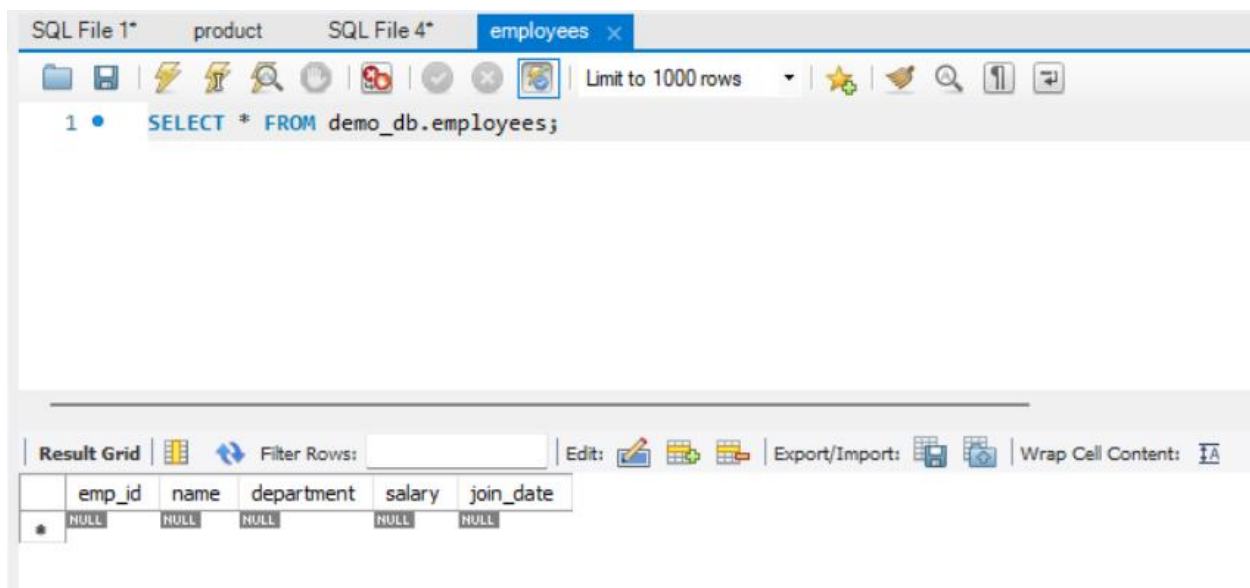


Name :- Sakshi Kadam

Creating table :-)

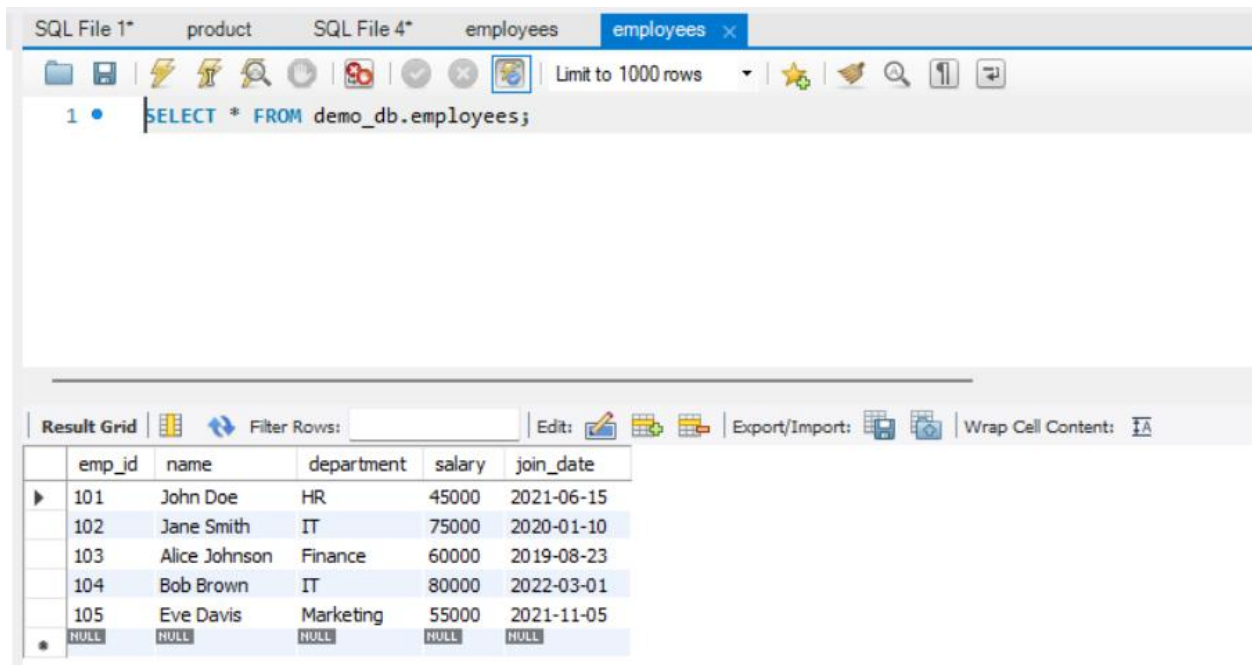
CREATE TABLE employees (emp_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, department VARCHAR(50), salary DOUBLE, join_date DATE);



Insert 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');



The screenshot shows a SQL IDE interface with a tab labeled 'employees'. The query editor contains the SQL statement: `SELECT * FROM demo_db.employees;`. Below the editor, the 'Result Grid' displays the query results in a table format. The table has five columns: emp_id, name, department, salary, and join_date. It contains five rows of employee data, followed by a row of NULL values. The interface also includes a toolbar with various icons and a 'Limit to 1000 rows' dropdown.

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

Select Query :-)

1. *SELECT * FROM employees;*

SQL File 1* product **SQL File 4*** x employees employees

Limit to 1000 rows

```

1 • USE demo_db;
2
3 • SELECT * FROM employees;

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	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 , emp_id FROM employees;*

SQL File 1* product **SQL File 4*** x employees employees

Limit to 1000 rows

```

1 • USE demo_db;
2
3 • SELECT name, department, emp_id FROM employees;

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	name	department	emp_id
▶	John Doe	HR	101
	Jane Smith	IT	102
	Alice Johnson	Finance	103
	Bob Brown	IT	104
	Eve Davis	Marketing	105
*	NULL	NULL	NULL

3. *SELECT * FROM employees WHERE department = 'HR';*

The screenshot shows a SQL IDE interface with a toolbar and a query editor. The query editor contains the following SQL code:

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

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has columns for emp_id, name, department, salary, and join_date. The results are as follows:

emp_id	name	department	salary	join_date
101	John Doe	HR	45000	2021-06-15
*	NULL	NULL	NULL	NULL

4. AND, IN BETWEEN & LIKE

***SELECT * FROM employees
WHERE department = 'IT' AND salary > 75000;***

The screenshot shows a SQL IDE interface with a toolbar and a query editor. The query editor contains the following SQL code:

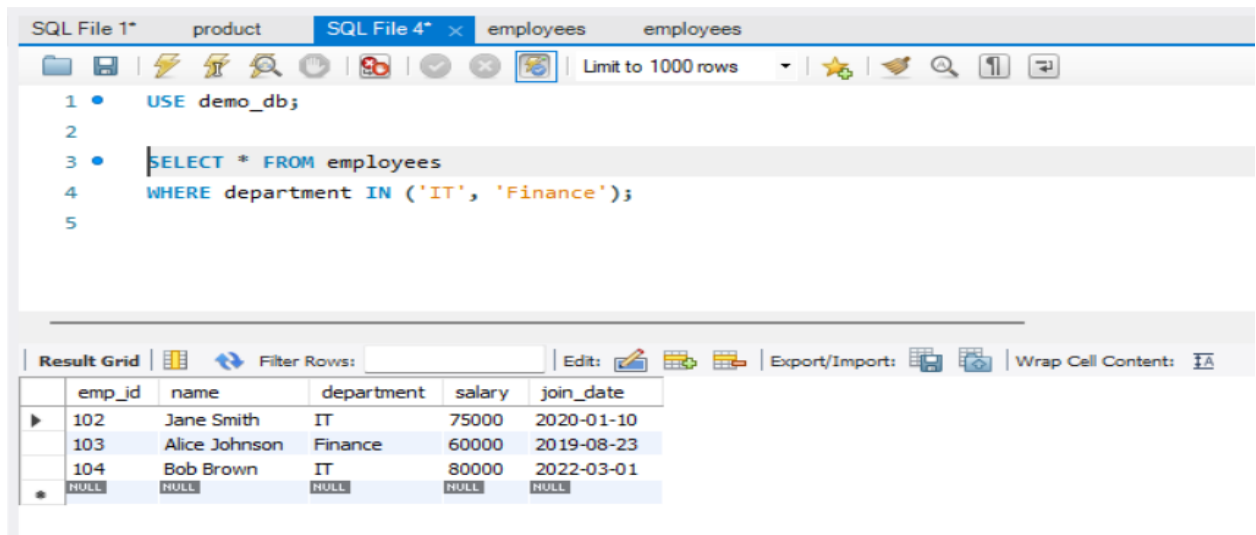
```
1 • USE demo_db;
2
3 • SELECT * FROM employees
4 WHERE department = 'IT' AND salary > 75000;
5
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has columns for emp_id, name, department, salary, and join_date. The results are as follows:

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

5. SELECT * FROM employees

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



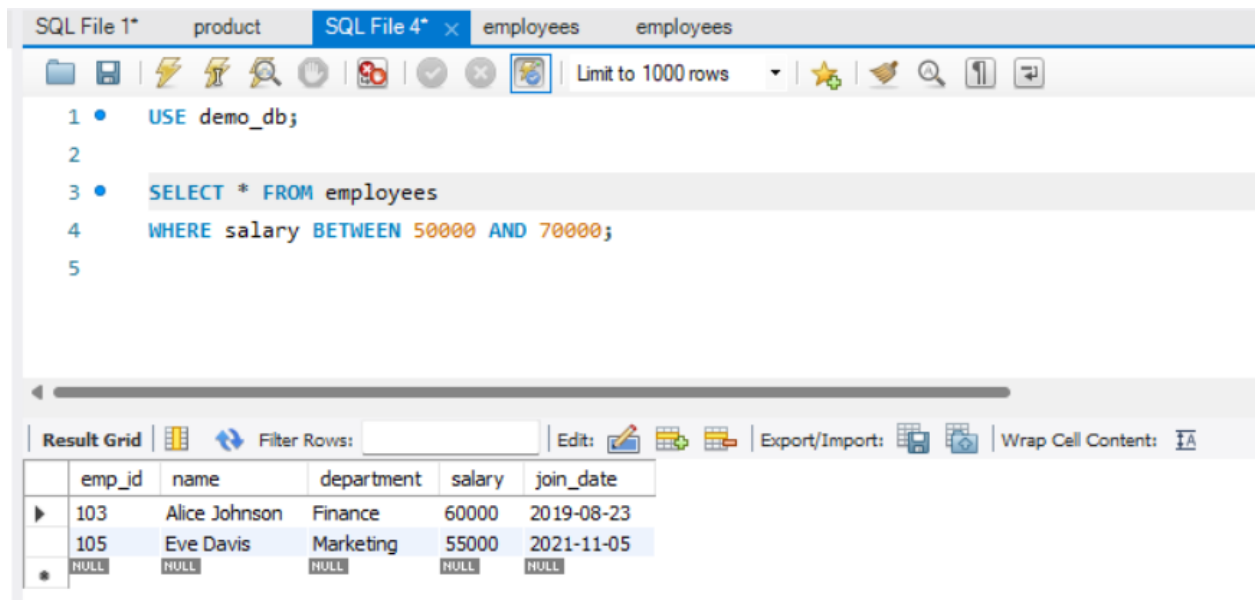
The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

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

The result grid displays 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

***6. SELECT * FROM employees
WHERE salary BETWEEN 50000 AND 70000;***



The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • USE demo_db;  
2  
3 • SELECT * FROM employees  
4 WHERE salary BETWEEN 50000 AND 70000;  
5
```

The result grid displays the following data:

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

***7. SELECT * FROM employees
WHERE name LIKE 'J%'; -- Names starting with J***

SQL File 1* product **SQL File 4*** × employees employees

Limit to 1000 rows

```

1 • USE demo_db;
2
3 • SELECT * FROM employees
4   WHERE name LIKE 'J%'; -- Names starting with J
5

```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	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

8. CLAUSE -ORDER BY, WHERE, HAVING

***SELECT * FROM employees
ORDER BY salary DESC;***

SQL File 1* product **SQL File 4*** × employees employees

Limit to 1000 rows

```

1 • USE demo_db;
2
3 • SELECT * FROM employees
4   ORDER BY salary DESC;
5
6

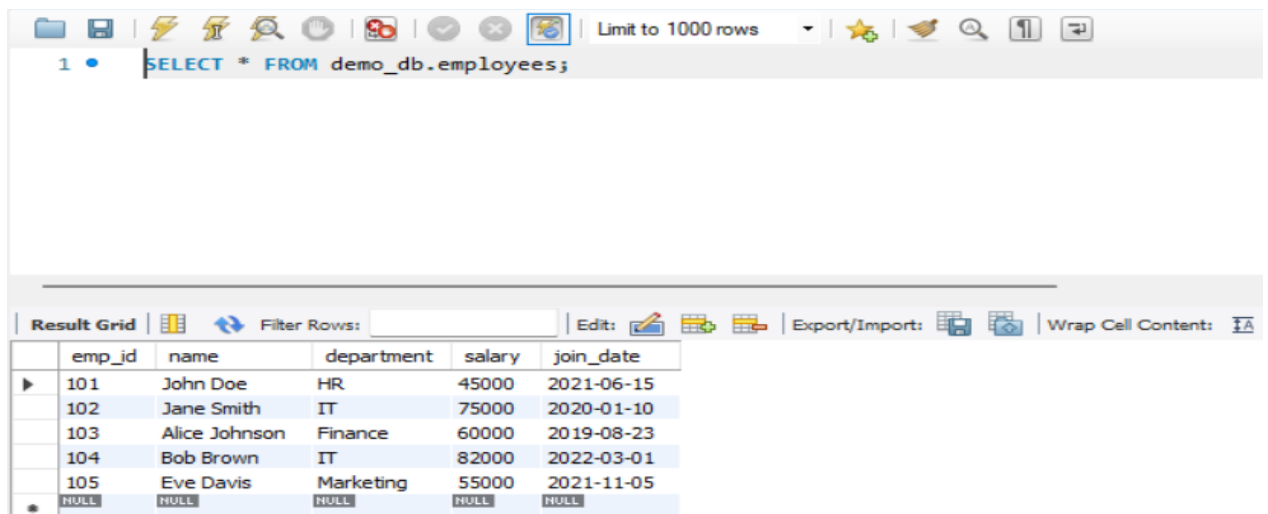
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	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

9. UPDATE QUERY

UPDATE employees SET salary = 82000 WHERE emp_id = 104;



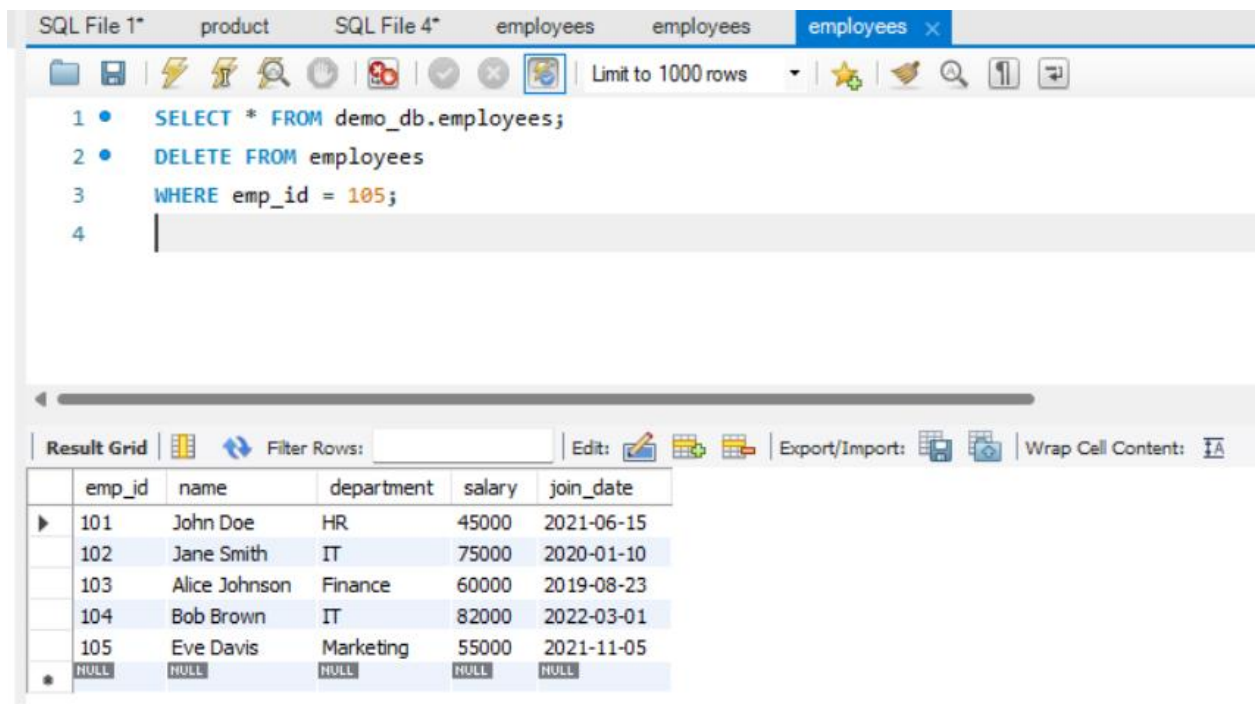
Limit to 1000 rows

```
1 • SELECT * FROM demo_db.employees;
```

Result Grid

	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

10. ***DELETE FROM employees WHERE emp_id = 105;***



SQL File 1* product SQL File 4* employees employees employees x

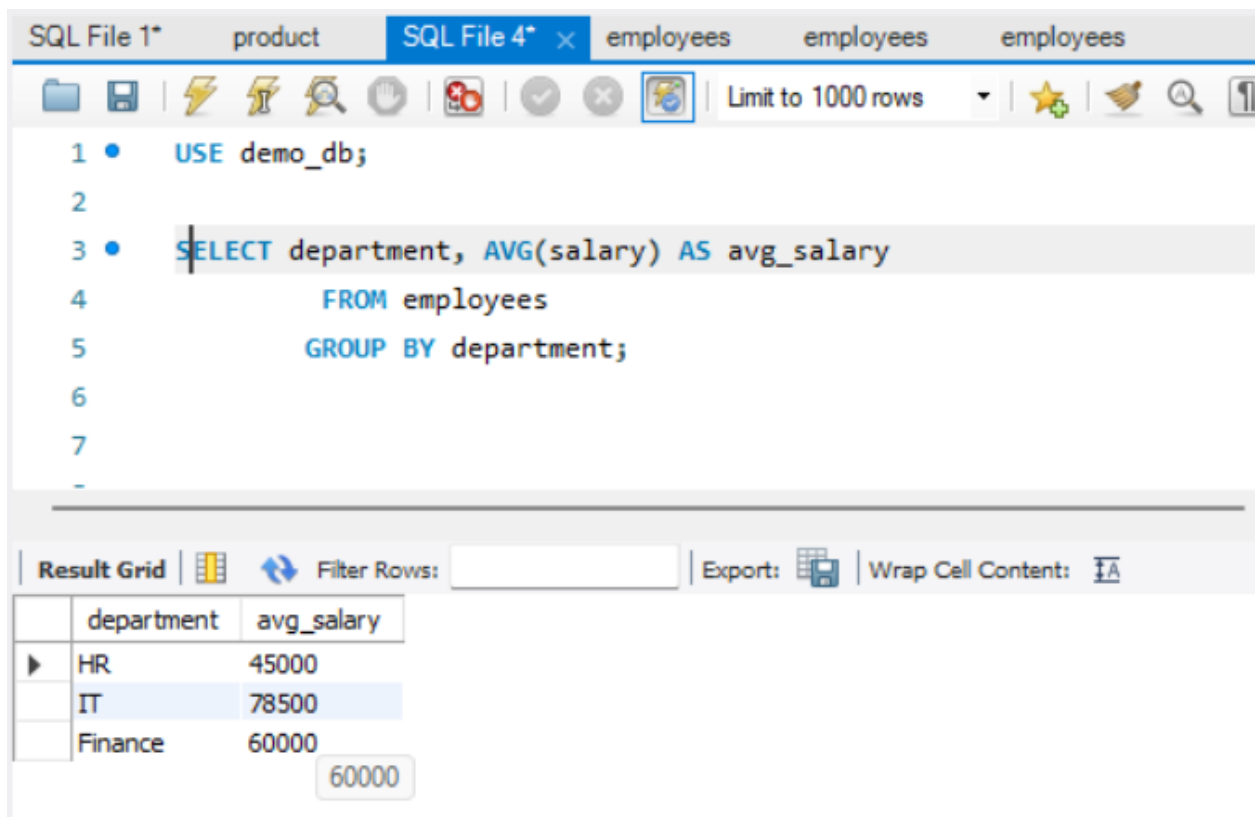
Limit to 1000 rows

```
1 • SELECT * FROM demo_db.employees;
2 • DELETE FROM employees
3   WHERE emp_id = 105;
4
```

Result Grid

	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

11. ***SELECT department, AVG(salary) AS avg_salary
FROM employees
GROUP BY department;***



The screenshot shows a SQL IDE interface with a tab labeled "SQL File 4* x" containing the following SQL query:

```
1 • USE demo_db;  
2  
3 • SELECT department, AVG(salary) AS avg_salary  
4       FROM employees  
5       GROUP BY department;  
6  
7  
-
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query:

	department	avg_salary
▶	HR	45000
	IT	78500
	Finance	60000

A tooltip showing the value "60000" is visible over the Finance row's avg_salary cell.

12. ***SELECT department, COUNT(*) AS emp_count
FROM employees
GROUP BY department***

HAVING COUNT(*) > 1;

The screenshot shows a SQL IDE interface with a toolbar at the top containing icons for file operations, execution, and search. Below the toolbar, the SQL query is displayed in a text editor. The query is as follows:

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

Below the query editor, there is a section for the results. It includes a 'Result Grid' button, a 'Filter Rows' input field, and an 'Export' button. The results are displayed in a table with two columns: 'department' and 'emp_count'.

	department	emp_count
▶	IT	2