**Step 1 — Installing PostgreSQL**

sudo apt install postgresql postgresql-contrib  
sudo apt install postgresql postgresql-contrib

## Step 2 — Using PostgreSQL Roles and Databases

sudo -i -u postgres  
psql  
\q

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

**If you don't already have PostgreSQL installed**

Add PostgreSQL's third party repository to get the latest PostgreSQL packages

echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb\_release -c -s)-pgdg main" | sudo tee /etc/apt/sources.list.d/pgdg.list

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

sudo apt-get update

## Step 1 — Installing TimescaleDB

sudo add-apt-repository ppa:timescale/timescaledb-ppa  
sudo apt update  
sudo apt install timescaledb-postgresql-12  
sudo -u postgres psql postgres  
\password postgres

## Step 2 — Configuring TimescaleDB

sudo timescaledb-tune

**Step 3** — **Restart PostgreSQL instance**

sudo service postgresql restart  
sudo -u postgres psql postgres  
\password postgres

The script will ask you to confirm changes. These changes are then written to your postgresql.conf and will take effect on the restart.

Now, let’s take a look at some basic operations from the [TimescaleDB tutorial](javascript:;) which can give you an idea of how to work with the new database system.

To create a hypertable, you start with a regular SQL table and then convert it into a hypertable via the function create\_hypertable.

-- Create extension timescaledb

CREATE EXTENSION timescaledb;

Create a regular table

CREATE TABLE conditions (

time TIMESTAMPTZ NOT NULL,

location TEXT NOT NULL,

temperature DOUBLE PRECISION NULL,

humidity DOUBLE PRECISION NULL

);

Convert it to hypertable is simple as:

SELECT create\_hypertable('conditions', 'time');

Inserting data into the hypertable is done via normal SQL commands:

INSERT INTO conditions(time, location, temperature, humidity)

VALUES (NOW(), 'office', 70.0, 50.0);

Selecting data, is old good SQL.

SELECT \* FROM conditions ORDER BY time DESC LIMIT 10;

As we can see below we can do a group by, order by, and functions. In addition, TimescaleDB includes functions for time-series analysis that are not present in vanilla PostgreSQL.

SELECT time\_bucket('15 minutes', time) AS fifteen\_min,

location, COUNT(\*),

MAX(temperature) AS max\_temp,

MAX(humidity) AS max\_hum

FROM conditions

WHERE time > NOW() - interval '3 hours'

GROUP BY fifteen\_min, location

ORDER BY fifteen\_min DESC, max\_temp DESC;

### BASIC COMMANDS :- Connect to your PostgreSQL server psql -h HOSTNAME -p PORT -U USERNAME -W -d DATABASENAME psql -U postgres -h localhost -d tutorial

#### Option 2: Use a service URI

The Service URI begins with postgres://.

psql postgres://[USERNAME]:[PASSWORD]@[HOSTNAME]:[PORT]/[DATABASENAME]?sslmode=require

*# To access your database (e.g., devices\_small)*

psql -U postgres -h localhost -d devices\_small

# (1) unzip the archive

tar -xvzf devices\_small.tar.gz

# (2) import the .sql file to the database

psql -U postgres -d devices\_small < devices.sql

# (3) import data from .csv files to the database

psql -U postgres -d devices\_small -c "\COPY readings FROM devices\_small\_readings.csv CSV"

psql -U postgres -d devices\_small -c "\COPY device\_info FROM devices\_small\_device\_info.csv CSV"

#### Common psql commands

Here is a table of common commands you'll find yourself using a lot:

| **Command** | **Actions** |
| --- | --- |
| \l | List available databases |
| \c dbname | Connect to a new database |
| \dt | List available tables |
| \d tablename | Describe the details of given table |
| \dn | List all schemas in the current database |
| \df | List functions in the current database |
| \h | Get help on syntax of SQL commands |
| \? | Lists all psql slash commands |
| \set | System variables list |
| \timing | Shows how long a query took to execute |
| \x | Show expanded query results |
| \q | Quit psql |

#### SELECT \* FROM rides LIMIT 5; How many rides took place on each day? SELECT date\_trunc('day', pickup\_datetime) as day, COUNT(\*) FROM rides GROUP BY day ORDER BY day;

What is the average fare amount for passengers?  
SELECT date\_trunc('day', pickup\_datetime)

AS day, avg(fare\_amount)

FROM rides

WHERE passenger\_count = 1

AND pickup\_datetime < '2016-01-08'

#### GROUP BY day ORDER BY day;

How many rides took place for each rate type?  
-- How many rides of each rate type took place in the month?

SELECT rate\_code, COUNT(vendor\_id) AS num\_trips

FROM rides

WHERE pickup\_datetime < '2016-02-01'

GROUP BY rate\_code  
ORDER BY rate\_code;

How many rides of each rate type took place?

-- Join rides with rates to get more information on rate\_code

SELECT rates.description, COUNT(vendor\_id) AS num\_trips FROM rides

JOIN rates ON rides.rate\_code = rates.rate\_code

WHERE pickup\_datetime < '2016-02-01'

GROUP BY rates.description  
 ORDER BY rates.description;

#### 