In [1]:

```
%load_ext sql
```

In [2]:

```
Password_DB = [redacted]
```

In [3]:

```
from urllib.parse import quote
password = quote([redacted])
connection_url = f"postgresql://postgres:{password}@localhost:5432/cohort_
```

In [4]:

```
%sql $connection_url
```

1. Problem Statement: Assume there are two tables, orders and order_details . Write an SQL query to retrieve all orders along with their corresponding order details where the total price of the order is greater than 100.

In [6]:

```
%%sql

CREATE TABLE orders1 (
order_id INT PRIMARY KEY,
customer_id INT,
order_date DATE
);

CREATE TABLE order_details (
detail_id INT PRIMARY KEY,
order_id INT,
product_id INT,
quantity INT,
price DECIMAL(10, 2)
);

```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
Done.
```

Out[6]:

```
[]
```

In [7]:

```
%%sql

INSERT INTO orders1 (order_id, customer_id, order_date) VALUES
(1, 1, '2021-01-01'),
(2, 2, '2021-01-02'),
(3, 1, '2021-01-03'),
(4, 3, '2021-01-04');

INSERT INTO order_details (detail_id, order_id, product_id, quantity, pric
(1, 1, 1, 2, 50),
(2, 1, 2, 1, 30),
(3, 2, 3, 3, 20),
(4, 3, 1, 3, 40);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
4 rows affected.
```

Out[7]:

```
[]
```

In [22]:

```
%%sql

INSERT INTO order_details (detail_id, order_id, product_id, quantity, pric
(5, 4, 2, 2, 25),
(6, 4, 3, 1, 15);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
2 rows affected.
```

Out[22]:

```
[]
```

In [23]:

```
%%sql

SELECT * FROM orders1
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[23]:

order_id	customer_id	order_date
1	1	2021-01-01
2	2	2021-01-02
3	1	2021-01-03
4	3	2021-01-04

In [24]:

```
%%sql

SELECT * FROM order_details
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[24]:

detail_id	order_id	product_id	quantity	price
1	1	1	2	50.00
2	1	2	1	30.00
3	2	3	3	20.00
4	3	1	3	40.00
5	4	2	2	25.00
6	4	3	1	15.00

In [28]:

```
%%sql

SELECT * from order_details WHERE price*quantity > 100
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
1 rows affected.
```

Out[28]:

detail_id	order_id	product_id	quantity	price
4	3	1	3	40.00

In [7]:

```
%%sql

SELECT * FROM orders1 as o
JOIN order_details as d
ON o.order_id = d.order_id

WHERE price*quantity > 100

```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
1 rows affected.
```

Out[7]:

order_id	customer_id	order_date	detail_id	order_id_1	product_id	quantity	price
3	1	2021-01-03	4	3	1	3	40.00

2. Problem Statement: Assume there are two tables, employees and departments . Write an SQL query to retrieve all employees along with their corresponding department name.

In [8]:

```
%%sql


CREATE TABLE departments (
dept_id INT PRIMARY KEY,
dept_name VARCHAR(50)
);

CREATE TABLE employees (
emp_id INT PRIMARY KEY,
emp_name VARCHAR(50),
dept_id INT,
hire_date DATE,
salary DECIMAL(10, 2)
);

```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
Done.
```

Out[8]:

```
[]
```

In [9]:

```
%%sql

INSERT INTO departments (dept_id, dept_name) VALUES
(1, 'IT'),
(2, 'Finance'),
(3, 'HR');

INSERT INTO employees (emp_id, emp_name, dept_id, hire_date, salary) VALUE
(1, 'John', 1, '2020-01-01', 50000),
(2, 'Jane', 2, '2020-02-01', 60000),
(3, 'Mark', 1, '2020-03-01', 55000),
(4, 'Mike', 3, '2020-04-01', 65000);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
4 rows affected.
```

Out[9]:

```
[]
```

In [10]:

```
%%sql

SELECT * FROM departments
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[10]:

dept_id	dept_name
1	IT
2	Finance
3	HR

In [11]:

```
%%sql

SELECT * FROM employees
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[11]:

emp_id	emp_name	dept_id	hire_date	salary
1	John	1	2020-01-01	50000.00
2	Jane	2	2020-02-01	60000.00
3	Mark	1	2020-03-01	55000.00
4	Mike	3	2020-04-01	65000.00

In [12]:

```
%%sql

SELECT emp_name FROM employees
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[12]:

emp_name
John
Jane
Mark
Mike

In [15]:

```
%%sql

SELECT e.emp_name, d.dept_name FROM employees as e

JOIN departmentS as d
ON e.dept_id = d.dept_id



```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[15]:

emp_name	dept_name
John	IT
Jane	Finance
Mark	IT
Mike	HR

3. Problem Statement: Assume there is a table sales with columns product_id , sale_date , and

amount . Write an SQL query to retrieve the total sales amount for each product for the month of January 2021.

In [16]:

```
%%sql

CREATE TABLE sales (
sale_id INT PRIMARY KEY,
product_id INT,
sale_date DATE,
amount DECIMAL(10, 2)
);

```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[16]: []

In [17]:

```
%%sql

INSERT INTO sales (sale_id, product_id, sale_date, amount) VALUES
(1, 1, '2021-01-01', 100),
(2, 2, '2021-01-02', 200),
(3, 1, '2021-01-03', 150),
(4, 3, '2021-01-04', 300),
(5, 2, '2021-02-01', 250),
(6, 3, '2021-02-02', 350);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[17]: []

In [18]:

```
%%sql

SELECT * FROM sales
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[18]:

sale_id	product_id	sale_date	amount
1	1	2021-01-01	100.00
2	2	2021-01-02	200.00
3	1	2021-01-03	150.00
4	3	2021-01-04	300.00
5	2	2021-02-01	250.00
6	3	2021-02-02	350.00

In [32]:

```
%%sql

SELECT sum(amount) as total_sales_amount from sales WHERE EXTRACT(MONTH FR
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
1 rows affected.
```

Out[32]:

total_sales_amount
750.00

4. Problem Statement: Assume there is a table logins with columns user_id and login_time . Write an SQL query to retrieve the number of logins for each user for the month of January 2021.

In [33]:

```
%%sql

CREATE TABLE logins (
login_id INT PRIMARY KEY,
user_id INT,
login_time TIMESTAMP
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[33]: []

In [35]:

```
%%sql

INSERT INTO logins (login_id, user_id, login_time) VALUES
(1, 1, '2021-01-01 12:00:00'),
(2, 2, '2021-01-01 13:00:00'),
(3, 1, '2021-01-02 10:00:00'),
(4, 3, '2021-01-02 11:00:00'),
(5, 2, '2021-02-01 12:00:00'),
(6, 3, '2021-02-01 13:00:00');
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[35]: []

In [36]:

```
%%sql

SELECT * FROM logins
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[36]:

login_id	user_id	login_time
1	1	2021-01-01 12:00:00
2	2	2021-01-01 13:00:00
3	1	2021-01-02 10:00:00
4	3	2021-01-02 11:00:00
5	2	2021-02-01 12:00:00
6	3	2021-02-01 13:00:00

In [343]:

```
%%sql

SELECT user_id, count(*)from logins WHERE EXTRACT(MONTH FROM login_time)=
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[343]:

user_id	count
1	2
2	1
3	1

5. Assume there are two tables, customers and orders . Write an SQL query to retrieve all customers who have placed at least one order.

In [73]:

```
%%sql

CREATE TABLE orders2 (
order_id INT PRIMARY KEY,
customer_id INT,
order_date DATE
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[73]: []

In [77]:

```
%%sql

CREATE TABLE customers (
customer_id INT PRIMARY KEY,
customer_name VARCHAR(50),
address VARCHAR(100)
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[77]:

```
[]
```

In [78]:

```
%%sql

INSERT INTO customers (customer_id, customer_name, address) VALUES
(1, 'John', '123 Main St'),
(2, 'Jane', '456 Oak Ave'),
(3, 'Mark', '789 Elm St');

INSERT INTO orders2 (order_id, customer_id, order_date) VALUES
(1, 1, '2021-01-01'),
(2, 2, '2021-01-02'),
(3, 1, '2021-01-03'),
(4, 3, '2021-01-04');
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
4 rows affected.
```

Out[78]:

```
[]
```

In [79]:

```
%%sql

SELECT * FROM customers
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[79]:

customer_id	customer_name	address
1	John	123 Main St
2	Jane	456 Oak Ave
3	Mark	789 Elm St

In [80]:

```
%%sql

SELECT * FROM orders2
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[80]:

order_id	customer_id	order_date
1	1	2021-01-01
2	2	2021-01-02
3	1	2021-01-03
4	3	2021-01-04

In [82]:

```
%%sql

SELECT * from orders WHERE customer_id >= 1;
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[82]:

order_id	customer_id	order_date	order_amount
1	1	2021-01-01	50.00
2	1	2021-02-01	75.00
3	2	2021-02-15	125.00
4	3	2021-03-01	200.00

In [83]:

```
%%sql

SELECT customer_id from orders WHERE customer_id >= 1;
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[83]:

customer_id
1
1
2
3

In [85]:

```
%%sql

SELECT customer_name from customers WHERE customer_id IN

(SELECT customer_id from orders WHERE customer_id >= 1)
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[85]:

customer_name
John
Jane
Mark

6. Assume there is a table transactions with columns transaction_id , user_id , and amount . Write an SQL query to retrieve the average transaction amount for each user.

In [86]:

```
%%sql

CREATE TABLE transactions (
transaction_id INT PRIMARY KEY,
user_id INT,
amount DECIMAL(10, 2)
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[86]: []

In [87]:

```
%%sql

INSERT INTO transactions (transaction_id, user_id, amount) VALUES
(1, 1, 50),
(2, 2, 100),
(3, 1, 75),
(4, 3, 200),
(5, 2, 125),
(6, 3, 150);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[87]: []

In [88]:

```
%%sql

SELECT * FROM transactions
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[88]:

transaction_id	user_id	amount
1	1	50.00
2	2	100.00
3	1	75.00
4	3	200.00
5	2	125.00
6	3	150.00

In [105]:

```
%%sql

SELECT user_id, count(*), SUM(amount) as Total_Number_amount from transact
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[105]:

user_id	count	total_number_amount
3	2	350.00
2	2	225.00
1	2	125.00

In [345]:

```
%%sql

SELECT user_id, round(AVG(amount),1) as average_transaction_amount from tr
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[345]:

user_id	average_transaction_amount
3	175.0
2	112.5
1	62.5

7 Assume there is a table products with columns product_id and price . Write an SQL query to retrieve the top 3 most expensive products.

In [107]:

```
%%sql

CREATE TABLE products1 (
product_id INT PRIMARY KEY,
price DECIMAL(10, 2)
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[107]:

```
[]
```

In [108]:

```
%%sql

INSERT INTO products1 (product_id, price) VALUES
(1, 100),
(2, 200),
(3, 150),
(4, 300),
(5, 250);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
5 rows affected.
```

Out[108]:

```
[]
```

In [110]:

```
%%sql

SELECT * FROM products1
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
5 rows affected.
```

Out[110]:

product_id	price
1	100.00
2	200.00
3	150.00
4	300.00
5	250.00

In [114]:

```
%%sql

SELECT product_id, price from products1 ORDER BY price DESC LIMIT 3
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[114]:

product_id	price
4	300.00
5	250.00
2	200.00

8. Assume there are two tables, students and grades . Write an SQL query to retrieve the average grade for each student.

In [115]:

```
%%sql

CREATE TABLE students (
student_id INT PRIMARY KEY,
student_name VARCHAR(50),
address VARCHAR(100)
);

CREATE TABLE grades (
grade_id INT PRIMARY KEY,student_id INT,
course_name VARCHAR(50),
grade DECIMAL(10, 2)
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
Done.
```

Out[115]: []

In [116]:

```
%%sql

INSERT INTO students (student_id, student_name, address) VALUES
(1, 'John', '123 Main St'),
(2, 'Jane', '456 Oak Ave'),
(3, 'Mark', '789 Elm St');

INSERT INTO grades (grade_id, student_id, course_name, grade) VALUES
(1, 1, 'Math', 90),
(2, 2, 'Math', 95),
(3, 1, 'Science', 80),
(4, 3, 'Math', 85),
(5, 2, 'Science', 92),
(6, 3, 'Science', 88);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
6 rows affected.
```

Out[116]: []

In [117]:

```
%%sql

SELECT * FROM students
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[117]:

student_id	student_name	address
1	John	123 Main St
2	Jane	456 Oak Ave
3	Mark	789 Elm St

In [118]:

```
%%sql

SELECT * FROM grades
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[118]:

grade_id	student_id	course_name	grade
1	1	Math	90.00
2	2	Math	95.00
3	1	Science	80.00
4	3	Math	85.00
5	2	Science	92.00
6	3	Science	88.00

In [281]:

```
%%sql

SELECT s.student_name, round(AVG(g.grade),1) as average_grade from student
JOIN grades g ON s.student_id = g.student_id
GROUP BY student_name

```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[281]:

student_name	average_grade
Jane	93.5
Mark	86.5
John	85.0

9. Assume there are two tables, employees and salaries . Write an SQL query to retrieve all

In [206]:

```
%%sql

CREATE TABLE employees2 (
emp_id INT PRIMARY KEY,
emp_name VARCHAR(50),
hire_date DATE
);

CREATE TABLE salaries2 (
salary_id INT PRIMARY KEY,
emp_id INT,
salary DECIMAL(10, 2),
start_date DATE,
end_date DATE
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
Done.
```

Out[206]: []

In [207]:

```
%%sql

INSERT INTO employees2 (emp_id, emp_name, hire_date) VALUES
(1, 'John', '2020-01-01'),
(2, 'Jane', '2020-02-01'),
(3, 'Mark', '2020-03-01'),
(4, 'Mike', '2020-04-01');

INSERT INTO salaries2 (salary_id, emp_id, salary, start_date, end_date) VA
(1, 1, 50000, '2020-01-01', '2020-12-31'),
(2, 2, 60000, '2020-01-01', '2020-12-31'),
(3, 1, 55000, '2021-01-01', '2021-12-31'),
(4, 3, 65000, '2021-01-01', '2021-12-31'),
(5, 2, 70000, '2021-01-01', '2021-12-31'),
(6, 4, 75000, '2021-01-01', '2021-12-31');
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
6 rows affected.
```

Out[207]: []

In [209]:

```
%%sql
SELECT * FROM employees2
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
4 rows affected.
```

Out[209]:

emp_id	emp_name	hire_date
1	John	2020-01-01
2	Jane	2020-02-01
3	Mark	2020-03-01
4	Mike	2020-04-01

In [211]:

```
%%sql

SELECT * FROM salaries2
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[211]:

salary_id	emp_id	salary	start_date	end_date
1	1	50000.00	2020-01-01	2020-12-31
2	2	60000.00	2020-01-01	2020-12-31
3	1	55000.00	2021-01-01	2021-12-31
4	3	65000.00	2021-01-01	2021-12-31
5	2	70000.00	2021-01-01	2021-12-31
6	4	75000.00	2021-01-01	2021-12-31

In [212]:

```
%%sql

SELECT salary from salaries2
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[212]:

salary
50000.00
60000.00
55000.00
65000.00
70000.00
75000.00

In [221]:

```
%%sql
SELECT * from employees2 as e
JOIN salaries2 as s
ON e.emp_id =  s.emp_id
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[221]:

emp_id	emp_name	hire_date	salary_id	emp_id_1	salary	start_date	end_date
1	John	2020-01-01	1	1	50000.00	2020-01-01	2020-12-31
2	Jane	2020-02-01	2	2	60000.00	2020-01-01	2020-12-31
1	John	2020-01-01	3	1	55000.00	2021-01-01	2021-12-31
3	Mark	2020-03-01	4	3	65000.00	2021-01-01	2021-12-31
2	Jane	2020-02-01	5	2	70000.00	2021-01-01	2021-12-31
4	Mike	2020-04-01	6	4	75000.00	2021-01-01	2021-12-31

In [222]:

```
%%sql


SELECT e.emp_name, s.salary from employees2 as e
JOIN salaries2 as s
ON e.emp_id =  s.emp_id
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[222]:

emp_name	salary
John	50000.00
Jane	60000.00
John	55000.00
Mark	65000.00
Jane	70000.00
Mike	75000.00

10. Assume there is a table orders with columns order_id , order_date , and total_price . Write an SQL query to retrieve the total sales for each month.

In [241]:

```
%%sql

CREATE TABLE orders4 (
order_id INT PRIMARY KEY,
order_date DATE,
total_price DECIMAL(10, 2)
);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
Done.
```

Out[241]:

```
[]
```

In [242]:

```
%%sql

INSERT INTO orders4 (order_id, order_date, total_price) VALUES
(1, '2021-01-01', 100),
(2, '2021-01-02', 200),
(3, '2021-02-01', 150),
(4, '2021-02-02', 300),
(5, '2021-03-01', 250),
(6, '2021-03-02', 350);
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[242]:

```
[]
```

In [243]:

```
%%sql

SELECT * FROM orders4
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
6 rows affected.
```

Out[243]:

order_id	order_date	total_price
1	2021-01-01	100.00
2	2021-01-02	200.00
3	2021-02-01	150.00
4	2021-02-02	300.00
5	2021-03-01	250.00
6	2021-03-02	350.00

In [293]:

```
%%sql

SELECT EXTRACT(MONTH from order_date) as months, SUM(total_price) as total
```

```
 * postgresql://postgres:***@localhost:5432/cohort_4
3 rows affected.
```

Out[293]:

months	total_sales
1	300.00
2	450.00
3	600.00

In [ ]: