# MYSQL BIBLE

**Select Query:  To retrieve data from a tale.**
Eg. SELECT * FROM Actor {selects all fields from actor table}

```
SELECT first_name, last_name FROM `actor` {to display more than on fields}
```

***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'
```

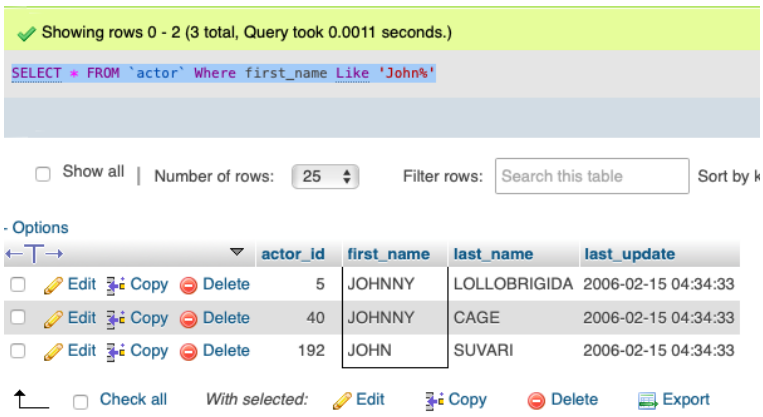
**Selecting multiple records using IN (IN Means matches any of the following)**

```
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%'
```

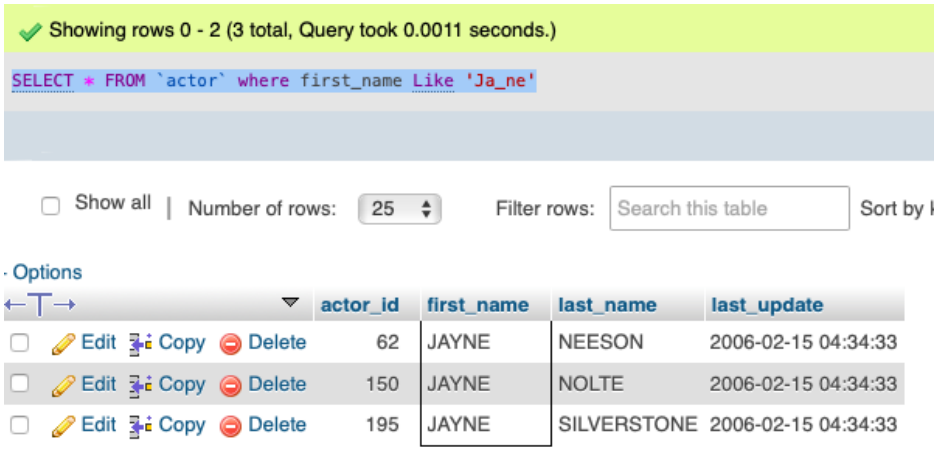
**Result:**



**Using _ Wildcard (more restrictive it will oly return based on a character) not similiar.**

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

Result:

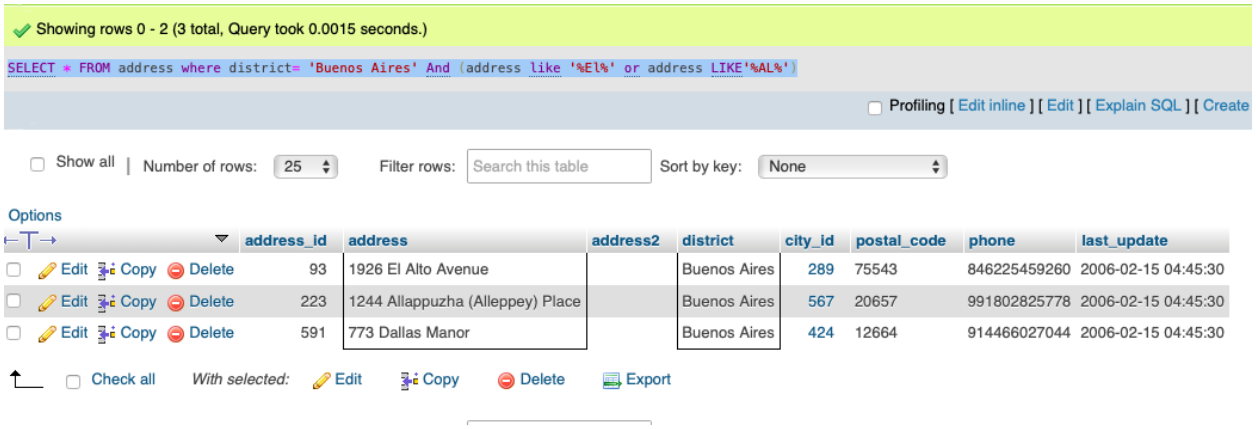



**You cannot combine LIKE with IN**

**Combining AND Or LIKE**

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

Result:




**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

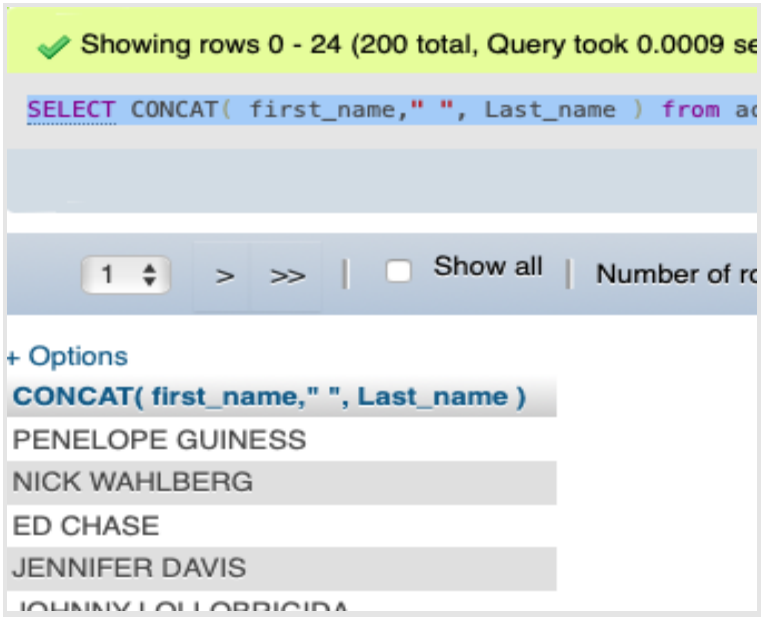
+ Options

| | first_name | Length( first_name) |
|---|---|---|
| Edit Copy Delete | PENELOPE | 8 |
| Edit Copy Delete | NICK | 4 |
| Edit Copy Delete | ED | 2 |
| Edit Copy Delete | JENNIFER | 8 |
| Edit Copy Delete | JOHNNY | 6 |
| Edit Copy Delete | BETTE | 5 |
| Edit Copy Delete | GRACE | 5 |
| Edit Copy Delete | MATTHEW | 7 |

Concatenate: To join strings of text

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

Results:




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

Showing rows 0 - 24 (200 total, Query took 0.0009 secon

```
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:

- Options

| CONCAT( first_name," ", Last_name) | LENGTH(CONCAT(first_name," ",last_name)) |
|---|---|
| PENELOPE GUINESS | 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
```

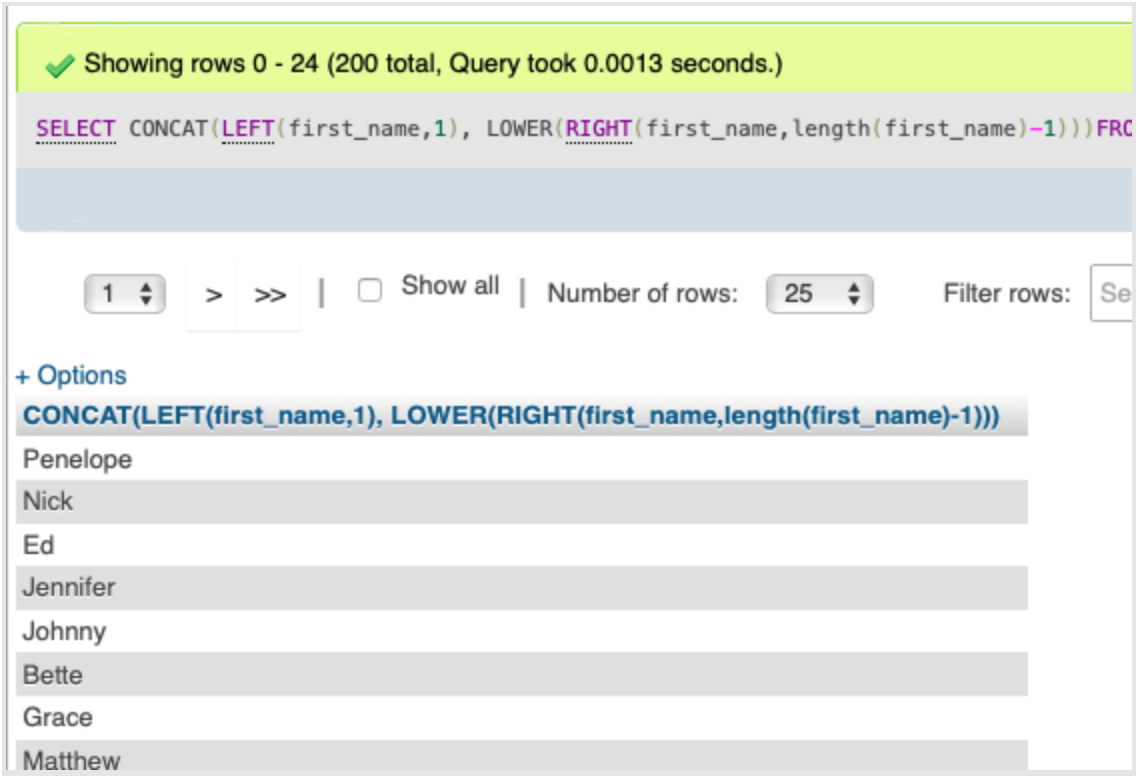
Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Creat

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 |

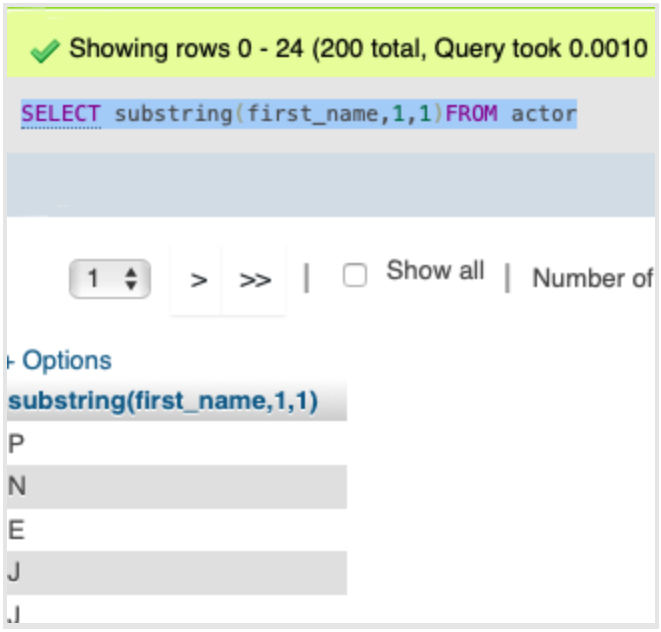
```
Changing the Case using UPPER or LOWER

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
```



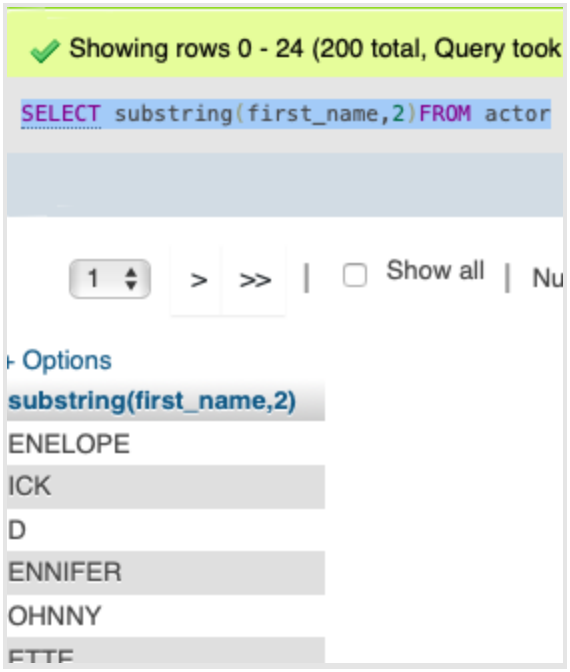
To select a partial string eg, the first letter of the first name

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

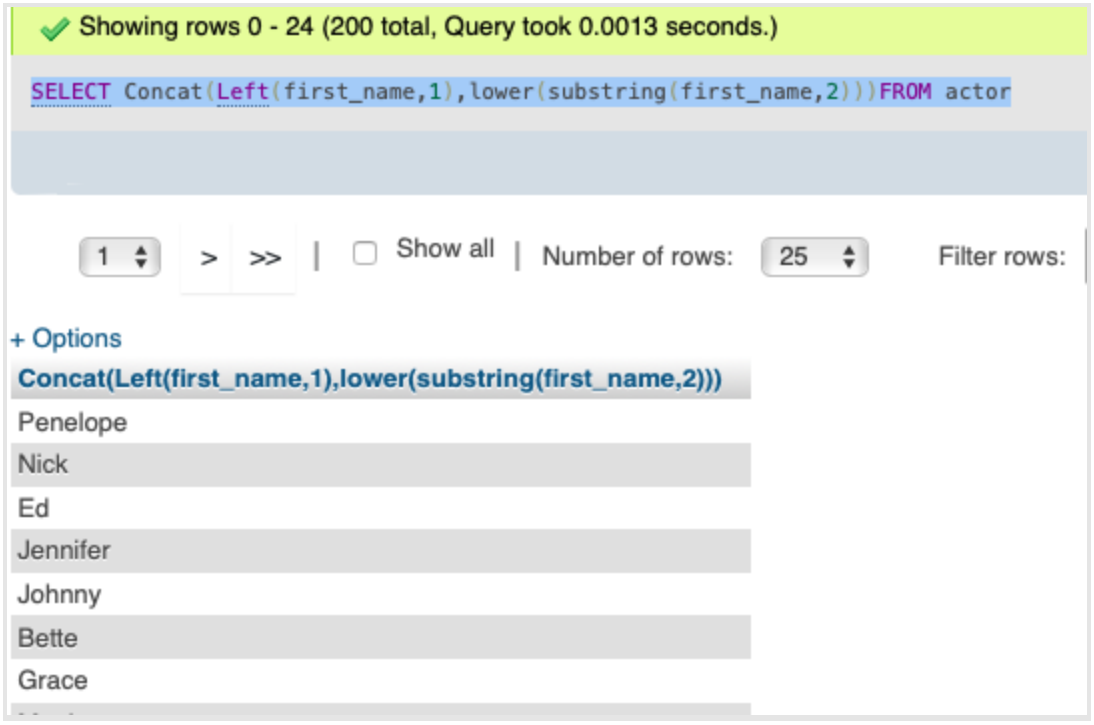
The first 1 represents the first letter and the second 1 represents the number of characters




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

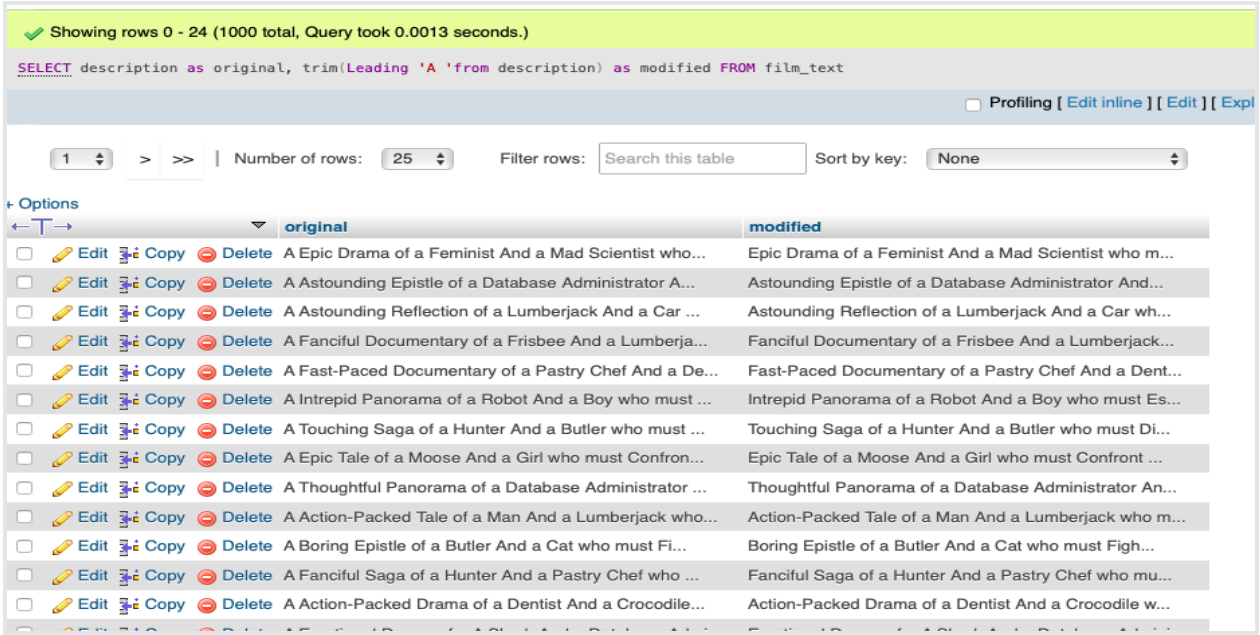


```
Substring can be used in space of combining LEFT, RIGHT and Length
SELECT Concat(Left(first_name,1),lower(substring(first_name,2)))FROM actor
Or
SELECT Concat(Substring(first_name,1,1),lower(substring(first_name,2)))FROM actor
```




```
TRIM Removes leading spaces to help you locate a record that may have spaces
SELECT * FROM actor Where TRIM(first_name)= 'GRACE'
```

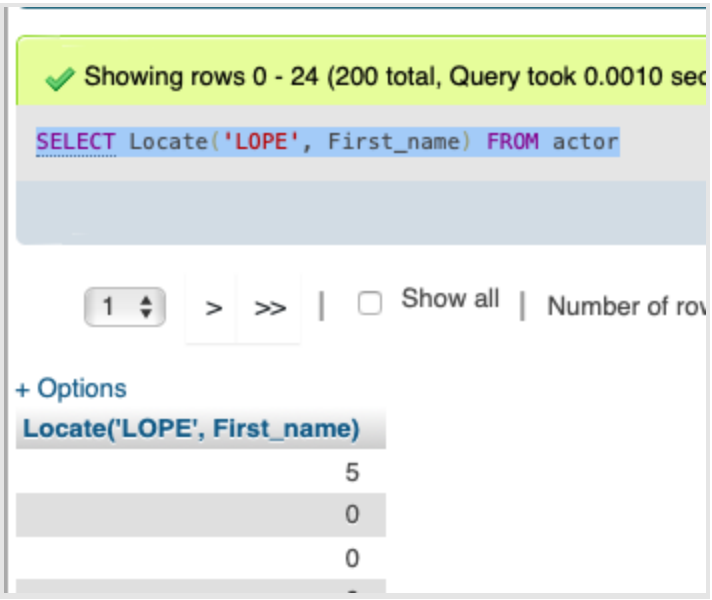
```
Modifying redundant text
SELECT description as original, trim(Leading 'A 'from description) as modified FROM film_text
```



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

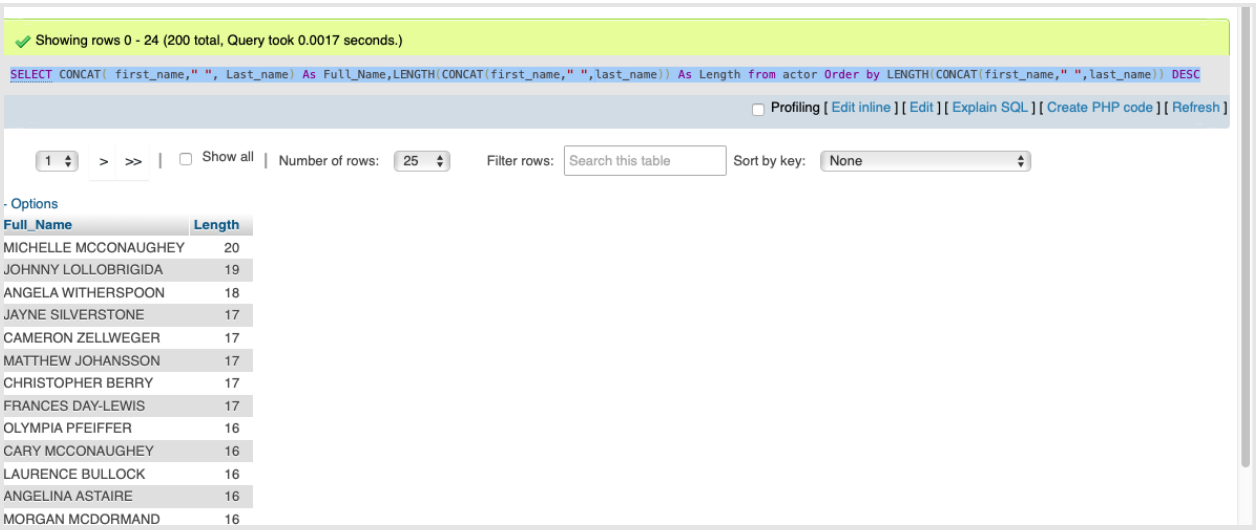
LOCATE/POSITION (POSTGRE)/ CHARINDEX(Microsoft)/INSTR Oracle
To locate the position of characters in a string

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




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
```




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`

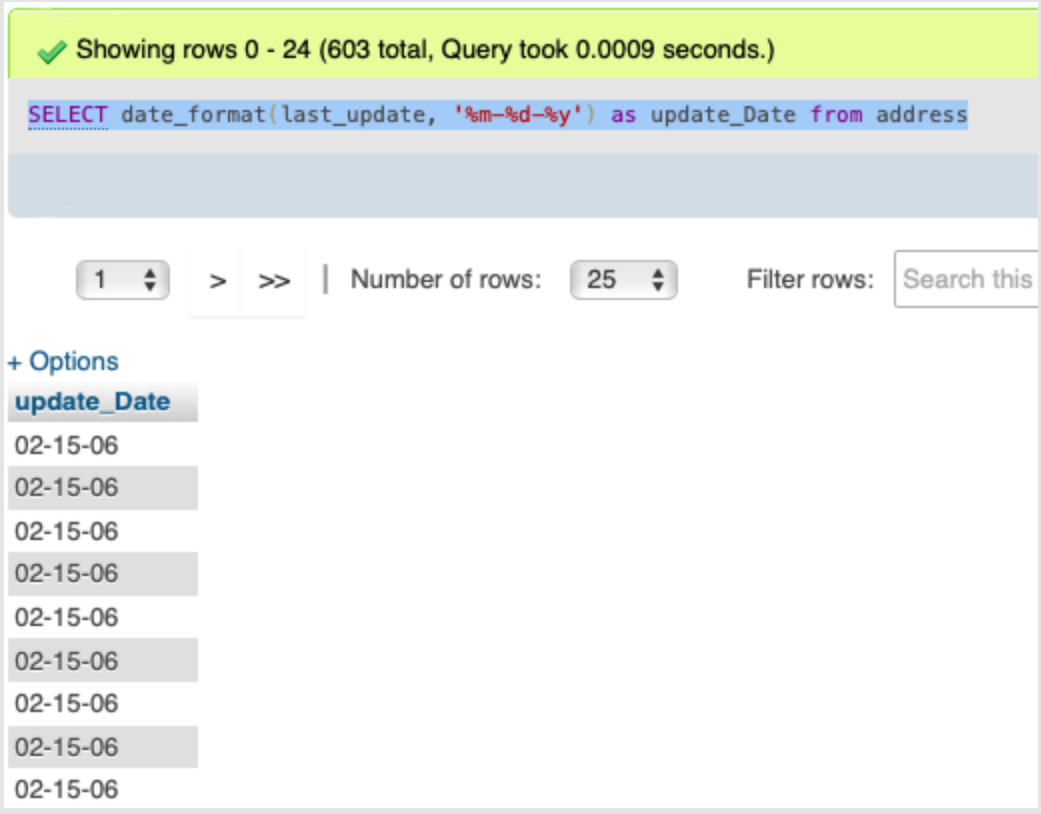
**To select with a ALIAS using a filter**

```
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
```


**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
```

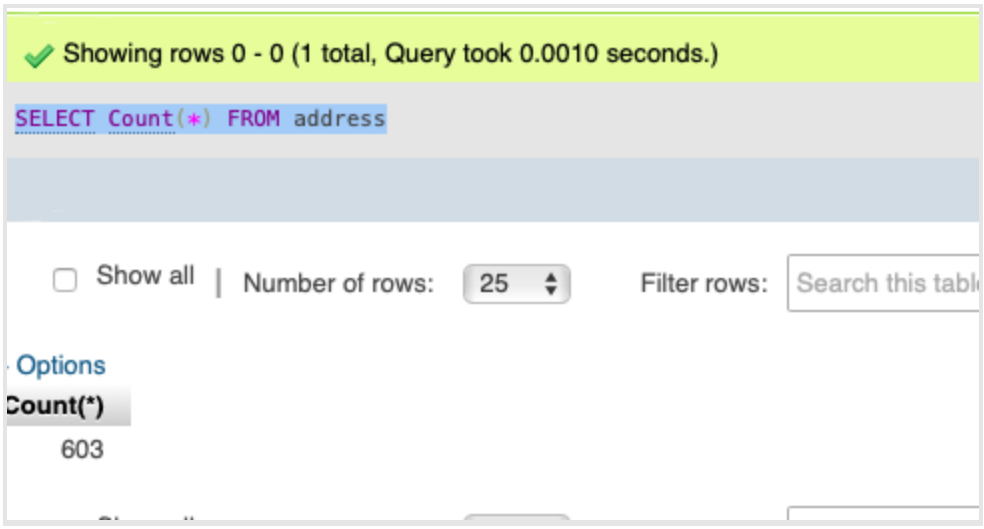



```
SELECT date_format(last_update, '%M-%D,-%Y') as update_Date from address
```

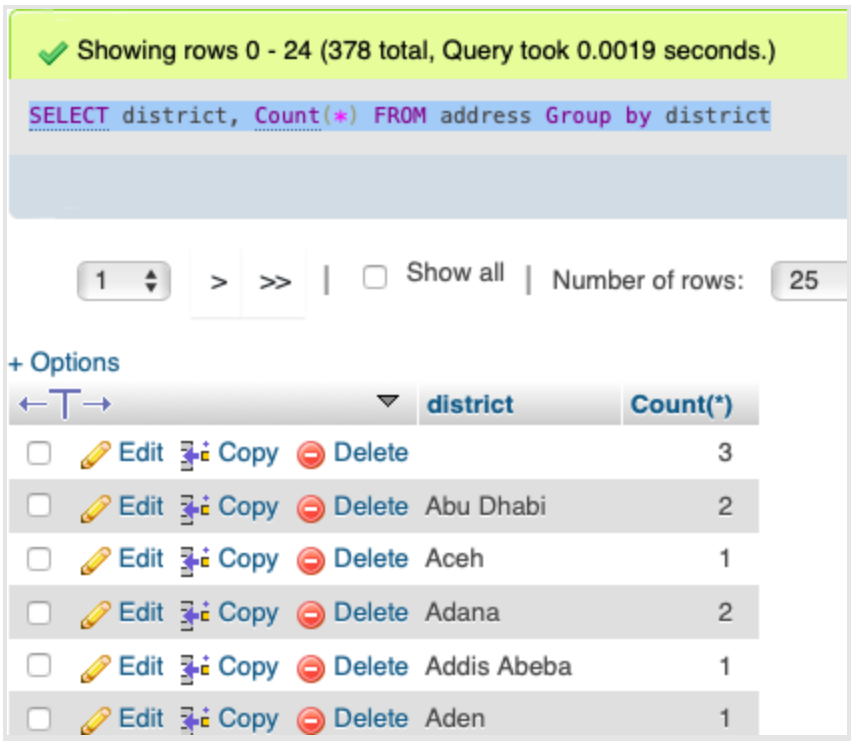

**GROUPING SQL**

**To get a count of rows within a table:**

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

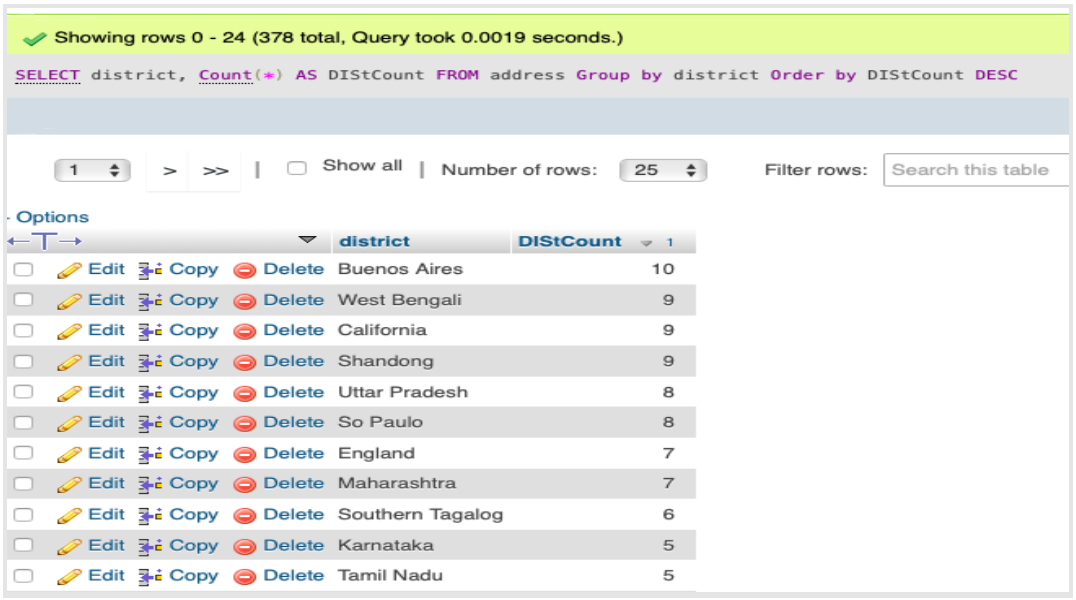



```
SELECT district, Count(*) FROM address Group by district
```

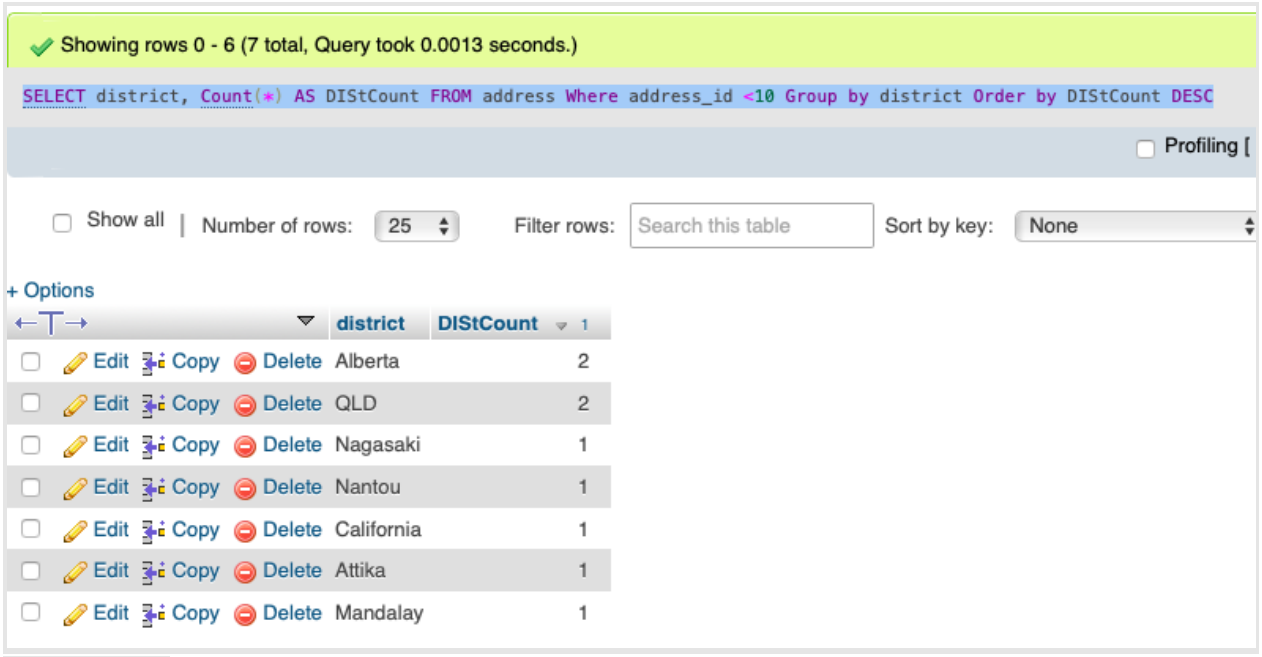



**Aggregating functions such as COUNT should be used with Group BY**
**They can also use order by to show which district is the most popular in the database;**

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




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



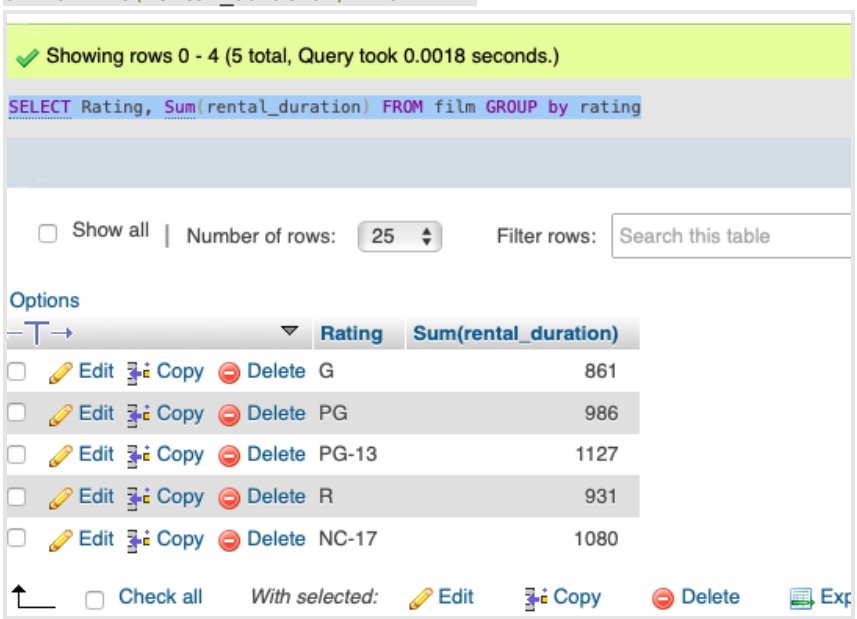

```
SQL Structure
SELECT
FROM
WHERE
GROUPBY
ORDERBY
```

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

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

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

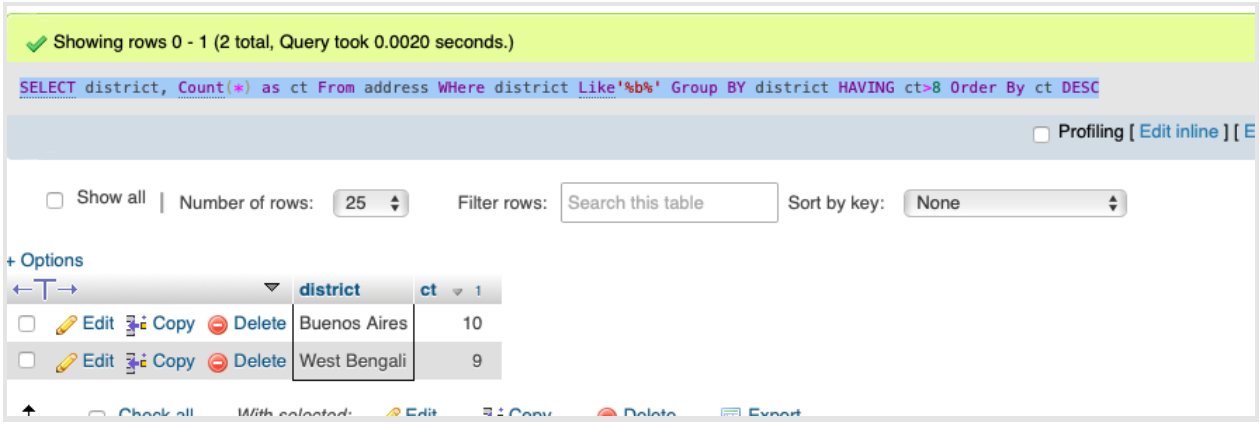
```
SELECT AVG(rental_duration) FROM film
```



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




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

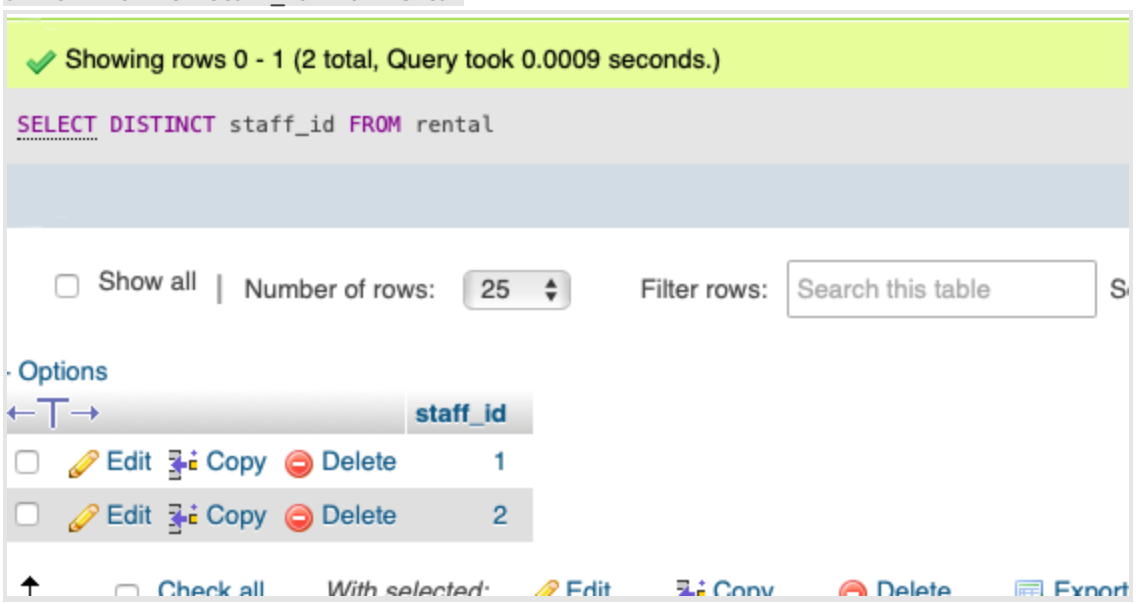
To get the avg of a string function like year

Distinct Keyword

```
SELECT Distinct district FROM address
```

To see how many staff we have in rental

```
SELECT DISTINCT staff_id FROM rental
```

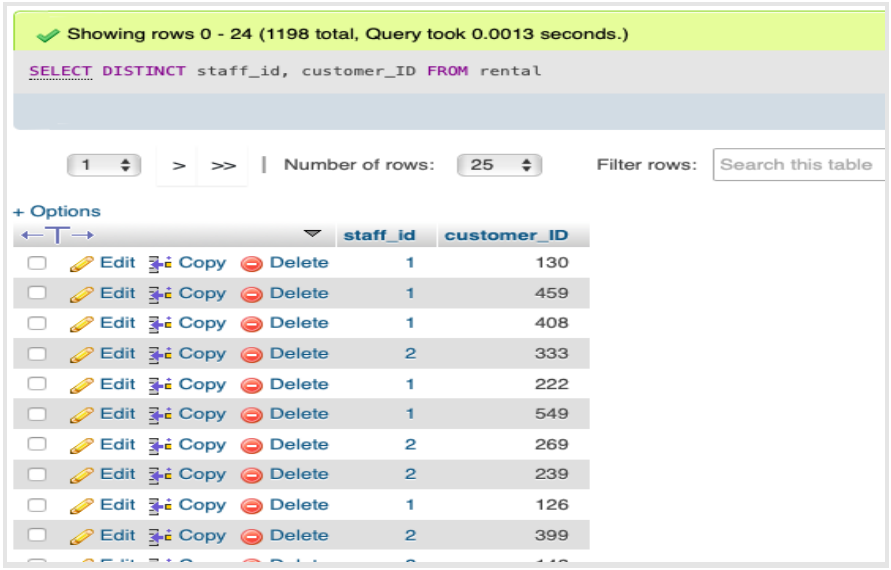



To see how many customers served

```
SELECT DISTINCT customer_ID FROM rental
```

To see how many customer served by each staff

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




**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
```

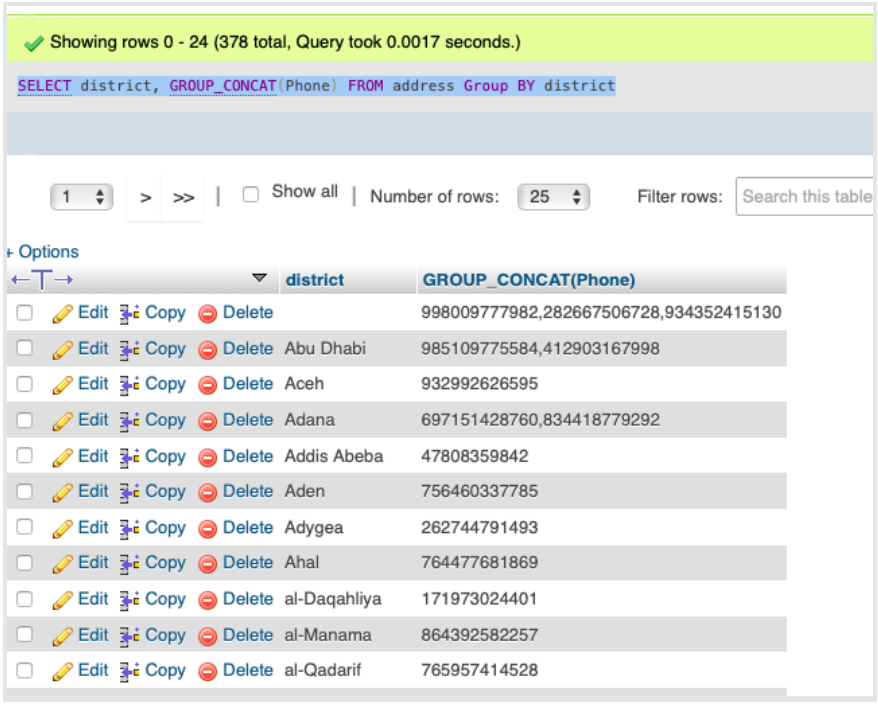
```
SELECT CONCAT(customer_ID, inventory_id) as conc, count(*) as ct FROM rental GROUP by conc Order by ct DESC
```

**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**


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
```


```
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
```




```
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 ';') |
|---|---|---|
| Edit Copy Delete | | 282667506728;934352415130;998009777982 |
| Edit Copy Delete | Abu Dhabi | 412903167998;985109775584 |
| Edit Copy Delete | Aceh | 932992626595 |
| Edit Copy Delete | Adana | 697151428760;834418779292 |
| Edit Copy Delete | Addis Abeba | 47808359842 |
| Edit Copy Delete | Aden | 756460337785 |
| Edit Copy Delete | Adygea | 262744791493 |
| Edit Copy Delete | Ahal | 764477681869 |
| Edit Copy Delete | al-Daqahliya | 171973024401 |
| Edit Copy 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: selct * 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 |
|---|---|---|---|---|---|---|---|
| Edit Copy Delete | 1 | 1 | 1 | 76 | 2.99 | 2005-05-25 11:30:37 | 2006-02-15 22:12:30 |
| Edit Copy 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 ] [ Creat

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

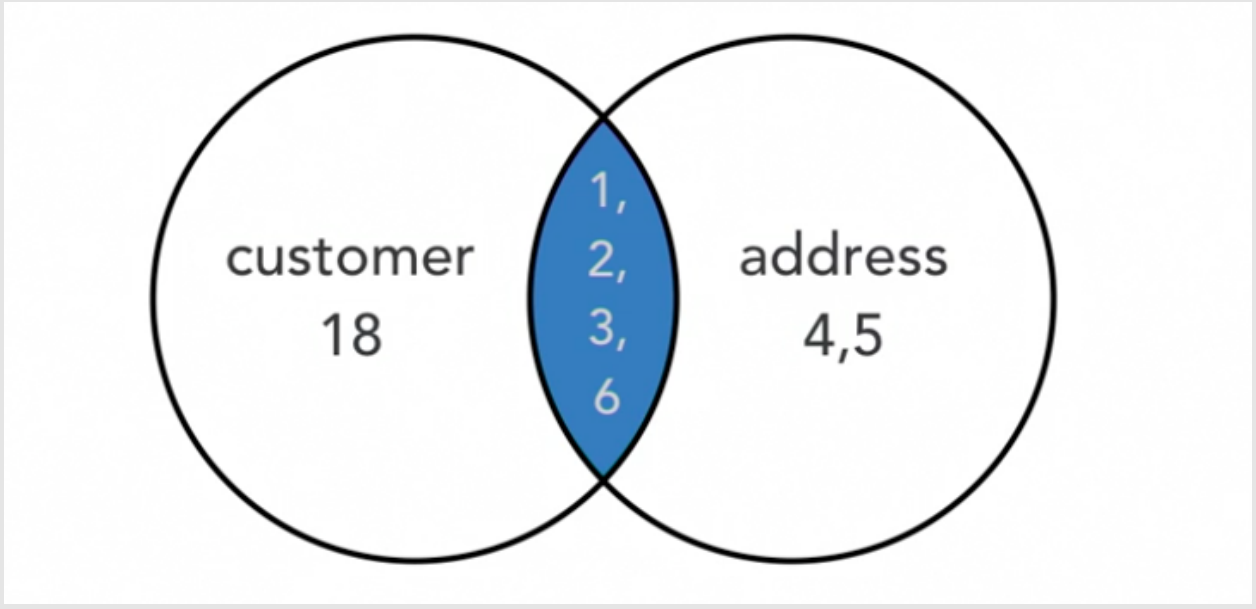
+ Options

| customer_id | store_id | first_name | last_name | email | address_id | active | create_date | last_update | addre |
|---|---|---|---|---|---|---|---|---|---|
| 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 ] [

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

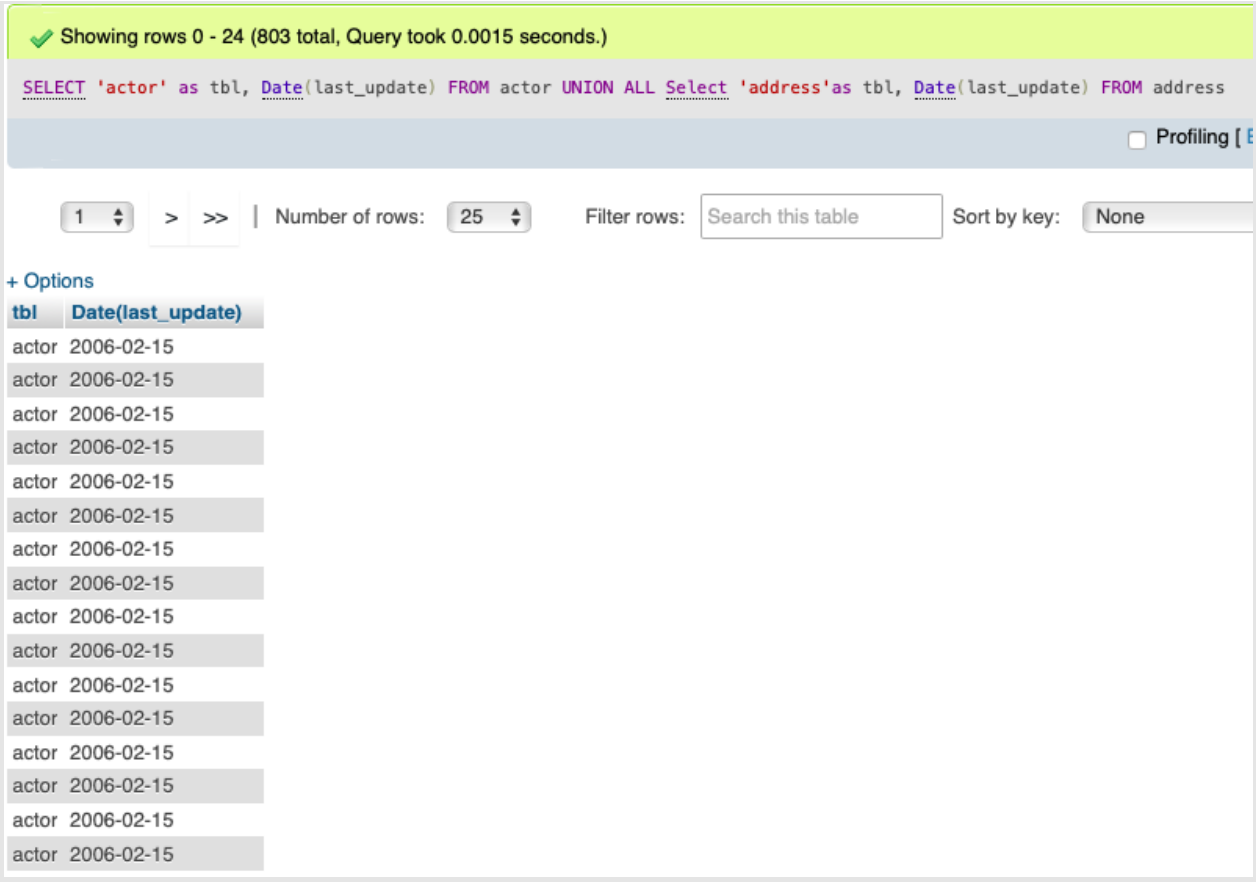
+ Options

| 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

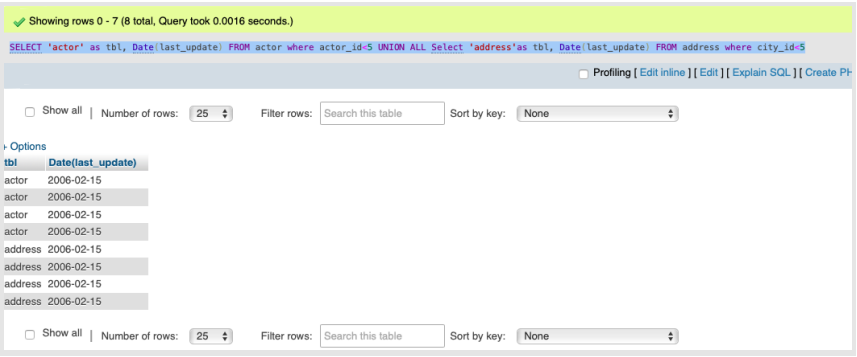
+ Options

| 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 |

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 PH

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 |
|---|---|---|---|---|---|---|---|
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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 |
| 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`