

Assignment: 3

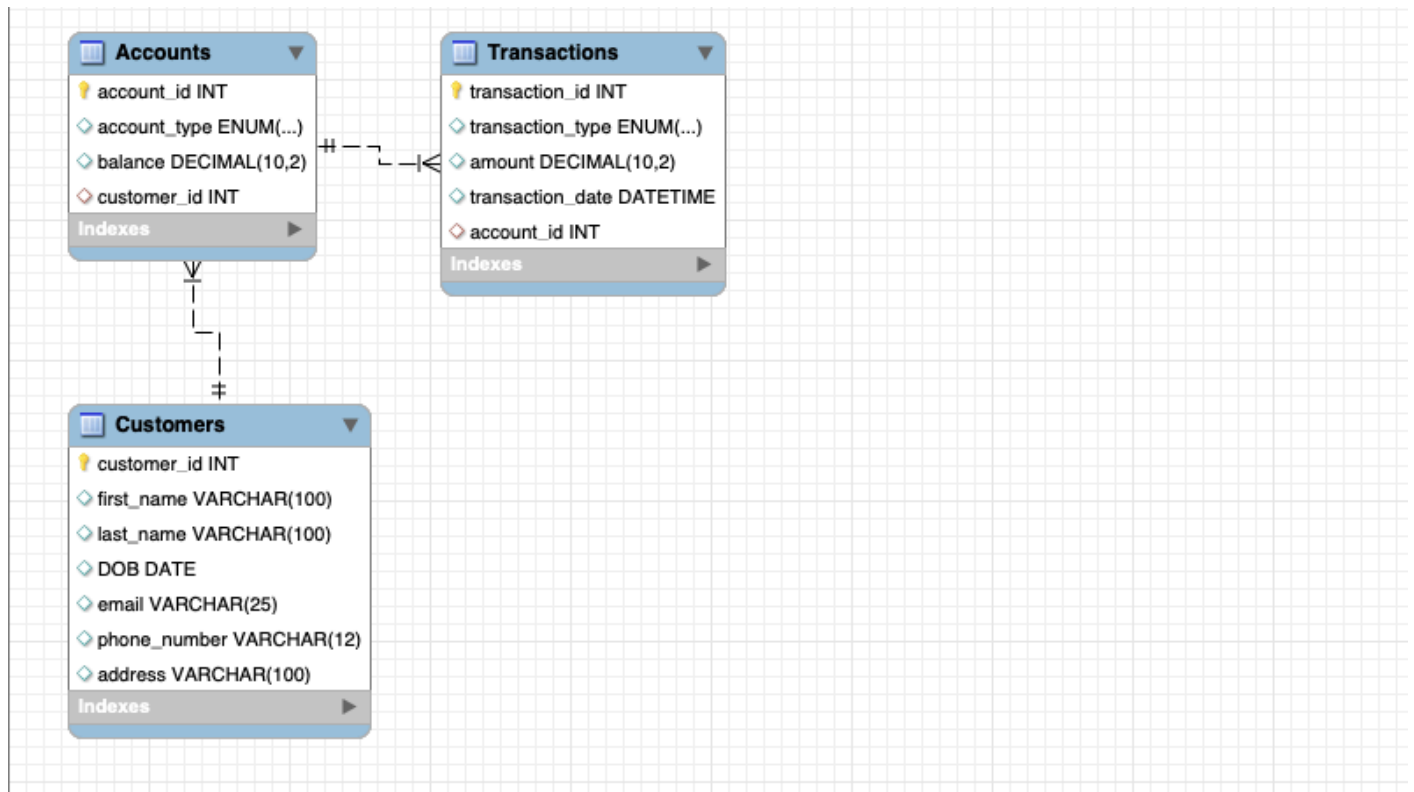
Banking system

Tasks 1: Database Design:

1. Create the database named "HMBank".
 2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema
 - 3.
 - 4.
 5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.
 6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.
- Customers
 - Accounts
 - Transactions

```
1 • CREATE Database if not exists HMBank;
2 • use HMBank;
3
4 • CREATE table if not exists Customers(
5     customer_id INT PRIMARY KEY,
6     first_name VARCHAR(100),
7     last_name VARCHAR(100),
8     DOB DATE,
9     email VARCHAR(25),
10    phone_number VARCHAR(12),
11    address VARCHAR(100)
12 );
13
14 • CREATE table if not exists Accounts(
15     account_id INT PRIMARY KEY,
16     account_type ENUM('savings', 'current' , 'zero balance'),
17     balance DECIMAL (10,2),
18     customer_id int ,FOREIGN KEY(customer_id) REFERENCES Customers(customer_id)
19 );
20
21 • CREATE table if not exists Transactions(
22     transaction_id INT PRIMARY KEY,
23     transaction_type ENUM('deposit', 'withdrawal', 'transfer'),
24     amount DECIMAL(10,2),
25     transaction_date DATETIME,
26     account_id int ,FOREIGN KEY(account_id) REFERENCES Accounts(account_id)
27 );
```

4. Create an ERD



Tasks 2: Select, Where, Between, AND, LIKE:

1. Insert at least 10 sample records into each of the following tables.

- Customers
- Accounts
- Transactions

```

29 • INSERT INTO Customers VALUES(1, 'Ajay', 'Chaudhary', '2000-11-24', 'ajay.chaudhary@gmail.com', '63678765
30 (2, 'Tarun', 'Singh', '2000-09-20', 'tarun.singh@gmail.com', '9823454523', 'Charbagh'),
31 (3, 'Sridhar', 'Mishra', '1998-12-10', 'sridhar.mishra@gmail.com', '9345324567', 'Indira nagar'),
32 (4, 'Raghav', 'Aggarwal', '1994-01-11', 'raghav.aggarwal@gmail.com', '8834234576', 'Munshi puliya'),
33 (5, 'Abhishek', 'Verma', '1994-07-17', 'abhishek.verma@gmail.com', '7052218976', 'Shaeed path'),
34 (6, 'Paras', 'Gupta', '1993-11-03', 'paras.gupta@gmail.com', '7275385128', 'Polytechnic road'),
35 (7, 'Aman', 'Yadav', '2001-10-10', 'aman.yadav@gmail.com', '9453678782', 'Hazratganj'),
36 (8, 'Shivansh', 'Singh', '2002-01-27', 'shivansh.singh@gmail.com', '7633221456', 'Aminabad'),
37 (9, 'Rishabh', 'Dixit', '2001-02-07', 'rishabh.dixit@gmail.com', '9867563423', 'Qaiserbagh'),
38 (10, 'Mitali', 'Srivastava', '1996-04-05', 'mitali.srivast@gmail.com', '8734234565', 'Nawabganj');
39 • select * from Customers;
40
41 • INSERT INTO Accounts VALUES(101, 'current', 50000.00, 1),

```

100% 1:40

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

	customer_id	first_name	last_name	DOB	email	phone_number	address	
▶	1	Ajay	Chaudhary	2000-11-24	ajay.chaudhary@gmail.com	6367876567	Alambagh	
	2	Tarun	Singh	2000-09-20	tarun.singh@gmail.com	9823454523	Charbagh	
	3	Sridhar	Mishra	1998-12-10	sridhar.mishra@gmail.com	9345324567	Indira nagar	
	4	Raghav	Aggarwal	1994-01-11	raghav.aggarwal@gmail.com	8834234576	Munshi puliya	
	5	Abhishek	Verma	1994-07-17	abhishek.verma@gmail.com	7052218976	Shaeed path	
	6	Paras	Gupta	1993-11-03	paras.gupta@gmail.com	7275385128	Polytechnic road	
	7	Aman	Yadav	2001-10-10	aman.yadav@gmail.com	9453678782	Hazratganj	
	8	Shivansh	Singh	2002-01-27	shivansh.singh@gmail.com	7633221456	Aminabad	
	9	Rishabh	Dixit	2001-02-07	rishabh.dixit@gmail.com	9867563423	Qaiserbagh	
	10	Mitali	Srivastava	1996-04-05	mitali.srivast@gmail.com	8734234565	Nawabganj	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Result Grid

Form Editor

Field Types

```

> db1
  > sys
    > Tables
    > Views
    > Stored...
    > Functio...
  > TechShop
    > Tables
    > Views
    > Stored...
    > Functio...

41 • INSERT INTO Accounts VALUES(101, 'current', 50000.00, 1),
42 (102, 'savings', 25000.00, 2),
43 (103, 'zero balance', 0.00, 3),
44 (104, 'current', 20000.00, 4),
45 (105, 'savings', 75000.00, 5),
46 (106, 'zero balance', 0.00, 6),
47 (107, 'current', 42500.00, 7),
48 (108, 'savings', 33000.00, 8),
49 (109, 'zero balance', 0.00, 9),
50 (110, 'current', 82000.00, 10);
51 • select * from Accounts;
52

```

100% 1:52

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

	account_id	account_ty...	balance	customer_id
▶	101	current	50000.00	1
	102	savings	25000.00	2
	103	zero balance	0.00	3
	104	current	20000.00	4
	105	savings	75000.00	5
	106	zero balance	0.00	6
	107	current	42500.00	7
	108	savings	33000.00	8
	109	zero balance	0.00	9
	110	current	82000.00	10
	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Field Types

Object Info Session
No object selected

SQL Query Editor showing a query to insert data into the Transactions table and then select all records.

```

INSERT INTO Transactions VALUES
54 (1001, 'deposit', 1000.00, '2023-01-10', 101),
55 (1002, 'withdrawal', 500.00, '2023-02-05', 102),
56 (1003, 'deposit', 3000.00, '2023-03-15', 103),
57 (1004, 'withdrawal', 6000.00, '2023-04-25', 104),
58 (1005, 'deposit', 7000.00, '2023-05-20', 105),
59 (1006, 'deposit', 9000.00, '2023-06-19', 106),
60 (1007, 'deposit', 1000.00, '2023-07-11', 107),
61 (1008, 'withdrawal', 10000.00, '2023-08-13', 108),
62 (1009, 'deposit', 5500.00, '2023-09-27', 109),
63 (1010, 'withdrawal', 1000.00, '2023-10-06', 110);
64 select * from Transactions;

```

Result Grid showing the inserted data:

transaction...	transaction_ty...	amount	transaction_date	account_id
1001	deposit	1000.00	2023-01-10 00:00:00	101
1002	withdrawal	500.00	2023-02-05 00:00:00	102
1003	deposit	3000.00	2023-03-15 00:00:00	103
1004	withdrawal	6000.00	2023-04-25 00:00:00	104
1005	deposit	7000.00	2023-05-20 00:00:00	105
1006	deposit	9000.00	2023-06-19 00:00:00	106
1007	deposit	1000.00	2023-07-11 00:00:00	107
1008	withdrawal	10000.00	2023-08-13 00:00:00	108
1009	deposit	5500.00	2023-09-27 00:00:00	109
1010	withdrawal	1000.00	2023-10-06 00:00:00	110
NULL	NULL	NULL	NULL	NULL

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

SQL Query Editor showing a query to retrieve customer information.

```

66 SELECT c.first_name, c.last_name, a.account_type, c.email
67 FROM Customers c
68 JOIN Accounts a ON c.customer_id = a.customer_id;

```

Result Grid showing the retrieved data:

first_name	last_name	account_ty...	email
Ajay	Chaudhary	current	ajay.chaudhary@gmail.com
Tarun	Singh	savings	tarun.singh@gmail.com
Sridhar	Mishra	zero balance	sridhar.mishra@gmail.com
Raghav	Aggarwal	current	raghav.aggarwal@gmail.com
Abhishek	Verma	savings	abhishek.verma@gmail.com
Paras	Gupta	zero balance	paras.gupta@gmail.com
Aman	Yadav	current	aman.yadav@gmail.com
Shivansh	Singh	savings	shivansh.singh@gmail.com
Rishabh	Dixit	zero balance	rishabh.dixit@gmail.com
Mitali	Srivastava	current	mitali.srivastava@gmail.com

3. Write a SQL query to list all transaction corresponding customer.

SQL Query Editor showing a query to list all transactions with customer information.

```

71 SELECT t.transaction_id, t.transaction_type, t.amount, t.transaction_date
72 FROM Transactions t
73 JOIN Accounts a ON t.account_id = a.account_id
74 JOIN Customers c ON a.customer_id = c.customer_id;

```

Result Grid showing the retrieved data:

transaction...	transaction_ty...	amount	transaction_date
1001	deposit	1000.00	2023-01-10 00:00:00
1002	withdrawal	500.00	2023-02-05 00:00:00
1003	deposit	3000.00	2023-03-15 00:00:00
1004	withdrawal	6000.00	2023-04-25 00:00:00
1005	deposit	7000.00	2023-05-20 00:00:00
1006	deposit	9000.00	2023-06-19 00:00:00
1007	deposit	1000.00	2023-07-11 00:00:00
1008	withdrawal	10000.00	2023-08-13 00:00:00
1009	deposit	5500.00	2023-09-27 00:00:00
1010	withdrawal	1000.00	2023-10-06 00:00:00

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

```

76 • UPDATE Accounts
77   SET balance = balance + 1000
78   WHERE account_id = 103;
79 • select * from Accounts;
80
81
100% 24:79

```

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

account_id	account_ty...	balance	customer_id
101	current	50000.00	1
102	savings	25000.00	2
103	zero balance	1000.00	3
104	current	20000.00	4
105	savings	75000.00	5
106	zero balance	0.00	6
107	current	42500.00	7
108	savings	33000.00	8
109	zero balance	0.00	9
110	current	82000.00	10
NULL	NULL	NULL	NULL

Accounts 11 Apply

5. Write a SQL query to Combine first and last names of customers as a full name.

```

81 • SELECT CONCAT(first_name, ' ', last_name) AS full_name
82   FROM Customers;
83
84
85
100% 1:86

```

Result Grid Filter Rows: Search Export:

full_name
Ajay Chaudhary
Tarun Singh
Sridhar Mishra
Raghav Aggarwal
Abhishek Verma
Paras Gupta
Aman Yadav
Shivansh Singh
Rishabh Dixit
Mitali Srivastava

Result 12 Read Only

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

```

84 • DELETE FROM Accounts
85   WHERE balance = 0 AND account_type = 'savings';
86 • select * from Accounts;
87
88
100% 1:87

```

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

account_id	account_ty...	balance	customer_id
101	current	50000.00	1
102	savings	25000.00	2
103	zero balance	1000.00	3
104	current	20000.00	4
105	savings	75000.00	5
106	zero balance	0.00	6
107	current	42500.00	7
108	savings	33000.00	8
109	zero balance	0.00	9
110	current	82000.00	10
NULL	NULL	NULL	NULL

Accounts 13 Apply

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

Object Info Session

No object selected

100%
31:89

Result Grid

Filter Rows:

Search

Edit:

Export/Import:

customer_id	first_name	last_name	DOB	email	phone_number	address
1	Ajay	Chaudhary	2000-11-24	ajay.chaudhary@gmail.com	6367876567	Alambagh
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customers 14
Apply

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	357 08:35:52	SELECT * FROM Customers WHERE address LIKE 'Alam...	1 row(s) returned	0.0028 sec / 0.00002...

8. Write a SQL query to Get the account balance for a specific account.

```
91 • SELECT balance
92 FROM Accounts
93 WHERE account_id = 105;
94
```

100% 24:93

Result Grid Filter Rows: Search Export:

balance
75000.00

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

95 • **SELECT ***

96 **FROM Accounts**

97 **WHERE account_type = 'current' AND balance > 10000.00;**

100% 55:97

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

	account_id	account_ty...	balance	customer_id
▶	101	current	50000.00	1
◀	104	current	20000.00	4
	107	current	42500.00	7
◀	110	current	82000.00	10
	NULL	NULL	NULL	NULL

Accounts 16

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

```
99 • SELECT *
100 FROM Transactions
101 WHERE account_id = 107;
102
```

100% 1:102

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

transaction...	transaction_ty...	amount	transaction_date	account_id
1007	deposit	1000.00	2023-07-11 00:00:00	107
NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor

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

```
103 • SELECT account_id, balance * 1.5 / 100 AS interest_accrued
104 FROM Accounts
105 WHERE account_type = 'savings';
106
```

100% 32:105

Result Grid Filter Rows: Search Export:

account_id	interest_accru...
102	375.0000000
105	1125.0000000
108	495.0000000

Result Grid Form Editor

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

Tables Views Stored... Functio...

```
107 • SELECT *
108 FROM Accounts
109 WHERE balance < 50000;
110
```

100% 23:109

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

account_id	account_ty...	balance	customer_id
102	savings	25000.00	2
103	zero balance	1000.00	3
104	current	20000.00	4
106	zero balance	0.00	6
107	current	42500.00	7
108	savings	33000.00	8
109	zero balance	0.00	9
NULL	NULL	NULL	NULL

Accounts 19 Apply

Object Info Session No object selected

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 362	08:44:15	SELECT * FROM Accounts WHERE balance < 50000 LIM...	7 row(s) returned	0.0012 sec / 0.00001...

Result Grid Form Editor Field Types

13. Write a SQL query to Find customers not living in a specific city.

```
111 • SELECT *
112 FROM Customers
113 WHERE address NOT LIKE 'Alambagh';
114
115
```

100% 1:114

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

	customer_id	first_name	last_name	DOB	email	phone_number	address
▶	2	Tarun	Singh	2000-09-20	tarun.singh@gmail.com	9823454523	Charbagh
	3	Sridhar	Mishra	1998-12-10	sridhar.mishra@gmail.com	9345324567	Indira nagar
	4	Raghav	Aggarwal	1994-01-11	raghav.aggarwal@gmail.com	8834234576	Munshi puliya
	5	Abhishek	Verma	1994-07-17	abhishek.verma@gmail.com	7052218976	Shaeed path
	6	Paras	Gupta	1993-11-03	paras.gupta@gmail.com	7275385128	Polytechnic road
	7	Aman	Yadav	2001-10-10	aman.yadav@gmail.com	9453678782	Hazratganj
	8	Shivansh	Singh	2002-01-27	shivansh.singh@gmail.com	7633221456	Aminabad
	9	Rishabh	Dixit	2001-02-07	rishabh.dixit@gmail.com	9867563423	Qaiserbagh
	10	Mitali	Srivastava	1996-04-05	mitali.srivastava@gmail.com	8734234565	Nawabganj
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid Form Editor Field Types

Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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

```
115 • SELECT AVG(balance) AS average_balance
116 FROM Accounts;
117
```

100% 15:116

Result Grid Filter Rows: Search Export:

	average_balance
▶	32850.000000

Object Info Session No object selected

Result Grid Form Editor Field Types

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

```
118 • SELECT *
119 FROM Accounts
120 ORDER BY balance DESC
121 LIMIT 10;
122
```

100% 10:121

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

	account_id	account_ty...	balance	customer_id
▶	110	current	82000.00	10
	105	savings	75000.00	5
	101	current	50000.00	1
	107	current	42500.00	7
	108	savings	33000.00	8
	102	savings	25000.00	2
	104	current	20000.00	4
	103	zero balance	1000.00	3
	106	zero balance	0.00	6
	109	zero balance	0.00	9
	NULL	NULL	NULL	NULL

Object Info Session No object selected

Accounts 22 Apply

Result Grid Form Editor Field Types

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

The screenshot shows a SQL query editor with the following code:

```
123 • SELECT SUM(amount) AS total_deposits
124 FROM Transactions
125 WHERE transaction_type = 'deposit' AND DATE(transaction_date) = '2023-01-10';
126
127
128
```

The result grid below the query shows a single row with the total deposits for the specified date.

total_deposits
1000.00

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

The screenshot shows two SQL queries and their corresponding result grids.

Query 1: Find the Oldest Customer

```
127 • SELECT *
128 FROM Customers
129 ORDER BY DOB ASC
130 LIMIT 1;
131
132
```

The result grid for this query shows the oldest customer:

customer_id	first_name	last_name	DOB	email	phone_number	address
6	Paras	Gupta	1993-11-03	paras.gupta@gmail.com	7275385128	Polytechnic road

Query 2: Find the Newest Customer

```
132 • SELECT *
133 FROM Customers
134 ORDER BY DOB DESC
135 LIMIT 1;
136
137
```

The result grid for this query shows the newest customer:

customer_id	first_name	last_name	DOB	email	phone_number	address
8	Shivansh	Singh	2002-01-27	shivansh.singh@gmail.com	7633221456	Aminabad

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

```
137 • SELECT t.*, a.account_type
138 FROM Transactions t
139 JOIN Accounts a ON t.account_id = a.account_id;
140
```

100% 1:140

Result Grid Filter Rows: Search Export:

transaction...	transaction_ty...	amount	transaction_date	account_id	account_ty...
1001	deposit	1000.00	2023-01-10 00:00:00	101	current
1002	withdrawal	500.00	2023-02-05 00:00:00	102	savings
1003	deposit	3000.00	2023-03-15 00:00:00	103	zero balance
1004	withdrawal	6000.00	2023-04-25 00:00:00	104	current
1005	deposit	7000.00	2023-05-20 00:00:00	105	savings
1006	deposit	9000.00	2023-06-19 00:00:00	106	zero balance
1007	deposit	1000.00	2023-07-11 00:00:00	107	current
1008	withdrawal	10000.00	2023-08-13 00:00:00	108	savings
1009	deposit	5500.00	2023-09-27 00:00:00	109	zero balance
1010	withdrawal	1000.00	2023-10-06 00:00:00	110	current

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

```
141 • SELECT c.*, a.*
142 FROM Customers c
143 LEFT JOIN Accounts a ON c.customer_id = a.customer_id;
144
```

100% 1:140

Result Grid Filter Rows: Search Export:

customer_id	first_name	last_name	DOB	email	phone_number	address	account_id	account_ty...	balance	customer_id
1	Ajay	Chaudhary	2000-11-24	ajay.chaudhary@gmail.com	6367876567	Alambagh	101	current	50000.00	1
2	Tarun	Singh	2000-09-20	tarun.singh@gmail.com	9823454523	Charbagh	102	savings	25000.00	2
3	Sridhar	Mishra	1998-12-10	sridhar.mishra@gmail.com	9345324567	Indira nagar	103	zero balance	1000.00	3
4	Raghav	Aggarwal	1994-01-11	raghav.aggarwal@gmail.com	8834234576	Munshi puliya	104	current	20000.00	4
5	Abhishek	Verma	1994-07-17	abhishek.verma@gmail.com	7052218976	Shaeed path	105	savings	75000.00	5
6	Paras	Gupta	1993-11-03	paras.gupta@gmail.com	7275385128	Polytechnic road	106	zero balance	0.00	6
7	Aman	Yadav	2001-10-10	aman.yadav@gmail.com	9453678782	Hazratganj	107	current	42500.00	7
8	Shivansh	Singh	2002-01-27	shivansh.singh@gmail.com	7633221456	Aminabad	108	savings	33000.00	8
9	Rishabh	Dixit	2001-02-07	rishabh.dixit@gmail.com	9867563423	Qaiserbagh	109	zero balance	0.00	9
10	Mitali	Srivastava	1996-04-05	mitali.srivastava@gmail.com	8734234565	Nawabganj	110	current	82000.00	10

Result 29

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

```
146 • SELECT t.*, c.*
147 FROM Transactions t
148 JOIN Accounts a ON t.account_id = a.account_id
149 JOIN Customers c ON a.customer_id = c.customer_id
150 WHERE t.account_id = 105;
151
```

100% 26:150

Result Grid Filter Rows: Search Export:

transaction...	transaction_ty...	amount	transaction_date	account_id	customer_id	first_name	last_name	DOB	email	phone_number	address
1005	deposit	7000.00	2023-05-20 00:00:00	105	5	Abhishek	Verma	1994-07-17	abhishek.verma@gmail.com	7052218976	Shaeed path

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

```
152 • SELECT customer_id
153 FROM Accounts
154 GROUP BY customer_id
155 HAVING COUNT(*) > 1;
156
157
```

100% 1:156

Result Grid Filter Rows: Search Export:

customer_id

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

```
157 • SELECT account_id,
158 SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) -
159 SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS difference
160 FROM Transactions
161 GROUP BY account_id;
162
```

100% 21:161

Result Grid Filter Rows: Search Export:

account_id	difference
101	1000.00
102	-500.00
103	3000.00
104	-6000.00
105	7000.00
106	9000.00
107	1000.00
108	-10000.00
109	5500.00
110	-1000.00

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

```
163 • SELECT account_id, AVG(balance) AS average_daily_balance
164 FROM Accounts
165 JOIN Transactions ON Accounts.account_id = Transactions.account_id
166 WHERE transaction_date BETWEEN '2023-01-10' AND '2023-12-31'
167 GROUP BY account_id;
168
```

11. Calculate the total balance for each account type.

```
169 • SELECT account_type, SUM(balance) AS total_balance
170 FROM Accounts
171 GROUP BY account_type;
172
```

100% 23:171 1 error found

Result Grid Filter Rows: Search Export:

account_ty...	total_balan...
current	194500.00
savings	133000.00
zero balance	1000.00

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

```
169 • SELECT account_type, SUM(balance) AS total_balance
170 FROM Accounts
171 GROUP BY account_type;
172
173 • SELECT account_id, COUNT(*) AS transaction_count
174 FROM Transactions
175 GROUP BY account_id
176 ORDER BY transaction_count DESC;
```

100%

33:176

1 error found

Result Grid

Filter Rows:

Search

Export:

	account_id	transaction_co...
▶ 101	1	
▶ 102	1	
▶ 103	1	
▶ 104	1	
▶ 105	1	
▶ 106	1	
▶ 107	1	
▶ 108	1	
▶ 109	1	
▶ 110	1	

Result Grid

Form Editor

Field Types

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

```
178 • SELECT c.customer_id, c.first_name, c.last_name, a.account_type, SUM(a.balance) AS total_balance
179 FROM Customers c
180 JOIN Accounts a ON c.customer_id = a.customer_id
181 GROUP BY c.customer_id, a.account_type
182 HAVING total_balance > 30000;
```

100% 30:182 1 error found

Result Grid Filter Rows: Search Export:

	customer_id	first_name	last_name	account_ty...	total_balan...
▶ 1	Ajay	Chaudhary	current	50000.00	
5	Abhishek	Verma	savings	75000.00	
7	Aman	Yadav	current	42500.00	
8	Shivansh	Singh	savings	33000.00	
10	Mitali	Srivastava	current	82000.00	

Result Grid Form Editor

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

```
184 SELECT *
185 FROM Transactions
186 WHERE (account_id,transaction_type,amount,transaction_date) IN (
187 SELECT account_id, transaction_type,amount,transaction_date
188 FROM Transactions
189 GROUP BY account_id,transaction_type,amount,transaction_date
190 HAVING COUNT(*)>1
191 );
```

192
103

100%

3:191

1 error found

Result Grid

Filter Rows:

Search

Edit:

Export/Import:

transaction...

transaction_ty...

amount

transaction_d...

account_id

▶

NULL

NULL

NULL

NULL

Transactions 37

Apply

Action Output

Time

Action

Response

Duration / Fetch Time

394

10:43:57

SELECT * FROM Transactions WHERE (account_id,transaction_type,amount,transaction_...

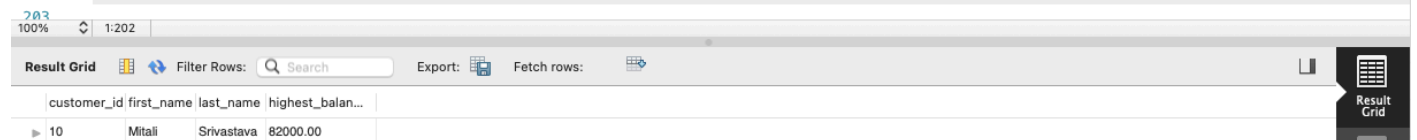
0 row(s) returned

0.0068 sec / 0.00001...

Tasks 4: Subquery and its type:

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

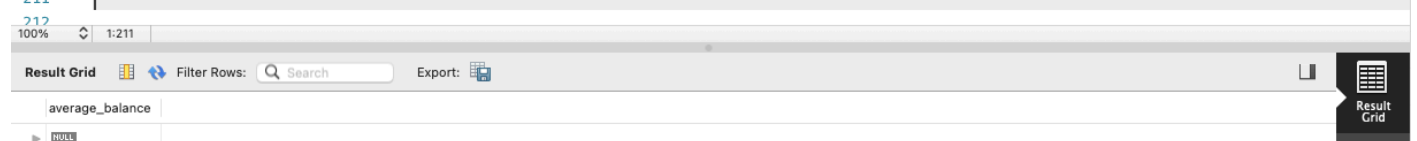
```
196 • SELECT C.customer_id, C.first_name, C.last_name, MAX(A.balance) AS highest_balance
197 FROM Customers C
198 JOIN Accounts A ON C.customer_id = A.customer_id
199 GROUP BY C.customer_id, C.first_name, C.last_name
200 ORDER BY highest_balance DESC
201 LIMIT 1;
202
```



customer_id	first_name	last_name	highest_balance
10	Mitali	Srivastava	82000.00

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

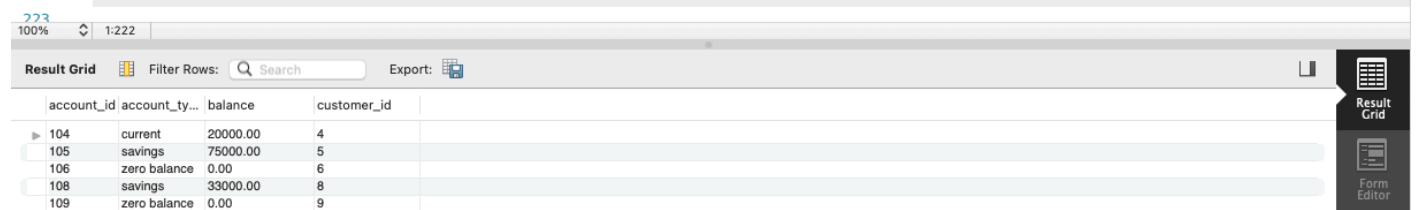
```
203 • SELECT AVG(sub.avg_balance) AS average_balance
204 FROM (
205     SELECT C.customer_id, COUNT(A.account_id) AS num_accounts, AVG(A.balance) AS avg_balance
206     FROM Customers C
207     JOIN Accounts A ON C.customer_id = A.customer_id
208     GROUP BY C.customer_id
209     HAVING num_accounts > 1
210 ) sub;
211
```



average_balance
NULL

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

```
212 • WITH AverageTransaction AS (
213     SELECT AVG(amount) AS avg_transaction_amount
214     FROM Transactions
215 )
216
217 SELECT A.*
218 FROM Accounts A
219 JOIN Transactions T ON A.account_id = T.account_id
220 CROSS JOIN AverageTransaction AT
221 WHERE T.amount > AT.avg_transaction_amount;
222
```



account_id	account_ty...	balance	customer_id
104	current	20000.00	4
105	savings	75000.00	5
106	zero balance	0.00	6
108	savings	33000.00	8
109	zero balance	0.00	9

4. Identify customers who have no recorded transactions.

```
224 • SELECT C.*
225 FROM Customers C
226 LEFT JOIN Accounts A ON C.customer_id = A.customer_id
227 LEFT JOIN Transactions T ON A.account_id = T.account_id
228 WHERE T.transaction_id IS NULL;
```

100% 1:230

Result Grid Filter Rows: Search Export:

customer_id	first_name	last_name	DOB	email	phone_number	address
-------------	------------	-----------	-----	-------	--------------	---------

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

```
230 • SELECT COALESCE(SUM(A.balance), 0) AS total_balance
231 FROM Accounts A
232 LEFT JOIN Transactions T ON A.account_id = T.account_id
233 WHERE T.transaction_id IS NULL;
```

100% 32:233

Result Grid Filter Rows: Search Export:

total_balance
0.00

6. Retrieve transactions for accounts with the lowest balance.

```
242 WITH AccountsWithLowestBalance AS (
243     SELECT account_id, balance
244     FROM Accounts
245     WHERE balance = (SELECT MIN(balance) FROM Accounts)
246 )
247 SELECT T.*
248 FROM Transactions T
249 JOIN AccountsWithLowestBalance ALB ON T.account_id = ALB.account_id;
```

100% 1:253

Result Grid Filter Rows: Search Export:

transaction...	transaction_ty...	amount	transaction_date	account_id
1006	deposit	9000.00	2023-06-19 00:00:00	106
1009	deposit	5500.00	2023-09-27 00:00:00	109

7. Identify customers who have accounts of multiple types.

```
259 • SELECT C.customer_id, C.first_name, C.last_name
260 FROM Customers C
261 JOIN Accounts A ON C.customer_id = A.customer_id
262 GROUP BY C.customer_id, C.first_name, C.last_name
263 HAVING COUNT(DISTINCT A.account_type) > 1;
```

100% 43:263

Result Grid Filter Rows: Search Export:

customer_id	first_name	last_name
-------------	------------	-----------

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

```
265 • SELECT
266     account_type,
267     COUNT(*) AS account_count,
268     (COUNT(*) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS percentage
269 FROM
270     Accounts
271 GROUP BY
272     account_type;
273
```

100% 1:273

Result Grid Filter Rows: Search Export:

account_ty...	account_cou...	percentage
current	4	40.00000
savings	3	30.00000
zero balance	3	30.00000

Result Grid

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

```
275 • SELECT T.*
276 FROM Transactions T
277 JOIN Accounts A ON T.account_id = A.account_id
278 JOIN Customers C ON A.customer_id = C.customer_id
279 WHERE C.customer_id = 1;
```

100% 25:279

Result Grid Filter Rows: Search Export:

transaction...	transaction_ty...	amount	transaction_date	account_id
1001	deposit	1000.00	2023-01-10 00:00:00	101

Result Grid

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

```
275 • SELECT T.*
276 FROM Transactions T
277 JOIN Accounts A ON T.account_id = A.account_id
278 JOIN Customers C ON A.customer_id = C.customer_id
279 WHERE C.customer_id = 1;
```

100% 25:279

Result Grid Filter Rows: Search Export:

transaction...	transaction_ty...	amount	transaction_date	account_id
1001	deposit	1000.00	2023-01-10 00:00:00	101

Result Grid