

Where Clause

To define some conditions

```
SELECT col1, col2 FROM table_name  
WHERE conditions;
```



```
18 • INSERT INTO user
19   (id, age, name, email, followers, following)
20   VALUES
21   (1, 14, "adam", "adam@yahoo.in", 123, 145),
22   (2, 15, "bob", "bob123@gmail.com", 200, 200),
23   (3, 16, "casey", "casey@email.com", 300, 306),
24   (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT *
27   FROM user
28   WHERE followers >= 200;
29
30
31
32
33
```

Result Grid



Filter Rows:



Search

Edit:



id	age	name	email	followers	following
2	15	bob	bob123@gmail.com	200	200
3	16	casey	casey@email.com	300	306
4	17	donald	donald@gmail.com	200	105
NULL	NULL	NULL	NULL	NULL	NULL

Query 1

Limit to 1000 rows

```
19 (id, age, name, email, followers, following)
20 VALUES
21 (1, 14, "adam", "adam@yahoo.in", 123, 145),
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT name, followers
27 FROM user
28 WHERE age < 16;
```

Where Clause

Operators

Arithmetic Operators : +(addition) , -(subtraction), *(multiplication), /(division), %(modulus)

Comparison Operators : = (equal to), != (not equal to), > , >=, <, <=

Logical Operators : AND, OR , NOT, IN, BETWEEN, ALL, LIKE, ANY

Bitwise Operators : & (Bitwise AND), | (Bitwise OR)



Where Clause

Frequently used Operators

AND (to check for both conditions to be true)

OR (to check for one of the conditions to be true)

BETWEEN (selects for a given range)

IN (matches any value in the list)

NOT (to negate the given condition)

VALUES

```
(1, 14, "adam", "adam@yahoo.in", 123, 145),  
(2, 15, "bob", "bob123@gmail.com", 200, 200),  
(3, 16, "casey", "casey@email.com", 300, 306),  
(4, 17, "donald", "donald@gmail.com", 200, 105);
```

SELECT name, age

FROM user

WHERE age > 15 AND followers > 200;

Query 1

Limit to 1000 rows

```
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT name, age, followers
27 FROM user
```

160% 36:28

Result Grid



Filter Rows:



Search

Export:



name	age	followers
casey	16	300

objects

Instagram

Tables

post

user

Views

Stored Procedures

Functions

Limit to 1000 rows

```
19 (id, age, name, email, followers, following)
20 VALUES
21 (1, 14, "adam", "adam@yahoo.in", 123, 145),
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT name, age, followers
27 FROM user
28 WHERE age > 15 OR followers > 200;
29
```

Result Grid



Filter Rows:



name



age

followers

casey

16

300

donald

17

200

```
2  (2, 15, "bob", "bob123@gmail.com", 200, 200),  
3  (3, 16, "casey", "casey@email.com", 300, 306),  
4  (4, 17, "donald", "donald@gmail.com", 200, 105);
```

```
5  
6 • SELECT name, age, followers  
7 FROM user  
8 WHERE age BETWEEN 15 AND 17;  
9
```

27

FROM user

160%



29:28

Result Grid



Filter Rows:



Search

name

age

followers

bob

15

200

casey

16

300

donald

17

200

```
(4, 17, "donald", "donald@gmail.com", 200, 105);
```

- ```
SELECT name, age, followers, | I
FROM user
WHERE email IN ("donald@gmail.com", "bob123@gmail.com", "abc@gmail.com");
```

```
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • SELECT name, followers, email
27 FROM user
```

160% 74:28

Result Grid



Filter Rows:



Search

Export:



| name   | followers | email            |
|--------|-----------|------------------|
| bob    | 200       | bob123@gmail.com |
| donald | 200       | donald@gmail.com |
|        |           |                  |
|        |           |                  |
|        |           |                  |
|        |           |                  |
|        |           |                  |



The image shows a screenshot of a SQL editor interface. At the top, there is a toolbar with various icons for file operations, execution, and search. A text box in the toolbar indicates "Limit to 1000 rows". Below the toolbar, a list of SQL queries is displayed, each preceded by a line number and a bullet point. The queries are as follows:

```
18 • INSERT INTO user
19 (id, age, name, email, followers, following)
20 VALUES
21 (1, 14, "adam", "adam@yahoo.in", 123, 145),
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • INSERT INTO user
27 (id, age, name, email, followers, following)
28 VALUES
29 (5, 14, "eve", "eve@yahoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
32 • SELECT name, followers, email
33 FROM user
34 WHERE age IN (14, 16);
35
36
```

```
28 VALUES
29 (5, 14, "eve", "eve@yahoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
32 • SELECT name, age, email
33 FROM user
```

160% 23:34

Result Grid



Filter Rows:

Search

Export:



| name  | age | email           |
|-------|-----|-----------------|
| adam  | 14  | adam@yahoo.in   |
| casey | 16  | casey@email.com |
| eve   | 14  | eve@yahoo.in    |
| farah | 16  | farah@gmail.com |



5  
6  
7  
8  
9  
0  
1  
2  
3  
4  
5

- **INSERT INTO user**  
(id, age, name, email, followers, following)  
**VALUES**  
(5, 14, "eve", "eve@yahoo.in", 400, 145),  
(6, 16, "farah", "farah@gmail.com", 10000, 1000);
- **SELECT name, age, email**  
**FROM user**  
**WHERE age NOT IN (14, 16);**

160%



27:34

Result Grid



Filter Rows:



Search

Export:



name

age

email

bob

15

bob123@gmail.com

donald

17

donald@gmail.com

# Limit Clause

Sets an upper limit on number of (tuples) rows to be returned

```
SELECT col1, col2 FROM table_name
LIMIT number;
```

```
28 VALUES
29 (5, 14, "eve", "eve@yahoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
32 • SELECT name, age, email
33 FROM user
34 WHERE age > 14
35 LIMIT 2;
36
37
```

160%



9:35

Result Grid



Filter Rows:



Search

Export:

name

age

email

bob

15

bob123@gmail.com

casey

16

casey@email.com

```
30 (6, 16, "farah", "farah@gmail.com", 1000
```

```
31
```

```
32 • SELECT name, age, email |
```

```
33 FROM user
```

```
34 LIMIT 2;
```

```
35
```

```
36
```

Result Grid



Filter Rows:



Search

Export:



Fetch

|  | name  | age | email            |  |
|--|-------|-----|------------------|--|
|  | adam  | 14  | adam@yahoo.in    |  |
|  | bob   | 15  | bob123@gmail.com |  |
|  | casey | 16  | casey@email.com  |  |
|  |       |     |                  |  |
|  |       |     |                  |  |

# Order by Clause

To sort in ascending (ASC) or descending order (DESC)

```
SELECT col1, col2 FROM table_name
ORDER BY col_name(s) ASC;
```



snehagupta7385@g

data

ASC

1 ✓

2

3

4 ✓

DESC

4 ✓

3

2

1 ✓

```
6 • INSERT INTO user
 (id, age, name, email, followers, following)
8 VALUES
9 (5, 14, "eve", "eve@yahoo.in", 400, 145),
0 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
1
2 • SELECT name, age, followers
3 FROM user
4 ORDER BY followers ASC;
```

Result Grid



Filter Rows:



Search

Export:



| name   | age | followers |
|--------|-----|-----------|
| adam   | 14  | 123       |
| bob    | 15  | 200       |
| donald | 17  | 200       |
| casey  | 16  | 300       |
| eve    | 14  | 400       |
| farah  | 16  | 10000     |

```
0 (6, 16, "Taran", "Taran@gmail.com", 10000,
1
2 • SELECT name, age, followers
3 FROM user
4 ORDER BY followers DESC; |
```

snehagupta7385@gm

Result Grid



Filter Rows:



Search

name

age

followers

farah

16

10000

eve

14

400

casey

16

300

bob

15

200

donald

17

200

adam

14

123

# Aggregate Functions

Aggregate functions perform a calculation on a set of values, and return a single value.

- COUNT()
- MAX()
- MIN()
- SUM()
- AVG()

Example :

```
SELECT max(marks)
FROM student;
```



```
29 (5, 14, "eve", "eve@yahoo.in", 40)
30 (6, 16, "farah", "farah@gmail.com", 50)
31
32 • SELECT max(followers)
33 FROM user;
34
35
```

# Result Grid

max(f...

10000



0 (6, 16, "farah", ')

1

2 • **SELECT** max(age)

3 **FROM** user;



1

2

3

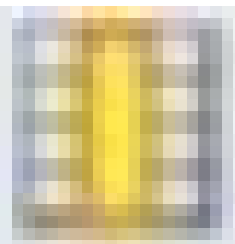
4

5

6

- `SELECT count(age)`  
`FROM user`  
`WHERE age = 14;`

# Result Grid



CO...

2



```
20 VALUES
21 (1, 14, "adam", "adam@yahoo.in", 123, 145),
22 (2, 15, "bob", "bob123@gmail.com", 200, 200),
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • INSERT INTO user
27 (id, age, name, email, followers, following)
28 VALUES
29 (5, 14, "eve", "eve@yahoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
32 • SELECT min(age)
33 FROM user;
34
```



```
SELECT avg(age)
FROM user;
```

avg(age)

15.3333

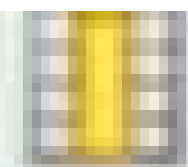




(5, 14, "eve", "eve@yahoo.in", 4  
(6, 16, "farah", "farah@gmail.co

- `SELECT sum(followers)`  
`FROM user;`

## Result Grid



|  | sum(f... |
|--|----------|
|  | 11223    |
|  |          |
|  |          |
|  |          |



# Group by Clause

Groups rows that have the same values into summary rows.

It collects data from multiple records and groups the result by one or more column.

```
SELECT col1, col2
FROM table_name
GROUP BY col_name(s);
```

snehagupta7385@gmail.com

\*Generally we use group by with some *aggregation function*.



Limit to 1000 rows

```
23 (3, 16, "casey", "casey@email.com", 300, 306),
24 (4, 17, "donald", "donald@gmail.com", 200, 105);
25
26 • INSERT INTO user
27 (id, age, name, email, followers, following)
28 VALUES
29 (5, 14, "eve", "eve@yahoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
32 • SELECT count(id)
33 FROM user
34 GROUP BY age;
35
36
37
38
39
40
41 • CREATE TABLE post (
42 id INT PRIMARY KEY,
```

| count(id) |
|-----------|
| 2         |
| 1         |
| 2         |
| 1         |

- ```
SELECT age, count(id)  
FROM user  
GROUP BY age;
```

age		count(id)
14		2
15		1
16		2
17		1

- ```
SELECT age
FROM user
GROUP BY age;
```



day

14

15

16

17



- ```
SELECT age, max(followers)
FROM user
GROUP BY age;
```

	age	max(f...
	14	400
	15	200
	16	10000
	17	200

Having Clause

Similar to Where i.e. applies some condition on rows.

But it is used when we want to apply any **condition after grouping**.

where
(condition)

```
SELECT col1, col2  
FROM table_name  
GROUP BY col_name(s)  
HAVING condition;
```

- WHERE is for the table, HAVING is for a group
- Grouping is necessary for HAVING



Having Clause

Similar to Where i.e. applies some condition on rows.

But it is used when we want to apply any **condition after grouping**.

where
(condition)

```
SELECT col1, col2  
FROM table_name  
GROUP BY col_name(s)  
HAVING condition;
```

} select age, max(followers)
From user
Group By age

- WHERE is for the table, HAVING is for a group
- Grouping is necessary for HAVING



(0, 10, 'Ratan', 'Ratan@gmail.com', 100)

- ```
SELECT age, max(followers)
FROM user
GROUP BY age
HAVING max(followers) > 200;
```

age

max(f...

14

400

16

10000

# General Order

**SELECT** *column(s)*

**FROM** *table\_name*

**WHERE** *condition*

**GROUP BY** *column(s)*

**HAVING** *condition*

**ORDER BY** *column(s)* **ASC;**



- ```
SELECT age, max(followers)
FROM user
GROUP BY age
HAVING max(followers) > 200
ORDER BY age DESC;
```

	age	max(followers)
	16	10000
	14	400

Table Queries

Update (to update existing rows)

UPDATE *table_name*

SET *col1 = val1, col2 = val2*

WHERE *condition;*

```

5
6 • INSERT INTO user
7   (id, age, name, email, followers, following)
8   VALUES
9   (5, 14, "eve", "eve@yahoo.in", 400, 145),
10  (6, 16, "farah", "farah@gmail.com", 10000, 1000);
11
12 • UPDATE user
13   SET followers = 600
14   WHERE age = 16;

```



snehagupta7385@gm

	Time	Action	Response	Duration
62	13:29:28	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 LIMIT 0, 1000	2 row(s) returned	0.0
63	13:55:14	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 LIMIT 0, 1000	2 row(s) returned	0.0
64	13:55:37	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 ORDER BY age DESC LIMIT 0, 1...	2 row(s) returned	0.0
65	13:58:09	UPDATE user SET followers = 600 WHERE age = 16	Error Code: 1175. You are using safe update mode an...	0.0

- **UPDATE user**

SET followers = 600

WHERE age = 16;

- **SET SQL_SAFE_UPDATES = 0;**

```

30 (6, 16, "Taran", "Taran@gmail.com", 10000, 1000);
31
32 • UPDATE user
33 SET followers = 600
34 WHERE age = 16;
35
36 SET D
37
38
39
40
41

```

Object Info Session
object selected

Action Output

	Time	Action	Response	Duration
62	13:29:28	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 LIMIT 0, 1000	2 row(s) returned	0.0004
63	13:55:14	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 LIMIT 0, 1000	2 row(s) returned	0.0006
64	13:55:37	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 ORDER BY age DESC LIMIT 0, 1...	2 row(s) returned	0.0007
65	13:58:09	UPDATE user SET followers = 600 WHERE age = 16	Error Code: 1175. You are using safe update mode an...	0.0014

```
29 (5, 14, "eve", "eve@yanoo.in", 400, 145),
30 (6, 16, "farah", "farah@gmail.com", 10000, 1000);
31
```

```
32 • UPDATE user
33 SET followers = 600
34 WHERE age = 16;
```

```
36 • SET SQL_SAFE_UPDATES = 0;
```

```
37
38
39
40
41
```

160% 1:32

Action Output

	Time	Action	Response	Duration / Fetch Ti
64	13:55:37	SELECT age, max(followers) FROM user GROUP BY age HAVING max(followers) > 200 ORDER BY age DESC LIMIT 0, 1...	2 row(s) returned	0.00071 sec / 0.000...
65	13:58:09	UPDATE user SET followers = 600 WHERE age = 16	Error Code: 1175. You are using safe update mode an...	0.0014 sec
66	13:59:06	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00031 sec
67	13:59:11	UPDATE user SET followers = 600 WHERE age = 16	2 row(s) affected Rows matched: 2 Changed: 2 Warn...	0.0018 sec

35

36 • **SELECT * FROM user;**

37

160% 20:36

Result Grid



Filter Rows:



Search

Edit:



Export/Import:



id	age	name	email	followers	following
1	14	adam	adam@yahoo.in	123	145
2	15	bob	bob123@gmail.com	200	200
3	16	casey	casey@email.com	600	306
4	17	donald	donald@gmail.com	200	105
5	14	eve	eve@yahoo.in	400	145
6	16	farah	farah@gmail.com	600	1000
NULL	NULL	NULL	NULL	NULL	NULL

Table Queries

Delete (to delete existing rows)



DELETE FROM *table_name*

WHERE *condition;*

- **INSERT INTO user**
(id, age, name, email, followers)
VALUES
(5, 14, "eve", "eve@yahoo.in", 4)
(6, 16, "farah", "farah@gmail.co
I
- **DELETE FROM user**
WHERE age = 14;
- **SELECT * FROM user;**
- **SET SQL_SAFE_UPDATES = 0;**

Result Grid



Filter Rows:



Search

Edit:



E

	id	age	name	email	followers	following	
	2	15	bob	bob123@gmail.com	200	200	
	3	16	casey	casey@email.com	600	306	
	4	17	donald	donald@gmail.com	200	105	
	6	16	farah	farah@gmail.com	600	1000	
	NULL	NULL	NULL	NULL	NULL	NULL	

Table Queries

Alter (to change the schema)

columns

name
datatype
constraints
add
delete

ADD Column

ALTER TABLE *table_name*

ADD COLUMN *column_name datatype constraint;*

ehagupta7385@gmail.com

DROP Column

ALTER TABLE *table_name*

DROP COLUMN *column_name;*

RENAME Table

ALTER TABLE *table_name*

RENAME TO *new_table_name;*

```
ALTER TABLE user  
ADD COLUMN city VARCHAR(25) DEFAULT "Delhi";
```

33 **ADD COLUMN** city **VARCHAR(25)** **DEFAULT** "Delhi"

34

35 • **SELECT * FROM user;**

36

160%



20:35

Result Grid



Filter Rows:



Search

Edit:



Export/Import:



id	age	name	email	followers	following	city
2	15	bob	bob123@gmail.com	200	200	Delhi
3	16	casey	casey@email.com	600	306	Delhi
4	17	donald	donald@gmail.com	200	105	Delhi
6	16	farah	farah@gmail.com	600	1000	Delhi
NULL	NULL	NULL	NULL	NULL	NULL	NULL

385@gmail.com

```
ALTER TABLE user  
DROP COLUMN age;
```

37

38 • **SELECT * FROM user;**

39

160% 20:38

Result Grid



Filter Rows:



Search

Edit:



Export/Import

id	name	email	followers	following	city	
2	bob	bob123@gmail.com	200	200	Delni	
3	casey	casey@email.com	600	306	Delhi	
4	donald	donald@gmail.com	200	105	Delhi	
6	farah	farah@gmail.com	600	1000	Delhi	
NULL	NULL	NULL	NULL	NULL	NULL	


```
DROP COLUMN age;
```

- ```
ALTER TABLE user
RENAME TO instaUser;
```
- ```
SELECT * FROM user;
```

40

41 • **SELECT * FROM instaUser;**

42

160%



25:41

Result Grid



Filter Rows:



Search

Edit:



Export/Import

id	name	email	followers	following	city	
2	bob	bob123@gmail.com	200	200	Delhi	
3	casey	casey@email.com	600	306	Delhi	
4	donald	donald@gmail.com	200	105	Delhi	
6	farah	farah@gmail.com	600	1000	Delhi	
NULL	NULL	NULL	NULL	NULL	NULL	

Table Queries

Alter (to change the schema)

CHANGE Column (rename)

ALTER TABLE *table_name*

CHANGE COLUMN *old_name new_name new_datatype new_constraint;*

MODIFY Column (modify datatype/ constraint)

ALTER TABLE *table_name*

MODIFY *col_name new_datatype new_constraint;*



7
3 • `ALTER TABLE instaUser`
9 `RENAME TO user;`

1 • `ALTER TABLE user`
2 `CHANGE COLUMN followers subs INT DEFAULT 0;`
3

44 • **SELECT * FROM user;**

160% 20:44

Result Grid



Filter Rows:



Search

Edit:



Export/Import:



id	name	email	subs	following	city
2	bob	bob123@gmail.com	200	200	Delhi
3	casey	casey@email.com	600	306	Delhi
4	donald	donald@gmail.com	200	105	Delhi
6	farah	farah@gmail.com	600	1000	Delhi
NULL	NULL	NULL	NULL	NULL	NULL



```
ALTER TABLE user  
MODIFY subs INT DEFAULT 5;
```

25

26 • **INSERT INTO user**

27 (id, age, **name**, email, **following**)

28 **VALUES**

29 (6, 19, "gemini", "gem@yahoo.in", 145);

30

31 **ALTER TABLE user**

45

46 • **SELECT * FROM user;**

47

160% 20:46

Result Grid



Filter Rows:



Search

Edit:



Export/Import:



id	name	email	subs	following	city	
2	bob	bob123@gmail.com	200	200	Delhi	
3	casey	casey@email.com	600	306	Delhi	
4	donald	donald@gmail.com	200	105	Delhi	
6	farah	farah@gmail.com	600	1000	Delhi	
7	gemini	gem@yahoo.in	5	120	Delhi	
NULL	NULL	NULL	NULL	NULL	NULL	

Table Queries

Truncate (to delete table's data)

TRUNCATE TABLE *table_name* ;

- `TRUNCATE TABLE user;`

Action Output

	Time	Action	Response	Duration / Fetch Time
83	14:09:40	INSERT INTO user (id, name, email, following) VALUES (6, "gemini", "gem@yahoo.in", 120)	Error Code: 1062. Duplicate entry '6' for key 'user.pri...	0.0000 / 1 sec
84	14:09:53	INSERT INTO user (id, name, email, following) VALUES (7, "gemini", "gem@yahoo.in", 120)	1 row(s) affected	0.0016 sec
85	14:09:58	SELECT * FROM user LIMIT 0, 1000	5 row(s) returned	0.00072 sec / 0.0000...
86	14:11:14	TRUNCATE TABLE user	Error Code: 1701. Cannot truncate a table referenced i...	0.0012 sec

52

53

54 • **DROP TABLE** post;

55

56 • **CREATE TABLE** post (

--

```
TRUNCATE TABLE user;
```

45

46 • **SELECT * FROM user;**

47

160% 20:46

Result Grid



Filter Rows:



Search

Edit:



Export/Import:



id	name	email	subs	following	city
----	------	-------	------	-----------	------

NULL

NULL

NULL

NULL

NULL

NULL

47

48 • **DROP TABLE user;**

49

```
SELECT * FROM user;
```


90	14:13:35	DROP TABLE user	0 row(s) affected	0.0044 sec
91	14:13:40	SELECT * FROM user LIMIT 0, 1000	Error Code: 1146. Table 'instagram.user' doesn't exist	0.00094 sec