



ASSIGNMENT TITLE

Filtering and Sorting

Submitted By : Sachin Yadav

Course : Data Analytics With AI – September batch Live

Institute : PW Skills

Date : 10 Dec. 25

Question 1: Show employees working in either the 'IT' or 'HR' departments.

SQL Query:

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query', 'Query History', and 'Scratch Pad'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT *
2 FROM Employees
3 WHERE Department IN ('IT', 'HR');
4
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns and data types:

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

The `IN` operator checks whether the department is either IT or HR. It returns all employees belonging to these two departments.

Question 2: Retrieve employees whose department is in ‘Sales’, ‘IT’, or ‘Finance’.

SQL Query:

Query

Query History

1

2

3

4

SELECT *

FROM Employees

WHERE Department IN ('Sales', 'IT', 'Finance');

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

📦

📥

📈

SQL

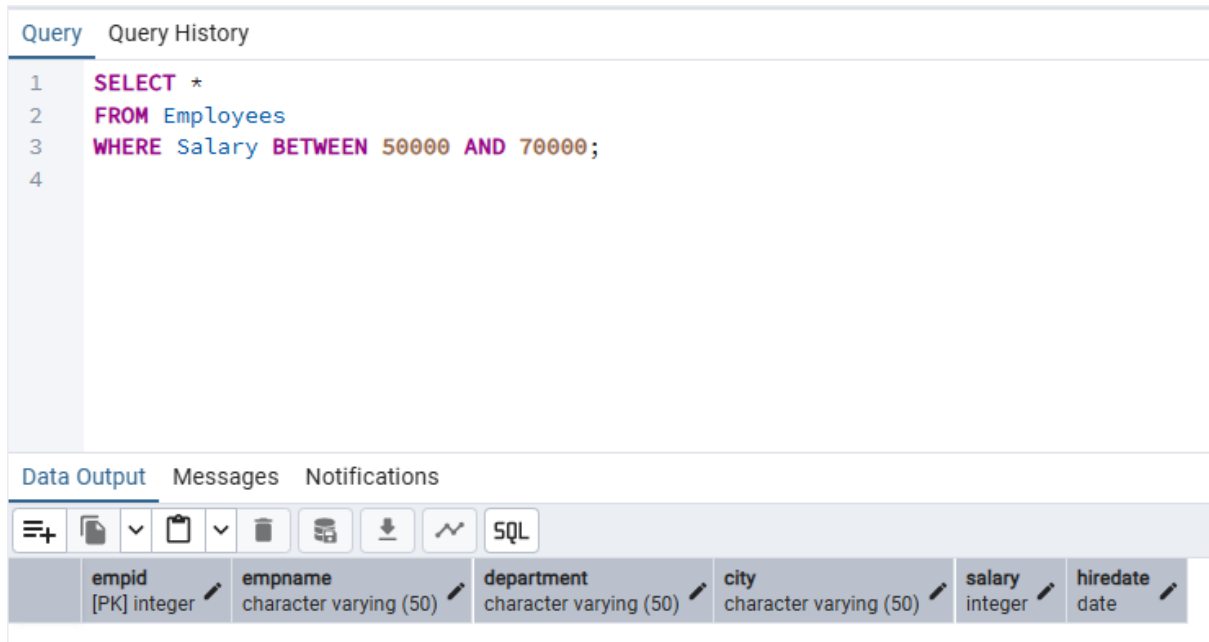
empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

Multiple departments are filtered using IN for cleaner, readable syntax.

Question 3: Display employees whose salary is between ₹50,000 and ₹70,000.

SQL Query:



The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query across four lines:

```
1 SELECT *
2 FROM Employees
3 WHERE salary BETWEEN 50000 AND 70000;
4
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table structure with the following columns and data types:

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

The BETWEEN operator is inclusive of both boundary values.

Question 4: List employees whose names start with the letter 'A'.

SQL Query:

[Query](#) [Query History](#)

```
1 SELECT *
2 FROM Employees
3 WHERE EmpName LIKE 'A%';
4
```

[Data Output](#) [Messages](#) [Notifications](#)



	empid [PK] integer	empname character varying (50)	department character varying (50)	city character varying (50)	salary integer	hiredate date
--	------------------------------	--	---	---------------------------------------	--------------------------	-------------------------

Explanation:

A% matches names beginning with A (example: Aman, Arjun, Anjali).

Question 5: Find employees whose names contain the substring 'an'.

SQL Query:

Query

Query History

1

2

3

4

SELECT *

FROM Employees

WHERE EmpName LIKE '%an%';

Data Output

Messages

Notifications

≡+

▼

▼

SQL

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

%an% matches any name that has "an" anywhere (Aman, Karan, Anjali).

Question 6: Show employees who are from ‘Delhi’ or ‘Mumbai’ and earn more than ₹55,000.

SQL Query:



The screenshot shows a SQL query editor with a query window and a data output window. The query window contains the following SQL query:

```
1 SELECT *
2 FROM Employees
3 WHERE City IN ('Delhi', 'Mumbai')
4 AND Salary > 55000;
5
```

The data output window shows a table with the following columns and data types:

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

Two filters are applied: (1) City must be Delhi or Mumbai; (2) Salary > 55,000.

Question 7: Display all employees except those from the 'HR' department.

SQL Query:

The screenshot shows an SQL IDE interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query in a text editor. The query is as follows:

```
1 SELECT *
2 FROM Employees
3 WHERE Department <> 'HR';
4
```

Below the text editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of employee data. The table has six columns: empid, empname, department, city, salary, and hiredate. Each column header is followed by its data type and a small edit icon. The data types are: empid [PK] integer, empname character varying (50), department character varying (50), city character varying (50), salary integer, and hiredate date.

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

The <> operator excludes HR employees.

Question 8: Get all employees hired between 2019 and 2022, ordered by HireDate (oldest first).

SQL Query:

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT *
2 FROM Employees
3 WHERE HireDate BETWEEN '2019-01-01' AND '2022-12-31'
4 ORDER BY HireDate ASC;
5
```

Below the query editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns and data types:

empid	empname	department	city	salary	hiredate
[PK] integer	character varying (50)	character varying (50)	character varying (50)	integer	date

Explanation:

- BETWEEN is inclusive of both years.
- ORDER BY ASC shows the oldest employees first.