Subquery Exercise

** Task 1: Retrieve the names of all users who have placed orders.**

```
SELECT
    name
FROM
    User_info
WHERE
    ID IN (SELECT user_id FROM Orders);
```

Task 2: Find the names of all riders who have been verified and have delivered orders in the "Chittagong" city

```
SELECT
     name
FROM
     Rider_info
WHERE
     verified = true
AND city_id =
(SELECT city_id FROM city where city_name = "Chittagong" )
AND ID IN (SELECT rider_id FROM Orders);
```

Task 3: Find the total number of orders placed by each user who has placed more than 3 orders. Display the user's name and the total number of orders they have placed.

```
SELECT
    U.name,
    COUNT(O.order_id) AS TotalOrders
FROM
    Orders O
JOIN
    User_info U
ON
    O.user_id = U.ID
WHERE U.ID IN ( SELECT user_id FROM Orders
                    GROUP BY user_id
                    HAVING COUNT(order_id) > 3
)
GROUP BY U.name;
```

Task 4: Find the names of riders who have delivered orders for restaurants that serve "Italian" cuisine.

```
SELECT rd.name FROM Rider_info rd
WHERE rd.ID
IN
(SELECT rider_id FROM Orders WHERE restaurant_id
IN
(SELECT restaurant_id FROM
Restaurant_info WHERE cuisine_type_id
IN
( SELECT cuisine_type_id FROM Cuisine_Type
WHERE cuisine_name = "Italian"
))) ;
```

Task 5: Find the user who has placed the most expensive order. Display the user's name and the order amount.

```
SELECT name FROM User_info WHERE ID = (
SELECT user_id FROM Orders WHERE total_amount =
(SELECT MAX(total_amount) FROM Orders));
```

Task 6: Find the names of restaurants that have more than 4 menu items available.

```
SELECT R.name FROM Restaurant_info R
WHERE R.restaurant_id IN
(SELECT M.restaurant_id FROM MenuItems M
GROUP BY restaurant_id
HAVING COUNT(M.name) > 4);
```

Task 7: Find the names of users who have placed at least one order in each city where they have registered.

```
SELECT U.name FROM User_info U
WHERE U.ID IN(SELECT O.user_id FROM Orders O
GROUP BY O.user_id
HAVING COUNT(DISTINCT U.CITY_ID) =
(SELECT COUNT(city_id) FROM User_info WHERE ID = U.ID));
```

Task 8: Find the total number of orders placed by each user and the total amount they've spent on those orders. Display the user's name, the total number of orders, and the total amount spent. Sort the results in descending order based on the total amount spent.

```
SELECT
(SELECT name FROM User_info WHERE ID = USER_ID ),
TotalOrders,TotalAmountSpent
FROM
(SELECT O.user_id AS USER_ID,
COUNT(O.order_id) AS TotalOrders,
SUM(O.total_amount) AS TotalAmountSpent
FROM Orders O
GROUP BY O.user_id
) AS OrderInfoData
ORDER BY TotalAmountSpent DESC;
```

Task 9: Find the total number of orders placed by each user in the "Chittagong" city and the total amount they've spent on those orders. Display the user's name, the total number of orders, and the total amount spent. Sort the results in descending order based on the total amount spent.

```
SELECT (SELECT name
FROM User_info WHERE ID = USER_ID) AS USER_NAME,
TOTAL_ORDER ,TOTAL_SPENT
FROM (SELECT O.user_id AS
USER_ID,COUNT(O.order_id)
AS TOTAL_ORDER ,SUM(O.total_amount)
AS TOTAL_SPENT FROM Orders O
WHERE O.user_id IN(SELECT ID FROM User_info
WHERE CITY_ID =
(SELECT city_id FROM city WHERE city_name="Chittagong"))
GROUP BY O.user_id) AS INFO_DATA
ORDER BY TOTAL_SPENT DESC;
```

Task 10: Find the names of restaurants that have an average menu item price greater than $20. Display the restaurant names and the average menu item price.

```
SELECT
(SELECT name FROM Restaurant_info
WHERE restaurant_id = RID ) AS Restaurant_Name,APRICE FROM
(SELECT M.restaurant_id AS RID
,AVG(M.price) AS APRICE FROM MenuItems M
GROUP BY M.restaurant_id
HAVING APRICE > 20) AS TMP_TABLE;
```