CHAPTER 1

Topic 1

Database Introduction

1.1 Introduction to DBMS

1. Data

– The facts and figures that can be recorded in system and that have special meaning assigned to it is called as data.

Example

– Data of a customer like name, telephone number, address and product purchased date etc.

2. Database

– A database is a collection of data items stored in one place and having some common base (Background) between them.

Example

- College database contains data such as teachers, students, books, canteen but college is common (Base) between all above data items.

Student table

Sid	Name	Class	Major

Course Table

Cid	Name	Hours

Department Table

Did	Name

Marks Table

Sid	Cid	Marks	Grade

Fig. 1.1.1 : Sample Student Database

3. Database Management System (DBMS)

– A **Database Management System (DBMS)** is a collection of software or programs which help user in creation and maintenance of a database (set of information). Hence it is also known as a computerized record-keeping system.

Examples :

– MS Access, Fox Pro by Microsoft.

– Oracle by Oracle corp.

– SQL Server By Microsoft.

1.2 File System v/s Database System

1. **Redundancy can be reduced**
2. **Inconsistency can be avoided**
3. **Data can be shared**
4. **Centralized control of data**
5. **Standards can be enforced**
6. **Security restrictions can be applied**
7. **Integrity can be maintained**
9. **New applications** may be developed using the existing database.

Database Management System	File Processing System
Computerized record – keeping system is used in DBMS	Collection of individual files accessed by applications programs is called File Processing System
DBMS allows flexible access to data	File – Processing System is designed to allow predetermined access to data
It co-ordinates both the physical and logical	It co-ordinates only the physical access to data
DBMS provides multiple user interface	Data is isolated in the file system
Unauthorized access is restricted in DBMS	Unauthorized access cannot be restricted
Redundancy can be controlled	Redundancy cannot be controlled

1.3 Database Users

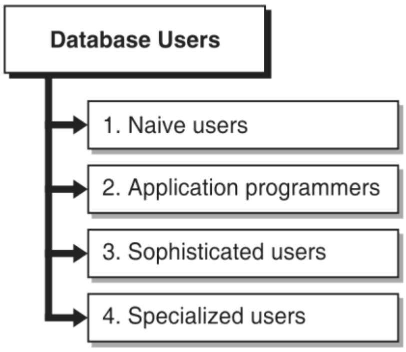


Fig. 1.4.1 : Database Users

1. Naive users

– Naive users are users who interact with the system using application programs that have been developed previously.

2. Application programmers

– Application programmers responsible for writing application programs that use the database.

– **Rapid Application Development (RAD)** tools are available nowadays that enable an application programmer to construct application without writing code.

3. Sophisticated users

– Sophisticated users interact with application without writing programs by using a database query language.

4. Specialized users

– Creates the actual database and implements technical controls needed to enforce various policy decisions.

CHAPTER 1

Topic 2

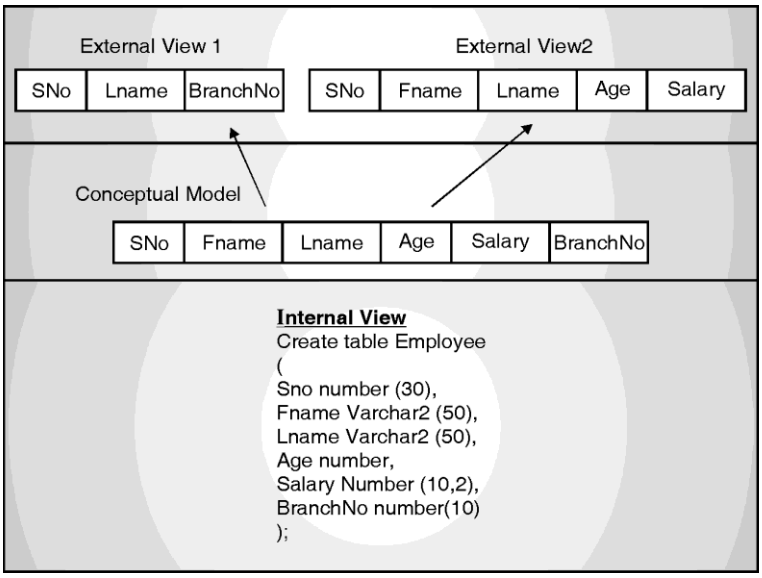
Database Architecture

2.1 Three-Levels Schema Architecture

1. Introduction

- The goal of the three-schema architecture is to separate the front end (user applications interface) and the back end (physical database).

2. Database architecture



```
Create table Employee
(
Sno number (30),
Fname Varchar2 (50),
Lname Varchar2 (50),
Age number,
Salary Number (10,2),
BranchNo number(10)
);
```

Fig. 2.1.1 : Database schema levels

(I) Internal Level (Physical Level)

- The internal level is very close to physical storage of data.
- The internal (or physical) database is stored on secondary storage devices, mainly the magnetic disk.

Example :

```
Create table Employee
(   Sno         number (30),
    Fname       varchar2 (50),
    Lname       varchar2 (50),
    Age             number,
    Salary      number (10,2),
    BranchNo    number (10) );
```

(II) Conceptual level

- The conceptual schema hides the internal details of physical storage and targets on describing entities, data types, relationships and constraints.
- The conceptual schema contains all the information to build relevant external records. As the conceptual model is derived from the physical model.

(III) External level (view level)

- The external level is the one closest to the user, i.e., it is related with the way data is viewed by individual end users.
- External views are the proper interface between the user and the database, as an individual user can hardly be expected to be interested in the entire database.
- The external model is derived from the conceptual model.
-

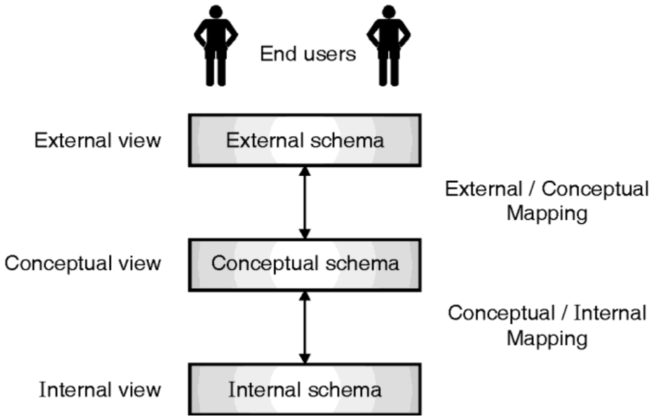


Fig. 2.1.2 : Three level schema architecture

(IV) Mapping

- These mappings may be time-consuming, so small databases do not support external views.
- **External / conceptual mapping :** The DBMS must transform a request on an external schema into a request against the conceptual schema.
- **Conceptual / internal mapping :** A certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

2.4 Detailed DBMS Architecture

- A database system can be separated into two different modules that deal with all operations of the overall system.

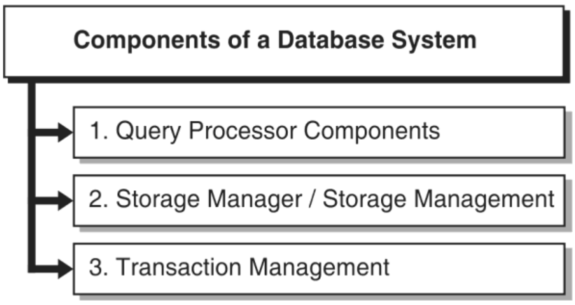

Components of a Database System
1. Query Processor Components
2. Storage Manager / Storage Management
3. Transaction Management

Fig. 2.4.1 : Components of a Database System

- The storage manager is important because databases typically require a huge amount of storage space.

2.4.1 Query Processor Components

Q. Write a short notes on : Query processor. **(3 Marks)**

1. Introduction

The query processor will accept query from user and solves it by accessing the database.

2. Parts of query processor

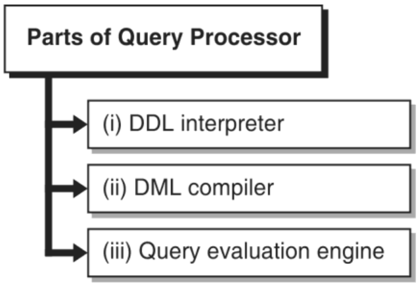


Fig. 2.4.2 : Parts of Query Processor

(i) DDL interpreter

This will interpret DDL statements and fetch the definitions in the data dictionary.

(ii) DML compiler

- A query can usually be translated into any of a number of alternative evaluation plans for same query result DML compiler will select best plan for **query optimization**.

(iii) Query evaluation engine

This engine will execute low-level instructions generated by the DML compiler on DBMS.

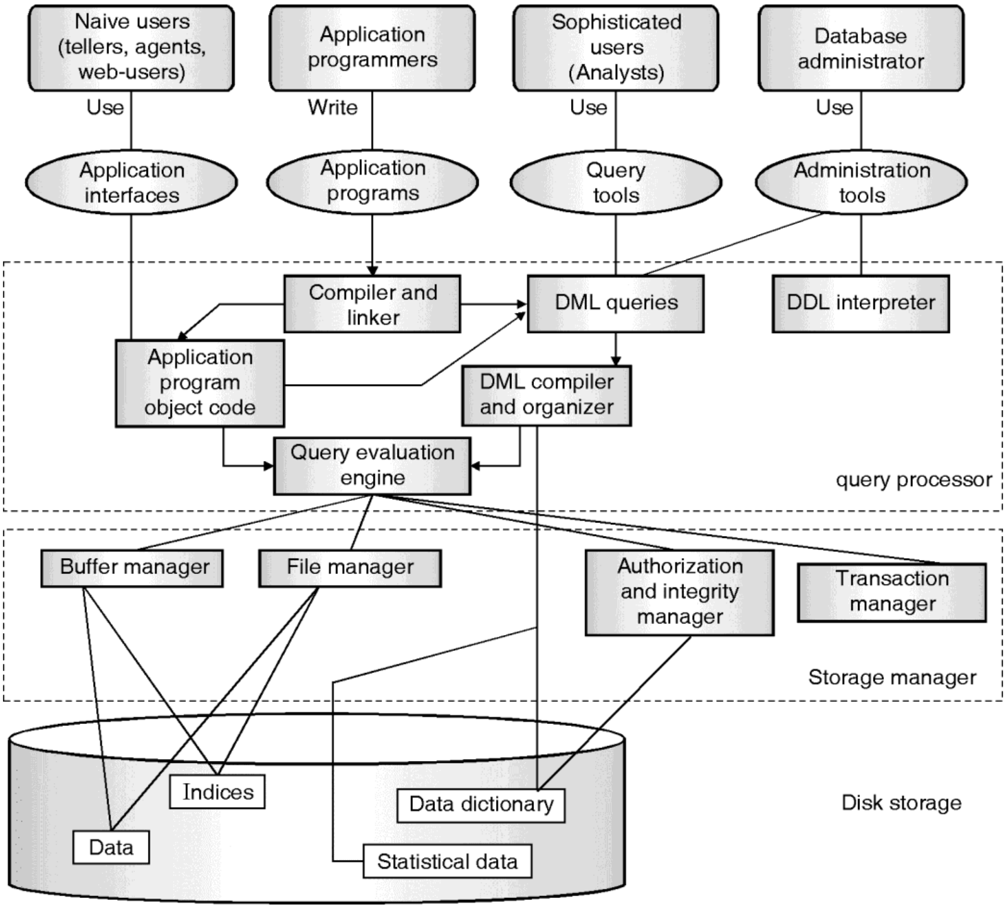


Fig. 2.4.3 : Components of DBMS

2.4.2 Storage Manager / Storage Management

- A **storage manager** is a program module which acts like interface between the data stored in the database and the application programs and queries submitted to the system.
- The data is stored on the disk using the file system.
- The storage manager is programme which is responsible for the interaction with the file manager.
- The storage manager components include
  - **Authorization and integrity manager**
  - **Transaction manager**
  - **File manager**

    Buffer manager
- Data structures implemented by storage manager,
  - **Data files**
  - **Data dictionary**
  - **Indices**

CHAPTER 2

Topic 1

ER Diagrams

3.1 Entities

(1) Introduction

An **entity** is anything in real world with its own independent existence.

Example : Student, faculty, subject having independent existence.

(2) Entity set : Entity set is collection of all entities of same type.

(a) Strong entity type
(b) Weak entity type

Strong entity type

- Entity type which has its own key attributes by which we can identify specific entity uniquely is called as strong entity type.

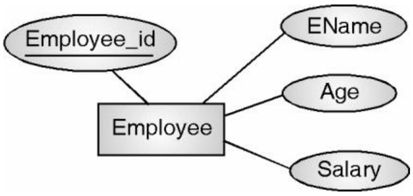


Fig. 3.2.1 : Employee entity

Weak entity type

- Entity type which cannot form distinct key from their attributes and takes help from corresponding strong entity is called as weak entity type.

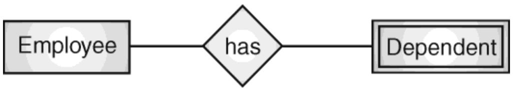


Fig. 3.2.2 : Weak entity "dependent"

3.2 Attributes

Properties that describe an entity are known as **attribute**.

Example : For an employee of ID 30 value of name attribute is 'Jayendra'

(a) Composite attributes

- Composite attributes which can be divided into subparts.
- **Example :** Name attribute can be divided into First_Name and Last_Name

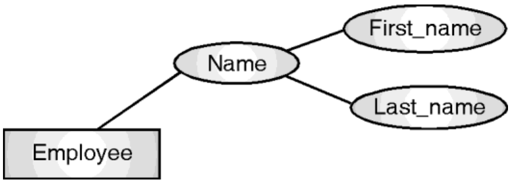
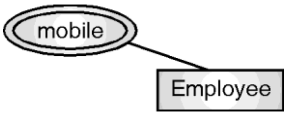


Fig. 3.3.1(b) : Composite attributes

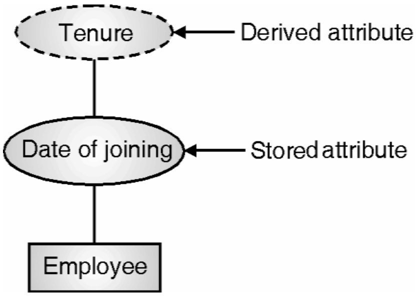
(b) Multi valued attributes

- The attribute having many values for a particular entity is called as multi-valued attribute
- Example : Each Employee has multiple mobile numbers



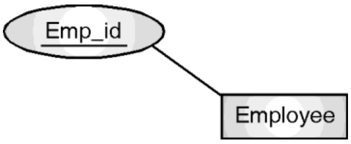
(c) Derived attributes

- The value of this attributes which can be derived from the value of related stored attribute is called as derived attributes



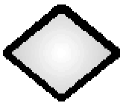
(d) Key attributes

This is an attribute of an entity which must have a unique value by which any row can be identified is called as key attribute of entity



3.3 Relationships

(1) A relationship is an association among one or more than one entity.

Type	Notation
Relationship	

3.4 Relationship Types based on Constraints

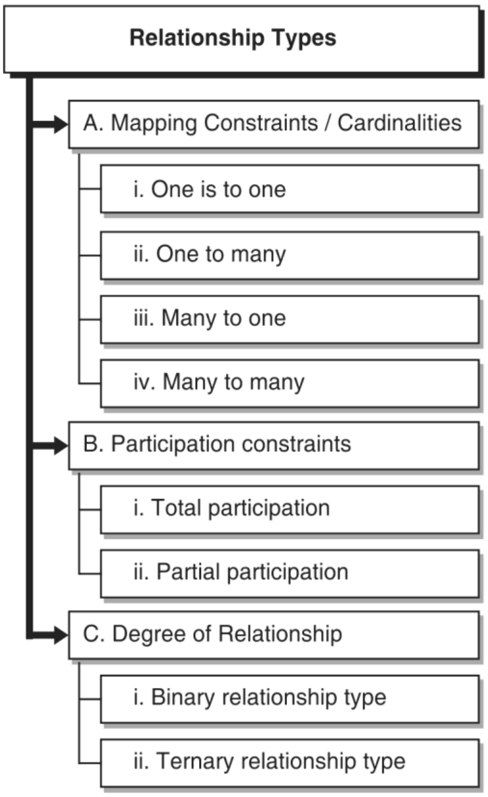

Relationship Types
A. Mapping Constraints / Cardinalities
i. One is to one
ii. One to many
iii. Many to one
iv. Many to many
B. Participation constraints
i. Total participation
ii. Partial participation
C. Degree of Relationship
i. Binary relationship type
ii. Ternary relationship type

Fig. 3.5.1 : Relationship Types

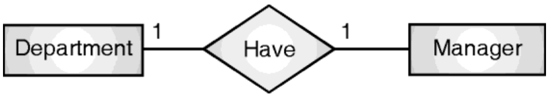
(A) Mapping Constraints / Cardinalities

– Number of entities from each side participating in a relationship set.

Types of mapping constraints

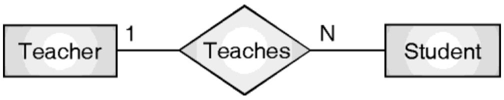
(i) One is to one

– That is one row in table is related with only one row in other table.



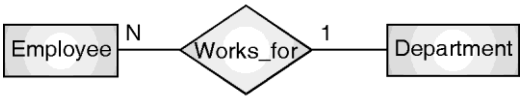
(ii) One to many

– A associated with any number of entities in B.



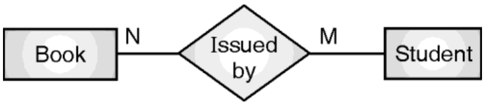
(iii) Many to one

- A associated with at most one entity in B.



(iv) Many to many

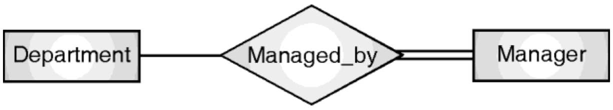
- A associated with any number of entities in entity B.



(B) Participation constraints

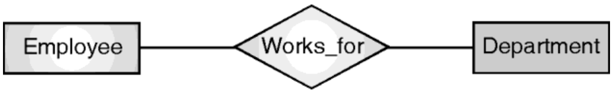
(i) Total participation

- In case of total participation every object in an entity must participate in a relationship.



(ii) Partial participation

- In case of partial participation more than one object in an entity may participate in a relationship.

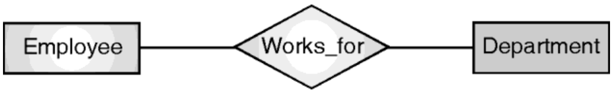


(C) Degree of Relationship (Binary Vs ternary)

– The degree of the relationship type is number of participating entity types.

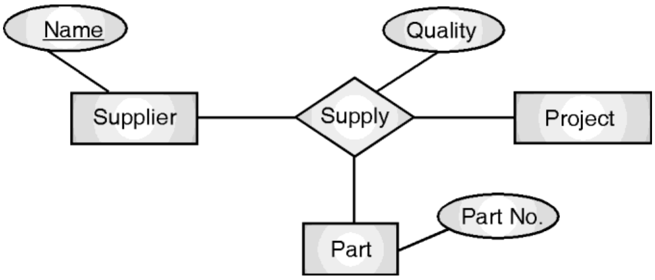
(i) Binary relationship type

- A relationship of degree two.



(ii) Ternary relationship type

– A relationship of degree three.



3.5 Sample ER Model

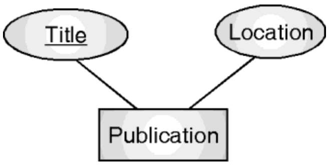
A publication may be a book or an article. Articles are published in Journals. Publication has title and location. Book having their title and category. Article includes title, Topic and date. Publication is written by Authors stores Name, address and mobile number. Publication also belongs to particular subject which has their names.

Step 1 : Identify entities

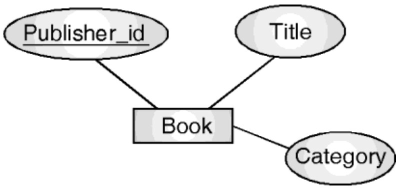
1. Publication 2. Book
3. Article 4. Journal
5. Subject 6. Author

Step 2 : Identify attributes

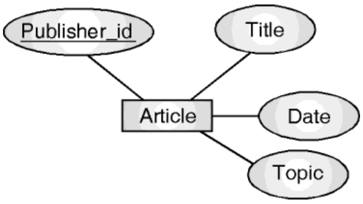
1. Publication (Title, Location)



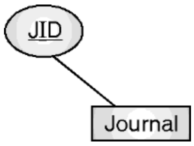
2. Book (Publisher_id, Title, Category)



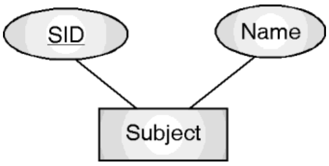
3. Article (Publisher_id, Title, Date, Topic)



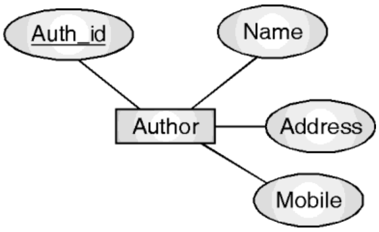
4. Journal (<u>JID</u>)



5. Subject (<u>SID</u>, Name)

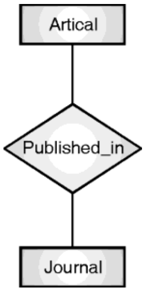


6. Author (<u>Auth_id</u>, Name, Address, Mobile)

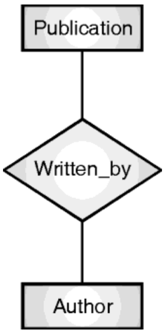


Step 3 : Identify relationships

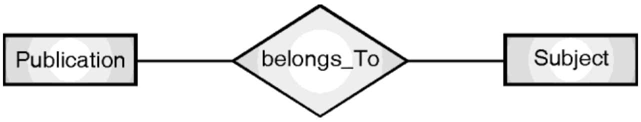
1. Articles are published in Journal.



2. Publication is written by Author.



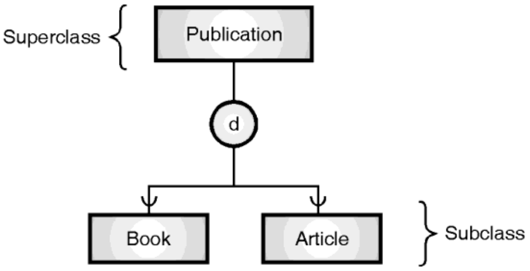
3. Publication belongs to a particular subject.



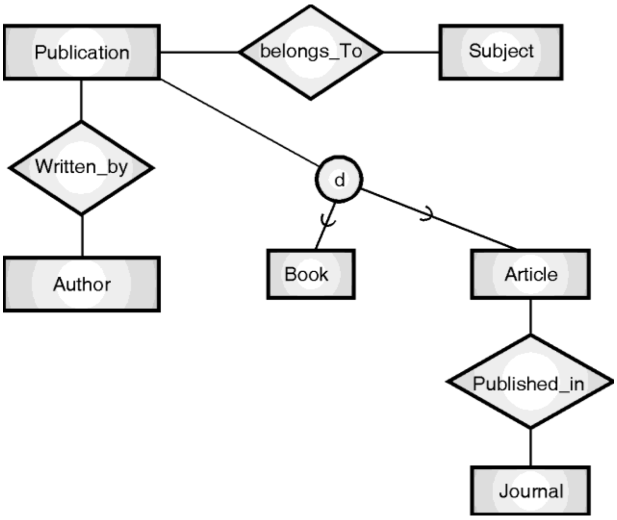
Step 4 : Identify inheritance relations

Publication can be

BOOK or ARTICLE.



Step 5 : Merging all above relations we will get final ER model

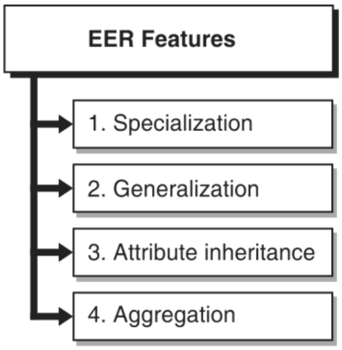


CHAPTER 2

Topic 2

EER Diagrams

- EER model includes all the modeling concept of ER model. In addition, it also includes the concept of aggregation, specialization and generalization.



3.6.1 Specialization

- Top down approach of superclass / subclass relationship.
- Specialization is a process of defining a set of subclass of entity type, this entity type is called super **class of specialization.**

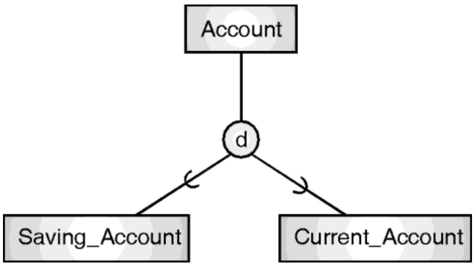


Fig. 3.6.2 : Specialization

3.6.2 Generalization

This is reverse process of specialization or this is bottom up approach of Superclass/subclass relationship.

3.6.3 Attribute inheritance

- The attributes of higher and lower level entities created by specialization and generalizations are attributes inheritance.

3.7 Aggregation

– Aggregation is meant to represent a relationship between a whole object and its component parts.

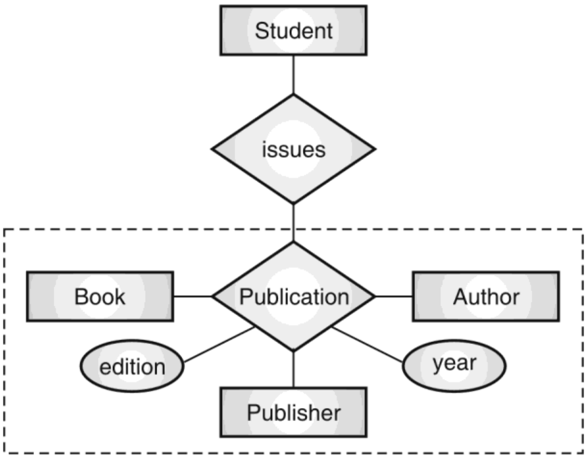


Fig. 3.7.1 : Aggregation

3.8 Solved ER Designing Examples

A publication may be a book or an article. Articles are published in Journals. Publication has title and location. Book having their title and category. Article includes title, Topic and date. Publication is written by Authors stores Name, address and mobile number. Publication also belongs to particular subject which has their names.

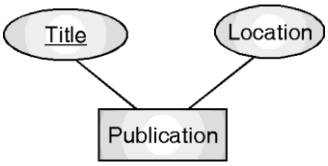
Soln. :

Step 1 : Identify entities

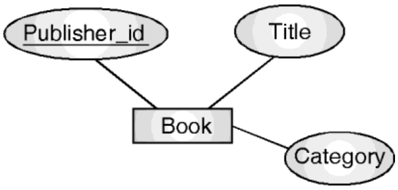
1. Publication	2. Book	3. Article
4. Journal	5. Subject	6. Author

Step 2 : Identify attributes

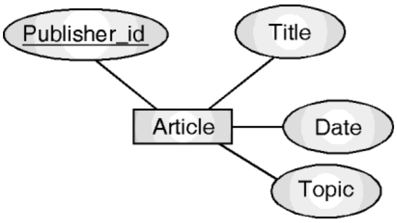
1. Publication (<u>Title</u>, Location)



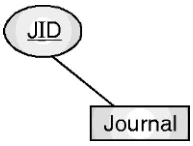
2. Book (<u>Publisher_id</u>, Title, Category)



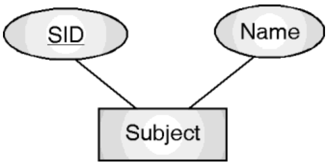
3. Article (<u>Publisher_id</u>, Title, Date, Topic)



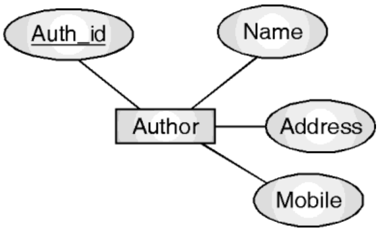
4. Journal (<u>JID</u>)



5. Subject (<u>SID</u>, Name)

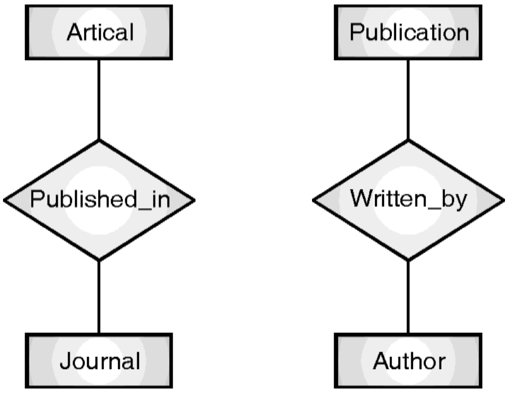


6. Author (<u>Auth_id</u>, Name, Address, Mobile)

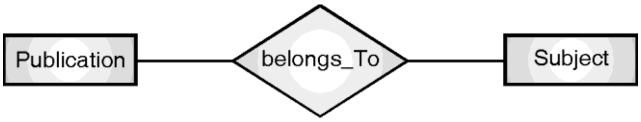


Step 3 : Identify relationships

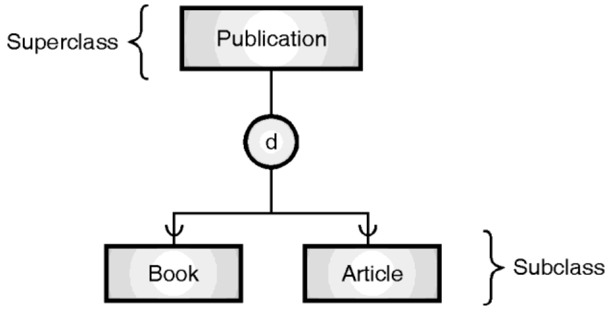
1. Articles are published in Journal.
2. Publication is written by Author.



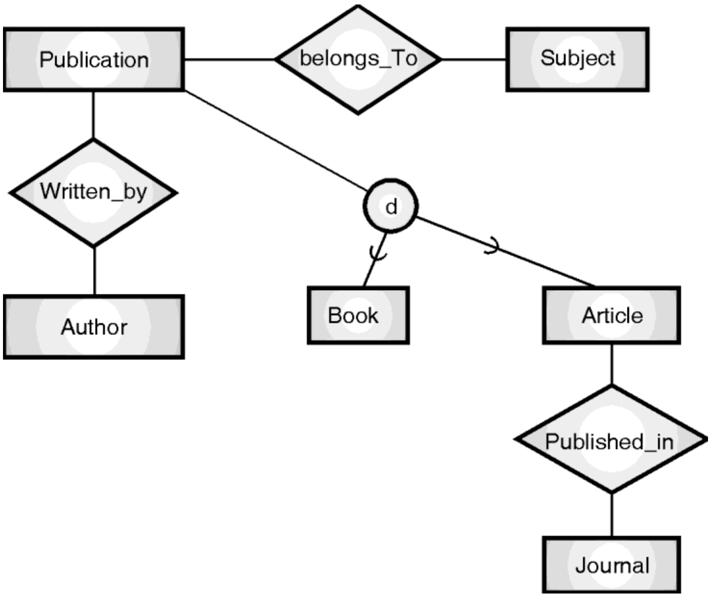
3. Publication belongs to a particular subject



Step 4 : Identify inheritance relations
Publication can be
BOOK or ARTICLE.



Step 5 : Merging all above relations we will get final ER model



CHAPTER 2

Topic 3

EER to Tables

4.1 Relational Model

1. Introduction

– The relational model first proposed by E. F. Codd hence he is known as father of relational model.

– Tables are known as **relations**, columns are known as **attributes** and rows (or records) are known as **tuples**.

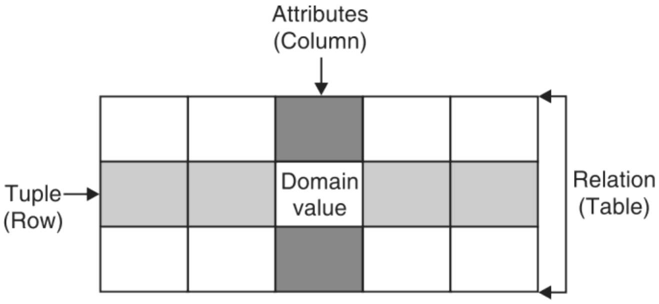


Fig. 4.1.1 : Relational Algebra Notations

2. Relation (Table)

– The table containing rows and columns represents entity in relational model, it is called as Relation.

Id	Name	Age	Class	Branch
105	Mahesh	25	BE	IT
106	Suhas	28	FE	CS

3. Attributes (Column)

– Relation has its own properties which describes that relation (table) such properties are known as attributes

4. Tuple (Row / Records)

– A single row in relational table which contains all the information about a single entity is called as Tuple.

–

Stud_Id	Name	Age	Std	Div
105	Mahesh	25	BE	A

5. Domain (Data Value)

- The intersection column and row in a relational table which represents data of entity is called as Domain.

Name
Mahesh

4.2 Mapping Entities to Tables

(1) Regular entity types

- Regular entity sets can be represented as table in relational model.
- Attributes of entity set can be converted to the columns (attributes) of the tables in relational model.

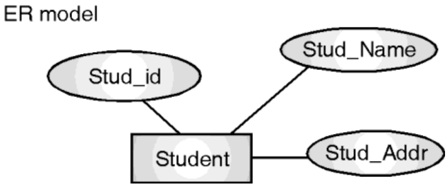


Fig. 4.8.1 : Regular entity

Stud_id	Stud_Name	Stud_Addr
1	Snehal	Mumbai
2	Pratiksha	Mumbai
3	Supriya	Mumbai
4	Tanmay	Goa

(2) Weak entity types

For each weak entity type with owner entity, create a table and include all simple attributes of weak entity type as columns of table, including foreign key attributes as the primary key attribute of the table that correspond to the owner entity type.

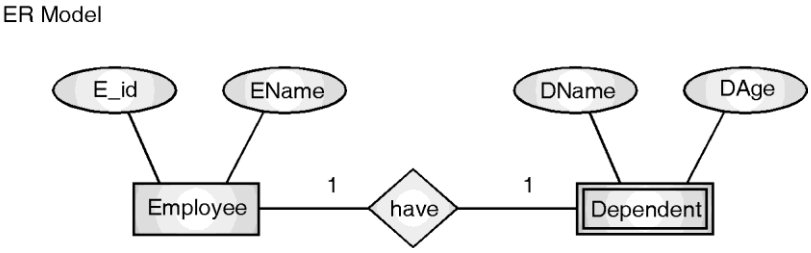


Fig. 4.8.2 : Weak entity

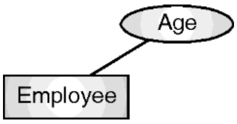
E_id	Ename	DName	DAge
1	Sachin	Jyoti	23
2	Suhas	Manju	22
3	Jayendra	Tanya	27

4.3 Mapping Attributes to Columns of Table

Q. Explain how to convert attribute in ER to relational Table. **(5 Marks)**

(a) Simple attributes

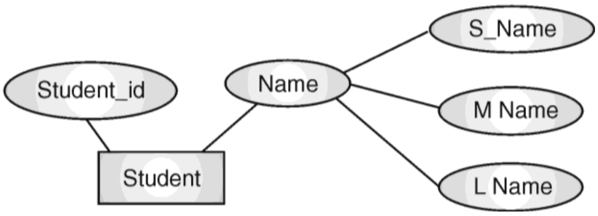
Simple attribute can be directly converted to a column (Attribute) in relational model.



Eid	Age
1	23
2	24
3	43
4	28

(b) Composite attributes

These attributes need to be stored as set of simple component attributes (Columns) in relational model by avoiding actual attribute



Student_id	S_Name	MName	LName
1	Harshad	Rupali	Malar
2	Bipin	Anand	Shinde
3	Aanand	Ganesh	Panchal
4	Tushar	Bipin	Pimple

(c) Multi valued attributes

Multi valued attributes are mapped as a relation which includes combination of the primary key of table and multi valued attribute as a composite primary key

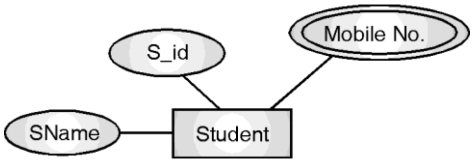
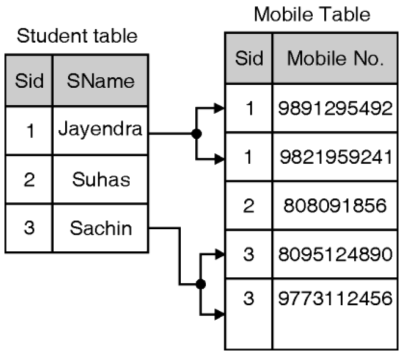


Fig. 4.9.3 : Multi valued attribute

Mobile table	
Sid	**Mobile No.**
1	9891295492
1	9821959241
2	8080918456
3	8095124890
3	9773112456



Student table

Sid	SName
1	Jayendra
2	Suhas
3	Sachin

Mobile Table

Sid	Mobile No.
1	9891295492
1	9821959241
2	808091856
3	8095124890
3	9773112456

Fig. 4.9.4

(d) Derived attributes

There is no need to store such attribute in relational model

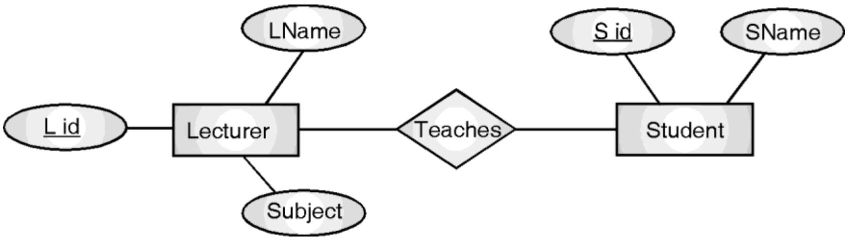
(e) Key attributes

Key attribute in ER Model can be directly converted to primary key attribute of relational model.

4.4 Mapping Relationships

(a) Foreign key approach

If binary relationship type does not possess many attributes then we can map such relation using foreign key.

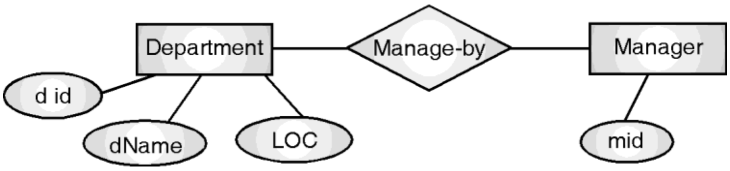


Lecturer table		
Lid	**LName**	**Subject**
1	Omprakash	IP
2	Yogesh	INS
3	Amit	PM

Student table		
Sid	**SName**	**Lid**
1	Bency	1
2	Deepak	1
3	Yogita	1
4	Snehal	2
5	Pratiksha	2

(b) Merged relationship approach

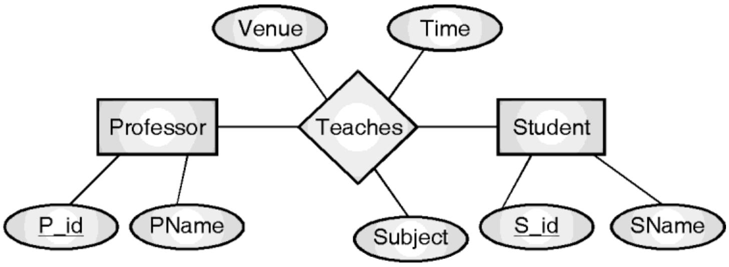
When participation is total it is possible to merge relation and involved entities as a single relation and then map it to a table.



Department table			
did	**Dname**	**Loc**	**mid**
10	IDF	Mahape	11
20	Mayban	Pune	22
30	Lax	Mumbai	33

(c) Cross reference approach

– A relationship type in EER is mapped to new table in relational model.



Professor table	
Pid	**PName**
1	Om
2	Nitin

Student table	
Sid	**SName**
1	Snehal
2	Tanmay

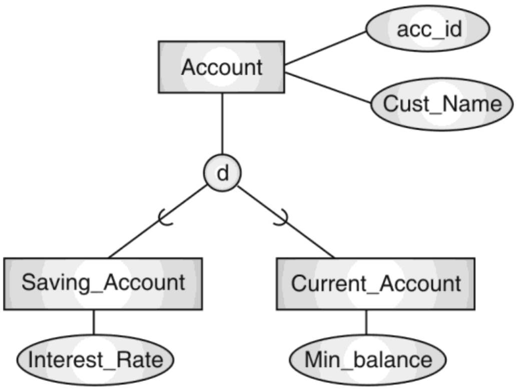
Teacher relation table				
Pid	**Sid**	**Venue**	**Time**	**Subject**
1	1	SFIT	1 pm	ADBMS
1	2	XIE	1 pm	DBMS

4.5 Mapping Inheritance constraints

– Each super class and subclass entity sets represents table in relational model.

– Attributes of entity set is converted to the columns of the tables in relational model.

– The primary key column of super class is also added to all subclasses and treated as a primary key column for all tables in relational model.



Account table	
acc_id	**Cust_Name**
1	Snehal
2	Tanmay
3	Nikhil

Saving_Account table	
acc_id	**Interest_Rate**
1	10
2	12
3	15

Current_Account	
acc_id	**Min balance**
1	1000
2	2000
3	500