Postgres Servr Status

**root@my:~# service postgresql status**

● postgresql.service - PostgreSQL RDBMS

Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)

Active: active (exited) since Thu 2020-04-30 11:08:34 IST; 2s ago

Process: 2702 ExecStart=/bin/true (code=exited, status=0/SUCCESS)

Main PID: 2702 (code=exited, status=0/SUCCESS)

Apr 30 11:08:34 my systemd[1]: Starting PostgreSQL RDBMS...

Apr 30 11:08:34 my systemd[1]: Finished PostgreSQL RDBMS.

Login Postgres user

**root@my:~# sudo su – postgresql**

[postgres@my](mailto:postgres@my):~$

psostgersql Shell

**postgres@my:~$ psql**

psql (12.2 (Debian 12.2-4), server 11.5 (Debian 11.5-1))

Type "help" for help.

List the database

**postgres=# \l**

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+----------+----------+---------+---------+-----------------------

msf | msf | UTF8 | C.UTF-8 | C.UTF-8 |

msf\_test | msf | UTF8 | C.UTF-8 | C.UTF-8 |

postgres | postgres | UTF8 | C.UTF-8 | C.UTF-8 |

template0 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

testdb1 | postgres | UTF8 | C.UTF-8 | C.UTF-8 |

(6 rows)

selecting Database

**postgres=# \c testdb1**

psql (12.2 (Debian 12.2-4), server 11.5 (Debian 11.5-1))

You are now connected to database "testdb1" as user "postgres".

testdb1=#

**testdb1=# \c postgres**

psql (12.2 (Debian 12.2-4), server 11.5 (Debian 11.5-1))

You are now connected to database "postgres" as user "postgres".

postgres=#

Set the search path

**postgres=# SET search\_path TO information;**

SET

List the Tables

**postgres=# \d**

List of relations

Schema | Name | Type | Owner

-------------+----------+-------+----------

information | Teachers | table | postgres

information | students | table | postgres

information | test | table | postgres

(3 rows)

List the Schema

**postgres=# \dn**

List of schemas

Name | Owner

-------------+----------

information | postgres

public | postgres

samplee | postgres

(3 rows)

postgres=#

List the User

**postgres=# \du**

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

msf | | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

user\_1 | Superuser | {}

Alter the User paermission

**postgres=# ALTER USER user\_1 WITH CreateDB;**

ALTER ROLE

**postgres=# \du**

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

msf | | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

user\_1 | Superuser, Create DB | {}

**postgres=# ALTER USER user\_1 WITH Createrole;**

ALTER ROLE

**postgres=# \du**

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

msf | | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

user\_1 | Superuser, Create role, Create DB | {}

postgres=#

Create user with password

**postgres=# CREATE USER User\_2 WITH PASSWORD '12345678' SUPERUSER;**

CREATE ROLE

**postgres=# \du**

List of roles

Role name | Attributes | Member of

-----------+------------------------------------------------------------+-----------

msf | | {}

postgres | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

user\_1 | Superuser, Create role, Create DB | {}

user\_2 | Superuser | {}

postgres=#

Change user Password

**postgres=# ALTER USER user\_1 PASSWORD 'test123';**

ALTER ROLE

Switching database connection

**postgres=# \connect testdb1**

psql (12.2 (Debian 12.2-4), server 11.5 (Debian 11.5-1))

You are now connected to database "testdb1" as user "postgres".

testdb1=#

Inserting data into Table

**postgres=# INSERT INTO test (no,name,age) VALUES (1,'ram',25);**

INSERT 0 1

Select data from table

**postgres=# SELECT \* FROM test;**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

**postgres=# SELECT \* from TEST;**

no | name | age

----+-------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

5 | arun | 25

6 | anbu | 24

7 | bala | 34

10 | deva | 28

14 | mani | 41

8 | kavi | 21

11 | roja | 21

9 | rani | 23

12 | guna | 26

17 | guru | 29

13 | mathi | 43

(16 rows)

**Remove duplicat value**

**postgres=# SELECT DISTINCT name FROM test;**

name

-------

mathi

deva

arun

guna

raja

anbu

rani

bala

raj

kavi

ram

guru

roja

ravi

mani

(15 rows)

Display data in order by

**postgres=# SELECT \* FROM test ORDER BY name;**

no | name | age

----+-------+-----

6 | anbu | 24

5 | arun | 25

7 | bala | 34

10 | deva | 28

12 | guna | 26

17 | guru | 29

8 | kavi | 21

14 | mani | 41

13 | mathi | 43

2 | raj | 35

2 | raja | 35

3 | raja | 25

1 | ram | 25

9 | rani | 23

4 | ravi | 25

11 | roja | 21

(16 rows)

**postgres=# SELECT DISTINCT ON (name) name, age FROM test ORDER BY name;**

name | age

-------+-----

anbu | 24

arun | 25

bala | 34

deva | 28

guna | 26

guru | 29

kavi | 21

mani | 41

mathi | 43

raj | 35

raja | 35

ram | 25

rani | 23

ravi | 25

roja | 21

(15 rows)

Display particular row value in table

**postgres=# SELECT name,no FROM test WHERE name = 'raja';**

name | no

------+----

raja | 2

raja | 3

(2 rows)

**postgres=# SELECT name,no,age FROM test WHERE name IN ('raj','raja','ram');**

name | no | age

------+----+-----

ram | 1 | 25

raj | 2 | 35

raja | 2 | 35

raja | 3 | 25

(4 rows)

**postgres=# SELECT name,no,age FROM test WHERE name = 'raja' AND no=3;**

name | no | age

------+----+-----

raja | 3 | 25

(1 row)

**postgres=# SELECT name,no,age FROM test WHERE name = 'raja' OR no=3;**

name | no | age

------+----+-----

raja | 2 | 35

raja | 3 | 25

(2 rows)

**postgres=# SELECT \* FROM test WHERE name != 'raja';**

no | name | age

----+-------+-----

1 | ram | 25

2 | raj | 35

4 | ravi | 25

5 | arun | 25

6 | anbu | 24

7 | bala | 34

10 | deva | 28

14 | mani | 41

8 | kavi | 21

11 | roja | 21

9 | rani | 23

12 | guna | 26

17 | guru | 29

13 | mathi | 43

(14 rows)

**postgres=# SELECT \* FROM test WHERE no > 5;**

no | name | age

----+-------+-----

6 | anbu | 24

7 | bala | 34

10 | deva | 28

14 | mani | 41

8 | kavi | 21

11 | roja | 21

9 | rani | 23

12 | guna | 26

17 | guru | 29

13 | mathi | 43

(10 rows)

**postgres=# SELECT \* FROM test WHERE no <= 5;**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

5 | arun | 25

(6 rows)

**postgres=# SELECT \* FROM test WHERE name LIKE 'ra%';**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

9 | rani | 23

(6 rows)

Limit

**postgres=# SELECT \* FROM test LIMIT 5;**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

(5 rows)

**postgres=# SELECT \* FROM test ORDER BY age LIMIT 5;**

no | name | age

----+------+-----

8 | kavi | 21

11 | roja | 21

9 | rani | 23

6 | anbu | 24

5 | arun | 25

(5 rows)

**postgres=# SELECT \* FROM test ORDER BY age LIMIT 5 OFFSET 3;**

no | name | age

----+------+-----

6 | anbu | 24

1 | ram | 25

3 | raja | 25

4 | ravi | 25

5 | arun | 25

(5 rows)

Fetching First row only

**postgres=# SELECT \* FROM test ORDER BY age FETCH FIRST ROW ONLY;**

no | name | age

----+------+-----

8 | kavi | 21

(1 row)

Fetching First 5 row only

**postgres=# SELECT \* FROM test ORDER BY age FETCH FIRST 5 ROW ONLY;**

no | name | age

----+------+-----

8 | kavi | 21

11 | roja | 21

9 | rani | 23

6 | anbu | 24

5 | arun | 25

(5 rows)

Getting perticular row (IN)

**postgres=# SELECT \* FROM test WHERE age IN (21,41,25) ORDER BY no ASC;**

no | name | age

----+------+-----

1 | ram | 25

3 | raja | 25

4 | ravi | 25

5 | arun | 25

8 | kavi | 21

11 | roja | 21

14 | mani | 41

(7 rows)

Getting perticular row ( NOT IN)

**postgres=# SELECT \* FROM test WHERE age NOT IN (21,41,25) ORDER BY no ASC;**

no | name | age

----+-------+-----

2 | raja | 35

2 | raj | 35

6 | anbu | 24

7 | bala | 34

9 | rani | 23

10 | deva | 28

12 | guna | 26

13 | mathi | 43

17 | guru | 29

(9 rows)

# Between

**postgres=# SELECT \* FROM test WHERE age BETWEEN 25 AND 35;**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

5 | arun | 25

7 | bala | 34

10 | deva | 28

12 | guna | 26

17 | guru | 29

**postgres=# SELECT \* FROM test WHERE name LIKE 'r%' AND LENGTH(name) BETWEEN 3 AND 5 ORDER BY name;**

no | name | age

----+------+-----

2 | raj | 35

2 | raja | 35

3 | raja | 25

1 | ram | 25

9 | rani | 23

4 | ravi | 25

11 | roja | 21

(7 rows)

**postgres=# SELECT \* FROM test WHERE name LIKE 'r%' AND LENGTH(name) BETWEEN 1 AND 3 ORDER BY name;**

no | name | age

----+------+-----

2 | raj | 35

1 | ram | 25

(2 rows)

postgres=#

Null and IS NULL

**postgres=# SELECT \* FROM contact WHERE phone IS NULL ORDER BY id;**

id | name | address | phone | email

----+---------+------------+-------+----------------------

2 | RAM | SALEM | | test1@gmail.com

9 | JEEVA | MADURI | | manostogle@gmail.com

12 | PRAVEEN | METTUR | | gopraveen@gmial.com

13 | VINOTH | SALEM | |

14 | TAMIL | DHARMAPURI | | tamildago@gmial.com

(5 rows)

postgres=#

Alias Name (column alias)

**postgres=# SELECT id || ' ' || name FROM contact ORDER BY id;**

?column?

-------------

1 RAMESH

2 RAM

3 ARUN

4 SAKTHI

5 KAVI

6 BABU

7 MANO

8 GOKUL

9 JEEVA

10 GUNA

11 NAVEEN

12 PRAVEEN

13 VINOTH

14 TAMIL

15 DEVA

(15 rows)

**postgres=# SELECT id || ' ' || name AS name\_list FROM contact ORDER BY id;**

name\_list

-------------

1 RAMESH

2 RAM

3 ARUN

4 SAKTHI

5 KAVI

6 BABU

7 MANO

8 GOKUL

9 JEEVA

10 GUNA

11 NAVEEN

12 PRAVEEN

13 VINOTH

14 TAMIL

15 DEVA

(15 rows)

Alias Name (table alias)

**postgres=# SELECT \* FROM contact AS table\_1;**

id | name | address | phone | email

----+---------+------------+---------------+------------------------

1 | RAMESH | SALEM | +917485963625 | test@gmail.com

2 | RAM | SALEM | | test1@gmail.com

3 | ARUN | CHENNAI | +918569741236 | testgoogle@gmail.com

6 | BABU | MADURI | +916365784595 | babugoogle@gmail.com

15 | DEVA | CHENNAI | +919327845965 | devatestogle@gmail.com

10 | GUNA | NAMAKKAL | +917845278455 | gunastogle@gmail.com

8 | GOKUL | KOVAI | +918694527845 | gunastogle@gmail.com

7 | MANO | METTUR | +919544527845 | manostogle@gmail.com

9 | JEEVA | MADURI | | manostogle@gmail.com

5 | KAVI | MADURI | +918759684798 |

13 | VINOTH | SALEM | |

4 | SAKTHI | DHARMAPURI | +917510032364 |

11 | NAVEEN | SALEM | +917783632364 |

14 | TAMIL | DHARMAPURI | | tamildago@gmial.com

12 | PRAVEEN | METTUR | | gopraveen@gmial.com

(15 rows)

Join (inner join)

**SELECT test.no, test.name,test\_1.no, test\_1.name FROM test INNER JOIN test\_1 ON test.name = test\_1.name;**

no | name | no | name

----+-------+----+-------

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

(2 rows)

**postgres=# SELECT \* FROM test INNER JOIN test\_1 on test.name = test\_1.name;**

no | name | age | no | name | city | age

----+-------+-----+----+-------+---------+-----

13 | mathi | 43 | 1 | mathi | salem | 43

4 | ravi | 25 | 2 | ravi | chennai | 24

(2 rows)

**SELECT test.no,test.name,test\_1.no,test\_1.name FROM test INNER JOIN test\_1 on test.name = test\_1.name ORDER BY test.name;**

no | name | no | name

----+-------+----+-------

6 | anbu | 5 | anbu

14 | mani | 7 | mani

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

(4 rows)

**SELECT test.no,test.name,test\_1.no,test\_1.name, contact.id, contact.name FROM test INNER JOIN test\_1 on test.name = test\_1.name INNER JOIN contact ON test.name = test\_1.name;**

no | name | no | name | id | name

----+-------+----+-------+----+---------

13 | mathi | 1 | mathi | 1 | RAMESH

13 | mathi | 1 | mathi | 2 | RAM

13 | mathi | 1 | mathi | 3 | ARUN

13 | mathi | 1 | mathi | 6 | BABU

13 | mathi | 1 | mathi | 15 | DEVA

13 | mathi | 1 | mathi | 10 | GUNA

13 | mathi | 1 | mathi | 8 | GOKUL

13 | mathi | 1 | mathi | 7 | MANO

13 | mathi | 1 | mathi | 9 | JEEVA

13 | mathi | 1 | mathi | 5 | KAVI

13 | mathi | 1 | mathi | 13 | VINOTH

13 | mathi | 1 | mathi | 4 | SAKTHI

13 | mathi | 1 | mathi | 11 | NAVEEN

13 | mathi | 1 | mathi | 14 | TAMIL

13 | mathi | 1 | mathi | 12 | PRAVEEN

4 | ravi | 2 | ravi | 1 | RAMESH

4 | ravi | 2 | ravi | 2 | RAM

4 | ravi | 2 | ravi | 3 | ARUN

4 | ravi | 2 | ravi | 6 | BABU

4 | ravi | 2 | ravi | 15 | DEVA

4 | ravi | 2 | ravi | 10 | GUNA

4 | ravi | 2 | ravi | 8 | GOKUL

4 | ravi | 2 | ravi | 7 | MANO

4 | ravi | 2 | ravi | 9 | JEEVA

4 | ravi | 2 | ravi | 5 | KAVI

4 | ravi | 2 | ravi | 13 | VINOTH

4 | ravi | 2 | ravi | 4 | SAKTHI

4 | ravi | 2 | ravi | 11 | NAVEEN

4 | ravi | 2 | ravi | 14 | TAMIL

4 | ravi | 2 | ravi | 12 | PRAVEEN

6 | anbu | 5 | anbu | 1 | RAMESH

6 | anbu | 5 | anbu | 2 | RAM

6 | anbu | 5 | anbu | 3 | ARUN

6 | anbu | 5 | anbu | 6 | BABU

6 | anbu | 5 | anbu | 15 | DEVA

6 | anbu | 5 | anbu | 10 | GUNA

6 | anbu | 5 | anbu | 8 | GOKUL

6 | anbu | 5 | anbu | 7 | MANO

6 | anbu | 5 | anbu | 9 | JEEVA

6 | anbu | 5 | anbu | 5 | KAVI

6 | anbu | 5 | anbu | 13 | VINOTH

6 | anbu | 5 | anbu | 4 | SAKTHI

6 | anbu | 5 | anbu | 11 | NAVEEN

6 | anbu | 5 | anbu | 14 | TAMIL

6 | anbu | 5 | anbu | 12 | PRAVEEN

14 | mani | 7 | mani | 1 | RAMESH

14 | mani | 7 | mani | 2 | RAM

14 | mani | 7 | mani | 3 | ARUN

14 | mani | 7 | mani | 6 | BABU

14 | mani | 7 | mani | 15 | DEVA

:

Join (Left join)

**SELECT test.no,test.name, test\_1.no, test\_1.name FROM test LEFT JOIN test\_1 ON test.name = test\_1.name;**

no | name | no | name

----+-------+----+-------

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

11 | roja | |

1 | ram | |

3 | raja | |

2 | raja | |

17 | guru | |

9 | rani | |

2 | raj | |

6 | anbu | |

7 | bala | |

5 | arun | |

8 | kavi | |

10 | deva | |

12 | guna | |

14 | mani | |

(16 rows)

**postgres=# SELECT \* FROM test LEFT JOIN test\_1 on test.name = test\_1.name;**

no | name | age | no | name | city | age

----+-------+-----+----+-------+---------+-----

13 | mathi | 43 | 1 | mathi | salem | 43

4 | ravi | 25 | 2 | ravi | chennai | 24

11 | roja | 21 | | | |

1 | ram | 25 | | | |

3 | raja | 25 | | | |

2 | raja | 35 | | | |

17 | guru | 29 | | | |

9 | rani | 23 | | | |

2 | raj | 35 | | | |

6 | anbu | 24 | | | |

7 | bala | 34 | | | |

5 | arun | 25 | | | |

8 | kavi | 21 | | | |

10 | deva | 28 | | | |

12 | guna | 26 | | | |

14 | mani | 41 | | | |

(16 rows)

**SELECT test.no,test.name,test\_1.no,test\_1.name FROM test LEFT JOIN test\_1 on test.name = test\_1.name ORDER BY test.no;**

no | name | no | name

----+-------+----+-------

1 | ram | |

2 | raj | |

2 | raja | |

3 | raja | |

4 | ravi | 2 | ravi

5 | arun | |

6 | anbu | 5 | anbu

7 | bala | |

8 | kavi | |

9 | rani | |

10 | deva | |

11 | roja | |

12 | guna | |

13 | mathi | 1 | mathi

14 | mani | 7 | mani

17 | guru | |

(16 rows)

Join (Right join)

**postgres=# SELECT test.no,test.name,test\_1.no,test\_1.name FROM test RIGHT JOIN test\_1 on test.name = test\_1.name;**

no | name | no | name

----+-------+----+--------

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

6 | anbu | 5 | anbu

14 | mani | 7 | mani

| | 3 | dinesh

| | 4 | mohan

| | 8 | kannan

| | 6 | kumar

(8 rows)

**postgres=# SELECT \* FROM test RIGHT JOIN test\_1 on test.name = test\_1.name;**

no | name | age | no | name | city | age

----+-------+-----+----+--------+------------+-----

13 | mathi | 43 | 1 | mathi | salem | 43

4 | ravi | 25 | 2 | ravi | chennai | 24

6 | anbu | 24 | 5 | anbu | madurai | 24

14 | mani | 41 | 7 | mani | mettur | 35

| | | 3 | dinesh | mecheri | 27

| | | 4 | mohan | salem | 28

| | | 8 | kannan | chennai | 39

| | | 6 | kumar | dharmapuri | 32

(8 rows)

**postgres=# SELECT test.no,test.name,test\_1.no,test\_1.name FROM test RIGHT JOIN test\_1 on test.name = test\_1.name ORDER BY test.no;**

no | name | no | name

----+-------+----+--------

4 | ravi | 2 | ravi

6 | anbu | 5 | anbu

13 | mathi | 1 | mathi

14 | mani | 7 | mani

| | 6 | kumar

| | 3 | dinesh

| | 4 | mohan

| | 8 | kannan

(8 rows)

join (Full outer)

**postgres=# SELECT test.no,test.name,test\_1.no,test\_1.name FROM test FULL OUTER JOIN test\_1 on test.name = test\_1.name;**

no | name | no | name

----+-------+----+--------

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

6 | anbu | 5 | anbu

14 | mani | 7 | mani

| | 3 | dinesh

| | 4 | mohan

| | 8 | kannan

| | 6 | kumar

11 | roja | |

1 | ram | |

3 | raja | |

2 | raja | |

17 | guru | |

9 | rani | |

2 | raj | |

7 | bala | |

5 | arun | |

8 | kavi | |

10 | deva | |

12 | guna | |

(20 rows)

**postgres=# SELECT test.no,test.name,test\_1.no,test\_1.name FROM test FULL OUTER JOIN test\_1 on test.name = test\_1.name ORDER BY test.no;**

no | name | no | name

----+-------+----+--------

1 | ram | |

2 | raja | |

2 | raj | |

3 | raja | |

4 | ravi | 2 | ravi

5 | arun | |

6 | anbu | 5 | anbu

7 | bala | |

8 | kavi | |

9 | rani | |

10 | deva | |

11 | roja | |

12 | guna | |

13 | mathi | 1 | mathi

14 | mani | 7 | mani

17 | guru | |

| | 3 | dinesh

| | 4 | mohan

| | 6 | kumar

| | 8 | kannan

(20 rows)

join(Full join)

**postgres=# SELECT test.no,test.name,test\_1.no,test\_1.name FROM test FULL JOIN test\_1 on test.name = test\_1.name;**

no | name | no | name

----+-------+----+--------

13 | mathi | 1 | mathi

4 | ravi | 2 | ravi

6 | anbu | 5 | anbu

14 | mani | 7 | mani

| | 3 | dinesh

| | 4 | mohan

| | 8 | kannan

| | 6 | kumar

11 | roja | |

1 | ram | |

3 | raja | |

2 | raja | |

17 | guru | |

9 | rani | |

2 | raj | |

7 | bala | |

5 | arun | |

8 | kavi | |

10 | deva | |

12 | guna | |

(20 rows)

**Natural join**

**postgres=# SELECT \* FROM products;**

product\_id | product\_name | category\_id

------------+-----------------+-------------

1 | iPhone | 1

2 | Samsung Galaxy | 1

3 | HP Elite | 2

4 | Lenovo Thinkpad | 2

5 | iPad | 3

6 | Kindle Fire | 3

(6 rows)

**postgres=# SELECT \* FROM categories;**

category\_id | category\_name

-------------+---------------

1 | Smart Phone

2 | Laptop

3 | Tablet

4 | Smart Phone

5 | Laptop

6 | Tablet

(6 rows)

**postgres=# SELECT \* FROM products NATURAL JOIN categories;**

category\_id | product\_id | product\_name | category\_name

-------------+------------+-----------------+---------------

1 | 1 | iPhone | Smart Phone

1 | 2 | Samsung Galaxy | Smart Phone

2 | 3 | HP Elite | Laptop

2 | 4 | Lenovo Thinkpad | Laptop

3 | 5 | iPad | Tablet

3 | 6 | Kindle Fire | Tablet

(6 rows)

Cross join

**postgres=#SELECT \* FROM products CROSS JOIN categories;**

product\_id | product\_name | category\_id | category\_id | category\_name

------------+-----------------+-------------+-------------+---------------

1 | iPhone | 1 | 1 | Smart Phone

1 | iPhone | 1 | 2 | Laptop

1 | iPhone | 1 | 3 | Tablet

1 | iPhone | 1 | 4 | Smart Phone

1 | iPhone | 1 | 5 | Laptop

1 | iPhone | 1 | 6 | Tablet

2 | Samsung Galaxy | 1 | 1 | Smart Phone

2 | Samsung Galaxy | 1 | 2 | Laptop

2 | Samsung Galaxy | 1 | 3 | Tablet

2 | Samsung Galaxy | 1 | 4 | Smart Phone

2 | Samsung Galaxy | 1 | 5 | Laptop

2 | Samsung Galaxy | 1 | 6 | Tablet

3 | HP Elite | 2 | 1 | Smart Phone

3 | HP Elite | 2 | 2 | Laptop

3 | HP Elite | 2 | 3 | Tablet

3 | HP Elite | 2 | 4 | Smart Phone

3 | HP Elite | 2 | 5 | Laptop

3 | HP Elite | 2 | 6 | Tablet

4 | Lenovo Thinkpad | 2 | 1 | Smart Phone

4 | Lenovo Thinkpad | 2 | 2 | Laptop

4 | Lenovo Thinkpad | 2 | 3 | Tablet

4 | Lenovo Thinkpad | 2 | 4 | Smart Phone

4 | Lenovo Thinkpad | 2 | 5 | Laptop

4 | Lenovo Thinkpad | 2 | 6 | Tablet

5 | iPad | 3 | 1 | Smart Phone

5 | iPad | 3 | 2 | Laptop

5 | iPad | 3 | 3 | Tablet

5 | iPad | 3 | 4 | Smart Phone

5 | iPad | 3 | 5 | Laptop

5 | iPad | 3 | 6 | Tablet

6 | Kindle Fire | 3 | 1 | Smart Phone

6 | Kindle Fire | 3 | 2 | Laptop

6 | Kindle Fire | 3 | 3 | Tablet

6 | Kindle Fire | 3 | 4 | Smart Phone

6 | Kindle Fire | 3 | 5 | Laptop

6 | Kindle Fire | 3 | 6 | Tablet

(36 rows)

Group

**postgres=# SELECT no FROM test GROUP BY no;**

no

----

11

9

3

17

5

4

10

6

14

13

2

7

12

1

8

(15 rows)

**postgres=# SELECT age,COUNT(\*) FROM test GROUP BY age;**

age | count

-----+-------

43 | 1

26 | 1

21 | 2

28 | 1

29 | 1

34 | 1

35 | 2

41 | 1

24 | 1

25 | 4

23 | 1

(11 rows)

**postgres=# SELECT age,COUNT(\*) FROM test GROUP BY age;**

age | count

-----+-------

43 | 1

26 | 1

21 | 2

28 | 1

29 | 1

34 | 1

35 | 2

41 | 1

24 | 1

25 | 4

23 | 1

**postgres=# SELECT age,COUNT(\*) FROM test GROUP BY age ORDER BY age;**

age | count

-----+-------

21 | 2

23 | 1

24 | 1

25 | 4

26 | 1

28 | 1

29 | 1

34 | 1

35 | 2

41 | 1

43 | 1

(11 rows)

**postgres=# SELECT age,SUM(age) FROM test GROUP BY age ORDER BY age;**

age | sum

-----+-----

21 | 42

23 | 23

24 | 24

25 | 100

26 | 26

28 | 28

29 | 29

34 | 34

35 | 70

41 | 41

43 | 43

**postgres=# SELECT age,SUM(age),COUNT(age) FROM test GROUP BY age ORDER BY age;**

age | sum | count

-----+-----+-------

21 | 42 | 2

23 | 23 | 1

24 | 24 | 1

25 | 100 | 4

26 | 26 | 1

28 | 28 | 1

29 | 29 | 1

34 | 34 | 1

35 | 70 | 2

41 | 41 | 1

43 | 43 | 1

(11 rows)

Having

**postgres=# SELECT age,SUM(age) FROM test GROUP BY age HAVING SUM(age) >50;**

age | sum

-----+-----

35 | 70

25 | 100

(2 rows)

**postgres=# SELECT age,SUM(age) FROM test GROUP BY age HAVING SUM(age) >40;**

age | sum

-----+-----

43 | 43

21 | 42

35 | 70

41 | 41

25 | 100

(5 rows)

Union

**postgres=# SELECT name,age FROM test UNION SELECT name,age FROM test\_1;**

name | age

--------+-----

guna | 26

ravi | 24

kannan | 39

mani | 35

dinesh | 27

anbu | 24

rani | 23

roja | 21

mathi | 43

ram | 25

bala | 34

kavi | 21

arun | 25

deva | 28

raj | 35

mani | 41

raja | 25

mohan | 28

raja | 35

ravi | 25

guru | 29

kumar | 32

(22 rows)

Union All

**postgres=# SELECT name,age FROM test UNION ALL SELECT name,age FROM test\_1;**

name | age

--------+-----

ram | 25

raj | 35

raja | 35

raja | 25

ravi | 25

arun | 25

anbu | 24

bala | 34

deva | 28

mani | 41

kavi | 21

roja | 21

rani | 23

guna | 26

guru | 29

mathi | 43

mathi | 43

ravi | 24

anbu | 24

mani | 35

dinesh | 27

mohan | 28

kannan | 39

kumar | 32

kavi | 21

raja | 25

guna | 26

arun | 25

(28 rows)

Union With Order By

**postgres=# SELECT name,age FROM test UNION SELECT name,age FROM test\_1 ORDER BY name;**

name | age

--------+-----

anbu | 24

arun | 25

bala | 34

deva | 28

dinesh | 27

guna | 26

guru | 29

kannan | 39

kavi | 21

kumar | 32

mani | 41

mani | 35

mathi | 43

mohan | 28

raj | 35

raja | 35

raja | 25

ram | 25

rani | 23

ravi | 25

ravi | 24

roja | 21

(22 rows)

Intersect

**postgres=# SELECT name FROM test INTERSECT SELECT name FROM test\_1;**

name

-------

mathi

arun

guna

raja

anbu

kavi

ravi

mani

(8 rows)

**postgres=# SELECT name FROM test INTERSECT SELECT name FROM test\_1 ORDER BY name;**

name

-------

anbu

arun

guna

kavi

mani

mathi

raja

ravi

(8 rows)

EXCEPT

**postgres=# SELECT name FROM test EXCEPT SELECT name FROM test\_1;**

name

------

deva

rani

bala

raj

ram

guru

roja

(7 rows)

**postgres=# SELECT name FROM test EXCEPT SELECT name FROM test\_1 ORDER BY name;**

name

------

bala

deva

guru

raj

ram

rani

roja

(7 rows)

**postgres=# SELECT DISTINCT test.name,test.no FROM test EXCEPT SELECT DISTINCT test\_1.name,test\_1.no FROM test\_1 INNER JOIN test ON test\_1.name = test.name;**

name | no

-------+----

arun | 5

guru | 17

roja | 11

bala | 7

deva | 10

raja | 2

raja | 3

mani | 14

mathi | 13

ram | 1

kavi | 8

ravi | 4

rani | 9

raj | 2

anbu | 6

(15 rows)

Grouping SET

**postgres=# SELECT no,name FROM test GROUP BY GROUPING SETS(no,name);**

no | name

----+-------

11 |

9 |

3 |

17 |

5 |

4 |

10 |

6 |

14 |

13 |

2 |

7 |

12 |

1 |

8 |

| mathi

| deva

| arun

| guna

| raja

| anbu

| rani

| bala

| raj

| kavi

| ram

| guru

| roja

| ravi

| mani

(30 rows)

**postgres=# SELECT no,name,age FROM test GROUP BY GROUPING SETS(no,name,age);**

no | name | age

----+-------+-----

11 | |

9 | |

3 | |

17 | |

5 | |

4 | |

10 | |

6 | |

14 | |

13 | |

2 | |

7 | |

12 | |

1 | |

8 | |

| mathi |

| deva |

| arun |

| guna |

| raja |

| anbu |

| rani |

| bala |

| raj |

| kavi |

| ram |

| guru |

| roja |

| ravi |

| mani |

| | 43

| | 26

| | 21

| | 28

| | 29

| | 34

| | 35

| | 41

| | 24

| | 25

| | 23

(41 rows)

**postgres=# SELECT no,name,SUM(age) FROM test GROUP BY GROUPING SETS(no,name,age);**

no | name | sum

----+-------+-----

11 | | 21

9 | | 23

3 | | 25

17 | | 29

5 | | 25

4 | | 25

10 | | 28

6 | | 24

14 | | 41

13 | | 43

2 | | 70

7 | | 34

12 | | 26

1 | | 25

8 | | 21

| mathi | 43

| deva | 28

| arun | 25

| guna | 26

| raja | 60

| anbu | 24

| rani | 23

| bala | 34

| raj | 35

| kavi | 21

| ram | 25

| guru | 29

| roja | 21

| ravi | 25

| mani | 41

| | 43

| | 26

| | 42

| | 28

| | 29

| | 34

| | 70

| | 41

| | 24

| | 100

| | 23

(41 rows)

**postgres=# SELECT no,name,age FROM test GROUP BY GROUPING SETS((no,name,age),(no,name),(no));**

no | name | age

----+-------+-----

13 | mathi | 43

9 | rani | 23

2 | raja | 35

3 | raja | 25

17 | guru | 29

5 | arun | 25

4 | ravi | 25

11 | roja | 21

12 | guna | 26

8 | kavi | 21

6 | anbu | 24

2 | raj | 35

10 | deva | 28

7 | bala | 34

14 | mani | 41

1 | ram | 25

2 | raja |

9 | rani |

1 | ram |

17 | guru |

12 | guna |

3 | raja |

14 | mani |

7 | bala |

8 | kavi |

13 | mathi |

10 | deva |

2 | raj |

6 | anbu |

4 | ravi |

11 | roja |

5 | arun |

11 | |

9 | |

3 | |

17 | |

5 | |

4 | |

10 | |

6 | |

14 | |

13 | |

2 | |

7 | |

12 | |

1 | |

8 | |

(47 rows)

Grouping set(function)

**postgres=# SELECT GROUPING(no) grouping\_no,GROUPING(name) grouping\_name,age FROM test GROUP BY GROUPING SETS((no,name,age));**

grouping\_no | grouping\_name | age

-------------+---------------+-----

0 | 0 | 43

0 | 0 | 23

0 | 0 | 35

0 | 0 | 25

0 | 0 | 29

0 | 0 | 25

0 | 0 | 25

0 | 0 | 21

0 | 0 | 26

0 | 0 | 21

0 | 0 | 24

0 | 0 | 35

0 | 0 | 28

0 | 0 | 34

0 | 0 | 41

0 | 0 | 25

(16 rows)

CUBE

**SELECT name,age FROM test GROUP BY CUBE(name,age);**

name | age

-------+-----

|

mathi | 43

ram | 25

guna | 26

bala | 34

kavi | 21

arun | 25

deva | 28

raj | 35

mani | 41

raja | 25

raja | 35

anbu | 24

rani | 23

ravi | 25

roja | 21

guru | 29

mathi |

deva |

arun |

guna |

raja |

anbu |

rani |

bala |

raj |

kavi |

ram |

guru |

roja |

ravi |

mani |

| 43

| 26

| 21

| 28

| 29

| 34

| 35

| 41

| 24

| 25

| 23

(43 rows)

RollUP

**postgres=# SELECT no,name,SUM(age) FROM test GROUP BY no,ROLLUP(name,age);**

no | name | sum

----+-------+-----

13 | mathi | 43

9 | rani | 23

2 | raja | 35

3 | raja | 25

17 | guru | 29

5 | arun | 25

4 | ravi | 25

11 | roja | 21

12 | guna | 26

8 | kavi | 21

6 | anbu | 24

2 | raj | 35

10 | deva | 28

7 | bala | 34

14 | mani | 41

1 | ram | 25

2 | raja | 35

9 | rani | 23

1 | ram | 25

17 | guru | 29

12 | guna | 26

3 | raja | 25

14 | mani | 41

7 | bala | 34

8 | kavi | 21

13 | mathi | 43

10 | deva | 28

2 | raj | 35

6 | anbu | 24

4 | ravi | 25

11 | roja | 21

5 | arun | 25

11 | | 21

9 | | 23

3 | | 25

17 | | 29

5 | | 25

4 | | 25

10 | | 28

6 | | 24

14 | | 41

13 | | 43

2 | | 70

7 | | 34

12 | | 26

1 | | 25

8 | | 21

(47 rows)

Subquery

**postgres=# SELECT no,name,age FROM test WHERE EXISTS(SELECT no,name FROM test\_1 WHERE test\_1.name = test.name) ORDER BY name;**

no | name | age

----+-------+-----

6 | anbu | 24

5 | arun | 25

12 | guna | 26

8 | kavi | 21

14 | mani | 41

13 | mathi | 43

3 | raja | 25

2 | raja | 35

4 | ravi | 25

(9 rows)

**postgres=# SELECT film\_id,title FROM film WHERE FILM\_id IN(SELECT inventory.film\_id FROM rental INNER JOIN inventory ON inventory.inventory\_id = rental.inventory\_id WHERE return\_date BETWEEN '2005-05-29' AND '2005-05-30');**

film\_id | title

---------+-----------------------

307 | Fellowship Autumn

255 | Driving Polish

388 | Gunfight Moon

130 | Celebrity Horn

563 | Massacre Usual

397 | Hanky October

898 | Tourist Pelican

228 | Detective Vision

347 | Games Bowfinger

1000 | Zorro Ark

624 | Nightmare Chill

179 | Conquerer Nuts

54 | Banger Pinocchio

684 | Pizza Jumanji

267 | Eagles Panky

68 | Betrayed Rear

868 | Superfly Trip

138 | Chariots Conspiracy

418 | Hobbit Alien

681 | Pirates Roxanne

52 | Ballroom Mockingbird

83 | Blues Instinct

858 | Submarine Bed

45 | Attraction Newton

958 | Wardrobe Phantom

783 | Shane Darkness

# EXISTS

**postgres=# SELECT no,name,age FROM test WHERE EXISTS(SELECT no FROM test\_1 WHERE test\_1.no = test.no);**

no | name | age

----+------+-----

1 | ram | 25

2 | raj | 35

2 | raja | 35

3 | raja | 25

4 | ravi | 25

5 | arun | 25

6 | anbu | 24

7 | bala | 34

10 | deva | 28

8 | kavi | 21

11 | roja | 21

9 | rani | 23

12 | guna | 26

(13 rows)

**postgres=# SELECT no,name,age FROM test WHERE EXISTS(SELECT no FROM test\_1 WHERE test\_1.no = test.no) ORDER BY name;**

no | name | age

----+------+-----

6 | anbu | 24

5 | arun | 25

7 | bala | 34

10 | deva | 28

12 | guna | 26

8 | kavi | 21

2 | raj | 35

3 | raja | 25

2 | raja | 35

1 | ram | 25

9 | rani | 23

4 | ravi | 25

11 | roja | 21

(13 rows)