```sql
select a.Account_id ,c.Customer_id,c.Customer_FirstName ,
c.Customer_LastName , avg(a.balance) as Average_Balance from Accounts a
join Customers c on a.Customer_id=c.Customer_id
group by a.Account_id ,c.Customer_id,c.Customer_FirstName ,
c.Customer_LastName
```

96 %

Results Messages

	Account_id	Customer_id	Customer_FirstName	Customer_LastName	Average_Balance
1	101	1	John	Doe	5000.000000
2	102	2	Alice	Brown	10000.000000
3	103	3	Michael	Smith	7000.000000
4	104	4	Emma	Johnson	0.000000
5	105	5	David	Williams	3000.000000
6	106	6	Sophia	Miller	12000.000000
7	107	7	Daniel	Wilson	8000.000000
8	108	8	Olivia	Anderson	0.000000
9	109	9	James	Thomas	15000.000000
10	110	10	Emily	Harris	6000.000000

```sql
--2.Retrieve the top 10 highest account balances.

select top 5 (c.Customer_FirstName+' '+c.Customer_LastName)
as Customer_Name  , max(a.Balance) as Max_Balance from Customers c
join Accounts a on a.Customer_id=c.Customer_id
group by c.Customer_FirstName,c.Customer_LastName
order by Max_Balance desc
```

sults Messages

Customer_Name	Max_Balance
James Thomas	15000.00
Sophia Miller	12000.00
Alice Brown	10000.00
Daniel Wilson	8000.00
Michael Smith	7000.00

```sql
select c.Customer_id,c.Customer_FirstName , a.Account_id
, t.Transaction_id ,t.Transaction_date, sum(t.Amount) as Total_deposit
from Customers c
join Accounts a on a.Customer_id=c.Customer_id
join Transactions t on t.Account_id=a.Account_id
where t.Transaction_date='2024-03-01' and t.Transaction_type='deposit'
group by c.Customer_id,c.Customer_FirstName , a.Account_id
, t.Transaction_id ,t.Transaction_date
```

Results Messages

Customer_id	Customer_FirstName	Account_id	Transaction_id	Transaction_date	Total_deposit
1	John	101	1	2024-03-01	2000
5	David	105	5	2024-03-01	1500

```sql
--4.
select t.Transaction_id, t.Transaction_type,t.Transaction_date
, a.Account_Type from Transactions t
join Accounts a on a.Account_id=t.Account_id
order by a.Account_Type desc
```

5 %

Results Messages

	Transaction_id	Transaction_type	Transaction_date	Account_Type
1	4	transfer	2024-03-01	Zero_balance
2	8	deposit	2024-03-08	Zero_balance
3	3	deposit	2024-03-03	Saving
4	7	transfer	2024-03-07	Saving
5	5	deposit	2024-03-01	Saving
6	1	deposit	2024-03-01	Saving
7	10	deposit	2024-03-10	Saving
8	2	withdrawal	2024-03-02	Current
9	6	withdrawal	2024-03-06	Current
10	9	withdrawal	2024-03-09	Current

```sql
--6.
select c.Customer_FirstName,a.Balance,a.Account_Type,
a.Account_id from Customers c
join Accounts a on a.Customer_id=c.Customer_id
```

95 %

Results Messages

	Customer_FirstName	Balance	Account_Type	Account_id
1	John	5000.00	Saving	101
2	Alice	10000.00	Current	102
3	Michael	7000.00	Saving	103
4	Emma	0.00	Zero_balance	104
5	David	3000.00	Saving	105
6	Sophia	12000.00	Current	106
7	Daniel	8000.00	Saving	107
8	Olivia	0.00	Zero_balance	108
9	James	15000.00	Current	109
10	Emily	6000.00	Saving	110

```sql
--specific account.

select c.Customer_FirstName, c.Customer_DOB ,
a.Account_id,t.Transaction_type,t.Transaction_id,
t.Transaction_date  from Customers c
join Accounts a on a.Customer_id=c.Customer_id
join Transactions t on a.Account_id=t.Account_id
```

Results Messages

	Customer_FirstName	Customer_DOB	Account_id	Transaction_type	Transaction_id	Transaction_date
	John	1990-05-15	101	deposit	1	2024-03-01
	Alice	1985-08-20	102	withdrawal	2	2024-03-02
	Michael	1992-02-10	103	deposit	3	2024-03-03
	Emma	1995-12-05	104	transfer	4	2024-03-01
	David	1988-04-25	105	deposit	5	2024-03-01
	Sophia	1993-07-19	106	withdrawal	6	2024-03-06
	Daniel	1987-10-30	107	transfer	7	2024-03-07
	Olivia	1991-06-18	108	deposit	8	2024-03-08
	James	1994-09-09	109	withdrawal	9	2024-03-09
0	Emily	1989-11-23	110	deposit	10	2024-03-10

```sql
--8.Identify customers who have more than one account
 SELECT c.Customer_id, c.Customer_FirstName, c.Customer_LastName,
 COUNT(a.Account_id) AS Account_Count
FROM Customers c
JOIN Accounts a ON c.Customer_id = a.Customer_id
GROUP BY c.Customer_id, c.Customer_FirstName, c.Customer_LastName
HAVING COUNT(a.Account_id) > 1
```

esults Messages

Customer_id	Customer_FirstName	Customer_LastName	Account_Count
1	John	Doe	2
6	Sophia	Miller	2

```sql
select Account_id ,count(*) as Total_Transactions from Transactions
group by Account_id order by Total_Transactions
```

5 %

Results Messages

	Account_id	Total_Transactions
1	101	1
2	102	1
3	103	1
4	104	1
5	105	1
6	106	1
7	107	1
8	108	1
9	109	1
10	110	1

```sql
--8.Identify customers who have more than one account
 SELECT c.Customer_id, c.Customer_FirstName, c.Customer_LastName,
COUNT(a.Account_id) AS Account_Count
FROM Customers c
JOIN Accounts a ON c.Customer_id = a.Customer_id
GROUP BY c.Customer_id, c.Customer_FirstName, c.Customer_LastName
HAVING COUNT(a.Account_id) > 1
```

86 %

Results Messages

	Customer_id	Customer_FirstName	Customer_LastName	Account_Count
1	1	John	Doe	2
2	6	Sophia	Miller	2

```sql
--9.difference in transaction amounts between deposits and withdrawals

select  top 5 account_id,
sum(case when transaction_type = 'deposit' then amount else 0 end) -
sum(case when transaction_type = 'withdrawal' then amount else 0 end) as amount_
from transactions
group by account_id
order by amount_difference desc
```

Results | Messages

account_id	amount_difference
108	4000
110	3500
103	3000
101	2600
102	1500

```sql
--10.Calculate the average daily balance for each account
--over a specified period.
select account_id, avg(balance) as avg_daily_balance
from accounts
where account_id in (select distinct account_id from transactions
where transaction_date between '2024-01-01' and '2024-12-31')
group by account_id


 --11.Calculate the total balance for each account type
```

Results | Messages

account_id	avg_daily_balance
101	5000.000000
102	10000.000000
103	7000.000000
104	0.000000
105	3000.000000
106	12000.000000
107	8000.000000
108	0.000000

```sql
 --11.Calculate the total balance for e
 select Account_Type, sum(Balance) as T
 group by Account_Type

--12.the highest number of transactions
 select Account_id ,count(Transaction_T
 from Transactions group by Account_id
 order by Num_Transactions desc

 --13.high aggregate account balances,
 SELECT C.Customer_id,  C.Customer_Firs
 A.Account_Type, SUM(A.Balance) AS Tota
FROM Customers C
```

Results | Messages

Account_Type	Total_Amount
Current	57000.00
Saving	49000.00
Zero_balance	0.00

Account_id	Num_Transactions
101	3
102	2
103	1

```sql
--11.Calculate the total balance for each account type
select Account_Type, sum(Balance) as Total_Amount from Accounts
group by Account_Type

--12.the highest number of transactions order by descending order
select Account_id ,count(Transaction_Type) as Num_Transactions
from Transactions group by Account_id
order by Num_Transactions desc

--13.high aggregate account balances, along with their account types
SELECT C.Customer_id,  C.Customer_FirstName,C.Customer_LastName,
A.Account_Type, SUM(A.Balance) AS Total_Balance
FROM Customers C
```

86 %

Results | Messages

	Account_Type	Total_Amount
1	Current	57000.00
2	Saving	49000.00
3	Zero_balance	0.00

	Account_id	Num_Transactions
1	101	3
2	102	2
3	103	1
4	104	1
5	105	1
6	106	1
7	107	1
8	108	1
9	109	1

```sql
--13.high aggregate account balances, along with their account types
SELECT C.Customer_id,  C.Customer_FirstName,C.Customer_LastName,
A.Account_Type, SUM(A.Balance) AS Total_Balance
FROM Customers C
JOIN Accounts A ON C.Customer_id = A.Customer_id
GROUP BY C.Customer_id, C.Customer_FirstName,
C.Customer_LastName, A.Account_Type
HAVING SUM(A.Balance) > 10000
ORDER BY Total_Balance DESC
```

Results | Messages

	Customer_id	Customer_FirstName	Customer_LastName	Account_Type	Total_Balance
	1	John	Doe	Current	20000.00
	6	Sophia	Miller	Saving	20000.00
	9	James	Thomas	Current	15000.00
	6	Sophia	Miller	Current	12000.00

```sql
--14. list duplicate transactions based on transaction amount, date, and a
select Account_id, Amount, Transaction_date, count(*) as duplicate_count
from transactions
group by account_id, amount, transaction_date
having count(*) > 0;

-------------------------------------------------------------------------

 select * from Accounts
 select * from Customers
 select * from Transactions

 --1.Retrieve the customer(s) with the highest account balance.
```

86 %

Results | Messages

	Account_id	Amount	Transaction_date	duplicate_count
1	101	2000	2024-03-01	1
2	101	3400	2024-03-07	1
3	101	4000	2024-03-02	1
4	102	500	2024-03-02	1
5	102	2000	2024-03-10	1
6	103	3000	2024-03-03	1
7	104	1000	2024-03-01	1