Assignment-3

Task-1

1, Create the database named "HMBank"

```
create database HMBank;
use HMBank;
```

✓	37	14:03:18	create database HMBank	1 row(s) affected
✓	38	14:03:53	use HMBank	0 row(s) affected

2, Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

Customer:

```
4 •  create table customers
5    (customerid int primary key, firstname text, lastname text,
6    dateofbirth date, email text, phonenumber varchar(10),
7    address varchar(15));
8 •  desc customers;
9
```

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

Field	Type	Null	Key	Default	Extra
customerid	int	NO	PRI	NULL	
firstname	text	YES		NULL	
lastname	text	YES		NULL	
dateofbirth	date	YES		NULL	
email	text	YES		NULL	
phonenumber	varchar(10)	YES		NULL	
address	varchar(15)	YES		NULL	

Accounts:

```
25 •  create table accounts
26    (accountid int primary key, customerid int , foreign key(customerid) references customers(customerid),accounttype text,
27    balance decimal(10,2));
28 •  desc accounts;
```

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

Field	Type	Null	Key	Default	Extra
accountid	int	NO	PRI	NULL	
customerid	int	YES	MUL	NULL	
accounttype	text	YES		NULL	
balance	decimal(10,2)	YES		NULL	

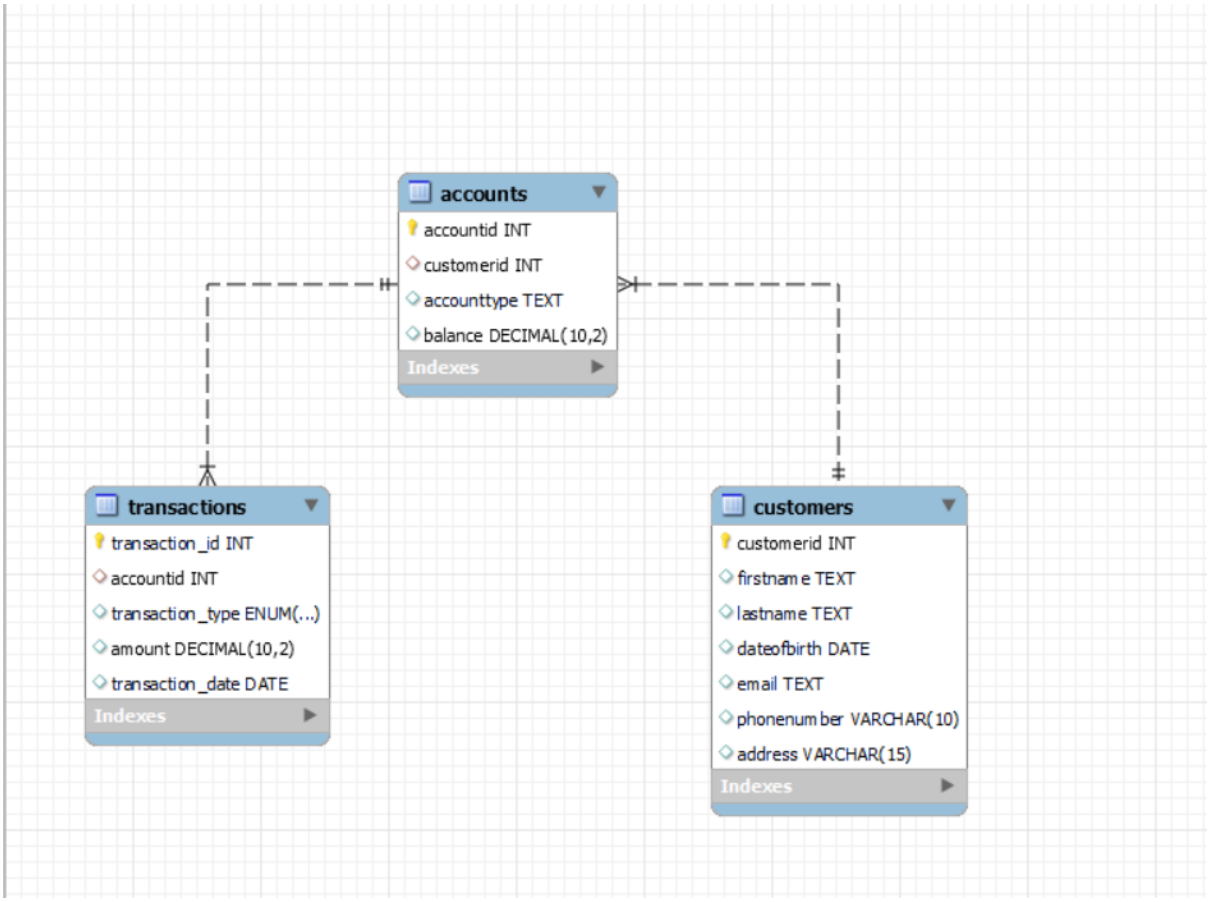
Transactions:

```
15 •  create table transactions
16    (transaction_id int primary key, accountid int, foreign key(accountid) references accounts(accountid), transaction_type
17    enum("deposit","transfer", "withdrawal"), amount decimal(10,2), transaction_date date);
18 •  desc transactions;
```

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

Field	Type	Null	Key	Default	Extra
transaction_id	int	NO	PRI	NULL	
accountid	int	YES	MUL	NULL	
transaction_type	enum('deposit','transfer','withdrawal')	YES		NULL	
amount	decimal(10,2)	YES		NULL	
transaction_date	date	YES		NULL	

3, Create an ERD (Entity Relationship Diagram) for the database.



5 Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships. • Customers • Accounts • Transactions,

Customers:

```
INSERT INTO customers
VALUES
  (1, 'Aarav', 'Sharma', '1990-05-15', 'aarav@gmail.com', '9876543210', '123 Main Street'),
  (2, 'Diya', 'Patel', '1988-09-22',   'diya@gmail.com',  '8765432109', '456 Oak Avenue'),
  (3, 'Vivaan', 'Gupta', '1995-03-10', 'vivaan@gmail.com', '7654321098', '789 Pine Road'),
  (4, 'Ananya', 'Kumar', '1992-07-08', 'ananya@gmail.com', '6543210987', '101 Cedar Lane'),
  (5, 'Advait', 'Verma', '1987-12-30', 'advait@gmail.com', '5432109876', '202 Elm Street'),
  (6, 'Ishita', 'Singh', '1993-04-18', 'ishita@gmail.com', '4321098765', '303 Birch Ave'),
  (7, 'Aryan', 'Yadav', '1989-11-25', 'aryan@gmail.com', '3210987654', '404 Maple Road'),
  (8, 'Sanya', 'Mishra', '1997-01-12', 'sanya@gmail.com', '2109876543', '5PinecrestDrive'),
  (9, 'Arjun', 'Reddy', '1994-06-05', 'arjun@gmail.com',  '1098765432', '606 OakwoodLane'),
  (10, 'Zara', 'Malhotra', '1991-08-20', 'zara@gmail.com', '9876543210', '707 Cedar Ridge');
select * from customers;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customerid	firstname	lastname	dateofbirth	email	phonenumber	address
1	Aarav	Sharma	1990-05-15	aarav@gmail.com	9876543210	123 Main Street
2	Diya	Patel	1988-09-22	diya@gmail.com	8765432109	456 Oak Avenue
3	Vivaan	Gupta	1995-03-10	vivaan@gmail.com	7654321098	789 Pine Road
4	Ananya	Kumar	1992-07-08	ananya@gmail.com	6543210987	101 Cedar Lane
5	Advait	Verma	1987-12-30	advait@gmail.com	5432109876	202 Elm Street
6	Ishita	Singh	1993-04-18	ishita@gmail.com	4321098765	303 Birch Ave
7	Aryan	Yadav	1989-11-25	aryan@gmail.com	3210987654	404 Maple Road
8	Sanya	Mishra	1997-01-12	sanya@gmail.com	2109876543	5PinecrestDrive
9	Arjun	Reddy	1994-06-05	arjun@gmail.com	1098765432	606 OakwoodLane
10	Zara	Malhotra	1991-08-20	zara@gmail.com	9876543210	707 Cedar Ridge
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Accounts:

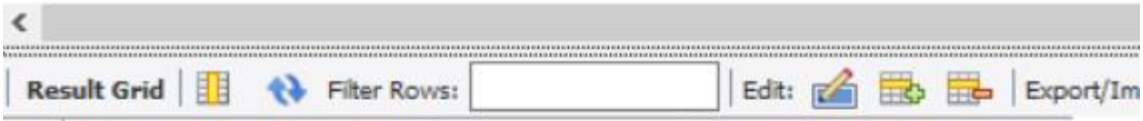
```
30 •  INSERT INTO accounts
31    VALUES
32      (100, 1, 'savings', 5000.00),
33      (101, 2, 'current', 10000.00),
34      (102, 3, 'savings', 7500.50),
35      (103, 4, 'zerobalance', 0.00),
36      (104, 5, 'current', 12000.75),
37      (105, 6, 'savings', 6000.25),
38      (106, 7, 'savings', 9000.00),
39      (107, 8, 'current', 8500.50),
40      (108, 9, 'zerobalance', 400.00),
41      (109, 10, 'savings', 11000.00),
42      (110, 6, 'zerobalance', 600.00);
43 •  select * from accounts;
```

Result Grid | Filter Rows: | Edit:

accountid	customerid	accounttype	balance
100	1	savings	5000.00
101	2	current	10000.00
102	3	savings	7500.50
103	4	zerobalance	200.00
104	5	current	12000.75
105	6	savings	6000.25
106	7	savings	9000.00
107	8	current	8500.50
108	9	zerobalance	400.00
109	10	savings	11000.00
110	6	zerobalance	600.00
NULL	NULL	NULL	NULL

Transactions:

```
53 •  INSERT INTO transactions
54    VALUES
55      (1, 100, 'deposit', 1000.00, '2024-01-13'),
56      (2, 101, 'withdrawal', 500.00, '2024-01-14'),
57      (3, 102, 'transfer', 1000.50, '2024-01-15'),
58      (4, 103, 'deposit', 200.00, '2024-01-16'),
59      (5, 104, 'withdrawal', 800.75, '2024-01-17'),
60      (6, 105, 'transfer', 300.25, '2024-01-18'),
61      (7, 106, 'deposit', 700.00, '2024-01-19'),
62      (8, 107, 'withdrawal', 450.50, '2024-01-20'),
63      (10,109, 'deposit', 1500.00, '2024-01-22');
64 •    select * from transactions;
```



transaction_id	accountid	transaction_type	amount	transaction_date
1	100	deposit	1000.00	2024-01-13
2	101	withdrawal	500.00	2024-01-14
3	102	transfer	1000.50	2024-01-15
4	103	deposit	200.00	2024-01-16
5	104	withdrawal	800.75	2024-01-17
6	105	transfer	300.25	2024-01-18
7	106	deposit	700.00	2024-01-19
8	107	withdrawal	450.50	2024-01-20
10	109	deposit	1500.00	2024-01-22
NULL	NULL	NULL	NULL	NULL

2

1, Write a SQL query to retrieve the name, account type and email of all customers.

```
72 •  select customers.firstname,customers.lastname, customers.email, accounts.accounttype from customers join accounts where
73     customers.customerid= accounts.customerid;
74
```

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

firstname	lastname	email	accounttype
Aarav	Sharma	aarav@gmail.com	savings
Diya	Patel	diya@gmail.com	current
Vivaan	Gupta	vivaan@gmail.com	savings
Ananya	Kumar	ananya@gmail.com	zerobalance
Advait	Verma	advait@gmail.com	current
Ishita	Singh	ishita@gmail.com	savings
Ishita	Singh	ishita@gmail.com	zerobalance
Aryan	Yadav	aryan@gmail.com	savings
Sanya	Mishra	sanya@gmail.com	current
Arjun	Reddy	arjun@gmail.com	zerobalance
Zara	Malhotra	zara@gmail.com	savings

2, Write a SQL query to list all transaction corresponding customer

```
74 •  select c.customerid, c.firstname,t.* from customers c join transactions t on accountid=104;
```

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

customerid	firstname	transaction_id	accountid	transaction_type	amount	transaction_date
1	Aarav	5	104	withdrawal	800.75	2024-01-17

3, Write a SQL query to increase the balance of a specific account by a certain amount.

```
75 •  update accounts set balance= balance+200 where accountid="103";
76 •  select * from accounts;
77
```

Result Grid | Filter Rows: | Edit: | Export/Import:

accountid	customerid	accounttype	balance
100	1	savings	5000.00
101	2	current	10000.00
102	3	savings	7500.50
103	4	zerobalance	200.00
104	5	current	12000.75
105	6	savings	6000.25
106	7	savings	9000.00
107	8	current	8500.50
108	9	zerobalance	400.00
109	10	savings	11000.00
NULL	NULL	NULL	NULL

4, Write a SQL query to Combine first and last names of customers as a full_name.

```
80
81 •  SELECT  *, CONCAT(FIRSTNAME, LASTNAME) AS fullname FROM customers;
82
```

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

customerid	firstname	lastname	dateofbirth	email	phonenumber	address	fullname
1	Aarav	Sharma	1990-05-15	aarav@gmail.com	9876543210	123 Main Street	AaravSharma
2	Diya	Patel	1988-09-22	diya@gmail.com	8765432109	456 Oak Avenue	DiyaPatel
3	Vivaan	Gupta	1995-03-10	vivaan@gmail.com	7654321098	789 Pine Road	VivaanGupta
4	Ananya	Kumar	1992-07-08	ananya@gmail.com	6543210987	101 Cedar Lane	AnanyaKumar
5	Advait	Verma	1987-12-30	advait@gmail.com	5432109876	202 Elm Street	AdvaitVerma
6	Ishita	Singh	1993-04-18	ishita@gmail.com	4321098765	303 Birch Ave	IshitaSingh
7	Aryan	Yadav	1989-11-25	aryan@gmail.com	3210987654	404 Maple Road	AryanYadav
8	Sanya	Mishra	1997-01-12	sanya@gmail.com	2109876543	5PinecrestDrive	SanyaMishra
9	Arjun	Reddy	1994-06-05	arjun@gmail.com	1098765432	606 OakwoodLane	ArjunReddy
10	Zara	Malhotra	1991-08-20	zara@gmail.com	9876543210	707 Cedar Ridge	ZaraMalhotra

5, Write a SQL query to remove accounts with a balance of zero where the account type is savings.

```
83 •  delete from accounts where balance="0" and accounttype="savings";
84 •  select * from accounts;
85
```

Result Grid | Filter Rows: | Edit: | Export/Import:

accountid	customerid	accounttype	balance
100	1	savings	5000.00
101	2	current	10000.00
102	3	savings	7500.50
103	4	zerobalance	200.00
104	5	current	12000.75
105	6	savings	6000.25
106	7	savings	9000.00
107	8	current	8500.50
108	9	zerobalance	400.00
109	10	savings	11000.00
NULL	NULL	NULL	NULL

Here we can see that there is no change in the table because there is no account with zero balance which has balance with 0.

6, Write a SQL query to Find customers living in a specific city.

```
86 •  select * from customers where address="456 Oak Avenue";
87
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cel

customerid	firstname	lastname	dateofbirth	email	phonenumber	address
2	Diya	Patel	1988-09-22	diya@gmail.com	8765432109	456 Oak Avenue
NULL	NULL	NULL	NULL	NULL	NULL	NULL

7, Write a SQL query to Get the account balance for a specific account

```
89 •  select balance from accounts where accountid="104";
90
```

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

balance
12000.75

8, Write a SQL query to List all current accounts with a balance greater than $1,000

```
92 •  select * from accounts where accounttype="current" and balance >1000;
93
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

accountid	customerid	accounttype	balance
101	2	current	10000.00
104	5	current	12000.75
107	8	current	8500.50
NULL	NULL	NULL	NULL

9, Write a SQL query to Retrieve all transactions for a specific account.

```
96 •  select * from transactions where accountid="104";
97
```

Result Grid | Filter Rows: | Edit: | Export/Import:

transaction_id	accountid	transaction_type	amount	transaction_date
5	104	withdrawal	800.75	2024-01-17
NULL	NULL	NULL	NULL	NULL

10, Write a SQL query to Calculate the interest accrued on savings accounts based on a given interest rate.

```
98 •  SELECT accountid,balance,accounttype,
99    balance * (5 / 100) AS interest_accrued FROM accounts where accounttype='savings';
100
```

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

accountid	balance	accounttype	interest_accrued
100	5000.00	savings	250.000000
102	7500.50	savings	375.025000
105	6000.25	savings	300.012500
106	9000.00	savings	450.000000
109	11000.00	savings	550.000000

Here I have taken interest rate as 5 so according to that I have displayed the interest.

11, Write a SQL query to Identify accounts where the balance is less than a specified overdraft limit.

```
104
105 •   select * from accounts where balance<"500";
106
```

Result Grid | Filter Rows: | Edit:

accountid	customerid	accounttype	balance
103	4	zerobalance	200.00
108	9	zerobalance	400.00
NULL	NULL	NULL	NULL

12, Write a SQL query to Find customers not living in a specific city

```
102 •   SELECT * FROM customers WHERE address <> '707 cedar ridge';
103
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

customerid	firstname	lastname	dateofbirth	email	phonenumber	address
1	Aarav	Sharma	1990-05-15	aarav@gmail.com	9876543210	123 Main Street
2	Diya	Patel	1988-09-22	diya@gmail.com	8765432109	456 Oak Avenue
3	Vivaan	Gupta	1995-03-10	vivaan@gmail.com	7654321098	789 Pine Road
4	Ananya	Kumar	1992-07-08	ananya@gmail.com	6543210987	101 Cedar Lane
5	Advait	Verma	1987-12-30	advait@gmail.com	5432109876	202 Elm Street
6	Ishita	Singh	1993-04-18	ishita@gmail.com	4321098765	303 Birch Ave
7	Aryan	Yadav	1989-11-25	aryan@gmail.com	3210987654	404 Maple Road
8	Sanya	Mishra	1997-01-12	sanya@gmail.com	2109876543	5PinecrestDrive
9	Arjun	Reddy	1994-06-05	arjun@gmail.com	1098765432	606 OakwoodLane
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Task-3

1, Write a SQL query to Find the average account balance for all customers.

```
116 •   select avg(balance) from accounts;
117
```

Result Grid | Filter Rows: | Export:

avg(balance)
6382.000000

2, Write a SQL query to Retrieve the top 10 highest account balances.

```
119 •   select c.firstname,c.lastname,a.AccountID, a.Balance
120     from Accounts as a , customers as c where c.customerid=a.customerid
121     order by Balance desc limit 10;
122
123
```

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

firstname	lastname	AccountID	Balance
Advait	Verma	104	12000.75
Zara	Malhotra	109	11000.00
Diya	Patel	101	10000.00
Aryan	Yadav	106	9000.00
Sanya	Mishra	107	8500.50
Vivaan	Gupta	102	7500.50
Ishita	Singh	105	6000.25
Aarav	Sharma	100	5000.00
Ishita	Singh	110	600.00
Arjun	Reddy	108	400.00

3, Write a SQL query to Calculate Total Deposits for All Customers in specific date

```
121 •   select accountid,count(accountid) as TotalDeposits from Transactions
122     where Transaction_Date = "2024-01-13" and transaction_type='deposit'
123     Group by accountid;
124
```

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

accountid	TotalDeposits
100	1

4, Write a SQL query to Find the Oldest and Newest Customers.

Oldest customer:

```
123 •   SELECT
124         CustomerID,
125         FirstName,
126         LastName,
127         dateofbirth
128     FROM Customers
129     ORDER BY customerid ASC limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap C

CustomerID	FirstName	LastName	dateofbirth
1	Aarav	Sharma	1990-05-15

Newest Customer:

```
131 •   SELECT
132             CustomerID,
133             FirstName,
134             LastName,
135             dateofbirth
136     FROM Customers
137     ORDER BY customerid DESC limit 1;
138
139
```

Result Grid | Filter Rows: | Export:

CustomerID	FirstName	LastName	dateofbirth
10	Zara	Malhotra	1991-08-20

5, Write a SQL query to Retrieve transaction details along with the account type.

```
140 •   select t.*, a.accounttype from transactions as t
141     join accounts a on  a.accountid = t.accountid where t.accountid=101;
142
143
```

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

transaction_id	accountid	transaction_type	amount	transaction_date	accounttype
2	101	withdrawal	500.00	2024-01-14	current

6, Write a SQL query to Get a list of customers along with their account details.

```
151 •   select firstname,lastname, a.* from customers as c
152     right outer join accounts a on c.customerid=a.customerid;
153
```

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

firstname	lastname	accountid	customerid	accounttype	balance
Aarav	Sharma	100	1	savings	5000.00
Diya	Patel	101	2	current	10000.00
Vivaan	Gupta	102	3	savings	7500.50
Ananya	Kumar	103	4	zerobalance	200.00
Advait	Verma	104	5	current	12000.75
Ishita	Singh	105	6	savings	6000.25
Aryan	Yadav	106	7	savings	9000.00
Sanya	Mishra	107	8	current	8500.50
Arjun	Reddy	108	9	zerobalance	400.00
Zara	Malhotra	109	10	savings	11000.00
Ishita	Singh	110	6	zerobalance	600.00

.

7, Write a SQL query to Retrieve transaction details along with customer information for a specific account.

```
148 •   select c.*, t.* from customers as c
149     join accounts a on c.customerid=a.customerid
150     join transactions t on a.accountid=t.accountid where t.accountid=100;
```

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

customerid	firstname	lastname	dateofbirth	email	phonenumber	address	transaction_id	accountid	transaction_type	amount	transaction_date
1	Aarav	Sharma	1990-05-15	aarav@gmail.com	9876543210	123 Main Street	1	100	deposit	1000.00	2024-01-13

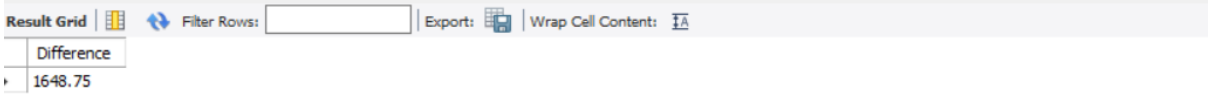
8, Write a SQL query to Identify customers who have more than one account.

```
160 •   SELECT c.CustomerID,
161            c.FirstName,
162            c.LastName
163     FROM Customers c
164     JOIN Accounts a ON c.CustomerID = a.CustomerID
165     GROUP BY c.CustomerID, c.FirstName, c.LastName
166     HAVING COUNT(a.AccountID) > 1;
```

CustomerID	FirstName	LastName
6	Ishita	Singh

9, Write a SQL query to Calculate the difference in transaction amounts between deposits and Withdrawals

```
164 •  select
165      (select COALESCE(SUM(Amount), 0) from Transactions where Transaction_Type = 'Deposit') -
166      (select COALESCE(SUM(Amount), 0) from Transactions where Transaction_Type = 'Withdrawal') as Difference;
```



Difference
1648.75

10, Write a SQL query to Calculate the average daily balance for each account over a specified period.

```
184 •  select accountid, avg(amount) as average_daily_balance from transactions
185    where transaction_date between '2024-01-13' and '2024-2-15'
186    group by accountid;
```

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

accountid	average_daily_balance
100	1000.000000
101	500.000000
102	1000.500000
103	200.000000
104	800.750000
105	300.250000
106	700.000000
107	450.500000
109	1500.000000

11, Calculate the total balance for each account type.

```
193 •  select AccountType, SUM(Balance) as TotalBalance
194    from Accounts group by AccountType;
195
```

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

AccountType	TotalBalance
savings	38500.75
current	30501.25
zerobalance	1200.00

12, . Identify accounts with the highest number of transactions order by descending order

```
204 •  select a.AccountID,COUNT(Transaction_ID) as TransactionCount from Transactions t
205    right join accounts a on a.accountid= t.accountid
206    group by AccountID order by TransactionCount desc;
207
```

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

AccountID	TransactionCount
100	1
101	1
102	1
103	1
104	1
105	1
106	1
107	1
109	1
108	0

13, List customers with high aggregate account balances, along with their account types

```
204 •  select c.CustomerID,c.FirstName,c.LastName,a.AccountType,
205    SUM(a.Balance) as AggregateBalance from Customers c
206    join Accounts a on c.CustomerID = a.CustomerID
207    group by c.CustomerID, c.FirstName, c.LastName, a.AccountType
208    ORDER BY AggregateBalance DESC;
```

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

CustomerID	FirstName	LastName	AccountType	AggregateBalance
5	Advait	Verma	current	12000.75
10	Zara	Malhotra	savings	11000.00
2	Diya	Patel	current	10000.00
7	Aryan	Yadav	savings	9000.00
8	Sanya	Mishra	current	8500.50
3	Vivaan	Gupta	savings	7500.50
6	Ishita	Singh	savings	6000.25
1	Aarav	Sharma	savings	5000.00
6	Ishita	Singh	zerobalance	600.00
9	Arjun	Reddy	zerobalance	400.00
4	Ananya	Kumar	zerobalance	200.00

14, Identify and list duplicate transactions based on transaction amount, date, and account.

```
201 •   select Amount,Transaction_Date,AccountID,
202     COUNT(Transaction_ID) as DuplicateCount from Transactions
203     group by Amount, Transaction_Date, AccountID
204     having COUNT(Transaction_ID) > 1;
205
```

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

Amount	Transaction_Date	AccountID	DuplicateCount

Here nothing is displayed because there was no duplicate transactions on the table

Task -4

1, Retrieve the customer(s) with the highest account balance

```
208 •   select c.CustomerID,c.FirstName,c.LastName,a.AccountID,a.Balance
209     from Customers c
210     join Accounts a ON c.CustomerID = a.CustomerID
211     where a.Balance = (select MAX(Balance) from Accounts);
```

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

CustomerID	FirstName	LastName	AccountID	Balance
5	Advait	Verma	104	12000.75

2, Calculate the average account balance for customers who have more than one account.

```
226 •   select c.CustomerID, c.FirstName,c.LastName,
227     avg(a.Balance) AS AverageAccountBalance
228     from Customers c
229     join Accounts a on c.CustomerID = a.CustomerID
230     where c.CustomerID in (select CustomerID
231     from Accounts a1 where a.customerid=a1.customerid
232     having COUNT(AccountID) >1
233         );
234
```

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

CustomerID	FirstName	LastName	AverageAccountBalance
6	Ishita	Singh	3300.125000

Here we

3, Retrieve accounts with transactions whose amounts exceed the average transaction amount.

```
225 • with AverageTransaction as (
226         select avg(Amount) as AvgTransactionAmount
227         from Transactions
228     )
229     select a.AccountID,a.AccountType,t.*,t.Amount
230     from Accounts a
231     join Transactions t on a.AccountID = t.AccountID
232     cross join AverageTransaction avgT
233     where t.Amount > avgT.AvgTransactionAmount
234
235
```

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

AccountID	AccountType	transaction_id	accountid	transaction_type	amount	transaction_date	Amount
100	savings	1	100	deposit	1000.00	2024-01-13	1000.00
102	savings	3	102	transfer	1000.50	2024-01-15	1000.50
104	current	5	104	withdrawal	800.75	2024-01-17	800.75
106	savings	7	106	deposit	700.00	2024-01-19	700.00
109	savings	10	109	deposit	1500.00	2024-01-22	1500.00

4, Identify customers who have no recorded transactions.

```
254 •  SELECT c.CustomerID, c.FirstName, c.LastName
255    FROM Customers c
256    JOIN accounts a on c.customerid= a.customerid
257    left join Transactions t ON a.accountid=t.accountid
258    WHERE t.Transaction_ID IS NULL;
259
```

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

CustomerID	FirstName	LastName
9	Arjun	Reddy

5, Calculate the total balance of accounts with no recorded transactions

```
260 •  select sum(balance) from accounts where not exists (select accountid from transactions
261    where accounts.accountid=transactions.accountid);
262
```

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

sum(balance)
400.00

6, Retrieve transactions for accounts with the lowest balance.

```
265 • WITH LowestBalanceAccounts AS (
266   SELECT AccountID FROM Accounts
267   ORDER BY Balance LIMIT 1
268   )
269
270   SELECT t.Transaction_ID,t.AccountID,t.Transaction_Date,t.Amount
271   FROM Transactions t
272   JOIN
273       LowestBalanceAccounts lba ON t.AccountID = lba.AccountID;
```

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

Transaction_ID	AccountID	Transaction_Date	Amount
4	103	2024-01-16	200.00

7, Identify customers who have accounts of multiple types

```
271 •   SELECT a.customerid FROM accounts a GROUP BY customerid
272     HAVING COUNT(DISTINCT accounttype) > 1;
273
274
```

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

customerid
6

8, Calculate the percentage of each account type out of the total number of accounts

```
277 •  SELECT
278        AccountType,
279        COUNT(AccountID) AS NumberOfAccounts,
280        (COUNT(AccountID) * 100.0 / (SELECT COUNT(AccountID) FROM Accounts)) AS Percentage
281    FROM
282        Accounts
283    GROUP BY
284        AccountType;
285
```

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

AccountType	NumberOfAccounts	Percentage
savings	5	45.45455
current	3	27.27273
zerobalance	3	27.27273

9, Retrieve all transactions for a customer with a given customer_id

```
271 •  select t.Transaction_ID,t.Transaction_Date,t.Amount
272    from Transactions  ,customers as c
273    join accounts a  on c.customerid= a.customerid
274    join transactions t  on a.accountID = t.accountID
275    where c.CustomerID = 4;
```

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

Transaction_ID	Transaction_Date	Amount
4	2024-01-16	200.00

10, Calculate the total balance for each account type, including a subquery within the SELECT clause.

```
296 •  SELECT AccountType,
297        (SELECT SUM(Balance) FROM Accounts a2 WHERE a2.AccountType = a1.AccountType) AS TotalBalance
298    FROM Accounts a1
299    GROUP BY AccountType;
300
```

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

AccountType	TotalBalance
savings	38500.75
current	30501.25
zerobalance	1200.00

Submitted By:

Devaki Akash