

Case Study Questions

The following case study questions include some general data exploration analysis for the nodes and transactions before diving right into the core business questions and finishes with a challenging final request!

A. Customer Nodes Exploration

1. How many unique nodes are there on the Data Bank system?

```
Select count(distinct(node_id)) as unique_nodes
From customer_nodes;
```

	unique_nodes
▶	5

2. What is the number of nodes per region?

```
Select c.region_id, r.region_name, count(c.node_id) as number_of_nodes
From customer_nodes as c
Join regions as r On r.region_id = c.region_id
Group by c.region_id
Order by c.region_id;
```

region_id	region_name	number_of_nodes
1	Australia	770
2	America	735
3	Africa	714
4	Asia	665
5	Europe	616

3. How many customers are allocated to each region?

```
Select c.region_id, r.region_name, count(distinct(customer_id)) as customers
From customer_nodes as c
Join regions as r On r.region_id = c.region_id
Group by c.region_id
Order by c.region_id;
```

region_id	region_name	customers
1	Australia	110
2	America	105
3	Africa	102
4	Asia	95
5	Europe	88

4. How many days on average are customers reallocated to a different node?

```
SELECT round(avg(datediff(end_date, start_date)), 2) AS avg_reallocation_days
FROM customer_nodes
WHERE end_date != '9999-12-31';
```

avg_reallocation_days
14.63

5. What is the median, 80th and 95th percentile for this same reallocation days metric for each region?

```
Create View reallocation_days_table as
Select c.customer_id, c.region_id, c.node_id, r.region_name, datediff(c.end_date, c.start_date) as reallocation_day
From customer_nodes as c
Join regions as r On r.region_id = c.region_id
Where c.end_date != '9999-12-31';
```

customer_id	region_id	node_id	region_name	reallocation_day
1	3	4	Africa	1
2	3	5	Africa	14
3	5	4	Europe	22
4	5	4	Europe	12
5	3	3	Africa	8

```
Create View percentile As
Select *, percent_rank() over(partition by region_id order by reallocation_day)*100 as percent
From reallocation_days_table;
```

customer_id	region_id	node_id	region_name	reallocation_day	percent
14	1	2	Australia	0	0
161	1	2	Australia	0	0
204	1	3	Australia	0	0
14	1	5	Australia	0	0
51	1	2	Australia	0	0

```
Select region_id, region_name, reallocation_day
From percentile
Where percent > 95
Group by region_id;
```

region_id	region_name	reallocation_day
1	Australia	29
2	America	29
3	Africa	29
4	Asia	29
5	Europe	29

```

Select region_id, region_name, reallocation_day
From percentile
Where percent>80
Group by region_id;

```

region_id	region_name	reallocation_day
1	Australia	24
2	America	24
3	Africa	25
4	Asia	24
5	Europe	25

B. Customer Transactions

1. What is the unique count and total amount for each transaction type?

```

Select txn_type, count(txn_amount) as unique_count, sum(txn_amount) as total_amount
From customer_transactions
Group by txn_type
Order by txn_type;

```

txn_type	unique_count	total_amount
deposit	2671	1359168
purchase	1617	806537
withdrawal	1580	793003

2. What is the average total historical deposit counts and amounts for all customers?

```

Create View deposit As
Select customer_id, txn_type, count(*) as txn_count, avg(txn_amount) as average_amount
From customer_transactions
Where txn_type = 'deposit'
Group by customer_id, txn_type;

```

customer_id	txn_type	txn_count	average_amount
429	deposit	3	389.6667
155	deposit	6	607.8333
398	deposit	6	654.0000
255	deposit	3	341.0000
185	deposit	7	746.5714

```
Select round(avg(txn_count),0) as avg_count, round(avg(average_amount),2) as avg_amount
From deposit;
```

avg_count	avg_amount
5	508.61

3. For each month - how many Data Bank customers make more than 1 deposit and either 1 purchase or 1 withdrawal in a single month?

```
Create View transaction_count As
Select customer_id, month(txn_date) as txn_month,
sum(if(txn_type='deposit',1,0)) as deposit_count,
sum(if(txn_type='purchase',1,0)) as purchase_count,
sum(if(txn_type='withdrawal',1,0)) as withdrawal_count
From customer_transactions
Group by customer_id, txn_month;
```

customer_id	txn_month	deposit_count	purchase_count	withdrawal_count
429	1	1	0	0
155	1	1	3	0
398	1	3	2	2
255	1	1	1	0
185	1	1	0	0

ult 21 x

```
Select txn_month, count(distinct customer_id) as customer_count
From transaction_count
Where deposit_count > 1 and (purchase_count =1 or withdrawal_count =1)
Group by txn_month;
```

txn_month	customer_count
1	115
2	108
3	113
4	50

4. What is the closing balance for each customer at the end of the month?

```

Create View last_day_month As
Select customer_id, last_day(txn_date) as closing_month,
txn_type, txn_amount,
sum(case when txn_type = 'withdrawal' or txn_type = 'purchase' then (-txn_amount) Else txn_amount End) as transaction_balance
From customer_transactions
Group by customer_id, last_day(txn_date)
Order by customer_id;

```

customer_id	closing_month	txn_type	txn_amount	transaction_balance
1	2020-01-31	deposit	312	312
1	2020-03-31	purchase	612	-952
2	2020-01-31	deposit	549	549
2	2020-03-31	deposit	61	61
3	2020-01-31	deposit	144	144

```

SELECT customer_id, closing_month, COALESCE(transaction_balance, 0) AS monthly_change,
SUM(transaction_balance) OVER (PARTITION BY customer_id ORDER BY closing_month
ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS closing_balance
FROM last_day_month;

```

customer_id	closing_month	monthly_change	closing_balance
1	2020-01-31	312	312
1	2020-03-31	-952	-640
2	2020-01-31	549	549
2	2020-03-31	61	610
3	2020-01-31	144	144
3	2020-02-29	-965	-821
3	2020-03-31	-401	-1222
3	2020-04-30	493	-729
4	2020-01-31	848	848
4	2020-03-31	-193	655
5	2020-01-31	954	954
5	2020-03-31	-2877	-1923

5. What is the percentage of customers who increase their closing balance by more than 5%?

```

Create View sequence_table As
SELECT customer_id, closing_month, closing_balance,
ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY closing_month) AS sequence
FROM table_ex_4;

```

customer_id	closing_month	closing_balance	sequence
1	2020-01-31	312	1
1	2020-03-31	-640	2
2	2020-01-31	549	1
2	2020-03-31	610	2
3	2020-01-31	144	1
3	2020-02-29	-821	2
3	2020-03-31	-1222	3
3	2020-04-30	-729	4
4	2020-01-31	848	1
4	2020-03-31	655	2
5	2020-01-31	954	1

```
Create View next_balance_table As
SELECT customer_id, closing_month, closing_balance,
LEAD(closing_balance) OVER (PARTITION BY customer_id ORDER BY closing_month) AS next_balance
FROM sequence_table;
```

customer_id	closing_month	closing_balance	next_balance
1	2020-01-31	312	-640
1	2020-03-31	-640	NULL
2	2020-01-31	549	610
2	2020-03-31	610	NULL
3	2020-01-31	144	-821
3	2020-02-29	-821	-1222
3	2020-03-31	-1222	-729
3	2020-04-30	-729	NULL
4	2020-01-31	848	655
4	2020-03-31	655	NULL
5	2020-01-31	954	-1923

```
SELECT customer_id, closing_month, closing_balance, next_balance,
ROUND((1.0 * (next_balance - closing_balance)) / closing_balance,2) AS variance_value
FROM next_balance_table
WHERE closing_month = '2020-01-31' and next_balance is not null
GROUP BY customer_id, closing_month, closing_balance, next_balance
HAVING round(1.0 *(next_balance - closing_balance) / closing_balance,2) > 5.0;
```

customer_id	closing_month	closing_balance	next_balance	variance_value
183	2020-01-31	-540	-3729	5.91
368	2020-01-31	-526	-3490	5.63
398	2020-01-31	-429	-3171	6.39
384	2020-01-31	-10	-2486	247.60
309	2020-01-31	-363	-2404	5.62
420	2020-01-31	-280	-2117	6.56
267	2020-01-31	-193	-2068	9.72
272	2020-01-31	-228	-1673	6.34
56	2020-01-31	-67	-1646	23.57
265	2020-01-31	-25	-1481	58.24
171	2020-01-31	-197	-1400	6.11
187	2020-01-31	-211	-1379	5.54

Result 122 ✕