PostgreSQL

Database: is a computer server where you can store, manipulate and retrieve data.

Postgres is the database engine, where SQL is Structured Query Language.

Data are stored in tables, which are made up of columns and rows. The tables are named and the columns are also named, rows are mostly represented by an id. For example:

Table named coins has the following columns:

['id', 'bitcoin\_value', 'etherem\_value', 'tether\_value', 'binance\_value',

'date\_time']

and Rows:

|  | **id** | **bitcoin\_value** | **etherem\_value** | **tether\_value** | **binance\_value** | **date\_time** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | 29681.9264 | 1767.9025 | 0.9991 | 1.0003 | 2022-06-04 13:04:07.945000+00:00 |
| **1** | 2 | 29681.9264 | 1767.9025 | 0.9991 | 1.0003 | 2022-06-04 13:04:16.774000+00:00 |

Relational database: SQL allows us to relate two or more data base or tables

Tables are related by the use of ids. Example:

Table1 named: post

|  | **id** | **title** | **body** | **date\_time** | **attached\_image** | **author\_id** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Disclaimer | The Site Does Not Guarantee That The Sites Wil... | 2022-06-07 09:45:45.596102+00:00 |  | 1 |

Table2 named: user

|  | **id** | **password** | **last\_login** | **username** | **email** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1 | $Xj... | 2022-06-04 12:59:54.037000+00:00 | chukwunonsodavid | nonsoilonze@gmail.com |

The author\_id in Table 1 is related to Table 2 by the id number 1.

1. Install Postgres and PGadmin.
2. Open Terminal
3. Run~ psql -U postgres
4. Enter a password
5. Run~ \l -> to list all the database in the system

postgres=# \l

**List of databases**

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

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

bitblog | postgres | UTF8 | C | C |

postgres | postgres | UTF8 | C | C |

template0 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

(4 rows)

1. Run ~ CREATE DATABASE databasename; 🡪 to create a databae

postgres=# CREATE DATABASE may;

CREATE DATABASE

postgres=# \l

List of databases

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

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

bitblog | postgres | UTF8 | C | C |

may | postgres | UTF8 | C | C |

postgres | postgres | UTF8 | C | C |

template0 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

(5 rows)

chukwunonsodavid@King2-MacBook-Pro ~ % psql --help

psql is the PostgreSQL interactive terminal.

Usage:

psql [OPTION]... [DBNAME [USERNAME]]

General options:

-c, --command=COMMAND run only single command (SQL or internal) and exit

-d, --dbname=DBNAME database name to connect to (default: "chukwunonsodavid")

-f, --file=FILENAME execute commands from file, then exit

-l, --list list available databases, then exit

-v, --set=, --variable=NAME=VALUE

set psql variable NAME to VALUE

(e.g., -v ON\_ERROR\_STOP=1)

-V, --version output version information, then exit

-X, --no-psqlrc do not read startup file (~/.psqlrc)

-1 ("one"), --single-transaction

execute as a single transaction (if non-interactive)

-?, --help[=options] show this help, then exit

--help=commands list backslash commands, then exit

--help=variables list special variables, then exit

Input and output options:

-a, --echo-all echo all input from script

-b, --echo-errors echo failed commands

-e, --echo-queries echo commands sent to server

-E, --echo-hidden display queries that internal commands generate

-L, --log-file=FILENAME send session log to file

-n, --no-readline disable enhanced command line editing (readline)

-o, --output=FILENAME send query results to file (or |pipe)

-q, --quiet run quietly (no messages, only query output)

-s, --single-step single-step mode (confirm each query)

-S, --single-line single-line mode (end of line terminates SQL command)

Output format options:

-A, --no-align unaligned table output mode

--csv CSV (Comma-Separated Values) table output mode

-F, --field-separator=STRING

field separator for unaligned output (default: "|")

-H, --html HTML table output mode

-P, --pset=VAR[=ARG] set printing option VAR to ARG (see \pset command)

-R, --record-separator=STRING

record separator for unaligned output (default: newline)

-t, --tuples-only print rows only

-T, --table-attr=TEXT set HTML table tag attributes (e.g., width, border)

-x, --expanded turn on expanded table output

-z, --field-separator-zero

set field separator for unaligned output to zero byte

-0, --record-separator-zero

set record separator for unaligned output to zero byte

Connection options:

-h, --host=HOSTNAME database server host or socket directory (default: "local socket")

-p, --port=PORT database server port (default: "5432")

-U, --username=USERNAME database user name (default: "chukwunonsodavid")

-w, --no-password never prompt for password

-W, --password force password prompt (should happen automatically)

For more information, type "\?" (for internal commands) or "\help" (for SQL

commands) from within psql, or consult the psql section in the PostgreSQL

documentation.

Report bugs to <pgsql-bugs@lists.postgresql.org>.

PostgreSQL home page: <https://www.postgresql.org/>

1. To connect to a database

chukwunonsodavid@King2-MacBook-Pro ~ % psql -h localhost -p 5432 -U postgres may

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

may=#

or

chukwunonsodavid@King2-MacBook-Pro ~ % psql -U postgres

Password for user postgres:

psql (14.3)

Type "help" for help.

postgres=# \l

List of databases

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

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

bitblog | postgres | UTF8 | C | C |

may | postgres | UTF8 | C | C |

postgres | postgres | UTF8 | C | C |

template0 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | C | C | =c/postgres +

| | | | | postgres=CTc/postgres

(5 rows)

postgres=# \c may

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

may=#

1. To delete a data base: DROP DATABASE nameofdatabase;
2. Create table code:

CREATE TABLE table\_name(

Column name + data type +constraints if any

)

Example:

CREATE TABLE person (

id int,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

gender VARCHAR(6),

date\_of\_birth TIMESTAMP,

)

To check for other data types in postgres: [www.postgresql.org/docs/11/datatype.html](http://www.postgresql.org/docs/11/datatype.html)

**Table 8.1. Data Types**

| **Name** | **Aliases** | **Description** |
| --- | --- | --- |
| bigint | int8 | signed eight-byte integer |
| bigserial | serial8 | autoincrementing eight-byte integer |
| bit [ (***n***) ] |  | fixed-length bit string |
| bit varying [ (***n***) ] | varbit [ (***n***) ] | variable-length bit string |
| boolean | bool | logical Boolean (true/false) |
| box |  | rectangular box on a plane |
| bytea |  | binary data (“byte array”) |
| character [ (***n***) ] | char [ (***n***) ] | fixed-length character string |
| character varying [ (***n***) ] | varchar [ (***n***) ] | variable-length character string |
| cidr |  | IPv4 or IPv6 network address |
| circle |  | circle on a plane |
| date |  | calendar date (year, month, day) |
| double precision | float8 | double precision floating-point number (8 bytes) |
| inet |  | IPv4 or IPv6 host address |
| integer | int, int4 | signed four-byte integer |
| interval [ ***fields*** ] [ (***p***) ] |  | time span |
| json |  | textual JSON data |
| jsonb |  | binary JSON data, decomposed |
| line |  | infinite line on a plane |
| lseg |  | line segment on a plane |
| macaddr |  | MAC (Media Access Control) address |
| macaddr8 |  | MAC (Media Access Control) address (EUI-64 format) |
| money |  | currency amount |
| numeric [ (***p***, ***s***) ] | decimal [ (***p***, ***s***) ] | exact numeric of selectable precision |
| path |  | geometric path on a plane |
| pg\_lsn |  | PostgreSQL Log Sequence Number |
| point |  | geometric point on a plane |
| polygon |  | closed geometric path on a plane |
| real | float4 | single precision floating-point number (4 bytes) |
| smallint | int2 | signed two-byte integer |
| smallserial | serial2 | autoincrementing two-byte integer |
| serial | serial4 | autoincrementing four-byte integer |
| text |  | variable-length character string |
| time [ (***p***) ] [ without time zone ] |  | time of day (no time zone) |
| time [ (***p***) ] with time zone | timetz | time of day, including time zone |
| timestamp [ (***p***) ] [ without time zone ] |  | date and time (no time zone) |
| timestamp [ (***p***) ] with time zone | timestamptz | date and time, including time zone |
| tsquery |  | text search query |
| tsvector |  | text search document |
| txid\_snapshot |  | user-level transaction ID snapshot |
| uuid |  | universally unique identifier |
| xml |  | XML data |

may=# CREATE TABLE person(

may(# id INT,

may(# first\_name VARCHAR(50),

may(# last\_name VARCHAR(50),

may(# gender VARCHAR(6),

may(# date\_of\_birth DATE);

CREATE TABLE

may=# \d

List of relations

Schema | Name | Type | Owner

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

public | person | table | postgres

(1 row)

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | integer | | |

first\_name | character varying(50) | | |

last\_name | character varying(50) | | |

gender | character varying(6) | | |

date\_of\_birth | date | | |

1. Delete table code: DROP TABLE tablename ;

Example :

may=# DROP TABLE person;

DROP TABLE

may=# \d

Did not find any relations.

may=# \d person

Did not find any relation named "person".

1. Creating the table with Nullable attributes:

may=# CREATE TABLE person(

may(# id BIGSERIAL NOT NULL PRIMARY KEY,

may(# first\_name VARCHAR(50) NOT NULL,

may(# other\_name VARCHAR(50),

may(# last\_name VARCHAR(50) NOT NULL,

may(# gender VARCHAR(10) NOT NULL,

may(# date\_of\_birth DATE NOT NULL,

may(# email VARCHAR(200));

CREATE TABLE

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('person\_id\_seq'::regclass)

first\_name | character varying(50) | | not null |

other\_name | character varying(50) | | |

last\_name | character varying(50) | | not null |

gender | character varying(10) | | not null |

date\_of\_birth | date | | not null |

email | character varying(200) | | |

Indexes:

"person\_pkey" PRIMARY KEY, btree (id)

may=# \d

List of relations

Schema | Name | Type | Owner

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

public | person | table | postgres

public | person\_id\_seq | sequence | postgres

(2 rows)

1. Insert records into tables:

The table is empty

|  | **id** | **first\_name** | **other\_name** | **last\_name** | **gender** | **date\_of\_birth** | **email** |
| --- | --- | --- | --- | --- | --- | --- | --- |

Code:

INSERT INTO tablename (

first\_name,

last\_name,

gender,

date\_of\_birth)

VALUES (‘Chukwunonso’, ‘Ilonze’, ‘Male’, DATE ‘1992-03-08’) ;

Or

INSERT INTO tablename (

first\_name,

other\_name,

last\_name,

gender,

date\_of\_birth,

email)

VALUES (‘Chukwunonso’, ‘David’, ‘Ilonze’, ‘Male’, DATE ‘1992-03-08’, ‘nonsoilonze@gmail.com’) ;

may=# INSERT INTO person (

may(# first\_name,

may(# last\_name,

may(# gender,

may(# date\_of\_birth)

may-# VALUES ('Precious', 'Ezuruike', 'Female', DATE '1994-09-10');

INSERT 0 1

|  | **id** | **first\_name** | **other\_name** | **last\_name** | **gender** | **date\_of\_birth** | **email** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Precious | None | Ezuruike | Female | 1994-09-10 | None |

may=# INSERT INTO person (

may(# first\_name,

may(# last\_name,

may(# other\_name,

may(# gender,

may(# date\_of\_birth,

may(# email)

may-# VALUES ('Chukwunonso', 'Ilonze', 'David', 'Male', DATE '1992-03-08', 'nonsoilonze@gmail.com');

INSERT 0 1

|  | **id** | **first\_name** | **other\_name** | **last\_name** | **gender** | **date\_of\_birth** | **email** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Precious | None | Ezuruike | Female | 1994-09-10 | None |
| **1** | 2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com |

Visit <https://mockaroo.com> to generate data

Download it as sql file.

Locate the file and run this command

\i /Users/chukwunonsodavid/Downloads/person.sql

create table person (

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(50),

gender VARCHAR(50),

date\_of\_birth DATE,

other\_name VARCHAR(50)

);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Melonie', 'Scopyn', 'mscopyn0@cpanel.net', 'Agender', '2004/08/04', null);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Gabrila', 'Acres', 'gacres1@smh.com.au', 'Female', '2009/10/29', null);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Dionisio', 'Eddicott', 'deddicott2@ycombinator.com', 'Male', '2016/03/15', 'Solène');

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Benoite', 'Janu', 'bjanu3@symantec.com', 'Female', '2017/09/01', null);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Barrie', 'Marco', null, 'Agender', '2015/07/06', null);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Conan', 'Barlass', 'cbarlass5@is.gd', 'Male', '2011/05/19', null);

insert into person (first\_name, last\_name, email, gender, date\_of\_birth, other\_name) values ('Lilyan', 'Espley', 'lespley6@nature.com', 'Female', '1997/08/13', 'Cléopatre');

|  | **id** | **first\_name** | **other\_name** | **last\_name** | **gender** | **date\_of\_birth** | **email** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Precious | None | Ezuruike | Female | 1994-09-10 | None |
| **1** | 2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com |
| **2** | 3 | Melonie | None | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net |
| **3** | 4 | Gabrila | None | Acres | Female | 2009-10-29 | gacres1@smh.com.au |
| **4** | 5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **95** | 96 | Susan | Marie-thérèse | Balsdon | Female | 2016-09-16 | sbalsdon2n@ning.com |
| **96** | 97 | Bobine | None | Walstow | Female | 2015-03-07 | None |
| **97** | 98 | Cully | Gwenaëlle | Planke | Non-binary | 2012-11-28 | cplanke2p@disqus.com |
| **98** | 99 | Gae | None | De Ferraris | Female | 1995-03-01 | None |
| **99** | 100 | Dyana | None | O'Hederscoll | Female | 1999-12-18 | dohederscoll2r@cnn.com |

100 rows × 7 columns

may=# SELECT \* FROM person;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

3 | Melonie | | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net

4 | Gabrila | | Acres | Female | 2009-10-29 | gacres1@smh.com.au

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

7 | Barrie | | Marco | Agender | 2015-07-06 |

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

11 | Tanner | | Gobert | Male | 2012-10-18 |

12 | Ebba | | Gorvin | Female | 2002-05-31 | egorvin9@livejournal.com

13 | Perri | Léa | Caen | Female | 1994-06-08 | pcaena@dedecms.com

14 | Evelyn | | Lannen | Male | 2015-10-17 | elannenb@paginegialle.it

15 | Wanids | | Reynault | Female | 2002-09-06 | wreynaultc@google.ca

16 | Nappie | | Courtman | Male | 2007-01-30 | ncourtmand@imageshack.us

17 | Mufi | Esbjörn | Scamp | Female | 2016-12-07 | mscampe@tinyurl.com

18 | Charisse | Irène | Overthrow | Female | 1997-04-17 | coverthrowf@netvibes.com

19 | Karmen | | Cheine | Female | 2002-05-15 |

may=# SELECT email FROM person;

email

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

nonsoilonze@gmail.com

mscopyn0@cpanel.net

gacres1@smh.com.au

deddicott2@ycombinator.com

bjanu3@symantec.com

cbarlass5@is.gd

lespley6@nature.com

egorvin9@livejournal.com

pcaena@dedecms.com

elannenb@paginegialle.it

wreynaultc@google.ca

ncourtmand@imageshack.us

mscampe@tinyurl.com

coverthrowf@netvibes.com

[pbrownbridgeh@dell.com](mailto:pbrownbridgeh@dell.com)

may=# SELECT id, email FROM person;

id | email

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

1 |

2 | nonsoilonze@gmail.com

3 | mscopyn0@cpanel.net

4 | gacres1@smh.com.au

5 | deddicott2@ycombinator.com

6 | bjanu3@symantec.com

7 |

8 | cbarlass5@is.gd

9 | lespley6@nature.com

10 |

11 |

12 | egorvin9@livejournal.com

13 | pcaena@dedecms.com

14 | elannenb@paginegialle.it

15 | wreynaultc@google.ca

16 | ncourtmand@imageshack.us

17 | mscampe@tinyurl.com

18 | coverthrowf@netvibes.com

19 |

20 | pbrownbridgeh@dell.com

21 |

may=# SELECT \* FROM person ORDER BY first\_name;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

52 | Abe | | Beldam | Male | 2019-10-22 | abeldam1d@dot.gov

54 | Adrian | Mélys | Cauthra | Female | 2004-10-27 |

43 | Amabelle | | Stanmore | Female | 2008-10-28 | astanmore14@buzzfeed.com

93 | Athene | Cécile | Stollenberg | Female | 1993-01-22 | astollenberg2k@i2i.jp

48 | Barbette | | Burrell | Female | 2014-03-24 |

7 | Barrie | | Marco | Agender | 2015-07-06 |

45 | Bartlet | | Garn | Non-binary | 2000-05-08 | bgarn16@ted.com

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

69 | Berty | Maëline | Michelle | Female | 2016-02-26 | bmichelle1w@umich.edu

97 | Bobine | | Walstow | Female | 2015-03-07 |

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

32 | Carmelita | | Dominici | Female | 1995-04-08 |

89 | Cesar | | Arnout | Male | 2009-08-20 | carnout2g@ehow.com

18 | Charisse | Irène | Overthrow | Female | 1997-04-17 | coverthrowf@netvibes.com

83 | Chucho | | Widdop | Male | 2020-02-28 | cwiddop2a@bbc.co.uk

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

55 | Clyde | Eliès | Stuart | Male | 1999-09-01 | cstuart1h@myspace.com

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

98 | Cully | Gwenaëlle | Planke | Non-binary | 2012-11-28 | cplanke2p@disqus.com

:

may=# SELECT \* FROM person ORDER BY first\_name DESC;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

81 | Zack | | Pettet | Male | 2016-01-28 |

75 | Zachary | | Jumonet | Male | 2005-06-02 | zjumonet22@springer.com

26 | Woodie | | Arne | Male | 2017-03-06 |

87 | Wood | | Martyn | Male | 1997-10-28 |

15 | Wanids | | Reynault | Female | 2002-09-06 | wreynaultc@google.ca

36 | Wallas | | De Pero | Male | 2013-11-13 |

60 | Trudy | Athéna | Buckberry | Female | 2003-02-20 | tbuckberry1n@constantcontact.com

59 | Tawnya | | Cisco | Female | 2018-12-06 |

79 | Tanya | Adélaïde | Reppaport | Female | 1998-02-03 |

11 | Tanner | | Gobert | Male | 2012-10-18 |

44 | Tab | | Skeats | Male | 1993-05-17 | tskeats15@house.gov

96 | Susan | Marie-thérèse | Balsdon | Female | 2016-09-16 | sbalsdon2n@ning.com

72 | Silvie | Cinéma | Follacaro | Female | 1995-06-01 | sfollacaro1z@house.gov

67 | Scott | | Huxton | Male | 2007-04-25 |

42 | Ruthi | | Dufall | Female | 2017-04-04 | rdufall13@cdc.gov

63 | Rube | | Levings | Male | 2017-02-02 | rlevings1q@ucla.edu

27 | Ronny | Maï | Claworth | Male | 1997-10-25 | rclawortho@dion.ne.jp

30 | Rochelle | | Drakes | Female | 2019-05-19 | rdrakesr@paginegialle.it

:

DISTINCT:

may=# SELECT DISTINCT gender FROM person ORDER BY gender;

gender

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

Agender

Female

Male

Non-binary

Polygender

(5 rows)

may=# SELECT DISTINCT first\_name FROM person ORDER BY first\_name;

first\_name

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

Abe

Adrian

Amabelle

Athene

Barbette

Barrie

Bartlet

Benoite

Berty

Bobine

Bondy

Carmelita

Cesar

Charisse

Chucho

Chukwunonso

Clyde

Conan

Cully

Daune

Desiree

:

WHERE:

may=# SELECT \* FROM person WHERE gender ='Female';

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

4 | Gabrila | | Acres | Female | 2009-10-29 | gacres1@smh.com.au

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

12 | Ebba | | Gorvin | Female | 2002-05-31 | egorvin9@livejournal.com

13 | Perri | Léa | Caen | Female | 1994-06-08 | pcaena@dedecms.com

15 | Wanids | | Reynault | Female | 2002-09-06 | wreynaultc@google.ca

17 | Mufi | Esbjörn | Scamp | Female | 2016-12-07 | mscampe@tinyurl.com

18 | Charisse | Irène | Overthrow | Female | 1997-04-17 | coverthrowf@netvibes.com

19 | Karmen | | Cheine | Female | 2002-05-15 |

20 | Philippa | | Brownbridge | Female | 2017-12-28 | pbrownbridgeh@dell.com

21 | Gweneth | | Atley | Female | 2014-09-12 |

25 | Jeri | | Hold | Female | 2011-09-01 |

29 | Emilee | | Wattinham | Female | 1998-11-07 |

30 | Rochelle | | Drakes | Female | 2019-05-19 | rdrakesr@paginegialle.it

31 | Dora | | Antrobus | Female | 2005-04-23 | dantrobuss@rakuten.co.jp

32 | Carmelita | | Dominici | Female | 1995-04-08 |

37 | Patience | Erwéi | Fenners | Female | 2000-05-19 | pfennersy@bandcamp.com

39 | Margaux | | Brownsea | Female | 2005-03-04 | mbrownsea10@sitemeter.com

41 | Ophelie | | Spellecy | Female | 2012-03-08 | ospellecy12@cbslocal.com

42 | Ruthi | | Dufall | Female | 2017-04-04 | rdufall13@cdc.gov

:

may=# SELECT \* FROM person WHERE gender ='Male';

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

11 | Tanner | | Gobert | Male | 2012-10-18 |

14 | Evelyn | | Lannen | Male | 2015-10-17 | elannenb@paginegialle.it

16 | Nappie | | Courtman | Male | 2007-01-30 | ncourtmand@imageshack.us

22 | Emmett | | Seldner | Male | 1994-05-11 |

23 | Kilian | Rachèle | Shugg | Male | 1999-01-31 | kshuggk@gov.uk

24 | Lindy | | January | Male | 2004-01-06 | ljanuaryl@ycombinator.com

26 | Woodie | | Arne | Male | 2017-03-06 |

27 | Ronny | Maï | Claworth | Male | 1997-10-25 | rclawortho@dion.ne.jp

28 | Orin | | Shaw | Male | 2018-05-08 |

33 | Jordan | | Roxburch | Male | 2007-01-15 |

34 | Donal | | Finlayson | Male | 2003-10-30 |

35 | Karoly | | Kelston | Male | 2019-07-23 | kkelstonw@imageshack.us

36 | Wallas | | De Pero | Male | 2013-11-13 |

38 | Dougie | Léonore | Koop | Male | 2018-01-14 | dkoopz@merriam-webster.com

40 | Enrico | | Telling | Male | 2017-10-23 | etelling11@usda.gov

44 | Tab | | Skeats | Male | 1993-05-17 | tskeats15@house.gov

47 | Garald | Nélie | Veelers | Male | 2015-02-08 | gveelers18@diigo.com

:

may=# SELECT \* FROM person WHERE gender ='Non-binary';

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

45 | Bartlet | | Garn | Non-binary | 2000-05-08 | bgarn16@ted.com

65 | Ewart | | Loveland | Non-binary | 2018-09-25 | eloveland1s@examiner.com

98 | Cully | Gwenaëlle | Planke | Non-binary | 2012-11-28 | cplanke2p@disqus.com

(3 rows)

may=# SELECT \* FROM person WHERE gender ='Polygender';

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

46 | Munroe | | Kubin | Polygender | 1996-12-31 | mkubin17@elegantthemes.com

80 | Perrine | | Loyns | Polygender | 2003-10-06 | ployns27@discovery.com

(2 rows)

may=# SELECT first\_name, last\_name, gender FROM person WHERE gender ='Non-binary';

first\_name | last\_name | gender

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

Bartlet | Garn | Non-binary

Ewart | Loveland | Non-binary

Cully | Planke | Non-binary

(3 rows)

may=# SELECT first\_name, last\_name, gender FROM person WHERE gender ='Non-binary' ORDER BY first\_name;

first\_name | last\_name | gender

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

Bartlet | Garn | Non-binary

Cully | Planke | Non-binary

Ewart | Loveland | Non-binary

(3 rows)

may=# SELECT first\_name, last\_name, gender FROM person WHERE gender ='Non-binary' AND first\_name ='Cully';

first\_name | last\_name | gender

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

Cully | Planke | Non-binary

(1 row)

may=# SELECT first\_name, last\_name, gender FROM person WHERE gender ='Non-binary' OR gender = 'Polygender';

first\_name | last\_name | gender

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

Bartlet | Garn | Non-binary

Munroe | Kubin | Polygender

Ewart | Loveland | Non-binary

Perrine | Loyns | Polygender

Cully | Planke | Non-binary

(5 rows)

Comparison Operations:

Not equal to: <> (which is represented by != in most programming language)

may=# SELECT \* FROM person LIMIT 10;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

3 | Melonie | | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net

4 | Gabrila | | Acres | Female | 2009-10-29 | gacres1@smh.com.au

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

7 | Barrie | | Marco | Agender | 2015-07-06 |

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

(10 rows)

may=# SELECT \* FROM person WHERE gender ='Male' LIMIT 10;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

11 | Tanner | | Gobert | Male | 2012-10-18 |

14 | Evelyn | | Lannen | Male | 2015-10-17 | elannenb@paginegialle.it

16 | Nappie | | Courtman | Male | 2007-01-30 | ncourtmand@imageshack.us

22 | Emmett | | Seldner | Male | 1994-05-11 |

23 | Kilian | Rachèle | Shugg | Male | 1999-01-31 | kshuggk@gov.uk

24 | Lindy | | January | Male | 2004-01-06 | ljanuaryl@ycombinator.com

(10 rows)

may=# SELECT \* FROM person OFFSET 5 LIMIT 10;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

7 | Barrie | | Marco | Agender | 2015-07-06 |

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

11 | Tanner | | Gobert | Male | 2012-10-18 |

12 | Ebba | | Gorvin | Female | 2002-05-31 | egorvin9@livejournal.com

13 | Perri | Léa | Caen | Female | 1994-06-08 | pcaena@dedecms.com

14 | Evelyn | | Lannen | Male | 2015-10-17 | elannenb@paginegialle.it

15 | Wanids | | Reynault | Female | 2002-09-06 | wreynaultc@google.ca

(10 rows)

may=# SELECT \* FROM person OFFSET 95;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

96 | Susan | Marie-thérèse | Balsdon | Female | 2016-09-16 | sbalsdon2n@ning.com

97 | Bobine | | Walstow | Female | 2015-03-07 |

98 | Cully | Gwenaëlle | Planke | Non-binary | 2012-11-28 | cplanke2p@disqus.com

99 | Gae | | De Ferraris | Female | 1995-03-01 |

100 | Dyana | | O'Hederscoll | Female | 1999-12-18 | dohederscoll2r@cnn.com

(5 rows)

may=# SELECT \* FROM person OFFSET 5 FETCH FIRST 5 ROW ONLY;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

7 | Barrie | | Marco | Agender | 2015-07-06 |

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

(5 rows)

may=# SELECT \* FROM person OFFSET 5 LIMIT 5;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

7 | Barrie | | Marco | Agender | 2015-07-06 |

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

9 | Lilyan | Cléopatre | Espley | Female | 1997-08-13 | lespley6@nature.com

10 | Bondy | | Eaglesham | Male | 1995-05-06 |

(5 rows)

**may=# SELECT first\_name, last\_name, gender FROM person WHERE gender ='Non-binary' OR gender = 'Polygender';**

first\_name | last\_name | gender

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

Bartlet | Garn | Non-binary

Munroe | Kubin | Polygender

Ewart | Loveland | Non-binary

Perrine | Loyns | Polygender

Cully | Planke | Non-binary

(5 rows)

**may=# SELECT first\_name, last\_name, gender FROM person WHERE gender IN ('Non-binary', 'Polygender');**

first\_name | last\_name | gender

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

Bartlet | Garn | Non-binary

Munroe | Kubin | Polygender

Ewart | Loveland | Non-binary

Perrine | Loyns | Polygender

Cully | Planke | Non-binary

(5 rows)

**may=# SELECT \* FROM person WHERE date\_of\_birth BETWEEN '2017-09-01' AND '2022-09-01';**

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

20 | Philippa | | Brownbridge | Female | 2017-12-28 | pbrownbridgeh@dell.com

28 | Orin | | Shaw | Male | 2018-05-08 |

30 | Rochelle | | Drakes | Female | 2019-05-19 | rdrakesr@paginegialle.it

35 | Karoly | | Kelston | Male | 2019-07-23 | kkelstonw@imageshack.us

38 | Dougie | Léonore | Koop | Male | 2018-01-14 | dkoopz@merriam-webster.com

40 | Enrico | | Telling | Male | 2017-10-23 | etelling11@usda.gov

49 | Lynnette | | Rentoll | Female | 2022-02-13 | lrentoll1a@freewebs.com

52 | Abe | | Beldam | Male | 2019-10-22 | abeldam1d@dot.gov

59 | Tawnya | | Cisco | Female | 2018-12-06 |

65 | Ewart | | Loveland | Non-binary | 2018-09-25 | eloveland1s@examiner.com

76 | Piggy | Salomé | Stoaks | Male | 2021-04-16 |

83 | Chucho | | Widdop | Male | 2020-02-28 | cwiddop2a@bbc.co.uk

(13 rows)

**may=# SELECT \* FROM person WHERE first\_name LIKE 'C%';**

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

8 | Conan | | Barlass | Male | 2011-05-19 | cbarlass5@is.gd

18 | Charisse | Irène | Overthrow | Female | 1997-04-17 | coverthrowf@netvibes.com

32 | Carmelita | | Dominici | Female | 1995-04-08 |

55 | Clyde | Eliès | Stuart | Male | 1999-09-01 | cstuart1h@myspace.com

83 | Chucho | | Widdop | Male | 2020-02-28 | cwiddop2a@bbc.co.uk

89 | Cesar | | Arnout | Male | 2009-08-20 | carnout2g@ehow.com

98 | Cully | Gwenaëlle | Planke | Non-binary | 2012-11-28 | cplanke2p@disqus.com

(8 rows)

**may=# SELECT \* FROM person WHERE first\_name LIKE 'P%';**

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

13 | Perri | Léa | Caen | Female | 1994-06-08 | pcaena@dedecms.com

20 | Philippa | | Brownbridge | Female | 2017-12-28 | pbrownbridgeh@dell.com

37 | Patience | Erwéi | Fenners | Female | 2000-05-19 | pfennersy@bandcamp.com

66 | Patrica | | Dever | Female | 2001-11-15 | pdever1t@usnews.com

76 | Piggy | Salomé | Stoaks | Male | 2021-04-16 |

80 | Perrine | | Loyns | Polygender | 2003-10-06 | ployns27@discovery.com

85 | Peyton | Liè | Spriddle | Male | 2002-07-07 | pspriddle2c@upenn.edu

(8 rows)

**may=# SELECT \* FROM person WHERE first\_name LIKE 'p%';**

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

(0 rows)

Note: LIKE is case sensitive.

While ILIKE is not:

**may=# SELECT \* FROM person WHERE first\_name ILIKE 'p%';**

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

13 | Perri | Léa | Caen | Female | 1994-06-08 | pcaena@dedecms.com

20 | Philippa | | Brownbridge | Female | 2017-12-28 | pbrownbridgeh@dell.com

37 | Patience | Erwéi | Fenners | Female | 2000-05-19 | pfennersy@bandcamp.com

66 | Patrica | | Dever | Female | 2001-11-15 | pdever1t@usnews.com

76 | Piggy | Salomé | Stoaks | Male | 2021-04-16 |

80 | Perrine | | Loyns | Polygender | 2003-10-06 | ployns27@discovery.com

85 | Peyton | Liè | Spriddle | Male | 2002-07-07 | pspriddle2c@upenn.edu

(8 rows)

GROUP BY:

**may=# SELECT DISTINCT gender FROM person ORDER BY gender;**

gender

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

Agender

Female

Male

Non-binary

Polygender

(5 rows)

**may=# SELECT gender FROM person GROUP BY gender;**

gender

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

Male

Polygender

Non-binary

Female

Agender

(5 rows)

**may=# SELECT gender, COUNT(\*) FROM person GROUP BY gender;**

gender | count

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

Male | 40

Polygender | 2

Non-binary | 3

Female | 53

Agender | 2

(5 rows)

**may=# SELECT gender, COUNT(\*) FROM person GROUP BY gender HAVING COUNT(\*) > 5;**

gender | count

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

Male | 40

Female | 53

(2 rows)

**GO TO MOCKAROO AND DOWNLOAD CAR.SQL FILE**

**may=# \i /Users/chukwunonsodavid/Downloads/car.sql**

**may=# \d**

List of relations

Schema | Name | Type | Owner

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

public | car | table | postgres

public | car\_id\_seq | sequence | postgres

public | person | table | postgres

public | person\_id\_seq | sequence | postgres

(4 rows)

may=# \d car

Table "public.car"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('car\_id\_seq'::regclass)

make | character varying(100) | | not null |

model | character varying(100) | | not null |

price | numeric(19,2) | | not null |

Indexes:

"car\_pkey" PRIMARY KEY, btree (id)

**may=# SELECT \* FROM car**;

id | make | model | price

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

1 | Hyundai | Elantra | 8118.17

2 | Ford | F-Series | 3575.86

3 | Toyota | TundraMax | 2180.36

4 | Chevrolet | Suburban 1500 | 3112.80

5 | GMC | Jimmy | 2916.69

6 | Dodge | Caravan | 8536.79

7 | Chevrolet | Tahoe | 6971.71

8 | Subaru | Impreza | 9163.17

9 | Dodge | Viper | 1913.57

10 | Scion | xB | 8090.91

**may=# SELECT \* FROM car LIMIT 10;**

id | make | model | price

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

1 | Hyundai | Elantra | 8118.17

2 | Ford | F-Series | 3575.86

3 | Toyota | TundraMax | 2180.36

4 | Chevrolet | Suburban 1500 | 3112.80

5 | GMC | Jimmy | 2916.69

6 | Dodge | Caravan | 8536.79

7 | Chevrolet | Tahoe | 6971.71

8 | Subaru | Impreza | 9163.17

9 | Dodge | Viper | 1913.57

10 | Scion | xB | 8090.91

(10 rows)

**may=# SELECT \* FROM car ORDER BY price DESC LIMIT 10;**

id | make | model | price

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

54 | Suzuki | SJ | 9975.88

25 | Isuzu | i-290 | 9916.52

27 | Suzuki | SJ | 9850.64

65 | Land Rover | Freelander | 9803.25

28 | Pontiac | G6 | 9732.28

93 | Saturn | L-Series | 9649.27

49 | Lotus | Esprit | 9556.46

70 | Mazda | RX-7 | 9554.13

38 | Volvo | C70 | 9500.06

16 | Isuzu | Axiom | 9461.07

(10 rows)

**Maximum value:**

**may=# SELECT MAX(price) FROM car;**

max

---------

9975.88

(1 row)

**may=# SELECT \* FROM car ORDER BY price LIMIT 10;**

id | make | model | price

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

40 | Subaru | Forester | 1159.65

12 | Bentley | Continental GTC | 1236.22

68 | Toyota | T100 | 1244.64

60 | Acura | NSX | 1313.03

39 | Volkswagen | Passat | 1351.91

35 | Ford | Explorer Sport | 1570.40

30 | Acura | TL | 1643.03

75 | Chrysler | Concorde | 1643.31

58 | Ford | Thunderbird | 1689.17

71 | Mercury | Lynx | 1722.21

(10 rows)

**Minimum value:**

**may=# SELECT MIN(price) FROM car;**

min

---------

1159.65

(1 row)

**Average:**

**may=# SELECT AVG(price) FROM car;**

avg

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

5368.8782000000000000

(1 row)

**may=# SELECT ROUND(AVG(price)) FROM car;**

round

-------

5369

(1 row)

GROUPING THE MINIMUM VALUES FOR EACH CAR:

may=# SELECT make, MIN(price) FROM car GROUP BY make;

make | min

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

Honda | 1754.76

McLaren | 3522.95

Ford | 1570.40

Scion | 8090.91

Dodge | 1913.57

Chevrolet | 3112.80

Saturn | 9649.27

Infiniti | 8159.48

Bentley | 1236.22

Pontiac | 2204.06

Audi | 1834.99

Lexus | 4967.03

Kia | 5530.28

Lotus | 8339.48

Mercedes-Benz | 2528.78

Oldsmobile | 2786.00

Land Rover | 3751.98

Jeep | 2546.62

Buick | 2171.98

Volvo | 8132.56

Acura | 1313.03

Nissan | 3687.63

Citroën | 7444.65

Suzuki | 4433.83

Hyundai | 4404.77

Mitsubishi | 5044.14

Subaru | 1159.65

Mazda | 6009.85

BMW | 3124.58

Volkswagen | 1351.91

Mercury | 1722.21

Eagle | 7749.72

Hummer | 4396.99

Chrysler | 1643.31

Isuzu | 8027.91

Toyota | 1244.64

GMC | 2916.69

(37 rows)

may=# SELECT make, MAX(price) FROM car GROUP BY make;

make | max

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

Honda | 6322.97

McLaren | 3522.95

Ford | 7127.88

Scion | 8090.91

Dodge | 8536.79

Chevrolet | 7101.08

Saturn | 9649.27

Infiniti | 8159.48

Bentley | 1236.22

Pontiac | 9732.28

Audi | 6004.54

Lexus | 4967.03

Kia | 5530.28

Lotus | 9556.46

Mercedes-Benz | 5622.98

Oldsmobile | 3589.84

Land Rover | 9803.25

Jeep | 6399.58

Buick | 9069.88

Volvo | 9500.06

Acura | 1643.03

Nissan | 6779.28

Citroën | 7444.65

Suzuki | 9975.88

Hyundai | 8118.17

Mitsubishi | 7883.24

Subaru | 9163.17

Mazda | 9554.13

BMW | 5324.20

Volkswagen | 7056.01

Mercury | 6985.56

Eagle | 7749.72

Hummer | 4396.99

Chrysler | 5645.77

Isuzu | 9916.52

Toyota | 6071.17

GMC | 5374.73

(37 rows)

**may=# SELECT make, MAX(price) FROM car GROUP BY make ORDER BY max LIMIT 10;**

make | max

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

Bentley | 1236.22

Acura | 1643.03

McLaren | 3522.95

Oldsmobile | 3589.84

Hummer | 4396.99

Lexus | 4967.03

BMW | 5324.20

GMC | 5374.73

Kia | 5530.28

Mercedes-Benz | 5622.98

(10 rows)

**may=# SELECT SUM(price) FROM car;**

sum

-----------

536887.82

(1 row)

**may=# SELECT make, SUM(price) FROM car GROUP BY make;**

make | sum

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

Honda | 12445.34

McLaren | 3522.95

Ford | 51191.90

Scion | 8090.91

Dodge | 14844.57

Chevrolet | 29697.34

Saturn | 9649.27

Infiniti | 8159.48

Bentley | 1236.22

Pontiac | 34434.21

Audi | 7839.53

Lexus | 4967.03

Kia | 5530.28

Lotus | 26633.30

Mercedes-Benz | 11979.37

Oldsmobile | 6375.84

Land Rover | 18066.09

Jeep | 8946.20

Buick | 11241.86

Volvo | 17632.62

Acura | 2956.06

Nissan | 21471.70

Citroën | 7444.65

Suzuki | 24260.35

Hyundai | 20163.65

Mitsubishi | 26008.40

Subaru | 15819.21

Mazda | 15563.98

BMW | 8448.78

Volkswagen | 15287.26

Mercury | 11231.01

Eagle | 7749.72

Hummer | 4396.99

Chrysler | 10999.31

Isuzu | 27405.50

Toyota | 11892.11

GMC | 13304.83

(37 rows)

**may=# SELECT make, SUM(price) FROM car GROUP BY make ORDER BY sum;**

make | sum

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

Bentley | 1236.22

Acura | 2956.06

McLaren | 3522.95

Hummer | 4396.99

Lexus | 4967.03

Kia | 5530.28

Oldsmobile | 6375.84

Citroën | 7444.65

Eagle | 7749.72

Audi | 7839.53

Scion | 8090.91

Infiniti | 8159.48

BMW | 8448.78

Jeep | 8946.20

Saturn | 9649.27

Chrysler | 10999.31

Mercury | 11231.01

Buick | 11241.86

Toyota | 11892.11

Mercedes-Benz | 11979.37

Honda | 12445.34

GMC | 13304.83

Dodge | 14844.57

Volkswagen | 15287.26

Mazda | 15563.98

Subaru | 15819.21

Volvo | 17632.62

Land Rover | 18066.09

Hyundai | 20163.65

Nissan | 21471.70

Suzuki | 24260.35

Mitsubishi | 26008.40

Lotus | 26633.30

Isuzu | 27405.50

Chevrolet | 29697.34

Pontiac | 34434.21

Ford | 51191.90

(37 rows)

**may=# SELECT id, make, model, price, price \* .15 FROM car LIMIT 10;**

id | make | model | price | ?column?

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

1 | Hyundai | Elantra | 8118.17 | 1217.7255

2 | Ford | F-Series | 3575.86 | 536.3790

3 | Toyota | TundraMax | 2180.36 | 327.0540

4 | Chevrolet | Suburban 1500 | 3112.80 | 466.9200

5 | GMC | Jimmy | 2916.69 | 437.5035

6 | Dodge | Caravan | 8536.79 | 1280.5185

7 | Chevrolet | Tahoe | 6971.71 | 1045.7565

8 | Subaru | Impreza | 9163.17 | 1374.4755

9 | Dodge | Viper | 1913.57 | 287.0355

10 | Scion | xB | 8090.91 | 1213.6365

(10 rows)

**may=# SELECT id, make, model, price, ROUND(price \* .15, 2) FROM car LIMIT 10;**

id | make | model | price | round

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

1 | Hyundai | Elantra | 8118.17 | 1217.73

2 | Ford | F-Series | 3575.86 | 536.38

3 | Toyota | TundraMax | 2180.36 | 327.05

4 | Chevrolet | Suburban 1500 | 3112.80 | 466.92

5 | GMC | Jimmy | 2916.69 | 437.50

6 | Dodge | Caravan | 8536.79 | 1280.52

7 | Chevrolet | Tahoe | 6971.71 | 1045.76

8 | Subaru | Impreza | 9163.17 | 1374.48

9 | Dodge | Viper | 1913.57 | 287.04

10 | Scion | xB | 8090.91 | 1213.64

(10 rows)

**may=# SELECT id, make, model, price, ROUND(price \* .10, 2) AS ten\_percent FROM car LIMIT 10;**

id | make | model | price | ten\_percent

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

1 | Hyundai | Elantra | 8118.17 | 811.82

2 | Ford | F-Series | 3575.86 | 357.59

3 | Toyota | TundraMax | 2180.36 | 218.04

4 | Chevrolet | Suburban 1500 | 3112.80 | 311.28

5 | GMC | Jimmy | 2916.69 | 291.67

6 | Dodge | Caravan | 8536.79 | 853.68

7 | Chevrolet | Tahoe | 6971.71 | 697.17

8 | Subaru | Impreza | 9163.17 | 916.32

9 | Dodge | Viper | 1913.57 | 191.36

10 | Scion | xB | 8090.91 | 809.09

(10 rows)

may=# SELECT id, email FROM person LIMIT 10;

id | email

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

1 |

2 | nonsoilonze@gmail.com

3 | mscopyn0@cpanel.net

4 | gacres1@smh.com.au

5 | deddicott2@ycombinator.com

6 | bjanu3@symantec.com

7 |

8 | cbarlass5@is.gd

9 | lespley6@nature.com

10 |

(10 rows)

**Use of COALESCE to return value when table content is empty.**

**may=# SELECT id, COALESCE(email, 'no email') FROM person LIMIT 10;**

id | coalesce

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

1 | no email

2 | nonsoilonze@gmail.com

3 | mscopyn0@cpanel.net

4 | gacres1@smh.com.au

5 | deddicott2@ycombinator.com

6 | bjanu3@symantec.com

7 | no email

8 | cbarlass5@is.gd

9 | lespley6@nature.com

10 | no email

(10 rows)

**may=# SELECT id, COALESCE(email, 'no email') AS email FROM person LIMIT 10;**

id | email

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

1 | no email

2 | nonsoilonze@gmail.com

3 | mscopyn0@cpanel.net

4 | gacres1@smh.com.au

5 | deddicott2@ycombinator.com

6 | bjanu3@symantec.com

7 | no email

8 | cbarlass5@is.gd

9 | lespley6@nature.com

10 | no email

(10 rows)

may=# SELECT now();

now

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

2022-07-04 21:58:48.123015+01

(1 row)

may=# SELECT now() - INTERVAL '10 YEARS';

?column?

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

2012-07-04 22:01:05.213701+01

(1 row)

may=# SELECT now() - INTERVAL '10 DAYS';

?column?

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

2022-06-24 22:03:15.617134+01

(1 row)

may=# SELECT now() + INTERVAL '10 MONTHS';

?column?

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

2023-05-04 22:27:23.122871+01

(1 row)

may=# SELECT now()::DATE;

now

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

2022-07-04

(1 row)

**EXTRACT**

may=# SELECT EXTRACT(YEAR FROM now());

extract

---------

2022

(1 row)

may=# SELECT EXTRACT(DAY FROM now());

extract

---------

4

(1 row)

^

may=# SELECT EXTRACT(HOUR FROM now());

extract

---------

22

(1 row)

may=# SELECT first\_name, last\_name, gender, date\_of\_birth FROM person;

first\_name | last\_name | gender | date\_of\_birth

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

Precious | Ezuruike | Female | 1994-09-10

Chukwunonso | Ilonze | Male | 1992-03-08

Melonie | Scopyn | Agender | 2004-08-04

Gabrila | Acres | Female | 2009-10-29

Dionisio | Eddicott | Male | 2016-03-15

Benoite | Janu | Female | 2017-09-01

Barrie | Marco | Agender | 2015-07-06

Conan | Barlass | Male | 2011-05-19

Lilyan | Espley | Female | 1997-08-13

Bondy | Eaglesham | Male | 1995-05-06

Tanner | Gobert | Male | 2012-10-18

Ebba | Gorvin | Female | 2002-05-31

Perri | Caen | Female | 1994-06-08

Evelyn | Lannen | Male | 2015-10-17

Wanids | Reynault | Female | 2002-09-06

Nappie | Courtman | Male | 2007-01-30

Mufi | Scamp | Female | 2016-12-07

Charisse | Overthrow | Female | 1997-04-17

Karmen | Cheine | Female | 2002-05-15

Philippa | Brownbridge | Female | 2017-12-28

AGE:

may=# SELECT first\_name, last\_name, gender, date\_of\_birth, AGE(now(), date\_of\_birth) AS age FROM person;

first\_name | last\_name | gender | date\_of\_birth | age

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

Precious | Ezuruike | Female | 1994-09-10 | 27 years 9 mons 24 days 22:38:15.195024

Chukwunonso | Ilonze | Male | 1992-03-08 | 30 years 3 mons 27 days 22:38:15.195024

Melonie | Scopyn | Agender | 2004-08-04 | 17 years 11 mons 22:38:15.195024

Gabrila | Acres | Female | 2009-10-29 | 12 years 8 mons 6 days 22:38:15.195024

Dionisio | Eddicott | Male | 2016-03-15 | 6 years 3 mons 20 days 22:38:15.195024

Benoite | Janu | Female | 2017-09-01 | 4 years 10 mons 3 days 22:38:15.195024

Barrie | Marco | Agender | 2015-07-06 | 6 years 11 mons 29 days 22:38:15.195024

Conan | Barlass | Male | 2011-05-19 | 11 years 1 mon 16 days 22:38:15.195024

Lilyan | Espley | Female | 1997-08-13 | 24 years 10 mons 22 days 22:38:15.195024

Bondy | Eaglesham | Male | 1995-05-06 | 27 years 1 mon 29 days 22:38:15.195024

Tanner | Gobert | Male | 2012-10-18 | 9 years 8 mons 17 days 22:38:15.195024

Ebba | Gorvin | Female | 2002-05-31 | 20 years 1 mon 4 days 22:38:15.195024

Perri | Caen | Female | 1994-06-08 | 28 years 26 days 22:38:15.195024

Evelyn | Lannen | Male | 2015-10-17 | 6 years 8 mons 18 days 22:38:15.195024

Wanids | Reynault | Female | 2002-09-06 | 19 years 9 mons 28 days 22:38:15.195024

Nappie | Courtman | Male | 2007-01-30 | 15 years 5 mons 5 days 22:38:15.195024

Mufi | Scamp | Female | 2016-12-07 | 5 years 6 mons 28 days 22:38:15.195024

Charisse | Overthrow | Female | 1997-04-17 | 25 years 2 mons 17 days 22:38:15.195024

Karmen | Cheine | Female | 2002-05-15 | 20 years 1 mon 20 days 22:38:15.195024

Philippa | Brownbridge | Female | 2017-12-28 | 4 years 6 mons 7 days 22:38:15.195024

How to add item to the same id or primary key:

1st: alter the constraint

Code: ALTER TABLE person DROP CONSTRAINT person\_pkey;

2nd: insert the item

Code : insert into person(id, first\_name etc);

3rd: Add primary key

Code : ALTER TABLE person ADD PRIMARY KEY(id);

DELETE A ROW:

DELETE FROM person WHERE id = 1;

may=# SELECT \* FROM person LIMIT 5;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | | Ezuruike | Female | 1994-09-10 |

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

3 | Melonie | | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net

4 | Gabrila | | Acres | Female | 2009-10-29 | gacres1@smh.com.au

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

(5 rows)

may=# DELETE FROM person WHERE id = 1;

DELETE 1

may=# SELECT \* FROM person LIMIT 5;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

2 | Chukwunonso | David | Ilonze | Male | 1992-03-08 | nonsoilonze@gmail.com

3 | Melonie | | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net

4 | Gabrila | | Acres | Female | 2009-10-29 | gacres1@smh.com.au

5 | Dionisio | Solène | Eddicott | Male | 2016-03-15 | deddicott2@ycombinator.com

6 | Benoite | | Janu | Female | 2017-09-01 | bjanu3@symantec.com

(5 rows)

may=# INSERT INTO person (id, first\_name, other\_name, last\_name, gender, date\_of\_birth, email)

VALUES(1, 'Precious', 'Adaugo', 'Ezuruike', 'Female', '1994-09-10', 'pezuruike3@gmail.com');

INSERT 0 1

may=# SELECT \* FROM person WHERE id = 1;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

1 | Precious | Adaugo | Ezuruike | Female | 1994-09-10 | pezuruike3@gmail.com

(1 row)

may=# ALTER TABLE person ADD PRIMARY KEY(first\_name, last\_name);

ALTER TABLE

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('person\_id\_seq'::regclass)

first\_name | character varying(50) | | not null |

other\_name | character varying(50) | | |

last\_name | character varying(50) | | not null |

gender | character varying(10) | | not null |

date\_of\_birth | date | | not null |

email | character varying(200) | | |

Indexes:

"person\_pkey" PRIMARY KEY, btree (first\_name, last\_name)

may=# ALTER TABLE person ADD PRIMARY KEY(id);

ERROR: multiple primary keys for table "person" are not allowed

may=# ALTER TABLE person DROP CONSTRAINT person\_pkey;

ALTER TABLE

may=# ALTER TABLE person ADD PRIMARY KEY(id);

ALTER TABLE

Unique Constraint:

may=# SELECT email, count(\*) FROM person GROUP BY email;

email | count

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

carnout2g@ehow.com | 1

| 32

kshuggk@gov.uk | 1

gberrey29@mtv.com | 1

ljanuaryl@ycombinator.com | 1

ncourtmand@imageshack.us | 1

hmcguire2h@ox.ac.uk | 1

eloveland1s@examiner.com | 1

mscampe@tinyurl.com | 1

pbrownbridgeh@dell.com | 1

hbloxsum1f@census.gov | 1

rclawortho@dion.ne.jp | 1

may=# SELECT email, count(\*) FROM person GROUP BY email HAVING COUNT(\*) >1;

email | count

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

| 32

(1 row)

may=# SELECT first\_name, count(\*) FROM person GROUP BY first\_name HAVING COUNT(\*) >1;

first\_name | count

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

(0 rows)

may=# ALTER TABLE person ADD CONSTRAINT unique\_email\_address UNIQUE(email);

ALTER TABLE

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('person\_id\_seq'::regclass)

first\_name | character varying(50) | | not null |

other\_name | character varying(50) | | |

last\_name | character varying(50) | | not null |

gender | character varying(10) | | not null |

date\_of\_birth | date | | not null |

email | character varying(200) | | |

Indexes:

"person\_pkey" PRIMARY KEY, btree (id)

"unique\_email\_address" UNIQUE CONSTRAINT, btree (email)

may=# ALTER TABLE person DROP CONSTRAINT unique\_email\_address;

ALTER TABLE

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('person\_id\_seq'::regclass)

first\_name | character varying(50) | | not null |

other\_name | character varying(50) | | |

last\_name | character varying(50) | | not null |

gender | character varying(10) | | not null |

date\_of\_birth | date | | not null |

email | character varying(200) | | |

Indexes:

"person\_pkey" PRIMARY KEY, btree (id)

may=# ALTER TABLE person ADD UNIQUE(email);

ALTER TABLE

may=# \d person

Table "public.person"

Column | Type | Collation | Nullable | Default

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

id | bigint | | not null | nextval('person\_id\_seq'::regclass)

first\_name | character varying(50) | | not null |

other\_name | character varying(50) | | |

last\_name | character varying(50) | | not null |

gender | character varying(10) | | not null |

date\_of\_birth | date | | not null |

email | character varying(200) | | |

Indexes:

"person\_pkey" PRIMARY KEY, btree (id)

"person\_email\_key" UNIQUE CONSTRAINT, btree (email)

Check Constraint:

may=# SELECT DISTINCT gender FROM person;

gender

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

Male

Polygender

Non-binary

Female

Agender

(5 rows)

may=# ALTER TABLE person ADD CONSTRAINT gender\_constraint CHECK(gender = 'Female OR gender = 'Male');

DELETE:

DELETE FROM person;

DELETE FROM person WHERE id =1;

DELETE FROM person WHERE gender = ‘Female’ AND country\_of\_birth =’England’;

may=# SELECT \* FROM person WHERE id = 3;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

3 | Melonie | Jake | Scopyn | Agender | 2004-08-04 | mscopyn0@cpanel.net

(1 row)

may=# UPDATE person SET other\_name = 'Jake' WHERE id = 3;

UPDATE 1

may=# INSERT INTO person(id, first\_name, other\_name, last\_name, gender, date\_of\_birth, email)

VALUES (3, 'Melonie', 'Jake', 'Scopyn', 'Agender', DATE '1952-09-25', 'mscopyn0@cpanel.net')

;

ERROR: duplicate key value violates unique constraint "person\_pkey"

DETAIL: Key (id)=(3) already exists.

may=# INSERT INTO person(id, first\_name, other\_name, last\_name, gender, date\_of\_birth, email)

VALUES (3, 'Melonie', 'Jake', 'Scopyn', 'Female', DATE '1952-09-25', 'mscopyn0@cpanel.net')

ON CONFLICT (id) DO UPDATE SET gender = EXCLUDED.gender;

INSERT 0 1

may=# SELECT \* FROM person WHERE id = 3;

id | first\_name | other\_name | last\_name | gender | date\_of\_birth | email

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

3 | Melonie | Jake | Scopyn | Female | 2004-08-04 | mscopyn0@cpanel.net

(1 row)

ADDING RELATIONSHIP BETTWEEN TWO TABLES:

may=# \d

List of relations

Schema | Name | Type | Owner

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

public | car | table | postgres

public | car\_id\_seq | sequence | postgres

public | person | table | postgres

public | person\_id\_seq | sequence | postgres

(4 rows)

may=# \dt

List of relations

Schema | Name | Type | Owner

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

public | car | table | postgres

public | person | table | postgres

(2 rows)

**Previous tables:**

create table person (

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(50),

gender VARCHAR(50),

date\_of\_birth DATE,

other\_name VARCHAR(50)

);

**And**

create table car (

id BIGSERIAL NOT NULL PRIMARY KEY,

make VARCHAR(100) NOT NULL,

model VARCHAR(100) NOT NULL,

price NUMERIC(19, 2) NOT NULL

);

**Create another sql file called person&car.sql**

create table car (

id BIGSERIAL NOT NULL PRIMARY KEY,

make VARCHAR(100) NOT NULL,

model VARCHAR(100) NOT NULL,

price NUMERIC(19, 2) NOT NULL

);

create table person (

id BIGSERIAL NOT NULL PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(50),

gender VARCHAR(50),

date\_of\_birth DATE,

car\_id BIGINT REFERENCES car (id),

UNIQUE(car\_id)

);

insert into car (id, make, model, price) values (1, 'Hyundai', 'Elantra', '8118.17');

insert into car (id, make, model, price) values (2, 'Ford', 'F-Series', '3575.86');

insert into car (id, make, model, price) values (3, 'Toyota', 'TundraMax', '2180.36');

insert into car (id, make, model, price) values (4, 'Chevrolet', 'Suburban 1500', '3112.80');

insert into person (id, first\_name, last\_name, email, gender, date\_of\_birth) values (1,'Melonie', 'Scopyn', 'mscopyn0@cpanel.net', 'Agender', '2004/08/04');

insert into person (id, first\_name, last\_name, email, gender, date\_of\_birth) values (2, 'Gabrila', 'Acres', 'gacres1@smh.com.au', 'Female', '2009/10/29');

insert into person (id, first\_name, last\_name, email, gender, date\_of\_birth) values (3, 'Dionisio', 'Eddicott', 'deddicott2@ycombinator.com', 'Male', '2016/03/15');

insert into person (id, first\_name, last\_name, email, gender, date\_of\_birth) values (4, 'Benoite', 'Janu', 'bjanu3@symantec.com', 'Female', '2017/09/01');

insert into person (id, first\_name, last\_name, email, gender, date\_of\_birth) values (5, 'Barrie', 'Marco', 'bar@gmail.com', 'Agender', '2015/07/06');

may=# SELECT \* FROM car;

id | make | model | price

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

1 | Hyundai | Elantra | 8118.17

2 | Ford | F-Series | 3575.86

3 | Toyota | TundraMax | 2180.36

4 | Chevrolet | Suburban 1500 | 3112.80

(4 rows)

may=# SELECT \* FROM person;

id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id

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

1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 |

2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 |

3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 |

4 | Benoite | Janu | bjanu3@symantec.com | Female | 2017-09-01 |

5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 |

(5 rows)

Updating Foreign keys columns:

may=# UPDATE person SET car\_id = 1 WHERE id = 1;

UPDATE 1

may=# SELECT \* FROM person;

id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id

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

2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 |

3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 |

4 | Benoite | Janu | bjanu3@symantec.com | Female | 2017-09-01 |

5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 |

1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 | 1

(5 rows)

may=# UPDATE person SET car\_id = 1 WHERE id = 2;

ERROR: duplicate key value violates unique constraint "person\_car\_id\_key"

DETAIL: Key (car\_id)=(1) already exists.

may=# UPDATE person SET car\_id = 3 WHERE id = 2;

UPDATE 1

may=# UPDATE person SET car\_id = 4 WHERE id = 3;

UPDATE 1

may=# UPDATE person SET car\_id = 2 WHERE id = 5;

UPDATE 1

may=# SELECT \* FROM person;

id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id

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

4 | Benoite | Janu | bjanu3@symantec.com | Female | 2017-09-01 |

1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 | 1

2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 | 3

3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 | 4

5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 | 2

(5 rows)

INNER JOINS

may=# SELECT \* FROM person

may-# JOIN car ON person.car\_id = car.id;

id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id | id | make | model | price

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

1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 | 1 | 1 | Hyundai | Elantra | 8118.17

5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 | 2 | 2 | Ford | F-Series | 3575.86

2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 | 3 | 3 | Toyota | TundraMax | 2180.36

3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 | 4 | 4 | Chevrolet | Suburban 1500 | 3112.80

(4 rows)

may=# \x

Expanded display is on.

may=# SELECT \* FROM person

JOIN car ON person.car\_id = car.id;

-[ RECORD 1 ]-+---------------------------

id | 1

first\_name | Melonie

last\_name | Scopyn

email | mscopyn0@cpanel.net

gender | Agender

date\_of\_birth | 2004-08-04

car\_id | 1

id | 1

make | Hyundai

model | Elantra

price | 8118.17

-[ RECORD 2 ]-+---------------------------

id | 5

first\_name | Barrie

last\_name | Marco

email | bar@gmail.com

gender | Agender

date\_of\_birth | 2015-07-06

car\_id | 2

id | 2

make | Ford

model | F-Series

price | 3575.86

-[ RECORD 3 ]-+---------------------------

id | 2

first\_name | Gabrila

last\_name | Acres

email | gacres1@smh.com.au

gender | Female

date\_of\_birth | 2009-10-29

car\_id | 3

id | 3

make | Toyota

model | TundraMax

price | 2180.36

-[ RECORD 4 ]-+---------------------------

id | 3

first\_name | Dionisio

last\_name | Eddicott

email | deddicott2@ycombinator.com

gender | Male

date\_of\_birth | 2016-03-15

car\_id | 4

id | 4

make | Chevrolet

model | Suburban 1500

price | 3112.80

may=# SELECT \* FROM car

JOIN person ON car.id = person.car\_id;

id | make | model | price | id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id

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

1 | Hyundai | Elantra | 8118.17 | 1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 | 1

2 | Ford | F-Series | 3575.86 | 5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 | 2

3 | Toyota | TundraMax | 2180.36 | 2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 | 3

4 | Chevrolet | Suburban 1500 | 3112.80 | 3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 | 4

(4 rows)

may=# SELECT person.first\_name, person.last\_name, car.make FROM car

JOIN person ON car.id = person.car\_id;

first\_name | last\_name | make

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

Melonie | Scopyn | Hyundai

Barrie | Marco | Ford

Gabrila | Acres | Toyota

Dionisio | Eddicott | Chevrolet

(4 rows)

may=# SELECT person.first\_name, person.last\_name, car.make FROM person

JOIN car ON person.car\_id = car.id;

first\_name | last\_name | make

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

Melonie | Scopyn | Hyundai

Barrie | Marco | Ford

Gabrila | Acres | Toyota

Dionisio | Eddicott | Chevrolet

(4 rows)

LEFT JOINS:

may=# SELECT person.first\_name, person.last\_name, car.make FROM person

LEFT JOIN car ON person.car\_id = car.id;

first\_name | last\_name | make

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

Melonie | Scopyn | Hyundai

Barrie | Marco | Ford

Gabrila | Acres | Toyota

Dionisio | Eddicott | Chevrolet

Benoite | Janu |

(5 rows)

may=# SELECT person.first\_name, person.last\_name, car.make FROM car

LEFT JOIN person ON car.id = person.car\_id;

first\_name | last\_name | make

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

Melonie | Scopyn | Hyundai

Barrie | Marco | Ford

Gabrila | Acres | Toyota

Dionisio | Eddicott | Chevrolet

(4 rows)

may=# DELETE FROM car WHERE id = 4;

ERROR: update or delete on table "car" violates foreign key constraint "person\_car\_id\_fkey" on table "person"

DETAIL: Key (id)=(4) is still referenced from table "person".

**EXPORTING QUERY RESULTS TO CSV:**

LEFT JOIN car ON person.car\_id = car.id) TO '/Users/chukwunonsodavid/Code/PostgreSQL/personlcar.csv' DELIMITER ','CSV HEADER;

COPY 5

may=# \copy (SELECT \* FROM person) TO '/Users/chukwunonsodavid/Code/PostgreSQL/person.csv' DELIMITER ','CSV HEADER;

COPY 5

**SERIAL & SEQUENCES:**

may=# SELECT \* FROM car\_id\_seq;

last\_value | log\_cnt | is\_called

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

4 | 29 | t

(1 row)

may=# SELECT \* FROM person\_id\_seq;

last\_value | log\_cnt | is\_called

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

5 | 28 | t

(1 row)

may=# SELECT nextval('person\_id\_seq'::regclass);

nextval

---------

6

(1 row)

may=# SELECT \* FROM person\_id\_seq;

last\_value | log\_cnt | is\_called

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

6 | 32 | t

(1 row)

may=# SELECT \* FROM person;

id | first\_name | last\_name | email | gender | date\_of\_birth | car\_id

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

1 | Melonie | Scopyn | mscopyn0@cpanel.net | Agender | 2004-08-04 |

2 | Gabrila | Acres | gacres1@smh.com.au | Female | 2009-10-29 |

3 | Dionisio | Eddicott | deddicott2@ycombinator.com | Male | 2016-03-15 |

4 | Benoite | Janu | bjanu3@symantec.com | Female | 2017-09-01 |

5 | Barrie | Marco | bar@gmail.com | Agender | 2015-07-06 |

(5 rows)

may=# ALTER SEQUENCE person\_id\_seq RESTART WITH 6;

ALTER SEQUENCE

CREATE EXTENSION:

may=# SELECT \* FROM pg\_available\_extensions;

may=# CREATE EXTENSION plpython3u