

Unit-1

Structure Query Language

❖ Introduction to SQL :

- Structure Query Language (SQL) is a language that provides an interface to relational database systems.
- SQL was developed by IBM in the 1970s for use in System R.
- SQL is also known as SEQUEL.
- SQL is used for creating and modifying tables and other database.
- SQL has been a command language for communication with the Oracle 9i server any tool or application.
- When an SQL statement is entered, it is stored in a part of memory called the SQL buffer and remains there until a new SQL statement is entered.

❖ Features of SQL :

1. SQL can be used by a range of users, including those with little or no programming experience.
2. It is a non procedural language.
3. It reduces the amount of time required for creating and maintaining systems.
4. It is English-like language.

❖ Components of SQL :

1. DDL (Data Definition Language) :

- DDL commands used to create, modify and delete database structure but not data.
- These commands used by DBA, a database designer or application developer.
- **Examples:**
 - i. **CREATE** : To create objects in the database
 - ii. **ALTER** : Alters the structure of the database
 - iii. **DROP** : Delete objects from the database
 - iv. **TRUNCATE** : Remove all records from a table, including all spaces allocated for the records are removed

- v. **COMMENT** : Add comments to the data dictionary
- vi. **GRANT** : Gives user's access permissions to database
- vii. **REVOKE** : Withdraw access permission given with GRANT command

2. DML(Data Manipulation Language) :

- DML commands used to changing data within the database structure.
- Manipulation of data means inserting, updating, deleting data from database structure.
- **Examples:**
 - i. **INSERT** : Insert data into table
 - ii. **UPDATE** : Updates existing data within a table
 - iii. **DELETE** : Deletes all records from a table, the space for the records remain
 - iv. **CALL** : Call a PL/SQL or Java subprogram
 - v. **EXPLAIN PLAN** : Explain access path to data
 - vi. **LOCK TABLE** : Control concurrency

3. DCL(Data Control Language) :

- DCL commands used to control access to data and to the database.
- Sometimes DCL commands grouped with DML commands.
- **Examples :**
 - i. **COMMIT** : Save work done
 - ii. **ROLLBACK** : Restore database to original from the last COMMIT
 - iii. **SAVEPOINT**: Identify a point in a transaction, so we can rollback from that point when failure occurs
 - iv. **SET TRANSACTION**: Change transaction options like setting rollback segment
 - v. **GRANT/REVOKE**: Grant or take back permissions to or from the oracle users

4. DQL(Data Query Language):

- DQL commands used to getting data from the database.
- It is also used for ordering data.
- **Example:**
 - i. **SELECT**: Used to retrieve data from the database. This command is the heart of SQL.

❖ Rules for SQL :

1. SQL starts with a verb (action word) ex. SELECT statements. This verb may have additional adjectives. Ex. FROM
2. A space separates clauses. Ex. DROP TABLE EMP;
3. A ';' is used to end SQL statements.
4. Statements may be split across lines but keyword may not.
5. A comma is used to separate parameter.
6. Character and date literals must be enclosed within single quotes.

❖ Oracle Data Types:

- Data types come in several forms and sizes, allowing the programmer to create tables suited to the scope of the project.
 - Following are the basic data types of oracle.
1. **CHAR(size):** This data type is used to store character strings values of fixed length.
 - The size in brackets determines the number of character the cell can hold.
 - This data type can hold maximum 255 characters.
 2. **VARCHAR(size) or VARCHAR2(size):** This data type is used to store variable length alphanumeric data.
 - It is more flexible form of the CHAR data type.
 - This data type can hold 4000 characters.
 3. **DATE:** This data type is used to represent date and time.
 - The standard format is DD-MON-YY.
 4. **NUMBER(P,S):** The NUMBER data type is used to store numbers.
 - The Precision (P) determines the maximum length of the data.
 - The Scale (S) determines the number of places to the right of the decimal.
 5. **LONG:** This data type is used to store variable length character strings up to 2 GB.
 - LONG data can be used to store arrays of binary data in ASCII format.
 - Only one LONG value can be defined per table.
 6. **RAW/LONG RAW:** The RAW/LONG RAW data types are used to store binary data, such as digitized picture or image.
 - Maximum length of LONG RAW data is 255 bytes or 2GB.

❖ **The CREATE TABLE command:**

- **Use:** CREATE TABLE command is used to create a new table in database.
 - It defines each column of the table uniquely.
 - Each column has a minimum three attributes- a name, datatype and size.
- **Rules For Creating Tables:**
 1. A name can have maximum up to 30 characters.
 2. A name should begin with alphabet.
 3. A-Z, a-z or 0-9 are allowed.
 4. Special character is allowed (_, \$, #)
 5. Keywords are not allowed.
- **Syntax:**

```
CREATE TABLE TableName (ColumnName1 DataType(size),  
                          ColumnName2 DataType(size));
```

- **Example:**

```
CREATE TABLE "Emp" (Eno NUMBER(3), Ename VARCHAR2(15));
```

❖ **Inserting Data Into Tables: (INSERT INTO Command)**

- **Use:** INSERT INTO command is used to insert data into a table.
 - This command creates a new row in the table and
 - Loads the values passed by the INSERT INTO in specified columns.
 - In the INSERT INTO SQL sentence, tables and column have a one to one relationship.
- **Syntax:**

```
INSERT INTO TableName (ColumnName1, ColumnName2)  
VALUES (expression1, expression2);
```

- **Example:**

```
INSERT INTO Emp (Eno, Ename) VALUES (1,'Rahul');
```

```
INSERT INTO Emp (Eno, Ename) VALUES (2,'Pankaj');
```

❖ **Viewing Data In The Tables: (SELECT Command)**

- **Use:** SELECT command is used to retrieve rows selected from one or more tables.
- **Syntax:**

SELECT * FROM TableName;

- **Example:**

SELECT * FROM Emp;

➤ **Filtering Table Data:**

- While viewing data from a table, it is rare that we required all the data each time. So SQL provides a method of filtering table data.
- There are mainly three ways of filtering data.
 1. Selected columns and all rows
 2. Selected rows and all columns
 3. Selected columns and selected rows

- **Selected columns and all rows:** The retrieval of specific columns from a table can be done as shown below.

- **Syntax:**

SELECT ColumnName1, ColumnName2 **FROM** TableName;

- **Example:**

SELECT Ename **FROM** Emp;

- **Selected Rows and all Columns:** If information of a particular client is to be retrieved from a table, its retrieval must be based on a specific condition.
- Oracle provides the option of using a WHERE clause in an SQL query to apply a filter on the rows retrieved.
- The oracle engine displays only those records that satisfy the specified condition.
- **Syntax:**

SELECT * FROM TableName **WHERE** Condition;

- **Example:**

SELECT * FROM Emp WHERE Ename='Pratik';

SELECT * FROM Emp WHERE Eno=2;

- **Selected Rows and Selected Columns:** To view a specific set of rows and columns from a table we use specific column name and WHERE condition.

- **Syntax:**

SELECT ColumnName1, ColumnName2 FROM TableName WHERE Condition;

- **Example:**

SELECT Ename FROM Emp WHERE Eno=3;

SELECT Ename, Ecity FROM Emp WHERE Ecity='Hmt';

❖ **Eliminating Duplicate Rows Using SELECT**

- To view only unique rows the DISTINCT clause is used.
- The DISTINCT clause allows removing duplicates from the result set.
- The DISTINCT clause can only be used with select statements.
- **Syntax:**

SELECT DISTINCT ColumnName1, ColumnName2 FROM TableName;

- **Example:**

SELECT DISTICNT * FROM Emp;

❖ **Sorting Data In a Table**

- We can sort table data in oracle in ascending or descending order.
- ORDER BY clause is used to sort table data with SELECT statement.
- By default sorting order is ascending order.
- We can sort data in descending order by specifying DESC clause in select statement.

- **Syntax:**

```
SELECT * FROM TableName  
  
ORDER BY ColumnName1, ColumnName2 SORT ORDER;
```

- **Example:**

```
SELECT * FROM Emp ORDER BY Ename;  
SELECT * FROM Emp ORDER BY Ename DESC;
```

❖ **Creating a Table from a Table**

- We can create a table from existing table by following command.

- **Syntax:**

```
CREATE TABLE TableName (ColumnName, ColumnName)  
  
AS SELECT ColumnName, ColumnName FROM TableName;
```

- **Example:**

```
CREATE TABLE Emp_Addr (Addr1,Addr2,PinCode)  
  
AS SELECT Addr1, Addr2, PinCode FROM Emp;
```

- The source table is the table identified in the SELECT section of the SQL sentence.
- The Target table is one identified in the CREATE section of this SQL sentence.
- This SQL sentence creates table with data from the Source table.
- To create a Target table without data from the source table, the SELECT statement must have WHERE clause.

- **Example:**

```
CREATE TABLE Emp_Addr (Addr1,Addr2,PinCode)  
  
AS SELECT Addr1, Addr2, PinCode FROM Eid WHERE 1=2;
```

❖ **Inserting data into Table from another Table:**

- We can insert data into table from another table as below.

- **Syntax:**

```
INSERT INTO TableName
      SELECT ColumnName1, ColumnNameN FROM TableName;
```

- **Example:**

```
INSERT INTO Emp_Addr
      SELECT Addr1, Addr2, PInCode FROM Emp;
```

- If you want to insert specific records from a table then it is possible using WHERE clause.

- **Example:**

```
INSERT INTO Emp_Addr
      SELECT Addr1, Addr2, PInCode FROM Emp WHERE Ecity='Hmt';
```

❖ Delete Operations:

- The DELETE command deletes all rows from the table.

- **Syntax:**

```
DELETE FROM TableName;
```

- **Example:**

```
DELETE FROM Emp;
```

- Above statement deleted all rows from Emp table.
- If we want delete specific rows from the table we can use **WHERE** clause to do so.

- **Syntax:**

```
DELETE FROM TableName WHERE Condition;
```

- **Example:**

```
DELETE FROM Emp WHERE Eid=101;
```

- Above statement will delete only one record from the table whose Employee id is 101.

❖ Updating The Contents of a Table:

- The UPDATE statement updates columns in the table's row with new values.

- **Syntax:**

```
UPDATE TableName
      SET ColumnName1=Expr1, ColumnName2=Expr2;
```

- The SET clause indicates which column data should be modified and what new data they should have.

- **Example:**

UPDATE Emp **SET** Ecity='hmt';

- Above statement updates all rows of the table with value hmt in city column.
- If we want to update specific rows in the table we can use WHERE condition.

▪ **Example:**

UPDATE Emp **SET** Ecity='hmt' **WHERE** Eid=101;

- above statement updates city with value hmt of the employee whose id is 101.

❖ **Modifying The Structure of Tables:**

- ALTER TABLE command used for modifying the structure of the table.
- ALTER TABLE allows changing the structure of an existing table.
- **Uses of ALTER TABLE:**
 - To add or delete columns
 - To create or destroy indexes
 - To change the data type of existing columns
 - To rename columns or rename table.

❖ **Adding New Columns:**

▪ **Syntax:**

ALTER TABLE TableName
 ADD(NewColumnName Datatype(size),
 NewColumnName Datatype(size)...);

▪ **Example:**

ALTER TABLE Emp **ADD**(EDOB Date);

- Above SQL sentence add new column called EDOB to Emp table.

❖ **Dropping A Column From A Table:**

▪ **Syntax:**

ALTER TABLE TableName
 DROP COLUMN ColumnName;

▪ **Example:**

ALTER TABLE Emp **DROP COLUMN** Ecity;

- Above SQL statement delete Ecity column from Emp table.

❖ **Modifying Existing Columns:**

▪ **Syntax:**

ALTER TABLE TableName

MODIFY(ColumnName NewDataType(NewSize));

▪ **Example:**

ALTER TABLE Emp **MODIFY** (ENAME Varchar2(25));

- Above SQL statement change column datatype and size of ENAME column.

❖ **Restriction on the ALTER TABLE:**

- Following tasks cannot be performed by using alter table.
 1. Change the name of the table.
 2. Change the name of the column.
 3. Decrease the size of a column if table data exist.

❖ **RENAMING TABLES**

- Oracle allows renaming of tables.

▪ **Syntax:**

RENAME TableName **TO** NewTableName;

▪ **Example:**

RENAME Emp **TO** Employee;

❖ **TRUNCATING TABLES**

- **Use:** TRUNCATE TABLE command empties a table completely.
- logically, this is equivalent to a DELETE statement that deletes all rows, but there some practical differences...

TRUNCATE

- 1) TRUNCATE delete all table records permanently.
- 2) We can't use WHERE condition with TRUNCATE command.
- 3) TRUNCATE is DDL command
- 4) TRUNCATE works slower then DELETE

DELETE

- 1) We can ROLLBACK data which is deleted by DELETE command.
- 2) We can use WHERE condition with DELETE command.
- 3) DELETE is DML command.
- 4) DELETE works faster than TRUNCATE.

▪ **Syntax:**

TRUNCATE TABLE TableName;

▪ **Example:**

TRUNCATE TABLE Emp;

❖ **DESTROYING TABLES**

- Sometimes tables within particular database become non-usable and need to be deleted.
- At that time DROP TABLE command is used to destroy table.
- Table dropped by DROP TABLE cannot be recovered.

▪ **Syntax:**

DROP TABLE TableName;

▪ **Example:**

DROP TABLE Emp;

❖ **DATA CONSTRAINTS:**

- “Business rules, which are enforced on data being stored in a table, are called Constraints.”
- Constraints super control the data being entered into a table for permanent storage.
- Constraints could be column level or table level. Column level constraints are applied only to one column, whereas table level constraints are applied to the whole table.
- Constraints can be specified when a table is created with the CREATE TABLE statement or you can use ALTER TABLE statement to create constraints in existing table.
- there two types of data constraints
 1. I/O constraints
 2. Business rules constraints

1) I/O constraints:

- Input/output data constraint determines the speed at which data can be inserted or extracted from oracle table.
- I/O constraints divided into three constraints.
 - i. PRIMARY KEY CONSTRAINT
 - ii. FOREIGN KEY CONSTRAINT
 - iii. UNIQUE KEY CONSTRAINT

I. PRIMARY KEY CONSTRAINT

- A primary key is one or more column in a table used to uniquely identify each row in the table.
- A table can have only one primary key.
- A primary key column in a table has special attributes:
 - It defines the column, as a mandatory column.
 - The data must be UNIQUE.
-