Table of Contents

1. Direct comparison operators 1
1.1. Tests for exact matches 1
1.2. Tests for non-matches 2
1.3. Tests for inequalities 3
2. Tests that require conversions 3

1. Direct comparison operators

In this section we will look at the direct comparison operators.

These operators compare two expressions. The SQL comparison operators are;

```
=           !=     <>
>           >=     <          <=
```

1.1. Tests for exact matches

Demo 01: Display only rows with an exact match on Salary.

```
select emp_id
, name_last as "Employee"
, salary
from a_emp.employees
where salary = 12000;
+--------+----------+----------+
| emp_id | Employee | salary   |
+--------+----------+----------+
|    108 | Green    | 12000.00 |
+--------+----------+----------+
```

Demo 02: Some queries do not return any rows. This does not mean the query is incorrect. We just do not have any matching rows. Depending on the client the results might be shown with a header only or just with a message.

```
select emp_id
, name_last as "Employee"
, salary
from a_emp.employees
where salary = 18888
;
Empty set (0.00 sec)
```

Demo 03: Display only location rows with a country-id of US.

```
select loc_city
, loc_street_address
from a_emp.locations
where loc_country_id ='US';
+---------------------+---------------------+
| loc_city            | loc_street_address  |
+---------------------+---------------------+
| Southlake           | 2014 Jabberwocky Rd |
| South San Francisco | 2011 Interiors Blvd |
| San Francisco       | 50 Pacific Ave      |
+---------------------+---------------------+
```

Demo 04: MySQL is not case specific on text comparisons.

```
select loc_city
, loc_street_address
from a_emp.locations
where loc_city ='SAN FRANCISCO';
+---------------+--------------------+
| loc_city      | loc_street_address |
+---------------+--------------------+
| San Francisco | 50 Pacific Ave     |
+---------------+--------------------+
```

Demo 05: Test date values using the default date format; enclose the date literal in single quotes.

```
select ord_id
, cust_id
, ord_mode
from a_oe.order_headers
where ord_date = '2011-12-15';
+--------+---------+----------+
| ord_id | cust_id | ord_mode |
+--------+---------+----------+
|    126 |  409190 | DIRECT   |
|    127 |  915001 | ONLINE   |
|    128 |  409030 | ONLINE   |
|    129 |  915001 | DIRECT   |
+--------+---------+----------+
```

Demo 06: Using a Row equality test.

```
select prod_id, prod_name, catg_id, prod_warranty_period
from a_prd.products
where row(catg_id, prod_warranty_period ) = row('HW', 12);
+---------+-----------------+---------+----------------------+
| prod_id | prod_name       | catg_id | prod_warranty_period |
+---------+-----------------+---------+----------------------+
|    1000 | Hand Mixer      | HW      |                   12 |
|    1090 | Gas grill       | HW      |                   12 |
|    1110 | Pancake griddle | HW      |                   12 |
|    1160 | Mixer Deluxe    | HW      |                   12 |
+---------+-----------------+---------+----------------------+
```

1.2. Tests for non-matches

Demo 07: Use the not equals operator to exclude rows. You can use != or <>

```
select loc_city
, loc_street_address
from a_emp.locations
where loc_country_id !='US';
+-------------+-----------------------+
| loc_city    | loc_street_address    |
+-------------+-----------------------+
| Toronto     | 147 Spadina Ave       |
| Munich      | Schwanthalerstr. 7031 |
| Mexico City | Mariano Escobedo 9991 |
+-------------+-----------------------+
```

1.3. Tests for inequalities

Demo 08: Finding jobs with a max salary less than $60,000. Do not include formatting characters- such as the $ or the comma in the literal.

```
select job_id, max_salary
, job_title
from a_emp.jobs
where max_salary <60000;
+--------+------------+-----------+
| job_id | max_salary | job_title |
+--------+------------+-----------+
|      8 |   30000.00 | Sales Rep |
+--------+------------+-----------+
```

Demo 09: Finding jobs with a max salary greater than or equal to 60000.

```
select job_id, max_salary
, job_title
from a_emp.jobs
where max_salary >= 60000;
+--------+------------+---------------+
| job_id | max_salary | job_title     |
+--------+------------+---------------+
|      1 |  100000.00 | President     |
|      2 |   75000.00 | Marketing     |
|      4 |   60000.00 | Sales Manager |
|     16 |  120000.00 | Programmer    |
+--------+------------+---------------+
```

2. Tests that require conversions

These are queries that you could try to run that might not work at all in some dbms; that might work with invalid conversions; or that might turn out OK. In any case you should not run these types of queries- care about your data!

Demo 10: Comparing a string to a number: You should test the numeric salary attribute against a number- not against a string.

```
select emp_id
, name_last as "Employee"
, salary
from a_emp.employees
where salary = '15000';
```

Demo 11: -- comparing a date to a number

```
select emp_id
, hire_date
from a_emp.employees
where hire_date> 34567;
```