

SQL

SELECT

Description: Used to select data from a database table. If `DISTINCT` is used, no duplicate rows are returned.

Syntax (without `DISTINCT`):

```
SELECT column(s)
FROM table;
```

Syntax (with `DISTINCT`):

```
SELECT DISTINCT column(s)
FROM table;
```

WHERE

Description: Used to filter records, returning only those which meet provided conditions.

Syntax:

```
SELECT column(s)
FROM table
WHERE condition(s);
```

Condition types:

- `=, >, >=, <, <=`
- `<>` (not equal)
- `BETWEEN min AND max`
- `LIKE %pattern` (where `%` is a wildcard)
- `LIKE pattern%`
- `LIKE %pattern%`

ORDER BY

Description: Used to sort the result set in ascending (default) or descending order.

Syntax:

```
SELECT column(s)
FROM table
ORDER BY column(s) ASC|DESC;
```

LIMIT

Description: Used to give the top-n elements of a given category.

Syntax:

```
SELECT column(s)
FROM table
LIMIT n;
```

CREATE TABLE

Description: Used to create a new table.

Syntax:

```
CREATE TABLE tableName(  
    column1 datatype,  
    column2 datatype,  
    ...  
    columnN datatype,  
    PRIMARY KEY (one or more columns)  
);
```

Common Column Data Types:

VARCHAR(N) - strings of up to N characters (e.g., 'Whitaker')

INTEGER - integers (e.g., 10)

FLOAT - floats (e.g., 1.54)

DATETIME - date/time representation (e.g., '2017-05-25 18:20:32')

Alias for tables names

Description: You can use an alias to give a table a different name. You assign a table an alias by using the AS keyword as the following syntax for a partial query:

```
table_name AS table_alias
```

Examples:

```
SELECT g.name, g.platform FROM Games AS g  
WHERE g.name LIKE "%Pokemon%"
```

As a shortcut you can also write this same query without the AS

```
SELECT g.name, g.platform FROM Games g  
WHERE g.name LIKE "%Pokemon%"
```

INSERT INTO

Description: Used to insert a new record (row) into an existing table, where the listed values correspond to the listed columns.

Syntax:

```
INSERT INTO table_name  
    (column1, column2, ..., columnN)  
VALUES  
    (value1, value2, ..., valueN);
```

DELETE

Description: Used to remove a record (row) which matches condition(s) from an existing table.

Syntax:

```
DELETE FROM tableName  
WHERE condition(s);
```

UPDATE

Description: Used to modify the existing records in a table.

Syntax:

```
UPDATE table_name
SET column1 = value1, column2 =
value2, ... WHERE condition(s);
```

JOIN

Description: Used to select values from more than one table. Note that you can write multi-table queries either with equivalent WHERE clauses, or using the JOIN keyword.

Syntax/Examples:

```
SELECT col(s)
FROM table1, table2, ...
WHERE table1.a = table2.b
AND table2.c > '42';
```

OR

```
SELECT col(s)
FROM table1
JOIN table2 on table1.a = table2.b
JOIN ...
WHERE table2.c > '42';
```