Assignment: SQL Subqueries, CASE, String Functions, and COALESCE

Question 1. Write an SQL query to find the names of restaurants that have at least one menu item with a price greater than $10. **Answer 1:**

```
SELECT DISTINCT
    R.name
FROM
    Restaurant_info R
WHERE
    restaurant_id IN
    (
        SELECT
            M.restaurant_id AS RestaurantID
            FROM
                MenuItems M
            WHERE
                M.price >10
    );
```

Question 2. Write an SQL query to retrieve the user names and their corresponding orders where the order total is greater than the average order total for all users.

Answer 2 :

```
SELECT
   U.name AS UserName,
   O.order_id AS Order_ID,
   O.total_amount AS TOTAL_AMOUNT
FROM
   User_info U
JOIN
   Orders O
ON
   U.ID = O.user_id

WHERE O.total_amount >
   (
       SELECT
            AVG(OIN.total_amount)
       FROM
           Orders OIN
   );
```

Question 3. Write an SQL query to list the names of users whose last names start with 'S' or ends with ‘e’. **Answer 3 :**

```
SELECT
    name
FROM
    User_info
WHERE
        SUBSTRING_INDEX(name," ",-1)  LIKE "S%"
    OR
        SUBSTRING_INDEX(name," ",-1) LIKE "%e"
;
```

Question 4. Write an SQL query to find the total order amounts for each restaurant. If a restaurant has no orders, display the restaurant name and a total amount of 0. Use the COALESCE function to handle null values.

Answer 4:

```
SELECT
    R.name AS RestaurantName,
    COALESCE(SUM(O.total_amount),0) AS TOTAL_AMOUNT
FROM
    Restaurant_info R
LEFT JOIN
    Orders O
ON
    R.restaurant_id = O.restaurant_id
GROUP BY
    R.name;
```

Question 5. Write a query to find out how many orders were placed using cash or credit.

Answer 5 :

```
SELECT
    PT.name AS PaymentMethod,
    COUNT(O.order_id) AS ORDER_COUNT
FROM
    Orders O
JOIN
    Payment_Transactions PTR
ON
    O.order_id = PTR.order_id
JOIN
    Payment_type PT
ON
    PTR.pay_type_id = PT.pay_type_id
```

```
GROUP BY
    PT.name;
```