UNIT - I

1 Introduction to DBMS

Syllabus

Introduction : *Basic concepts, Advantages of DBMS over file processing systems, Data abstraction, Database languages, Data models, Data independence, Components of a DBMS, Overall structure of DBMS, Multi-user DBMS architecture, System catalogs, Data Modeling : Basic concepts, Entity, Attributes, Relationships, Constraints, Keys.*

Contents

Part I : Introduction

1.1 Basic Concepts

- **Definition :** A database management system (DBMS) is collection of **interrelated data** and various **programs** that are used to handle the data.
- The **primary goal** of DBMS is to provide a way to **store and retrieve** the required information from the database in convenient and efficient manner.
- For managing the data in the database two important **tasks** are conducted -
  - **Define the structure** for storage of information.
  - Provide mechanism for manipulation of information.
- In addition, the database systems must ensure the **safety of information** stored.

Database System Applications

There are wide range of applications that make use of database systems. Some of the applications are -

1) **Accounting :** Database systems are used in maintaining information employees, salaries, and payroll taxes.
2) **Manufacturing :** For management of supply chain and tracking production of items in factories database systems are maintained.
3) For maintaining customer, product and purchase information the databases are used.
4) **Banking :** In banking sector, for customer information, accounts and loan and for performing banking applications the DBMS is used.
5) For purchase on credit cards and generation of monthly statements database systems are useful.
6) **Universities :** The database systems are used in universities for maintaining student information, course registration, and accounting.
7) **Reservation systems :** In airline/railway reservation systems, the database is used to maintain the reservation and schedule information.
8) **Telecommunication :** In telecommunications for keeping records of the calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about communication networks the database systems are used.

1.1.1 Characteristics of Database Approach

Following are the characteristics of database system :

1) Representation of some aspects of **real world applications**.
2) Systematic **management of information**.
3) Representing the data by **multiple views**.
4) Efficient and easy implementation of various **operations** such as insertion, deletion and updation.
5) It maintains **data for** some specific **purpose**.
6) It represents **logical relationship** between records and data.

1.2 Advantages of DBMS over File Processing Systems

SPPU : May-18, Marks 5

1.2.1 Advantages of DBMS

Following are the advantages of DBMS :

1) DBMS **removes the data redundancy** that means there is no duplication of data in database.
2) DBMS allows to **retrieve the desired data** in required format.
3) **Data** can be **isolated** in separate tables for convenient and efficient use.
4) Data can be **accessed efficiently** using a simple query language.
5) The **data integrity** can be maintained. That means – the constraints can be applied on data and it should be in some specific range.
6) The **atomicity** of data can be maintained. That means, if some operation is performed on one particular table of the database, then the change must be reflected for the entire database.
7) The DBMS allows **concurrent access** to multiple users by using the synchronization technique.
8) The **security policies** can be applied to DBMS to allow the user to access only desired part of the database system.

1.2.2 Disadvantages of DBMS

1) **Complex design :** Database design is complex, **difficult and time consuming**.
2) **Hardware and software cost : Large amount of investment** is needed to setup the required hardware or to repair software failure.

3) **Damaged part :** If one part of database is corrupted or damaged, then **entire database** may get affected.

4) **Conversion cost :** If the current system is in conventional file system and if we need to convert it to database systems then large amount of cost is incurred in purchasing different tools, and adopting different techniques as per the requirement.

5) **Training :** For designing and maintaining the database systems, the people need to be trained.

1.2.3 File Processing System Vs. DBMS

- Earlier database systems are created in response to manage the commercial data. These data is typically stored in files. To allow users to manipulate these files, various programs are written for
  1) Addition of new data
  2) Updating the data
  3) Deleting the data.
- As per the need for addition of new data, separate application programs were required to write. Thus as the time goes by, the system acquires more files and more application programs.
- This typical **file processing system** is supported by conventional operating system. Thus the file processing system can be described as -
- "The system that stores the permanent records in files and it needs different application programs to extract or add the records".
- Before introducing database management system, this file processing system was in use. However, such a system has many drawbacks. Let us discuss them.

Disadvantages of Traditional File Processing System

The traditional file system has following disadvantages :

1) **Data redundancy :** Data redundancy means duplication of data at several places. Since different programmers create different files and these files might have different structures, there are chances that some information may appear repeatedly in some or more format at several places.

2) **Data inconsistency :** Data inconsistency occurs when various copies of same data may no longer get matched. For example changed address of an employee may be reflected in one department and may not be available (or old address present) for other department.

3) **Difficulty in accessing data :** The conventional file system does not allow to retrieve the desired data in efficient and convenient manner.

4) **Data isolation :** As the data is scattered over several files and files may be in different formats, it becomes difficult to retrieve the desired data from the file for writing the new application.

5) **Integrity problems :** Data integrity means data values entered in the database fall within a specified range and are of specific format. With the use of several files enforcing such constraint on the data becomes difficult.

6) **Atomicity problems :** An atomicity means particular operation must be carried out entirely or not at all with the database. It is difficult to ensure atomicity in conventional file processing system.

7) **Concurrent access anomalies :** For efficient execution, multiple users update data simultaneously, in such a case data need to be synchronized. As in traditional file systems, data is distributed over multiple files, one cannot access these files concurrently.

8) **Security problems :** Every user is not allowed to access all the data of database system. Since application program in file system are added in an ad hoc manner, enforcing such security constraints become difficult.

Database systems offer solutions to all the above mentioned problems.

Difference between Database System and Conventional File System

Sr. No.	Database systems	Conventional file systems
1.	Data redundancy is **less**.	Data redundancy is **more**.
2.	**Security** is **high**.	Security is very **low**.
3.	Database systems are used when security **constraints are high**.	Conventional file systems are used where there is **less demand** for **security constraints**.
4.	Database systems define the data in a **structured** manner. Also there is well defined co-relation among the data.	File systems define the data in **un-structured** manner. Data is usually in isolated form.
5.	Data **inconsistency** is **less** in database systems.	Data **inconsistency** is **more** in file systems.
6.	**User is unknown** to the physical address of the data used in database systems.	User locates the **physical address of file** to access the data in conventional file systems.
7.	We can retrieve the data in any **desired format** using database systems.	We cannot retrieve the data in any desired format using file systems.
8.	There is ability to access the data **concurrently** using database systems.	There is **no ability to concurrently** access the data using conventional file system.

Review Question

1. *State and explain the disadvantages of the File processing system.* **SPPU : May-18, (End Sem), Marks 5**

1.3 Data Abstraction

Definition of data abstraction : Data abstraction means retrieving only the required amount of information about the system and **hiding** background details.

There are several levels of abstraction that simplify user interactions with the system. These are :

1) Physical level :

- This is the **lowest level.**
- This level describes how the data are **stored.**
- The database administrators decide how to store data at the physical level.
- This level describes complex low-level data structures.

2) Logical level :

- This is the next **higher level,** which describes what data are stored in the database?.
- This level also describes the **relationship between the data.**
- The logical level thus describes the entire database in terms of a small number of relatively simple structures.
- The database administrators use a logical level of abstraction for deciding what information to keep in the database.

3) View level :

- This is the **highest level** of abstraction that describes only part of the entire database.
- The view level can provide access to only part of the database.
- This level helps in **simplifying the interaction** with the system.
- It can provide **multiple views** of the same system.
- For example – A Clerk at the reservation system can see only part of the database and access the passenger's required information.

Fig. 1.3.1 shows the relationship between the three levels of abstraction.

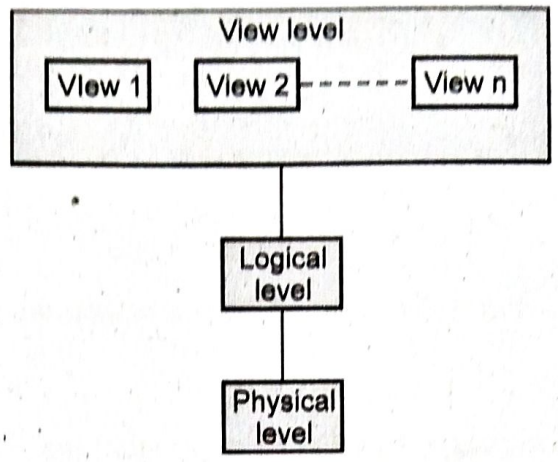


Fig. 1.3.1 : Levels of data abstraction

For example : Consider following record

```
type employee = record
                    empID : numeric(10)
                    empname : char(20)
                    dept_no : numeric(10)
                    salary : numeric(8,2)
                    end
```

This code defines a new record **employee** with four fields. Each field is associated with field name and its **type**. There are several other records such as **department** with fields dept_no, dept_name, building customer with fields cust_id,cust_name .

- **At the physical level**, the record - **customer, employee, department** can be described as block of **consecutive storage locations**. Many database systems hide lowest level storage details from database programmer.
- The type definition of the records is decided **at the logical level**. The **programmer** work of the record at this level, similarly **database administrators** also work at this level of abstraction.
- There is specific view of the record is allowed at the view level. For instance - customer can view the name of the employee, or id of the employee but cannot access employee's salary.

1.4 Database Languages

There are three types of languages supported by database systems.

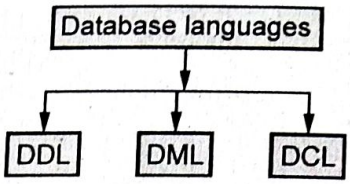


Fig. 1.4.1 Types of database languages

(1) DDL

- Data Definition Language (DDL) is a specialized language used to specify a database schema by a set of definitions.
- It is a language used for **creating** and **modifying** the structures of tables, views, indexes, etc.
- DDL is also used to specify additional properties of data.
- Some of the common commands used in DDL are -**CREATE, ALTER, DROP.**
- The **primary use** of CREATE command is to build a new table. Using ALTER command, the users can add up some additional column and drop existing columns. Using DROP command, the user can delete table or view.

(2) DML

- DML stands for Data Manipulation Language.
- This language enables users to access or manipulate data as organized by appropriate data model.
- The types of access are -
  - **Retrieval** of information stored in the database
  - **Insertion** of new information into the database.
  - **Deletion** of information from the database.
  - **Modification** of information stored in database.
- There are two types of DML -
  - **Procedural DML** - Require a user to specify what data are needed and how to get those data.
  - **Declarative DML** - Require a user to specify what data are needed without specifying how to get those data.
- **Query** is a statement used for requesting the retrieval of information. This retrieval of information using some specific language is called **query language.**

(3) DCL

- The Data Control Language (DCL) is used to control access to data stored in the database. This is also called as authorization.
- The typical command used in DCL are GRANT and REVOKE.
  - **GRANT :** This command is used to give access rights or privileges to the database.
  - **REVOKE :** The revoke command removes user access rights or privileges to the database objects

1.5 Data Models

SPPU : Nov.-19, Marks 5

- **Definition :** It is a collection of conceptual tools for describing data, relationships among data, semantics (meaning) of data and constraints.
- Data model is **a structure below the database.**
- Data model provides a way to describe the design of database at physical, logical and view level.
- There are various data models used in database systems and these are as follows -

(1) Relational model :

- The relation model consists of **collection of tables** which stores data and also represents the relationship among the data.
- The **table** is also known as **relation.**
- The table contains one or more **columns** and each column has unique name.
- Each table contains **record of particular type,** and each record type defines a fixed **number of fields or attributes.**
- **For example** – The following figure shows the relational model by showing the relationship between Student and Result database. For example – Student **Ram** lives in city **Chennai** and his marks are **78.** Thus the relationship between these two databases is maintained by the **SeatNo.** Column

Seat No	Name	City
101	Ram	Chennai
102	Shyam	Pune

SeatNo	Marks
101	78
102	95

Advantages :

(i) **Structural independence :** Structural independence is an ability that allows us to make changes in one database structure without affecting other. The relational model have structural independence. Hence making required changes in the database is convenient in relational database model.

(ii) **Conceptual simplicity :** The relational model allows the designer to simply focus on logical design and not on physical design. Hence relational models are conceptually simple to understand.

(iii) **Query capability :** Using simple query language (such as SQL) user can get information from the database or designer can manipulate the database structure.

(iv) **Easy design, maintenance and usage :** The relational models can be designed logically hence they are easy to maintain and use.

Disadvantages :

i) Relational model **requires powerful hardware** and large data storage devices.

ii) May lead to **slower processing time.**

iii) Poorly designed systems lead to **poor implementation** of database systems.

(2) Entity relationship model :

- As the name suggests the entity relationship model uses collection of basic objects called **entities** and **relationships.**
- The entity is a thing or object in the real world.
- The entity relationship model is widely used in database design.
- For example - Following is a representation of Entity Relationship model in which the relationship **works_for** is between entities **Employee** and **Department.**

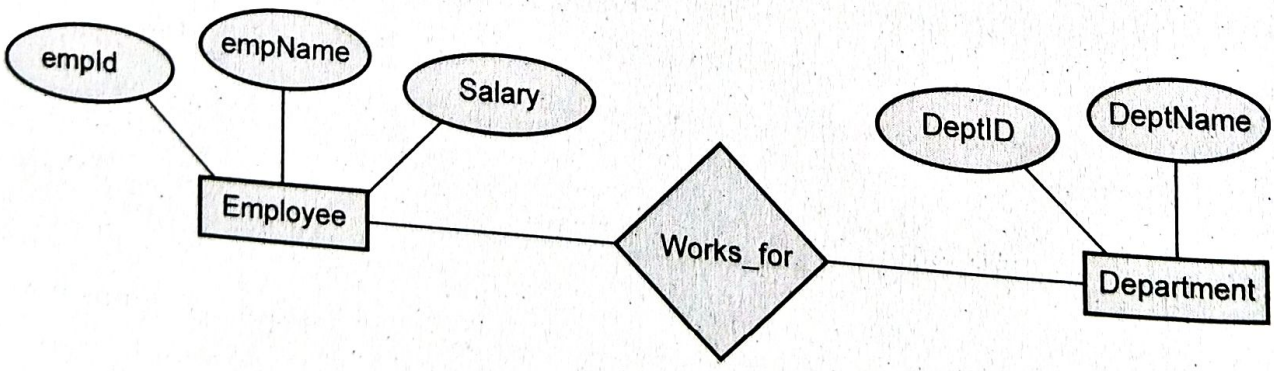


Fig. 1.5.1

Advantages :

i) **Simple :** It is simple to draw ER diagram when we know entities and relationships.

ii) **Easy to understand :** The design of ER diagram is very logical and hence they are easy to design and understand.

iii) **Effective :** It is effective communication tool.

iv) **Integrated :** The ER model can be easily integrated with Relational model.

v) **Easy conversion :** ER model can be converted easily into other type of models.

Disadvantages :

i) **Loss of information :** While drawing ER model some information can be hidden or lost.

ii) **Limited relationships :** The ER model can represent limited relationships as compared to other models.

iii) **No representation for data manipulation :** It is not possible to represent data manipulation in ER model.

iv) **No industry standard :** There is no industry standard for notations of ER diagram.

(3) Object Based Data Model :

- The **object oriented languages** like C++, Java, C# are becoming the dominant in software development.
- This led to object based data model.
- The object based data model combines **object oriented features** with relational data model.

Advantages :

i) **Enriched modelling :** The object based data model has capability of modelling the real world objects.

ii) **Reusability :** There are certain features of object oriented design such as inheritance, polymorphism which help in reusability.

iii) **Support for schema evolution :** There is a tight coupling between data and applications, hence there is strong support for schema evolution.

iv) **Improved performance :** Using object based data model there can be significant improvement in performance using object based data model.

Disadvantages :

i) **Lack of universal data model :** There is no universally agreed data model for an object based data model, and most models lack a theoretical foundation.

ii) **Lack of experience :** In comparison with relational database management the use of object based data model is limited. This model is more dependent on the skilled programmer.

iii) **Complex :** More functionalities present in object based data model make the design complex.

(4) Semi-structured data model :

- The semi-structured data model permits the **specification of data** where individual data items of same type may have **different sets of attributes.**
- The Extensible Markup Language **(XML)** is widely used to represent semi-structured data model.

Advantages

i) Data is not constrained by fixed schema.

ii) It is **flexible.**

iii) It is **portable.**

Disadvantage

i) **Queries** are **less efficient** than other types of data model.

(5) Hierarchical Model

- In this model each entity has only **one parent** but can have **several children.** At the top of hierarchy there is only one node called **root**. Refer Fig. 1.5.2.
- This model represents the relationship in 1 : N types. That means one university can have multiple courses. One course can have multiple projects and so on.

Advantage

1. This model groups the data into tables and defines the relationship between the tables.

Disadvantages

1. For searching any data, we have to start form the root and move downwards and visit each child node. Thus traversing through each node is required.
2. For addition of some information about child node, sometimes the parent information need to be modified.

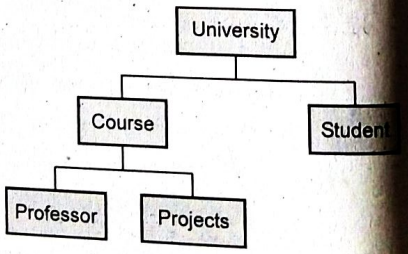


Fig. 1.5.2 Hierarchical model

3. It fails to handle many to many relationship (M : N) efficiently.
   It can cause duplication and data redundancy.

(6) Network Model

- This is enhanced version of hierarchical model. It overcomes the drawback of hierarchical model. It helps to address M : N relationship. That means, this model is not having single parent concept. Any child in this model can have multiple parents. Refer Fig. 1.5.3.
- The **main difference** between network model and hierarchical model is to allow many to many relationship.

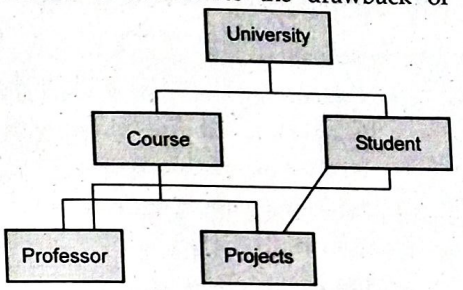


Fig. 1.5.3 Network model

Advantages

1. **Capability to handle more Relationships :** Since the network model allows many to many relationship, it helps in modeling the real life situations.
2. **Ease of data access :** The data access is easier and flexible than hierarchical model.
3. **Data Integrity :** In network model every member is associated with some other member in the model.
4. **Conformance to Standards :** The network model structure can be designed as per the standards.

Disadvantages

1. **Complex to implement :** For all the records the pointers need to be maintained, hence the database structure becomes complex.
2. **Complicated Operations :** The simple operations such as insertion, deletion and modification becomes complex due to adjustment of multiple pointer.
3. **Difficult to change structure :** The structural changes are difficult.

Review Question

1. *List out different Data Models. Explain any two.* **SPPU : Nov.-19, (End Sem), Marks 5**

1.6 Components of a DBMS

SPPU : Nov.-17,18, Marks 6

The functional components of database systems can be broadly classified into :

i) Storage manager and **ii) Query processor components.**

- Storage manager is to maintain the large storage space required for database.
- Query processor is helps to access the data of the database.

Let us first discuss the overall structure of DBMS.

1.6.1 Overall Structure of DBMS

Three Schema Architecture

- **Definition :** Database schema is a collection of database objects like tables, views, indexes and so on associated with one particular database username. This username is called the **schema owner**.
- For example Student Schema can be owner of STUDENT and MARKS tables. The Course schema can be the owner of SUBJECT table.
- The **goal** of three-schema architecture is to separate the user application from the physical database.
- The architecture of database is divided into **three levels** based on three types of schema - internal schema, conceptual schema or external schema.

1. Internal level :

- It contains **internal schema**.
- This schema represents the **physical storage structure of database**.
- This schema is maintained by the software and user is not allowed to modify it.
- This level is closest to the physical storage. It typically describes the record layout of the files and types of files, access paths etc.

2. Conceptual level :

- It contains **conceptual schema**.
- This schema **hides the details** of internal level.
- This level is also called as logical level as it contains the constructs used for designing the database.
- It contains information like table name, their columns, indexes and constraints, database operations.
- A representational data model is used to describe conceptual schema when a database system is implemented.

3. External level :

- It contains the external schema or user views.
- At this level, the user will get to see only the data stored in the database. Either they

will see whole data values or any specific records. They will not have any information about how they are stored in the database.

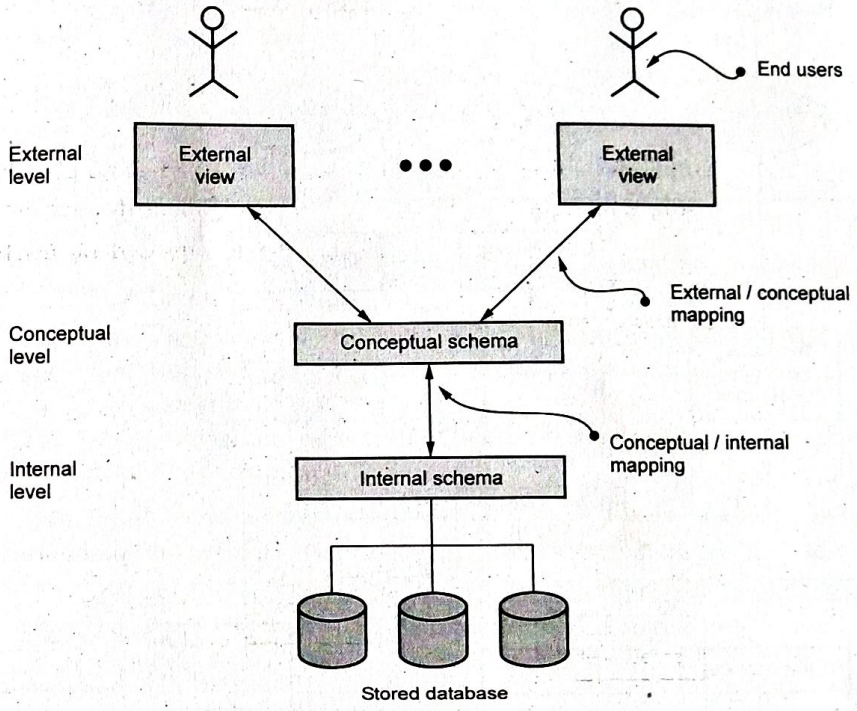


Fig. 1.6.1 Three schema architecture

- The processes of transforming requests and results between levels are called **mappings.**
- In the three schema architecture there are two mappings –

  1) External - Conceptual Mapping and

  2) Conceptual - Internal Mapping

1.6.2 Architecture of DBMS

- The typical structure of typical DBMS is based on relational data model as shown in Fig. 1.6.2. (Refer page 1-16).
- Consider the top part of Fig. 1.6.2. It shows **application interfaces** used by naïve users, application programs created by application programmers, query tools used by sophisticated users and administration tools used by database administrator
- The lowest part of the architecture is for **disk storage.**

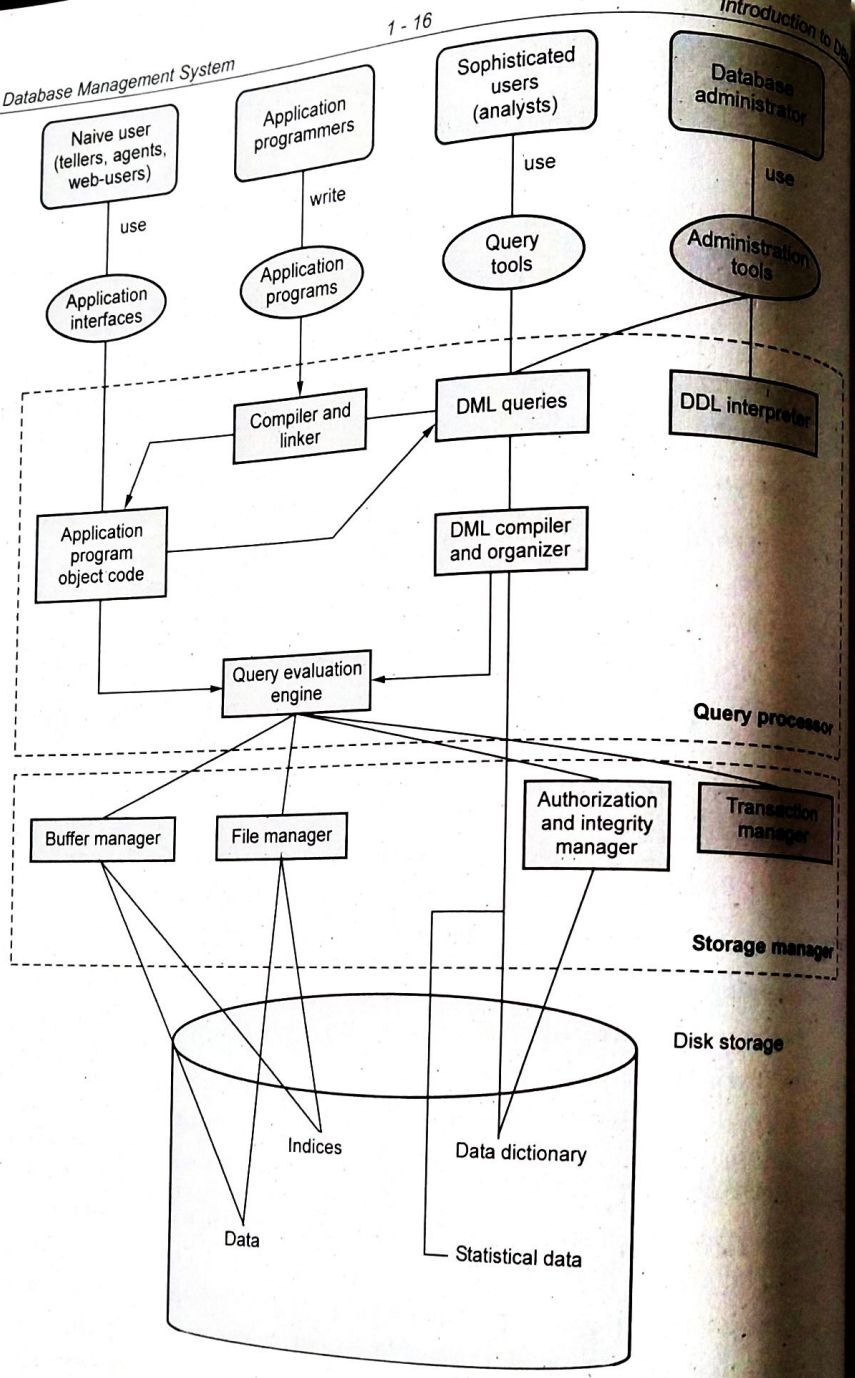


Fig. 1.6.2 Architecture of database

- The two important components of database architecture are - **Query processor and storage manager.**

Query processor :

- The interactive query processor helps the database system to simplify and facilitate access to data. It consists of DDL interpreter, DML compiler and query evaluation engine.
- With the following components of query processor, various functionalities are performed -
  - i) **DDL interpreter :** This is basically a translator which interprets the DDL statements in data dictionaries.
  - ii) **DML compiler :** It translates DML statements query language into an evaluation plan. This plan consists of the instructions which query evaluation engine understands.
  - iii) **Query evaluation engine :** It executes the low-level instructions generated by the DML compiler.
- When a user issues a query, the parsed query is presented to a query optimizer, which uses information about how the data is stored to produce an efficient execution plan for evaluating the query. An execution plan is a blueprint for evaluating a query. It is evaluated by **query evaluation engine**.

Storage Manager:

- Storage manager is the component of database system that provides interface between the low level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible for storing, retrieving, and updating data in the database. The storage manager components include -
  - i) **Authorization and integrity manager :** Validates the users who want to access the data and tests for integrity constraints.
  - ii) **Transaction manager :** Ensures that the database remains in consistent despite of system failures and concurrent transaction execution proceeds without conflicting.
  - iii) **File manager :** Manages allocation of space on disk storage and representation of the information on disk.
  - iv) **Buffer manager :** Manages the fetching of data from disk storage into main memory. The buffer manager also decides what data to cache in main memory. Buffer manager is a crucial part of database system.

- Storage manager implements several data structures such as -

i) **Data files :** Used for storing database itself.

ii) **Data dictionary :** Used for storing metadata, particularly schema of database.

iii) **Indices :** Indices are used to provide fast access to data items present in the database

1.6.3 Multi-user DBMS Architecture

Single Tier Architecture

- In this architecture application logic, presentation and data management all are combined in a single tier. For running the application, there was a use of mainframes. The application was accessible by **dumb terminals** that could perform only data input and display. It is as shown by following figure -

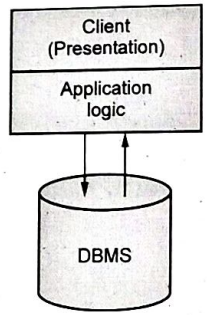


Fig. 1.6.3 Single tier architecture

- The advantage of this architecture is that it is easily maintained by central administrator.
- The main drawback of this architecture is that - it could support limited number of users.

Client Server Architecture

- The client server architecture is also called as **two tier architecture.**
- It consists of **Client computer** and **Server computer** which can communicate through well -defined protocol.

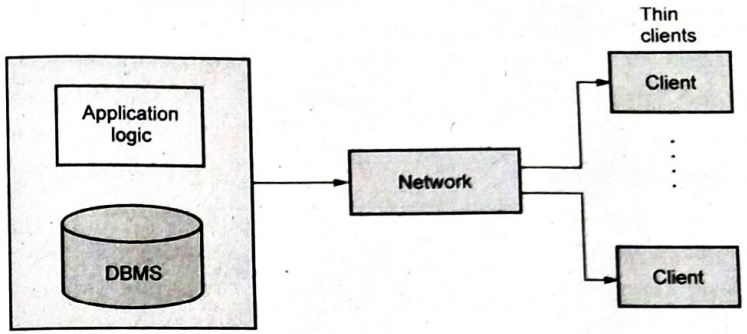


Fig. 1.6.4 Two server architecture

- In traditional client server architecture, the client computer implements simply the graphical user interface part. The Server computer implements application logic and data management part. In such architecture, the client is called as **thin client**.
- On the other hand there is some client server architecture, in which client implements graphical interface as well as business (application) logic part. Only data management part is taken care by server. Such client is called as **thick client.**
- The thick client model has several disadvantages as compared to thin client. Those are -
  - There is no central place to update and maintain the business logic as application program runs on the client computer only.
  - Client should run the business logic with reliability without affecting any security aspects.
  - The thick client architecture can handle only limited number of users.

Three Tier Architecture

The three tier architecture is made up of three tiers as –

(1) Presentation Tier : The presentation tier is comprised of graphical user interface. The user always expects the GUI which is easy to input. He/She expects the results in some organized format. The use web-based interfaces are getting popular.

(2) Middle Tier : This is a tier in which the application or business logic executes. The complex business processes get executed at this tier. The business logic can be implemented in some suitable programming language like C++ or Java.

(3) Data Management Tier : This tier takes care of data management activities. The database management (DBMS) systems are located in this architecture.

The typical three tier architecture is represented by Fig. 1.6.5.

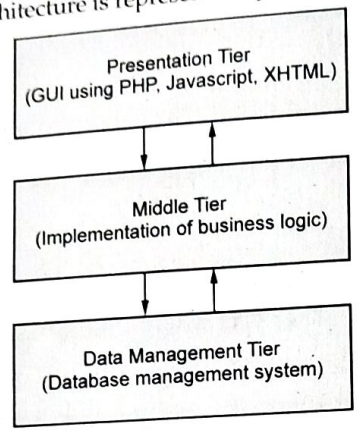


Fig. 1.6.5 Three tier architecture

Advantages of Three Tier Architecture

The Three tier architecture has following advantages –

(1) **Heterogeneous Systems :** Different platforms and different software components can be used at different tiers. Also it is easy to replace or modify some code present at any tier without affecting the other code.

(2) **Thin Clients :** Clients need enough computation power for presentation layer.

(3) **Integrated Data Access :** In many applications, data must be accessed from several sources. This is can be done transparently in three tier architecture by using the middle tier.

(4) **Scalability to Many Clients :** Multiple clients can access the system through middle tier.

(5) **Software Development Benefits :** By separating presentation, business logic and data management activities by means of separate tiers, it is easy to debug, maintain and modify the system as per the requirements. Interaction between tiers occur through well defined standardized APIs (Application Programming interfaces). Hence it is possible to create reusable components using three tier architecture.

Review Questions

1. *Draw and list different components of database system structure.* **SPPU : Nov.-17, (End Sem), Marks 4**
2. *Explain different database architectures.* **SPPU : Nov 17, (End Sem), Marks 6**
3. *Describe the three level architecture of DBMS. Explain how it is useful for achieving data independence.* **SPPU : Nov.-18, (End Sem), Marks 5**

1.7 Data Independence

SPPU : Nov.-19, Oct.-19, Marks 5

- **Definition :** Data independence is an ability by which one can change the data at one level without affecting the data at another level. Here level can be **physical, conceptual** or **external.**
- Data independence is one of the important characteristics of database management system.
- By this property, the structure of the database or the values stored in the database can be easily modified by without changing the application programs.
- There are two types of data independence

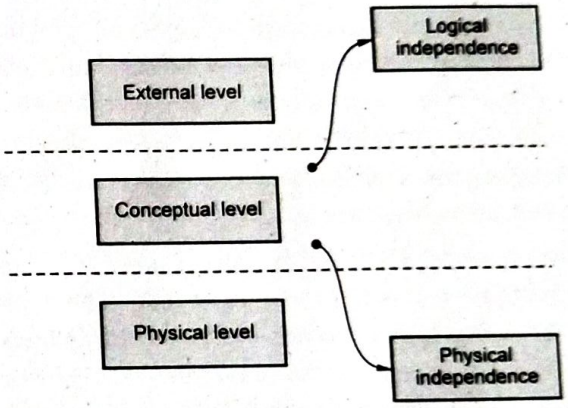


Fig. 1.7.1 Data independence

1. **Physical Independence :** This is a kind of data independence which allows the modification of physical schema without requiring any change to the conceptual schema. For example - if there is any change in memory size of database server then it will not affect the logical structure of any data object.
2. **Logical Independence :** This is a kind of data independence which allows the modification of conceptual schema without requiring any change to the external schema. For example - Any change in the table structure such as addition or deletion of some column does not affect user views.

- By these data independence the time and cost acquired by changes in any one level can be reduced and abstract view of data can be provided to the user.

Review Questions

1. *Describe the three level architecture of DBMS. Explain how is it useful for achieving data independence.* **SPPU : Nov.-19, (End Sem), Marks 5**
2. *Differentiate between logical and physical data independence.* **SPPU : Oct.-19, (In Sem), Marks 2**

SPPU : May 19, Nov.-18, Marks 5

1.8 Database Users

There **four different types** of database system **users** differentiated by the way they interact with the system. Different types of user interfaces for different types of users are -

i) **Naïve users :** This type of users interact with the system with the help of previously created program(known as application program). Typically a form interface is used by this type of user to interact with the system. For example - we may feel up the booking form for booking a ticket on an online system.

ii) **Application programmers :** These are computer professionals who write application programs. Normally Rapid Application Development(RAD) tools are used to quickly design forms and reports.

iii) **Sophisticated users :** These are the type of users who interact with the system without writing programs. These users may submit the database query to retrieve the desired information or tools from data analysis software. Database analysts fall in this category of database users.

iv) **Specialized users :** Specialized users are sophisticated users who write specialized database application that does not fit into the traditional data-processing framework. Among these applications are computer aided-design systems, knowledge-base and expert systems etc.

Responsibilities of DBA

Database Administrator (DBA) is a person who have a **central control** over both data and programs that access data in DBMS. The **functions of DBA** are -

i) **Schema definition :** DBA creates a database Schema using DDL statements.

ii) **Schema and physical organization modification :** In order to improve the overall performance of database management system DBA carried out changes in schema or physical organization.

iii) **Granting authorization for data access :** Granting authorization for data access means giving special permissions to the users for accessing the database. This task is done by DBA so that privacy of database can be maintained.

iv) **Maintenance :** Maintenance involve different types of tasks such as monitoring jobs and their performances, taking periodic back-ups, making free space available for normal operations or upgrading disk space as per the requirements. These all tasks are carried out by DBA.

Example 1.8.1 *Explain the problems that may arrive if the DBA does not discharge the responsibilities properly.* **SPPU : May 19, (End Sem), Marks 5**

Solution : Following are the problems that may arrive if DBA does not discharge the responsibilities properly -

1) The database can not perform without file manger interaction. If nothing is stored in the files then obviously we can not retrieve anything.
2) The consistency in database must operations must be maintained. If it is not, then it will create major problems. For instance – account balance may go below the minimum allowed, employees can earn too much overtime and so on.
3) If authorization for the authentic user is not done, then unauthorized users may access the database or users authorized to access part of the database may be able to access parts of the database for which they lack authority.
4) Data can be lost permanently.
5) Consistency constraints may be violated despite proper integrity enforcement in each transaction. For example, incorrect bank balances might be reflected due to simultaneous withdrawals and deposits, and so on.

Review Question

1. *List the responsibilities of DBA.* **SPPU : Nov.-18, (End Sem), Marks 5**

1.9 System Catalogs

- The system catalogue is a collection of tables and views that contain important information about a database or metadata(meta data means: data about data).
- For each scheme, following things must be at-least included –
  - The names of the tables in the database
  - The names of columns of each table
  - The data type of each column
  - The constraints present on the tables(primary key,NULL, NOT NULL, and some other types of constraints)
  - The access privileges for the elements of the databases.

- The storage structures and indexes

Such a database is often called as **system catalog**.

- Basically the system catalog defines the structure of database. It is sometimes referred as data dictionary.
- The system catalogue is basically a group of objects that contain information that defines other objects in the database.
- For example – Consider Student table as given below and corresponding system catalog.

Roll	Name	RegID	CourseID
101	Ankita	AA123	101
102	Aniket	AA227	105
103	Ashwini	SA823	101
104	Siddharth	HQ256	103
105	Nandan	YG491	105

System catlog (Meta data)

Table name	Column	Data type	Description
Student	Roll	int	Primary key of table
Student	Name	varchar(30)	Student's name
Student	RegID	varchar(10)	Registration ID
Student	CourseID	int	Course ID, Ref : Course table

Part II: Data Modelling

1.10 Basic Concepts

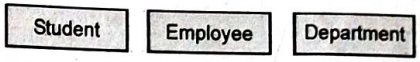
SPPU : May-18, Nov.-17,19, Marks 5

- Data Modelling in database management system is based on Entity Relationship modelling(ER Model)
- Entity Relational model is a model for identifying entities to be represented in the database and representation of how those entities are related.
- ER data model represents the overall **logical structure** of database.
- The E-R model is very useful in mapping the meanings and interactions of real-world entities onto a conceptual schema or database.
- The ER model consists of three basic concepts –

1. Entities 2. Relationships 3. Attributes

1.10.1 Entity and Entity Sets

- **Entity :** An entity is an object that exists and is distinguishable from other objects. For example - Student named "Poonam" is an entity and can be identified by her name. The entity can be concrete or abstract. The concrete entity can be - Person, Book, Bank. The abstract entity can be like - holiday, concept entity is represented as a box.



- **Entity set :** The entity set is a set of entities of the same types. For example - All students studying in class X of the School. The entity set need not be disjoint. Each entity in entity set have the same set of attributes and the set of attributes will distinguish it from other entity sets. No other entity set will have exactly the same set of attributes.

1.10.2 Relationship and Relationship Sets

- **Relationship:** Relationship is an association among two or more entities.
- **Relationship Set:** The **relationship set** is a collection of similar relationships. For example - Following Fig. 1.10.1 shows the relationship **works_for** for the two entities **Employee** and **Departments.**

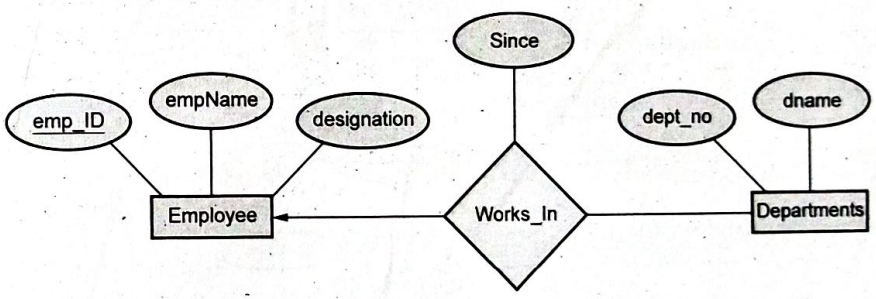


Fig. 1.10.1 : Relation set

- The association between entity sets is called as **participation**. that is, the entity sets E1, E2, . . . , En participate in relationship set R.
- The function that an entity plays in a relationship is called that entity's **role**.

1.10.3 Attributes

- Attributes define the properties of a data object of entity. For example: if student is an entity, his ID, name, address, date of birth, class are its attributes. The attributes help in determining the unique entity. Refer Fig. 1.10.2 for Student entity set with

attributes - ID, name, address. Note that entity is shown by rectangular box and attributes are shown in oval. The primary key is underlined.

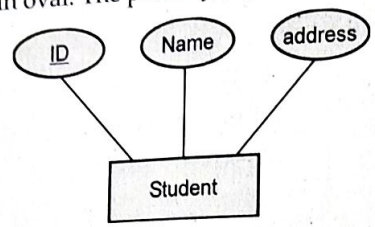


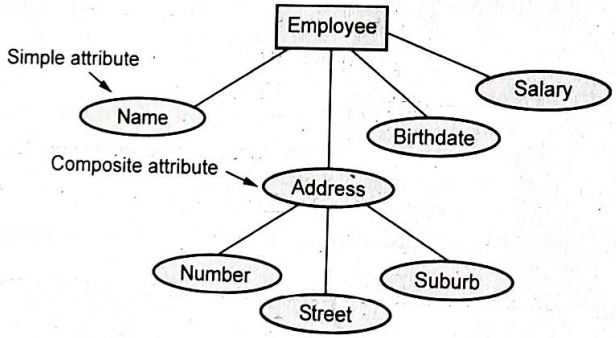
Fig. 1.10.2 : Student entity set with attributes

1.10.3.1 Types of Attributes

Following are the types of attributes -

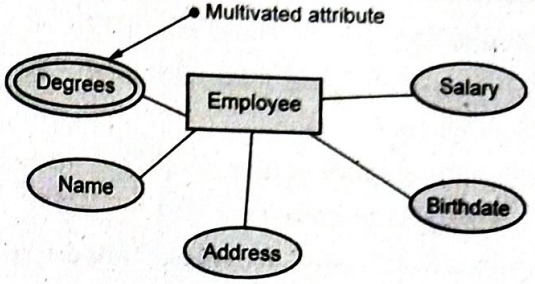
1) Simple and Composite Attributes :

1) Simple attributes are attributes that are drawn from the atomic value domains
   For example - Name = {Parth} ; Age = {23}
2) Composite attributes: Attributes that consist of a hierarchy of attributes
   For example – Address may consists of "Number", "Street" and "Suburb"
   Hence, **Address = {59 + 'JM Road' + 'ShivajiNagar'}**



2) Single valued and multivalued :

- There are some attributes that can be represented using a single value. For example - StudentID attribute for a Student is specific only one studentID.
- Multivalued attributes : Attributes that have a set of values for each entity. It is represented by concentric ovals
  For example - Degrees of a person: ' BSc' , 'MTech', 'PhD'



3) Derived attribute :

Derived attributes are the attributes that contain values that are calculated from other attributes. To represent derived attribute there is dotted ellipse inside the solid ellipse. For example – Age can be derived from attribute DateOfBirth. In this situation, DateOfBirth might be called Stored Attribute.

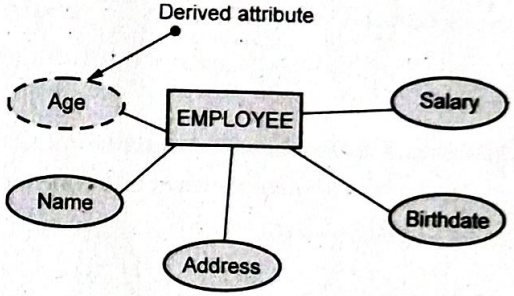


Fig. 1.10.3

Review Questions

1. *Define entity and entity-set. What is E-R model?* **SPPU : May-18. (End Sem). Marks 5**
2. *Explain different types of attributes of an entity with example.* **SPPU : Nov.-17. (End Sem). Marks 3; Nov.-19. (End Sem). Marks 4**

1.11 Constraints

- Relationship types have certain rules that limit the possible combination of entities that can take part in relationship. These rules or restrictions are called **structural constraints.**
- The common type of structural constraint is represented by the **cardinality ratio.**
- The **cardinality ratio** for a binary relationship specifies the **maximum** number of relationship instances that an entity can participate in.

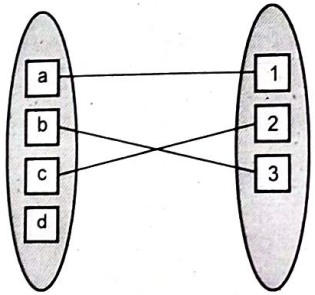
1.11.1 Types of Cardinality

Mapping Cardinality represents the number of entities to which another entity can be associated via a relationship set.

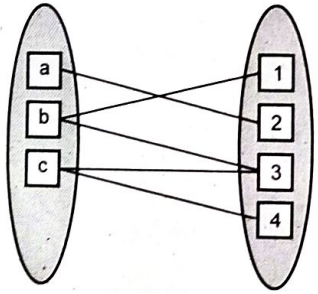
The mapping cardinalities are used in representing the binary relationship sets.

Various types of mapping cardinalities are -

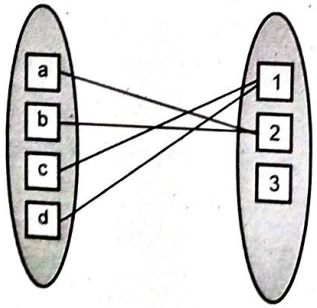
1. **One to One :** An entity A is associated with at least one entity on B and an entity B is associated with at one entity on A. This can be represented as



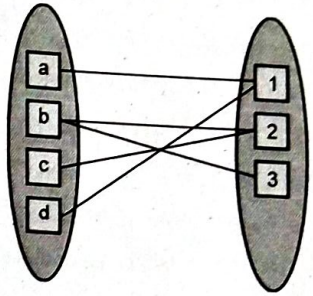
2. **One to Many :** An entity in A is associated with any number of entities in B. An entity in B, however, can be associated with at most one entity in A.



3. **Many to One :** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number of entities in A.



4. **Many to many :** An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.



1.12 Keys

- Keys are used to identify entities uniquely from the given entity set.
- A key can be a an attribute or a set of attributes that help us to identify the entity uniquely.
- Keys also help to identify relationships uniquely, and thus distinguish relationships from each other.
- The primary key of an entity set allows us to distinguish among the various entities of the set.
- For example – If a Student table contains the information about various students as given below –

RollNo	Name	City	Course
101	Ram	Pune	Computer
102	Sita	Pune	Electronics
103	Laxman	Chennai	Mechanical

In above table, **RollNo** is a primary key because, it uniquely identifies the student record.

1.13 Multiple Choice Questions

Q.1 A DBMS provides users with the conceptual representation of _____.

a register b data

c logical view d physical view

Q.2 DBMS helps to achieve _____.

[a] data Independence [b] centralized Control of Data
[c] neither a nor b [d] both a and b

Q.3 In view of total database content is _____.

[a] conceptual view [b] internal view
[c] external view [d] physical view

Q.4 The main purpose of DBMS is to provide _____ view of data to user.

[a] completer [b] abstract
[c] partial [d] none of these

Q.5 ___ means to hide certain details of how data is stored.

[a] Data Integrity [b] Data independence
[c] Data abstraction [d] Data separation

Q.6 How many levels of data abstraction are there ?

[a] One [b] Two
[c] Three [d] Four

Q.7 A _____view of data expresses the way a user thinks about data _____.

[a] logical view [b] physical view
[c] both [d] none

Q.8 A physical view of data refers to the way data is handled at a_____ its storage and retrieval.

[a] high level [b] low level
[c] medium level [d] all of these

Q.9 Architecture of the database can be viewed as _____.

[a] two levels [b] three levels
[c] four levels [d] one level

Q.10 In the architecture of a database system external level is the _____.

[a] physical level [b] logical level
[c] conceptual level [d] view level

Q.11 In hierarchical model records are organized as _____.

[a] lists [b] links
[c] tree [d] graph

Q.12 There are ___ levels of data independence

[a] one [b] two
[c] three [d] four

Q.13 The ability to modify the schema of database in one level without affecting the schema definition in higher level is called as _____.

[a] data isolation [b] data abstraction
[c] data hiding [d] data independence

Q.14 Which of the following is record based on logical model ?

[a] Network Model [b] Object Oriented Model
[c] E-R Model [d] None of these

Q.15 The DDL is used to specify the_____.

[a] conceptual schemas [b] internal schemas
[c] both [d] none

Q.16 DCL stands for _____.

[a] Data Control Language [b] Data Console Language
[c] Data Console Level [d] Data Control Level

Q.17 Which of the following is / are the DDL statements ?

[a] Create [b] Drop
[c] Alter [d] All of the above

Q.18 Which are the three levels of abstraction ?

[a] Physical [b] Logical
[c] External [d] All of these

Q.19 The statement in SQL which allows to change the definition of a table is _____.

[a] create [b] alter
[c] select [d] update

Q.20 Which of the following is NOT a basic element of all versions of the E - R model ?

[a] Entities [b] Relationships
[c] Attributes [d] Primary key

Q.21 Data independence means _____.

a data is defined separately and not included in programs

b programs are not dependent on the physical attributes of data.

c programs are not dependent on the logical attributes of data

d both (b) and (c)

Q.22 E-R model uses this symbol to represent weak entity set _____.

a dotted rectangle

b diamond

c doubly outlined rectangle

d none of these

Q.23 _____ express the number of entities to which another entity can be associated via a relationship set.

a Mapping cardinality

b Relational cardinality

c Participation constraints

d None of the mentioned

Q.24 In E-R diagram derived attribute is represented by _____.

a rectangle

b circle

c dashed Ellipse

d diamond

Q.25 DBA stands for _____.

a Data Building Administrator

b Database Access

c Database Authentication

d Database Administrator

Q.26 _____ represents the number of entities to which another entity can be associated

a Degree

b Cardinality

c Modality

d None of these

Q.27 Data Model is collection of conceptual tools for describing _____.

a Data

b Schema

c constraints

d All of the above

Q.28 Which of the following is example of Object based logical model ?

a Relational Model

b Hierarchical Model

c Network Model

d Entity Relationship Model

Q.29 Entity Relationship model consists of collection of basic objects called _____ and relationship among these objects.

a functions

b models

c entity

d all of these

Answer Keys for Multiple Choice Questions :

Q.1	b	Q.2	d	Q.3	a	Q.4	b
Q.5	c	Q.6	c	Q.7	a	Q.8	b
Q.9	b	Q.10	d	Q.11	c	Q.12	b
Q.13	d	Q.14	a	Q.15	a	Q.16	a
Q.17	d	Q.18	d	Q.19	b	Q.20	d
Q.21	d	Q.22	c	Q.23	a	Q.24	c
Q.25	d	Q.26	b	Q.27	d	Q.28	d
Q.29	c						

❑❑❑

UNIT - II

2 Relational Model

Syllabus

ER and EER diagrams : *Components of ER model, Conventions, Converting ER diagrams into tables.*

Relational Model : *Basic concepts, Attributes and Domains, Codd's rules.*

Relational Integrity : *Nulls, Entity, Referential integrities, Enterprise constraints, Views, Schema diagram.*

Contents

Part I : ER and EER Diagrams

2.1 Components of ER Model

SPPU : Aug 17, Marks 2

- Entity Relational model is a model for identifying entities tobe represented in the database and representation of how those entities are related.
- The ER data model specifies enterprise schema that represents the overall **logical structure** of a database graphically.
- E-R diagrams are used to model real-world objects like a person, a car, a company and the relation between these real-world objects.

Features of ER model

i) E-R diagrams are used to represent E-R model in a database, which makes them easy to be converted into relations (tables).

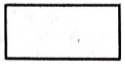
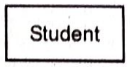
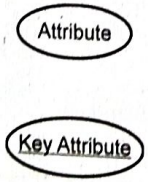
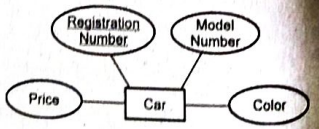
ii) E-R diagrams provide the purpose of real-world modeling of objects which makes them intently useful.

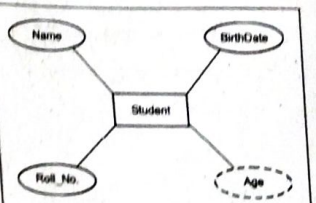
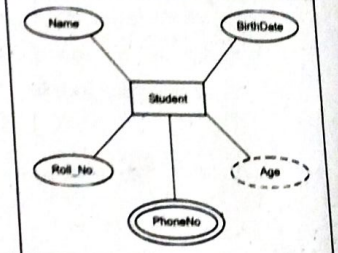
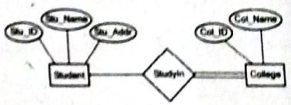
iii) E-R diagrams require **no technical knowledge** and no hardware support.

iv) These diagrams are very **easy to understand** and easy to create even by a naive user.

v) It gives a standard solution of **visualizing** the **data logically.**

Various Components used in ER Model are -

Component	Symbol	Example
Entity : Any real-world object can be represented as an entity about which data can be stored in a database. All the real world objects like a book, an organization, a product, a car, a person are the examples of an entity.		
Relationship : Rhombus is used to setup relationships between two or more entities.		
Attribute : Each entity has a set of properties. These properties of each entity are termed as attributes. For example, a car entity would be described by attributes such as price, registration number, model number, color etc		

Derived attribute : Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth. To represent a derived attribute, another dotted ellipse is created inside the main ellipse		
Multivalued attribute : An attribute that can hold multiple values is known as multivalued attribute. We represent it with double ellipses in an E-R Diagram. E.g. A person can have more than one phone numbers so the phone number attribute is multivalued.		
Total participation : Each entity is involved in the relationship. Total participation is represented by double lines.		

Review Question

1. *List different components used in ER diagram with their meaning.* SPPU : Aug 17, Marks 2

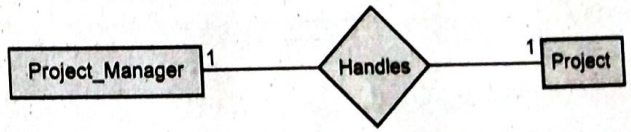
2.2 Conventions

SPPU : Oct.-19, Marks 2

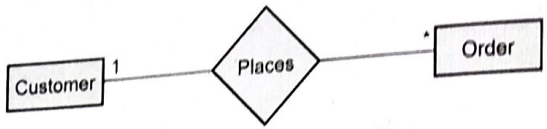
2.2.1 Mapping Cardinality Representation

There are four types of relationships that are considered for key constraints.

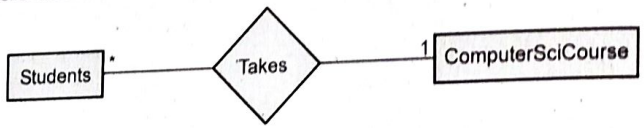
i) **One to one relation :** When entity A is associated with at the most one entity B then it shares one to one relation. For example - There is one project manager who manages only one project.



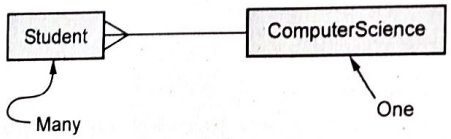
ii) One to many : When entity A is associated with more than one entities at a time then there is one to many relation. For example - One customer places order at time.



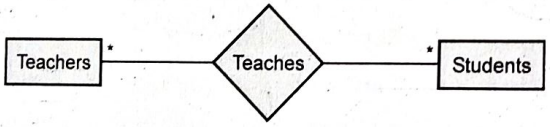
iii)Many to one : When more than one entities are associated with only one entity then there is is many to one relation. For example - Many student take ComputerSciCourse.



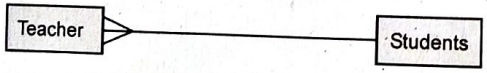
Alternate representation can be



iv) Many to many : When more than one entities are associated with more than one entities. For example -Many teachers can teach many students.

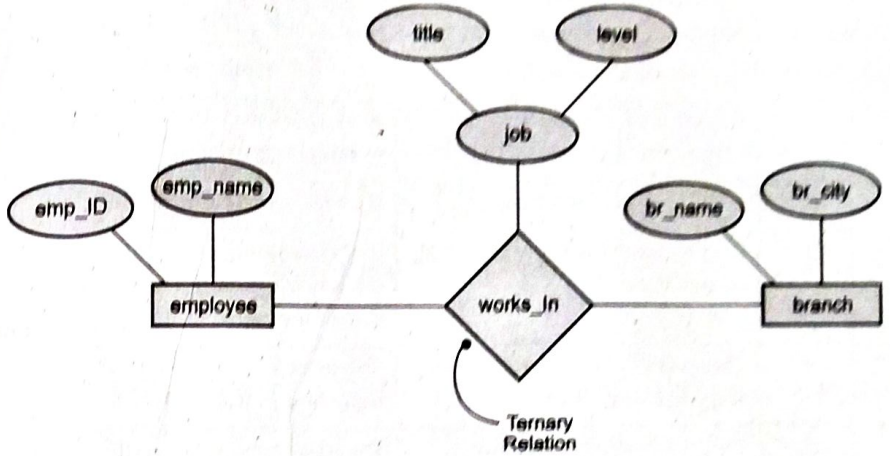


Alternate representation can be



2.2.2 Ternary Relationship

The relationship in which three entities are involved is called ternary relationship. For example -



2.2.3 Weak Entity Set

- A weak entity is an entity that cannot be uniquely identified by its attributes alone. The entity set which does not have sufficient attributes to form a primary key is called as weak entity set.

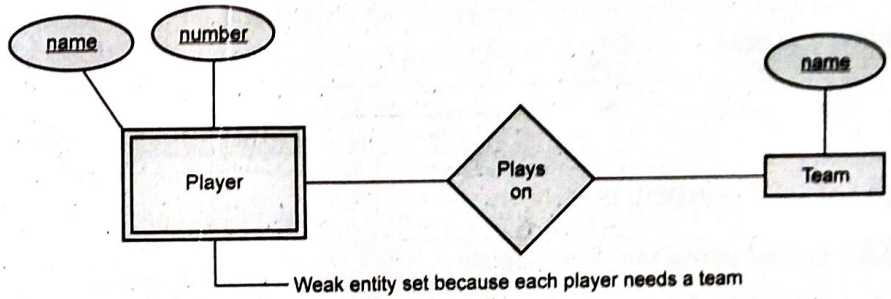


Fig. 2.2.1 : Weak entity set

- **Strong Entity Set**

  The entity set that has primary key is called as strong entity set

Weak entity rules

- A weak entity set has **one or more many-one** relationships to other (supporting) entity sets.
- The key for a weak entity set is its own underlined attributes and the keys for the supporting entity sets. For example - player-number and team-name is a key for Players.

Difference between Strong and Weak Entity Set

Sr. No.	Strong entity set	Weak entity set
1	It has its own primary key.	It does not have sufficient attribute to form a primary key on its own.
2.	It is represented by rectangle	It is represented by double rectangle.
3.	It represents the primary key which is underlined.	It represents the partial key or discriminator which is represented by dashed underline.
4.	The member of strong entity set is called as dominant entity set	The member of weak entity set is called subordinate entity set.
5.	The relationship between two strong entity sets is represented by diamond symbol.	The relationship between strong entity set and weak entity set is represented by double diamond symbol.
6.	The primary key is one of the 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.

Review Question

1. *Differentiate between – Strong entity set and Weak entity set*

SPPU : Oct.-19, (In Sem), Marks 2

2.3 Enhanced ER(EER) Model

2.3.1 Specialization and Generalization

- Some entities have relationships that form hierarchies. For instance, Employee can be an hourly employee or contracted employee.
- In this relationship hierarchies, some entities can act as superclass and some other entities can act as subclass.
- **Superclass :** An entity type that represents a general concept at a high level, is called superclass.
- **Subclass :** An entity type that represents a specific concept at lower levels, is called subclass.
- The subclass is said to inherit from superclass. When a subclass inherits from one or more superclasses, it inherits all their attributes. In addition to the inherited attributes, a subclass can also define its own specific attributes.

- The process of making subclasses from a general concept is called **specialization.** This is **top-down** process. In this process, the sub-groups are identified within an entity set which have attributes that are not shared by all entities.
- The process of making superclass from subclasses is called **generalization.** This is a **bottom up** process. In this process multiple sets are synthesized into high level entities.
- The symbol used for specialization/ Generalization is

- **For example –** There can be two subclass entities namely **Hourly_Emps** and **Contract_Emps** which are subclasses of **Empoyee** class. We might have attributes **hours_worked** and **hourly_wage** defined for **Hourly_Emps** and an attribute **contractid** defined for **ContractEmps.**

  Therefore, the attributes defined for an **Hourly_Emps** entity are the attributes for **Employees** plus **Hourly_Emps.** We say that the attributes for the entity set **Employees** are inherited by the entity set **Hourly_Emps** and that **Hourly-Emps ISA** (read is a) Employees. It can be represented by following Fig. 2.3.1.

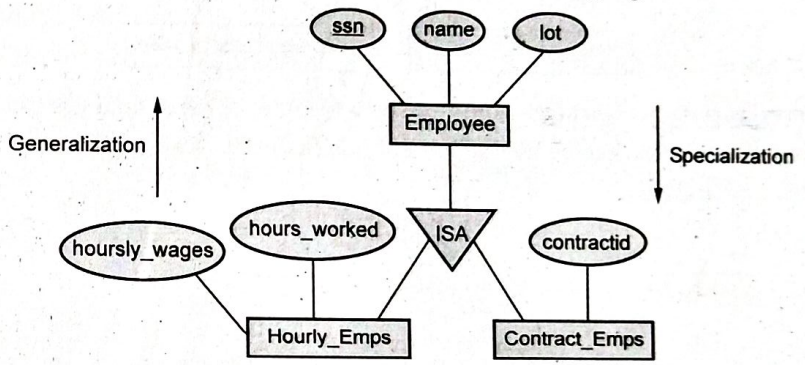


Fig. 2.3.1 Example of Generalization and Specialization

2.3.2 Constraints on Specialization/Generalization

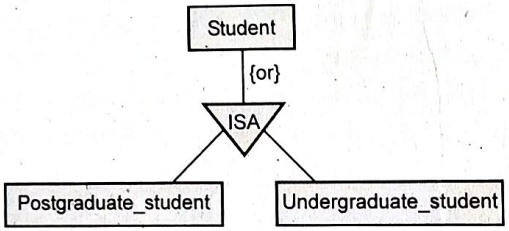
There are four types of constraints on specialization/generalization relationship. These are -

1) **Membership constraints :** This is a kind of constraints that involves determining which entities can be members of a given lower-level entity. There are two types of membership constraints -

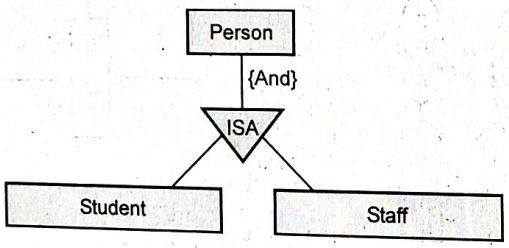
i) **Condition defined :** In condition-defined lower-level entity sets,membership is evaluated on the basis of whether or not an entity satisfies an explicit condition or predicate. For example - Consider the high-level entity Set **Employee** that has attribute **Employee_type**. All Employee entities are evaluated on defining **Employee_type** attribute. All entities that satisfy the condition student type = "ContractEmployee" are included in Contracted Employee. Since all the lower-level entities are evaluated on the basis of the same attribute this type of generalization is said to be **attribute-defined**.

ii) **User defined :** This is kind of entity set that in which the membership is manually defined.

2) **Disjoint constraints :** The disjoint constraint only applies when a superclass has more than one subclass. If the subclasses are disjoint, then an entity occurrence can be a member of only one of the subclasses. For entity Student has either **Postgraduate_Student** entity or **Undergraduate_Student**

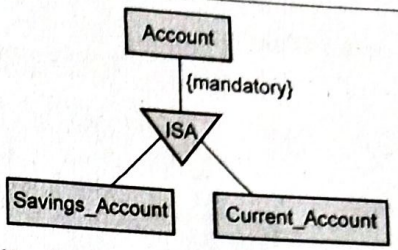


3) **Overlapping :** When some entity can be a member of more than one subclasses. For example - Person can be both a Student or a Staff. The **And** can be used to represent this constraint.

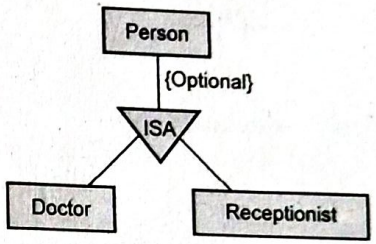


4) **Completeness :** It specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within the generalization/specialization. This constraint may be one of the following -

i) **Total generalization or specialization :** Each higher-level entity must belong to a lower-level entity set. For example - Account in the bank must either Savings account or Current Account. The **mandatory** can be used to represent this constraint.



ii) **Partial generalization or specialization :** Some higher-level entities may not belong to any lower-level entity set.



2.3.3 Aggregation

A feature of the entity relationship model that allows a relationship set to participate in another relationship set. This is indicated on an ER diagram by drawing a dashed box around the aggregation.

For example - We treat the relationship set work and the entity sets employee and project as a higher-level entity set called work.

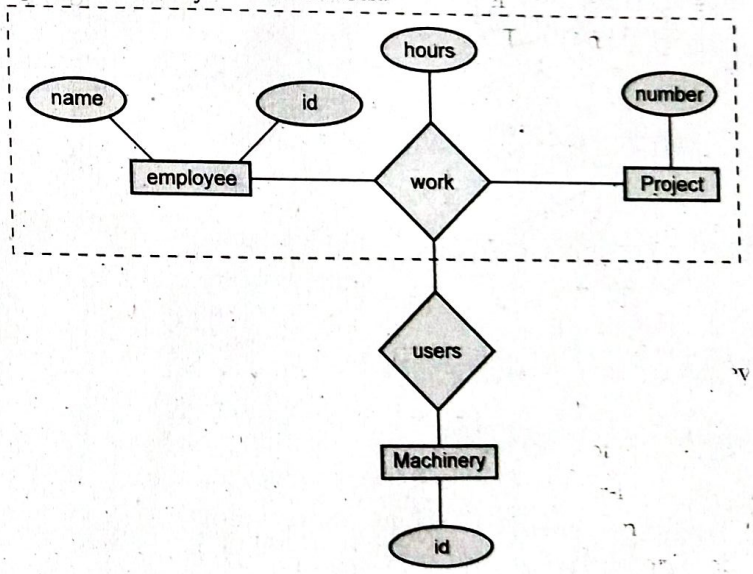


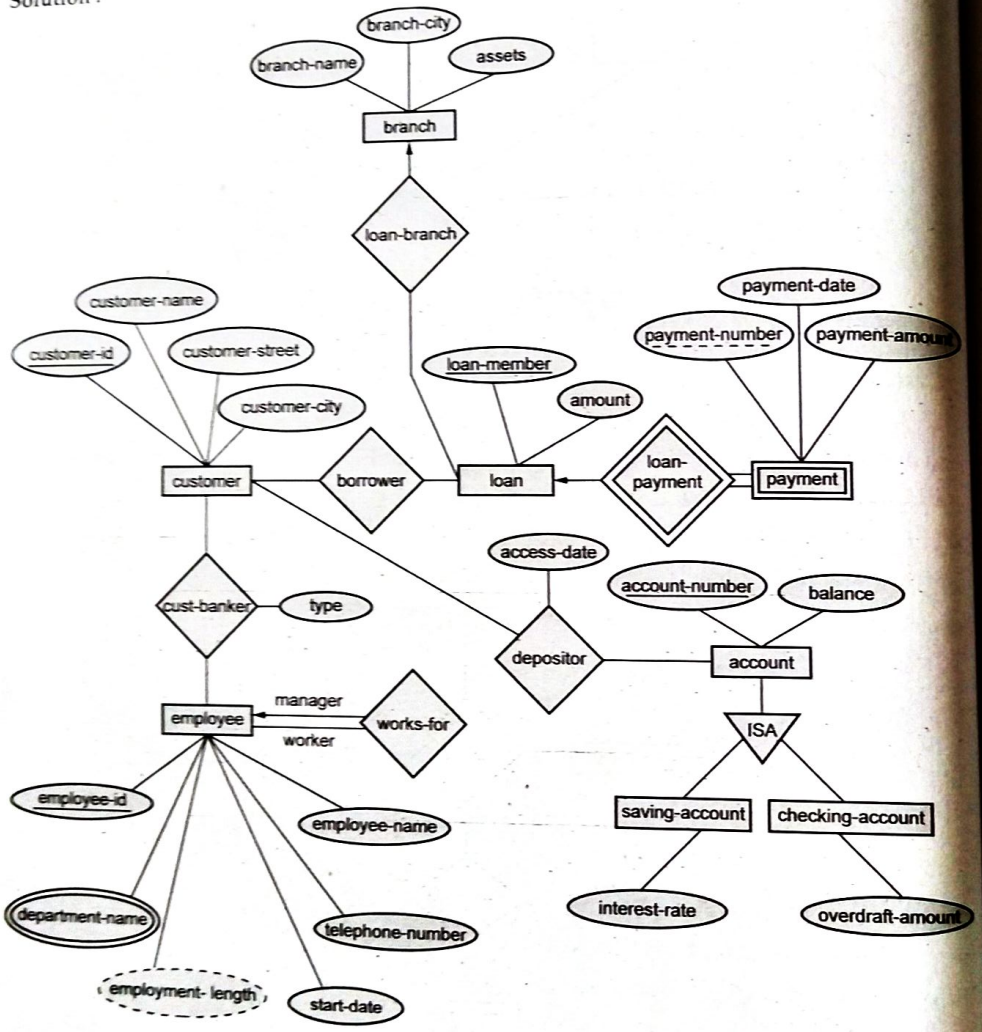
Fig. 2.3.2 : ER model with aggregation

2.4 Examples Based on ER Diagram

SPPU : Aug 17, Nov.-17, Oct.-19, Marks 6

Example 2.4.1 *Draw an ER diagram for banking system(Home-Loan Application)*

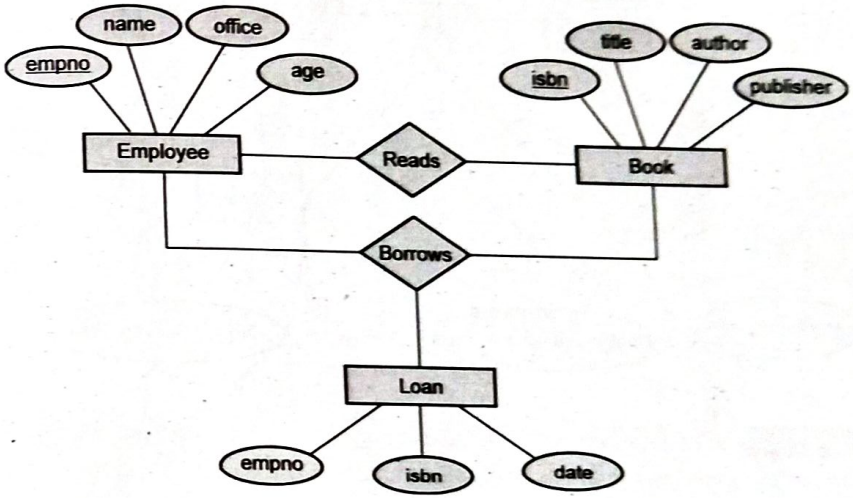
Solution :



Example 2.4.2 *Consider the relation schema given in following Figure. Design and draw an ER diagram that capture the information of this schema.*

Employee(empno,name,office,age)
Books(isbn,title,authors,publisher)
Loan(empno,isbn,date)

Solution :



Example 2.4.3 *A car rental company maintains a database for all vehicles in its current fleet. For all vehicles, it includes the vehicle identification number license number, manufacturer, model, date of purchase and color. Special data are included for certain types of vehicles.*

Trucks : Cargo capacity

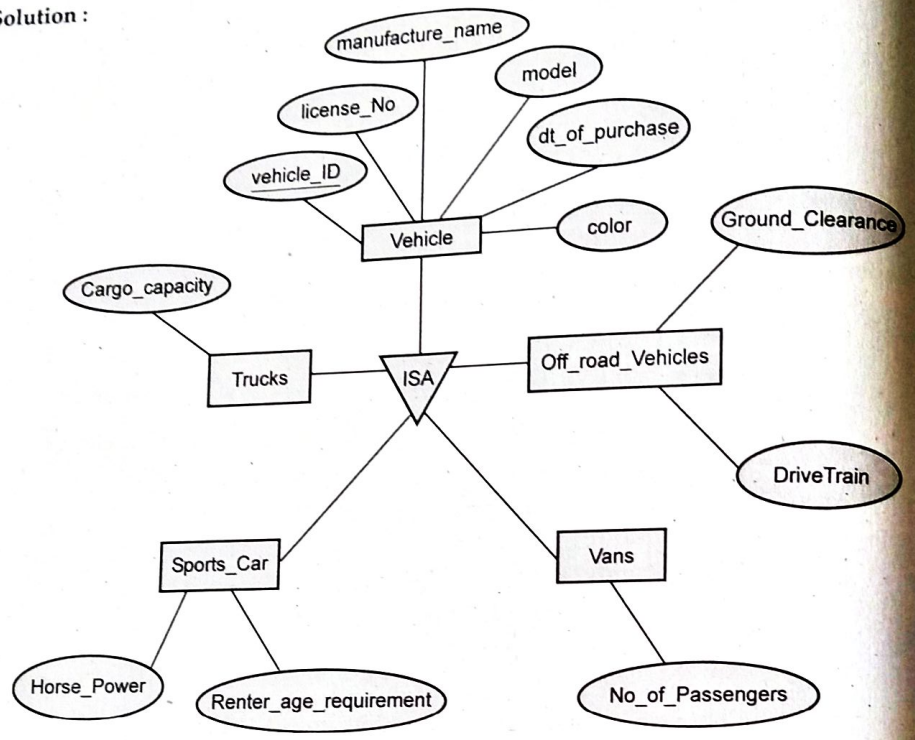
Sports cars : horsepower, renter age requirement

Vans : number of passengers

Off-road vehicles : ground clearance, drivetrain (four-or two-wheel drive)

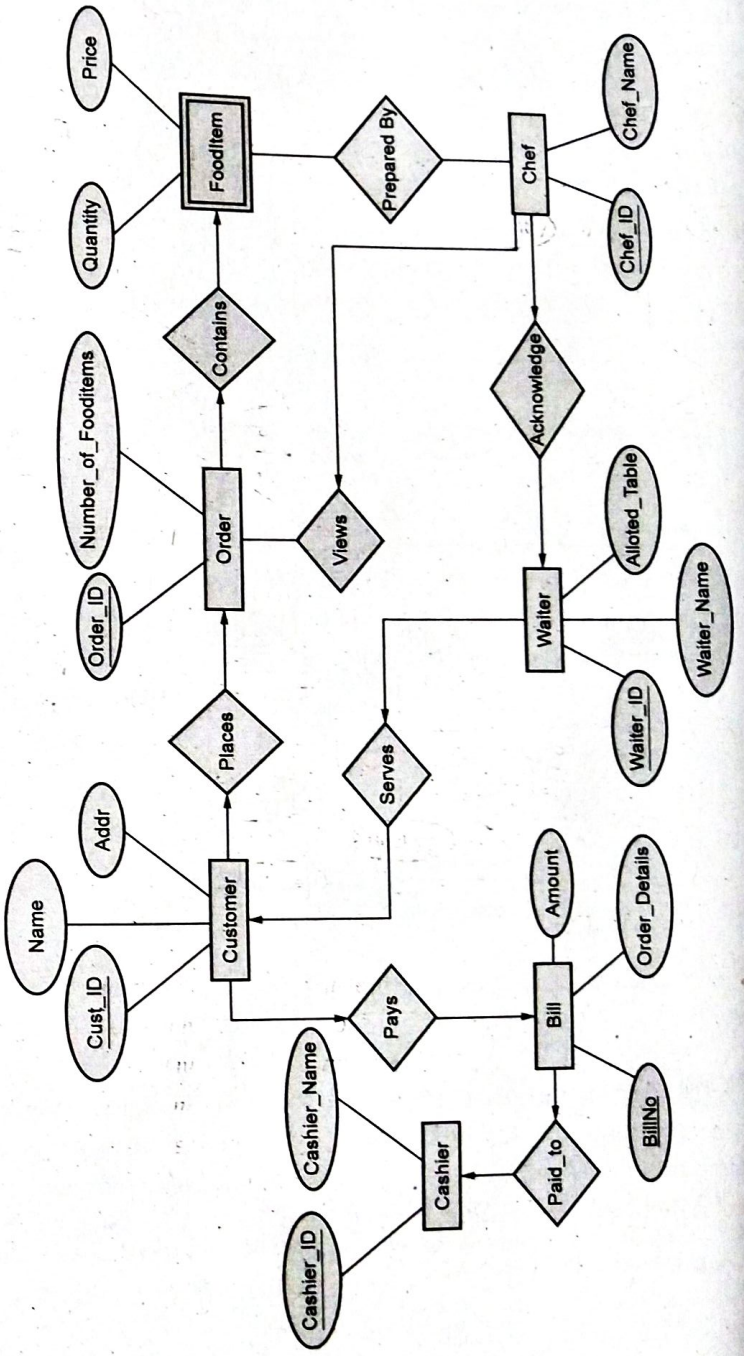
Construct an ER model for the car rental company database.

Solution :



Example 2.4.4 *Draw E-R diagram for the "Restaurant Menu Ordering System", which will facilitate the food items ordering and services within a restaurant. The entire restaurant scenario is detailed as follows. The customer is able to view the food items menu, call the waiter, place orders and obtain the final bill through the computer kept in their table. The Waiters through their wireless tablet PC are able to initialize a table for customers, control the table functions to assist customers, orders, send orders to food preparation staff (chef) and finalize the customer's bill. The Food preparation staffs (chefs), with their touch-display interfaces to the system, are able to view orders sent to the kitchen by waiters. During preparation they are able to let the waiter know the status of each item, and can send notifications when items are completed. The system should have full accountability and logging facilities, and should support supervisor actions to account for exceptional circumstances, such as a meal being refunded or walked out on.*

Solution :



Example 2.4.5 *A university registrar's office maintains data about the following entities :*

(1) courses, including number, title, credits, syllabus, and prerequisites;

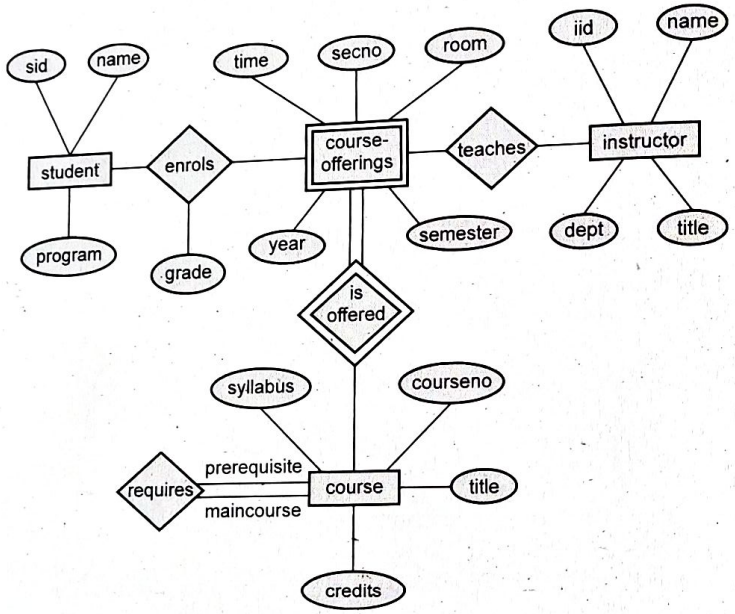
(2) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;

(3) students, including student-id, name, and program; and

(4) instructors, including identification number, name, department, and title.

Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

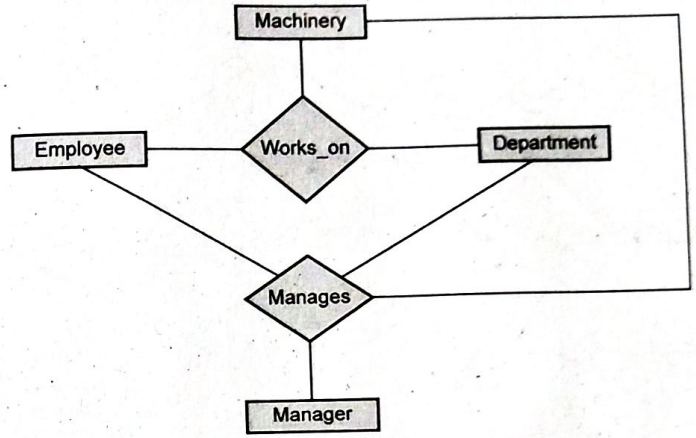
Solution :



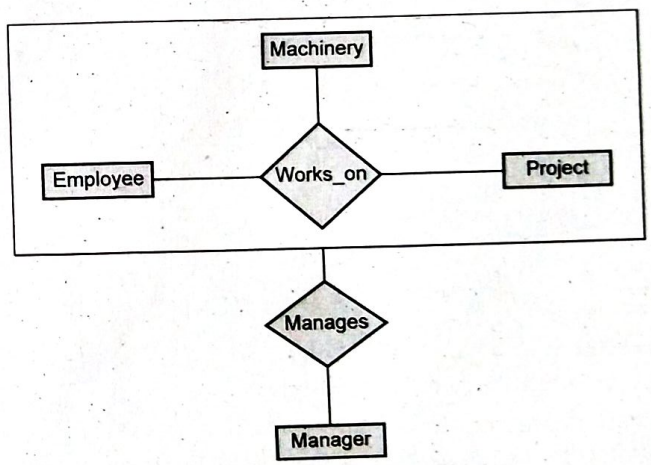
Example 2.4.6 *What is aggregation in ER model ? Develop an ER diagram using aggregation that captures following information : Employees work for projects. An employee working for particular project uses various machinery. Assume necessary attributes. State any assumptions you make. Also discuss about the ER diagram you have designed.*

Solution : Aggregation : Refer section 2.3.3.

ER Diagram : The ER diagram for above described scenario can be drawn as follows -



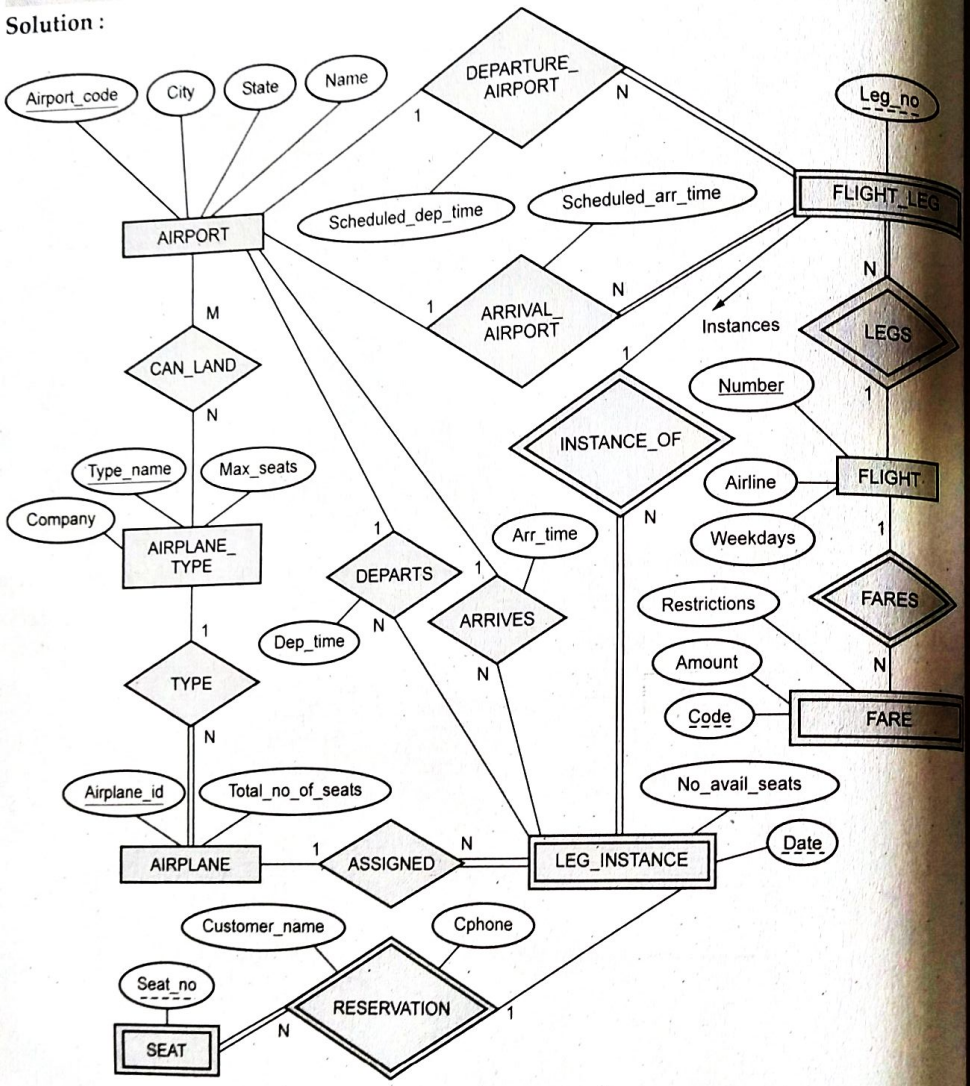
The above ER model contains the redundant information, because every Employee, Project, Machinery combination in **works_on** relationship is also considered in **manages** relationship. To avoid this redundancy problem we can make use of aggregation relationship in ER diagram as follows -



We can then create a binary relationship **manages** for between **Manager** and **(Employee, Project, Machinery).**

Example 2.4.7 *Draw an ER diagram of Airline reservation system taking into account at least five entities. Indicate all keys, constraints and assumptions that are made*

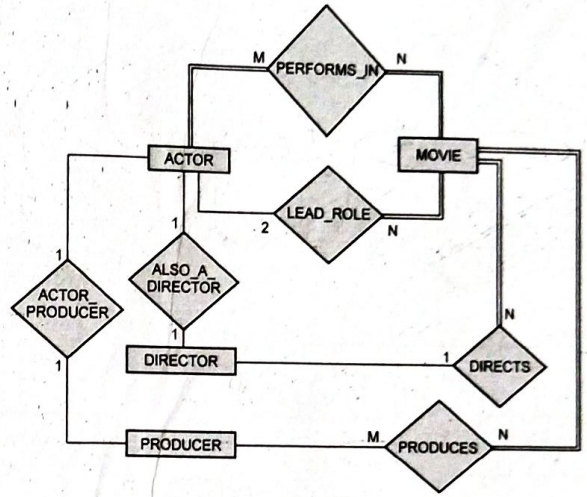
Solution :



A LEG is nonstop portion of flight. The LEG_INSTANCE is a particular occurrence of a LEG on particular date.

Example 2.4.8 *Draw an ER diagram of movie database. Assume your own entities (minimum 4) attributes and relationships.*

Solution :



Example 2.4.9 *Draw an ER diagram to represent the Election Information system based on the following description.*

In Indian national election, a state is divided into a number of constituencies depending upon the population of the state. Several candidates contest elections in each constituency. Record the number of votes obtained by each candidate. The system also maintains the voter list and a voter normally belongs to a particular constituency.

Note that the party details must also be taken care in the design.

Solution :

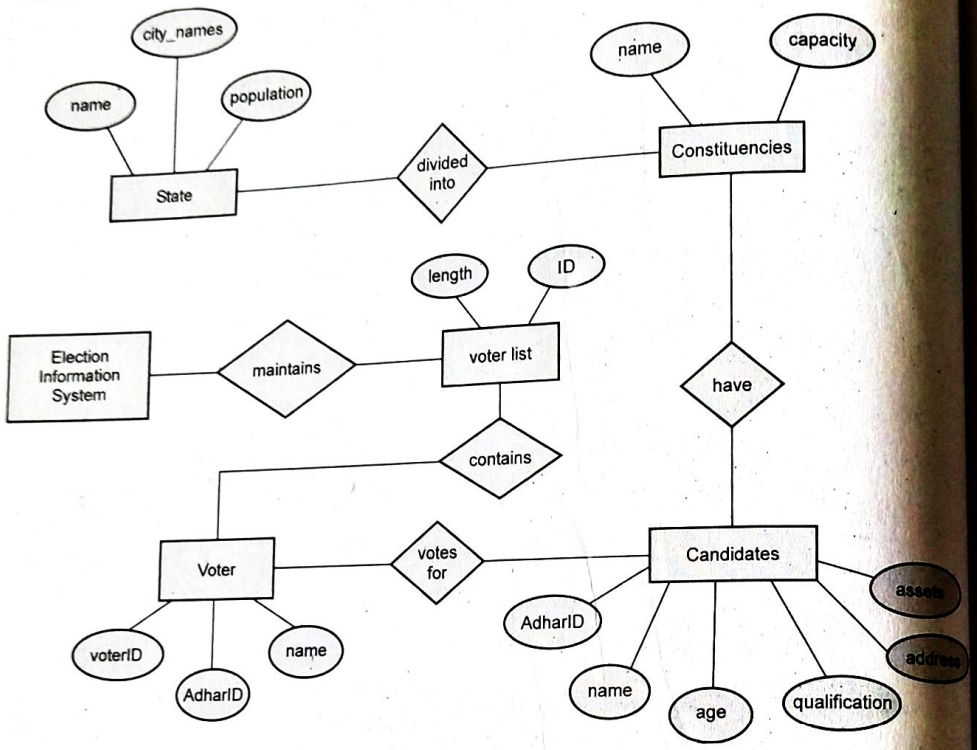
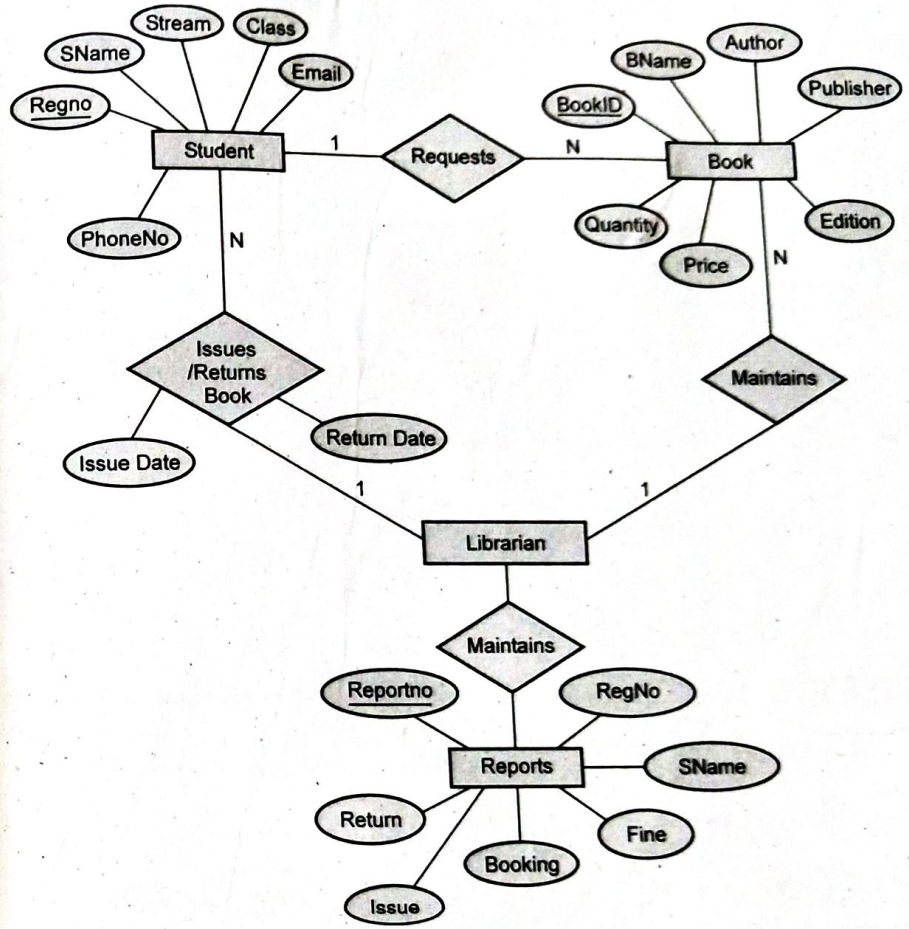


Fig. 2. 4.1 Election information system

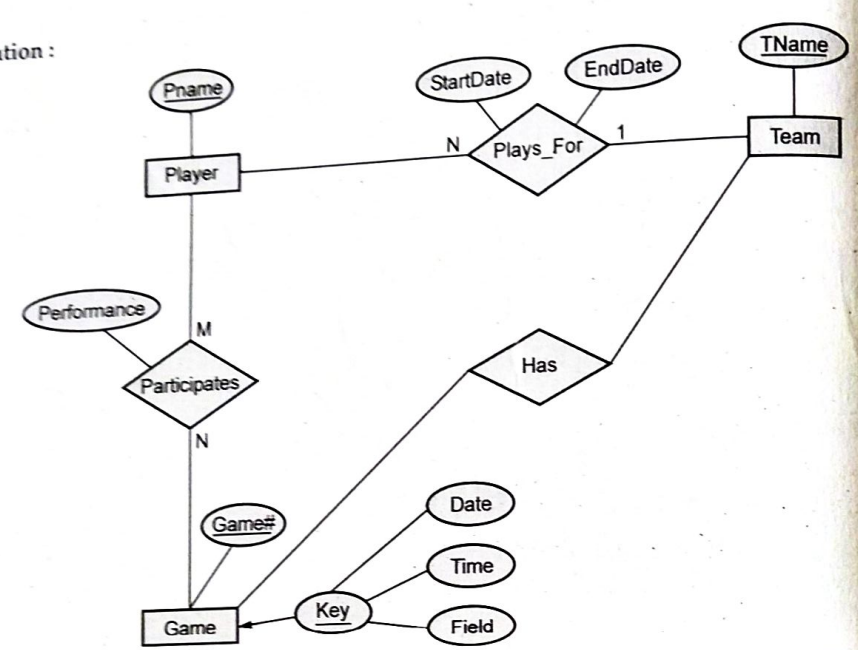
Example 2.4.10 *Construct an E R diagram for library management system*

Solution :



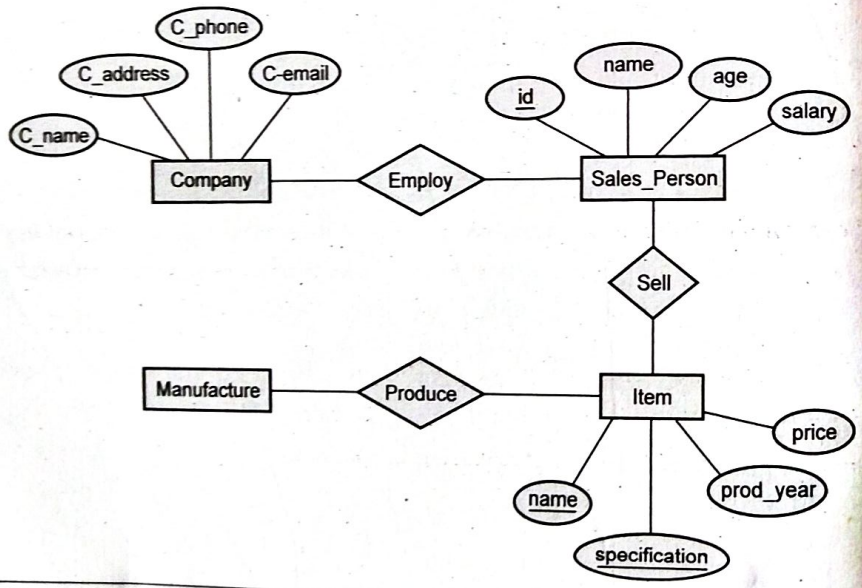
Example 2.4.11 *A database is to be constructed to keep track of the teams and games of a sport league. A team has number of players not all of whom participate in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of the game. Design an ER diagram completely with attributes, keys and constraints for the above description.*

Solution :



Example 2.4.12 *Draw an ER diagram for a small marketing company database. Assume suitable data.*

Solution :



Example 2.4.13 *A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram -*

1. Professors can teach the same course in several semesters, and each ordering must be recorded.

2. Professors can teach the same course in several semesters, and only the most recent such ordering needs to be recorded (Assume this condition applies in all subsequent questions).

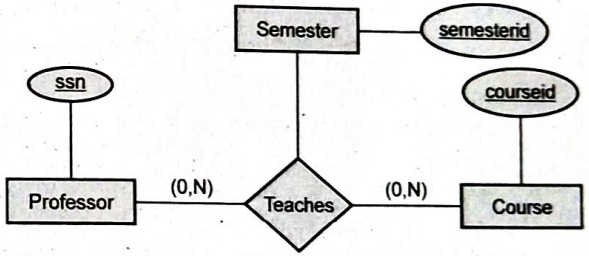
3. Every professor must teach some course.

4. Every professor teaches exactly one course.

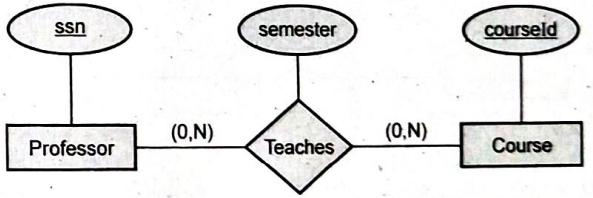
5. Every professor teaches exactly one course and every course must be taught by some professor.

Solution :

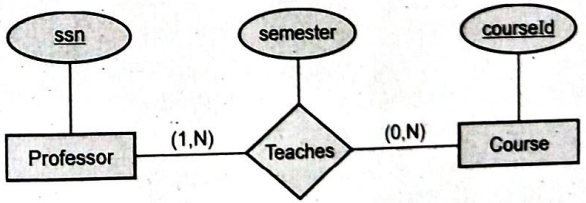
1.



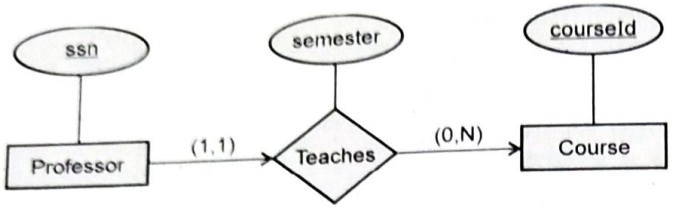
2.



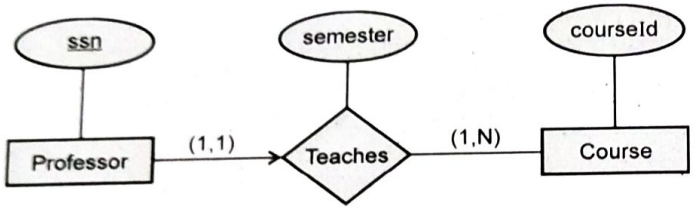
3.



4.

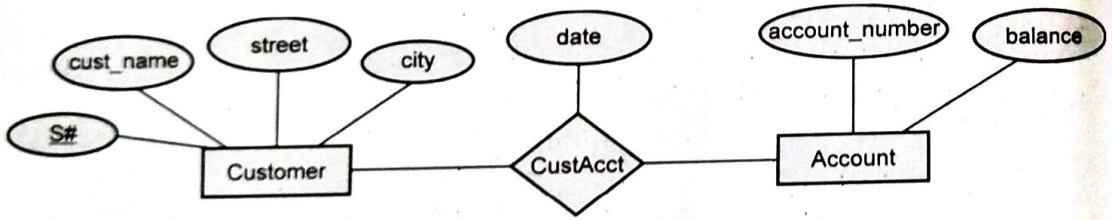


5.



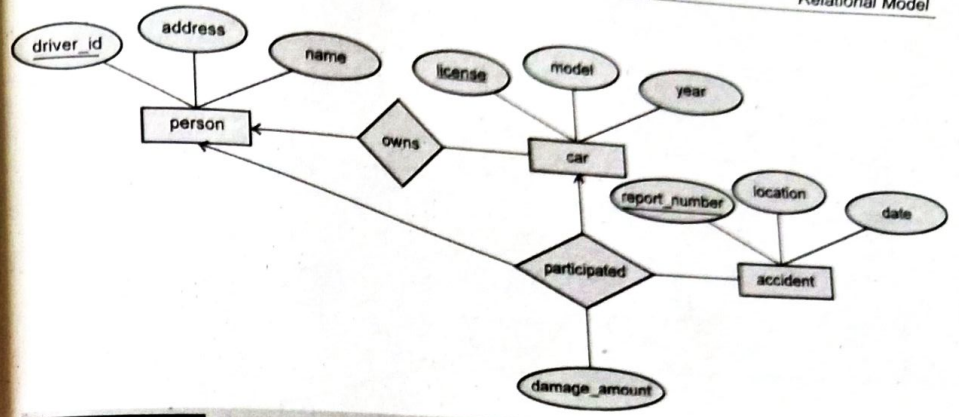
Example 2.4.13 *Construct an ER diagram of customer account relationship. Customer entity with attributes of S#, Customer name, street, customer city, and account entity with attributes account no, and balance. The customer account relationship with date attributes.*

Solution :



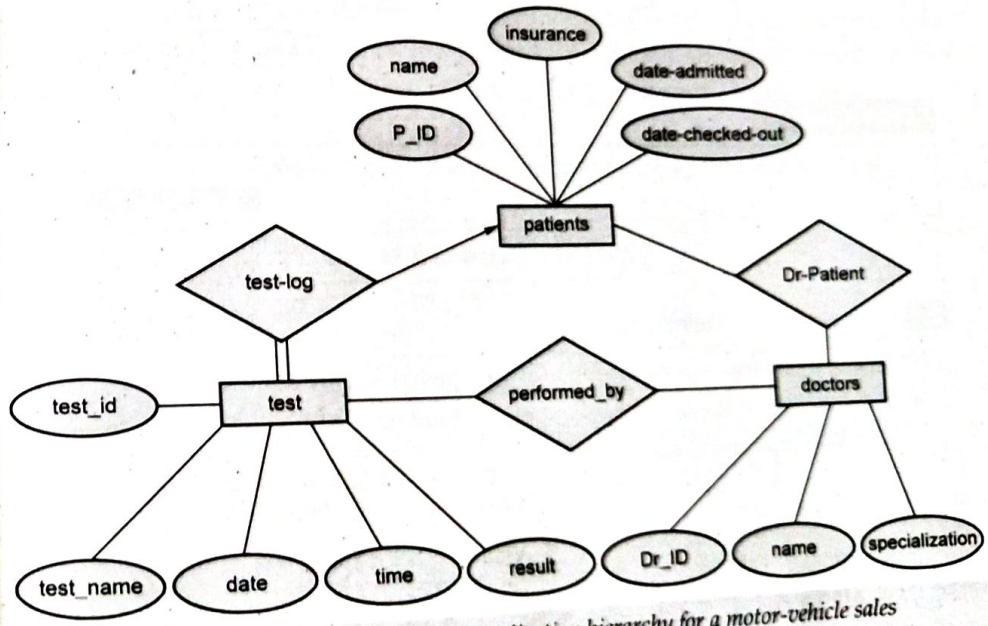
Example 2.4.14 *Construct an E-R diagram for a car insurance company whose customers own one or more cars each.Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars and has one or more premium payments associated with it. Each payment is for particular period of time and has an associated due date and date when the payment was received.* **SPPU : Aug 17, Marks 4**

Solution :



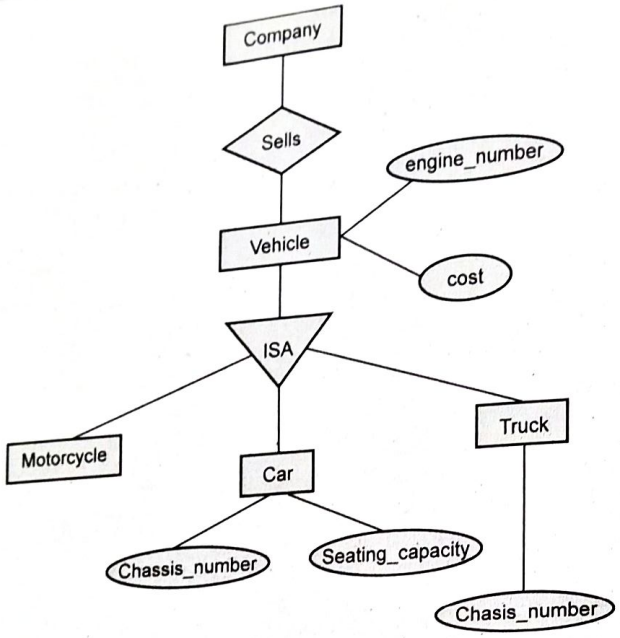
Example 2.4.15 *List E-R diagram symbol and draw an E-R diagram for a hospital management system with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examination conducted.* **SPPU : Nov -17, (End Sem), Marks 6**

Solution :



Example 2.4.16 *Design generalization-specialization hierarchy for a motor-vehicle sales company. The company sell motorcycles which have an engine number and cost. Cars which have chassis number, an engine number, seating capacity and cost; trucks which have chassis number an engine number and cost.* **SPPU : Oct.-19, (In Sem), Marks 6**

Solution :



Review Question

1. *Explain specialization, generalization and aggregation with example.* **SPPU : Aug 17, Marks 5**

Part II : Relational Model

2.5 Introduction to Relational Model

- Relation database is a **collection of tables** having unique names.
- For example – Consider the example of **Student table** in which the information about the student is stored.

RollNo	Name	Phone
001	AAA	1111111111
002	BBB	2222222222
003	CCC	3333333333

Fig. 2.5.1 Student table

The above table consists of three column headers **RollNo, Name** and **Phone**. Each row of the table indicates the information of each student by means of his Roll Number, Name and Phone number.

Similarly consider another table named **Course** as follows –

CourseID	CourseName	Credits
101	Mechanical	4
102	Computer Science	6
103	Electrical	5
104	Civil	3

Fig. 2.5.2 Course table

Clearly, in above table the columns are **CourseID, CourseName** and **Credits.** The **CourseID 101** is associated with the course named **Mechanical** and associated with the course of mechanical there are **4 credit points.** Thus the relation is represented by the table in the relation model. Similarly we can establish the relationship among the two tables by defining the third table. For example – Consider the table **Admission** as

RollNo	CourseID
001	102
002	104
003	101

Fig. 2.5.3 Admission Table

From this third table we can easily find out that the course to which the **RollNo 001** is admitted is **computer Science**.

2.6 Basic Terminologies

SPPU : Aug.-17, Marks 5

There are some commonly used terms in Relational Model and those are -

Table or relation : In relational model, table is a collection of data items arranged in rows and columns. The table cannot have duplicate data or rows. Below is an example of student table

Roll No	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333
004	DDD	67	4444444444

Tuple or record or row : The single entry in the table is called tuple. The tuple represents a set of related data. In above **Student** table there are four tuples. One of the tuple can be represented as

001	AAA	88	1111111111

Attribute or columns : It is a part of table that contains several records. Each record can be broken down into several small parts of data known as attributes. For example the above table consists of four attributes such as **RollNo,Name,Marks** and **Phone**.

Relation schema : A relation schema describes the structure of the relation, with the name of the relation (i.e. name of table), its attributes and their names and type.

Relation Instance : It refers to specific instance of relation i.e. containing a specific set of rows. For example – the following is a relation instance – which contains the records with marks above 80.

RollNo	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333

Domain : For each attribute of relation, there is a set of permitted values called domain. For example – in above table, the domain of attribute **Marks** is set of all possible permitted marks of the students. Similarly the domain of **Name** attribute is all possible names of students.

That means Domain of Marks attribute is (88,83,98)

Atomic : The domain is atomic if elements of the domain are considered to be indivisible units. For example in above **Student** table, the attribute **Phone** is non-atomic.

NULL attribute : A null is a special symbol, independent of data type, which means either unknown or inapplicable. It does not mean zero or blank. For example - Consider a salary table that contains NULL

Emp#	Job Name	Salary	Commission
E10	Sales	12500	32090
E11	Null	25000	8000
E12	Sales	44000	0
E13	Sales	44000	Null

Degree : It is nothing but total number of columns present in the relational database. In given Student table –

Roll No	Name	Marks	Phone
001	AAA	88	1111111111
002	BBB	83	2222222222
003	CCC	98	3333333333

The degree is 4.

Cardinality : It is total number of tuples present in the relational database. In above given table the cardinality is 3

Example 2.6.1 *Find out following for given Staff table*

i) No of Columns *ii) No of tuples*

iii) Different attributes *iv) Degree* *v) Cardinality*

StaffID	Name	Sex	Designation	Salary	DOJ
S001	John	M	Manager	50000	1 Oct. 2012
S002	Ram	M	Executive	20000	20 Jan. 2015
S003	Meena	F	Supervisor	40000	12 Aug. 2011

Solution :

i) No of Columns = 6

ii) No of Tuples= 3

iii) Different attributes are StaffID, Name,Sex, Designation, Salary, DOJ

iv) Degree= Total number of columns=6

v) Cardinality =Total number of rows = 3

Review Question

1. *What is schema definition? Explain the types of attributes with an example.* **SPPU : Aug.-17, Marks 5**

SPPU : May-19, Nov.-18, Marks 6

2.7 Codd's rules

Codd proposed 13 rules for relational database management system, which are popularly known as **Codd's 12 rule** : These rules are as follows –

Rule 0 : This rule states for a database to be relational, it must use its relational capabilities to manage the database.

Rule 1 : The Information rule - All information in an RDBMS is represented logically only by storing the values in tables.

Rule 2 : The Guaranteed Access rule - Each item of data in an RDBMS is guaranteed to be logically accessible by specifying the table name, primary key value, and column name.

Rule 3 : The Systematic Treatment of Null Values rule - Null values are supported in a fully relational DBMS for representing missing information or inapplicable information in a systematic way which is independent of the data type.

Rule 4 : The Dynamic Online Catalog Based on the Relational Model rule - Database dictionary which is called as catalog-is the structure description of the complete Database and it must be stored online. This Catalog must be governed by same rules as rest of the database. The same query language should be used on catalog as used to query database.

Rule 5 : The Comprehensive Data Sublanguage rule - At least one well structured, well-defined language must be there which can access all the data present in the database.

Rule 6 : The View Updating rule - All views of the data which are theoretically updatable must be updatable in practice by the DBMS.

Rule 7 : Relational level operation - The High-level Insert, Update, and Delete rule: There must be insert, delete and update operations at each level of relations.

Rule 8 : The Physical Data Independence rule - Physical storage should not matter the system. Whenever any changes are made in either storage representations or access methods then it should not affect the application.

Rule 9 : The Logical Data Independence rule - If any changes are made in table structure then the logical view of the user should not get affected. Fpr Rule example - if a table is split into two tables internally, the view of the table to the user should be an entire table and not the split tables.

Rule 10 : The Integrity Independence rule - The Integrity constraints must be defined by the RDBMS stored in the system and it should not be enforced by the external application programs.

Rule 11 : The Distribution Independence rule - An RDBMS must have distribution independence. That means, even if database is scattered geographically, user should get a feel as if it is stored in one piece at one location.

Rule 12 : The Non-sub-version rule - If low-level language is allowed to access the system, then that low-level language must not be able to subvert or bypass the integrity rules which are expressed in a higher-level language.

Review Question

1. *Specify Codd's Norms to be satisfied by RDBMS?*

SPPU : May-19, (End Sem), Marks 5; Nov.-18, (End Sem), Marks 6

Part III : Relational Integrity

2.8 Keys

SPPU : Aug.-17, Marks 4

Keys are used to specify the tuples distinctly in the given relation.

Various types of keys used in relational model are – Superkey, Candidate Keys, primary keys, foreign keys. Let us discuss them with suitable example

1) Super Key(SK):

It is a set of one or more attributes within a table that can uniquely identify each record within a table. For example – Consider the Student table as follows –

Reg No.	Roll No	Phone	Name	Marks
R101	001	1111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Fig. 2.8.1 Student

The superkey can be represented as follows

(RollNo, Phone, Name) Superkey

(Name, Marks) Not a Superkey

Superkey

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Clearly using the (RegNo) and (RollNo,Phone,Name) we can identify the records uniquely but (Name, Marks) of two students can be same, hence this combination not necessarily help in identifying the record uniquely.

2) Candidate Key(CK) :

The candidate key is a subset of superset. In other words candidate key is a single attribute or least or minimal combination of attributes that uniquely identify each record in the table. For example - in above given **Student** table, the candidate key is RegNo, (RollNo,Phone). The candidate key can be

Candidate key

Candidate key

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Thus every candidate key is a superkey but every superkey is not a candidate key.

3) Primary Key(PK) :

The primary key is a candidate key chosen by the database designer to identify the tuple in the relation uniquely. For example – Consider the following representation of primary key in the student table

Primary key

RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

Other than the above mentioned primary key, various possible primary keys can be **(RollNo), (RollNo,Name), (RollNo, Phone)**

The relation among super key, candidate key and primary can be denoted by

Candidate Key = Super Key – Primary Key

Rules for Primary Key

i) The primary key may have one or more attributes.

ii) There is **only one primary key** in the relation.

iii) The value of primary key attribute can not be NULL.

iv) The value of primary key attribute does not get changed.

4) Alternate key :

The alternate key is a candidate key which is not chosen by the database designer to uniquely identify the tuples. For example –

Primary key

Alternate key

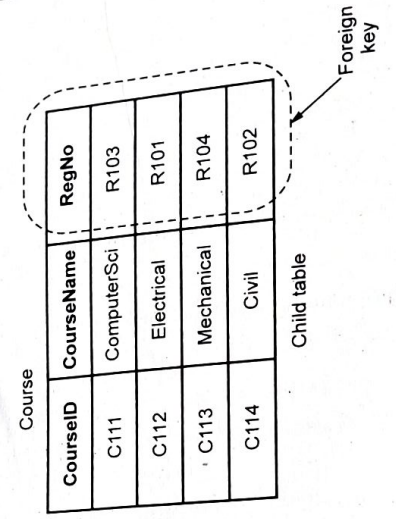
RegNo.	RollNo	Phone	Name	Marks
R101	001	111111111	AAA	88
R102	002	2222222222	BBB	83
R103	003	3333333333	CCC	98
R104	004	4444444444	DDD	67

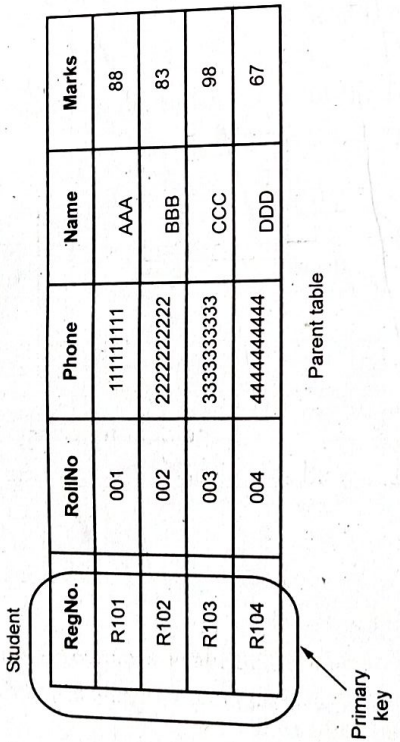
5) Foreign key :

Foreign key is a single attribute or collection of attributes in one table that refers to the primary key of other table.

- Thus foreign keys refer to primary key.
- The table containing the primary key is called **parent table** and the table containing foreign key is called **child table**.

- Example -





From above example, we can see that two tables are linked. For instance we could easily find out that the 'Student CCC has opted for ComputerSci course'

Review Question

1. *Explain the distinction among the terms primary key, candidate key and super key*
   SPPU : Aug.-17. Marks 4

2.9 Constraints

Constraints mean some rules or restrictions that are set on the database.

There are three main types of constraints.

1. Domain Constraint
2. Key Constraint or NULL Constraint
3. Integrity Constraint
   - i) Entity Integrity Constraint
   - ii) Referential Integrity Constraint

1. Domain Constraint

- Domain constraint defines the domain or set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.
- For example - Consider the Student table as follows.

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	1234	98	3333333333
004	DDD	67	4444444444

Name cannot be numeric value.
It must be a string.
Hence this is not allowed !!!

The above relation does not satisfy the domain constraint.

2. Key Constraint or Null Constraint

- Keys are used to identify particular record from the table. Primary key is normally used to identify the record uniquely.
- Hence the key constraint can be stated as -
  - All values of primary key must be unique.
  - The value of primary key must not be NULL.
- For example - Consider the Student table as follows. For this relation, the **Roll No** is a primary key. It is expected to find the desired record using this primary key.

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	CCC	98	3333333333
004	DDD	67	4444444444

Duplicate values ? This is not allowed !!!

The above relation does not satisfy key constraint as the primary key is not having unique value.

3. Integrity Constraint

- Integrity constraints are rules that are to be applied on database columns to ensure the validity of data.
- For example -
  i) The Employee ID and Department ID must consist of two digits.
  ii) Every Employee ID must start with letter.

i) Entity Integrity Constraint

- **This rule states that** "In the relations, the value of attribute of primary key can not be null".
- The NULL represents a value for an attribute that is currently unknown or is not applicable for this tuple. The Nulls are always **to deal with incomplete or exceptional data.**
- The primary key value helps in uniquely identifying every row in the table. Thus if the users of the database want to retrieve any row from the table or perform any action on that table, they must know the value of the key for that row. Hence it is necessary that the primary key should not have the NULL value.

- For example -

Roll No	Name	Marks	Phone
001	AAA	88	111111111
001	BBB	83	2222222222
003	CCC	98	3333333333
	DDD	67	4444444444

NULL is not allowed !!!

ii) Referential Integrity Constraint

- In relationships, data is linked between two or more tables.
- This is achieved by having the foreign key (in the associated table) reference a primary key value (in the primary - or parent - table). Because of this, we need to ensure that data on both sides of the relationship remain intact.
- **The referential integrity rule states that** "whenever **a foreign key** value is used it must **reference a valid, existing primary** key in the parent table".
- For example - Consider two tables

Primary key

RegNo	Roll No	Name	Marks	Phone
R101	001	AAA	88	111111111
R102	001	BBB	83	2222222222
R103	003	CCC	98	3333333333
R104	004	DDD	67	4444444444

Student table as parent table

Relation

Foreign key

CourseID	CourseName	RegNo
c111	ComputerSci	R103
c112	Electrical	R101
c113	Mechanical	R555
c114	Civil	R102

This value is not present in parent table

Course table as child table

In above relation, the registration no. R555 is not existing still if it is present in the course table, then we say that it is not following referential integrity constraint.

2.10 Enterprise Constraints

Enterprise constraints are also called as semantic constraints.

The enterprise constraints are basically the additional rules specified by users or database administrators. These constraints are normally based on multiple tables.

Examples of enterprise constraints are –

1) The salary of teacher should not exceed the salary of Principal.
2) A Student can not opt for more than two courses at a time.
3) A class can have maximum 50 students.

2.11 Views

- The view of the system help the user to retrieve data efficiently.
- A **view** is a virtual or logical table that allows to view or manipulate parts of the tables.
- A view also has rows and columns as they are in a real table in the database.
- We can create a view by selecting fields from one or more tables present in the database.
- A view can either have all the rows of a table or specific rows based on certain condition.
- For example

Consider table **Empolyee** and create a view **EmployeeDetails** whose Salary is <10000

Employee Table		
EmpID	**EName**	**Salary**
101	Archana	20000
102	Madhura	5000
103	Poonam	8000
104	Shruti	15000
105	Monika	7000

→

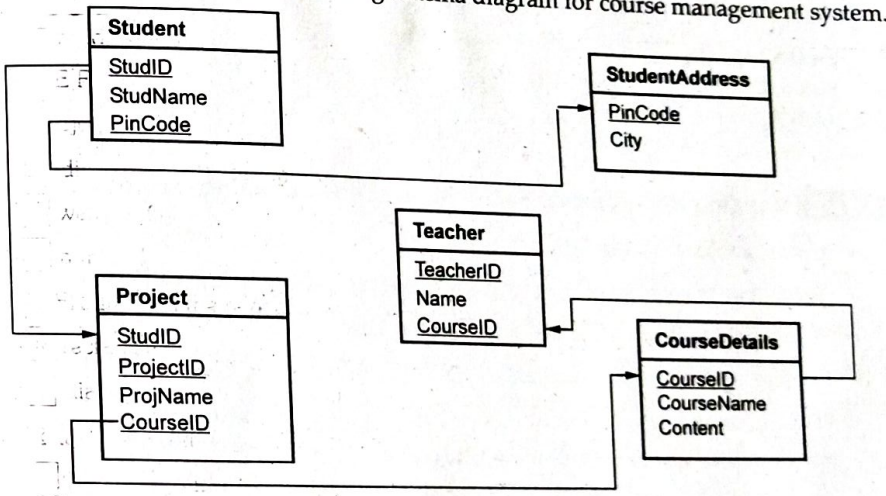
EmployeeDetails View	
EmpID	**Ename**
102	Madhura
103	Poonam
105	Monika

Role of Views in Database:

1) Views are valuable in the **context of security.**
2) We can define view that give a group of uses **access to only allowed part** of the database and **can restrict other part.**

2.12 Schema Diagrams

- Schema diagrams are used to represent the relationship among the various schemas.
- Each relation appears as a box, with the relation name at the top, and the attributes listed inside the box.
- Primary key attributes are shown underlined.
- Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation.
- For example – Consider following schema diagram for course management system.



2.13 Converting ER diagrams into tables

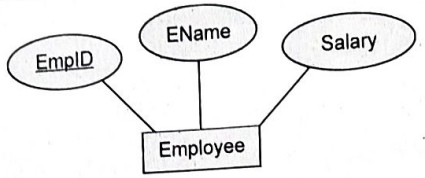
SPPU : Nov.-18, Marks 4

In this section we will discuss how to map various ER model constructs to Relational Model construct.

2.13.1 Mapping of Entity Set to Relationship

- An entity set is mapped to a relation in a straightforward way.
- Each attribute of entity set becomes an attribute of the table.
- The primary key attribute of entity set becomes an entity of the table.

- For example - Consider following ER diagram.



The converted employee table is as follows -

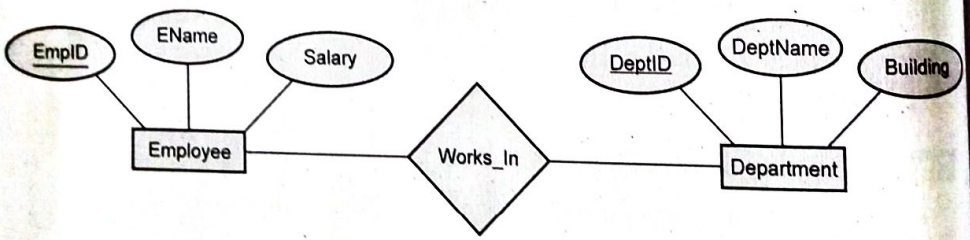
EmpID	EName	Salary
201	Poonam	30000
202	Ashwini	35000
203	Sharda	40000

The SQL statement captures the information for above ER diageam as follows -

```
CREATE TABLE Employee( EmpID CHAR(11),
    EName CHAR(30),
    Salary INTEGER,
PRIMARY KEY(EmpID))
```

2.13.2 Mapping Relationship Sets(without Constraints) to Tables

- Create a table for the relationship set.
- Add all primary keys of the participating entity sets as fields of the table.
- Add a field for each attribute of the relationship.
- Declare a primary key using all key fields from the entity sets.
- Declare foreign key constraints for all these fields from the entity sets.
  For example - Consider following ER model

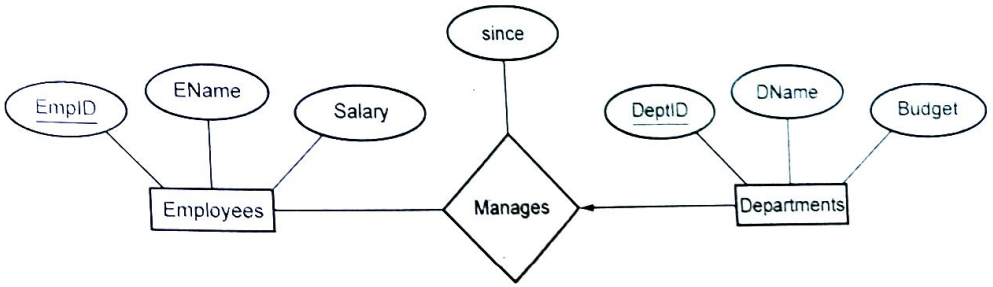


The SQL statement captures the information for relationship present in above ER diagram as follows -

```
CREATE TABLE Works_In  (EmpID CHAR(11),
                        DeptID CHAR(11),
                        EName CHAR(30),
                        Salary INTEGER,
                        DeptName CHAR(20),
                        Building CHAR(10),
                        PRIMARY KEY(EmpID,DeptID),
                        FOREIGN KEY (EmpID) REFERENCES Employee,
                        FOREIGN KEY (DeptID) REFERENCES Department
                        )
```

2.13.3 Mapping Relationship Sets(With Constraints) to Tables

- If a relationship set involves **n** entity sets and some **m** of them are linked via arrows in the ER diagram, the key for anyone of these **m** entity sets constitutes a key for the relation to which the relationship set is mapped.
- Hence we have **m** candidate keys, and one of these should be designated as the **primary key.**
- There are two approaches used to convert a relationship sets with key constraints into table.
- Approach 1 :
  - By this approach the relationship associated with more than one entities is separately represented using a table. For example - Consider following ER diagram. Each **Dept has at most one manager, according to the key constraint on Manages.**



Here the constraint is each department has at the most one manager to manage it. Hence no two tuples can have same DeptID. Hence there can be a separate table named **Manages** with DeptID as Primary Key. The table can be defined using following SQL statement

```
CREATE TABLE Manages(EmpID CHAR(11),
                    DeptID INTEGER,
                    Since DATE,
                    PRIMARY KEY(DeptID),
                    FOREIGN KEY (EmpID) REFERENCES Employees,
                    FOREIGN KEY (DeptID) REFERENCES Departments)
```

- Approach 2 :
  - In this approach , it is preferred **to translate a relationship set with key constraints**.
  - It is a superior approach because, it avoids creating a distinct table for the relationship set.
  - The idea is to include the information about the relationship set in the table corresponding to the entity set with the key, taking advantage of the key constraint.
  - This approach eliminates the need for a separate Manages relation, and queries asking for a department's manager can be answered without combining information from two relations.
  - The only drawback to this approach is that space could be wasted if several departments have no managers.
  - The following SQL statement, defining a **Dep_Mgr** relation that captures the information in both **Departments** and **Manages**, illustrates the second approach to translating relationship sets with key constraints :

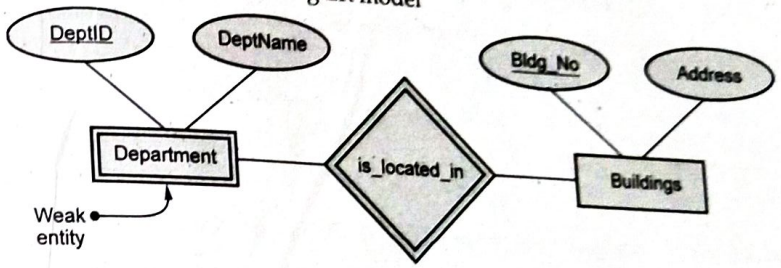
```
CREATE TABLE Dep_Mgr ( DeptID INTEGER,
                       DName CHAR(20),
                       Budget REAL,
                       EmpID CHAR (11),
                       since DATE,
                       PRIMARY KEY (DeptID),
                       FOREIGN KEY (EmpID) REFERENCES Employees)
```


2.13.4 Mapping Weak Entity Sets to Relational Mapping

A weak entity can be identified uniquely only by considering the primary key of another (owner) entity. Following steps are used for mapping Weka Entity Set to Relational Mapping

- Create a table for the weak entity set.
- Make each attribute of the weak entity set a field of the table.
- Add fields for the primary key attributes of the identifying owner.
- Declare a foreign key constraint on these identifying owner fields.
- Instruct the system to automatically delete any tuples in the table for which there are no owners

For example - Consider following ER model

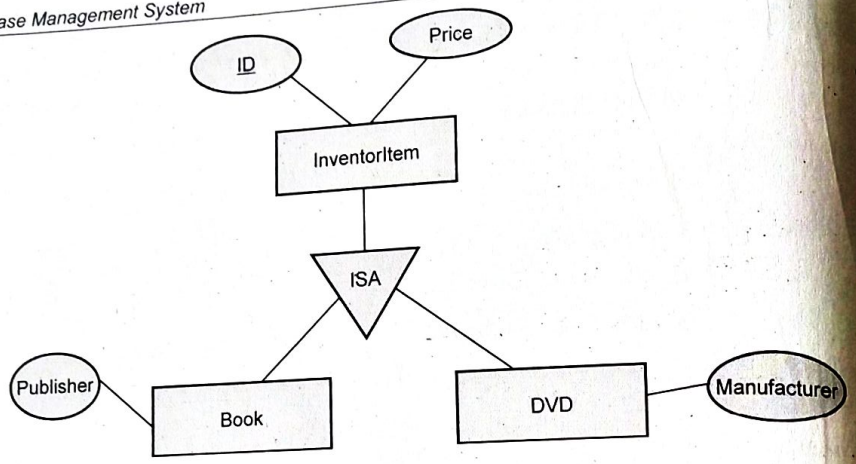


Following SQL Statement illustrates this mapping

```
CREATE TABLE Department(DeptID CHAR(11),
        DeptName CHAR(20),
        Bldg_No CHAR(5),
        PRIMARY KEY (DeptID,Bldg_No),
        FOREIGN KEY(Bldg_No) References Buildings on delete cascade
        )
```

2.13.5 Mapping of Specialization / Generalization(EER Construct) to Relational Mapping

The specialialization/Generalization relationship(Enhanced ER Construct) can be mapped to database tables(relations) using three methods. To demonstrate the methods, we will take the – InventoryItem, Book, DVD



Method 1 : All the entities in the relationship are mapped to individual tables

InventoryItem(ID , Price)
Book(ID,Publisher)
DVD(ID, Manufacturer)

Method 2 : Only subclasses are mapped to tables. The attributes in the superclass are duplicated in all subclasses. For example -

Book(ID, Price,Publisher)
DVD(ID, Price,Manufacturer)

Method 3 : Only the superclass is mapped to a table. The attributes in the subclasses are taken to the superclass. For example -

InventoryItem(ID , Price,Publisher,Manufacturer)

This method will introduce **null** values. When we insert a **Book** record in the table, the **Manufacturer** column value will be null. In the same way, when we insert a **DVD** record in the table, the **Publisher** value will be null.

Example 2.13.1 *Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Also construct appropriate tables for the ER diagram you have drawn.*

Solution :

ER Diagram - Refer example 2.4.15.

Relational Mapping

patients (P_id, name, insurance, date-admitted, date-checked-out)
doctors (Dr_id, name, specialization)
test (testid, testname, date, time, result)
doctor-patient (P_id, Dr_id)
test-log (testid, P_id) performed-by (testid, Dr_id)

Review Question

1. *Writing short note on – Mapping ISA relationship of E-R diagram to tables*

SPPU : Nov.-18, (End Sem), Marks 4

2.14 Multiple Choice Questions

Q.1 ER model was introduced by _____.

[a] E.F. Codd [b] P.P. Chen
[c] Bjarne Stroustrup [d] None of these

Q.2 ER diagram an ellipse represents _____.

[a] weak entity [b] relationship
[c] attribute [d] entity class

Q.3 In ER diagram the relationship is represented by _____.

[a] rectangle [b] ellipse
[c] diamond [d] circle

Q.4 Relationship among entities of a single class is called as _____.

[a] IS-A relationship [b] recursive relationship
[c] HAS -A relationship [d] none of these

Q.5 The relationship used for connecting entities of different types when identifiers are different _____.

[a] HAS-A relationship [b] IS-A relationship
[c] recursive relationship [d] none of these

Q.6 Which is not an example of strong entity type ?

[a] Employee [b] Department
[c] Emp_ID [d] Course

Q.7 If person is an entity then "Ankita" and "Prajkta" is the entity _____.
[a] characteristics [b] field
[c] identifier [d] instance

Q.8 Properties that describe the characteristics of entities are called :
[a] entities [b] attributes
[c] identifiers [d] relationships

Q.9 A student RollNumber, Name and Marks are all examples of _____.
[a] entities [b] attributes
[c] relationships [d] none of these

Q.10 An attribute which consists of a group of attributes is called as _____.
[a] composite attribute [b] single valued attribute
[c] multi-valued attribute [d] none of these

Q.11 The attribute "age" is calculated from "date_of_Birth". The attribute "age" is _____.
[a] single valued [b] multi valued
[c] composite [d] derived

Q.12 The most common type of relationship used in data modeling is _____.
[a] unary [b] binary
[c] ternary [d] none of these

Q.13 An entity type whose existence depends on another entity type is called ___ entity.
[a] strong [b] weak
[c] dependent [d] mutually dependent

Q.14 Every weak entity set can be converted into a strong entity set by :
[a] Using generalization [b] Adding appropriate attributes
[c] Using aggregation [d] None of the above

Q.15 In a one-to-many relationship, the entity that is on the one side of the relationship is called a(n) _____ entity.
[a] parent [b] child
[c] instance [d] subtype

Q.16 Which of the following is NOT a basic element of all versions of the E-R model ?
[a] Entities [b] Attributes
[c] Relationships [d] Primary keys

Q.17 The entity employee has three candidate keys : 1) EmpID 2) Email 3) Date_of_joining 4) Designation. Suggest best primary key for this entity :
[a] EmpID [b] Email
[c] Date_of_joining [d] Designation

Q.18 Foreign key is one in which the ____ of one relation is referenced in another relation.
[a] candidate key [b] constraint
[c] primary key [d] foreign key

Q.19 The relationship between the two tables are created using _____.
[a] primary key [b] check constraints
[c] candidate key [d] foreign key

Q.20 Constraints are preferred methods for enforcing _____.
[a] data abstraction [b] data access
[c] data inheritance [d] data integrity

Answer Keys for Multiple Choice Questions :

Q.1	b	Q.2	c	Q.3	c	Q.4	b
Q.5	a	Q.6	c	Q.7	d	Q.8	b
Q.9	b	Q.10	c	Q.11	d	Q.12	b
Q.13	b	Q.14	b	Q.15	a	Q.16	d
Q.17	a	Q.18	c	Q.19	d	Q.20	d

❑❑❑