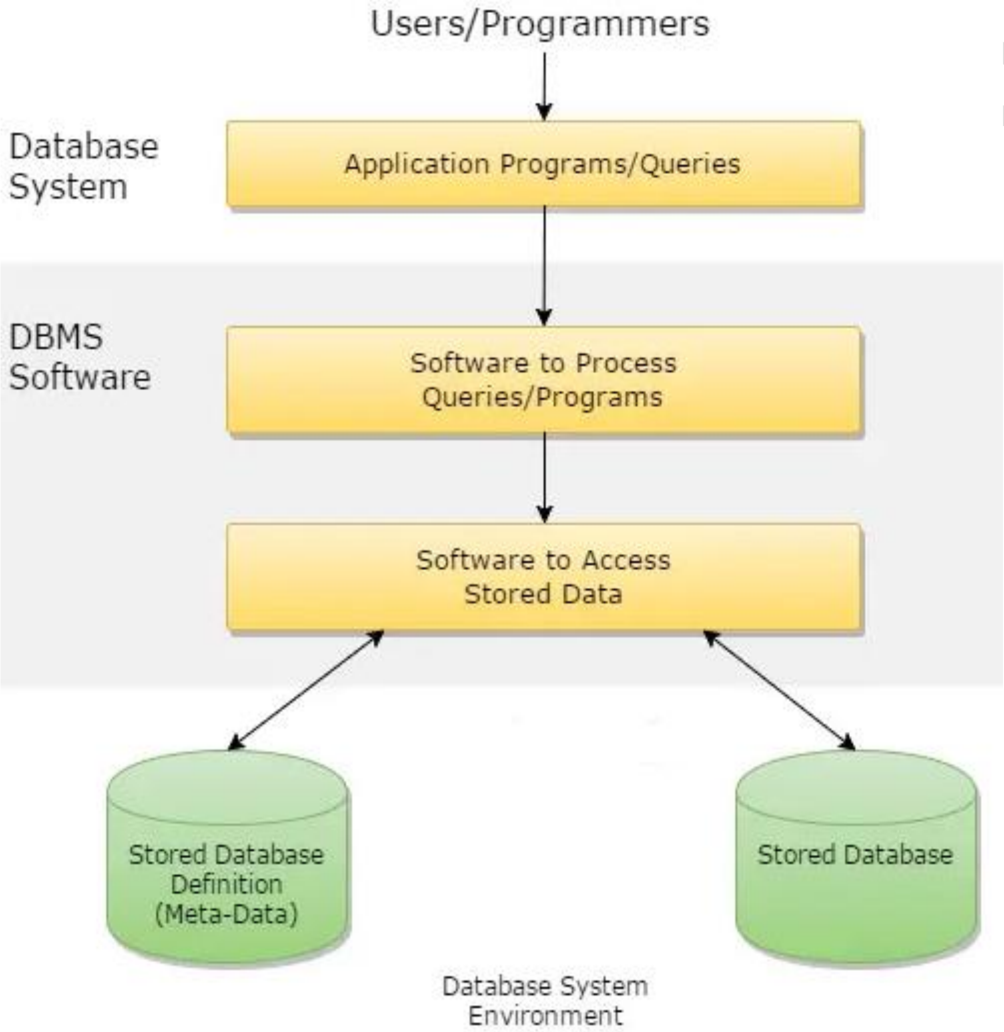
Unit – I
Introduction to Database System and SQL commands

1.1 Concepts and Definitions:
Database and database systems and database environment

A database environment is a collective system of components that comprise and regulates the group of data, management, and use of data, which consist of software, hardware, people, techniques of handling database, and the data also**.**


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database
Database System Environment

1.2 Data, Information, Data Item or Fields, Records, Files, Metadata, Data dictionary and it's components,

Data

- ❖ "DATA is a raw fact, anything can be data".
- ❖ It is not be meaningful.
- ❖ It is used analysis and reasoning.
- ❖ It is input processing unit.
- ❖ Data is basic row material which taken by certain observation, certain experiment and storing in paper, stored memory, human mind.
- ❖ Ex:- marks of student ,account information

INFORMATION

- ❖ This is a processed form of the data.
- ❖ It is always meaningful.
- ❖ "Meaningful data is called information".
- ❖ It is output processing units.
- ❖ Whenever we organize the data and process is done on it then we get proper information.

Data Item or Fields:

- ❖ A field is a character or group of characters that have a specific meaning.
- ❖ It is also called a data item. It is represented in the database by a value.
- ❖ For Example, customer id, name, society and city are all fields for customer Data.

Records:

- ❖ A record is a collection of logically related fields. For examples, collection of fields (id, name, society & city) forms a record for customer.

Files:

- ❖ A group of related records. Files are frequently classified by the application for which they are primarily used (employee file).

Metadata:

- ❖ Metadata is data about data.
- ❖ Data such as table name, column name, data type, authorized user and user access privileges for any table is called metadata for that table.

1.3 Schemas, Sub-schemas, and Instances

Schemas :

- ❖ The plan or formulation of database is known as schema.
- ❖ Schema gives the names of the entities and attributes. It specifies relationship among them.
- ❖ Schema includes the definition of database name, record type, components that make up records.
- ❖ Schema can be categorized in two ways:
- ❖ Logical schema (Table, View)
- ❖ Physical Schema (Secondary storage)

Sub-schemas :

- It is a subset of schema and inherits the same property that schema has.

Instances :

- The contents of the database at any point of time (or current state), it is referred to as INSATANCE of database.

1.4 Data types

- PL/SQL is super set of the SQL. So, it supports all the data types provided by SQL.
- Along with this, in PL/SQL Oracle provides subtypes of the data types.
- For example, the data type NUMBER has a subtype called INTEGER.
- These subtypes can be used in PL/SQL block to make the data type compatible with the data types of the other programming languages.

The various data types can be given as below:

Category	Data Type	Sub types/values
Numerical	NUMBER	BINARY_INTEGER, DEC, DECIMAL, DOUBLE PRECISION, FLOAT, INTEGER, INT, NATURAL, POSITIVE, REAL, SMALLINT
Character	CHAR, LONG, VARCHAR2	CHARACTER,VARCHAR, STRING, NCHAR, NVARCHAR2
Date	DATE	
Binary	RAW, LONG RAW	
Boolean	BOOLEAN	Can have value like TRUE, FALSE and NULL.
RowID	ROWID	Stores values of address of each record.

1.5 Database Language commands: Data Definition Language (DDL): CREATE, ALTER, TRUNCATE, DROP

- **<u>Create:</u> It used to create table.**

Syntax:
CREATE TABLE TableName (column1 datatype (size), column2 datatype(size)...ColumnN datatype(size));

Example:
CREATE TABLE student(Rollno number(15),Name varchar2(20),Age number(5),DOB date);
Output: Table created.

- **To describe a table use command DESCRIBE,**

Syntax:
DESCRIBE TableName;
OR
DESC TableName;

Example: DESC student;

Output:

```
Name       ? Null                  Type
----------------------------------------------------------------------------------------------------
Rollno                            NUMBER (15)
Name                               VARCHAR (20)
Age                                NUMBER (5)
DOB                                DATE
```

- **ALTER:** **It used to modify table structure.**

A. Adding New Columns

Syntax:
Alter Table TableName Add (NewColumnName1 Datatype (size), NewColumnName2 Datatype (size)…. NewColumnNameN Datatype (size));

Example:
ALTER TABLE student ADD (marks number (15)); //Add column marks
Output: Table altered.

B. Modifying Existing Columns

Syntax:
Alter Table TableName Modify (ColumnName1 NewDatatype (NewSize), ColumnName2 NewDatatype (NewSize)……ColumnNameN NewDatatype (NewSize));

Example: ALTER TABLE student Modify (marks number (20)); // Change size of column
Output: Table altered

C. Dropping(deleting) Existing Columns

Syntax: Alter Table TableName DROP CLOUMN ColumnName1, ColumnName2;

Example: ALTER TABLE student DROP COLUMN marks; // Remove marks column
Output: Table altered.

❖ **<u>DROP</u>: It used to delete or destroy table from a database.**

Syntax: DROP TABLE TableName

Example: DROP TABLE student;
Output: Table Dropped.

❖ **<u>TRUNCATE</u>:**

- TRUNCATE TABLE used to delete all data from a table
- Logically, this is equivalent 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 student;
Output: Table Truncated.

1.6 Database Language: Data Manipulation Language (DML): INSERT, SELECT, UPDATE, DELETE

❖ DML is a set of SQL Commands used to insert, modify and delete data in a database.
❖ These SQL commands are used for storing, retrieving, modifying, and deleting data.
❖ It is normally used by general users who are accessing database via pre-developed applications.
❖ These Data Manipulation Language commands are:
1) INSERT
2) UPDATE
3) DELETE
4) SELECT

1) SQL INSERT Statement

❖ The INSERT Statement is used to add new rows of data to a table.

Syntax:
INSERT INTO TableName(ColumnName1, ColumnName2…ColumnNameN) values (Expression1, Expression2….ExpressionN);

Example:
INSERT INTO Student (Sno, Sname, age, Branch) values (1,’Niyati’, 17,’CE’);
Output: 1 row inserted

❖ We can insert NULL value in Table Using INSERT Statement.

Example:
INSERT INTO Student (Sno, Sname, age, Branch) values (1,'Niyati', 17, NULL);

Output: 1 row inserted.

2) SQL UPDATE Statement

❖ The UPDATE Statement is used to modify the existing rows in a table.

Syntax:
UPDATE TableName SET Column_Name1 = value1, Column_Name2 = value2 WHERE condition;

Example:
UPDATE Student SET Name='nilam' WHERE Rollno=1;
Output: 1 row Updated.

NOTE:

❖ In the Update statement, WHERE clause identifies the rows that get affected.
❖ If you do not include the WHERE clause, column values for all the rows get affected.

3) SQL Delete Statement

❖ The DELETE Statement is used to delete rows from a table.

Syntax:
DELETE FROM TableName WHERE condition;

Example:

❖ To delete an employee with id 100 from the employee table, the SQL delete query would be like,
DELETE FROM Student WHERE Rollno = 100;

❖ To delete all the rows from the employee table, the query would be like,
DELETE FROM Student;

4) SQL SELECT Statement

❖ The SELECT command is used to retrieve selected rows from the Tables.

Syntax:

a. SELECT * FROM TableName;
b. SELECT ColumnName1, ColumnName2**…** ColumnNameN From TableName;
c. SELECT * FROM TableName WHERE Condition;

d. SELECT ColumnName1, ColumnName2**…** ColumnNameN From TableName WHERE Condition;

Example:

a. SELECT * FROM student;
b. SELECT Rollno, Name FROM student;
c. SELECT * FROM student where Rollno=101;
d. SELECT Rollno, Name FROM student where Rollno=101;

1.7 Transactional Control: Commit, Save point, Rollback

- TCL – Transaction Control Language

- **Definition:** A Transaction is a set of database operations that performs a particular task.

- A transaction must be completely successful or completely fail without doing anything to maintain database consistency.
- Example, Successfully Complete Transaction: A task, say fund transfer from one account to another account.
- To complete a transaction needs to update balance in two accounts – source and destination.
- It requires two database operations: one, to debit an amount in source account and to credit that amount in destination account.
- Example, Fail Transaction: In a fund transfer, if a system failure occurs after debiting amount from source account, but before crediting it to destination account then database will be in inconsistent state.
- So, balance should be updated in both the accounts in both accounts or not in anyone.
- We can say that a transaction is considered as a sequence of database operations.
- These operations involve various data manipulation operations such as insert, update and delete.
- These operations are performed in two steps:
    - Changes are made in memory only.
    - Changes are permanently saved to hard disk.
- A transaction begins with the execution of first SQL statement after a COMMIT and can be undone using ROLLBACK command.
- A transaction can be closed by using COMMIT or ROLLBACK command. When a transaction is closed, all the locks acquired during that transaction are released.
- TCL commands are used to manage transactions, that are given below:
    - Commit
    - Rollback
    - Savepoint

COMMIT: Committing a Transaction

- There are two ways to commit a transaction:
  - Explicitly
  - Implicitly

Explicit Commit:

- To commit a transaction explicitly, user needs to request COMMIT command explicitly.
- A COMMIT command terminates the current transaction and makes all the changes permanent.
- Various data manipulation operations such as insert, update and delete are not effect permanently until they are committed.

- **Syntax:**

```
COMMIT;
```

- **Output:**

```
Commit complete
```

Implicit Commit:

- There are some operations which forces a **COMMIT** to occur automatically, even user don't specify the **COMMIT** command.
- Some of commands are given below:

  Quit Command:

  To end SQL*PLUS session disconnecting from the Oracle.

  Exit Command:

  To end SQL*PLUS session disconnecting from the Oracle.

  Data Definition Language(DDL) commands:

  Commands like CREATE.., ALTER.., DROP.. are immediate and makes all prior changes, made during current transaction permanent.

ROLLBACK: Canceling a Transaction Completely

- A transaction can be cancelled using **ROLLBACK** command either **completely or partially**.
- A **ROLLBACK** command terminates the current transaction and undone any changes made during the transaction.
- Oracle also performs auto rollback. In situation like, Computer failure, Oracle automatically rollbacks any uncommitted work, when the database bought back next time.

SAVEPOINT: Cancelling a Transaction Partially

- A **ROLLBACK** command can also used to terminate the current transaction partially.

- **Syntax:**
  ROLLBACK TO SAVEPOINT savepoint_name;
- **Output:**
  Rollback Complete.

- It is required to create **a savepoint to cancel transaction partially**.
- A savepoint marks and save the current point in the processing of a transaction.
- A savepoint can be created using command **SAVEPOINT** as given below:

- **Syntax:**
  SAVEPOINT savepoint_name**;**
- **Output:**
  Savepoint created.

- When a **ROLLBACK** is used with **SAVEPOINT**, part of the transaction is cancelled.
- All the operations performed after creating a savepoint are undone.
- It is also possible to create more than one savepoint within a single transaction.

1.8 DCL Commands: Grant and Revoke

GRANT – Granting Privileges

- **GRANT** command is used to granting privileges means to give permission to some user to access database object or a part of a database object.
- This command provides various types of access to database object such as tables, views and sequences.

Syntax:

```
GRANT   object privileges
ON object name
TO user name
[ WITH GRANT OPTION ];
```

- The owner of a database object can grant all privileges or specific privileges to other users.
- The **WITH GRANT OPTION** allows the grantee. User to which privilege is granted to in turn grant object privilege to other users.
- User can grant all or specific privileges owned by him/her.

Example:

```
GRANT ALL
ON     Customer
TO     user2
WITH GRANT OPTION;
```

Output:

```
Grant Succeeded.
```

- Observe the use of **WITH GRANT OPTION**. If this option is not specified, user2 will have privileges, but he cannot grant these privileges to other users.

REVOKE – Revoking Privileges

- Revoking privileges means to deny (decline) permission to user given previously.
- The owner on an object can revoke privileges granted to another user. A user of the object, who is not an owner, but has been granted privileges using **WITH GRANT OPTION**, can revoke the privilege from the grantee.

  Syntax: **REVOKE** object privileges
  ON object name **FROM** user name**;**

  Example:

```
Revoke SELECT, INSERT
ON     Customer
FROM  user2;
```

GRANT	REVOKE
This DCL command grants permissions to the user on the database objects.	This DCL command removes permissions if any granted to the users on database objects.
It assigns access rights to users.	It revokes the user access rights of users.
For each user you need to specify the permissions.	If access for one user is removed; all the particular permissions provided by users to others will be removed.
When the access is decentralized granting permissions will be easy.	If decentralized access removing the granted permissions is difficult.