

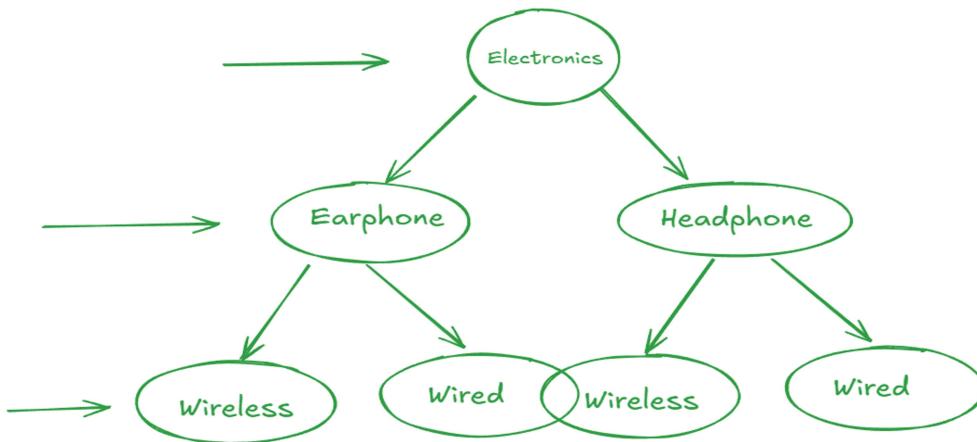
## Filtering Rows - p1

### Filtering Data - p1

```
SELECT * FROM products WHERE category = 'Electronics' AND subcategory = 'Headphones' AND connectivity = 'Wireless';
```

The screenshot shows a product search interface. At the top, there are navigation links: Electronics, TVs & Appliances, Men, Women, Baby & Kids, Home & Furniture, and Sports, Books & More. Below this is a 'Filters' section with 'CLEAR ALL' and two selected filters: 'Bluetooth' and '4★ & above'. Under 'CATEGORIES', 'Headphones' is expanded, showing 'Wireless Headphones'. A 'PRICE' filter shows a range from 'Min' to '2600+'. The main area displays three products: a black over-ear headset, a black on-ear headset, and a dark blue over-ear headset. Each product has a small heart icon and a 'View Details' button below it.

Headphones [-----] earphone[----]



Comparison Operator:-

LIKE : wildcard : % & \_

Operator	Description
=	Equal to.
<> or !=	Not equal to
<	Less than.
>	Greater than.
<=	Less than or equal to
>=	Greater than or equal to

## Equal to Operator :-

In SQL, the equal to operator is represented by the symbol `=`. It is used to determine if they are equal. Here's an example of how it is used in a

SQL

```
SELECT * FROM table_name WHERE column_name = value;
```

Sample Question : Retrieve all data where release year is 1916 from films table.

```
mysql> SELECT * FROM films where year = 1916;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | Title          | Year | Country | Runtime | Language | MPAA Rating | Budget | Box Office |
+----+-----+-----+-----+-----+-----+-----+-----+
|  1 | Intolerance: Love's Struggle Throughout the Ages | 1916 | USA     | 123    | Not Rated |             | 385907.0 |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

## Not equal to

In SQL, the not equal to operator can be represented by either expressions to determine if they are not equal. Here's an example

```
SELECT * FROM table_name WHERE column_name <> value;
```

or

```
SELECT * FROM table_name WHERE column_name != value;
```

```
mysql> SELECT * FROM films where COUNTRY LIKE "INDIA";
+----+-----+-----+-----+-----+
| id | Title          | Year | Country | Runtime | Language |
+----+-----+-----+-----+-----+
| 1069 | Earth          | 1998 | India   | 110    | Hindi    |
| 1383 | Fiza           | 2000 | India   | 167    | Hindi    |
| 1526 | Ayurveda: Art of Being | 2001 | India   | 102    | English  |
| 1608 | Monsoon Wedding | 2001 | India   | 114    | Hindi    |
| 2275 | Veer-Zaara     | 2004 | India   | 192    | Hindi    |
| 2314 | Chocolate: Deep Dark Secrets | 2005 | India   | 160    | Hindi    |
| 2402 | Neal 'N' Nikki | 2005 | India   | 97     | English  |
| 2609 | Kabhi Alvida Naa Kehna | 2006 | India   | 193    | Hindi    |
| 2611 | Krish          | 2006 | India   | 168    | Hindi    |
| 2613 | Lage Raho Munna Bhai | 2006 | India   | 144    | Hindi    |
| 2656 | Rang De Basanti | 2006 | India   | 157    | Hindi    |
| 2806 | Gandhi, My Father | 2007 | India   | 136    | Hindi    |
| 2849 | Namastey London | 2007 | India   | 128    | Hindi    |
| 3072 | Roadside Romeo | 2008 | India   | 93     | Hindi    |
| 3322 | Paa             | 2009 | India   | 133    | Hindi    |
| 3336 | Rocket Singh: Salesman of the Year | 2009 | India   | 150    | Hindi    |
| 3508 | Housefull      | 2010 | India   | 144    | Hindi    |
| 3525 | Kites           | 2010 | India   | 90     | English  |
| 3549 | My Name Is Khan | 2010 | India   | 128    | Hindi    |
| 3711 | Dum Maaro Dum | 2011 | India   | 128    | Hindi    |
| 3962 | Jab Tak Hai Jaan | 2012 | India   | 176    | Hindi    |
| 3969 | Khiladi 786 | 2012 | India   | 141    | Hindi    |
| 4120 | ABCD (Any Body Can Dance) | 2013 | India   | 160    | Hindi    |
| 4176 | Faith Connections | 2013 | India   | 115    | Hindi    |
| 4306 | The Lunchbox    | 2013 | India   | 104    | Hindi    |
| 4340 | Yeh Jawaani Hai Deewani | 2013 | India   | 160    | Hindi    |
| 4420 | Fugly           | 2014 | India   | 134    | Hindi    |
| 4494 | Ramanujan      | 2014 | India   | 153    | English  |
| 4615 | Baahubali: The Beginning | 2015 | India   | 159    | Telugu   |
| 4808 | VaaLu           | 2015 | India   | 155    | Tamil    |
| 4824 | Airlift         | 2016 | India   | 130    | Hindi    |
+----+-----+-----+-----+-----+
31 rows in set (0.02 sec)
```

## Less than

In SQL, the less than operator is represented by the symbol <. It expressions to determine if the left operand is less than the right it is used in a SQL query:

```
SELECT * FROM table_name WHERE column_name < value;
```

Sample Question: Retrieve title, year where runtime is less than 150 minutes in films table.

6 • **SELECT title , year FROM films where runtime < 150;**

title	year
Intolerance: Love's Struggle Throughout the Ages	1916
Over the Hill to the Poorhouse	1920
Metropolis	1927
Pandora's Box	1929
The Broadway Melody	1929
Hell's Angels	1930
A Farewell to Arms	1932
42nd Street	1933
She Done Him Wrong	1933

## Greater than.

In SQL, the greater than operator is represented by the symbol >. It expressions to determine if the left operand is greater than the how it is used in a SQL query:

```
SQL
SELECT * FROM table_name WHERE column_name > value;
```

Sample Question: Retrieve title, year where runtime is greater than 150 minutes in films table.

mysql> sELECT title , year FROM Films WHERE Runtime > 150;	
title	year
The Big Parade	1925
Gone with the Wind	1939
The Best Years of Our Lives	1946
Quo Vadis	1951
The Greatest Show on Earth	1952
Seven Samurai	1954
The Bridge on the River Kwai	1957
Judgment at Nuremberg	1961

## Less than or equal to

In SQL, the less than or equal to operator is represented by the `<=` operator. It compares two expressions to determine if the left operand is less than or equal to the right operand. Here's an example of how it is used in a SQL query:

```
SELECT * FROM table_name WHERE column_name <= value;
```

Sample Question: Retrieve title, year where year is less than or equal to 2000 in films table.

9 • **SELECT Title , Year FROM films where Year <= 2000;**

10

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	Title	Year			
▶	Intolerance: Love's Struggle Throughout the Ages	1916			
	Over the Hill to the Poorhouse	1920			
	The Big Parade	1925			
	Metropolis	1927			
	Pandora's Box	1929			
	The Broadway Melody	1929			
	Hell's Angels	1930			
	A Farewell to Arms	1932			
	42nd Street	1933			

## Greater than or equal to

In SQL, the greater than or equal to operator is represented by the `>=` operator. It compares two expressions to determine if the left operand is greater than or equal to the right operand. Here's an example of how it is used in a SQL query:

```
SELECT * FROM table_name WHERE column_name >= value;
```

Sample Question: Retrieve title, year where year is greater than or equal to 2000 in films table.

10 • `SELECT Title , Year FROM films where Year >= 2000;`

11

The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'Title' and 'Year'. The data consists of seven rows, all of which have the 'Year' value set to 2000. The rows listed are: 102 Dalmatians, 28 Days, 3 Strikes, Aberdeen, All the Pretty Horses, Almost Famous, American Psycho, Amores Perros, and An Everlasting Piece.

Title	Year
102 Dalmatians	2000
28 Days	2000
3 Strikes	2000
Aberdeen	2000
All the Pretty Horses	2000
Almost Famous	2000
American Psycho	2000
Amores Perros	2000
An Everlasting Piece	2000

## Using Multiple Criteria in WHERE

We can have multiple criteria to fulfill in `where` clause. To do so, we will study three additional keywords: `OR`, `AND` and `BETWEEN`.

*OR --- IN  
AND --- NOT IN*

Logical Operator can be used to filter based on fulfilling multiple criteria at the same time.

### 1. `OR` operator

The `OR` operator is used when you want to filter data based on multiple criteria, and it's sufficient for at least one of the conditions to be true. It is employed to combine conditions that allow you to select records that meet at least one of the specified criteria.

Example: Select all films released in either 1994 or 2000 from the films database.

```
SELECT title
FROM films
WHERE year = 1994 OR year = 2000;
```

```
mysql> SELECT title , year FROM Films where Year = 1994 OR year = 2000;
+-----+-----+
| title | year |
+-----+-----+
| 3 Ninjas Kick Back | 1994 |
| A Low Down Dirty Shame | 1994 |
| Ace Ventura: Pet Detective | 1994 |
| Baby's Day Out | 1994 |
| Beverly Hills Cop III | 1994 |
+-----+-----+
224 rows in set (0.01 sec)
```

```
mysql> SELECT title , year FROM Films where Year = 1994 OR year = 1996 OR year = 1998 OR year = 2000;  
+-----  
457 rows in set (0.00 sec)
```

Year = 1994 OR country = "USA"

```
mysql> SELECT title , year FROM Films where Year = 1994 OR year = 1996 OR year = 1998 OR year = 2000;  
+-----  
| title | year |  
+-----  
| 3 Ninjas Kick Back | 1994 |  
| A Low Down Dirty Shame | 1994 |  
| Ace Ventura: Pet Detective | 1994 |  
| Baby's Day Out | 1994 |  
| Beverly Hills Cop III | 1994 |  
| Bullets Over Broadway | 1994 |  
| Clear and Present Danger | 1994 |  
| Clerks | 1994 |  
| Crooklyn | 1994 |
```

```
mysql> SELECT title , year FROM Films where Year IN (1994 , 1996 , 1998 , 2000);  
+-----  
| title | year |  
+-----  
| 3 Ninjas Kick Back | 1994 |  
| A Low Down Dirty Shame | 1994 |  
| Ace Ventura: Pet Detective | 1994 |  
| Baby's Day Out | 1994 |  
| Beverly Hills Cop III | 1994 |  
| Bullets Over Broadway | 1994 |  
| Clear and Present Danger | 1994 |  
| Clerks | 1994 |  
| Crooklyn | 1994 |
```

YEAR IN (1994 , 1996 , 1998 , 2000) OR Country IN ("USA" , "China" , "India");

```
mysql> SELECT title , year FROM Films where Year = 1994 OR year = 1996 OR year = 1998 OR year = 2000 OR Country = "USA" OR Country = "China" OR Country = "India";  
+-----
```

3869 rows in set (0.02 sec)

```
mysql> SELECT title , year FROM Films where Year IN (1994 , 1996 , 1998 , 2000) OR Country IN ("USA" , "China" , "India");  
+-----  
| title | year |  
+-----  
| Intolerance: Love's Struggle Throughout the Ages | 1916 |  
| Over the Hill to the Poorhouse | 1920 |  
| The Big Parade | 1925 |  
| The Broadway Melody | 1929 |  
| Hell's Angels | 1930 |  
| A Farewell to Arms | 1932 |  
| 42nd Street | 1933 |  
| She Done Him Wrong | 1933 |  
| It Happened One Night | 1934 |  
| Top Hat | 1935 |  
| Modern Times | 1936 |
```

## 2. AND operator

The AND operator is used when you want to filter data based on multiple criteria, and all the specified conditions must be true. It is employed to combine conditions that require records to satisfy all the specified criteria.

Example: Select titles of films released between 1994 and 2000 from the films database, with 1994 and 2000 inclusive.

```
SELECT title
FROM films
WHERE release_year >= 1994 AND release_year <= 2000;
```

```
SELECT title , year FROM Films where Year IN (1994 , 1996 , 1998 , 2000) AND Country IN ("USA" , "China" , "India");
```

```
mysql> SELECT title , year , Country FROM Films where Year IN (1994 , 1996 , 1998 , 2000) AND Country IN ("USA" , "China" , "India");
+-----+-----+-----+
| title | year | Country |
+-----+-----+-----+
| 3 Ninjas Kick Back | 1994 | USA
| A Low Down Dirty Shame | 1994 | USA
| Ace Ventura: Pet Detective | 1994 | USA
| Baby's Day Out | 1994 | USA
| Beverly Hills Cop III | 1994 | USA
| Bullets Over Broadway | 1994 | USA
| Clear and Present Danger | 1994 | USA
| Clerks | 1994 | USA
| Crooklyn | 1994 | USA
| Disclosure | 1994 | USA
| Don Juan DeMarco | 1994 | USA
| Dumb & Dumber | 1994 | USA
| Ed Wood | 1994 | USA
```

Whenever you take the inclusive value - Range [Between]



```
mysql> SELECT Title , year FROM Films where year >= 1994 AND year <= 2000;
+-----+-----+
| Title | year |
+-----+-----+
| 3 Ninjas Kick Back | 1994
| A Low Down Dirty Shame | 1994
| Ace Ventura: Pet Detective | 1994
| Baby's Day Out | 1994
| Beverly Hills Cop III | 1994
| Bullets Over Broadway | 1994
| Clear and Present Danger | 1994
| Clerks | 1994
| Crooklyn | 1994
| Disclosure | 1994
| Don Juan DeMarco | 1994
```

```
mysql> SELECT Title , year FROM Films where year BETWEEN 1994 AND 2000;
+-----+-----+
| Title | year |
+-----+-----+
| 3 Ninjas Kick Back | 1994
| A Low Down Dirty Shame | 1994
```

"Inclusive Range"

AND , AND , !AND ----- NOT IN

```
mysql> SELECT title , Year FROM Films where year NOT IN (1994 , 1996 , 1998 , 2000);  
+-----+-----+  
| title | Year |
```

```
mysql> SELECT title , Year FROM Films where year <> 1994 AND year <> 1996 AND year <> 1998 AND year != 2000;  
+-----+-----+  
| title | Year |  
+-----+-----+  
| Total | 1916 |
```

title	year
	1990
	1991
	1992
	1994
	1996
	2000
	2001
	2002
	2004

↑ False

Short Circuiting

1994 != 1994 -- False

1994 <> 1994 -- False

1994 = 1994 -- True

Short Circuiting

OR ----- T || T || F || T || F || T ----- True

AND ----- F && T && F && T && F && T && F && T && T ----- False

Short Circuiting

## ▼ Question

[https://sqlbolt.com/lesson/select\\_queries\\_with\\_constraints?authuser=0](https://sqlbolt.com/lesson/select_queries_with_constraints?authuser=0)

Table: Movies

<b>Id</b>	<b>Title</b>	<b>Director</b>	<b>Year</b>	<b>Length_minutes</b>
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

```
SELECT * FROM movies;
```

### Exercise 2 — Tasks

1. Find the movie with a row **id** of 6

```
SELECT * FROM Movies where id = 6;
```

2. Find the movies released in the **year**s

between 2000 and 2010

```
SELECT * FROM Movies where year BETWEEN 2000 AND 2010;
```

3. Find the movies **not** released in the **year**s

between 2000 and 2010

```
SELECT * FROM Movies where year NOT BETWEEN 2000 AND 2010;
```

4. Find the first 5 Pixar movies and their release

**year**

```
SELECT title , year FROM Movies LIMIT 5;
```