**Assignment No:1**

**Title:** Implementation of SQL DDL statements such as Create, Alter, Drop, Rename, Truncate, and Commit.

**Problem Statement:** Craft and implement SQL DDL statements that demonstrate the use of various SQL objects, such as creating tables with defined schemas, constructing views for specific queries, setting up indexes for performance optimization, generating sequences for unique identifier generation, and establishing synonyms for alternative naming of database objects.

**Objective:** • To acquire knowledge of database query languages.

**Outcome:** Implement different SQL and PLSQL operations using suitable databases.

**Tools Required:** Ubuntu OS, Mysql.

**Theory:**

**1. Introduction:**

**1.1 What is Database?**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems. So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational DataBase Management System (RDBMS)** is a software that:

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

**1.2 RDBMS Terminology:**

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:** A database is a collection of tables, with related data.
* **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Primary Key:** A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:** A foreign key is the linking pin between two tables.
* **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:** An index in a database resembles an index at the back of a book.
* **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

**2. DDL, DML, DCL, TCL**

**2.1 DDL: Data Definition Language** (DDL) statements are used to define the database structure or schema. Some examples:

* CREATE - to create objects in the database
* ALTER - alters the structure of the database
* DROP - delete objects from the database
* TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
* RENAME - rename an object

**2.2 DML: Data Manipulation Language** (DML) statements are used for managing data within schema objects. Some examples:

* SELECT - retrieve data from the a database
* INSERT - insert data into a table
* UPDATE - updates existing data within a table
* DELETE - deletes all records from a table, the space for the records remain

**2.3 DCL: Data Control Language** (DCL) statements. Some examples:

* GRANT - gives user's access privileges to database
* REVOKE - withdraw access privileges given with the GRANT command

**2.4 TCL: Transaction Control** (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

* COMMIT - save work done
* SAVEPOINT - identify a point in a transaction to which you can later roll back
* ROLLBACK - restore database to original since the last COMMIT
* SET TRANSACTION - Change transaction options like isolation level and what rollback segment to use

**3. Tables**

**3.1 Create MySQL Tables**

The CREATE TABLE statement is used to create a table in a database.

|  |  |
| --- | --- |
| **Syntax:**  CREATE TABLE table\_name ( column\_name1 data\_type(size), column\_name2 data\_type(size), column\_name3 data\_type(size), .... ); | **Example:**  CREATE TABLE Persons (  PersonID int, LastName varchar(255), FirstName varchar(255), Address varchar(255),  City varchar(255) ); |

**3.2 ALTER Table**

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

|  |  |
| --- | --- |
| **Syntax:**  ALTER TABLE table\_name ADD column\_name datatype; | **Example:**  ALTER TABLE Persons ADD DateOfBirth date; |
| ALTER TABLE table\_name DROP COLUMN column\_name; | ALTER TABLE Persons DROP COLUMN DateOfBirth; |
| ALTER TABLE table\_name MODIFY COLUMN column\_name datatype; | ALTER TABLE Persons MODIFY COLUMN DateOfBirth int; |

**3.3 DROP Table**

The DROP TABLE Statement is used to drop a table.

|  |  |
| --- | --- |
| **Syntax:**  DROP TABLE table\_name; | **Example:**  DROP TABLE Persons; |

**3.4 TRUNCATE Table**

The TRUNCATE TABLE Statement is used to truncate (delete all rows) a table.

|  |  |
| --- | --- |
| **Syntax:**  TRUNCATE TABLE table\_name; | **Example:**  TRUNCATE TABLE Persons; |

**4. 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.

|  |
| --- |
| **Syntax:**  CREATE VIEW view\_name AS  SELECT column\_name(s)  FROM table\_name  WHERE condition; |

**4.1 SQL Dropping a View**

You can delete a view with the DROP VIEW command.

|  |
| --- |
| **Syntax:**  DROP VIEW view\_name; |

**5. CREATE INDEX**

Index in SQL is created on existing tables to retrieve the rows quickly. When there are thousands of records in a table, retrieving information will take a long time. Therefore indexes are created on columns which are accessed frequently, so that the information can be retrieved quickly. Indexes can be created on a single column or a group of columns. When a index is created, it first sorts the data and then it assigns a ROWID for each row.

|  |
| --- |
| **Syntax:**  CREATE INDEX index\_name  ON table\_name (column\_name1,column\_name2...); |

**5.1 DROP INDEX**

|  |
| --- |
| **Syntax:**  DROP INDEX index\_name; |

**6. CREATE SYNONYM**

Use the CREATE SYNONYM statement to create a synonym, which is an alternative name for a table, view, sequence, procedure, stored function, package, materialized view. Synonyms provide both data independence and location transparency. Synonyms permit applications to function without modification regardless of which user owns the table or view and regardless of which database holds the table or view.

You can refer to synonyms in the following DML statements: SELECT, INSERT, UPDATE, DELETE.

|  |
| --- |
| **Syntax:**  Create synonym synonym\_name for table\_name; |

**Conclusion:**

We have successfully implemented SQL DDL statements which demonstrate the use of SQL objects such as Create, Alter, Drop, Rename, Truncate, and Commit..