

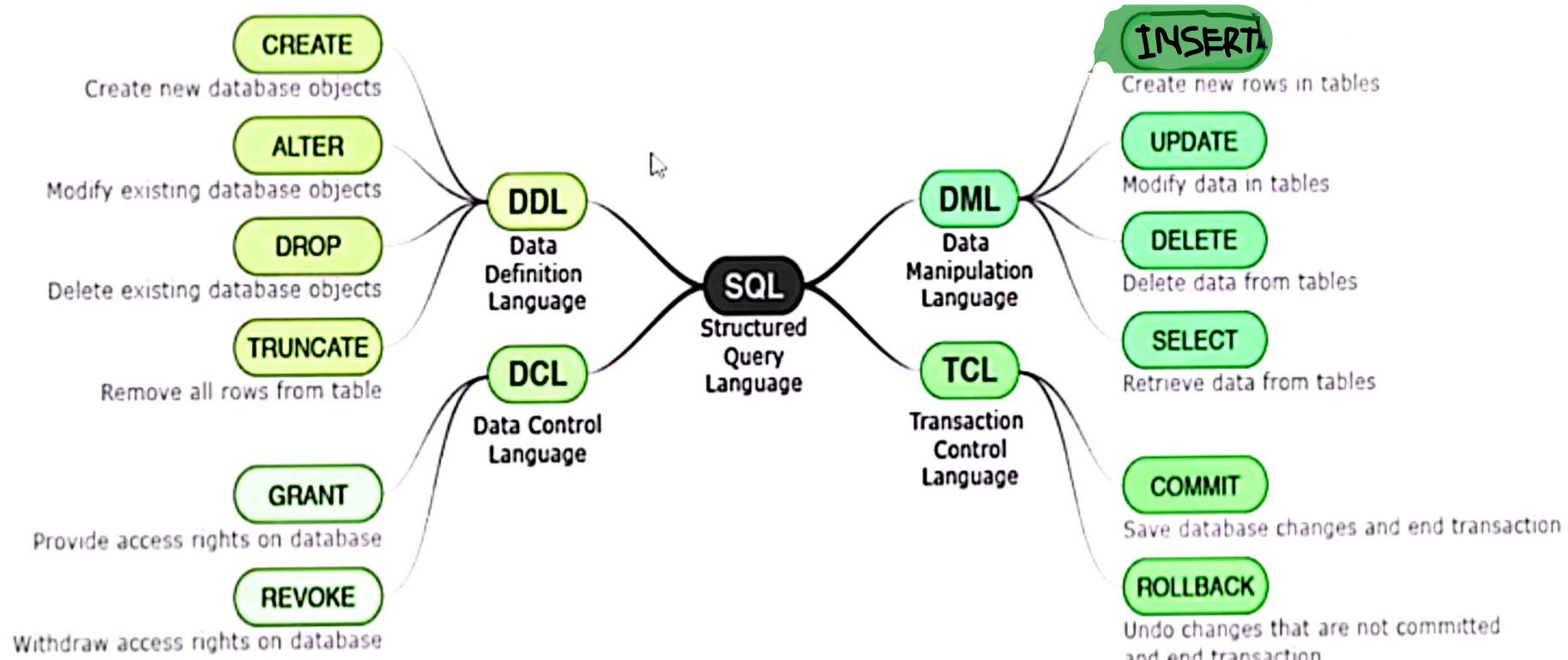
# What is SQL?

08 February 2023 14:56

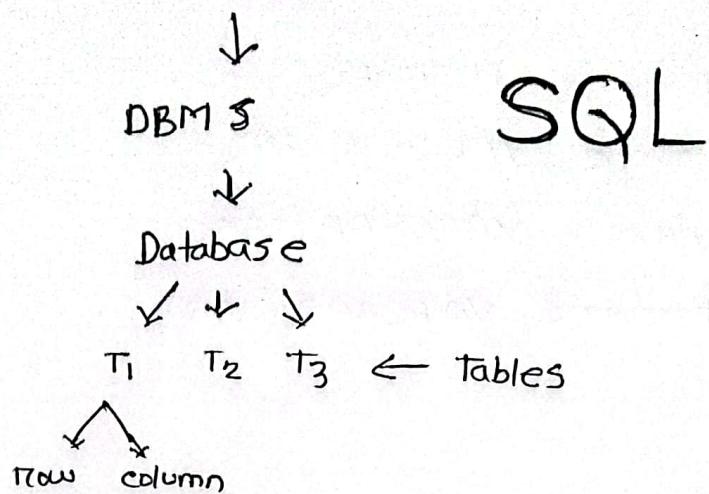
- SQL (Structured Query Language) is a programming language used for managing and manipulating data in relational databases. It allows you to insert, update, retrieve, and delete data in a database. It is widely used for data management in many applications, websites, and businesses. In simple terms, SQL is used to communicate with and control databases.

# Types of SQL commands

08 February 2023 14:56



Database Server



### \* DDL commands \*

DDL = Data definition language.

For DDL command for database.

\* For create database

SQL = CREATE DATABASE campus x

= CREATE " IF NOT EXIST campus x (better)

\* For delete database

SQL = DROP DATABASE campus x

" " IF EXIST campus x (better) .

For DDL command for table:

④ Create: we create a table for user.

| userid | name             | email   | password |
|--------|------------------|---------|----------|
| int    | varchar<br>[256] | varchar | varchar  |

SQL: CREATE TABLE users(  
col-name data-type,  
" " " "  
> " " " "

Always my column data will lower letter

```
SQL = CREATE TABLE users(  
    user_id INTEGER,  
    name VARCHAR(255),  
    email VARCHAR(255)  
    password VARCHAR(255)  
)
```

④ Table is created but null rows.

So now we will insert some value:-

SQL → Insert

⑤ Empty table (Empty)

SQL: TRUNCATE TABLE ~~IF EXIST~~ users.

⑥ DROP table

SQL: DROP TABLE IF EXIST users

# Constraints in MySQL

08 February 2023 15:01

....

Constraints in databases are rules or conditions that must be met for data to be inserted, updated, or deleted in a database table. They are used to enforce the integrity of the data stored in a database and to prevent data from becoming inconsistent or corrupted.

I

1. NOT NULL
2. UNIQUE(combo)  
-> Another way of creating constraint
3. PRIMARY KEY
4. AUTO INCREMENT
5. CHECK
6. DEFAULT
7. FOREIGN KEY

## Referential Actions

1. RESTRICT
2. CASCADE
3. SET NULL
4. SET DEFAULT

## ④ Constraint in MySQL:

① NOT NULL → Train ticket sitnumber.

② UNIQUE → email must unique

③ Primary key → which is most unique column

④ Auto increment →

|  | id |
|--|----|
|  | 1  |
|  | 2  |
|  | 3  |
|  | 4  |
|  | .  |

suppose udemy course  
auto increment  
student id with  
subscription.

⑤ CHECK -

⑥ DEFAULT -

⑦ FOREIGN KEY - relation make.

① NOT NULL:

SQL: CREATE TABLE users(

userid INTEGER NOT NULL,

name VARCHAR(255) NOT NULL,

email VARCHAR(255),

password VARCHAR(255),

)

Try insert value one time

again insert all null the see bouse

## ② UNIQUE

SQL: CREATE TABLE user(

    user\_id INTEGER NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL)

Another way.

In this writing if i try to unique two  
combo column not possible have  
to use ↓

CREATE TABLE user(

    user\_id INTEGER NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) " " ,  
    password " " " ,

    CONSTRAINT users\_email\_unique UNIQUE  
        (name, email)

### ③ Primary key creating.

SQL: CREATE TABLE users (

user\_id INTEGER NOT NULL PRIMARY KEY,

name VARCHAR(255) " ",

email " ",

password " ",

CONSTRAINT user\_email\_unique UNIQUE (name, email)

)

Another way

CONSTRAINT users\_pk PRIMARY KEY (user\_id)

↑  
use this approach if one column can't be  
primary key i have to use two column then.

### ④ AUTO INCREMENT

CREATE TABLE users (

user\_id INTEGER PRIMARY KEY AUTO\_INCREMENT

name VARCHAR(255) NOT NULL,

email " ",

Pass " ",

NOT NULL,

### ⑤ CHECK :

SQL: CREATE TABLE students (

student-id INTEGER PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(255) NOT NULL,

age INTEGER CHECK (age > 6 AND age < 25)

)

ON CONSTRAINT students-age-check CHECK( )

\* TRY by insert

### ⑥ DEFAULT:

SQL: CREATE TABLE ticket (

ticket-id INTEGER PRIMARY KEY,

name VARCHAR(255) NOT NULL,

travel\_date DATETIME DEFAULT CURRENT\_TIMESTAMP

)

## ⑤ FOREIGN KEY

SQL

```
CREATE TABLE customers (
    cid INTEGER PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) .. " UNIQUE",
)
```



```
CREATE TABLE orders (
```

student  
order\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,  
customer\_id INTEGER NOT NULL

order\_rate order\_fk FOREIGN KEY (cid) REFERENCES  
customers (cid),

↑  
Retrict mode

\* CASCADE - The References customers row is

deleted then all his sub Foreign key will  
delete.

- And similarly if update in main key  
then foreign key will also update by value.

\* SET NULL - customer\_id deleted the hi's foreign  
key order value will null.

(\*) Set Default; customer\_id set default.

## \* CASCADE

```
CREATE TABLE orders(
    orden_id INTEGER PRIMARY KEY,
    cid INTEGER NOT NULL,
    orden_date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT orden_fk FOREIGN KEY (cid)
        REFERENCES customers(cid)
    ON DELETE CASCADE,
    ON UPDATE CASCADE)
```

\* For setnull same as &

ON DELETE SETNULL

" update "

\*

④ ALTER TABLE Command: Adding a new column

SQl:

```
ALTER TABLE customers ADD TABLECOLUMN password VARCHAR(255) NOT NULL.
```

```
ALTER TABLE customers ADD COLUMN surname VARCHAR(255) NOT NULL AFTER name  
(After other only column).
```

ALTER TABLE

# ALTER TABLE command

08 February 2023 15:02

The "**ALTER TABLE**" statement in SQL is used to modify the structure of an existing table. Some of the things that can be done using the ALTER TABLE statement include

1. Add columns
2. Delete columns
3. Modify columns



ALTER TABLE customers

ADD COLUMN pan\_number VARCHAR(255) AFTER surname,

ADD COLUMN joining\_date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP

### ④ Deleting columns.

ALTER TABLE customers DROP COLUMN pan\_table

\* Deleting Multiple rows

ALTER TABLE customers,

DROP COLUMN password,

DROP COLUMN joining\_date.

### ⑤ Modify

ALTER TABLE customers

MODIFY COLUMN surname INTEGER

\* ALTER TABLE customers ADD COLUMN age INTEGER

\* ALTER TABLE customers ADD CONSTRAINT CHECK (age > 13)

~~ALTER TABLE customers MODIFY CONSTRAINT~~

~~customer\_age\_check CHECK (age > 6)~~