Assignment 2

Create a Databse 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)`

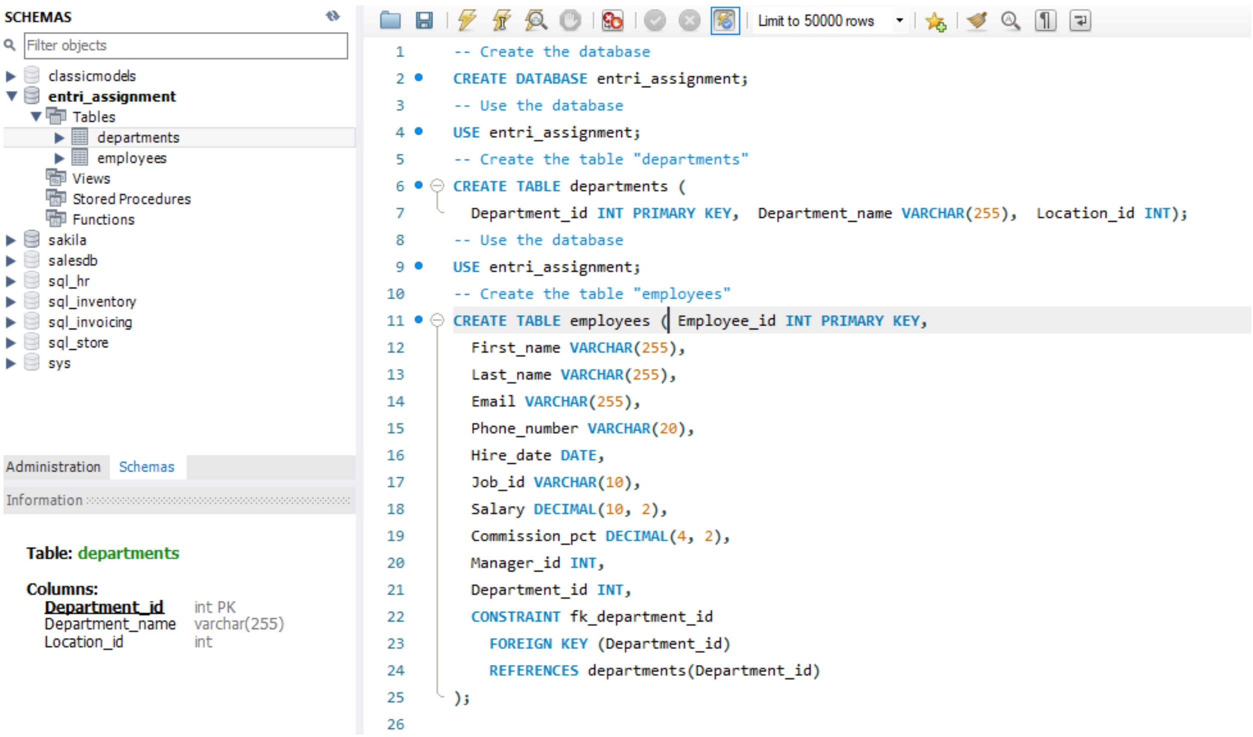


Table: departments

Columns:

Department_id	int PK
Department_name	varchar(255)
Location_id	int

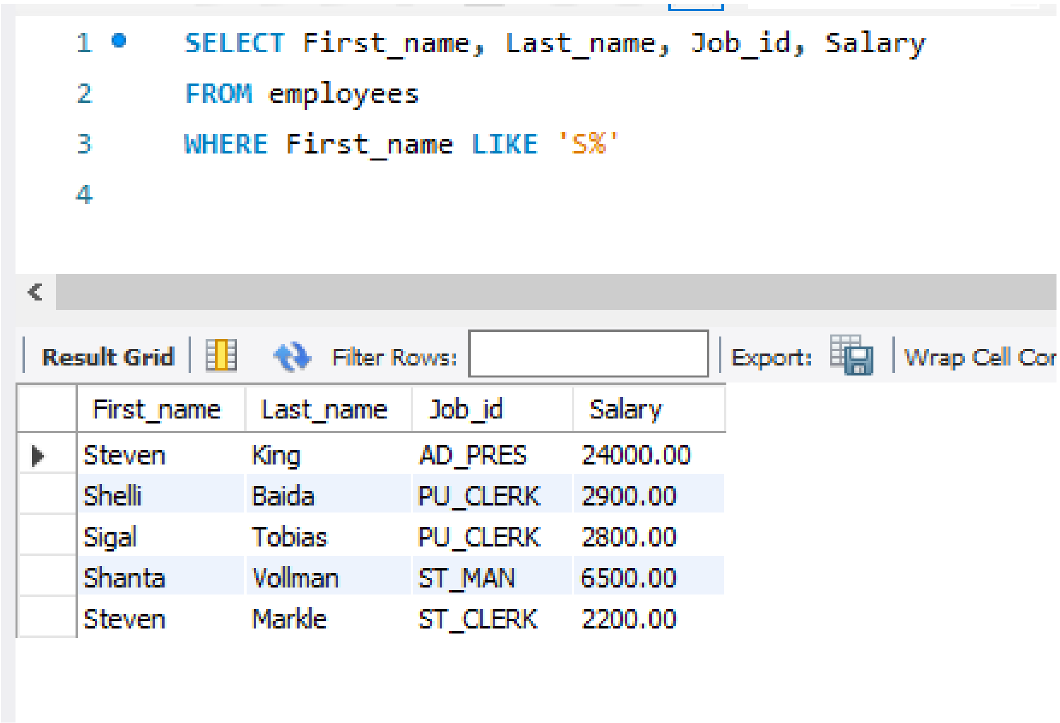
```sql
-- Create the database
CREATE DATABASE entri_assignment;
-- Use the database
USE entri_assignment;
-- Create the table "departments"
CREATE TABLE departments (
  Department_id INT PRIMARY KEY,  Department_name VARCHAR(255),  Location_id INT);
-- Use the database
USE entri_assignment;
-- Create the table "employees"
CREATE TABLE employees ( Employee_id INT PRIMARY KEY,
  First_name VARCHAR(255),
  Last_name VARCHAR(255),
  Email VARCHAR(255),
  Phone_number VARCHAR(20),
  Hire_date DATE,
  Job_id VARCHAR(10),
  Salary DECIMAL(10, 2),
  Commission_pct DECIMAL(4, 2),
  Manager_id INT,
  Department_id INT,
  CONSTRAINT fk_department_id
    FOREIGN KEY (Department_id)
    REFERENCES departments(Department_id)
);
```

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%'



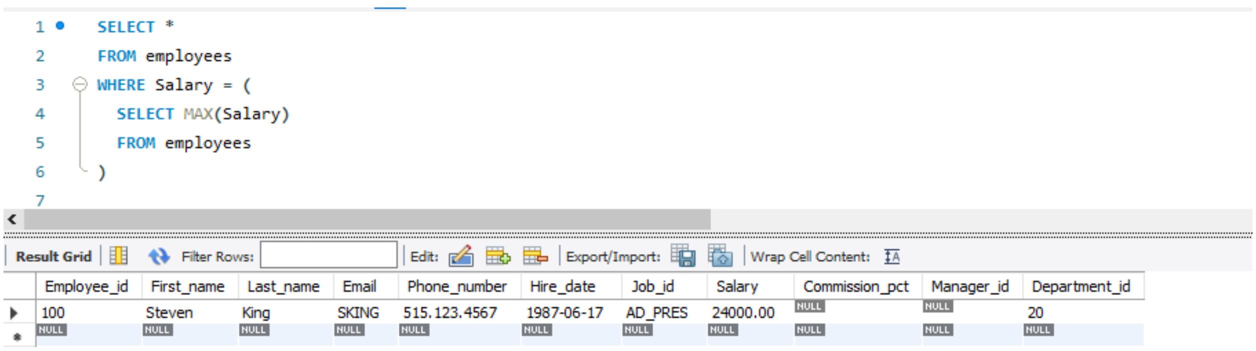
```
SELECT First_name, Last_name, Job_id, Salary
FROM employees
WHERE First_name LIKE 'S%'

```

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)

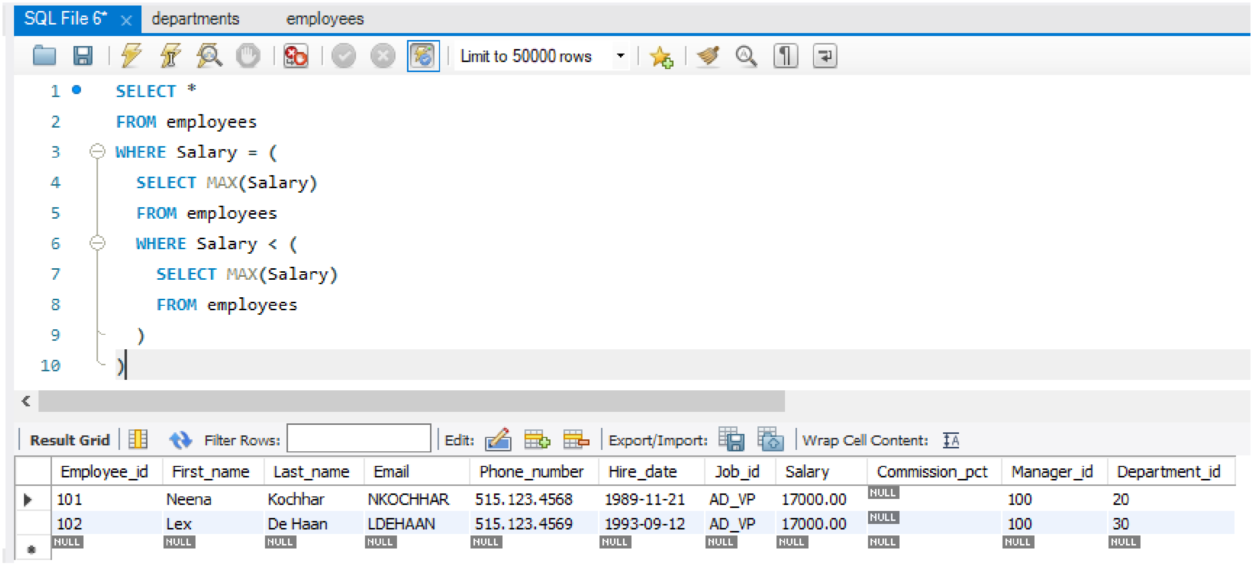


```
SELECT *
FROM employees
WHERE Salary = (
  SELECT MAX(Salary)
  FROM employees
)
```

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
  )
)
```



```
SELECT *
FROM employees
WHERE Salary = (
  SELECT MAX(Salary)
  FROM employees
  WHERE Salary < (
    SELECT MAX(Salary)
    FROM employees
  )
)
```

	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

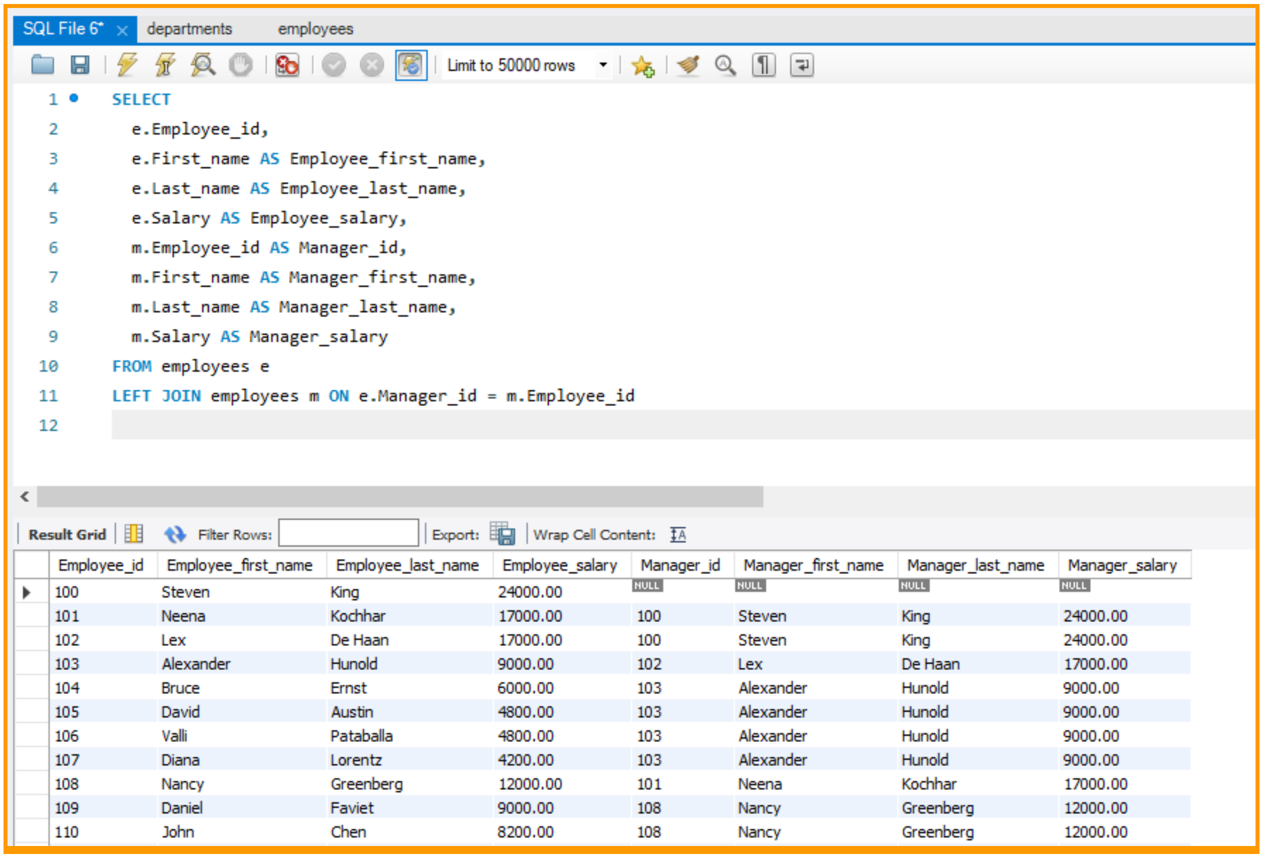
SELECT

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



```
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
```

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)

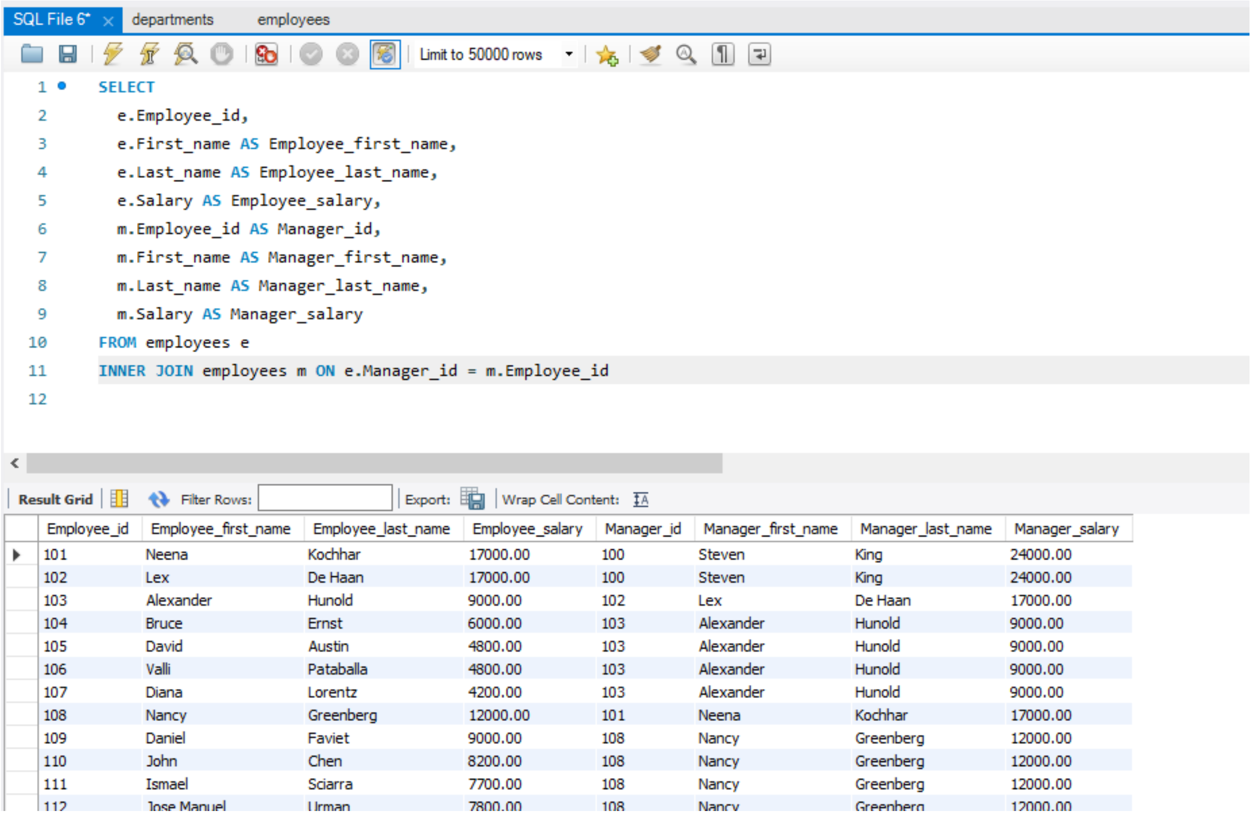
SELECT

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



```
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
```

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

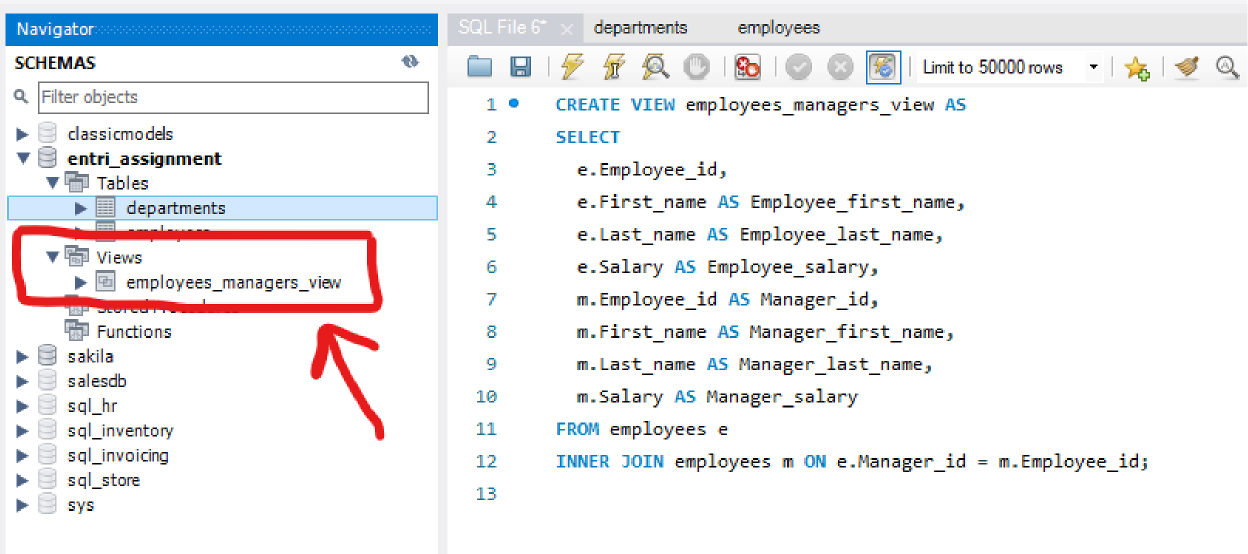
SELECT

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;



```sql
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;
```

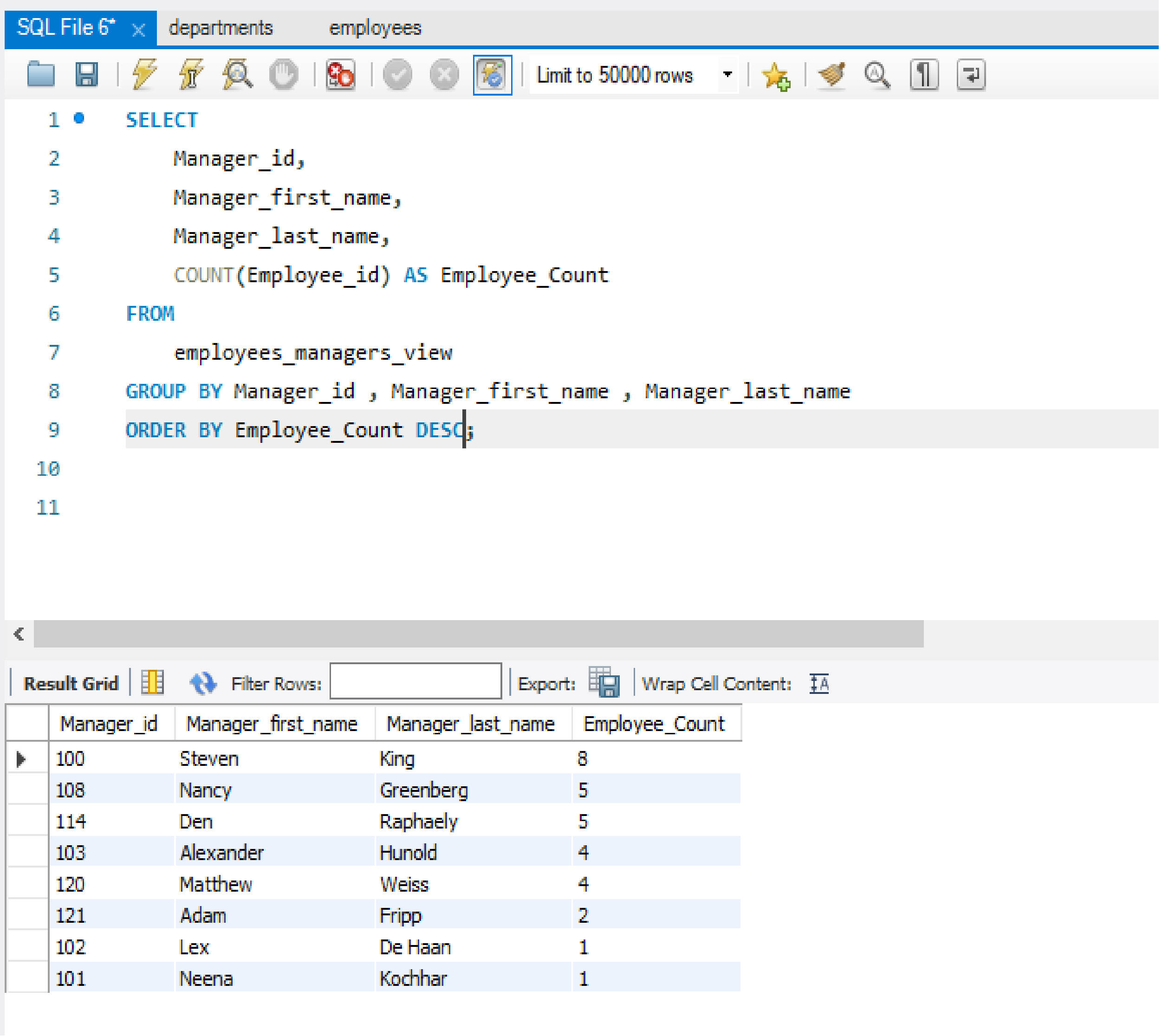
7. Write a query to show count of employees under each manager in descending order (from view)

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;



```
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;
```

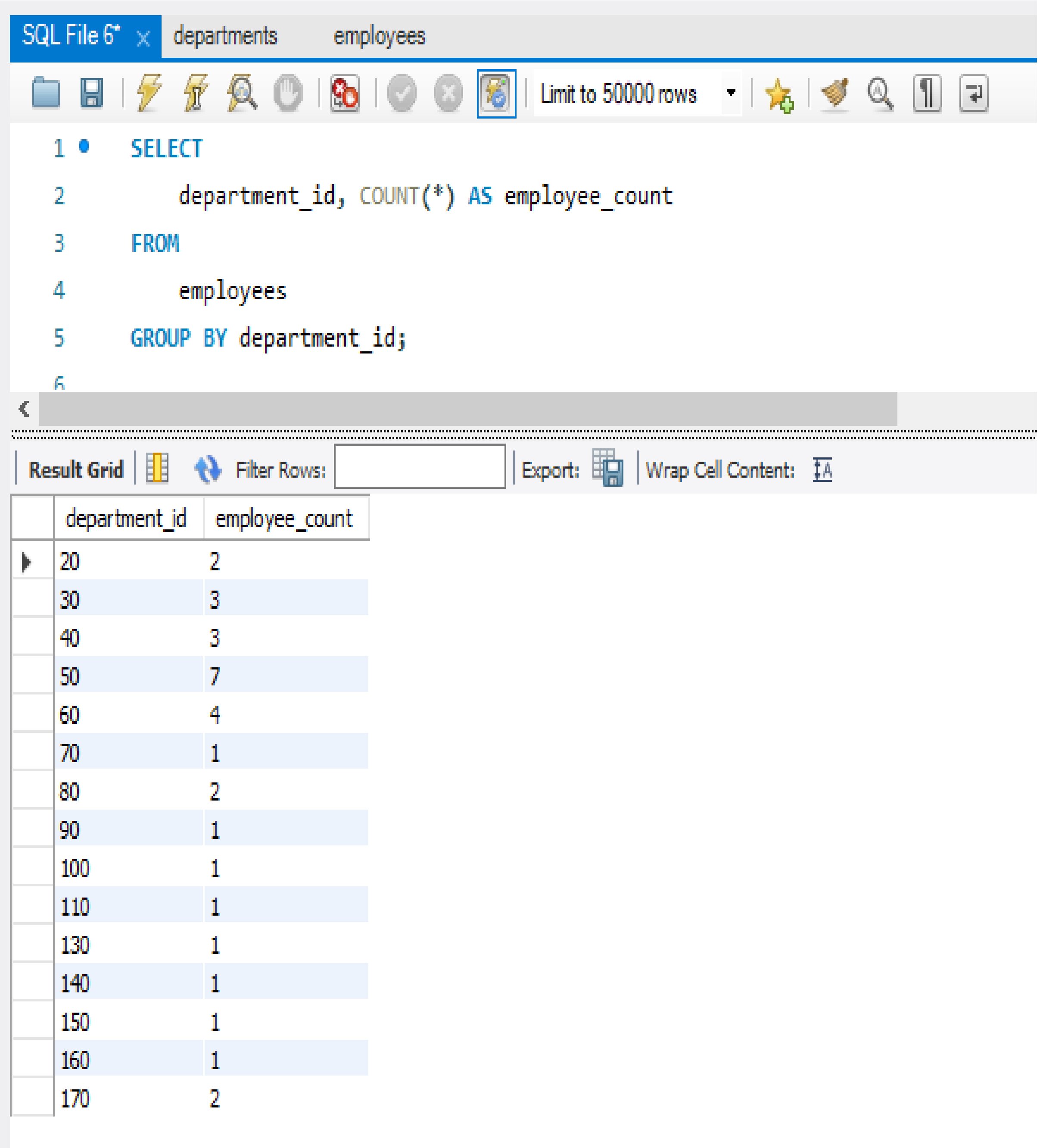
Manager_id	Manager_first_name	Manager_last_name	Employee_Count
100	Steven	King	8
108	Nancy	Greenberg	5
114	Den	Raphaely	5
103	Alexander	Hunold	4
120	Matthew	Weiss	4
121	Adam	Fripp	2
102	Lex	De Haan	1
101	Neena	Kochhar	1

8. Find the count of employees in each department

SELECT department_id, COUNT(*) AS employee_count

FROM employees

GROUP BY department_id;



```sql
SELECT
    department_id, COUNT(*) AS employee_count
FROM
    employees
GROUP BY department_id;
```

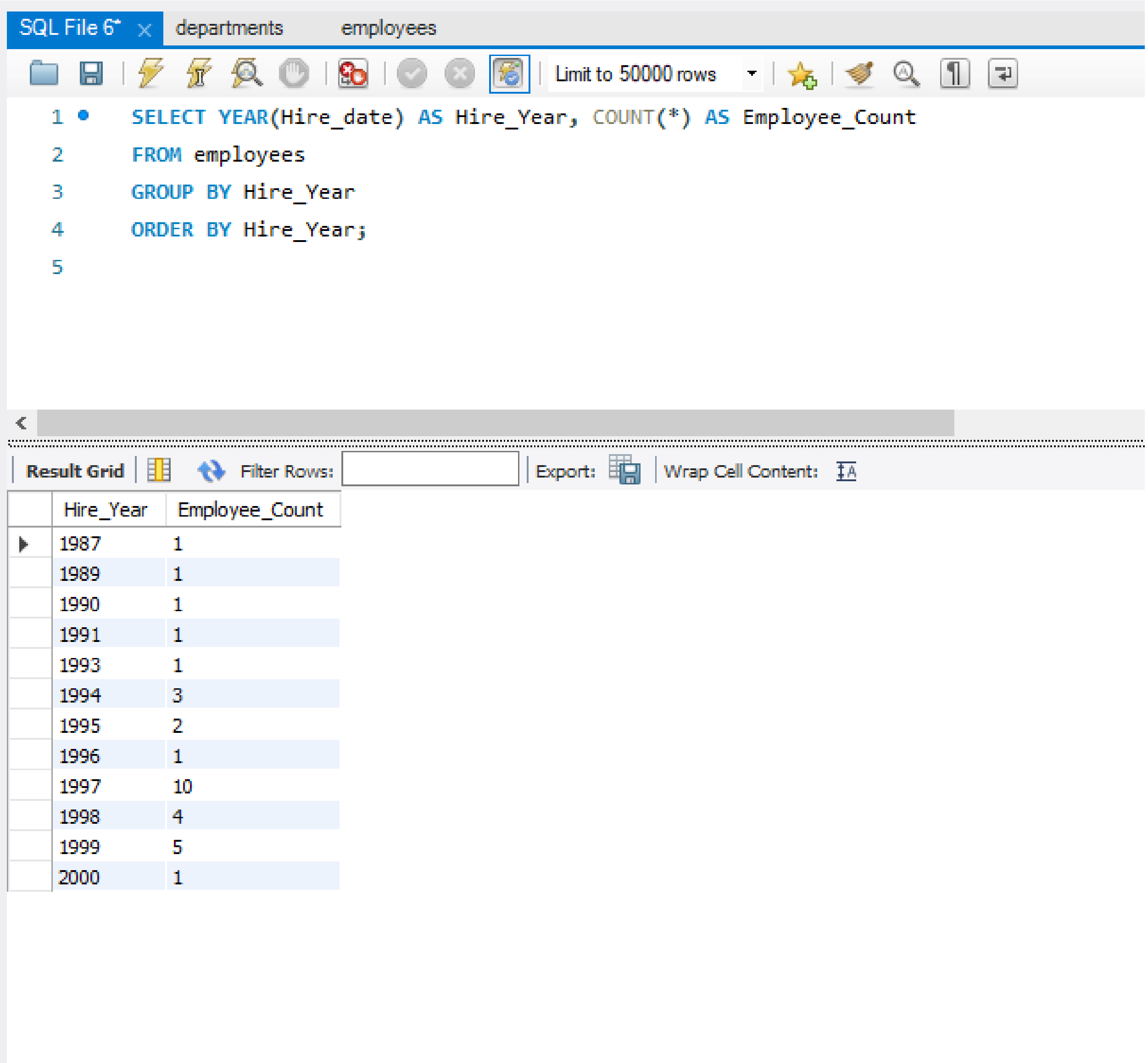
department_id	employee_count
20	2
30	3
40	3
50	7
60	4
70	1
80	2
90	1
100	1
110	1
130	1
140	1
150	1
160	1
170	2

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;



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

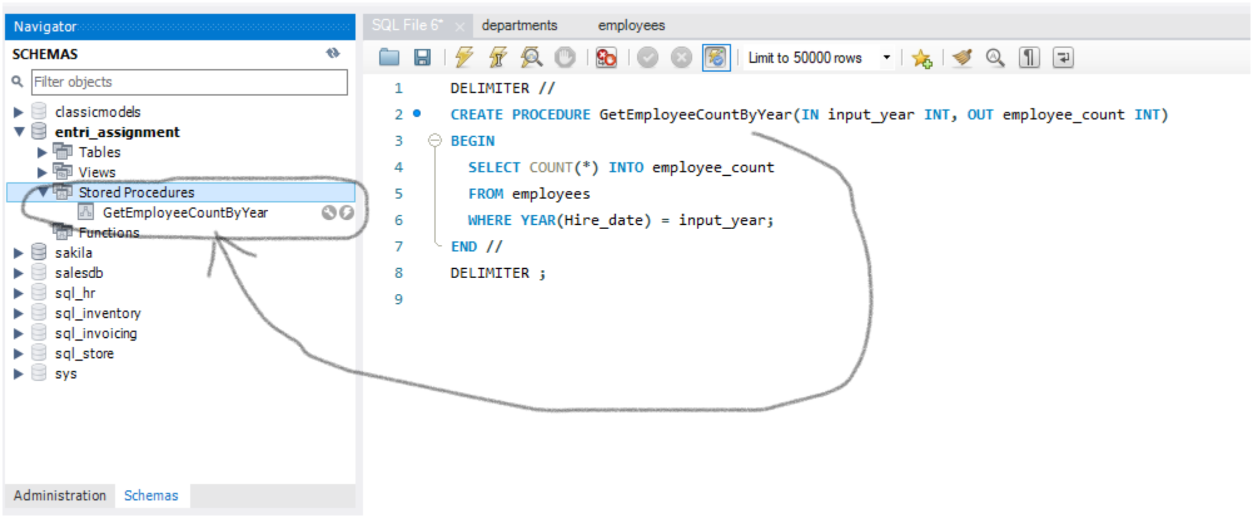
Hire_Year	Employee_Count
1987	1
1989	1
1990	1
1991	1
1993	1
1994	3
1995	2
1996	1
1997	10
1998	4
1999	5
2000	1

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

DELIMITER //

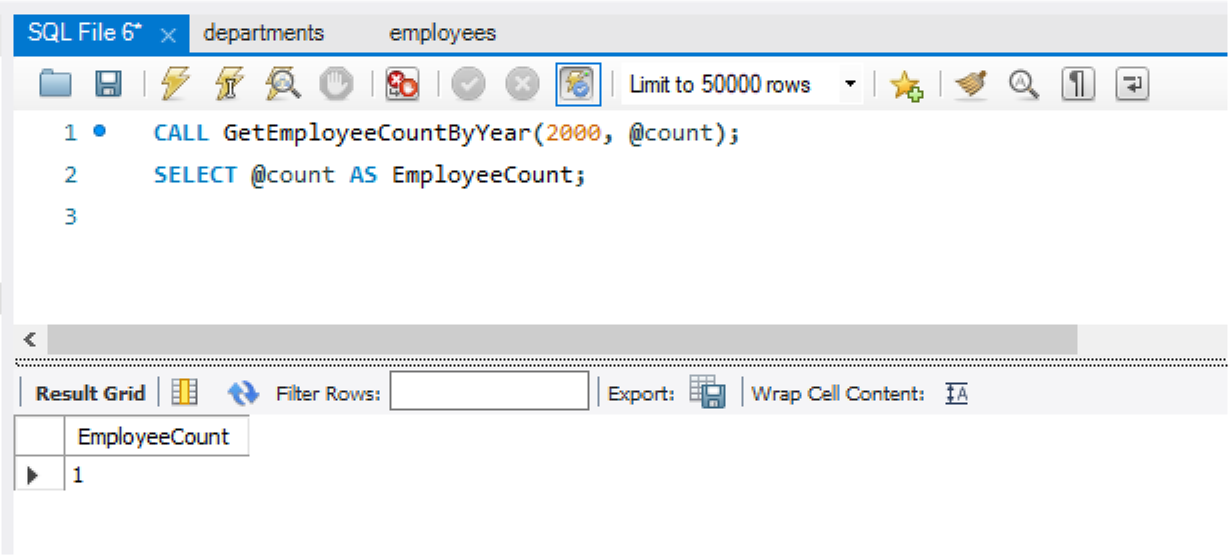
CREATE PROCEDURE GetEmployeeCountByYear(IN input_year INT, OUT employee_count INT)

```
BEGIN

 SELECT COUNT(*) INTO employee_count

 FROM employees

 WHERE YEAR(Hire_date) = input_year;

END //

DELIMITER ;
```



```
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:



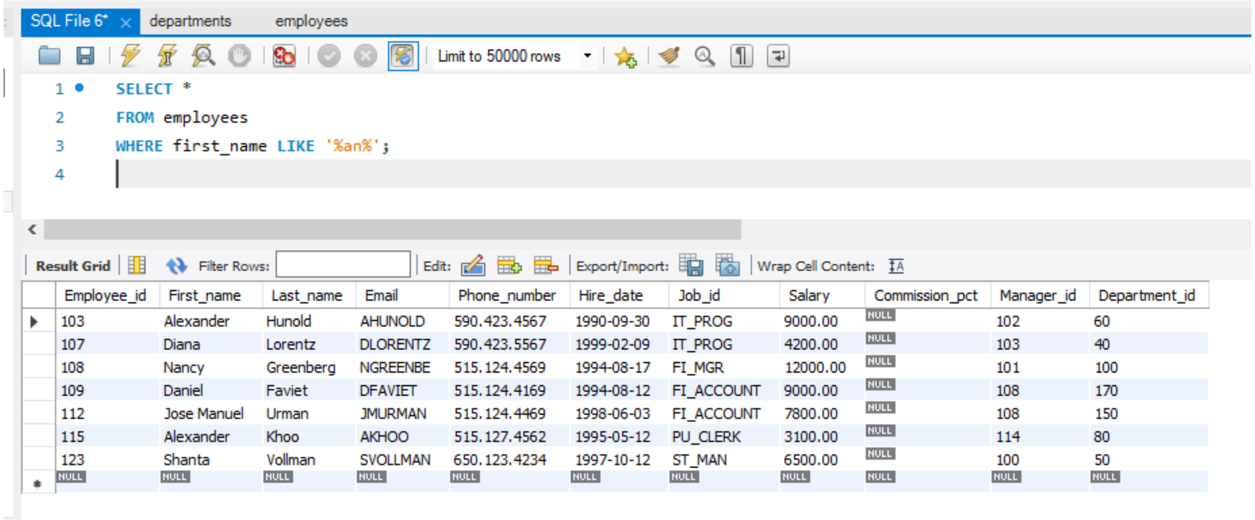
```sql
CALL GetEmployeeCountByYear(2000, @count);
SELECT @count AS EmployeeCount;
```

EmployeeCount
1

11.Select the employees whose first_name contains “an”

SELECT * FROM employees

WHERE first_name LIKE '%an%';

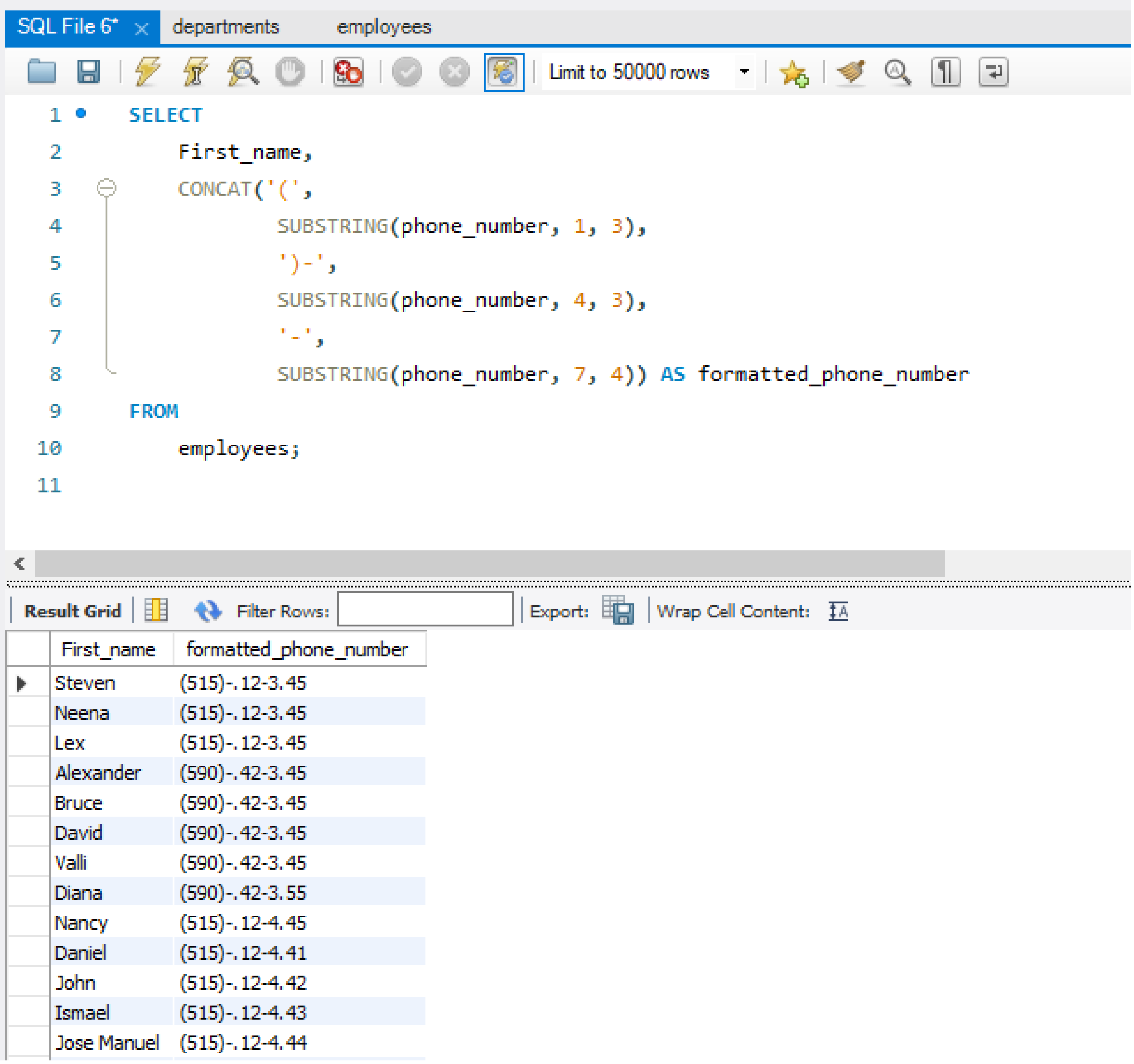


```sql
SELECT *
FROM employees
WHERE first_name LIKE '%an%';
```

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	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;
```



```
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;
```

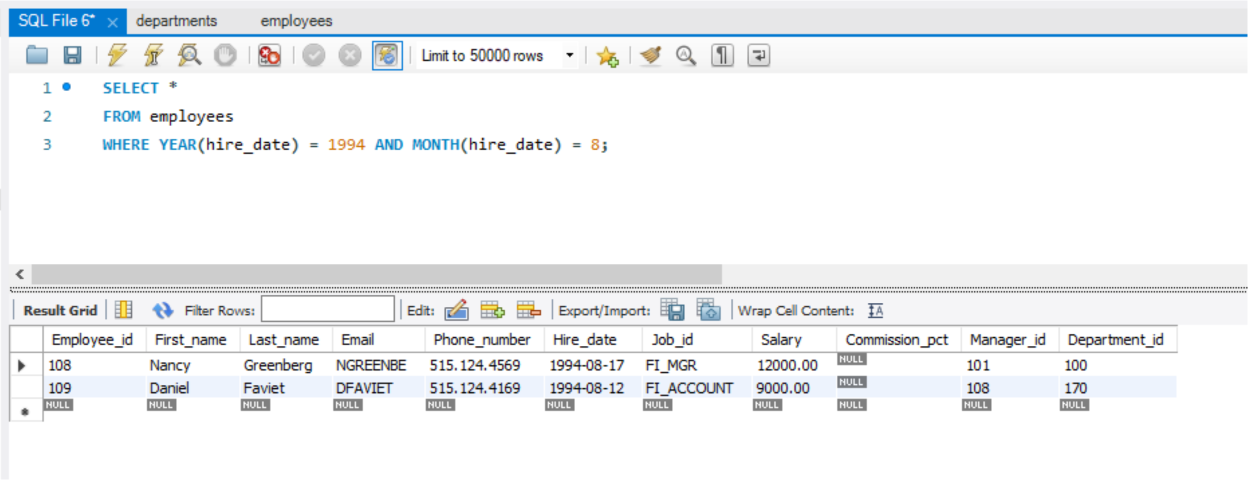
First_name	formatted_phone_number
Steven	(515)-.12-3.45
Neena	(515)-.12-3.45
Lex	(515)-.12-3.45
Alexander	(590)-.42-3.45
Bruce	(590)-.42-3.45
David	(590)-.42-3.45
Valli	(590)-.42-3.45
Diana	(590)-.42-3.55
Nancy	(515)-.12-4.45
Daniel	(515)-.12-4.41
John	(515)-.12-4.42
Ismael	(515)-.12-4.43
Jose Manuel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



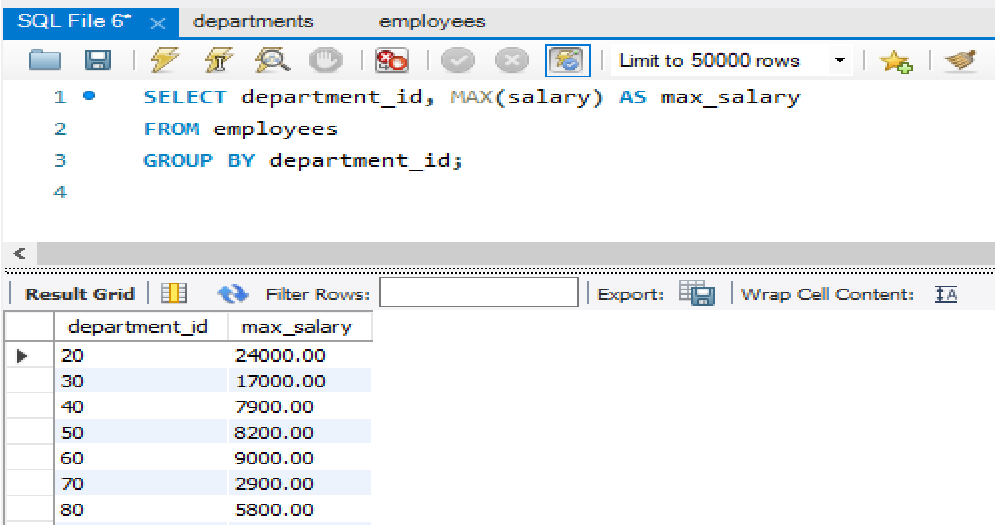
```
SELECT *
FROM employees
WHERE YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;
```

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	NULL

14. Find the maximum salary from each department.

SELECT department_id, MAX(salary) AS max_salary

FROM employees GROUP BY department_id;



```
SELECT department_id, MAX(salary) AS max_salary
FROM employees
GROUP BY department_id;
```

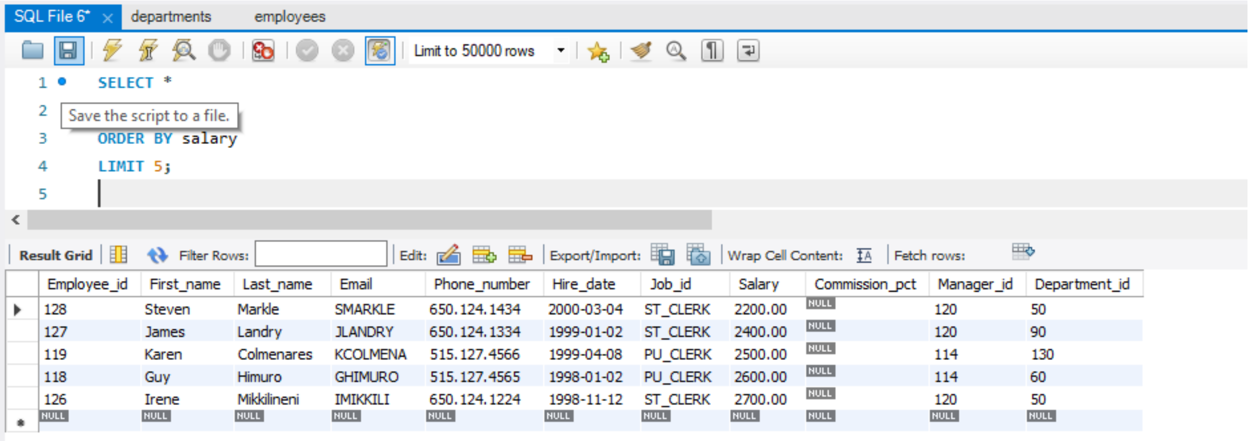
department_id	max_salary
20	24000.00
30	17000.00
40	7900.00
50	8200.00
60	9000.00
70	2900.00
80	5800.00

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

SELECT * FROM employees

ORDER BY salary

LIMIT 5;



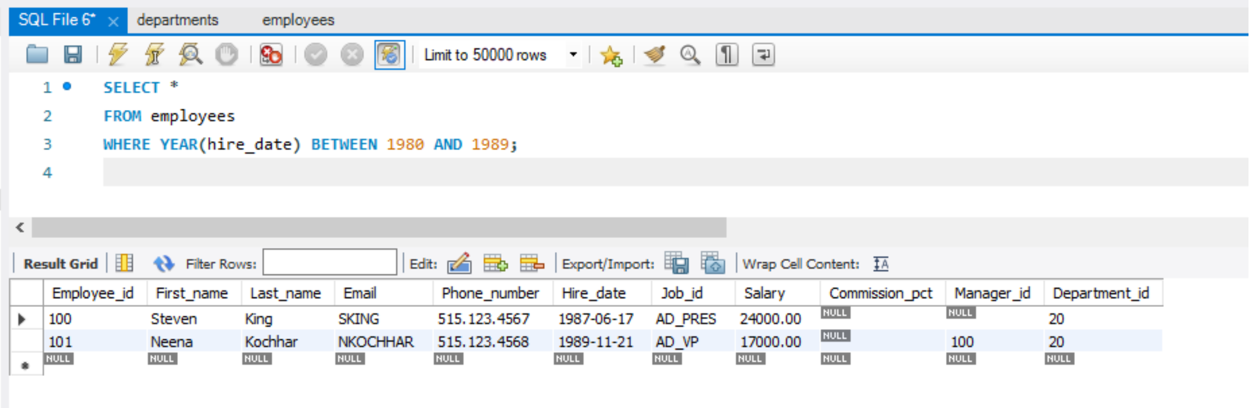
```
SELECT *

ORDER BY salary
LIMIT 5;
```

	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	KCOLMENA	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
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

16. Find the employees hired in the 80s

SELECT * FROM employees

WHERE YEAR(hire_date) BETWEEN 1980 AND 1989;



```
SELECT *
FROM employees
WHERE YEAR(hire_date) BETWEEN 1980 AND 1989;
```

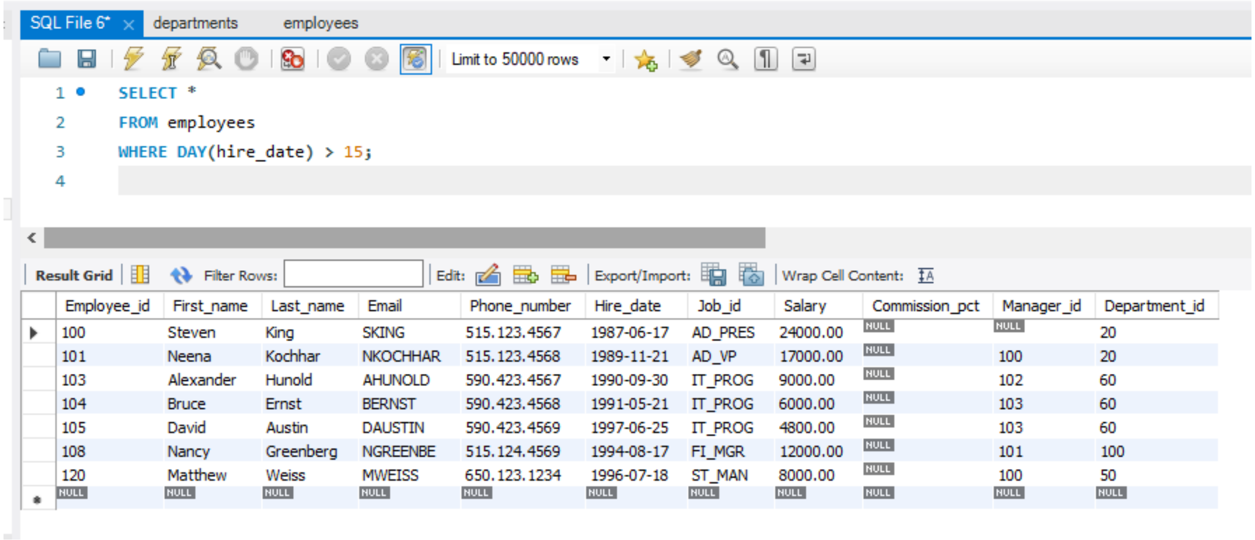
	Employee_id	First_name	Last_name	Email	Phone_number	Hire_date	Job_id	Salary	Commission_pct	Manager_id	Department_id
▶	100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	20
	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD_VP	17000.00	NULL	100	20
*	NULL	NULL	NULL	NULL	NULL	NULL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

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

SELECT *

FROM employees

WHERE DAY(hire_date) > 15;



```sql
SELECT *
FROM employees
WHERE DAY(hire_date) > 15;
```

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
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-09-30	IT_PROG	9000.00	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800.00	NULL	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18	ST_MAN	8000.00	NULL	100	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL