

Assignment 2

Create a Database name entri_assignment

Create a Table with name departments

Department_id (pk) Department_name Location_id

Create a Table with name employees

Employee_id (pk) ,first_name,last_name ,email,phone_number,hire_date,
job_id, salary, commission_pct, manager_id, department_id (fk
reference to departments)

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the 'entri_assignment' database, including tables 'departments' and 'employees'. Below this, the 'Table: departments' is detailed with columns: 'Department_id' (int PK), 'Department_name' (varchar(255)), and 'Location_id' (int). The main pane on the right shows the SQL code for creating the database and tables. The code includes comments and SQL statements for creating the database 'entri_assignment', using it, creating the 'departments' table with a primary key on 'Department_id', and creating the 'employees' table with a primary key on 'Employee_id' and a foreign key on 'Department_id' that references the 'departments' table.

```
1 -- Create the database
2 • CREATE DATABASE entri_assignment;
3 -- Use the database
4 • USE entri_assignment;
5 -- Create the table "departments"
6 • CREATE TABLE departments (
7     Department_id INT PRIMARY KEY, Department_name VARCHAR(255), Location_id INT);
8 -- Use the database
9 • USE entri_assignment;
10 -- Create the table "employees"
11 • CREATE TABLE employees ( Employee_id INT PRIMARY KEY,
12     First_name VARCHAR(255),
13     Last_name VARCHAR(255),
14     Email VARCHAR(255),
15     Phone_number VARCHAR(20),
16     Hire_date DATE,
17     Job_id VARCHAR(10),
18     Salary DECIMAL(10, 2),
19     Commission_pct DECIMAL(4, 2),
20     Manager_id INT,
21     Department_id INT,
22     CONSTRAINT fk_department_id
23     FOREIGN KEY (Department_id)
24     REFERENCES departments(Department_id)
25 );
26
```

1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S

SELECT First_name, Last_name, Job_id, Salary

FROM employees

WHERE First_name LIKE 'S%'

```
1 • SELECT First_name, Last_name, Job_id, Salary
2 FROM employees
3 WHERE First_name LIKE 'S%'
4
```

< Result Grid Filter Rows: Export: Wrap Cell Cor

	First_name	Last_name	Job_id	Salary
▶	Steven	King	AD_PRES	24000.00
	Shelli	Baida	PU_CLERK	2900.00
	Sigal	Tobias	PU_CLERK	2800.00
	Shanta	Vollman	ST_MAN	6500.00
	Steven	Markle	ST_CLERK	2200.00

2. Write a query to select employee with the highest salary (using inner query)

SELECT *

FROM employees WHERE Salary = (SELECT MAX(Salary) FROM employees)

```

1 • SELECT *
2 FROM employees
3 WHERE Salary = (
4     SELECT MAX(Salary)
5     FROM employees
6 )
7

```

	Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
▶	100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Select the employee with the second-highest salary

SELECT *

FROM employees

WHERE Salary = (

SELECT MAX(Salary)

FROM employees

WHERE Salary < (

SELECT MAX(Salary)

FROM employees

)

)

SQL File 6* x departments employees

Limit to 50000 rows

```

1 • SELECT *
2   FROM employees
3  WHERE Salary = (
4      SELECT MAX(Salary)
5      FROM employees
6  WHERE Salary < (
7      SELECT MAX(Salary)
8      FROM employees
9  )
10 )

```

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

	Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
▶	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD_VP	17000.00	NULL	100	20
	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-09-12	AD_VP	17000.00	NULL	100	30
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Write a query to select employees and their corresponding managers and their salaries

SELECT

e.Employee_id, e.First_name AS Employee_first_name, e.Last_name AS Employee_last_name, e.Salary AS Employee_salary,

m.Employee_id AS Manager_id, m.First_name AS Manager_first_name, m.Last_name AS Manager_last_name, m.Salary AS Manager_salary

FROM employees e

LEFT JOIN employees m **ON** e.Manager_id = m.Employee_id

SQL File 6* x departments employees

Limit to 50000 rows

```

1 • SELECT
2     e.Employee_id,
3     e.First_name AS Employee_first_name,
4     e.Last_name AS Employee_last_name,
5     e.Salary AS Employee_salary,
6     m.Employee_id AS Manager_id,
7     m.First_name AS Manager_first_name,
8     m.Last_name AS Manager_last_name,
9     m.Salary AS Manager_salary
10  FROM employees e
11  LEFT JOIN employees m ON e.Manager_id = m.Employee_id
12

```

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

	Employee_id	Employee_first_name	Employee_last_name	Employee_salary	Manager_id	Manager_first_name	Manager_last_name	Manager_salary
▶	100	Steven	King	24000.00	NULL	NULL	NULL	NULL
	101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
	102	Lex	De Haan	17000.00	100	Steven	King	24000.00
	103	Alexander	Hunold	9000.00	102	Lex	De Haan	17000.00
	104	Bruce	Ernst	6000.00	103	Alexander	Hunold	9000.00
	105	David	Austin	4800.00	103	Alexander	Hunold	9000.00
	106	Valli	Pataballa	4800.00	103	Alexander	Hunold	9000.00
	107	Diana	Lorentz	4200.00	103	Alexander	Hunold	9000.00
	108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
	109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
	110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00

5. Write a query to select employees and their corresponding managers and their salaries (SELF Join)

SELECT

e.Employee_id, e.First_name AS Employee_first_name, e.Last_name AS Employee_last_name, e.Salary AS Employee_salary,

m.Employee_id AS Manager_id, m.First_name AS Manager_first_name, m.Last_name AS Manager_last_name, m.Salary AS Manager_salary

FROM employees e

INNER JOIN employees m **ON** e.Manager_id = m.Employee_id

SQL File 6* departments employees

Limit to 50000 rows

```

1 • SELECT
2     e.Employee_id,
3     e.First_name AS Employee_first_name,
4     e.Last_name AS Employee_last_name,
5     e.Salary AS Employee_salary,
6     m.Employee_id AS Manager_id,
7     m.First_name AS Manager_first_name,
8     m.Last_name AS Manager_last_name,
9     m.Salary AS Manager_salary
10  FROM employees e
11  INNER JOIN employees m ON e.Manager_id = m.Employee_id
12

```

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

	Employee_id	Employee_first_name	Employee_last_name	Employee_salary	Manager_id	Manager_first_name	Manager_last_name	Manager_salary
▶	101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
	102	Lex	De Haan	17000.00	100	Steven	King	24000.00
	103	Alexander	Hunold	9000.00	102	Lex	De Haan	17000.00
	104	Bruce	Ernst	6000.00	103	Alexander	Hunold	9000.00
	105	David	Austin	4800.00	103	Alexander	Hunold	9000.00
	106	Valli	Pataballa	4800.00	103	Alexander	Hunold	9000.00
	107	Diana	Lorentz	4200.00	103	Alexander	Hunold	9000.00
	108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
	109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
	110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00
	111	Ismael	Sciarra	7700.00	108	Nancy	Greenberg	12000.00
	112	Jose Manuel	Urman	7800.00	108	Nancy	Greenberg	12000.00

6. Create a view for the above query

```
CREATE VIEW employees_managers_view AS
```

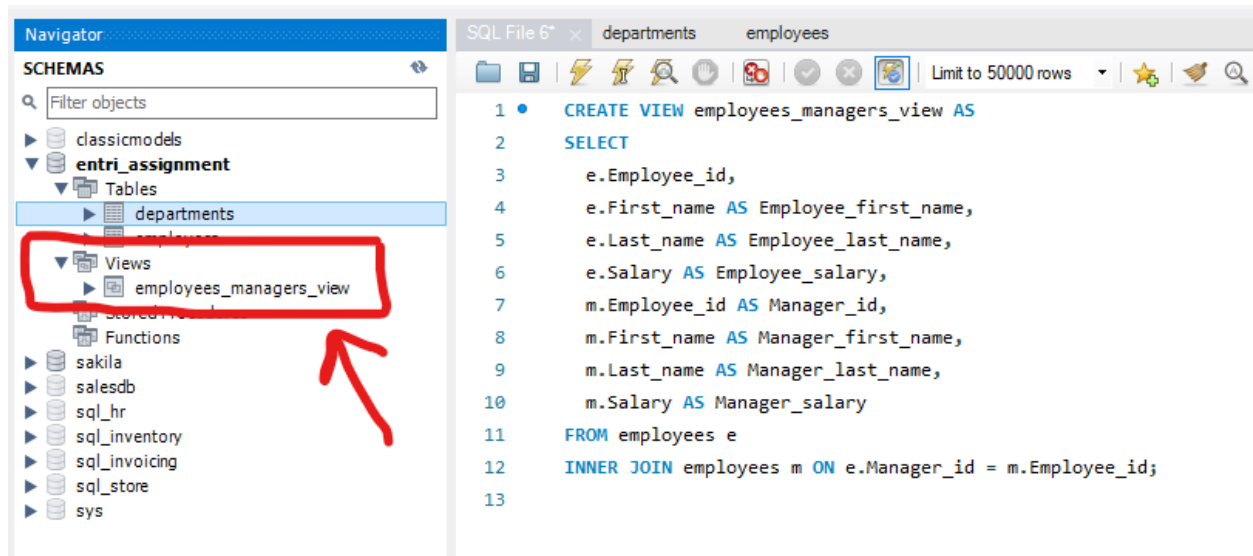
```
SELECT
```

```
e.Employee_id, e.First_name AS Employee_first_name, e.Last_name AS
Employee_last_name, e.Salary AS Employee_salary,
```

```
m.Employee_id AS Manager_id, m.First_name AS Manager_first_name,
m.Last_name AS Manager_last_name, m.Salary AS Manager_salary
```

```
FROM employees e
```

```
INNER JOIN employees m ON e.Manager_id = m.Employee_id;
```



7. Write a query to show count of employees under each manager in descending order (from view)

```
SELECT Manager_id, Manager_first_name, Manager_last_name,  
COUNT(Employee_id) AS Employee_Count
```

```
FROM employees_managers_view
```

```
GROUP BY Manager_id, Manager_first_name, Manager_last_name
```

```
ORDER BY Employee_Count DESC;
```

SQL File 6* x departments employees

Limit to 50000 rows

```

1 • SELECT
2     Manager_id,
3     Manager_first_name,
4     Manager_last_name,
5     COUNT(Employee_id) AS Employee_Count
6 FROM
7     employees_managers_view
8 GROUP BY Manager_id , Manager_first_name , Manager_last_name
9 ORDER BY Employee_Count DESC;
10
11

```

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

	Manager_id	Manager_first_name	Manager_last_name	Employee_Count
▶	100	Steven	King	8
	108	Nancy	Greenberg	5
	114	Den	Raphaely	5
	103	Alexander	Hunold	4
	120	Matthew	Weiss	4
	121	Adam	Fripp	2
	102	Lex	De Haan	1
	101	Neena	Kochhar	1

8. Find the count of employees in each department

```
SELECT department_id, COUNT(*) AS employee_count
```

```
FROM employees
```

```
GROUP BY department_id;
```


SQL File 6* x departments employees

Limit to 50000 rows

```
1 • SELECT
2     department_id, COUNT(*) AS employee_count
3 FROM
4     employees
5 GROUP BY department_id;
6
```

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

	department_id	employee_count
▶	20	2
	30	3
	40	3
	50	7
	60	4
	70	1
	80	2
	90	1
	100	1
	110	1
	130	1
	140	1
	150	1
	160	1
	170	2

9. Get the count of employees hired year wise

```
SELECT YEAR(Hire_date) AS Hire_Year, COUNT(*) AS Employee_Count
FROM employees
GROUP BY Hire_Year ORDER BY Hire_Year;
```

SQL File 6* x departments employees

Limit to 50000 rows

```
1 • SELECT YEAR(Hire_date) AS Hire_Year, COUNT(*) AS Employee_Count
2 FROM employees
3 GROUP BY Hire_Year
4 ORDER BY Hire_Year;
5
```

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

	Hire_Year	Employee_Count
▶	1987	1
	1989	1
	1990	1
	1991	1
	1993	1
	1994	3
	1995	2
	1996	1
	1997	10
	1998	4
	1999	5
	2000	1

10 . create a stored procedure to get the “ Get the count of employees hired in the input year”(IN year , OUT count)

DELIMITER //

CREATE PROCEDURE GetEmployeeCountByYear(IN input_year INT, OUT employee_count INT)

BEGIN

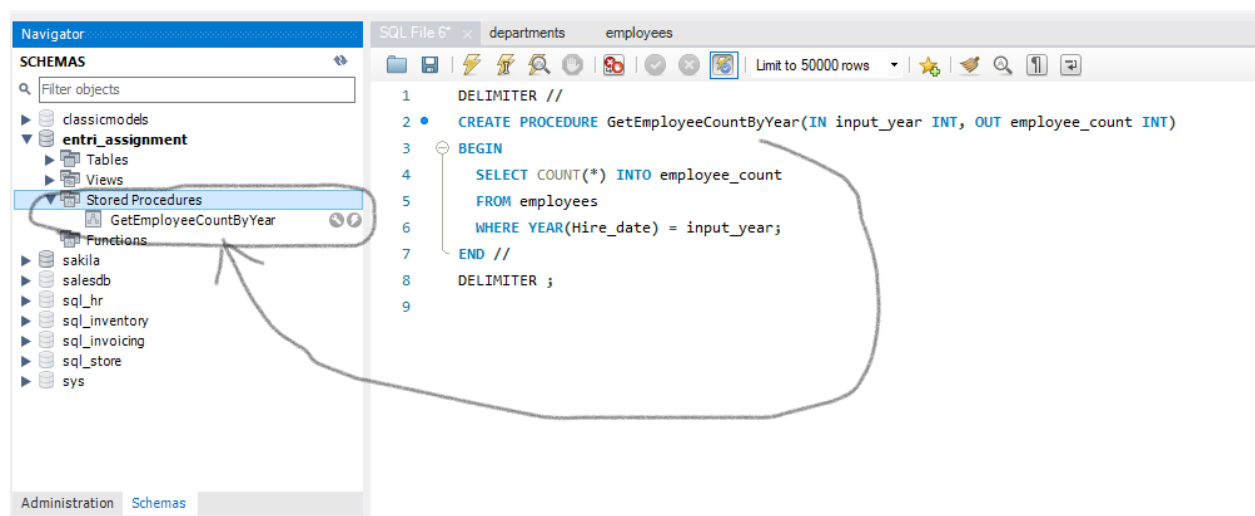
SELECT COUNT(*) INTO employee_count

FROM employees

WHERE YEAR(Hire_date) = input_year;

END //

DELIMITER ;



To use this stored procedure, you can call it and pass the input year as an argument. Here's an example:

SQL File 6* x departments employees

Limit to 50000 rows

```

1 • CALL GetEmployeeCountByYear(2000, @count);
2   SELECT @count AS EmployeeCount;
3

```

Result Grid

EmployeeCount
1

11. Select the employees whose first_name contains "an"

SELECT * FROM employees

WHERE first_name LIKE '%an%';

SQL File 6* x departments employees

Limit to 50000 rows

```

1 • SELECT *
2   FROM employees
3  WHERE first_name LIKE '%an%';
4

```

Result Grid

Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-09-30	IT_PROG	9000.00	NULL	102	60
107	Diana	Lorentz	DLorentz	590.423.5567	1999-02-09	IT_PROG	4200.00	NULL	103	40
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-12	FI_ACCOUNT	9000.00	NULL	108	170
112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-06-03	FI_ACCOUNT	7800.00	NULL	108	150
115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-12	PU_CLERK	3100.00	NULL	114	80
123	Shanta	Vollman	SVOLLMAN	650.123.4234	1997-10-12	ST_MAN	6500.00	NULL	100	50
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

12. Select employee first name and the corresponding phone number in the format (_ _)-(_ _)-(_ _ _)

SELECT

First_name,

CONCAT('(', SUBSTRING(phone_number, 1, 3), '-')-
SUBSTRING(phone_number, 4, 3), '-', SUBSTRING(phone_number, 7, 4))
AS formatted_phone_number FROM employees;

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 6'. The query is as follows:

```
1 • SELECT
2     First_name,
3     CONCAT('(',
4         SUBSTRING(phone_number, 1, 3),
5         ')-',
6         SUBSTRING(phone_number, 4, 3),
7         '-')',
8         SUBSTRING(phone_number, 7, 4)) AS formatted_phone_number
9 FROM
10    employees;
11
```

The results are displayed in a table with two columns: 'First_name' and 'formatted_phone_number'.

First_name	formatted_phone_number
Steven	(515)-.12-3.45
Neena	(515)-.12-3.45
Lex	(515)-.12-3.45
Alexander	(590)-.42-3.45
Bruce	(590)-.42-3.45
David	(590)-.42-3.45
Valli	(590)-.42-3.45
Diana	(590)-.42-3.55
Nancy	(515)-.12-4.45
Daniel	(515)-.12-4.41
John	(515)-.12-4.42
Ismael	(515)-.12-4.43
Jose Manuel	(515)-.12-4.44

13. Find the employees who joined in August, 1994.

SELECT *

FROM employees

WHERE YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 6*' containing the following SQL code:

```
1 • SELECT *
2 FROM employees
3 WHERE YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;
```

Below the query window, the 'Result Grid' displays the results of the query. The table has 12 columns: Employee_id, First_name, Last_name, Email, Phone_number, Hire_date, Job_id, Salary, Commission_pct, Manager_id, and Department_id. The results show three rows of data for employees hired in August 1994.

Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-12	FI_ACCOUNT	9000.00	NULL	108	170
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

14. Find the maximum salary from each department.

SELECT department_id, MAX(salary) AS max_salary

FROM employees GROUP BY department_id;

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 6*' containing the following SQL code:

```
1 • SELECT department_id, MAX(salary) AS max_salary
2 FROM employees
3 GROUP BY department_id;
4
```

Below the query window, the 'Result Grid' displays the results of the query. The table has two columns: department_id and max_salary. The results show the maximum salary for each department.

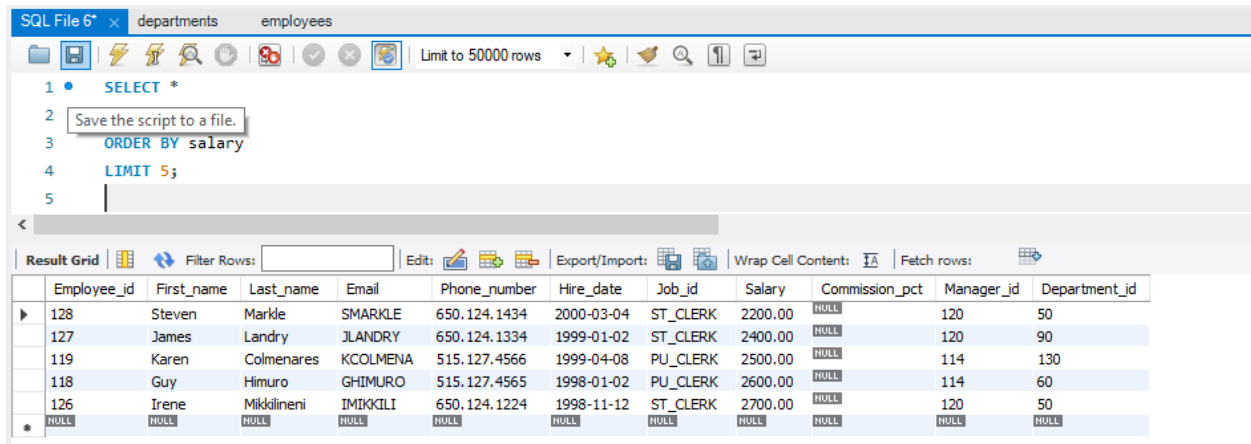
department_id	max_salary
20	24000.00
30	17000.00
40	7900.00
50	8200.00
60	9000.00
70	2900.00
80	5800.00

15. Write a SQL query to display the 5 least earning employees

SELECT * FROM employees

ORDER BY salary

LIMIT 5;



The screenshot shows a SQL File 6 window with a query editor and a result grid. The query is:

```
1 SELECT *
2
3 ORDER BY salary
4
5 LIMIT 5;
```

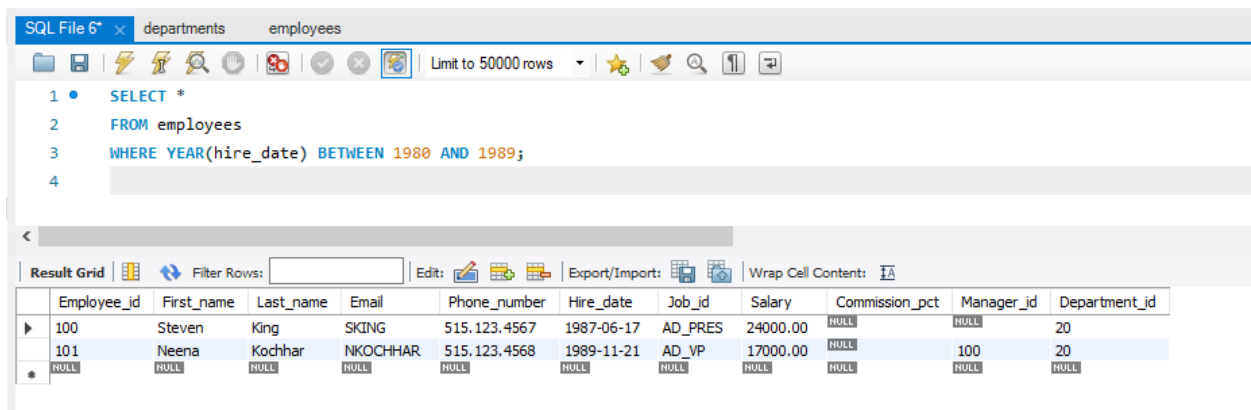
The result grid displays the top 5 employees by salary:

Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
128	Steven	Markle	SMARKLE	650.124.1434	2000-03-04	ST_CLERK	2200.00	NULL	120	50
127	James	Landry	JLANDRY	650.124.1334	1999-01-02	ST_CLERK	2400.00	NULL	120	90
119	Karen	Colmenares	KCOLMENAS	515.127.4566	1999-04-08	PU_CLERK	2500.00	NULL	114	130
118	Guy	Himuro	GHIMURO	515.127.4565	1998-01-02	PU_CLERK	2600.00	NULL	114	60
126	Irene	Mikkilineni	IMIKKILI	650.124.1224	1998-11-12	ST_CLERK	2700.00	NULL	120	50

16. Find the employees hired in the 80s

SELECT * FROM employees

WHERE YEAR(hire_date) BETWEEN 1980 AND 1989;



The screenshot shows a SQL File 6 window with a query and its results. The query is:

```
1 SELECT *
2
3 FROM employees
4
5 WHERE YEAR(hire_date) BETWEEN 1980 AND 1989;
```

The result grid displays the employees hired in the 1980s:

Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD PRES	24000.00	NULL	NULL	20
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD VP	17000.00	NULL	100	20

17. Find the employees who joined the company after 15th of the month

SELECT *

FROM employees

```
WHERE DAY(hire_date) > 15;
```

[illegible]