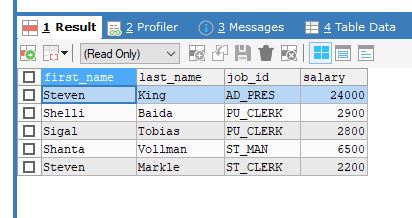
**Solve SQL Exercises (Lab Exam)**

**1. Select employees first name, last name, job\_id and salary whose  first name starts with alphabet S**

**CODE: SELECT first\_name, last\_name, job\_id, salary FROM employees WHERE first\_name LIKE 's%';**

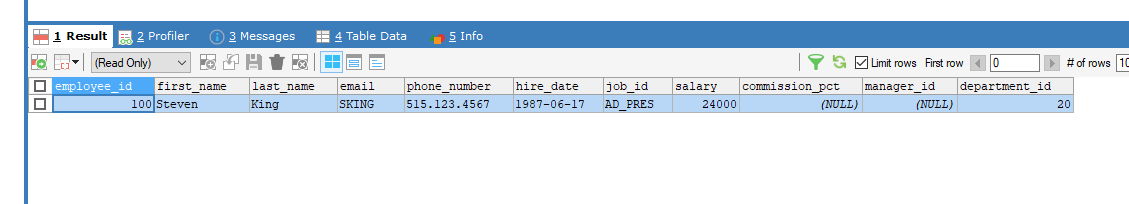
**SCREENSHOT:**

****

**2. Write a query to select employee with the highest salary**

**CODE: SELECT \*  FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);**

**SCREENSHOT:**

****

**3. Select employee with the second highest salary**

The above query selects only one person with the second-highest salary. But  what if there are more than 1 person with the same salary? Or, what if we want  to select the 3rd or 4th highest salary? So, let’s try a generic approach.

**CODE: SELECT first\_name, last\_name, salary FROM employees WHERE salary < (SELECT MAX(salary) FROM employees);**

**SCREENSHOT: **

**4. Fetch employees with 2nd or 3rd highest salary**

**CODE: SELECT first\_name, last\_name, salary FROM employees WHERE salary < 17000 ORDER BY salary DESC;**

**SCREENSHOT: **

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

Now, this is a classic example of **SELF JOIN** in SQL exercises. Also, I am  using the **CONCAT** function to concatenate the first name and last name of

**6. Write a query to show count of employees under each manager  in descending order**

**7. Find the count of employees in each department**

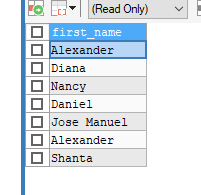
**8. Get the count of employees hired year wise**

**9. Find the salary range of employees**

**10. Write a query to divide people into three groups based on their  salaries**

**11. Select the employees whose first\_name contains “an”**

**CODE: SELECT first\_name FROM employees WHERE first\_name LIKE '%an%';**

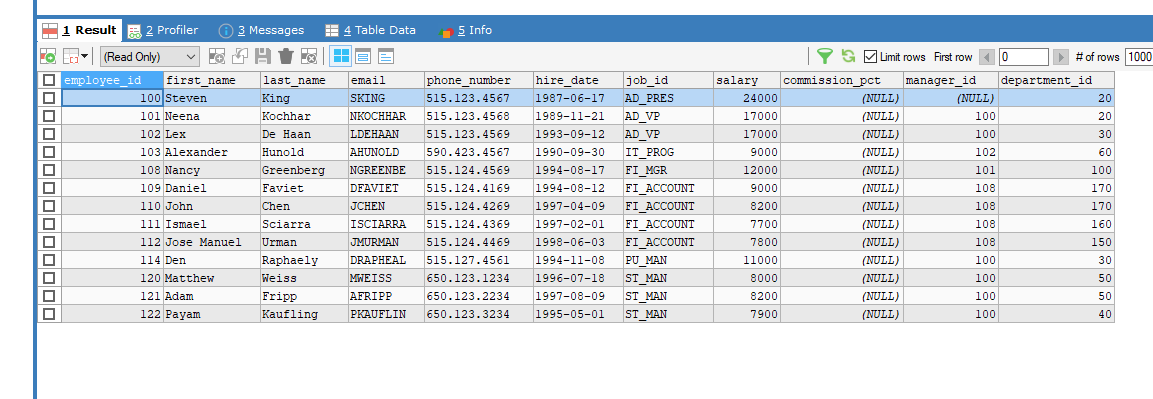
**SCREENSHOT: **

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

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

**14. Write an SQL query to display employees who earn more than  the average salary in that company**

**CODE: SELECT \* FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);**

**SCREENSHOT: **

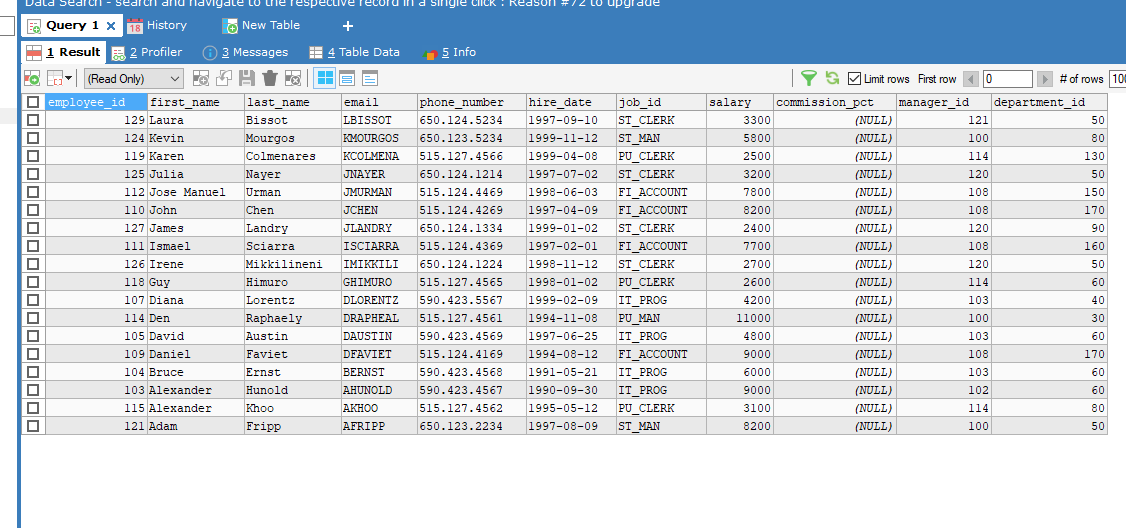
**15. Find the maximum salary from each department.**

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

**17. Find the employees hired in the 80s**

**18. Display the employees first name and the name in reverse order**

**CODE: SELECT \* FROM EMPLOYEES ORDER BY first\_name DESC;**

**SCREENSHOT: **

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

**20. Display the managers and the reporting employees who work  in different departments**