Module - I

Introduction of DBMS orientation - Data - Information - Database - filesystem - DBMS

- Data is the raw material that is to be processed for information or for collection of details.
- Data is plain fact and it has to be processed for further information.
- Data is useless unless it is processed or has been made into something.

Examples of Data:

(i) Student Data on Admission Forms: When students get admission in a college, they fill admission form. This form contains raw facts (data of student) like name, father's name address, etc.

(ii) Students examination data: In examination data about obtained marks of different subjects for all students is collected.

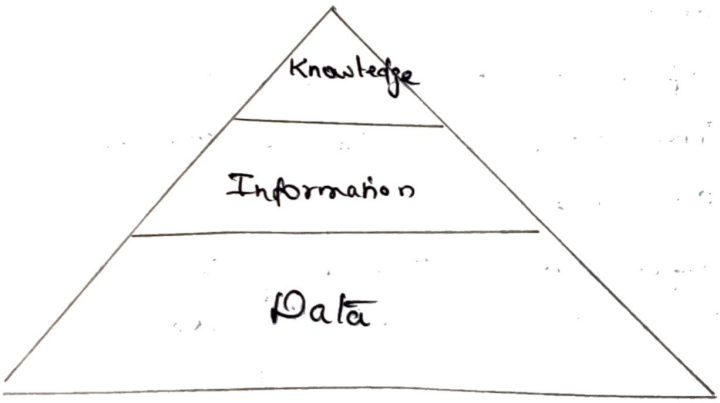
(iii) Survey Data: Different companies collect data by survey to know the opinion of people about their product.

- Information:
  - Information is processed data
  - The data that can be made useful is known as information.
  - Information is basically the data plus the meaning of what the data was collected for.

- Data does not depend upon Information but information depends upon data.

Key Differences:

	Data	Information
1.	Data is the input language for a computer	information is the output language for human.
2.	Data is unprocessed facts or mere figures.	Information is processed data which has been made sense of.
3.	Data doesn't depend on information.	Information depends on data and without it, information can't be processed
4.	Data is not specific.	Information is specific enough to generate meaning.
5.	Data is the raw material that is collected	information is a detailed meaning generated from the data



Files

- A file represents a sequence of bytes on the disk where a group of related data is stored.
- File is created for permanent storage of data.

Example: File created in 'C' language.

```
#include <stdio.h>
struct emp
{
   char name[10];
   int age;
};
void main()
{
   struct emp e;
   FILE *p, *q;
   p = fopen("one.txt", "a")
   q = fopen("one.txt", "r")
   printf("Enter Name and age");
   scanf("%s %d", e.name, e.age);
   fprintf(p,"%s %d", e.name, e.age);
   fclose(p);
   do
      {
         fscanf(q, "%s %d", e.name, e.age);
         printf("%s %d", e.name, e.age);
      } while( !feof(q));
}
```

File System to DBMS

- File System manages data using files in hard disk. Users are allowed to create, delete and update the files according to their requirements.
- Let us consider the example of file based university Management system. Data of student is available to their respective Departments, Academics section, Result section, Accounts Section, Result section, Accounts section, Hostel office etc. Some of the data is common for all sections like Roll no, Name, Father Name, Address and phone number of students, but some data is available to a particular section only like Hostel allotment number which is a part of Hostel office.

Issues in File System:

(i) Redundancy of data: Data is said to be redundant if same data is copied at many places. If a student wants to change phone number, he has to get it updated at various sections. Similarly old records must be deleted from all sections representing that student.

(ii) Inconsistency of Data: Data is said to be inconsistent if multiple copies of same data doesn't match with each other. If phone number is different in Accounts section and Academics Section, It will be inconsistent.

(iii) Difficult Data Access: A user should know the exact location of file to access data, so the process is very cumbersome and tedious. If user wants to search student hostel allotment number of a student from 10000 unsorted student records how difficult it can be.

iv) Unauthorized Access: File System may lead to unauthorized access to data. If a student gets access to file having his marks, he can change it in unauthorized way.

v) No concurrent Access: The access of same data by multiple users at same time is known as concurrency. File System doesn't allow concurrency as data can be accessed by only one user at a time.

vi) No Backup and Recovery: File System doesn't incorporate any backup and recovery of data if a file is lost or corrupted.

DataBase:

- Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organises the data in the form of tables, views schemas, reports etc.

For example, University database organizes the data about students, faculty and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.

Database Management System:

- The software which is used to manage database is called Database Management System (DBMS)
- For example, MySQL, Oracle etc are popular commertial DBMS used in different applications.

DBMS allows users the following tasks:

i) Data Definition: It helps in creation, modification and removal of definitions that define the organization of data in database

ii) Data Updation: It helps in insertion, modification, and deletion of the actual data in the database.

iii) Data retrieval: It helps in retrieval of data from the database which can be used by applications for various purposes.

iv) User Administration: It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control and recovering information corrupted by unexpected failure.

Advantages of DBMS:

- DBMS helps in efficient organization of data in database which has following advantages over typical file system.

i) Minimized redundancy and data consistency:- Data is normalized in DBMS to minimize the redundancy which helps in keeping data consistent.

for example: student information can be kept at one place in DBMS and accessed by different users.

ii) Simplified Data Access: A user need only name of the relation not exact location to access data, so the process is very simple.

iii) Multiple data views: Different views of same data can be created to cater the needs of different users. For example, faculty salary information can be hidden from student view of data but shown in admin view.

iv) Data security: only authorized users are allowed to access the data in DBMS. Also, data can be encrypted by DBMS which makes it secure.

v) Concurrent access to data: Data can be accessed concurrently by different users at same time in DBMS

vi) Backup and Recovery mechanism: DBMS backup and recovery mechanism helps to avoid data loss and data inconsistency in case of catastrophic failures.

Need for DBMS:

- A Data Base Management System is a system software for easy, efficient and reliable data processing and management. It can be used for:
  - ✓ creation of a database
  - ✓ Retrieval of information from the database
  - ✓ updating the database
  - ✓ Managing a database.

File System Vs DBMS - Keys and its Importance

- File System:

i) It is a software system that manages and controls the data files in a computer system.

ii) File System doesn't support multi-user access.

iii) Data consistency is less in the file system.

iv) File system is not secured.

v) File system is used for storing the unstructured data.

vi) In the file system, data redundancy is high.

vii) No data backup and recovery process is present in a file system.

viii) Handling of a file system is easy.

ix) cost of a file system is less than the DBMS.

x) If one application fails, it doesn't affect other application in a system.

xi) In the file system, data can't be shared because it is distributed in different files.

xii) These system doesn't provide concurrency facility.

xiii) Example: NTFS (New Technology file system)
EXT (Extended file system) etc.

- DBMS:

i) It is a Software system used for creating and managing the database. DBMS provides a systematic way to access, update and delete data.

ii) Database Management system supports multi-user access.

iii) Data consistency is more due to the use of normalization.

iv) DBMS is highly secured

v) Database management System is used for storing the structure data.

vi) In DBMS, Data redundancy is low.

vii) There is a backup recovery for data in DBMS

viii) Handling a DBMS is complex.

ix) Cost of database management system is more than the file System.

x) If the database fails, it affects all application which depends on it.

xi) In DBMS, data can be shared as it is stored at one place in a database.

xii) This system provides concurrency facility.

xiii) Example: Oracle, MySQL, SQL Server, DB2

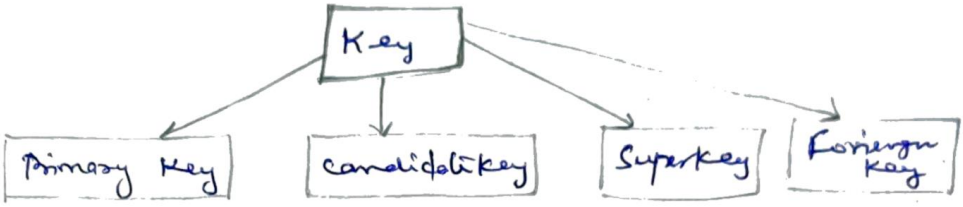
Keys and its Importance:

- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.
- For example: In student table, ID is used as a key because it is unique for each student. In PERSON table, Passport_number, license_number, SSN are keys since they are unique for each person

Student
ID
Name
Address
Course

Person
Name
DOB
Passport-NO
License-No
SSN

Types of Keys:



Candidate Key:

- The candidate keys in a table are defined as the set of keys that is minimal and can uniquely identify any data row in the table.

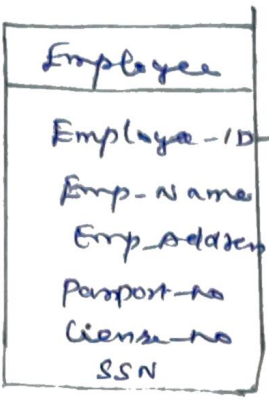
Example: In the Employee table, ~~ID~~ is best suited for the primary key, rest of the attributes like SSN, Passport_number, license_number etc, are considered as a candidate key.

Employee	
Employe-ID	– candidate key
Emp_Name	
Emp_Address	
Passport-NO	– candidate key.
License-NO	
SSN	

Primary Key:

- The primary key is selected from one of the candidate keys and becomes the identifying key of a table. It can uniquely identify any row of the table.

Example: In employee table, ID can be primary key since it is unique for each employee.



Super key:

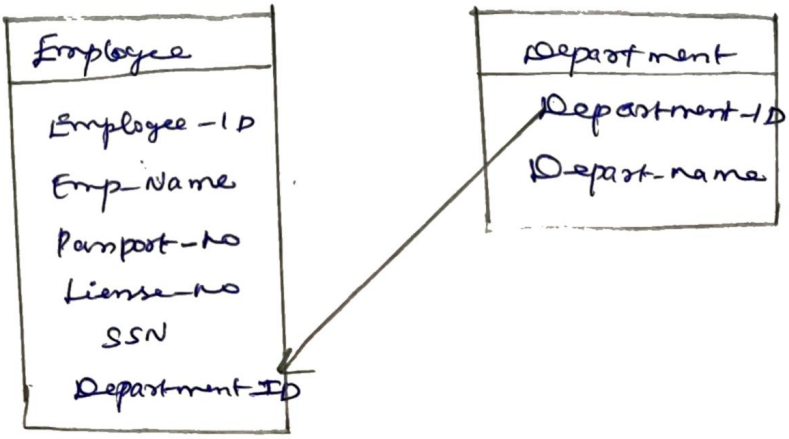
- It is the superset of primary key. The super key contains a set of attributes, including the primary key, which can uniquely identify any data row in the table.

Example:

The super key would be Employee-ID, (Employee-ID, Employee-name) etc.

Foreign key:

- A foreign key is an attribute value in a table that acts as the primary key in another table. Hence, the foreign key is useful in linking together these two tables.



Composite Key:

- If any single attribute of a table is not capable of being the key, i.e. it can't identify a row uniquely, then we combine two or more attributes to form a key. This is known as composite key.

Secondary Key:

- only one of the candidate keys is selected as the primary key. The rest of them are known as secondary keys.

Data Models & Entity Relationship Model.

Data Models:

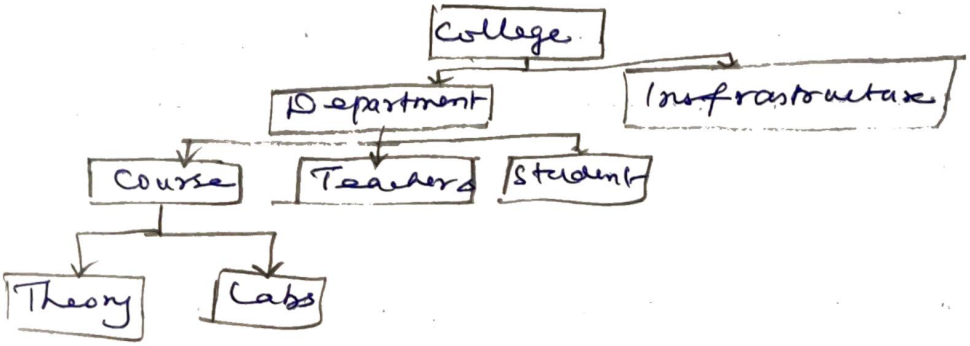
- Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.

Types:

(i) Hierarchical Model
(ii) Network Model
(iii) Entity-relationship Model
(iv) Relational Model.

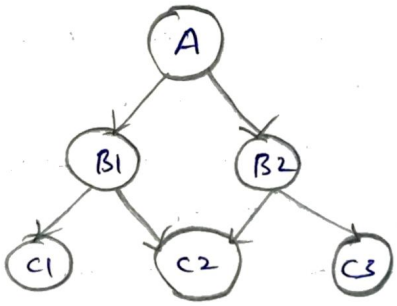
Hierarchical Model:

- This database model organises data into a tree-like structure, with a single root, to which all the other data is linked. The hierarchy starts from the Root data, and expands like a tree adding child nodes to the parent nodes.
- In this model, a child node will only have a single parent node.
- This model efficiently describes many real-world relationships like index of book, recipes etc.



Network Model :-

- This is an extension of the hierarchical model. In this model data is organized more like a graph, and are allowed to have more than one parent node.
- In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast.
- This was most widly used database model, before relational model was introduced.



Entity - Relationship Model :

- Entity Relationship (ER) model is based on the notion of ~~rela~~ real world entities and relationships among them. While formulating real-world scenario into the database model, the ER model creates entity set, relationship set, general attributes and constraints.
- ER model is best used for the conceptual design of a database.

- ER model is based on -
  1) Entities and their attributes.
  2) Relationships among entities.

Entity:

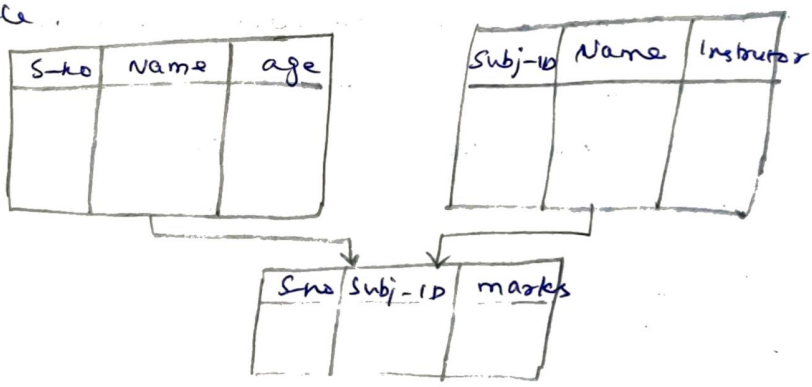
- An entity in an ER model is a real world entity having properties called attributes. Every attribute is defined by its set of values called domain.
- For example, in University database, a "student" is considered as an entity.
- Student has various attributes like name, age, class, etc.

Relationship:

- The logical association among entities is called relationship.
- Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.
- Mapping cardinalities:
  i) one to one
  ii) one to many
  iii) many to one
  iv) many to many

Relational model:

- In this model, data is organized in two-dimensional tables and the relationship is maintained by storing a common field.
- This model was introduced by E.F Codd in 1970, and since then it has been the mostly widely used database model, infact, we can say the only database model used around the world.
- The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.



ER Diagram.

- ER Model is represented by means of an ER diagram. Any object for, example entities attributes of an entity, relationship sets, and attributes of relationship sets can be represented with the help of ER diagram.

Entity:

- An entity can be a real-world object.

For example, in a University database, Students, Teachers, classes and courses offered can be considered as entities.

- An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values.
- For example, a Student set may contain all the students of a college, likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

Attributes:

- Entities are represented by means of their properties, called attributes. All attributes have values. For example, Student entity may have name, class, and age as attributes.

Types of attributes:

(i) Simple attribute - Simple attributes are atomic values, which can't be divided further. For example, a student's phone number is an atomic value of 10 digits.

(ii) Composite attributes - Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first-name and last-name.

(iii) Derived attribute:- Derived attributes are the attributes that don't exist in the database, but their values are derived from other attributes present in the database. Example, age can be derived can be derived from date-of-birth

iv) Single-value attribute: Single-value attributes contain single value. For example, Social_Security_Number.

v) Multi-value attribute - Multi-value attributes may contain more than one values. For example, a person can have more than one phone no, email-id etc.

Entity-set and Keys:

- Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.
- For example, the roll-number of a student makes him/her identifiable among students.

Super key - A set of attributes (one or more) that collectively identifies an entity in an entity set.

Candidate key - A minimal super key is called a candidate key. An entity set may have more than one candidate key.

Relationship:

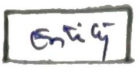
- The association among entities is called a relationship.
- For example, an employee works at a department, a student enrolls in a course. Here, works, enrolls are called relationships.

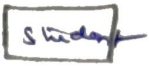
Relationship Set:

- A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes.
- These attributes are called descriptive attributes.

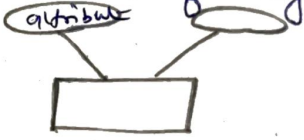
Components of ER Diagram:

1) Entity - simple rectangular box represents an entity



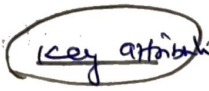


2) Attribute for any entity: - Ellipse is used to represent attributes of any, it is connected to the entity.



3) Weak Entity: A weak entity is represented using double rectangular boxes. It is generally connected to another entity.

4) Key Attribute for any entity: - To represent a key attribute, the attribute name inside the ellipse is underlined.



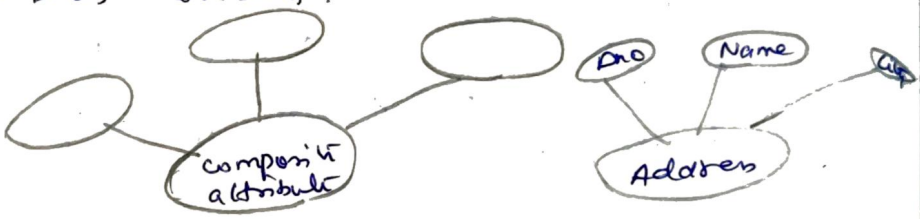
5) Derived Attribute for any entity: Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth.

6) Multivalued Attributes for any entity :-

Double ellipse, one inside another, represent the attribute which can have multiple values.

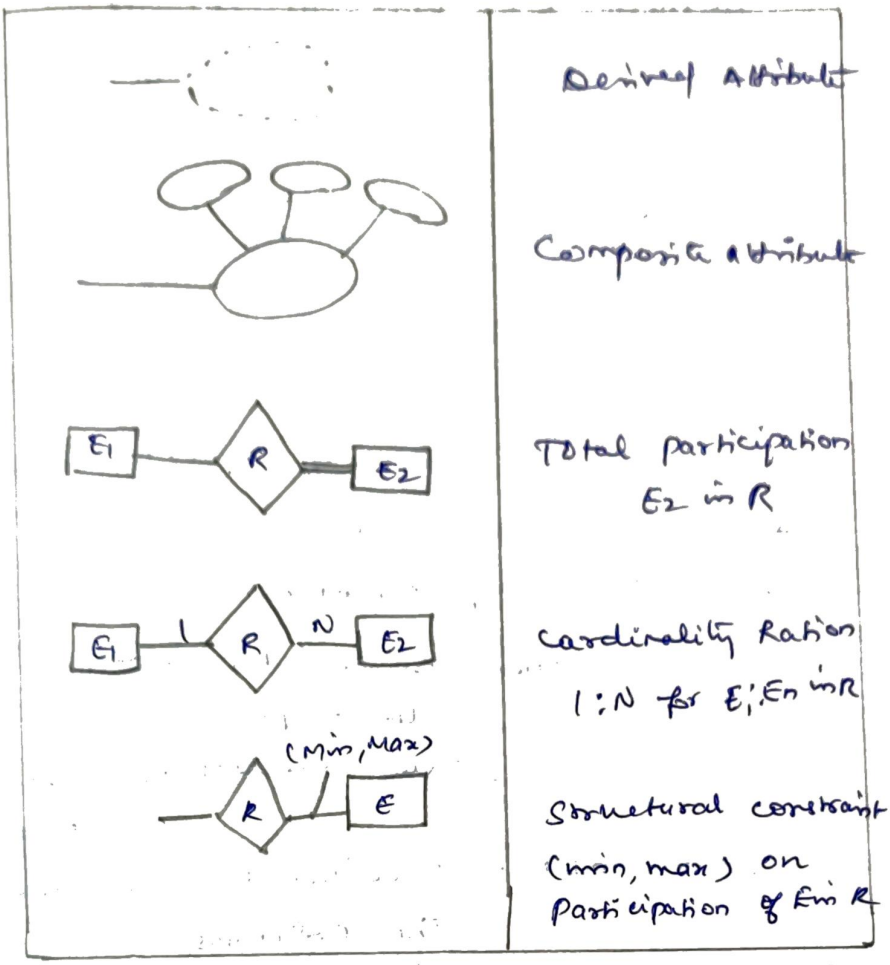
#) Composite Attribute for any entity :

- A composite attribute is the attribute, which also has attributes.

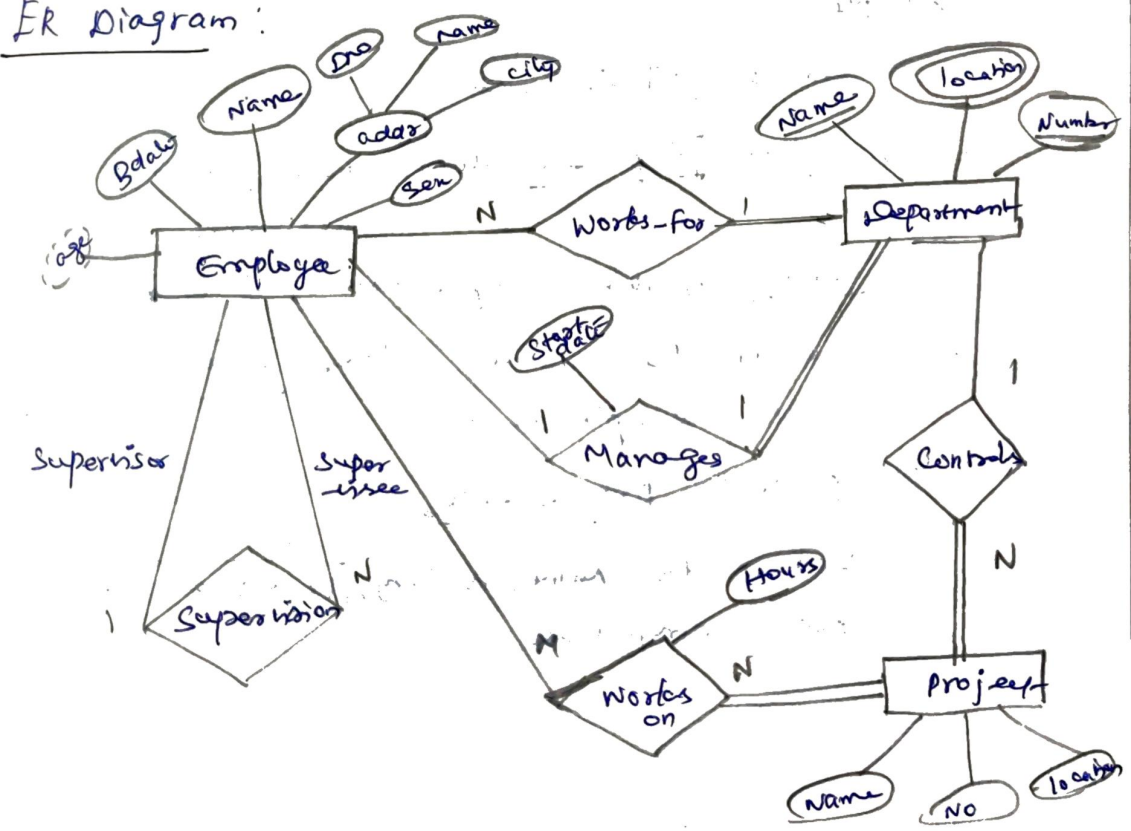


Notation for ER diagrams:

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key attribute
	Multivalued Attribute


Derived Attribute
Composite Attribute
E1
R
E2
Total participation E2 in R
E1
1
R
N
E2
Cardinality Ration 1:N for E1:En in R
(Min, Max)
R
E
Structural constraint (min, max) on Participation of E in R

ER Diagram:


Name
SSN
Bdate
Sex
Address
DNo
Name
City
Employee
N
Works-for
1
Department
Name
Location
Number
Start date
1
1
Manages
1
Controls
N
Supervisor
Super-visee
1
N
Supervision
M
Works on
N
Hours
Project
Name
NO
Location

Retrieving Data Using the SQL SELECT Statement

Basic SELECT statement:

Syntax:

```
Select *| {[Distinct] column| expression [alias],.}
from table;
```

In Syntax:

- Select – is a list of one or more columns
- * – Select all columns
- Distinct – suppress duplicate
- column|expression – selects the named column or the expression.
- alias – gives selected columns different headings
- From table – specifies the table containing the columns.

Example:

1) `Select * from departments;`
2) `Select depart_id, depart_name from departments;`
3) using Arithmetic operators: (+, −, *, /)

```
Select last_name, salary, salary+500
from employees;
```

4) Operator precedence:

```
Select last_name, salary, 12*(salary+100)
from employees;
```

5) Defining a Null Value

A Null is a value in unavailable, unassigned, unknown or inapplicable.

- A Null is not the same as a zero or blank space.

Eg: Select last_name, 12 * salary * comm_pct from employees;

6) Defining a Column Alias:

- Renames a column heading
- Is useful with calculations
- Immediately follows the column name.
- Requires double quotation marks if it contains spaces or special characters, or if it is case-sensitive

- Example:

Select last_name as Name, comm_pict as commision, salary * 12 "Annual salary" from employee;

7) Concatenation operator:

- Links columns or character strings to other columns
- Is represented by two vertical bars (||)
- Creates a resultant column that is a character expression

Select last_name || job_id as "A Employee" from employees;

8) Literal character strings

- A literal is a character, a number, or a date that is included in the select statement.
- Date and character literal values must be enclosed by single quotation marks.
- Each character string is output once for each row returned.

```
Select last_name || ' is a ' || job_id As
"Employee Details" from employee;
```

9) Alternative Quote (q) Operator:

- Specify your own quotation mark delimiter
- Choose any delimiter
- Increase readability and usability.

Eg:

```
Select dept_name || q'[, It's assigned
manager id : ]' || manager_id as
"Department and Manager' from employee;
```

(0) Duplicate Rows:

- Distinct → removes duplicate values.
- Select distinct depart_id from employees;

- Oracle SQL Developer, the visual tool for database development, simplies the following tasks:
  i) Browsing and managing database objects
  ii) Executing SQL statements and scripts
  iii) Editing and debugging PL/SQL statements
  iv) Creating reports

Sorting rows & Substitution variables & Pattern matching

1) Limiting the Rows that are selected:

- restrict the rows that are returned by using the WHERE clause.

```
Select *| {[Distinct] column|expression [alias],...} from table
Where condition(s);
```

- Eg:
  1) Select employee_id, last_name, job_id from employees where depart_id = 90;
  2) Select last_name, job_id from employees where last_name = 'whalen';

2) Comparison conditions:

Operator	Meaning
=	equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
<>	not equal to
Between ... and ...	Between two values
in	match any of a list of values
like	Match a character pattern
is Null	Is a null value

Eg: Select last_name, salary from employee where salary <= 3000;

*) Using between condition:

Select last_name, salary from employees
where salary between 2500 and 3500;
(lower limit — upper limit)

*) Using in condition:

select employee_id, last_name from employees
where manager_id in (100, 101, 102);

3) Using like condition (Pattern matching)

- Use the like condition to perform wildcard searches of valid search string values.
- Search conditions can contain either literal characters or numbers:
  % → denotes zero or many chacters.
  _ → denotes one character.

Eg:

Select first_name, last_name from employees
where first_name like 'S%';

Select last_name from employee
where last_name like '_o%';

4) Using NULL conditions:

Select last_name, manager_id from employees
where manage_id is Null;

5) Logical conditions.

- And - returns true if both component conditions are true
- Or - returns true if either component condition is true
- NOT - returns true if the condition is false

Eg:

Select employee_id, last_name, job_id from employees where salary > 10000 and job_id like '%MAN%';

Select employee_id, job_id from employee where job_id not in ('it prog', 'sr_clerk');

order by clause

- Sort retrieved rows with the order by clause
  - ASC : ascending order, default
  - Desc : descending order
- Eg:
  *) Select last_name, job_id, dept_id from employee order by hire_date;
  *) Select last_name, job_id from employee order by hire_date desc;
  *) Select last_name, depart_id, salary from employee order by depart_id, salary desc;

Substitution Variables

— Use substitution Variables to:

*) Temporarily store values with single-ampersand (&) and double-ampersand (&&) substitution.

— Use substitution Variables to supplement the following:

- Where conditions
- Order by clauses
- Column expressions
- Table names
- Entire select statements

— Eg:

*) Select employee_id, last_name, salary from employees where `employee_id = &emp_num;`

*) Select last_name, dept_id, salary * 12 from employee where `job_id = '&job_title';`

*) Select employee_id, last_name, `&&column_name` from employee `order by &column_name;`

*) Select employee_id, last_name, `&column_name` from employees where `&condition` order by `&order_column;`

Assignment Question:

ER Diagrams:

1. Construct an ER Diagram for a hospital with a set of patients and a set of medical doctors. A log of the various conducted tests is associated with each patient, construct the normalized relations from this ER Diagram

2. Construct an E-R Diagram for a car insurance company with a set of customers, each of who owns a number of cars. Each car has a number of accidents associated with it. Construct the normalized relation from this ER diagram.

Practice Questions: – select statements:

Table Name:

Nobel
year
Subject
winner

1) Write the SQL query, it display the nobel prize winner of the year 1950.
2) Show who won the 1962 prize for "literature"
3) Give the name of the winners since the year 2000 (including 2000)
4) Show the year, subject and name of winners for 1980 excluding chemistry and Medicine.
5) Show the Winners with first name John.