DBMS LAB 3

Q1

Write a query that displays the employee's last names all in uppercase and the length of the last names, for all employees whose name starts with S, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

Query

```
SELECT UPPER(LAST_NAME) AS LAST_NAME_UPPER, LENGTH(LAST_NAME) AS
NAME_LENGTH
FROM hr.employees
WHERE LAST_NAME LIKE 'S%' OR
LAST_NAME LIKE 'A%' OR
LAST_NAME LIKE 'M%'
ORDER BY LAST_NAME;
```

Output

LAST_NAME_UPPER	NAME_LENGTH
ABEL	4
ANDE	4
ATKINSON	8
AUSTIN	6
MALLIN	6
MARKLE	6
MARLOW	6
MARVINS	7
MATOS	5
MAVRIS	6
MCCAIN	6
MCEWEN	6
MIKKILINENI	11
MOURGOS	7
SARCHAND	8
SCIARRA	7
SEO	3
SEWALL	6
SMITH	5
SMITH	5
STILES	6
SULLIVAN	8
SULLY	5

Download CSV

23 rows selected.

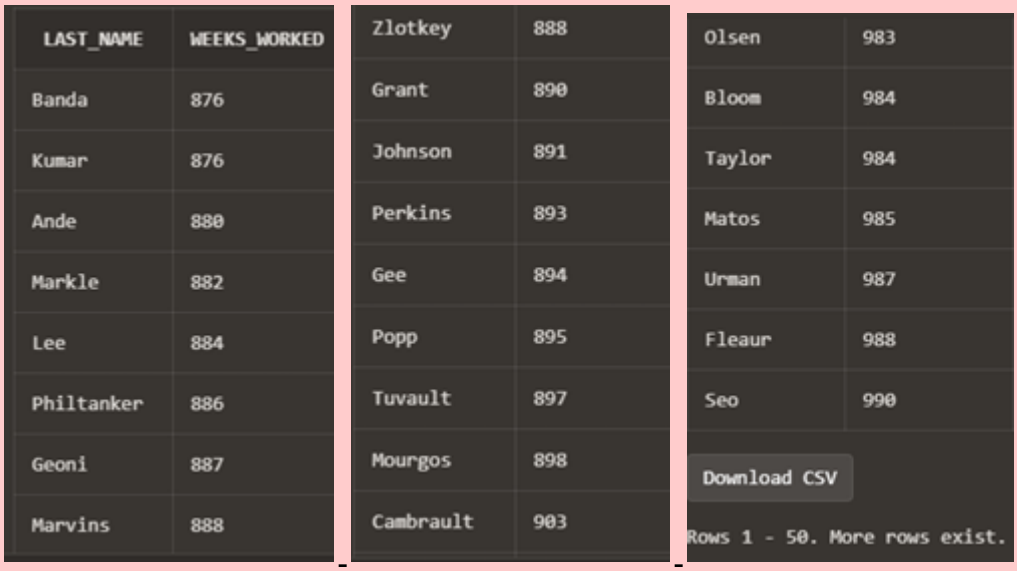
Q2

For each employee, display the employee's last name, and calculate the number of weeks between today and the date the employee was hired. Label the column WEEKS_WORKED . Order your results by the number of WEEKs employed. Round the number of weeks up to the closest whole number.

Query

```
SELECT LAST_NAME,
ROUND((SYSDATE-HIRE_DATE)/7) AS WEEKS_WORKED
 FROM hr.employees
 ORDER BY WEEKS_WORKED;
```

Output



LAST_NAME	WEEKS_WORKED
Banda	876
Kumar	876
Ande	880
Markle	882
Lee	884
Philtanker	886
Geoni	887
Marvins	888
Zlotkey	888
Grant	890
Johnson	891
Perkins	893
Gee	894
Popp	895
Tuvault	897
Mourgos	898
Cambrault	903
Olsen	983
Bloom	984
Taylor	984
Matos	985
Urman	987
Fleaur	988
Seo	990

Download CSV

Rows 1 - 50. More rows exist.

Q3

Study the NULLIF Function and solve the following question: Write a query to display: employee_id, salary, commission_pct, total_earnings, which is calculated as: total_earnings = salary + (salary×commission_pct) If commission_pct is NULL, it should be treated as 0 to prevent calculation errors.

Query

```
SELECT EMPLOYEE_ID, SALARY,
COMMISSION_PCT,
 (SALARY + SALARY* NVL(COMMISSION_PCT, 0)) AS TOTAL_EARNINGS
FROM hr.employees;
```

Output

EMPLOYEE_ID	SALARY	COMMISSION_PCT	TOTAL_EARNINGS
100	24000	-	24000
101	17000	-	17000
102	17000	-	17000
103	9000	-	9000
104	6000	-	6000
105	4800	-	4800
106	4800	-	4800
107	4200	-	4200

143	2600	-	2600
144	2500	-	2500
145	14000	.4	19600
146	13500	.3	17550
147	12000	.3	15600
148	11000	.3	14300
149	10500	.2	12600

Download CSV

Rows 1 - 50. More rows exist.

Q4

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 similar to "Monday, the Thirty-First of July, 2000.

Query

SELECT LAST_NAME,

TO_CHAR(HIRE_DATE, 'Day, "the" Ddsp "of" Month,YYYY') AS HIRE_DATE,
TO_CHAR(NEXT_DAY(ADD_MONTHS(HIRE_DATE, 6), 'MONDAY'), 'Day, "the" Ddsp "of" Month, YYYY') AS REVIEW

FROM hr.employees;

Output

LAST_NAME	HIRE_DATE	REVIEW
King	Tuesday , the Seventeen of June ,2003	Monday , the Twenty-Two of December , 2003
Kochhar	Wednesday, the Twenty-One of September,2005	Monday , the Twenty-Seven of March , 2006
De Haan	Saturday , the Thirteen of January ,2001	Monday , the Sixteen of July , 2001
Hunold	Tuesday , the Three of January ,2006	Monday , the Ten of July , 2006
Ernst	Monday , the Twenty-One of May ,2007	Monday , the Twenty-Six of November , 2007
Austin	Saturday , the Twenty-Five of June ,2005	Monday , the Twenty-Six of December , 2005
Pataballa	Sunday , the Five of February ,2006	Monday , the Seven of August , 2006
Matos	Wednesday, the Fifteen of March ,2006	Monday , the Eighteen of September, 2006
Vargas	Sunday , the Nine of July ,2006	Monday , the Fifteen of January , 2007
Russell	Friday , the One of October ,2004	Monday , the Four of April , 2005
Partners	Wednesday, the Five of January ,2005	Monday , the Eleven of July , 2005
Errazuriz	Thursday , the Ten of March ,2005	Monday , the Twelve of September, 2005
Cambrault	Monday , the Fifteen of October ,2007	Monday , the Twenty-One of April , 2008
Zlotkey	Tuesday , the Twenty-Nine of January ,2008	Monday , the Four of August , 2008

Download CSV

Rows 1 - 50. More rows exist.

Q5

Create a query to display the first name and salary for all employees. Format the salary to be 15 characters long, right-padded with &. Label the column SALARY.

Query

```
SELECT FIRST_NAME, RPAD(TO_CHAR(SALARY), 15, '&') AS SALARY

FROM hr.employees;
```

Output

FIRST_NAME	SALARY
Steven	24000&&&&&&&&&&&
Neena	17000&&&&&&&&&&&
Lex	17000&&&&&&&&&&&
Alexander	9000&&&&&&&&&&&&
Bruce	6000&&&&&&&&&&&&
David	4800&&&&&&&&&&&&
Valli	4800&&&&&&&&&&&&
Diana	4200&&&&&&&&&&&&
Randall	2600&&&&&&&&&&&&
Peter	2500&&&&&&&&&&&&
John	14000&&&&&&&&&&&
Karen	13500&&&&&&&&&&&
Alberto	12000&&&&&&&&&&&
Gerald	11000&&&&&&&&&&&
Eleni	10500&&&&&&&&&&&

Download CSV

Rows 1 - 50. More rows exist.

Q6

Write a query that produces the following for each employee: earns monthly but wants Label the column Dream Salaries.

Query

```
SELECT LAST_NAME|' earns ' |SALARY|' monthly but wants ' |(SALARY * 3) AS "Dream Salaries"

 FROM hr.employees;
```

Output

Dream Salaries
King earns 24000 monthly but wants 72000
Kochhar earns 17000 monthly but wants 51000
De Haan earns 17000 monthly but wants 51000
Hunold earns 9000 monthly but wants 27000
Ernst earns 6000 monthly but wants 18000
Austin earns 4800 monthly but wants 14400
Pataballa earns 4800 monthly but wants 14400
Lorentz earns 4200 monthly but wants 12600
Matos earns 2600 monthly but wants 7800
Vargas earns 2500 monthly but wants 7500
Russell earns 14000 monthly but wants 42000
Partners earns 13500 monthly but wants 40500
Errazuriz earns 12000 monthly but wants 36000
Cambrault earns 11000 monthly but wants 33000
Zlotkey earns 10500 monthly but wants 31500

Download CSV

Rows 1 - 50. More rows exist.

#

Q7

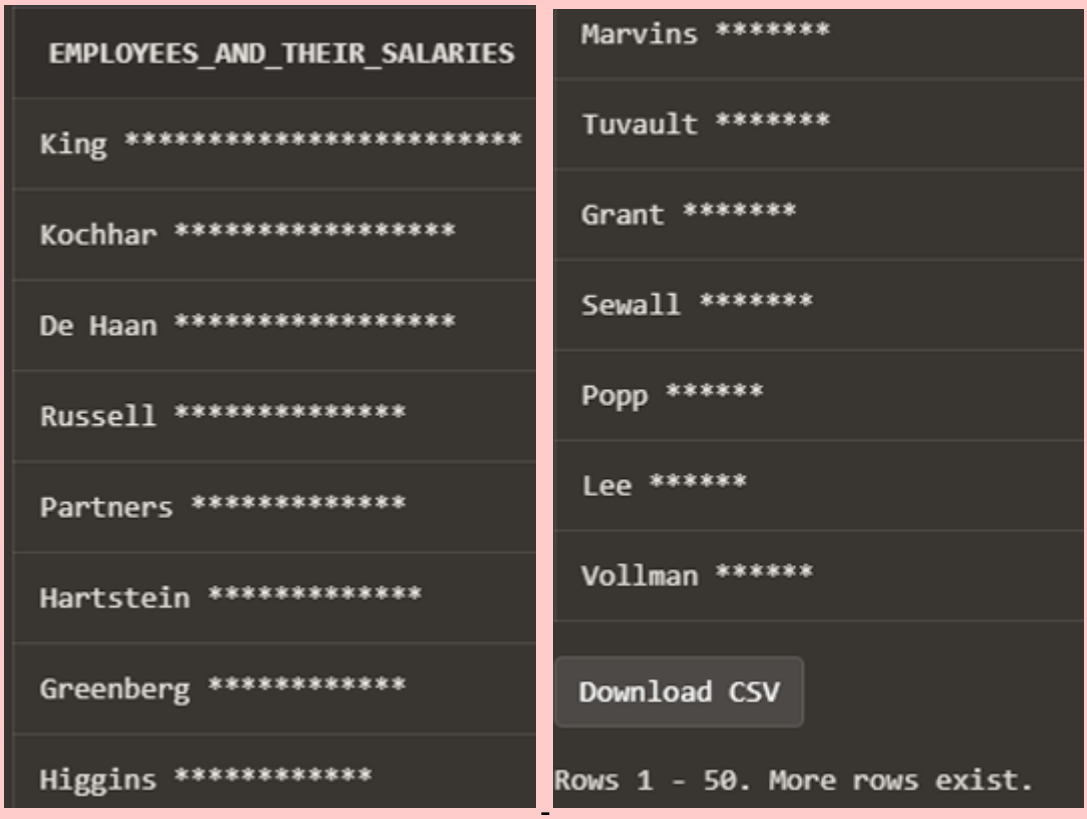
Create a query that displays the employees' last names and indicates the amounts of their annual salaries with asterisks. Each asterisk signifies a thousand dollars. Sort the data in descending order of salary. Label the column EMPLOYEES_AND_THEIR_SALARIES.

Query

```
SELECT LAST_NAME|' ' |RPAD('*', TRUNC(salary / 1000), '*') AS EMPLOYEES_AND_THEIR_SALARIES
FROM hr.employees ORDER

BY SALARY DESC;
```

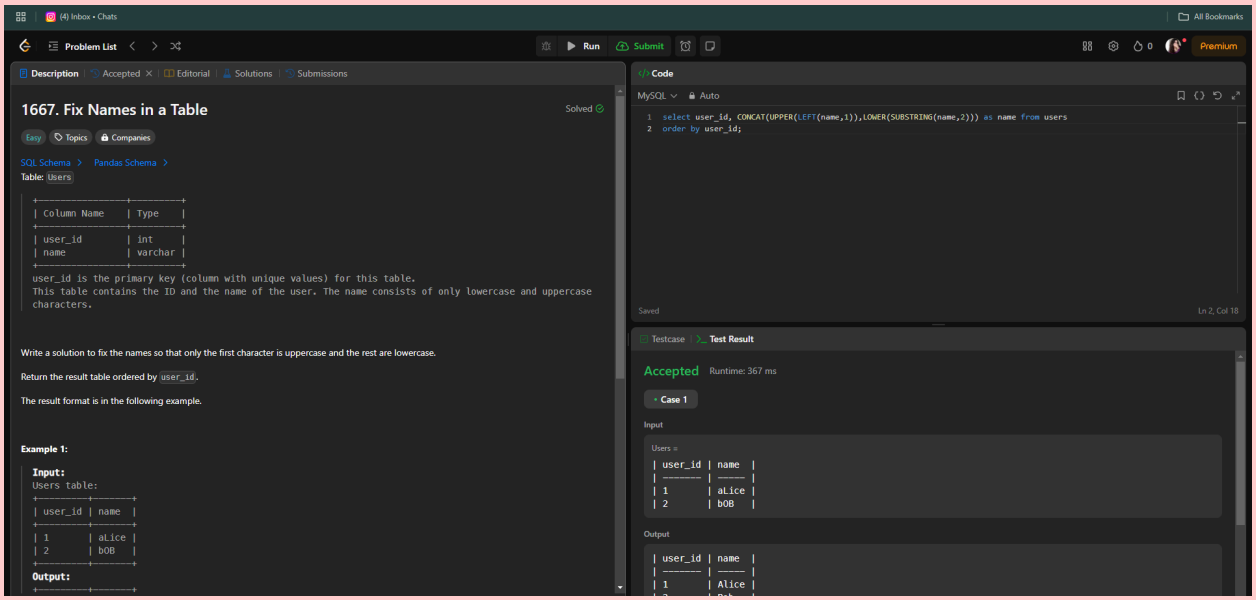
Output



Q8

Solve first 2 questions of "Advanced String Functions / Regex / Clause" section of SQL 50 Badge on leetcode.

1667. FIX NAMES IN A TABLE



1527. PATIENTS WITH A CONDITION