

SQL ASSINGMENT

Gayatri Barhate

Creating table

```
CREATE TABLE employees (  
    emp_id INT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    department VARCHAR(50),  
    salary DOUBLE,  
    join_date DATE  
);
```

```
1 • USE demo_db;  
2 • CREATE TABLE employees (  
3     emp_id INT Primary KEY,  
4     name VARCHAR(100) NOT NULL,  
5     department VARCHAR(50),  
6     salary DOUBLE,  
7     join_date date);  
8  
9
```

The screenshot shows a database management interface. At the top, there's a 'Result Grid' tab and a 'Filter Rows' input field. Below the filter, a table structure is displayed with columns: emp_id, name, department, salary, and join_date. The first row of data shows all columns as NULL. On the right side, there's a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'.

emp_id	name	department	salary	join_date
*	NULL	NULL	NULL	NULL

Insertion of query

INSERT INTO employees (emp_id, name, department, salary, join_date)

VALUES

(101, 'John Doe', 'HR', 45000, '2021-06-15'),

(102, 'Jane Smith', 'IT', 75000, '2020-01-10'),

(103, 'Alice Johnson', 'Finance', 60000, '2019-08-23'),

(104, 'Bob Brown', 'IT', 80000, '2022-03-01'),

(105, 'Eve Davis', 'Marketing', 55000, '2021-11-05');

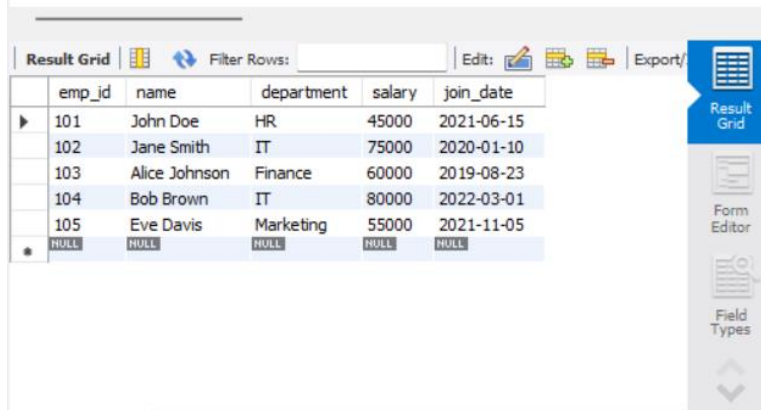
```
1 • USE demo_db;
2 • INSERT INTO employees (emp_id, name, department, salary, join_
3 (101, 'John Doe', 'HR', 45000, '2021-06-15'),
4 (102, 'Jane Smith', 'IT', 75000, '2020-01-10'),
5 (103, 'Alice Johnson', 'Finance', 60000, '2019-08-23'),
6 (104, 'Bob Brown', 'IT', 80000, '2022-03-01'),
7 (105, 'Eve Davis', 'Marketing', 55000, '2021-11-05');
8
9
```

emp_id	name	department	salary	join_date
101	John Doe	HR	45000	2021-06-15
102	Jane Smith	IT	75000	2020-01-10
103	Alice Johnson	Finance	60000	2019-08-23
104	Bob Brown	IT	80000	2022-03-01
105	Eve Davis	Marketing	55000	2021-11-05
* NOLE	NOLE	NOLE	NOLE	NOLE

Selection of query

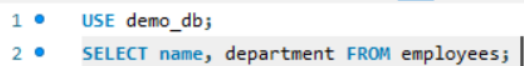
1. SELECT * FROM employees;

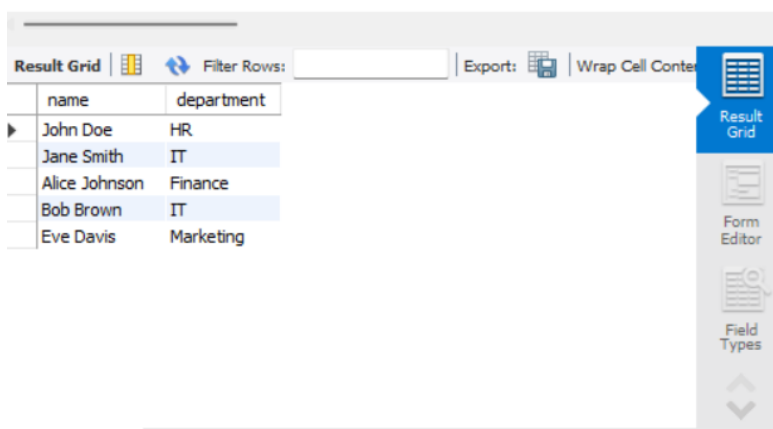
- 1 • `USE demo_db;`
- 2 • `SELECT * FROM employees;`



emp_id	name	department	salary	join_date
101	John Doe	HR	45000	2021-06-15
102	Jane Smith	IT	75000	2020-01-10
103	Alice Johnson	Finance	60000	2019-08-23
104	Bob Brown	IT	80000	2022-03-01
105	Eve Davis	Marketing	55000	2021-11-05
NULL	NULL	NULL	NULL	NULL

2. `SELECT name, department FROM employees;`

- 
- 1 • `USE demo_db;`
 - 2 • `SELECT name, department FROM employees;`



name	department
John Doe	HR
Jane Smith	IT
Alice Johnson	Finance
Bob Brown	IT
Eve Davis	Marketing

3. `SELECT * FROM employees WHERE department = 'IT';`

```
1 • USE demo_db;
2 • SELECT * FROM employees WHERE department = 'IT';
```

emp_id	name	department	salary	join_date
102	Jane Smith	IT	75000	2020-01-10
104	Bob Brown	IT	80000	2022-03-01
NULL	NULL	NULL	NULL	NULL

AND, IN BETWEEN & LIKE

SELECT * FROM employees

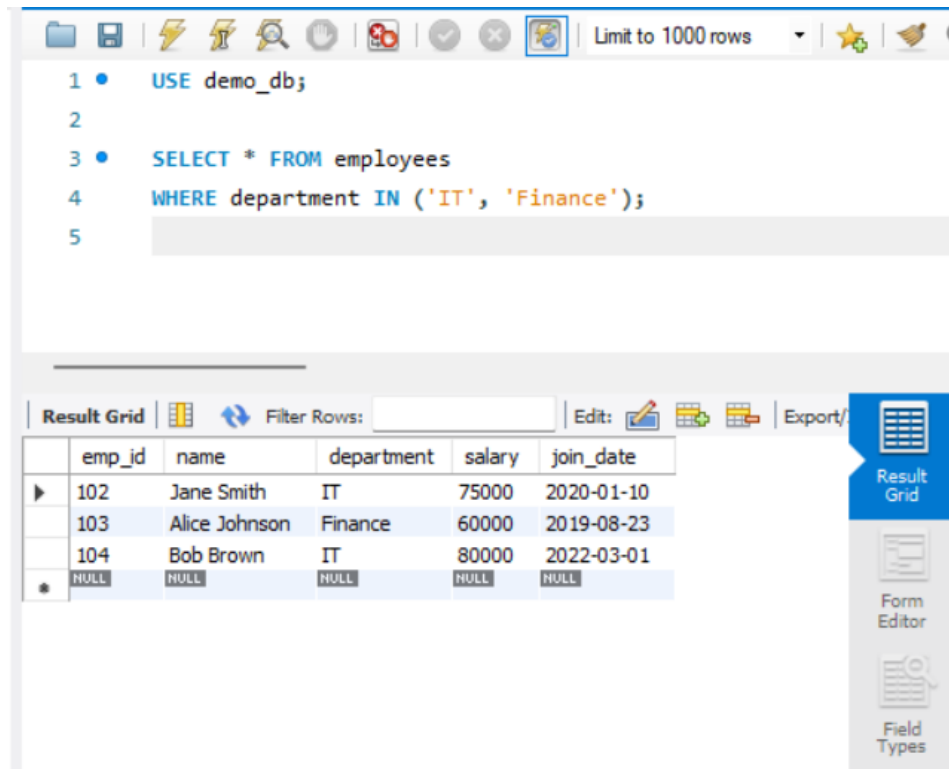
WHERE department = 'IT' AND salary > 75000;

```
1 • USE demo_db;
2
3 • SELECT * FROM employees
4 WHERE department = 'IT' AND salary > 75000;
5
```

emp_id	name	department	salary	join_date
104	Bob Brown	IT	80000	2022-03-01
NULL	NULL	NULL	NULL	NULL

SELECT * FROM employees

WHERE department IN ('IT', 'Finance');



The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
1 • USE demo_db;  
2  
3 • SELECT * FROM employees  
4   WHERE department IN ('IT', 'Finance');  
5
```

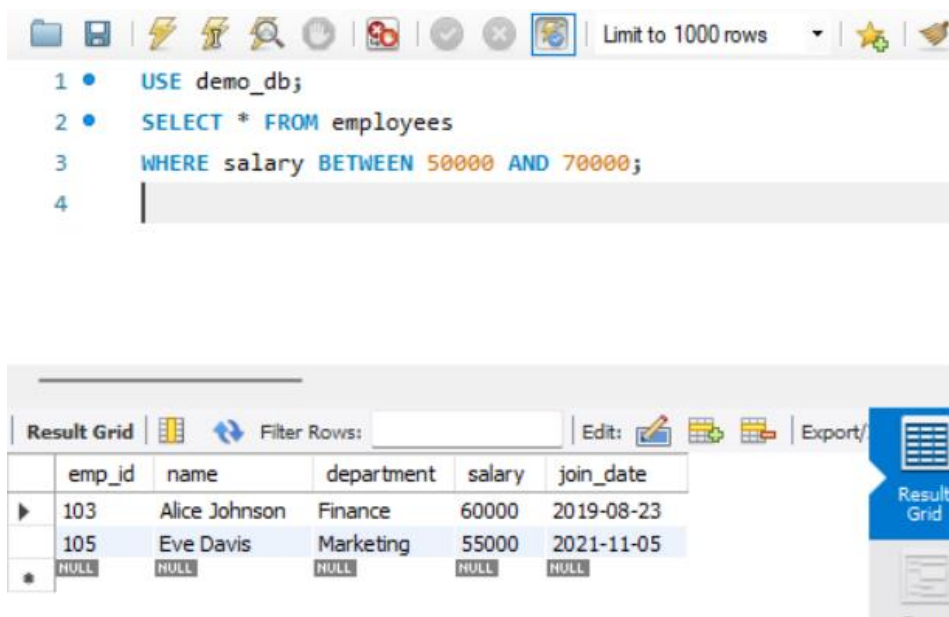
Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input field and an 'Export' button. The results are displayed in a table with the following data:

	emp_id	name	department	salary	join_date
▶	102	Jane Smith	IT	75000	2020-01-10
	103	Alice Johnson	Finance	60000	2019-08-23
	104	Bob Brown	IT	80000	2022-03-01
*	NULL	NULL	NULL	NULL	NULL

On the right side of the interface, there are three buttons: 'Result Grid' (highlighted in blue), 'Form Editor', and 'Field Types'.

SELECT * FROM employees

WHERE salary BETWEEN 50000 AND 70000;



The screenshot shows the same database query editor interface. The SQL editor contains the following query:

```
1 • USE demo_db;  
2 • SELECT * FROM employees  
3   WHERE salary BETWEEN 50000 AND 70000;  
4
```

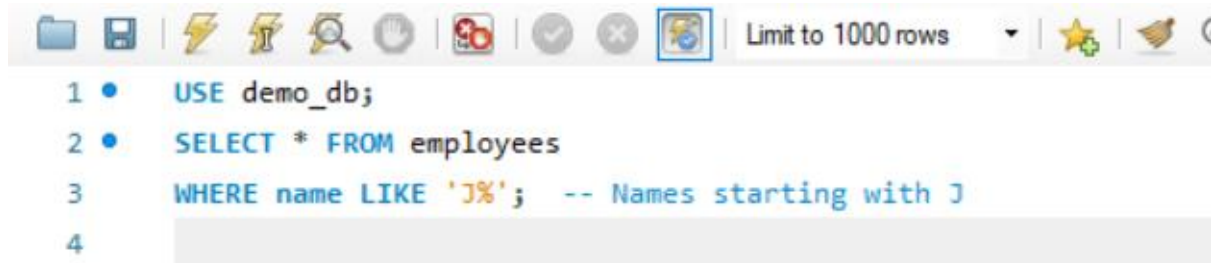
The 'Result Grid' section shows the results of the query in a table:

	emp_id	name	department	salary	join_date
▶	103	Alice Johnson	Finance	60000	2019-08-23
	105	Eve Davis	Marketing	55000	2021-11-05
*	NULL	NULL	NULL	NULL	NULL

The 'Result Grid' button on the right is highlighted in blue.

SELECT * FROM employees

WHERE name LIKE 'J%'; -- Names starting with J



A screenshot of a SQL query result grid. The toolbar at the top includes a 'Result Grid' button, a 'Filter Rows' input field, and buttons for 'Edit', 'Export', and 'Import'. The result grid displays the following data:

	emp_id	name	department	salary	join_date
▶	101	John Doe	HR	45000	2021-06-15
	102	Jane Smith	IT	75000	2020-01-10
✱	NULL	NULL	NULL	NULL	NULL

CLAUSE -ORDER BY, WHERE, HAVING

SELECT * FROM employees

ORDER BY salary DESC;

The screenshot shows a database query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and search. Below the toolbar, a SQL query is entered in a text area:

```
1 • USE demo_db;  
2 • SELECT * FROM employees  
3 ORDER BY salary DESC;  
4
```

Below the query editor, there is a section for the results. It includes a "Result Grid" tab, a "Filter Rows:" input field, and an "Edit:" button. The results are displayed in a table with the following columns: emp_id, name, department, salary, and join_date.

	emp_id	name	department	salary	join_date
▶	104	Bob Brown	IT	80000	2022-03-01
	102	Jane Smith	IT	75000	2020-01-10
	103	Alice Johnson	Finance	60000	2019-08-23
	105	Eve Davis	Marketing	55000	2021-11-05
	101	John Doe	HR	45000	2021-06-15
*	NULL	NULL	NULL	NULL	NULL

On the right side of the interface, there is a vertical sidebar with buttons for "Result Grid" (highlighted in blue), "Form Editor", and a search icon.

UPDATE QUERY

UPDATE employees

SET salary = 82000

WHERE emp_id = 104;

```

1 • USE demo_db;
2 • UPDATE employees
3   SET salary = 82000
4   WHERE emp_id = 104;
5

```

emp_id	name	department	salary	join_date
101	John Doe	HR	45000	2021-06-15
102	Jane Smith	IT	75000	2020-01-10
103	Alice Johnson	Finance	60000	2019-08-23
104	Bob Brown	IT	82000	2022-03-01
105	Eve Davis	Marketing	55000	2021-11-05
NULL	NULL	NULL	NULL	NULL

DELETE FROM employees

WHERE emp_id = 105;

```

1 • USE demo_db;
2 • DELETE FROM employees
3   WHERE emp_id = 105;
4

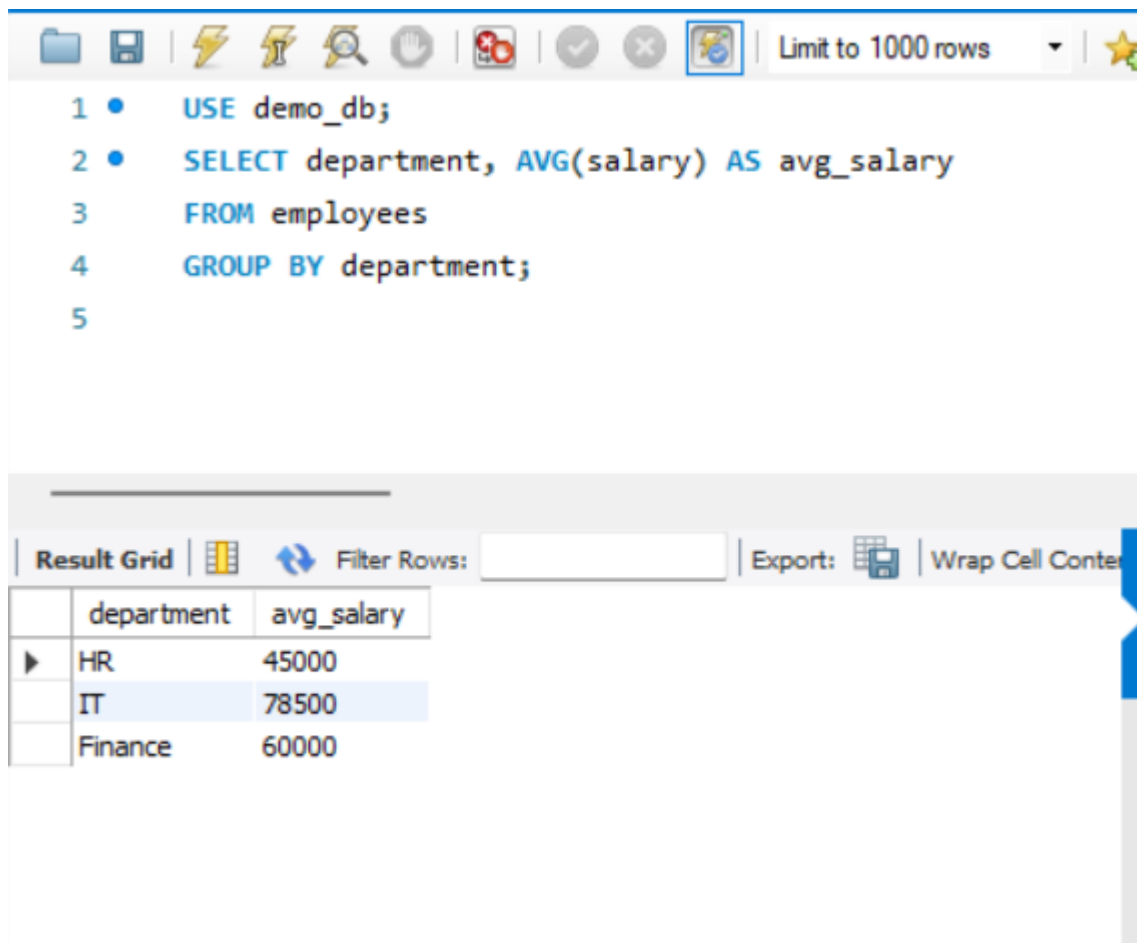
```

emp_id	name	department	salary	join_date
101	John Doe	HR	45000	2021-06-15
102	Jane Smith	IT	75000	2020-01-10
103	Alice Johnson	Finance	60000	2019-08-23
104	Bob Brown	IT	82000	2022-03-01
NULL	NULL	NULL	NULL	NULL

SELECT department, AVG(salary) AS avg_salary

FROM employees

GROUP BY department;



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL code:

```
1 • USE demo_db;  
2 • SELECT department, AVG(salary) AS avg_salary  
3 FROM employees  
4 GROUP BY department;  
5
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has two columns: 'department' and 'avg_salary'. The results are as follows:

	department	avg_salary
▶	HR	45000
	IT	78500
	Finance	60000

```
SELECT department, COUNT(*) AS emp_count  
FROM employees  
GROUP BY department  
HAVING COUNT(*) > 1;
```

Limit to 1000 rows

```
1 • USE demo_db;
2
3 • SELECT department, COUNT(*) AS emp_count
4 FROM employees
5 GROUP BY department
6 HAVING COUNT(*) > 1;
7
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

	department	emp_count
▶	IT	2