**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;

1. Generating Email Addresses
   1. 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, SUBSTR(FIRST\_NAME,1,1)||SUBSTR(LAST\_NAME,1,3)||'@mymail.com' AS EMAIL

FROM CUSTOMERS;

* 1. 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, SUBSTR(FIRST\_NAME,1,1)||SUBSTR(LAST\_NAME,-3)||'@mymail.com' AS EMAIL

FROM CUSTOMERS;

1. 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;

1. Phone Numbers :
   1. 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\_NUMBER

FROM CUSTOMERS;

* 1. 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\_NUMBER

FROM CUSTOMERS;

**Numeric Functions**

1. 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\*1.197,

ROUND(MONTHLY\_DISCOUNT\*1.197),

FLOOR(MONTHLY\_DISCOUNT\*1.197),

CEIL(MONTHLY\_DISCOUNT\*1.197)

FROM CUSTOMERS;

**Date Functions**

1. From *Customers* table, for all customers, display the first name, join date, join date minus 10 days, 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+1)

FROM CUSTOMERS;

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

SELECT FIRST\_NAME, BIRTH\_DATE, TRUNC((SYSDATE-BIRTH\_DATE+1)/365.25) AS AGE

FROM CUSTOMERS

WHERE TRUNC((SYSDATE-BIRTH\_DATE+1)/365.25) >50;

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

SELECT \*

FROM CUSTOMERS

WHERE TO\_CHAR(BIRTH\_DATE,'DD-MON') = TO\_CHAR(SYSDATE,'DD-MON');

1. 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 10 years since they joined the company.

SELECT FIRST\_NAME, JOIN\_DATE, TRUNC((SYSDATE-JOIN\_DATE+1)/365.25)

FROM CUSTOMERS

WHERE TRUNC((SYSDATE-JOIN\_DATE+1)/365.25) = 10;

**Conversion functions**

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

SELECT CONCAT(FIRST\_NAME,JOIN\_DATE), CONCAT( LAST\_NAME,MONTHLY\_DISCOUNT)

FROM CUSTOMERS;

1. From *Customers* table, for all customers whose last name starts with a *d* or *k,*display:
   1. last name
   2. state in uppercase concatenated with customer number
   3. join date concatenated with birthdate  
      in the WHERE clause instead of using LIKE, try to define the filtering condition using SUBSTRING.

SELECT LAST\_NAME, CONCAT(UPPER(STATE),CUSTOMER\_ID), CONCAT(JOIN\_DATE,BIRTH\_DATE)

FROM CUSTOMERS

WHERE SUBSTR(LAST\_NAME,1,1) IN ('D','K');

**Null-Related Functions**

1. Phone numbers report:
   1. 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, NVL(SECONDARY\_PHONE\_NUM,'N/A')

FROM CUSTOMERS

WHERE PACK\_ID = 27;

* 1. 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') AS MAIN\_PHONE\_NUM,

NVL(SECONDARY\_PHONE\_NUM,'N/A') AS SECONDARY\_PHONE\_NUM

FROM CUSTOMERS

WHERE TO\_CHAR(BIRTH\_DATE,'YYYY') = '1972';

**CASE Function**

1. From *Customers* table, for all customers, display the first name, last name, monthly discount and a discount grade based on these conditions :
   1. If the discount is between 0 and 10 – discount grade level is A.
   2. If the discount is between 11 and 20 – discount grade level is B.
   3. If the discount is between 21 and 30 – discount grade level is C.
   4. 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) GRADE

FROM CUSTOMERS;

**SQL Multi Row Function**

**Aggregate Functions**

Part 1 – Basic Usage

1. Display the lowest last name alphabetically (*Customers* table).

SELECT MIN(LAST\_NAME)

FROM CUSTOMERS;

1. Display the average monthly payment (*Packages* table).

SELECT AVG(MONTHLY\_PAYMENT)

FROM PACKAGES;

1. Display the highest last name alphabetically (*Customers* table).

SELECT MAX(LAST\_NAME)

FROM CUSTOMERS;

1. Display the number of internet packages (*Packages* table).

SELECT COUNT(\*)

FROM PACKAGES;

1. Display the number of records in *Customers* table.

SELECT COUNT(\*)

FROM CUSTOMERS;

1. Display the number of distinct states (*Customers* table).

SELECT COUNT(DISTINCT STATE)

FROM CUSTOMERS;

1. Display the number of distinct internet speeds (*Packages* table).

SELECT COUNT(DISTINCT SPEED)

FROM PACKAGES;

1. Display the number of values (exclude Nulls) in Fax column (*Customers* table).

SELECT COUNT(FAX)

FROM CUSTOMERS

WHERE FAX IS NOT NULL;

1. Display the number of Null values in Fax column (*Customers* table).

SELECT COUNT(\*)

FROM CUSTOMERS

WHERE FAX IS NULL;

1. Display the highest, lowest and average monthly discount (*Customers* table).

SELECT MAX(MONTHLY\_DISCOUNT), MIN(MONTHLY\_DISCOUNT), AVG(MONTHLY\_DISCOUNT)

FROM CUSTOMERS;

Part 2 – GROUP BY and HAVING clauses

1. Display the state and the number of customers for each state (*Customers* table).

SELECT STATE, COUNT(STATE)

FROM CUSTOMERS

GROUP BY STATE;

1. Display the internet speed and the average monthly payment for each speed (*Packages* table).

SELECT SPEED, AVG(MONTHLY\_PAYMENT)

FROM PACKAGES

GROUP BY SPEED;

1. Display the state and the number of distinct cities for each state (*Customers* table).

SELECT STATE, COUNT(DISTINCT CITY)

FROM CUSTOMERS

GROUP BY STATE;

1. Display the sector number and the highest monthly payment for each sector (*Packages* table).

SELECT SECTOR\_ID, MAX(MONTHLY\_PAYMENT)

FROM PACKAGES

GROUP BY SECTOR\_ID;

1. Package number and average monthly discount (*Customers* table) –
   1. Display the package number and the average monthly discount for each package.

SELECT PACK\_ID, AVG(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

GROUP BY PACK\_ID;

* 1. Display the package number and the average monthly discount for each package, only for packages whose number equals 22 or 13.

SELECT PACK\_ID, AVG(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

WHERE PACK\_ID IN (22,13)

GROUP BY PACK\_ID;

1. Display the highest, lowest and average monthly payment for each internet speed (*Packages* table).

SELECT MAX(MONTHLY\_PAYMENT), MIN(MONTHLY\_PAYMENT), AVG(MONTHLY\_PAYMENT)

FROM PACKAGES

GROUP BY SPEED;

1. The number of customer in each internet package (*Customers* table) –
   1. Display the package number and the number of customers for each package number.

SELECT PACK\_ID, COUNT(CUSTOMER\_ID)

FROM CUSTOMERS

GROUP BY PACK\_ID;

* 1. Modify the query to display the package number and number of customers for each package number, only for the customers whose monthly discount is greater than 20.

SELECT PACK\_ID, COUNT(CUSTOMER\_ID)

FROM CUSTOMERS

WHERE MONTHLY\_DISCOUNT > 20

GROUP BY PACK\_ID;

* 1. Modify the query to display the package number and number of customers for each package number, only for the packages with more than 100 customers.

SELECT PACK\_ID, COUNT(CUSTOMER\_ID)

FROM CUSTOMERS

GROUP BY PACK\_ID

HAVING COUNT(CUSTOMER\_ID)>100;

1. Display the state, city and number of customers for each state and city.

SELECT STATE, CITY, COUNT(CUSTOMER\_ID)

FROM CUSTOMERS

GROUP BY STATE,CITY;

1. Cities and the average monthly discount (*Customers* table) –
   1. Display the city and the average monthly discount for each city

SELECT STATE, AVG(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

GROUP BY STATE;

* 1. Display the city and the average monthly discount for each city, only for the customers whose monthly discount is greater than 20

SELECT STATE, AVG(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

WHERE MONTHLY\_DISCOUNT>20

GROUP BY STATE;

1. States and the lowest monthly discount (*Customers* table) –
   1. Display the state and the lowest monthly discount for each state.

SELECT STATE, MIN(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

GROUP BY STATE;

* 1. Display the state and lowest monthly discount for each state, only for states where the lowest monthly discount is greater than 10

SELECT STATE, MIN(MONTHLY\_DISCOUNT)

FROM CUSTOMERS

GROUP BY STATE

HAVING MIN(MONTHLY\_DISCOUNT)>10;

1. Display the internet speed and number of package for each internet speed, only for the internet speeds with more than 8 packages.

SELECT SPEED, COUNT(\*)

FROM PACKAGES

GROUP BY SPEED

HAVING COUNT(\*)>8;