

EXERCISE-6

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

```
SELECT DATE('now') AS "Date";
```

Output

Date
2025-08-26

2. The HR department needs a report to display the employee number, last name, salary, and increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

```
SELECT
    employee_id,
    last_name,
    salary,
    CAST(salary * 1.155 AS INTEGER) AS "New Salary"
FROM employees;
```

Output

employee_id	last_name	salary	New Salary
100	Jackson	5000	5775
101	Andrews	4200	4851
102	Morrison	3900	4504
103	Hunt	3600	4158
104	Smith	7000	8085
105	Johnson	5500	6352
106	Miller	4700	5428

3. Modify your query lab_03_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase.

```
SELECT
    employee_id,
    last_name,
    salary,
    CAST(salary * 1.155 AS INTEGER) AS "New Salary",
    CAST(salary * 1.155 AS INTEGER) - salary AS "Increase"
FROM employees;
```

Output

employee_id	last_name	salary	New Salary	Increase
100	Jackson	5000	5775	775
101	Andrews	4200	4851	651
102	Morrison	3900	4504	604
103	Hunt	3600	4158	558
104	Smith	7000	8085	1085
105	Johnson	5500	6352	852
106	Miller	4700	5428	728

4. Write a query that displays the last name (with the first letter uppercase and all other letters

lowercase) and the length of the last name for all employees whose name starts with the letters J, A,

or M. Give each column an appropriate label. Sort the results by the employees' last names.

```
SELECT
    UPPER(SUBSTR(last_name, 1, 1)) || LOWER(SUBSTR(last_name, 2)) AS "Last Name",
    LENGTH(last_name) AS "Length"
FROM employees
WHERE UPPER(SUBSTR(last_name, 1, 1)) IN ('J', 'A', 'M')
ORDER BY last_name;
```

Output

Last Name	Length
Andrews	7
Jackson	7
Johnson	7
Miller	6
Morrison	8

5. Rewrite the query so that the user is prompted to enter a letter that starts the last name. For

example, if the user enters H when prompted for a letter, then the output should show all

employees whose last name starts with the letter H.

```
SELECT
    UPPER(SUBSTR(last_name, 1, 1)) || LOWER(SUBSTR(last_name, 2)) AS "Last Name",
    LENGTH(last_name) AS "Length"
FROM employees
WHERE UPPER(SUBSTR(last_name, 1, 1)) = 'H'
ORDER BY last_name;
```

Output

Last Name	Length
Hunt	4

6. The HR department wants to find the length of employment for each employee. For each

employee, display the last name and calculate the number of months between today and the date on

which 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.

```
SELECT
    last_name,
    CEIL((JULIANDAY('now') - JULIANDAY(hire_date)) / 30.4375) AS "MONTHS_WORKED"
FROM employees
ORDER BY "MONTHS_WORKED";
```

Output

last_name	MONTHS_WORKED
Smith	33
Morrison	46
Hunt	66
Andrews	75
Johnson	84
Miller	114
Jackson	128

7. Create a report that produces the following for each employee:

<employee last name> earns <salary> monthly but wants <3 times salary>. Label the column

Dream Salaries.

```
SELECT
    last_name || ' earns ' || salary || ' monthly but wants ' || (salary * 3) AS "Dream Salaries"
FROM employees;
```

Output

Dream Salaries
Jackson earns 5000.0 monthly but wants 15000.0
Andrews earns 4200.0 monthly but wants 12600.0
Morrison earns 3900.0 monthly but wants 11700.0
Hunt earns 3600.0 monthly but wants 10800.0
Smith earns 7000.0 monthly but wants 21000.0
Johnson earns 5500.0 monthly but wants 16500.0
Miller earns 4700.0 monthly but wants 14100.0

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 the \$ symbol. Label the column SALARY.

```
SELECT
    last_name,
    printf('%15s', '$' || salary) AS "SALARY"
FROM employees;
```

Output

last_name	SALARY
Jackson	\$5000.0
Andrews	\$4200.0
Morrison	\$3900.0
Hunt	\$3600.0
Smith	\$7000.0
Johnson	\$5500.0
Miller	\$4700.0

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."

```
SELECT
    last_name,
    hire_date,
    DATE(hire_date, '+6 months', 'weekday 1') AS "REVIEW" -- Monday = 1
FROM employees;
```

Output

last_name	hire_date	REVIEW
Jackson	2015-01-10	2015-07-13
Andrews	2019-06-20	2019-12-23
Morrison	2021-11-15	2022-05-16
Hunt	2020-03-05	2020-09-07
Smith	2022-12-01	2023-06-05
Johnson	2018-09-17	2019-03-18
Miller	2016-02-28	2016-08-29

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.

```
SELECT
    last_name,
    hire_date,
    STRFTIME('%w', hire_date) AS "DAY_INDEX", -- 0 = Sunday, 1 = Monday
    CASE STRFTIME('%w', hire_date)
        WHEN '0' THEN 'Sunday'
        WHEN '1' THEN 'Monday'
        WHEN '2' THEN 'Tuesday'
        WHEN '3' THEN 'Wednesday'
        WHEN '4' THEN 'Thursday'
        WHEN '5' THEN 'Friday'
        WHEN '6' THEN 'Saturday'
    END AS "DAY"
FROM employees
ORDER BY
    CASE STRFTIME('%w', hire_date)
        WHEN '1' THEN 1
        WHEN '2' THEN 2
        WHEN '3' THEN 3
        WHEN '4' THEN 4
        WHEN '5' THEN 5
        WHEN '6' THEN 6
        WHEN '0' THEN 7
    END;
```

Output

last_name	hire_date	REVIEW
Jackson	2015-01-10	2015-07-13
Andrews	2019-06-20	2019-12-23
Morrison	2021-11-15	2022-05-16
Hunt	2020-03-05	2020-09-07
Smith	2022-12-01	2023-06-05
Johnson	2018-09-17	2019-03-18
Miller	2016-02-28	2016-08-29