**RollNo.**

**Name:**

**Batch:**

**ASSIGNMENT 1**

**AIM:-**

Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index and Sequence, also implement SQL DML statements: Insert, Select, Update, Delete with operators, functions, Set operators, Clauses.

**OBJECTIVE:-**

To implement DDL & DML operations.

**THEORY:-**

group condition restricts the groups of rows returned to those groups for which the specified condition is true.

**LIKE OPERATOR**

The LIKE operator is used to list all rows in a table whose column values match a specified pattern. It is useful when you want to search rows to match a specific pattern, or when you do not know the entire value. For this purpose, we use a character '%'.

WEBSITE:- <https://www.w3schools.com/sql/sql_like.asp>

**BETWEEN.. AND OPERATOR**

The operator BETWEEN and AND, are used to compare data for a range of values.

WEBSITE:-AND: <https://www.w3schools.com/sql/sql_and.asp>

:-BETWEEN: <https://www.w3schools.com/sql/sql_between.asp>

**SET OPERATOR**

SQL set operators combine results from two or more SELECT statements.

SQL set operators combine rows from different queries with strong preconditions - all involved SELECTS must.

They retrieve the same number of columns and the data types of corresponding columns.

UNION :- <https://www.w3schools.com/sql/sql_union.asp>

UNION ALL

INTERSECT:-https://www.simplilearn.com/tutorials/sql-tutorial/sql-intersect#:~:text=SQL%20INTERSECT%20operator%20combines%20two,in%20both%20A%20and%20B.

INTERSECT ALL

EXCEPT :- <https://www.tutorialspoint.com/sql/sql-except-clause.htm>

EXCEPT ALL

**DDL Statements:**

Table, View, Index and Sequence

**E.g.**

**TABLE**

**CREATE:-** [**https://www.w3schools.com/sql/sql\_create\_table.asp**](https://www.w3schools.com/sql/sql_create_table.asp)

Create table person (P\_ID int(10),Last\_Name Varchar(15), First\_Name varchar(15), Address varchar(20),City varchar(10));

Select \* from person ;

**ALTER:-** [**https://www.w3schools.com/sql/sql\_alter.asp**](https://www.w3schools.com/sql/sql_alter.asp)

ALTER TABLE PERSON ADD DESIGNATION VARCHAR(20);

**DROP :-** [**https://www.w3schools.com/sql/sql\_drop\_table.asp**](https://www.w3schools.com/sql/sql_drop_table.asp)

DROP TABLE PERSON;

**TRUNCATE**

TRUNCATE TABLE PERSON;

**RENAME:-** [**https://www.javatpoint.com/sql-rename-table**](https://www.javatpoint.com/sql-rename-table)

RENAME TABLE PERSON AS P1;

**VIEW:-** **https://www.w3schools.com/sql/sql\_view.asp**

CREATE VIEW PERSON\_ORDER

AS

SELECT First\_Name,Last\_Name,City

From PERSON;

**INDEX**

CREATE INDEX ID

ON PERSON (P\_ID);

**DROP INDEX:-** [**https://www.w3schools.com/sql/sql\_create\_index.asp**](https://www.w3schools.com/sql/sql_create_index.asp)

ALTER TABLE PER**Data Definition Language (DDL):**

DDL is abbreviation of Data Definition Language. It is used to create and modify the structure of database objects in database.

Examples: CREATE, ALTER, DROP statements.

**CREATE**

The CREATE TABLE Statement is used to create tables to store data.

Syntax for the CREATE TABLE Statement is:

CREATE TABLE table\_name   
(column\_name1 datatype,   
column\_name2 datatype,   
... column\_nameN datatype   
);

**DROP**

The SQL DROP command is used to remove an object from the database. If you drop a table, all the rows in the table is deleted and the table structure is removed from the database.

Syntax

DROP TABLE table\_name;

**ALTER**

The SQL ALTER TABLE command is used to modify the definition (structure) of a table by modifying the definition of its columns.

The ALTER command is used to perform the following functions.

1. Add, drop, modify table columns   
   2) Add and drop constraints   
   3) Enable and Disable constraints

**RENAME**

The SQL RENAME command is used to change the name of the table or a database object.

If you change the object's name any reference to the old name will be affected. You have to manually change the old name to the new name in every reference.

Syntax to rename a table

RENAME TABLE old\_table\_name To new\_table\_name;

**TRUNCATE**

The SQL TRUNCATE command is used to delete all the rows from the table and free the space containing the table.

Syntax

TRUNCATE TABLE table\_name;

**VIEW**

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain condition.

The Syntax to create a sql view is

CREATE VIEW view\_name   
AS   
SELECT column\_list   
FROM table\_name [WHERE condition];

SQL Dropping a View

You can delete a view with the DROP VIEW command.

Syntax:

DROP VIEW view\_name

**AUTO-INCREMENT:-** [**https://www.w3schools.com/sql/sql\_autoincrement.asp**](https://www.w3schools.com/sql/sql_autoincrement.asp)

My SQL uses the AUTO\_INCREMENT keyword to perform an auto-increment feature.

By default, the starting value for AUTO\_INCREMENT is 1, and it will increment by 1 for each new record.

To let the AUTO\_INCREMENT sequence start with another value, use the following SQL statement:

ALTER TABLE TABLE\_NAME AUTO\_INCREMENT=NUMBER

**INDEX**

Index in sql is created on existing tables to retrieve the rows quickly.

When an index is created, it first sorts the data and then it assigns a ROWID for each row.

Syntax to create Index:

CREATE INDEX index\_name   
ON table\_name (column\_name1,column\_name2...);

DROP INDEX

An index can be dropped using SQL DROP command.

Care should be taken when dropping an index because performance may be slowed or improved.

The basic syntax is as follows:

DROP INDEX index\_name;

**SYNONYM**

We can simplify access to objects by creating a synonym (another name for an object).

Syntax:

CREATE [PUBLIC] SYNONYM synonym

FOR object;

Where,

PUBLIC: creates synonym accessible to all users

SYNONYM: It is the synonym to be created

object: identifies the object for which the synonym is created.

To refer to a table owned by another user, you need to prefix the table name with the name of the user who created it followed by a period.

Creating a synonym eliminates the need to qualify the object name with the schema and provides you with an alternative name for a table, view, sequence, procedure, or other objects.

**SEQUENCE**

A sequence is a database item that generates a sequence of integers.

Syntax of CREATE SEQUENCE statement:

CREATE SEQUENCE sequence\_name  
[START WITH start\_n  
[INCREMENT BY n]  
[ { MAXVALUE maximum\_n | NOMAXVALUE } ]  
[ { MINVALUE minimum\_n | NOMINVALUE } ]  
[ { CYCLE | NOCYCLE } ]  
[ { CACHE cache\_n | NOCACHE } ]  
[ { ORDER | NOORDER } ];

**Foreign key:-** [**https://www.w3schools.com/sql/sql\_foreignkey.asp**](https://www.w3schools.com/sql/sql_foreignkey.asp)

This constraint identifies any column referencing the PRIMARY KEY in another table. For a column to be defined as a Foreign Key, it should be defined as a Primary Key in the table which it is referring.

One or more columns can be defined as Foreign key.

**Syntax** to define a Foreign key at column level:

[CONSTRAINT constraint\_name] REFERENCES Referenced\_Table\_name(column\_name)

**Primary key:-** [**https://www.w3schools.com/sql/sql\_primarykey.asp**](https://www.w3schools.com/sql/sql_primarykey.asp)

This constraint defines a column or combination of columns which uniquely identifies each row in the table.

Syntax to define a Primary key at column level:

column\_name datatype[CONSTRAINT constraint\_name] PRIMARY KEY

**Check constraint:-** [**https://www.w3schools.com/sql/sql\_constraints.asp**](https://www.w3schools.com/sql/sql_constraints.asp)

The constraint can be applied for a single column or a group of columns.

Syntax to define a Check constraint:

[CONSTRAINT constraint\_name] CHECK (condition)

NOT NULL:- <https://www.w3schools.com/sql/sql_notnull.asp>

**Unique key:-** [**https://www.w3schools.com/sql/sql\_notnull.asp**](https://www.w3schools.com/sql/sql_notnull.asp)

his constraint ensures that a column or a group of columns in each row have a distinct value. A column(s) can have a null value but the values cannot be duplicated.

Syntax to define a Unique key at column level:[CONSTRAINT constraint\_name] UNIQUE

**Data Manipulation Language (DML):**

DML is abbreviation of Data Manipulation Language. It is used to retrieve, store, modify, delete, insert, and update data in database.

Examples: SELECT, UPDATE, INSERT statements

**SELECT**

The most commonly used SQL command is SELECT statement. The SQL SELECT statement is used to retrieve data from a table in the database. A query may retrieve information from specified columns or from all of the columns in the table. To create a simple SQL SELECT Statement, you must specify the column(s) name and the table name. The whole query is called SQL SELECT Statement.

**WHERE**

SQL offers a feature called WHERE clause, which we can use to restrict the data that is retrieved. The condition you provide in the WHERE clause filters the rows retrieved from the table and gives you only those rows which you expected to see. WHERE clause can be used along with SELECT, DELETE, UPDATE statements. The WHERE clause is used when you want to retrieve specific information from a table excluding other irrelevant data.

**INSERT**

The INSERT Statement is used to add new rows of data to a table. Insert data to a table in two ways,

Syntax

INSERT INTO TABLE\_NAME [ (col1, col2, col3,...colN)]   
VALUES (value1, value2, value3,...valueN);

col1, col2,...colN -- the names of the columns in the table into which you want to insert data.

**DELETE**

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

Syntax : DELETE FROM table\_name [WHERE condition];

table\_name -- the table name which has to be updated.

The WHERE clause in the sql delete command is optional and it identifies the rows in the column that gets deleted.

If you do not include the WHERE clause all the rows in the table is deleted, so be careful while writing a DELETE query without WHERE clause.

**UPDATE**

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

Syntax:

UPDATE table\_name   
 SET column\_name1 = value1,   
 column\_name2 = value2, ...   
 [WHERE condition]

table\_name - the table name which has to be updated.

column\_name1, column\_name2.. - the columns that gets changed.

value1, value2... - are the new values.

**GROUP BY CLAUSE**

To divide the rows in a table into groups we can use Group By clause. Then we can use the group functions to return summary information for each group.

Syntax:

SELECT column, group\_function(column)

FROM table

[WHERE condition]

[GROUP BY expression]

Expression specifies columns whose values determine the basis for grouping rows

**HAVING CLAUSE**

Use the having clause to restrict groups:

1) Rows are grouped

2)The group function is applied

3) Groups matching the HAVING clause are displayed.

Syntax:

SELECT column, group\_function

FROM table

[WHERE condition]

[GROUP BY expression]

[HAVING group condition]

[OREDER BY column];

SON

DROP INDEX ID;

**Copy your Queries and Output Snapshots:**

**CREATE TABLE**

**# Create Command :**

**# Create table :**

**# Insert Command :**

**# Select Command :**

**# Update Command :** [**https://www.w3schools.com/sql/sql\_update.asp**](https://www.w3schools.com/sql/sql_update.asp)

**# Delete Command :**

**# Alter command for adding column:**

**# Drop Command :**

**# Rename Command :**