Question 1. Retrieve a list of user information with their name and date of registration who uses android phones. First look at the structure of the user_info table in database and then write your query.

Code:

```
SELECT Name,registration_date
    FROM user_info
WHERE operating_system='Android';
```

Output:

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

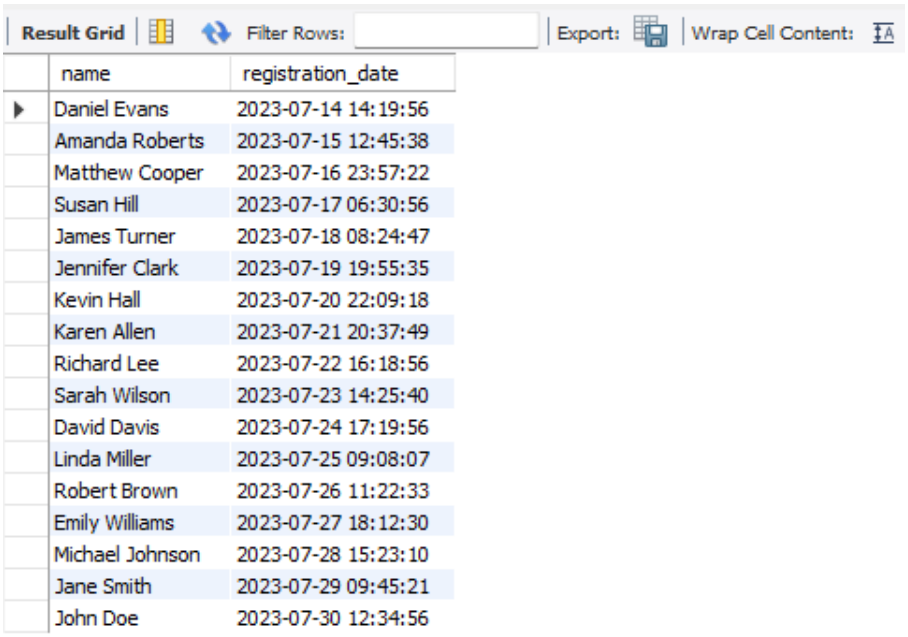
name	registration_date
John Doe	2023-07-30 12:34:56
Michael Johnson	2023-07-28 15:23:10
Robert Brown	2023-07-26 11:22:33
David Davis	2023-07-24 17:19:56
Richard Lee	2023-07-22 16:18:56
Kevin Hall	2023-07-20 22:09:18
James Turner	2023-07-18 08:24:47
Matthew Cooper	2023-07-16 23:57:22
Daniel Evans	2023-07-14 14:19:56
Thomas Adams	2023-07-12 18:56:10
Charles Morgan	2023-07-10 10:34:18
Andrew James	2023-07-08 21:10:59
Joseph Scott	2023-07-06 12:08:27
Michael Hall	2023-07-04 19:23:12
Jason Allen	2023-07-02 11:34:56
Christopher Lewis	2023-06-30 22:18:40
Mark Harris	2023-06-28 13:25:45
Matthew Allen	2023-06-26 11:14:10
William Turner	2023-06-24 09:50:40
John Smith	2023-06-22 16:18:55

Question 2. Edit your query above to find out users who have registered on or after 14th of july and sort the list of users in ascending order.

Code:

```sql
SELECT NAME, REGISTRATION_DATE
    FROM USER_INFO
WHERE REGISTRATION_DATE >= '2023-07-14'
    ORDER BY REGISTRATION_DATE ASC;
```

Output:



name	registration_date
Daniel Evans	2023-07-14 14:19:56
Amanda Roberts	2023-07-15 12:45:38
Matthew Cooper	2023-07-16 23:57:22
Susan Hill	2023-07-17 06:30:56
James Turner	2023-07-18 08:24:47
Jennifer Clark	2023-07-19 19:55:35
Kevin Hall	2023-07-20 22:09:18
Karen Allen	2023-07-21 20:37:49
Richard Lee	2023-07-22 16:18:56
Sarah Wilson	2023-07-23 14:25:40
David Davis	2023-07-24 17:19:56
Linda Miller	2023-07-25 09:08:07
Robert Brown	2023-07-26 11:22:33
Emily Williams	2023-07-27 18:12:30
Michael Johnson	2023-07-28 15:23:10
Jane Smith	2023-07-29 09:45:21
John Doe	2023-07-30 12:34:56

Question 3: Imagine you are tasked with retrieving a list of all restaurants and their menu items. Some restaurants may not have any menu items yet. Write an SQL query that performs a LEFT JOIN between the "Restaurant_info" and "MenuItems" tables to achieve this. Your query should include the restaurant name and the name of the menu items, if available.

Code:

```
SELECT R.NAME,M.NAME
    FROM RESTAURANT_INFO R
LEFT JOIN MENUITEMS M
    ON R.RESTAURANT_ID=M.RESTAURANT_ID;
```

Output:

NAME	NAME
Restaurant A	Item2
Restaurant A	Item8
Restaurant A	Item14
Restaurant A	Item22
Restaurant B	Item3
Restaurant B	Item35
Restaurant B	Item50
Restaurant C	Item4
Restaurant C	Item39
Restaurant C	Item43
Restaurant D	Item19
Restaurant D	Item30
Restaurant D	Item42
Restaurant E	Item18
Restaurant E	Item23
Restaurant E	Item37
Restaurant F	Item11
Restaurant F	Item15
Restaurant F	Item16
Restaurant F	Item46
Restaurant F	Item48

Question 4: Extend the previous query to include the restaurant's contact number and the availability status of each menu item. If a restaurant does not have any menu items, display "No Menu Items" in the menu item column.

Code:

```
SELECT R.NAME, CONTACT_NUMBER,
    COALESCE(M.NAME,'NO MENU ITEMS') MENU_ITEM_NAME, AVAILABILITY
    FROM RESTAURANT_INFO R
LEFT OUTER JOIN MENUITEMS M
    ON R.RESTAURANT_ID = M.RESTAURANT_ID
```

Output:

NAME	CONTACT_NUMBER	MENU_ITEM_NAME	AVAILABILITY
Restaurant A	9876543210	Item2	1
Restaurant A	9876543210	Item8	0
Restaurant A	9876543210	Item14	1
Restaurant A	9876543210	Item22	1
Restaurant B	1234567890	Item3	1
Restaurant B	1234567890	Item35	1
Restaurant B	1234567890	Item50	0
Restaurant C	5555555555	Item4	1
Restaurant C	5555555555	Item39	1
Restaurant C	5555555555	Item43	0
Restaurant D	7777777777	Item19	0
Restaurant D	7777777777	Item30	1
Restaurant D	7777777777	Item42	1
Restaurant E	8888888888	Item18	0
Restaurant E	8888888888	Item23	0
Restaurant E	8888888888	Item37	1
Restaurant F	2222222222	Item11	1
Restaurant F	2222222222	Item15	1
Restaurant F	2222222222	Item16	1
Restaurant F	2222222222	Item46	1
Restaurant F	2222222222	Item48	0

Question 5: 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.

Code:

```
SELECT NAME,COUNT(ORDER_ID) NUMBER_OF_ORDER
    FROM USER_INFO A
INNER JOIN ORDERS B ON A.ID=B.USER_ID
    GROUP BY USER_ID
    ORDER BY NUMBER_OF_ORDER DESC;
```

Output:

NAME	NUMBER_OF_ORDER
Rebecca Bailey	5
Karen Allen	5
Elizabeth King	4
Susan Hill	4
Patricia Scott	4
Charles Morgan	4
Matthew Cooper	3
Amanda Roberts	3
Michelle Lee	3
Kimberly Green	3
Joseph Scott	3
Jane Smith	2
William Turner	2
Robert Brown	2
Linda Miller	2
David Davis	2
Emily Davis	2
Thomas Adams	2
Mark Harris	2
Michael Johnson	1
James Turner	1

Question 6: 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.

Code:

```
SELECT R.NAME,AVG(PRICE) AVERAGE_PRICE
    FROM MENUITEMS M
INNER JOIN RESTAURANT_INFO R
    ON M.RESTAURANT_ID=R.RESTAURANT_ID
GROUP BY R.RESTAURANT_ID
ORDER BY AVERAGE_PRICE ASC;
```

Output:

NAME	AVERAGE_PRICE
Restaurant C	27.6667
Restaurant J	29.5000
Restaurant A	31.0000
Restaurant D	34.3333
Restaurant L	36.3333
Restaurant H	44.5000
Restaurant B	47.0000
Restaurant E	58.0000
Restaurant I	60.0000
Restaurant K	60.0000
Restaurant F	61.1667
Restaurant G	63.3333

Question 7: Identify the restaurant with the highest total sales (sum of order amounts). Display the restaurant name and the total sales amount.

Code:

```
SELECT R.NAME,SUM(TOTAL_AMOUNT) TOTALS_SALES
    FROM ORDERS O
INNER JOIN RESTAURANT_INFO R
    ON O.RESTAURANT_ID=R.RESTAURANT_ID
GROUP BY R.RESTAURANT_ID
ORDER BY R.NAME DESC LIMIT 1;
```

Output:

	NAME	TOTALS_SALES
▶	Restaurant L	1809

Question 8: 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.

Code:

```
SELECT CITY_NAME,COUNT(ORDER_ID) ORDER_NUMBER
    FROM ORDERS O
LEFT JOIN
    (SELECT NAME,R.CITY_ID,C.CITY_NAME,RESTAURANT_ID FROM RESTAURANT_INFO R
    LEFT JOIN CITY C
    ON R.CITY_ID = C.CITY_ID) M
    ON O.RESTAURANT_ID = M.RESTAURANT_ID
GROUP BY CITY_NAME ;
```

Output:

	CITY_NAME	ORDER_NUMBER
▶	Rangpur	12
	Khulna	14
	Chittagong	5
	Barisal	17
	Sylhet	5
	Comilla	9
	Jessore	3
	Dhaka	6

Question 9: Write an SQL query to find the names of restaurants that have at least one menu item with a price greater than $10.

Code:

```
SELECT DISTINCT NAME
    FROM RESTAURANT_INFO
WHERE RESTAURANT_ID IN
    (
    SELECT RESTAURANT_ID FROM MENUITEMS
    );
```

Output:

	NAME
▶	Restaurant A
	Restaurant B
	Restaurant C
	Restaurant D
	Restaurant E
	Restaurant F
	Restaurant G
	Restaurant H
	Restaurant I
	Restaurant J
	Restaurant K
	Restaurant L

Question 10: 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.

Code:

```
SELECT NAME, ORDER_NUMBER
    FROM USER_INFO U
INNER JOIN
    (SELECT USER_ID, COUNT(ORDER_ID) ORDER_NUMBER FROM ORDERS
        WHERE
        TOTAL_AMOUNT> (
            SELECT AVG(TOTAL_AMOUNT)
                FROM ORDERS)
            GROUP BY USER_ID
         ) T
          ON U.ID=T.USER_ID;
```

Output:

NAME	ORDER_NUMBER
Jane Smith	2
Linda Miller	1
David Davis	1
Sarah Wilson	1
Karen Allen	2
Susan Hill	2
Amanda Roberts	1
Elizabeth King	1
Charles Morgan	2
Rebecca Bailey	1
Michael Hall	1
Kimberly Green	2
Mark Harris	2
Michelle Lee	2
William Turner	1
Patricia Scott	2

Question 11: Write an SQL query to list the names of users whose last names start with 'S' or ends with ‘e’.

Code:

```
SELECT NAME FROM USER_INFO
WHERE
    SUBSTRING_INDEX(NAME, ' ', -1 ) LIKE 'S%'
    OR
    SUBSTRING_INDEX(NAME, ' ', -1 ) LIKE '%E';
```

Output:

NAME
John Doe
Jane Smith
Richard Lee
Joseph Scott
Stephanie White
Michelle Lee
Patricia Scott
John Smith

Question 12: 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.

Code:

```
SELECT R.NAME, TOTAL_AMOUNT
    FROM RESTAURANT_INFO R
INNER JOIN
    (SELECT RESTAURANT_ID,COALESCE(SUM(TOTAL_AMOUNT),0 ) TOTAL_AMOUNT
        FROM ORDERS
    GROUP BY RESTAURANT_ID ) T
    ON R.RESTAURANT_ID=T.RESTAURANT_ID;
```

Output:

NAME	TOTAL_AMOUNT
Restaurant A	586
Restaurant B	892
Restaurant C	1999
Restaurant D	1915
Restaurant E	1683
Restaurant F	643
Restaurant G	2542
Restaurant H	578
Restaurant I	844
Restaurant J	1752
Restaurant K	521
Restaurant L	1809

Question 13: Write a query to find out how many orders were placed using cash or credit.

Code:

```
SELECT NAME, TOTAL_AMOUNT
    FROM PAYMENT_TYPE P
INNER JOIN
    (SELECT PAY_TYPE_ID, COUNT(PAY_TYPE_ID) TOTAL_AMOUNT
        FROM PAYMENT_TRANSACTIONS
    GROUP BY PAY_TYPE_ID) T
    ON P.PAY_TYPE_ID=T.PAY_TYPE_ID;
```

Output:

	NAME	TOTAL_AMOUNT
▶	Cash	33
	Credit	38