Table of Contents

[1. Direct comparison operators 1](#_Toc363583601)

[1.1. Tests for exact matches 1](#_Toc363583602)

[1.2. Tests for non-matches 2](#_Toc363583603)

[1.3. Tests for inequalities 3](#_Toc363583604)

[2. Tests that require conversions 3](#_Toc363583605)

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;

= != <>

> >= < <=

Tests for exact matches

1. 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 |

+--------+----------+----------+

1. 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)

1. 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 |

+---------------------+---------------------+

1. 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 |

+---------------+--------------------+

1. 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 |

+--------+---------+----------+

1. 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 |

+---------+-----------------+---------+----------------------+

Tests for non-matches

1. 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 |

+-------------+-----------------------+

Tests for inequalities

1. 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 |

+--------+------------+-----------+

1. 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 |

+--------+------------+---------------+

1. 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!

1. 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';

1. -- comparing a date to a number

select emp\_id

, hire\_date

from a\_emp.employees

where hire\_date> 34567;