

✓ SETUP (RUN THIS SECTION BUT DON'T EDIT)

#The output of the installation is not displayed when %%capture is used at the start of the cell

%%capture

Install postgresql server

!sudo apt-get -y -qq update

!sudo apt-get -y -qq install postgresql

!sudo service postgresql start

Setup a password `postgres` for username `postgres`

!sudo -u postgres psql -U postgres -c "ALTER USER postgres PASSWORD 'postgres';"

Setup a database with name `sampledb` to be used

!sudo -u postgres psql -U postgres -c 'DROP DATABASE IF EXISTS sampledb;'

!sudo -u postgres psql -U postgres -c 'CREATE DATABASE sampledb;'



[Show hidden output](#)

✓ Set Colab Database Variable

set connection

Now sampledb is the default database we are going to use

%env DATABASE_URL=postgresql://postgres:postgres@localhost:5432/sampledb



env: DATABASE_URL=postgresql://postgres:postgres@localhost:5432/sampledb

#To load the sql extention to start using %%sql

%load_ext sql

%config SqlMagic.style = '_DEPRECATED_DEFAULT'



The sql extension is already loaded. To reload it, use:

%reload_ext sql

✓ HOMEWORK QUESTION

%%sql


```
CREATE TABLE Orders (  
    Order_ID INT PRIMARY KEY,  
    Customer_ID INT,  
    Order_Date DATE,  
    Order_Amount FLOAT);
```

```
INSERT INTO Orders
```

```
VALUES
```

```
(1, 101, '2024-10-30', 800),  
(2, 102, '2024-10-30', 200),  
(3, 103, '2024-10-31', 300),  
(4, 104, '2024-10-31', 450),  
(5, 105, '2024-10-31', 100),  
(6, 106, '2024-10-31', 500),  
(7, 107, '2024-10-31', 600),  
(8, 108, '2024-10-31', 700),  
(9, 109, '2024-10-31', 800),  
(10, 110, '2024-10-31', 900),  
(11, 111, '2024-10-31', 1000),  
(12, 112, '2024-11-01', 1000),  
(13, 113, '2024-11-02', 850.50),  
(14, 114, '2024-11-03', 1200),  
(15, 115, '2024-11-03', 400),  
(16, 116, '2024-11-04', 300),  
(17, 117, '2024-11-04', 550),  
(18, 118, '2024-11-05', 700),  
(19, 119, '2024-11-05', 950),  
(20, 120, '2024-11-06', 220),  
(21, 121, '2024-11-06', 600),  
(22, 122, '2024-11-07', 1300),  
(23, 123, '2024-11-07', 150),  
(24, 124, '2024-11-08', 800),  
(25, 125, '2024-11-08', 450),  
(26, 126, '2024-11-09', 500),  
(27, 127, '2024-11-09', 1100),  
(28, 128, '2024-11-10', 250),  
(29, 129, '2024-11-10', 950.75),  
(30, 130, '2024-11-11', 300),  
(31, 131, '2024-11-11', 120),  
(32, 132, '2024-11-12', 700),  
(33, 133, '2024-11-12', 330),  
(34, 134, '2024-11-13', 900),  
(35, 135, '2024-11-13', 220),  
(36, 136, '2024-11-14', 1500),  
(37, 137, '2024-11-14', 750),  
(38, 138, '2024-11-15', 980),
```


```
(39, 139, '2024-11-15', 400),
(40, 140, '2024-11-16', 150);
```

 Done.
40 rows affected.
[]

Start coding or [generate](#) with AI.

%%sql

```
-- Write a query using the Orders table that shows the orders which
--   Were placed on Oct 31, 2024 AND Exceed $250...
-- Write Query below
SELECT *
FROM Orders
WHERE Order_Date = '2024-10-31'
AND Order_Amount > 250;
```

 * postgresql://postgres:***@localhost:5432/sampled
8 rows affected.

order_id	customer_id	order_date	order_amount
3	103	2024-10-31	300.0
4	104	2024-10-31	450.0
6	106	2024-10-31	500.0
7	107	2024-10-31	600.0
8	108	2024-10-31	700.0
9	109	2024-10-31	800.0
10	110	2024-10-31	900.0
11	111	2024-10-31	1000.0

%%sql

```
-- Write a query using the Orders table to calculate the maximum order amount for each day.
-- The query should return the customer who placed the highest order
-- and the corresponding order amount for that day.
SELECT Order_Date, MAX(Order_Amount) AS Max_Order_Amount
FROM Orders
GROUP BY Order_Date;
```



```
* postgresql://postgres:***@localhost:5432/sampledb
```

18 rows affected.

order_date max_order_amount

2024-11-11	300.0
2024-10-31	1000.0
2024-11-02	850.5
2024-11-15	980.0
2024-11-12	700.0
2024-11-08	800.0
2024-11-16	150.0
2024-11-01	1000.0
2024-11-05	950.0
2024-11-03	1200.0
2024-11-14	1500.0
2024-11-10	950.75
2024-11-06	600.0
2024-11-04	550.0
2024-11-13	900.0
2024-11-09	1100.0
2024-10-30	800.0
2024-11-07	1300.0