**ASSIGNMENT – 6**

**ON: SINGLE ROW FUNCTIONS IN SQL**

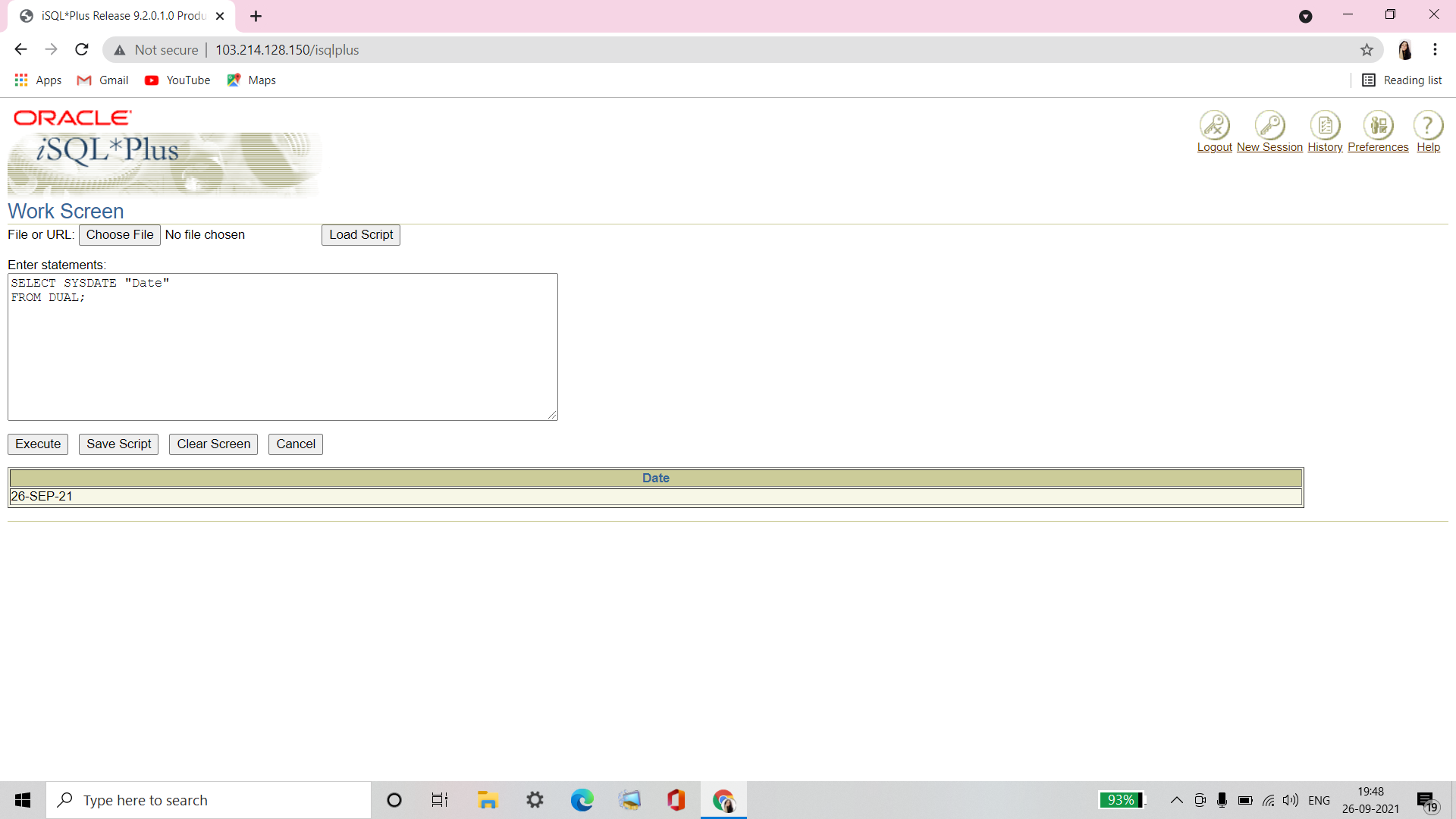
**Q1. Write a query to display the current date. Label the column Date.**

Ans1.

SELECT SYSDATE "Date"

FROM DUAL;

**Verification table -**



**Q2. 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 lab6\_2.sql.**

Ans2.

SELECT EMPLOYEE\_ID, LAST\_NAME, SALARY,

ROUND(SALARY\*1.15, 0) "New Salary"

FROM EMPLOYEES;

**Q3. Run your query in the file lab lab6\_2.sql.**

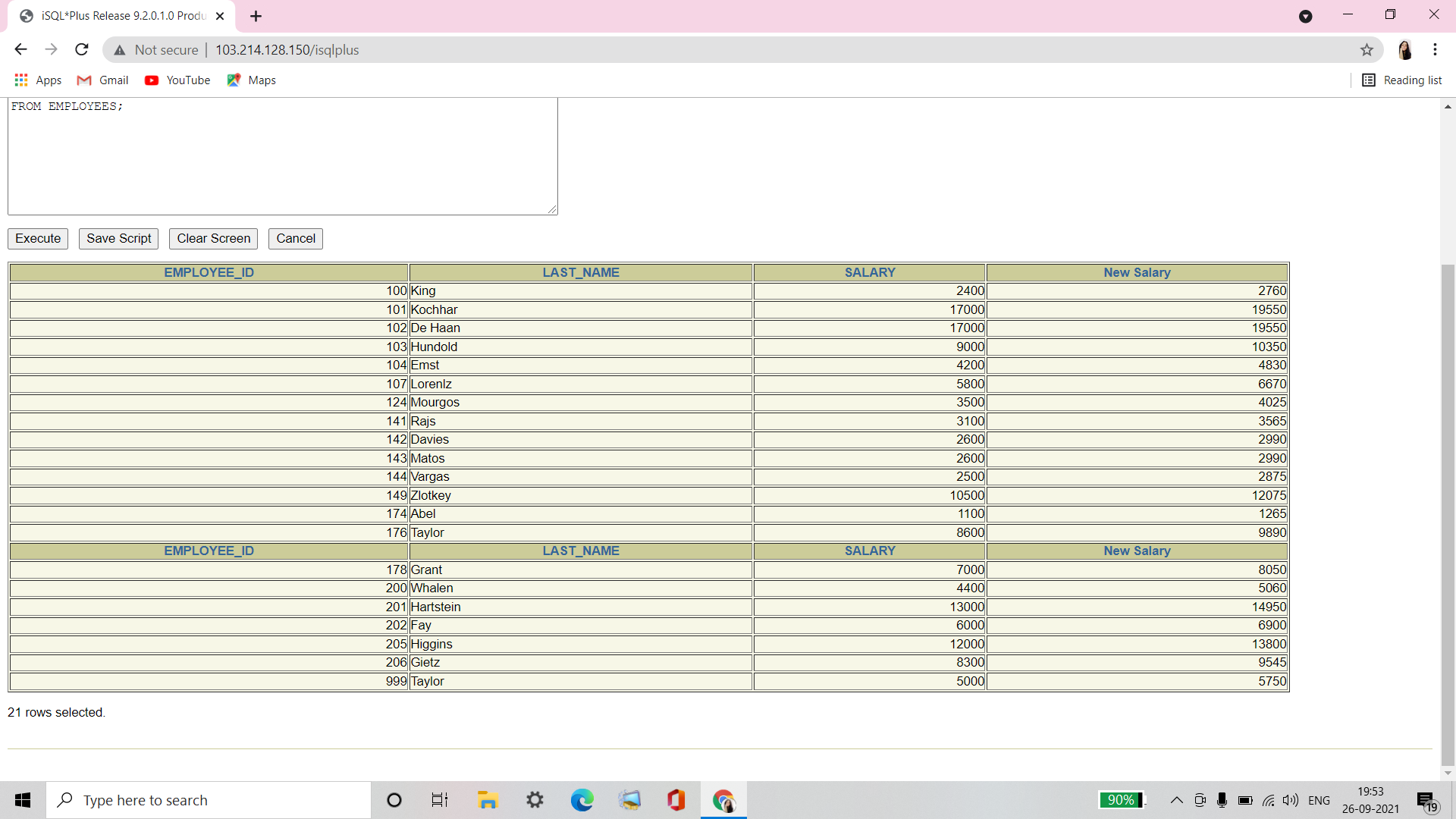
Ans3.

SELECT EMPLOYEE\_ID, LAST\_NAME, SALARY,

ROUND(SALARY\*1.15, 0) "New Salary"

FROM EMPLOYEES;

**Verification table -**



**Q4. Modify your query lab6\_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 lab6\_4.sql. Run the revised query.**

Ans4.

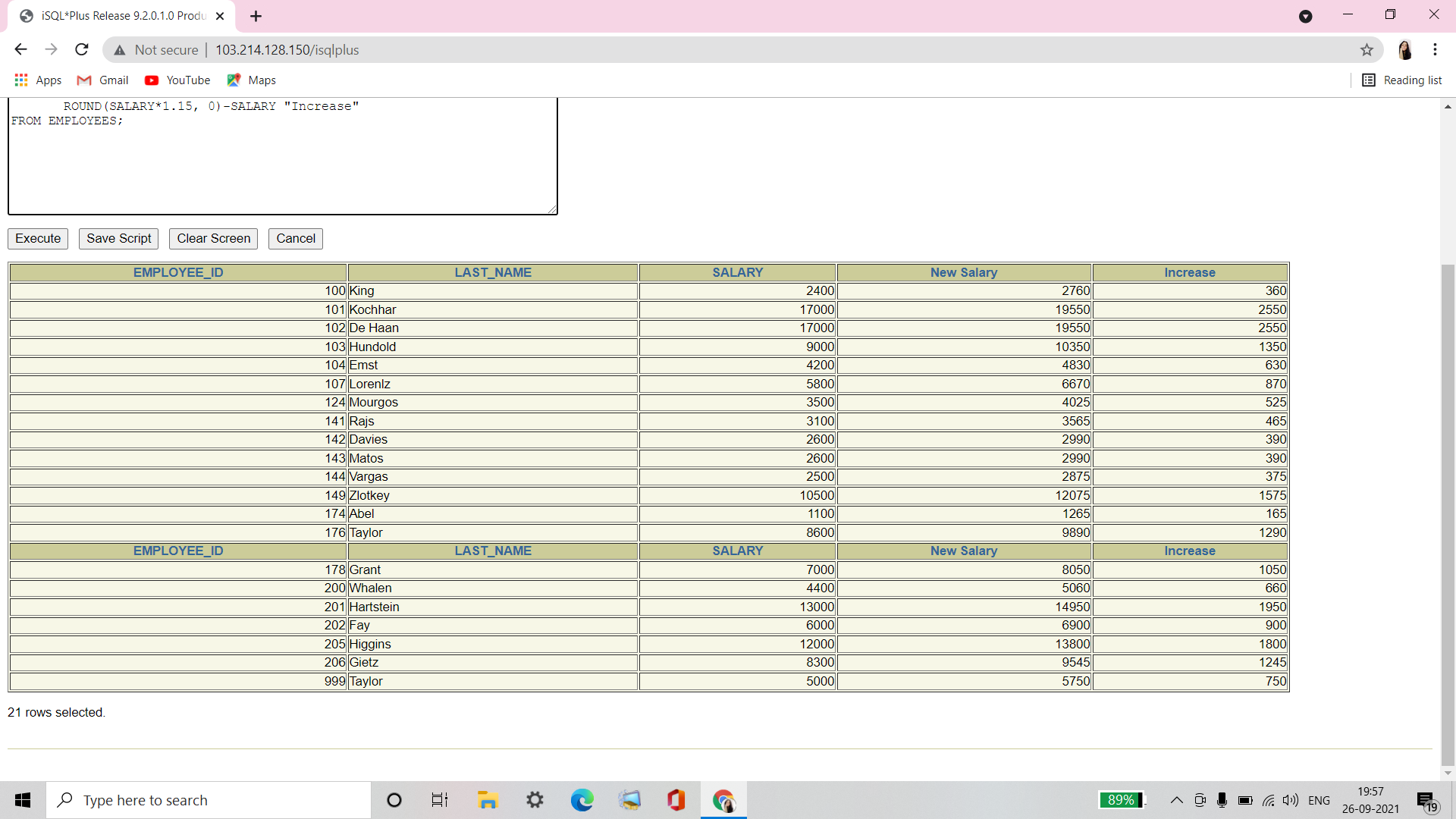
SELECT EMPLOYEE\_ID, LAST\_NAME, SALARY,

ROUND(SALARY\*1.15, 0) "New Salary",

ROUND(SALARY\*1.15, 0)-SALARY "Increase"

FROM EMPLOYEES;

**Verification table -**

****

**Q5. 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 name 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.**

Ans5.

SELECT INITCAP(LAST\_NAME) "Name",

LENGTH (LAST\_NAME) "Length"

FROM EMPLOYEES

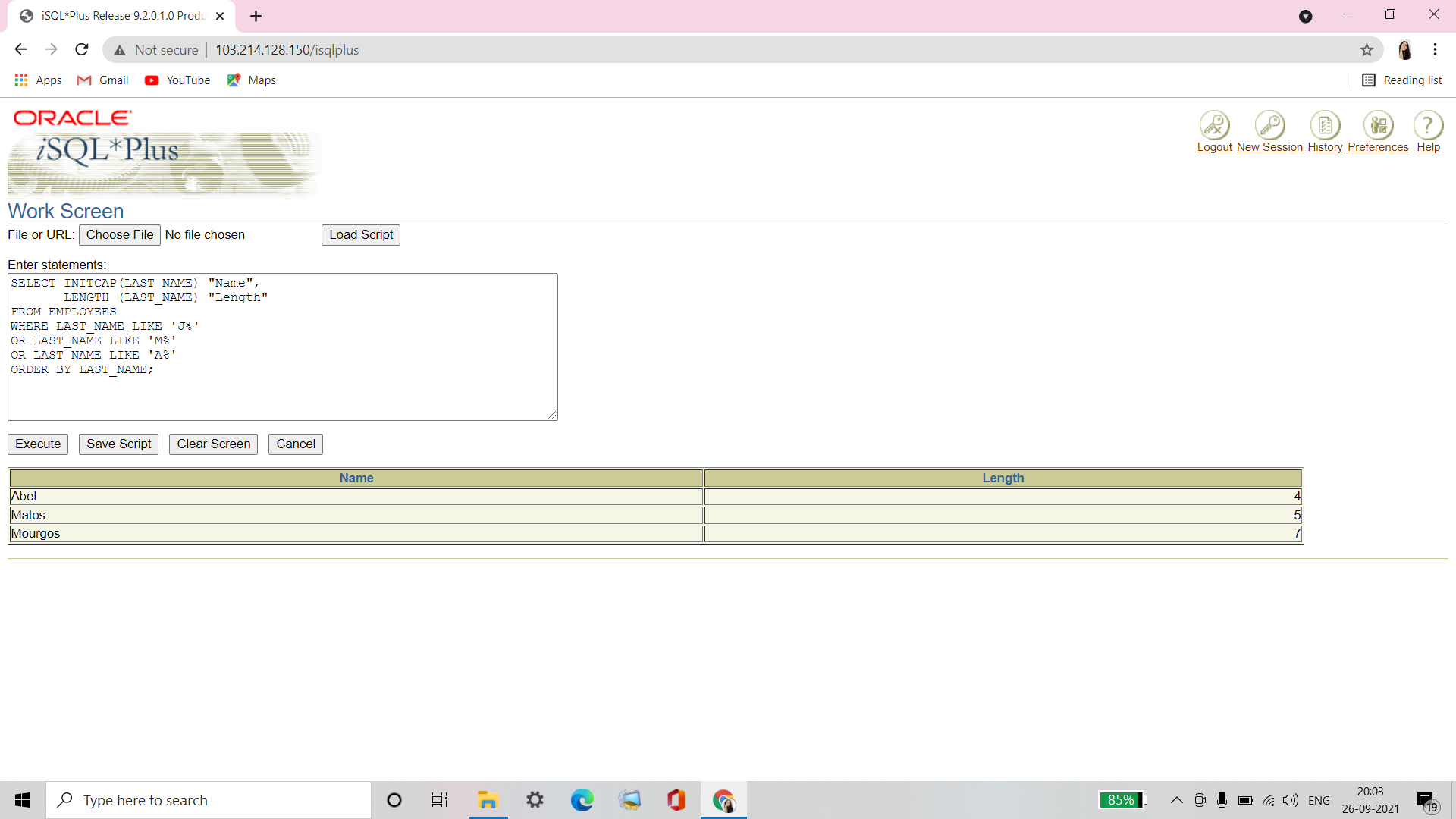
WHERE LAST\_NAME LIKE 'J%'

OR LAST\_NAME LIKE 'M%'

OR LAST\_NAME LIKE 'A%'

ORDER BY LAST\_NAME;

**Verification table -**



**Q6. For each employee, display the employee’s last name, and calculate the number of month 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.**

**Note: Your results will differ.**

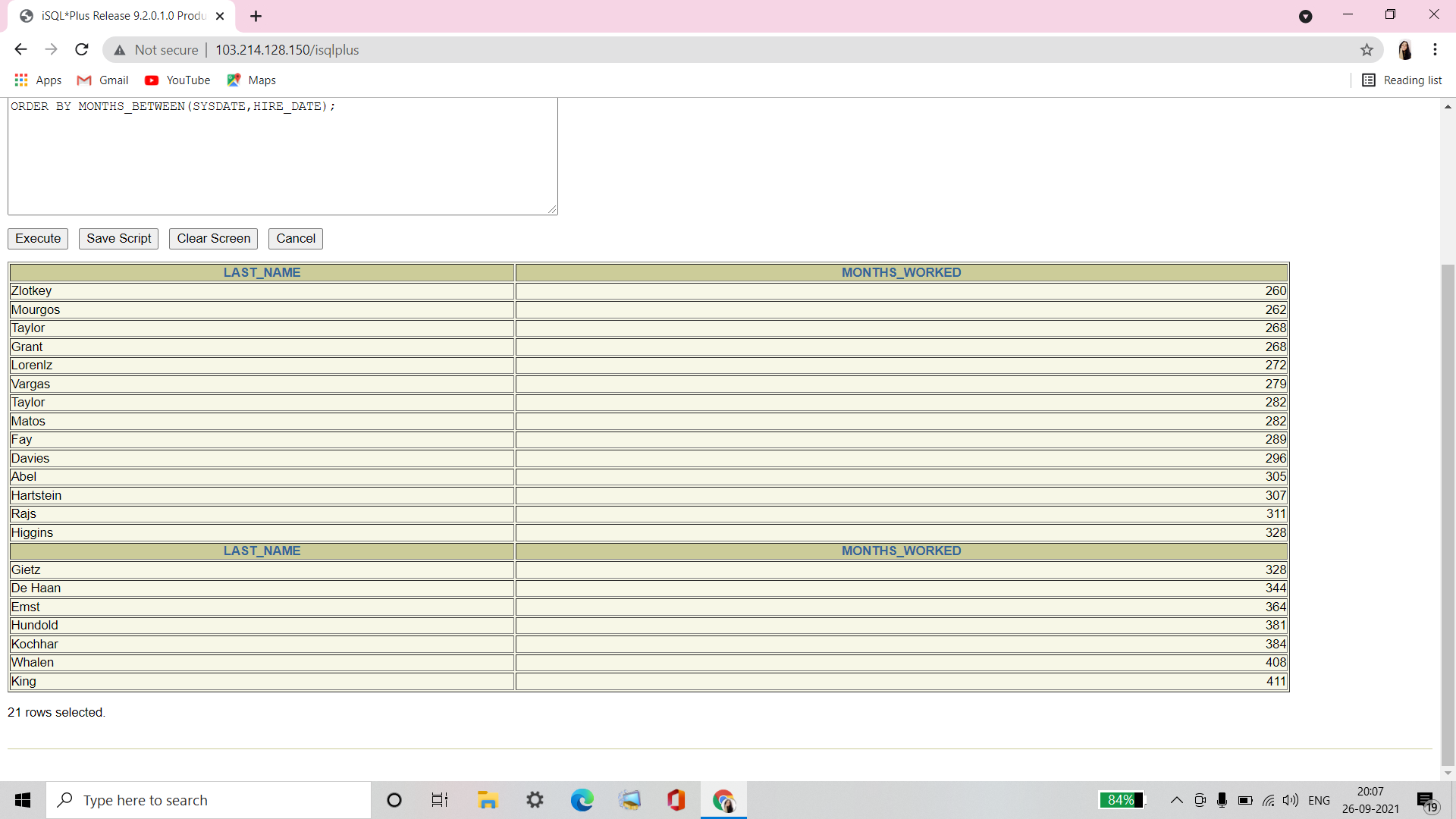
Ans6.

SELECT LAST\_NAME,ROUND(MONTHS\_BETWEEN (SYSDATE,HIRE\_DATE)) MONTHS\_WORKED

FROM EMPLOYEES

ORDER BY MONTHS\_BETWEEN(SYSDATE,HIRE\_DATE);

**Verification table -**



**Q7. 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.**

Ans7.

SELECT LAST\_NAME || ' earns '

|| TO\_CHAR(SALARY, 'fm$99,999.00')

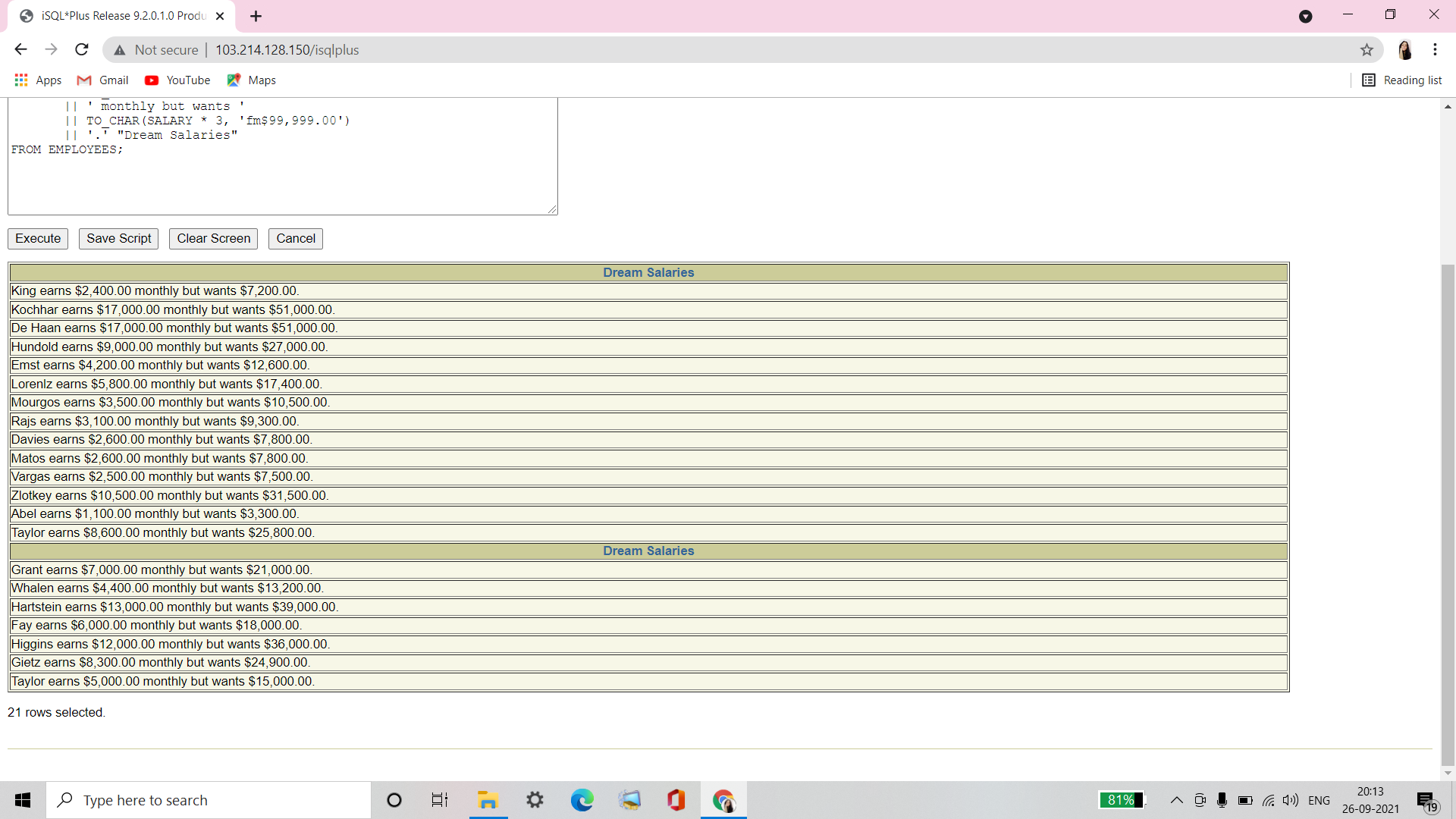
|| ' monthly but wants '

|| TO\_CHAR(SALARY \* 3, 'fm$99,999.00')

|| '.' "Dream Salaries"

FROM EMPLOYEES;

**Verification table -**



**Q8. 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.**

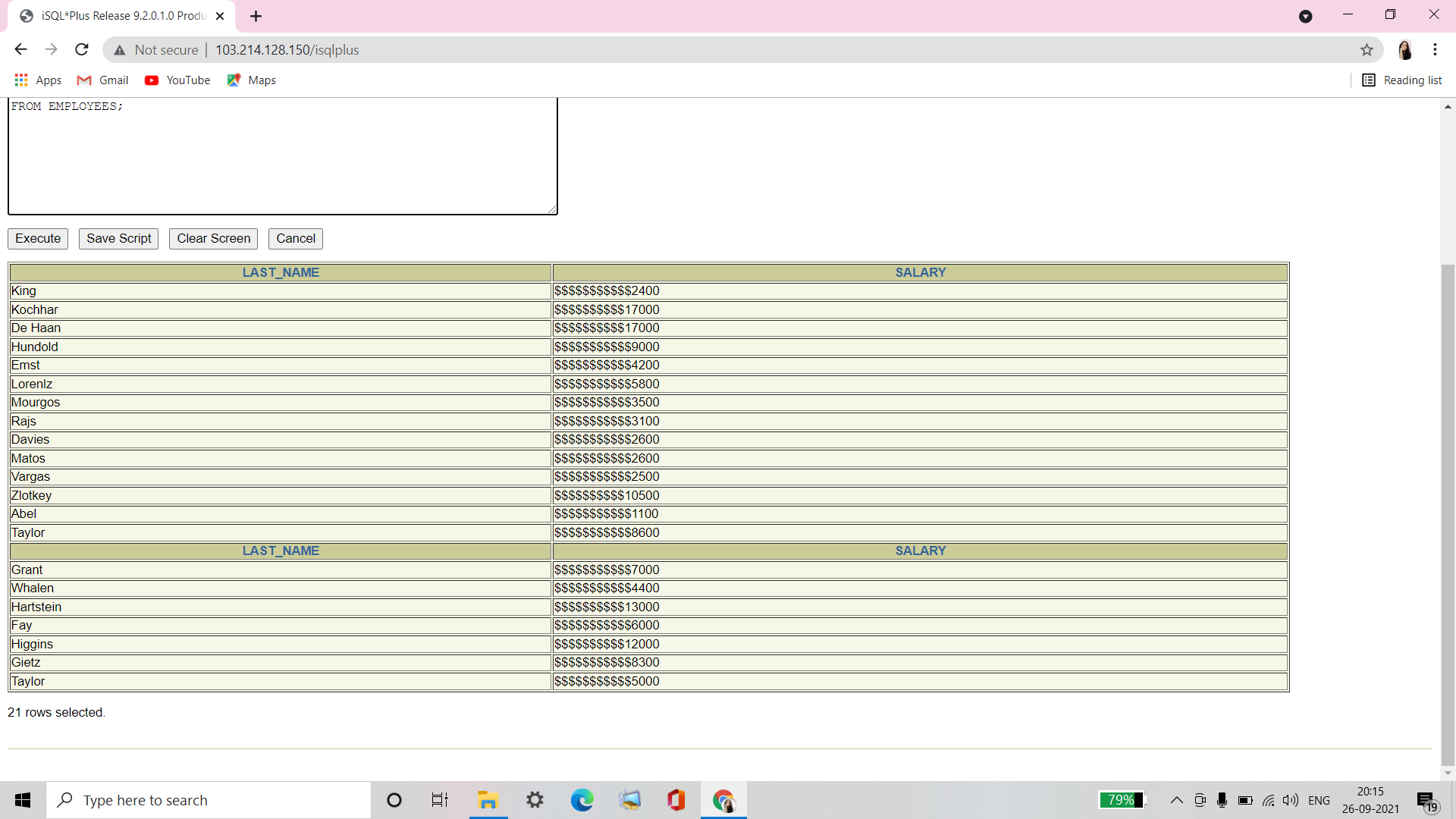
Ans8.

SELECT LAST\_NAME,

LPAD(SALARY, 15, '$') SALARY

FROM EMPLOYEES;

**Verification table -**



**Q9. 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.”**

Ans9.

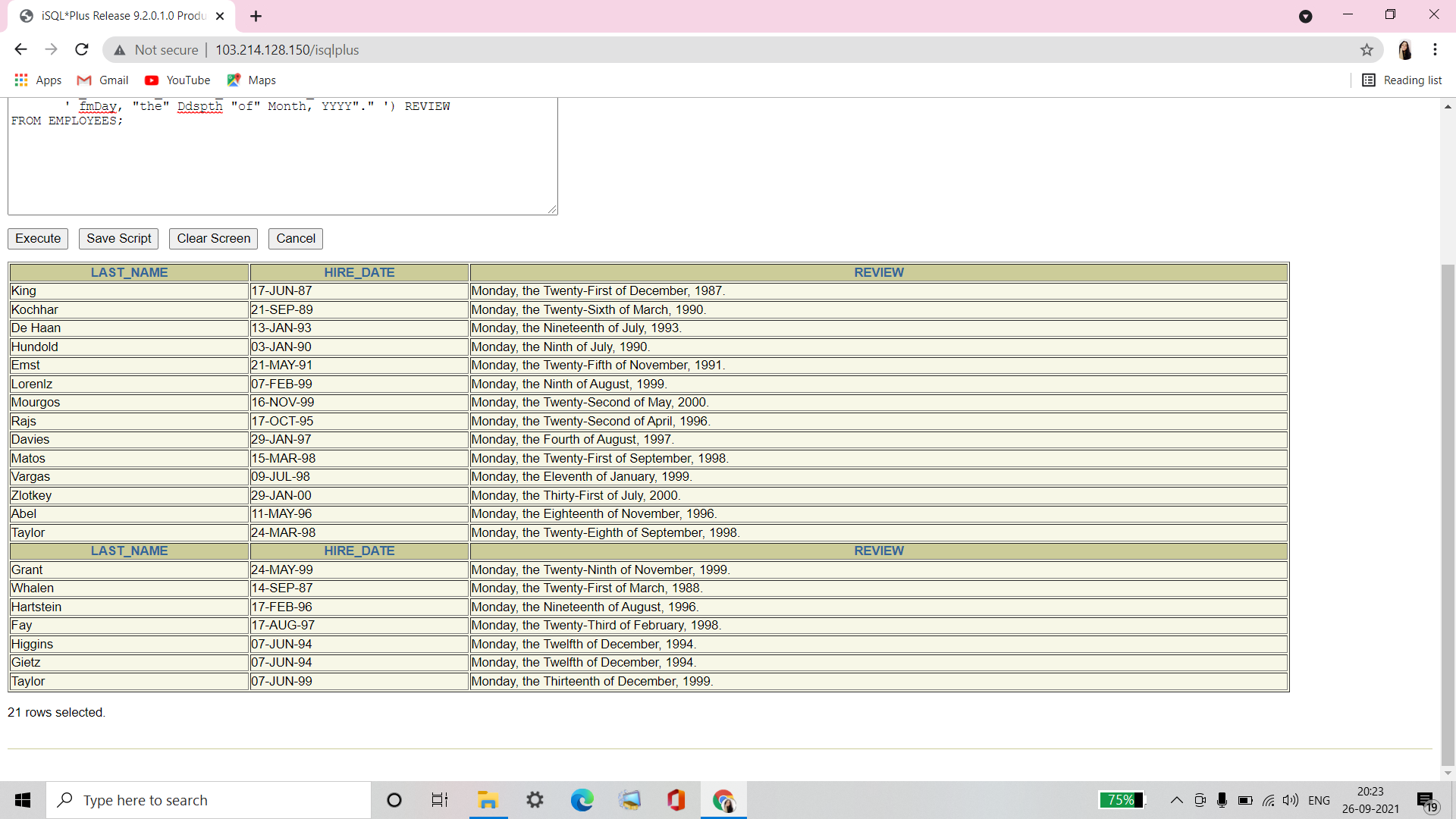
SELECT LAST\_NAME, HIRE\_DATE,

TO\_CHAR(NEXT\_DAY(ADD\_MONTHS(HIRE\_DATE, 6),'MONDAY'),

' fmDay, "the" Ddspth "of" Month, YYYY"." ') REVIEW

FROM EMPLOYEES;

**Verification table -**



**Q10. 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.**

Ans10.

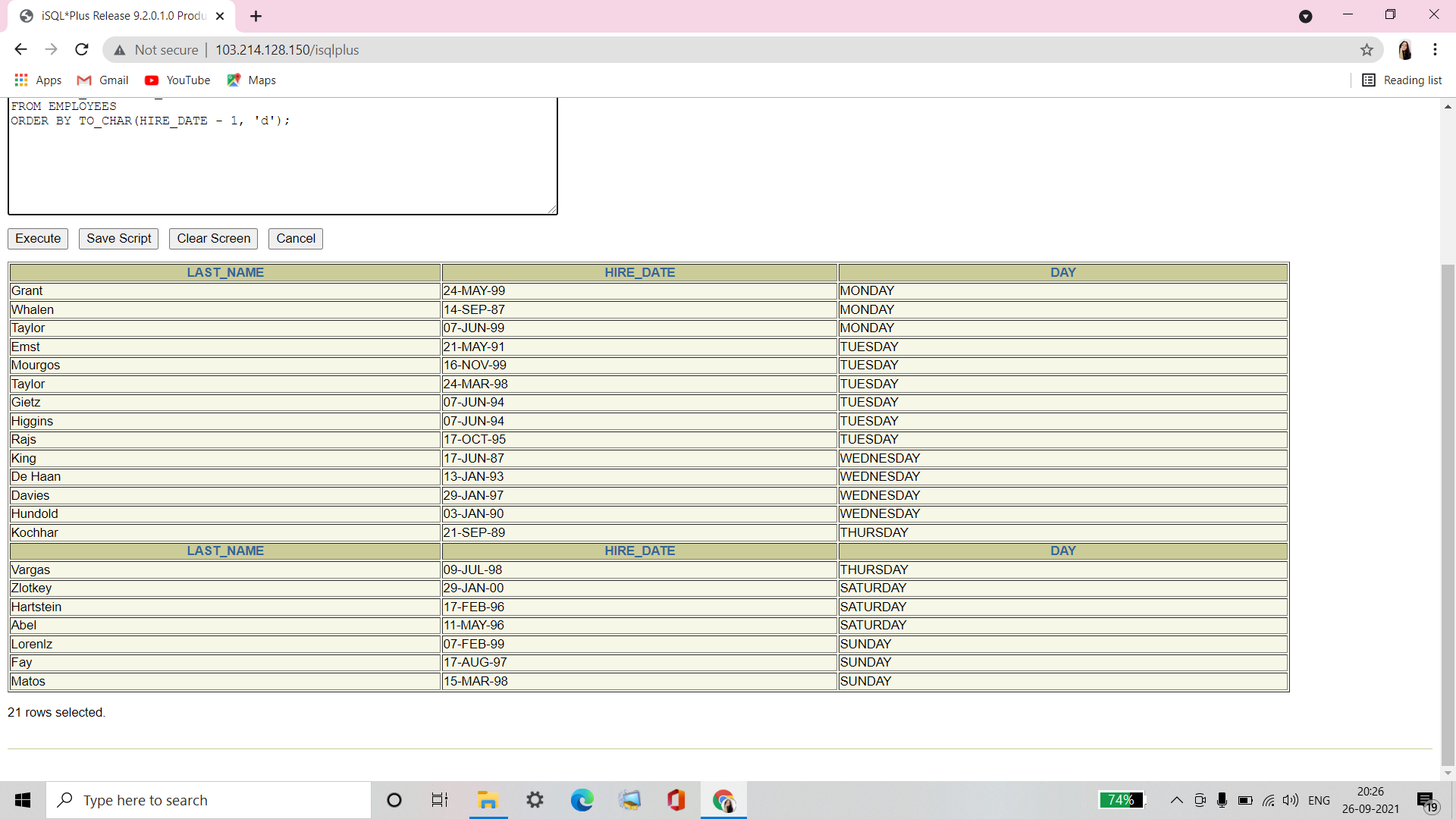
SELECT LAST\_NAME, HIRE\_DATE,

TO\_CHAR(HIRE\_DATE, 'DAY') DAY

FROM EMPLOYEES

ORDER BY TO\_CHAR(HIRE\_DATE - 1, 'd');

**Verification table -**



**Q11. 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.**

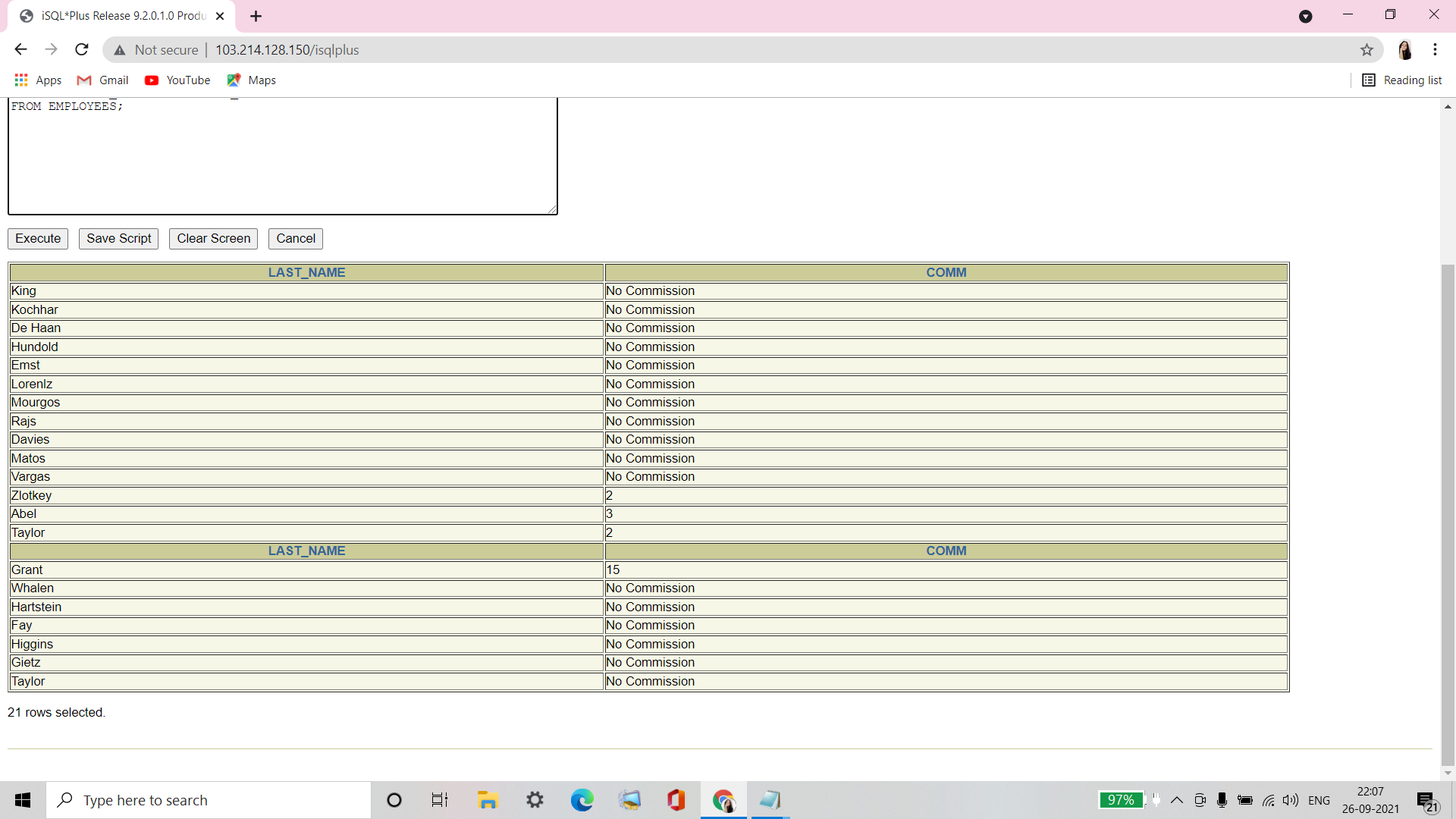
Ans11.

SELECT LAST\_NAME,

NVL(TO\_CHAR(COMMISSION\_PCT), 'No Commission') COMM

FROM EMPLOYEES;

**Verification table -**



**Q12. 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\_SALAIES.**

Ans12.

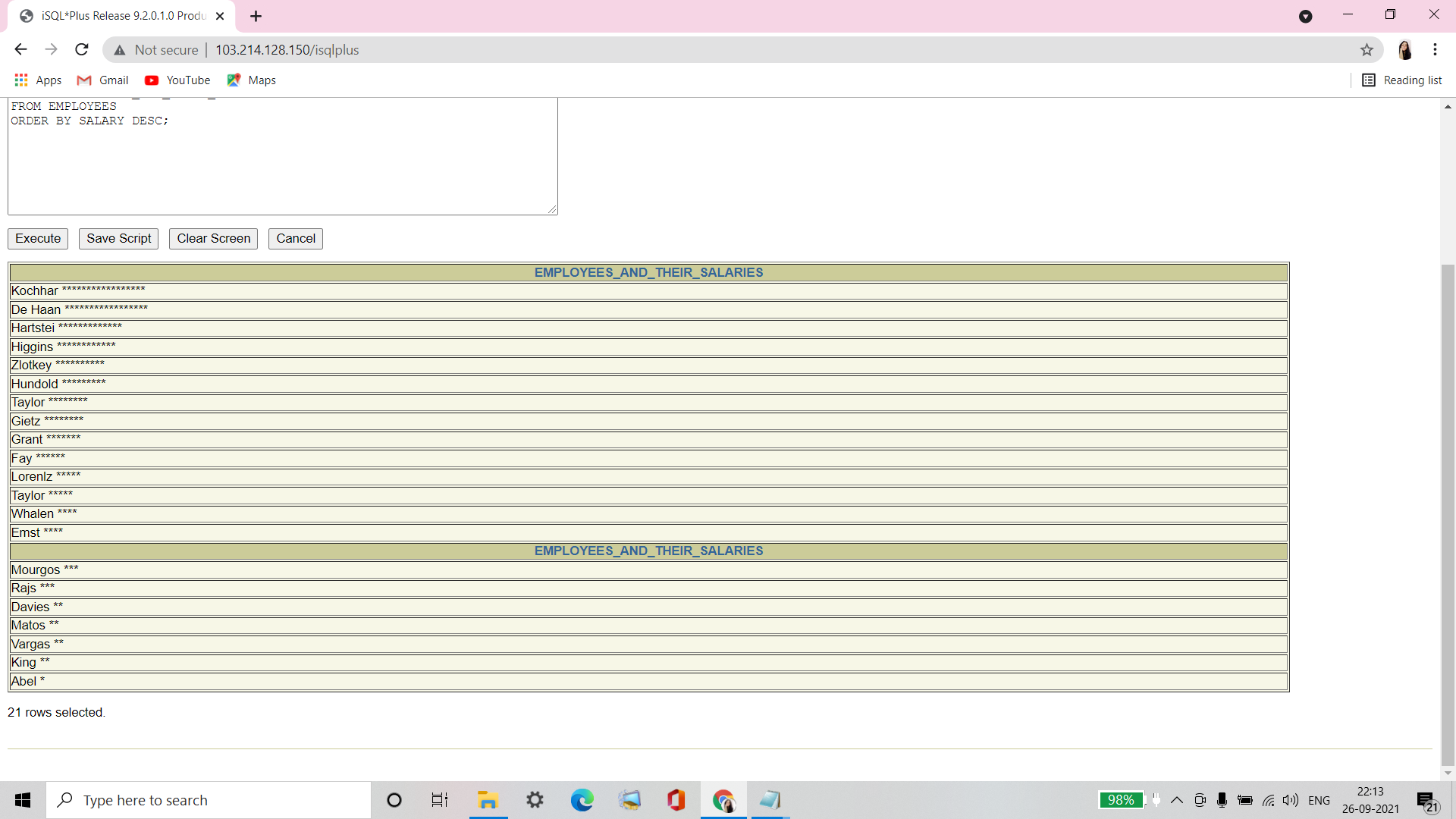
SELECT RPAD(LAST\_NAME, 8) || ' ' || RPAD(' ', SALARY/1000+1, '\*')

EMPLOYEES\_AND\_THEIR\_SALARIES

FROM EMPLOYEES

ORDER BY SALARY DESC;

**Verification table -**



**Q13. 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** |

Ans13.

SELECT JOB\_ID, DECODE (JOB\_ID,

'ST\_CLERK', 'E',

'SA\_REP', 'D',

'IT\_PROG', 'C',

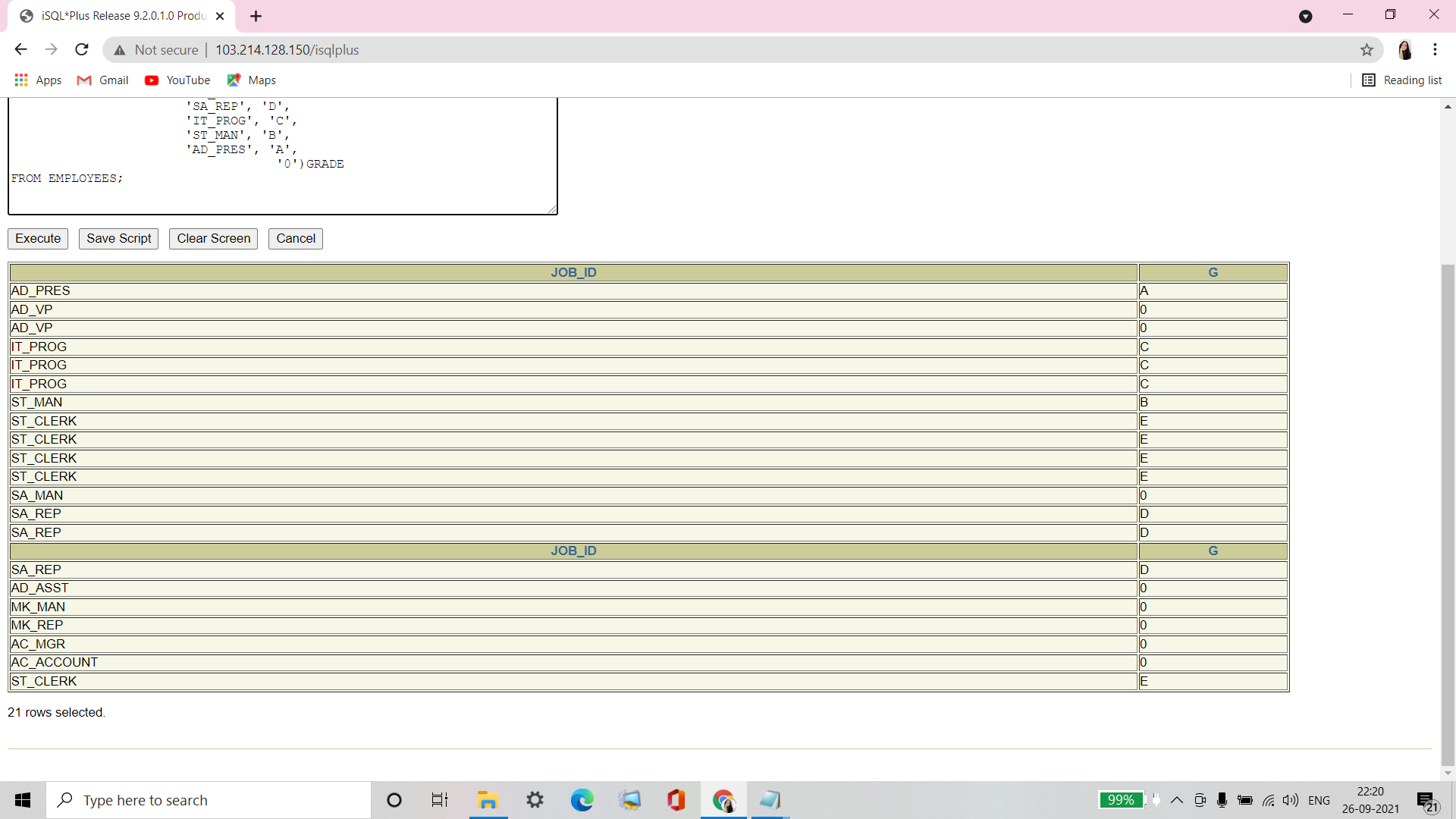
'ST\_MAN', 'B',

'AD\_PRES', 'A',

'0')GRADE

FROM EMPLOYEES;

**Verification table -**



**Q14. Rewrite the statement in the preceding question using the CASE syntax.**

Ans14.

SELECT JOB\_ID, CASE JOB\_ID

WHEN 'ST\_CLERK' THEN 'E'

WHEN 'SA\_REP' THEN 'D'

WHEN 'IT\_PROG' THEN 'C'

WHEN 'ST\_MAN' THEN 'B'

WHEN 'AD\_PRES' THEN 'A'

ELSE '0' END GRADE

FROM EMPLOYEES;

**Verification table -**

