

# MySQL WHERE Clause: AND, OR, IN, NOT IN Query Example

By Richard Peterson ⌚ Updated September 30, 2022

## What is WHERE Clause in MySQL?

**WHERE Clause** in MySQL is a keyword used to specify the exact criteria of data or rows that will be affected by the specified SQL statement. The WHERE clause can be used with SQL statements like INSERT, UPDATE, SELECT, and DELETE to filter records and perform various operations on the data.

We looked at how to query data from a [database](#) using the SELECT statement in the previous tutorial. The SELECT statement returned all the results from the queried database table.

There are, however, times when we want to restrict the query results to a specified condition. The WHERE clause in SQL comes handy in such situations.





WHERE clause in MySQL

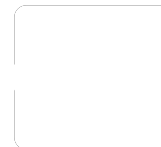
## WHERE clause Syntax

The basic syntax for the WHERE clause when used in a MySQL SELECT WHERE statement is as follows.

```
SELECT * FROM tableName WHERE condition;
```

Guru99

### EXPLORE MORE



Learn Java Programming  
with Beginners Tutorial

08:32

Linux Tutorial for  
Beginners: Introduction  
to Linux Operating



00:02

35:04

---

## HERE

- “**SELECT \* FROM tableName**” is the standard SELECT statement
- “**WHERE**” is the keyword that restricts our select query result set and “**condition**” is the filter to be applied on the results. The filter could be a range, single value or sub query.

Let's now look at a **practical example**.

Suppose we want to get a member's personal details from members table given the membership number 1, we would use the following script to achieve that.

```
SELECT * FROM `members` WHERE `membership_number` = 1;
```

Executing the above script in MySQL workbench on the “myflixdb” would produce the following results.

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contct_number
1	Janet Jones	Female	21-07-1980	First Street Plot No 4	Private Bag	0759 253 542



## WHERE clause combined with – *AND* LOGICAL Operator

The WHERE condition in MySQL when used together with the AND logical operator, is only executed if ALL filter criteria specified are met.

Let's now look at a practical example – Suppose we want to get a list of all the movies in category 2 that were released in 2008, we would use the script shown below to achieve that.

```
SELECT * FROM `movies` WHERE `category_id` = 2 AND `year_released` = 2008;
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

movie_id	title	director	year_released	category_id
2	Forgetting Sarah Marshal	Nicholas Stoller	2008	2

## WHERE clause combined with – *OR* LOGICAL Operator

The WHERE clause when used together with the OR operator, is only executed if any or the entire specified filter criteria is met.

The following script gets all the movies in either category 1 or category 2



```
SELECT * FROM `movies` WHERE `category_id` = 1 OR `category_id` = 2;
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

movie_id	title	director	year_released	category_id
1	Pirates of the Caribbean 4	Rob Marshall	2011	1
2	Forgetting Sarah Marshal	Nicholas Stoller	2008	2

## WHERE clause combined with – /IN Keyword

The WHERE in MySQL clause, when used together with the IN keyword only affects the rows whose values matches the list of values provided in the IN keyword. The MySQL IN statement helps to reduce number of OR clauses you may have to use.

The following MySQL WHERE IN query gives rows where membership\_number is either 1 , 2 or 3

```
SELECT * FROM `members` WHERE `membership_number` IN (1,2,3);
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contact_number
-------------------	------------	--------	---------------	------------------	----------------	----------------



1	Janet Jones	Female	21-07-1980	First Street Plot No 4	Private Bag	0759 253 542
2	Janet Smith Jones	Female	23-06-1980	Melrose 123	NULL	NULL
3	Robert Phil	Male	12-07-1989	3rd Street 34	NULL	12345



## WHERE clause combined with – *NOT IN* Keyword

The WHERE clause when used together with the NOT IN keyword DOES NOT affects the rows whose values matches the list of values provided in the NOT IN keyword.

The following query gives rows where membership\_number is NOT 1 , 2 or 3

```
SELECT * FROM `members` WHERE `membership_number` NOT IN (1,2,3);
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contct_number
4	Gloria Williams	Female	14-02-1984	2nd Street 23	NULL	NULL



## WHERE clause combined with – COMPARISON Operators

The less than (<), equal to (=), not equal to (≠) comparison operators can be used with the WHERE Clause

### =Equal To

The following script gets all the female members from the members table using the equal to comparison operator.

```
SELECT * FROM `members` WHERE `gender` = 'Female';
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contct_number
1	Janet Jones	Female	21-07-1980	First Street Plot No 4	Private Bag	0759 253 542
2	Janet Smith Jones	Female	23-06-1980	Melrose 123	NULL	NULL
4	Gloria Williams	Female	14-02-1984	2nd Street 23	NULL	NULL



## > Greater than

The following script gets all the payments that are greater than 2,000 from the payments table.

```
SELECT * FROM `payments` WHERE `amount_paid` > 2000;
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.

payment_id	membership_number	payment_date	description	amount_paid	external_reference_number
1	1	23-07-2012	Movie rental payment	2500	11
3	3	30-07-2012	Movie rental payment	6000	NULL

## <> Not Equal To

The following script gets all the movies whose category id is not 1.

```
SELECT * FROM `movies` WHERE `category_id` <> 1;
```

Executing the above script in MySQL workbench against the “myflixdb” produces the following results.





movie_id	title	director	year_released	category_id
2	Forgetting Sarah Marshal	Nicholas Stoller	2008	2
5	Daddy's Little Girls	NULL	2007	8
6	Angels and Demons	NULL	2007	6
7	Davinci Code	NULL	2007	6
9	Honey mooners	John Schultz	2005	8

## Summary

- The SQL WHERE clause is used to restrict the number of rows affected by a SELECT, UPDATE or DELETE query.
- The WHERE condition in SQL can be used in conjunction with logical operators such as AND and OR, comparison operators such as ,= etc.
- When used with the AND logical operator, all the criteria must be met.
- When used with the OR logical operator, any of the criteria must be met.
- The key word IN is used to select rows matching a list of values.

## Brain Teaser

Let's suppose that we want to get a list of rented movies that have not been returned on time 25/06/2012. We can use the SQL WHERE statement clause together with the less than comparison operator and AND logical operator to achieve that.

```
SELECT * FROM `movierentals` WHERE `return_date` < '2012-06-25' AND movie_returned = 0;
```



Executing the above script in MySQL workbench gives the following results.

reference_number	transaction_date	return_date	membership_number	movie_id	movie_returned
14	21-06-2012	24-06-2012	2	2	0

## You Might Like:

- [What is Normalization in DBMS \(SQL\)? 1NF, 2NF, 3NF Example](#)
- [Database Design in DBMS Tutorial: Learn Data Modeling](#)
- [MySQL Workbench Tutorial: What is, How to Install & Use](#)
- [What is a Database? Definition, Meaning, Types with Example](#)
- [MariaDB Tutorial: Learn Syntax, Commands with Examples](#)

[Prev](#)[Report a Bug](#)[Next](#)[About](#)[About Us](#)[Advertise with Us](#)[Write For Us](#)

[Contact Us](#)

## Career Suggestion

[SAP Career Suggestion Tool](#)

[Software Testing as a Career](#)

## Interesting

[eBook](#)

[Blog](#)

[Quiz](#)

[SAP eBook](#)

## Execute online

[Execute Java Online](#)

[Execute Javascript](#)

[Execute HTML](#)

[Execute Python](#)

