**Theory: Structured Query Language (SQL):-**

Structured Query Language (SQL) is the set of statements with which all programs and users access data in an Oracle database. Application programs and Oracle tools often allow users access to the database without using SQL directly, but these applications in turn must use SQL when executing the user's request.

## 1. CREATE TABLE

Use the CREATE TABLE statement to create one of the following types of tables:

* A **relational table** is the basic structure to hold user data.
* An **object table** is a table that uses an object type for a column definition. An object table is explicitly defined to hold object instances of a particular type.

You can also create an object type and then use it in a column when creating a relational table.

Tables are created with no data unless a query is specified. You can add rows to a table with the INSERT statement. After creating a table, you can define additional columns, partitions, and integrity constraints with the ADD clause of the ALTER TABLE statement. You can change the definition of an existing column or partition with the MODIFY clause of the ALTER TABLE statement.

**CREATE TABLE Syntax:-**

**CREATE TABLE <TABLENAME>**

**{(COLUMNNAME DATATYPE [NOT NULL] [UNIQUE]**

**[DEFAULT DEFAULTOPTION] [CHECK (SEARCHCONDITION)] [,….]}**

**{PRIMARY KEY (LISTOFCOLUMNS),]**

**{[UNIQUE (LISTOFCOLUMNS),] [,…]}**

**{[Foreign key (listofForeignkeyColumns)**

**REFERENCES ParentTableName [(listofCandidateKeyColumns)],**

**[MATCH {PARTIAL | FULL}**

**[ ON UPDATE referentialAction]**

**[ ON DELETE referentialAction]] [,…]}**

**{[ CHECK (searchCondition)] [,…]})**

**2. INSERT**

Use the INSERT statement to add rows to a table, a view's base table, a partition of a partitioned table or a subpartition of a composite-partitioned table, or an object table or an object view's base table.

INSERT INTO *table\_name*  
VALUES (*value1*,*value2*,*value3*,...);

The second form specifies both the column names and the values to be inserted:

INSERT INTO *table\_name* (*column1*,*column2*,*column3*,...)  
VALUES (*value1*,*value2*,*value3*,...);

## 3. ALTER TABLE

Use the ALTER TABLE statement to alter the definition of a nonpartitioned table, a partitioned table, a table partition, or a table subpartition. For object tables or relational tables with object columns, use ALTER TABLE to convert the table to the latest definition of its referenced type after the type has been altered.

ALTER TABLE syntax:-

**ALTER TABLE TableName**

**[ADD [COLUMN] columnName dataType[NOT NULL] [UNIQUE]**

**[DEFAULT defaultOption ] [CHECK (searchCondition )]]**

**[DROP [ COLUMN] columnName [RESTRICT | CASCADE]]**

**[ADD [ CONSTRAINT [ConstraintName]] tableConstraintDefinition]**

**[DROP CONSTRAINT ConstraintName [RESTRICT | CASCADE]]**

[**ALTER [COLUMN] SET DEFAULT defaultOption]**

**[ALTER [COLUMN] DROP DEFAULT]**

**4. SELECT**

Use a SELECT statement or subquery to retrieve data from one or more tables, object tables, views, object views, or materialized views.

SELECT TABLE syntax:-

SELECT column\_name,column\_name  
FROM table\_name;

and

SELECT \* FROM table\_name;

## 5. DROP TABLE

Use the DROP TABLE statement to remove a table or an object table and all its data from the database.

DROP TABLE Syntax:-

DROP TABLE TableName [RESTRICT | CASCADE ]

**FAQs:**

How to establish referential integrity?

Which referential actions to be used while creating tables?

**Oral/Review Questions:**

What are DDL, DML, DCL, and TCL?