

Important Question for DBMS

1. Explain advantages of DBMS.
2. Function and responsibility of DBA.
3. Component of Data dictionary.
4. Draw and Explain E-R model Symbol.
5. Explain NOT NULL and CHECK constraint.
6. What is Data Model? Explain all Data Model.
7. Explain Three-tier Architecture.
8. Explain Group by having and order by with Example.
9. Explain DDL and DML commands with Example.
10. Explain Data Types of DBMS.
11. Explain Generalization and Specialization with example.
12. Give the Difference between primary key and unique key.
13. Explain foreign key with example.
14. Explain different types of join with example.
15. What is sub query? Explain with Example.
16. List set operators and Explain all with Example.
17. Explain types of keys with example.
18. Explain group function with example.
19. Explain logical operators with example.
20. List types of DBMS and Explain distributed and Centralized DBMS.
21. Advantages of Three-tier architecture.
22. Explain schema, sub-schema and instance.
23. Explain Data Independence
24. What is Data Dictionary? Explain its types.
25. Define Following
 - a) Data
 - b) Information
 - c) System Catalog
 - d) Data Warehouse
 - e) Entity
 - f) Attribute
 - g) Metadata
26. Explain relational algebra operation
 - a) Selection
 - b) Projection
 - c) Intersection
 - d) Cartesian product
 - e) Rename

1. Explain advantages of DBMS.

Ans.

1. **Minimal Data Redundancy.** Due to centralized database, it is possible to avoid unnecessary duplication of information. Ex. All information about bank customer can be kept centralized.
2. **Improved Data Inconsistency.** Data inconsistency occurs due to data redundancy. Due to centralized database it is possible to avoid unnecessary duplication of information.
3. **Efficient Data Access.** DBMS provides a variety of techniques to retrieve data.
4. **Improved Data Sharing.** In DBMS Data is maintained centrally, all authorized users and application programs can share this data easily.
5. **Improved Data Integrity.** Data integrity means that data contained in the database is both correct and consistent. DBMS software provides different way to implement such types of rules.
6. **Guaranteed Atomicity.** Any operation on database must be atomic. It is responsibility of the DBMS software ensure such kind of atomicity.
7. **Improved Concurrent Access.** Multiple users are allowed to access data simultaneously (concurrently). As Database is maintained centrally, data can be shared easily among multiple users.
8. **Improved Security.** DBMS software provides way to control the access to data for different user according to their requirements.

2. Function and services of DBA.

Ans.

1. **Defining conceptual schema and database creation (Schema Definition).** The DBA Defines the overall logical structure of database. According to this schema database will be developed to stored required data for system.
2. **Storage structure and access method definition.** DBA decides how the data is to be represented in the stored database process is called physical database design.
3. **Assisting Application Programmers.** The DBA provides assistance to application programmers to develop application programs.
4. **Granting authorization to users (Approving Data Access).** The DBA determines which user access to which part of the database. This is required to prevent unauthorized access of a database.
5. **Physical organization modification.** The DBA modifies the physical organization of the database to improve performance.
6. **Monitoring performance.** The DBA monitors performance of the system and making changes in physical or logical schema if required.

7. **Backup and Recovery.** Database is a valuable asset for any organization. It should not be lost or damaged. The DBA ensures this by taking proper backup of Database.

3. Component of Data dictionary.

Ans.

1. **Entities.** It is real physical object or event. The user is interested in keeping track of
OR Any item about which information is stored is called entity.

Example. Employee of company

2. **Attributes.** An attribute is a property or characteristic (field) of an entity.

Example. Employee's EMP_NO, EMP_SALARY etc.

3. **Entity set.** A collection of entities of same type are called entity set.

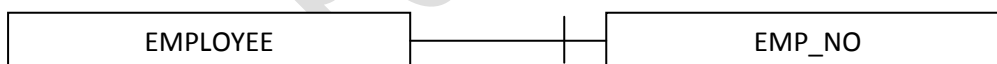
Example. Employee of all companies.

4. **Relationships.** The association or the ways that different entities relate to each other is called relationships.

Relationship could be of following type.

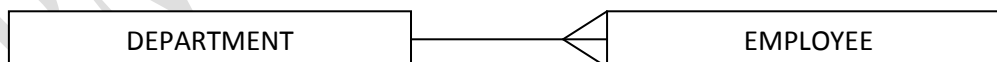
- **One to One (1:1) Relationship**

Example: One employee having one employee number.



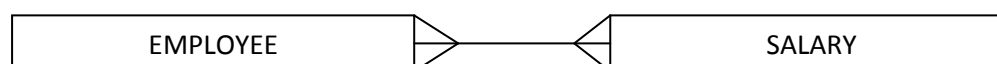
- **One to Many(1:m) Relationship**

Example: One Department having more than one employee OR One salary is given to many employees.



- **Many to Many (n:m) Relationship**

Example: Much salary is given to many employees.



5. **Key.** The data item for which a computer uses to identify a record in a database system is referred to as key. There are several keys available in a table. Two main keys are primary key and foreign key.

4. Explain Data Types of DBMS.

Ans.

✓ There are basic four data types available in SQL.

- 1) Numerical
- 2) Character
- 3) Date
- 4) Binary

Numerical Data Types

✓ Used to store zero, negative and positive numerical values.

✓ These values can be fixed-point or floating-point.

Sr No.	Data Type	Description
1	NUMBER(P,S)	Floating-point number. P=Precision, Precision can be up to 38 digits. S=scale, Ex. number(6,2)=1234.79
2	NUMBER(P)	Fixed-length number. Ex. Number(6)=123456
3	NUMBER	Floating-point number with a precision of 38 digits.

Character/String Types

Sr No.	Data Type	Description
1	CHAR(size)	Stores character string of fixed length. Default size is 1. Maximum length is 255 characters.
2	VARCHAR(size)/ VARCHAR2(size)/	Stores character strings of fixed length. More flexible than CHAR Maximum length is 2000 characters.
3	LONG	Stores large amount of character strings of variable length. Maximum length is up to 2 GB. Only one column per table can be defined as LONG.

Binary Data Types

Sr No.	Data Type	Description
1	RAW	Stores binary type data. Maximum length is up to 255 bytes.
2	LONG RAW	Stores large amount of binary type data. Maximum length is up to 2 GB.

DATE Data Types

- ✓ Used to store date and time.
- ✓ The information stored about date and time is century, year, month, day, hour, minute and second.
- ✓ The standard format is DD-MON-YY to store date.

5. Explain NOT NULL and CHECK constraint.

Ans.

1. NOT NULL

- ✓ A Null value indicates not applicable or not known.
- ✓ A Null value is different from zero or blank space.
- ✓ A Column define as not null does not contain null value. Such type of column become mandatory (compulsory) and does not contain empty value.
- ✓ **Syntax.** ColumnName Datatype (Size) NOT NULL
- ✓ **Example.** Create table Account(Ano number(3),Balance number(8) NOT NULL, Branch varchar2(20));
- ✓ Now if we insert null value to balance column then it will generate an error.

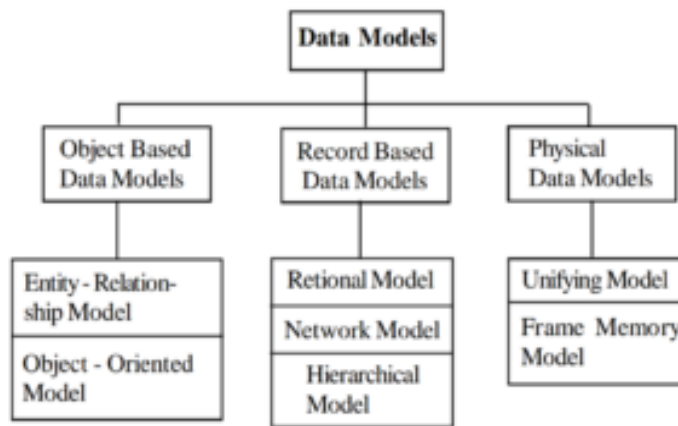
2. CHECK

- ✓ The check constrain is used to implement business rules. So it is also known as Business Constrain. For Ex. Business rules balance cannot be a negative.
- ✓ Once check constrain is implemented apply on column any insert or update operation on that table must follow these constrain.
- ✓ If any operation violate (break) constrain then it will be reject.
- ✓ **Syntax.** ColumnName Datatype (Size) CHECK(Condition)
- ✓ **Example.** Create table Account (Ano number (3), Balance number (8) CHECK (Balance>0), Branch varchar2 (20));
- ✓ A Condition must be some valid logical expression.
- ✓ On a violation of this oracle display error message " Check Constrain Violated".

6. What is Data Model? Explain all Data Model.

Ans.

Definition. A data model is a collection of conceptual tools for describing Data, Data Relationships, Data schema and Consistency constraints.



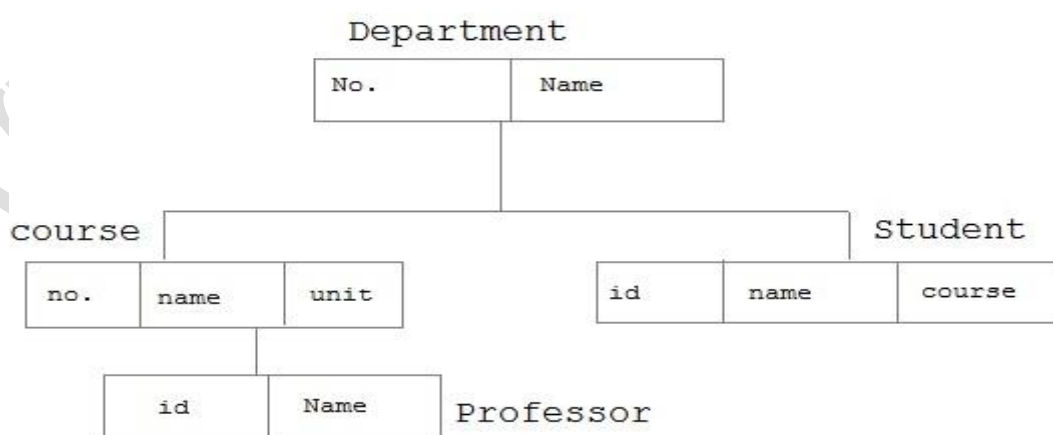
A. Record Based Data Models

The most widely used Record Based Data Models are

- Hierarchical Data Model
- Network Data Model
- Relational Data Model.

1) Hierarchical Data Model

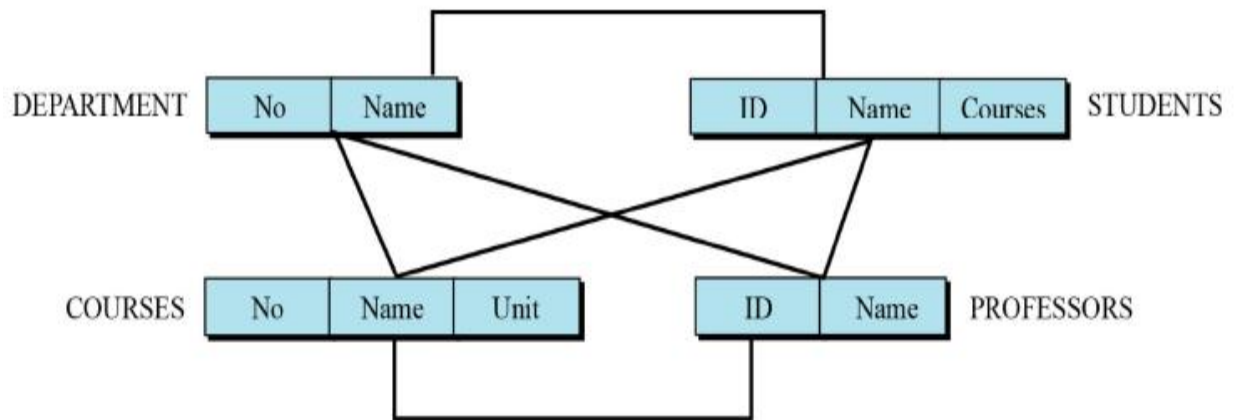
- ✓ In Hierarchical data model tree concept is used to represent data and relationship among this data.
- ✓ The nodes of tree are connected with pointer or link.
- ✓ In tree structure each child records can have only one parent record, and each parent record have zero or more than one child so one-to-one or one-to-many relationship is possible.
- ✓ The records are represented by rectangular box and relationship between these records is represented by arcs (pointer).
- ✓ This type of structure is also known as tree structure diagram.



2) Network Data 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.

- ✓ Network Database Model is same like Hierarchical Model, but the only difference is that it allows a record to have more than one parent.
- ✓ It replaces the hierarchical tree with a graph.
- ✓ It represents the data as record types and many-to-many relationship.
- ✓ This model is easy to design and understand.
- ✓ This was the most widely used database model, before Relational Model was introduced.



3) Relational Data Model.

- ✓ It represents data as relations or tables.
- ✓ Relational database simplifies the database structure by making use of tables and columns.
- ✓ Relational data model describe a database as a collection of tables to represent data and relationship among those data.
- ✓ Each table is called relation.
- ✓ Each relation has number of columns that is attributes of table.
- ✓ Each relation has number of unlimited rows called tuples.
- ✓ The relationship among table (relation) can be represented by primary key- foreign key.

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

7. Explain Group by having and order by with Example.

Ans.

- ✓ The SQL GROUP BY Clause is used along with the group functions to retrieve data grouped according to one or more columns.

Syntax.

Select column1, column2..... columnN, Aggregate Function (argument) From
TableName Group By column1, column2..... columnN;

Example.

Id	Name	Dept	Age	Salary	Location
100	Ramesh	Electrical	24	25000	Bangalore
101	Hrithik	Electronics	28	35000	Bangalore
102	Harsha	Aeronautics	28	35000	Mysore
103	Soumya	Electronics	22	20000	Bangalore
104	Priya	InfoTech	25	30000	Mangalore

SELECT dept, SUM (salary) FROM employee GROUP BY dept;

The output would be like:

Dept	Salary
Electrical	25000
Electronics	55000
Aeronautics	35000
InfoTech	30000

1) SQL ORDER BY

- ✓ The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- ✓ The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

Syntax.

SELECT column1, column2....FROM table_name
ORDER BY column1, column2 ... ASC|DESC;

Id	Name	Dept	Age	Salary	Location
----	------	------	-----	--------	----------

100	Ramesh	Electrical	24	25000	Bangalore
101	Hrithik	Electronics	28	35000	Pune
103	Soumya	Electronics	22	20000	Bombay
104	Priya	InfoTech	25	30000	Mangalore

Example.

SELECT name, salary FROM employee ORDER BY salary;

The output would be like

Name	Salary
Soumya	20000
Ramesh	25000
Priya	30000
Hrithik	35000

- ✓ By default, the ORDER BY Clause sorts data in ascending order.
- ✓ If you want to sort the data in descending order, you must explicitly specify it as shown below.

SELECT name, salary FROM employee ORDER BY name, salary DESC;

2) SQL HAVING Clause

- ✓ Having clause is used to filter data based on the group functions.
- ✓ This is similar to WHERE condition but is used with group functions.
- ✓ Group functions cannot be used in WHERE Clause but can be used in HAVING clause.
- ✓ The HAVING clause must follow the GROUP BY clause in a query and must also precedes the ORDER BY clause if used.

Syntax.

SELECT column1, column2 FROM TableName WHERE [conditions] GROUP BY column1, column2 HAVING [conditions];

Id	Name	Dept	Age	Salary	Location
100	Ramesh	Electrical	24	25000	Bangalore
101	Hrithik	Electronics	28	35000	Pune
103	Soumya	Electronics	22	20000	Bombay

104	Priya	InfoTech	25	30000	Mangalore
-----	-------	----------	----	-------	-----------

SELECT dept, SUM (salary) FROM employee GROUP BY dept HAVING SUM (salary) > 25000;

Dept	salary
Electronics	55000
Aeronautics	35000
InfoTech	30000

8. Explain DDL and DML commands with Example.

Ans:

DDL (Data-Definition Language)

- ✓ It is set of SQL commands used to create, modify and delete database objects like table.
- ✓ Provides commands:
 1. CREATE
 2. ALTER
 3. TRUNCATE
 4. DROP
 5. RENAME

1) CREATE

- ✓ Create command used to create a table in database.

Syntax:

```
CREATE TABLE TableName
(Column1 datatype (size), ... ColumnN datatype (size));
```

Example:

Create table customer (cno varchar2 (6), name varchar2 (20), city varchar2 (15));

2) ALTER

- ✓ Used to modify structures of a table.
- ✓ Used to either add ,modify, or drop columns in a table

Syntax:

Adding New Columns

Alter table TableName **Add** (NewColumnName Datatype (size), NewColumnName Datatype (size)...);

Modifying Existing Columns

Alter table TableName **Modify** (ColumnName NewDatatype (NewSize));

Dropping (deleting) Existing Columns

Alter table TableName **Drop** column ColumnName;

Example:

- 1) Alter table student **ADD** (Gender varchar2 (10));
- 2) Alter table student **modify** (sname varchar2 (15));
- 3) alter table student **drop** column dept;

3) TRUNCATE TABLE

- ✓ TRUNCATE TABLE used to delete all data from a table
- ✓ Logically, this is same to DELETE statement that deletes all rows
- ✓ TRUNCATE command is faster than DELETE command
- ✓ The number of deleted rows are not returned

Syntax:

TRUNCATE TABLE TableName;

Example:

TRUNCATE TABLE client_master;

4) DROP TABLE

- ✓ DROP command is used to delete or destroy table from a database

Syntax:

DROP TABLE TableName;

Example:

DROP TABLE client_master;

5) RENAME TABLE

- ✓ RENAME command is used to rename the existing table Name.

Syntax:

Rename TableName to NewTableName;

Example:

Rename Student to StudentMaster;

DML Commands - (Data-Manipulation language)

- ✓ DML commands are generally used for manipulating data stored in tables.
- ✓ So DML commands are work on data stored in tables rather than table definition.
- ✓ It include following commands

1. INSERT
2. UPDATE
3. DELETE

1) INSERT

- ✓ Used to insert data into a table or create a new row in table.

Syntax:

Insert into TableName (columnname1, columnname2) values (value1, value2);

Example:

Insert into client_master (clientno, name) values ('C01','Preeti');

2) UPDATE

- ✓ The UPDATE command is used to change or modify data of a table
- ✓ Update command used to update Selected rows from table or All the rows from table

Syntax:

Update TableName Set columnname1 = value1, Columnname2= value2 where Condition;

Example:

Update client_master Set city='Bombay' where clientno='C01';

3) DELETE

- ✓ Delete command used to delete data or rows from a table
- ✓ delete command used to delete Selected rows from table or All the rows from table

Syntax:

Delete from TableName Where Condition;

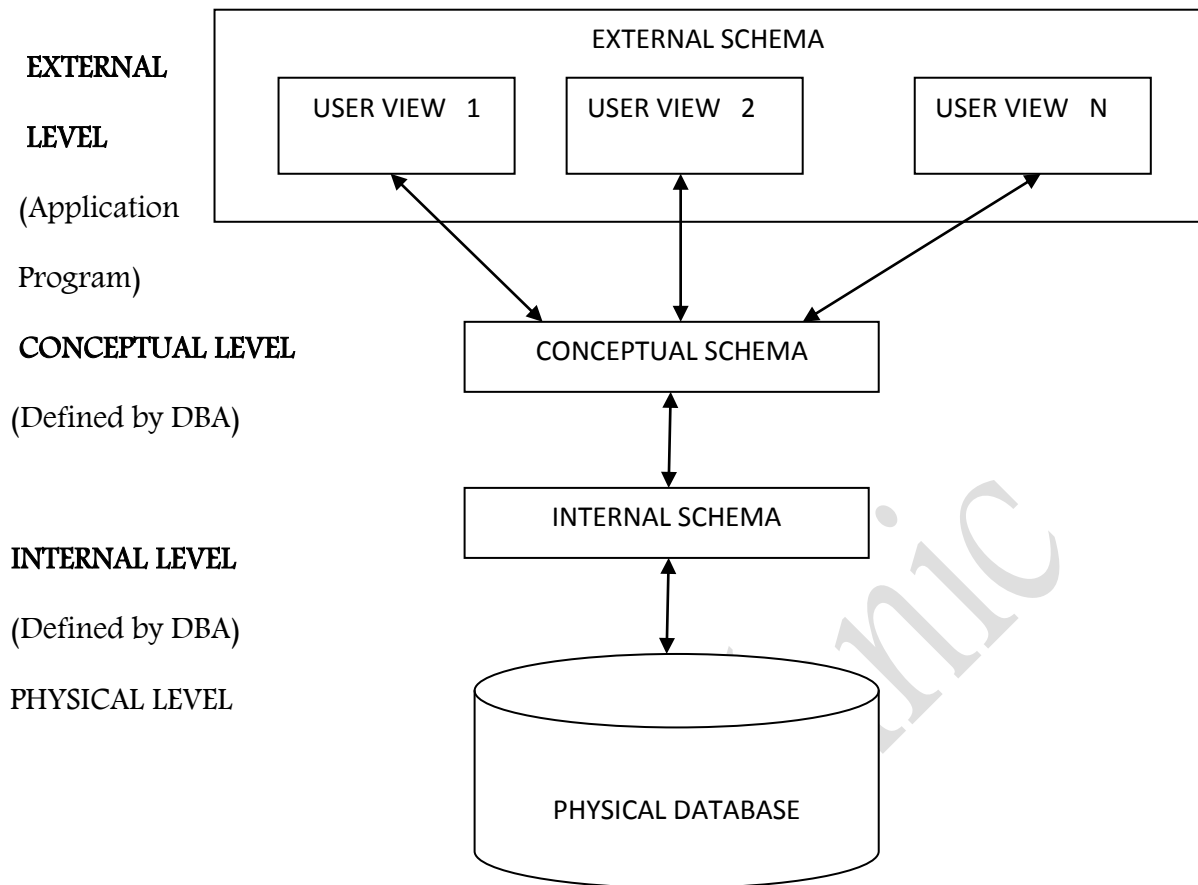
Example:

Delete from client_master Where client_no='C01';

9. Explain Three-tier Architecture.

Ans:

In 1975, **ANSI-SPARC** produced three-tier architecture with a system catalog. The architecture of most commercial DBMS available market is based on ANSI-SPARC model.



Internal (Physical) Level.

- ✓ It is the lowest level of data abstraction.
- ✓ It is the physical representation of the database on the computer and this view is found at the lowest level of abstraction of database.
- ✓ It is also known as physical level.

Conceptual (Logical) Level.

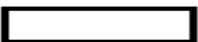

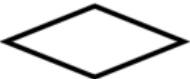

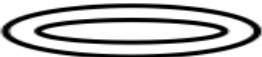






- ✓ It is middle level or next level in the 3-tier architecture.
- ✓ It contains the logical structure of entire database as seen by DBA.
- ✓ It describes what data are stored in the database and what relationship exists among those data.
- ✓ It is also known as logical level.

External (View) Level.

- ✓ It is user's view of database.
- ✓ This is the highest level of data abstraction.
- ✓ It is known as a view level.
- ✓ Each user has a view of the "real world" represented in a form that is familiar for that user the eternal view Example: Customer information for transaction.

10. Draw and Explain E-R model Symbol.

Ans.

	Represents Entity
	Represents Attribute
	Represents Relationship
	Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)
	Represents Multivalued Attributes
	Represents Derived Attributes
	Represents Total Participation of Entity
	Represents Weak Entity
	Represents Weak Relationships
	Represents Composite Attributes
	Represents Key Attributes / Single Valued Attributes

11. Explain Generalization and Specialization with example.

Ans.

Specialization

- ✓ "Specialization is the process of defining sub-classes of a super-class.
- ✓ Specialization identifies the sub-sets of an entity set.
- ✓ Specialization is a top-down process to define super-class/sub-class relationships.
- ✓ **Example.** Consider the Employee entity set having attributes eid, name, address, bdate, contact_no, salary, work_hrs, rate_per_hr, etc.
- ✓ Employees can be either of two types
- ✓ **Permanent** employees having fixed salary
- ✓ **Part-time** employees having variable salary, based on number of hours worked and rate per hour.

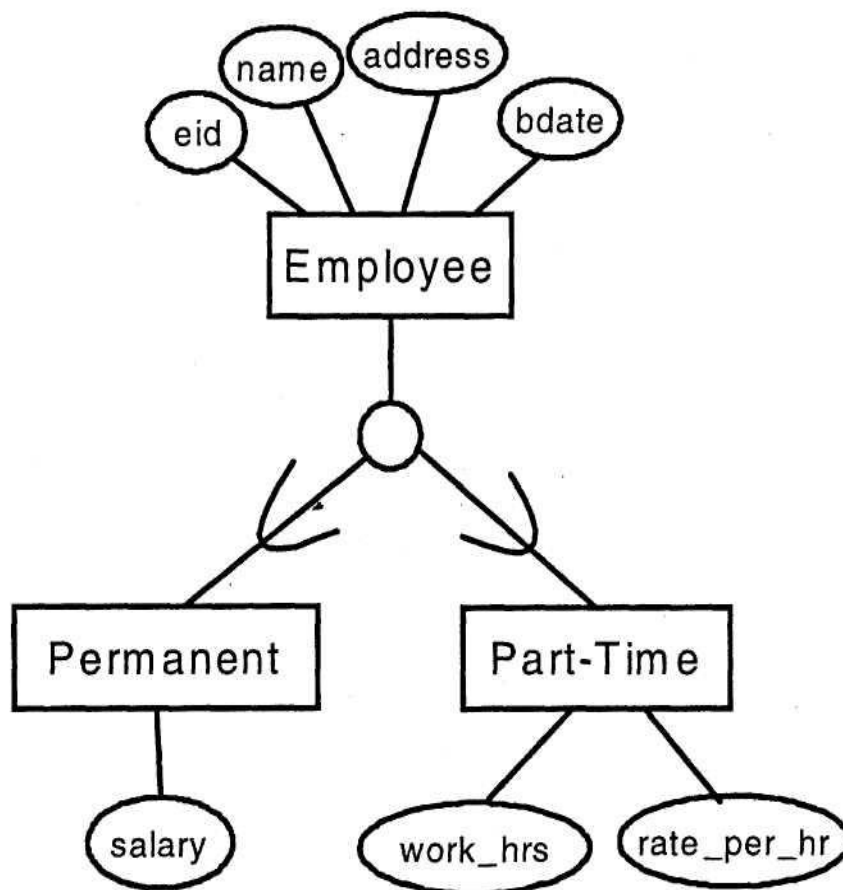


Figure 5.5 : Employee super-class/sub-class relationship

Generalization

- ✓ "Generalization is the process of defining a super-class from sub-classes based on some common characteristics."
- ✓ Generalization identifies a super-class from different entity sets.
- ✓ Some attributes are common to given entity sets.
- ✓ Generalization is a bottom-up process to define super-class/sub-class relationships.
- ✓ **Example.** consider the two entity sets Car and Truck

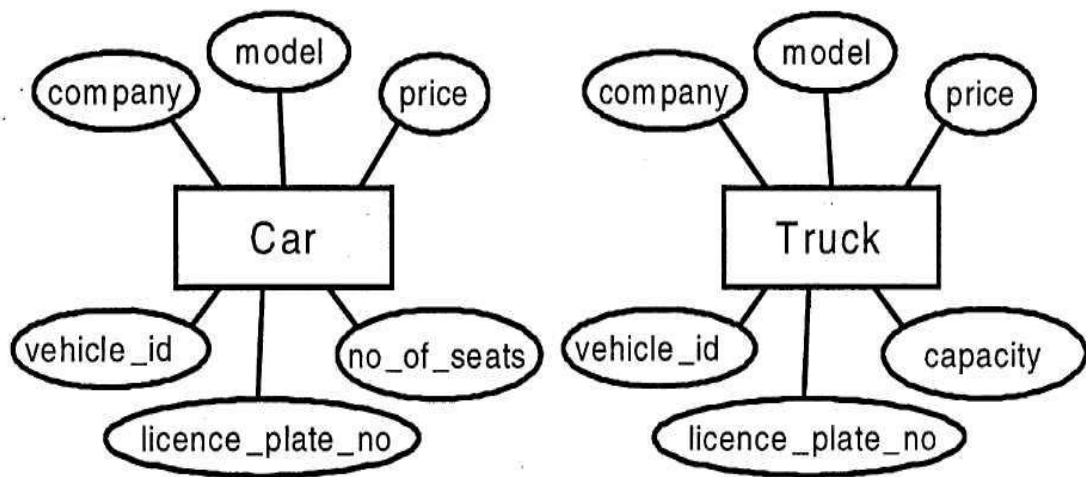


Figure 5.6 : A Car and A Truck entity set

- ✓ They have many attributes in common.
- ✓ **Example.** vehicle_id, licence_plat_no, company, model, and price are common attributes to both of these entity sets
- ✓ Vehicle can be defined as a super-class for two sub-classes Car and Truck.

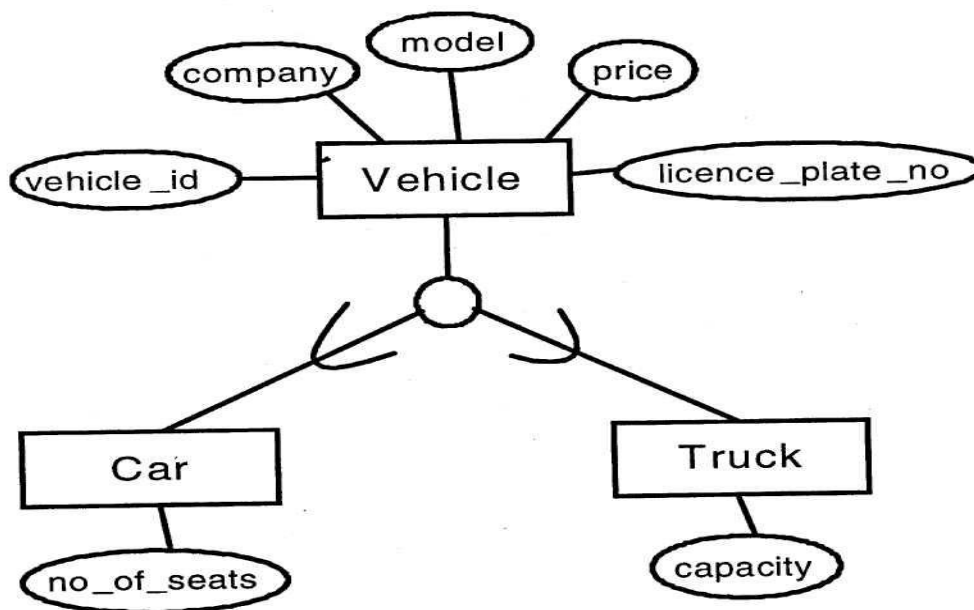


Figure 5.7 : Generalization to Vehicle super-class

12. Give the Difference between primary key and unique key.

Ans.

Primary Key	Unique Key
A Column defines as primary key does not contain duplicate value as well as null value.	A Column defines as unique key cannot have duplicate value but it have null value.

Syntax. ColumnName Datatype (Size) Primary Key.	Syntax. ColumnName Datatype (Size) Unique Key.
Example. Create table Account(Ano number(3) Primary Key, Balance number(8), Branch varchar2(20));	Example. Create table Account(Ano number(3) Unique Key, Balance number(8), Branch varchar2(20));
Primary Key does not allow null value in column.	Unique Key allows null value as well as duplicate null value in a column.
A table cannot have more than one primary key.	A Table can have more than one column define as unique key.

13. Explain foreign key with example.

Ans.

- ✓ A foreign key constrain is also called referential integrity constrain which is define between two table.
- ✓ A Foreign key is a set of one or more column whose value is derive from the primary key or unique key of other table.
- ✓ A Table in which Foreign key is defined is called Primary Table or Master Table or Parent Table.
- ✓ **Syntax.**
ColumnName Datatype (Size) foreign key (ColumnName) References TableName (ColumnName) [On delete/Update Cascade]
- ✓ **Example.**
Create table student (RollNo number (3) Primary key, Name varchar2 (15), Sid number (3) foreign key (Sid) reference Subject (Sid)) on delete cascade;
On Delete Cascade
- ✓ With this option, when any record from parent table is deleted, all the corresponding records from child table are also deleted automatically.

14. Explain different types of join with example.

Ans.

- ✓ In DBMS it is possible to retrieve information from multiple table using join.
- ✓ Join combines two or more tables.
- ✓ Join combines columns from different tables.
- ✓ There are different types of joins available in a SQL.
 1. Cross Join (Cartesian product or simple join).
 2. Inner Join (Equi join).
 3. Self-Join

4. Outer Join

INNER JOIN

- ✓ The inner join is the most common used join in all joins.
- ✓ The inner join combines only that record which contains common value in both relations.
- ✓ Inner join is totally reverse from cross join and it provides consistent record as an output.
- ✓ **Syntax:** select column1,column2,...,columnN from table1,table2 where table1.column1 op table2.column1;
- ✓ In inner join, if both the columns are compared for equality (=op) then it is referred as 'Equi join'.
- ✓ If both the columns are compared for non-equality then join operation is referred as non-equi join.

Example.

Relation: Student

ID	NAME	DEPT
1	Anisha	CE
2	Tanisha	EE

Relation: Hostel

NAME	HOSTELNAME	ROOMNO
Kanisha	ABC	14
Anisha	XYZ	12
Tanisha	PQR	13

Example.

Select id, Student.Name, dept, RoomNo, HostelName form Student,Hostel where Student .Name=Hostel.Name;

Output.

ID	NAME	DEPT	HOSTELNAME	ROOMNO
1	Anisha	CE	XYZ	12
2	Tanisha	EE	PQR	13

OUTER JOIN:

- ✓ The outer join operation is an extension of inner join operation.
- ✓ In inner join operation, it may be possible to lose information.
- ✓ The outer join with such kind of missing information.
- ✓ Outer join operation can be divided into three different forms:
 1. Left Outer Join
 2. Right Outer Join
 3. Full Outer Join

15. What is sub query? Explain with Example.

Ans.

- ✓ A Subquery is a SQL statement that contains one statement within another statement.
- ✓ A Subquery is a SQL query within in a Query.
- ✓ Subquery must be enclosed within parentheses.

Syntax.

SELECT column_name [, column_name] FROM table1 [, table2] WHERE column_name
OPERATOR (SELECT column_name [, column_name] FROM table1 [, table2] [WHERE])

Example.

Id	Name	Dept	Age	Salary	Location
100	Ramesh	Electrical	24	2500	Bangalore
101	Hrithik	Electronics	28	6500	Bangalore
102	Harsha	Aeronautics	28	5500	Mysore
103	Soumya	Electronics	22	2000	Bangalore
104	Priya	InfoTech	25	3000	Mangalore

Example.

SELECT * FROM CUSTOMERS WHERE ID IN (SELECT ID FROM CUSTOMERS WHERE
SALARY > 4500);

Output.

Id	Name	Dept	Age	Salary	Location
101	Hrithik	Electronics	28	6500	Bangalore
102	Harsha	Aeronautics	28	5500	Mysore

16. List set operators and Explain all with Example.

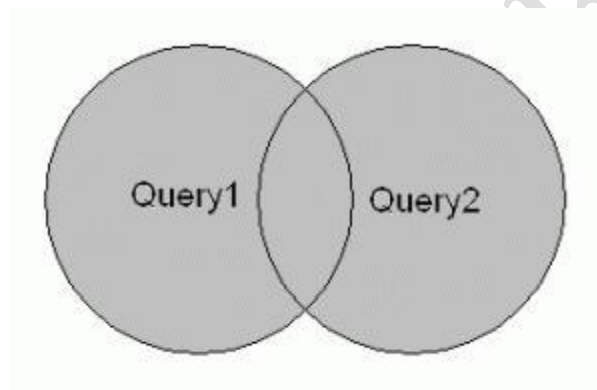
Ans.

There are four types of Set operators.

- 1) Union
- 2) Union all
- 3) Intersect
- 4) Minus

1) **UNION Clause**

- ✓ The UNION clause merges the output of query1 and query2.
- ✓ The output will be as a single set of rows and columns.
- ✓ The output will contain all the records from query1 and query2 according to following.



- ✓ The number of columns, name of each column and their data types in both relations must be same.
- ✓ **Syntax.** `SELECT * FROM table1 UNION SELECT * FROM table2;`
- ✓ **Example.** `SELECT * FROM Emp UNION SELECT * FROM Cust;`

Relation: Emp

ID	Name
1	Manisha
2	Tanisha
3	Kavisha

Relation: Cust

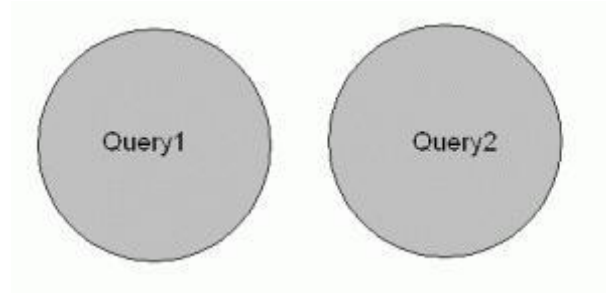
ID	Name
1	Manisha
2	Tanisha
4	Disha

Output:

ID	Name
1	Manisha
2	Tanisha
3	Kavisha
4	Disha

2) UNION ALL Clause

- ✓ The UNION ALL clause merges the output of query1 and query2.
- ✓ The output will be as a single set of rows and columns.
- ✓ The output will contain all the records from query1 and query2 according to following.



- ✓ The number of columns, name of each column and their data types in both relations must be same.

Syntax. SELECT * FROM table1 UNION ALL SELECT * FROM table2;

Example. SELECT * FROM EMP UNION ALL SELECT * FROM Cust;

Relation: EMP

ID	Name
1	Manisha
2	Tanisha
3	Kavisha

Relation: Cust

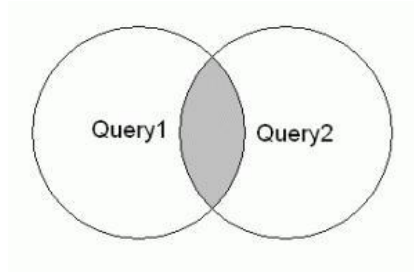
ID	Name
1	Manisha
2	Tanisha
4	Disha

Output.

ID	Name
1	Manisha
2	Tanisha
3	Kavisha
1	Manisha
2	Tanisha
4	Disha

3) INTERSECT Clause

- ✓ The Intersect clause merges the output of query1 and query2.
- ✓ The output will be as a single set of rows and columns.
- ✓ The output will contain those records which are common to query1 and query2 according to following.



- ✓ The number of columns, name of each column and their data types in both relations must be same.

Output: A single set of Records which is common in both queries.

Syntax:

`SELECT * FROM table1 INTERSECT SELECT * FROM table2;`

- ✓ **Example:** `SELECT * FROM Emp INTERSECT SELECT * FROM Cust;`

Relation: Emp

ID	Name
1	Manisha
2	Tanisha
3	Kavisha

Relation: Cust

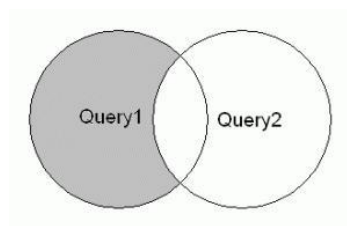
ID	Name
1	Manisha
2	Tanisha
4	Disha

Output:

ID	Name
1	Manisha
2	Tanisha

4) MINUS Clause

- ✓ The MINUS clause merges the output of query1 and query2.
- ✓ The output will be as a single set of rows and columns.
- ✓ The output will contain those records which are present in query1 but not present in query2 according to following.
- ✓ The number of columns, name of each column and their data types in both relations must be same.



Syntax: `SELECT * FROM table1 MINUS SELECT * FROM table2;`

Example. SELECT * FROM EMP MINUS SELECT * FROM Cust;

Relation: Emp

ID	Name
1	Manisha
2	Tanisha
3	Kavisha

Relation: Cust

ID	Name
1	Manisha
2	Tanisha
4	Disha

Output.

ID	Name
3	Kavisha

17. Explain types of keys with example.

Ans. Types of keys are.

- 1) Super Key
- 2) Candidate Key
- 3) Primary Key
- 4) Alternate Key
- 5) Foreign Key

1) Super Key

- ✓ "A super key is a set of one or more attributes that allows identifying each tuple uniquely in a relation."
- ✓ **Example.** Relation student has Sid, sname, contact_no.
- ✓ Here, Sid, {Sid, sname} and contact_no is a super key.

2) Candidate Key

- ✓ Candidate key is a subset of super key.
- ✓ "If a relation contains more than one relation keys, then, they each are called candidate key."
- ✓ A candidate key is a least combination of attributes that uniquely identify each record in a relation.
- ✓ **Example.** Relation student has Sid, sname, contact_no.
- ✓ Here, Sid and contact_no is candidate key.

3) Primary Key

- ✓ "A primary key is a candidate key that is chosen by the database designer to identify tuples uniquely in a relation."
- ✓ A primary key is a candidate key that is most appropriate to become main key for any relation.
- ✓ **Example.** Relation student has Sid, sname, contact_no.
- ✓ Here, Sid is a primary key.

4) Alternate Key

- ✓ "An alternate key is a candidate key that is not chosen by the database designer".
- ✓ Candidate key which are not selected as a primary key are known as alternate key or secondary key.
- ✓ **Example:** in above relation table Sid and phone is candidate key.
- ✓ Sid is a primary key so phone is an alternate key.

5) Foreign Key

- ✓ "A foreign key is a set of one or more attributes whose values are derived from the key attribute of another relation."
- ✓ This key is used to join two more tables.
- ✓ **Example:** Account: ano, balance, bname
Branch: bname, baddress
- ✓ Here, bname is foreign key for Account table and primary key for Branch table.

18. Explain group function with example.

Ans.

1) AVG()

- ✓ It returns average value of the specified column, ignoring NULL values.
Syntax: AVG (Column name)
Example: select AVG (sell_price) from Product_Master;
Output: 2012.345

2) MIN()

- ✓ It returns the minimum value in the specified column.
Syntax: MIN (column name)
Example: select MIN (sell_price) from Product_Master;
Output: 250

3) MAX()

- ✓ It returns the maximum value in the specified column.
Syntax: MAX (column name)
Example: select MAX (sell_price) from Product_Master;
Output: 1500

4) SUM()

- ✓ It returns the sum of values of the specified column.
Syntax: SUM (column name)
Example: select SUM (salary) from salesman_master;
Output: 65000

5) COUNT()

- ✓ It returns the number of rows in specified column or the total number of rows in the table.

Syntax: COUNT (column name)

Example: select COUNT (salesman_no) from salesman_master;

Output: 15

Syntax: COUNT (*)

Example: select COUNT (*) from salesman_master;

Output: 50

19. Explain logical operators with example.

Ans:

There are three types of Logical Operators

- 1) The AND Operator
- 2) The OR Operator
- 3) The NOT Operator

1) The AND Operator

- ✓ The AND Operator is used to combine two or more conditions in WHERE Clause.
- ✓ It requires all the conditions to be true to consider entire clause true.
- ✓ **Example.**

```
SQL>select * from employee;
```

ENAME	SALARY	CITY
-----	-----	-----
Preet	20000	Jamnagar
Parimal	15000	Rajkot
Khanjan	12000	Surat
Mehul	8000	Ahmedabad

```
SQL>select * from employee where city = 'jamnagar' AND salary > 15000;
```

ENAME	SALARY	CITY
-----	-----	-----
Preet	20000	Jamnagar

2) The OR Operator

- ✓ The OP operators is also used to combine two or more conditions in WHERE and HAVING clauses.
- ✓ It requires any one condition to be true to consider entire clause true.

Example.

SQL>select * from employee where city = 'Rajkot' OR city = 'Surat';

ENAME	SALARY	CITY
-----	-----	-----
Parimal	15000	Rajkot
Khanjan	12000	Surat

3) The NOT Operator

- ✓ The NOT operator is used to negate the result of any condition or group of conditions.

Example.

SQL>select * from employee where not (city = 'Ahmedabad');

ENAME	SALARY	CITY
-----	-----	-----
Preet	20000	Jamnagar
Parimal	15000	Rajkot
Khanjan	12000	Surat

20. List types of DBMS and Explain distributed and Centralized DBMS.

Ans.

DBMS can also be classified according to number of users, database site locations, expected type and extent of use.

(a) On the basis of no of users.

1. Single user system
2. Multi user system

(b) On the basis of site location.

1. Centralized DBMS
2. Parallel DBMS
3. Distributed DBMS
4. Client/Server DBMS

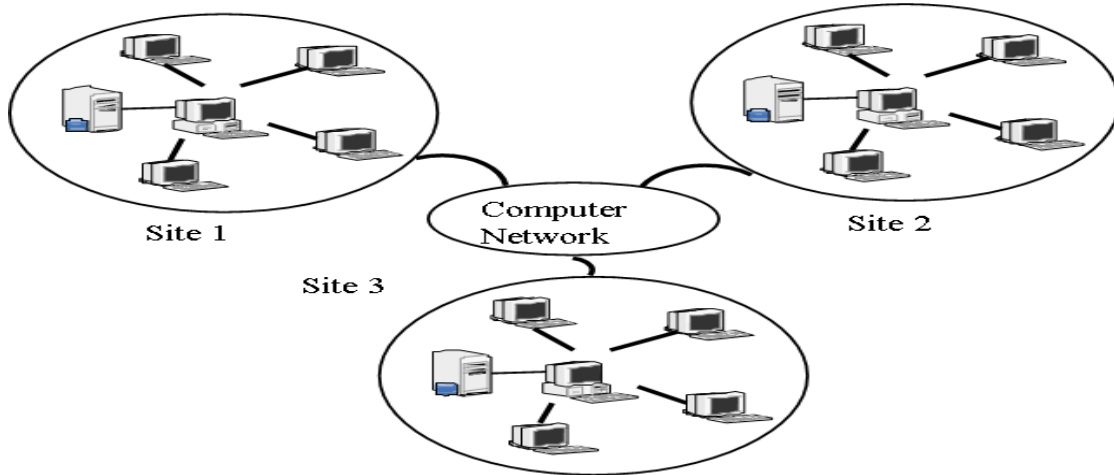
(c) On the basis of type and extent of use.

1. Transactional or Production DBMS
2. Decision Support DBMS
3. Data Warehouse

DISTRIBUTED DATABASE SYSTEM.

- ✓ In this system multiple computers are used.
- ✓ Enable users to access data from remote system.
- ✓ Managed by variety of different DBMS software's running on different computer machines supported by different of operating system.

- ✓ In distributed database system, one application can operate at different location.
- ✓ Each machine can have data and application of its own.
- ✓ Each machine will act as a server for some users and a client for others.



Advantages:

- ✓ Provide greater efficiency and better performance.
- ✓ Response time and Throughput is high.
- ✓ A single database (on server) can be shared across several distinct client (application) systems.

Disadvantages:

- ✓ Recovery from failure is more complex.

CENTRALISED DATABASE SYSTEM:

- ✓ It consists of a single database server and multiple users connected to the server.
- ✓ It is physically located at single place.
- ✓ Multiple users can access this database from local machines or from remote machine.
- ✓ Management of system and its data are controlled centrally.

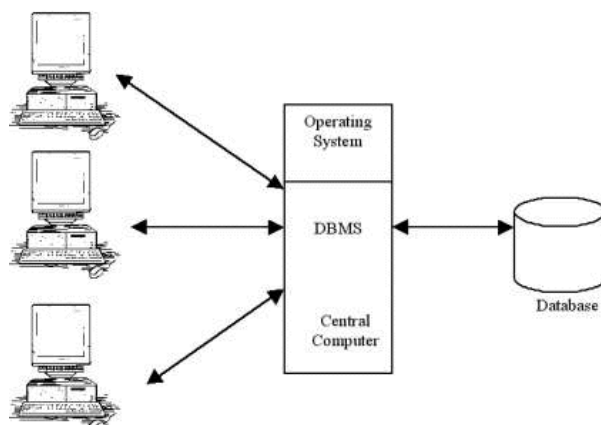


FIGURE: CENTRALISED DATABASE SYSTEM

Advantages:

- ✓ Functions like update, backup, query, control access are easier to perform.

- ✓ It provides security in terms of authorization.
- ✓ **Example:** small enterprise (Personal Computer) , large enterprise (Mainframe)

Disadvantages:

- ✓ When central site goes down then everyone (user) is blocked until system comes back.
- ✓ Communication costs from terminals to central site can be expensive.

21. Advantages of Three-tier architecture.

Ans.

Main objective is to isolate each user's view of the database from way the database is physically stored / represented.

1. Each user is able to access same data but have different customized view.
Each user can change the way to view the data and that change does not affect other user's same database.
2. User's interaction with database is independent of physical data storage
3. DBA can change database storage structure without affecting user's view.
4. DBA is able to change conceptual structure of database without affecting all users.

22. Explain schema, sub-schema and instance.

Ans.

Schema.

Definition: The overall logical design of the database is known as Schema.

- ✓ Schema includes Table Names, Column names, data-types and size of columns, various constraints at logical level.
- ✓ Schema changes infrequently.
- ✓ Cause of change is insertion/deletion of tables or columns.
- ✓ Example of schema is variable (Column Name) declaration like Rollno of student.

There are three Type of schema.

- 1) **Internal schema:** describe physical storage structures and access paths.
- 2) **Conceptual schema:** describe structure and constraints for the database.
- 3) **External schema:** describe different views of the database at view level.

Instances:

Definition: The collection of information stored in the database at a particular moment is known as instance.

- ✓ Instance defines actual data or information stored in tables in form of different records/rows.
- ✓ Instance changes frequently.

- ✓ Insert, delete or update operation make changes on database.
- ✓ **Example.** value of the variable (Column Name) like 1 Rollno of student.

Sub-Schema.

Definition. A subschema is a subset of the schema and inherits the same property that a schema has.

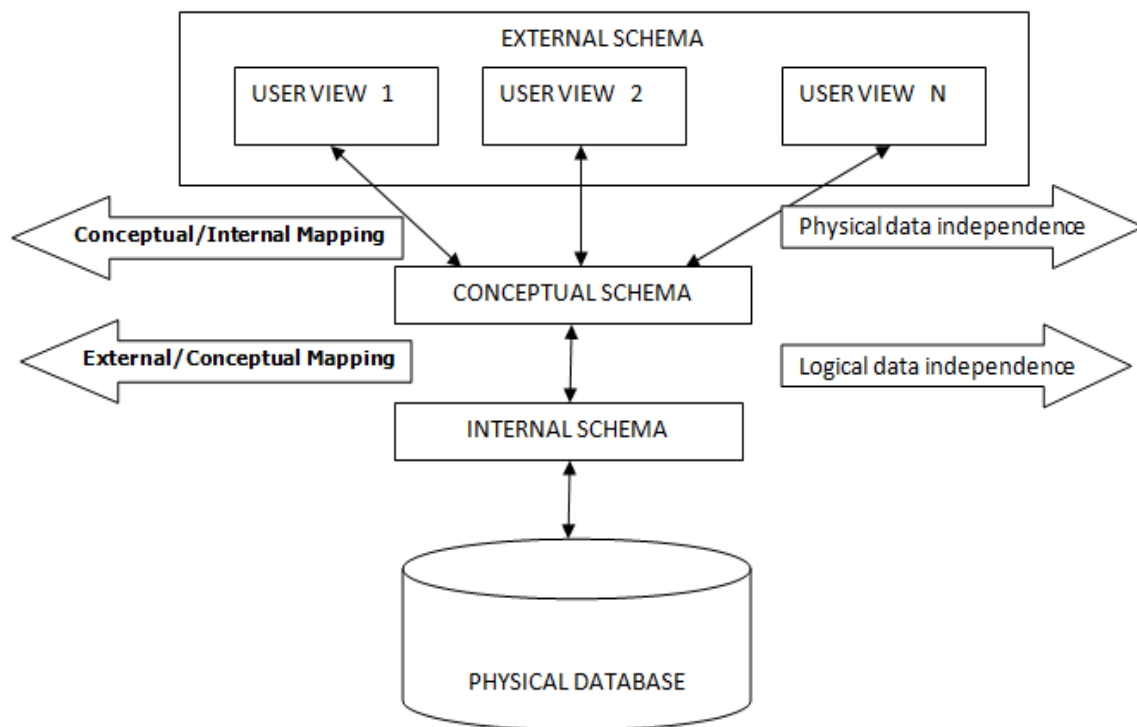
- ✓ Subschema refers to an application programmer's (user's) view of the data item types and record types, which he or she uses.

23. Explain Data Independence

Ans.

Definition. It is the ability to change a schema definition at one level without affecting a schema definition in the next higher level.

- ✓ Application programs do not depend on any one particular physical representation or access technique.



- ✓ There are Two types of Data Independence:

1. Physical data independence

- ✓ Physical data independence is the ability to modify the physical (internal) schema without requiring any change in applications programs.
- ✓ Physical data independence separates conceptual level from the internal level.
- ✓ It is easy to achieve physical data independence.

2. Logical data independence

- ✓ Logical data independence is the ability to modify the logical (conceptual) schema without requiring any change in applications programs.
- ✓ Logical data independence separates external level from the conceptual level.
- ✓ It is difficult to achieve logical data independence.

24. What is Data Dictionary? Explain its types.

Ans.

Definition.

- ✓ Data dictionary (Information Repository) is a mini database management system that manages metadata.
- ✓ General structure of data dictionary given below.
- ✓ A useful data dictionary stores and manages following types of information:
 1. Descriptions of schema of database.
 2. Detailed information on physical database design.
 3. Description of database users their responsibilities and their access rights.

TYPES OF DATA DICTIONARY.

1. **Active data dictionary.** It is managed automatically by the database management system.
2. **Passive data dictionary.** It is also called non-integrated data dictionary and it is used only for documentation purpose. It is simply a self-contained application and a set of files is used for documenting.

25. Define Following

- | | |
|-------------------|--------------|
| a) Data | e) Entity |
| b) Information | f) Attribute |
| c) System Catalog | g) Metadata |
| d) Data Warehouse | |

Ans.

- a) Data:
 - ✓ Data are the raw materials to derive information.
 - ✓ Data means known facts that can be recorded and have implicit meaning.
- b) Information:
 - ✓ Information is the product derived from data.
- c) System Catalog:
 - ✓ System catalog describes table-related information such as table name, column name, datatype, size of column etc.

d) Data Warehouse:

- ✓ Data warehouse is a decision support database that is maintained separately from the organization's operational database.

e) Entity:

- ✓ It is real physical object or event.
- ✓ Eg: student of college.

f) Attribute:

- ✓ It is the property of an entity.
- ✓ Eg: roll no, name branch etc of student.

g) Metadata:

- ✓ Metadata means data about data.
- ✓ It is also known as system catalog which is self describe database.

26. Explain relational algebra operation

Ans.

a) Selection

Operation.

- ✓ The select operation selects tuples that satisfy a given predicate.
- ✓ It is denoted by sigma (σ).
- ✓ It selects particular rows from Relation.

Symbol. σ (Sigma)

Notation. $\sigma_{\text{(condition)}} \langle \text{Relation} \rangle$

Operators.

- ✓ The following operators can be used in condition.
- ✓ $=, \neq, >, <, \leq, \geq, \wedge$ (AND), OR etc.

Example. Find out all accounts which belong to branch named "Surat".

Customer Relation.

cid	cname	city
101	Shweta	Bombay
102	Neha	Surat
103	Preeti	Surat
104	Ami	Bombay

$\sigma_{\text{(city='Surat')}} \langle \text{Customer} \rangle$

Output.

cid	cname	city
-----	-------	------

102	Neha	Surat
103	Preeti	Surat

b) Projection

Operation.

- ✓ Selects specified attributes of a relation.
- ✓ It select particular column from relation but select all rows of relation.

Symbol: π (pi)

Notation: $\pi_{(\text{attribute list})} <\text{Relation}>$

Example. List out Cid and cname from customer Relation.

Customer Relation.

cid	cname	city
101	Shweta	Bombay
102	Neha	Surat
103	Preeti	Surat
104	Ami	Bombay

$\pi_{(\text{Cid, cname})} <\text{Customer}>$

Output.

cid	cname
101	Shweta
102	Neha
103	Preeti
104	Ami

c) Intersection

Operation.

- ✓ Selects tuples those are in both relations.

Symbol: \cap (Intersection)

Notation: Relation 1 \cap Relation2

Requirement.

- ✓ Intersection must be taken between compatible relations.
Relations R and S are Compatible if
- ✓ Both have same arity.
- ✓ Attribute Domain must be compatible.

Example-6: $_{(\text{name})} (\text{Employee}) \cap _{(\text{name})} (\text{Customer})$

Emp	
Id	Name
1	Manisha
2	Anisha
3	Nisha

Cst	
Id	Name
1	Manisha
2	Anisha
4	Isha

Emp \cap Cst	
Id	Name
1	Manisha
2	Anisha

d) Cartesian product

Operation.

- ✓ Combines information of two relations.
- ✓ It is also known as Cross-product operation and similar to mathematical Cartesian-product operation.

Symbol. X (cross)

Notation. Relation 1 X Relation2

Resultant Relation.

- ✓ If Relation1 and Relation2 have n1 and n2 attributes respectively then, Resultant relation = n1 + n2 attributes
- ✓ If both relations having same name of attribute, it can be distinguished by combining 'Relation-name .Attribute-name'.
- ✓ If Relation1 and Relation2 have n1 and n2 tuples respectively, then Resultant relation will have n1 * n2 attributes.
- ✓ **Example.** Combine only consistent information from EMP and Dept relation.

Emp		
Empid	Empname	Deptname
S01	Manisha	Finance
S02	Anisha	Sales
S03	Nisha	Finance

Dept	
Deptame	Manager
Finance	Arun
Sales	Rohit
Production	Kishan

Output: Emp × Dept

Emp × Dept				
Empid	Empname	Emp.Deptname	Dept.Deptname	Manager
S01	Manisha	Finance	Finance	Arun
S01	Manisha	Finance	Sales	Rohit
S01	Manisha	Finance	Production	Kishan
S02	Anisha	Sales	Finance	Arun
S02	Anisha	Sales	Sales	Rohit
S02	Anisha	Sales	Production	Kishan
S03	Nisha	Finance	Finance	Arun
S03	Nisha	Finance	Sales	Rohit
S03	Nisha	Finance	Production	Kishan

e) Rename

Operation.

- ✓ It is used to rename the relation or attributes.

Symbol. ρ (Rho)

Notation. $\rho_A(B)$

- ✓ Rename Relation B to A.

Example. $\rho_{\text{vpmp_student}}(\text{Student})$