<https://www.tutorialspoint.com/sql/sql-drop-table.html>

Types of SQL Databases:

Relational databases are used to store and manage the data objects that are related to one another, i.e. the process of handling data in a relational database is done based on a relational model.( structured way) (using tables). A system used to manage these relational databases is known as Relational Database Management System (RDBMS).

There are many popular RDBMS available:

MySQL MS SQL Server

ORACLE MS ACCESS

PostgreSQL SQLite

For create a database/ schema

CREATE DATABASE / SCHEMA database\_name;

CREATE DATABASE sampleDB

USE database\_name;

USE sampleDB;

DROP DATABASE database\_name;

DROP DATABASE sampleDB;

CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

**for** description table or table info

DESC CUSTOMERS;

To insert data

INSERT INTO CUSTOMERS VALUES

(1, 'Ramesh', 32, 'Ahmedabad', 2000.00 ),

(2, 'Khilan', 25, 'Delhi', 1500),

(3, 'kaushik', 23, 'Kota', 2000),

(4, 'Chaitali', 25, 'Mumbai', 6500),

(5, 'Hardik', 27, 'Bhopal', 8500),

(6, 'Komal', 22, 'Hyderabad', 4500),

(7, 'Muffy', 24, 'Indore', 10000);

Update data in table

UPDATE CUSTOMERS SET ADDRESS = 'Pune' WHERE ID = 6;

Change table name column name with datatype

ALTER TABLE table\_name

{ADD|DROP|MODIFY} column\_name {data\_type};

All records delete but table structure as it is

TRUNCATE TABLE CUSTOMERS;

Changes table column\_name, modify and drop

ALTER TABLE CUSTOMERS

ADD|DROP|MODIFY

Add

city varchar(10);

change table name:

ALTER TABLE table\_name RENAME TO new\_table\_name;

SQL types of operators:

Arithmetic operators

Comparison operators

Logical operators

Operators used to negate conditions

Arithmetic operators

SQL Arithmetic Operators are used to perform mathematical operations on the numerical values.

+,-,/,\*, and

% modules 5%2=1

ALL:- TRUE if all of a set of comparisons are TRUE.

AND :- TRUE if all the conditions separated by AND are TRUE.

ANY: TRUE if all of a set of comparisons are TRUE.

BETWEEN: TRUE if the operand lies within the range of comparisons.

EXISTS: TRUE if the subquery returns one or more records

IN: TRUE if the operand is equal to one of a list of expressions.

LIKE: TRUE if the operand matches a pattern specially with wildcard.

NOT: Reverses the value of any other Boolean operator.

OR: TRUE if any of the conditions separated by OR is TRUE

IS NULL: TRUE if the expression value is NULL.

SOME: TRUE if some of a set of comparisons are TRUE.

UNIQUE: The UNIQUE operator searches every row of a specified table for uniqueness (no duplicates).

What is SQL Expression?

An SQL expression is a combination of one or more values, operators and SQL functions that are all evaluated to a value. These SQL EXPRESSION(s) are like formulae and they are written in query language.

Expressions are used in WHERE clause of an SQL query.

* Boolean Expressions
* Numeric Expressions
* Date and time Expressions

Boolean Expressions that check for equality of two values using SQL comparison operators. Here, equality of these values is a condition.

Boolean Expressions can also contain one value paired with an SQL logical operator. In this case, the logic specified acts like a condition.

select count(\*) from customers;

select count(\*)+1 as total\_members from customers;

select sum(id) as total from customers;

SELECT CURRENT\_TIMESTAMP;

In sql can we connect one schema to another schema

Yes, in SQL, it's possible to communicate or reference objects (tables, views, procedures, etc.) from one schema to another schema within the same database or even across databases, depending on the database management system (DBMS) we're using.

Here are some common ways to reference objects between schemas:

Within the Same Database:

Using Fully Qualified Names:

Objects can be referenced using their fully qualified names, including the schema name.

For example:

sql

Copy code

SELECT \* FROM schema\_name.table\_name;

Changing the Default Schema:

Some databases allow changing the **default** schema **for** a user's session. This way, when an object name is used without a schema qualifier, the database system uses the default schema.

Syntax may vary across different database systems. For instance:

PostgreSQL: SET search\_path TO schema\_name;

SQL Server: ALTER USER username SET DEFAULT\_SCHEMA = schema\_name;

Across Different Databases:

Using Linked Servers (in some DBMS like SQL Server):

Linked Servers allow querying and modifying data from another database or instance within the same query.

Example in SQL Server:

sql

Copy code

SELECT \* FROM linked\_server\_name.schema\_name.table\_name;

Using Database Links (in Oracle):

Oracle Database Links enable communication between two databases.

Example in Oracle:

sql

Copy code

SELECT \* FROM schema\_name.table\_name@database\_link\_name;

When referencing objects between schemas or databases, ensure proper permissions and access rights are granted to the user executing these operations. The syntax and methods can vary between different database systems, so refer to the specific documentation **for** your database management system **for** detailed instructions and specific functionalities.

RENAME DATABASE OldDatabaseName TO NewDatabaseName;

ALTER TABLE BUYERS RENAME TO CUSTOMERS;

TRUNCATE a table completely in one go instead of deleting table records one by one

The SQL **TRUNCATE TABLE** command is used to empty a table.

And table structure as it is.

TRUNCATE TABLE table\_name;

DROP TABLE command to delete a table but it will remove the complete table structure from the database

|  |  |
| --- | --- |
| **DELETE** | **TRUNCATE** |
| The DELETE command in SQL removes one or more rows from a table based on the conditions specified in a WHERE Clause. | SQL's TRUNCATE command is used to remove all of the rows from a table, regardless of whether or not any conditions are met. |
| It is a DML(Data Manipulation Language) command. | It is a DDL(Data Definition Language) command. |
| There is a need to make a manual COMMIT after making changes to the DELETE command, for the modifications to be committed. | When you use the TRUNCATE command, the modifications made to the table are committed automatically. |
| It deletes rows one at a time and applies same criteria to each deletion. | It removes all of the information in one go. |
| The WHERE clause serves as the condition in this case. | The WHERE Clause is not available. |
| All rows are locked after deletion. | TRUNCATE utilizes a table lock, which locks the pages so they cannot be deleted. |
| It makes a record of each and every transaction in the log file. | The only activity recorded is the deallocation of the pages on which the data is stored. |
| It consumes a greater amount of transaction space compared to TRUNCATE command. | It takes comparatively less amount of transaction space. |
| If there is an identity column, the table identity is not reset to the value it had when the table was created. | It returns the table identity to a value it was given as a seed. |
| It requires authorization to delete. | It requires table alter permission. |
| When it comes to large databases, it is much slower. | It is much faster. |
| DROP | TRUNCATE |
| The DROP command in SQL removes an entire table from a database including its definition, indexes, constraints, data etc. | The TRUNCATE command is used to remove all of the rows from a table, regardless of whether or not any conditions are met and resets the table definition. |
| It is a DDL(Data Definition Language) command. | It is also a DDL(Data Definition Language) command. |
| The table space is completely freed from the memory. | The table still exists in the memory. |
| All the integrity constraints are removed. | The integrity constraints still exist in the table. |
| Requires ALTER and CONTROL permissions on the table schema and table respectively, to be able to perform this command. | Only requires the ALTER permissions to truncate the table. |
| DROP command is much slower than TRUNCATE but faster than DELETE. | TRUNCATE command is faster than both DROP and DELETE commands. |

SQL **Cloning Operation** allows to create the exact copy of an existing table along with its definition. There are three types of cloning

* Simple Cloning
* Shallow Cloning
* Deep Cloning

Simple cloning operation creates a new replica table from the existing table and copies all the records in newly created table.

CREATE TABLE new\_table SELECT \* FROM original\_table;

Shallow cloning operation creates a new replica table from the existing table but does not copy any data records into newly created table, so only new but empty table is created.

CREATE TABLE NEW\_TABLE LIKE ORIGINAL\_TABLE;

Deep cloning operation is a combination of simple cloning and shallow cloning. It not only copies the structure of the existing table but also its data into the newly created table.

CREATE TABLE NEW\_TABLE LIKE ORIGINAL\_TABLE;

INSERT INTO NEW\_TABLE SELECT \* FROM ORIGINAL\_TABLE;

Temporary tables are pretty much what their name describes: they are the tables which are created in a database to store temporary data. We can perform SQL operations similar to the operations on permanent tables like CREATE, UPDATE, DELETE, INSERT, JOIN, etc. But these tables will be automatically deleted once the current client session is terminated.

CREATE TEMPORARY TABLE CUSTOMERS (

ID INT NOT NULL,

NAME VARCHAR(20) NOT NULL,

AGE INT NOT NULL,

PRIMARY KEY(ID) );

Though all the temporary tables are deleted by MySQL when your database connection gets terminated

DROP TEMPORARY TABLE CUSTOMERS;

The SQL **ALTER TABLE** command is a part of Data Definition Language (DDL) and modifies the structure of a table. The ALTER TABLE command can add or delete columns, create or destroy indexes, change the type of existing columns, or rename columns or the table itself.

add a new column to a table:

ALTER TABLE TABLE\_NAME ADD COLUMN\_NAME datatypes;

Drop column:

ALTER TABLE TABLE\_NAME DROP COLUMN COLUMN\_NAME;

ADD PRIMARY KEY:

ALTER TABLE TABLE\_NAME

ADD CONSTRAINT constraint\_name

PRIMARY KEY(COLUMN1, COLUMN2…);

DROP PRIMARY KEY:

ALTER TABLE TABLE\_NAME DROP PRIMARY\_KEY;

ADD CONSTRAINTS KEY:

ALTER TABLE table\_name

ADD CONSTRAINT constraint\_name

UNIQUE (COLUMN1, COLUMN2…);

DROP CONSTRAINTS KEY:

ALTER TABLE table\_name DROP CONSTRAINT constraint\_name;

RENAME COLUMN NAME:

ALTER TABLE table\_name RENAME COLUMN column\_name TO new\_column\_name;

MODIFY DATATYPES

ALTER TABLE table\_name MODIFY COLUMN column\_name DATATYPE;

The SQL **DROP TABLE** statement is a Data Definition Language (DDL) command that is used to remove a table's definition, and its data, indexes, triggers, constraints and permission specifications (if any).

once a table is DROP then all the information available in that table will also be lost forever.

To drop a table in a database, one must require ALTER permission on the said table and CONTROL permissions on the table schema.

Even though it is a data definition language command, it is different from TRUNCATE TABLE statement as the DROP statement completely frees the table from the memory.

DROP TABLE causes an implicit commit, except when used with the TEMPORARY keyword.

IF EXISTS CLAUSE

DROP TABLE IF EXISTS TABLE\_NAME;

The **SQL DELETE** is a command of Data Manipulation Language (DML), so it does not delete or modify the table structure but it delete only the data contained within the table.

The **SQL DELETE TABLE** statement is used to delete the existing records from a table in a database.

DELETE FROM table\_name WHERE condition;

We can use the **SQL DELETE TABLE** statement without a WHERE clause to delete all records in a table in SQL. This statement will remove all the rows from the specified table, effectively resetting the table to its original state

DELETE FROM table\_name;