SQL for Data Analysis

```sql
-- Table creation of departments --
CREATE TABLE departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL
);

-- Sample Data
INSERT INTO departments (dept_id, dept_name) VALUES
(101, 'Data Science'),
(102, 'Marketing'),
(103, 'HR');

-- Table creation of employees --
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    dept_id INT,
    salary DECIMAL(10, 2),
    join_date DATE,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- Sample Data
INSERT INTO employees (emp_id, name, dept_id, salary, join_date) VALUES
(1, 'Alice', 101, 60000, '2021-01-15'),
(2, 'Bob', 102, 55000, '2020-03-20'),
(3, 'Charlie', 101, 70000, '2019-05-12'),
(4, 'David', 103, 48000, '2022-11-01'),
(5, 'Eve', NULL, 52000, '2023-06-25');
-- we can add some more data i only added 5 people data to understand easily--
```

SQL for Data Analysis

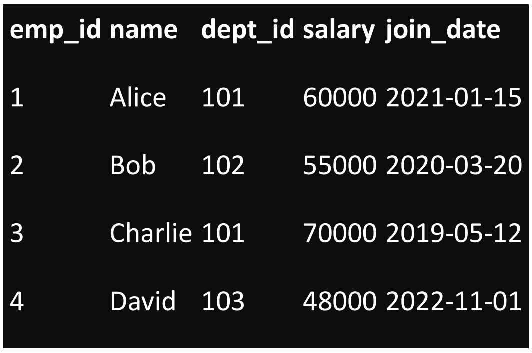
```sql
--using order by  & group by --
SELECT name, salary
FROM employees
ORDER BY salary DESC
LIMIT 3;

SELECT d.dept_name, AVG(e.salary) AS avg_salary
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id
GROUP BY d.dept_name;

-- INNER JOIN to get all employees with their department names
SELECT e.name, d.dept_name
FROM employees e
INNER JOIN departments d ON e.dept_id = d.dept_id;

-- LEFT JOIN to find departments without employees
SELECT d.dept_name, e.name
FROM departments d
LEFT JOIN employees e ON d.dept_id = e.dept_id
WHERE e.name IS NULL;
-- there is no any department without employee so we get nothing--

--Subquery--
-- Find employees earning above average salary
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

SQL for Data Analysis

```
--Aggregate Functions (SUM, AVG)--

-- Total salary budget

SELECT SUM(salary) AS total_salary_budget

FROM employees;

SELECT AVG(salary) AS total_salary_budget

FROM employees;


-- View for department-wise salary stats

CREATE VIEW dept_salary_stats AS

SELECT d.dept_name, COUNT(e.emp_id) AS num_employees, AVG(e.salary) AS avg_salary

FROM employees e

JOIN departments d ON e.dept_id = d.dept_id

GROUP BY d.dept_name;


-- Create index on dept_id to speed up joins

CREATE INDEX idx_dept_id ON employees(dept_id);
```

OUTPUT'S:-

employees

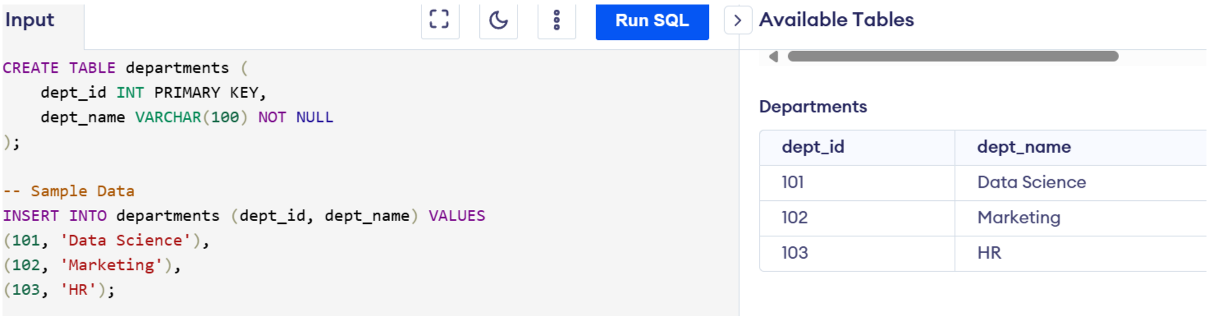


emp_id	name	dept_id	salary	join_date
1	Alice	101	60000	2021-01-15
2	Bob	102	55000	2020-03-20
3	Charlie	101	70000	2019-05-12
4	David	103	48000	2022-11-01

SQL for Data Analysis

departments

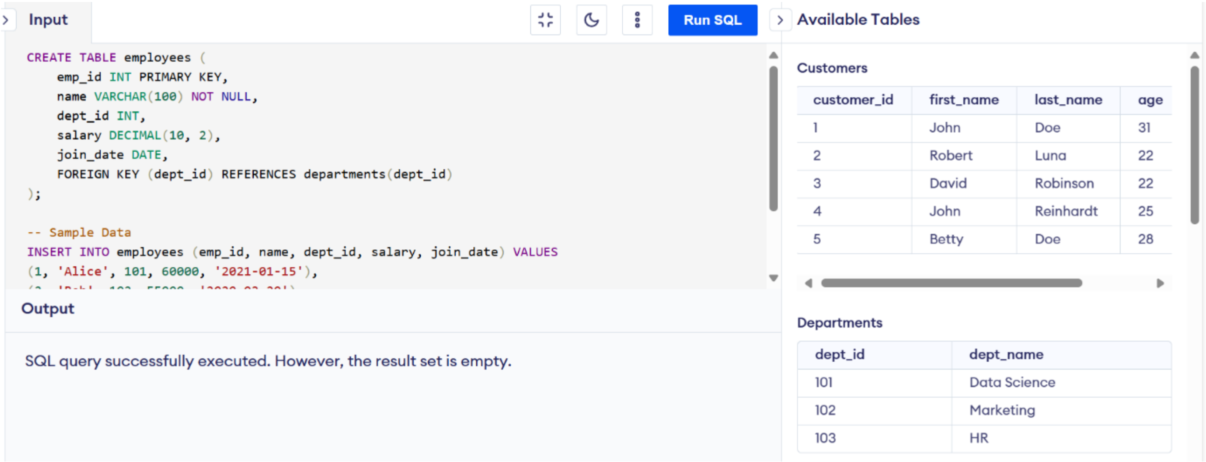
dept_id	dept_name
101	Data Science
102	Marketing
103	HR



```
CREATE TABLE departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(100) NOT NULL
);

-- Sample Data
INSERT INTO departments (dept_id, dept_name) VALUES
(101, 'Data Science'),
(102, 'Marketing'),
(103, 'HR');
```

Departments

dept_id	dept_name
101	Data Science
102	Marketing
103	HR



```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    dept_id INT,
    salary DECIMAL(10, 2),
    join_date DATE,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- Sample Data
INSERT INTO employees (emp_id, name, dept_id, salary, join_date) VALUES
(1, 'Alice', 101, 60000, '2021-01-15'),
```

SQL query successfully executed. However, the result set is empty.

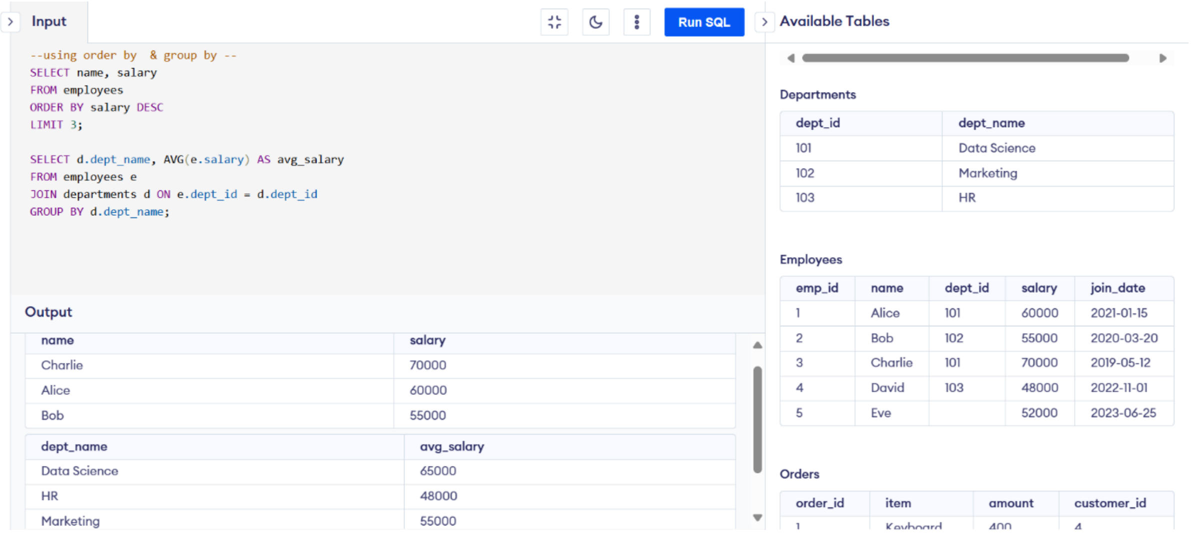
Customers

customer_id	first_name	last_name	age
1	John	Doe	31
2	Robert	Luna	22
3	David	Robinson	22
4	John	Reinhardt	25
5	Betty	Doe	28

Departments

dept_id	dept_name
101	Data Science
102	Marketing
103	HR

SQL for Data Analysis



```sql
--using order by  & group by --
SELECT name, salary
FROM employees
ORDER BY salary DESC
LIMIT 3;

SELECT d.dept_name, AVG(e.salary) AS avg_salary
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id
GROUP BY d.dept_name;
```

name	salary
Charlie	70000
Alice	60000
Bob	55000

dept_name	avg_salary
Data Science	65000
HR	48000
Marketing	55000

Departments

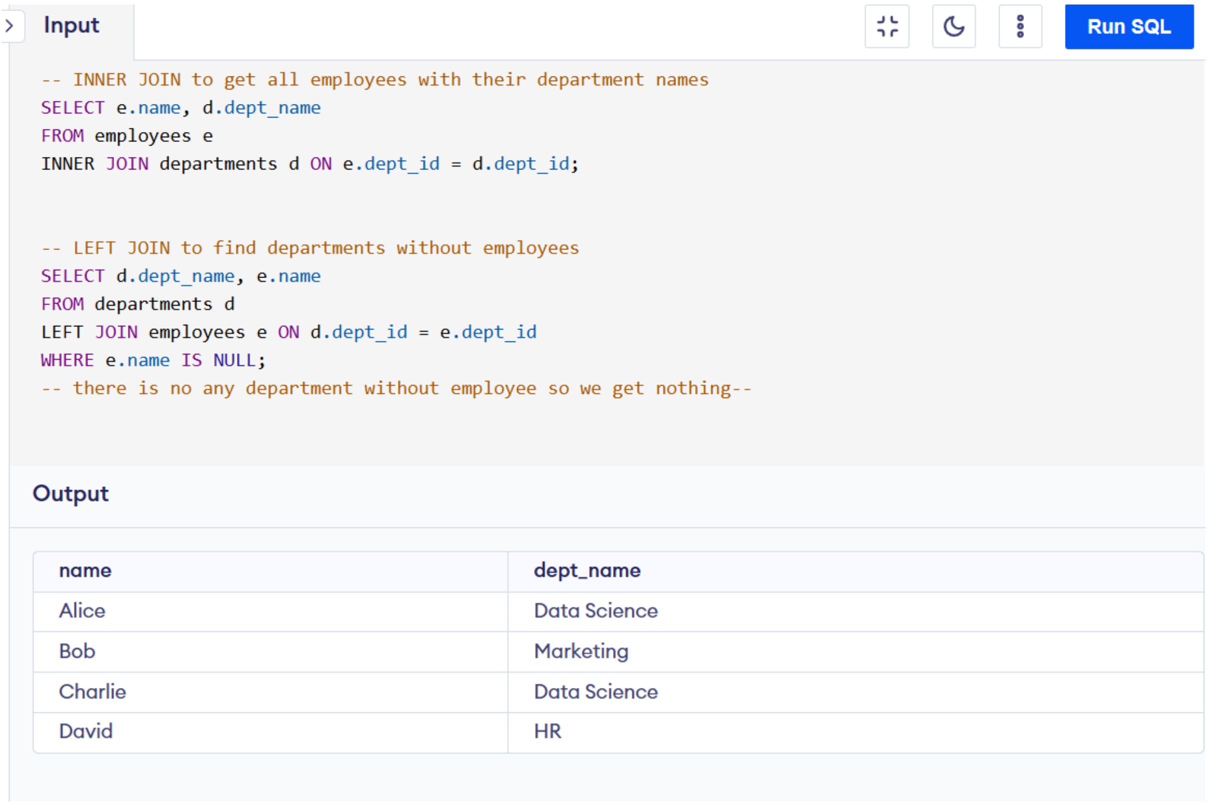
dept_id	dept_name
101	Data Science
102	Marketing
103	HR

Employees

emp_id	name	dept_id	salary	join_date
1	Alice	101	60000	2021-01-15
2	Bob	102	55000	2020-03-20
3	Charlie	101	70000	2019-05-12
4	David	103	48000	2022-11-01
5	Eve		52000	2023-06-25

Orders

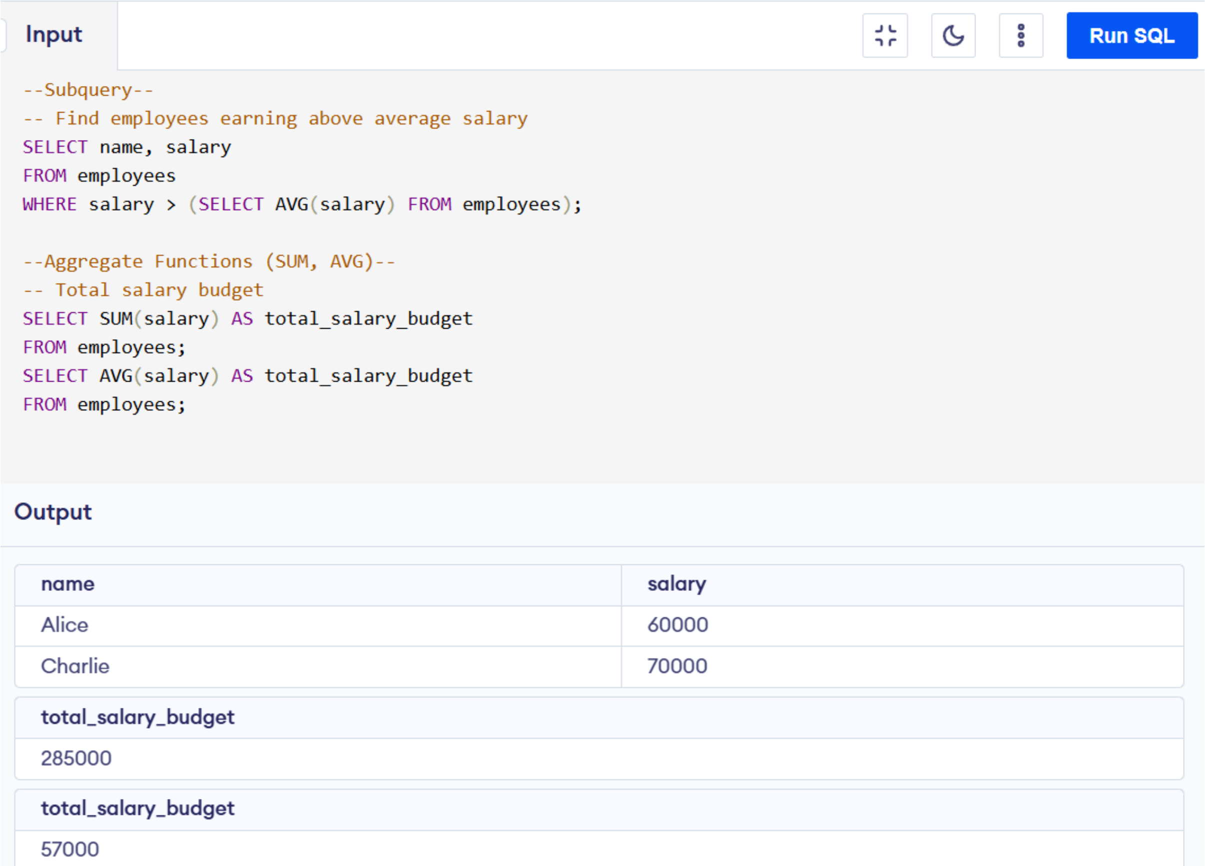
order_id	item	amount	customer_id
1	Keyboard	400	4



```sql
-- INNER JOIN to get all employees with their department names
SELECT e.name, d.dept_name
FROM employees e
INNER JOIN departments d ON e.dept_id = d.dept_id;


-- LEFT JOIN to find departments without employees
SELECT d.dept_name, e.name
FROM departments d
LEFT JOIN employees e ON d.dept_id = e.dept_id
WHERE e.name IS NULL;
-- there is no any department without employee so we get nothing--
```

name	dept_name
Alice	Data Science
Bob	Marketing
Charlie	Data Science
David	HR

SQL for Data Analysis



```
--Subquery--
-- Find employees earning above average salary
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);

--Aggregate Functions (SUM, AVG)--
-- Total salary budget
SELECT SUM(salary) AS total_salary_budget
FROM employees;
SELECT AVG(salary) AS total_salary_budget
FROM employees;
```

name	salary
Alice	60000
Charlie	70000

total_salary_budget
285000

total_salary_budget
57000

If u have any queries feel free to contact me :- babipepakayala162129@gmail.com