Postgres database

\?
\l → list of database

CREATE DATABASE test; → Don't forget add ;

Connect to database

```
psql -h ec2-52-203-27-62.compute-1.amazonaws.com -p 5432 phanerrlasffas dr34mpeu1lc13
```

\c database name → connect to the database.

DROP DATABASE db_name; → Delete database.

Create a table in postgres.

```
CREATE TABLE table_name(
        Column_name + data type + constraints if any
)
```

This is not a good way to create table.

```
CREATE TABLE person(
id int,
first_name VARCHAR(50),
last_name VARCHAR(50),
date_of_birth TIMESTAMP
);
```

\d → check all the table in database.
\d person

```
CREATE TABLE person(
id BIGSERIAL NOT NULL PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
date_of_birth DATE NOT NULL,
email VARCHAR(60)
```

);

How to insert records into tables.

INSERT INTO person(first_name, last_name, date_of_birth) VALUES ('Amit', 'Singh', DATE '1988-01-09','amitoct9@gmail.com');

SELECT * FROM person;

\i /home/aps/Downloads/sqls/person.sql → To insert sql data from a sql file.

Generate sql commands → https://www.mockaroo.com/

SELECT FROM
SELECT * FROM person; → select everything from table.

SELECT first_name, last_name FROM person;

Order by

SELECT * FROM person ORDER BY country_of_birth;
SELECT * FROM person ORDER BY ID DESC;

DISTINCT

SELECT DISTINCT country_of_birth FROM person ORDER BY country_of_birth DESC;

WHERE clause and AND

SELECT * FROM person WHERE gender = 'Female';

SELECT * FROM person WHERE country_of_birth = 'India' AND first_name = 'Jeffie';

SELECT * FROM person WHERE gender = 'Female' AND (country_of_birth = 'India' OR country_of_birth = 'China');

Comparison operator

SELECT 1 = 1;

SELECT 1 <> 2; → Not equal to.

Limit, Offset and Fetch

```
SELECT * FROM person LIMIT 10;


SELECT * FROM person OFFSET 5 LIMIT 5;

SELECT * FROM person OFFSET 5 FETCH FIRST 5 ROW ONLY;
```

IN

```
SELECT * FROM person WHERE country_of_birth IN ('France', 'China', 'Brazil');
```

Between

```
SELECT * FROM person WHERE date_of_birth BETWEEN DATE '2020-01-01' AND
'2021-01-01';
```

Like and iLike

```
SELECT * FROM person WHERE email like '%.com';
SELECT * FROM person WHERE email like '%@over-blog.com';
SELECT * FROM person WHERE email like '________@%';


SELECT * FROM person WHERE country_of_birth LIKE 'P%';

SELECT * FROM person WHERE country_of_birth ILIKE 'p%';
```

Group By

```
SELECT country_of_birth, COUNT(*) FROM person GROUP BY country_of_birth;
SELECT country_of_birth, COUNT(*) FROM person GROUP BY country_of_birth ORDER BY
country_of_birth;
```

Group By having

```
SELECT country_of_birth, COUNT(*) FROM person GROUP BY country_of_birth HAVING
COUNT(*) > 5 ORDER BY country_of_birth;
```

Calculating min, max and average
SELECT MAX(price) FROM car;

SELECT MIN(price) FROM car;

SELECT AVG(price) FROM car;

SELECT make, model, MIN(price) FROM car GROUP BY make, model;
SELECT make, model, MAX(price) FROM car GROUP BY make, model;

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

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

SELECT make, ROUND(AVG(price)) FROM car GROUP BY make;

SUM
SELECT SUM(price) FROM car;

SELECT make, SUM(price) FROM car GROUP BY make;
SELECT make,model, SUM(price) FROM car GROUP BY make, model ORDER BY make;

Basics of Arithmetic Operator

+ - * ^

Arithmetic Operators (ROUND)

SELECT id, make, model, price, price * 0.10 FROM car; → It will return 10% discount on original price.

SELECT id, make, model, price,ROUND(price * 0.10, 2), ROUND(price - (price * 0.10), 2) FROM car;

Alias

SELECT id, make, model, price AS original_price,ROUND(price * 0.10, 2) AS discount_10_per, ROUND(price - (price * 0.10), 2) AS discounted_price_after_10_percent FROM car;

Coalesce

SELECT COALESCE(null, null, 1) AS number;

SELECT COALESCE(null, null, 1, 10) AS number;

SELECT COALESCE(email, 'Email not provided') FROM person;

NULLIF

SELECT COALESCE(10 / NULLIF(0, 0), 0);

Timestamps and Dates

SELECT NOW();

SELECT NOW()::DATE;

SELECT NOW()::TIME;

Adding And Subtracting With Dates

SELECT NOW() - INTERVAL '1 YEAR';

SELECT NOW() - INTERVAL '10 YEARS';
SELECT NOW() + INTERVAL '10 YEARS';

SELECT NOW()::DATE + INTERVAL '10 YEARS';

SELECT (NOW() + INTERVAL '10 YEARS')::DATE;

Extracting Fields From Timestamp

SELECT EXTRACT(YEAR FROM NOW());

YEAR → MONTH, DAY, DOW

SELECT EXTRACT(CENTURY FROM NOW());

Age Function

```
SELECT first_name, last_name, gender, country_of_birth, date_of_birth, AGE(NOW(),
date_of_birth) AS age FROM person;
```

What Are Primary Keys

PRIMARY KEY(PK) → Uniquely identify a record in tables.

Understanding Primary Keys

```
\d person
```

```
ALTER TABLE person DROP CONSTRAINT person_pkey;
```

```
\d person
```

```
SELECT * FROM person WHERE id = 1;
```

Adding Primary Key

```
DELETE FROM person WHERE id = 1;
```

```
ALTER TABLE person ADD PRIMARY KEY(id);
```

Check Constraints

```
SELECT email, COUNT(*) FROM person GROUP BY email;
```

```
SELECT email, COUNT(*) FROM person GROUP BY email HAVING COUNT(*) > 1;
```

```
SELECT first_name, COUNT(*) FROM person GROUP BY first_name HAVING COUNT(*) > 1;
```

```
SELECT first_name, COUNT(*) FROM person GROUP BY first_name HAVING COUNT(*) >= 2;
```

```
ALTER TABLE person ADD CONSTRAINT unique_email_address UNIQUE(email);
```

ALTER TABLE person DROP CONSTRAINT unique_email_address; → DROP unique CONSTRAINT

ALTER TABLE person DROP CONSTRAINT person_email_key1;

ALTER TABLE person ADD UNIQUE(email);

Check Constraints

SELECT DISTINCT gender FROM person;

DELETE FROM person WHERE gender IN ('Polygender','Non-binary');

ALTER TABLE person ADD CONSTRAINT gender_constraint CHECK (gender = 'Female' OR gender = 'Male');

How to Delete Records

DELETE FROM person WHERE id = 2;

DELETE FROM person WHERE gender = 'Female' AND country_of_birth = 'China';

Update Records

UPDATE person SET email = 'jeri@mail.com' WHERE id = 1033;

UPDATE person SET first_name = 'Aps', last_name = 'Singh', email = 'aps@mail.com' WHERE id = 1033;

On Conflict Do Nothing

```
insert into person (id, first_name, last_name, email, gender, date_of_birth, country_of_birth)
values (1054,'Logan', 'Offer', 'loffer0@imgur.com', 'Male', '2020-07-10', 'Bangladesh');
```

```
insert into person (id, first_name, last_name, email, gender, date_of_birth, country_of_birth)
values (1054,'Logan', 'Offer', 'loffer0@imgur.com', 'Male', '2020-07-10', 'Bangladesh')
dr34mpeu1lc13-> ON CONFLICT (id) DO NOTHING;
```

Upsert

```
insert into person (id, first_name, last_name, email, gender, date_of_birth, country_of_birth)
values (1054,'Logan', 'Offer', 'loffer0@imgur.in', 'Male', '2020-07-10', 'Bangladesh')
ON CONFLICT (id) DO UPDATE SET email = EXCLUDED.email;
```

```
insert into person (id, first_name, last_name, email, gender, date_of_birth, country_of_birth)
values (1054,'Logan', 'SIngh', 'loffer0@imgur.in', 'Male', '2020-07-10', 'Bangladesh')
ON CONFLICT (id) DO UPDATE SET email = EXCLUDED.email, last_name =
EXCLUDED.last_name;
```

Relationship/Foreign Keys and Joins

```
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) NOT NULL,
   last_name VARCHAR(50) NOT NULL,
   gender VARCHAR(7) NOT NULL,
   email VARCHAR(100),
   date_of_birth DATE NOT NULL,
   country_of_birth VARCHAR(50) NOT NULL,
  car_id BIGINT REFERENCES car (id),
  UNIQUE(car_id)
);
```

Updating Foreign Keys Columns

```
SELECT * FROM person;

SELECT * FROM car;

UPDATE person SET car_id = 2 WHERE id =1;
```

Inner Joins

Take common in both tables.

```
SELECT * FROM person
JOIN car ON person.car_id = car.id;
```

```
\x
```

```
SELECT * FROM person
JOIN car ON person.car_id = car.id;
```

Select specific columns in join.

```
SELECT person.first_name, car.make, car.model, car.price
FROM person
JOIN car ON person.car_id = car.id;
```

LEFT Joins

```
SELECT * FROM person
LEFT JOIN car ON person.car_id = car.id;

SELECT * FROM person WHERE car_id IS NULL;
SELECT * FROM person
LEFT JOIN car ON car.id = person.car_id
WHERE car.* IS NULL;
```

Deleting Records With Foreign Keys

```
dr34mpeu1lc13=> insert into car (id, make, model, price) values (13, 'Volkswagen', 'Golf',
'15411.02');
INSERT 0 1
dr34mpeu1lc13=> insert into person (id, first_name, last_name, email, gender, date_of_birth,
country_of_birth) values (9000,'Amit', 'Singh', null, 'Male', '2020-07-10', 'Bangladesh');
INSERT 0 1
dr34mpeu1lc13=> SELECT * FROM person WHERE id = 9000;
  id  | first_name | last_name | gender | email | date_of_birth | country_of
_birth | car_id
------+------------+-----------+--------+-------+---------------+-----------
-------+--------
 9000 | Amit       | Singh     | Male   |       | 2020-07-10    | Bangladesh
       |
(1 row)

dr34mpeu1lc13=> SELECT * FROM car WHERE id = 13;
 id |    make    | model |  price
----+------------+-------+----------
 13 | Volkswagen | Golf  | 15411.02
(1 row)

dr34mpeu1lc13=> UPDATE person SET car_id = 13 WHERE id = 9000;
UPDATE 1
dr34mpeu1lc13=> SELECT * FROM person WHERE id = 9000;
  id  | first_name | last_name | gender | email | date_of_birth | country_of
_birth | car_id
------+------------+-----------+--------+-------+---------------+-----------
-------+--------
 9000 | Amit       | Singh     | Male   |       | 2020-07-10    | Bangladesh
       |     13
(1 row)

dr34mpeu1lc13=> SELECT * FROM person WHERE id = 9000;
  id  | first_name | last_name | gender | email | date_of_birth | country_of_birth | car_id
------+------------+-----------+--------+-------+---------------+------------------+--------
 9000 | Amit       | Singh     | Male   |       | 2020-07-10    | Bangladesh       |     13
(1 row)
```

```
dr34mpeu1lc13=> SELECT * FROM car WHERE id = 13;
 id |    make    | model |  price
----+------------+-------+----------
 13 | Volkswagen | Golf  | 15411.02
(1 row)
```

```
DELETE FROM person WHERE id = 9000;

 DELETE FROM car WHERE id = 13;
```

Exporting Query Results to CSV

```
ELECT * FROM person
LEFT JOIN car ON car.id = person.car_id;
```

```
\copy (SELECT * FROM person LEFT JOIN car ON car.id = person.car_id) TO
'/home/aps/Downloads/sqls/results.csv' DELIMITER ',' CSV HEADER;

\copy (SELECT * FROM person) TO '/home/aps/Downloads/sqls/resultss.csv' DELIMITER ','
CSV HEADER;
```

Serial & Sequences

```
SELECT * FROM person_id_seq;
```

```
\d person
```

```
SELECT nextval('person_id_seq'::regclass);
```

Extensions

```
SELECT * FROM pg_available_extensions;
```