

## Assignment No 2

**Title:** SQL Queries:

- a. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.
- b. Write at least 10 SQL queries on the suitable database application using SQL DML statements.

**Objective:** Understand the concept of DDL & DML Commands and its operations with Operators, functions.

**Software Required:** MYSQL Workbench

### Theory:

#### **What is SQL?**

SQL stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

#### **SQL Languages:**

**There is three types of commands are used in database that is :-**

- 1) DDL (Data Definition Language)
- 2) DML (Data Manipulation Language)
- 3) DCL (Data Control Language)

#### **DDL (Data Definition Language) :-**

- i) It Is Stands For Data Definition Language
- ii) It is used for manipulate the data.
- iii) DDL statements are used to define the database structure or schema.

#### **Operations performs on DDL are:-**

##### **1. Create operation:-**

- a) **Create database :-** The CREATE DATABASE statement is used to create a new SQL database.

Syntax:- CREATE DATABASE *databasename*;

- b) **Create Table :-** It is used to Create a Table.

Syntax:- CREATE TABLE *table\_name* ( *column1 datatype*, *column2 datatype*, *column3 datatype*);

**2. Alter Table :-** The ALTER TABLE statement is used to add, delete, or modify columns in an existing table. The ALTER TABLE statement is also used to add and drop various constraints on an existing table and alters the structure of the database

##### **a) ALTER TABLE - ADD Column**

To add a column in a table, use the following syntax :-

```
ALTER TABLE table_name
ADD column_name datatype;
```

#### **b) ALTER TABLE - DROP COLUMN**

To delete a column in a table, use the following syntax:-

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

#### **c) ALTER TABLE - ALTER/MODIFY COLUMN**

To change the data type of a column in a table, use the following syntax :-

```
ALTER TABLE table_name
ALTER COLUMN column_name datatype;
```

**3. Drop Table :-** The DROP TABLE statement is used to drop an existing table in a database.

Syntax:- DROP TABLE *table\_name*;

**4. Truncate Table :-** The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

Syntax:- TRUNCATE TABLE *table\_name*;

**5. Rename Table :-** It is used to rename an object . It is used for give another name to the table.

Syntax :- Rename *old\_table\_name* to *New\_table\_name* ;

### **SQL Objects :-**

- 1) **Table:** A table is a collection of related data held in a structure format within a database it consists of column and row a table is a set of data elements using a model of vertical column and horizontal rows the cell being the init where a row and column insert .
- 2) **View:** In SQL, a view is a virtual table based on the result-set of an SQL statement.A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

#### **a) CREATE VIEW:**

Syntax :- CREATE VIEW *view\_name* AS SELECT *column1, column2, ...*  
FROM *table\_name* WHERE *condition*;

#### **b) SQL CREATE OR REPLACE VIEW :**

Syntax:- CREATE OR REPLACE VIEW *view\_name* AS SELECT *column1, column2, ...*

FROM table\_name WHERE condition;

**c) SQL Dropping a View:** You can delete a view with the DROP VIEW command.

Syntax:- DROP VIEW view\_name;

**3) Index:** Indexes are used to retrieve data from the database very fast. The users cannot see the indexes, they are just used to speed up searches/queries.

a) **CREATE INDEX :-** Creates an index on a table. Duplicate values are allowed:

Syntax:- CREATE INDEX index\_name

ON table\_name (column1, column2, ...);

b) **CREATE UNIQUE INDEX :-**Creates a unique index on a table. Duplicate values are not allowed:

Syntax:- CREATE UNIQUE INDEX index\_name

ON table\_name (column1, column2, ...);

**4) Sequence :-** Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table. Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

Syntax :- Create table < table\_name > (variable\_name datatype primary key auto increment , variable\_name data type);

## **DML (Data Manipulation Language) :-**

### **What is DML ?**

A popular data manipulation language is that of [Structured Query Language](#) (SQL), which is used to retrieve and manipulate [data](#) in a [relational database](#). It is used to manipulate **data itself**. Data manipulation language comprises the SQL data change statements, which modify stored data but not the schema or database objects. For example, with SQL, it would be instructions such as **insert, update, delete, select**.

**Insert:** The INSERT statement is used to insert new records in a table.

Syntax: It is possible to write the INSERT statement in two ways. The first way specifies both the column names and the values to be inserted:

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

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. The INSERT syntax would be as follows:

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

**Select:** The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.

Syntax:

```
SELECT column1, column2, ... FROM table_name;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```

**Update:** The UPDATE statement is used to modify the existing records in a table.

Syntax:

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ... WHERE  
condition;
```

**Delete:** The DELETE statement is used to delete existing records in a table.

Syntax:

```
DELETE FROM table_name WHERE  
condition;
```

### **What is Functions?**

A user-defined function is a Transact-SQL or common language runtime (CLR) routine that accepts parameters, performs an action, such as a complex calculation, and returns the result of that action as a value. The return value can either be a scalar (single) value or a table. SQL Server has many built-in functions. This reference contains the string, numeric, date, conversion, and advanced.

**The SQL Functions are:** MIN(),MAX(),COUNT(),AVG(),SUM() and ORDERBY().

**MIN():**The MIN() function returns the smallest value of the selected column.

Syntax:

```
SELECT MIN(column_name) FROM table_name  
WHERE condition;
```

**MAX():**The MAX() function returns the largest value of the selected column.

Syntax:

```
SELECT MAX(column_name) FROM table_name  
WHERE condition;
```

**COUNT():**The COUNT() function returns the number of rows that matches a specified criteria.

Syntax:

```
SELECT COUNT(column_name) FROM table_name  
WHERE condition;
```

**AVG():**The AVG() function returns the average value of a numeric column.

Syntax:

```
SELECT AVG(column_name) FROM table_name  
WHERE condition;
```

**SUM():**The SUM() function returns the total sum of a numeric column.

Syntax:

```
SELECT SUM(column_name) FROM table_name  
WHERE condition;
```

**ORDERBY():** 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;
```

### **What is operators?**

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.

**The Operators are:** AND, OR, NOT, BETWEEN, LESS THAN, GREATER THAN.

**AND:** The WHERE clause can be combined with AND operators. The AND operators are used to filter records based on more than one condition. The AND operator displays a record if all the conditions separated by AND is TRUE.

Syntax:

```
SELECT column1, column2, ... FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

**OR:** The WHERE clause can be combined with OR operators. The OR operators are used to filter records based on more than one condition: The OR operator displays a record if any of the conditions separated by OR is TRUE.

Syntax:

```
SELECT column1, column2, ... FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

**NOT:**The WHERE clause can be combined with AND, OR, and NOT operators. The NOT operator displays a record if the condition(s) is NOT TRUE.

Syntax:

```
SELECT column1, column2, ... FROM table_name  
WHERE NOT condition;
```

**BETWEEN:** The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates. The BETWEEN operator is inclusive: begin and end values are included.

Syntax:

```
SELECT column_name(s) FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

**LESS THAN:** Less than operator is used to find out less value from table. We can use WHERE clause.

Syntax:

SELECT \* from table\_name WHERE condition;

**GREATER THAN:** Greater than operator is used to find out greater value from table. We can use WHERE clause.

Syntax:

SELECT \* from table\_name WHERE condition;

**Conclusion :-** Here we understood the DDL & DML Commands and its operations with Operators, functions.