MySQL Quick Reference

```
# admin commands
show databases;
use <database>;
show tables;
describe <database>;
create table <table>();
show create table <table>;
drop database <database>;
```

```
# aggregate functions
avg()
min()
max()
sum()
count()
```

```
# control flow
IF("" AND "")
CASE WHEN <condition> THEN
```

```
# query commands
database();
user();
select * from <table>;
```

```
# string functions
upper()
strcmp()
ascii()
replace(str,from,to)
substr(str,start,len)
sort()
round()
datediff()
date_format(date,"%d-%m")
```

```
# import and export
MySQL -u root -p <bus.sql;
```

```
# keywords
select
where
between
like
in
and
or
limit
distinct
order by
group by
```

```
# dates
YEAR()
DAY()
MONTHNAME()
```

Stored functions and procedures

```
# stored function add numbers
CREATE FUNCTION add2Nums(n1 integer,
  n2 integer)
RETURNS integer
DETERMINISTIC
BEGINS
    RETURN n1 + n2;
END

SELECT add2Nums(3, 10);
```

FUNCTIONS	PROCEDURES
Return a single value	Retun 0 or more values
Only select	select, **insert**, **update**, **delete**
Cant use stored procedures	can use stored functions
Does **not** support transactions	Support transactions

```
# stored function return discount
CREATE FUNCTION discount(age INT(11))
RETURNS VARCHAR(3)
DETERMINISTIC
BEGIN
    IF age < 16 THEN
        RETURN “0%”;
    ELSEIF age < 26 THEN
        RETURN “10%”;
    ELSEIF age < 40 THEN
        RETURN “20%”;
    ELSEIF age < 60 THEN
        RETURN “30%”;
    ELSE
        RETURN “40%”;
    END IF;
END
```

```
Create procedure
CREATE PROCEDURE make_mileage(mk VARCHAR(20),
  ml INT(11))
DETERMINISTIC
BEGIN
    SELECT * FROM CAR
    WHERE make LIKE mk
    AND mileage < ml
    ORDER BY mileage;
END
```

```
# Routine management
select <function>;
call <procedure>;
SELECT name, type from MYSQL.PROCE;
SHOW CREATE FUNCTION <function>;
DROP FUNCTION <function>
```

Sample Queries

```
# Show all details of all teachers who were born in January,
# February or March, and who can teach as far as Junior Cert only.

mysql> select * from teacher
    -> where (month(dob) != 1
    -> or month(dob) != 2
    -> or month(dob) != 3)
    -> and level = 'j';
+-----+---------------+-------+------------+------------+
| tid | Name          | level | experience | dob        |
+-----+---------------+-------+------------+------------+
|   3 | Ms. Smith     | J     |          4 | 1980-03-23 |
|   5 | Mr. Kavanagh  | J     |         50 | 1949-11-01 |
|   6 | Mr. Picasso   | J     |         42 | 1939-03-30 |
+-----+---------------+-------+------------+------------+
```

```
# Show all details of all subjects whose 3rd or 4th letter
# is “l”. Sort them by name.

mysql> SELECT * FROM school.subject
    -> where name like '___i%'
    -> or name like '____i%'
    -> order by name;
+----------+--------------+---------------+
| Name     | Teacher      | OnLeavingCert |
+----------+--------------+---------------+
| English  | Mr. Kavanagh |             1 |
| Religion | Fr. Lynch    |             1 |
+----------+--------------+---------------+
```

```
# Show the name of all teachers who have 10, 15, 20, 25,
# 30, 35, 40, 45, 50, 55 or 60 years experience. Sort from
# youngest to oldest.

mysql> SELECT * FROM school.teacher
    -> where experience in (10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60);
+-----+---------------+-------+------------+------------+
| tid | Name          | level | experience | dob        |
+-----+---------------+-------+------------+------------+
|   1 | Mr. Pasteur   | L     |         15 | 1960-02-02 |
|   4 | Mr. Hawking   | L     |         40 | 1951-02-19 |
|   5 | Mr. Kavanagh  | J     |         50 | 1949-11-01 |
|   7 | Fr. Lynch     | L     |         55 | 1939-03-31 |
+-----+---------------+-------+------------+------------+
```

```
# Show all details of all Females born in the 1950s and hired between
# September 1st 1988 and February 28th 1991.

mysql> SELECT * FROM employees.employees
    -> where gender = 'F'
    -> and (year(birth_date)>='1950' and year(birth_date)<='1959')
    -> and (hire_date>='1988-09-01' and hire_date<='1991-02-28')
    -> ;
+--------+------------+------------+------------+--------+------------+
| emp_no | birth_date | first_name | last_name  | gender | hire_date  |
+--------+------------+------------+------------+--------+------------+
|  10006 | 1953-04-20 | Anneke     | Preusig    | F      | 1989-06-02 |
|  10007 | 1957-05-23 | Tzvetan    | Zielinski  | F      | 1989-02-10 |
| ....   |            |            |            |        |            |
```

```
|  10088 | 1954-02-25 | Jungsoon   | Syrzycki   | F      | 1988-09-02 |
|  10099 | 1956-05-25 | Valter     | Sullins    | F      | 1988-10-18 |
+--------+------------+------------+------------+--------+------------+
```

```
# Show the following details, in the following order, for the first
# 15 employees, in emp_no order: ID, Title, Name, Surname, Gender.
# Title should be “Mr.” if the employee is Male, and “Ms.” if the employee is female.

mysql> SELECT emp_no as ID,
    ->    if(gender = 'M', 'Mr.','Ms.') as Gender,
    ->        first_name as Name,
    ->        last_name as Surname,
    ->        gender as Gender
    -> FROM employees.employees
    -> order by emp_no
    -> limit 15;

+-------+--------+-----------+-----------+--------+
| ID    | Gender | Name      | Surname   | Gender |
+-------+--------+-----------+-----------+--------+
| 10001 | Mr.    | Georgi    | Facello   | M      |
| 10002 | Ms.    | Bezalel   | Simmel    | F      |
   .........
| 10014 | Mr.    | Berni     | Genin     | M      |
| 10015 | Mr.    | Guoxiang  | Nooteboom | M      |
+-------+--------+-----------+-----------+--------+
```

```
# Show the following details emp_no, the maximum salary for each employee,
# and the tax bracket the employee’s maximum salary is in (Tax Bracket).
# Tax brackets are defined as follows:

# Max Salary      Tax Bracket
# Under 40,000    30%
# Under 60,000    40%
# Under 80,000    50%
# Over 80,000     60%

mysql> SELECT emp_no as 'Employee Number',
    ->    max(salary) as 'Max Salary',
    ->        CASE
    ->          when max(salary) < 40000 then '30%'
    ->          when max(salary) < 60000 then '40%'
    ->          when max(salary) < 80000 then '50%'
    ->          else '60%'
    ->        END as 'Tax Bracket'
    -> FROM employees.salaries
    -> group by emp_no
    -> limit 15;
+-----------------+------------+-------------+
| Employee Number | Max Salary | Tax Bracket |
+-----------------+------------+-------------+
|           10001 |      88958 | 60%         |
|           10002 |      72527 | 50%         |
|           10003 |      43699 | 40%         |
    ..........
|           10013 |      68901 | 50%         |
|           10014 |      60598 | 50%         |
|           10015 |      40000 | 40%         |
+-----------------+------------+-------------+
```

```
# Show the manu_code and manu_name as well as associated reg,
# for each manufacturer who has vehicles listed in the vehicle table.
```

```
mysql> SELECT gm.manu_code, gm.manu_name, gv.reg
    -> FROM garage.manufacturer as gm
    -> inner join garage.vehicle as gv
    -> on gm.manu_code=gv.manu_code;

+-----------+-----------+--------------+
| manu_code | manu_name | reg          |
+-----------+-----------+--------------+
| FOR       | Ford      | 2009-RN-12   |
| FOR       | Ford      | 2011-G-995   |
   .........
| TOY       | Toyota    | 2010-G-13345 |
| TOY       | Toyota    | 2016-D-12345 |
+-----------+-----------+--------------+
```

```
# lots of left joins ...
SELECT DISTINCT(ee.emp_no), ee.first_name, ee.last_name, ed.name AS department
FROM employees2.employees AS ee
LEFT JOIN employees2.salaries AS es
ON ee.emp_no=es.emp_no
LEFT JOIN employees2.dept AS ed
ON es.dept_no=ed.dept_no;
```