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

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

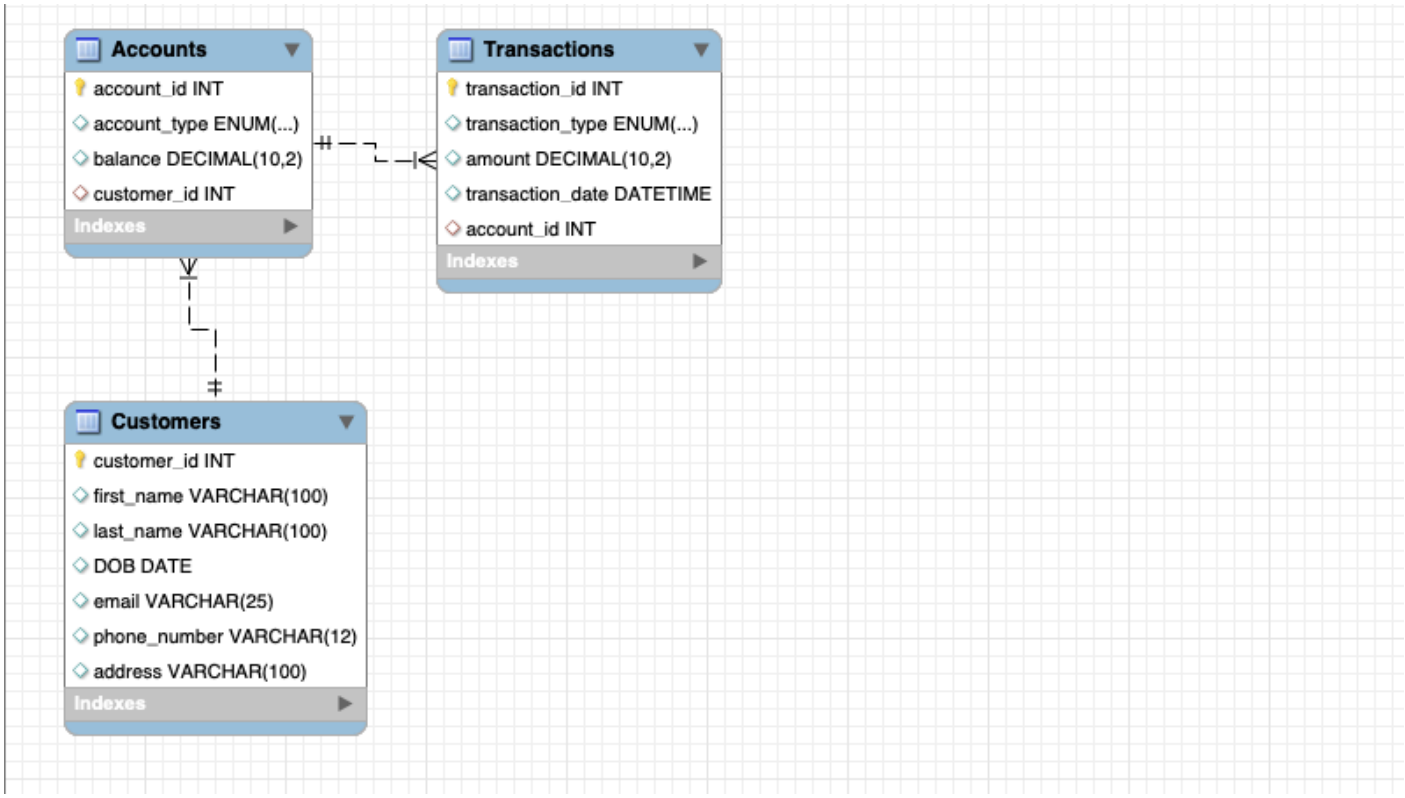
```sql
CREATE Database if not exists HMBank;
use HMBank;

CREATE table if not exists Customers(
customer_id INT PRIMARY KEY,
first_name VARCHAR(100),
last_name VARCHAR(100),
DOB DATE,
email VARCHAR(25),
phone_number VARCHAR(12),
address VARCHAR(100)
);

CREATE table if not exists Accounts(
account_id INT PRIMARY KEY,
account_type ENUM('savings', 'current' , 'zero balance'),
balance DECIMAL (10,2),
customer_id int ,FOREIGN KEY(customer_id) REFERENCES Customers(customer_id)
);
```

```sql
CREATE table if not exists Transactions(
transaction_id INT PRIMARY KEY,
transaction_type ENUM('deposit', 'withdrawal', 'transfer'),
amount DECIMAL(10,2),
transaction_date DATETIME,
account_id int ,FOREIGN KEY(account_id) REFERENCES Accounts(account_id)
);
```

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

```sql
INSERT INTO Customers VALUES(1, 'Ajay', 'Chaudhary', '2000-11-24', 'ajay.chaudhary@gmail.com', '63678765
(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.srivas@gmail.com', '8734234565', 'Nawabganj');
select * from Customers;

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.srivas@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...

Object Info | Session

No object selected

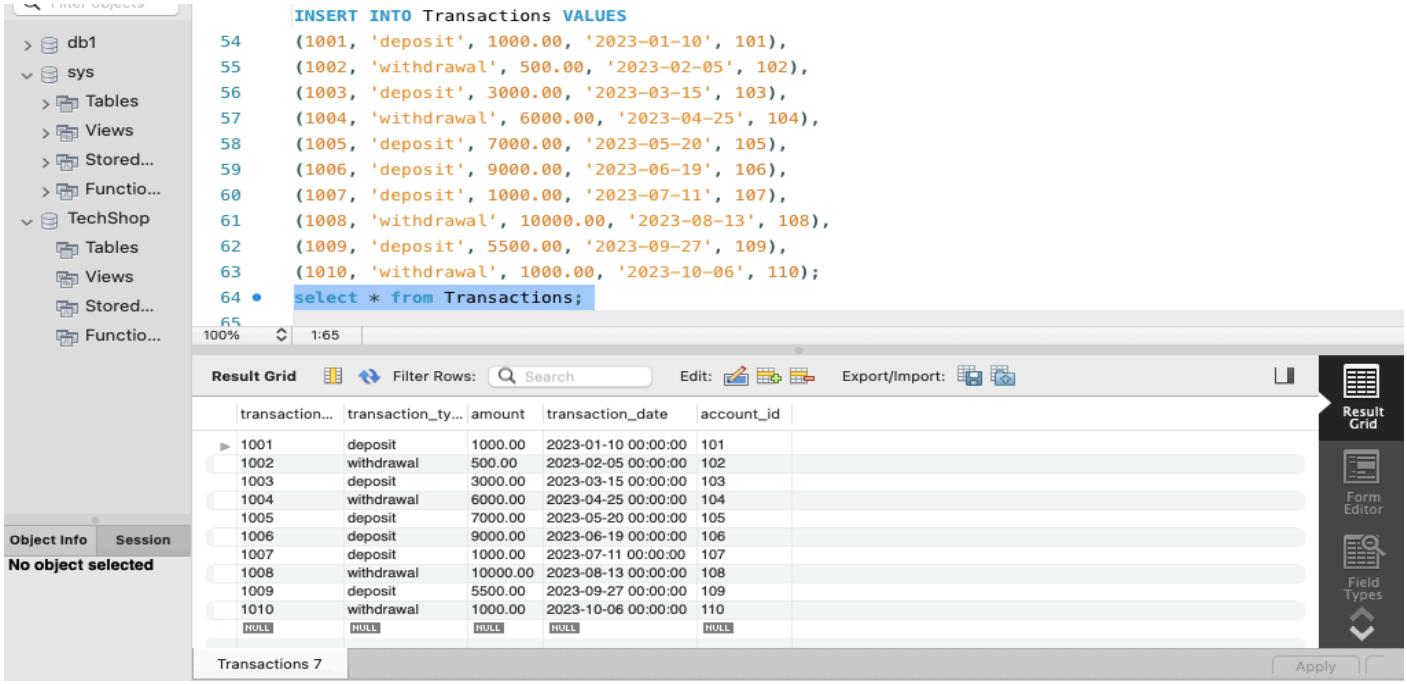
```sql
INSERT INTO Accounts VALUES(101, 'current', 50000.00, 1),
(102, 'savings', 25000.00, 2),
(103, 'zero balance', 0.00, 3),
(104, 'current', 20000.00,4),
(105, 'savings', 75000.00, 5),
(106, 'zero balance', 0.00, 6),
(107, 'current', 42500.00, 7),
(108, 'savings', 33000.00, 8),
(109, 'zero balance', 0.00, 9),
(110, 'current', 82000.00, 10);
select * from Accounts;
```

100% 1:52

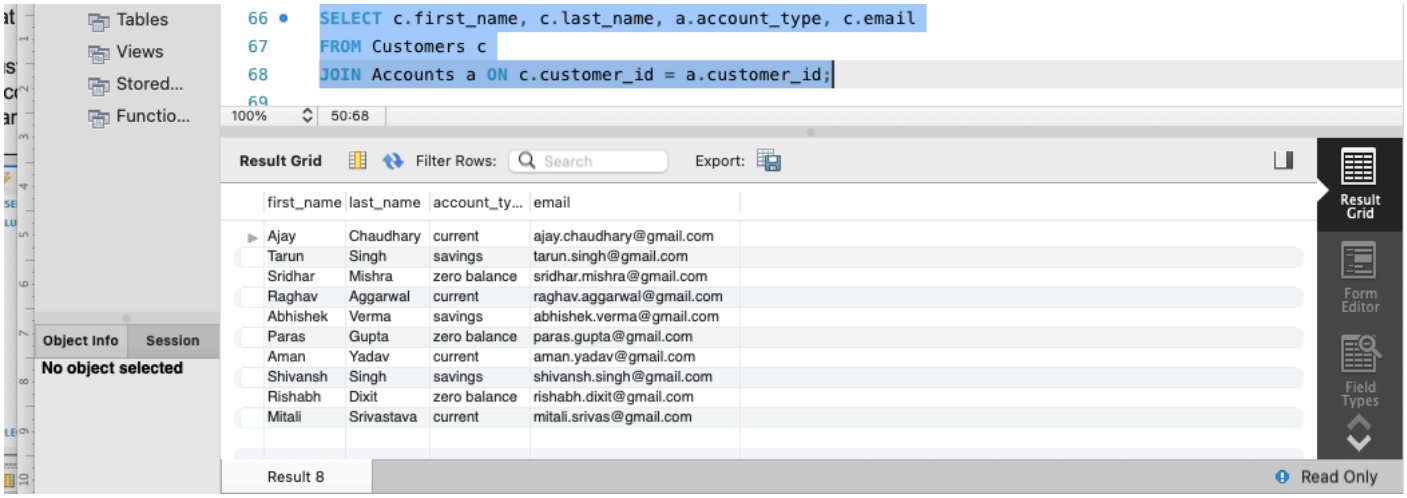
Result Grid | Filter Rows: Search | Edit: | Export/Import:

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

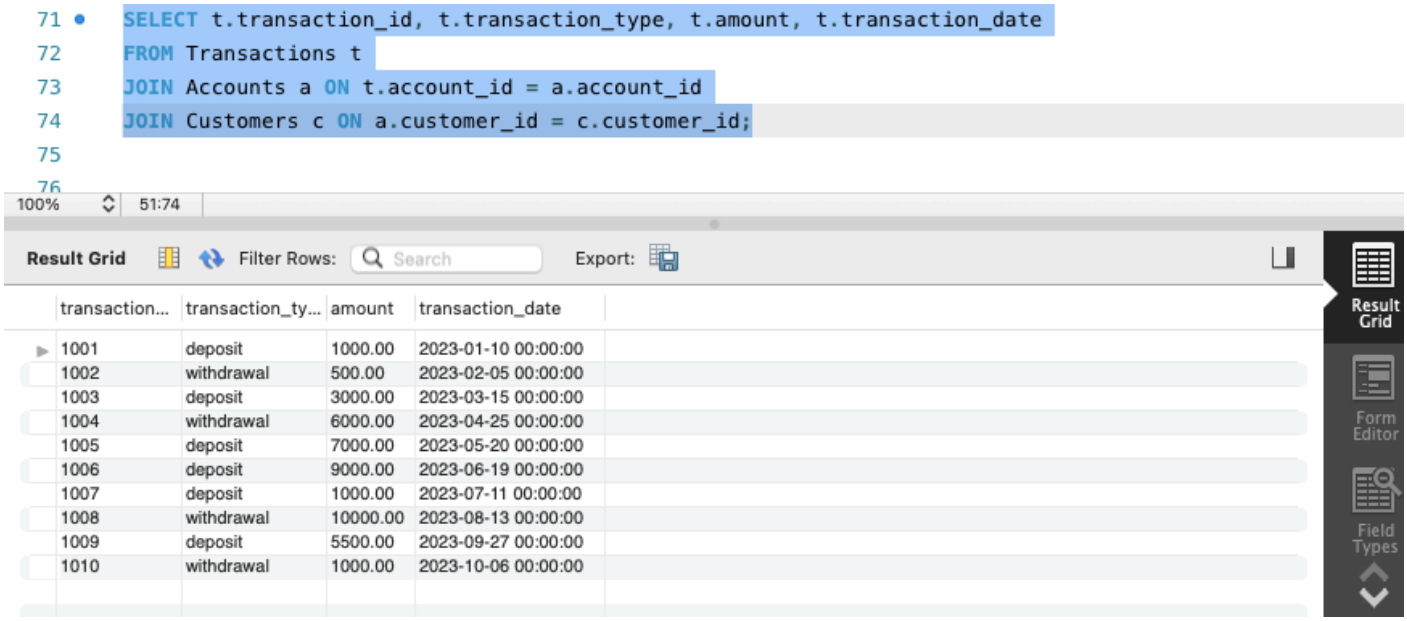
Result Grid | Form Editor | Field Types



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



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



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

Result Grid | Form Editor | Field Types

Accounts 11 Apply

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

TechShop
Tables
Views
Stored...
Functio...

```
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 Grid | Form Editor | Field Types

Object Info Session
No object selected

Result 12 Read Only

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

TechShop
Tables
Views
Stored...
Functio...

```
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:

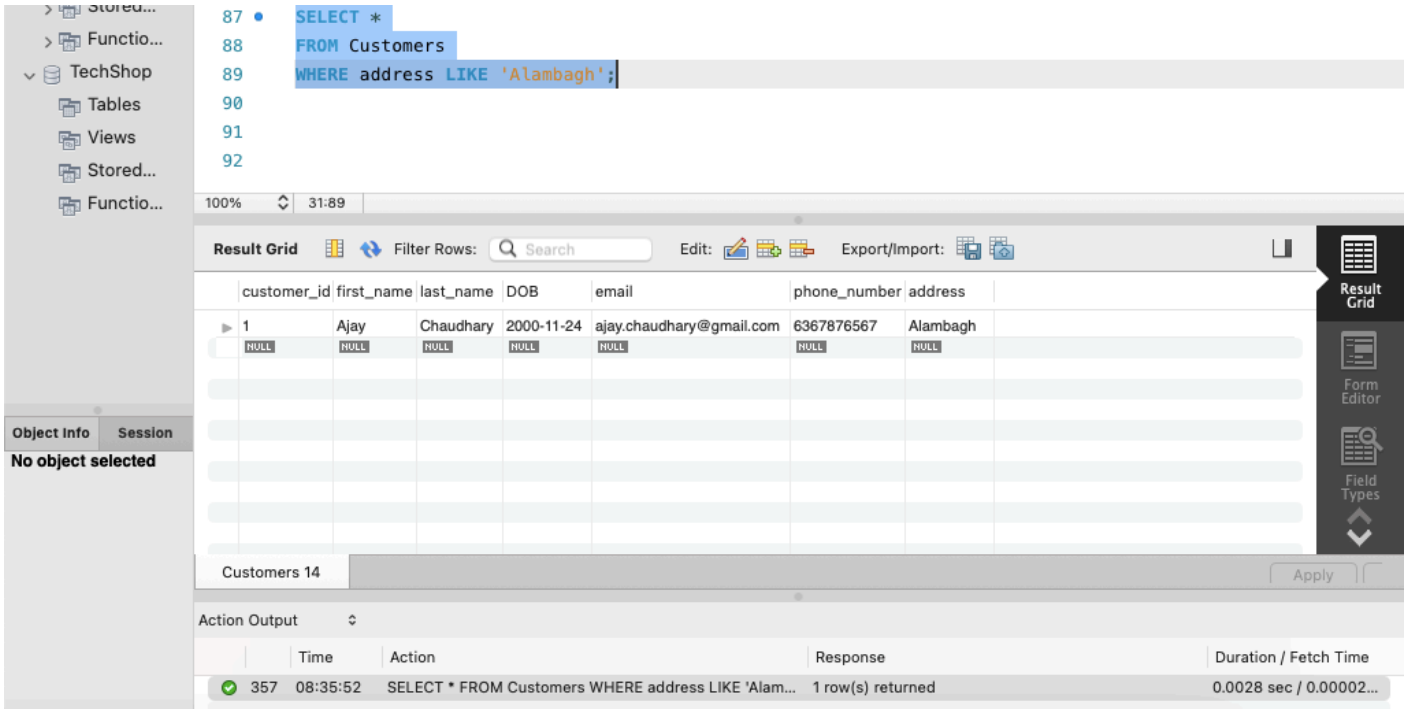
account_id	account_ty...	balance	customer_id
101	current	50000.00	1
102	savings	25000.00	2
103	zero balance	1000.00	3
104	current	20000.00	4
105	savings	75000.00	5
106	zero balance	0.00	6
107	current	42500.00	7
108	savings	33000.00	8
109	zero balance	0.00	9
110	current	82000.00	10
NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

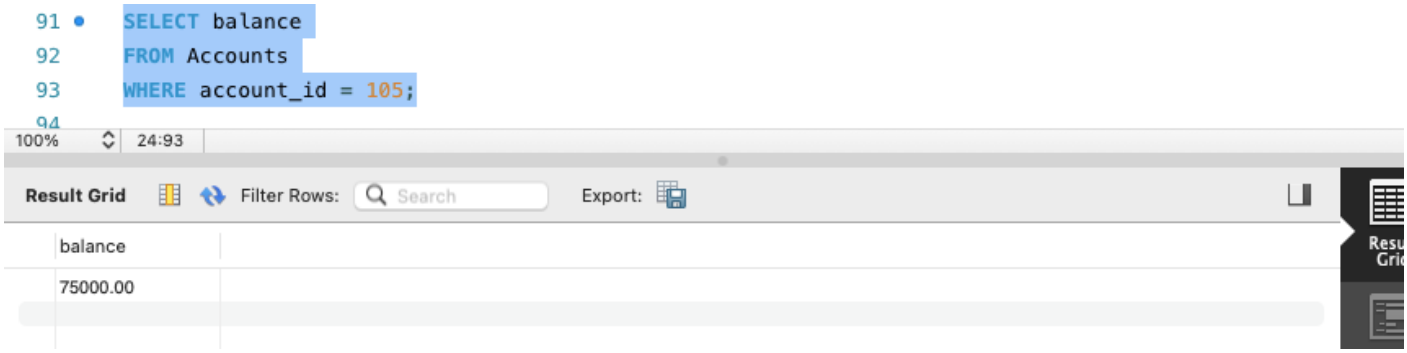
Object Info Session
No object selected

Accounts 13 Apply

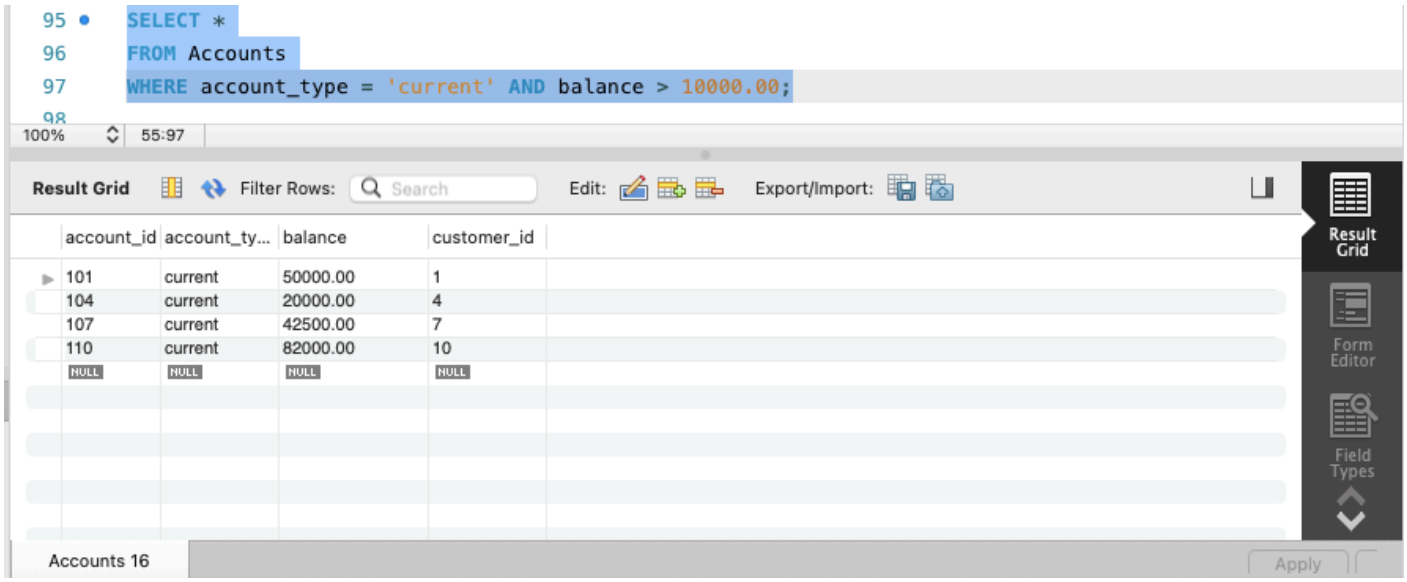
7. Write a SQL query to Find customers living in a specific city.



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



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



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

```
SELECT *
FROM Transactions
WHERE account_id = 107;
```

Result Grid

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

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

```
SELECT account_id, balance * 1.5 / 100 AS interest_accrued
FROM Accounts
WHERE account_type = 'savings';
```

Result Grid

account_id	interest_accru...
102	375.0000000
105	1125.0000000
108	495.0000000

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

```
SELECT *
FROM Accounts
WHERE balance < 50000;
```

Result Grid

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

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...

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

```
SELECT *
FROM Customers
WHERE address NOT LIKE 'Alambagh';
```

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.srivas@gmail.com	8734234565	Nawabganj

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

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

```
SELECT AVG(balance) AS average_balance
FROM Accounts;
```

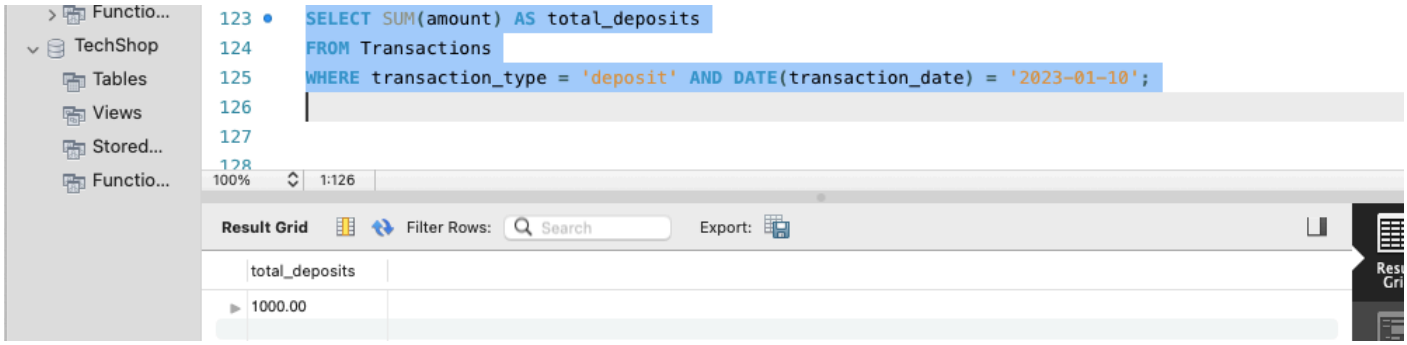
average_balance
32850.000000

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

```
SELECT *
FROM Accounts
ORDER BY balance DESC
LIMIT 10;
```

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

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



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

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

100% 8:130

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

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

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

100% 1:136

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

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

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

```
SELECT t.*, a.account_type
FROM Transactions t
JOIN Accounts a ON t.account_id = a.account_id;
```

Result Grid

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.

```
SELECT c.*, a.*
FROM Customers c
LEFT JOIN Accounts a ON c.customer_id = a.customer_id;
```

Result Grid

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.srivas@gmail.com	8734234565	Nawabganj	110	current	82000.00	10

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

```
SELECT t.*, c.*
FROM Transactions t
JOIN Accounts a ON t.account_id = a.account_id
JOIN Customers c ON a.customer_id = c.customer_id
WHERE t.account_id = 105;
```

Result Grid

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.

```sql
SELECT customer_id
FROM Accounts
GROUP BY customer_id
HAVING COUNT(*) > 1;
```

Result Grid

customer_id

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

```sql
SELECT 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 difference
FROM Transactions
GROUP BY account_id;
```

Result Grid

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.

```sql
SELECT account_id, AVG(balance) AS average_daily_balance
FROM Accounts
JOIN Transactions ON Accounts.account_id = Transactions.account_id
WHERE transaction_date BETWEEN '2023-01-10' AND '2023-12-31'
GROUP BY account_id;
```

11. Calculate the total balance for each account type.

```sql
SELECT account_type, SUM(balance) AS total_balance
FROM Accounts
GROUP BY account_type;
```

Result Grid

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.

```
SELECT account_type, SUM(balance) AS total_balance
FROM Accounts
GROUP BY account_type;

SELECT account_id, COUNT(*) AS transaction_count
FROM Transactions
GROUP BY account_id
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

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

```
SELECT c.customer_id, c.first_name, c.last_name, 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, a.account_type
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

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

```
SELECT *
FROM Transactions
WHERE (account_id,transaction_type,amount,transaction_date) IN (
SELECT account_id, transaction_type,amount,transaction_date
FROM Transactions
GROUP BY account_id,transaction_type,amount,transaction_date
HAVING COUNT(*)>1
);
```

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	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.

```
SELECT C.customer_id, C.first_name, C.last_name, MAX(A.balance) AS highest_balance
FROM Customers C
JOIN Accounts A ON C.customer_id = A.customer_id
GROUP BY C.customer_id, C.first_name, C.last_name
ORDER BY highest_balance DESC
LIMIT 1;
```

Result Grid

customer_id	first_name	last_name	highest_balan...
10	Mitali	Srivastava	82000.00

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

```
SELECT AVG(sub.avg_balance) AS average_balance
FROM (
    SELECT C.customer_id, COUNT(A.account_id) AS num_accounts, AVG(A.balance) AS avg_balance
    FROM Customers C
    JOIN Accounts A ON C.customer_id = A.customer_id
    GROUP BY C.customer_id
    HAVING num_accounts > 1
) sub;
```

Result Grid

average_balance
NULL

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

```
WITH AverageTransaction AS (
    SELECT AVG(amount) AS avg_transaction_amount
    FROM Transactions
)
SELECT A.*
FROM Accounts A
JOIN Transactions T ON A.account_id = T.account_id
CROSS JOIN AverageTransaction AT
WHERE T.amount > AT.avg_transaction_amount;
```

Result Grid

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;
229
```

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;
234
```

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
248    SELECT T.*
249    FROM Transactions T
250    JOIN AccountsWithLowestBalance ALB ON T.account_id = ALB.account_id;
251
252
253
```

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;
264
```

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
274
```

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

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;
280
```

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

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;
280
```

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