

1) explain two tier and three tier architecture

2-Tier Architecture

- The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.

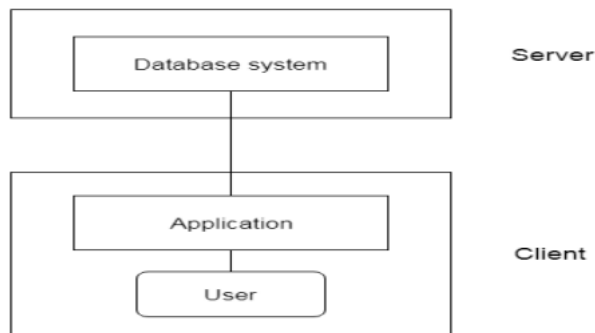


Fig: 2-tier Architecture

3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.
- The application on the client-end interacts with an application server which further communicates with the database system.
- The 3-Tier architecture is used in case of large web application.

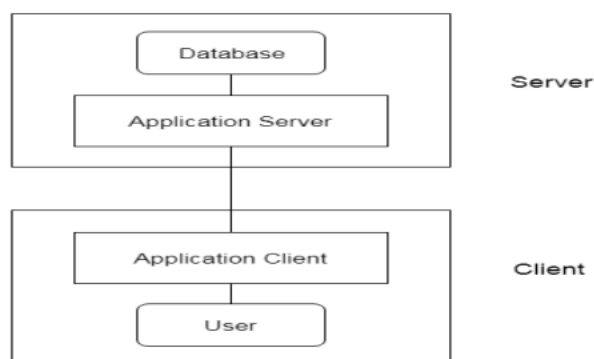


Fig: 3-tier Architecture

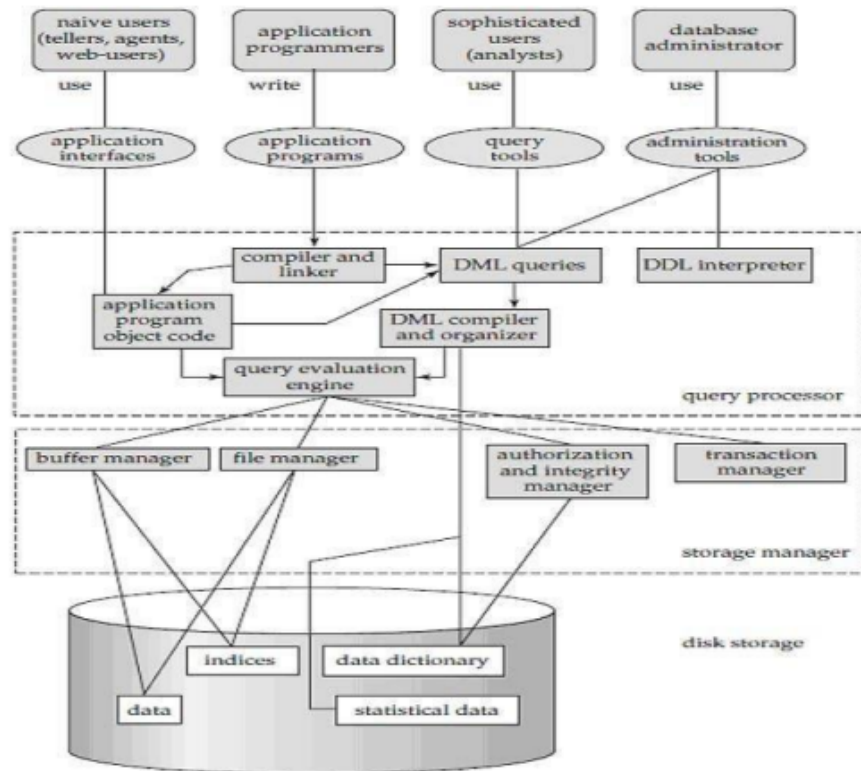
2) compare dbms vs file system

| | |
|--|---|
| The file system provides the details of data representation and storage of data. | DBMS gives an abstract view of data that hides the details |
| Storing and retrieving of data can't be done efficiently in a file system. | DBMS is efficient to use as there are a wide variety of methods to store and retrieve data. |
| It does not offer data recovery processes. | There is a backup recovery for data in DBMS. |
| The file system doesn't have a crash recovery mechanism. | DBMS provides a crash recovery mechanism |
| Protecting a file system is very difficult. | DBMS offers good protection mechanism. |
| In a file management system, the redundancy of data is greater. | The redundancy of data is low in the DBMS system. |
| Data inconsistency is higher in the file system. | Data inconsistency is low in a database management system. |
| The file system offers lesser security. | Database Management System offers high security. |
| File System allows you to stores the data as isolated data files and entities. | Database Management System stores data as well as defined constraints and interrelation. |
| Not provide support for complicated transactions. | Easy to implement complicated transactions. |
| The centralization process is hard in File Management System. | Centralization is easy to achieve in the DBMS system. |
| It doesn't offer backup and recovery of data if it is lost. | DBMS system provides backup and recovery of data even if it is lost. |
| There is no efficient query processing in the file system. | You can easily query data in a database using the SQL language. |
| These system doesn't offer concurrency. | DBMS system provides a concurrency facility. |

3) Illustrate dbms system structure

Structure of a DBMS: Database System Structure

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be broadly divided into the **storage manager** and the **query processor** components.

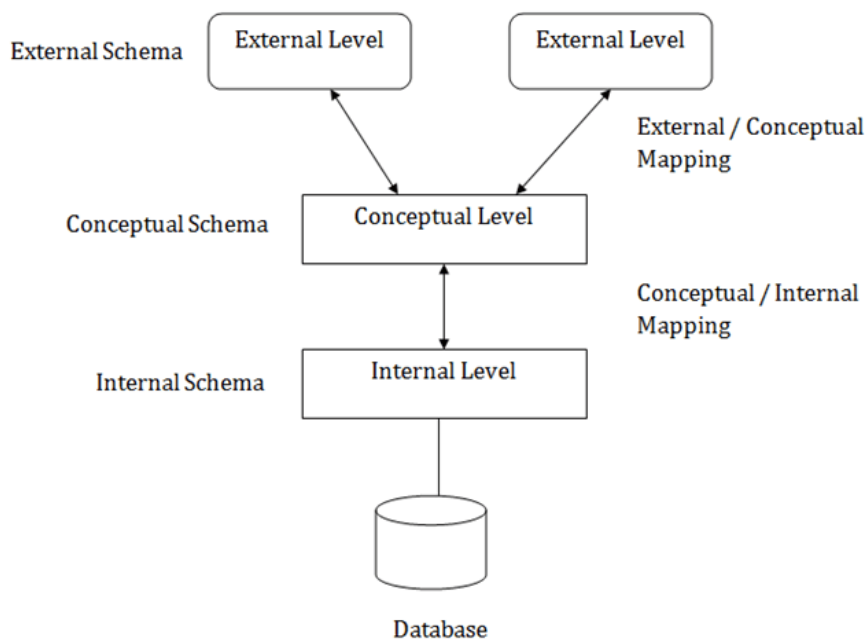


4) explain three schema architecture

Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:



5) List out various data types in sql.

BASIC SQL

Data Types

- char(n): Fixed length character string, with user-specified length n .
- varchar(n): Variable length character strings, with user-specified maximum length n .
- int: Integer (a finite subset of the integers that is machine-dependent).
- Smallint: Small integer (a machine-dependent subset of the integer domain type).
- number(p,d): Fixed point number, with user-specified precision of p digits, with d digits to the right of decimal point. (ex., numeric(3,1), allows 44.5 to be stores exactly, but not 444.5 or 0.32).
- real, double precision: Floating point and double-precision floating point numbers, with machine-dependent precision.
- float(n): Floating point number, with user-specified precision of at least n digits.
- Date: Date format in dd-MON-yyyy.Example: 17-AUG-2021

6) what is model? explain various data models with neat diagram.

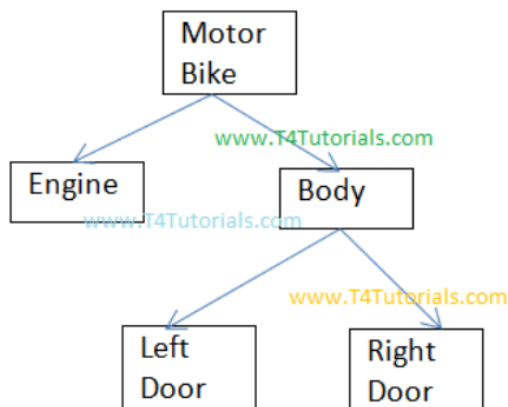
DBMS DATA MODELS Underlying the structure of a database is the data model: a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. The data models can be classified into four different categories:

1.Hierarchy Database

It is almost the first good database we used to organize the data. It is developed by computer software development company IBM for IMS (Information Management system).

The design of this model is very simple as we arrange the data in the tree-like structure.

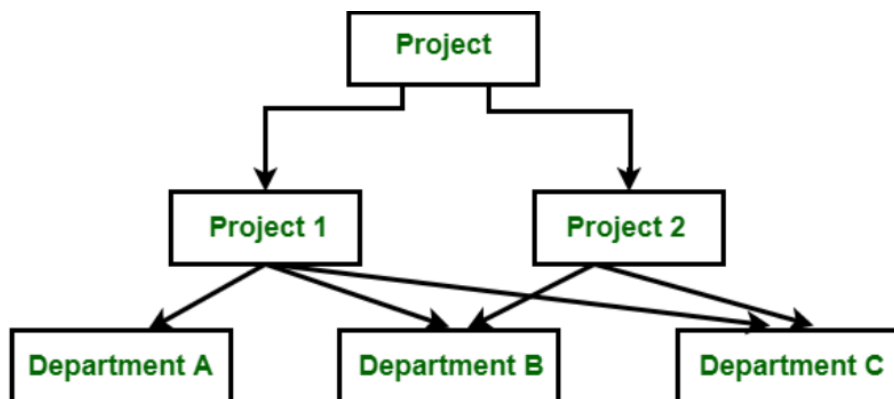
This tree model has nodes and branches which represent fields and different areas of data.



2.Network Database

You can understand the network model as a Graph in data structures as follows –

- We store the data in a graph-like manner which means it can have more than 1 root node.
- This is the reason we said this model as an advanced Hierarchy model and known it as a modified version of the Hierarchical database.



3. Relational Model.

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns, and each column has a unique name. Tables are also known as relations. The relational model is an example of a record-based model.

| SID | SName | SAge | SClass | SSection |
|------|--------|------|--------|----------|
| 1101 | Alex | 14 | 9 | A |
| 1102 | Maria | 15 | 9 | A |
| 1103 | Maya | 14 | 10 | B |
| 1104 | Bob | 14 | 9 | A |
| 1105 | Newton | 15 | 10 | B |

4. Entity-Relationship Model.

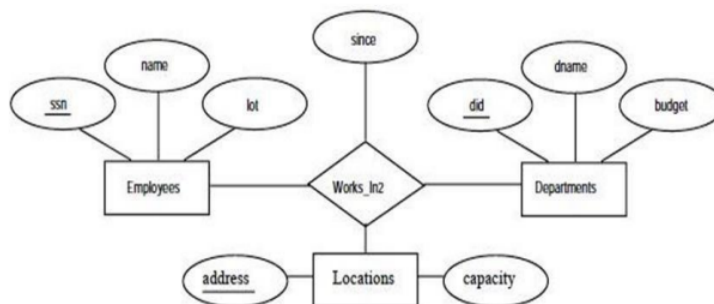
An **Entity-relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

The entity-relationship (E-R) data model uses a collection of basic objects, called entities, and relationships among these objects.

Mapping cardinalities –

- one to one
- one to many
- many to one
- many to many

Suppose that each department has offices in several locations and we want to record the locations at which each employee works. The ER diagram for this variant of Works In, which we call Works In2.



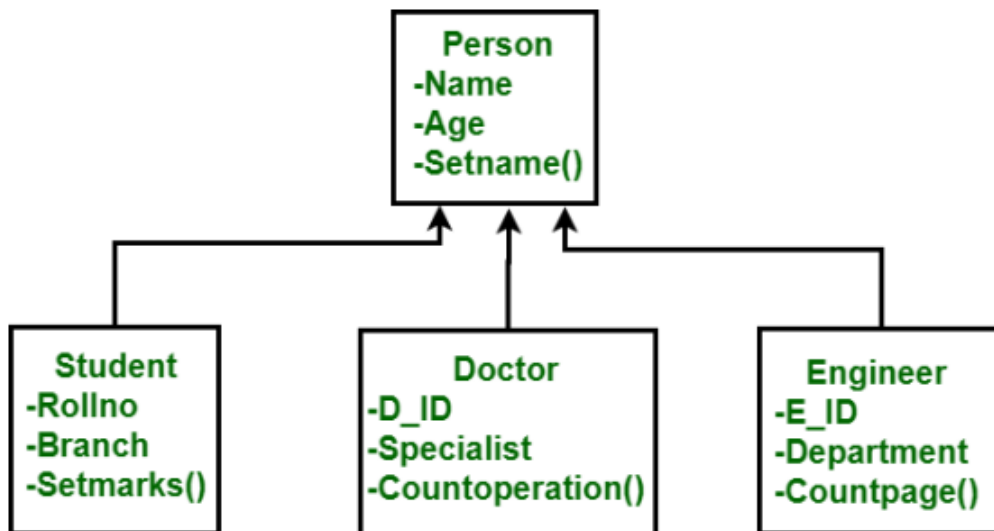
5. Object-oriented database (OOB) Model

OOB is totally based on object-oriented programming as it can show our data in the form of objects.

This database comes up with functionality as similar to OOP as Polymorphism, Encapsulation, Inheritance, Abstraction.

In this model, we have the capability to handle different type of data as you can't do this in any other model

It model comes up with functionalities like Reusability, more flexibility etc which gives this model an upper hand above others.



7) explain ddl and dml commands.

1.Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. CREATE It is used to create a new table in the database.

Example:

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);
```

b. DROP: It is used to delete both the structure and record stored in the table.

Example:

```
DROP TABLE EMPLOYEE;
```

c. ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Example:

1. ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
2. ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

c. TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

Example:

```
TRUNCATE TABLE EMPLOYEE;
```


3. Data Manipulation Language

DML commands are used to modify the database. It is responsible for all form of changes in the database.

The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- o INSERT
- o UPDATE
- o DELETE
- o SELECT

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Example:

```
INSERT INTO EMP VALUES (7369, 'SMITH', 'CLERK', 7902, '17-DEC-80', 800, NULL, 20);
```

b. UPDATE: This command is used to update or modify the value of a column in the table.

Example:

```
UPDATE DEPT  
SET DNAME='IT';
```

OR

```
UPDATE DEPT  
SET DNAME='IT'  
WHERE DEPTNO=10;
```

c. DELETE: It is used to remove one or more row from a table.

Example:

```
DELETE FROM DEPT  
WHERE DEPTNO=20;
```

d. SELECT: It is a Data Query Language, but used with DML commands. used to view a single row or multiple rows or single column or multiple columns of a table.

Example:

```
SELECT * FROM EMP;
```

Select with WHERE Clause

```
SELECT column1, column2, columnN  
FROM table name  
WHERE [condition]
```

```
SQL> SELECT ID, NAME, SALARY  
FROM CUSTOMERS  
WHERE SALARY > 2000;
```

8) explain select clause in sql.

The **SELECT** statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

d. SELECT: It is a Data Query Language, but used with DML commands. used to view a single row or multiple rows or single column or multiple columns of a table.

Example:

SELECT * FROM EMP;

Select with WHERE Clause

```
SELECT column1, column2, columnN  
FROM table name  
WHERE [condition]
```

```
SQL> SELECT ID, NAME, SALARY  
FROM CUSTOMERS  
WHERE SALARY > 2000;
```

9) explain various data models in dbms.

10) what are the various operators in sql.

What is an Operator in SQL?

An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations. These Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.

- Arithmetic operators
- Comparison operators
- Logical operators

SQL Arithmetic Operators

| Operator | Description |
|----------|-------------|
| + | Add |
| - | Subtract |
| * | Multiply |
| / | Divide |
| % | Modulo |

SQL Compound Operators

| Operator | Description |
|----------|--------------------------|
| += | Add equals |
| -= | Subtract equals |
| *= | Multiply equals |
| /= | Divide equals |
| %= | Modulo equals |
| &= | Bitwise AND equals |
| ^-= | Bitwise exclusive equals |
| *= | Bitwise OR equals |

SQL Comparison Operators

| Operator | Description |
|----------|--------------------------|
| = | Equal to |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal to |
| <= | Less than or equal to |
| <> | Not equal to |

SQL Logical Operators

| Operator | Description |
|----------|--|
| ALL | TRUE if all of the subquery values meet the condition |
| AND | TRUE if all the conditions separated by AND is TRUE |
| ANY | TRUE if any of the subquery values meet the condition |
| BETWEEN | TRUE if the operand is within the range of comparisons |
| EXISTS | TRUE if the subquery returns one or more records |
| IN | TRUE if the operand is equal to one of a list of expressions |
| LIKE | TRUE if the operand matches a pattern |
| NOT | Displays a record if the condition(s) is NOT TRUE |
| OR | TRUE if any of the conditions separated by OR is TRUE |
| SOME | TRUE if any of the subquery values meet the condition |

SQL Bitwise Operators

| Operator | Description |
|----------|----------------------|
| & | Bitwise AND |
| | Bitwise OR |
| ^ | Bitwise exclusive OR |

11) compare primary key and foreign key.

PRIMARY KEY VERSUS FOREIGN KEY

| PRIMARY KEY | FOREIGN KEY |
|---|--|
| A specific choice of a minimal set of attributes or columns that uniquely specify a tuple or a row in a table | A field or collection of fields in one table that uniquely identifies a row of another table or the same table |
| Related to a single table | Related to two tables |
| Value cannot be null | Value can be null |
| Cannot have duplicate values | Can have duplicate values |
| There can only be a single primary key in a table | There can be multiple foreign keys in a table |
| Used to identify the records of the table uniquely | Used to link two tables together |
| | Visit www.PEDIAA.com |

12) Explain entity, entity set, attribute, relationship, relationship set.

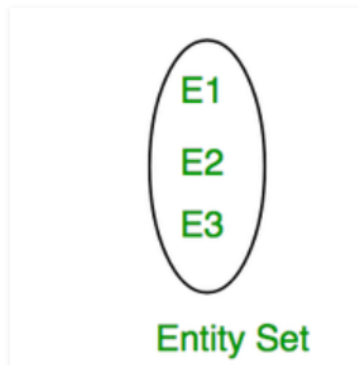
Entity

Any thing that has an independent existence and about which we collect data. It is also known as entity type.

In ER modeling, notation for entity is given below.



3. Entity Set : An entity set is a collection or set of all entities of a particular entity type at any point in time. The type of all the entities should be the same.



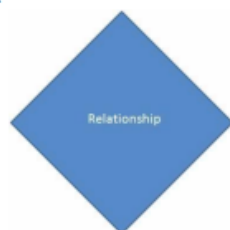
Attributes

Properties/characteristics which describe entities are called attributes. In ER modeling, notation for attribute is given below.



Relationships

Associations between entities are called relationships Example : An employee works for an organization. Here "works for" is a relation between the entities employee and organization. In ER modeling, notation for relationship is given below.



Relationship Set:

A set of relationships of same type is known as relationship set. The following relationship set depicts S1 is enrolled in C2, S2 is enrolled in C1 and S3 is enrolled in C3.

