



MySQL Commands

SELECT

Syntax

```
SELECT column_name FROM table_name;
```

Selecting all columns with the wildcard *

```
SELECT * FROM members;
```

```
SELECT * FROM state;
```

```
SELECT * FROM level;
```

Selecting and limiting the number of rows returned

```
SELECT * FROM state LIMIT 10;
```

Selecting individual columns

```
SELECT f_name, l_name, city FROM members;
```

WHERE and LIKE clauses

Syntax

```
SELECT column_name FROM table_name WHERE a_column_name = 'some_value';
```

Selecting to match criteria with WHERE

```
SELECT f_name, l_name, city FROM members WHERE l_name = 'stewart';
```

```
SELECT f_name, l_name, city FROM members WHERE username = 'goatboy';
```

Using LIKE

Syntax

```
SELECT column_name FROM table_name WHERE a_column_name LIKE 'some_pattern%';
```

LIKE uses % to indicate missing pattern (not case-sensitive)

- `%.gif` ...would search for patterns ending in ".gif"

- n% ...would search for anything starting with the letter "n"
- %tra% ...searches for anything with the pattern "tra" inside if it at any position. It could return "tramp," "Sinatra," "intravenous" etc. if those words were in the search columns.

Examples:

```
SELECT f_name, l_name, city FROM members WHERE l_name LIKE 'd%';
```

```
SELECT long_form FROM state WHERE long_form LIKE 'n%';
```

ORDER BY

Ordering sorts data in a set of records. It will by default order in ascending order if not specified. Otherwise, you can specify descending.

Syntax

```
SELECT column_name(s)  
FROM table_name  
ORDER BY column_name(s) ASC|DESC;
```

Examples

```
SELECT *  
FROM members  
ORDER BY username;
```

```
SELECT *  
FROM members  
ORDER BY username DESC;
```

//Using a limit!

```
SELECT *  
FROM state  
ORDER BY long_form DESC LIMIT 10;
```

JOIN

Syntax

```
SELECT table_a.some_column_name, table_b.some_column_name
FROM table_a, table_b
WHERE (table_a.some_column_name = table_b.some_column_name) AND (table_a.some_column_name =
table_b.some_column_name);
```

Joining tables to get values from each table to temporarily create one larger table in memory

```
SELECT members.l_name, members.f_name, members.city, members.zip, state.state, level.type, level.donation
FROM members, state, level
WHERE (members.state = state.id) AND (members.type = level.id);
```

INSERT

Syntax

```
INSERT INTO table_name
VALUES (value_a,value_b,value_c');
```

This command inserts NEW column values into a NEW row(s). An example of when you would use this type of query is when someone fills out a web form, whose data you want to store in a database. This could be anything from setting up a user account, buying goods, etc.

It is important that you have a place in the syntax for each field/column, or else the values will be inserted into the wrong columns. The order of the values (see below) is identical to the way that the table's columns are ordered.

You do NOT use this command to merely update an existing row. For that, use UPDATE.

```
INSERT INTO members
VALUES ('lcotnoir','cotnoir','leigh','555 Pacific Ocean','#2','San
Diego','5','92182','lcotnoir@mail.sdsu.edu','1');
```

UPDATE

Syntax

```
UPDATE table_name SET column_name='some_value' WHERE some_column='some_value';
```

This command is used to update existing records. A good example is if a user changes his or her contact information in a form.

```
UPDATE members SET address2='Apartment 77' WHERE username='smokey';
```

```
UPDATE members SET address2='Unit 245' WHERE username='smokey';
```

You can also update multiple values at once

```
UPDATE members SET f_name='tim',l_name='tiny' WHERE username='smokey';
```

You can also update multiple values at once on multiple tables!

```
UPDATE members,state SET members.state='51',state.id='51' WHERE state.id='5' AND  
members.state='5';
```

```
UPDATE members,state SET members.state='5',state.id='5' WHERE state.id='51' AND  
members.state='51';
```

DELETE

Syntax

```
DELETE FROM table_name WHERE some_column = 'some_value';
```

Used to delete complete rows/records from a table.

```
DELETE FROM members WHERE username = 'lcothnoir';
```