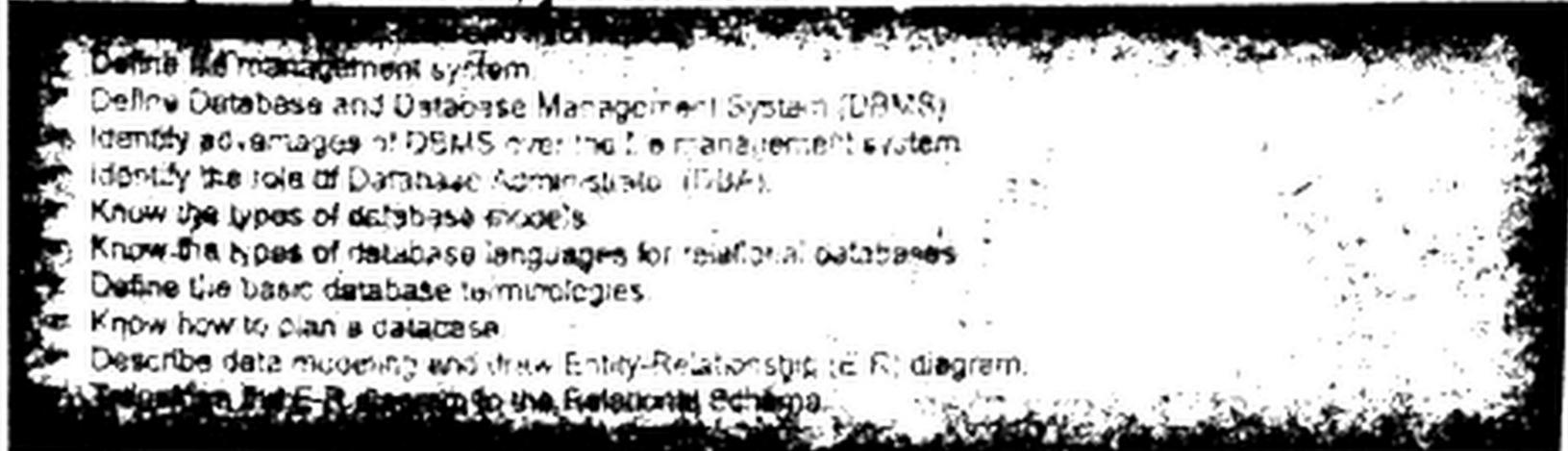


# DATABASE FUNDAMENTALS

## CHAPTER

# 7

After completing this lesson, you will be able to:



### UNIT INTRODUCTION

A database is a collection of related files that are usually integrated, linked or cross-referenced to one another. This unit is dedicated to database and database management systems. It introduces the types of database models and provides the basic knowledge about data modeling and E-R diagram for planning and development of a database system. It presents the advantages of using a database management system over the old file management system. It describes the responsibilities of database administrator for the performance, integrity, aim and security of a database.

### 7.1 INTRODUCTION TO DATABASE

**Q.1** Give a brief introduction to database and database systems.

**Answer**

#### Databases and Database Systems

Databases and database systems have become an essential part of everyday life modern society. Most of us encounter several activities that involve some interaction with database.

#### Explanation

If we go to the bank to deposit or withdraw money; if we make a hotel or airline reservation or if we access a computerized library catalogue to search for a book chances are that our activities will involve someone accessing a database. Even purchasing items from a supermarket nowadays in many cases involves an automatic update of the database that keeps the inventory of supermarket items.

#### 7.1.1 DATA AND INFORMATION

**Q.2** Differentiate between data and information with the help of examples.

**Answer**

#### Data

Data refers to the facts and figures in raw form i.e. not in organized form. Data is the raw material given as input to the computer for processing.

#### Example

The item code, description, quantity and price of items purchased in a store are data. When this data is entered into the computer and processed, it prints the bill, which is the output of the computer.

#### Information

Information is the processed or organized form of data. When the computer processes data and it is properly arranged and organized, it is converted to information. It is also called the output of computer.

and is the meaningful form of data.

#### Example

The names of students and their marks in all the subjects is data. When the computer processes it, the result sheets and report cards produced is information.

### 7.1.2 FILE MANAGEMENT SYSTEM

**Q.3** What is file management system? Explain the problem in this system with the help of an example.

**Answer**

#### File Management System / File Based Management System

A file (or file based) management system is a collection of programs used for managing data stored in various files. Each program within a file management system is developed independently and it defines and manages its own data.

#### Problem in File Management System

In file management system, records in one file are not related to the records in any other file. This approach leads to many problems, which include data duplication in different files, data inconsistency, sharing of data and lack of flexibility in organizing and querying the data.

#### Example

A company may store a salesman's name, address telephone number and commission rate in a file in Accounts Department and the same information may be stored in another file in Sales Department. The Accounts Department, for example, may change the commission rate of salesman but the Sales Department may fail to update its file. This will produce reports calculated with out-of-date figures. The inconsistency in files occurred due to duplication of same data in files. This is the main problem faced in file management systems.

### 7.1.3 DATABASE

**Q.4** What is the solution of the problems in traditional file management system?

**Answer**

#### Solution of the problems in traditional file management system

In order to solve the problems of traditional file management system for managing data, the concept of a database was introduced.

**Q.5** Explain 'database' its size, creation and manipulation in detail.

**Answer**

#### Database

A database is a collection of related data.

#### Explanation

Consider the names, telephone numbers and addresses of the people you know. You may have recorded this data in an indexed address book or you stored it on your computer's hard disk using software such as Microsoft Access or Excel. This is a collection of data having implicit meaning and hence is a database.

#### Size of Database

A database can be of any size and complexity.

#### Explanation

A database containing names and telephone numbers of your friends may have only a few records. On the other hand, a database containing information about all the citizens of a country for National Identity Card (NIC) may contain millions of records.

#### Creation and Manipulation of Database

A database may be created and maintained manually or it may be computerized. The library card catalogue is an example of a manually created and maintained database. A computerized database is

created and maintained by a database management system.

#### Examples

- Library Management System
- Stock Control System
- Examination Control System

### 7.1.4 DATABASE MANAGEMENT SYSTEM (DBMS)

**Q.6** What is meant by DBMS? Explain it in detail.

**Answer**

#### DBMS

A database management system (DBMS) is a set of programs that allow users to create, maintain and manipulate database, and store or retrieve data from those database files. It provides user-friendly access and controls between user and database. Its main purpose is to improve data sharing, data access, decision making and increase end user productivity.

#### Operations in manipulation of data

Manipulation of data includes the following operations:

1. **Adding new data**  
Adding details of new students
2. **Deleting unwanted data**  
- Deleting the details of students who have completed course.
3. **Changing existing data**  
Modifying the fee paid by the students.

#### Function/Purpose of DBMS

The DBMS helps to create an environment in which users have better access to data. DBMS helps to give an integrated view of the organization's operations. The DBMS makes it possible to share the data in the database among multiple applications and users.

#### Examples

Some examples of the database systems managed by DBMS are:

- Customer information system
- Inventory information
- Library management
- Accounting and bookkeeping

#### Examples of DBMS Applications

Examples of DBMS applications are as follows:

- Microsoft Access
- Microsoft SQL Server
- Sybase
- Oracle
- MySQL

### 7.1.5 ADVANTAGES OF DBMS OVER FILE MANAGEMENT SYSTEM

**Q.7** Describe the advantages of DBMS over file management system.

**Answer**

DBMS has many advantages over the file management system because it provides solutions to all the problems faced in file management system.

#### Problems in file management system

The following problems were faced in file management system:

- Reduced Data Redundancy
- Data consistency
- Program-Data Dependency
- Lack of Flexibility
- Data not Sharable
- Backup and Recovery
- Data Security

#### 1. Reduced Data Redundancy

Data redundancy is the duplication of data in many different files in file management system.

##### Example

A salesman's data may be held on a file in the Sales Department and also Personnel Department.

When data is to be updated, it must be changed in both files. This results in wastage of storage space and may lead to data inconsistency.

##### DBMS not faced with the problem

This problem is not faced in DBMS because all the data belonging to the entire organization is centralized in a common pool of data, accessible by all the programs.

#### 2. Data consistency

In file management system, same data may be held in several different files, it has to be updated in each separate file when it changes. If data is not updated in any file, it causes data inconsistency.

##### Solution in DBMS

This problem is solved in DBMS because all the data is centralized for use by all the programs

##### Example

In a school, students' information such as name, address, phone number and class are held in a file in Admission Office. The same data with tuition fee and some other data are also kept in another file in Accounts Office. If a student's address changes it must be updated on both files. In case, it is updated in only one file, the other file will have out-of-date data. This will cause data inconsistency.

#### 3. Program-Data Dependency

In file management system, every computer program in each Department has to specify exactly what data fields constitute a record. Any change to the format to the data field of a record, such as adding a new field or changing the length of a field means that every program, which uses that file has to be changed.

##### Example

If a new field, place of birth of students is to be added in student data files in Admission Office and Accounts Office, run both the programs need to be changed.

#### 4. Lack of Flexibility

In file management system, when information of a non-routine nature is needed, it can take weeks to assemble the data from the various files and write new programs produce the required reports.

##### Example

A report about students is required in a school that has to merge information stored in various files. This will require lot of time and effort by the programmer to write a program to print such a report in file management system, whereas, it can be feeling very easily produced in database management system.

#### 5. Data not Sharable

When a copy of a data file is made for sharing data with other people in a file management system, it will soon lead to data inconsistency if data in one of the files is not updated. Therefore, it is difficult to share data with others in a file management system.

#### 6. Backup and Recovery

DBMS provides facilities for backup and recovery from failures including disk crash, power failure, software errors, which may bring the database from the consistent state to a state prior to the failure.

## 7. Data Security

In Conventional filing systems, there is no centralized security system, which restricts users according to their role in the organization. DBMS makes it easier to enforce security restrictions since database is centralized. Users are provided permissions to access data according to their rights.

### 7.1.6 ROLE OF DATABASE ADMINISTRATOR (DBA)

**Q.8 Who is database administrator? What are his responsibilities?**

**Answer**

#### Database Administrator

Database Administrator (DBA) is the person responsible for supervising the database and the use of DBMS in an organization.

#### Responsibilities of DBA

DBA has the following responsibilities:

1. Designing the database and enforcing the operational policies and procedures for its usage.
2. Planning security measures and backup of database.
3. Controlling privileges and permissions of database users.
4. Allocating passwords to users
5. Planning recovery procedures if hardware or software failure occurs and ensuring that no data is lost.
6. Providing training to new employees about using the database.

### 7.1.7 DATABASE MODELS

**Q.9 Define database model. Explain its types.**

**Answer**

#### Database Model

A database model is the theoretical foundation of a database and determines in which manner data can be stored, organized and manipulated. It defines a way of structuring data.

#### Types of Database Models

There are five types of database models, which are as follows:

- Hierarchical Database Model
- Network Database Model
- Relational Database Model
- Object-Oriented Database Model
- Object Relational Database Model

#### 1. Hierarchical Database Model

In a hierarchical model, data is organized into a tree-like structure as shown in Fig.7.1.

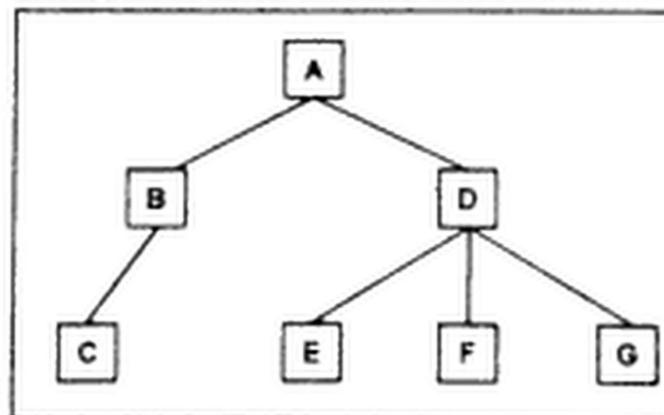


Fig. 7.1 Hierarchical Database Model

It is a logical construct with owner and subordinate relationship.

#### Members and Owners

Data elements in a subordinate relationship are called members and those having subordinates are

called owners. In a hierarchical structure, subordinates can have only a single owner and there is only one data element that has no owner. This structure is very suitable in describing many relationships in the real world.

**Examples of hierarchical model**

Some examples of hierarchical model are table of contents, departments of an organization and types of memories.

**2. Network Database Model**

A network model is a logical structure in which some of the data elements can have more than one owner data element as shown in Fig 7 2

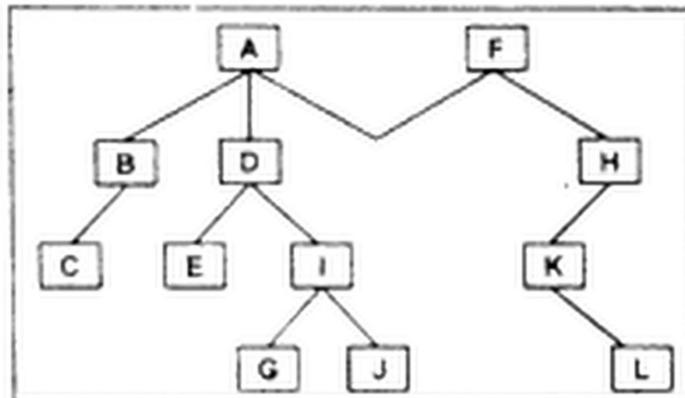


Fig. 7 2 Network Database Model

A network is a more complex structure than a hierarchical model. A hierarchical model can be thought of as a network with some discipline imposed on it. An application with which crew members, aircraft, routes and schedules are related is by its nature a network model.

**3. Relational Database Model**

In a relational database model, data is held in tables as shown in Fig.7.3 and the tables are linked by means of common fields.

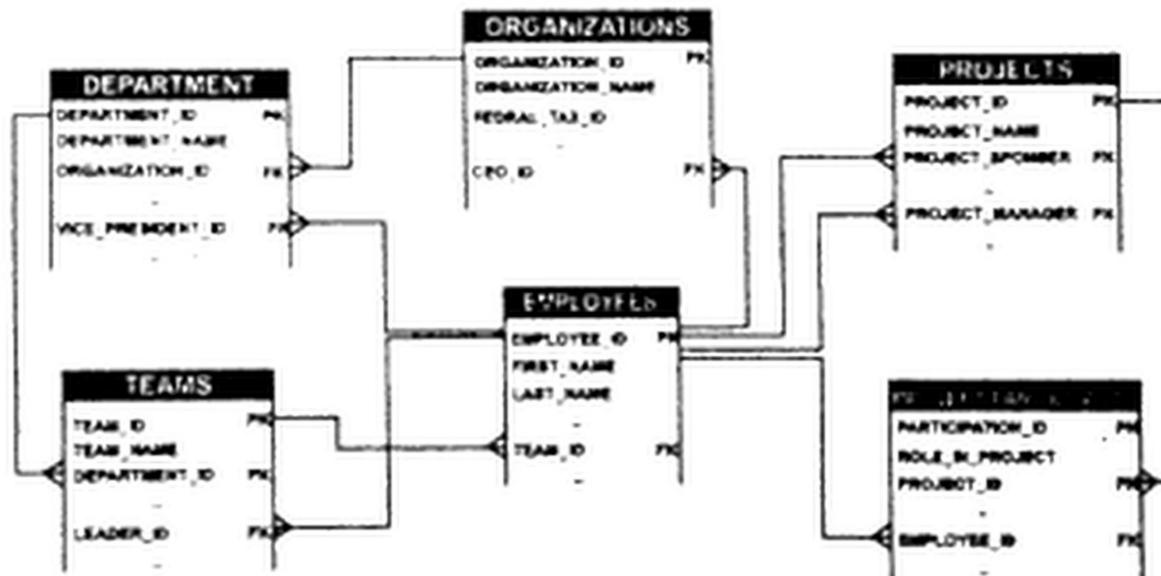


Fig.7.3 Table in a relational database

**Relation**

The "relation" in a relational database refers to the various tables in the database, which are linked with each other.

**Table**

A table in database consists of rows and columns. One row of table holds one record. Each column in the table holds one field or attribute.

**Use of relational database model**

The relational database model is used to make database management systems more independent of any particular application. It is the most popular database model used in business applications.

**4. Object-Oriented Database Model**

Object-oriented databases is also called Object Database Management Systems (ODBMS). Object

databases store objects rather than data such as integers, real numbers and strings. Objects are used in object oriented programming languages such as C++ and Java.

### Characteristics of Real world

Real world objects have two characteristics, that is, state and behaviour.

#### Example

The object radio has four states and five behaviours as shown in Fig 7.4.

Object : Radio	
States	Behaviours
On	turn on
Off	turn off
Current channel	tune channel
Volume	increase/decrease volume

Fig.7.4 States and Behaviour of object Radio

### Object

An object stores its states in variables and exposes its behaviour through functions in programming languages.

#### 5. Object Relational Database Model

An Object Relational Database (ORD) is a database management system similar to relational database, but with an object-oriented database. In object relational database, the approach is essentially that of relational database. The data resides in the database and is manipulated collectively with queries in a query language.

## 7.1.8 DATABASE LANGUAGES

**Q.10** What is meant by SQL? Explain SQL languages in detail.

**Answer**

### SQL

SQL (Structured Query Language) is a standard language for accessing and manipulating databases.

#### Types of SQL languages

There are three types of SQL languages for relational databases. These are:

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)

#### 1. Data Definition Language

Data Definition Language (DDL) is a computer language for defining different data structures. DDL statements create, modify and remove data objects such as tables.

#### Tasks of DDL

Some important tasks of DDL are:

- To create objects in a database model
- To alter the structure of the database
- To delete objects from the database

#### 2. Data Manipulation Language

Data Manipulation Language (DML) statements are used to manage or manipulate data.

#### Tasks of DML

Some examples of DML tasks are:

- To retrieve data from a database
- To insert data into a table

- To update existing data within a table
- To delete records from a table

### 3. Data Control Language

Data Control Language (DCL) is used for controlling the data. A user can access the data based on the privileges given to him

#### Tasks of DCL

Some important tasks of DCL are:

- To allow specified users to perform specific tasks
- To disallow specified users from performing specified tasks
- To cancel previously granted or denied permissions

## 7.2 BASIC DATABASE TERMINOLOGIES

### Q.11 What is the purpose of DBMS?

Answer

#### Effective DBMS

An effective DBMS provides users with timely, accurate and relevant information. This information is stored in computer files, which needs to be suitably organized and properly maintained so that users can easily access the information they need.

### Q.12 Briefly explain the database term 'table'.

Answer

#### Table/Relation

A database table is a file that contains data about a single entity. An entity can be a person, place or event, etc..

#### Example

'Student', 'Teacher', 'Stock', etc. are entities.

#### Composition of table

A database table is composed of rows and columns. Rows hold the records and columns hold the fields. Data item is inserted at each row and column intersection. Tables are also called relations in RDBMS.

Each table in a database holds data about a different but related subject.

Fig. 7.5 shows a database table containing data about the entity 'Student'.

Roll No.	Name	Class	Section	Date of Birth	Address
1	Muhammad Ali	XI	A	20-6-1999	Islamabad
2	Zahid Saleem	XI	A	15-4-2000	Rawalpindi
3	Mustafa	XI	A	6-6-2001	Rawalpindi
4	Ahmed	XI	A	12-8-1999	Islamabad
5	Umar Gul	XI	A	23-7-2000	Islamabad

Fig. 7.5 A Table/Relation

### Q.13 Explain record/tuple/row and field/attribute/column.

Answer

#### Record/Tuple/Row

A collection of related fields treated as a single unit is called a record. All the information about one

person or item is held in a record. When records are stored in a table, rows represent records and columns represent fields. In relational database, rows are also known as tuples. Database table in Fig. 7.5 has five records, represented row wise, about the entity 'Student'.

#### Fields/Attribute/Column

A field or attribute is part of a record and contains a single piece of data for the subject of the record. In the database table illustrated in Fig. 7.5, each record contains six fields:

<b>Roll No.</b>	Used to assign a unique roll number to a student.
<b>Name</b>	Used to give name of the student.
<b>Class</b>	Used to give class of the student.
<b>Section</b>	Used to store the section of the student.
<b>Date of Birth</b>	Used to give date of birth of the student.
<b>Address</b>	Used to give the address of the student.

Fields appear as columns in a database table.

#### Q.14 Define file and tables/relations with respect to database.

**Answer**

##### File

A file is a collection of records.

##### Example

A stock file contains records for items of stock, a payroll file contains records for employees and so on.

##### Tables/Relations

In a relational database, records are stored in files called tables/relations.

Fig. 7.5 shows a table/relation of 'Student' file.

#### Q.15 Define view and data types with respect to database.

**Answer**

##### View

View is made up of rows and columns. It may display information that is restricted to a part of table. It may also present selected data from several tables simultaneously. View cannot exist independently of tables.

##### Data Types

Every field in a table is assigned a data type. Data types available in a relational database are character data, integers/real numbers, Boolean data, date/time, etc..

##### 1. Character

Character is used to store text and combinations of text and numbers

##### 2. Integer

Integer is used to store whole numbers such as 34 and 2073.

##### 3. Real numbers

Real numbers are used to store numbers that have fractional part such as 23.75.

##### 4. Boolean data

Boolean data is used for True or False values. Null values are not allowed.

##### 5. Date and time

Date and time is used for storing date and time.

#### Q.16 Define key with respect to database and explain its types.

**Answer**

##### Key

Key is an attribute (or field) that is used to identify records in a table. The purpose of key is to link data together between tables without repeating all of the data in every table.

### Types of keys

The following are the types of keys used in databases:

- Primary Key
- Candidate or Alternate Key
- Secondary Key
- Foreign Key

#### 1. Primary Key

Each entity in a database must have a unique key field known as primary key to identify a record.

##### Example

Roll Number of a student can be used as primary key in a student database

(Fig. 7.5) since it is unique key field.

#### 2. Candidate or Alternate Key

A key field that can act as a primary key field in a table to unique identifies each record but it is not chosen, as primary key is known as candidate or alternate key.

##### Example

Roll Number of a student is chosen as primary key field in a student database N.I.C. Number of a student is unique and it can also act as primary key. Therefore, N.I.C Number is a candidate or alternate key.

#### 3. Secondary Key

Sometimes a record in a table need to be searched on field other than primary key such a field is known as secondary key.

##### Example

A student table needs to be search by name. Then name becomes secondary key.

#### 4. Foreign Key

A key field used in a relationship between tables whose value matches a primary key in the other table is known as foreign key.

##### Example

Suppose a student database has two tables. Student table that contains students' particulars and another Result table that contains the results. Student table is the primary table in which Roll Number is primary key. To link the student table and the Result table, the field Roll Number can also be used as foreign key in the Result table.

## 7.3 PLANNING A DATABASE

**Q.17** What is database planning? Explain the steps for planning a database.

**Answer**

### Database planning

Database planning is a systematic approach to the development of database that moves from concept to design and development to implementation. A well-designed database promotes consistent data entry and retrieval. Database should be planned in a systematic way to save time, efforts and make it perform the expected tasks. The time and work required to plan a database depends on its complexity.

### Steps for planning a database

The following steps are involved in planning a database:

- Problem Identification/Definition
- Feasibility Study
- Requirement Analysis
- Identify Entities and Attributes
- Assigning Names to Tables and Columns

### 1. Problem Identification/Definition

In this step, the nature and scope of the problem, to be solved, is identified and the problem is clearly defined. The database developers must know what type of information is given and what are the unknowns. They have to analyze the problem to gather as much information as possible for finding a solution.

#### Example

The Examination Section Head of a College has been getting complaints of poor Examination services from the Examination department. This may lead an initial investigation to find whether a new system can solve the problem. If the report suggests a new system, this leads to the next phase, which is the feasibility study of the new system.

### 2. Feasibility Study

The purpose of feasibility study is to find one or more solutions of the problem and to suggest the most desirable and economical solution. For this purpose, the database developer needs to generate several solutions of the problem to accomplish the desired task and propose one solution.

#### Steps of Feasibility study

Feasibility study includes the following steps:

- Investigate the problem
- Find out all the possible solutions available
- Study all the solutions to determine their feasibility
- List the issues with each solution
- Select the preferred solution for implementation
- Document the results in a feasibility report

### 3. Requirement Analysis

The purpose of requirement analysis is to obtain thorough and detailed understanding of the problem. It is important to create a complete and accurate presentation of all the requirements. Only then, it is possible to develop a database that fully satisfies the requirements.

#### Example

To develop 'College Examination System' a detailed study is needed to analyze the requirements. This will include the following:

- (i) Entities required i.e. the number of tables (database files) required (like Student, Exam, Exam Type, Result, Courses, etc.)
- (ii) Fields required for each entity in each table (like Roll No., Name, Class, Section, etc. for Student entity)
- (iii) Key field in each table Data types to fields
- (iv) Relationship between entities
- (v) Queries
- (vi) Forms design
- (vii) Reports

### 4. Identify Entities and Attributes

After requirements identification, the next step is to identify the entities and its attributes. An entity is the main data object that is of significant interest to the organization. It is usually a person, place, thing, or event to be recorded in the database.

An Attribute is a property that describes an entity.

#### Example 1

If employee is an entity then the employee's name, age, address, salary and job etc. are the attributes.

#### Example 2

To develop 'College Examination System' the following entities and attributes will be required as shown in Fig. 7.6.

Entities	Attributes
Student	Roll_number, Name, Class, Section, Date of birth, Address
Course	Course_Id, Course_name, Description
Exam	Exam_Id, Exam_name, Start_date
Exam_result	Marks

Fig. 7.6 Entities and Attributes of Examination System

### 5. Assigning Names to Tables and Columns

Once entities and attributes are identified, entities are converted to tables and attributes to columns of the tables. There are no standard conventions for naming tables and columns but all names should be meaningful and consistent throughout the database.

#### Example 1

"Student" for Student's table. In case of attributes, meaningful names should be used wherever possible.

#### Example 2

The College Exam System may have the table name 'Student' and the columns (attributes) may be assigned the names, Roll number, Name, Class, Section, Date of birth, Address. The table Student is shown in Fig 7.7 with three records.

Roll_number	Name	Class	Section	Date of birth	Address
1	Fahad	XI	B	15/05/1986	Sialkot
2	Meerab	XI	B	26/02/1987	Lahore
3	Rameen	XI	B	23/08/1988	Lahore

Fig 7.7 Table Student with three records

## 7.4 DATA MODELING AND ENTITY-RELATIONSHIP DIAGRAM

### 7.4.1 DATA MODELING

**Q.18** Briefly explain data modeling.

**Answer**

**Data Modeling / Entity-Relationship diagram (ERD)**

Data modeling is the process of designing logical structure of a database with a diagram using text and symbols to represent the way data needs to flow. This diagram is called Entity-Relationship diagram (ERD)

Data models define how data is connected to each other and how they are processed and stored inside the system. Data models are built during the analysis and design phases of a project to ensure that the requirements for a new application are fully understood. A data model can be thought of as a diagram or flowchart that illustrates the relationships between data. Well-documented conceptual, logical and physical data models allow stake-holders to identify errors and make changes before any programming code has been written.

Data model designers often use multiple models to view the same data and ensure that all processes, entities, relationships and data flows have been identified.

**Q.19** Write down the different approaches to data modeling.

**Answer**

**Approaches to data modeling**

There are several different approaches to data modeling including:

#### 1. Conceptual Data Modeling

Conceptual Data Modeling identifies the highest-level relationships between different entities.

#### 2. Enterprise Data Modeling

Enterprise Data Modeling is similar to conceptual data modeling, but addresses the unique

requirements of a specific business.

### 3. Logical Data Modeling

Logical Data Modeling illustrates the specific entities, attributes and relationships involved in a business function serves as the basis for the creation of the physical data model.

### 4. Physical Data Modeling

Physical Data Modeling represents an application and database-specific implementation of a logical data model.

### Identify and state the data needs

When a system developer designs a new database system, one crucial task is to identify and state the data needs of the organization. This describes how the data elements in the system are to be grouped

**Q.20 Define the terms which are used in building picture of the data requirements.**

**Answer**

#### Terms used in data requirements

The following terms are used in building a picture of the data requirements.

- Entity
- Attribute
- Relationship
- Keys

#### 1. Entity

An entity is a thing of interest to an organization about which data is to be held.

#### Examples

Examples of entities include Student, Customer, Employee, Stock Item, Supplier, etc.

#### 2. Attribute

An attribute is a property or characteristic of an entity.

#### Examples

Examples of attributes associated with a Customer include Customer ID, Surname, Initials, Title, Address, etc.

#### 3. Relationship

A relationship is a link or association between entities.

#### Example

An example is the link between Dentist and Patient; one dentist has many patients, but each patient has only one dentist.

#### 5. Keys

Key is an attribute used to identify a record in a database. Keys are used to create links between tables to avoid duplication of data in various tables.

### 7.4.2 Entity-Relationship (ER) Diagram

**Q.21 What is E-R diagram? Explain in detail.**

**Answer**

#### Entity-Relationship (ER) diagram or model

The Entity-Relationship (ER) diagram or model defines the conceptual view of a database. It is made up of entities and the associations among them. At view level, the ER model is considered a good option for designing databases. Any object, for example, entities, attributes of an entity, relationship sets, and attributes of relationship sets, can be represented with the help of an ER diagram. ER Model is best used for the conceptual design of a database.

ER Model is based on:

- Entities and their attributes

➤ Relationships among entities  
 These concepts are explained in Fig. 7.8.

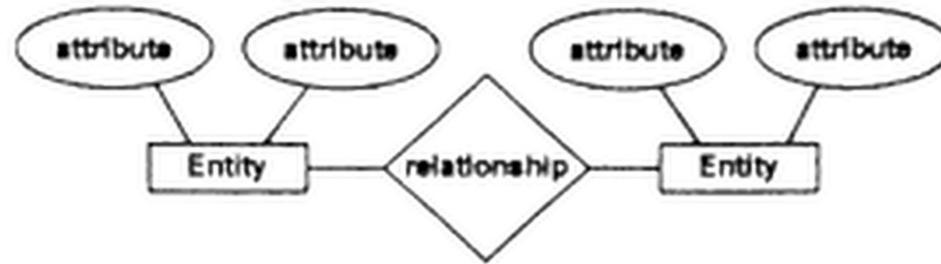


Fig.7.8 ER diagram concept

1. Entity

An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**.

Example

In a school database, a 'Student' is considered as an entity. Student has various attributes like roll number, name, age, class, etc..

Representation of entities

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent. Fig. 7.9 shows some entities.

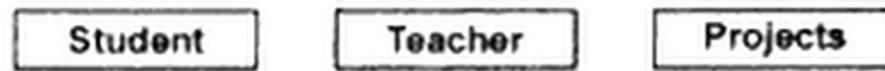


Fig.7.9 Entities Attributes

2. Attributes

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

Example

Fig 7.10 shows three attributes 'Roll No.', 'Name' and 'BirthDate' of entity 'Student'. The primary key attribute is underlined, for example, Roll\_No. attribute.

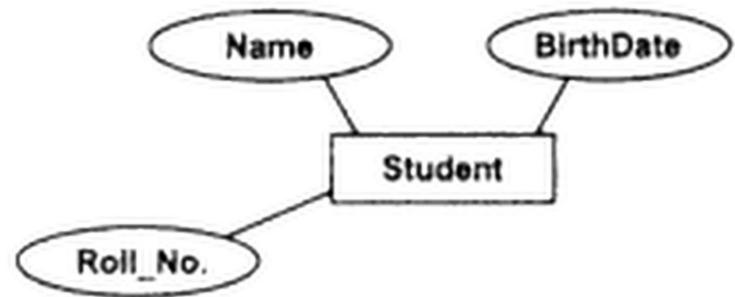


Fig.7.10 Attributes

3. Relationship

The logical association among entities is called relationship. Relationships are represented by a diamond symbol connected to the related entities. Fig 7.11 shows some relationships between entities.

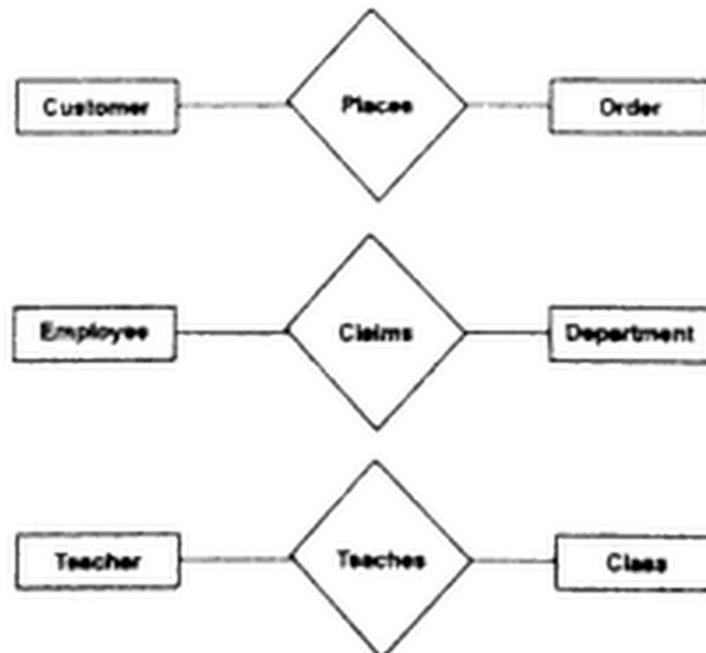


Fig.7.11 Relationships Degree of a Relationship

**Q.22 Describe the degree of the relationship.**

**Answer**

**Degree of the relationship**

The number of participating entities in a relationship is known as the degree of the relationship.

**Types of relationship**

Relationship has two types:

**1. Unary relationship**

It exists when an association is maintained within a single entity. It is also known as a recursive relationship. Fig. 7.12 shows a unary relationship, in which only one entity i.e. 'Machine Operator' is maintaining the recursive relationship.

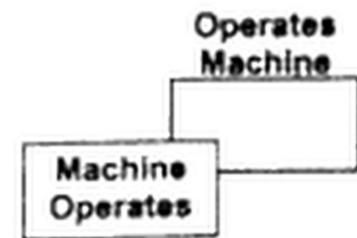


Fig.7.12 Unary Relationship

**2. Binary relationship**

It exists when two entities are associated with each other through a relationship or if there are two entity types involved.

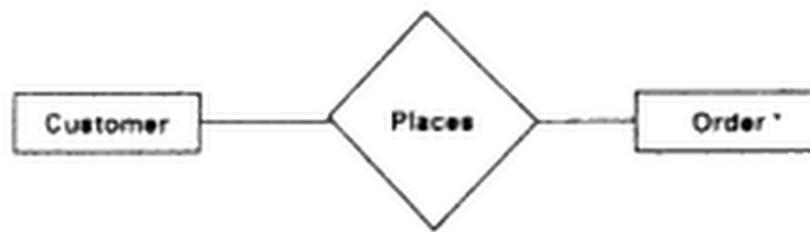


Fig.7.13 Binary Relationship

**Q.23 Explain the association between two entity types.**

**Answer**

**Association between entity types**

In binary relationship, the associations between two entity types may be described as:

- One-to-one
- One-to-many
- Many-to-many

**1. One-to-one**

When only one instance of an entity is associated with the relationship, it is marked as '1:1'. Fig. 7.14 reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.

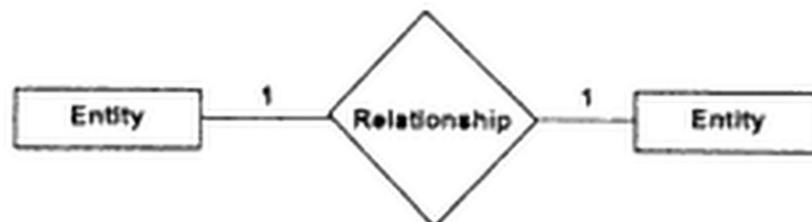


Fig.7.14 One-to-one Relationship

**2. One-to-many**

When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. Fig. 7.15 reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship.

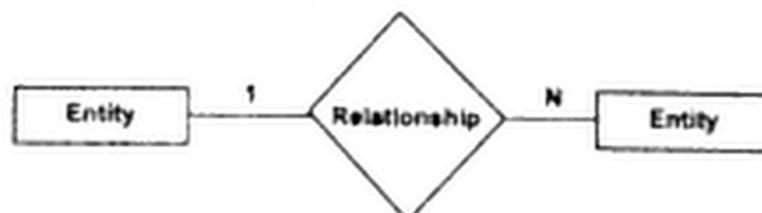


Fig.7.15 One-to-many Relationship

**3. Many-to-many**

When more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.

Fig. 7.15 shows many-to-many relationship.



Fig 7.16 Many-to-many Relationship

**7.4.3 CARDINALITY AND MODALITY**

**Q.24 Explain cardinality and modality with the help of examples.**

**Answer**

Cardinality and modality specify the business rules in a relationship.

**Cardinality**

Cardinality refers to the maximum number of times an instance in one entity can be associated with instances in the related entity. An instance in a database is the actual content of the database at a particular point in time

**Modality**

Modality refers to the minimum number of times an instance in one entity can be associated with an instance in the related entity.

**Explanation of Cardinality and Modality**

Cardinality can be 1 or many and the symbol is placed on the outside ends of the relationship line, closest to the entity.

Modality can be 1 or 0 and the symbol is place on the inside, next to the cardinality symbol. When cardinality and modality are specified together in a relationship between entities, for a cardinality of 1, a straight line is drawn.

For a cardinality of many, a foot with three toes is drawn. For a modality of 1 a straight line is drawn and for 0 a circle is drawn. Cardinality and modality are shown at both ends of the relationship line.

**Example 1**

Let us draw E-R diagram of entity 'Student' and 'Seat' that specifies the cardinality modality. A student fills a seat. This is a one-to-one relationship. This relationship can be as shown in Fig.7.16.



Fig 7.17 Relationship between Student and Seat

**Cardinality**

One student can fill a maximum of one seat. One seat can be filled by a maximum of one student. Each side of the relationship has a cardinality of one.

**Modality**

The modality on each side is also one. A student must fill at least one seat, and one seat must be filled by at least one student.

**Example 2**

Let us see the relationship between the entities 'Teacher' and 'Course'. A teacher teaches one or more courses. This is a one-to-many relationship. This relationship is shown in ' Fig.7.17.



Fig 7.18 Relationship between Teacher and Course

**Cardinality**

One teacher can teach many courses. One course is taught by only one teacher. The cardinality is one-to-many.

**Modality**

The modality is one on both ends of the relationship. One teacher must teach at least one course and one teacher must teach a course.

**Example 3**

A 'Branch' of a company has many 'Departments' and a 'Manager' manages each department. The E-R diagram that shows the cardinality and modality for this is shown in Fig 7.18.

There are three entities in this example, which are Branch, Department and Manager. Following are the cardinalities and modalities between these entities:

**1. Cardinality between Branch and Department**

- i. Each Branch has one or more departments.
- ii. One Department can only be in one branch.

Therefore, cardinality between Branch and Department is one-to-many as shown in the E-R diagram (Fig 7.18).

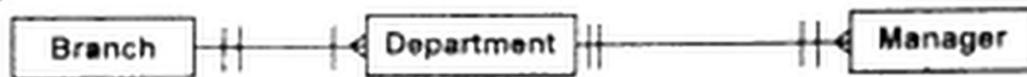


Fig 7.19 Relationship between Branch, Department and Manager

**2. Cardinality between Department and Manager**

- i. Each Department is managed by a Manager.
- ii. Each Manager belongs to only one Department.

Therefore, cardinality between Department and Manager is one-to-one.

**3. Modality between Branch and Department**

Modality between Branch and Department is one on both ends of the relationship because each Branch must have at least one Department and each Department belongs to only one Branch.

**4. Modality between Department and Manager**

Modality between Department and Manager is also one on both ends of the relationship because a Department must have at least one Manager and a Manager must manage one department.

**7.4.4 ENTITY- RELATIONSHIP (ER) DIAGRAM -- EXAMPLES**

The following are few examples of E-R Diagrams for some systems like Library Management System, Student Management System and Ticket Booking System

**Q.25 Draw E-R diagram for library management system.**

**Answer**

**Library Management System**

Library Management Systems consists of three entities. The entities are Books, Readers and Staff. E-R diagram for Library Management System is shown in Fig.7.19.

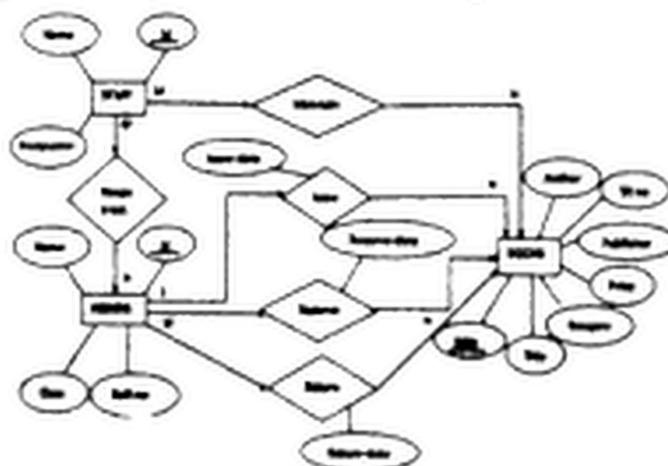


Fig.7.20 E-R Diagram for Library Management System

**Entities**

1. **STAFF:** The entity STAFF is the all staff of the library and its attributes are Id (primary key), Name and Designation.
2. **BOOKS:** The entity BOOKS is the all-available books in the library. Its attributes are ISBN (primary key), Title, Category, Price, Publisher, Sr no and Author.
3. **READERS:** The entity READERS is all the members of the library. Its attributes are Id (primary key), Name, Class and Roll no.

**Relationships**

1. The relationship between Staff and Books is many-to-many because many staff members are maintaining the books
2. The relationship between Staff and Readers is many-to-many because many staff members keep track of many readers.
3. The relationship between Readers and Books is one-to-many because a reader can borrow one or more books.
4. The relationship between Readers and Books is many to many as many readers can reserve and also return many books.

**Q.26 Draw E-R diagram for student management system.****Answer****Student Management System**

Student Management System registers students for various courses and stores and prints their results. It consists of three entities, which are Student, Course and Result.

E-R diagram for student management system is shown in Fig.7.10.

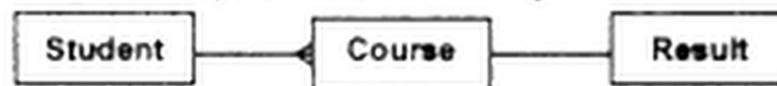


Fig 7.10 Student Management System

**Entities**

1. **Student:** The entity Student is set of all the students and its attributes are Admission\_No, Student\_Name, Address, DOB (Date of Birth) and Tel\_No.
2. **Course:** The entity Course is a set of all the courses offered and its attributes are Course\_Code, Course\_Name, Credits, Semester and Year.
3. **Result:** The entity Result is a set of all the results of students and its attributes are Marks\_Obtained, Max\_Marks and Grade.

**Relationships**

1. The relationship between Student and Course is one-to-many because a student can register for one or more courses.
2. The relationship between Course and/Result is one-to-one because there is only one result for each course.

**Q.27 Draw E-R diagram for ticket booking system.****Answer****Ticket Booking System**

Ticket Booking System has three entities. The entities are Airlines, Flights and Tickets. The E-R diagram for this is shown in Fig.7.11.



Fig.7.11 E-R Diagram of Ticket Booking System

**Entities**

1. **Airline:** The entity Airline is set of all the airlines and its attributes are Airline\_Code and

Airline\_Name.

2. Flight: Attributes for Flight are Flight\_Code, Departure\_Location, Arrival\_Location, Departure\_Time, Arrival\_Time, Departure\_Date, Price and Seats\_Available.
3. Ticket: Attributes for Ticket are Ticket\_No, Passenger\_Name, Address, Tel\_No and Email\_Address.

#### Relationships

1. The type of relationship between Airline and Flight is one-to-many because one airline has many flights
2. The relationship between Flight and Ticket is also one-to-many since many tickets are booked for a single flight

## 7.5 RELATIONAL SCHEMA

**Q.28** What is meant by schema? Briefly explain database schema.

**Answer**

#### Schema

The word schema is a Greek word, which means shape or plan

#### Database Schema

A database schema represents the structure of a database system in a formal language. It refers to the organization of data to construct a database

### 7.5.1 TRANSFORM E-R DIAGRAM TO RELATIONAL SCHEMA

The E-R diagram represents the conceptual level of database design. It describes real-world entities. Although, E-R diagram is constructed in such a way to allow easy transition to relational model, it is not a simple process. A relational schema is at the logical level of database design. In a relational database, the schema defines the objects in the database.

**Q.29** Write down the steps to transform E-R diagram to relational schema.

**Answer**

#### Steps to transform E-R diagram to relational schema

The following steps transform E-R diagram to relational schema:

1. Transforming Entity to Relational Schema

An entity turns into a table.

#### Example

Suppose you have an entity named Student that has attributes, StudentID, Name, Class and Address. The entity Student will turn into a Student table.

2. Transforming Attribute to Relational Schema

Each attribute turns into a column (field) in the table.

#### Example

The attributes of entity named Student will become columns in the Student table.

3. Transforming Relationships to Relational Schema

To transform a relationship in the E-R diagram to relational schema, you have to turn the primary key of the entity to the primary key in the table and make sure that foreign keys are in place in the schema to allow join between the tables

4. Transforming One-to-Many Relationship

To represent the one-to-many relationship, take the primary key of the table on the "one" side and insert it as a foreign key into the table on the "many" side. This is the most common use of a foreign key. It may make sense to rename the foreign key to reflect its relationship to the table you are inserting into

5. Transforming One-to-One Relationship

To represent a one-to-one relationship, you have to decide whether it makes more sense to keep it as two separate tables or to join them together to make one big table. If you have to choose two tables for

in a one-to-one relationship, you must decide which table will have the primary key and which table will have the primary key of the other as foreign key.

**Q.30** Give an example which explain the transformation of E-R diagram to relational schema.

**Answer**

**Example**

Suppose we want to develop a student database management system to store student particulars and their results of various examinations throughout the year. The E-R diagram of this database has two entities, that is, Student and Examination. The relationship between these entities will be one-to-many as shown Fig 7.13 since students have many examinations throughout the year.



Fig.7.13 E-R diagram of Student and Examination

**Entity Student**

It has the attributes, STUDENTID, NAME, CLASS and ADDRESS.

**Entity Examination**

It has the attributes, EXAM NAME, ENG, URDU, ISL, MATHS, PHY and COMP.

**E-R diagram into relational schema**

Now, let us transform this E-R diagram into relational schema.

The entities, Student and Examination, will turn into Student table and Examination table respectively

The attributes of both entities will turn into columns of the Student and Examination tables as shown in Fig.7.14.

**Student Table**

STUDENTID	NAME	CLASS	ADDRESS

**Examination Table**

RESULTID	EXAM NAME	ENG	URDU	ISL	MATH	PHY	COMP

Fig.7.14 Student and Examination tables

**Create relational schema between the entities**

To create relational schema between the entities, STUDENTID will be primary key in that Student table since it has unique values and it will become foreign key in the Examination table. It makes sense to rename the foreign key as RESULTID in the Examination table to reflect its relationship to the Student table. The Student and Examination tables will have one-to-many relationship. For a single record in Student table, there may be multiple corresponding records in Examination table representing results of many examinations throughout the years, such as 1st Term, 2nd Term, Final, etc..

### 7.5.2 NORMALIZATION OF RELATIONAL DATABASE

**Q.31** Why database should be normalize? How many forms of normalization are there?

**Answer**

Databases are normalized to avoid data redundancy and inconsistency.

**Forms of normalization**

There are five normal forms for normalization of a database. The first three normal forms will be

discussed here to understand the concept of normalization.

### Q.32 Explain first normal form.

**Answer**

#### First Normal Form

To get to first normal form, we break up our data into its related data groups.

#### Explanation

To understand how to obtain the first normal form, consider the table shown in Fig. 7.15.

It stores information about members of library and the books borrowed by them.

MemberID	Name	TelNo	BookID	Title	Author	Pub_Date
481	Raja Imran	4815568	20056	Introduction to Java	Ali Ahmed	2006
481	Raja Imran	4815568	36211	Using Internet	Rizwan Khan	2009
532	Nasir Ali	2309116	20568	Computer Networks	Amjad Ali	1998
532	Nasir Ali	2309116	44195	Learning MS Word	Imran Khan	2010
532	Nasir Ali	2309116	31128	Web Designing	Waqas Arif	2011
669	Arif Khan	4212957	62617	Mastering MS Excel	Syed Fawad	2005

Fig. 7.15 Table in Un-normal Form

In this example, the primary key is the MemberID and we have two fields, Name and Tel\_No repeating themselves across the tuples for each of the different BookID entries. The redundant data will take lot of space to store the data and more time will be required to search through.

The rules of first normal form break this table into two and relate them to each. The repeated information in the table is removed. This will create the two tables shown in Fig.7.16.

MemberID	Name	Tel_No
481	Raja Imran	4815568
532	Nasir Ali	2309116
669	Arif Khan	4212957

MemberID	BookID	Title	Author	Pub_Date
481	20056	Introduction to Java	Ali Ahmed	2006
481	36211	Using Internet	Rizwan Khan	2009
532	20568	Computer Networks	Amjad Ali	1998
532	44195	Learning MS Word	Imran Khan	2010
532	31128	Web Designing	Waqas Arif	2011
669	62617	Mastering MS Excel	Syed Fawad	2005

Fig. 7.16 Tables in First Normal Form

The other normal forms are also obtained in a similar way by further breaking up the table and removing repeating information if it exists.

### Q.33 Explain second normal form.

**Answer**

#### Second Normal Form (Removes Partial Functional Dependencies)

To move into second normal form, all partial functional dependencies must be removed. Partial

functional dependencies occur when there is a composite key and only one part of the key is needed to determine one or more other attributes. To remove partial functional dependencies, we have to project out the partial functional dependencies into its own table, creating two tables from one.

### Explanation

In our example, we have to further check if we are keeping more data than we need to in our relation. If we look at Fig. 7.16, we can see that for every UnitCode, we are also storing the UnitName. It will make more sense, if we look up UnitCode 443 in another table and find the UnitName. This is how we can avoid the duplicate information entered in the UnitName field. UnitName is only dependent on UnitCode and has no dependency on StudentID. To be in second normal form, all partial dependencies must be removed. When we apply the rules of second normal form and remove the partial functional dependency in our database, we get the following tables shown in Fig. 7.17, which are in second normal form.

StudentID	UnitCode
481	UC443
481	UC471
481	UC422
532	UC443
669	UC471
669	UC468

UnitCode	UnitName
UC443	Database Design
UC471	Operating System
UC422	Digital Electronics
UC468	Accounting

Fig. 7.17 Table in Second Normal Form

### Q.34 Explain third normal form.

Answer

#### Third Normal Form (Removes Transitive Functional Dependencies)

The next step is to move into third normal form. In order to reach third normal form, all transitive functional dependencies must be removed. Similar to partial functional dependencies, transitive functional dependencies occur when a non-key attribute can determine another non-key attribute. Transitive functional dependencies are removed by projecting out the transitively dependent attributes to another table while leaving the determinate as a foreign key in the original table.

UnitCode	UnitName	CourseCode	CourseName
UC443	Database Design	COMP2010	Computing
UC471	Operating System	COMP2010	Computing
UC422	Digital Electronics	COMP2010	Computing
UC468	Accounting	BUS2010	E-Commerce

Fig 7.18 Table with CourseCode and CourseName attributes

### Explanation

In our example above, we have CourseName that is dependent on CourseCode and

CourseCode is dependent on UnitCode. Therefore, CourseName is transitively dependent on UnitCode. To remove the transitive dependency, break the table into two tables, leaving the determinant as foreign key in the original table. This is shown in the tables of Fig 7.19. Here, CourseCode is foreign key in the first table and CourseCode is primary key in the second table.

UnitCode	UnitName	CourseCode
UC443	Database Design	COMP2010
UC471	Operating System	COMP2010
UC422	Digital Electronics	COMP2010
UC488	Accounting	BUS2010

CourseCode	CourseName
COMP2010	Computing
BUS2010	E-Commerce

Fig 7.19 Table in Third Normal Form

### KEY POINTS

- ☞ Data refers to facts and figures. It is raw material that is input in the computer for processing.
- ☞ Information is organized form of data that is meaningful and easily understandable by computer users. It is the output produced by computer after processing the data.
- ☞ A database is a collection of related data. A database consists of one or more data files and data files contain fields.
- ☞ A Database Management System (DBMS) is a collection of programs that enables users to create and maintain a database.
- ☞ Database administrator (DBA) is the person in charge for supervising the database and that use of DBMS in an organization.
- ☞ In a Relational Database Model, data is stored in tables and the tables are linked by means of common fields. Each column of table holds one field and each row one record. It is the most commonly used database model.
- ☞ All the information about one person or item is held in a record. When records are stored in tables, rows represent records and columns represent fields.
- ☞ Key is an attribute (or field) that is used to identify records in tables.
- ☞ Entity is a thing of interest to an organization about which data is to be held. For example employees in an organization or books in a library.
- ☞ Attribute is a property or characteristics of an entity.
- ☞ Relationship is a link between entities. In a relational database, it is the link between tables.
- ☞ Data modeling is the process of creating a conceptual data model that identifies the data needed by an organization to achieve its objectives.
- ☞ An instance in a database is the actual content of the database at a particular point in time.
- ☞ Cardinality refers to the maximum number of times an instance in one entity can be associated with instances in the related entity.
- ☞ Modality refers to the minimum number of times an instance in one entity can be associated with an instance in the related entity.
- ☞ Entity-Relationship (E-R) diagram is a diagrammatic way of representing the relationship between the entities of a database.
- ☞ Normalization is a series of steps to create normal forms used to ensure that data anomalies are not created when information is inserted, updated or deleted.

