Practice SQL Scalar Functions

This page provides exercises and solutions to help you practice SQL Scalar Functions. These exercises are based on the Oracle HR schema, and may be performed online or by running the sample schema scripts on your local database server. For additional exercises in other subjects, use this link.

String Functions Practice

1. Display the first name in lower case and last name in upper case, for all employees whose employee number is in the range between 80 and 150.
2. Display the first name and last name for all employees whose family name is King, perform this exercise with a case-insensitive search (regardless of the capitalization used for the values within *last name* column).
3. Generating new email address
    1. For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three first letters of last name, concatenated with *@oracle.com*.
    2. For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three last letters of last name, concatenated with *@oracle.com*.
4. Using the CONCAT function
    1. For each employee, use the CONCAT function to display the first name concatenated with the last name.
    2. For each employee, use the CONCAT function to display the first

name concatenated with the last name, concatenated with hire date.

5. Display the last name for all employees where last name's length is greater than 8 characters.
6. Phone numbers:
    1. For each employee, display the first name, last name, phone number and a new phone number using the REPLACE function. in the new phone number replace all occurrences of *515* with *815.*
    2. For each employee, display the first name, last name, phone number and a new phone number using the REPLACE function. in the new phone number replace all prefixes of *515* with *815.*

Numeric Functions Practice

1. For each employee, display :
    1. first name
    2. salary
    3. salary after a raise of 12%
    4. salary after a raise of 12%, expressed as a whole number (ROUND).
    5. salary after a raise of 12%, round down to the nearest whole number.

Date Functions Practice

1. For each employee, display the first name, hire date, hire date minus 10 days, hire date plus one month, and the day difference between current date and hire date.
2. For each employee, display the first name, last name, hire date, number of months he works in the company, and number of years he works in the company.
3. For each employee, display the first name, hire date, and hire date plus one year.
4. For each employee, display the first name, hire date, hire date rounded up to the nearest year, and hire date rounded up to the nearest month.

Conversion Functions Practice

1. For each employee, display the first name, the day of his hire date, and the year of his hire date.
2. Display the last name in upper case, the salary in format model : '9,999.999', and hire date in format model: 'DD/MM/YYYY', for all employees whose last name begins with the letter *D* or *K*.

NULL-Related Functions Practice

1. Commission Percentage
    1. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display 0 instead of NULL.
    2. For each employee, display the first name, last name, salary and commission percentage. If an employee doesn't earn a commission, display "No Commission" instead of NULL.

Case Function Practice

1. For each employee, display the first name, last name, salary, and a salary grade based on these conditions :
    1. if the salary is between 0 and 5000 – salary grade level is A
    2. if the salary is between 5001 and 15000 – salary grade level is B
    3. if the salary is between 15001 and 20000 – salary grade level is C
    4. for any other range – salary grade level is D

Solutions – Oracle

These solutions apply to Oracle, for solutions that apply to SQL Server click here.

```
01 -- 1. LOWER / UPPER
02 SELECT LOWER(first_name) , UPPER(last_name)
03 FROM employees
04 WHERE employee_id BETWEEN 80 AND 150
05 -- 2. LOWER / UPPER
06 SELECT first_name , last_name
```

```
FROM employees
WHERE UPPER(last_name) = 'KING'
-- or
SELECT first_name , last_name
FROM employees
WHERE LOWER(last_name) = 'king'
-- 3. SUBSTR / LENGTH
-- a
SELECT first_name , last_name ,
    SUBSTR(first_name , 1 , 1) || SUBSTR(last_name , 1 , 3)
FROM employees
-- b
SELECT first_name , last_name ,
       SUBSTR(first_name , 1 , 1)
       || SUBSTR(last_name , LENGTH(last_name)-2 , 3) || '@o
FROM employees
-- 4. CONCAT
-- a
SELECT CONCAT (first_name , last_name)
FROM employees
-- b
SELECT CONCAT(first_name , CONCAT(last_name , hire_date))
FROM employees
-- 5. LENGTH
SELECT last_name
FROM employees
WHERE LENGTH(last_name) > 8
-- 6. REPLACE
-- a
SELECT first_name , last_name , REPLACE(phone_number , '515'
FROM employees
-- b
SELECT first_name , last_name , REPLACE(SUBSTR(phone_number
FROM employees
-- 7. ROUND / TRUNC
SELECT first_name , salary , salary * 1.12 , ROUND(salary *
FROM employees
-- 8. ADD_MONTHS
SELECT first_name , hire_date , hire_date - 10 , ADD_MONTHS(
```

```
FROM employees
-- 9. MONTHS_BETWEEN
SELECT first_name , last_name , hire_Date ,
MONTHS_BETWEEN(sysdate , hire_date ) ,
MONTHS_BETWEEN(sysdate , hire_date ) / 12
FROM employees
-- 10. ADD_MONTHS
SELECT first_name , hire_date ,  ADD_MONTHS(hire_date , 12)
FROM employees
-- 11. TRUNC / ROUND
SELECT first_name , hire_date ,
       ROUND(hire_date , 'year') ,
       ROUND(hire_date , 'month')
FROM employees
-- 12. TO_CHAR
SELECT first_name , TO_CHAR(hire_date , 'Day') , TO_CHAR(hir
FROM employees
-- 13. UPPER / TO_CHAR / SUBSTR
SELECT UPPER(last_name) , TO_CHAR(salary , '9,999.999') ,
       TO_CHAR(hire_date , 'DD/MM/YYYY')
FROM employees
WHERE SUBSTR(last_name , 1 , 1) IN ('D' , 'K')
-- 14. NVL / TO_CHAR
-- a
SELECT first_name , last_name , salary , commission_pct ,
       NVL(commission_pct , 0)
FROM employees
-- b
SELECT first_name , last_name , salary , commission_pct , NVl
FROM employees
-- 15. CASE
SELECT first_name , last_name , salary ,
       CASE WHEN salary BETWEEN 0 AND 5000 THEN 'A'
            WHEN salary BETWEEN 5001 AND 15000 THEN 'B'
            WHEN salary BETWEEN 15001 AND 20000 THEN 'C'
            ELSE 'D'
       END AS "SAL_RANKS"
FROM employees
```

Solutions – SQL Server

```
01  -- 1. LOWER / UPPER
02  SELECT LOWER(first_name) , UPPER(last_name)
03  FROM employees
04  WHERE employee_id BETWEEN 80 AND 150
05  -- 2. LOWER / UPPER
06  -- Remember that in SQL Server the search is case-insensitiv
07  SELECT first_name , last_name
08  FROM employees
09  WHERE UPPER(last_name) = 'KING'
10  -- or
11  SELECT first_name , last_name
12  FROM employees
13  WHERE LOWER(last_name) = 'king'
14  -- 3. SUBSTRING / LEN
15  -- a
16  SELECT first_name , last_name ,
17      SUBSTRING(first_name , 1 , 1) + SUBSTRING(last_name , 1
18  FROM employees
19  -- b
20  SELECT first_name , last_name ,
21         SUBSTRING(first_name , 1 , 1)
22         + SUBSTRING(last_name , LEN(last_name)-2 , 3) + '@ora
23  FROM employees
24  -- 4. CONCAT
25  -- a
26  SELECT CONCAT (first_name , last_name)
27  FROM employees
28  -- b
29  SELECT CONCAT(first_name , CONCAT(last_name , hire_date))
30  FROM employees
31  -- 5. LEN
32  SELECT last_name
33  FROM employees
34  WHERE LEN(last_name) > 8
35  -- 6. REPLACE
36  -- a
37  SELECT first_name , last_name , REPLACE(phone_number , '515'
```

```
FROM employees
-- b
SELECT first_name , last_name ,
REPLACE(SUBSTRING(phone_number , 1 ,3) , '515', '815') +
        SUBSTRING(phone_number , 4, LEN(phone_number)-3)
FROM employees
-- 7. ROUND / FLOOR
SELECT first_name , salary , salary * 1.12 , ROUND(salary * 1
FROM employees
-- 8. DATEADD
SELECT first_name , hire_date , hire_date - 10 , DATEADD(MON
FROM employees
-- 9. DATEDIFF
SELECT first_name , last_name , hire_Date ,
DATEDIFF(DAY, GETDATE() , hire_date ) ,
DATEDIFF(YEAR, GETDATE() , hire_date )
FROM employees
-- 10. DATEADD
SELECT first_name , hire_date ,  DATEADD(YEAR, 1, hire_date)
FROM employees
-- 11. DATEADD
SELECT first_name , hire_date ,
       DATEADD(MONTH, datediff(MONTH, 0, hire_date, 0),
       DATEADD(MONTH, datediff(YEAR, 0, hire_date, 0)
FROM employees
-- 12. DAY / YEAR
SELECT first_name , DAY(hire_date) , YEAR(hire_date)
FROM employees
-- 13. UPPER / TO_CHAR / SUBSTRING
SELECT UPPER(last_name) , FORMAT(salary, 'N', 'en-us'),
       FORMAT (hire_date, 'd', 'en-gb' )
FROM employees
WHERE SUBSTRING(last_name , 1 , 1) IN ('D' , 'K')
-- 14. ISNULL / CAST
-- a
SELECT first_name , last_name , salary , commission_pct ,
       ISNULL(commission_pct , 0)
FROM employees
-- b
```

```
77 SELECT first_name , last_name , salary , commission_pct , ISN
78 FROM employees
79 -- 15. CASE
80 SELECT first_name , last_name , salary ,
81        CASE WHEN salary BETWEEN 0 AND 5000 THEN 'A'
82             WHEN salary BETWEEN 5001 AND 15000 THEN 'B'
83             WHEN salary BETWEEN 15001 AND 20000 THEN 'C'
84             ELSE 'D'
85        END AS 'SAL_RANKS'
86 FROM employees
```

Share this:

Google Facebook LinkedIn Twitter Print Email

Login