RELATIONAL DATABASE MANAGEMENT

PGDCA 103

BLOCK 1:

INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

Dr. Babasaheb Ambedkar Open University
Ahmedabad

RELATIONAL DATABASE MANAGEMENT

kmro
KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Ms. Katherine Abraham

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

RELATIONAL DATABASE MANAGEMENT

Contents

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

UNIT 1 INTRODUCTION TO DATABASE SYSTEM

Basic Concepts of Data and Database, Database management system, Data Administrator and its functions, Advantages of DBMS over file systems, Applications of DBMS, Three-tier architecture of DBMS, Data Models, Components of DBMS, Overview of Languages of DBMS (DDL, DML, DCL)

UNIT 2 CONCEPTUAL MODELLING

Introduction, Basic Concept of Entity-Relationship Diagram, Relationship concept and its types (unary, binary and ternary), Cardinalities (one-one, one-many, many-one and many-many), Overview of Extended ER model, Case Study of Extended ER model

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

UNIT 1 DATABASE INTEGRITY CONCEPTS

Introduction, Domain Integrity constraints, Entity Integrity constraints, Referential Integrity constraints

UNIT 2 NORMALIZATION

Introduction, Basic Concept of designing, Need of normalization, Decomposition (lossy and lossless), Functional Dependency, Full Functional Dependency, Armstrong Axioms of Functional Dependencies, Normalization Rules (1NF, 2NF, 3NF and BCNF), Examples based on normalization

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

UNIT 1 RELATIONAL ALGEBRA

Introduction, Basic concepts of relational algebra, symbols of relational algebra, operations on relational algebra, Examples based on relational algebra

UNIT 2 QUERY LANGUAGE

Introduction, Introduction to Structured Query Language, Data Definition Language (DDL), Data Manipulation Statement (DML), Data types in MySQL, Operators (Arithmentic, Comparison, Logical), SQL Functions, Group function (AVG, MAX, MIN, SUM and COUNT), COMMIT and ROLLBACK

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

UNIT 1 DATA RETRIEVAL SQL STATEMENT

Introduction, Single table query without condition, Single table query with condition, Group by clause, Orders by Clause, Self Join Natural Join and Sub query, Examples based on SQL concepts

UNIT 2 TYPES OF DATABASE SYSTEMS

Introduction, Centralized Database System, Parallel Database Systems, Distributed Database Systems, Client-Server Database System

Dr. Babasaheb
Ambedkar
Open University

PGDCA 103

RELATIONAL DATABASE MANAGEMENT

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

UNIT 1
INTRODUCTION TO DATABASE SYSTEM 03

UNIT 2
CONCEPTUAL MODELLING 30

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

Block Introduction

Database is a collection of files or records in electronic form, which is accumulating in addition that, are investigated by computer. Database involves single or multiple computer files to programme data in a tremendously programmed format. Database Management System, is a software program that that allows the formation as well as management of databases. A database is a prearranged arrangement of data essentials designed for the trouble-free collection of information.

In this block, we will learn and study about basic of database concepts and idea about conceptualisation. We will cover the topics related to Database domain, model, architecture and languages. The student will be given with the knowledge about different rules and criteria's of ER model.

The block will focus on a basic understanding about DBMS languages and ER concepts that will give the knowledge about cardinalities. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Block Objective

After learning this block, you will be able to understand:

- Basic Concepts of Database
- Generalisation of Database management system
- Data Administrator and its functions
- About Three-tier architecture of DBMS
- Basic of Data Models
- Overview of Languages of DBMS (DDL, DML, DCL)

- Concept of Entity-Relationship Diagram
- About Relationship concepts and its types
- Idea about Cardinalities

Block Structure

Unit 1: Introduction to Database System

Unit 2: Conceptual Modelling

UNIT 1: INTRODUCTION TO DATABASE SYSTEM

Unit Structure

1.0 Learning Objectives

1.1 Introduction

1.2 Basic Concepts of Data and Database

1.3 Database Management System

1.4 Data Administrator and Its Functions

1.5 Advantages Of DBMS Over File Systems

1.6 Applications Of DBMS

1.7 Three-Tier Architecture Of DBMS

1.8 Data Models

1.9 Components Of DBMS

1.10 Overview Of Languages Of DBMS (DDL, DML, DCL)

1.11 Let Us Sum Up

1.12 Answers For Check Your Progress

1.13 Glossary

1.14 Assignment

1.15 Activities

1.16 Case Study

1.17 Further Readings

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About File compilation in Database.
- Knowledge about Database Understanding of Metadata and Data dictionary.
- Know how about DBMS.
- Basic of database scheme.
- Overview of Data Independence.

1.1 Introduction

Database is a compilation of files or records in electronic form, which is stored on furthermore can be investigate by computer. A database real meaning is involving one or multiple computer files to encode data in an extremely prearranged format. Database Management System (DBM), is a software program that allows the formation as well as management of databases. A database is a prearranged arrangement of data essentials designed for the trouble-free collection of information. Unlike a set of flat files, a database surrounds both data as well as structural information that are used in taking out data from the files where data lives in.

In actuality, the database systems are more or less works as Relational Database Management System (RDBMS) as they have unique smart features that can store the required data across multiple tables.

A database management system (DBMS) is the grouping of computer software programs used for producing, organizing, retrieving, analysing as well as sorting information in computer based databases. This type of software is informally called as database software.

1.2 Basic Concepts of Data and Database

A database is presently the name given to a collection of data. The data is prearranged in several etiquette so that the information enclosed inside the database can be easily recovered. Some of the easy databases that you know are things such as phone books or rolodexes. At the same time, as data processing has turned out to be more complicated, there are certain methods for collecting, storing as well as retrieving information. Databases have turned out to be the foundation stone for an irresistible amount of computing background in survival.

Databases over and over again fall into one of two broad categories. The initial category encompasses specific purpose and limited databases. In the academic world, these surround data to carry out a relatively incomplete role simply in a meticulous project. The database possibly will be planned to offer the researcher with an exacting set of data, however have no particular function or role at the ending of the project.

The next category encompasses general purpose, resource databases. A good illustration of a resource database is regional archaeological sites as well as monuments records (SMRs) that can be national monuments records. Such databases are not projected as precise. Resource databases more often than not challenge to be inclusive within their area of conversation that are maintained as well as updated, plus are prepared as per the interested parties. Such databases challenge to be inclusive in order to contain unexpected enquiries as well as research, but these cover a broad array of data which in turn needs a complex data structure, or technique of storing the information.

The CISP database is anticipated to be a source database moreover as a result has a complex data structure. Such data structure, on the other hand, provides huge power as well as flexibility equally for the retrieval as well as for the handling of the data, other than the future development of the database to contain other information in addition to materials.

Data

It is a peculiar form of information that are arranged or organised in special ways. Basically, every software carries data and programs. It is found that a program is a mixture of instructions that helps in carry out data. You can arrange data in numerous forms in shape of numeric or text and can be explained on paper by way of bits and bytes kept in memory.

Metadata

Metadata also a form of data dictionary which tells about a particular data. With the help of database it can be explained through program data independence. Metadata keeps the below mentioned information about every data element inside. These are:

- Name
- Type

- Range of values
- Source
- Access authorization

The above listed data an element applies with all related application programs so that if any alteration in data structure required, then affected programs will further be obtained. It will handle all database operation along with data integrity and correctness. Such type of data is mostly utilised by developers in order to construct programs, queries, controls and procedures so that the data can be controlled and carry out certain programs.

Data dictionary

The file system in database will describes basic arrangement of a database. So data dictionary carries certain list of related files which is present inside database. These will contain number of records with details about names and types of every field. It is seen that mostly some DBMS makes such files hidden from users so that it should not be damaged or destroyed. It is many times seen that data dictionary will not contain real data from database, but only enter such information in book so as to handle it. If the data dictionary is not present, then DBMS will not be able to work with data which is present inside database.

Check your progress 1

1. Database is:

a. Collection of files

b. Collection of records

c. Collection of information

d. All of above

2. Database Management System is:

a. Software Program

b. Collection of Record

c. Collection of Fields

d. None of above

3. Open Database Connectivity is:

a. Driver

b. Software

c. Fields

d. None of above

1.3 Database Management System

Database Management System or DBMS in diminutive refers to the know-how of storing as well as retrieving users' data with highest efficiency all along with suitable security method.

A DBMS creates it probable for end users to create, read, update as well as delete data in a database. The DBMS fundamentally provides an interface involving the database as well as end users or application programs, that guarantee that the data is again and again organized plus remains easily available.

The DBMS administers 3 important things:

- the data
- the database engine
- the database schema

These above 3 foundational elements help provide concurrency, security, data integrity as well as uniform administration measures. Characteristically, database administration responsibilities maintained by the DBMS consists of:

- change management
- performance monitoring/tuning
- backup and recovery

Many database management systems are in addition answerable for mechanical rollbacks, restarts along with recovery as well as the logging also auditing of activity.

The DBMS is for the most part useful for providing a federal view of data that can be admittance by numerous users, as of multiple locations, in a guarded behaviour. A DBMS can bind what data the end user will observe, as well as how to facilitate he end user viewing the data, as long as numerous views of a single database representation. End users as well as software programs are free from having to know where the data is physically situated and otherwise on what type of storage space media it resides inform the reason that the DBMS handles all requests.

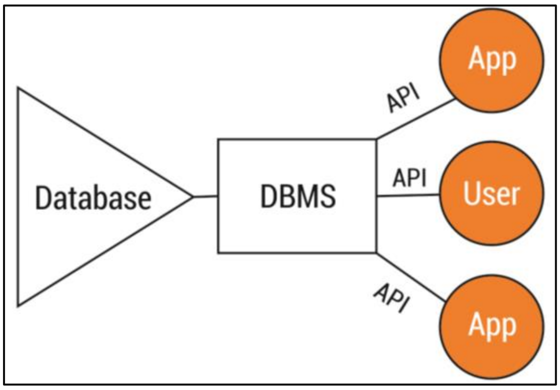


Fig 1.1 DBMS system

The DBMS can present equally logical as well as physical data independence. With this aim, it saves the users as well as applications from needing to be familiar with where data is stored regarding changes to the physical structure of data in terms of storage and hardware. As long as programs make use of the application programming interface (API) designed for the database with the aim to show by DBMS, developers that won't require modifications programs as changes have also been added to the database.

By means of relational DBMSs (RDBMSs), such API is SQL, which is a basic programming language that defines, protect as well as access data in a RDBMS. With DBMS, it is easy to store as well as manage data that comes with advantages, but also fixed cost. One of the major advantages of using a DBMS is that it gives permission to end users as well as application programmer's the right to use as well as allow the use of same data along with managing data integrity. The data has a superior protection plus maintained when it can be shared using a DBMS as a substitute for producing new iterations of the equivalent data stored in new files for each original application. The DBMS makes accessible a central store of data with the intention that it can accessible by multiple users in controlled approach.

A DBMS can in addition make available a lot of views of a particular database schema. A view describes what data the user sees in addition to how that user observes the data. The DBMS offers a level of concepts and wiched between the conceptual schema that defines the rational structure of the database as well as the physical schema that explain the files, indexes in addition to other physical mechanisms used by the database. As soon as a DBMS is used, systems can be customized a large amount further easily when business requirements change. New categories of data can be added to the database without disrupting the existing system and applications can be insulated from how data is structured and stored.

Database Schema

A database scheme is the structural arrangement to facilitate the rational view of the whole database. It illustrates how the data is predetermined and furthermore how the relation is stuck among them that are correlated. It positions mutually each and every restriction that is to be functional on the data. A database scheme portrays its entities as well as relationship that exist among all. Further, it controls a vivid detail of the database that is able to correspond by way of scheme illustration as shown in fig 1.2. It is the database designer who sketches the representation to help out programmers to be responsive of databases in accumulation to work constructively.

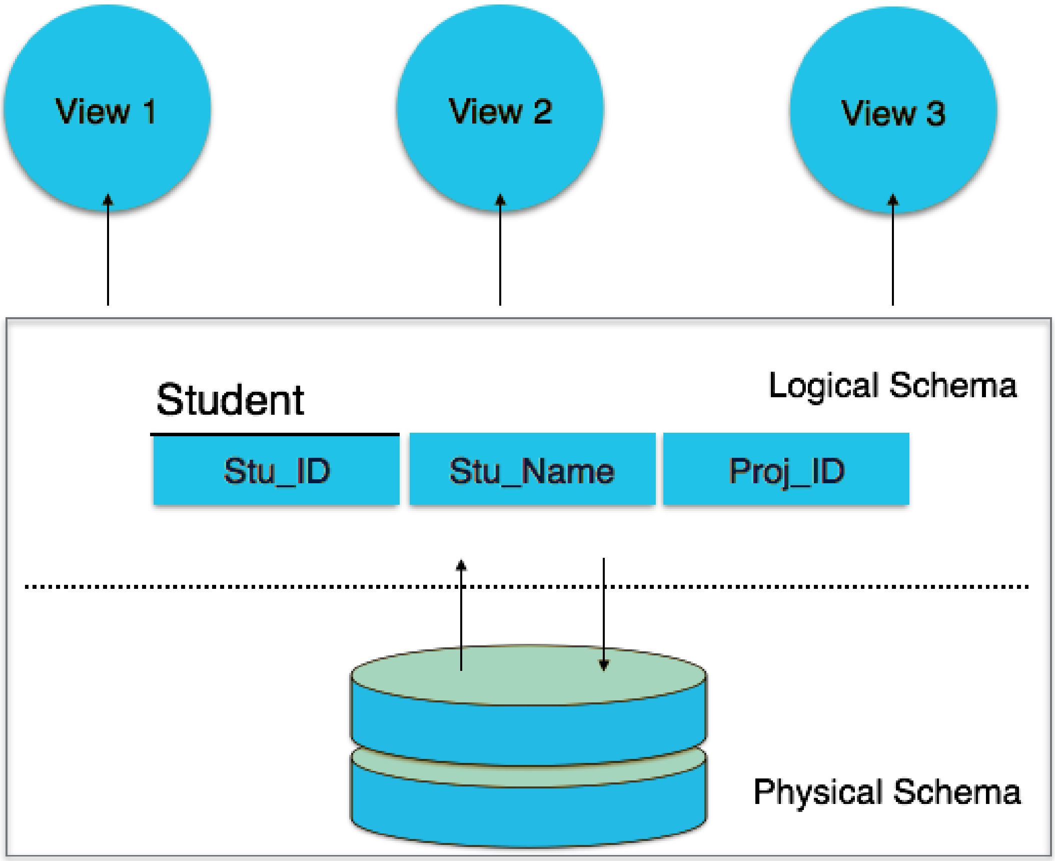


Fig 1.2 Database schema

A database schema can be separated roughly into two groups -

Physical Database Schema - Such type of schema relates to the definite storage space of data as well as forms of storage space like files, indices, etc. It explains how the data be determine with the purpose of storage in secondary part.

Logical Database Schema - Such type of scheme illustrates search and every rational constraint with the intent of pre-requisites to be practical on the data stored. It shows tables, views, as well as integrity limitations.

Database Instance

It is imperative with the idea of differentiation that occurs with two terms separately. Database representation is the structure of database. It is designed rapidly as if the database doesn't survive at all. It is observed that as soon as the database is carried out, after that it turns out to be hard to prepare any alteration in it. A database representation does not manage every data or information.

A database graphic is a situation of a performing database with data at numerous prearranged times. It contains a snap of the database. Database incidents have a predisposition to modify by way of time. A DBMS formulates crystal-clear as a result that every incidence is stated to be in applicable situation, by consistently sub sequenting each and every validation, constraints, as well as situation to facilitate the database designers to work.

Data Independence

A database system in general controls a bunch of data in accumulation to users' data. For illustration, it stores data in relation to data, recognized as metadata, to locate as well as to get back data without difficulty. It is somewhat tricky to change or bring up to date a position of metadata as it stores it in the database. Although since a DBMS is spread out, it desires to transform over time to please the needs of the users. If the whole data is dependent, it would roll out to be multifaceted yet boring.

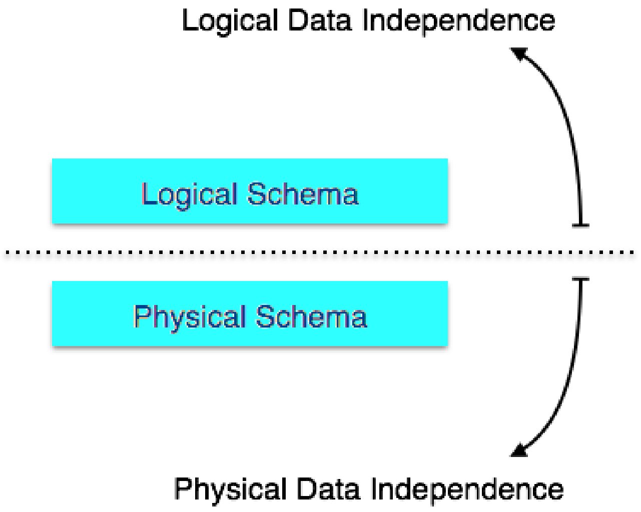


Fig 1.3 Data Independence

Metadata itself go subsequent to a layered structural design, as a result when we amend data at a particular layer, it does not have an outcome on the data at supplementary stage. This data is self-determining, but never the less it is mapped to each one.

Logical Data Independence

Logical data independence relates to a type database which shows details about how the data will be handled or arranged inside the system. If you happen to analyse a relation that was created in database by considering several constraints and at the same time practice on such relation, then you will find that it is direct which shows particular data storage on a disk.

If we do several amendments on table arrangement, it should not revolutionize the data which is available in the disk.

Physical Data Independence

Each and every representation in Physical Data Independence is rational; furthermore the real data is accumulated in a bit that is created on the disk. Such type of data independence is authorized to mutate the physical data so as not to cover the collision of representation or logical data.

For instance, in a situation, if we hope to transform or improve the storage system itself, we have to revolutionize the hard-disks by means of SSD, further which should not give several contacts on logical data or diagram.

Check your progress 2

1. The database schema is written in

   a. HLL

   b. DML

   c. DDL

   d. DCL

2. A logical schema

   a. is the entire database.

   b. is a standard way of organizing information into accessible parts.

   c. describes how data is actually stored on disk.

   d. both a. and c.

3. One of the advantage of database management is

   a. data depends on programs.

   b. data dismissal increases.

   c. data is incorporated in addition to admittance by multiple programs.

   d. none of the above.

1.4 Data Administrator and Its Functions

A data administration also recognized as database administration manager, data architect as well as information centre manager, is a far above the ground level function which is an in charge for on the whole management of data assets in an group. In order to carry out its duties, the data administrator should recognize a high-quality agreement of system analysis as well as programming. These are the functions of a data administrator:

1. Ideate about data policies, process and standards.
2. Development of organization's IT view, project model, cost/profit model, recommendations of database impression with supervision map.
3. Data divergence motion.
4. Data investigation which covers the aim as well as description of model data needs, industry rules, functioning needs and saving commercial data glossary.
5. Inside promotional impression.
6. Administering the data storage area.

Check your progress 3

1. __is a person that is responsible for data, metadata, and policies related to data usage.

a. Data administrator	c. Database steward
b. Database administrator	d. Both a and b

1.5 Advantages of DBMS Over File Systems

A database is designed for data storage space as well as data recovery. As soon as you place data into a database table, the database table is indexed in different ways. As a result of that it depends on a number of columns you searched by which data at all times survive as quickly as possible.

Individually, the database does such activity by loading as much data into RAM as feasible also called as buffer cache. This cache seizes all the data that has been newly used inside the database; hence it can be established faster than by looking for it on the disk. As the data in the cache turns out to be idle, it is noticeable to be glowing so that if new data desires to be loaded into cache the database can create room for it.

Database servers in addition hold up numerous people reading as well as writing the data at the similar time. Databases agree for locks to be engaged on precise rows, data pages, or entire tables depending on the need at the time, as a result of this, one person can perform a transaction beside the database where as other users can’t update those records, pages, or tables until the initial transaction has finished.

The database approach offers a number of potential advantages as compared to traditional file processing systems.

1. **Program-Data Independence -** The parting of data descriptions on or after the application programs uses is referred as Data independence. By means of such database approach, data descriptions are accumulated in a central location referred to as repository. Such quality of database systems permit an organization’s data to alter with no altering the application programs so as to process the data.

2. **Data Redundancy and Inconsistency -** During File-processing, files having dissimilar formats as well as application programs might generate by changed programmers. Likewise dissimilar programs could be on paper in numerous programming languages. The identical information positioned at dissimilar files cause laying-off as well as inconsistency and thus higher storage as well as admittance costs.

3. **Difficulty in accessing Data -** In conventional file arrangement, the data or information is collected and stored in various files. It is easy storing data in files, as you can take the data back any time where there is a need on

accepting the prior approval and on request of application program. This is a tire some procedure.

4. **Data isolation -** On account of data there is a wide complexity of files, which are conceivably in asymmetric formats. It is adverse to formulate new application programs to accumulate the appropriate data.

5. **Combined access -** There continues no central administration of declaration in conventional file organization. As an effect, the concurrent admittance of data proximate many users to use it.

6. **Safety Problems -** In reality there continues no main administration of data in conventional file arrangement wherefore, safety, compulsion is not alleviated in File-processing approximation.

7. **Integrity Problem -** The declaration values acquired in the database should satisfy certain categories of consistency constraints. For prototype, the ledger of bank assets feasibly will never fall down lower than an affirmed charge. These constraints are essential in the approach nearby accumulating together appropriate code in the distinct application programs. On the other hand, when authentic constraints are accumulated, it endures to alter the programs to set up into sequence. The difficulty is combined when prohibitions approach numerous data elements from distinct files.

8. **Advanced Data assigning -** A database is arranged as an assorted corporate resource. Legal inward as well as exterior users are categorized with approval to develop application of the database. Additionally, every user is allowed with one or additional user observation to create potential use. A consumer witness is a feasible reputation of numerous divisions of the database that is essential for an employer.

9. **Improved Creativeness of Application Progress -** One of the most important benefits of database is to move in line as it is an important aspect which considerably lowers the amount as well as time of establishing new business submissions.

Conveniently there will be two important points where data base gathering can frequently be urbanized into a lot supplementary fast as compared to conventional file applications.

a) Contemplating that the database as well as the associated data contains support applications subsequently planned and exercised the programmer can essence on the catalogued functions compulsory for the fresh application, with no worry about file arrangement or low-level facilitation facts.

b) The data base management approximation approach is an allotment of high-level productivity gadgets like forms plus logs generators along with high-level dialects to facilitate computerize operations of the database concept as well as facilitation.

Check your progress 4

1. Which among the following is an advantage of database management?

a. In this, data is dependent on programs.

b. In this, data redundancy increases.

c. In this, data is integrated and can be accessed by multiple programs.

d. none of the above.

1.6 Applications of DBMS

Database management systems is applicable on developing and forming a database, pile it up with data or information and forms different ways to reserve all along by way of modifying the particular information not including the technological aspects of data storage as well as reclamation. Apart from this, some of the characteristics of database management systems consist of:

- User admission as well as security administration systems presenting appropriate data contact to numerous users at the same time protecting perceptive data.
- Data encouragement to formulate definite and consistent accessibility of data.
- The access logs makes smoother for database manager to justify the use of database that is being applied.
- The rules enforcement will gives surety about prescribed data that is kept in every field. This is stated as date fields which are adjusted for particular data having dates in specific range.

- In DBMS, formulae like counting, averaging as well as summing up makes arithmetical examination of simple data.
- Performance monitoring as well as optimization tools that shows permit to users to adjust database settings that relates to speed along with efficiency.

Numerous web applications depends on DBMS that will vary from search engines as well as article, directories to social networks network like Facebook along with Twitter.

Check your progress 5

1. The purpose of DBMS is to:

a. Keep record	c. Display record
b. Maintain record	d. all

1.7 Three-tier architecture of DBMS

The plan of a DBMS applies only to its architecture. This could be centralized or decentralized as well as hierarchical. It is found that the DBMS architecture is single tier or multi-tier. With N-tier architecture, it is seen that the DBMS is divided into complete system that relates to independent N modules, which is independently altered, changed or further replaced.

In single tier DBMS architecture, with single entity, the user can simply sit on database as well as can use such database. If the user makes any alteration, then this will simply alter the DBMS directly. For this, it will not carry any handy tools to be used by end users. The creator or designers of database will simply wish to apply single-tier architecture.

In case of 2-tier DBMS, the application will pass through DBMS in order to access straight. Programmers if applying 2-tier architecture will simply access DBMS through its application. We see that the application in 2 tier DBMS is a wholly self-governing of database in relation to process, plan and indoctrination.

3-tier Architecture

In case of 3-tier DBMS architecture, the difference among other tier arrangements is different from each other. This applies on the convolution of users by means of opinion as to how they apply data that is available in the database.

The 3 tier architecture is for the most part used architecture used in forming DBMS. Fig 1.4 shows this arrangement.

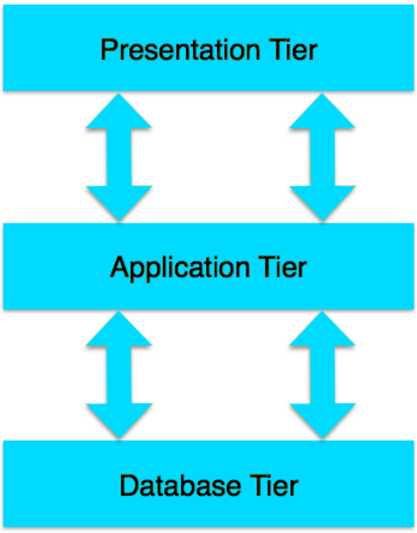


Fig 1.4 3-tier architecture

- **Database (Data) Tier -** In this arrangement, the database is available all along with its query processing languages.

- **Application (Middle) Tier -** In this type of arrangement, the database is available in application servers with programs to work with the database. This application is good for user as it explains a conceptual analysis of database. End-users are ignorant of some continuation of database away from the application.

- **User (Presentation) Tier -** In this type of arrangement, we see that the operator uses the database without having any knowledge about existence of database further than this layer. In this many views of database are shown with the help of applications.

- **Multiple-tier -** In this type of database architecture, you can modify or edit, as roughly each and every component is autonomous. Furthermore, it can be transformed separately.

Check your progress 6

1. Presentation tier is used by the:

a. Programmers

c. Software people

b. End users

d. All

1.8 Data Models

A Database model defines the logical design of data. The data model is an actual description of analysing real world objects along with several measures in relation to description. It is a concept which facilitates on required, intrinsic characteristic of an association as well as overlooks the unplanned properties. A data model approximates the arrangement itself. It should assign the foundational conceptions as well as emblems that consideration allocates database designers as well as end users unambiguously additionally correctly to broadcast their comprehension of the organizational data.

Data Model can be described as an accomplished coalition of considerations for accounting as well as skimming data, connections between data, as well as confinements on the data in an organization. A data model encompasses of three elements:

- A compositional featured, consisting relevantly a set of controls according to which databases can be developed.
- A manipulative measure, describing the categories of performance that are assigned on the data.
- Feasibly a set of collectivity rules, which promises that the data is correct.

The approach of a data model continues to approximate data also to construct the data perspicuous. There acquire been ample data models endeavoured in the composition. The model explicates the connections between contradictory elements of the data. In the history of database design, three models have been in use.

- Hierarchical Model
- Network Model
- Relational Model
- Entity-Relationship Model

The hierarchical model

The data is arranged in a hierarchy with the help of downward tree. In such models, pointers are applied to steer among stored data. Such model was initially a DBMS model.

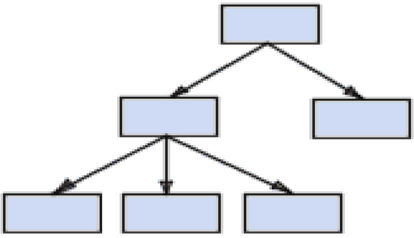


Fig 1.5 hierarchical structure

We see that in such type of model, every entity contains a single parent having many children. As seen, the top of hierarchy carries single entity known as root.

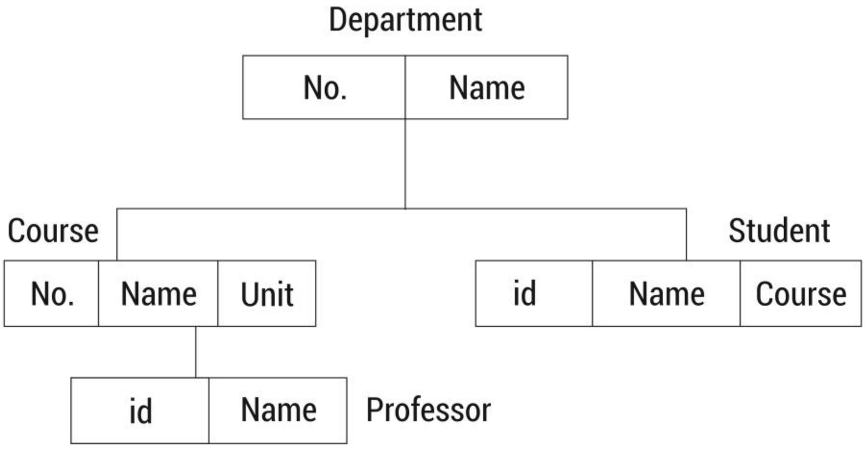


Fig 1.6 hierarchical model

Network model

Just like the hierarchical model, the network model also utilizes pointers in the way as the data is kept. On the other hand, it does not automatically make use of the downward tree structure.

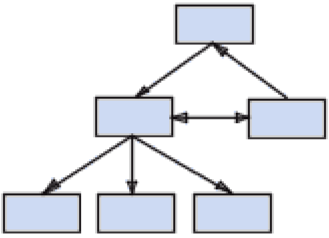


Fig 1.7 network structure

In the network model, entities are organised in a graph, in which some entities can be accessed through several path

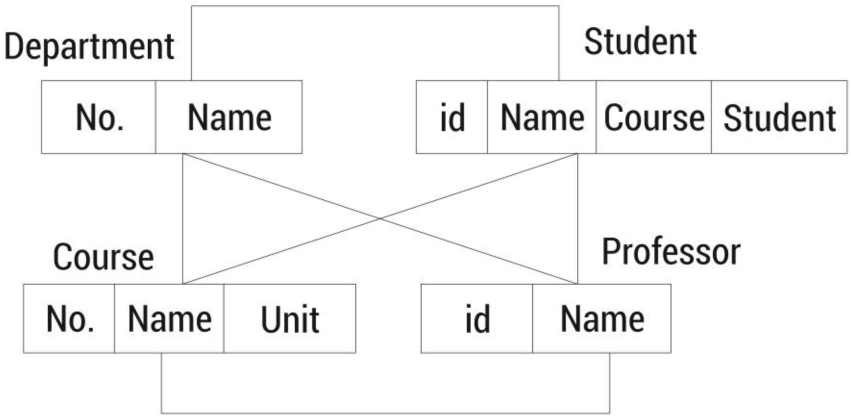


Fig 1.8 network model

Relational model

In this model, data is kept in 2 dimensional tables having rows as well as columns. In this, the data is calculated depending upon the relational theory of mathematics.

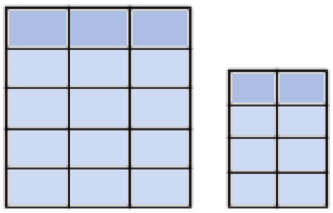


Fig 1.9 relational model

Further it is seen that, in such model, the data is gathered in 2-dimensional tables as shown above known as relations. It is observed that these tables or relations somewhat relate to each other. This arrangement is quiet famous as it is more scientific model as compared to rest. It is based on first-order predicate logic and defines a table as an n-ary relation.

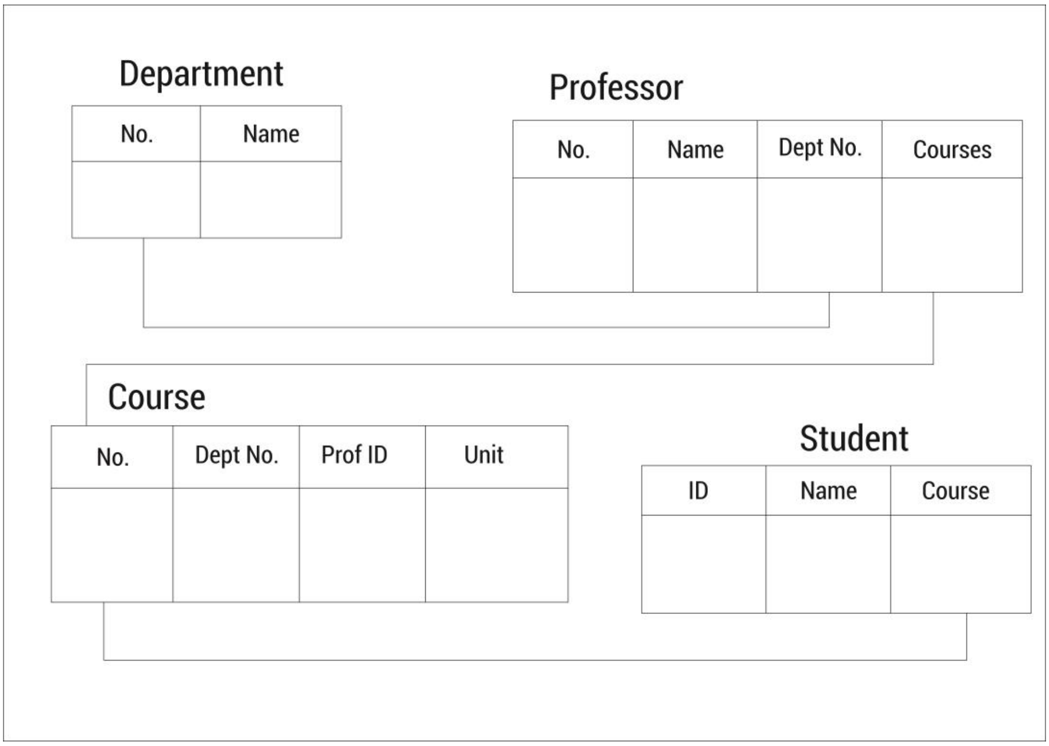


Fig 1.10 relational model

Entity-Relationship Model

Entity-Relationship (ER) Model is a type of database model structure that is based on conception of real world things as well as relations surrounded by them. At the same time, the preparation of the actual world situation hooked on the database model, ER Model generates entity set, relationship set, common characteristic as well as constraints. It is the most excellent model applied for theoretical plan of a database. It is based on –

- Entities and their attributes.
- Relationships among entities.

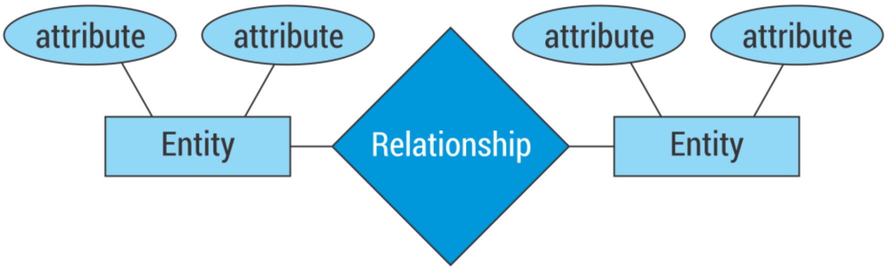


Fig 1.11 ER model

Entity

We can say that the unit of ER Model is an actual entity example that carries properties called attributes. Every characteristic is a definite by set of principles called domain. For instance, while considering school database, a student is considered as an entity having various characteristics such as name, age, class, etc.

Relationship

The rational relationship amongst entities is called correlation. Relationships are mapped by means of entities portraying different behaviour. Mapping cardinalities characterize the extent of relationship among two entities. In this, the mapping cardinalities are:

- one to one
- one to many
- many to one
- many to many

Check your progress 7

1. Hierarchical model is also called

a. Tree structure

b. Flex Structure

c. Normalize Structure

d. Table Structure

2. Which of the following is record based logical model?

a. Network Model

b. Object oriented model

c. E-R Model

d. None of these

1.9 Components of DBMS

Databases are composed of related tables, which in turn are composed of fields and records.

Field

Every record contains specific fields to store data. The fiel contains user information such as name, address, city, phone numbers etc. Normally the fields are defined by:

Field name

Data type

- Character: Details about customer telephone numbers and pin codes
- Numeric: it carries numbers that can be calculated using certain mathematical operators
- Date: it will show numeric dates which can be calculated mathematically.
- Logical: It state True or False or can also stress on Yes or No

Field size

- It shows the capacity of data storage

Record

It is a data in shape of values which is for related fields that is for single entity. It refers to particular person, product, company etc.

Table

It shows details of particular records. It shows the examples as employee table, product table, customer, and orders tables.

The table contains rows and column, so the records can be seen in terms of row and fields can be seen in terms of columns. We see that database is a mixture of rows and columns.

Relationships

In the table, the relationship exists in three types:

- One-to-One
- One-to-Many

- Many-to-Many

Among the above, the famous relationship in relational database is One-to-Many and Many-to-Many.

To see for One-to-Many relationship, we see that the presence of Customer table and Orders table having single customer that can do multiple orders.

In case of One-to-Many relationships, we find that it contains two tables, “one” table and “many” table.

Normally, it is found that an example of Many-to-Many relationship relates to Orders table and Products table.

If we see Many-to-Many relationship, we find that it will have three tables: two “one” tables and one “one” table.

Key Fields

Two fields are related to each other if they have common fields. It is seen that a key field in “one” table of One-to- Many relationship has to the main key. If we see for the similar key in “many” table of One-to-Many relationship such type of key will narrated as foreign key.

Check your progress 8

1. Database is composed of:

a. Table

b. Records

c. Fields

d. All of above

2. Data type consists of:

a. Character

b. Numeric

c. Date

d. Logic

e. All of above

3. Units of data within a database are generally called:

a. Driver

b. Records

c. Fields

d. None of above

1.10 Overview of Languages of DBMS (DDL, DML, DCL)

A DBMS is a software pack that does various tasks which will cover provision of facilities to allow the user to contact as well as change information in the database. The database is an intermediate link between the physical database, computer and the operating system and the users. To provide the various facilities to different types of users, a DBMS normally provides one or more spe-cialized programming languages called database languages.

Database languages come in different forms: -

1. Data Description Language (DDL)
2. Data Manipulation Language (DML)
3. Data Control Language (DCL)

DML

DML is abbreviation of Data Manipulation Language. It is used to retrieve, store, modify, delete, insert and update data in database. Some examples:

- SELECT - recover data from database
- INSERT - introduce data in table
- UPDATE - bring up to date the present data in a table
- DELETE - erase all account from a table
- MERGE - insert or update operation
- CALL - bring PL/SQL or Java subprogram
- EXPLAIN PLAN - give details about access path to data
- LOCK TABLE - manage concurrency

DDL

DDL is short form of Data Definition Language. It is applied to produce as well as change the arrangement of database substance in database. Some examples:

- CREATE - produce objects in the database
- ALTER - convert the arrangement of database
- DROP - obliterate materials from database
- TRUNCATE - acquire away each as well as every descriptions from table
- COMMENT - accumulate acknowledgment to data documentation
- RENAME - give another identity to the material

DCL

DCL continues compaction of Data Control Language. It is exercised to develop reverences, approvals, as well as referential completeness as well it is utilized to administer approach to database by guarding it. Some examples:

- GRANT - gives user's access privileges to database
- REVOKE - withdraw access privileges given with the GRANT command

Check your progress 9

1. DDL is:

a. Dynamic Data Language

b. Detailed Data Language

c. Data Definition Language

d. Data Derivation Language

1.11 Let Us Sum Up

A database involves one or multiple computer files to encode data in an extremely prearranged format. A Database Management System is a software program that allows the formation as well as management of databases. Database is a compilation of files or else records in electronic form, which can be searched by computer. DBMS is a type of software programming that is used to arrange and frame all information and data in a database.

It serves as a mixture of different software programs that are used to arrange, sort, set and organise computer-based databases. Open Database Connectivity (ODBC) is a driver that allows the database to integrate with other databases. Some of the common relational database management systems are

Microsoft Access, File maker, Microsoft SQL Server, MySQL, Oracle etc. Databases are composed of related tables, which in turn are composed of fields and records.

We see that a field is specific area that is present inside the record which will keep the particular data. A record is the collection of values for all the fields pertaining to one entity. A table is a collection of related records. Units of data within a database are generally called records.

Further it is seen that a DBMS is a software package which performs many tasks which will cover provision of facilities to allow the user to contact as well as can alter any information present inside database.

1.12 Answers for Check Your Progress

Check your progress 1

Answers: (1-d), (2-a), (3-a)

Check your progress 2

Answers: (1-c), (2-a), (3-c)

Check your progress 3

Answers: (1-a)

Check your progress 4

Answers: (1-c)

Check your progress 5

Answers: (1-d)

Check your progress 6

Answers: (1-d)

Check your progress 7

Answers: (1-a), (2-a)

Check your progress 8

Answers: (1-d), (2-e), (3-b)

Check your progress 9

Answers: (1-c)

1.13 Glossary

1. **Hardware** - It contains computer components.
2. **Software** - It is a program that contains operating system, utilities, files and applications programs from data stored in files.
3. **Procedures** - The instructions and rules that govern design and use of software component.
4. **Data** - The collection of facts.

1.14 Assignment

Explain the Components of DBMS?

1.15 Activities

Write features of Data Administrator and its functions?

1.16 Case Study

Compare DDL, DML and DCL?

1.17 Further Readings

1. Asher, H.B. (1984). Causal modelling. Sage University Paper series on quantitative applications in the social sciences, 07-003. Newbury Park, CA: Sage Publications.

2. Creswell, J.W. (1994). Research design: Qualitative and quantitative approaches. Thousand Oaks, CA: Sage Publications.

3. Kerlinger, F.N. (1979). Behavioural research: A conceptual approach. New York: Holt, Rinehart & Winston.

UNIT 2: CONCEPTUAL MODELLING

Unit Structure

2.0 Learning Objectives

2.1 Introduction

2.2 Basic concept of Entity-Relationship Diagram

2.2.1 Symbols of entities

2.2.2 Types of entities

2.2.3 Types of attributes.

2.3 Relationship concepts and its types

2.4 Cardinalities

2.5 Overview of Extended ER model

2.6 Case Study of Extended ER model

2.7 Let Us Sum Up

2.8 Answers for Check Your Progress

2.9 Glossary

2.10 Assignment

2.11 Activities

2.12 Case Study

2.13 Further Readings

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- The concept of Conceptual Modelling
- Knowledge about Entity-Relationship Diagram
- Basic of different types of entities Idea about Single and Multi value attribute

2.1 Introduction

The conceptual modelling is separated into three decades that is 1970s, 1980s as well as 1990s. In 1970s the database design was extremely significant as per Peter Chen's paper "Entity-Relationship Model that is Unified View of Data." This is highlighted in the area of data modelling as well as database design. Readily available discussions endeavour to expand high level data definition languages designed for defining conceptual schemas that can be Conceptual Schema Language (CSL). In 1980s, there sprang up a number of approaches to expand Chen's Entity Relationship Model. In fact information systems as well as the design of IS were considered attractive subjects. Next, in the start of 1990s, there were many queries such as schema integration, schema transformation as well as quality measures meant for conceptual schemas in region of database design. However this moment in time is also predisposed by object-oriented modelling methods in addition to languages in software engineering.

A conceptual model is a model prepared of the work of art of concepts, which are old to help people be acquainted with understanding or reproduce a subject that shows the model. A number of models comprising of physical objects that facilitate it together by making it work like the object it represents. A conceptual model recognizes the elevated relationships involving the unlike entities. Features of conceptual data model are:

- Vital entities as well as relations which is present.
- Having no particular attribute.
- Contains no main key.

The figure 2.1 is an example of a conceptual model.

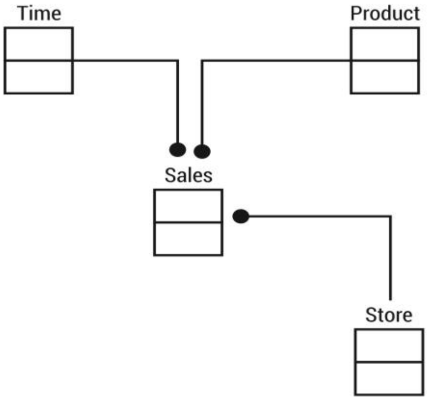


Fig 2.1 conceptual model

In fig 2.1 it is seen that the only information shown is by means of the conceptual modelling entities that explains the data as well as the relationships flanked by individual entities. No other information is exposed through the conceptual model.

2.2 Basic Concept of Entity-Relationship Diagram

An entity-relationship diagram, or ERD, is a type of chart which optically signifies the correlation involving database entities. The chart is an organizational data storage needs that carries three major mechanisms:

- Entities
- Attributes
- Relationships

This model chart will describe about the hypothetical dream of a database. This type of chart model will be comfortable in the region of real-world entities in addition to relationships surrounded by them. At the review stage, ER model is well thought-out as good alternative for designing databases.

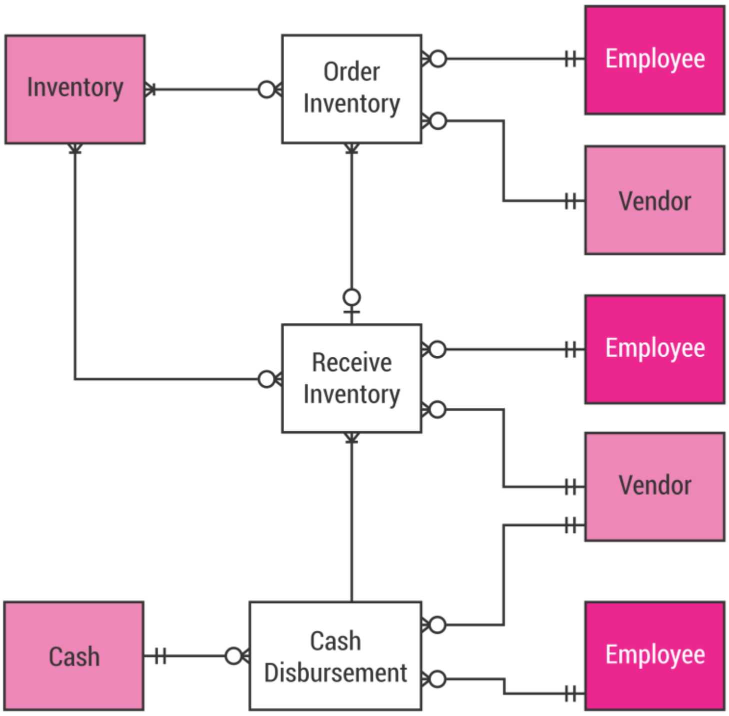


Fig 2.2 ERD model example

Entity-Relationship model works on the phenomena of actual-world entities as well as relationships that appear among them. At similar instance, preparing such actual-world condition in database models will lead to formation of entity set, relationship set, general attributes as well as constraints. It was established for theoretical design of database which relies on:

- Entities and their attributes.
- Relationships among entities.

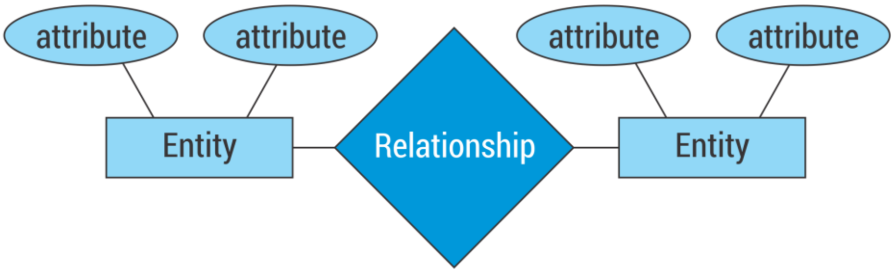


Fig 2.3 ER model

2.2.1 Symbols of entities

It seems that an entity is an actual-world entity that can be either animate or inanimate, but is identifiable. For instance, if we consider a school database, we see that students, teachers, classes as well as courses serves as entities. It is seen that every such entities carries various attributes or properties that give them their identity.

Moreover, it is found that an entity set is a mixture of similar types of entities. An entity set may perhaps hold entities with the characteristic sharing comparable values. For illustration, a set of students may comprise of each and every students of a school, similarly a set of Teachers will contain carryall teachers of school with all faculties. The Entity sets need not be disjoint.

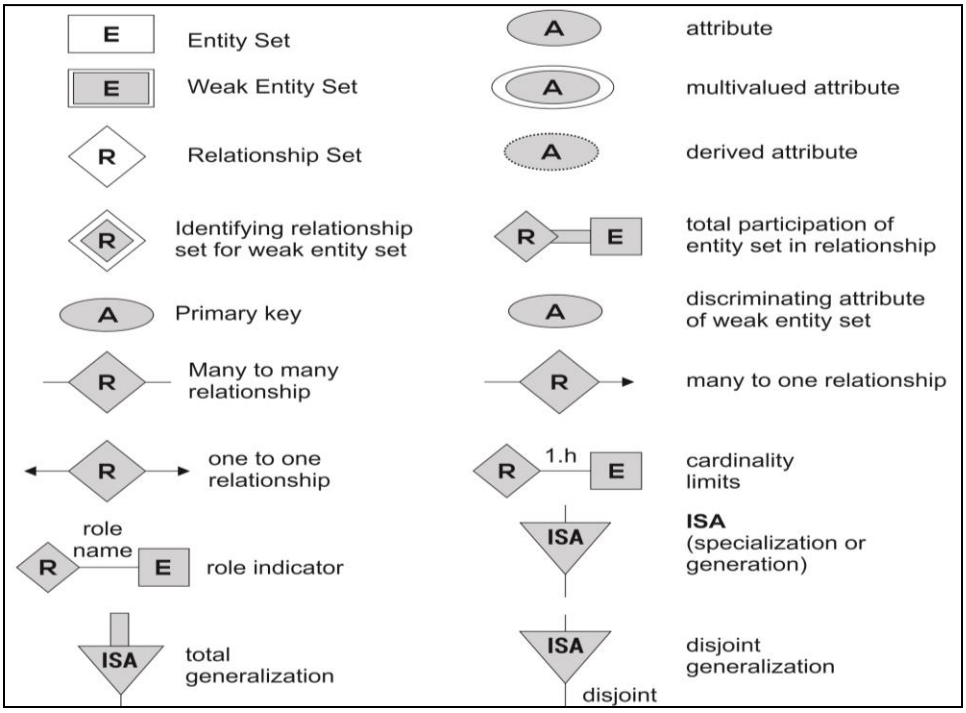


Fig 2.4 ER model symbol

2.2.2 Types of entities

Entity Relationship (ER) model carries various types of entities.

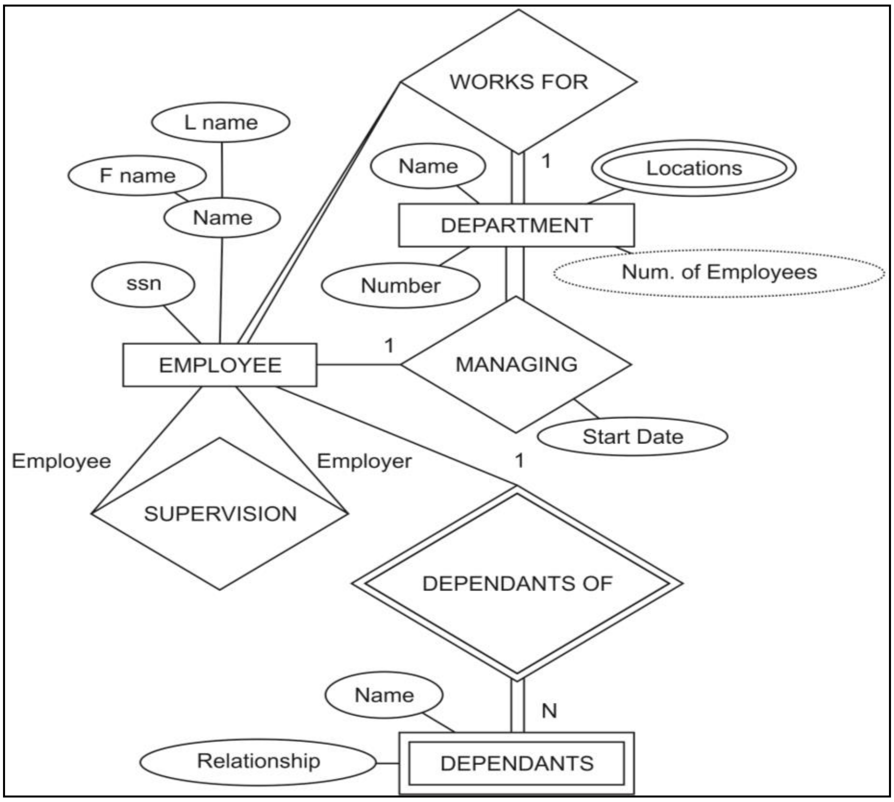


Fig 2.5 Entity in ER model

The continuation of an entity possibly will depend on the continuation of one or more previous entities, like existence dependent. It is that an entity whose continuation is not bound on every previous entity is called as not existence dependent entity. Entities which are marked as per their characteristics are:

- Strong Entities
- Weak Entities

Strong Entity and Weak Entity

The entity set which does not encompass enough attributes to figure a primary key is termed as Weak entity set. The entity set which carries a primary key is named as Strong entity set.

It is seen that a weak entity is reality dependent. It has a survival of a weak entity that depends on the continuation of a classifying entity set. In this the partial key is applied to make out previous attributes of a weak entity set. The primary key of a weak entity set is created by primary key of recognizing entity set as well as the discriminator of weak entity set. It is studied that the presence of weak entity is shown by double rectangle in the ER diagram as shown in fig 2.6.

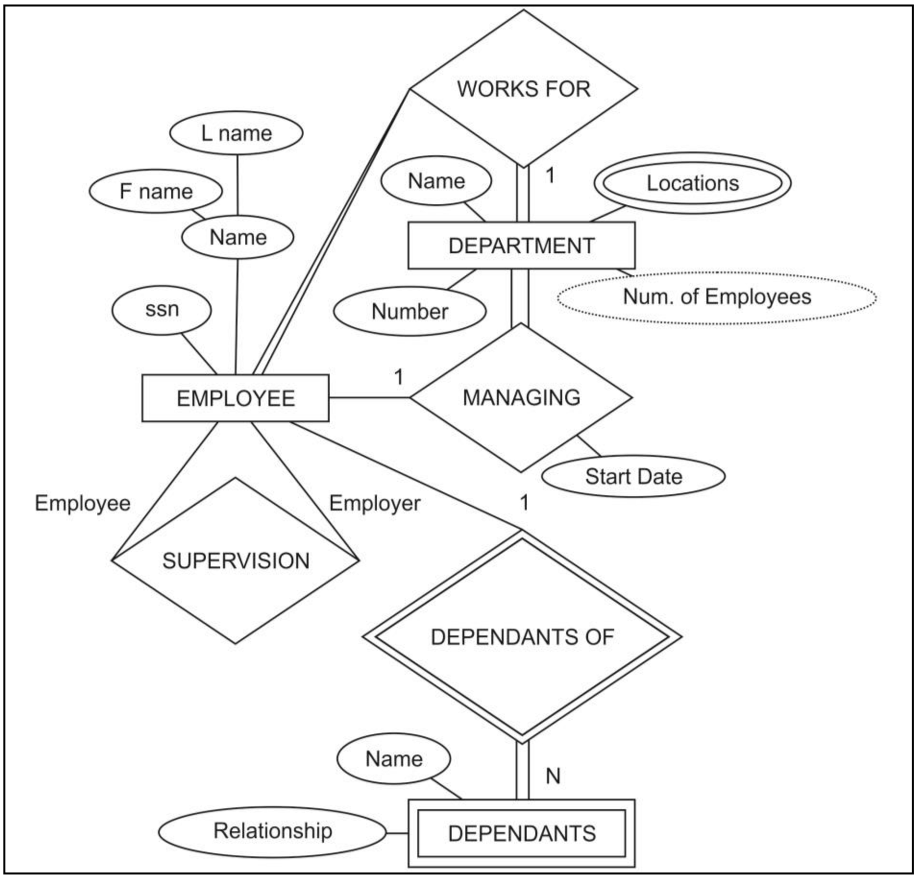


Fig 2.6 Entity in ER model

From the diagram 2.6, it is clear that the underlined is the discriminator of weak entity set that carries dashed line.

Further it is analysed that the relationship that exists among weak entity as well as strong entity set is termed as Identifying Relationship. For this consider diagram 1.8, which is related to a loan payment plan that identifies the relationship for payment entity. In this, a weak entity set is characterized by double outlined box in addition to equivalent categorized relation by a double outlined diamond shaped symbol.

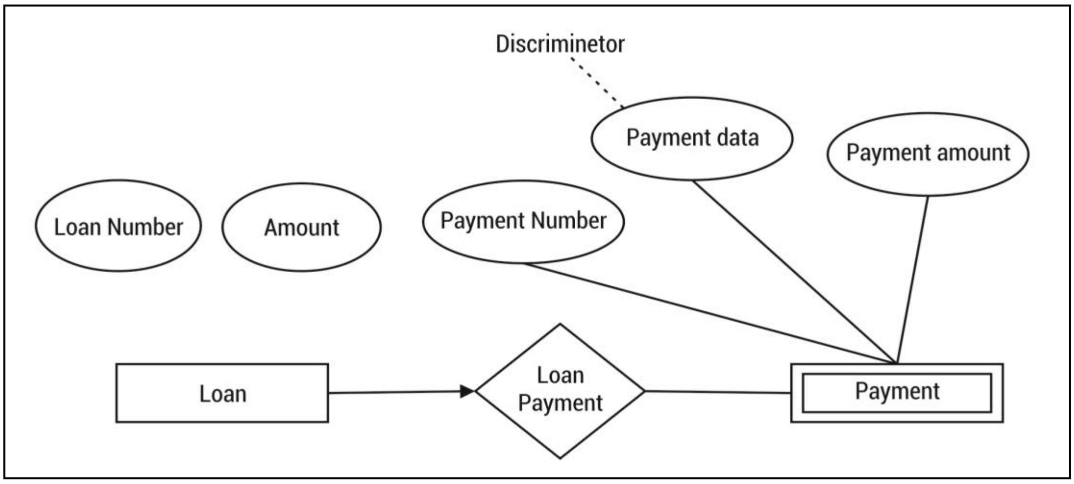


Fig 2.7 Example

In fig 2.7, we see that the double lines shows the complete participation of weak entity in strong entity place, which says that each payment be required to be connected by means of loan payment to various accounts. The arrow beginning from loan-payment in the direction of loan indicates shows the presence of single payment for each loan. In this, the discriminator of weak entity bunch is underlined with the help of dashed lines instead of bold line. The distinction between weak and strong entity is shown:

Strong Entity Set	Weak Entity Set
It has its own primary Key	It does not save sufficient attributes to form a primary key on its own
It is represented by a rectangle	It is represented by a double rectangle

It contains a primary key represented by an underline	It contains a Partial Key or discriminator represented by a dashed underline
The member of strong entity set is called as dominant entity set	The member of weak entity set is called as subordinate entity set
The Primary Key is one of its attributes which uniquely Identifies its member	The Primary Key of weak entity set is a combination of partial Key and Primary Key of the strong entity set
The relationship between two strong entity set is represent by a diamond symbol	The relationship between one strong and a weak entity set is represented by a double diamond sign it is known as identifying relationship
The line connecting strong entity set with the relationship is single	The line connecting weak entity set with the identifying relationship is double
Total participation in the relationship may or may not exist	Total Participation in the identifying relationship always exists

2.2.3 Types of attributes

Entities are characterized by way of their properties that are known as attributes. Every attributes contains certain values.

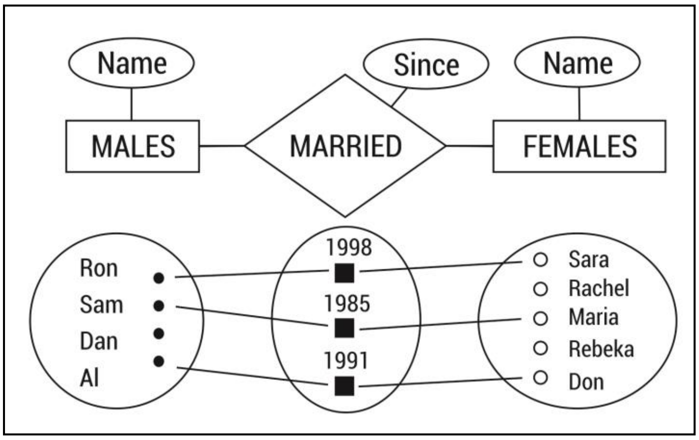


Fig 2.8 Arrangement of Attribute

From the figure, we found that a student entity having name, class as well as age serves as attributes. In such situation, there exists a domain or range of values to facilitate the attributes. In such case, the student's name cannot be a numeric value, but have to have an alphabetic; also the student's age should not be a negative number.

There are certain types of Attributes that exists in ER model:

Simple attribute: Simple attributes contain atomic values that cannot be separated more. In an example, a student's mobile number is an atomic value of 10 digits.

Composite attribute: Composite attributes is another type of attribute that are prepared from more than one simple attribute. In an example, if a student is writing his name, then the first name and last name constitute a composite attribute.

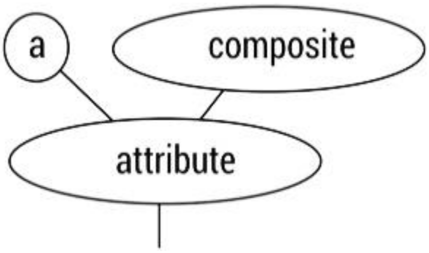


Fig 2.9 Composite attribute

Derived attribute: These are the attributes which do not present in physical database; however their values are copied from different attributes that are available in the database. In an example, an average salary in an organisation should not be kept directly in database, as an alternative, it should be derived.

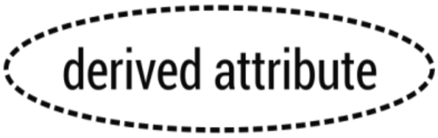


Fig 2.10 Derived attribute

Single-value attribute: The Single value attributes carries only one value. It can be a Social Security Number.

Multi-value attribute: Multi-value attribute is a type of attribute that carries more than one value. It can be a person that has more than one mobile number, email address, etc.

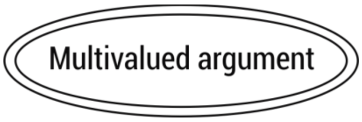


Fig 2.11 Multi value attributes

Such type of attribute can be:

- simple single-valued attributes
- simple multi-valued attributes
- composite single-valued attributes
- composite multi-valued attributes

Check your progress 1

1. Which of the following are the properties of entities?

   a. Groups

   b. Table

   c. Attributes

   d. Switchboards

2. E-R model uses this symbol to represent weak entity set?

   a. Dotted rectangle.

   b. Diamond

   c. Doubly outlined rectangle

d. None of these

3. Conceptual design

a. Is a documentation technique.

b. Needs data volume and processing frequencies to determine the size of the database.

c. Involves modelling independent of the DBMS.

d. Is designing the relational model.

2.3 Relationship concepts and its types

The level or degree of relationship is basically the number of entity types which takes part in the relationship. Since from the ER model, there are three on the whole common relationships that occurring ER modelling:

- Binary
- Unary
- Ternary

Binary Relationship

This type of relationship occurs when two entities take part shows the familiar relationship degree. For Example:

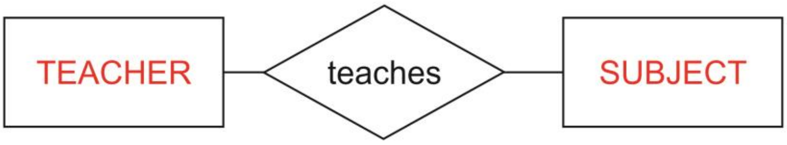


Fig 2.12 Binary relationship

Unary Relationship

This type of relationship exists when both participants in relationship behaves in similar manner. Consider fig 2.13.

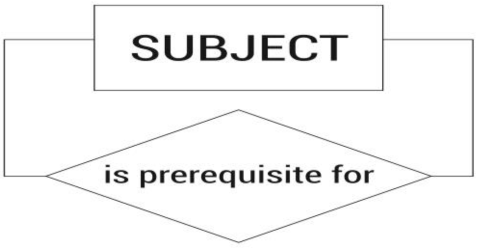


Fig 2.13 Unary relationship

In this, the subject gets prerequisites for different subjects.

Ternary Relationship

This type of relationship appears when three entities take part in a common relationship. Consider example in fig 2.14.

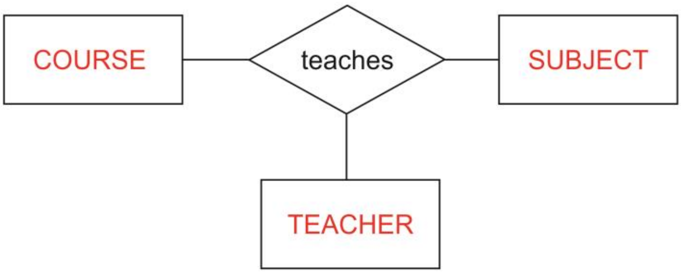


Fig 2.14 Ternary relationship

In the example above, it is seen that the University may further require the evidence that which teachers taught what particular subjects in which courses.

Check your progress 2

1. Which relationship exists when the maintenance is between two entities?

a. Unary	c. Ternary
b. Binary	d. Quaternary

2.4 Cardinalities

The cardinality can be seen as a relationship that exists for a number of occurrences of entity B so as to associate with entity A. There is least cardinality as well as highest cardinality for each relationship; though an indefinite greatest cardinality is given away as N. Cardinality limits are more often than not derived from the organisations, policies or external constraints.

Consider a case at the college where every Teacher is able to teach an indefinite utmost number of subjects as lengthy as his/her weekly hours carry out that will not go beyond 24. Teachers possibly will teach 0 subjects but since they are occupied in non- teaching projects, so the cardinality limits for Teacher can be (O, N) as shown in fig 2.15.

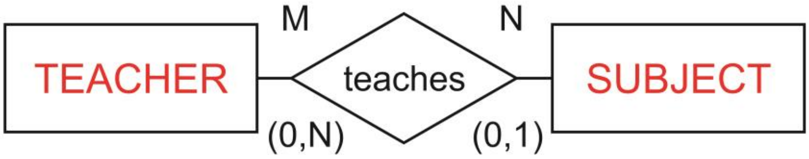


Fig 2.15 Cardinality limits for Teacher

As per the college policies, each Subject to be taught by only single teacher, further possibly, Subjects might not have allotted to a teacher, so in this case the cardinality limits for SUBJECT will be (0,1).

The connectivity of a relationship is its classification. It may be a one to one (1:1), one to many (1:M) or many to many (M:N) relationship. Relationships connectivity is represented by a 1, M or N next to the related entity.

One to one (1:1)

In this, a Principal manages one Department, where every Department is managed by single Principal Teacher as described in fig 2.16

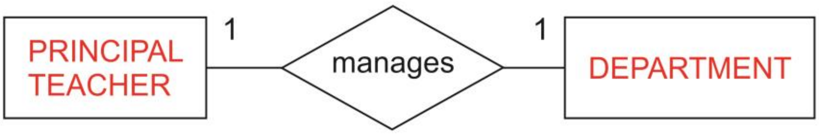


Fig 2.16 1:1 Cardinality of Principal Teacher

One to many (1:M)

In this, a Subject can be given many times, but each contribution belongs to one Subject as shown in fig 2.17 of one to many relationships

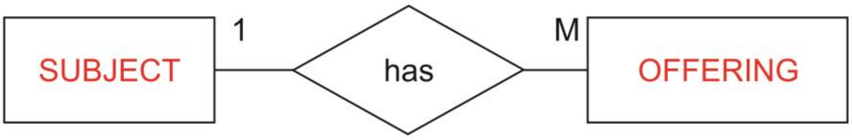


Fig 2.17 One to many relationships

Many to many (M:N)

In this case, a Teacher can teach several different Subjects, but every Subject may be taught by several Teachers. This is explained in fig 2.18 in many to many relationships:

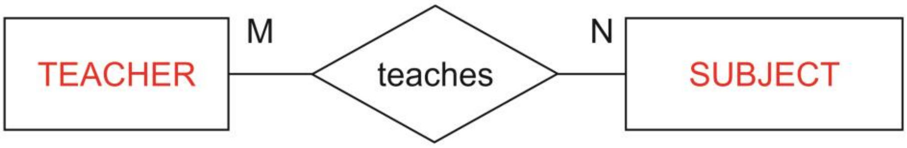


Fig 2.18 Many to many relationship

Many-to-one

An entity in A is related to at most one entity in B, but an entity in B is related to any number of entities in A.

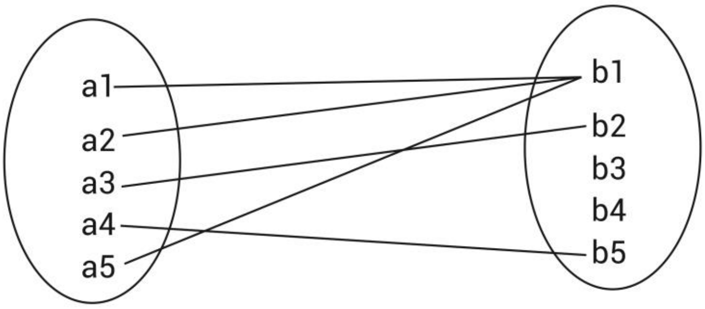


Fig 2.19 many to one relationship

Check your progress 3

1. Which cardinality is explained from, “in a School, a teacher can teach many different Subjects, but every subject can be taught by many teachers?”

   a. Many to many
   c. Many to one

   b. One to many
   d. One to one

2.5 Overview of Extended ER model

It is the process of assigning sub grouping inside an entity set and is commonly narrate as specialization. In such model, we apply IS for showing relationship having specialization. Is a type of relationship referred as super class-subclass relationship.

Example: Person IS an Employer

Person IS an Owner

Employee IS A General Manager

Generalization: In such extended model, the design process appears in bottom up manner. In this numerous entity sets are manufactured in a higher level entity set depending upon its features.

Example: Consumer as well as occupier entities can be conceptualized into a higher amplitude entity Person.

Accredit inheritance: Accredit of higher degree entity set are chosen by submerge level entity set.

Aggregation: Aggregation is a consideration where association sets are practising as higher degree entity sets. An association set is confined by entity set from inside because that alike entity sets conducted to endowment of associations.

Specialization: It is a mechanism which will explain about group of subclasses related to an entity type known as superclass. It explains based on special features that are available in entities in case of superclass. It contains varied specific variations of similar entity type with respect to certain special features as shown in fig 2.20. It is studied that subclass contains its own:

- Local attributes
- Particular relationship types

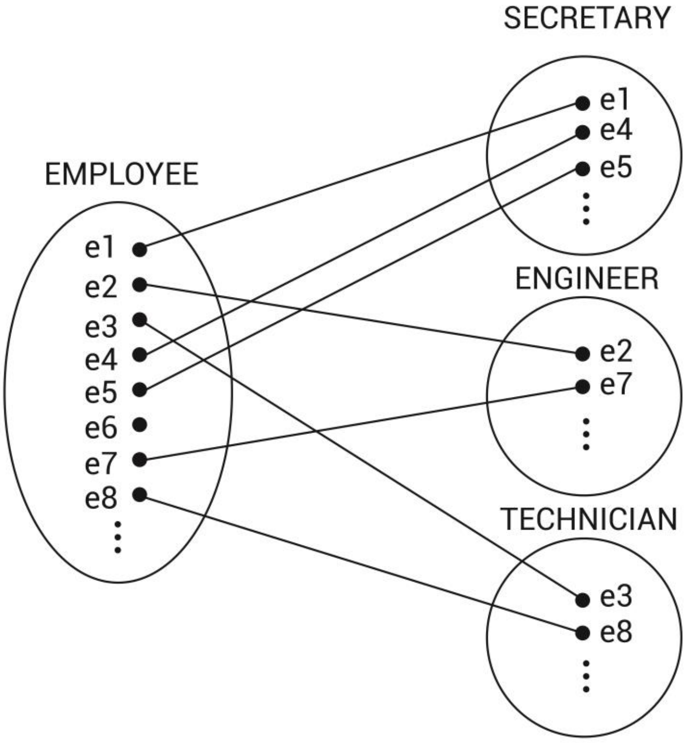


Fig 2.20 Instances of Specialization

Check your progress 4

1. In an Extended ER Model, relationship is part of:

a. contribution	c. concept
b. specialization	d. all

2.6 Case Study of Extended ER model

Enhanced Entity Relationship (EER) Model is high level data model which offers extension of original Entity Relationship (ER) model. EER Models contains several design details. The EER Modelling comes out as a result for modelling highly compound databases. Consider an Enhanced Entity Relationship model of the Dream Home as shown in fig 2.21

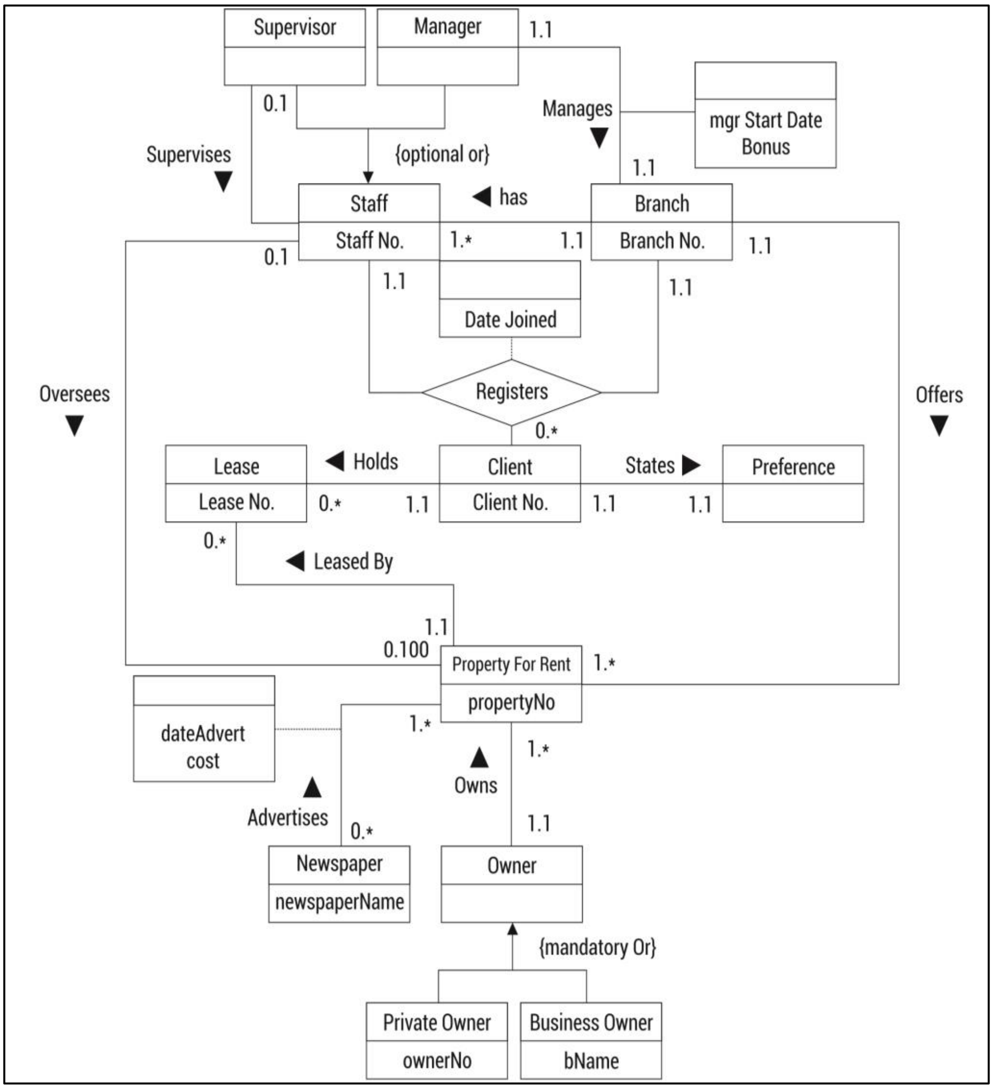


Fig 2.21Enhanced Entity Relationship model of Dream Home

Check your progress 5

1. EER Modelling, comes with:

a. low data model

b. high data model

c. complex data model

d. all

2.7 Let Us Sum Up

In the above unit we have learned that a conceptual model is a model which is prepared from work of art of concepts, which are old to help people be acquainted with understanding or reproduce a subject that shows the model.

It is studied that an Entity Relationship model is such type of model which works on the phenomena of actual-world entities as well as relationships that appear among them. Similarly preparing a real world condition in database models will led to formation of entity set, relationship set, general attributes as well as constraints. In case of a conceptual model, the modelling is done base on the work of art of concepts. The Entity-relationship diagram is a chart which signifies correlation involving database entities. The entity set is collection of similar types of entities which carries a primary key is named as strong entity set

2.8 Answers for Check Your Progress

Check your progress 1

Answers: (1-c), (2-c), (3-c)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-a)

Check your progress 4

Answers: (1-b)

Check your progress 5

Answers: (1-b)

2.9 Glossary

1. **Data Modelling -** A software engineering turns out to be more and more vital.
2. **Entity Relationship model -** It is a concept of real world entities and relationships among them.
3. **Entity set -** It is a collection of same types of entities.

2.10 Assignment

Is data modeling required for DBMS?

2.11 Activities

Is a unary and binary model having common features?

2.12 Case Study

Is one-one, one-many, many-one and many-many cardinalities are same?

2.13 Further Readings

1. Asher, H.B. (1984). Causal modelling. Sage University Paper series on quantitative applications in the social sciences, 07-003. Newbury Park, CA: Sage Publications.

2. Creswell, J.W. (1994). Research design: Qualitative and quantitative approaches. Thousand Oaks, CA: Sage Publications.

3. Kerlinger, F.N. (1979). Behavioural research: A conceptual approach. New York: Holt, Rinehart & Winston.

Block Summary

The students have given the basic of database concepts and idea about conceptualisation. The user was given a description on topics related to Database domain, model, architecture and languages. The model and origination of rules and criteria's of ER model was also detailed. The block focuses on basic understanding of DBMS languages and ER concepts that will give the knowledge about cardinalities. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

After reading this block the student will gather information on Database domain, model, architecture and languages with practice on certain rules and criteria's of ER model. The basic about advanced ER modelling is explained with conceptualisation of DBMS languages and ER concepts followed by different cardinalities.

Block Assignment

Short Answer Questions

1. What is Database Management System?
2. What is Entity-Relationship model?
3. What is an Entity set?
4. What is a hierarchical and network model?
5. Explain Cardinalities?

Long Answer Questions

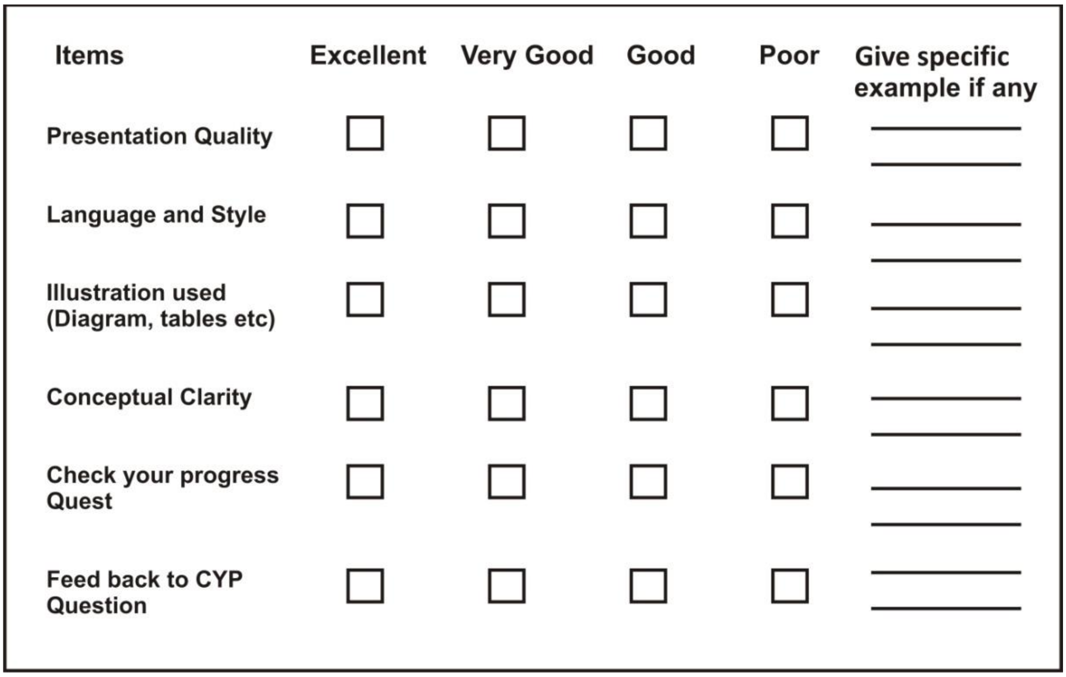
1. What is the basic concept of Entity-Relationship Diagram?
2. Write detail about Three-tier architecture of DBMS?
3. What are the advantages of DBMS over file systems?

Enrolment No. □

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

RELATIONAL DATABASE MANAGEMENT

PGDCA 103

BLOCK 2:

DATABASE INTEGRITY AND NORMALIZATION

स्वाध्याय : परमं तपः

Dr. Babasaheb Ambedkar Open University
Ahmedabad

RELATIONAL DATABASE MANAGEMENT

kmro
KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Ms. Katherine Abraham

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

RELATIONAL DATABASE MANAGEMENT

Contents

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

UNIT 1 INTRODUCTION TO DATABASE SYSTEM

Basic Concepts of Data and Database, Database management system, Data Administrator and its functions, Advantages of DBMS over file systems, Applications of DBMS, Three-tier architecture of DBMS, Data Models, Components of DBMS, Overview of Languages of DBMS (DDL, DML, DCL)

UNIT 2 CONCEPTUAL MODELLING

Introduction, Basic Concept of Entity-Relationship Diagram, Relationship concept and its types (unary, binary and ternary), Cardinalities (one-one, one-many, many-one and many-many), Overview of Extended ER model, Case Study of Extended ER model

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

UNIT 1 DATABASE INTEGRITY CONCEPTS

Introduction, Domain Integrity constraints, Entity Integrity constraints, Referential Integrity constraints

UNIT 2 NORMALIZATION

Introduction, Basic Concept of designing, Need of normalization, Decomposition (lossy and lossless), Functional Dependency, Full Functional Dependency, Armstrong Axioms of Functional Dependencies, Normalization Rules (1NF, 2NF, 3NF and BCNF), Examples based on normalization

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

UNIT 1 RELATIONAL ALGEBRA

Introduction, Basic concepts of relational algebra, symbols of relational algebra, operations on relational algebra, Examples based on relational algebra

UNIT 2 QUERY LANGUAGE

Introduction, Introduction to Structured Query Language, Data Definition Language (DDL), Data Manipulation Statement (DML), Data types in MySQL, Operators (Arithmentic, Comparison, Logical), SQL Functions, Group function (AVG, MAX, MIN, SUM and COUNT), COMMIT and ROLLBACK

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

UNIT 1 DATA RETRIEVAL SQL STATEMENT

Introduction, Single table query without condition, Single table query with condition, Group by clause, Orders by Clause, Self Join Natural Join and Sub query, Examples based on SQL concepts

UNIT 2 TYPES OF DATABASE SYSTEMS

Introduction, Centralized Database System, Parallel Database Systems, Distributed Database Systems, Client-Server Database System

Dr. Babasaheb
Ambedkar
Open University

PGDCA 103

RELATIONAL DATABASE MANAGEMENT

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

UNIT 1

DATABASE INTEGRITY CONCEPTS 03

UNIT 2

NORMALIZATION 17

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

Block Introduction

In DBMS, referential integrity explains about the database in total, where the things are arranged in way so that if a column is present in two or more tables inside the database, at that time, any change to a value in column in a single table will reflect in equivalent modification to facilitate value which exists in other tables. Normalization is a significant measurement of the database development process as; it frequently gives the appearance of real look of database and shows the working of how data are going to work together in database.

In this block, we will learn and study about Database Integrity Concepts and Normalization. We will cover the topics related to Domain Integrity constraints, Referential Integrity constraints, Entity Integrity constraints with concept and need of normalization. The student will be given with the knowledge about different rules and criteria's of normalization.

The block will focus on basic understanding about different integrity concepts and explains more on the concept of normalization and its requirement in making database. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Block Objective

After learning this block, you will be able to understand:

- The concept of Domain Integrity constraints.
- Understanding Entity Integrity constraints.
- Detail about Referential Integrity constraints.
- Idea about normalization.
- Basic of Functional and Full Functional Dependency.
- Idea about Armstrong Axioms of Functional Dependencies.

Block Structure

Unit 1: **Database Integrity Concepts**

Unit 2: **Normalization**

UNIT 1: DATABASE INTEGRITY CONCEPTS

Unit Structure

1.0 Learning Objectives

1.1 Introduction

1.2 Domain Integrity Constraints

1.3 Entity Integrity Constraints

1.4 Referential Integrity Constraints

1.4.1 Foreign Key

1.4.2 Candidate Key

1.5 Let Us Sum Up

1.6 Answers For Check Your Progress

1.7 Glossary

1.8 Assignment

1.9 Activities

1.10 Case Study

1.11 Further Readings

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Integrity constraints
- Primary key in Entity Integrity constraints
- Basic of features of Referential Integrity constraints

1.1 Introduction

Earlier than the implementation of database tables, one ought to define the integrity restrictions. Integrity can be called as something just similar to be right as well as consistent. The data in a database should be obliged to be right in addition to behave in good condition. There are different integrities constraints like domain integrity, entity integrity, referential integrity as well as foreign key integrity.

The Entity integrity shows that in each table the primary key should have both the conditions:

1. Primary key is unique inside the table
2. Primary key column(s) has no null values.

It is seen that, referential integrity explains about the database in total, where the things are arranged in way so that if a column is present in two or more tables inside the database, at that time, any change to a value in column in a single table will reflect in equivalent modification to facilitate value which exists in other tables. It shows that in RDBMS, there should be an arrangement to take suitable performance in order to spread change in single table from a particular table to the other tables where changes occur.

1.2 Domain Integrity constraints

The main aim of constraints on a Database is basically to preserve the accuracy as well as integrity of data. Domain integrity explains the meaning of a suitable set of values for a feature.

In this, you define:

- data type
- length or size
- null value allowed
- value unique or not

For an attribute.

Many of DBMS integrity agree to show the output format and/or input mask for the attribute. Such explanation guarantees of a particular attribute that contains right as well as proper value in a database.

Moreover, a domain is set of probable values of an attribute. The Domain for the states as well as territories of India includes Delhi, Mumbai, Goa, Amritsar, Tibet, Northern Territory as well as New Delhi. In a database furthermore we may impede the approximations to be accessed as DL, MUM, GA, AMT, TIB, NT as well as ND. These approximations are following the set of acceptable approximations for the region.

It is observed that complete approximations accessed in the column should be from the identical region. If, the acknowledged STATE acquires region approximations as discriminated over and above the data attained into this column in the database may be solitary DL, MUM, GA, AMT, TIB, NT as well as ND. It is likely for an augmented one to accredit to allocate identical region. On account of INDIA_theSTATE acquires equivalent region as STATE_Name in a distinguishing database.

The barriers on a region conforming to its region name, explanation, data category, area, as well as set of appropriate values. Several regions restrictions will carry standard approximations, amount of decimal places, style as well as maximum along with minimum approximation amount. Every region is given with a single area constraint.

Example:

A region constraint for Pune University database will show:

Region Name	Person Name
Explanation	Name of person with First, Middle or Surname.
Data Type	Character
Total Length	40 characters
Accepted Values	'A'-'Z', 'a'-'z', and '-'
Standard Value	' ' (null)
Style	Xxxxxxxx (where initial letter of the Name being capital)

In the above Pune University database, a Teacher entity contains two attributes: LName as well as FName, both sharing domain Person’s Name.

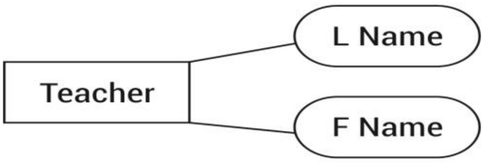


Fig 1.1domain constraint

Attribute Name	Lname
Domain Name	Person’s Name
Description	Last Name of Teacher
Data Type	Character
Max Length	40 characters
Allowable Values	'A'-'Z', 'a'-'z', and '-'
Default Value	' ' (null)
Format	Xxxxxxxx

Not Null Constraint

It is an SQL constraint which will make certain that every row in table will have standard finite approximation for particular column that is approximated as not null. This shows the no entry for null value whose syntax is:

```
[CONSTRAINT constraint name] NOT NULL
```

If we want to develop an employee table using Null values, then we will have:

```
CREATE TABLE employee
(id number (5),
name char (20) CONSTRAINT nm_nn NOT NULL,
dept char(10),
age number(2),
salary number(10),
location char(10)
);
```

Check constraint

It is another SQL constraint which explains more of business rule placed on certain column. In this, every rows should clear using business rule. It is seen that such type of constraint is applicable for an individual column or many columns. The syntax of check constraint is:

```
[CONSTRAINT constraint name] CHECK (condition)
```

If you want to generate employee table, then you can check the constraint at column level as well as table level. While doing so you have to choose gender of a person which will make you to show as:

Check Constraint at column level:

```
CREATE TABLE employee
(id number (5) PRIMARY KEY
name char (20)
dept char(10),
age number(2),
gender char(1) CHECK (gender in (‘M’,’F’))
salary number(10),
location char(10)
);
```

Check Constraint at table level:

```
CREATE TABLE employee
(id number (5) PRIMARY KEY
name char (20)
dept char(10),
age number(2),
gender char(1)
salary number(10),
location char(10)
CONSTRAINT gender_ck CHECK (gender in (‘M’,’F)’)
);
```

Check your progress 1

1. The main view of constraints on Database is:

   a. save accurateness

   b. save integrity of data

   c. both a and b

   d. neither a nor b

1.3 Entity Integrity constraints

Entity integrity is related to ensuring every row of a table will carry a one of its kind plus non-null primary key value. It further says that every row in a table shows an individual instance of an entity type model by the table. A necessity of E F Codd in his determining paper is to show primary key of an entity or any part of it without taking null value.

Entity integrity constraint gives explanation in relation to primary keys which are not empty. Conveniently, there ought to be an appropriate value in primary key field. It turns out as primary key value which is engaged to obtain in dependent rows in a table. It seems that if null values exist for primary keys, at that time it gives you an idea that such type or rows could not be made available.

Similarly, there are definite null values apart from primary key fields which shows that an individual doesn't need to be up to date with the value meant for such field. Null value is different from zero value or space.

In a feasible design forum we are constraining to examine that Primary Keys monopoly has been assigning to every table as well as there are a set of secret approximations that can be developed to create the row. Additionally it brings about conclusive evidence that foreign keys will referral the row to affix cells for inquiries on the database.

The Entity completeness commands describes about every Entity that conveys a Primary Key which is described as abnormal identifiers at every case of the entity. Lately following the entity possesses altered to an association or table, the authentication mechanism desires to acquire every row that holds extraordinary approximation in the primary key column. No null approximations are dispensed in the Primary Key column.

Example:

Consider an example of a Company that is involved in Car Rentals. We see that such company has a database where there exists a Car table where every car is assigned with proper as well as distinctive Registration _No. The database has certain cars that have no rate because either they are broken or it is a new brand where the Rate field carries null values as described below.

The entity integrity constraints promise that a definite row in a table can be recognized.

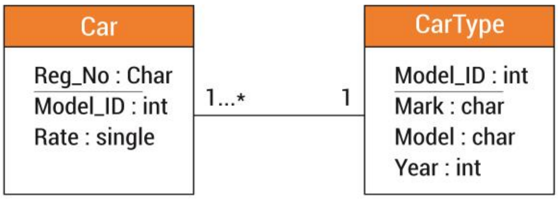
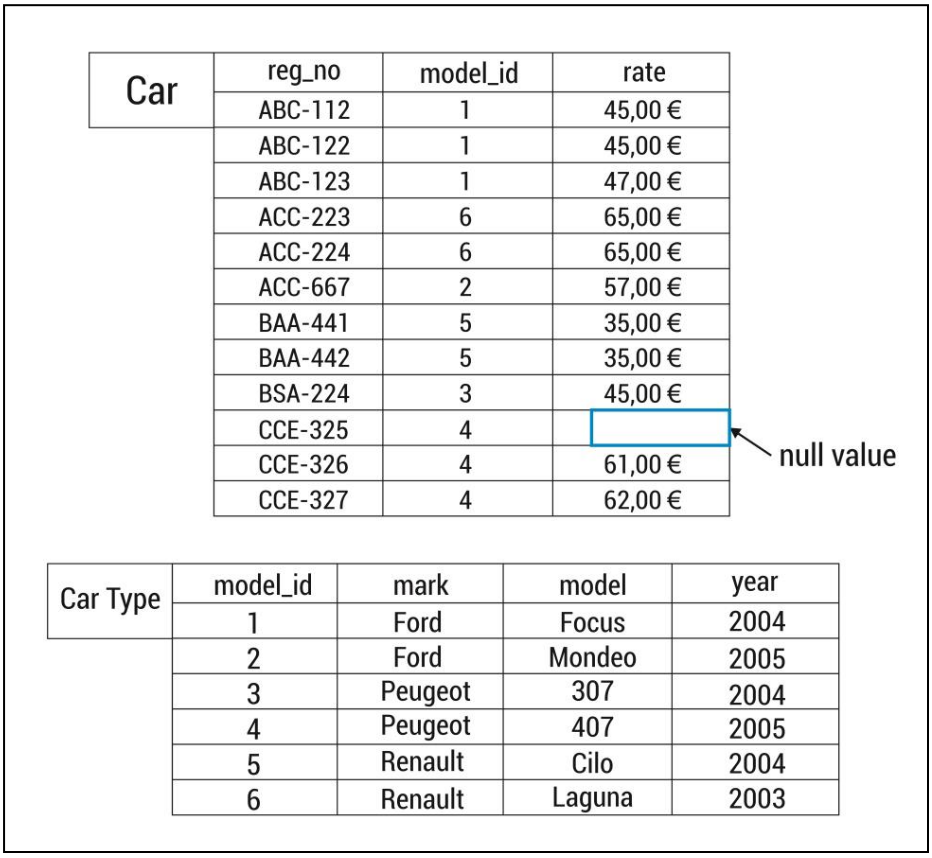


Fig 1.2 Example



Car	reg_no	model_id	rate
	ABC-112	1	45,00 €
	ABC-122	1	45,00 €
	ABC-123	1	47,00 €
	ACC-223	6	65,00 €
	ACC-224	6	65,00 €
	ACC-667	2	57,00 €
	BAA-441	5	35,00 €
	BAA-442	5	35,00 €
	BSA-224	3	45,00 €
	CCE-325	4	
	CCE-326	4	61,00 €
	CCE-327	4	62,00 €

null value

Car Type	model_id	mark	model	year
	1	Ford	Focus	2004
	2	Ford	Mondeo	2005
	3	Peugeot	307	2004
	4	Peugeot	407	2005
	5	Renault	Cilo	2004
	6	Renault	Laguna	2003

Fig 1.3 null value

Check your progress 2

1. Null values are:

   a. same as primary key fields

   b. same as secondary key fields

   c. not the primary key fields

   d. all

1.4 Referential Integrity constraints

Referential integrity is another type of database model that makes it sure to facilitate the interaction flanked by tables which remains steady. When a single table contains a foreign key to other table, then the idea of referential integrity explains with the aim that you might not put in a record inside the table which carries the foreign key except that there is an equivalent record in the linked table. Further, it is included that the performance which is called as cascading bring up to date further more cascading delete, which makes certain changes through to the linked table that are return in the primary table. The referential integrity constraint is particular among two tables in addition; it is applied so as used to preserve the steadiness that exists among rows flanked by the two tables. There are certain rules while applying, which are as follows:

Rules:

- You cannot eliminate a record from primary table if similar records be near in a connected table.
- You cannot change a primary key value into primary table if that record has connected records.
- You cannot go through a value in foreign key field of connected table that doesn't carry on in primary key of primary table.
- On the other hand, you can place a Null value in foreign key, by representing that the records are not connected.

1.4.1 Foreign key

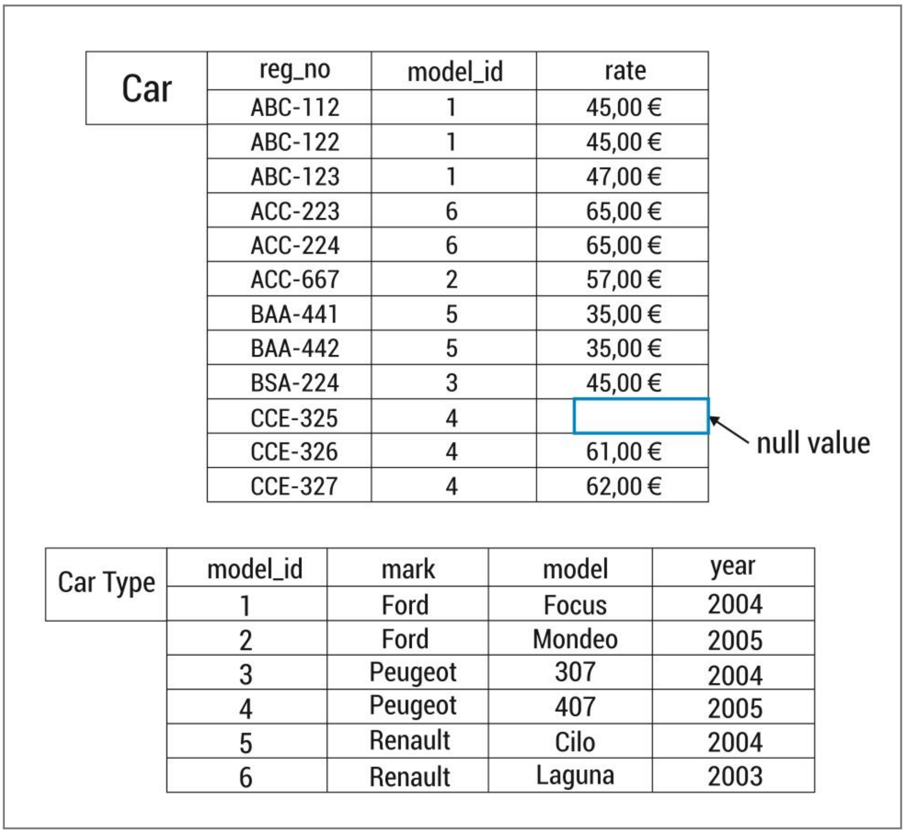
We see that an integrity constraint contains 2 foreign keys:

- cascade update related fields
- cascade delete related rows

Both such constraints affect the referential integrity constraint.

Cascade Update Related Fields

In such integrity constraint, you can alter primary key of row in primary table at any moment of time. Here the foreign key values are adjusted in the matching row which is held in related table. Such constraint will rule against rule 2 in the referential integrity constraints.



Car	reg_no	model_id	rate
	ABC-112	1	45,00 €
	ABC-122	1	45,00 €
	ABC-123	1	47,00 €
	ACC-223	6	65,00 €
	ACC-224	6	65,00 €
	ACC-667	2	57,00 €
	BAA-441	5	35,00 €
	BAA-442	5	35,00 €
	BSA-224	3	45,00 €
	CCE-325	4	
	CCE-326	4	61,00 €
	CCE-327	4	62,00 €

null value

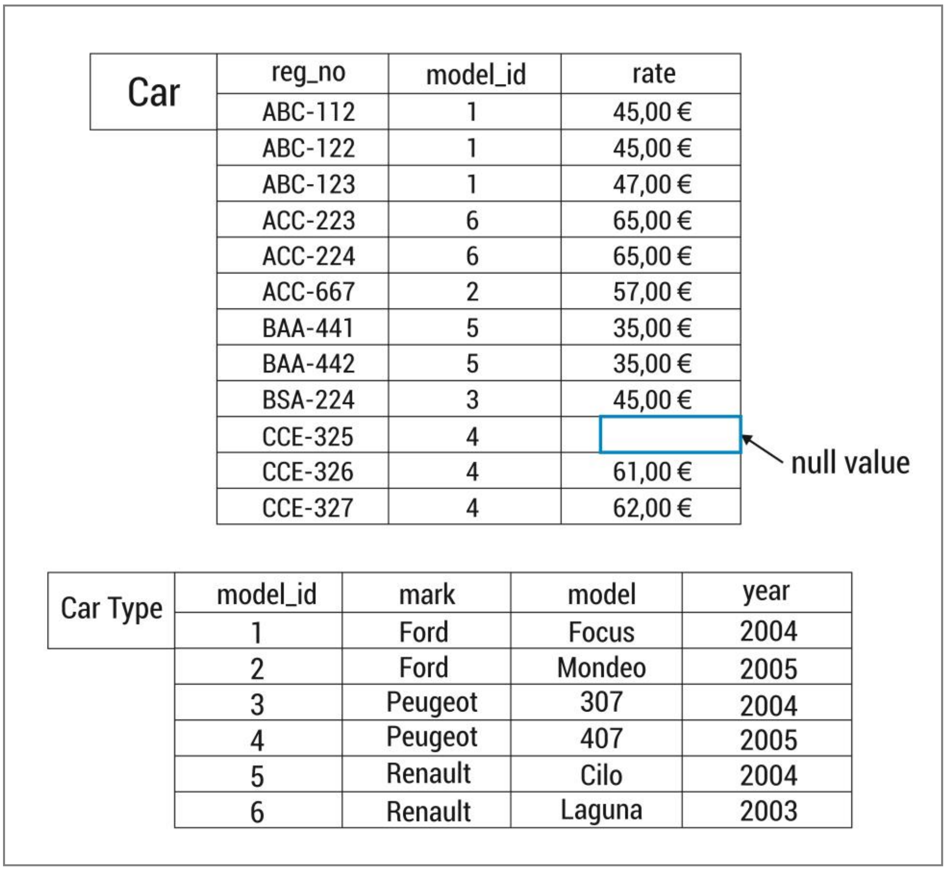
Car Type	model_id	mark	model	year
	1	Ford	Focus	2004
	2	Ford	Mondeo	2005
	3	Peugeot	307	2004
	4	Peugeot	407	2005
	5	Renault	Cilo	2004
	6	Renault	Laguna	2003

Fig 1.4 referential integrity constraints

Figure 1.3, shows with an explanation about constraint along with relationship that can be available with tables where Car is marked and CarType is written. You can also introduced model_id in this CarType table as shown above. You can alter the model_id 1 as (Ford Focus) to model_id 100 in CarTypetable, whereas the model_ids in Car table will be altered from 1 to 100 as for cars ABC-112, ABC-122, ABC-123.

Cascade Delete Related Rows

In such integrity constraint, you can remove a row located in the primary table anytime. In this, the matching rows are directly removed in the associated table. The result is that such constraint will rule against rule 1 placed in referential integrity constraints.



Car

reg_no	model_id	rate
ABC-112	1	45,00 €
ABC-122	1	45,00 €
ABC-123	1	47,00 €
ACC-223	6	65,00 €
ACC-224	6	65,00 €
ACC-667	2	57,00 €
BAA-441	5	35,00 €
BAA-442	5	35,00 €
BSA-224	3	45,00 €
CCE-325	4	
CCE-326	4	61,00 €
CCE-327	4	62,00 €

null value

Car Type

model_id	mark	model	year
1	Ford	Focus	2004
2	Ford	Mondeo	2005
3	Peugeot	307	2004
4	Peugeot	407	2005
5	Renault	Cilo	2004
6	Renault	Laguna	2003

Fig 1.5 referential integrity constraints

While explaining such constraint among the tables Car as well as CarType, then it becomes feasible to remove rows from CarType table. It is seen that when you remove the Ford Focus row from CarType table, then you will find that the cars ABC-112, ABC-122, ABC-123 will also be removed from Car table.

You can apply following rules to solve such problems:

Rule 1: You will not be able to remove any of rows located in CarType table as every car types are used in Car table.

Rule 2: You will not be able to remove any model_ids in the CarType table in view of the fact that every car types are in use in Car table.

Rule 3: Whatever values you enter in model_id field placed in the Car table should present in model_id field in CarType table.

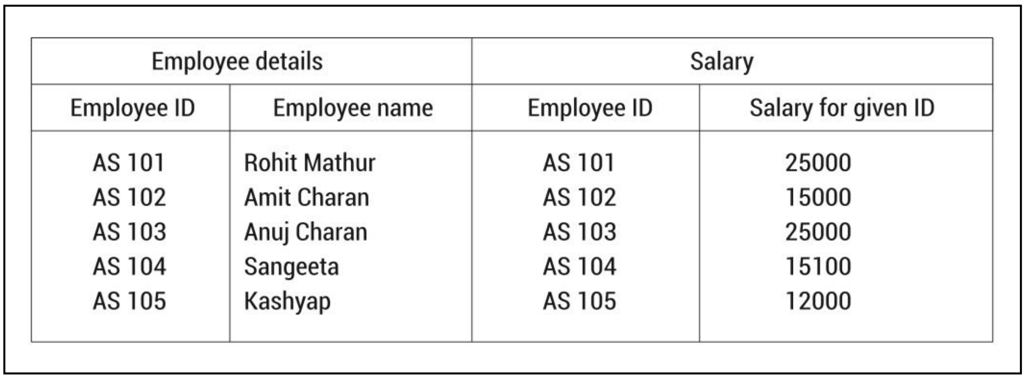
Rule 4: The model_id field in Car table encompasses null value where car types of such car are not identified.

1.4.2 Candidate key

Conveniently, there should be at slightest one smallest subset of attributes in the relation, which can be recognizing a tuple exclusively. Such type of negligible subset of attributes is termed as a key for particular relation. When there is more than one smallest subsets, then such type of key is known as candidate keys.

Example:

The accounts department of a company has segregated the Employee details in one table and Salary in another table. Further in the Employee details table, there are two columns, the employee ID and employee name. Further in the salary table, they have two columns, the employee ID and salary for given ID.



Employee details		Salary	
Employee ID	Employee name	Employee ID	Salary for given ID
AS 101	Rohit Mathur	AS 101	25000
AS 102	Amit Charan	AS 102	15000
AS 103	Anuj Charan	AS 103	25000
AS 104	Sangeeta	AS 104	15100
AS 105	Kashyap	AS 105	12000

Fig 1.6 Employee table

If the company wishes to remove an employee as he no longer works with it, then you have to delete his entry from the Employee table, as his identity is also present in Salary table. To remove an employee from the Salary table manually is very troublesome. If that employee presence is there in Companies other table, then he would have to be deleted from those tables also.

We can use referential integrity to solve such problem, so that in future we will not delete any name manually from any table. We will:

- First define the employee ID column in Employee table as primary key.
- Secondly, define the employee ID column in Salary table as foreign key which points to primary key located in employee ID column in Employee table.
- Third, add a constraint to the Salary table, which will add in selected cascading delete, where an employee can be deleted from Employee table any time.

From the above we see that if any entries that employee carries in Salary table will automatically get removed from Salary table.

Check your progress 3

1. A candidate key is also known as:

   a. Secondary key

   b. Alternate key

   c. Composite key

   d. Concatenate key

2. Which is not the rule of referential integrity constraint?

   a. You cannot remove a record from primary table if matching records are there in associated table.

   b. You can transform a primary key value in primary table if that record has associated records.

   c. You cannot enter a value in foreign key field of associated table that doesn't survive in primary key of primary table.

   d. You can put a Null value in foreign key, by indicating that records are not linked.

1.5 Let Us Sum Up

In this unit we have learnt that referential integrity is related to explanation concerning with database where things are arranged in certain ways where column present in two or more tables in database will alter value in column.We have studied that Entity integrity ensures every row of a table to carryout non null primary key value. In this every row in table shows individual instance of an entity type model by the table.

The Entity completeness describes about every Entity which is transferred as Primary Key showing abnormal identifiers at every case of the entity. Referential integrity is another type of database model that makes it sure to facilitate the interaction flanked by tables which remains steady. When a single table contains a foreign key to other table, then the idea of referential integrity explains with the aim that you might not put in a record inside the table which carries the foreign key except that there is an equivalent record in the linked table.

It is found that the number of DBMS describes the output format and/or input mask for the attribute. The constraints on domain contain a domain name, description, data type, size and acceptable values. Entity integrity ensures all rows of a table as well as non-null primary key value.

1.6 Answers for Check Your Progress

Check your progress 1

Answers: (1-c)

Check your progress 2

Answers: (1-c)

Check your progress 3

Answers: (1-b), (2-b)

1.7 Glossary

1. **Key -** It is a single or combination of many fields which access or retrieve data rows from table.
2. **Primary Key -** It is an attribute or combination of attributes having unique identity as row or record in relation.
3. **Secondary key -** It is a field or combination of fields which retrieves.
4. **Candidate Key -** It is a relation containing many fields or combination of fields used as primary key.
5. **Composite key -** It is a primary key containing two or more attributes.
6. **Sort key -** It is a field or combination of fields used to sequence the stored data.
7. **Foreign Key -** It is an attribute in relation having value that is matched with primary key in another relation.

1.8 Assignment

What is property in a table?

1.9 Activities

Can a composite key be act as candidate key?

1.10 Case Study

Can you use the same foreign key constraint in two different tables?

1.11 Further Readings

1. Dependency Structures of Data Base Relationships by W. W. Armstrong.
2. Cardinal numbers and Formalized Mathematics by Grzegorz Bancerek.
3. The fundamental properties of natural numbers by Grzegorz Bancerek.
4. Finite sequences and tuples of elements of a non-empty sets by Czesław Byliński.

UNIT 2: NORMALIZATION

Unit Structure

2.0 Learning Objectives

2.1 Introduction

2.2 Basic Concept of designing

2.3 Need of normalization

2.4 Decomposition

2.5 Functional Dependency

2.6 Full Functional Dependency

2.7 Armstrong Axioms of Functional Dependencies

2.8 Normalization Rules

2.9 Examples based on normalization

2.10 Let Us Sum Up

2.11 Answer for Check Your Progress

2.12 Glossary

2.13 Assignment

2.14 Activities

2.15 Case Study

2.16 Further Readings

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- About normalization
- Basic of Composite key
- Understanding of Normal Forms

2.1 Introduction

While designing a database, if its design is not just right, then it seems that it carries anomalies. Controlling a database with anomalies is highly impracticable. While for a designer, there exist three possibilities:

- **Updating anomalies -** If data items are spread as well as are not connected to each one properly, it could possibly lead to extraordinary situations. If we try to update one data piece having its copies spread over numerous places, the small number of instances gets modernized correctly at the same time few others are left by old values. Such examples will make the database in an inconsistent state.
- **Deleting anomalies -** When you make an effort to delete a record, however parts of record was remained undeleted as lack of knowledge, then the data at that time gets saved somewhere else.
- **Inserting anomalies -** When you try to place something in the data in record which is not available at all.

With this method, you can eliminate all anomalies as well as fetch the database to a steady state. By Normalization process, you can thoroughly inspect relations for anomalies furthermore, after detection; you can remove such anomalies by dividing relations into two new connected relations.

Normalization is a significant measurement of the database development process as; it frequently gives the appearance of real look of database and shows the working of how data are going to work together in database.

2.2 Basic Concept of Designing

The figure 2.1 shows the main phases of database design which is connected with application design.

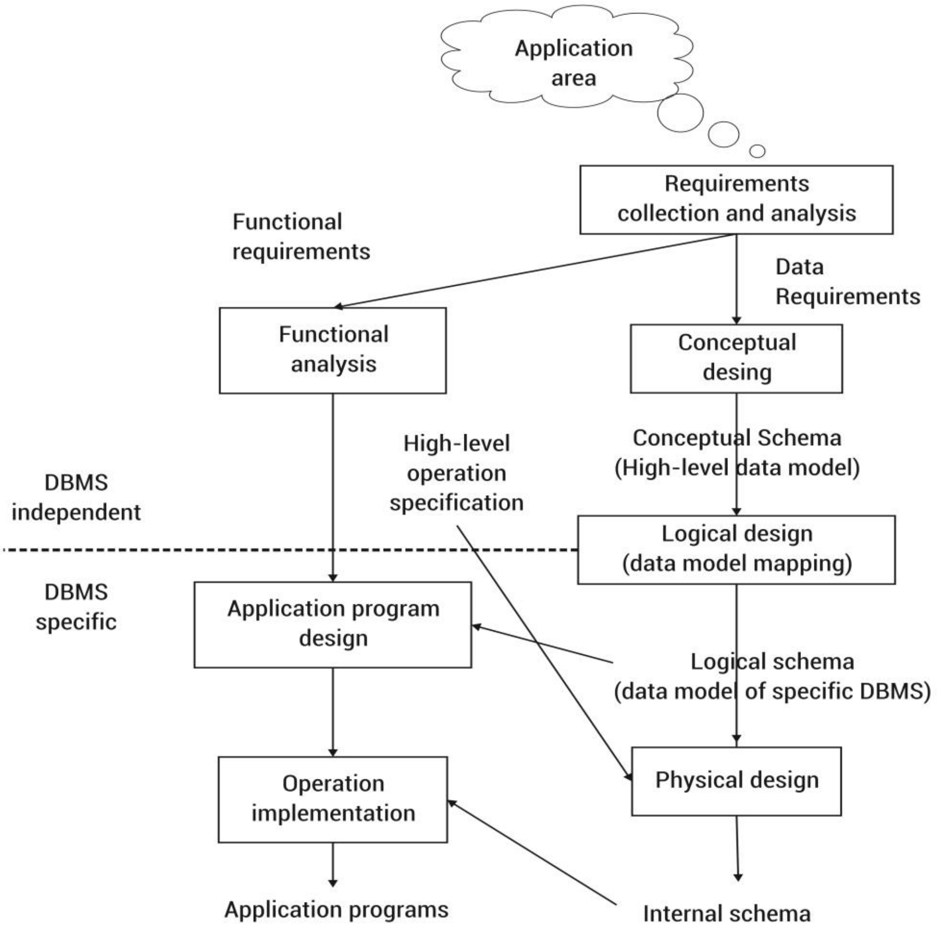

Application area
Requirements collection and analysis
Functional requirements
Data Requirements
Functional analysis
Conceptual desing
High-level operation specification
Conceptual Schema (High-level data model)
DBMS independent
DBMS specific
Logical design (data model mapping)
Application program design
Logical schema (data model of specific DBMS)
Operation implementation
Physical design
Application programs
Internal schema

Fig 2.1 Database design

The needs as well as compilation analysis phase creates together the data needs as well as functional needs. The data needs works as a foundation of database design. The data need sought to be specified as detailed as well as total form as feasible.

In identifying data needs, it is helpful to identify the famous functional needs of the application which comprises of user defined process with the purpose to apply database. The practical needs are employed as a foundation of application software design. We see that a number of phases are database management system free as well as needy. The main view is to design initially the database without assess mention relation to definite database system which simply focus on the data.

After gathering all relevant information, the next action is to produce a conceptual plan for the database, by means of high level theoretical data model which relates to conceptual design.

This phase results in Entity-Relationship (ER) diagram or else UML class diagram. These are high-level data model switch show how dissimilar entities are connected to each one. In these, the attributes carries each entity which carries all the explanation about concepts of application area.

All through the conceptual plan, the essential data model procedure can be used to identify the high-level user operations acknowledged all through the functional investigation.

It is further studied that normalization is the final portion of logical design. The objective of normalization is to get rid of redundancy as well as possibly update anomalies. Redundancy explains that similar data is kept more than once in a database. Bringing an anomaly up to date is an effect of laying-off. If a part of the data is put aside in additional one place, then the similar data is required to be updated in more than one place. By the method of normalization, you can alter the relation schema in order to decrease the laying-off. Every normalization period adds additional relations inside the database.

Check your progress 1

1.The Entity-Relationship diagram is also called as:

a. UML class diagram	c. MLU class diagram
b. ULM class diagram	d. MUL class diagram

2.3 Need of normalization

Normalization is a process where the table gets divides into various pieces also is applied to lower the redundancy in data that results in data inconsistency. If the tables are split into pieces, then to do manipulations on these, you will need joins as well as nested queries. Basically, database normalization is used to lower the redundancy as well as dependency. It significantly helps in dividing huge table into smaller ones meant for less redundancy as well as better relationship among data. Because of isolation of data, you can do additions, deletions as well as modifications of field in a single table plus spread throughout the rest of the database by the use of defined relationships. So normalization eliminates redundancy as well as potentially keeps informed anomalies. While normalising a database you will accomplish four goals:

1. Arranging data in logical groups with each group explains about small portion of the whole.
2. Minimizing the quantity of reproduction data stored in a database.
3. Construction of database where you use as well as control the data rapidly as well as professionally with no compromise on integrity of data storage.
4. Arranging the data in such a way that on modification, you can do changes in simply one position.

Check your progress 2

1. Normalization is used to:

a. reduce the table size c. expand the table size

b. increase the table size d. all of these

2.4 Decomposition

Decomposition is basically changing a relation by means of collecting smaller relations. In database, the idea of decomposition results in breaking of tables into various tables and reaching to higher normal form. There are two characteristics of good decompositions:

- Lossless
- Preserve dependencies

Lossless Decomposition

This shows the functioning without result in a loss. It retains everything. A good database should always carry such characteristics. We will explain this as:

- Assuming R to be a relation schema.
- Assuming F to be the set of functional dependencies on R.

Now we will form a decomposition of R. We see that the decomposition is lossless join decomposition of R, if one of following functional dependencies are in F^+

1) R1 ∩ R2 ----- R1

2) R1 ∩ R2 ------R2

3) R1 ∩ R2 ----- R1

4) R1 ∩ R2 ----- R2

We see that if R gets split into R1 as well as R2, then for lossless decomposition, at least one of two should hold true. Projecting on R1 and R2, and joining back, results in the relation you started with.

Now make sure that attributes is used in natural join (R1 ∩ R2) which is a candidate key for one of the two relations. It confirms that we will never have situation where false tuples originates, as for any value on join attributes, there will be a unique tuple in any relations. Further we see that decomposition is lossless if we can recover:

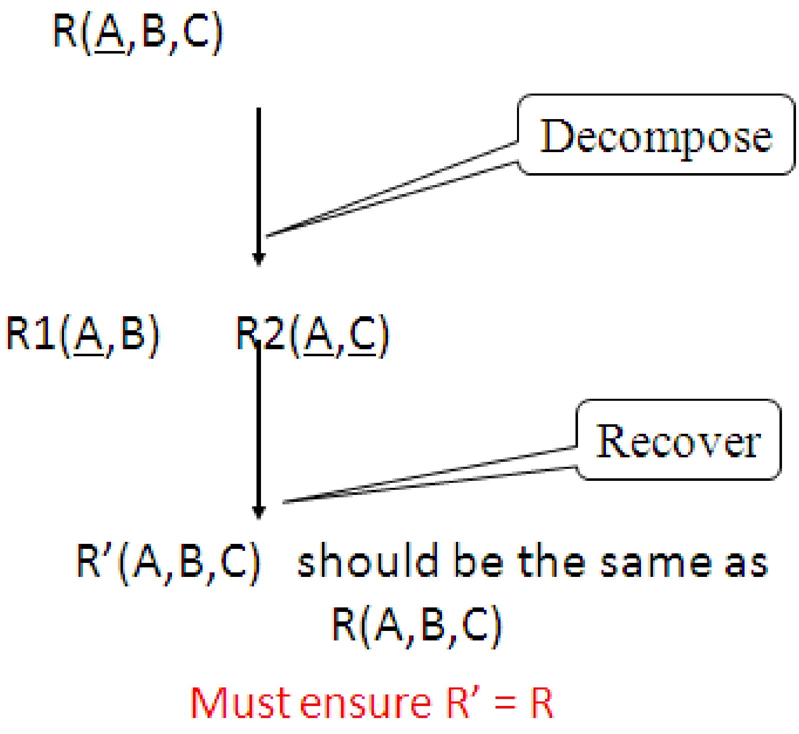
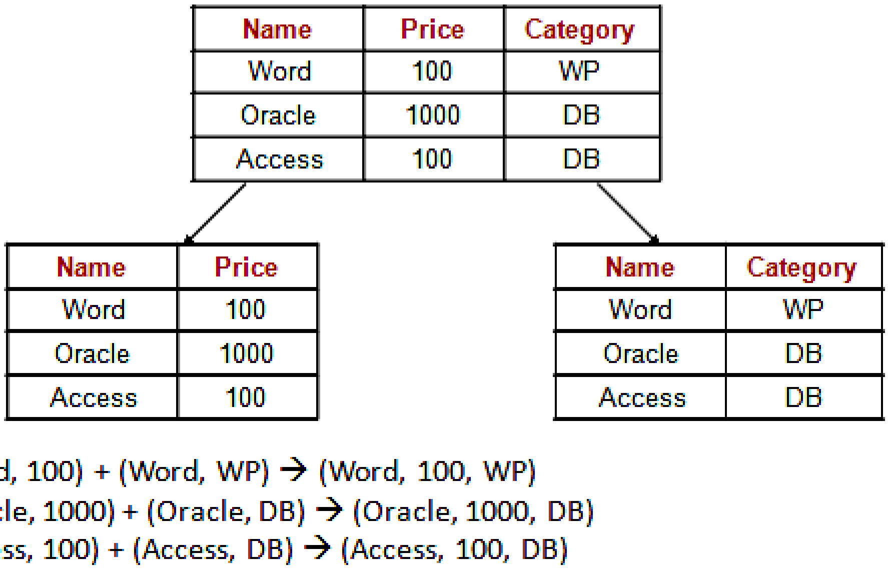


Fig 2.2 Lossless decomposition

Occasionally, the same set of data is replicated as:



Name	Price	Category
Word	100	WP
Oracle	1000	DB
Access	100	DB

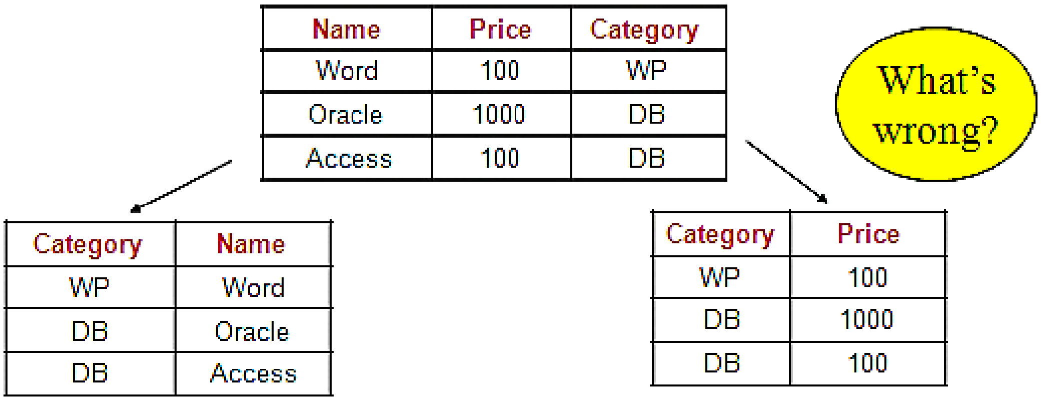
Name	Price
Word	100
Oracle	1000
Access	100

Name	Category
Word	WP
Oracle	DB
Access	DB

(Word, 100) + (Word, WP) → (Word, 100, WP)
(Oracle, 1000) + (Oracle, DB) → (Oracle, 1000, DB)
(Access, 100) + (Access, DB) → (Access, 100, DB)

Fig 2.3 data replication set

Many times it will not work as shown:



Name	Price	Category
Word	100	WP
Oracle	1000	DB
Access	100	DB

Category	Name
WP	Word
DB	Oracle
DB	Access

Category	Price
WP	100
DB	1000
DB	100

(Word, WP) +(100, WP) =(Word, 100, WP)

(Oracle, DB)+(1000, DB)=(Oracle, 1000, DB)

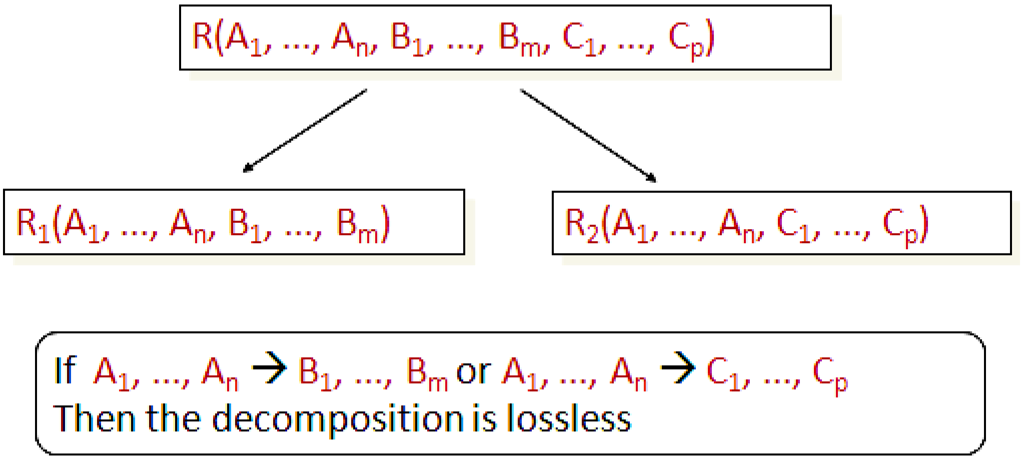
(Oracle, DB)+(100, DB)=(Oracle, 100, DB)

(Access, DB)+(1000, DB)=(Access, 1000, DB)

(Access, DB)+(100, DB)=(Access, 100, DB)

Fig 2.4 will not work data replication set

Make sure that lossless decomposition can only be when:



$R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_p)$

$R_1(A_1, \ldots, A_n, B_1, \ldots, B_m)$

$R_2(A_1, \ldots, A_n, C_1, \ldots, C_p)$

If $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ or $A_1, \ldots, A_n \rightarrow C_1, \ldots, C_p$
Then the decomposition is lossless

Fig 2.5 Lossless decomposition

Identifying Lossless decomposition:

Initially prepare a table for sub schemas of R

Now complete the table by filling eminent variables (equivalent to sub schemas)

- If initial row is full with famous variables, then it is lossless
- If no single row gets full, then you have to add famous variables

You can add famous variables by following:

- 2 or more rows with famous variables on LHS
- 1 or more rows with famous variables on RHS
- 1 or more rows with non-famous variables on RHS

Lossy Decomposition

In this type of decomposition, we lose the information not the record. Consider a relation R, which is decomposed into two relations R1 and R2.

We see that when relation R = (R1 JOIN R2) is expressed, then we find this as lossless join decomposition which according to rule is accepted.

Also when relation R (R1 JOIN R2) which acts as group then at that time, it will act as lossy join decomposition.

From the above we contemplate that, the lossy join decomposition, after combining R1 and R2 will generate extra records when it was compared with relation R. For this, we have consider a relation as shown R = (SN, Name, Address)

R1

SN	Name	Address
1111	Rohit	1 Pine
2222	Nishit	2 Oak
3333	Nishit	2 Oak

R1		R2	
SN	Name	Name	Address
1111	Rohit	Rohit	1 Pine
2222	Nishit	Nishit	2 Oak
3333	Nishit	Nishit	3Pine

If relations R1 and R2 are joined, then the result obtained is shown in the table:

R1 join R2

SN	Name	Address
1111	Rohit	1 Pine
2222	Nishit	2 Oak
2222	Nishit	3 Pine
3333	Nishit	2 Oak
3333	Nishit	3 Pine

We see that the information lost here is the address for person 2222 and 3333. If you see the original relation R, then person 2222 lives at 2 Oak. When we join relations R1 and R2, then person 2222 shows that either he lives at 2 Oak or 3 Pine. With this, it seems that the extra information will result in lossy decomposition. Here, the records were not lost, however what is lost is the information regarding which the records were in original relation.

Check your progress 3

1. Which is the popular property of decomposition?

a. Partition constraint

b. Dependency preservation

c. Redundancy

d. Security

2.5 Functional Dependency

Functional dependency is a connection which originates after one attribute exceptionally finds out one more attribute. It is an association that exists among two attributes having similar relational database table. In this, first attributes is called as determinant while second attribute is called as determined. It is visualised that for value of determinant there appears one value of determined.

If relation R with attributes X and Y, then functional dependency occurs among attributes is shown as X->Y. It means that attribute Y is functionally dependent on attribute X. Now the attribute X is a determinant set furthermore Y is a dependent attribute. So every value of attribute X is connected accurately with single attribute Y value. So it is clear that the functional dependency in database provides a restriction among two sets of attributes.

It is seen that functional dependency explains about Boyce-Codd normal form as well as third normal form. This will conserve dependency among the attributes, which results in removing the replication of information. Functional dependency is connected to candidate key, which exceptionally recognize a tuple furthermore determines the value of the entire attributes in the relation. In several cases, functionally dependent sets will not get reduced if:

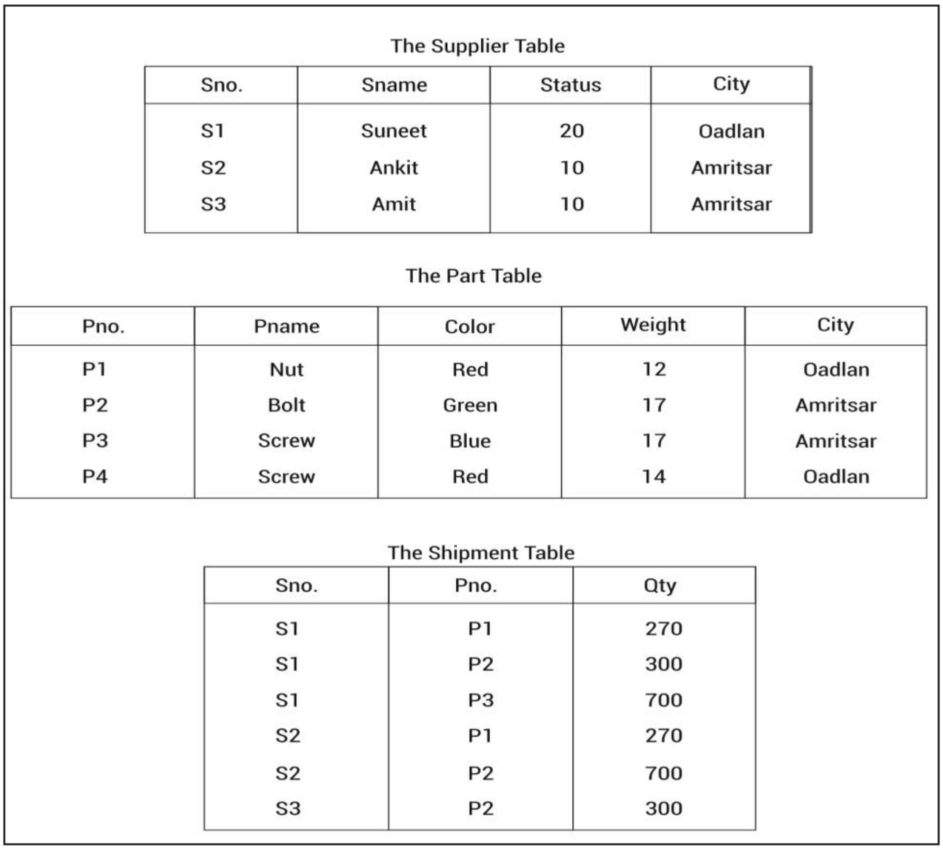
- Righth and set of functional dependency keeps only single attribute.
- Lefth and set of functional dependency not reduced as it alters the full content of set.
- Lowering any of accessible functional dependency that changes content of a set.

It is seen that there exists one of the famous property of functional dependency which is called as Armstrong's axiom that is utilised for database normalization. If R is the relation with three attributes X, Y, Z then Armstrong's axiom will result strong, when following situations are contented:

- Axiom of Transivity: If X->Y and Y->Z, then X->Z
- Axiom of Reflexivity (Subset Property): If Y is a subset of X, then X->Y
- Axiom of Augmentation: If X->Y, then XZ->YZ

Example

From the supplier database shown below



The Supplier Table

Sno.	Sname	Status	City
S1	Suneet	20	Oadlan
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

The Part Table

Pno.	Pname	Color	Weight	City
P1	Nut	Red	12	Oadlan
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Amritsar
P4	Screw	Red	14	Oadlan

The Shipment Table

Sno.	Pno.	Qty
S1	P1	270
S1	P2	300
S1	P3	700
S2	P1	270
S2	P2	700
S3	P2	300

Fig 2.6 database table

Consider a Supplier table, where:

Sno: Serial number of Supplier

Sname: Supplier name

City: Place of supplier

Status: Shows the city grades as A grade cities or B grad cities.

We explained that Sname is FD on Sno. As Sname has individual approximation for particular value of Sno (S1), then there exists individual Sname for supplier number S1.

Here FD is shown as Sno a Sname

In this, a means Sname is functionally depends on Sno. Further, city as well as status isin addition FD on Sno, as for every value of Sno there will be single city along with status.

FD is signifies:

Sno - City

Sno -Status

S. Sno – S (Sname, city, status)

Consider a shipment table, where:

S no: Number of Supplier

Pno: Part number of part

Qty: Quantity supplied for particular Part no

We see that Qty is FD on mixture of Sno, Pno for the reason that every grouping of Sno as well as Pno results simply for single Quantity.

SP (Sno, Pno) --> SP.QTY

In a dependency diagram we have attribute names as well as functional dependencies as shown.

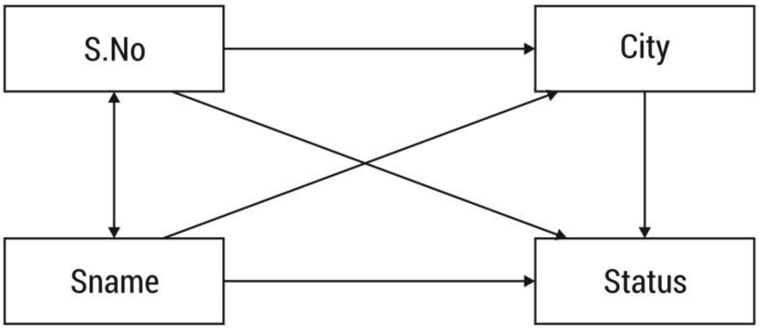


Fig 2.7 Dependency diagram

We see that from supplier table, we get following functional dependencies:

Sno: Sname

Sname: Sno

Sno: City

Sno: Status

Sname: City

Sname: Status

City: Status

Now the FD diagram of relation P can be shown as:

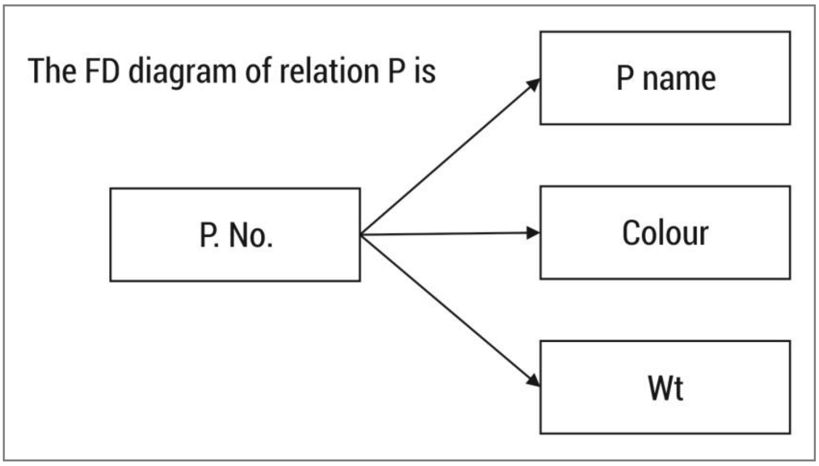


Fig 2.8 FD diagram

Now the functional dependencies for Part table:

Pno: Pname

Pno: Color

Pno: Wt

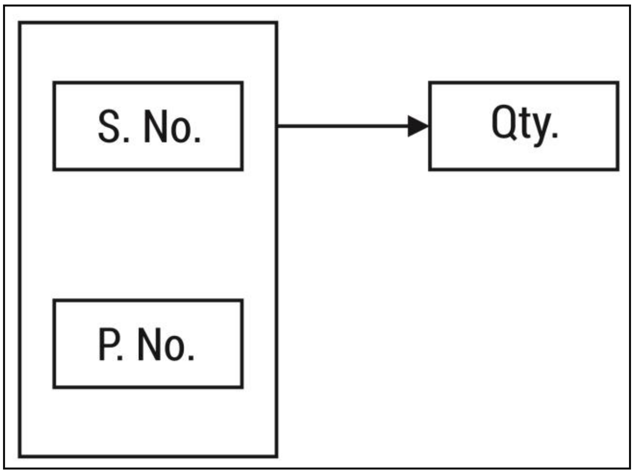


Fig 2.9 Functional dependencies

Finally we get functional dependencies the same as SP (Sno, Pno) - SP.QTY in Part table.

Check your progress 4

1. The functional dependency can be tested easily on the materialized view, using the constraints ____________.

   a. Primary key c. Unique

   b. Null d. Both b and c

2. In a relational database table, the first attributes is:

   a. determinant c. both a and b

   b. determined d. neither a nor b

2.6 Full Functional Dependency

Full functional dependency take place when you previously meet the needs for a functional dependency along with the set of attributes lying on the left side of the functional dependency statement cannot be condensed beyond.

In a relation R in addition to Functional Dependency X→Y

We see that, Y is fully functionally dependent on X as well as there must not exist any Z→Y, where Z is a appropriate subset of X.

Further, we see that if each and every non key attributes of the entity completely as well as functionally rely on key attribute of similar entity then such category of dependency is called as Full Functional Dependencies.

Example

In a student table entity having 4 column attribute as:

- Sid
- Sname
- Add
- Course Name

The table Sname, Add and Course Name are non-key attribute that are fully relying on Sid(Key Attribute).

In this, to get any information about student, you have to use only Key Attribute i.e Sid. Apart from this, all other information can be obtained on the bases of key attribute. We see that all attribute (Sname,Add,CourseName) completely rely on key attribute. Hence because of this, it is called as Full Fuctional Dependencies.

Check your progress 5

1. Consider a relation R(A,B,C,D,E) with functional dependencies as ABC -> DE and D -> AB. The number of super keys of R is:

a. 2 c. 10

b.7 d. 12

2.7 Armstrong Axioms of Functional Dependencies

If F continues set relevantly functional dependencies that time the fastener of F, described as F^+, endures the set dependently all-inclusive functional dependencies feasibly intimated nearby F. Armstrong’s codes are set of commands, when approached frequently brings about obstruction of functional dependencies.

Reflexive principle: If alpha endures a set of acknowledges as well as beta is_subset_of alpha, that time alpha accepts beta.

Augmentation principle: if a → b assumes along with y endures acknowledge set, that time ay → by additionally assumes. It continues affixing which acknowledges in associations, operates and not alteration in elementary dependencies.

Transitivity principle: It is equivalent as transitive principle in algebra, if a → b, we assumes along with b → c that will affirms if a → c that additionally gathers a → b which endures identified appropriately as shown by b.

Check your progress 6

1.The logically implied functional dependencies can be find by certain rules, such collection of rules is known as:

a. Axioms

b. Armstrong's axioms

c. Armstrong

d. Closure

2.8 Normalization Rules

Database Normalisation is a method of put in order the data in database. Basically normalization is a technique which was introduced for decomposing tables to keep away from data loss or unwanted features such as:

- Insertion
- Removing and updated Anamolies

It is a multi-step process that position data into table formation by keeping out double data from relation tables. The process of normalization applies mostly to:

- Get rid of useless data.
- Make certain about data dependencies.

Lacking of normalization will led to difficulty in handling and updating database, with no data loss. Insertion, Updation as well as Deletion Anomalies are very common if database is not normalized.

Normalization Rule

There are 4 different rules of normalization such as:

- First Normal Form
- Second Normal Form
- Third Normal Form
- BCNF

First Normal Form (1NF)

As per First Normal Form, no two Rows of data be required to hold repeating group of information. It seems that in FNF, every set of column should possess a unique value like multiple columns which cannot be used to carry out in similar row. Every table should be arranged in rows and every row should contain primary key that distinguishes it as only one of its kind.

The Primary key is more often than not a single column, but from time to time more than one column can be mutual to produce a single primary key. For instance consider a table which is not in First normal form.

Consider a table showing details of each student:

Student	**Age**	**Subject**
Nishit	15	Biology, Maths
Rohit	14	Maths
Amit	17	Maths

It is seen that in First Normal Form, in the least, the row ought tohave a column where more values can be kept, by separating with commas. Rather than that, we must separate such data into multiple rows. In the student table subsequent 1NF will be seen as:

Student	**Age**	**Subject**
Nishit	15	Biology
Nishit	15	Maths
Rohit	14	Maths
Amit	17	Maths

Applying First Normal Form, we see that data redundancy rises, since there are several columns having similar data in many rows having a unique identity.

Second Normal Form (2NF)

The quality of the second normal form is that there should not partly dependency of column on main key. It is found that a table containing end to end joining of primary key will not be part of primary key and be kept complete end to end joining of key in order to work further. It is found that if any column having single end to end key, then during that time the table will not go after with Second normal form.

In illustration of First Normal Form, it is seen that there are 2 rows which covers many subjects that are taken by the students. At the same time, this is searchable, additionally pursue First normal form, so it is inefficient usage of space. In the Table, the First Normal Form, the candidate key is {Student, Subject}, Age of Student simply rely on Student column, which is not correct as per Second Normal Form. To get second normal form, it is advisable to break subjects into independent table as well as match them by student names with the help of foreign keys. Now the new student table of 2NF will show:

Student	**Age**
Nishit	15
Rohit	14
Amit	17

It is seen that the candidate key is available in Student column, where Age is based on it. So we find another Subject Table for 2NF as:

Student	**Subject**
Nishit	Biology
Nishit	Maths
Amit	Maths
Rohit	Maths

In the candidate column with having {Student and Subject, bot will meet for Second Normal Form that will continuously updated Anomalies.

Third Normal Form (3NF)

Such type of normal form is good for every non-prime table features which depends on primary key. . It is seen that such type of transitive practical dependency should be avoided in the table and should favour for Second Normal form. To understand this, consider the following fields.

Student_Detail Table:

Student_id Student_name DOB Street City State Pincode

We see that Student_id is a Primary key whereas street, city and state has to depend on Pin code. So the dependency that exists among pin code with other fields are transitive. To have 3NF, we will re-shuffle the street, city and state with another table as shown:

New student_Detail Table:

Student_id Student_name DOB Pincode

Address Table

Pincode Street City State

The benefit of moving out the transitive dependency is, Reducing of data duplication approximation.

Getting more data integrity

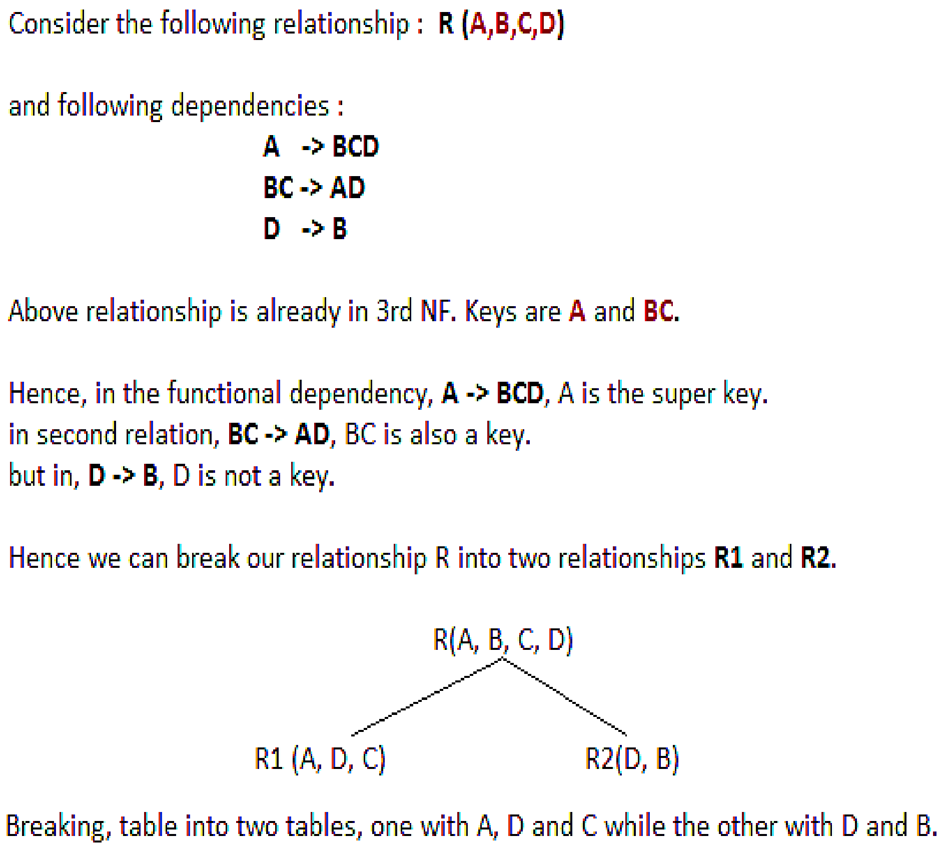
Boyce and Codd Normal Form (BCNF)

Boyce with Codd Normal Form is assumed to be the best edition in terms of Third Normal form. In this, the appearance will result with anomaly that is not managed by 3NF. . It is found that a 3NF table that has no multiple overlapping candidate keys will remained in BCNF.

The following conditions should match for table to be in BCNF:

R to remain in 3rd Normal Form

For each functional dependency (X -> Y), X to be super Key.



Consider the following relationship : **R (A,B,C,D)**

and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2.**

R(A, B, C, D)

R1 (A, D, C)

R2(D, B)

Breaking, table into two tables, one with A, D and C while the other with D and B.

Fig 2.10 Function dependency chart

Check your progress 7

1. F1, F2, F3, F4, and F5 are fields in table having functional dependencies as:

F1->F3

F2->F4

(F1,F2)->F5

Then with respect to normalization, the table will be in:

a. 1NF c. 3NF

b. 2NF d. None of the mentioned

2. In a relational schema R(A,B,C,D,E,P,G), following FDs are known:

AB->CD

DE->P

C->E

P->C

B->G

Now the relation schema R is:

a. in BCNF c. in 2NF, but not in 3NF

b. in 3NF, but not in BCNF d. not in 2NF

3. Which normal form is considered adequate for normal relational database design?

a. 2NF c. 4NF

b. 5NF d. 3NF

4. When the attribute of composite key depends on other composite, then a relation will be _____.

a. 2NF c. BCNF

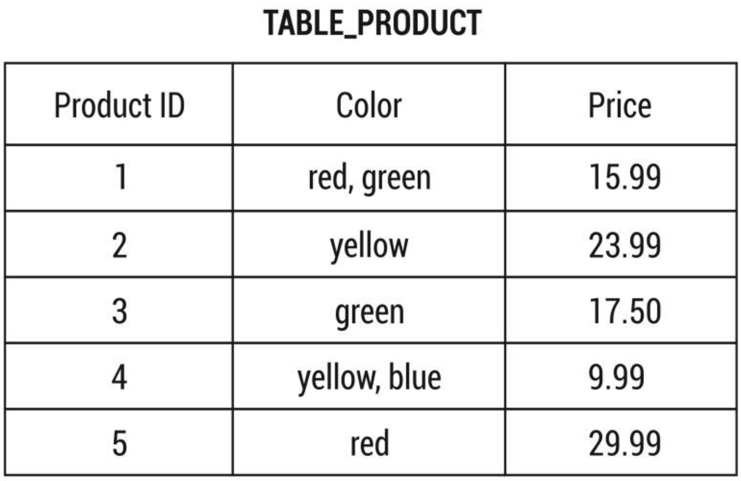
b. 3NF d. 1NF

2.9 Examples based on normalization

1st Normal Form Example

To bring about un-normalized table in first normal form, consider an example shown:

Table: 1st Normal Form

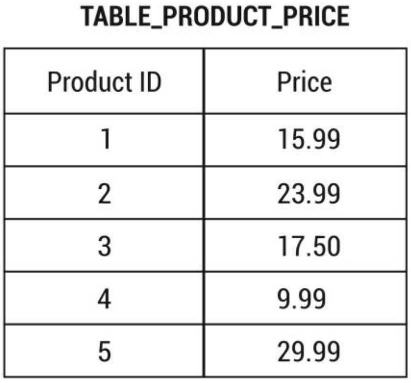


TABLE_PRODUCT

Product ID	Color	Price
1	red, green	15.99
2	yellow	23.99
3	green	17.50
4	yellow, blue	9.99
5	red	29.99

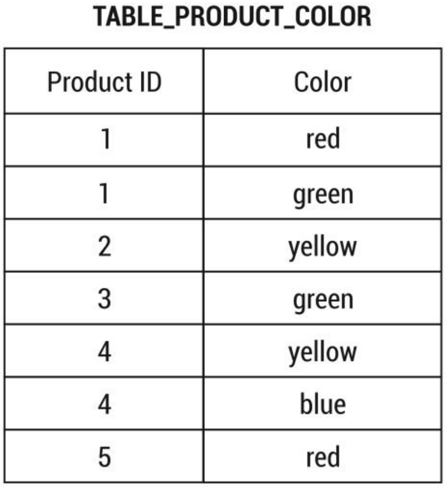
This table is not in first normal form as the [Color] column carries various values. If you see the first row that has values "red" and "green." Now to bring this table to first normal form, you have to distribute the table in two tables as:

Table Distributed tables



TABLE_PRODUCT_PRICE

Product ID	Price
1	15.99
2	23.99
3	17.50
4	9.99
5	29.99



TABLE_PRODUCT_COLOR

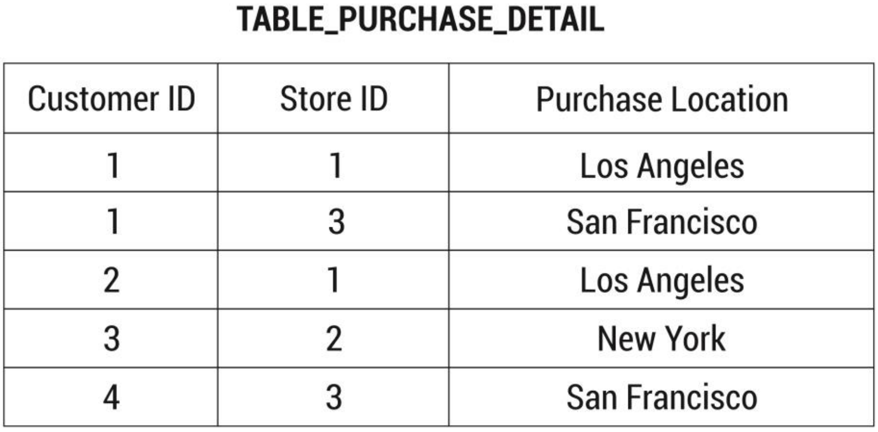
Product ID	Color
1	red
1	green
2	yellow
3	green
4	yellow
4	blue
5	red

Now first normal form is satisfied, as the columns on each table all hold just one value.

2nd Normal Form Example

Consider the following table:

Table: Normal Form



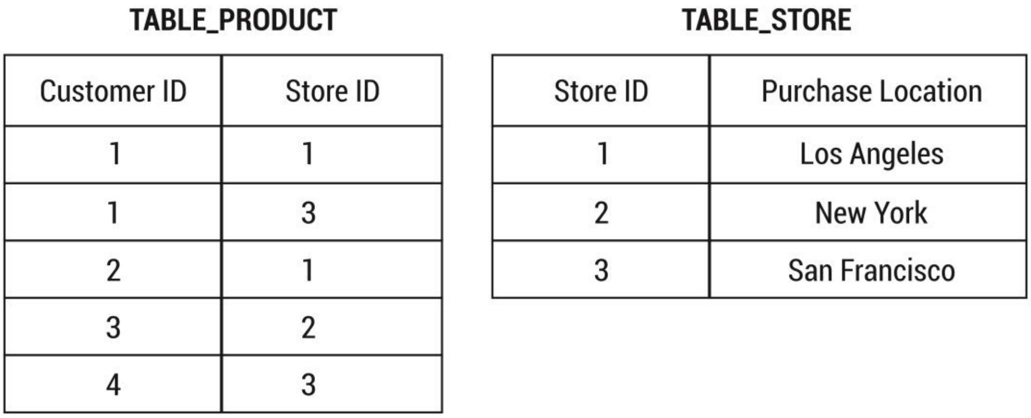
TABLE_PURCHASE_DETAIL

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

This table has a compound primary key [Customer ID, Store ID]. The non-key characteristic is [Purchase Location]. In this situation, [Purchase Location] merely depends on [Store ID], which is no more than single part of primary key.

As a result, this table does not please second normal form. To carry this table to second normal form, we rupture the table into two tables, further more as:

Table: Data



TABLE_PRODUCT

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

TABLE_STORE

Store ID	Purchase Location
1	Los Angeles
2	New York
3	San Francisco

What we have done is to take away the partial functional reliance that we originally have. At this time, in the table [TABLE_STORE], the column [Purchase Location] is completely needy on the primary key of that table, which is [Store ID].

Check your progress 8

1. In ________ normal form, a composite attribute is converted to individual attributes.

   a. First c. Third

   b. Second d. Fourth

2.10 Let Us Sum Up

In this unit we have studied that normalization is an important measurement of a database development process which shows a real look of database and displays working of data which will helped for particular database.

With the method of normalization we can change relation schema in order to decrease the laying-off. It is seen that all normalization period will give extra relations available in database.

It is studied that database normalization is used to lower the redundancy as well as dependency. It will probably help in dividing huge table into smaller ones so to possess low redundancy and to have better relationship among data.

In this unit, we have seen that decomposition is used to change relation by means of collecting smaller relations. As seen, use of decomposition in database results in breaking of tables into various tables and reaching to higher normal form.

Full functional dependency take place when you previously meet the needs for a functional dependency along with the set of attributes lying on the left side of the functional dependency statement cannot be condensed beyond

2.11 Answers for Check Your Progress

Check your progress 1

Answers: (1-a)

Check your progress 2

Answers: (1-a)

Check your progress 3

Answers: (1-b)

Check your progress 4

Answers: (1-c), (2-a)

Check your progress 5

Answers: (1-c)

Check your progress 6

Answers: (1-b)

Check your progress 7

Answers: (1-a), (2-d), (3-d), (4-b)

Check your progress 8

Answers: (1-a)

2.12 Glossary

1. **Normalization -** A process where data in relational construct is organized to lower redundancy and non-relational constructs.
2. **Physical model -** Data modelling level where you add database and database management system (DBMS) specific modelling information such as tables, columns, and data types.
3. **Primary key -** It is an attribute or attributes that uniquely identify an instance of an entity.
4. **Referential integrity -** It is an assertion where a foreign key values in an instance of child entity corresponds to values in parent entity.

2.13 Assignment

Collect the information about the value of normalizing a table, if normalization is done from second normal form to third normal form?

2.14 Activities

Why do I need to use a subquery for this SQL query?

2.15 Case Study

In data mining and statistical data analysis, when do you need to normalize data and why is it important to do so?

2.16 Further Readings

1. Dependency Structures of Data Base Relationships by W. W. Armstrong.
2. Cardinal numbers and Formalized Mathematics by GrzegorzBancerek.
3. The fundamental properties of natural numbers by GrzegorzBancerek.
4. Finite sequences and tuples of elements of a non-empty sets by CzesławByliński

Block Summary

While studying this block, the user will achieve knowledge and understanding about Database Integrity Concepts and the idea of Normalization. The block explains about Domain Integrity constraints, Referential Integrity constraints and Entity Integrity constraints with certain examples that will help the user to grab the concept.

The block explains about basic understanding of various integrity concepts and ideas of normalization that will make effective for making database. The features of decomposition in database are well explained through diagrams and illustration. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts

Block Assignment

Short Answer Questions

1. What are Domain Integrity constraints?
2. What is Entity Integrity constraint?
3. What is Functional Dependency?
4. Explain Entity Integrity constraints?
5. Is database normalization important?

Long Answer Questions

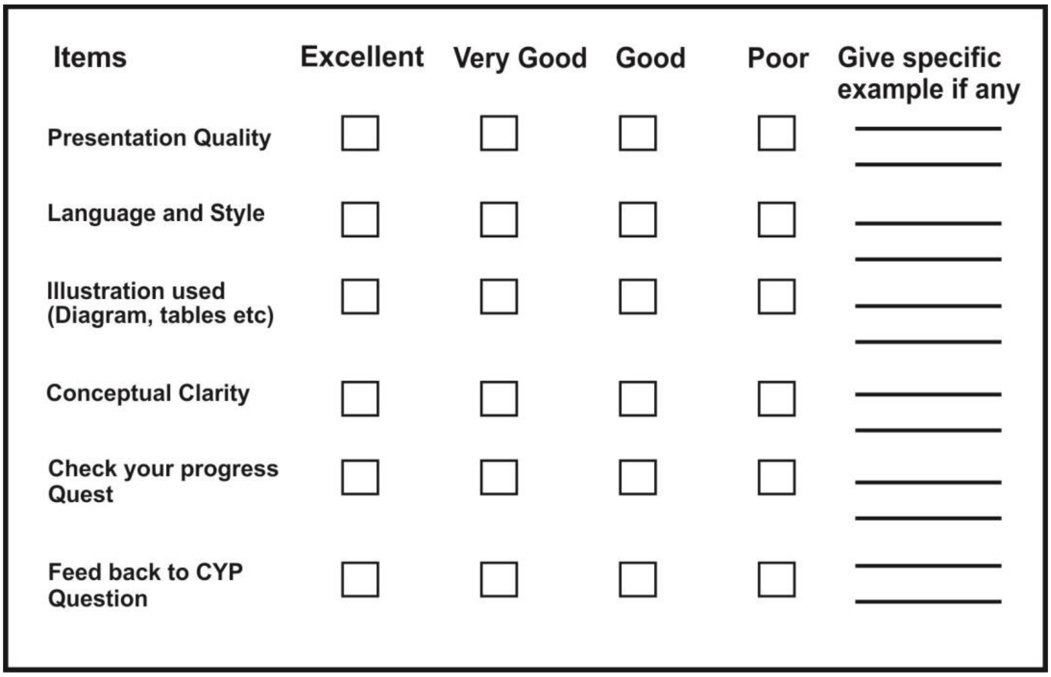
1. Explain the rules of normalization?
2. Explain Primary key and unique key?
3. How do you standardize a monthly distribution to account for the different length of each month?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	
Language and Style	□	□	□	□	
Illustration used (Diagram, tables etc)	□	□	□	□	
Conceptual Clarity	□	□	□	□	
Check your progress Quest	□	□	□	□	
Feed back to CYP Question	□	□	□	□	

3. Any Other Comments

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

………………………………………………………………………………………………

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

स्वाध्यायः परमं तपः

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

RELATIONAL DATABASE MANAGEMENT

PGDCA 103

BLOCK 3:

RELATIONAL ALGEBRA AND QUERY LANGUAGE

Dr. Babasaheb Ambedkar Open University
Ahmedabad

RELATIONAL DATABASE MANAGEMENT

KMRO
KNOWLEDGE MANAGEMENT & RESEARCH ORGANIZATION

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Ms. Katherine Abraham

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

RELATIONAL DATABASE MANAGEMENT

Contents

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

UNIT 1 INTRODUCTION TO DATABASE SYSTEM

Basic Concepts of Data and Database, Database management system, Data Administrator and its functions, Advantages of DBMS over file systems, Applications of DBMS, Three-tier architecture of DBMS, Data Models, Components of DBMS, Overview of Languages of DBMS (DDL, DML, DCL)

UNIT 2 CONCEPTUAL MODELLING

Introduction, Basic Concept of Entity-Relationship Diagram, Relationship concept and its types (unary, binary and ternary), Cardinalities (one-one, one-many, many-one and many-many), Overview of Extended ER model, Case Study of Extended ER model

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

UNIT 1 DATABASE INTEGRITY CONCEPTS

Introduction, Domain Integrity constraints, Entity Integrity constraints, Referential Integrity constraints

UNIT 2 NORMALIZATION

Introduction, Basic Concept of designing, Need of normalization, Decomposition (lossy and lossless), Functional Dependency, Full Functional Dependency, Armstrong Axioms of Functional Dependencies, Normalization Rules (1NF, 2NF, 3NF and BCNF), Examples based on normalization

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

UNIT 1 **RELATIONAL ALGEBRA**

Introduction, Basic concepts of relational algebra, symbols of relational algebra, operations on relational algebra, Examples based on relational algebra

UNIT 2 **QUERY LANGUAGE**

Introduction, Introduction to Structured Query Language, Data Definition Language (DDL), Data Manipulation Statement (DML), Data types in MySQL, Operators (Arithmentic, Comparison, Logical), SQL Functions, Group function (AVG, MAX, MIN, SUM and COUNT), COMMIT and ROLLBACK

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

UNIT 1 **DATA RETRIEVAL SQL STATEMENT**

Introduction, Single table query without condition, Single table query with condition, Group by clause, Orders by Clause, Self Join Natural Join and Sub query, Examples based on SQL concepts

UNIT 2 **TYPES OF DATABASE SYSTEMS**

Introduction, Centralized Database System, Parallel Database Systems, Distributed Database Systems, Client-Server Database System

Dr. Babasaheb
Ambedkar
Open University

PGDCA 103

RELATIONAL DATABASE MANAGEMENT

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

UNIT 1
RELATIONAL ALGEBRA 03

UNIT 2
QUERY LANGUAGE 18

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

Block Introduction

Relational database mechanisms convey query language which serves its consumers to query the database instances. The database delivers relational algebra as well as relational calculus as its query languages. Relational algebra endures a mechanical query language, which abducts instances of connections as input as well as accepts instances of connections as output. It exercises operators to conduct queries.

In this block, we will learn and study about relational algebra and query languages in SQL. We will cover the topics related to Structured Query Language, Data Definition Language and Data Manipulation Statement along with different operators. The student will be given with the knowledge about different rules and criteria's of query languages.

The block will focus on basic understanding about relational algebra concepts and explains more on the concept of Query Language and its requirement in making database. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Block Objective

After learning this block, you will be able to understand:

- The concepts of relational algebra
- About symbols of relational algebra
- About operations on relational algebra
- About Introduction to Structured Query Language
- About Data Definition Language (DDL) and Data Manipulation Statement (DML)
- About various Data types in MySQL
- Concept of Operators

- Concept of SQL Functions
- Various Group function

Block Structure

Unit 1: Relational Algebra

Unit 2: Query Language

UNIT 1: RELATIONAL ALGEBRA

Unit Structure

1.0 Learning Objectives

1.1 Introduction

1.2 Basic Concepts of Relational Algebra

1.3 Symbols of Relational Algebra

1.4 Operations on Relational Algebra

1.5 Examples Based on Relational Algebra

1.6 Let Us Sum Up

1.7 Answers for Check Your Progress

1.8 Glossary

1.9 Assignment

1.10 Activities

1.11 Case Study

1.12 Further Readings

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- Features of Relational algebra
- Operations of relational algebra

1.1 Introduction

Relational database mechanisms are apprehended to be composed with a query language that conserve contribute its consumers to query the database instances. There are dual categories of query languages − relational algebra as well as relational calculus. Relational algebra subsists a disciplined query language, which collects instances of relations as input furthermore conforms instances of relations as output. It exercises operators to conduct queries. An operator can be coupled unary or binary. They acknowledge relations as their input as well as agree relations as their output. Relational algebra is appeared recursively on a correlation as well as intermediate consequences are additionally examined relations.

1.2 Basic Concepts of Relational Algebra

Relation algebra is a formal structure consisting of sets and operations on those sets. It is used for manipulating relations. The operations of this algebra include the usual set operations (since relations are sets of tuples), and special operations defined for relations

- Select
- Project
- Union
- Set different
- Cartesian product
- Rename

Features

- Like general algebraic expression4+5*x+y, excluding numbers, we use relations as values along with operations as well as operators.
- Query language is not used in DBMSs.
- In this, inner or lower level operations of relational DBMS exists.
- Advance SQL queries are needed to open relational algebra operations mostly outer join.
- In this, a relation exists as sets of tuples without duplicates.

- You can instruct the DBMS of your requirement but cannot amend its working. It shows step by step representation of result.

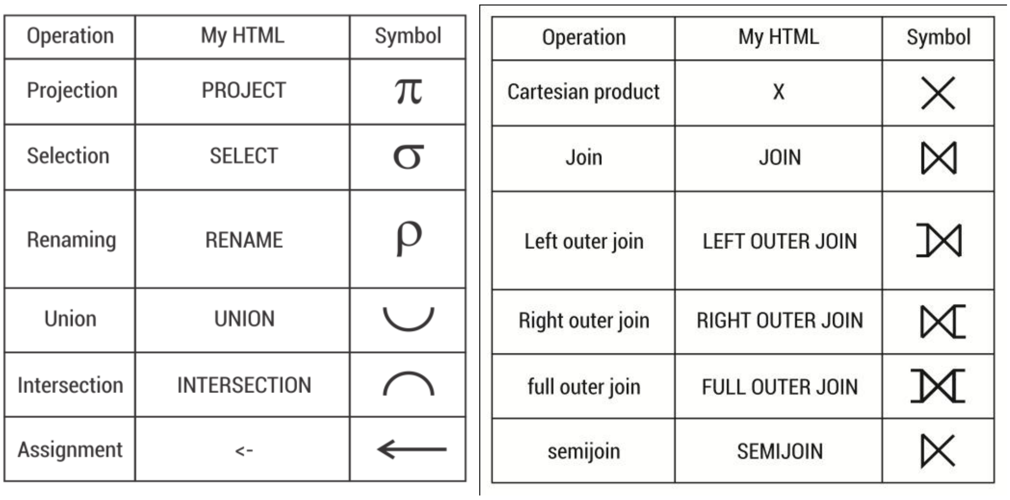
Check your progress 1

1. The term _______ is used to refer to a row.

a. Attribute	c. Field
b. Tuple	d. Instance

1.3 Symbols of Relational Algebra

Relational algebra carries varied symbols that have its own operations. Some of symbols do not support the browsers such as:



Operation	My HTML	Symbol
Projection	PROJECT	π
Selection	SELECT	σ
Renaming	RENAME	ρ
Union	UNION	∪
Intersection	INTERSECTION	∩
Assignment	<-	←

Operation	My HTML	Symbol
Cartesian product	X	×
Join	JOIN	⋈
Left outer join	LEFT OUTER JOIN	⟕
Right outer join	RIGHT OUTER JOIN	⟖
full outer join	FULL OUTER JOIN	⟗
semijoin	SEMIJOIN	⋉

Fig 1.1 Relational Algebra Symbol Chart

So finally we can write an expression in relational algebra as:

PROJECT$_{Namn}$(SELECT$_{Medlemsnummer< 3}$ (Medlem))

If we use symbols than the above expression will be:

πNamn (σMedlemsnummer<3 (Medlem))

Cartesian product

It is found that the Cartesian product of 2 tables will combine each row in a table with each row of another table.

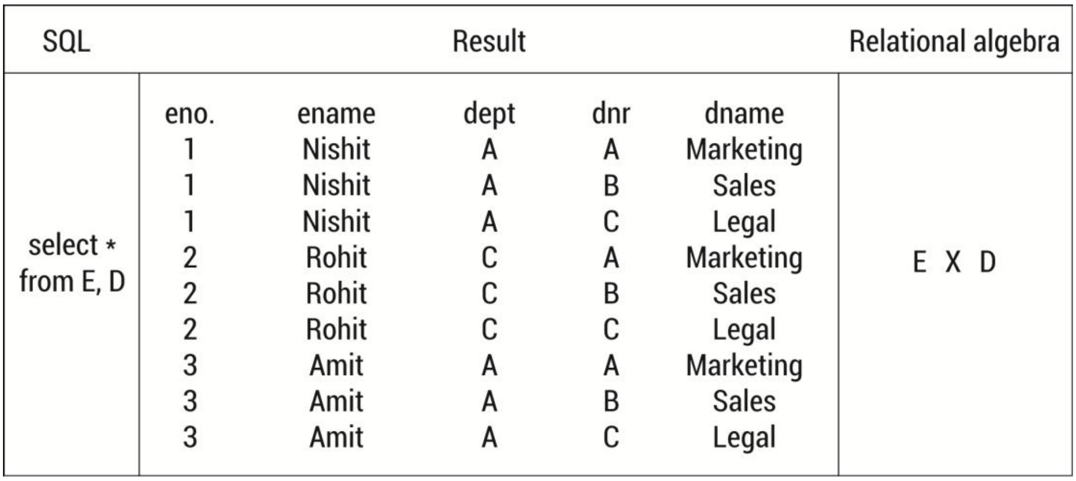
Example: The table E (for EMPLOYEE)

Enr	Ename	dept
1	Nishit	A
2	Rohit	C
3	Amit	A

Example: The table D (for DEPARTMENT)

Dnr	Dname
A	Marketing
B	Sales
C	Legal

The result in SQL will be:



SQL	Result					Relational algebra
	eno.	ename	dept	dnr	dname	
select * from E, D	1	Nishit	A	A	Marketing	E X D
	1	Nishit	A	B	Sales	
	1	Nishit	A	C	Legal	
	2	Rohit	C	A	Marketing	
	2	Rohit	C	B	Sales	
	2	Rohit	C	C	Legal	
	3	Amit	A	A	Marketing	
	3	Amit	A	B	Sales	
	3	Amit	A	C	Legal	

Fig 1.2 Output of SQL

Check your progress 2

1. Which among the following operation uses -, to locate for tuples which is obtained in single relation and not in others.

   a. Union c. Difference

   b. Set-difference d. Intersection

1.4 Operations on relational algebra

Activities in the Relational Data simulation are described by Relational Algebra. Relational Algebra subsists not subjugated on acknowledges, furthermore somewhat depends on the efficiency of approximations in the tuple. We see that there are five basic operations in relational algebra:

1. Union
2. Difference
3. Cartesian Product
4. Projection
5. Selection

If we consider a union of relations as R and S, then we can write:

$$R\ US = \{<x_1, \ldots, x_n> : <x_1, \ldots, x_n> \in R \mathrm{V} <x_1, \ldots, x_n> \in \mathrm{S}\}.$$

From the expression, we see that this is a set of tuples having R and S. We can use such operation which is in relations with same parity; consequently each and every tuples in the result have similar number of components.

Now we see for the difference of relations of R and S as:

$$R - S = \{<x_1, \ldots, x_n> \in R : <x_1, \ldots, x_n> \in \mathrm{S}\}.$$

From the expression, we see that this is a set of tuples which is in R and not in S, so we feel that R and S results in similar parity.

Now we will study for the cartesian product of relations in R and S as:

$$R\ \mathrm{x}S = \{<x_1, \ldots, x_n, y_1, \ldots, y_m> : <x, \ldots, x_n> \in R\ \Lambda <y_1, \ldots, y_m> \in \mathrm{S}\}.$$

If we think that relation R has parity n and relation S has parity m, then the relation R x S is examined as set of all possible (n+m)tuples which forms first n components form tuple in relation R and last m tuple in relation S. The result has the parity which can be written as n+m.

Since from the symbol table, the projection is indicated as π, so by assuming a relation R of parity k, we can write:

$$\pi_{i_1, i_2, \ldots, i_m}(R)$$

The expression will result that the projection of R is on top of components

$i_1,i_2,\ldots.i_m$

where

ij: different integers in range 1, ….k.

Now if the set of m tuples is written as $(a_1,a_2,\ldots\ldots a_m)$ in a similar way that there exists k-tuple in$(b_1, b_2,\ldots,b_n)$ in R for which $a_j=b_{ij}$ for j=1,2,….,m.

Consider an expression $\pi_{4,2}\,(R)$. If we find this by taking tuple t from R and forms a 2-tuple set from 4th as well as 2ndcomponent of t, then as an alternative of locations, characteristic names can also be applied.

If the relation scheme R(A,B,C,D) is shown for above relation, then by using the symbols, we can write the projection as:

$\pi_{D,B}\,(R)$

So, the resultant relation Q can be explained with the relation scheme as Q(D,D).

Here we see that the selection is represented as:

$\sigma_F(R)$

From the above selection where R is a relation and F is a formula, then: \$^{i}

- Operands are component numbers where component i is shown as,
- It uses arithmetic operators like $<, =, >, \geq, \leq, \neq$
- The use of logical operators such as ∧ (and), ∨ (or), ¬ (not) are also applied.

We see that apart from these, there are various algebraic operations such as:

- Intersection
- Quotient
- Join
- Natural Join

Intersection:

If R and S are relations, then the intersection can be written as:

$R \cap S = R - (R - S)$

Quotient:

If R and S are relations, then the quotient is written as R/S.

If there exists a parity r and s in relations R and S, and r>s with s≠0, so the quotient shows the set of all (r - s)tuples in (a_1, $a_{2,\dots,}a_{r-s}$) .

So, s-tuples is($a_{r-s+1,\dots,}a_{r)}$ in S and tuple is (a_1, $a_{2,\dots,}a_r$)in *R*, so we can write quotient of relation R and S as:

R $\% S = \pi 1,2,\dots,_{r-s}(R) - \pi_{1,2,\dots,r-s}((\pi_{1,2,\dots,r-s}(R) \text{ x } S) - R)$

Join:

Join or Ө-Join of relation R and S on columns i and j is written as: Λ

From the above:

θ= comparison operator

r = parity of R

Equijoin

We can from above expression that Ө and r are tuples in the Cartesian product of relations R and S where $i^{\text{-th}}$ component of relation R stands with Ө to j^{th} component of Relation S. So if θ is '$=$' then the operation is known as Equijoin.

Natural Join

It is different case of a Join which is applied when both relations R and S contains a column which are named by features.

If we think A_1, A_2, ….A_k being characteristics names applied for relations R and S, then we can explain Natural Join as:

$R \bowtie S = \pi_{i1,i2,\dots,im}\sigma R,A_1 = S.A_1 \; \Lambda \; R,A_k = S.A_k \; (R \text{ x } S)$

From the expression:

i_1, ….i_m = list of all components of relations RXS, excluding components s.A_1, …..S.A_k.

Semijoin

We see that a semijoin of relation R and S can be written as R >< S which is the projection on top of quality of R of natural join of R and S:

$$R >< S = \pi_R (R \bowtie S)$$

In the expression:

R= list of attributes of R

Selection:

This operation will find a group of tuples that is relational part and to get only such tuples. Selection will choose only tuples which follows particular predicate or circumstances. It is explained on individual relation and is represented as σ.

The table below shows details of particular book as:

Acc-no	Yr-pub	Title
734216	2007	Programming Principles
237235	2008	Structural Design
631523	2008	Electronic Devices Circuits
543211	2007	Switching Circuits
376112	2008	Engineering Drawing

We can see from the above table that , the relation showing books printed in year 2008 will be written as σ Yr-pub=2008. If from the relation of books, if we want to locate for books with Acc-no is greater than equal to 56782, then we have σ Acc-no>=56782(Book)

Projection:

In projection, the operation will comes back its argument relation by considering certain attributes. It is a kind of unary operation which is represented by Π. If we consider the above table and see for the total titles and Acc-no of Book relation, then we can write this as Π Acc-no, Title (Book)

Union:

This operation is applied, if someone needs attributes which can be shown on one or both the relations. This is represented as U. Consider an example of Bank of Maharashtra, Baroda Branch, where details of customer with following records were examined:

Borrower (customer_name, loan_number)

Depositor (customer_name, account_number)

Customer (customer_name, house_number, customer_city)

The details about bank customers having an account or loan or both, the operation can be specified as:

∏ customer_name (Borrower) U ∏ customer_name (Depositor)

In case of union operation, the condition, r U s to be true, there exists two **factors:**

The relation r and s must be of the same arity, i.e., they must have the same number of attributes. The domains of the ith attribute of r and the ith attribute of s must be the same for all i.

Difference:

In case of difference operation, it is applied to locate tuples present in single relation and not in other. It is written as –. If you want to locate the names of customers having a simple saving account without any loan, then you can write this as:

∏ customer_name (Depositor) - ∏ customer_name (Borrower)

Division:

Mostly the division operation is represented as ÷. If a(A. and b(B. be the relation, then a ÷ b will show limitations of tuples in a having different attribute names in relation to A, so the Header of a and not b, that results in tuples in b will be there in a. Consider a table "a"

X	Y
a	1
b	2
a	2
p	3
p	4

In case of table "b":-

Y
2
3

So a ÷ b

X
b
a
p

Cartesian product

This type of operation will allow to join information obtained from two relations. It can be written as a X b where a and b are relations. In the following relation "a":-

A	B
a	1
b	2
a	2

See another relation "b" :-

B	C
3	1a
2	2b

So we write a Xb as:-

r.A	r.B	s.B	s.C
a	1	3	1a
a	1	2	2b
b	2	3	1a
b	2	2	2b
a	2	3	1a
a	2	2	2b

We see that if relation a having n1 tuples and relation b having n2 tuples, then a X b will result in n1*n2 tuples.

Check your progress 3

1. _________is the the unary operation:

a. Selection　　c. Projection

b. Primitive　　d. Generalized selection

2. Which joined condition carries an equality operator:

a. Equijoins　　c. Natural

b. Cartesian　　d. Left

1.5 Examples based on relational algebra

Now suppose we need some information about cars that got manufactured in 1996 mode, and were detected with some faults during the inspection in the year 1999. To solve such problems:

First find the information about cars by taking the average values of all attributes of relation Car. In this, the information about inspections is kept and written in the inspection table and faults are kept in another registered table. So with this, we require 3 tables to find the information.

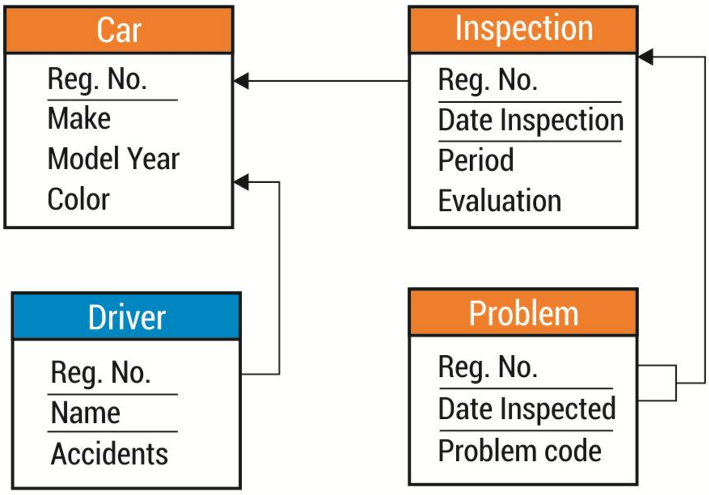


Fig 1.3 Value of quality Model Year

From the above figure, we see that cars of 1996 model are publicized as value of quality ModelYear in table shown in fig 1.3. We see tuples that shows cars that is manufactured in year 1996 with the help of selection.

C1996 = $\sigma_{\text{ModelYear=1996}}$ (Car)

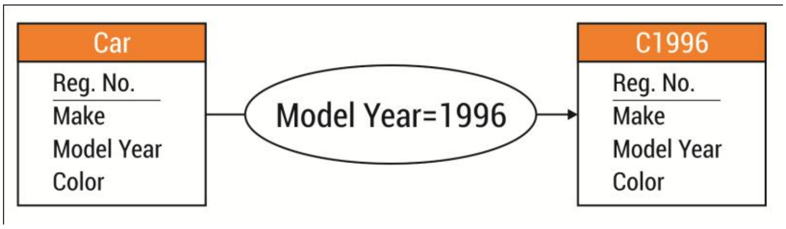


Fig 1.4 Value of quality Model Year

From the fig 1.4, we find that cars that are inspected in manufacturing year 1999, finds their selection as:

In1999 = $\sigma_{\text{Period=1999}}$ (Inspection)

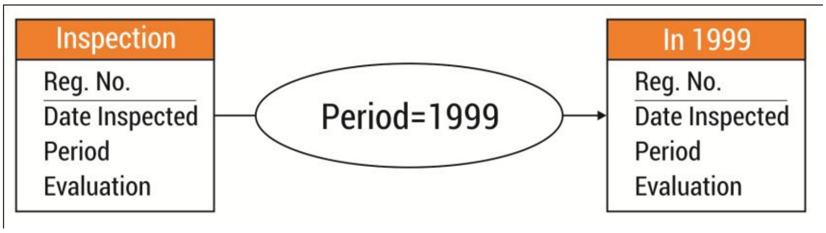


Fig 1.5 Inspection table

From the fig 1.5, there are 2 tables with manufacturing year 1999 that shows description of cars. While doing that a join operation is used. Applying natural join.

CI = C1966 * In1999 = $\sigma_{\text{ModelYear=1996}}$ (Car) * $\sigma_{\text{Period=1999}}$ (Inspection)

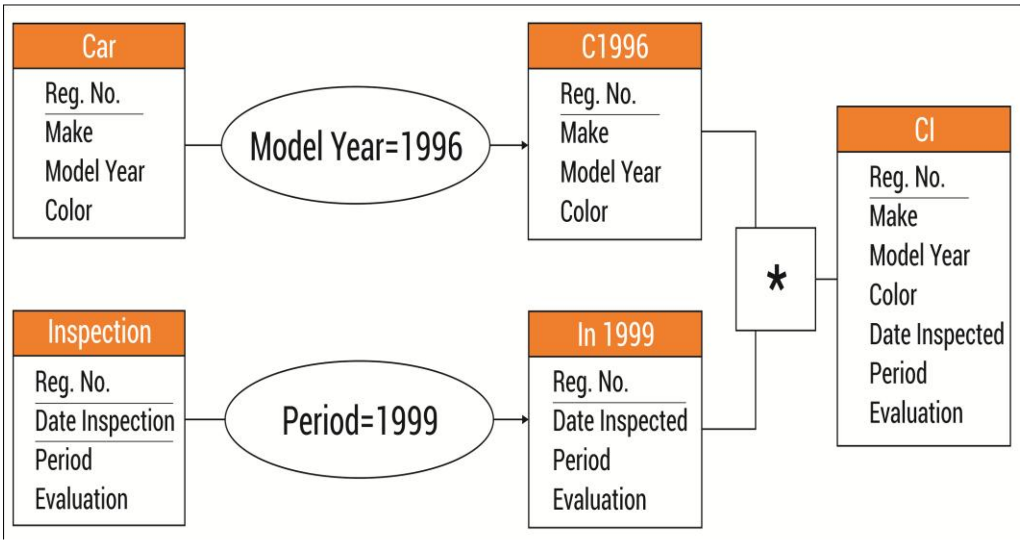


Fig 1.6 Natural Join

CIP = CI * Problem

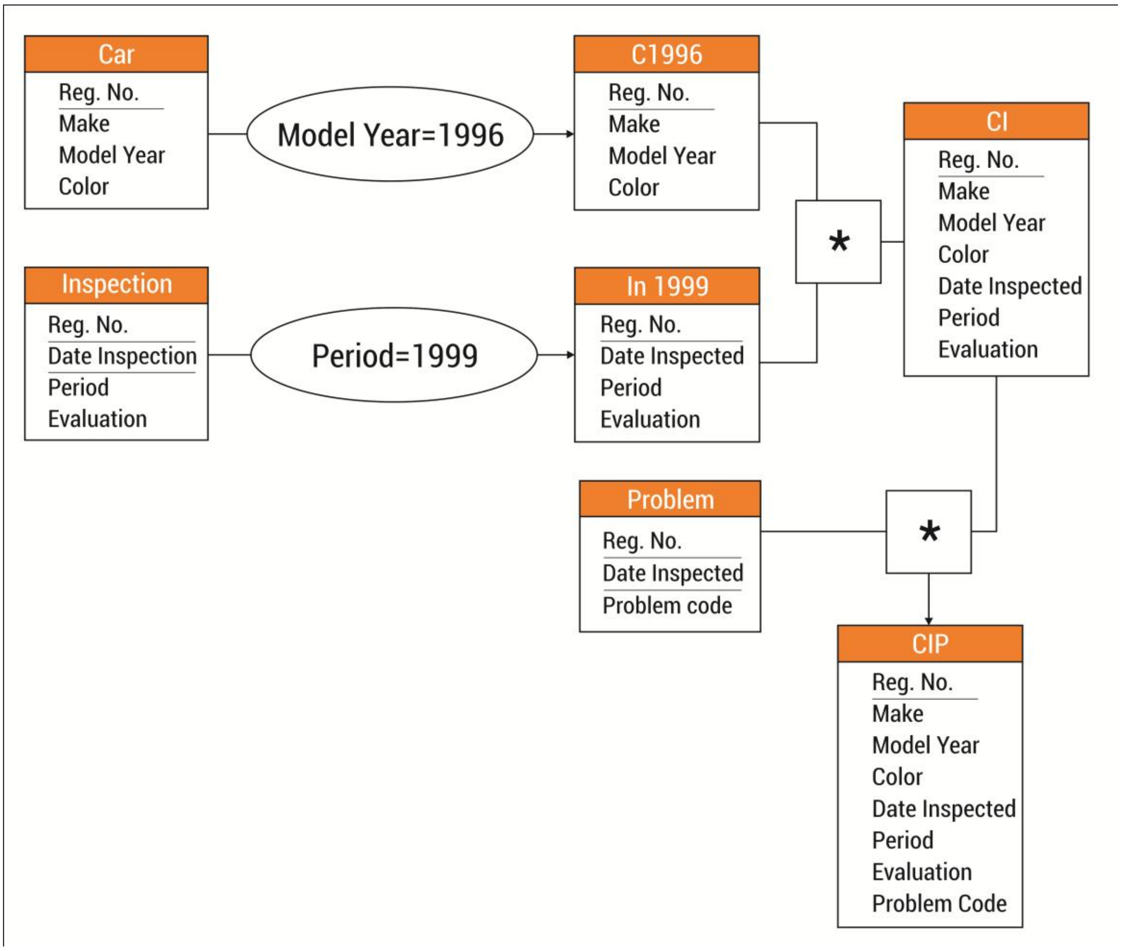


Fig 1.7 Natural Join chart

FaultyCar =

π RegNo,Make,ModelYear,Color (

($\sigma_{\text{ModelYear=1996}}$ (Car) * $\sigma_{\text{Period=1999}}$ (Inspection)) * Problem)

Check your progress 4

1.The common column is eliminated in

a. theta join c. natural join

b. outer join d. composed join

1.6 Let Us Sum Up

In this unit we have learnt that a relational algebra is a query language, which collects instances of relations as input furthermore conforms instances of relations as output. It exercises operators to conduct queries. An operator can be coupled unary or binary.

Relational Algebra depends on the efficiency of approximations in the tuple. There are five basic operations in relational algebra such as Union, Difference, Cartesian Product, Projection and Selection.

Further, relational algebra continues as methodical query language which abides practiced recursively on a relation as well as mediation results are additionally contemplated relations. It is a formal structure consisting of sets and operations on those sets.

1.7 Answers for Check Your Progress

Check your progress 1

Answers: (1-b)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-d), (2-a)

Check your progress 4

Answers: (1-c)

1.8 Glossary

1. **Attribute -** Represents a type of characteristic or property associated with a set of real or abstract things

2. **Denormalization -**To allow data redundancy in a table to improve query performance.

3. **Entity -** It shows a set of real or abstract things having common attributes or characteristics.
4. **Schema-** The structure of a database.

1.9 Assignment

Collect information on sigma and pi operations?

1.10 Activities

Are symbols a shortcut to solve relational algebra problems?

1.11 Case Study

Write details on relational algebra function that you can see in daily life?

1.12 Further Readings

1. Relational algebra in DBMS by Rasmus Ejlers Møgelberg.
2. The set-based relational algebra by J Van den Bussche.

Relational Algebra
Query Language

UNIT 2: QUERY LANGUAGE

Unit Structure

2.0 Learning Objectives

2.1 Introduction

2.2 Introduction to Structured Query Language

2.3 Data Definition Language (DDL)

2.4 Data Manipulation Statement(DML)

2.5 Data Types In Mysql

2.6 Operators (Arithmetic, Comparison, Logical)

2.7 SQL Functions

2.8 Group Function

2.9 COMMIT and ROLLBACK

2.10 Let Us Sum Up

2.11 Answers for Check Your Progress

2.12 Glossary

2.13 Assignment

2.14 Activities

2.15 Case Study

2.16 Further Readings

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- Concept of Rollback Command
- Basic about Group Functions
- Features of SQL functions
- Idea about SQL Arithmetic Operators

2.1 Introduction

Characteristically, visualizations look ahead to data in various particular forms. For instance, a pie chart possibly will imagine data in two columns: a text label as well as a numeric value. The data surrounded by data resource possibly will not accurately match such structure. For illustration, the data resource might have additional two columns; otherwise the order of the columns possibly will not be equivalent to the order accepted by pie chart. The query language makes available the capability to forward data manipulation in addition to formatting requests towards the data source, moreover guarantee that the returned data structure as well as contents will match the expected structure.

The sentence structure of the query language is comparable to SQL. Developers well-known with SQL must be able to promptly learn as well as use such query language. We see that the data sources are not essential to put into practice the query language, to execute all description of the language.

2.2 Introduction to Structured Query Language

SQL also termed as Structured Query Language is a measure of interactive as well as programming language for accomplishment of information from along with updating of database. Even though SQL is together with an ANSI as well as an ISO standard, lots of database products maintain SQL with proprietary expansion to the standard language. Queries acquire the shape of a command language to facilitate in selecting, inserting, updating and even finding the location of data, further more. It carries a programming interface.

SQL was initially formed by IBM in 1975 in addition to be known as SEQUEL which means "Structured English Query Language." In view of the fact that, it has undergone a numeral of changes among several comes from Oracle products. Nowadays, SQL is frequently carried for Web database development as well as in management. Despite the fact, SQL is currently adopting to be a standard language, conveniently as many variations of it exists as MySQL as well as MySQL. On applying a scripting language such as PHP, SQL commands can be carried out at the time when a Web page is to be loaded. This put together a possibility to produce dynamic Web pages that can facilitate to display various information at every time when they load.

Features:

- Executes queries against a database
- Retrieves data from a database
- Insert records in a database
- Update records in a database
- Delete records from a database
- Create new databases
- Create new tables in a database
- Create stored procedures in a database
- Create views in a database
- Set permissions on tables, procedures, and views

There are some common SQL Commands such as:

- SELECT –This command will extracts data from a database
- UPDATE - This command will update the data in a database
- DELETE - This command will deletes data from a database
- INSERT INTO - This command will inserts new data into database
- CREATE DATABASE - This command will generate a new database
- ALTER DATABASE - This command will adjust a database
- CREATE TABLE - This command will create new table
- ALTER TABLE - This command will alter a table
- DROP TABLE - This command will deletes a table
- CREATE INDEX - This command will create an index search key
- DROP INDEX - This command will delete an index

Check your progress 1

1. Which command is used to extract data from database?

a. SELECT

b. UPDATE

c. DELETE

d. INSERT

2.3 Data Definition Language (DDL)

Data Definition Language (DDL) is functional to produce as well as wipe out databases along with certain database objects. These commands are applied to database administrators during the period when you have to grouper want to quit various phases of database project. It is a vocabulary that is applied to explain data structures in SQL Server. By using such statements, you can create, alter or drop data structures where SQL Server is used.

The language further defines data structures and furthermore alters the data. For illustration, a DDL command is applied to add, remove or alter tables that are present inside the database. DDLs applied in database purpose are well thought-out to be a subset of Structured Query Language. On the other hand, a DDL may also explain about several types of data like XML.

In a Data Definition Language, there is a pre-defined syntax which will illustrate data. For instance, to construct a fresh table by means of SQL syntax, the CREATE command is applied, that will accompany by parameters for table name as well as column definitions. Further, DDL also explains about the name of every column in addition to the associated data type. Now, if a table is framed, it can be altered with the help of ALTER command. If the table is not required, then DROP command will be applied in order to delete the table.

The syntax of Create table:

```
CREATE TABLE <table name> (

<attribute name 1><data type 1>,

...

<attribute name n><data type n>);
```

The syntax of Alter table:

```
ALTER TABLE <table name>

ADD CONSTRAINT <constraint name> PRIMARY KEY (<attribute list>);
```

In view of the fact, DDL is a subset of SQL, which will not cover all feasible SQL commands. For instance, commands like SELECT as well as INSERT are well thought-out part of as part of Data Manipulation Language (DML), at the same time, access commands like CONNECT as well as EXECUTE are component of the Data Control Language (DCL). The DDL, DML, and DCL languages include for the most part of commands that are supported by SQL.

CREATE

In database, if you wish to create a table, then you need to make use of SQL DDL CREATE TABLE statement, which can be written as:

```
Table (
Column1 datatype [NULL | NOT NULL],
Column2 datatype [NULL | NOT NULL],
…
[UNIQUE (column_list),]
[PRIMARY KEY (column_list)]
);
```

You have to give certain values in the above program, where brackets ([]) is not mandatory. In the program, vertical bar (|) shows presence of items either left or right or can be in both the place. Further if you want to frame the students table, then you have to create following syntax:

```
Student (
Student_id INT NOT NULL,
Full_name VARCHAR(50) NOT NULL,
Favourite_colour CHAR(10) NULL,
UNIQUE (full_name),
PRIMARY KEY (student_id)
);
```

The above syntax will creates a student table that contains columns like student_id, full_name and favourite_colour. In this the database refer the consequent column name. Here, we see that the student_id along with full_name containing NOT NULL constraints, but favourite_colour column contains no particular constraint, hence we acknowledge NULL approximations if at all, students have no colour option. It is seen that while a UNIQUE constraint will make clear that no two students carry similar complete name. So, we can analysed that the main key for table is student_id column.

ALTER

With Alter database, you can change the complete features of database. Such features are kept in db.opt file inside the database directory. If you want to use Alter database, you should apply ALTER privilege on your database. The syntax of database is:

```
ALTER {DATABASE | SCHEMA} [db_name]
Alter_specification …
```

```
ALTER {DATABASE | SCHEMA} db_name
UPGRADE DATA DIRECTORY NAME
Alter_specification:
[DEFAULT] CHARACTER SET [=] charset_name
| [DEFAULT] COLLATE [=] collation_name
```

If you for the Alter function, you will finds its syntax as:

```
ALTER FUNCTION func_name [characteristic …]
Characteristic:
COMMENT 'string'
| LANGUAGE SQL
| {CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL
DATA}
| SQL SECURITY {DEFINER | INVOKER}
```

The above statement can be employed to alter certain features of stored function where many changes will be shown in ALTER FUNCTION statement.

This statement will not be able to change parameters of particular stored function and for doing this, you should drop additionally re-create function with the help of DROP and CREATE FUNCTION.

The Alter table description is shown below:

```
ALTER [ONLINE | OFFLINE] [IGNORE] TABLE tbl_name
[alter_specification [, alter_specification] …]
[partition_options]
Alter_specification:
Table_options
|ADD [COLUMN] col_name column_definition
[FIRST | AFTER col_name ]
|ADD [COLUMN] (col_namecolumn_definition, …)
|ADD {INDEX|KEY} [index_name]
[index_type] (index_col_name,…) [index_option] …
|ADD [CONSTRAINT [symbol]]
UNIQUE [INDEX|KEY] [index_name]
[index_type] (index_col_name,…) [index_option] …
|ADD FULLTEXT [INDEX|KEY] [index_name]
(index_col_name,…) [index_option] …
|ADD SPATIAL [INDEX|KEY] [index_name]
```

```
(index_col_name,…) [index_option]…
|ADD [CONSTRAINT [symbol]]
FOREIGN KEY [index_name] (index_col_name,…)
reference_definition
|ALTER  [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
|CHANGE [COLUMN] old_col_namenew_col_namecolumn_definition
[FIRST|AFTER col_name]
|MODIFY [COLUMN] col_namecolumn_definition
[FIRST|AFTER col_name]
|DROP [COLUMN] col_name
|DROP PRIMARY KEY
|DROP {INDEX|KEY} index_name
|DROP FOREIGN KEY fk_symbol
|DISABLE KEYS
|ENABLE KEYS
|RENAME [TO|AS] new_tbl_name


|ORDER BY col_name [, col_name] …
|CONVERT TO CHARACTER SET charset_name [COLLATE collation_name]
|[DEFAULT] CHARACTER SET [=] charset_name [COLLATE [=]
collation_name]
|DISCARD TABLESPACE
|IMPORT TABLESPACE
|ADD PARTITION (partition_defination)
|DROP PARTITION partition_names
|COALESCE PARTITION number
|REORGANIZE PARTITION [partition_names INTO (partition_definitions)]
|ANALYZE PARTITION {partition_names | ALL}
|CHECK PARTITION {partition_names | ALL}
|OPTIMIZE PARTITION {partition_names | ALL}
|REBUILD PARTITION {partition_names | ALL}
|REPAIR PARTITION {partition_names | ALL}
|PARTITION BY partitioning_expression
|REMOVE PARTITIONING
Index_col_name:
        Col_name [(length)] [ASC | DESC]
Index_type:
        USING {BTREE | HASH}
Index_option:
        KEY_BLOCK_SIZE [=] value
|index_type
|WITH PARSER parser_name
table_options:
table_option [[,] table_option] …
```

DROP

The drop table statement is used when you have to remove a table from particular tabular arrangement. In case of referential integrity arrangement in the table, the dropping tables become more complex.

The idea of Drop database is to put down all the tables and erased the database. If you are applying DROP DATABASE, you need to apply DROP privilege on certain database.

If this type of database exists, then it will not allow the fault or error to appear when database is not present. When, a default database is dropped, then it will not set and DATABASE() function will again back to NULL. The file extensions are:

.BAK .DAT .HSH .MRG

.MYD .MYI .TRG .TRN

.db .frm .ibd .ndb

.par

The db.opt file, if it exists

Check your progress 2

1. Which are the not most frequently used DDL statements:

a. CREATE

b. DROP

c. ALTER

d. None of these

2.4 Data Manipulation Language (DML)

DML is a type of statement which is used applied to work with data present in tables. When you are associated to the most part of multi-user databases, then you will find effect on private copy of your tables which is not viewed by anyone besides you complete it. You have already examined the SELECT statement which is an element of DML still however it simply gets back the data to a certain extent than modifying it.

It is seen that the insert statement is applied, clearly, to add recent rows to a table as shown:

```
INSERT INTO <table name>

VALUES (<value 1>, ...<value n>);
```

We see that the comma with an enclosed list of values is required to match the table structure accurately in the numeral attributes in addition to the data type of every attribute. In this character type values are at all times covered in single quotes; where number values are not in quotes and perhaps date values are frequently in the format 'yyyy-mm-dd'.

It is found that the update statement applies to alter values that are earlier in a table as highlighted below:

```
UPDATE <table name>

SET <attribute> = <expression>

WHERE <condition>;
```

From the above, an update expression can be a constant of any computed value or even result of SELECT statement which will back a single row as well as single column. As seen, if the WHERE clause is absent, after that the particular attribute is positioned to the similar value in each row of the table. You can also set numerous attribute values at a particular time that carries a comma enclosed list of attribute=expression pairs.

We see that the delete statement will work just that, for rows in a table as shown:

```
DELETE FROM <table name>

WHERE <condition>;
```

Now, if you try to remove the WHERE clause, subsequently each row of the table gets deleted and once more, you will not obtain a “do you really want to do this?” message.

Further, if you are applying a large multiuser system, you might require making your DML changes noticeable to other users of the database. Even though this might be finished automatically when you log out, you could also just type:

COMMIT;

Now, if all your changes are pushed in commit system plus you would like to re-establish your personal copy of the database to the method it was previously with you, then you have to just type:

ROLLBACK;

Though single-user systems don’t hold up commit as well as rollback statements, they are used in large systems to control transactions, which are series of changes to the database. Transactions are regularly enclosed in higher courses.

INSERT

This command will enter data inside the table. The syntax of INSERT command is INSERT into table-name values(data1,data2,..)

If the table as shown containing the Student detail fields, then:

S_id	S_Name	age

INSERT into Student values(95121,'Nishit',18);

We see that the above mentioned command will put a record inside student table as shown:

S_id	S_Name	age
95121	Nishit	18

Also we we study the example to Insert NULL value in a column. We see that below mentioned statements will put NULL value inside age column of student table.

INSERT inside Student(id,name) values(95122,'Rohit');

INSERT inside Student values(95122,'Rohit',null);

With the above commands, we find that there exists two column value while other are set as null.

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	

We will also study the standard example about Inserting standard value to column

INSERT into Student values(95123,'Amit',default)

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	
95123	Amit	16

If the age column of student table contains default age value as16 and on running below query, you will see that insert default value will be there in age column immaterial of default value.

INSERT into Student values(95123,'Amit')

DELETE

The Delete command will erase all data present inside the table. This command can delete particular row on applying certain conditions. The syntax of DELETE command is:

DELETE from table-name;

Now, if we consider an example of Deleting all Records from a Table, then command DELETE from Student, will erase all records from Student table. Now, if you want to erase a particular particular record from the table, the Student table shown below will explain you better as:

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	20
95123	Anuj	15

If you want to DELETE from Student where s_id=95123, then above command will erase record of s_id as 95123 from Student table, shown below:

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	20

UPDATE

The update command will adjust the row in a table. Its syntax is UPDATE table-name set column-name = value where condition;

If you see the particular example of update Student set age=20 where s_id=95122, then we have table as:;

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	20
95123	Amit	16

Now we will study another example where you can update multiple columns as:

UPDATE Student set s_name='Anuj',age=15 where s_id=95123, then such command will update two columns of record as:

S_id	S_Name	age
95121	Nishit	18
95122	Rohit	20
95123	Anuj	15

Check your progress 3

1. Which is used for data retrieval from the database:

a. DDL c. SDL

b. DML d. VDL

2. What are the types of DML:

a. Low level c. Procedural DML

b. High level d. All of these

2.5 Data types in MySQL

Appropriately explaining about the fields in a table is significant to largely optimization of particular database. In such cases, you should use only type as well as size of field and should not define a field as 10 characters broad if using only 2 characters. Particularly, such types of fields are called as data types. After typing of data, you will store information in those fields. MySQL uses lots of altered data types that can be categorized as:

- Numeric
- Date and time
- String
- Spatial

Numeric Data Types:

MySQL uses all the standard ANSI SQL numeric data types, so on approaching MySQL from other database, such definitions will result in the explanation shown below:

INT – It is a normal integer sized which is signed or unsigned. In case of signed, they have range from -2147483648 to 2147483647, where as in case of unsigned, they range from 0 to 4294967295.

TINYINT – It is a tiny integer which is also signed or unsigned. In case of signed, they possess range as -128 to 127, which in case of unsigned, the range is 0 to 255.

SMALLINT – It is also a tiny integer carrying 5 digits decimal width, which are also signed or unsigned. In case of signed, their range -32768 to 32767, while in unsigned, the range is 0 to 65535.

MEDIUMINT – It is a medium sized integer having 9 digit decimal width which are signed or unsigned. In case of signed, their range is -8388608 to 8388607, while in case of unsigned, their range is 0 to 16777215.

BIGINT – It is a large integer having a decimal width of 20 digits which are signed or unsigned. In case of signed, their range is -9223372036854775808 to 9223372036854775807, while in case of unsigned, their range is 0 to 18446744073709551615.

FLOAT (M,D) – It is a floating point number which is only signed with decimal precision of 24 places. In this, a display length (M) and number of decimals (D) can be defined. It is not used moreover has a default of 10,2, where 2 is number of decimals plus 10 is total digits.

DOUBLE (M,D) – It is a type of double precision floating point number having decimal precision of 53 place. These are only signed. It explains about length (M) and number of decimals (D). It delivers a common character relevantly 16,4 where 4 continues character of decimals.

DECIMAL (M,D) – It continues an alleviated signed elevating point character. In this unpack characters acquires 1 byte which exists exercised to demonstrate length (M) as well as character of decimals (D).

Numeric Types	Description
TINYINT	A very small integer
SMALLINT	A small integer
MEDIUMINT	A medium-sized integer
INT	A standard integer
BIGINT	A large integer
DECIMAL	A fixed-point number
FLOAT	A single-precision floating-point number
DOUBLE	A double-precision floating-point number
BIT	A bit field

Date and Time Types:

In MySQL, the date and time data types are:

DATE: It is in YYYY-MM-DD format that lies from 1000-01-01 to 9999-12-31. For instance, December 20th, 1971 would be stored as 1971-12-20.

DATETIME: A date as well as time combination could be YYYY-MM-DD HH:MM:SS format which is from 1000-01-01 00:00:00 to 9999-12-31 23:59:59. For instance, 2:30 in the afternoon on September20th, 1971 would be stored as 1971-09-20 14:30:00.

TIMESTAMP: A timestamp interspersed between midnight, January 1, 1970 all along 2037. It seems to be display proceeding with DATETIME format with no imperceptible hyphens. Therefore 2:30 in afternoon on September 20th, 1971 would consequence as 19710920143000 in YYYYMMDDHHMMSS format.

TIME: It keeps the time in HH:MM:SS format.

YEAR (M): It keeps year in 2-digit or 4-digits. If length is particular as 2 as **YEAR (2),** then YEAR can be from 1970 to 2069. If length is 4, then year be 1901 to 2155. The default length is 4.

Date and Time Types	Description
DATE	A date value in 'CCYY-MM-DD' format
TIME	A time value in 'hh:mm:ss' format
DATETIME	A date and time value in 'CCYY-MM-DD hh:mm:ss' format
TIMESTAMP	A timestamp value in 'CCYY-MM-DD hh:mm:ss' format
YEAR	A year value in CCYY or YY format

String Types: In mySQL, apart from storing of data in numeric and date types format, most data can also be stored in string format as shown:

CHAR (M): It is a fixed string lies from 1-255 characters in length having rightpad to particular length after stored. It has a value of 1.

VARCHAR (M): it is a variable length string lies from 1- 255 characters in length as VARCHAR (45).

BLOB or TEXT: This is a text field with highest length of 65535 characters. It keeps big binary data like images or files.

TINYBLOB or TINYTEXT: It is a TEXT column having maximum length as 255 characters.

MEDIUMBLOB or MEDIUMTEXT: It is also a TEXT column having max length of 16777215 characters.

LONGBLOB or LONGTEXT: It is a BLOB having max length as 4294967295 characters.

ENUM: It is an inventory serving as imagine list term. In this, if sorting is done to keeping fields A, B or C, then it can be explained as ENUM ('A', 'B', 'C').

String Types	Description
CHAR	A fixed-length non-binary (character) string
VARCHAR	A variable-length non-binary string
BINARY	A fixed-length binary string
VARBINARY	A variable-length binary string
TINYBLOB	A very small BLOB (binary large object)
BLOB	A small BLOB
MEDIUMBLOB	A medium-sized BLOB
LONGBLOB	A large BLOB
TINYTEXT	A very small non-binary string
TEXT	A small non-binary string
MEDIUMTEXT	A medium-sized non-binary string
LONGTEXT	A large non-binary string
ENUM	An enumeration; each column value may be assigned one enumeration member
SET	A set; each column value may be assigned zero or more set members

Spatial Data Types

MySQL supports many spatial data types that contain various kind of geometrical and geographical values as shown in the following table:

Operators

Spatial Data Types	Description
GEOMETRY	A Spatial value of any type
POINT	A point (a pair of XY co-ordinates)
LINESTRING	A curve (one or more POINT values)
POLYGON	A polygon
GEOMETRYCOLLECTION	A collection of GEOMETRY values
MULTILINESTRING	A collection of LINESTRING values
MULTIPOINT	A collection of POINT values
MULTIPOLYGON	A collection of POLYGON values

Arithmetic

Operator	Description	Example
+	Addition – Adds values on either side of the operator	a + b will give 30

-	Subtraction – Subtracts right hand operand from left hand operand	a – b will give -10
*	Multiplication – Multiplies values on either side of the operator	a * b will give 200
/	Division – Divides left hand operand by right hand operand	b / a will give 2
%	Modulus – Divides left hand operand by right hand operand and returns remainder	b % a will give 0

Comparison

=	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(a = b) is not true
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a!= b) is true.
<>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a<>b) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a>b) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a<b) is true.
>=	Checks of the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a>=b) is not true.

Logical

Operator	Description
ALL	The ALL operator is used to compare a value to all values in another value set.
AND	The AND operator allows the existence of multiple conditions in a SQL statement’s WHERE clause.
ANY	The ANY operator is used to compare a value to any applicable value in the list according to the condition.
BETWEEN	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
EXISTS	The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria.
IN	The IN operator is used to compare a value to a list of literal values that have been specified.
LIKE	The LIKE operator is used to compare a value to similar values using wildcard operators.
NOT	The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. This is negate operator.
OR	The OR operator is used to combine multiple condition in SQL statement’s WHERE clause.
IS NU LL	The NULL operator is used to compare a value with a NULL value.

Check your progress 4

1. Numeric Data is used to store

a. Whole numbers c. Rational numbers

b. Natural numbers d. Both a. and b.

2. Which Numeric Data type has the largest range?

a. Mediumint c. Int

b. Smallint d. Tinyint

2.6 Operators (Arithmetic, Comparison, Logical)

Operators are reserved word or characters mainly applied in SQL statements where clauses are used to carry out operation(s).They are used to identify situation in SQL statement additionally serves as combination for numerous situation in a statement. There are 4 types of operators in SQL as:

- Arithmetic operators
- Comparison operators
- Logical operators
- Operators used to negate conditions

SQL Arithmetic Operators:

Assume variable a holds 10 and variable b holds 20, then:

Operator	Description	Example
+	It is an addition operator that adds values on both side	a + b = 30
-	It is subtraction operator which subtracts right operand from left	a - b =-10
*	It is a multiplication operator, which multiplies values on both sides	a * b = 200
/	It is division operator which divides left hand operand by right hand	b/a=2
%	It is modulus operator which divide left operand by right and gives remainder	b % a = 0

SQL Comparison Operators:

Operator	**Description**	**Example**
=	It checks for approximations of two operands are equivalent, if yes that time definite	(a = b) exists as not definite.
!=	It checks if approximations of operands are equal, if not that time definite	(a!= b) exists definite.
<>	It checks if approximations of operands are equal or not , if not that time definite	(a<>b) exists definite.
>	It checks if approximations of left operand is more than right operand, if yes that time definite.	(a>b) exists not definite.
<	It checks if approximations of left operand is less than right operand, if yes that time definite.	(a<b) exists definite.
>=	It checks if approximations of left operand is more than or equal to right operand, if yes that time definite	(a>=b) exists not definite.
<=	It checks if approximations of left operand is less than or equal to right operand, if yes that time definite	(a<=b) exists definite
!<	It checks if approximations of left operand exists not less than value of right operand, if yes that time definite	(a!<b) is not definite
!>	It checks if approximations of left operand is not greater than right operand, if yes that time definite	(a!>b) exists definite

SQL Logical Operators:

Here is a list of all the logical operators available in SQL.

Operator	Description
ALL	The operator compares a value to all values in another value set.
AND	The operator shows presence of multiple conditions in statement's WHERE clause.
ANY	The operator compares value to any applicable value in list as per conditions.
BETWEEN	The operator searches for values which are in set of values with minimum and maximum value.
EXISTS	The operator searches for row in particular table fulfilling certain criterias.
IN	The operator evaluates a value with factual values that are specific.
LIKE	The operator compares a value with similar values by wildcard operators.
NOT	The Operator reverses use of logical operator.
OR	The operator joins mixture of situation in SQL statement's WHERE clause.
IS NU LL	The operator is used to assess a value with a NULL value.
UNIQUE	The operator explores each row of specific table for uniqueness.

Check your progress 5

1. In SQL the spaces at the end of the string are removed by _______ function.

a. Upper

b. String

c. Trim

d. Lower

2. _____ operator is used for appending two strings.

a.&

b. %

c. ||

d. _

2.7 SQL Functions

SQL contains many built-in functions to do calculations on data. These are:

SQL Aggregate Functions

SQL aggregate functions return a single value, calculated from values in a column.

Operator	Description
AVG()	This function will returns the average value
COUNT()	This function will returns amount of rows
FIRST()	This function will returns initial approximations
LAST()	This function will returns last approximations
MAX()	This function will return highest value
MIN()	This function will returns lowest approximations
SUM()	This function will returns total

SQL Scalar functions

SQL scalar functions return a single value, based on the input value.

Operator	Description
UCASE()	Coverts a field to upper case
LCASE()	Converts a field to lower case
MID()	Extract characters from a text field
LEN()	Returns the length of a text field
ROUND()	Rounds a numeric field to the number of decimals specified
NOW()	Returns the current system date and time
FORMAT()	Formats how a field is to be displayed

SQL Functions

Single row function

Such type function can be:

- character functions
- numeric functions
- date functions
- conversion functions

The single row functions carry out data items and require one/many input arguments that can be worked on individual row. In this, the argument is in column, literal or an expression.

These functions are used to choose SELECT statement as well as WHERE and ORDER BY clause.

These types of functions can be:

- General functions
- Case Conversion functions

Date function

This type of operation will return the date or numeric values. You will find the following functions under such operations:

MONTHS_BETWEEN

ADD_MONTHS

NEXT_DAY

LAST_DAY

ROUND and TRUNC

Further we will see that the above functions will:

MONTHS_BETWEEN:	Function returns the count of months between the two dates.
ADD_MONTHS:	Function adds ‘n’ number of months to an input date.
NEXT_DAY:	Function returns the next day of the date specified.
LAST_DAY	Function returns last day of the month of the input date.
ROUND and TRUNC:	Functions are used to round and truncates the date value.

Character function

The character function will input the character and returns back the character value. This function category will include:

CONCAT
SUBSTR
LPAD and RPAD
REPLACE
LENGTH
INSTR
TRIM

The functions of the above category can be:

CONCAT: function concatenates two string values.

LENGTH: function returns the length of the input string

SUBSTR function returns a portion of a string from a given start point to an end point

INSTR: function returns numeric position of a character or a string in a given string

LPAD and RPAD: functions pad the given string upto a specific length with a given character

TRIM: function trims the string input from the start or end.

REPLACE: function replaces characters from the input string with a given character

In the SELECT query, the idea about the CONCAT function that can concatenate two string approximations are shown:

```
SELECT CONCAT (first_name, last_name)
FROM employees
WHERE rownum < 5;
CONCAT (FIRST_NAME, LAST_NAME)
Nishit Mathur
Rohit Mathur
Amit Charan
Anuj Charan
```

Numeric function

The numeric or number function will take numeric input and gives out numeric values. The numeric function can be:

*ROUND

*TRUNC

*MOD

It is seen that the ROUND and TRUNC functions will round off and truncates the number approximation, whereas the MOD will return the remainder of division operation exists among two numbers.

Check your progress 6

1. Which SQL function Returns the current system date and time?

a. MID()	c. ROUND()
b. LEN()	d. NOW()

2.8 Group function

AVG() Function

The AVG() function returns the average value of a numeric column.

SQL AVG() Syntax

SELECT AVG(column_name) FROM table_name

Examples:

SELECT AVG(Price) AS PriceAverage FROM Products;

The Price Average will be 28.866363636363637

SELECT ProductName, Price FROM Products

WHERE Price>(SELECT AVG(Price) FROM Products);

COUNT() Function

It returns the number of rows which matches definite criterion.

SQL COUNT(column_name) Syntax

It will returns number of values of particular column:

SELECT COUNT(column_name) FROM table_name;

SQL COUNT(*) Syntax

This function will call back the number of records in a table:

SELECT COUNT(*) FROM table_name;

SQL COUNT(DISTINCT column_name) Syntax

This function will call back the number of different principles of particular column:

SELECT COUNT(DISTINCT column_name) FROM table_name;

Example:

Consider an order table:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10265	7	2	1996-07-25	1
10266	87	3	1996-07-26	3
10267	25	4	1996-07-29	1

If an SQL statement is used, then you have to count number of orders from "Customer ID"=7 available in the orders table:

Now we see that

SELECT COUNT (CustomerID) AS OrdersFromCustomerID7 FROM Orders

WHERE CustomerID=7;

OrdersFromCustomerID7 is 4

MAX() Function

The MAX() function returns the largest value of the selected column.

The Syntax of SQL MAX() function is:

SELECT MAX(column_name) FROM table_name;

Example:

Consider a Products table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24-12 oz bottles	19
3	Aniseed Syrup	1	2	12-550 ml bottles	10
4	Chef Anton’s	2	2	48-6 oz jars	21.35
5	Chef Mix	2	2	36 boxes	25

The following SQL statement gets the largest value of the "Price" column from the "Products" table:

SELECT MAX(Price) AS Highest Price FROM Products;

We see that the Highest Price is 263.5

MIN() Function

The MIN() function returns the smallest value of the selected column.

The syntax of SQL MIN() is:

SELECT MIN(column_name) FROM table_name;

Example:

Consider a Product table:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10x20 bags	18
2	Chang	1	1	24-12 bottles	19
3	Aniseed Syrup	1	2	12-550 bottles	10
4	Cajun Season	2	2	46-6 oz jars	21.35
5	Gumbo Mix	2	2	36 boxes	25

From the table find the smallest value of the "Price" column by:

SELECT MIN(Price) AS Smallest Order Price FROM Products;

Smallest Order Price is 2.5

SUM() Function

The SUM() function returns the total sum of a numeric column.

The Syntax of SUM() is:

SELECT SUM(column_name) FROM table_name;

Example:

Consider a table:

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

From the following SQL statement finds the sum of all the "Quantity" fields for OrderDetail:

SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;

TotalItemsOrdered is 12743

Check your progress 7

1.The __________function returns the smallest value of the selected column

a. MIN()	c. COUNT()
b. MAX()	d. AVG()

2.9 COMMIT and ROLLBACK

Commit and Rollback are commands that are used to control the transactions. A transaction is a sector of work that is carried out in oppose to database. They exist as series of work carried out to perform logical task either manually or by using certain database programs..

Transaction involves several changes in a database which involves addition, deletion or updation. Transaction can be controlled to make data working normal and to reduce any errors, if exists.

There are certain commands which are used to handle transactions such as:

- COMMIT: To Save the changes.
- ROLLBACK: To roll back the changes.
- SAVEPOINT: creates points within groups of transactions in which to ROLLBACK
- SET TRANSACTION: Places a name on a transaction.

COMMIT Command:

Such transaction command will save the changes made by transaction in the database. The syntax of such command is COMMIT. It safe guard the database by applying COMMIT or ROLLBACK commands.

Example:

The details shown in CUSTOMERS table with carry certain details of customer records as:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

If the records are erased from row having age as 25 and on applying COMMIT in database, then the syntax could be written as:

```
SQL> DELETE FROM CUSTOMERS

WHERE AGE = 25;

SQL>COMMIT
```

We see that the two rows from the table will get removed and the result of the SELECT statement will show:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
3	kaushik	23	Kota	2000.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

ROLLBACK Command:

It is a type of transactional command which is required to undo transactions that was not earlier saved to database.

This command will only be employed for undo transactions as last COMMIT or ROLLBACK command was forwarded.

The syntax is ROLLBACK;

Example:

Consider the CUSTOMERS table having the following records:

```
|ID   |NAME     |AGE   |ADDRESS     |SALARY     |
| 1   |Ramesh   | 32   |Ahmedabad   |2000.00    |
| 2   |Khilan   | 25   |Delhi       |1500.00    |
| 3   |kaushik  | 23   |Kota        |2000.00    |
| 4   |Chaitali | 25   |Mumbai      |6500.00    |
| 5   |Hardik   | 27   |Bhopal      |8500.00    |
| 6   |Komal    | 22   | MP         |4500.00    |
| 7   |Muffy    | 24   |Indore      |10000.00   |
```

In an example shown, the records are removed from table with age showing 25 and on using ROLLBACK in database, we see:

```
SQL> DELETE FROM CUSTOMERS

WHERE AGE = 25;

SQL> ROLLBACK;
```

We find that the delete operation will not impact the table additionally, the SELECT statement shows:

```
|ID   |NAME     |AGE   |ADDRESS     |SALARY     |
| 1   |Ramesh   | 32   |Ahmedabad   |2000.00    |
| 2   |Khilan   | 25   |Delhi       |1500.00    |
| 3   |kaushik  | 23   |Kota        |2000.00    |
| 4   |Chaitali | 25   |Mumbai      |6500.00    |
| 5   |Hardik   | 27   |Bhopal      |8500.00    |
| 6   |Komal    | 22   | MP         |4500.00    |
| 7   |Muffy    | 24   |Indore      |10000.00   |
```

Check your progress 8

1. Which command is not used to control the transactions?

   a. COMMIT

   b. ROLLBACK

   c. SELECT

   d. SAVEPOINT

2.10 Let Us Sum Up

In this unit we have learnt that a query language will transfers data to be carried out for formatting requests towards data source which will guarantee of the returned data structure and contents as per expected structure.

It is studied that a Data Definition Language is a standard syntax which will illustrate a data. It works by means of SQL syntax as CREATE command which is used by parameters for table name as well as column definitions.

Operators used in SQL statement's statements uses clauses for operation(s).They are used to identify situation in SQL statement additionally serves as combination for numerous situation in a statement. There are 4 types of operators in SQL as Arithmetic operators, Comparison operators, Logical operators and Operators used to negate conditions

The sentence structure of the query language is comparable to SQL. An SQL is Structured Query Language which measure interactivity and programming language for doing information by updating of database.

Data Definition Language (DDL) is applied to generate as well as wipe out databases as well as database objects

2.11 Answers for Check Your Progress

Check your progress 1

Answers: (1-a)

Check your progress 2

Answers: (1-d)

Check your progress 3

Answers: (1-b), (2-d)

Check your progress 4

Answers: (1-d), (2-c)

Check your progress 5

Answers: (1-c), (2-c)

Check your progress 6

Answers: (1-d)

Check your progress 7

Answers: (1-a)

Check your progress 8

Answers: (1-c)

2.12 Glossary

1. **SELECT -** A command that extracts data from a database
2. **UPDATE -** A command which will update the data in a database
3. **DELETE -** A command that deletes data from a database
4. **INSERT INTO -** A command which insert new data into database

2.13 Assignment

Data Definition Language (DDL) is applied to generate as well as wipe out databases as well as database objects? Comment.

2.14 Activities

State the various features of Query languages and compare?

2.15 Case Study

Write a draft showing the usability of MIN and MAX functions used in our daily life.

2.16 Further Readings

1. Structured Query Language: A Practical Introduction by Akeel I. Din.

2. The Structured Query Language by Jon Kabat-Zinn.

Block Summary

In this block, the student has understood and learned more on relational algebra and query languages in SQL. The block has given a full introduction on various topics such as structured query language, data definition language and data manipulation statement with different operators. The students have given more knowledge on different rules and criteria's of query languages.

The block focuses on basic understanding about relational algebra concepts and explains more on implementation of database. The concept of Query Language with examples and case studies were explained to the students. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Block Assignment

Short Answer Questions

1. Which symbol is used to denote the selection operation in relational algebra?
2. Which relational algebra operations do not require the participating tables to be union-compatible?
3. Explain the use of Data Definition Language?
4. Write short note on data types in MySQL?
5. What is Numeric Data Types?

Long Answer Questions

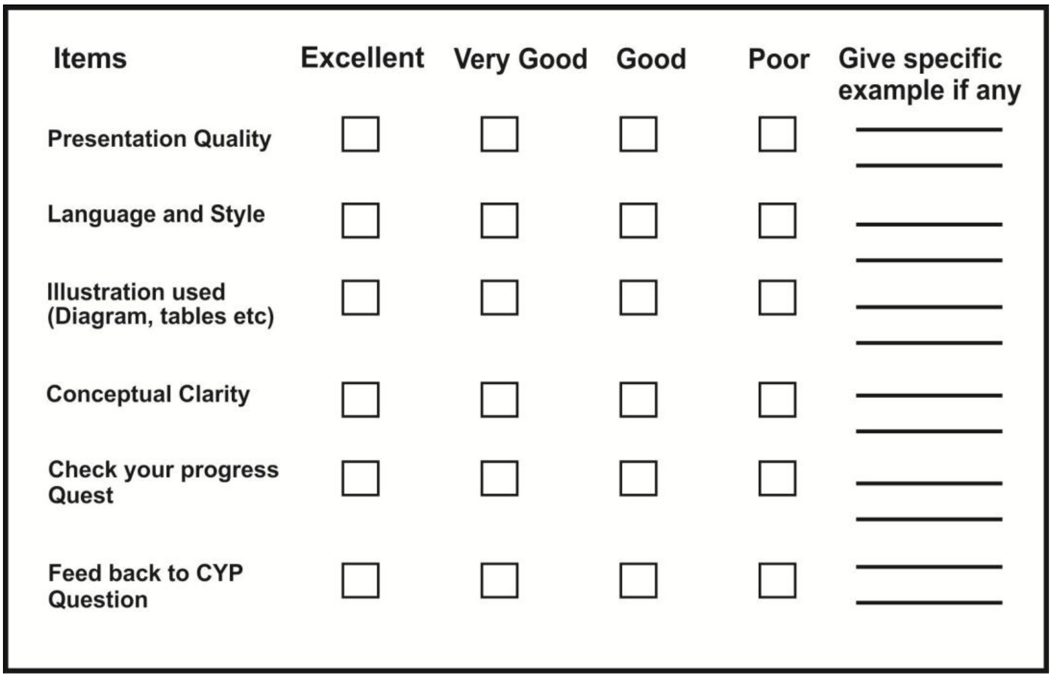
1. What is the function of COMMIT command?
2. Write the features of relational algebra?
3. What are the different Operators in SQL?

Enrolment No. □

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

………………………………………………………………………………………………

Relational Algebra
Query Language

> *Education is something which ought to be brought within the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.

RELATIONAL DATABASE MANAGEMENT

PGDCA 103

BLOCK 4:

DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

Dr. Babasaheb Ambedkar Open University
Ahmedabad

RELATIONAL DATABASE MANAGEMENT

Knowledge Management and
Research Organization
Pune

Editorial Panel

Author
Er. Nishit Mathur

Language Editor
Ms. Katherine Abraham

Graphic and Creative Panel
Ms. K. Jamdal
Ms. Lata Dawange
Ms. Pinaz Driver
Ms. Ashwini Wankhede
Mr. Kiran Shinde
Mr. Prashant Tikone
Mr. Akshay Mirajkar

Acknowledgment
Every attempt has been made to trace the copyright holders of material reproduced in this book. Should an infringement have occurred, we apologize for the same and will be pleased to make necessary correction/amendment in future edition of this book.
The content is developed by taking reference of online and print publications that are mentioned in Bibliography. The content developed represents the breadth of research excellence in this multidisciplinary academic field. Some of the information, illustrations and examples are taken "as is" and as available in the references mentioned in Bibliography for academic purpose and better understanding by learner.'

ROLE OF SELF INSTRUCTIONAL MATERIAL IN DISTANCE LEARNING

The need to plan effective instruction is imperative for a successful distance teaching repertoire. This is due to the fact that the instructional designer, the tutor, the author (s) and the student are often separated by distance and may never meet in person. This is an increasingly common scenario in distance education instruction. As much as possible, teaching by distance should stimulate the student's intellectual involvement and contain all the necessary learning instructional activities that are capable of guiding the student through the course objectives. Therefore, the course / self-instructional material are completely equipped with everything that the syllabus prescribes.

To ensure effective instruction, a number of instructional design ideas are used and these help students to acquire knowledge, intellectual skills, motor skills and necessary attitudinal changes. In this respect, students' assessment and course evaluation are incorporated in the text.

The nature of instructional activities used in distance education self-instructional materials depends on the domain of learning that they reinforce in the text, that is, the cognitive, psychomotor and affective. These are further interpreted in the acquisition of knowledge, intellectual skills and motor skills. Students may be encouraged to gain, apply and communicate (orally or in writing) the knowledge acquired. Intellectual-skills objectives may be met by designing instructions that make use of students' prior knowledge and experiences in the discourse as the foundation on which newly acquired knowledge is built.

The provision of exercises in the form of assignments, projects and tutorial feedback is necessary. Instructional activities that teach motor skills need to be graphically demonstrated and the correct practices provided during tutorials. Instructional activities for inculcating change in attitude and behavior should create interest and demonstrate need and benefits gained by adopting the required change. Information on the adoption and procedures for practice of new attitudes may then be introduced.

Teaching and learning at a distance eliminates interactive communication cues, such as pauses, intonation and gestures, associated with the face-to-face method of teaching. This is particularly so with the exclusive use of print media. Instructional activities built into the instructional repertoire provide this missing interaction between the student and the teacher. Therefore, the use of instructional activities to affect better distance teaching is not optional, but mandatory.

Our team of successful writers and authors has tried to reduce this.

Divide and to bring this Self Instructional Material as the best teaching and communication tool. Instructional activities are varied in order to assess the different facets of the domains of learning.

Distance education teaching repertoire involves extensive use of self-instructional materials, be they print or otherwise. These materials are designed to achieve certain pre-determined learning outcomes, namely goals and objectives that are contained in an instructional plan. Since the teaching process is affected over a distance, there is need to ensure that students actively participate in their learning by performing specific tasks that help them to understand the relevant concepts. Therefore, a set of exercises is built into the teaching repertoire in order to link what students and tutors do in the framework of the course outline. These could be in the form of students' assignments, a research project or a science practical exercise. Examples of instructional activities in distance education are too numerous to list. Instructional activities, when used in this context, help to motivate students, guide and measure students' performance (continuous assessment)

PREFACE

We have put in lots of hard work to make this book as user-friendly as possible, but we have not sacrificed quality. Experts were involved in preparing the materials. However, concepts are explained in easy language for you. We have included may tables and examples for easy understanding.

We sincerely hope this book will help you in every way you expect.

All the best for your studies from our team!

RELATIONAL DATABASE MANAGEMENT

Contents

BLOCK 1: INTRODUCTION TO DATABASE MANAGEMENT SYSTEM AND CONCEPTUAL MODELLING

UNIT 1 INTRODUCTION TO DATABASE SYSTEM

Basic Concepts of Data and Database, Database management system, Data Administrator and its functions, Advantages of DBMS over file systems, Applications of DBMS, Three-tier architecture of DBMS, Data Models, Components of DBMS, Overview of Languages of DBMS (DDL, DML, DCL)

UNIT 2 CONCEPTUAL MODELLING

Introduction, Basic Concept of Entity-Relationship Diagram, Relationship concept and its types (unary, binary and ternary), Cardinalities (one-one, one-many, many-one and many-many), Overview of Extended ER model, Case Study of Extended ER model

BLOCK 2: DATABASE INTEGRITY AND NORMALIZATION

UNIT 1 DATABASE INTEGRITY CONCEPTS

Introduction, Domain Integrity constraints, Entity Integrity constraints, Referential Integrity constraints

UNIT 2 NORMALIZATION

Introduction, Basic Concept of designing, Need of normalization, Decomposition (lossy and lossless), Functional Dependency, Full Functional Dependency, Armstrong Axioms of Functional Dependencies, Normalization Rules (1NF, 2NF, 3NF and BCNF), Examples based on normalization

BLOCK 3: RELATIONAL ALGEBRA AND QUERY LANGUAGE

UNIT 1 RELATIONAL ALGEBRA

Introduction, Basic concepts of relational algebra, symbols of relational algebra, operations on relational algebra, Examples based on relational algebra

UNIT 2 QUERY LANGUAGE

Introduction, Introduction to Structured Query Language, Data Definition Language (DDL), Data Manipulation Statement (DML), Data types in MySQL, Operators (Arithmentic, Comparison, Logical), SQL Functions, Group function (AVG, MAX, MIN, SUM and COUNT), COMMIT and ROLLBACK

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

UNIT 1 DATA RETRIEVAL SQL STATEMENT

Introduction, Single table query without condition, Single table query with condition, Group by clause, Orders by Clause, Self Join Natural Join and Sub query, Examples based on SQL concepts

UNIT 2 TYPES OF DATABASE SYSTEMS

Introduction, Centralized Database System, Parallel Database Systems, Distributed Database Systems, Client-Server Database System

Dr. Babasaheb
Ambedkar
Open University

PGDCA 103

RELATIONAL DATABASE MANAGEMENT

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

UNIT 1

DATA RETRIEVAL SQL STATEMENT 03

UNIT 2

TYPES OF DATABASE SYSTEMS 18

BLOCK 4: DATA RETRIEVAL SQL STATEMENT AND TYPES OF DATABASE SYSTEM

Block Introduction

SQL is a complete database language. SQL, marked Sequel or else S-Q-L, is a computer programming language applied for questioning relational databases following a nonprocedural methods. Once you take out the information from a database by means of SQL, this is called as querying the database.

In this block, we will learn and study about Data retrieval and different database systems in SQL. We will cover the topics related to Single table query, Group and Order by Clause and Natural Join and Sub query along with Centralized Database System. The student will be given with the knowledge about different rules and criteria's of data retrieval.

The block will focus on basic understanding about Database Systems concepts and explains more about Parallel Database Systems and Distributed Database Systems with more on Client-Server Database System. The students or programmers will get benefit while reading this block as it gives shortcuts and related examples that will clear all doubts.

Block Objective

After learning this block, you will be able to understand:

- Concept of Single table query.
- About Group by Clause.
- Detail of Join and Natural Join.
- Centralized Database System.
- About Parallel Database Systems.
- About Distributed Database Systems.
- About Client-Server Database System.

Block Structure

Unit 1: Data Retrieval SQL Statement: SELECT

Unit 2: Types of Database Systems

UNIT 1: DATA RETRIEVAL SQL STATEMENT: SELECT

Unit Structure

1.0 Learning Objectives

1.1 Introduction

1.2 Single table query without condition

1.3 Single table query with condition

1.4 Group by clause

1.5 Orders by Clause

1.6 Self Join Natural Join and Sub query

1.7 Examples based on SQL concepts

1.8 Let Us Sum Up

1.9 Answers for Check Your Progress

1.10 Glossary

1.11 Assignment

1.12 Activities

1.13 Case Study

1.14 Further Readings

1.0 Learning Objectives

After learning this unit, you will be able to understand:

- Basic about database language
- Understanding of ORDER BY clause
- Concept of Sub Queries
- Idea about various group syntaxes

Data Retrieval SQL Statement and Types of Database System

1.1 Introduction

SQL is a complete database language. It is a type of computer programming language which is used to ask questions from relational databases by using nonprocedural methods. Extracting information from a database results in querying of database in SQL.

A relational database is a place hooked on preparations all through the use of Relational Database Management System (RDBMS). An RDBMS bring out each and every necessity about jobs in DBMS software that will carry above besides with having ample amount of related functions which will facilitate the relational model smoother and faster to find as well as put into practice. RDBMS programmers will users controls the data with the help of particular data manipulation language. It is found that the database arrangement is constant all the way with application of data definition language. The system users makes use of several commands so as to store data and can see their data that gets inside the terminal using RDBMS on applying typing commands, otherwise it can be entered by using various graphic interface. The DBMS after that processes the commands.

1.2 Single table query without condition

If you want to locate your data file from the crunch of files by clicking and seeing each and every file, then such process at the same time is boring and lengthy. With use of Queries, all data can be seen at once which lowers tiredness and makes the searching faster as it uses techniques of searching through several tables and takes the desired information required by the user.

Check your progress 1

1. The purpose of query in database is to:

   a. remove boring work

   b. make simpler searching

   c. fast display of data

   d. all

Data Retrieval
SQL Statement:
SELECT

1.3 Single table query with condition

An expression is a mixture of mathematical or logical operators, constants, functions, as well as names of fields, controls, in addition to properties that estimate to a single value. You can make use of an expression as soon as you need data that does not exist directly in table. While selecting tables in one query and display price of a product as shown:

Categories

Category_id	parent_id
1	0
2	1
3	1
4	1
5	2
6	3

Products_to_categories

Product_id	category_id
54	0
55	2
56	2
57	2
58	3
59	3
60	4

Products

Product_id	price
54	10.50
55	11.20

```
56     | 1.00
57     | 22.20
58     | 32.0
59     | 32.0
60     | 22.0
```

From above we see that:

1. Table categories: parent_id = ‘1’ (result: 2, 3, 4)
2. Table producti_to_categories:category_id = result categories (result: 55, 56, 57, 58, 59, 60)
3. Table products: inner join or left join table product

   display price where

   product_id = result products_to_categories (result: 55, 56, 57, 58, 59, 60)

We find the output to be:

```
55 – 11.20
56 – 1.00
57 – 22.20
58 – 32.0
59 – 32.0
60 – 22.0
```

Now as per the query:

```
$sql_all = mysql_query(“SELECT cat.parent_id, cat.category_id FROM
categories cat WHERE cat.parent_id=’1’”);
While($row = mysql_fetch_array($sql_all))
{
Echo $row[‘categories_id’].<br/>’;
}
```

Check your progress 2

1.Expression contains:

a. operators	c. functions
b. constants	d. all

1.4 Group by clause

The SQL group by clause will assemble all of the rows collectively that hold data in the particular column(s) in addition to allocate aggregate functions to be carry out on the one or more columns. This can best be explicated by an example:

GROUP BY clause syntax:

```
SELECT column1,
SUM(column2)
FROM "list-of-tables"
GROUP BY "column-list";
```

Let's say you would like to retrieve a list of the highest paid salaries in each dept.:

```
SELECT max(salary), dept
FROM employee
GROUP BY dept;
```

Such statement will choose the utmost salary for the people in every exclusive department. Essentially, the salary for the person in every department will be demonstrated. There, the salary as well as their department will re-appear.

Check your progress 3

1. What is the meaning of "GROUP BY" clause in Mysql?

a. Group data by column values

b. Group data by row values

c. Both a and b

d. None of these

1.5 Order by Clause

The ORDER BY clause is applied in SELECT statement to arrange the output each in ascending or else in descending order. The syntax with SQL ORDER BY clause to sort data will be:

```
SELECT column-list

FROM table_name

[WHERE condition]

[ORDER BY column1, column2, ..columnN] [ASC | DESC];
```

Here you can apply more than one column in ORDER BY clause. In this, point has been made to ensure that no matter what column you are applying to arrange, but such column should be present in the column list.

Consider the case of customers with following records:

```
| ID   | NAME    | AGE   | ADDRESS   | SALARY|

| 001  | Rohit   | 41    | Goa       | 12000.00 |

| 002  | Karan   | 25    | Delhi     | 11500.00 |

| 003  | Nishit  | 43    | Kota      | 12000.00 |

| 004  | Amit    | 38    | Mumbai    | 16500.00 |

| 005  | Anuj    | 36    | Bhopal    | 18500.00 |

| 006  | Anika   | 22    | MP        | 14500.00 |

| 007  | Munish  | 24    | Indore    | 110000.00 |
```

If the result is to be obtained in increasing order by NAME and SALARY, then:

```
SQL> SELECT * FROM CUSTOMERS

ORDER BY NAME, SALARY;
```

It gives the output as:

```
 ID   | NAME        | AGE   | ADDRESS   | SALARY   |

| 004  | Amit        | 38    | Mumbai    | 16500.00 |

| 005  | Anuj        | 36    | Bhopal    | 18500.00 |

| 003  | Nishit      | 43    | Kota      | 12000.00 |

| 002  | Karan       | 25    | Delhi     | 11500.00 |
| 006  | Anika          | 22        | MP            | 14500.00 |
| 007  | Munish         | 24        | Indore        | 110000.00 |
| 001  | Rohit          | 41        | Goa           | 2000.00 |
```

If the result is to be obtained in descending order by NAME, then:

```
SQL> SELECT * FROM CUSTOMERS
ORDER BY NAME DESC;
```

It gives the output as:

```
| ID   | NAME    | AGE       | ADDRESS      | SALARY  |
| 001  | Rohit   | 41        | Goa          | 12000.00 |
| 007  | Munish  | 24        | Indore       | 110000.00 |
| 006  | Anika   | 22        | MP           | 14500.00 |
| 002  | Karan   | 25        | Delhi        | 11500.00 |
| 003  | Nishit  |43         | Kota         | 12000.00 |
| 005  | Anuj    | 36        | Bhopal       | 18500.00 |
| 004  | Amit    | 38        | Mumbai       | 16500.00 |
```

Check your progress 4

1. Select the appropriate option that will result while comparing clauses as GROUP BY clause similar as ORDER BY clause.

   a. Yes   c. Depends

   b. No   d. None of these

2. What is the meaning of "ORDER BY" clause in Mysql?

   a. Sorting your result set using column data   c. Both a and b

   b. Aggregation of fields   d. None of these

1.6 Self Join Natural Join and Sub query

Self- join

In SQL, self- join works with combining similar table with itself by fundamentally generating two copies of a table. On the other hand, it will show us as how to compare two dissimilar copies of table as there exists only single table. Further, it is analysed that when we perform self- join, then the table names will completely adopt the assumed name or in addition the column names will become indistinct. While analysing further, we will not be able to have an idea that which table columns can be referenced without using assumed names for duplicate names of the table. If you are unaware of the assumed names, then it is simple to use other name as prescribed in the table. It is seen that such name can be applied in the SQL query so as to reference a particular table. In such cases, we use assumed names as e1 and e2 for employee table in case of performing self- join.

If we compare other joins then there exist several conditions that can be used to do self- join. Instead of saying do a self- join without mentioning certain condition, we will apply predicate to the join. To learn more about self- join, consider an example. If in an employee table having 2 columns, one with employee name (employee_name), and other with employee location (employee_location), then:

employee_name	**employee_location**
Rohan	New Delhi
Sunil	Calcutta
Arun	Mumbai
Amit	Chennai
Aman	Bangalore

From the above arrangement, we require to trace which employees belong to similar location as employee named Rohan. The location of Rohan is New Delhi. If we think that that we cannot straight locate people who live in New Delhi, but if we apply simple query as:

```
SELECT employee_name
FROM employee
WHERE employee_location = “New Delhi”
```

On the other hand, we can write sub query as:

```
SELECT employee_name
FROM employee
WHERE employee_location in
(SELECT employee_location
FROM employee
WHERE employee_name = "Rohan")
```

This will help in visualizing final results of self- join. It is noted that a self-join is similar to any other join, where two tables are combined into single temporary table. Initially, you have to see for two different copies of employee table having their assumed names as e1 and e2 as shown in fig 1.1. If we write short column names from employee_name and employee_location to only Name and Location, then we can see assumed names table as:

e1

Name	Location
Rohan	New Delhi
Sunil	Calcutta
Arun	Mumbai
Amit	Chennai
Aman	Banglore

e2

Name	Location
Rohan	New Delhi
Sunil	Calcutta
Arun	Mumbai
Amit	Chennai
Aman	New Delhi

e1.employee_name	**e1.employee_location**	**e2.employee_name**	**e2.employee_location**
Rohan	New Delhi	Rohan	New Delhi
Aman	New Delhi	Rohan	New Delhi

Data Retrieval
SQL Statement
and Types of
Database
System

Sub Queries

In Relational databases, we find that there are certain tables which will relate to each other if both tables should possess common field. In SQL, customer and order tables corresponds as they uniformly encompass Cid column. It is possible to draw out data on or after one table through information of others. It is accepted away by associated ordinary keys of both tables.

We have to locate that which possesses an order location below the waiting position. We observe who is on the customer table in total and this relates to the pending order table. To do this, a sub query is obtained which is basically present within one another. The nested query contains pending order information while the main query has information what is looking for. So the syntax of sub queries is:

```
Select FieldName1, FieldName2 from TableName where CommonColumn Name in
(
        Select CommonColumnName from OtherTableName where ColumnName = This - Value
)
Select Cname from Customer where Cid in
(
        Select Cid from Oder where Ostatus = ‘Pending’
)
```

Natural join

Natural join is sort of binary operator that is explained mathematically such as A * B, where A and B exists as relations. We observed that the output is group of all possible combinations of tuples in A and B that will result commonly on its characteristics names. Here only a single column will turn back out of many columns.

Check your progress 5

1. The natural join is equal to:

   a. Cartesian Product

   b. Combination of Union and Cartesian product

   c. Combination of selection and Cartesian product

   d. Combination of projection and Cartesian product

2. Consider the join of a relation R with relation S. If R has m tuples and S has n tuples, then the maximum size of join is:

   a. mn

   b. m+n

   c. (m+n)/2

   d. 2(m+n)

1.7 Examples based on SQL concepts

There are many examples to show in SQL:

Example 1:

In an organisation, a survey was conducted regarding the details about the employees working. So if the employee details are marked as e_no, e_name, e_salary and d_code, if the department in which they are working are marked as d_code andd_name, then dependent can be represented as depndt_name and e_no, relation. Now the employees are to be highlight in purchase and accounts departments with single dependency.

So, we can write this as:

```
SELECT e_name FROM employee, dependent, dept

WHERE employee.d_cpd=dept.d_code

AND employee.e_no=dependent.e_no

AND d_name IN (‘ACCOUNTS’,’PURCHASE’)

GROUP BY e_name

HAVING COUNT (e_no)>=1
```

Purchase and accounts departments with single dependency

Example 2:

In an employee tables mentioned, if selection is made for rows where the Last_Name column contains Mathur as its value, then how will you explain the SELECT statement?

We will write the statement as:

SELECT * FROM Customers WHERE Last_Name='Mathur';

We see that the server back end shows the reply with result as:

```
| Cust_no        | Last_Name      | First_Name   |
| 1101           | Mathur         | Nishit       |
| 2152           | Mathur         | Rohit        |
| 2254           | Mathur         | Manish       |
```

To have the original Cust_No and First_Name columns, using certain rules can write this as:

SELECT Cust_No, First_Name FROM Customers WHERE Last_Name='Mathur';

We will see that the output will look like:

```
 | Cust_No | First_Name |
 | 1101    | Nishit     |
 | 2152    | Rohit      |
 | 2254    | Monish     |
```

To construct a WHERE clause locate in accurate matches by adding pattern matching operator as LIKE. This operator uses % symbol as wild card to compete0 or more characters. It will use underscore (_) as wild card to compete accurately single character. To show this, consider an example:

To select the First_Name and Nickname columns from the Friends table for rows in which the Nickname column contains the string "brain", use this statement:

SELECT First_Name, Nickname FROM Friends WHERE Nickname LIKE '%brain%';

The subsequent result set might look like:

```
 | First_Name          | Nickname |
| Mudit          | Baidu     |
| Sanjay         | Krishan   |
| Ashok          | Chand     |
```

Check your progress 6

1. In SQL, how to choose records from table named employee having FirstName value ending with "b"?

   a. SELECT * FROM employee WHERE FirstName='b'

   b. SELECT * FROM employee WHERE FirstName LIKE 'b%'

   c. SELECT * FROM employee WHERE FirstName LIKE '%b'

   d. SELECT * FROM employee WHERE FirstName='%b%'

1.8 Let Us Sum Up

In this unit we have studied that RDBMS programmers will control the data with the help of special data manipulation language. The database arrangement is standard application of database structures which is a fixed application of data definition language. Here the system users uses commands to store data and can see their data entering the terminal using RDBMS commands.

It found that self-join in SQL will work with combination of similar table by itself producing two copies of a table. It shows the way how to compare two dissimilar copies of table as there exists only single table. In performing self- join, the table names will completely adopt such assumed name along with column names as there are different.

1.9 Answers for Check Your Progress

Check your progress 1

Answers: (1-d)

Check your progress 2

Answers: (1-d)

Check your progress 3

Answers: (1-a)

Check your progress 4

Answers: (1-b), (2-a)

Check your progress 5

Answers: (1-d), (2-a)

Check your progress 6

Answers: (1-c)

1.10 Glossary

1. **SQL -** It is a complete database language.
2. **Expression -** These are mixture of mathematical or logical operators, constants, functions and fields.
3. **Group by clause -** It is a command which compiles row that holds data in column(s).
4. **Natural join -** A binary operator which is a set of all combinations of tuples.

1.11 Assignment

What is the output?

```
SELECT state, count(state)

FROM customers

GROUP BY state;
```

1.12 Activities

Explain about Single table query with and without condition?

1.13 Case Study

What will be the query, if you find the total amount of salary spent on each department?

1.14 Further Readings

1. Dependency Structures of Data Base Relationships by W. W. Armstrong.
2. Cardinal numbers and Formalized Mathematics by Grzegorz Bancerek.
3. The fundamental properties of natural numbers by Grzegorz Bancerek.
4. Finite sequences and tuples of elements of a non-empty sets by Czesław Byliński.

UNIT 2: TYPES OF DATABASE SYSTEMS

Unit Structure

2.0 Learning Objectives

2.1 Introduction

2.2 Centralized Database System

2.3 Parallel Database Systems

2.4 Distributed Database Systems

2.5 Client-Server Database System

2.6 Let Us Sum Up

2.7 Answers for Check Your Progress

2.8 Glossary

2.9 Assignment

2.10 Activities

2.11 Case Study

2.12 Further Readings

2.0 Learning Objectives

After learning this unit, you will be able to understand:

- About Hierarchical and Relational Databases
- Understanding of Client-Server Systems
- Features of parallel system
- Applications about distributed data system

2.1 Introduction

We observed that database is a mixture of files that can be stored or kept on computer storage devices which will handle database tables that are managed by a Database Management System (DBMS).There are various forms of DBMS products such as

- Relational
- Network
- Hierarchical

For the most part DBMS today is Relational Database Management Systems (RDBMS).Lots of Data Base Management Systems can be in touch directly with the help of programming languages like COBOL while others propose their individual programming language for functioning mutually with the database. Numerous DBMS applications additionally provide cover age as well as query tools to observe data in database.

Hierarchical Databases

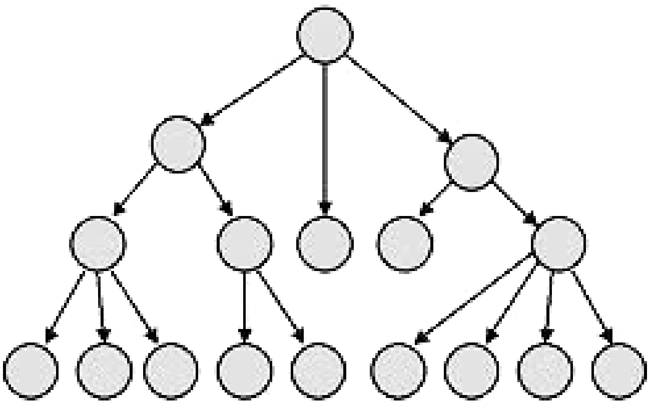


Fig 2.1 Hierarchical structure

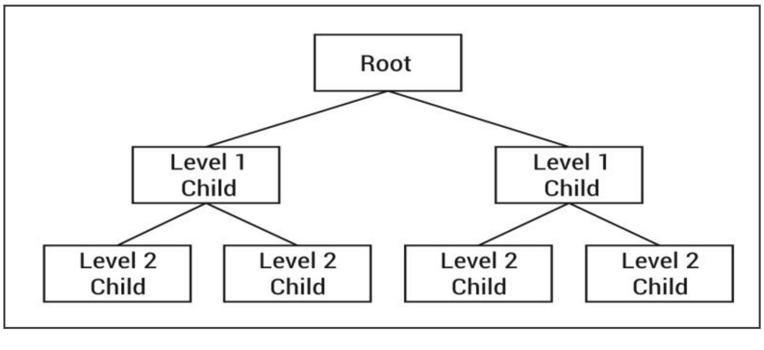


Fig 2.2 Hierarchical Databases

Network Database

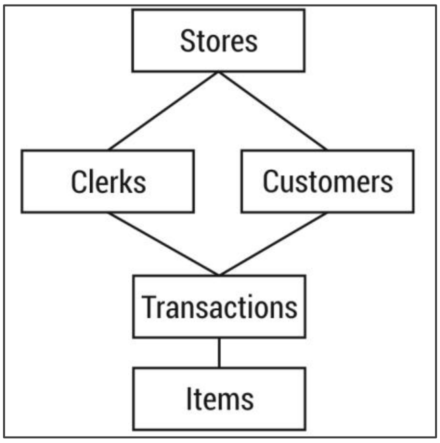


Fig 2.3 Network Databases

Relational Databases

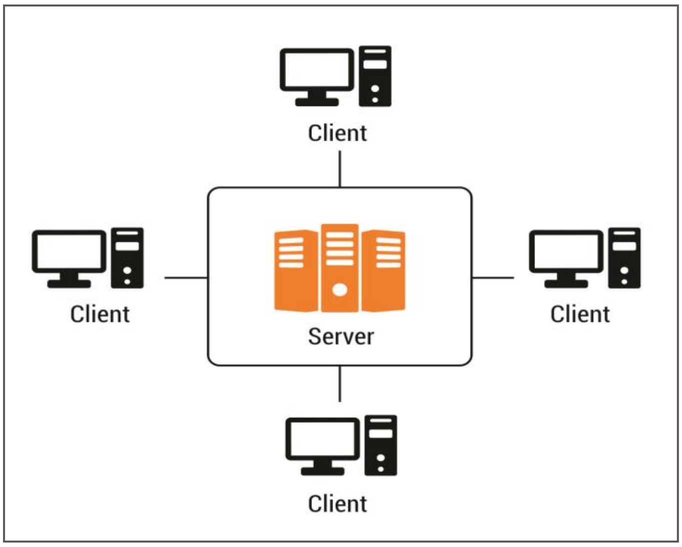


Fig 2.4 Relational Databases

2.2 Centralized Database System

The centralised database system runs on a single computer system furthermore do not interact with other computer systems. It is a general purpose computer system which carries one to few CPUs as well as number of device controllers which are joined with a common bus which gives admission to shared memory. It is also called as single user system which is a type of personal computer or workstation having particular desk-top unit, one user as well as single CPU along with hard drives installed with operating system.

It is also sometimes work as multiuser system which carries many disks, several memories, having many CPUs with multiuser Operating System by allowing many users to be connected to a system by means of different terminals. Such an arrangement is called as server systems.

The centralised system is designed to offer operations like:

- Processing
- Enforcement
- Image/video processing
- Data management tracking
- Reporting and statistics generation

These systems offer full integration for hybrid systems, with centralized control and monitoring. Moreover, it eases the road operator staff's workload by providing management tools such as point-of-sales and CRM applications. Finally, the system's database is designed to offer high performance, easy extensibility and expandability, and at the same time robustness in events such as network outages.

Fig 2.5 shows an arrangement of centralised computer system:

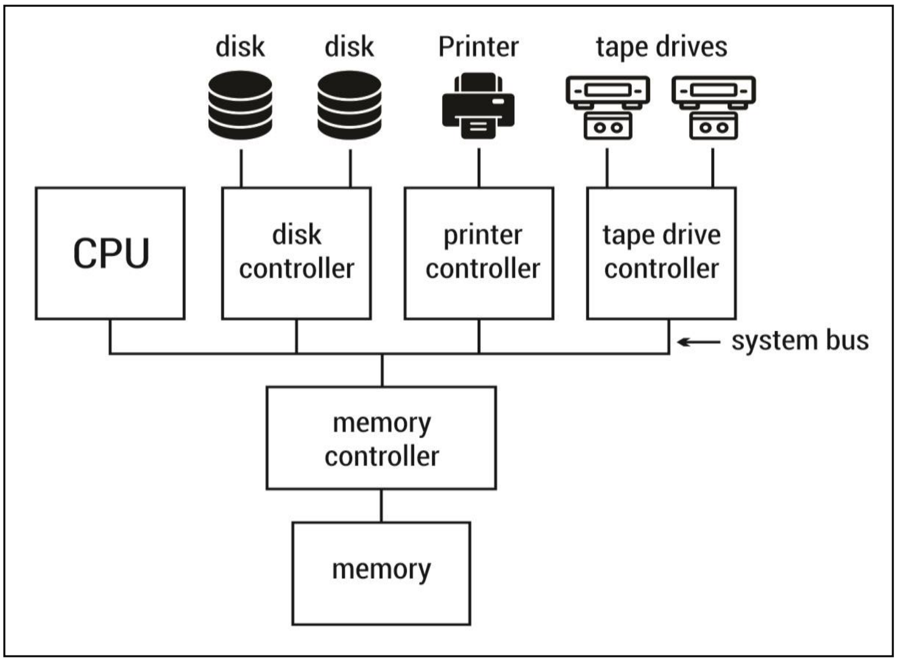


Fig 2.5 Centralised computer system

Fig 2.5 shows a Client-Server Systems, where a system satisfy requests produced at m client systems:

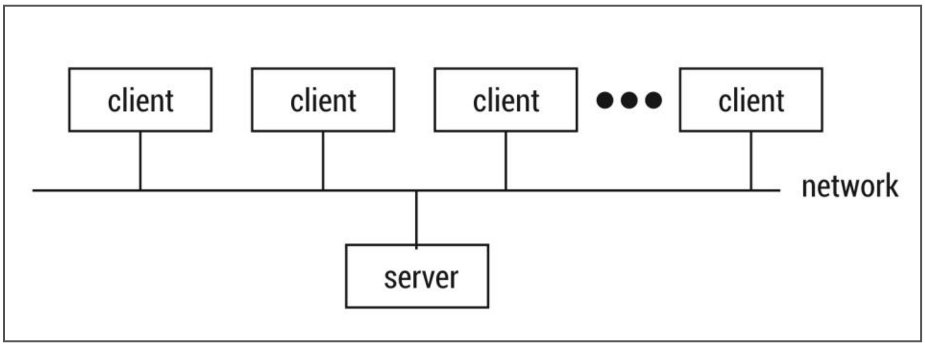


Fig 2.6 Client-Server System

In this, the database functionality is divided as:

- End-end: It Controls access structures, query evaluation and optimization concurrency control and recovery.
- Front-end: It Comprises of tools like forms, report-writers and graphical user interface facilities.
- The interface between the front-end and the back-end is through SQL or through an application program interface.

There are certain advantages:

- superior functionality for the cost
- elasticity in establishing resources in addition to growing facilities
- enhanced user boundary
- quick maintenance

Check your progress 1

1. In a Client Database System, _________controls the access structures, query evaluation and optimization.

   a. Back-end　　c. both

   b. Front-end　　d. none

2.3 Parallel Database Systems

Parallel database device architectures encompass development from implementation of attractive hardware to software corresponding dataflow architecture which is based on straight shared zero hardware. Such type of pioneering plan presents extraordinary speedup as well as scale up as soon as handing out relational database queries.

Parallel database systems are preliminary of dislocation conventional for mainframe computers for most important database as well as transaction processing everyday jobs. The accomplishment of these systems contradicts the 1983 theory that expected the end of database machines. After 10 years, future of highly-parallel database machines comes out to be depressing. Many database machine investigate on particular, over and over again in hardware like CCD memories, bubble memories, head-per-track disks plus optical disks. Nothing of such technologies satisfies such guarantees, so there exists knowledge about conventional processors, RAM as well as magnetic disks that will control the prospect for lots of years to draw closer.

The dataflow approach as shown in fig 2.7that needs system design to use message based client-server operating system that will connect parallel processes worked out by relational operators. It will require a high speed set of connections to interrelate parallel processors. Such facilities come out as strange decade ahead of though, today they are conventional of computer structural design. The client-server thought by means of high speed LANs is foundation for most part of desktop computers, workstation as well as workgroup software. Such type of similar client server mechanisms behaves as an outstanding foundation for distributed database knowledge.

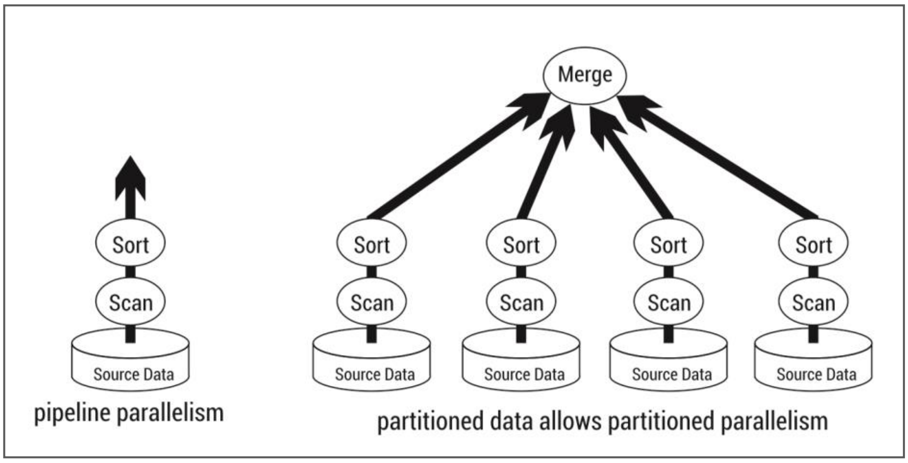


Fig 2.7 parallel database

The ideal parallel system should have following features:

Linear speedup:

Twice as much hardware can perform the task in half the elapsed time

Linear scale up:

Twice as much hardware can perform twice as large a task in the same elapsed time

Check your progress 2

1. The first concept of parallel database concept was disapproved in the year:

a. 1981	c. 1985
b. 1983	d. 1990

2.4 Distributed Database Systems

Earlier, the programs are kept in regular files. It is seen that every program has to maintain its own data as:

- huge overhead
- error-prone

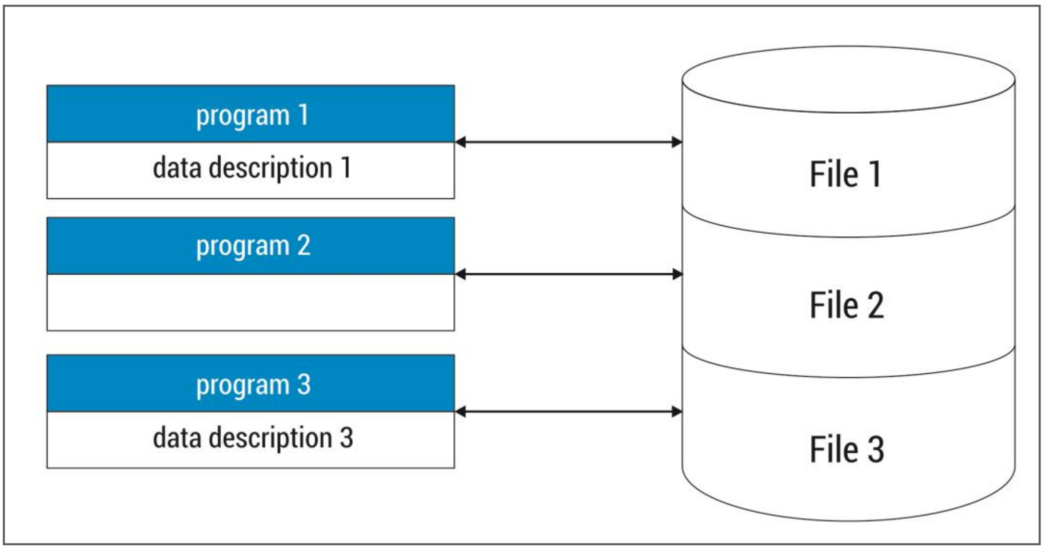


Fig 2.8 Distributed database system

Distributed database system is the amalgamation of what come into sight as two diametrically contrasting approaches in the direction of data processing:

- database systems
- computer network

Computer networks encourage a style of work that moves beside centralization. The most important concern to recognize such combination can be:

- Integration as not centralization
- Integration occurs without centralization

The main aim of distributed database systems is to:

- achieve data integration
- data distribution transparency

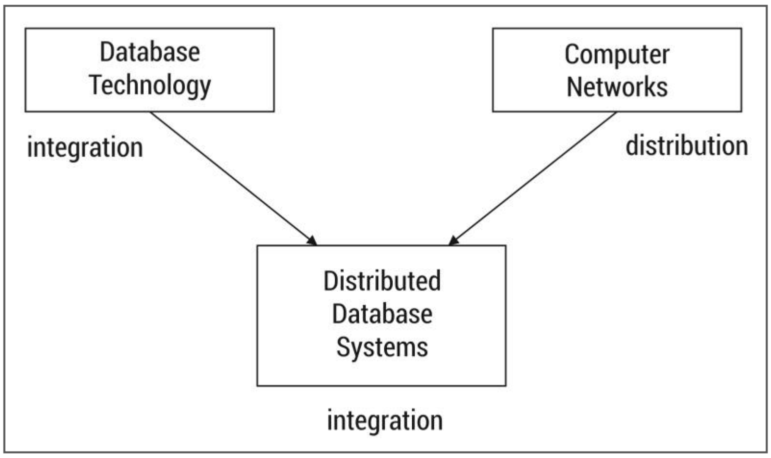
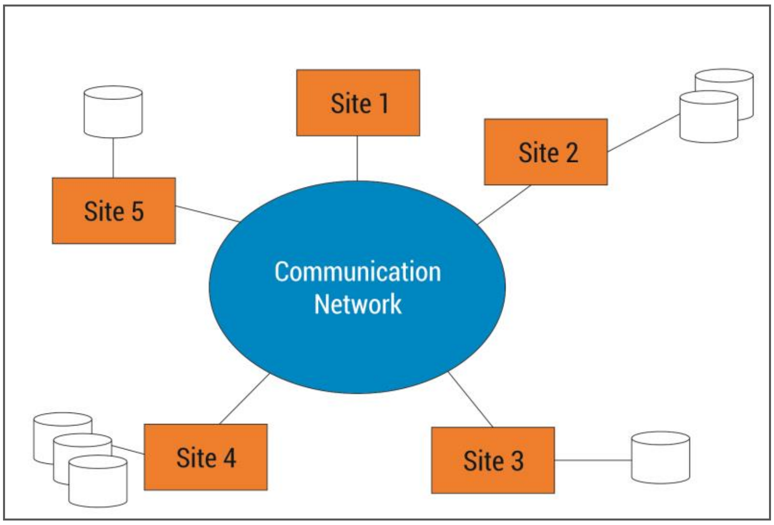


Fig 2.9 Distributed model

Further, it was observed that a distributed computing system is a mixture of independent dispensation elements that are organized by computer arrangement as shown in fig 2.10. The fundamentals work together in view to carry out the allocated task.



2.10 Fig Communication network

Applications

The distributed data system is used in:

- Manufacturing, especially multi-plant manufacturing
- Military command
- Airlines
- Hotel chains

The features of Distributed Database Systems are:

- Higher reliability
- Improved performance
- Easier system expansion
- Transparency of distributed and replicated data

Check your progress 3

1. Which is not the feature of Distributed database management system?

   a. Low reliability

   b. Improved performance

   c. Easier system expansion

   d. Transparency of distributed and replicated data

2.5 Client-Server Database System

Client/server is an extremely wide-ranging of software architecture that can be functional to an enormous range of software missions. However if you listen in business execs talking about client/server, probability is that you'll hear about SQL servers in the same conversation.

A Client server database system contains three main software components:

- client application (front end)
- data access layer (middleware)
- database server (database engine or back end)

In this, the client application is in charge for:

- accepting input from user
- accepting a query to database server depends on input
- receiving results from server
- formatting
- Presenting to the user.

In this database, the data access layer is moderately apparent as crystal towards the user, apart from that it might be tremendously clear to developer of client app. The database server acknowledges queries commencing from clients, processes them concurrently as well as returns the result back. There are various different query languages in the region around, near far away is SQL.

There are numerous compensation of such systems which are individual of its hardware requirement. The Client server database systems share out processing amongst clients that runs the user interface as well as definite application logic all along by way of server intended for controlling the database engine furthermore centralized data access logic. In an ideal world, such division of labour can put off the slowdowns connected with systems everywhere each and every one which process information centrally. On the other hand, such type of distribution creates a new possible restricted access called as a network. The network is in addition defenceless to congestion by exercising external database application.

Both in hardware as well as software, the CSDB systems are due as being scalable as well as modular. This is accurate to the level that the designers plan it. It is likely to build up CSDB systems which can develop smoothly as well as incrementally into large services furthermore whose components are replaceable with new products. It is unfavourably easy, on the other hand, to expand a CSDB system whose user capacity and/or transaction means low ceiling, or whose components are attached to one another. But with careful planning, a CSDB system can be better together than host-based design.

One of the important advantages of CSDB proponents is excellence of development tools. It is unquestionable that person in command advances have been equipped in development tools in earlier years, furthermore CSDB tools have completed for the most part advances of any:

- technical 4GLs
- object-oriented 4GLs
- graphical GUI builders
- object-oriented GUI builders
- graphical object-oriented languages
- data modelers
- component repositories

The advantages of CSDB systems are most important abundant organizations to imagine about its association. Migrating a user base of some size on or after a host-based database to CSDB agreement in fact involves re-engineering the business processes necessarily. Business processes are more often than not developed with the computers of their day in mind. Re-implementing an alive business process in new surroundings may not offer any of standard benefits if that process in particular adapts to inherit system or if it is fundamentally unproductive.

Check your progress 4

1. Which of the following items are not necessary for client/server?

   a. Assure that tools will connect with middleware.

   b. Understand the requirements.

   c. Determine network bandwidth capabilities.

   d. Include the use of a file server.

2.6 Let Us Sum Up

In this unit we have studied that a centralised database system gives a complete integration for hybrid systems with centralized control and monitoring. Such type of system will ease road operator staff's workload by applying management tools such as point-of-sales and CRM applications.

It is found that parallel database device architectures originate from attractive hardware and software which generalised dataflow architecture. This will improve the speed as well as scale up on applying relational database queries.

The client/server architecture is a wide range of software architecture which works for different software applications. This is good in business applications as it applies SQL server to develop comfortable client model.

2.7 Answers for Check Your Progress

Check your progress 1

Answers: (1-a)

Check your progress 2

Answers: (1-b)

Check your progress 3

Answers: (1-a)

Check your progress 4

Answers: (1-d)

2.8 Glossary

1. **Logical model -** It is a data model level where you frame a conceptual model which carries objects like entities, attributes and key groups.

2. **Logical/physical model -** it is a model which makes logical and physical models that are linked automatically.

2.9 Assignment

Explain the relative roles of Clients and Servers.

2.10 Activities

Compare distributed database with parallel database?

2.11 Case Study

Develop an activity on parallel database processing and study.

2.12 Further Readings

1. Asher, H.B. (1984), Causal modelling, Sage Publications.
2. Creswell, J.W. (1994), Research design: Qualitative and quantitative approaches.
3. Kerlinger, F.N. (1979). Behavioural research: A conceptual approach. Holt, Rinehart & Winston.

Block Summary

While studying this block, students will get knowledge and understanding about data retrieval and different database systems in SQL. The user will be detailed on various topics like Single table query, Group and Order by Clause and Natural Join and Sub query along with Centralized Database System. The student will be given knowledge about different rules and criteria's of data retrieval.

After completing this block, students will be able to gather sufficient knowledge on database and how to create database using SQL. The idea about database system and its features will help them to design a conceptual model in their labs. The concept of Parallel Database Systems and Distributed Database Systems along with its in reaction with Client-Server Database System will really help them in future. This block is useful for programmers as well as non programming students as it explains the crunch of database system with simplicity and shortcuts along with several illustrations that will made them to clear all their doubts.

Block Assignment

Short Answer Questions

1. Explain Centralised database System?
2. What is Natural Join and Sub query?
3. What are the advantages of parallel database system?
4. Write short note on single table query with examples?
5. What is the application of Client server model database?

Long Answer Questions

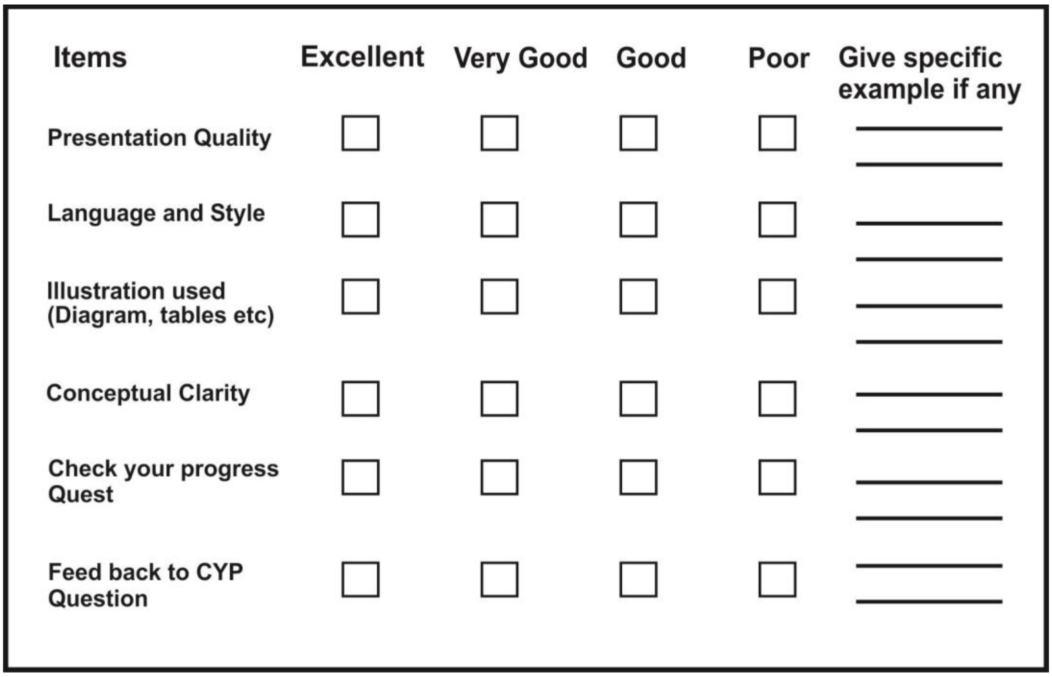
1. What are the different types of database systems?
2. Write the comparison about Group by clause and Order by Clause?
3. What are differences in Centralized and Distributed Database Systems?

Enrolment No.

1. How many hours did you need for studying the units?

Unit No	**1**	**2**	**3**	**4**
Nos of Hrs				

2. Please give your reactions to the following items based on your reading of the block:



Items	Excellent	Very Good	Good	Poor	Give specific example if any
Presentation Quality	□	□	□	□	________ ________
Language and Style	□	□	□	□	________ ________
Illustration used (Diagram, tables etc)	□	□	□	□	________ ________
Conceptual Clarity	□	□	□	□	________ ________
Check your progress Quest	□	□	□	□	________ ________
Feed back to CYP Question	□	□	□	□	________ ________

3. Any Other Comments

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

……………………………………………………………………………………………………

> *Education is something*
> *which ought to be*
> *brought within*
> *the reach of every one.*

- Dr. B. R. Ambedkar

Dr. Babasaheb Ambedkar Open University
Jyotirmay' Parisar, Opp. Shri Balaji Temple, Sarkhej-Gandhinagar Highway, Chharodi,
Ahmedabad-382 481.