EXPERIMENT -7

PRN : 22510064 ,S4,PARSHWA HERWADE

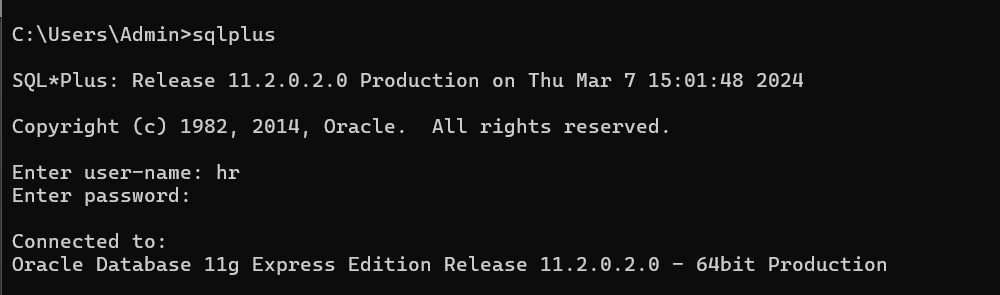
Subject : Database Engineering Lab (6CS274)

Topic : Implementation of Practice zones of introduction to Oracle 9i SQL vol-1.

Practice 1(Introduction)

1. Initiate an iSQL\*Plus session using the user ID and password provided by the instructor.

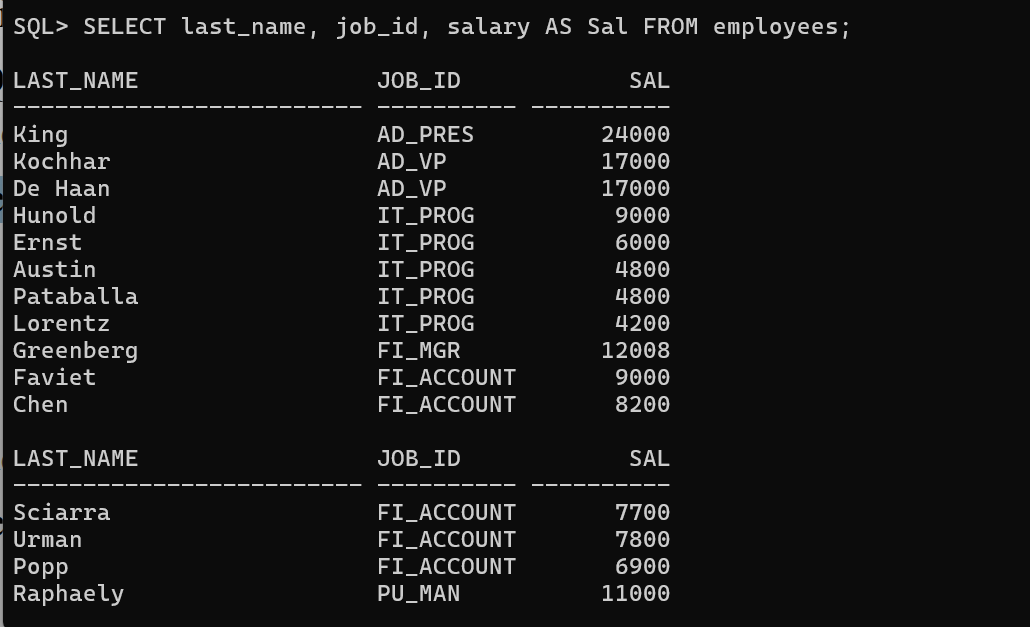
Query + Result :



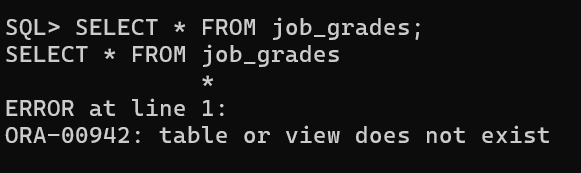
2. SQL\*Plus commands access the database. True/False  
Answer : TRUE

3. The following SELECT statement executes successfully: SELECT last\_name, job\_id, salary AS Sal FROM employees;

Answer : TRUE



4. The following SELECT statement executes successfully: SELECT \* FROM job\_grades; True/False

Ans: False  


5. There are four coding errors in this statement. Can you identify them? SELECT employee\_id, last\_name sal x 12 ANNUAL SALARY FROM employees;

Ans :

1)From should be written after sal

2)A comma is missing after the last\_name

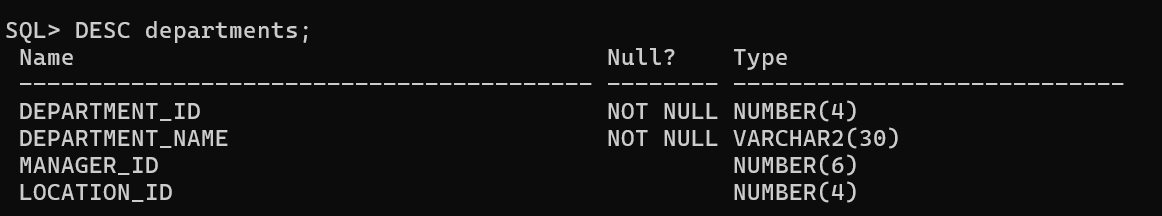
3)x 12 is redundant

4)There is no semicolon at the end of the query

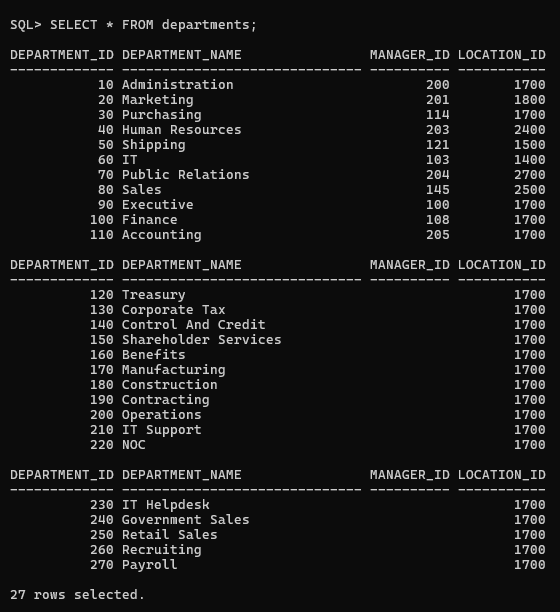
6. Show the structure of the DEPARTMENTS table. Select all data from the table.

Ans :

Structure :



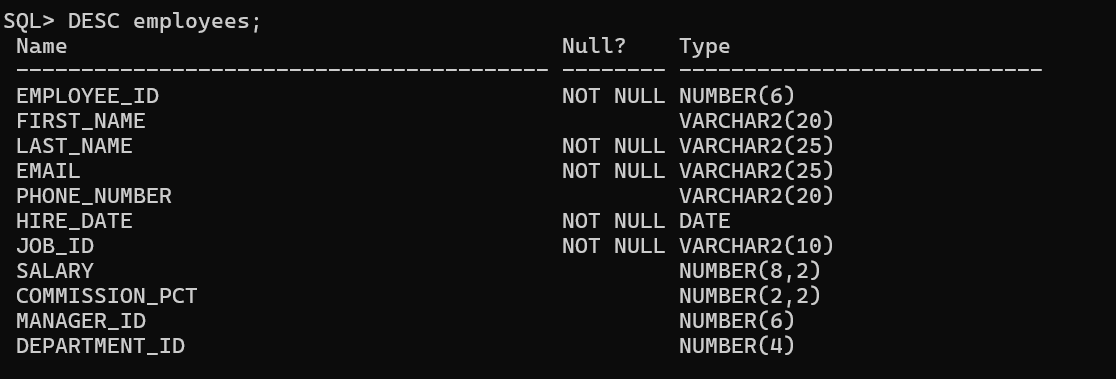
Data



7. Show the structure of the EMPLOYEES table. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first. Provide an alias STARTDATE for the HIRE\_DATE column. Save your SQL statement to a file named lab1\_7.sql.

Ans :

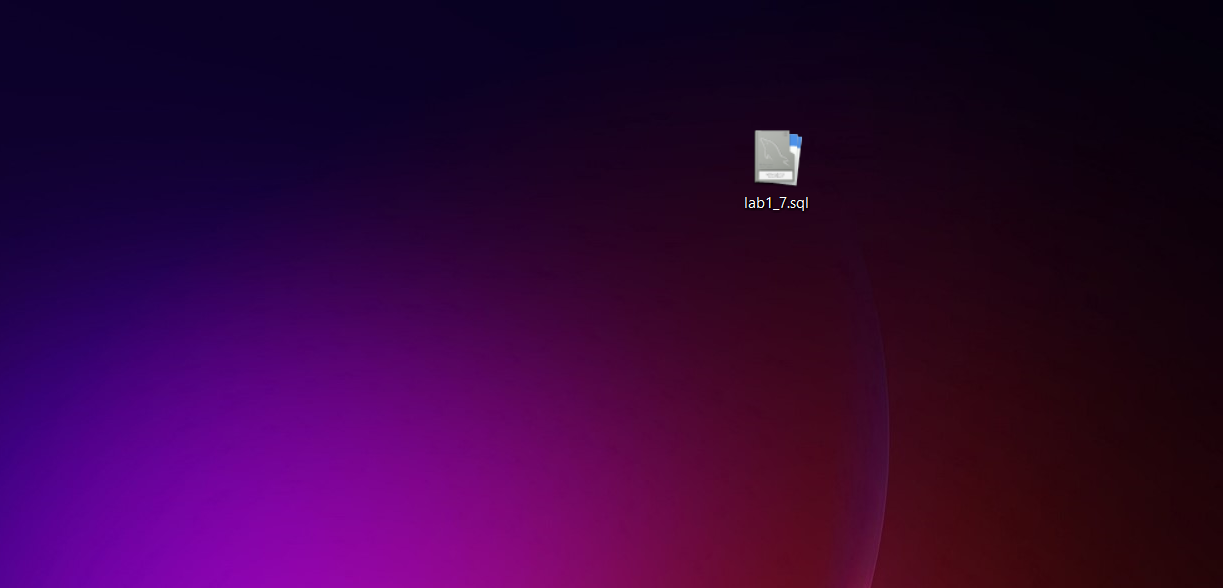
Structure of Employees->



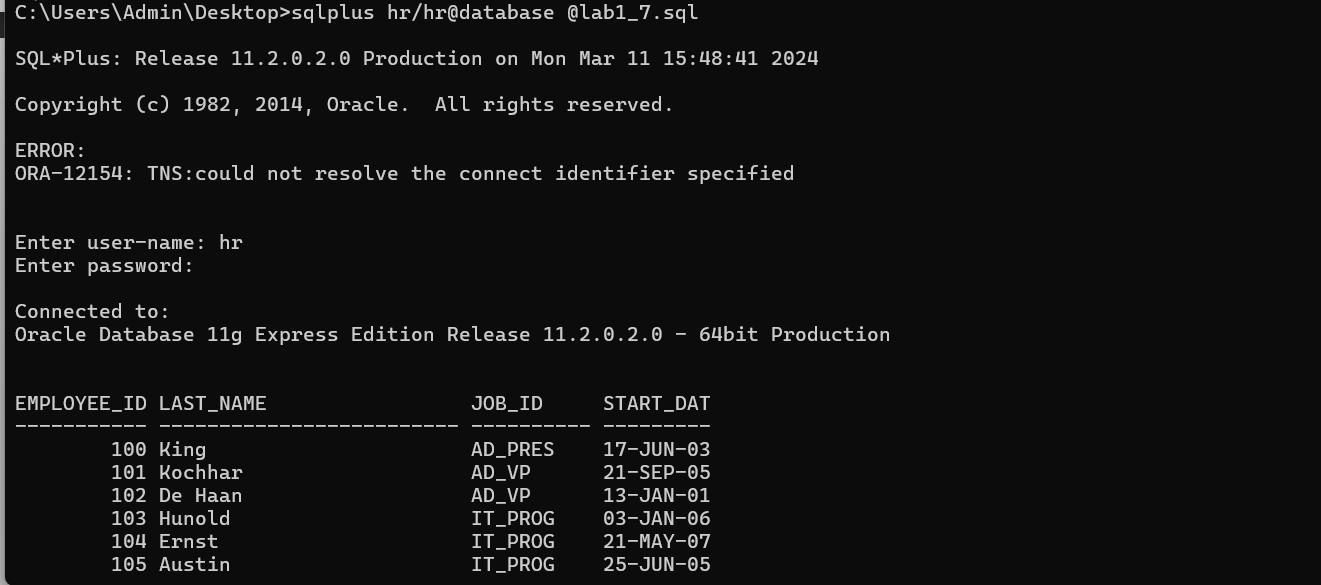
Specific Query->



Saving into lab1\_7.sql file by creating it on the desktop



8. Run your query in the file lab1\_7.sql



9. Create a query to display unique job codes from the EMPLOYEES table.



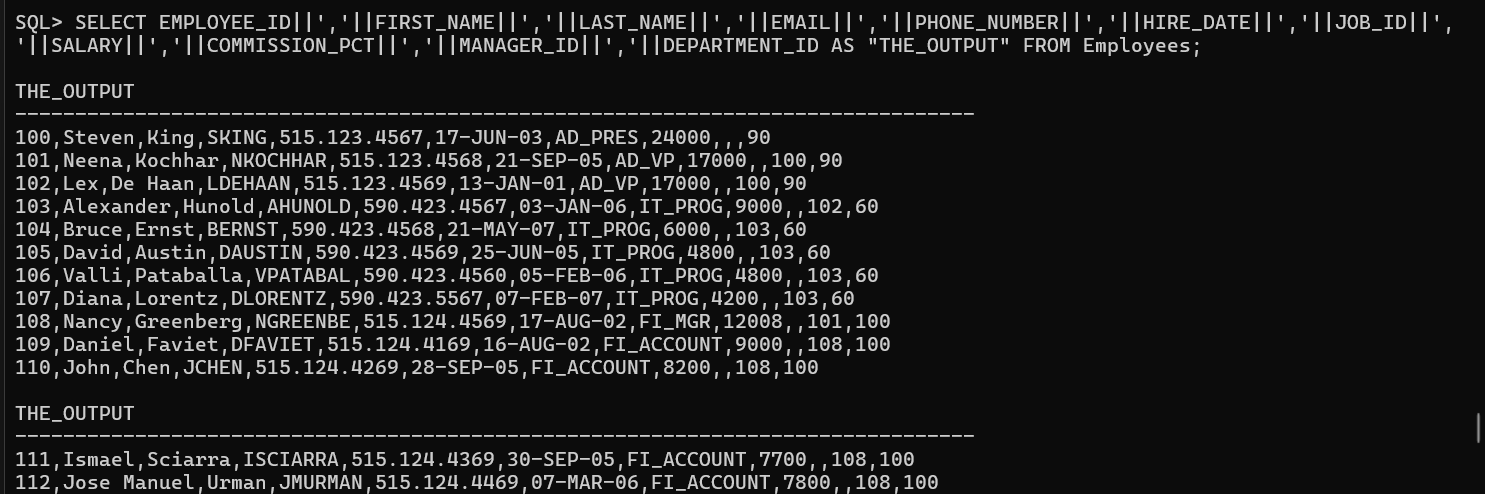
10. Copy the statement from lab1\_7.sql into the iSQL\*Plus Edit window. Name the column headings Emp #, Employee, Job, and Hire Date, respectively. Run your query again



11. Display the last name concatenated with the job ID, separated by a comma and space, and name the column Employee and Title.

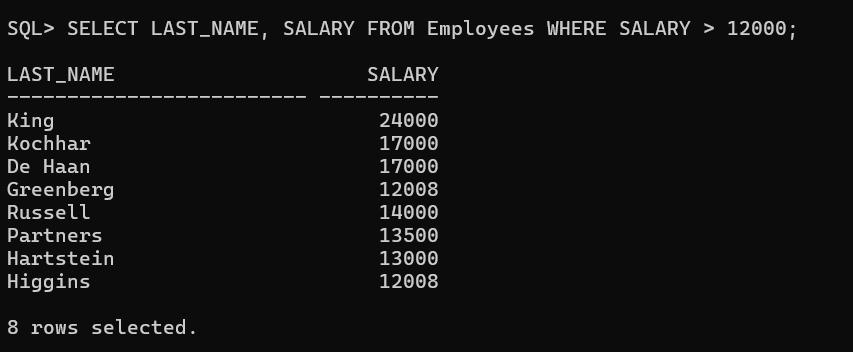


12. Create a query to display all the data from the EMPLOYEES table. Separate each column by a comma. Name the column THE\_OUTPUT.

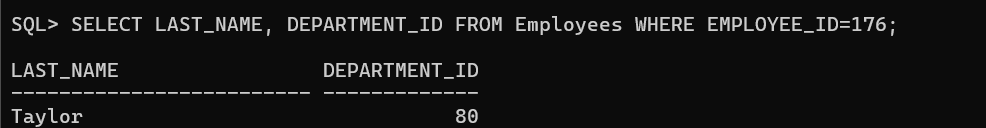


Practice-2(Restricting and Storing Data)

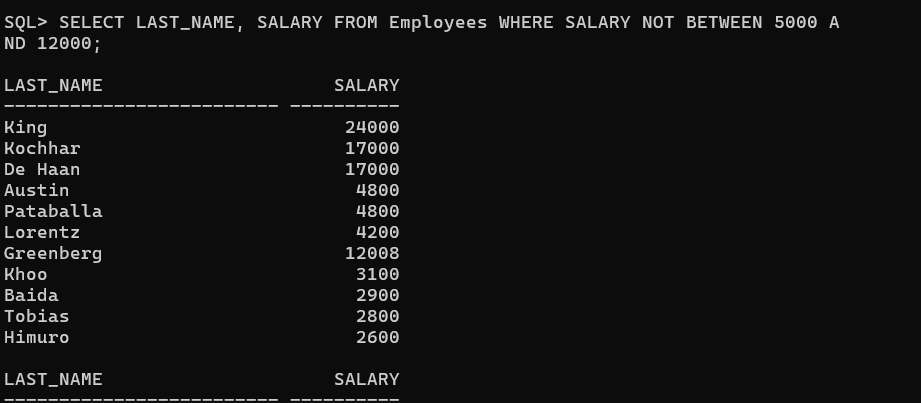
1. Create a query to display the last name and salary of employees earning more than $12,000. Place your SQL statement in a text file named lab2\_1.sql. Run your query.



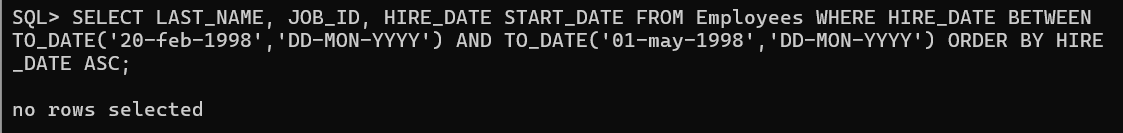
1. Create a query to display the employee last name and department number for employee number 176.



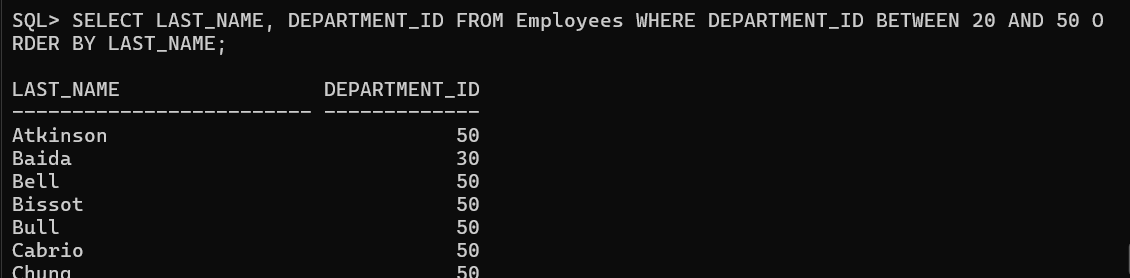
3. Modify lab2\_1.sql to display the last name and salary for all employees whose salary is not in the range of $5,000 and $12,000. Place your SQL statement in a text file named.



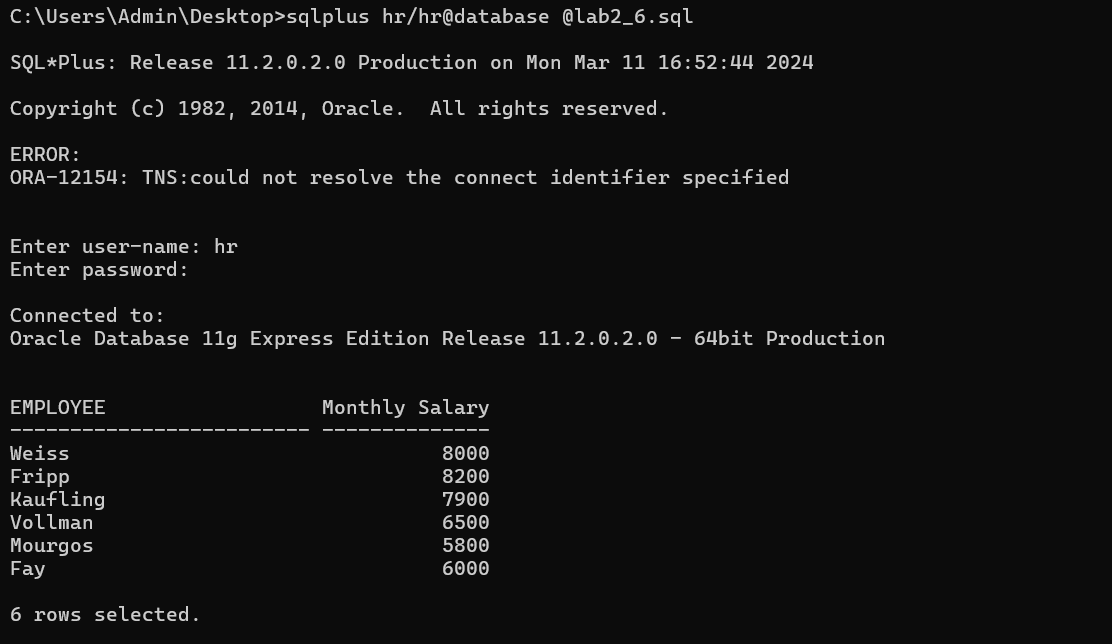
4.Display the employee last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date.



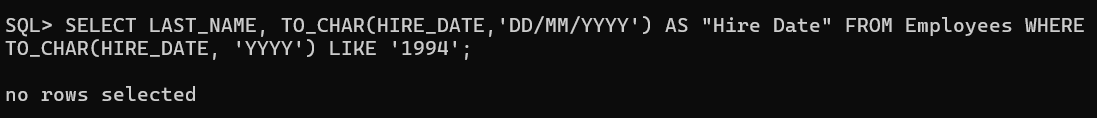
5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.



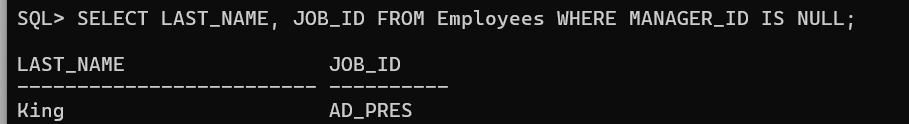
6. Modify lab2\_3.sql to list the last name and salary of employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Resave lab2\_3.sql as lab2\_6.sql. Run the statement in lab2\_6.sql.



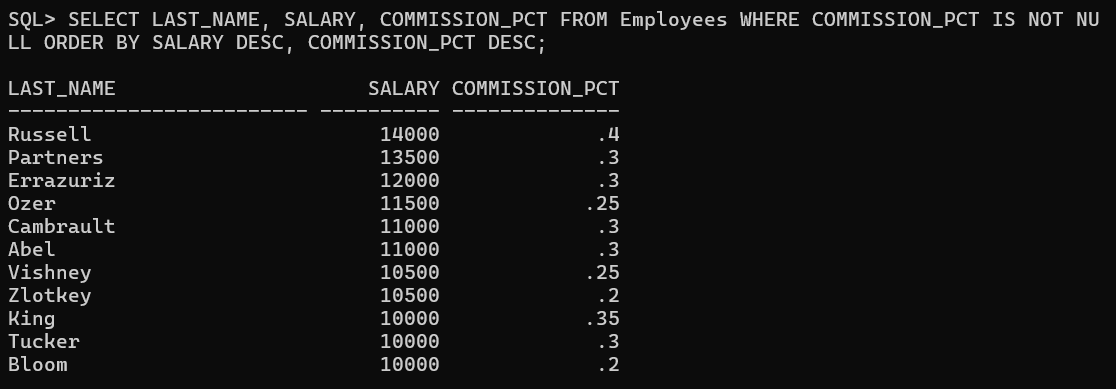
7.Display the last name and hire date of every employee who was hired in 1994.



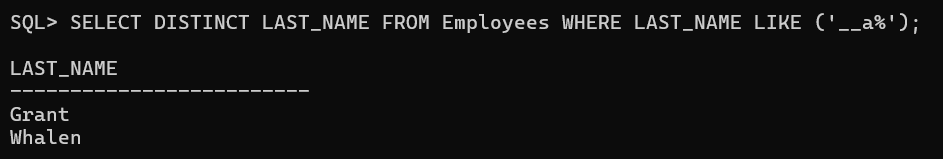
8. Display the last name and job title of all employees who do not have a manager.



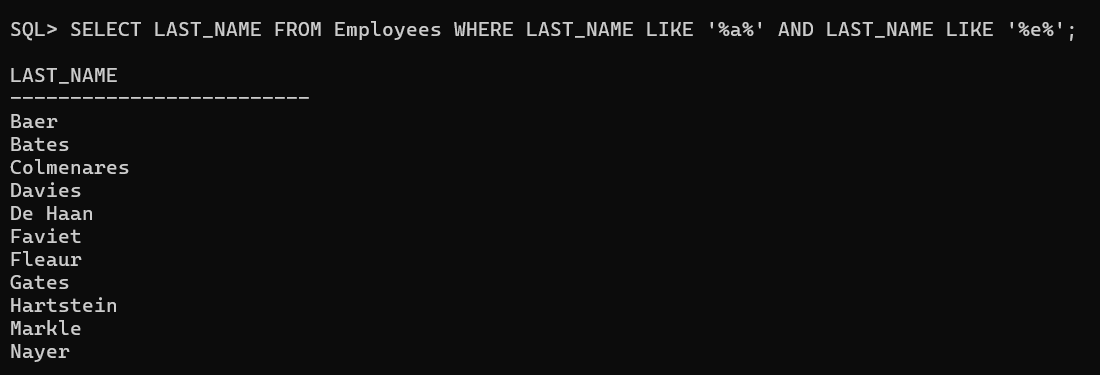
9. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions



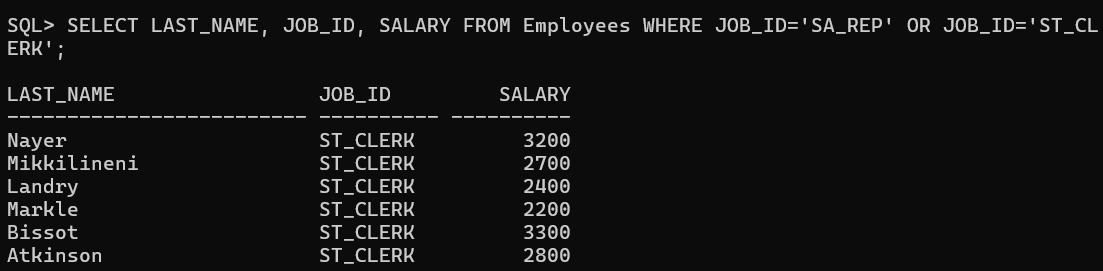
10. Display the last names of all employees where the third letter of the name is an a.



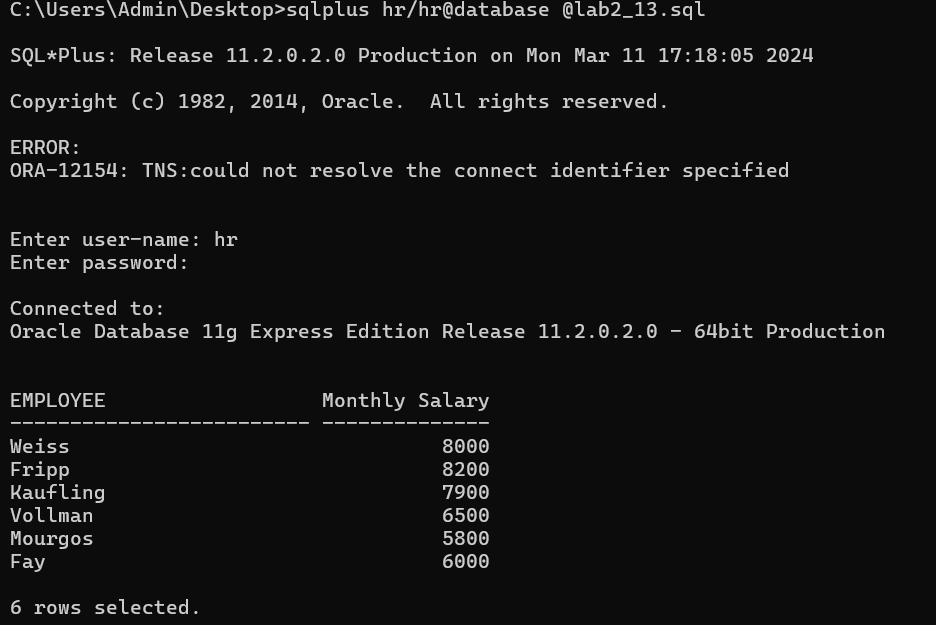
11. Display the last name of all employees who have an a and an e in their last name.



12. Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to $2,500, $3,500, or $7,000.

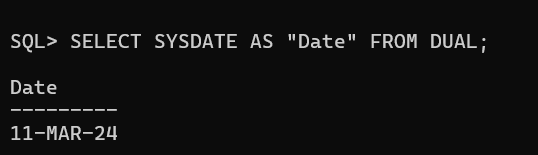


13.Modify lab2\_6.sql to display the last name, salary, and commission for all employees whose commission amount is 20%. Resave lab2\_6.sql as lab2\_13.sql. Rerun the statement in lab2\_13.sql.

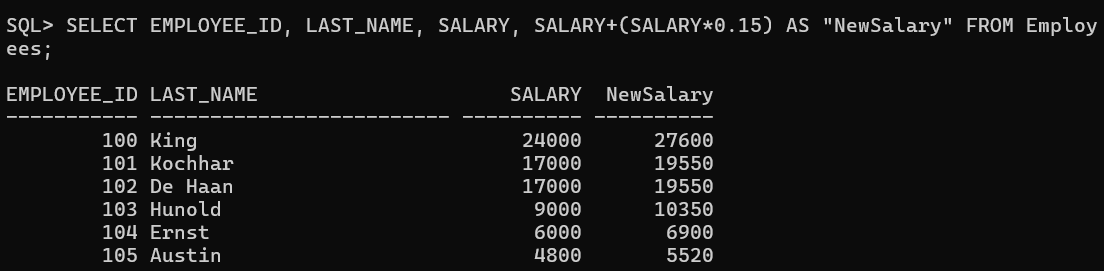


Practice Set-3(Single-row Functions)

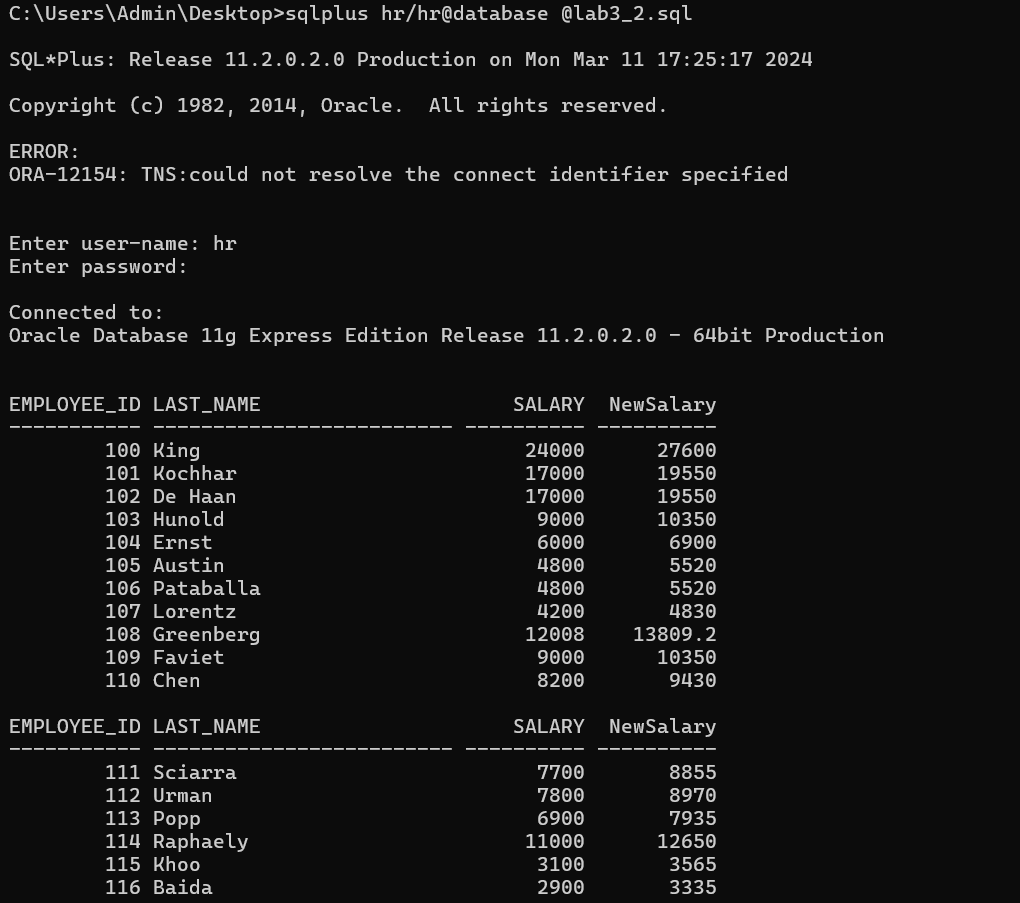
1. Write a query to display the current date. Label the column Date.



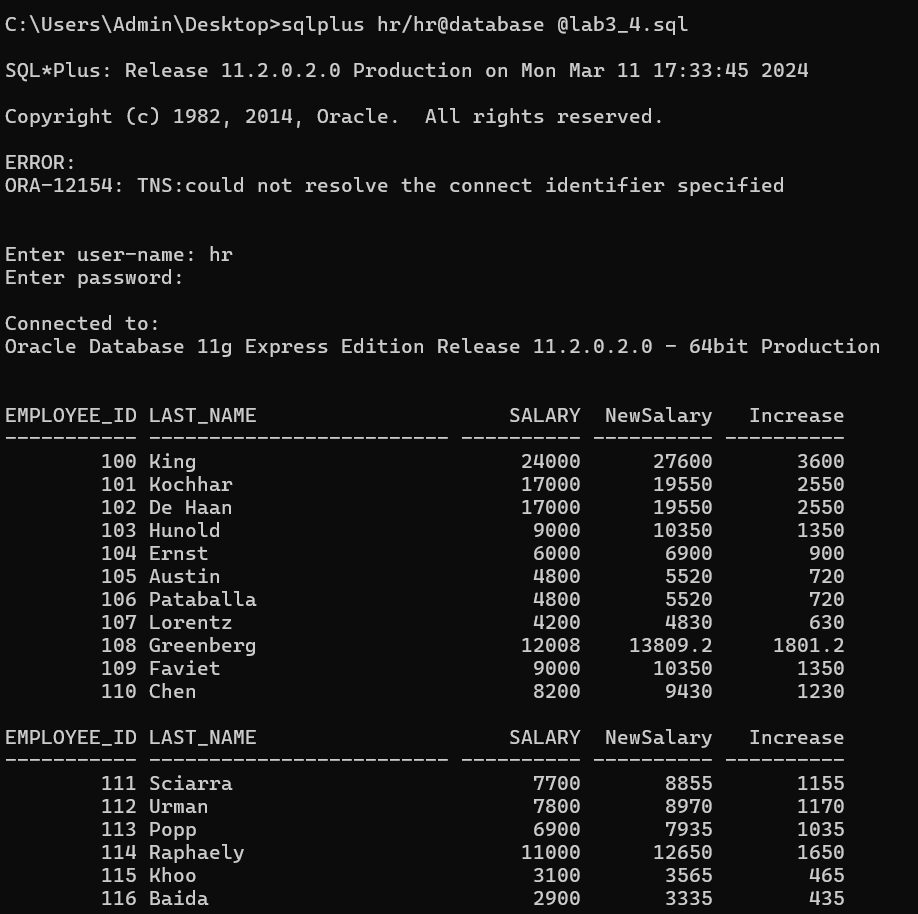
2.For each employee, display the employee number, last\_name, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary. Place your SQL statement in a text file named lab3\_2.sql.



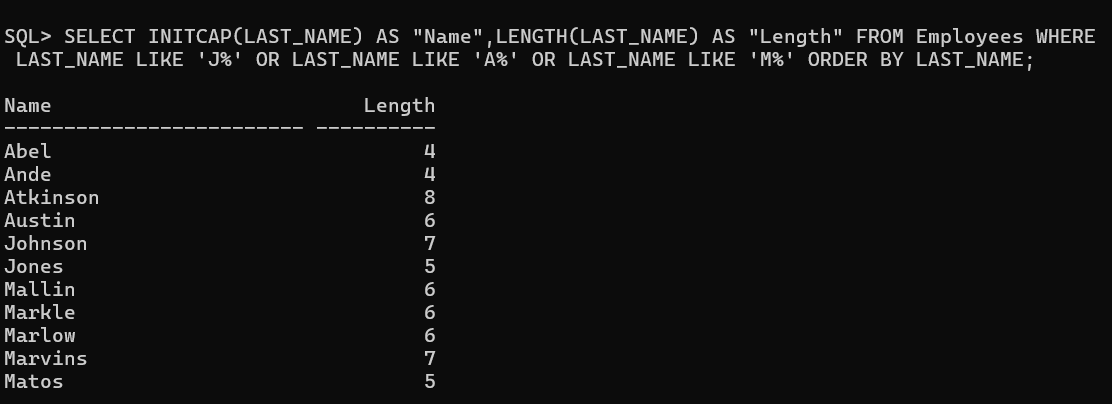
3. Run your query in the file lab3\_2.sql.



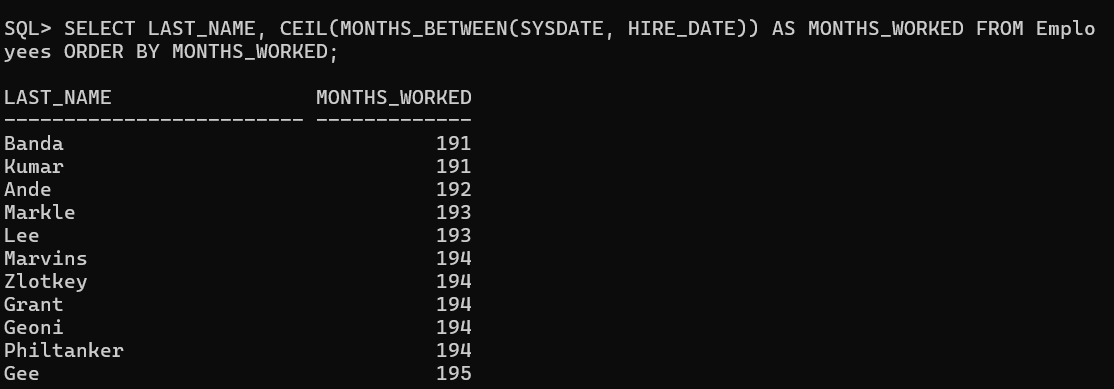
4. Modify your query lab3\_2.sql to add a column that subtracts the old salary from the new salary. Label the column Increase. Save the contents of the file as lab3\_4.sql. Run the revised query.



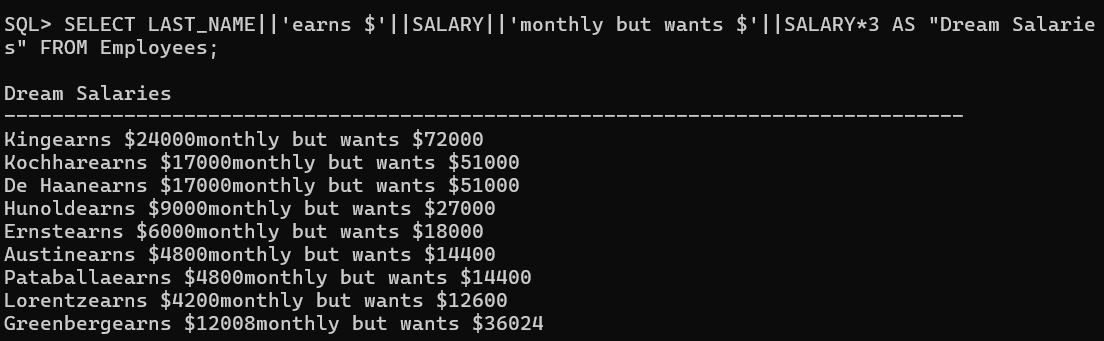
5. Write a query that displays the employee’s last names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees’ last names.



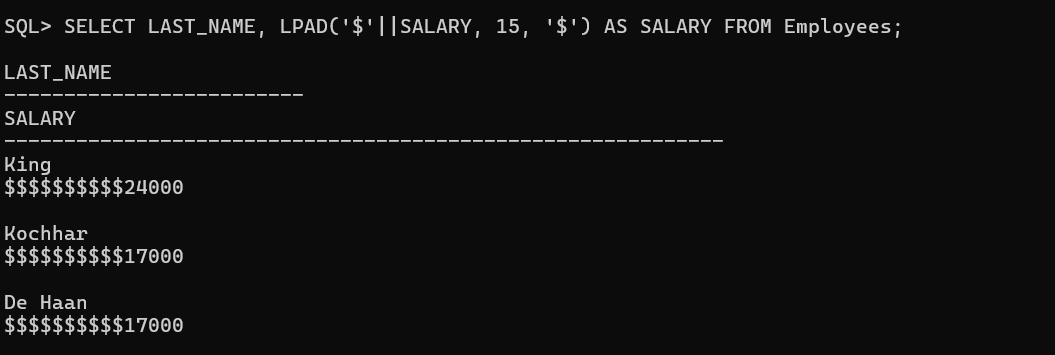
6. For each employee, display the employee’s last name, and calculate the number of months between today and the date the employee was hired. Label the column MONTHS\_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.



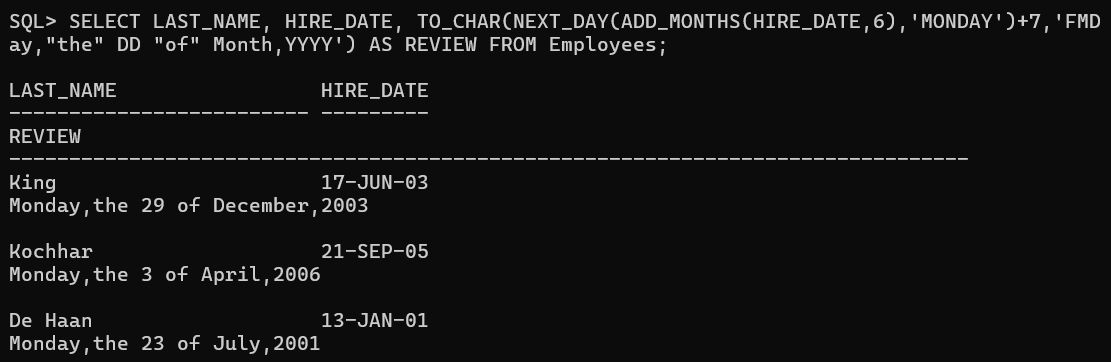
7. Write a query that produces the following for each employee: <employee last\_name> earns<salary> monthly but wants<3 times salary> . Label the column Dream Salaries.



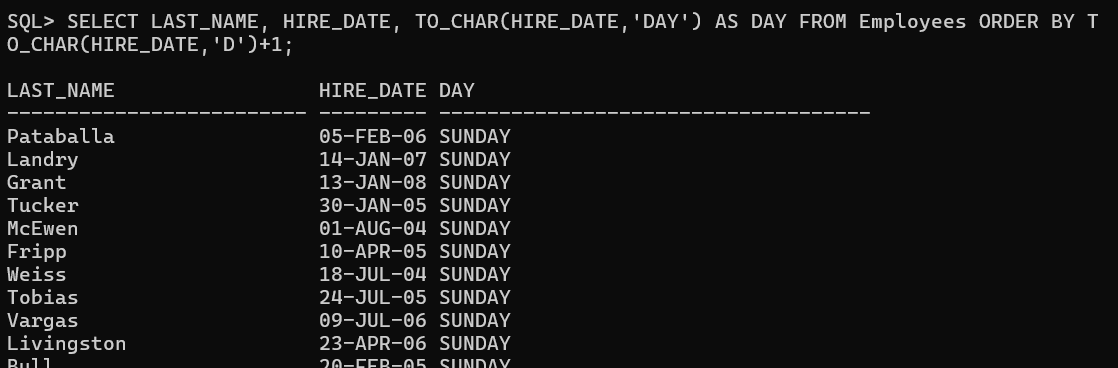
8. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with $. Label the column SALARY

0

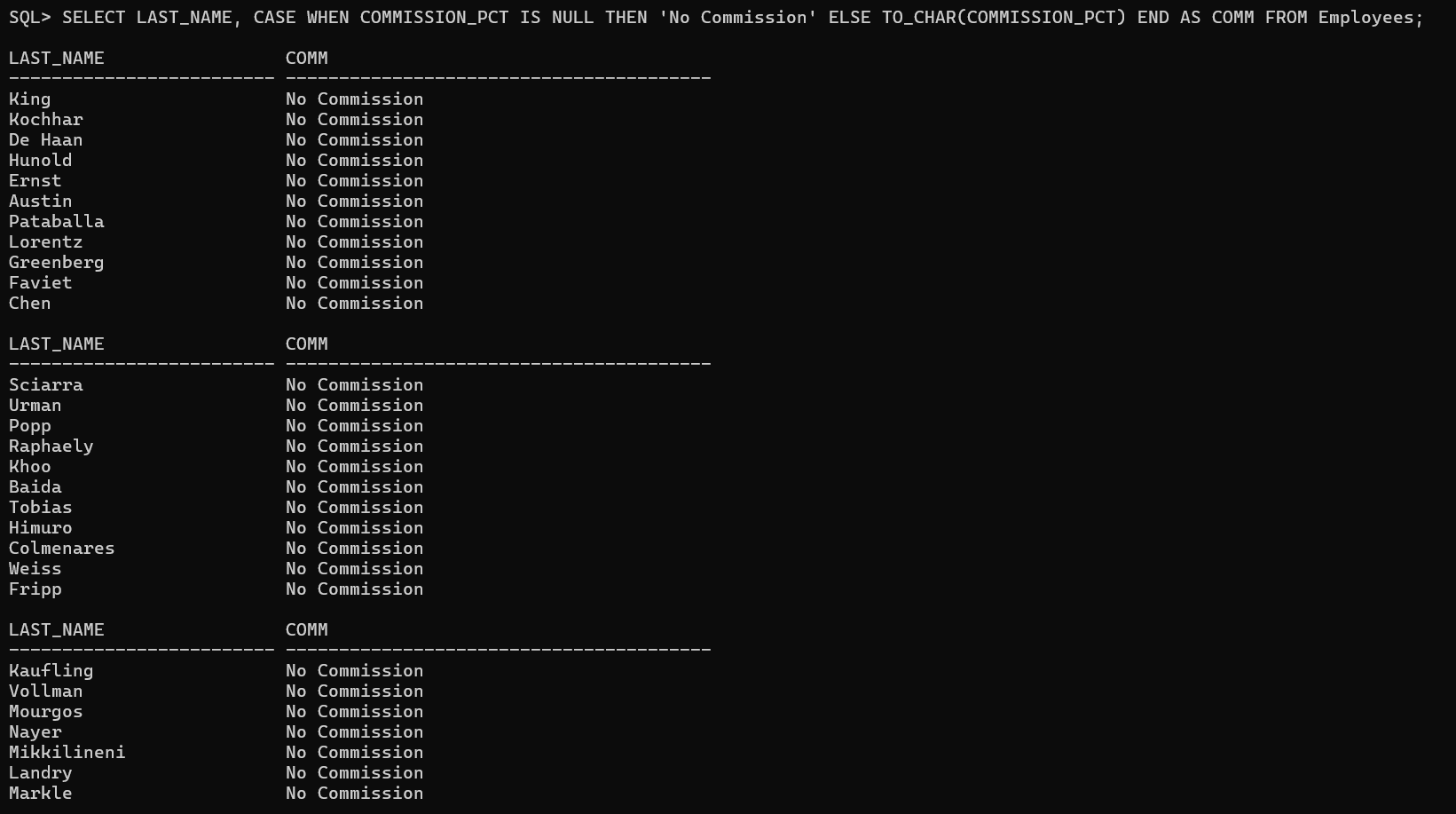
9. Display each employee’s last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to “Monday, the Thirty-First of July, 2000.”



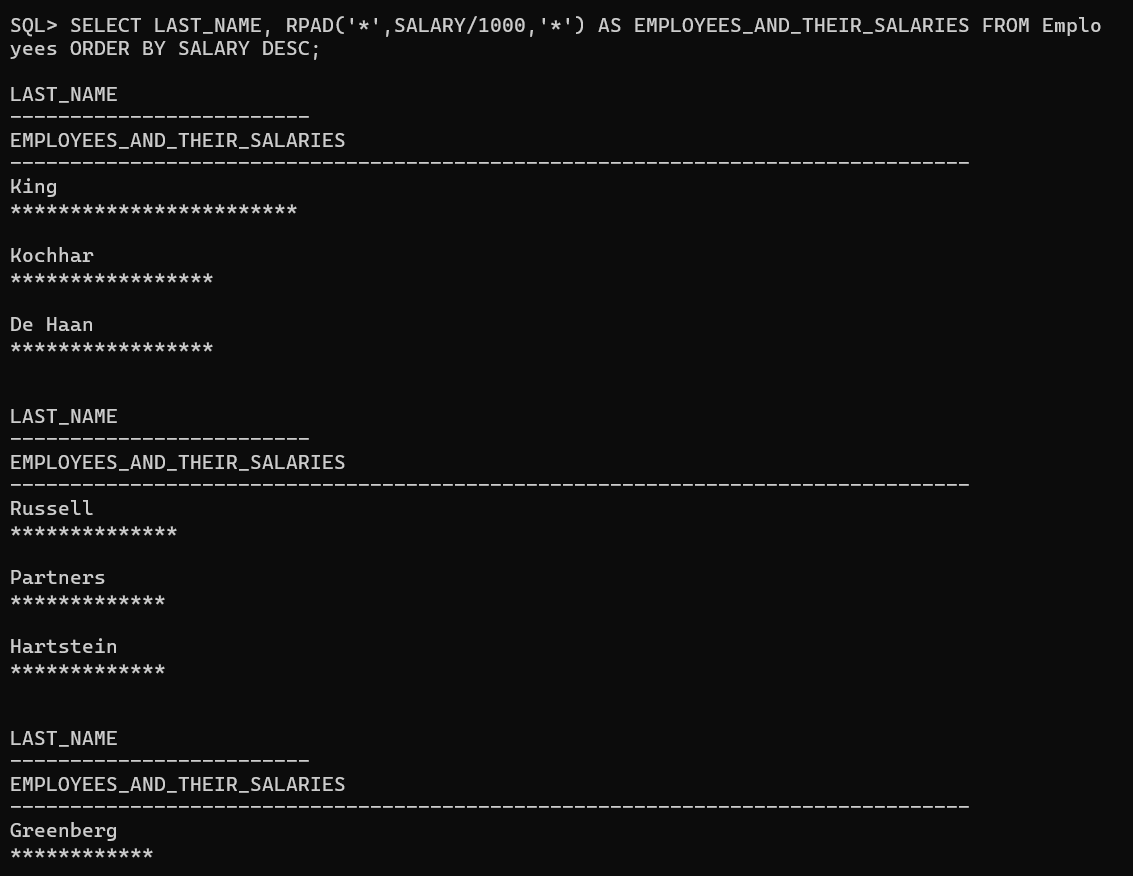
10. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week starting with Monday.



11. Create a query that displays the employees’ last names and commission amounts. If an employee does not earn commission, put “No Commission.” Label the column COMM.



12. Create a query that displays the employees’ last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES\_AND\_THEIR\_SALARIES.



13. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB\_ID, as per the following data:

Job Grade

AD\_PRES A

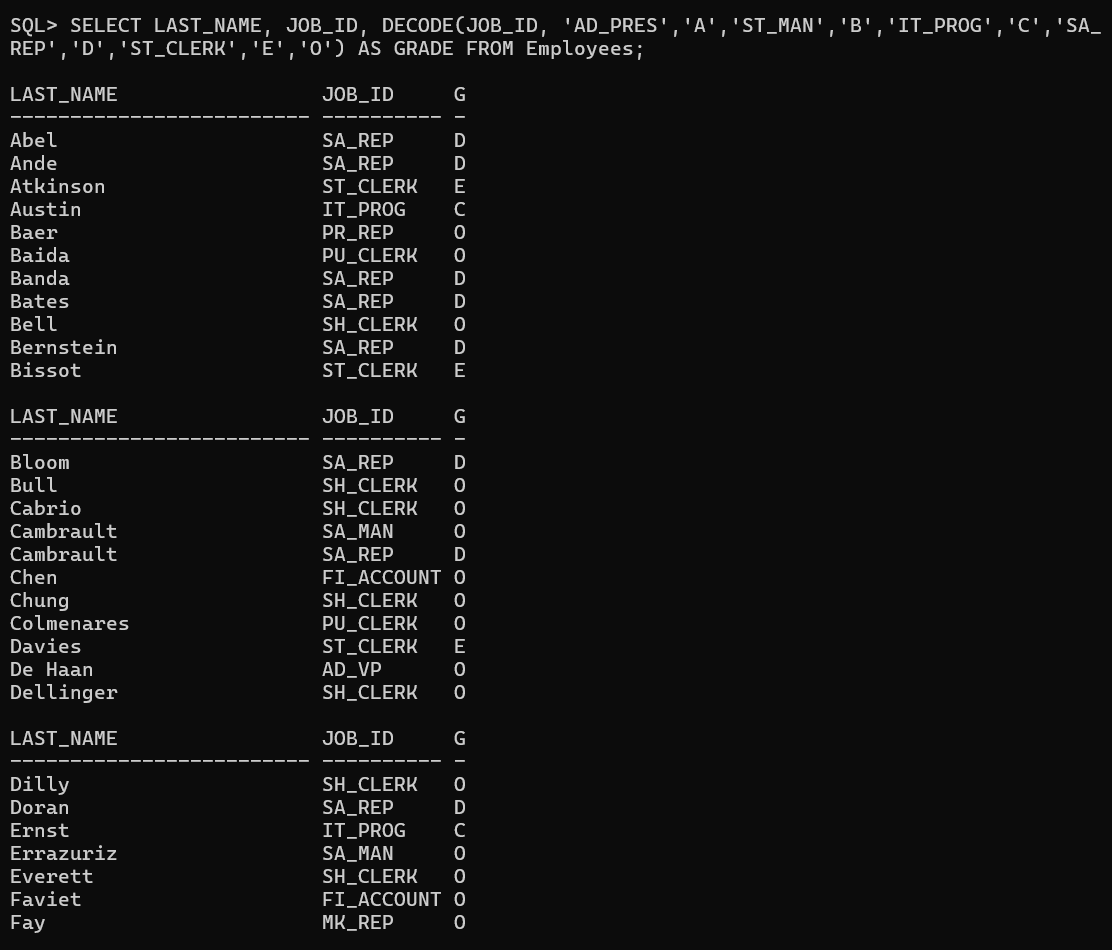
ST\_MAN B

IT\_PROG C

SA\_REP D

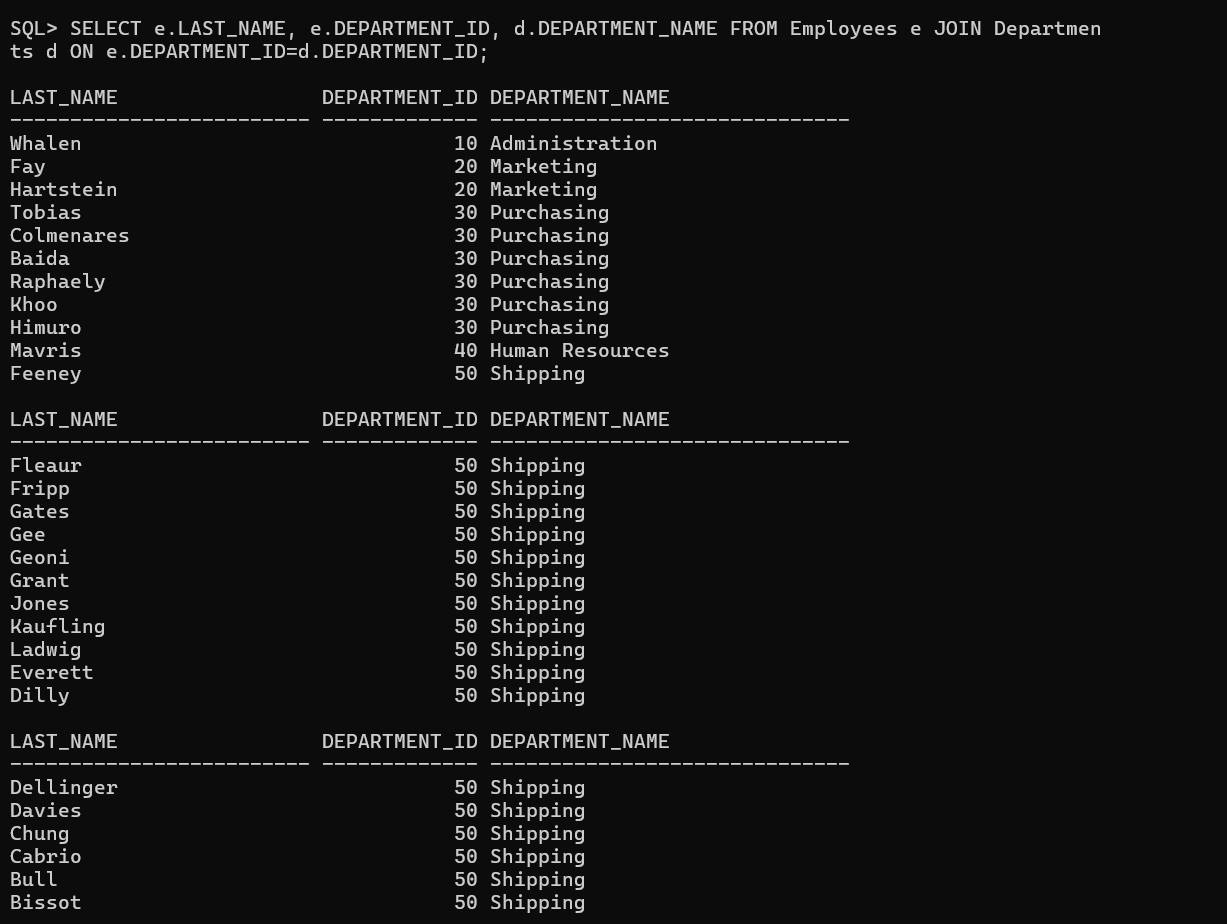
ST\_CLERK E

None of the above 0

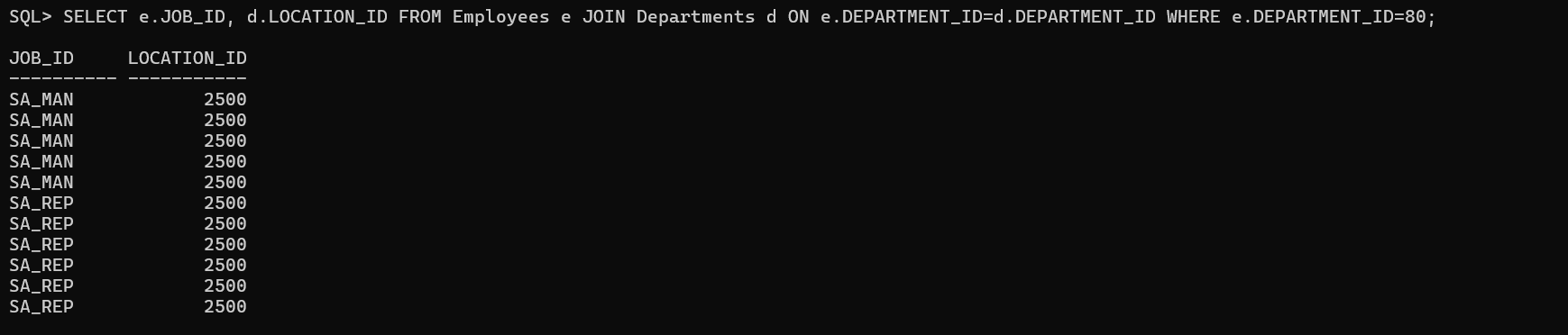


Practice Set-4(Displaying Data From Multiple Tables)

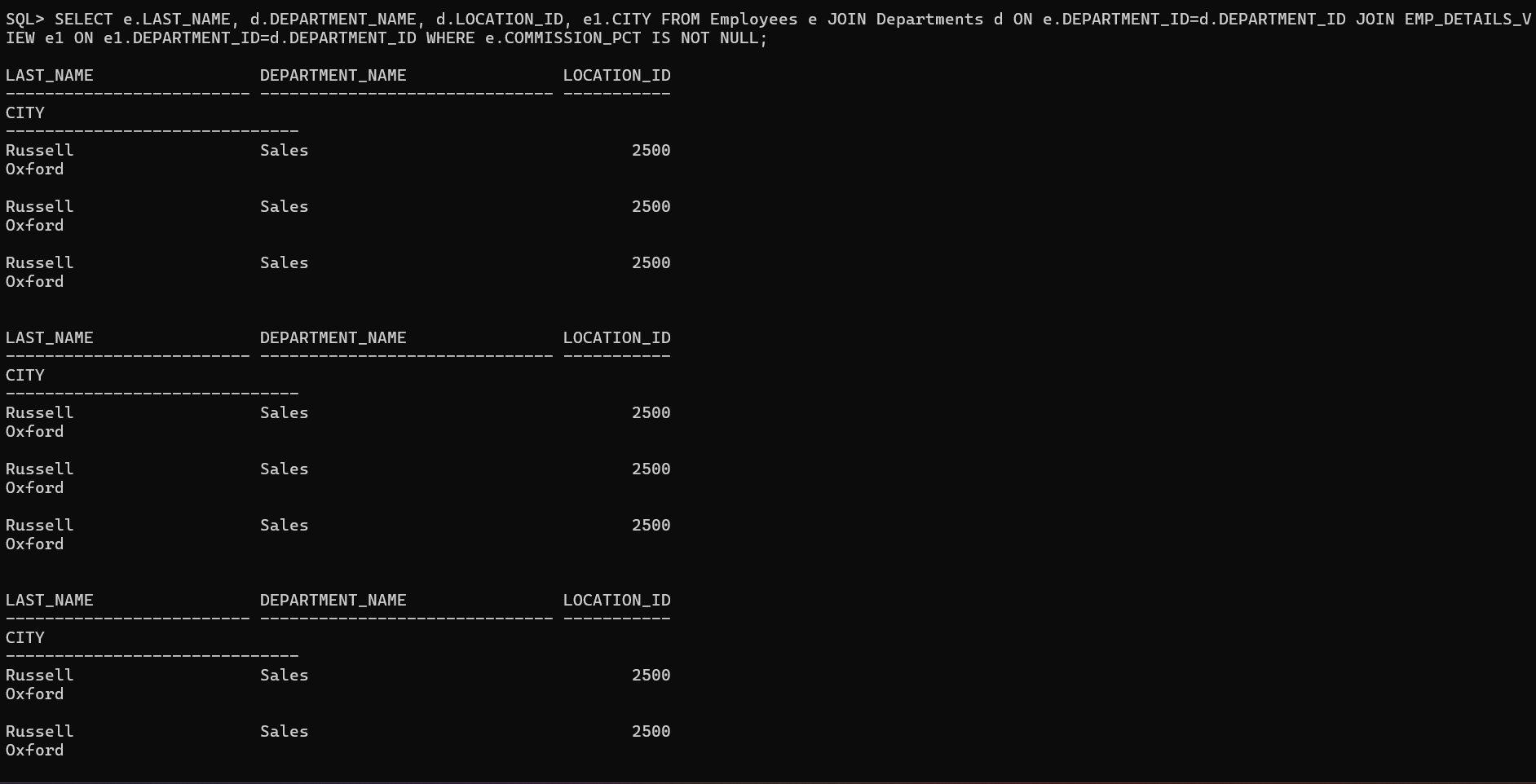
1. Write a query to display the last name, department number, and department name for all employees.



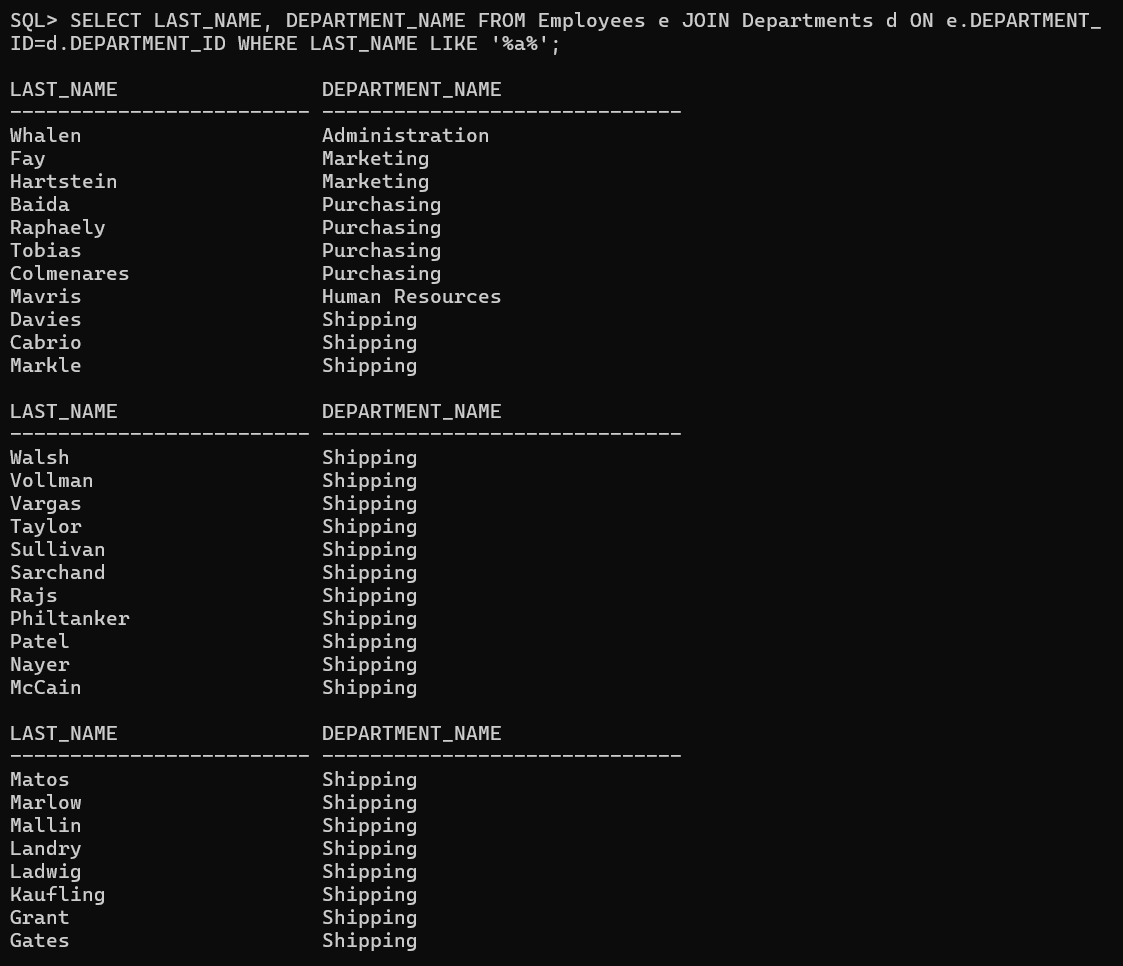
2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.



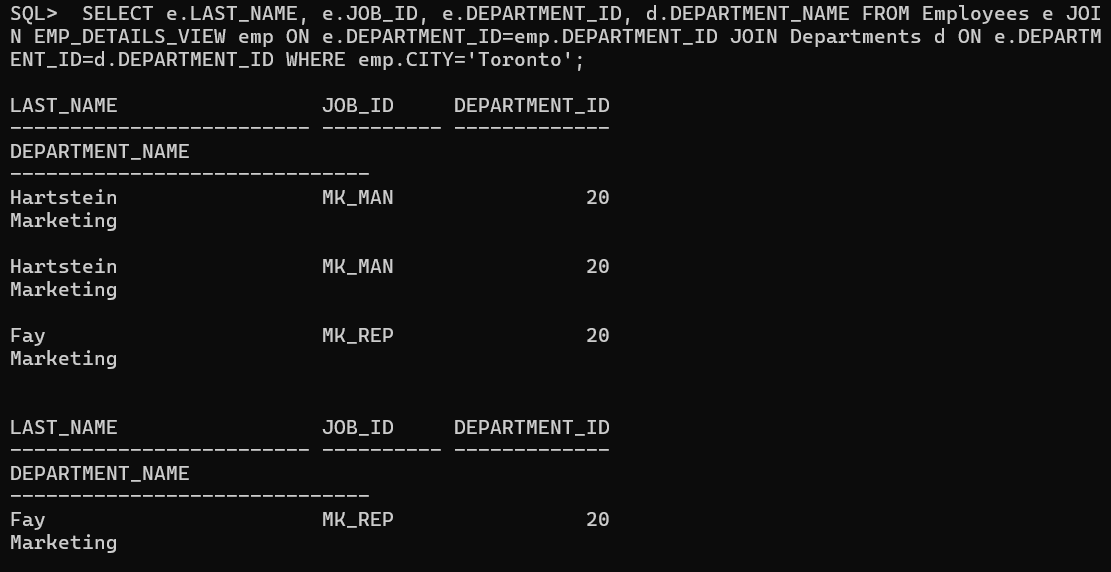
3. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission.



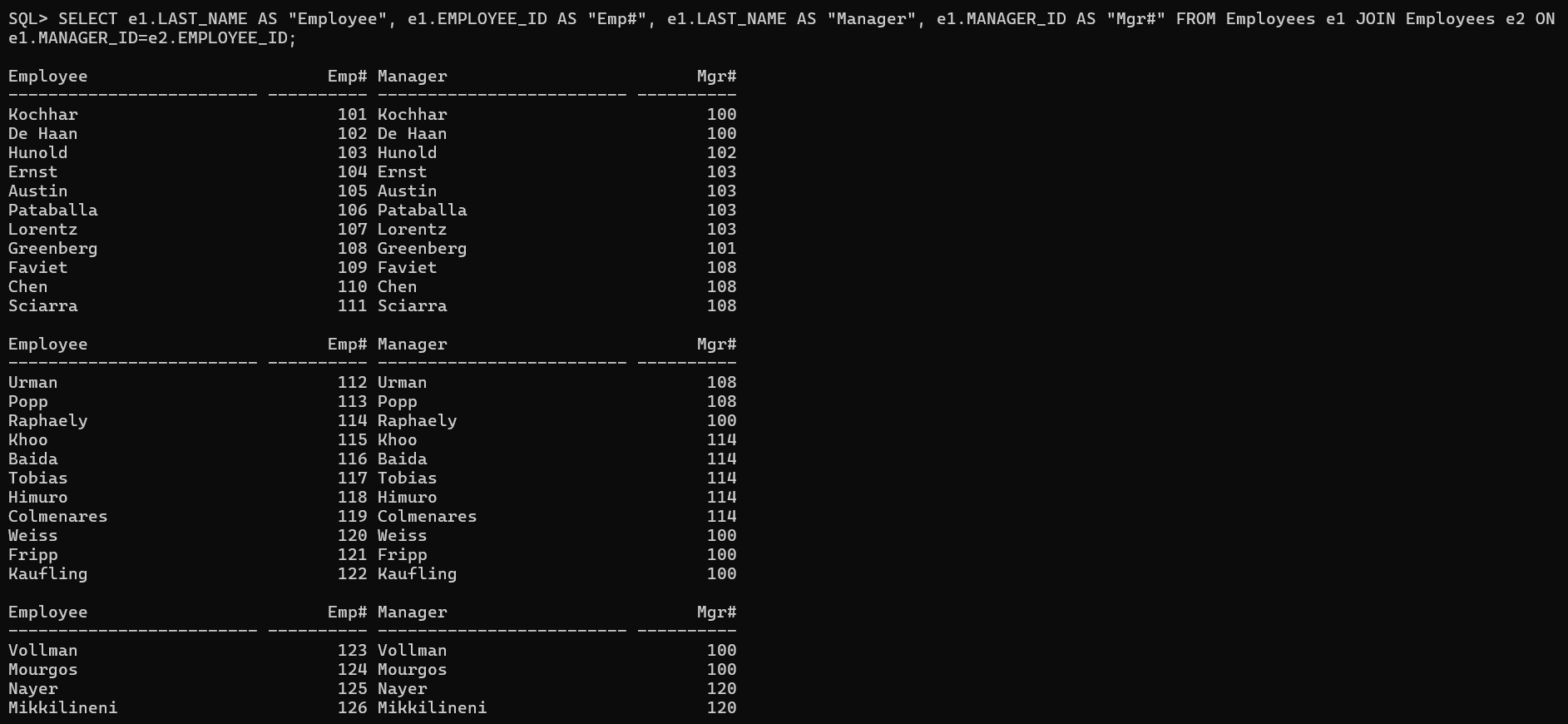
4. Display the employee last name and department name for all employees who have an a (lowercase) in their last names. Place your SQL statement in a text file named lab4\_4.sql.



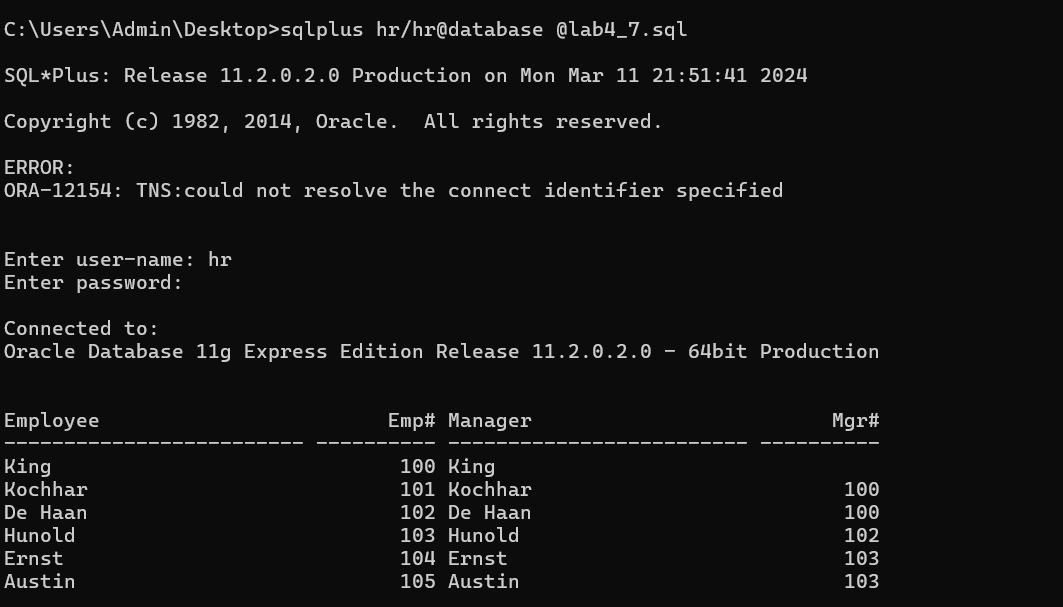
5. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.



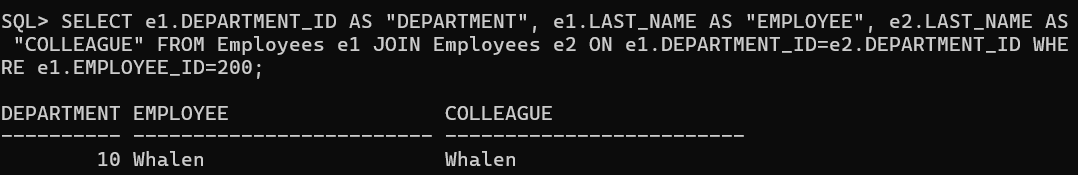
6. Display the employee last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Place your SQL statement in a text file named lab4\_6.sql.



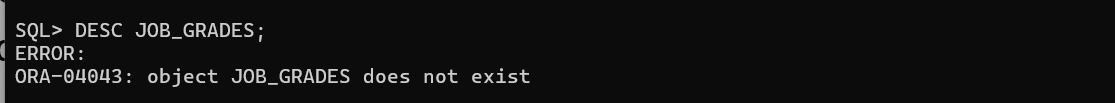
7. Modify lab4\_6.sql to display all employees including King, who has no manager. Order the results by the employee number. Place your SQL statement in a text file named lab4\_7.sql. Run the query in lab4\_7.sql.



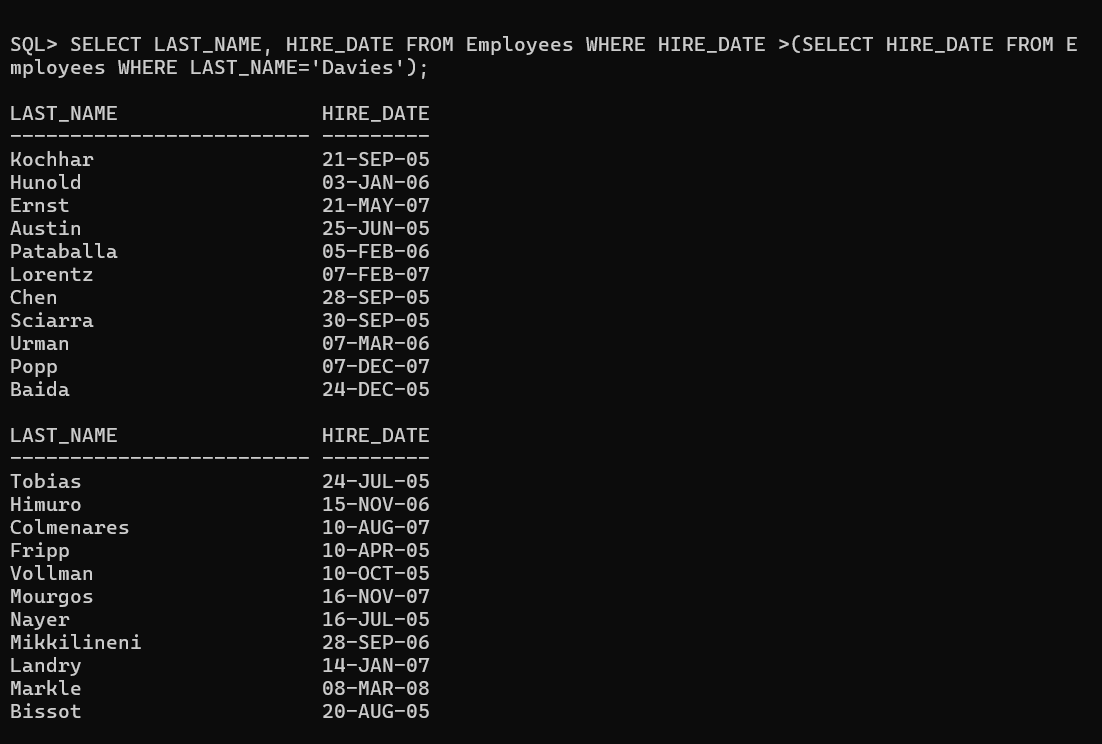
8. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label.



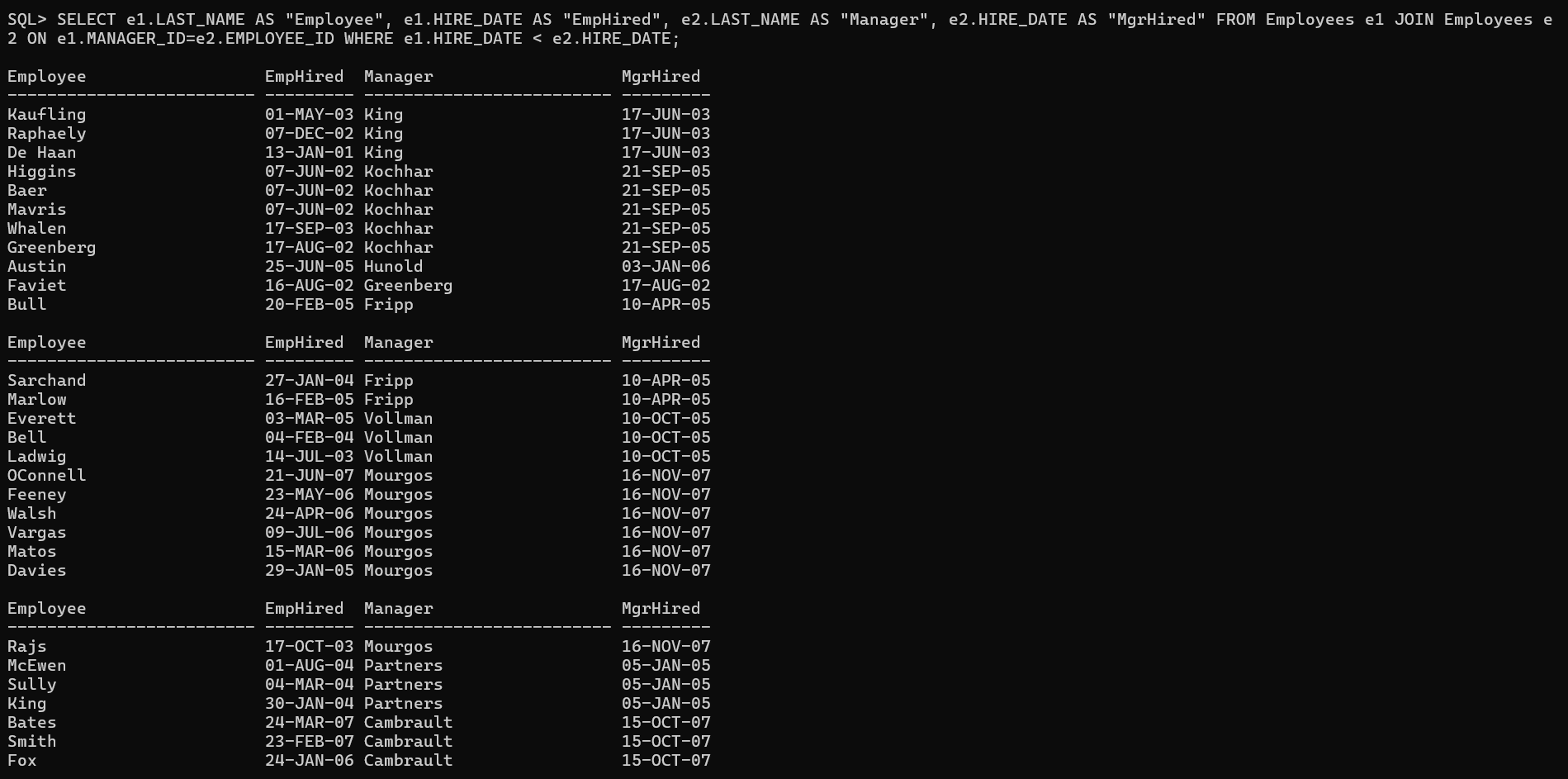
9. Show the structure of the JOB\_GRADES table. Create a query that displays the name, job, department name, salary, and grade for all employees.



10. . Create a query to display the name and hire date of any employee hired after employee Davies.



11. Display the names and hire dates for all employees who were hired before their managers, along with their manager’s names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.



Practice Set-5(Aggregating Data Using Group Functions)

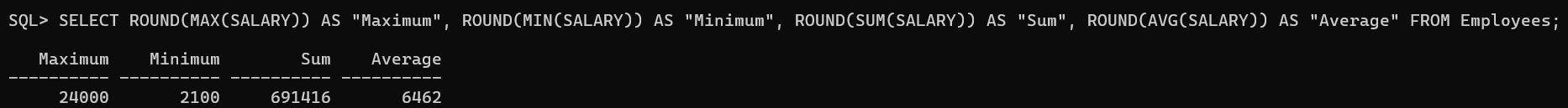
1. Determine the validity of the following three statements. Circle either True or False.

1.1 Group functions work across many rows to produce one result per group. True/False

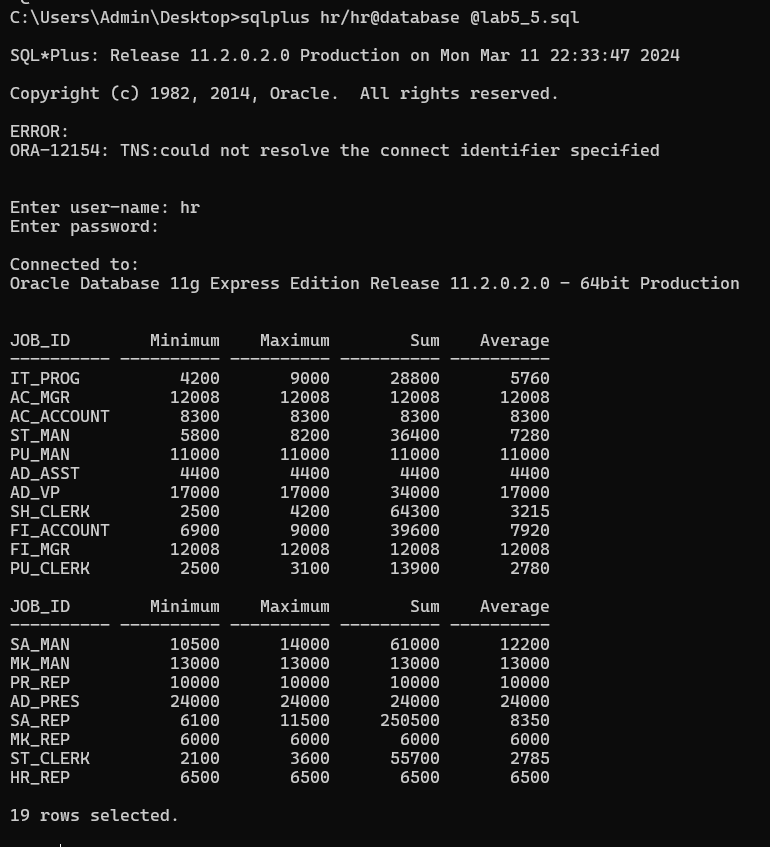
1.2 Group functions include nulls in calculations. True/False

1.3 The WHERE clause restricts rows prior to inclusion in a group calculation. True/False

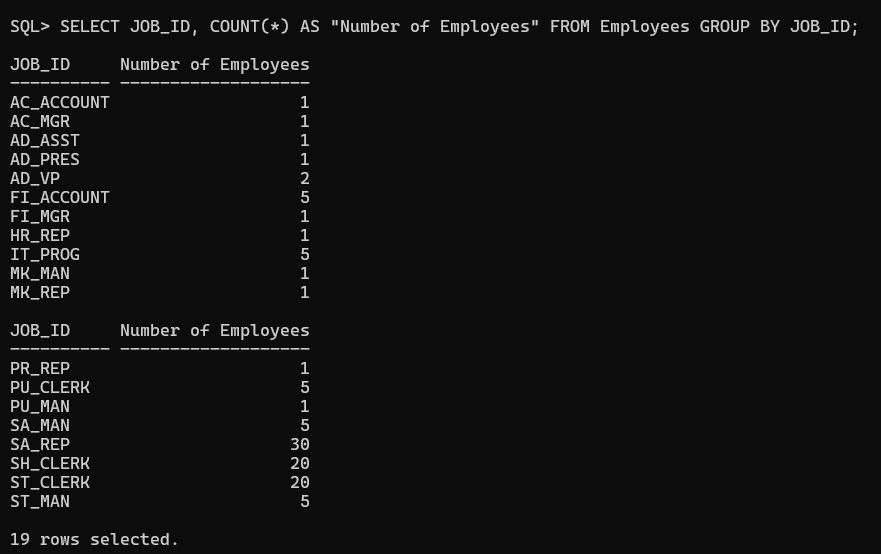
2.Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Place your SQL statement in a text file named lab5\_4.sql.



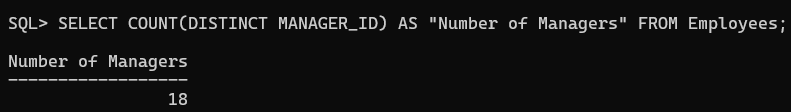
3. Modify the query in lab5\_4.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab5\_4.sql to lab5\_5.sql. Run the statement in lab5\_5.sql.



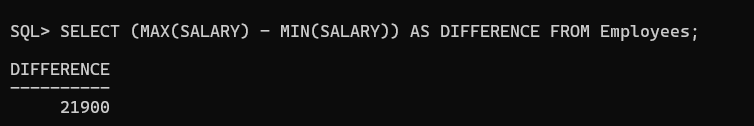
4. Write a query to display the number of people with the same job.



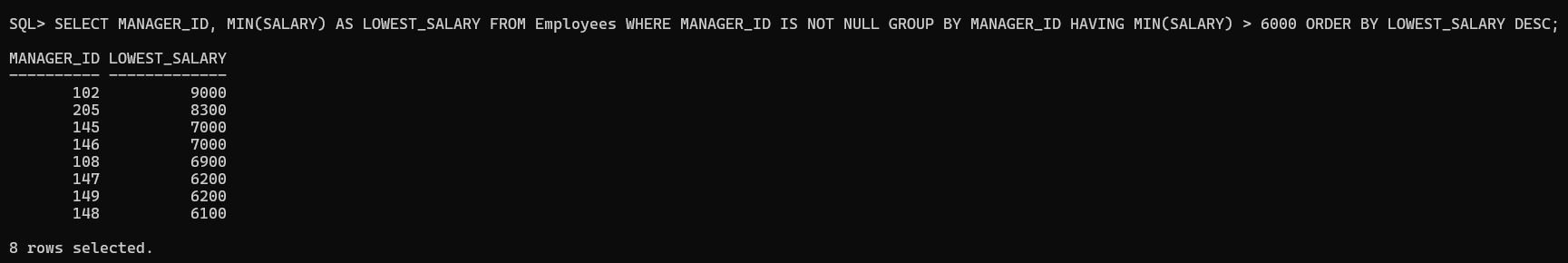
5. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER\_ID column to determine the number of managers.



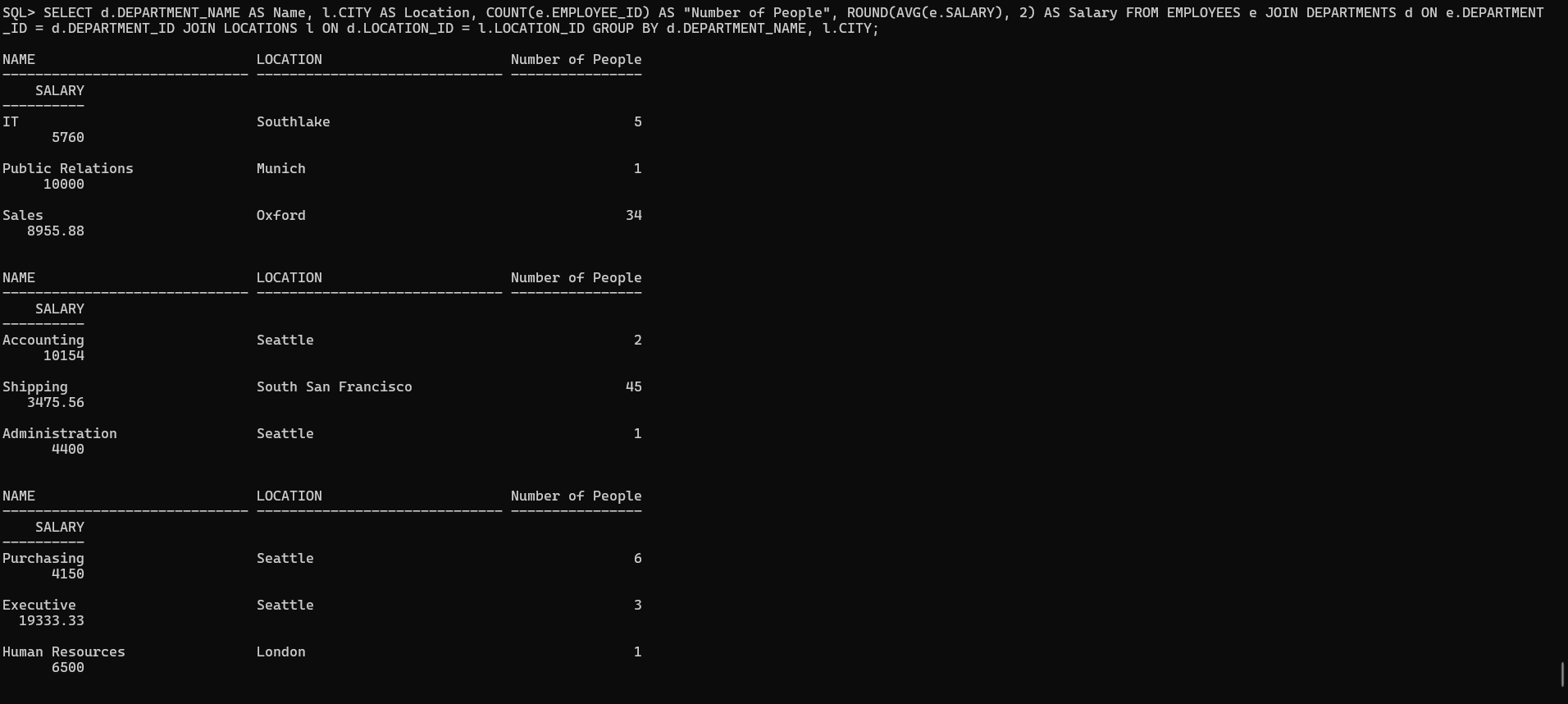
6. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.



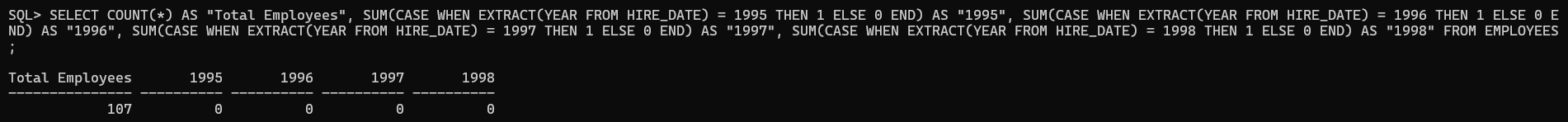
7. Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.



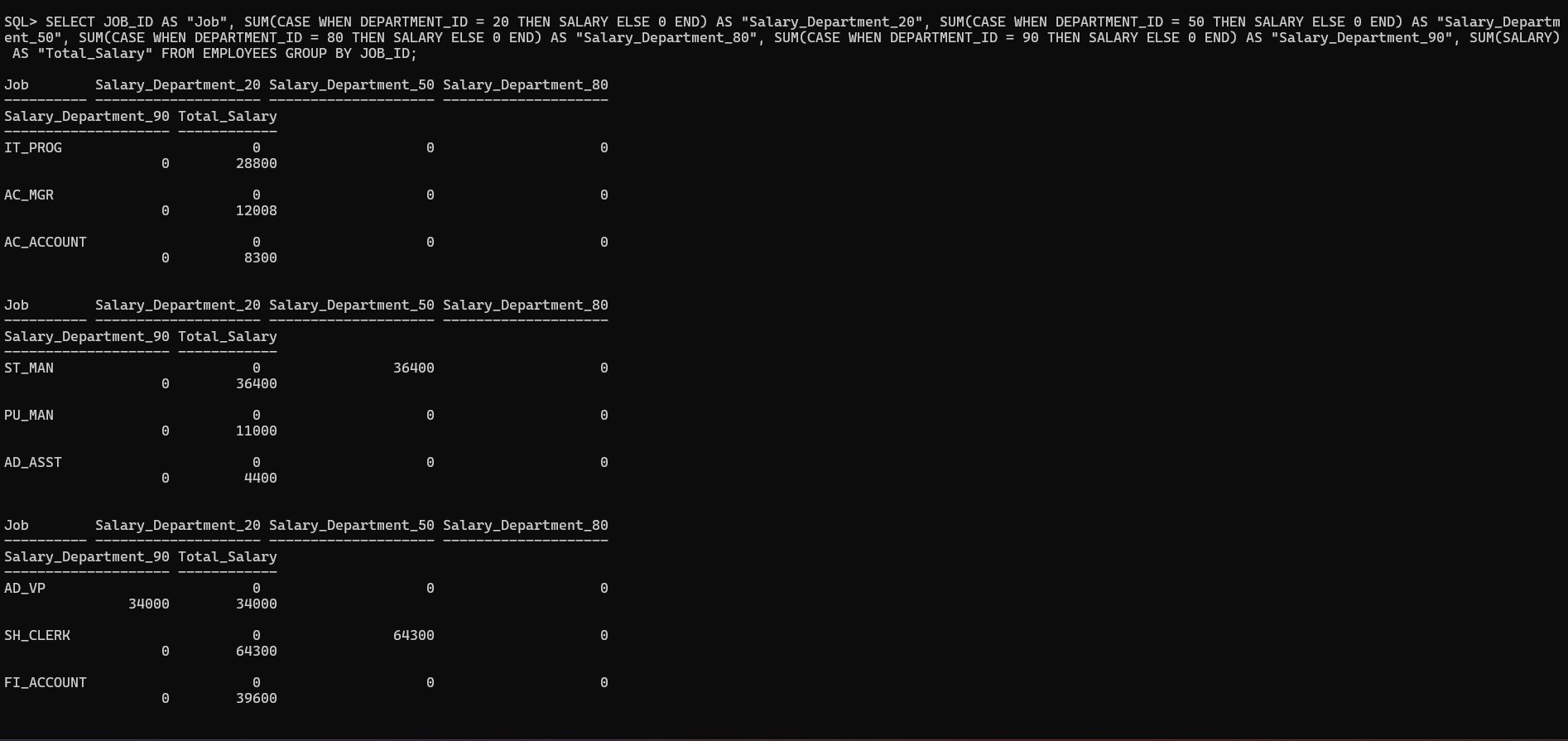
8. . Write a query to display each department’s name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.



9. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

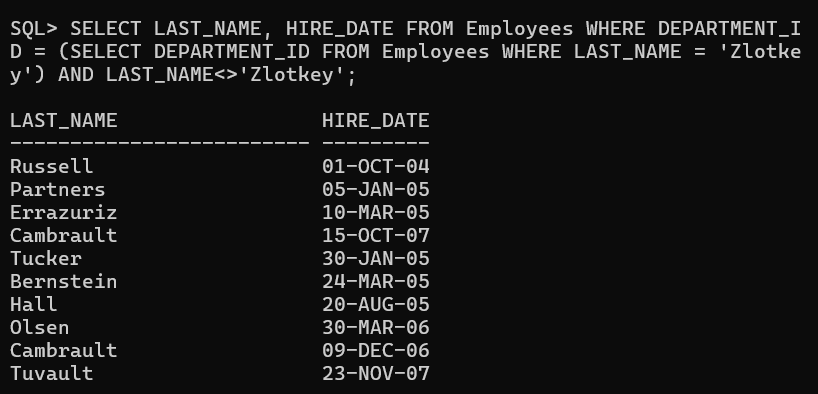


10. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

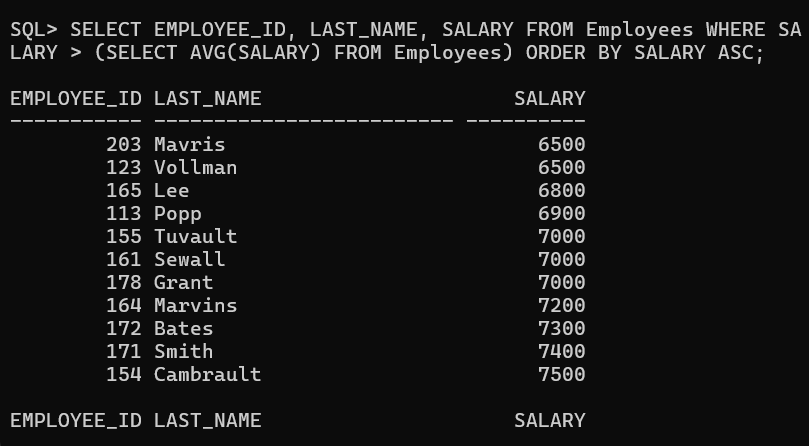


Practice Set-6(Subqueries)

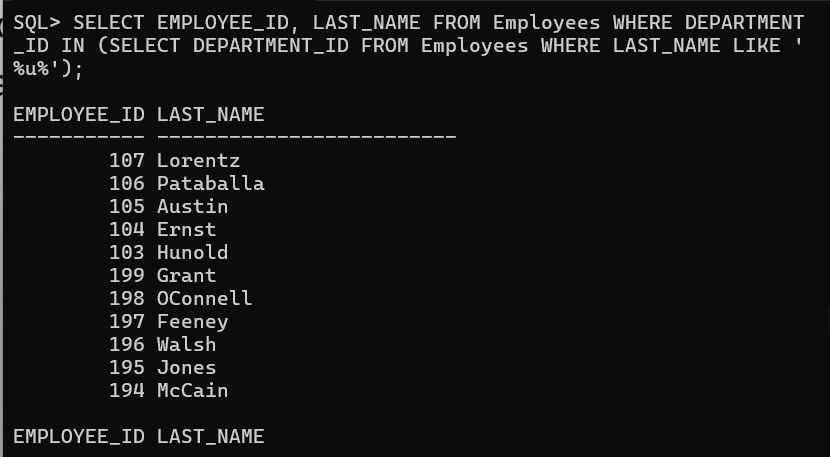
1. Write a query to display the last name and hire date of any employee in the same department as Zlotkey. Exclude Zlotkey.



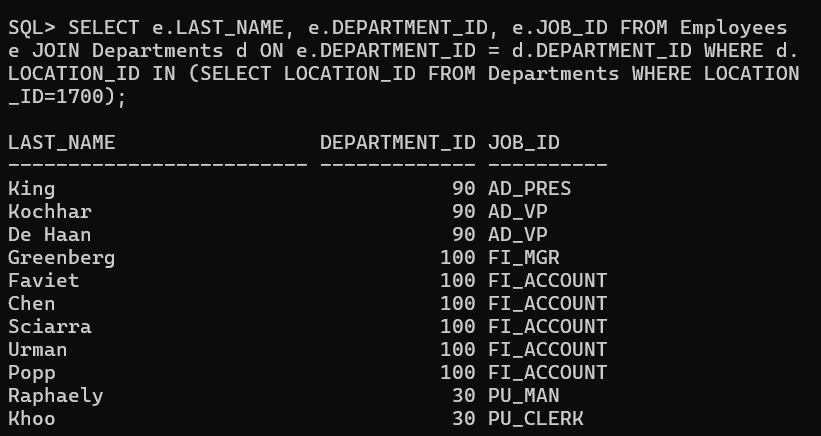
2. Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.



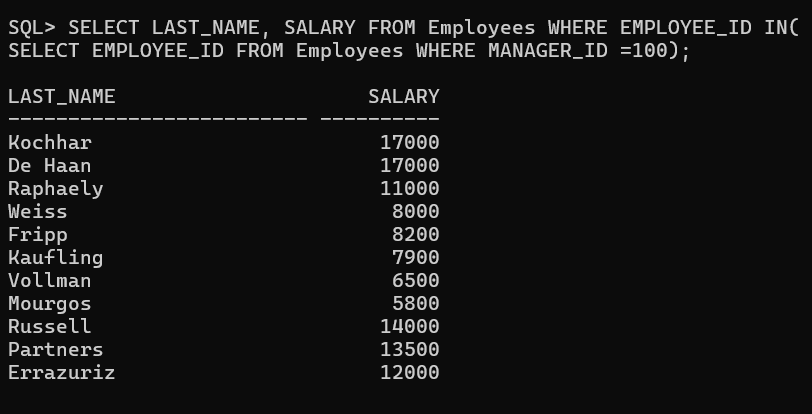
3. Write a query that displays the employee numbers and last names of all employees who work in a department with any employee whose last name contains a u. Place your SQL statement in a text file named lab6\_3.sql. Run your query.



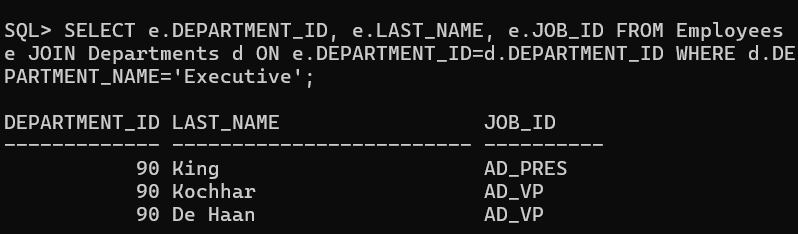
4.Display the last name, department number, and job ID of all employees whose department location ID is 1700.



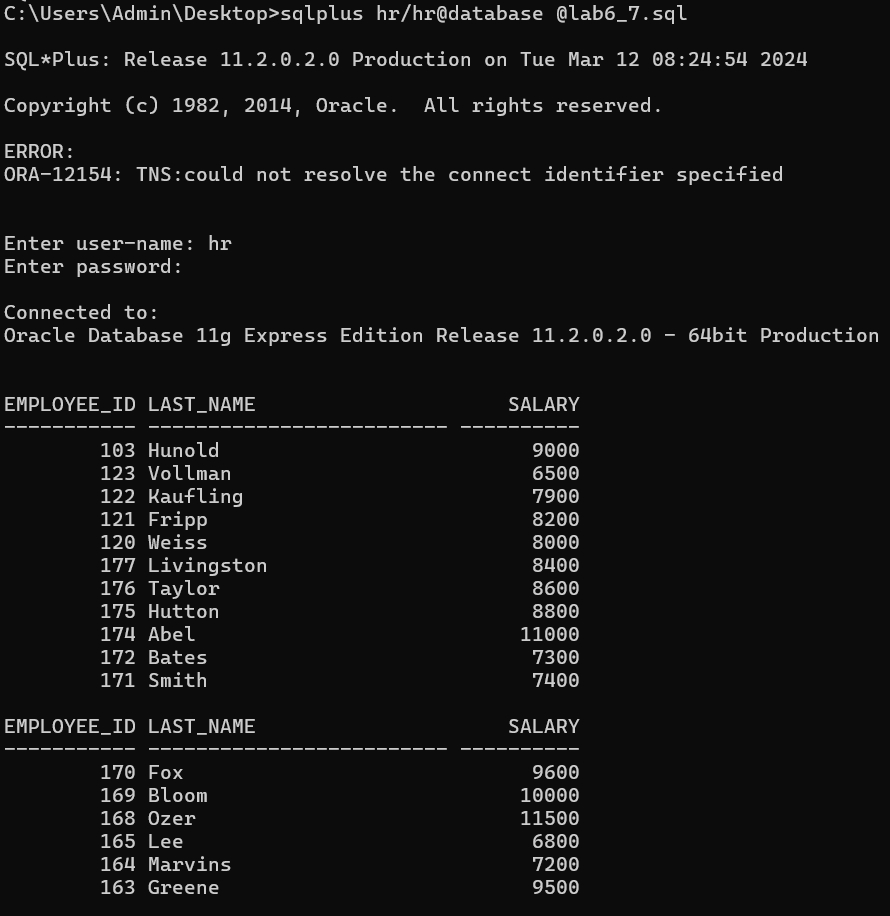
5. Display the last name and salary of every employee who reports to King.



6. Display the department number, last name, and job ID for every employee in the Executive department.



7. Modify the query in lab6\_3.sql to display the employee numbers, last names, and salaries of all employees who earn more than the average salary and who work in a department with any employee with a u in their name. Resave lab6\_3.sql to lab6\_7.sql. Run the statement in lab6\_7.sql.



Practice Set-7(Producing Readable Outputs)

Determine whether the following two statements are true or false:

1. The following statement is valid:

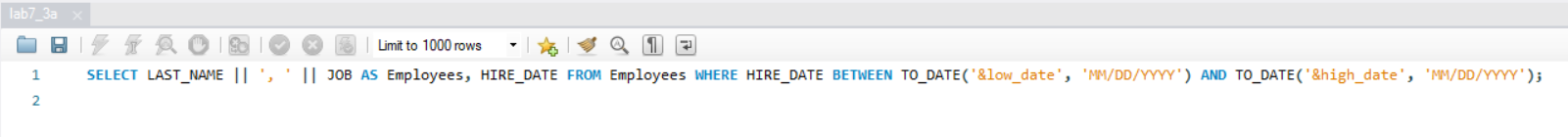
DEFINE & p\_val = 100

True/False

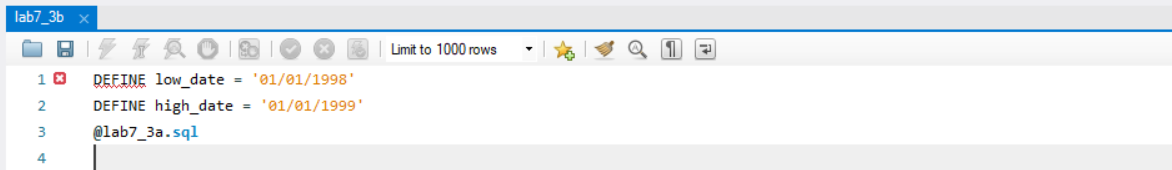
2. The DEFINE command is a SQL command.

True/False

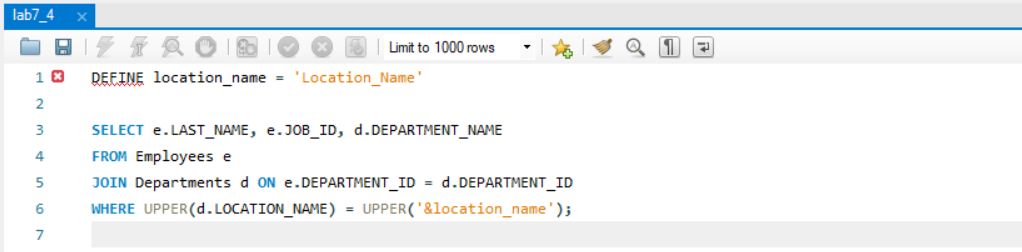
3. Write a script to display the employee last name, job, and hire date for all employees who started between a given range. Concatenate the name and job together, separated by a space and comma, and label the column Employees. In a separate SQL script file, use the DEFINE command to provide the two ranges. Use the format MM/DD/YYYY. Save the script files as lab7\_3a.sql and lab7\_3b.sql. DEFINE low\_date = 01/01/1998 DEFINE high\_date = 01/01/1999

Created 2 files successfully as lab7\_3a.sql as  


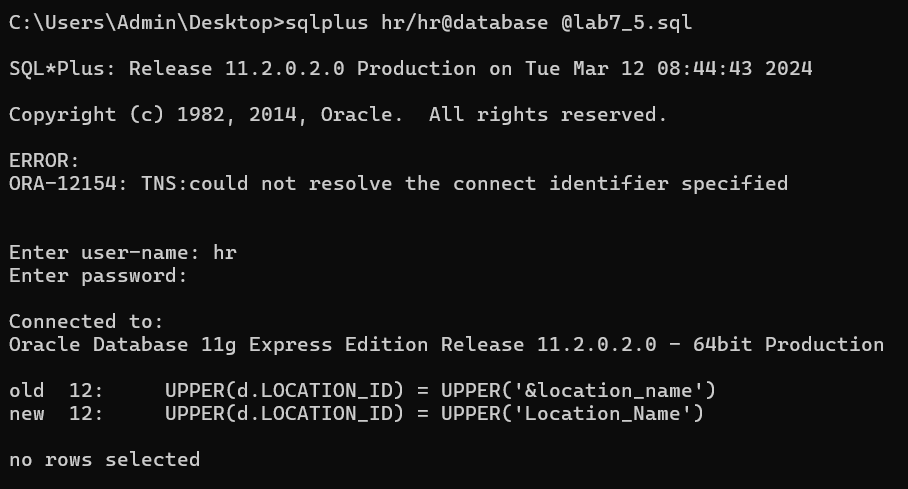
And lab7\_3b.sql as



4. Write a script to display the last names, job IDs, and department names for every employee in a given location. The search condition should allow for case-insensitive searches of the department location. Save the script file as lab7\_4.sql.

This is the required file ->  


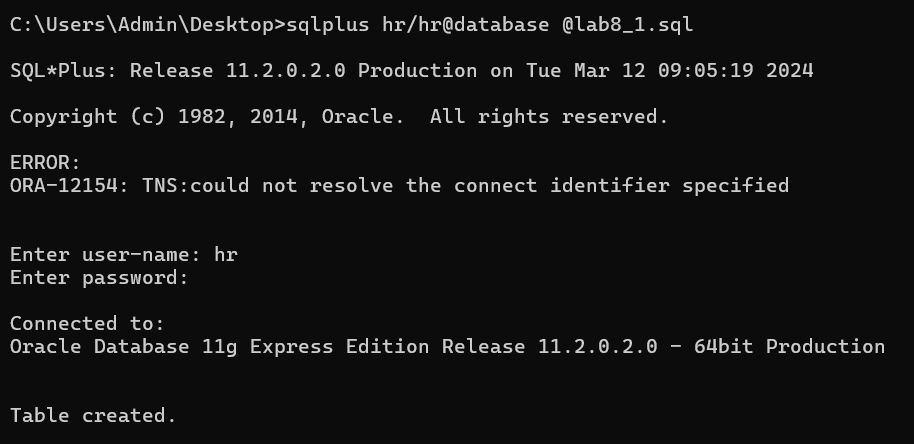
5. Modify the code in lab7\_4.sql to create a report containing the department name, employee last name, hire date, salary, and annual salary for each employee in a given location. Label the columns DEPARTMENT NAME, EMPLOYEE NAME, START DATE, SALARY, and ANNUAL SALARY, placing the labels on multiple lines. Resave the script as lab7\_5.sql, and execute the commands in the script.



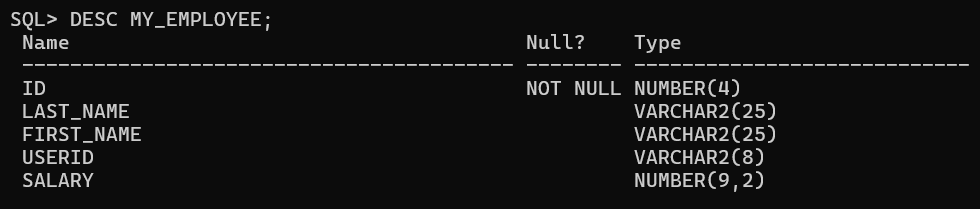
Practice -8(Manipulating Data)

Insert data into the MY\_EMPLOYEE table.

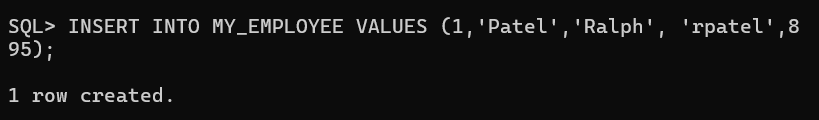
1. Run the statement in the lab8\_1.sql script to build the MY\_EMPLOYEE table to be used for the lab.



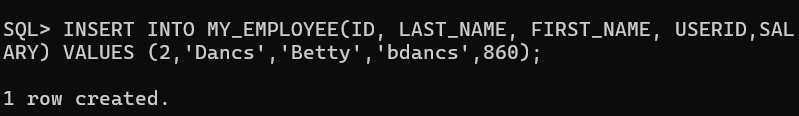
2. Describe the structure of the MY\_EMPLOYEE table to identify the column names.



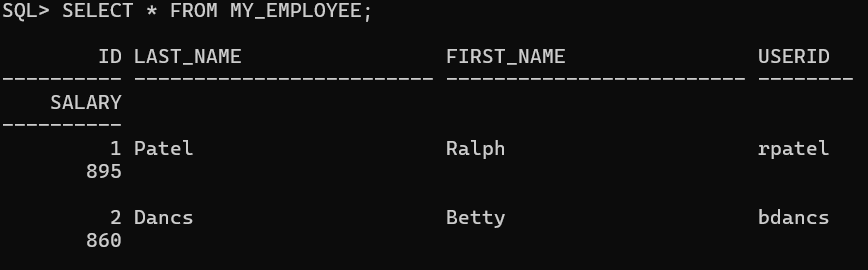
3. Add the first row of data to the MY\_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.



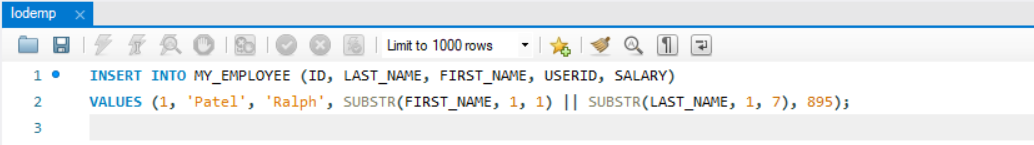
4. Populate the MY\_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.



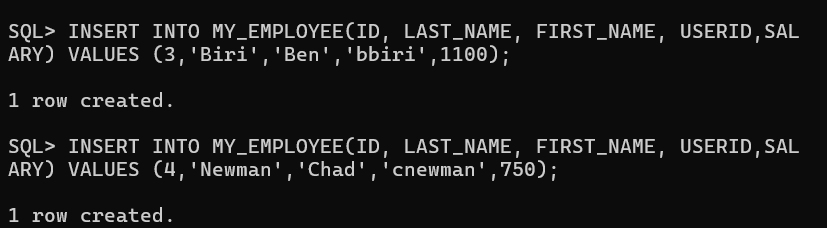
5. Confirm your addition to the table.



6. Write an insert statement in a text file named loademp.sql to load rows into the MY\_EMPLOYEE table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID.

This is the required file  


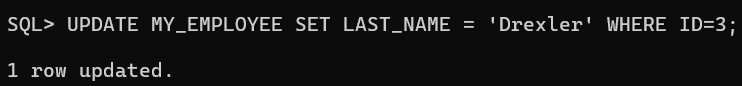
7. Populate the table with the next two rows of sample data by running the insert statement in the script that you created.



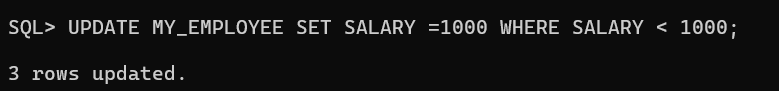
8. Confirm your additions to the table.



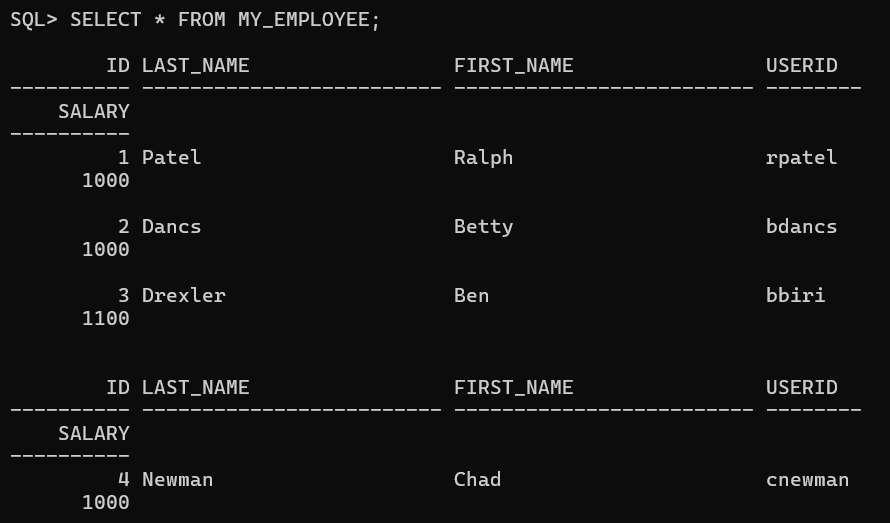
9. Change the last name of employee 3 to Drexler.



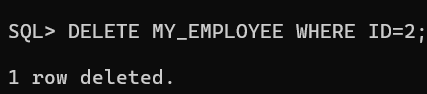
10. Change the salary to 1000 for all employees with a salary less than 900.



11. Verify your changes to the table.



12. Delete Betty Dancs from the MY\_EMPLOYEE table.



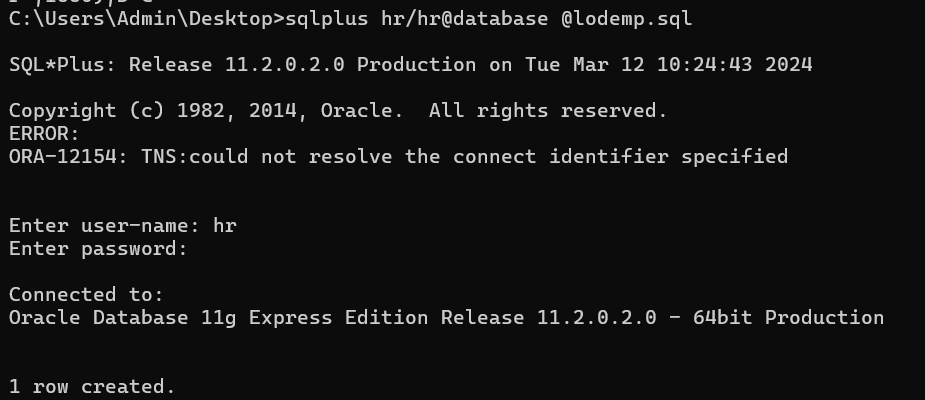
13. Confirm your changes to the table.



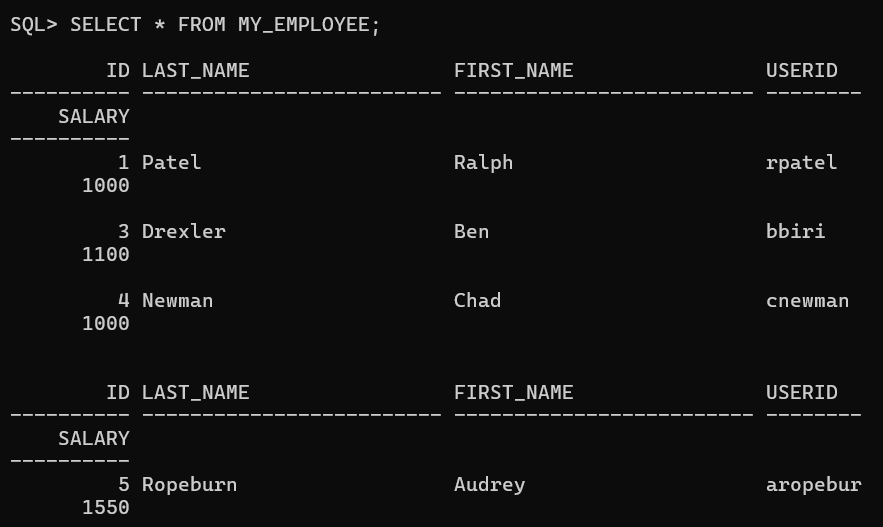
14. Commit all pending changes. Control data transaction to the MY\_EMPLOYEE table.



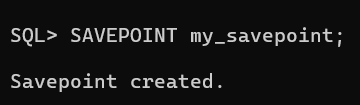
15. Populate the table with the last row of sample data by modifying the statements in the script that you created in step 6. Run the statements in the script.



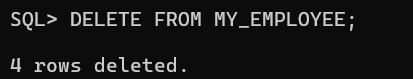
16. Confirm your addition to the table.



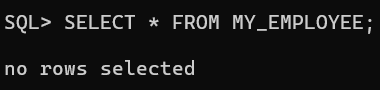
17. Mark an intermediate point in the processing of the transaction.



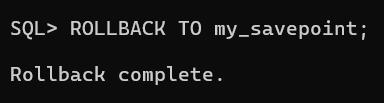
18. Empty the entire table.



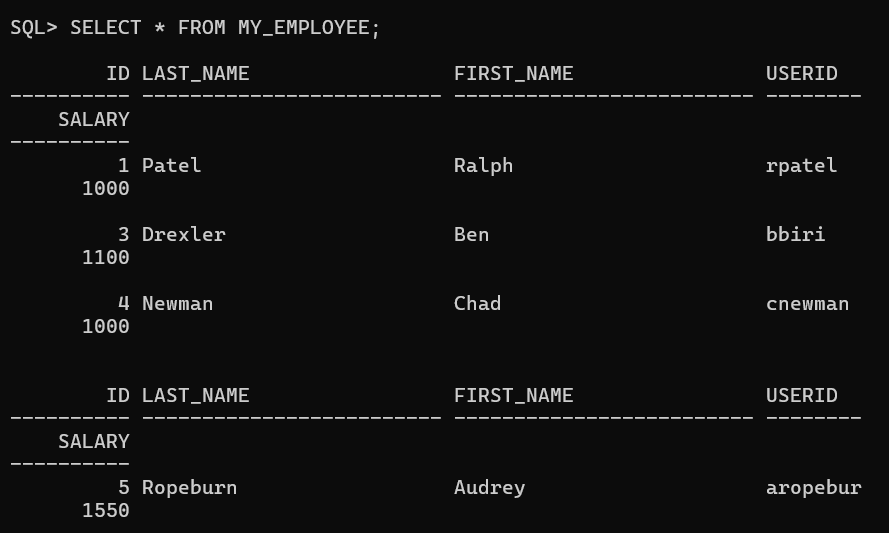
19. Confirm that the table is empty.



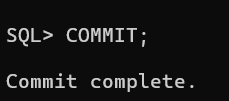
20. Discard the most recent DELETE operation without discarding the earlier INSERT operation.



21. Confirm that the new row is still intact.

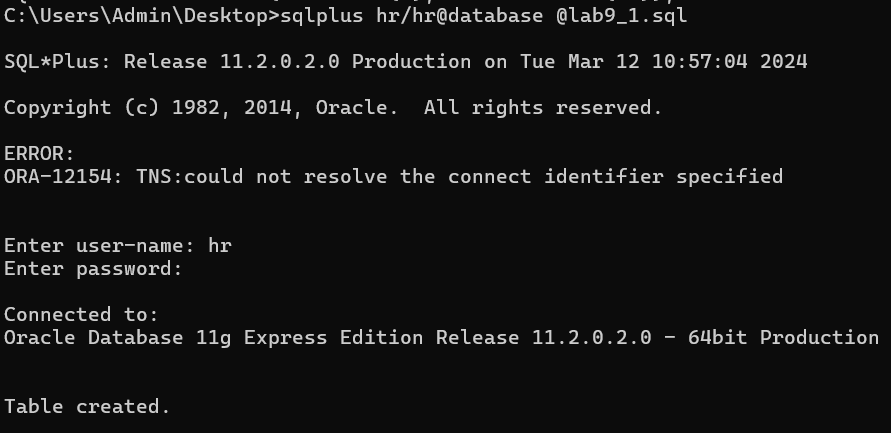


22. Make the data addition permanent.

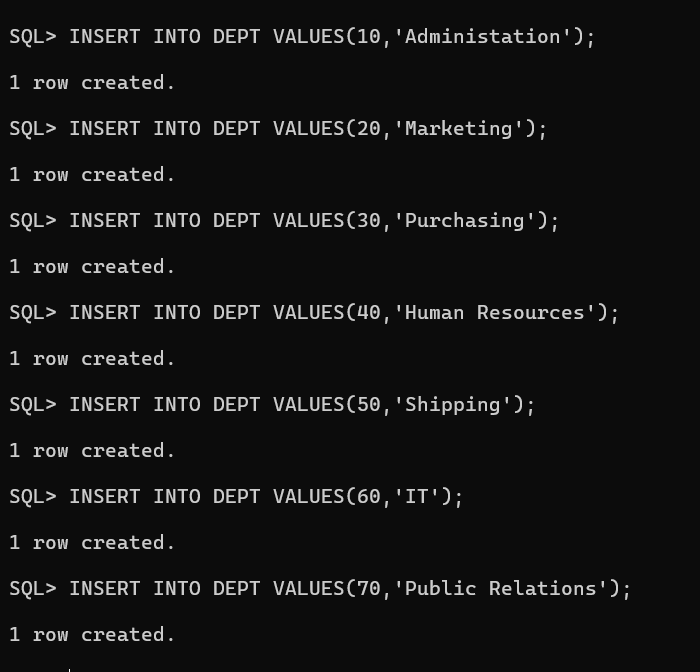


Practice-9(Creating and Managing Tables)

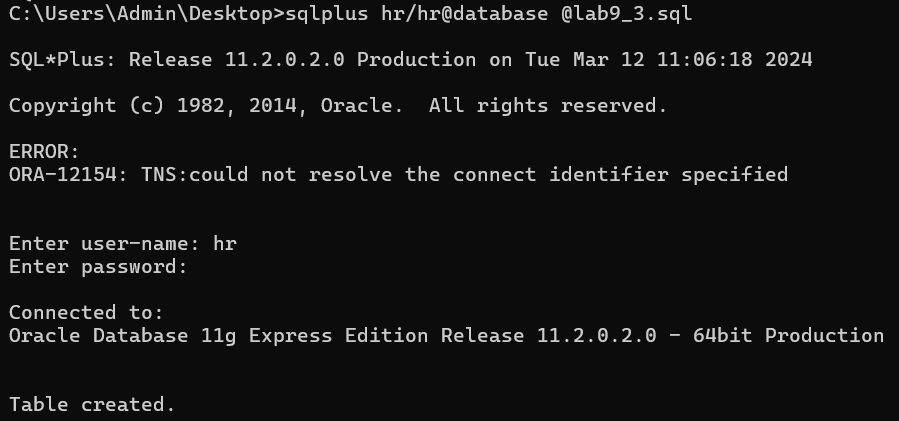
1. Create the DEPT table based on the following table instance chart. Place the syntax in a script called lab9\_1.sql, then execute the statement in the script to create the table. Confirm that the table is created.

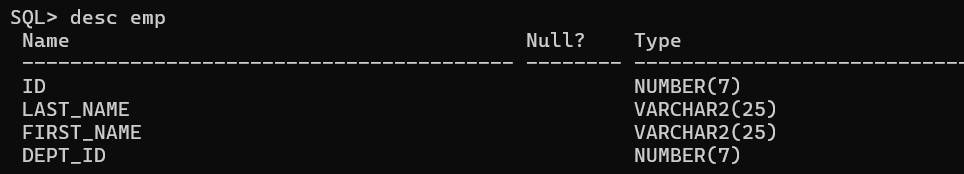


2. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need.

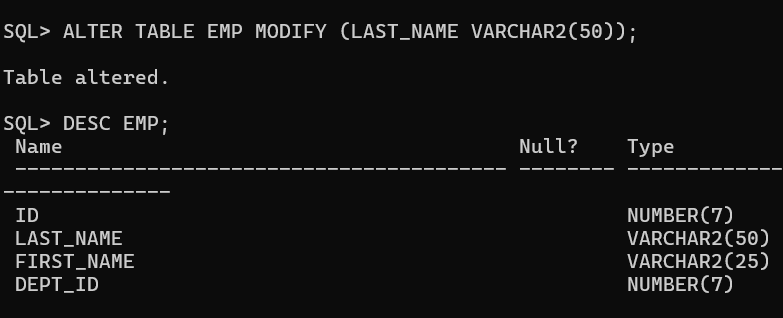


3. Create the EMP table based on the following table instance chart. Place the syntax in a script called lab9\_3.sql, and then execute the statement in the script to create the table. Confirm that the table is created.





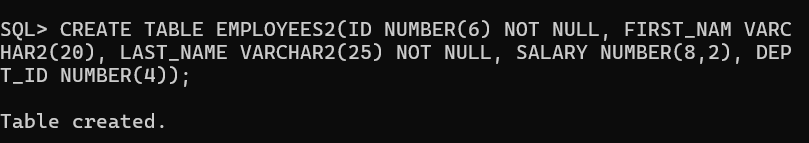
4. Modify the EMP table to allow for longer employee last names. Confirm your modification.



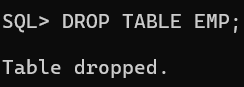
5. Confirm that both the DEPT and EMP tables are stored in the data dictionary. (Hint: USER\_TABLES)



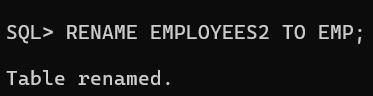
6. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and DEPARTMENT\_ID columns. Name the columns in your new table ID, FIRST\_NAME, LAST\_NAME, SALARY , and DEPT\_ID, respectively.



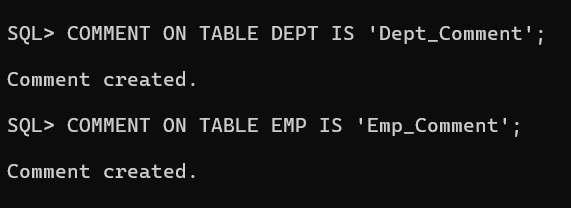
7. Drop the EMP table.

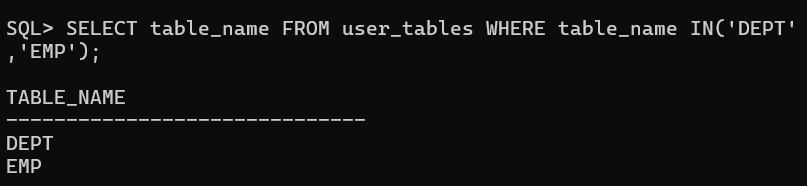


8. Rename the EMPLOYEES2 table as EMP.

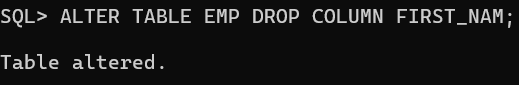


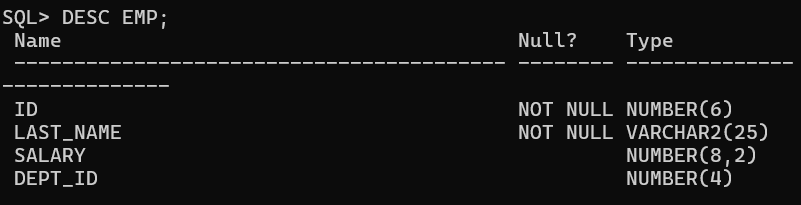
9. Add a comment to the DEPT and EMP table definitions describing the tables. Confirm your additions in the data dictionary.



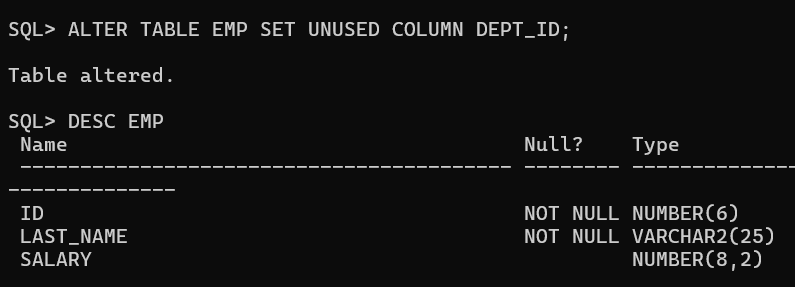


10. Drop the FIRST\_NAME column from the EMP table. Confirm your modification by checking the description of the table.

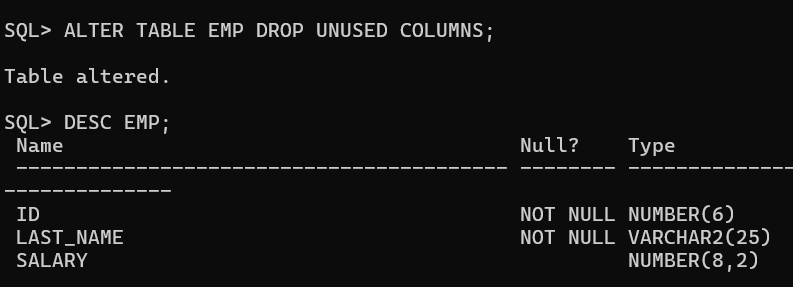




11. In the EMP table, mark the DEPT\_ID column in the EMP table as UNUSED. Confirm your modification by checking the description of the table.

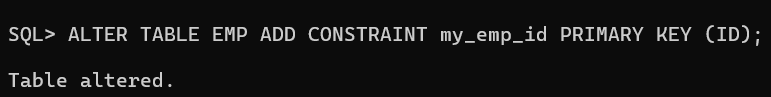


12. Drop all the UNUSED columns from the EMP table. Confirm your modification by checking the description of the table.

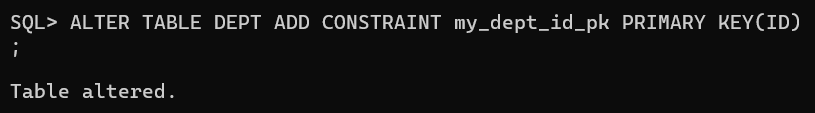


Practice-10(Including Constraints)

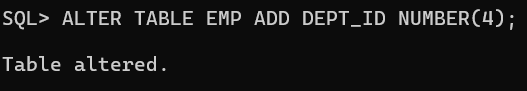
1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my\_emp\_id\_pk. Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

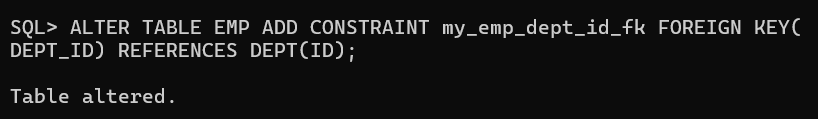


2. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my\_dept\_id\_pk. Hint: The constraint is enabled as soon as the ALTER TABLE command executes successfully.

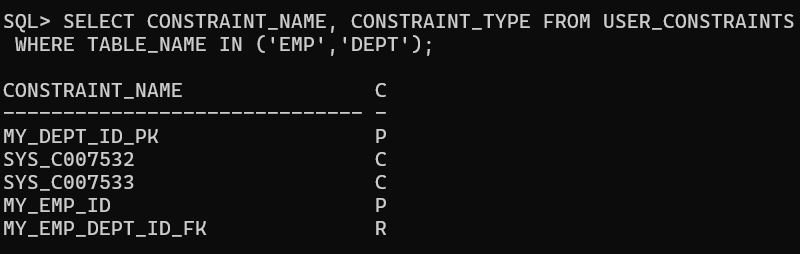


3. Add a column DEPT\_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my\_emp\_dept\_id\_fk.

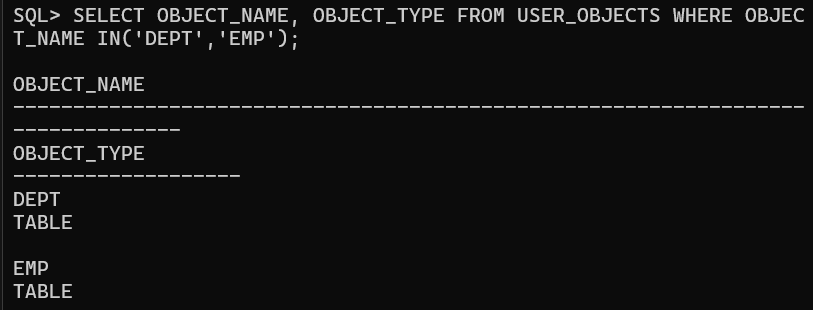




4. Confirm that the constraints were added by querying the USER\_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab10\_4.sql.



5. Display the object names and types from the USER\_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created.



6. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

