ISLAMIC UNIVERSITY OF TECHNOLOGY

Organization of Islamic Cooperation

Board Bazar, Gazipur

Lab 04

CSE 4508

Md. Mohsinul Kabir

# Answer 1

## Creating Tables

CREATE TABLE customers (

id NUMBER(9, 0),  
 name VARCHAR2(20),  
 dob DATE,  
 CONSTRAINT pk\_customers PRIMARY KEY (id)  
);  
  
CREATE TABLE addresses (  
 customer\_id NUMBER(9, 0),  
 district VARCHAR2(20),  
 division VARCHAR2(20),  
 CONSTRAINT pk\_addresses PRIMARY KEY (customer\_id),  
 CONSTRAINT fk\_customer\_address FOREIGN KEY (customer\_id)  
 REFERENCES customers (id)  
);  
  
CREATE TABLE subscriptions (

customer\_id NUMBER(9, 0),  
 subscriber\_type VARCHAR2(10),  
 subscriber\_level VARCHAR2(10),  
 lifetime\_usage NUMBER(12, 2),  
 CONSTRAINT pk\_subscriptions PRIMARY KEY (customer\_id),  
 CONSTRAINT fk\_customer\_subscription FOREIGN KEY (customer\_id)  
 REFERENCES customers (id),  
 CONSTRAINT chk\_subscriber\_type CHECK (subscriber\_type = 'Postpaid'  
 OR subscriber\_type = 'Prepaid'),  
 CONSTRAINT chk\_subscriber\_level CHECK (subscriber\_level = 'Bronze'  
 OR subscriber\_level = 'Silver' OR subscriber\_level = 'Platinum')  
);

SQL

## Inserting Data

INSERT INTO customers VALUES (1, 'A', TO\_DATE('1978/03/16', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (2, 'B', TO\_DATE('1985/07/23', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (3, 'C', TO\_DATE('1972/02/05', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (4, 'D', TO\_DATE('1977/03/12', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (5, 'E', TO\_DATE('1983/05/18', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (6, 'F', TO\_DATE('1993/01/25', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (7, 'G', TO\_DATE('1990/06/03', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (8, 'H', TO\_DATE('1991/03/21', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (9, 'I', TO\_DATE('1995/02/17', 'YYYY/MM/DD'));  
INSERT INTO customers VALUES (10, 'J', TO\_DATE('1997/08/12', 'YYYY/MM/DD'));  
  
INSERT INTO addresses VALUES (1, 'Dhaka', 'Dhaka');  
INSERT INTO addresses VALUES (2, 'Bandarban', 'Chittagong');  
INSERT INTO addresses VALUES (3, 'Dhaka', 'Dhaka');  
INSERT INTO addresses VALUES (4, 'Dhaka', 'Dhaka');  
INSERT INTO addresses VALUES (5, 'Gazipur', 'Dhaka');  
INSERT INTO addresses VALUES (6, 'Jessore', 'Khulna');  
INSERT INTO addresses VALUES (7, 'Bogura', 'Rajshahi');  
INSERT INTO addresses VALUES (8, 'Dhaka', 'Dhaka');  
INSERT INTO addresses VALUES (9, 'Chattogram', 'Chittagong');  
INSERT INTO addresses VALUES (10, 'Narsingdi', 'Dhaka');  
  
INSERT INTO subscriptions VALUES (1, 'Postpaid', 'Bronze', 478.25);  
INSERT INTO subscriptions VALUES (2, 'Postpaid', 'Silver', 2107.50);  
INSERT INTO subscriptions VALUES (3, 'Prepaid', 'Platinum', 5708.31);  
INSERT INTO subscriptions VALUES (4, 'Postpaid', 'Bronze', 320.20);  
INSERT INTO subscriptions VALUES (5, 'Prepaid', 'Bronze', 378.17);  
INSERT INTO subscriptions VALUES (6, 'Prepaid', 'Silver', 2416.48);  
INSERT INTO subscriptions VALUES (7, 'Postpaid', 'Silver', 2312.71);  
INSERT INTO subscriptions VALUES (8, 'Prepaid', 'Bronze', 217.05);  
INSERT INTO subscriptions VALUES (9, 'Postpaid', 'Bronze', 370.86);  
INSERT INTO subscriptions VALUES (10, 'Postpaid', 'Bronze', 451.10);

SQL

## Performing Queries

SELECT \*  
FROM (

SELECT customers.id, name, lifetime\_usage  
 FROM customers INNER JOIN subscriptions  
 ON customers.id = subscriptions.customer\_id  
 ORDER BY lifetime\_usage DESC  
)  
WHERE ROWNUM <= 5;

SQL

### Output

ID NAME LIFETIME\_USAGE  
---------- -------------------- --------------  
 3 C 5708.31  
 6 F 2416.48  
 7 G 2312.71  
 2 B 2107.5  
 1 A 478.25

TEXT

SELECT name, dob, district, division  
FROM (  
 (customers INNER JOIN addresses ON customers.id = addresses.customer\_id)  
 INNER JOIN subscriptions ON customers.id = subscriptions.customer\_id)  
WHERE lifetime\_usage > (  
 SELECT AVG(lifetime\_usage)  
 FROM subscriptions  
 WHERE subscriber\_type = 'Prepaid' AND subscriber\_level = 'Silver'  
);

SQL

### Output

NAME DOB DISTRICT DIVISION  
-------------------- --------- -------------------- --------------------  
C 05-FEB-72 Dhaka Dhaka

TEXT

SELECT COUNT(\*)  
FROM (

SELECT name, dob, district, division  
 FROM (  
 (customers INNER JOIN addresses  
 ON customers.id = addresses.customer\_id)  
 INNER JOIN subscriptions

ON customers.id = subscriptions.customer\_id  
 )  
 WHERE lifetime\_usage > (  
 SELECT AVG(lifetime\_usage)  
 FROM subscriptions  
 WHERE subscriber\_type = 'Prepaid' AND subscriber\_level = 'Silver'))  
WHERE division = 'Dhaka';

SQL

### Output

COUNT(\*)  
----------  
 1

TEXT

# Answer 2

## CONCAT

CONCAT allows us to join two strings. We can use either the function CONCAT (string1, string2) or the || operator.

SELECT ('Hello ' || 'World') output FROM DUAL;

SQL

OUTPUT  
-----------  
Hello World

TEXT

## INITCAP

INITCAP capitalizes every word in a string, i.e. it converts the first letter in each word to uppercase and the other letters to lowercase.

SELECT INITCAP('hELLO wORLD') output FROM DUAL;

SQL

OUTPUT  
-----------  
Hello World

TEXT

## INSTR

INSTR tells us the position of a substring within a string. The position returned is the starting position of the first occurrence of the substring. The format of the function is INSTR(mainString, substring).

SELECT INSTR('Hello World', 'World') output FROM DUAL;

SQL

OUTPUT  
----------  
 7

TEXT

We can give this function a third argument, the point from which it should start searching. If we do this, the function will return the position of the first occurrence of the substring it finds from that point forward.

SELECT INSTR('Hello World', 'l', 5) output FROM DUAL;

SQL

OUTPUT  
----------  
 10

TEXT

If the point at which we tell the function to start searching is a negative number, it will search backwards, i.e. if we give the position as -2, it will go to the second character from the end of the string and find the first occurrence of the substring going left from that point.

SELECT INSTR('Hello World', 'l', -1) output FROM DUAL;

SQL

OUTPUT  
----------  
 10

TEXT

We can give the function a fourth argument as well, which is the occurrence number it should search for. For example, we can tell the function to find the 3rd occurrence of the substring.

SELECT INSTR('Hello World', 'l', 1, 3) output FROM DUAL;

SQL

OUTPUT  
----------  
 10

TEXT

## LOWER

The LOWER function converts every character in a string to lowercase.

SELECT LOWER('Hello World') output FROM DUAL;

SQL

OUTPUT  
-----------  
hello world

TEXT

## UPPER

The UPPER function converts every character in a string to uppercase.

SELECT UPPER('hello world') output FROM DUAL;

SQL

OUTPUT  
-----------  
HELLO WORLD

TEXT

## LENGTH

The LENGTH function returns the total number of characters in a string.

SELECT LENGTH('Hello World') output FROM DUAL;

SQL

OUTPUT  
----------  
 11

TEXT

## LPAD

The LPAD function adds spaces to the left of a string until it becomes a specified length.

SELECT LPAD('Hello World', 15) output FROM DUAL;

SQL

OUTPUT  
---------------  
 Hello World

TEXT

We can add a third argument to the function to specify the character we wish to use to pad the string.

SELECT LPAD('Hello World', 15, '#') output FROM DUAL;

SQL

OUTPUT  
---------------  
####Hello World

TEXT

If the provided string is larger than the specified length, the string is truncated after the specified number of characters.

SELECT LPAD('Hello World', 6) output FROM DUAL;

SQL

OUTPUT  
------  
Hello

TEXT

## RPAD

RPAD works exactly like LPAD, except it adds the padding to the right of the string instead of the left.

SELECT RPAD('Hello World', 15, '#') output FROM DUAL;

SQL

OUTPUT  
---------------  
Hello World####

TEXT

## LTRIM

The LTRIM function removes any leading spaces from a string.

SELECT LTRIM(' Hello World') output FROM DUAL;

SQL

OUTPUT  
-----------  
Hello World

TEXT

We can also give the function a second argument to specify which character it should remove from the front of the string. If this is done, it will only remove that character, not leading spaces.

SELECT LTRIM('H ello World', 'H') output FROM DUAL;

SQL

OUTPUT  
-----------  
 ello World

TEXT

## RTRIM

The RTRIM function removes any trailing spaces from a string.

SELECT RTRIM('Hello World ') output FROM DUAL;

SQL

OUTPUT  
-----------  
Hello World

TEXT

We can also give the function a second argument to specify which character it should remove from the end of the string. If this is done, it will only remove that character, not trailing spaces.

SELECT RTRIM('Hello Worl d', 'd') output FROM DUAL;

SQL

OUTPUT  
-----------  
Hello Worl

TEXT

## SUBSTR

Provided a string and a starting position, the SUBSTR function extracts a substring from the provided starting position to the end of the string.

SELECT SUBSTR('Hello World', 6) output FROM DUAL;

SQL

OUTPUT  
------  
 World

TEXT

The start position can also be a negative number. Say the given start position is -8. In this case, the function will go to the 8th character from the right and then extract a substring starting from that character to the end of the string.

SELECT SUBSTR('Hello World', -8) output FROM DUAL;

SQL

OUTPUT  
--------  
lo World

TEXT

The function can take a third argument, the length of the substring. If this is provided, the function will extract a substring starting from the provided starting position and ending at whatever position is required to make the substring the specified length.

SELECT SUBSTR('Hello World', 2, 6) output FROM DUAL;

SQL

OUTPUT  
------  
ello W

TEXT

## COUNT

The COUNT function counts the number of occurrences of a provided argument in a specified table.

SELECT COUNT(name) FROM customers;

SQL

### Output

COUNT(NAME)  
-----------  
 10

TEXT