

# MYSQL BIBLE

**Select Query:** To retrieve data from a table.

Eg. `SELECT * FROM Actor` {selects all fields from actor table}

```
SELECT first_name, last_name FROM `actor` {to display more than one field}
```

**Please note: Backticks:** `` used in MYSQL on mac found under ~ button. But in other systems may differ.

### Filtering Data Using WHERE

```
SELECT first_name, last_name FROM `actor` WHERE first_name = 'PENELOPE'
```

Using a NOT indicator to select all but a certain filter we use `<>` in MYSQL (To show all records where first\_name not equal to PENELOPE)

```
SELECT first_name, last_name FROM `actor` WHERE first_name <> 'PENELOPE'
```

Or another NOT Indicator is `!=`

```
SELECT first_name, last_name FROM `actor` WHERE first_name != 'PENELOPE'
```

### Filtering not based on a string

```
SELECT first_name, last_name FROM `actor` WHERE actor_id = 5
```

```
SELECT first_name, last_name FROM `actor` WHERE actor_id <= 5
```

```
SELECT first_name, last_name FROM `actor` WHERE actor_id >= 5
```

### Selecting a Range (where actor ID is greater than 3 but less than 5)

```
SELECT actor_id, first_name, last_name FROM `actor` WHERE actor_id > 3 AND actor_id < 5
```

### Using AND OR

```
SELECT actor_id, first_name, last_name FROM `actor` WHERE first_name = 'PENELOPE' OR actor_id < 5 OR first_name = 'NICK'
```

### Selecting multiple records using IN (IN Means matches any of the following)

```
SELECT * FROM `actor` Where first_name IN ('PENELOPE', 'NICK', 'ED')
```

```
SELECT * FROM `actor` Where first_name NOT IN ('PENELOPE', 'NICK', 'ED')
```

Using WILDCARDS (%or \_ allows to pull up anything similar)

Using % Wildcard

```
SELECT * FROM `actor` Where first_name Like 'John%'
```

**Result:**

Showing rows 0 - 2 (3 total, Query took 0.0011 seconds.)

```
SELECT * FROM `actor` Where first_name Like 'John%'
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by k

Options

	actor_id	first_name	last_name	last_update
<input type="checkbox"/> Edit Copy Delete	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
<input type="checkbox"/> Edit Copy Delete	40	JOHNNY	CAGE	2006-02-15 04:34:33
<input type="checkbox"/> Edit Copy Delete	192	JOHN	SUVARI	2006-02-15 04:34:33

☐ Check all | With selected: Edit Copy Delete Export

Using `_` Wildcard (more restrictive it will only return based on a character) not similar.

```
SELECT * FROM `actor` where first_name Like 'Ja_ne'
```

Result:

✓ Showing rows 0 - 2 (3 total, Query took 0.0011 seconds.)

SELECT \* FROM `actor` where first\_name Like 'Ja\_ne'

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by:

Options

		actor_id	first_name	last_name	last_update
<input type="checkbox"/>	Edit Copy Delete	62	JAYNE	NEESON	2006-02-15 04:34:33
<input type="checkbox"/>	Edit Copy Delete	150	JAYNE	NOLTE	2006-02-15 04:34:33
<input type="checkbox"/>	Edit Copy Delete	195	JAYNE	SILVERSTONE	2006-02-15 04:34:33

You cannot combine LIKE with IN

Combining AND Or LIKE

```
SELECT * FROM address where district= 'Buenos Aires' And (address like '%E%' or address LIKE '%AL%')
```

Result:

✓ Showing rows 0 - 2 (3 total, Query took 0.0015 seconds.)

SELECT \* FROM address where district= 'Buenos Aires' And (address like '%E%' or address LIKE '%AL%')

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Options

		address_id	address	address2	district	city_id	postal_code	phone	last_update
<input type="checkbox"/>	Edit Copy Delete	93	1926 El Alto Avenue		Buenos Aires	289	75543	846225459260	2006-02-15 04:45:30
<input type="checkbox"/>	Edit Copy Delete	223	1244 Allappuzha (Alleppey) Place		Buenos Aires	567	20657	991802825778	2006-02-15 04:45:30
<input type="checkbox"/>	Edit Copy Delete	591	773 Dallas Manor		Buenos Aires	424	12664	914466027044	2006-02-15 04:45:30

↑ ☐ Check all With selected: Edit Copy Delete Export

To sort alphabetically

```
SELECT actor_id, first_name, Last_name FROM actor where first_name= 'PENELOPE' Order by last_name ASC
```

SQL requires queries need to be in certain order:

```
SELECT, FROM, WHERE, ORDER BY, ASC/DESC.
```

String Function is a Formula you can use on text (String just means text)  
Length FX

```
SELECT first_name, Length( first_name) from actor
```

Result:

✓ Showing rows 0 - 24 (200 total, Query took 0.0009 seconds.)

```
SELECT first_name, Length( first_name) from actor
```

1 > >> | ☐ Show all | Number of rows: 25

+ Options

	first_name	Length( first_name)
<input type="checkbox"/> Edit  Copy  Delete	PENELOPE	8
<input type="checkbox"/> Edit  Copy  Delete	NICK	4
<input type="checkbox"/> Edit  Copy  Delete	ED	2
<input type="checkbox"/> Edit  Copy  Delete	JENNIFER	8
<input type="checkbox"/> Edit  Copy  Delete	JOHNNY	6
<input type="checkbox"/> Edit  Copy  Delete	BETTE	5
<input type="checkbox"/> Edit  Copy  Delete	GRACE	5
<input type="checkbox"/> Edit  Copy  Delete	MATTHEW	7

Concatenate: To join strings of text

```
SELECT CONCAT( first_name," ", Last_name ) from actor
```

Results:

✓ Showing rows 0 - 24 (200 total, Query took 0.0009 seconds.)

```
SELECT CONCAT( first_name," ", Last_name ) from actor
```

1 > >> | ☐ Show all | Number of rows: 25

+ Options

```
CONCAT( first_name," ", Last_name )
```

PENELOPE GUINNESS
NICK WAHLBERG
ED CHASE
JENNIFER DAVIS
JOHNNY LOLOBRIGIDA

In oracle and postgres || || in microsoft + + can be used to join text together

✓ Showing rows 0 - 24 (200 total, Query took 0.0009 seconds)

```
1 SELECT first_name + ' ' + last_name
2 FROM actor
```

To combine two string fx

```
SELECT CONCAT( first_name," ", Last_name),LENGTH(CONCAT(first_name," ",last_name)) from actor
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0017 seconds.)

```
SELECT CONCAT( first_name," ", Last_name),LENGTH(CONCAT(first_name," ",last_name))
```

1

>

>>

☐

Show all

Number of rows:

25

Filter rows:

Search

- Options

**CONCAT( first\_name," ", Last\_name)** **LENGTH(CONCAT(first\_name," ",last\_name))**

PENELOPE GUINNESS	16
NICK WAHLBERG	13
ED CHASE	8
JENNIFER DAVIS	14
JOHNNY LOLLOBRIGIDA	19

Orderby can be used on joined string fx

```
SELECT CONCAT( first_name," ", Last_name),LENGTH(CONCAT(first_name," ",last_name)) from actor
Order by LENGTH(CONCAT(first_name," ",last_name)) DESC
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0016 seconds.)

```
SELECT CONCAT( first_name," ", Last_name),LENGTH(CONCAT(first_name," ",last_name)) from actor Order by LENGTH(CONCAT(first_name," ",last_name)) DESC
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create ]

1

>

>>

☐

Show all

Number of rows:

25

Filter rows:

Search this table

Sort by key:

None

- Options

**CONCAT( first\_name," ", Last\_name)** **LENGTH(CONCAT(first\_name," ",last\_name))** ▾ 1

MICHELLE MCCONAUGHEY	20
JOHNNY LOLLOBRIGIDA	19
ANGELA WITHERSPOON	18
JAYNE SILVERSTONE	17
CAMERON ZELLWEGER	17
MATTHEW JOHANSSON	17
CHRISTOPHER BERRY	17
FRANCES DAY-LEWIS	17
OLYMPIA PFEIFFER	16
CARY MCCONAUGHEY	16
LAURENCE BULLOCK	16
ANGELINA ASTAIRE	16

Changing the Case using UPPER or LOWER

```
SELECT LOWER (first_name) FROM actor
```

```
SELECT LOWER (first_name) FROM actor order by first_name ASC
```

Making Proper text

That is: First letter capital and rest lower case:

```
SELECT CONCAT(LEFT(first_name,1), LOWER(RIGHT(first_name,length(first_name)-1)))FROM actor
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0013 seconds.)

```
SELECT CONCAT(LEFT(first_name,1), LOWER(RIGHT(first_name,length(first_name)-1)))FROM actor
```

1

>

>>

|

☐ Show all

|

Number of rows:

25

Filter rows:

Se

+ Options

```
CONCAT(LEFT(first_name,1), LOWER(RIGHT(first_name,length(first_name)-1)))
```

Penelope

Nick

Ed

Jennifer

Johnny

Bette

Grace

Matthew

To select a partial string eg, the first letter of the first name

```
SELECT substring(first_name,1,1)FROM actor
```

The first 1 represents the first letter and the second 1 represents the number of characters

✓ Showing rows 0 - 24 (200 total, Query took 0.0010)

SELECT substring(first\_name,1,1)FROM actor

1 ▾ > >> | ☐ Show all | Number of

Options

substring(first\_name,1,1)

P

N

E

J

J

```
SELECT substring(first_name,2)FROM actor
```

✓ Showing rows 0 - 24 (200 total, Query took

SELECT substring(first\_name,2)FROM actor

1 ▾ > >> | ☐ Show all | Nu

Options

substring(first\_name,2)

ENELOPE

ICK

D

ENNIFER

OHNNY

ETTE

Substring can be used in space of combining LEFT, RIGHT and Length

```
SELECT Concat(Left(first_name,1),lower(substring(first_name,2))) FROM actor
```

Or

```
SELECT Concat(Substring(first_name,1,1),lower(substring(first_name,2))) FROM actor
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0013 seconds.)

```
SELECT Concat(Left(first_name,1),lower(substring(first_name,2))) FROM actor
```

1 > >> | ☐ Show all | Number of rows: 25 Filter rows:

+ Options

```
Concat(Left(first_name,1),lower(substring(first_name,2)))
```

Penelope

Nick

Ed

Jennifer

Johnny

Bette

Grace

TRIM Removes leading spaces to help you locate a record that may have spaces

```
SELECT * FROM actor Where TRIM(first_name) = 'GRACE'
```

Modifying redundant text

```
SELECT description as original, trim(Leading 'A 'from description) as modified FROM film_text
```

✓ Showing rows 0 - 24 (1000 total, Query took 0.0013 seconds.)

```
SELECT description as original, trim(Leading 'A 'from description) as modified FROM film_text
```

☐ Profiling [ Edit inline ] [ Edit ] [ Expl ]

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	original	modified
<input type="checkbox"/> Edit Copy Delete	A Epic Drama of a Feminist And a Mad Scientist who...	Epic Drama of a Feminist And a Mad Scientist who m...
<input type="checkbox"/> Edit Copy Delete	A Astounding Epistle of a Database Administrator A...	Astounding Epistle of a Database Administrator And...
<input type="checkbox"/> Edit Copy Delete	A Astounding Reflection of a Lumberjack And a Car ...	Astounding Reflection of a Lumberjack And a Car wh...
<input type="checkbox"/> Edit Copy Delete	A Fanciful Documentary of a Frisbee And a Lumberja...	Fanciful Documentary of a Frisbee And a Lumberjack...
<input type="checkbox"/> Edit Copy Delete	A Fast-Paced Documentary of a Pastry Chef And a De...	Fast-Paced Documentary of a Pastry Chef And a Dent...
<input type="checkbox"/> Edit Copy Delete	A Intrepid Panorama of a Robot And a Boy who must ...	Intrepid Panorama of a Robot And a Boy who must Es...
<input type="checkbox"/> Edit Copy Delete	A Touching Saga of a Hunter And a Butler who must ...	Touching Saga of a Hunter And a Butler who must Di...
<input type="checkbox"/> Edit Copy Delete	A Epic Tale of a Moose And a Girl who must Confron...	Epic Tale of a Moose And a Girl who must Confront ...
<input type="checkbox"/> Edit Copy Delete	A Thoughtful Panorama of a Database Administrator ...	Thoughtful Panorama of a Database Administrator An...
<input type="checkbox"/> Edit Copy Delete	A Action-Packed Tale of a Man And a Lumberjack who...	Action-Packed Tale of a Man And a Lumberjack who m...
<input type="checkbox"/> Edit Copy Delete	A Boring Epistle of a Butler And a Cat who must Fi...	Boring Epistle of a Butler And a Cat who must Figh...
<input type="checkbox"/> Edit Copy Delete	A Fanciful Saga of a Hunter And a Pastry Chef who ...	Fanciful Saga of a Hunter And a Pastry Chef who mu...
<input type="checkbox"/> Edit Copy Delete	A Action-Packed Drama of a Dentist And a Crocodile...	Action-Packed Drama of a Dentist And a Crocodile w...



LEADING Key word tells to only look at the start of a String.  
TRAILING looks from the right

LOCATE/POSITION (POSTGRE) / CHARINDEX(Microsoft) / INSTR Oracle  
To locate the position of characters in a string

```
SELECT Locate('LOPE', First_name) FROM actor
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0010 sec)

```
SELECT Locate('LOPE', First_name) FROM actor
```

1 > >> | ☐ Show all | Number of rows

+ Options

Locate('LOPE', First_name)
5
0
0
-

To Change column headings using an ALIAS AS

```
SELECT CONCAT( first_name," ", Last_name) As Full_Name,LENGTH(CONCAT(first_name," ",last_name))  
As Length from actor Order by LENGTH(CONCAT(first_name," ",last_name)) DESC
```

✓ Showing rows 0 - 24 (200 total, Query took 0.0017 seconds.)

```
SELECT CONCAT( first_name," ", Last_name) As Full_Name,LENGTH(CONCAT(first_name," ",last_name)) As Length from actor Order by LENGTH(CONCAT(first_name," ",last_name)) DESC
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

- Options

Full_Name	Length
MICHELLE MCCONAUGHEY	20
JOHNNY LOLLOBRIGIDA	19
ANGELA WITHERSPOON	18
JAYNE SILVERSTONE	17
CAMERON ZELLWEGER	17
MATTHEW JOHANSSON	17
CHRISTOPHER BERRY	17
FRANCES DAY-LEWIS	17
OLYMPIA PFEIFFER	16
CARY MCCONAUGHEY	16
LAURENCE BULLOCK	16
ANGELINA ASTAIRE	16
MORGAN MCDORMAND	16

Instead of repeating order by with a long expression:

```
SELECT CONCAT( first_name," ", Last_name) As Full_Name,LENGTH(CONCAT(first_name," ",last_name))  
As Length from actor Order by Length ASC
```

Or 

```
SELECT CONCAT( first_name," ", Last_name) As Full_Name,LENGTH(CONCAT(first_name," ",last_name))  
As Length from actor Order by Length DESC
```

To select with a ALIAS using a filter

```
SELECT CONCAT( first_name," ", Last_name) As Full_Name,LENGTH(CONCAT(first_name," ",last_name))  
As Length from actor Where LENGTH(CONCAT(first_name," ",last_name))> 17 Order byLength DESC
```

SELECT CONCAT( first\_name," ", Last\_name) As Full\_Name,LENGTH(CONCAT(first\_name," ",last\_name)) As Length from actor Where LENGTH(CONCAT(first\_name," ",last\_name))> 17 Order by Length DESC

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

Full_Name	Length
MICHELLE MCCONAUGHEY	20
JOHNNY LOLLOBRIGIDA	19
ANGELA WITHERSPOON	18

☐ Show all |  |  |  |

### Data FX

A data function is a fx that can be run on field with a date data type

```
SELECT * FROM address WHERE YEAR (last_update) = '2006'
```

```
SELECT * FROM address WHERE Date (last_update) = '2006-02-15'
```

```
SELECT * FROM address WHERE Date (last_update) = '2006-02-15' to change the date format
```

```
SELECT date_format(last_update, '%m-%d-%y') as update_Date from address
```

✓ Showing rows 0 - 24 (603 total, Query took 0.0009 seconds.)

```
SELECT date_format(last_update, '%m-%d-%y') as update_Date from address
```

1 | > >> | Number of rows: 25 | Filter rows: Search this

+ Options

update_Date
02-15-06
02-15-06
02-15-06
02-15-06
02-15-06
02-15-06
02-15-06
02-15-06

```
SELECT date_format(last_update, '%M-%D,-%Y') as update_Date from address
```

✓ Showing rows 0 - 24 (603 total, Query took 0.0010 seconds.)

```
SELECT date_format(last_update, '%M-%D, -%Y') as update_Date from address
```

1 > >> | Number of rows: 25 Filter rows: Search this table

- Options

update_Date
February-15th,-2006
February-15th,-2006
February-15th,-2006
February-15th,-2006
February-15th,-2006
February-15th,-2006

#### GROUPING SQL

To get a count of rows within a table:

```
SELECT Count(*) FROM address
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)

```
SELECT Count(*) FROM address
```

☐ Show all | Number of rows: 25 Filter rows: Search this table

- Options

Count(*)
603

```
SELECT district, Count(*) FROM address Group by district
```

✓ Showing rows 0 - 24 (378 total, Query took 0.0019 seconds.)

```
SELECT district, Count(*) FROM address Group by district
```

1 > >> | ☐ Show all | Number of rows: 25

+ Options

	district	Count(*)
<input type="checkbox"/> Edit Copy Delete		3
<input type="checkbox"/> Edit Copy Delete	Abu Dhabi	2
<input type="checkbox"/> Edit Copy Delete	Aceh	1
<input type="checkbox"/> Edit Copy Delete	Adana	2
<input type="checkbox"/> Edit Copy Delete	Addis Abeba	1
<input type="checkbox"/> Edit Copy Delete	Aden	1

Aggregating functions such as COUNT should be used with Group BY  
They can also use order by to show which district is the most popular in the database;

```
SELECT district, Count(*) AS DIStCount FROM address Group by district Order by DIStCount DESC
```

✓ Showing rows 0 - 24 (378 total, Query took 0.0019 seconds.)

```
SELECT district, Count(*) AS DIStCount FROM address Group by district Order by DIStCount DESC
```

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	district	DIStCount
<input type="checkbox"/> Edit Copy Delete	Buenos Aires	10
<input type="checkbox"/> Edit Copy Delete	West Bengali	9
<input type="checkbox"/> Edit Copy Delete	California	9
<input type="checkbox"/> Edit Copy Delete	Shandong	9
<input type="checkbox"/> Edit Copy Delete	Uttar Pradesh	8
<input type="checkbox"/> Edit Copy Delete	So Paulo	8
<input type="checkbox"/> Edit Copy Delete	England	7
<input type="checkbox"/> Edit Copy Delete	Maharashtra	7
<input type="checkbox"/> Edit Copy Delete	Southern Tagalog	6
<input type="checkbox"/> Edit Copy Delete	Karnataka	5
<input type="checkbox"/> Edit Copy Delete	Tamil Nadu	5

```
SELECT district, Count(*) AS DIStCount FROM address Where address_id <10 Group by district Order by DIStCount DESC
```

✓ Showing rows 0 - 6 (7 total, Query took 0.0013 seconds.)

```
SELECT district, Count(*) AS DIStCount FROM address Where address_id <10 Group by district Order by DIStCount DESC
```

☐ Profiling [

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	district	DIStCount
<input type="checkbox"/> Edit Copy Delete	Alberta	2
<input type="checkbox"/> Edit Copy Delete	QLD	2
<input type="checkbox"/> Edit Copy Delete	Nagasaki	1
<input type="checkbox"/> Edit Copy Delete	Nantou	1
<input type="checkbox"/> Edit Copy Delete	California	1
<input type="checkbox"/> Edit Copy Delete	Attika	1
<input type="checkbox"/> Edit Copy Delete	Mandalay	1

#### SQL Structure

SELECT

FROM

WHERE

GROUPBY

ORDERBY

```
SELECT MIN(rental_duration) FROM film
```

```
SELECT MAX(rental_duration) FROM film
```

```
SELECT AVG(rental_duration) FROM film
```

```
SELECT AVG(rental_duration) FROM film
```

✓ Showing rows 0 - 4 (5 total, Query took 0.0018 seconds.)

```
SELECT Rating, Sum(rental_duration) FROM film GROUP by rating
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Options

	Rating	Sum(rental_duration)
<input type="checkbox"/> Edit Copy Delete	G	861
<input type="checkbox"/> Edit Copy Delete	PG	986
<input type="checkbox"/> Edit Copy Delete	PG-13	1127
<input type="checkbox"/> Edit Copy Delete	R	931
<input type="checkbox"/> Edit Copy Delete	NC-17	1080

☐ Check all
 With selected:
 Edit
 Copy
 Delete
 Exp

Having Keyword: This is Filter for the GROUP BY KEY WORD

```
SELECT district, Count(*) as ct From address WHERE district Like '%b%' Group BY district HAVING ct>8 Order By ct DESC
```

If we wanted to see district with the most frequent that has B in it

✓ Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

```
SELECT district, Count(*) as ct From address WHERE district Like '%b%' Group BY district HAVING ct>8 Order By ct DESC
```

☐ Profiling [ Edit inline ] [ E

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	district	ct
<input type="checkbox"/> Edit Copy Delete	Buenos Aires	10
<input type="checkbox"/> Edit Copy Delete	West Bengali	9

↑ ☐ Check all | With selected: Edit Copy Delete Export

```
SELECT Avg(Year(last_update)) FROM `actor`
```

To get the avg of a string function like year

Distinct Keyword

```
SELECT Distinct district FROM address
```

To see how many staff we have in rental

```
SELECT DISTINCT staff_id FROM rental
```

✓ Showing rows 0 - 1 (2 total, Query took 0.0009 seconds.)

```
SELECT DISTINCT staff_id FROM rental
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | S

- Options

	staff_id
<input type="checkbox"/> Edit Copy Delete	1
<input type="checkbox"/> Edit Copy Delete	2

↑ ☐ Check all | With selected: Edit Copy Delete Export

To see how many customers served

```
SELECT DISTINCT customer_ID FROM rental
```

To see how many customer served by each staff

```
SELECT DISTINCT staff_id, customer_ID FROM rental
```

✓ Showing rows 0 - 24 (1198 total, Query took 0.0013 seconds.)

```
SELECT DISTINCT staff_id, customer_ID FROM rental
```

1 > >> | Number of rows: 25 Filter rows: Search this table

+ Options

			staff_id	customer_ID
<input type="checkbox"/>	Edit	Copy	Delete	1 130
<input type="checkbox"/>	Edit	Copy	Delete	1 459
<input type="checkbox"/>	Edit	Copy	Delete	1 408
<input type="checkbox"/>	Edit	Copy	Delete	2 333
<input type="checkbox"/>	Edit	Copy	Delete	1 222
<input type="checkbox"/>	Edit	Copy	Delete	1 549
<input type="checkbox"/>	Edit	Copy	Delete	2 269
<input type="checkbox"/>	Edit	Copy	Delete	2 239
<input type="checkbox"/>	Edit	Copy	Delete	1 126
<input type="checkbox"/>	Edit	Copy	Delete	2 399

Select distinct is used to remove duplicate values

To see how many customers have rented the same DVD twice ( or bought same inventory)

```
SELECT DISTINCT customer_ID, inventory_id
FROM rental
```

```
SELECT CONCAT(customer_ID, inventory_id) as conc, count(*) as ct FROM rental GROUP by conc Order
by ct DESC
```

If we wanted to create a unique ID for customer and rental to see which customer has rented twice and we aggregate using count to group and find any information for which customers has rented twice

✓ Showing rows 0 - 24 (16040 total, Query took 0.0306 seconds.)

```
SELECT CONCAT(customer_ID, inventory_id) as conc, count(*) as ct FROM rental GROUP by conc Order by ct DESC
```

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

Options

conc	ct
0263	2
93766	2
62706	2
23739	2
042859	1
533299	1
531073	1
44548	1
302708	1
393884	1
481834	1

When making a joint ID be sure to distinct spaces so that the new ID can be clear and accurate

```
SELECT CONCAT(customer_ID,"_" inventory_id) as conc, count(*) as ct FROM rental GROUP by conc
Order by ct DESC
```

✓ Showing rows 0 - 24 (16044 total, Query took 0.0364 seconds.)

```
SELECT CONCAT(customer_ID, '-', inventory_id) as conc, count(*) as ct FROM rental GROUP by conc Order by ct DESC
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Expl](#) ]

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

conc	ct
573_165	1
212_3270	1
291_162	1
5_2177	1
352_501	1
460_576	1
146_1494	1
267_3466	1

Distinct can be used with Year

```
SELECT DISTINCT Year(last_update) FROM address
```

Count can be used with Distinct

```
SELECT Count(DISTINCT district) FROM address
```

Merging Rows: if you wanted all the phone numbers in one row grouped by city:

```
SELECT district, GROUP_CONCAT(Phone) FROM address Group BY district
```

In a case where you would want to call all your customers from one city:

✓ Showing rows 0 - 24 (378 total, Query took 0.0017 seconds.)

```
SELECT district, GROUP_CONCAT(Phone) FROM address Group BY district
```

☐ Show all | Number of rows: 25 Filter rows: Search this table

+ Options

	district	GROUP_CONCAT(Phone)
<input type="checkbox"/> Edit Copy Delete		9980097777982,282667506728,934352415130
<input type="checkbox"/> Edit Copy Delete	Abu Dhabi	985109775584,412903167998
<input type="checkbox"/> Edit Copy Delete	Aceh	932992626595
<input type="checkbox"/> Edit Copy Delete	Adana	697151428760,834418779292
<input type="checkbox"/> Edit Copy Delete	Addis Abeba	47808359842
<input type="checkbox"/> Edit Copy Delete	Aden	756460337785
<input type="checkbox"/> Edit Copy Delete	Adygea	262744791493
<input type="checkbox"/> Edit Copy Delete	Ahal	764477681869
<input type="checkbox"/> Edit Copy Delete	al-Daqahliya	171973024401
<input type="checkbox"/> Edit Copy Delete	al-Manama	864392582257
<input type="checkbox"/> Edit Copy Delete	al-Qadarif	765957414528

```
SELECT district, GROUP_CONCAT(Phone Order by phone ASC SEPARATOR ';') FROM address Group BY district to include a semicolon in between numbers
```



✓ Showing rows 0 - 24 (378 total, Query took 0.0018 seconds.)

```
SELECT district, GROUP_CONCAT(Phone Order by phone ASC SEPARATOR ';') FROM address Group BY district
```

Profiling

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key:

Options

	district	GROUP_CONCAT(Phone Order by phone ASC SEPARATOR ';')
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		282667506728;934352415130;998009777982
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Abu Dhabi	412903167998;985109775584
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Aceh	932992626595
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Adana	697151428760;834418779292
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Addis Abeba	47808359842
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Aden	756460337785
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Adygea	262744791493
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	Ahal	764477681869
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	al-Daqahliya	171973024401
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	al-Manama	864392582257

To make query faster you can use LIMIT to reduce number of rows: (in Microsoft it is TOP in oracle it is an expression: select \* from table ROWNUM,=25)

```
SELECT * FROM payment LIMIT 25
```

✓ Showing rows 0 - 24 (25 total, Query took 0.0008 seconds.)

```
SELECT * FROM payment LIMIT 25
```

Profiling [ Edit inline ]

+ Options

	payment_id	customer_id	staff_id	rental_id	amount	payment_date	last_update
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	1	76	2.99	2005-05-25 11:30:37	2006-02-15 22:12:30
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	1	573	0.99	2005-05-28 10:35:23	2006-02-15 22:12:30

Merge data across Tables

Using JOIN

```
SELECT * FROM customer Join address On customer.address_id = address.address_id
```

✓ Showing rows 0 - 24 (599 total, Query took 0.0015 seconds.)

```
SELECT * FROM customer Join address On customer.address_id = address.address_id
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create ]

1 > >> | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

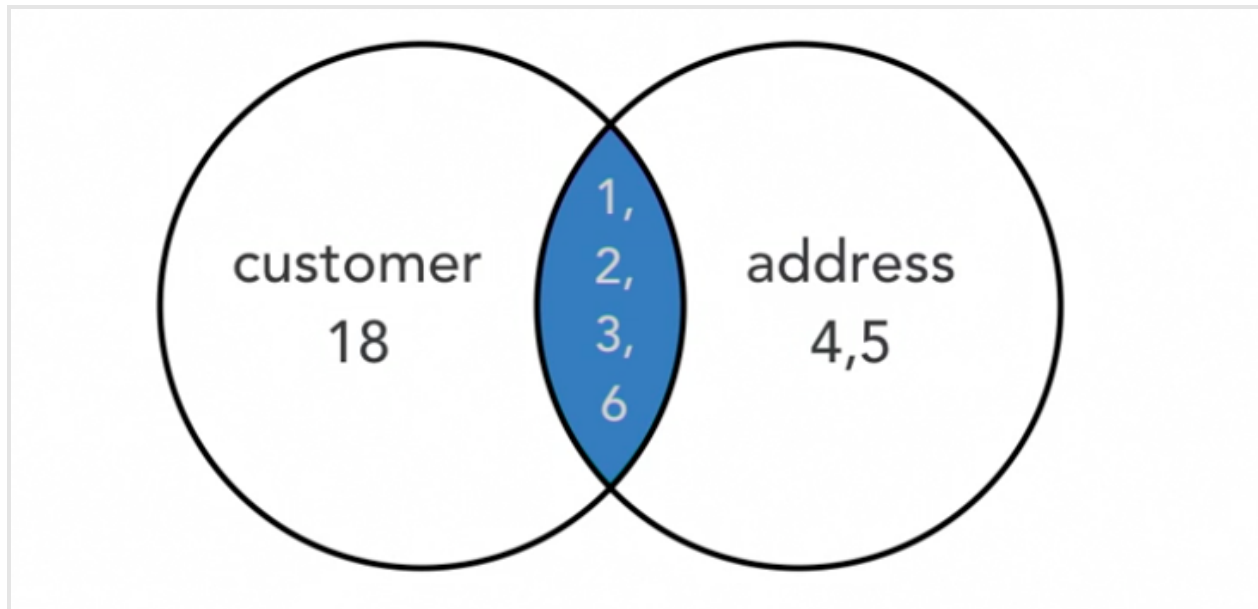
+ Options

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update	address
1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20	
2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20	
3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20	
4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20	
5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20	
6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20	

We could use aliases as well to join

```
SELECT * FROM customer as c Join address as a On c.address_id = a.address_id
```

~Join is short hand for innerJOIN which returns records found in both tables



We can add Filter with a WHERE but it comes after ON

```
SELECT * FROM customer as c
Join address as a
On c.address_id = a.address_id
WHERE district = 'Buenos Aires'
```

A report can be pulled up with a join:

```
SELECT c.first_name, c.last_name, a.address FROM customer as c Join address as a On c.address_id
= a.address_id WHERE district = 'Buenos Aires'
```

✓ Showing rows 0 - 9 (10 total, Query took 0.0025 seconds.)

```
SELECT c.first_name, c.last_name, a.address FROM customer as c Join address as a On c.address_id = a.address_id WHERE district = 'Buenos Aires'
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [View](#) ]

☐ Show all | Number of rows:  | Filter rows:  | Sort by key:

+ Options

first_name	last_name	address
JULIA	FLORES	1926 El Alto Avenue
FLORENCE	WOODS	1532 Dzerzinsk Way
WILLIE	HOWELL	1244 Allappuzha (Alleppey) Place
JASON	MORRISSEY	1427 A Corua (La Corua) Place
WILLIE	MARKHAM	1623 Kingstown Drive
LEONARD	SCHOFIELD	88 Nagaon Manor
MICHEAL	FORMAN	203 Tambaram Street
DARRYL	ASHCRAFT	166 Jinchang Street
JORDAN	ARCHULETA	1229 Varanasi (Benares) Manor
PERRY	SWAFFORD	773 Dallas Manor

## Union

### Joins Tables via columns

```
SELECT 'actor' as tbl, Date(last_update) FROM actor UNION ALL Select 'address' as tbl, Date(last_update) FROM address
```

✓ Showing rows 0 - 24 (803 total, Query took 0.0015 seconds.)

SELECT 'actor' as tbl, Date(last\_update) FROM actor UNION ALL Select 'address' as tbl, Date(last\_update) FROM address

Profiling [ E]

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

tbl	Date(last_update)
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15

Use union all allowed 200+ dat updates for actors and 600+ updates for address to be combined in a one table.

A filter canalso be put on a union

```
SELECT 'actor' as tbl, Date(last_update) FROM actor where actor_id<5 UNION ALL Select 'address' as tbl, Date(last_update) FROM address where city_id<5
```

✓ Showing rows 0 - 7 (8 total, Query took 0.0016 seconds.)

SELECT 'actor' as tbl, Date(last\_update) FROM actor where actor\_id<5 UNION ALL Select 'address' as tbl, Date(last\_update) FROM address where city\_id<5

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PT

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

tbl	Date(last_update)
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
actor	2006-02-15
address	2006-02-15
address	2006-02-15
address	2006-02-15
address	2006-02-15

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

## Using IN to Merge data among tables

```
SELECT * FROM rental WHERE customer_id IN (SELECT customer_id FROM customer Where first_name = 'Mary')
```

Showing rows 0 - 24 (32 total, Query took 0.0016 seconds.)

```
SELECT * FROM rental WHERE customer_id IN (SELECT customer_id FROM customer Where first_name = 'Mary')
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#)

1 > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update
<input type="checkbox"/> Edit Copy Delete	76	2005-05-25 11:30:37	3021	1	2005-06-03 12:00:37	2	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	573	2005-05-28 10:35:23	4020	1	2005-06-03 06:32:23	1	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	1185	2005-06-15 00:54:12	2785	1	2005-06-23 02:42:12	2	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	1422	2005-06-15 18:02:53	1021	1	2005-06-19 15:54:53	2	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	1476	2005-06-15 21:08:46	1407	1	2005-06-25 02:26:46	1	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	1725	2005-06-16 15:18:57	726	1	2005-06-17 21:05:57	1	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	2308	2005-06-18 08:41:48	197	1	2005-06-22 03:36:48	2	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	2363	2005-06-18 13:33:59	3497	1	2005-06-19 17:40:59	1	2006-02-15 21:30:53
<input type="checkbox"/> Edit Copy Delete	3284	2005-06-21 06:24:45	4566	1	2005-06-28 03:28:45	1	2006-02-15 21:30:53

## Select from a Selection with Subqueries

```
select F.first_name FROM (SELECT first_name FROM actor) as F
```

Another example:

```
select F.fn FROM (SELECT first_name as fn FROM actor) as F
```

## VIEWS:

```
SHOW CREATE VIEW actor_info
```

Returns info on how a view was created:

+ Options

View	Create View	character_set_client	collation_connection
actor_info	CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY INVOKER VIEW `actor_info` AS select 'a'.actor_id AS `actor_id`,`a`.`first_name` AS `first_name`,`a`.`last_name` AS `last_name`,group_concat(distinct concat('c`.`name`,`', (select group_concat('f`.`title` order by `f`.`title` ASC separator ',' ) from ((`film` `f` join `film_category` `fc` on(`f`.`film_id` = `fc`.`film_id`)) join `film_actor` `fa` on(`f`.`film_id` = `fa`.`film_id`)) where `fc`.`category_id` = `c`.`category_id` and `fa`.`actor_id` = `a`.`actor_id`)) order by `c`.`name` ASC separator ',' ) AS `film_info` from (((`actor` `a` left join `film_actor` `fa` on(`a`.`actor_id` = `fa`.`actor_id`)) left join `film_category` `fc` on(`fa`.`film_id` = `fc`.`film_id`)) left join `category` `c` on(`fc`.`category_id` = `c`.`category_id`)) group by `a`.`actor_id`,`a`.`first_name`,`a`.`last_name`	utf8	utf8_general_ci

Query results operations

## output:

actor_id	first_name	last_name	film_info
1	PENELOPE	GUINESS	Animation: ANACONDA CONFESSIONS; Children: LANGUAGE COWBOY; Classics: COLOR PHILADELPHIA, WESTWARD SEABISCUIT; Comedy: VERTIGO NORTHWEST; Documentary: ACADEMY DINOSAUR; Family: KING EVOLUTION, SPLASH GUMP; Foreign: MULHOLLAND BEAST; Games: BULWORTH COMMANDMENTS, HUMAN GRAFFITI; Horror: ELEPHANT TROJAN, LADY STAGE, RULES HUMAN; Music: WIZARD COLDBLOODED; New: ANGELS LIFE, OKLAHOMA JUMANJI; Sci-Fi: CHEAPER CLYDE; Sports: GLEAMING JAWBREAKER

## Using Variables

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
SET @Variable1 = 5
```

✓ Showing rows 0 - 3 (4 total, Query took 0.0011 seconds.)

```
SELECT * FROM actor where actor_id < @Variable1
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Export \]](#)

☐ Show all | Number of rows:  | Filter rows:  | Sort by key:

+ Options

	actor_id	first_name	last_name	last_update
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	1	PENELOPE	GUINNESS	2006-02-15 04:34:33
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	2	NICK	WAHLBERG	2006-02-15 04:34:33
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	3	ED	CHASE	2006-02-15 04:34:33
<input type="checkbox"/> <a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	4	JENNIFER	DAVIS	2006-02-15 04:34:33

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)