**SQL – Assignment**

1. Below is a sample table. Please write a SQL query based on the below table to return count of people whose grandparent is alive.

|  |  |  |
| --- | --- | --- |
| **Name of Person** | **Name of Parent** | **Status of Person (Alive/Dead)** |
| A | X | Alive |
| B | Y | Dead |
| X | X' | Alive |
| Y | Y' | Alive |
| X' | X'' | Alive |
| Y' | Y'' | Dead |

1. Below table (Table 1) has city and month wise sales data. Create a SQL query to return output as illustrated in Table 2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1** | | | |
| **City** | **Year** | **Month** | **Sales** |
| Delhi | 2020 | 5 | 4300 |
| Delhi | 2020 | 6 | 2000 |
| Delhi | 2020 | 7 | 2100 |
| Delhi | 2020 | 8 | 2200 |
| Delhi | 2020 | 9 | 1900 |
| Delhi | 2020 | 10 | 200 |
| Mumbai | 2020 | 5 | 4400 |
| Mumbai | 2020 | 6 | 2800 |
| Mumbai | 2020 | 7 | 6000 |
| Mumbai | 2020 | 8 | 9300 |
| Mumbai | 2020 | 9 | 4200 |
| Mumbai | 2020 | 10 | 9700 |
| Bangalore | 2020 | 5 | 1000 |
| Bangalore | 2020 | 6 | 2300 |
| Bangalore | 2020 | 7 | 6800 |
| Bangalore | 2020 | 8 | 7000 |
| Bangalore | 2020 | 9 | 2300 |
| Bangalore | 2020 | 10 | 8400 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Table 2** | | | | | | |
| **City** | **Year** | **Month** | **Sales** | **Previous Month Sales** | **Next Month Sales** | **YTD Sales** |
| Delhi | 2020 | 5 | 4300 |  | 2000 | 4300 |
| Delhi | 2020 | 6 | 2000 | 4300 | 2100 | 6300 |
| Delhi | 2020 | 7 | 2100 | 2000 | 2200 | 8400 |
| Delhi | 2020 | 8 | 2200 | 2100 | 1900 | 10600 |
| Delhi | 2020 | 9 | 1900 | 2200 | 200 | 12500 |
| Delhi | 2020 | 10 | 200 | 1900 |  | 12700 |
| Mumbai | 2020 | 5 | 4400 |  |  |  |
| Mumbai | 2020 | 6 | 2800 |  |  |  |
| Mumbai | 2020 | 7 | 6000 |  |  |  |
| Mumbai | 2020 | 8 | 9300 |  |  |  |
| Mumbai | 2020 | 9 | 4200 |  |  |  |
| Mumbai | 2020 | 10 | 9700 |  |  |  |
| Bangalore | 2020 | 5 | 1000 |  |  |  |
| Bangalore | 2020 | 6 | 2300 |  |  |  |
| Bangalore | 2020 | 7 | 6800 |  |  |  |
| Bangalore | 2020 | 8 | 7000 |  |  |  |
| Bangalore | 2020 | 9 | 2300 |  |  |  |
| Bangalore | 2020 | 10 | 8400 |  |  |  |

3. Find the monthly retention rate of users for each account separately for Dec 2020 and Jan 2021. Retention rate is the percentage of active users an account retains over a given period of time. In this case, assume the user is retained if he/she stays with the app in any future months. For example, if a user was active in Dec 2020 and has activity in any future month, consider them retained for Dec. You can assume all accounts are present in Dec 2020 and Jan 2021. Your output should have the account ID and the Jan 2021 retention rate divided by Dec 2020 retention rate.   
  
Table : **sf\_events**

|  |  |
| --- | --- |
| **Column name** | **Data Type** |
| Date | datetime |
| Account\_id | varchar |
| User\_id | varchar |

4 . Write a query to return Territory and corresponding Sales Growth. Compare growth between periods Q4-2021 vs Q3-2021. If Territory (say T123) has Sales worth 100 in Q3-2021 and Sales worth 100*inQ*3−2021*andSalesworth*110 in Q4-2021, then the Sales Growth will be 10% [ i.e. = ((110 - 100)/100) \* 100 ] Output the ID of the Territory and the Sales Growth. Only output these territories that had any sales in both quarters.  
  
Tables : fct\_customer\_sales , map\_customer\_territory  
  
**fct\_customer\_sales**

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| Cust\_id | Varchar |
| Prod\_sku\_id | Varchar |
| Order\_date | Datetime |
| Order\_value | Int |
| Order\_id | Varchar |

**Map\_customer\_territory**

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| Cust\_id | Varchar |
| Territory\_id | Varchar |

5. Definition of *Frequent* Customer: A Customer who has transacts on the platform atleast once in every 5 days since last transaction

**Table:**

**“**

CREATE TABLE SALES (

ORDER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

CUSTOMER\_ID INT,

SALARY NUMERIC(10,2),

ORDER\_DATE DATETIME

);

INSERT INTO SALES

(ORDER\_ID, CUSTOMER\_ID, SALARY, ORDER\_DATE) VALUES

(10001,90001, 10000, '2022-02-01 09.00.00'),

(10002,90001, 10000, '2022-02-03 09.00.00'),

(10003,90001, 10000, '2022-02-07 09.00.00'),

(10004,90001, 20000, '2022-02-09 09.00.00'),

(10005,90001, 20000, '2022-02-14 09.00.00'),

(10006,90001, 10000, '2022-02-14 09.00.00'),

(10007,90001, 10000, '2022-02-17 09.00.00'),

(10009,90001, 80000, '2022-02-21 09.00.00'),

(100020,90001, 10000, '2022-02-23 09.00.00'),

(100021,90001, 10000, '2022-02-28 09.00.00'),

(10010,90002, 10000, '2022-02-01 09.00.00'),

(10013,90002, 30000, '2022-02-09 09.00.00'),

(10014,90002, 10000, '2022-02-14 09.00.00'),

(10015,90002, 10000, '2022-02-14 09.00.00'),

(10016,90002, 70000, '2022-02-17 09.00.00'),

(10017,90002, 10000, '2022-02-21 09.00.00'),

(10019,90002, 10000, '2022-02-28 09.00.00');

select \* FROM SALES

“

Write a SQL query for below questions:

1. Find which customers are *Frequent.*
2. Evaluate cumulative sum of ORDER\_VALUE for each customer in ascending order of ORDER\_DATE

Output Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORDER\_ID** | **CUSTOMER\_ID** | **ORDER\_VALUE** | **ORDER\_DATE** | **CUM\_SUM** |
| 10001 | 90001 | 10000 | 2022-02-01 | 10000 |
| 10002 | 90001 | 10000 | 2022-02-03 | 20000 |
| 10003 | 90001 | 10000 | 2022-02-07 | 30000 |
| 10004 | 90001 | 20000 | 2022-02-09 | 50000 |
| 10005 | 90001 | 20000 | 2022-02-14 | 70000 |
| 10006 | 90001 | 10000 | 2022-02-14 | 80000 |
| 10007 | 90001 | 10000 | 2022-02-17 | 90000 |
| 10009 | 90001 | 80000 | 2022-02-21 | 170000 |
| 10010 | 90002 | 10000 | 2022-02-01 | 10000 |
| 10013 | 90002 | 30000 | 2022-02-09 | 40000 |
| 10014 | 90002 | 10000 | 2022-02-14 | 50000 |
| 10015 | 90002 | 10000 | 2022-02-14 | 60000 |
| 10016 | 90002 | 70000 | 2022-02-17 | 130000 |
| 10017 | 90002 | 10000 | 2022-02-21 | 140000 |
| 10019 | 90002 | 10000 | 2022-02-28 | 150000 |
| 100020 | 90001 | 10000 | 2022-02-23 | 160000 |
| 100021 | 90001 | 10000 | 2022-02-28 | 170000 |

1. Order IDs which constitute Top 80 percentile basis Order\_Value
2. Create a coupon\_flag which becomes active on alternate transactions, signifying availability of coupon. Assume coupon\_flag is 1 (Active) on first transaction, find number of days an offer was valid for each customer.