Practice 3 Solutions

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

```
SELECT    sysdate "Date"
FROM      dual;
```

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`.

```
SELECT    employee_id, last_name, salary,
          ROUND(salary * 1.15, 0) "New Salary"
FROM      employees;
```

3. Run your query in the file `lab3_2.sql`.

```
SELECT  employee_id, last_name, salary,
        ROUND(salary * 1.15, 0) "New Salary"
FROM    employees;
```

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.

```
SELECT    employee_id, last_name, salary,
  ROUND(salary * 1.15, 0) "New Salary",
  ROUND(salary * 1.15, 0) - salary "Increase"
FROM      employees;
```

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 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.

```
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;
```

---

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.

   Note: Your results will differ.

```
SELECT    last_name, ROUND(MONTHS_BETWEEN
                (SYSDATE, hire_date)) MONTHS_WORKED
FROM      employees
ORDER BY MONTHS_BETWEEN(SYSDATE, hire_date);
```

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`.

```
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;
```

---

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.

```
SELECT    last_name,
          LPAD(salary, 15, '$') SALARY
FROM      employees;
```

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,
  TO_CHAR(NEXT_DAY(ADD_MONTHS(hire_date, 6),'MONDAY'),
          'fmDay, "the" Ddspth "of" Month, YYYY') REVIEW
FROM      employees;
```

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,
          TO_CHAR(hire_date, 'DAY') DAY
FROM      employees
ORDER BY TO_CHAR(hire_date - 1, 'd');
```

---

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.

```
SELECT    last_name,
          NVL(TO_CHAR(commission_pct), 'No Commission') COMM
FROM      employees;
```

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.

```
SELECT    rpad(last_name, 8)||' '|| rpad(' ', salary/1000+1, '*')
          EMPLOYEES_AND_THEIR_SALARIES
FROM      employees
ORDER BY salary DESC;
```

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

```
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;
```

---

14. Rewrite the statement in the preceding question using the `CASE` syntax.

```
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;
```