

# SQL command

## To create database

Syntax :

```
CREATE DATABASE <DBname>;
```

## To create table

Syntax :

```
CREATE TABLE <tablename> (  
    column1 datatype,  
    .....  
);
```

## To drop database and table

Syntax :

```
DROP DATABASE <DBname>;  
DROP TABLE <tablename>;
```

# Add column

Syntax :

```
ALTER TABLE <tablename> ADD COLUMN  
<columnname> datatype;
```

# Drop column

Syntax :

```
ALTER TABLE <tablename> DROP COLUMN  
<columnname>;
```

# Alter column rename

Syntax :

```
ALTER TABLE <tablename> RENAME  
COLUMN <old_name> TO <new_name>;
```

# Add set default

Syntax :

```
ALTER TABLE <tablename> ALTER COLUMN  
<columnname> SET DEFAULT <value>;
```

@yez Got it

# Add constraint

Syntax :

```
ALTER TABLE <table name> ADD  
CONSTRAINT  constraint_name  
constraint_type;
```

# Drop constraint

Syntax :

```
ALTER TABLE <table name> DROP  
CONSTRAINT  constraint_name;
```

# Add primary key

Syntax :

```
ALTER TABLE <table name> ADD  
CONSTRAINT  constraint_name PRIMARY KEY  
(column_name);
```

# Add foreign key

Syntax :

```
ALTER TABLE <child_table>ADD CONSTRAINT  
constraint_name FOREIGN KEY (<column>)  
REFERENCES parent_table (<column>);
```

## select

Syntax :

```
SELECT * FROM <table_name>;
```

## Built-in method

Syntax :

```
SELECT COUNT(<column_name>),  
SUM(<column_name>),  
AVG(<column_name>),  
MAX(<column_name>),MIN(<column_name>)  
FROM <table_name>;
```

# WHERE

Syntax :

```
SELECT * FROM <table_name> WHERE  
condition;
```

## AND, OR, NOT

Syntax :

```
SELECT * FROM <table_name> WHERE  
column_name1 AND column_name2;
```

```
SELECT * FROM <table_name> WHERE  
column_name1 OR column_name2;
```

```
SELECT * FROM <table_name> WHERE  
NOT column_name;
```

# Insert

Syntax :

```
INSERT INTO <table_name> (column1,  
column2,...) VALUES (<value>, ....);
```

## Insert multi-line

Syntax :

```
INSERT INTO <table_name> (column1,  
column2,...) VALUES (<value>, ....),  
(<value>, ....),  
(<value>, ....);
```

## Update

Syntax :

```
UPDATE <table_name> SET column = value  
WHERE condition;
```

# Truncate

Syntax :

```
TRUNCATE TABLE <table_name>;
```

# Delete

Syntax :

```
DELETE FROM <table_name> WHERE  
condition;
```

# Between

Syntax :

```
SELECT column_name FROM <table_name>  
WHERE cloumn_name BETWEEN value1 AND  
value2;
```

# Alias cloumn

Syntax :

```
SELECT <column_name> AS <alias_name>  
FROM <table_name>;
```

# Alias table

Syntax :

```
SELECT <column_name> FROM <table_name>  
AS <alias_name>;
```

# Order by

Syntax :

```
SELECT <column_name> FROM <table_name>  
ORDER BY <cloumn_name> ASC | DESC;
```



# Group by

Syntax :

```
SELECT <cloumn_name> FROM <table_name>  
WHERE condition GROUP BY <column_name>  
ORDER BY <column_name>;
```

# Having by

Syntax :

```
SELECT <cloumn_name>  
FROM <table_name>  
WHERE condition  
GROUP BY <column_name>  
HAVING condition  
ORDER BY <column_name>;
```

# Union

Syntax :

```
SELECT <column_name> FROM  
<table_name1> UNION  
SELECT <column_name> FROM  
<table_name2>;
```

```
SELECT <column_name> FROM  
<table_name1> UNION ALL  
SELECT <column_name> FROM  
<table_name2>;
```

## Inner join

Syntax :

```
SELECT <c.column_name>,<p.column_name>  
FROM <table_name1> AS c INNER JOIN  
<table_name2> AS p ON <c.column_name> =  
<p.column_name>;
```

# Left join

Syntax :

```
SELECT <c.column_name>,<p.column_name>  
FROM <table_name1> AS c LEFT JOIN  
<table_name2> AS p ON <c.column_name> =  
<p.column_name>;
```

# Right join

Syntax :

```
SELECT <c.column_name>,<p.column_name>  
FROM <table_name1> AS c RIGHT JOIN  
<table_name2> AS p ON <c.column_name> =  
<p.column_name>;
```

# Full join

Syntax :

```
SELECT <c.column_name>,<p.column_name>  
FROM <table_name1> AS c FULL JOIN  
<table_name2> AS p ON <c.column_name> =  
<p.column_name>;
```