SAHIL VERMA - DS03

RE-MCT - Banking Data Analysis in SQL

1: Write a query to list all customers who haven't made any transactions in the last year. How can we make them active again? Provide appropriate region.

```sql
## SAHIL VERMA
select c.customer_id, c.first_name, c.state, t.transaction_date
from customerss c
left join accounts a on c.customer_id=a.customer_id
left join transactions t on a.account_number=t.account_number
and t.transaction_date < DATE_SUB(CURDATE(), INTERVAL 1 YEAR)
where t.transaction_id is null;
```

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

customer_id	first_name	state	transaction_date
4	Vanya	Karnataka	NULL
24	Ivan	Tamil Nadu	NULL
32	Madhav	Telangana	NULL
39	Baiju	Karnataka	NULL
40	Tanya	Karnataka	NULL
44	Keya	Maharashtra	NULL
49	Faiyaz	Uttar Pradesh	NULL

Q2 Summarize the total transaction amount per account per month.

```sql
## SAHIL VERMA
select account_number,
date_format(transaction_date, '%m') AS transaction_month,
round(sum(amount),2) AS total_transaction_amount
from transactions
group by account_number, transaction_month
order by account_number,transaction_month;
```

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

account_number	transaction_month	total_transaction_amount
1012982863	01	3510.6
1012982863	02	745.36
1012982863	04	3636.29
1012982863	05	16998.37
1012982863	07	162.98
1012982863	10	9445.09
1012982863	12	6419.55

Q3 Rank branches based on the total amount of deposits made in the last quarter.

```sql
## SAHIL VERMA
with last_quarter_deposit as (
select b.branch_id, t.transaction_type, t.transaction_date, round(sum(t.amount),2) as total_deposits
from branch b
join accounts a on b.branch_id=a.branch_id
join transactions t on a.account_number=t.account_number
group by b.branch_id, t.transaction_type, t.transaction_date
order by total_deposits desc
)
select *,
rank() over(order by total_deposits desc) as brank_ranking
from last_quarter_deposit
where transaction_type='Deposit' AND transaction_date <= DATE_SUB(CURDATE(), INTERVAL 3 MONTH);
```

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

branch_id	transaction_type	transaction_date	total_deposits	brank_ranking
6	Deposit	2023-08-23 11:11:28	4990.4	1
25	Deposit	2022-08-30 12:42:14	4987.77	2
2	Deposit	2022-09-26 10:28:02	4983.37	3
18	Deposit	2023-02-15 17:17:11	4973.02	4
20	Deposit	2023-12-14 15:09:35	4968.79	5
20	Deposit	2024-01-30 08:07:27	4963.87	6
19	Deposit	2023-12-19 10:47:35	4963.68	7

Q4 Find the name of the customer who has deposited the highest amount.

```sql
## SAHIL VERMA
select * from (select c.first_name, c.last_name, round(sum(t.amount),2) as deposit_amount
from customerss c
left join accounts a on c.customer_id=a.customer_id
left join transactions t on a.account_number=t.account_number
where t.transaction_type='Deposit'
group by c.first_name, c.last_name) as high_deposit
order by deposit_amount desc limit 1;
```

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

first_name	last_name	deposit_amount
Dishani	Deol	53763.75

Q5 Identify any accounts that have made more than two transactions in a single day, which could indicate fraudulent activity. How can you verify any fraudulent transaction?

```
54    ## SAHIL VERMA
55    select a.account_number, date_format(transaction_date, '%d') as day_of_month,
56    count(t.transaction_type) as no_of_transaction
57    from accounts a
58    join transactions t on a.account_number=t.account_number
59    group by a.account_number, day_of_month
60    having no_of_transaction>3;
```

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

account_number	day_of_month	no_of_transaction
1097521618	22	4
1032972218	18	4
1128643865	03	4

Q6 Calculate the average number of transactions per customer per account per month over the last year. average transaction per customer.

```
65    ## SAHIL VERMA
66    with MonthlyTransactions as (
67    SELECT c.customer_id, a.account_number,date_format(t.transaction_date, '%Y-%m') AS transaction_month,
68    count(t.transaction_id) AS transaction_count
69    from customerss c
70    left join accounts a ON c.customer_id = a.customer_id
71    left join transactions t ON a.account_number = t.account_number
72    where t.transaction_date >= date_add(CURDATE(), INTERVAL -1 YEAR)
73    group by c.customer_id, a.account_number, transaction_month
74    )
75    select customer_id, account_number, AVG(transaction_count) AS avg_transactions_per_month
76    from MonthlyTransactions
77    group by customer_id, account_number;
```

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

customer_id	account_number	avg_transactions_per_month
25	1036932307	1.7500
7	1044819961	2.0000
92	1116132425	1.0000
7	1084766312	1.3333
14	1170542404	1.0000
1	1032168449	1.0000
66	1053611792	1.0000

Q7 Write a query to find the daily transaction volume (total amount of all transactions) for the past month.

```
81  ## SAHIL VERMA
82  select date_format(transaction_date, '%Y-%m') as day_of_month, sum(amount) as total_amt_of_alltrans
83  from transactions
84  where transaction_date <= date_add(curdate(), interval -1 month)
85  group by day_of_month
86  order by day_of_month desc;
```

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

day_of_month	total_amt_of_alltrans
2024-07	81427.88
2024-06	255686.34999999992
2024-05	294164.31000000006
2024-04	212770.99999999994
2024-03	205383.22000000003
2024-02	247193.64000000007
2024-01	219896.71000000005

Q8 Calculate the total transaction amount performed by each age group in the past year. (Age groups: 0-17, 18-30, 31-60, 60+)

```
90  ## SAHIL VERMA
91  SELECT
92      CASE
93          WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 0 AND 17 THEN '0-17'
94          WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 18 AND 30 THEN '18-30'
95          WHEN TIMESTAMPDIFF(YEAR, c.date_of_birth, CURDATE()) BETWEEN 31 AND 68 THEN '31-60'
96          ELSE '60+'
97      END AS age_group,
98  round(SUM(T.amount),2) AS total_transaction_amount
99  FROM customerss c
100 JOIN accounts a ON c.customer_id = a.customer_id
101 JOIN transactions t ON a.account_number = t.account_number
102 WHERE t.transaction_date >= DATE_SUB(CURDATE(), interval 1 YEAR)
103 group by age_group;
```

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

age_group	total_transaction_amount
60+	319764.55
31-60	892399.15
18-30	199956.02

Q9 Find the branch with the highest average account balance.

```
107    ## SAHIL VERMA
108 •  select b.branch_name, round(avg(a.balance),2) as avg_balance
109    from branch b
110    join accounts a on b.branch_id=a.branch_id
111    group by b.branch_name
112    order by avg_balance desc;
```

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

branch_name	avg_balance
ICICI 56 Lake View	6997.23
BOB 101 Residency Road	6619.95
HDFC 303 Banjara Hills	6418.17
SBI 55 Info Park	6402.73
UBI 12 Green Avenue	6316.28
INDUS 123 MG Road	6088.83
BOB 78 Royal Street	5988.37

Q10 Calculate the average balance per customer at the end of each month in last year.

```
116    ## SAHIL VERMA
117 •  select c.customer_id,
118    YEAR(a.created_at) as yr,
119    month(a.created_at) as month,
120    avg(a.balance) as avg_balance_per_month
121    from customerss c
122    join accounts a on c.customer_id=a.customer_id
123    where a.created_at >= date_sub(curdate(), interval 1 year)
124    group by c.customer_id, yr, month;
```

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

customer_id	yr	month	avg_balance_per_month
5	2024	4	1141.6
8	2024	2	1451.93
16	2024	4	4960.615
19	2024	4	3931.98
26	2024	3	3262.49
28	2024	2	2340.48
29	2024	5	3040.2