Q1. Retrieve the total number of orders placed by each user. Display the user's name and the total number of orders they have placed. Sort the results in descending order based on the number of orders.

Solution:

```
SELECT
    U.name AS USER_NAME, COUNT(user_id) AS TOTAL_ORDERS
FROM
    orders AS O
        LEFT JOIN
    user_info AS U ON O.user_id = U.ID
GROUP BY user_id
ORDER BY TOTAL_ORDERS DESC;
```

Q2. Find the average price of menu items for each restaurant. Display the restaurant name and the average menu item price. Sort the results in ascending order based on the restaurant name.

Solution:

```
SELECT
    R.name AS RESTAURANT_NAME, AVG(price) AS AVG_ITEM_PRICE
FROM
    menuitems M
        LEFT JOIN
    restaurant_info R ON M.restaurant_id = R.restaurant_id
GROUP BY M.restaurant_id
ORDER BY RESTAURANT_NAME ASC;
```

Q3. Identify the restaurant with the highest total sales (sum of order amounts). Display the restaurant name and the total sales amount.

Solution:

```
SELECT
    R.name AS RESTAURANT_NAME,
    SUM(total_amount) AS TOTAL_SALES_AMOUNT
FROM
    orders O
        LEFT JOIN
    restaurant_info R ON O.restaurant_id = R.restaurant_id
GROUP BY O.restaurant_id
ORDER BY TOTAL_SALES_AMOUNT DESC;
```

Q4. Find the number of orders placed in each city. Display the city name and the number of orders. Sort the results in descending order based on the number of orders.

Solution:

```
SELECT
    C.city_name AS CITY, COUNT(C.city_id) AS TOTAL_ORDERS
FROM
    orders O
        INNER JOIN
    restaurant_info R
        INNER JOIN
    city C ON O.restaurant_id = R.restaurant_id
        AND R.city_id = C.city_id
GROUP BY C.city_id
ORDER BY TOTAL_ORDERS DESC;
```