

Islamic University of Technology

CSE 4508

Lab Report 3

Name : Amina

Student ID : 200041155

Section : 1

Lab Group : 1B (shifted from 1A)

Date of Performance : 30/08/23

Date of Submission : 31/08/23

Task A

At first we will create 3 tables in our database and insert some data within the tables.

```
create table Products(  
    product_id int primary key,  
    name varchar2(40),  
    unit_price int  
);  
  
create table Orders(  
    order_id int primary key,  
    customer_id int,  
    order_date varchar2(22)  
);  
  
create table Order_items(  
    order_id int,  
    product_id int,  
    quantity int,  
    CONSTRAINT fk_order_items_order_id FOREIGN KEY (order_id) REFERENCES  
Orders(order_id),  
    CONSTRAINT fk_order_items_product_id FOREIGN KEY (product_id) REFERENCES  
Products(product_id)  
);
```

```
insert into Products values(20 , 'Doll', 20);
insert into Products values(21 , 'Socks', 30);
insert into Products values(22 , '4- dimensional pocket', 3000);
insert into Products values(23 , 'Anywhere door', 2000);
insert into Products values(24 , 'Small light', 200);
insert into Products values(25 , 'Big light', 300);
insert into Products values(26 , 'Bamboo copter', 50);
insert into Products values(27 , 'Time machine', 5000);
insert into Products values(28 , 'Air canon', 200);
insert into Products values(29 , 'Animal beam', 220);
```

```
insert into Orders values(500 , 1000, 'July 20, 2023');
insert into Orders values(501 , 1001, 'June 13, 2023');
insert into Orders values(502 , 1002, 'June 24, 2023');
insert into Orders values(503 , 1003, 'July 26, 2023');
insert into Orders values(504 , 1004, 'June 17, 2023');
insert into Orders values(505 , 1005, 'July 19, 2023');
insert into Orders values(506 , 1006, 'June 20, 2023');
insert into Orders values(507 , 1007, 'July 30, 2023');
insert into Orders values(508 , 1008, 'July 23, 2023');
insert into Orders values(509 , 1009, 'June 12, 2023');
```

```
insert into Order_items values(509 , 23, 23);
insert into Order_items values(508 , 23, 2);
insert into Order_items values(507 , 24, 4);
insert into Order_items values(505 , 22, 5);
insert into Order_items values(501 , 22, 6);
insert into Order_items values(500 , 26, 7);
insert into Order_items values(508 , 26, 2);
insert into Order_items values(507 , 27, 3);
insert into Order_items values(507 , 28, 5);
insert into Order_items values(503 , 21, 6);
insert into Order_items values(502 , 20, 3);
```

Query for A1 :

```
select subtable.pid, sum(nvl(subtable.np, 0)) as total_revenue
from (select Products.product_id as pid, Order_items.order_id as oid,
(nvl(Products.unit_price, 0) * nvl(Order_items.quantity, 0)) as np
      from Products, Order_items
      where Products.product_id = Order_items.product_id) subtable, Orders
where Orders.order_id = subtable.oid AND
      Orders.order_date like '%June%'
group by subtable.pid;
```

Result :

PID	TOTAL_REVENUE
22	18000
20	60
23	46000

Query for A2 :

```
select *
from(select rownum as serial, totals.product_id as product_id,
totals.total_quantity as total_quantity
      from(select oi.product_id as product_id, sum(nvl(oi.quantity, 0)) as
total_quantity
            from order_items oi,(select *
                                from orders o
                                where o.order_date like '%July%') fromJuly
            where oi.order_id = fromJuly.order_id
            group by oi.product_id
            order by total_quantity desc) totals) finally
where finally.serial <=5;
```

Result :

SERIAL	PRODUCT_ID	TOTAL_QUANTITY
1	26	9
2	21	6
3	22	5
4	28	5
5	24	4

Query for A3 :

```
--A3

select customer_id from(
    select customer_id, sum(quantity * unit_price) as total_money
    from Products, Orders, Order_items
    where Products.product_id = Order_items.product_id and
           Orders.order_id = Order_items.order_id and
           order_date like '%July%'
    group by Orders.customer_id
)
where total_money >= 1000;
```

Result :

CUSTOMER_ID
1007
1005
1008

Query for A4 :

```
--A4
update Products set unit_price = 1.5 * unit_price
where product_id in
(
    select Products.product_id
    from Products, Orders, Order_items
    where Products.product_id = Order_items.product_id and
        Orders.order_id = Order_items.order_id and
        order_date like '%July%'
    group by Products.product_id
    having sum(quantity) >= 5
);
```

Result :

4 rows updated.

SQL> select * from Products;

PRODUCT_ID	NAME	UNIT_PRICE
20	Doll	20
21	Socks	45
22	4- dimensional pocket	4500
23	Anywhere door	2000
24	Small light	200
25	Big light	300
26	Bamboo copter	75
27	Time machine	5000
28	Air canon	300
29	Animal beam	220

10 rows selected.

Previous Table :

PRODUCT_ID	NAME	UNIT_PRICE
20	Doll	20
21	Socks	30
22	4- dimensional pocket	3000
23	Anywhere door	2000
24	Small light	200
25	Big light	300
26	Bamboo copter	50
27	Time machine	5000
28	Air canon	200
29	Animal beam	220

10 rows selected.

Task B

In Task B, we were asked to demonstrate the use of some built-in functions. The examples are given below.

CONCAT:

```
SQL> SELECT CONCAT('Mimi', 'Mina') AS concatenated_string from DUAL;

CONCATEN
-----
MimiMina
```

INSTR:

```
SQL> SELECT INSTR('Doraemon', 'mon') AS position FROM dual;

  POSITION
-----
        6
```

LOWER:

```
SQL> SELECT LOWER('NOBITA') AS lower_text FROM dual;

LOWER_
-----
nobita
```

UPPER:

```
SQL> SELECT UPPER('riruru') AS upper_text FROM dual;

UPPER_
-----
RIRURU
```

LENGTH:

```
SQL> SELECT LENGTH('Nobita sleeps') AS text_length FROM dual;

TEXT_LENGTH
-----
          13
```

L/R PAD:

```
SQL> SELECT LPAD('Karayel', 15, '#') AS left_padded_text FROM dual;

LEFT_PADDED_TEX
-----
#####Karayel
```

```
SQL> SELECT RPAD('Karayel', 15, '#') AS right_padded_text FROM dual;

RIGHT_PADDED_TE
-----
Karayel#####
```

L/R TRIM:

```
SQL> SELECT LTRIM('  Karayel  ') AS left_trimmed_text FROM dual;
```

```
LEFT_TRIMM
-----
Karayel
```

```
SQL> SELECT RTRIM('  Karayel  ') AS right_trimmed_text FROM dual;
```

```
RIGHT_TRIM
-----
  Karayel
```

SUBSTR:

```
SQL> SELECT SUBSTR('Computer Science', 10) AS extracted_substring FROM dual;
```

```
EXTRACT
-----
Science
```

COUNT:

```
SQL>
SQL> SELECT COUNT(DISTINCT customer_id) AS Unique_Customers FROM Orders;
```

```
UNIQUE_CUSTOMERS
-----
                10
```