**Topics**

**1) Keys**

1. Primary Key :
2. Candidate Key
3. Super Key
4. Alternate key
5. Composite or Compound Key
6. Unique Key
7. Foreign Key

**2) DDL**

Create , Alter , Drop , Truncate , Rename

**3) CRUD**

a) Insert

b) Read ( Select )

c) Update ( Update )

d) Delete

-------------------------------------------

1) **Keys ( Columns )**

**Primary Key**

The primary key is the most important key in the database. There can be only one primary key in a table.

It will not accept duplicate or null values. The primary key contains unique values.

**Candidate Key**

A candidate key is a set of one or multiple columns in a database table. It can identify a record uniquely just like a primary key. These are other unique columns that can become a primary key.

There can be any number of candidate keys that can be used in place of the primary key if required.

**Super Key**

A super key is a set of one or more keys that are used to identify data or records uniquely in a database table. It includes only those fields that have unique values.

**Alternate key**

The alternate key can be an alternative or a candidate for the primary key when needed but it is not the primary key. An alternate key is a function of all candidate keys except the primary key.

**Composite or Compound Key**

This is a combination of one or more columns that can uniquely identify the records in a table. The composite key can be a combination of primary and candidate keys.

**Unique Key**

The unique key is a set of one or more columns or fields of a table that can uniquely identify a record in the table. Other than the primary key there can also be other unique fields in a table.

The unique key cannot have duplicate values and can accept only one null value.

**Foreign Key**

A foreign key can be a common key in two database tables. Suppose a Company table where it has a column Employee\_Id which is also present in the Employee table in which it is the primary key.

Using a foreign key we can identify records from multiple tables. It accepts duplicate values as well as null values.

Foreign key also helps you to reduce data redundancy.

**2) DDL**

1. CREATE Command
2. DROP Command
3. ALTER Command
4. TRUNCATE Command
5. RENAME Command

**Create**

CREATE is a DDL command used to create databases, tables, triggers and other database objects.

Example : **CREATE** **Database** Database\_Name;

Create Table :

CREATE TABLE table\_name

(

column\_Name1 data\_type ( size of the column ) constraints ,

column\_Name2 data\_type ( size of the column) constraints,

column\_Name3 data\_type ( size of the column) constraints,

column\_NameN data\_type ( size of the column ) constraints

) ;

**Constraint Details**

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

FOREIGN KEY - Prevents actions that would destroy links between tables

CHECK - Ensures that the values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column if no value is specified

CREATE INDEX - Used to create and retrieve data from the database very quickly

**DROP Command**

DROP is a DDL command used to delete/remove the database objects from the SQL database. We can easily remove the entire table, view, or index from the database using this DDL command.

**DROP** **DATABASE** Database\_Name;

**ALTER Command**

ALTER is a DDL command which changes or modifies the existing structure of the database, and it also changes the schema of database objects.

We can also add and drop constraints of the table using the ALTER command.

Example : **ALTER** **TABLE** name\_of\_table **ADD** column\_name column\_definition;

ALTER TABLE Student **ADD** Father's\_Name Varchar(60); // Add Column

ALTER TABLE name\_of\_table **DROP** Column\_Name\_1 // Drop Column

ALTER TABLE table\_name **MODIFY** ( column\_name column\_datatype(size)); // Modify constraints

**TRUNCATE Command**

TRUNCATE is another DDL command which deletes or removes all the records from the table.

This command also removes the space allocated for storing the table records.

TRUNCATE TABLE Table\_Name;

**RENAME Command**

RENAME is a DDL command which is used to change the name of the database table.

Example :

RENAME TABLE Old\_Table\_Name TO New\_Table\_Name;

**CRUD**

1) Create ( Insert )

a) **from self data** : INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

b)**Insert values into all columns :** INSERT INTO table\_name VALUES (value1, value2, value3, ...);

c) **Insert from Another table** : INSERT INTO table2 SELECT <> FROM table1 WHERE condition;

2) (Read) **Select**

a) Simple Select : Select \* from <tableName> ;

i) TOP : SELECT TOP number|percent column\_name(s) FROM table\_name WHERE condition;

ii) LIMIT / ROW NUMBER :

SELECT column\_name(s) FROM table\_name WHERE condition LIMIT number;

SELECT column\_name(s) FROM table\_name ORDER BY column\_name(s) FETCH FIRST number ROWS ONLY;

iii) Aggregate :

a) Min() and Max() : SELECT MIN/MAX(column\_name) FROM table\_name WHERE condition;

b) Sum() , Avg() , Count() :

SELECT AVG/Count/Sum(Price<ColumnName>) FROM Products;

b) Where clause : Select \* from <tableName> where condition1 ;

**Multiple Condition** : AND , OR , NOT .

**Ordering** : SELECT column1, column2, FROM table\_name

ORDER BY column1, column2 ASC|DESC;

**Partial or Range Match** ( Like , In , between )

Like : SELECT column1, column2, FROM table\_name WHERE column LIKE pattern; // pattern : % -> Multiple Chars , \_ ,? -> Single chars

In : SELECT column\_name(s) FROM table\_name WHERE column\_name IN (value1, value2);

Between : SELECT column\_name(s) FROM table\_name WHERE column\_name Between (value1, value2);

c) **Group By and Having**

The GROUP BY Statement in SQL is used to arrange identical data into groups with the help of some functions.

GROUP BY clause is used with the SELECT statement.

In the query, the GROUP BY clause is placed after the WHERE clause.

In the query, the GROUP BY clause is placed before the ORDER BY clause if used.

In the query, the Group BY clause is placed before the Having clause.

Place condition in the having clause.

The Select Should have a function .

Ex: SELECT NAME, SUM(SALARY) FROM emp GROUP BY NAME;

Having clause is used to filter data after group by

SELECT col\_1, function\_name(col\_2)FROM tablename WHERE condition GROUP BY column1, column2 HAVING Condition ORDER BY column1, column2;

**SQL Structure :**

**Select** <columns, functions> **from** <TableName> <Alias> **where**

<condition> **GROUP BY** < columns> **HAVING** < Condition> ORDER BY

<columns> **LIMIT/ FETCH FIRST** <Count> ;

3) **Update**

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

UPDATE table\_name **SET** column1 = value1, column2 = value2 WHERE condition;

4) **Delete**

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

DELETE FROM table\_name WHERE condition;

Null Values :

A field with a NULL value is a field with no value.

a) is Null

b) is NotNull

SELECT column\_names FROM table\_name WHERE column\_name IS NULL/ IS NOT NULL;

Tip : == Null will give Incorrect results . s

**Question** : Difference between truncate , Drop and Delete .

Links : Online Editor : [https://www.programiz.com/sql/online-compiler/#google\_vignette](https://www.programiz.com/sql/online-compiler/" \l "google_vignette)

https://sqliteonline.com/