# **SQL Single-Row Functions\_V1**

**String Functions**

1. Display the customer number, first name in lowercase and last name in uppercase for all customers whose customer number is in the range of 80 and 150.

select customer\_id, lower(first\_name), upper(last\_name) from customers

where customer\_id between 80 and 150;

2. Generating Email Addresses

a. For all customers – display the last name, first name and email address. The email address will be composed from the first letter of first name concatenated with three first letters of last name concatenated with the string “@mymail.com” (For example: MANOJ KUMAR → [MKUM@mymail.com](mailto:MKUM@mymail.com)).

select first\_name, last\_name,

concat(concat(substr(first\_name,1,1), substr(last\_name,1,3)),'@mymail.com') as Email\_address from customers

b. For all customers – display the last name, first name and email address. The email address will be composed from the first letter of first name concatenated with three last letters of last name concatenated with the string “@mymail.com” (For example: MANOJ KUMAR → [MMAR@mymail.com](mailto:MMAR@mymail.com)).

select first\_name, last\_name,

concat(concat(substr(first\_name,1,1), (substr(last\_name,length(last\_name)-2,3))),'@mymail.com') as Email\_address

from customers;

…………..OR………………..

select first\_name, last\_name,

concat(concat(substr(first\_name,1,1), (substr(last\_name,-3,3))),'@mymail.com') as Email\_address

from customers;

3.Display the last name and the length of the last name for all customers where last name’s length is greater than 9 characters.

select last\_name, length(last\_name)

from customers

where length(last\_name)> 9;

4.Phone Numbers :

a. Display the first name, last name, main phone number and a new phone number using the REPLACE function. In the new phone number replace all occurrences of “515” with “$$$”.

select first\_name, last\_name, main\_phone\_num , replace(main\_phone\_num , '515', '$$$') as new\_phone\_num

from customers;

b. Display the first name, last name, main phone number and new phone number using the REPLACE function. In the new phone number replace all prefixes of “515” with “$$$” (only if the first 3 digits of the phone number contains the digits “515” replace those digits with “$$$”).

select first\_name, last\_name, main\_phone\_num,

replace(substr(main\_phone\_num ,1,3), '515', '$$$') || substr(main\_phone\_num,4) as new\_phone\_num

from customers

where main\_phone\_num like '%515%' ;

5.**Numeric Functions**

From *customers* table, for all customers, display :

* 1. first name.
  2. monthly discount.
  3. monthly discount after addition of 19.7%.
  4. monthly discount after addition of 19.7%, expressed as a whole number (ROUND).
  5. monthly discount after addition of 19.7%, round down to the nearest whole number (FLOOR).
  6. monthly discount after addition of 19.7%, round up to the nearest whole number (CEILING).

select first\_name, monthly\_discount, monthly\_discount \*(1+(19.7/100)), round(monthly\_discount \*(1+(19.7/100))),

floor(monthly\_discount \*(1+(19.7/100))), ceil(monthly\_discount \*(1+(19.7/100))) from customers;

**Date Functions**

6. From *Customers* table, for all customers, display the first name, join date, join date minus 10 days, and join date plus one month and the date difference between join date and current date.

select first\_name, join\_date,join\_date-10, add\_months(join\_date , 1), trunc(sysdate - join\_date) as date\_diff\_in\_days

from customers;

7.Display the first name, birthdate and age for all customers whose older than 50.

select first\_name, birth\_date,extract(year from sysdate)- extract(year from birth\_date) as age from customers

where extract(year from sysdate)- extract(year from birth\_date) > 50;

8.Display all the data from *Customers* table, for all customers whose birthdate is today.

select \* from customers

where extract(day from birth\_date) = extract(day from sysdate) and extract(month from birth\_date) = extract(month from sysdate);

9. Display the first name, join date and the difference in years between join date and current date for all customers where today have passed exactly 5 years since they joined the company.

select first\_name, join\_date, trunc(sysdate - join\_date)

from customers

where extract(year from sysdate)- extract(year from join\_date)= 10;

……….or…………..

select first\_name, join\_date, trunc(sysdate - join\_date)

from customers

where trunc(sysdate - join\_date)/365 = 10;

**Conversion functions**

10.Display the first name concatenated with the join date, and last name concatenated with the monthly discount, for all customers.

select first\_name ||''|| join\_date, last\_name ||''|| monthly\_discount from customers;

11.From *Customers* table, for all customers whose last name starts with a *d* or *k,*display:

last name

state in uppercase concatenated with customer number

join date concatenated with birthdate  
in the WHERE clause instead of using LIKE, try to define the filtering condition using SUBSTRING.

select last\_name, upper(state)|| customer\_id, join\_date || birth\_date from customers

where substr(lower(last\_name),1,1) = 'd' or substr(lower(last\_name),1,1) = 'k';

**Null-Related Functions**

12.Phone numbers report:

a.Display the first name, last name, birth date, main phone number and secondary phone number for all customers whose package number equals 27. Replace every null value in main phone number or in secondary phone number with ‘N/A’.

select first\_name, last\_name, birth\_date, main\_phone\_num, secondary\_phone\_num,

NVL(main\_phone\_num, 'N/A'), NVL(secondary\_phone\_num, 'N/A') from customers

where pack\_id = 27;

b.Display the first name, last name, birth date, main phone number, secondary phone number for all customers who was born on 1972. Replace every null value in main phone number or in secondary phone number with ‘N/A’.

select first\_name, last\_name, birth\_date, NVL(main\_phone\_num, 'N/A'), NVL(secondary\_phone\_num, 'N/A')

from customers

where extract(year from birth\_date) = '1972';

**CASE Function**

13.From *Customers* table, for all customers, display the first name, last name, monthly discount and a discount grade based on these conditions :

If the discount is between 0 and 10 – discount grade level is A.

If the discount is between 11 and 20 – discount grade level is B.

If the discount is between 21 and 30 – discount grade level is C.

for any other value – discount grade level is D.

select first\_name, last\_name, monthly\_discount,

case

when monthly\_discount between 0 and 10 then 'A'

when monthly\_discount between 11 and 20 then 'B'

when monthly\_discount between 21 and 30 then 'C'

else 'D'

end

as grade

from customers;