

TASK 1: Database Design:

Q1. Create the database named "HMBank"

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Query 1'. The code in the editor is as follows:

```
396
397 • UPDATE Payments
398     SET amount = amount + 80.00
399     WHERE student_id = 9;
400
401 • use codingchallenges;
402
403 • create database HMBank;
404 • use HMBank;
405
406 • create table Customers(
407     customer_id int primary key,
408     first_name text,
409     last_name text,
410     DOB date,
411     email text,
412     phone bigint
413 );
```

The 'Output' pane below the editor shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
1	17:08:09	create database HMBank	1 row(s) affected	0.047 sec
2	17:08:27	use HMBank	0 row(s) affected	0.015 sec

Q2. Define the schema for the Customers, Accounts, and Transactions tables based on the provided schema.

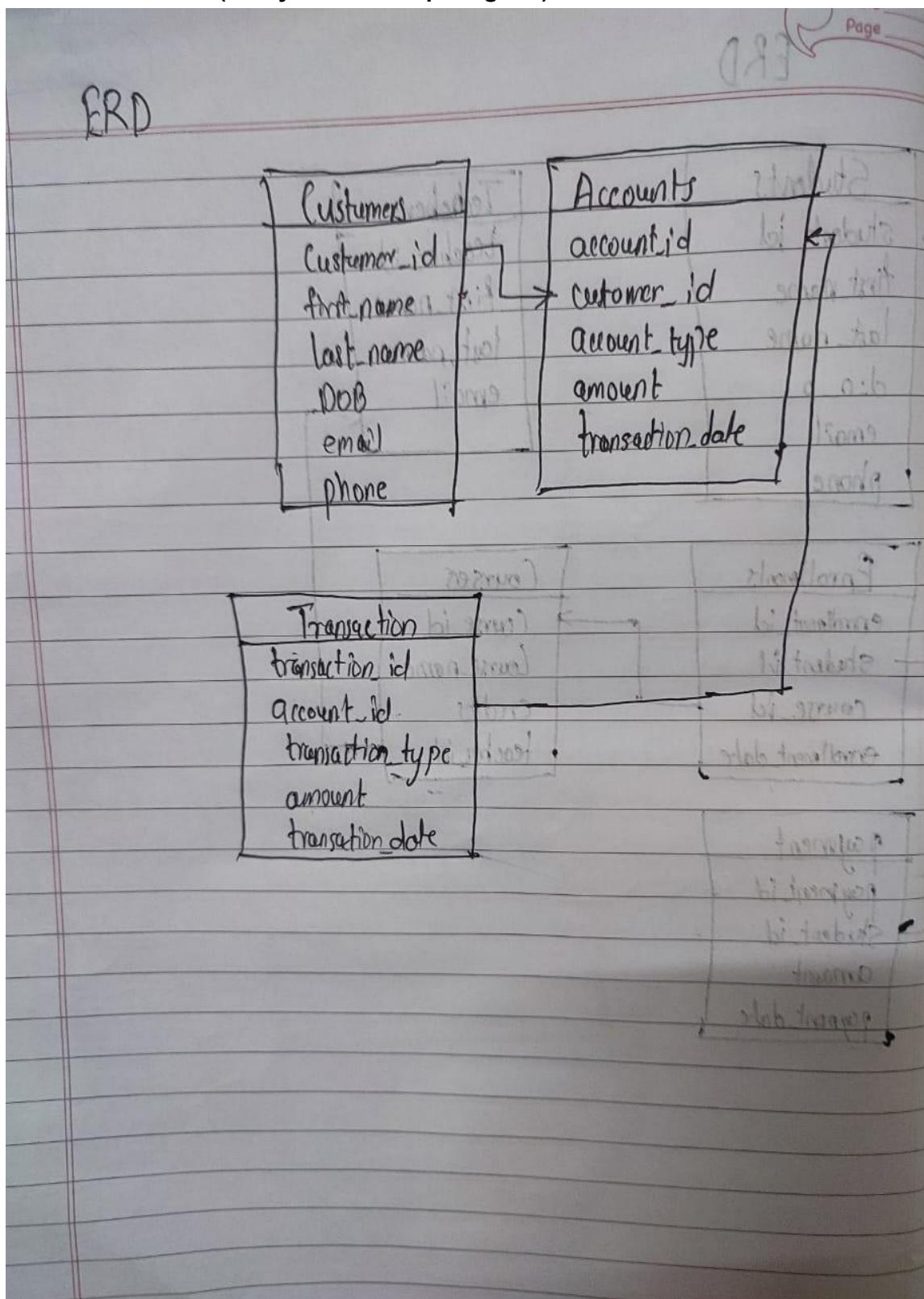
The screenshot shows the MySQL Workbench interface with a query editor window titled 'Query 1'. The code in the editor is as follows:

```
418     amount double,
419     transaction_date date,
420     foreign key (customer_id)references Customers(customer_id)
421 );
422 • alter table Accounts RENAME Accounts;
423 • create table Transactions(
424     transaction_id int primary key,
425     account_id int,
426     transaction_type enum('deposit','withdrawal','transfer'),
427     amount double,
428     transaction_date date,
429     foreign key(account_id)references Accounts(account_id)
430 );
431
432
433
434
```

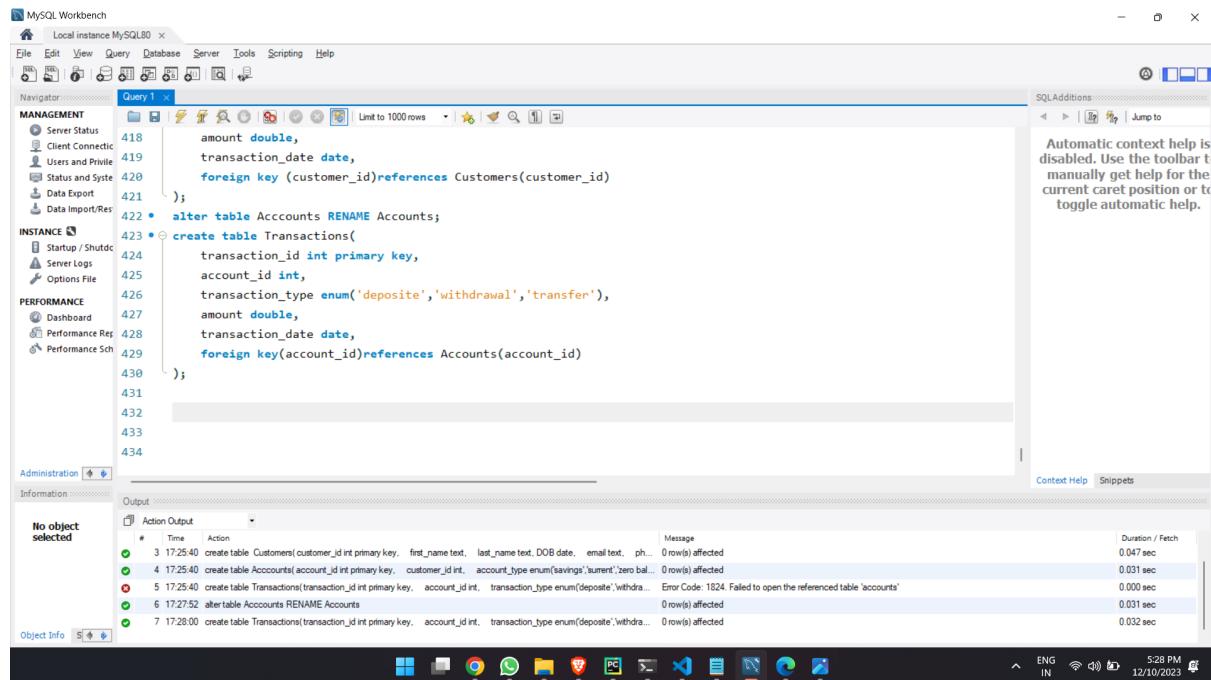
The 'Output' pane below the editor shows the results of the executed statements:

#	Time	Action	Message	Duration / Fetch
3	17:25:40	create table Customers(customer_id int primary key, first_name text, last_name text, DOB date, email text, ph...	0 row(s) affected	0.047 sec
4	17:25:40	create table Accounts(account_id int primary key, customer_id int, account_type enum('saving','current'), zero bal...	0 row(s) affected	0.031 sec
5	17:25:40	create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdr...	Error Code: 1824. Failed to open the referenced table 'accounts'	0.000 sec
6	17:27:52	alter table Accounts RENAME Accounts	0 row(s) affected	0.031 sec
7	17:28:00	create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdr...	0 row(s) affected	0.032 sec

Q4. Create an ERD (Entity Relationship Diagram) for the database.



Q5. Create appropriate Primary Key and Foreign Key constraints for referential integrity.



```

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status 418
    Client Connectic 419
    Users and Privilic 420
    Status and Syste 421
    Data Export 422
    Data Import/Res 423
INSTANCE
    Startup / Shuttic 424
    Server Logs 425
    Options File 426
PERFORMANCE
    Dashboard 427
    Performance Rep 428
    Performance Sch 429
Administration
Information
No object selected
Output
Action Output
# Time Action Message Duration / Fetch
1 17:25:40 create table Customers(customer_id int primary key, first_name text, last_name text, DOB date, email text, ph... 0 row(s) affected 0.047 sec
2 17:25:40 create table Accounts(account_id int primary key, customer_id int, account_type enum('savings','current','zero bal... 0 row(s) affected 0.031 sec
3 17:25:40 create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdrawal','transfer'), amount double, transaction_date date, foreign key(account_id)references Accounts(account_id) 0 row(s) affected 0.000 sec
4 17:27:52 alter table Accounts RENAME Accounts 0 row(s) affected 0.031 sec
5 17:28:00 create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdrawal','transfer'), amount double, transaction_date date, foreign key(account_id)references Accounts(account_id) 0 row(s) affected 0.032 sec
Object Info S

```

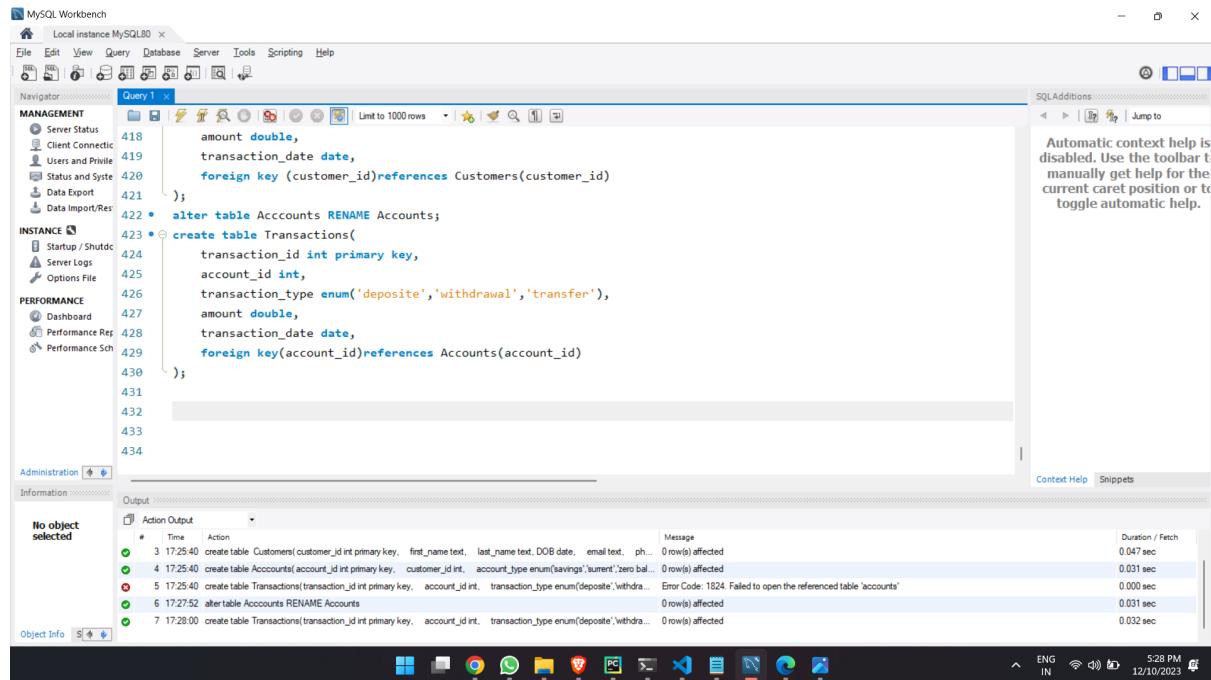
The screenshot shows the MySQL Workbench interface with a query editor containing SQL code to create three tables: Customers, Accounts, and Transactions. The Customers table has a primary key on customer_id and a foreign key on account_id referencing the Accounts table. The Accounts table has a primary key on account_id. The Transactions table has a primary key on transaction_id, a foreign key on account_id referencing the Accounts table, and a foreign key on customer_id referencing the Customers table. The output pane shows the execution results with 0 rows affected for each statement, indicating successful creation of the tables.

Q6. Write SQL scripts to create the mentioned tables with appropriate data types, constraints, and relationships.

-Customers

-Accounts

-Transactions



```

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
MANAGEMENT
    Server Status 418
    Client Connectic 419
    Users and Privilic 420
    Status and Syste 421
    Data Export 422
    Data Import/Res 423
INSTANCE
    Startup / Shuttic 424
    Server Logs 425
    Options File 426
PERFORMANCE
    Dashboard 427
    Performance Rep 428
    Performance Sch 429
Administration
Information
No object selected
Output
Action Output
# Time Action Message Duration / Fetch
1 17:25:40 create table Customers(customer_id int primary key, first_name text, last_name text, DOB date, email text, ph... 0 row(s) affected 0.047 sec
2 17:25:40 create table Accounts(account_id int primary key, customer_id int, account_type enum('savings','current','zero bal... 0 row(s) affected 0.031 sec
3 17:25:40 create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdrawal','transfer'), amount double, transaction_date date, foreign key(account_id)references Accounts(account_id) 0 row(s) affected 0.000 sec
4 17:27:52 alter table Accounts RENAME Accounts 0 row(s) affected 0.031 sec
5 17:28:00 create table Transactions(transaction_id int primary key, account_id int, transaction_type enum('deposit','withdrawal','transfer'), amount double, transaction_date date, foreign key(account_id)references Accounts(account_id) 0 row(s) affected 0.032 sec
Object Info S

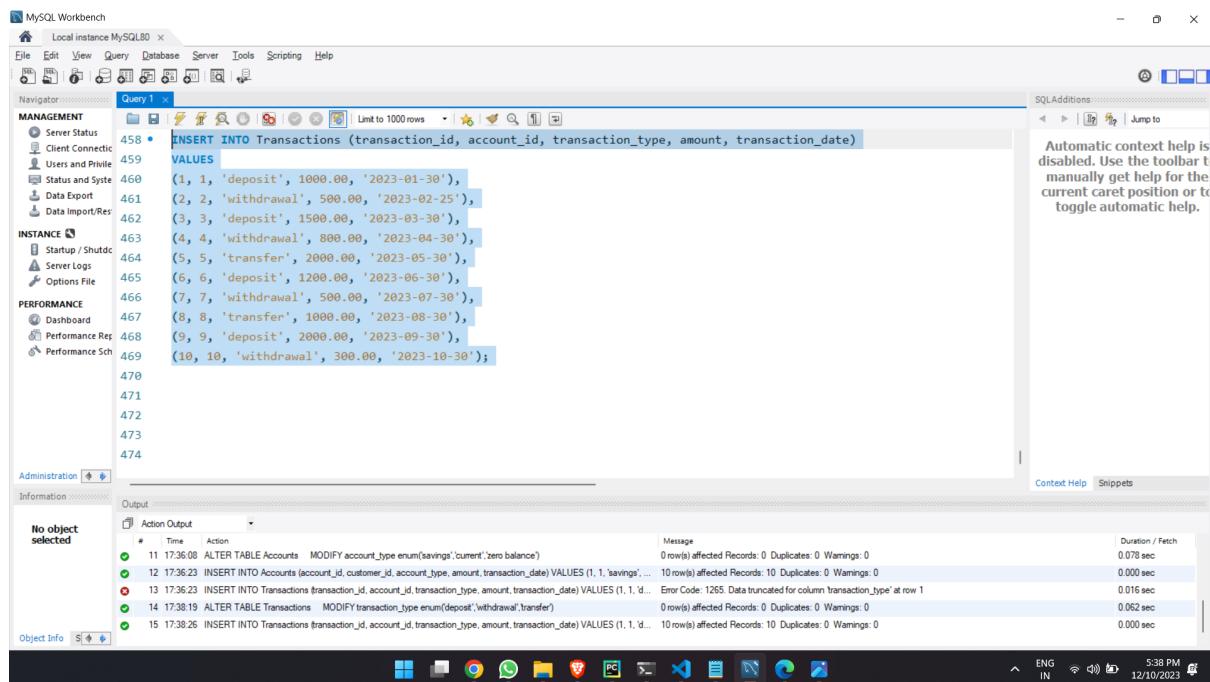
```

This screenshot is identical to the one above, showing the MySQL Workbench interface with the same SQL code for creating the Customers, Accounts, and Transactions tables. The output pane shows the execution results with 0 rows affected for each statement, indicating successful creation of the tables.

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

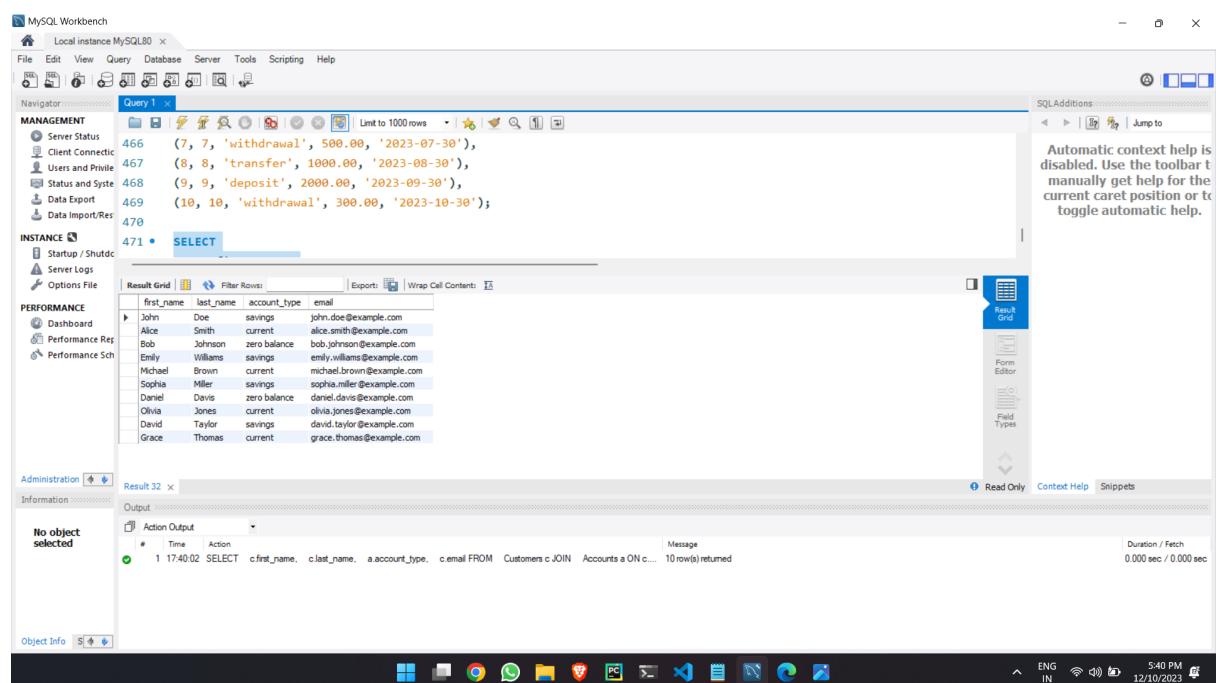
Q1. Insert at least 10 sample records into each of the following tables.

- Customers
 - Accounts
 - Transactions



Q2. Write SQL queries for the following tasks:

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



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

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
480
481 •  SELECT
482     c.first_name,
483     c.last_name,
484     t.transaction_id,
485     t.transaction_type,
486     t.amount
487
```

The results grid displays the following data:

first_name	last_name	transaction_id	transaction_type	amount	transaction_date
John	Doe	1	deposit	1000	2023-01-30
Alice	Smith	2	withdrawal	500	2023-02-25
Bob	Johnson	3	deposit	1500	2023-03-30
Emily	Williams	4	withdrawal	800	2023-04-30
Michael	Brown	5	transfer	200	2023-05-30
Sophia	Miller	6	deposit	1200	2023-06-30
Daniel	Davis	7	withdrawal	500	2023-07-30
Olivia	Jones	8	transfer	1000	2023-08-30
David	Taylor	9	deposit	2000	2023-09-30
Grace	Thomas	10	withdrawal	300	2023-10-30

The status bar at the bottom right shows: ENG IN 5:42 PM 12/10/2023.

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

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
483
484
485
486
487
488 FROM
489     Customers c
490 JOIN
491     Accounts a ON c.customer_id = a.customer_id
492 JOIN
493     Transactions t ON a.account_id = t.account_id;
494
495 • UPDATE Accounts
496 SET amount = amount + 500.00
497 WHERE account_id = 3;
498
499
500
```

The results grid shows the message from the output pane:

Message
1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

The status bar at the bottom right shows: ENG IN 5:52 PM 12/10/2023.

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

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
Transactions t ON a.account_id = t.account_id;
495 • UPDATE Accounts
496     SET amount = amount + 500.00
497 WHERE account_id = 3;
498
499 • SELECT
```
- Result Grid:** Shows the output of the SELECT query:

full_name
John Doe
Alice Smith
Bob Johnson
Emily Williams
Michael Brown
Sophia Miller
Daniel Davis
Olivia Jones
David Taylor
Grace Thomas
- Output Window:** Displays the execution log:

Action Output	#	Time	Action	Message	Duration / Fetch
1	17:52:55	UPDATE	Accounts SET amount = amount + 500.00 WHERE account_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.047 sec
2	17:53:43	SELECT	CONCAT(first_name, ' ', last_name) AS full_name FROM Customers LIMIT 0, 1000	10 row(s) returned	0.062 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```
Customers c
489 JOIN
490     Accounts a ON c.customer_id = a.customer_id
491
492 JOIN
493     Transactions t ON a.account_id = t.account_id;
494
495 • UPDATE Accounts
496     SET amount = amount + 500.00
497 WHERE account_id = 3;
498
499 • SELECT
500     CONCAT(first_name, ' ', last_name) AS full_name
501     FROM
502     Customers;
503
504 • DELETE FROM Accounts
505 WHERE account_type = 'savings' AND amount = 0.0;
```
- Output Window:** Displays the execution log:

Action Output	#	Time	Action	Message	Duration / Fetch
1	17:55:06	DELETE	FROM Accounts WHERE account_type = 'savings' AND amount = 0.0	0 row(s) affected	0.047 sec

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

According to schema we don't have their address

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

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is entered:

```
511 FROM
512 Accounts
513 WHERE
514 account_id = 5;
```

The Result Grid shows the output of the query:

account_id	account_type	amount
5	current	5000

In the Output pane, the following actions are listed:

#	Time	Action	Message	Duration / Fetch
1	17:55:06	DELETE FROM Accounts WHERE account_type = 'savings' AND amount > 0	0 row(s) affected	0.047 sec
2	17:56:39	SELECT * FROM Customers WHERE city = 'YourCity' LIMIT 0, 1000	Error Code: 1054. Unknown column 'city' in 'where clause'	0.000 sec
3	17:58:44	SELECT account_id, account_type, amount FROM Accounts WHERE account_id = 5 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is entered:

```
510 amount
511 FROM
512 Accounts
513 WHERE
514 account_id = 5;
515
516 • SELECT
```

The Result Grid shows the output of the query:

account_id	account_type	amount
2	current	7500
5	current	5000
8	current	6000
10	current	4000

In the Output pane, the following action is listed:

#	Time	Action	Message	Duration / Fetch
1	18:00:06	SELECT account_id, account_type, amount FROM Accounts WHERE account_type = 'current' AND amount > 1000	4 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
779    account_type;
780
781 • use HMBank;
782
783 • SELECT * from Transactions t
784 JOIN Accounts a ON t.account_id = a.account_id
785 WHERE a.account_id = 2;
786
787
788
```

The result grid shows one row of data:

transaction_id	account_id	transaction_type	amount	transaction_date	account_id	customer_id	account_type	amount	transaction_date
2	2	withdrawal	500	2023-02-25	2	2	current	7500	2023-02-20

The status bar at the bottom right indicates the time is 11:16 AM and the date is 12/12/2023.

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

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

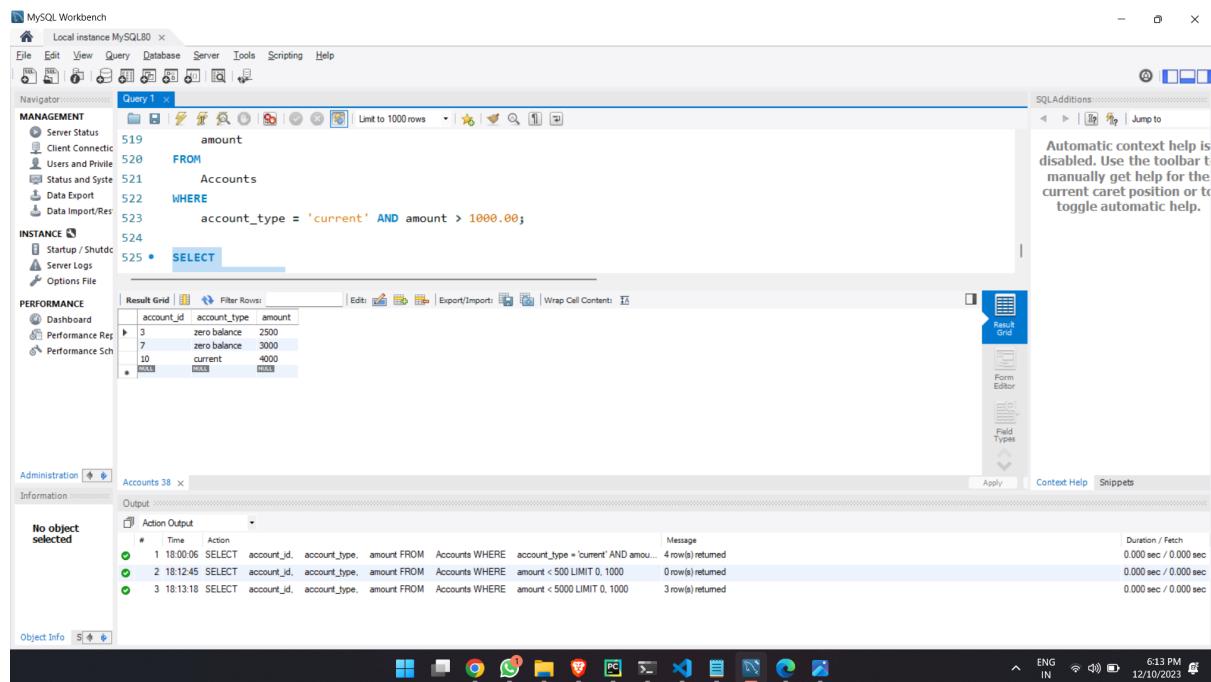
```
785 WHERE a.account_id = 2;
786
787 • SELECT a.account_id, c.first_name, c.last_name, a.amount AS current_balance, a.amount * 0.05 AS interest_accrued
788 FROM Accounts a
789 JOIN Customers c ON a.customer_id = c.customer_id
790 WHERE a.account_type = 'savings';
791
792
793
794
```

The result grid shows five rows of data:

account_id	first_name	last_name	current_balance	interest_accrued
1	John	Doe	5000	250
4	Emily	Williams	10000	500
6	Sophia	Miller	8000	400
9	David	Taylor	12000	600

The status bar at the bottom right indicates the time is 11:18 AM and the date is 12/12/2023.

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



The screenshot shows the MySQL Workbench interface. In the top-left pane, the Navigator displays 'MANAGEMENT' and 'INSTANCE' sections. In the center, the 'Query 1' editor contains the following SQL code:

```
519     amount
520   FROM
521     Accounts
522   WHERE
523     account_type = 'current' AND amount > 1000.00;
524
525 • SELECT
```

The results of the query are displayed in a 'Result Grid' table:

account_id	account_type	amount
3	zero balance	2500
7	zero balance	3000
10	current	4000
10	zero	4000

In the bottom-left pane, the 'Accounts 38' object info shows 'No object selected'. The bottom right corner of the window shows the system tray with icons for battery, signal, and date/time (6:13 PM, 12/10/2023).

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

We Don't Have address or city of any customer according to given schema

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

Q1. Write a SQL query to Find the average account balance for all customers.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
529 FROM
530   Accounts
531 WHERE
532   amount < 5000;
533
534 • SELECT
535   AVG(amount)
```

The result grid shows a single row with the value 6300. Below the query editor is the "Output" pane, which displays the execution log for the session:

#	Time	Action	Message	Duration / Fetch
1	18:00:06	SELECT account_id, account_type, amount FROM Accounts WHERE account_type = 'current' AND amount < 5000	4 row(s) returned	0.000 sec / 0.000 sec
2	18:12:45	SELECT account_id, account_type, amount FROM Accounts WHERE amount < 500 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	18:13:18	SELECT account_id, account_type, amount FROM Accounts WHERE amount < 5000 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
4	18:14:38	SELECT AVG(amount) FROM Accounts LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Q2. Write a SQL query to Retrieve the top 10 highest account balances.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
544 FROM
545   Accounts
546 ORDER BY
547   amount DESC
548 LIMIT 10;
549
550
```

The result grid shows 10 rows of account data. Below the query editor is the "Output" pane, which displays the execution log for the session:

#	Time	Action	Message	Duration / Fetch
1	18:00:06	SELECT account_id, account_type, amount FROM Accounts WHERE account_type = 'current' AND amount < 5000	4 row(s) returned	0.000 sec / 0.000 sec
2	18:12:45	SELECT account_id, account_type, amount FROM Accounts WHERE amount < 500 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
3	18:13:18	SELECT account_id, account_type, amount FROM Accounts WHERE amount < 5000 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
4	18:14:38	SELECT AVG(amount) FROM Accounts LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
5	18:15:42	SELECT account_id, account_type, amount FROM Accounts ORDER BY amount DESC LIMIT 10	10 row(s) returned	0.000 sec / 0.000 sec

Q3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
544 FROM
545     Accounts
546 ORDER BY
547     amount DESC
548 LIMIT 10;
```

Line 549: `550 • SELECT`
- Result Grid:** Shows the output of the query with one row:

total_deposits
1000
- Output Window:** Displays two log entries:

Action Output	Time	Action	Message	Duration / Fetch
1	18:18:44	SELECT	SUM(amount) AS total_deposits FROM Transactions WHERE transaction_type = 'deposit' AND trans...	0.000 sec / 0.000 sec
2	18:19:13	SELECT	SUM(amount) AS total_deposits FROM Transactions WHERE transaction_type = 'deposit' AND trans...	0.000 sec / 0.000 sec

Q4. Write a SQL query to Find the Oldest and Newest Customers.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
552 FROM
553     Transactions
554 WHERE
555     transaction_type = 'deposit'
556     AND transaction_date = '2023-01-30';
557
```

Line 558: `558 • SELECT`
- Result Grid:** Shows the output of the query with one row:

customer_id	first_name	last_name	DOB
2	Alice	Smith	1985-08-22
- Output Window:** Displays one log entry:

Action Output	Time	Action	Message	Duration / Fetch
1	18:20:48	SELECT	customer_id, first_name, last_name, DOB FROM Customers ORDER BY DOB ASC LIMIT 1	1 row(s) returned 0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT, INSTANCE, PERFORMANCE, Administration, Information

Query 1 X

```

553     Transactions
554     WHERE
555         transaction_type = 'deposit'
556         AND transaction_date = '2023-01-30';
557
558
559 • SELECT

```

Result Grid:

customer_id	first_name	last_name	DOB
6	Sophia	Miller	1997-09-18

Output:

#	Time	Action	Message	Duration / Fetch
1	18:20:48	SELECT customer_id, first_name, last_name, DOB FROM Customers ORDER BY DOB ASC LIMIT 1	1 row(s) returned	0.000 sec / 0.000 sec
2	18:21:27	SELECT customer_id, first_name, last_name, DOB FROM Customers ORDER BY DOB DESC LIMIT 1	1 row(s) returned	0.000 sec / 0.000 sec

Object Info: No object selected

System tray: ENG IN 6:21 PM 12/10/2023

Q5. Write a SQL query to Retrieve transaction details along with the account type.

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT, INSTANCE, PERFORMANCE, Administration, Information

Query 1 X

```

789 JOIN Customers c ON a.customer_id = c.customer_id
790 WHERE a.account_type = 'savings';
791
792 • SELECT t.transaction_id, t.account_id, t.transaction_type, t.amount, t.transaction_date, a.account_type
793 FROM Transactions t
794 JOIN Accounts a ON t.account_id = a.account_id;
795
796
797 • create database CarRentalSystem;
798 • use CarRentalSystem;

```

Result Grid:

transaction_id	account_id	transaction_type	amount	transaction_date	account_type
1	1	deposit	1000	2023-01-30	savings
2	2	withdrawal	500	2023-02-25	current
3	3	deposit	1500	2023-03-30	zero balance
4	4	withdrawal	800	2023-04-30	savings
5	5	transfer	2000	2023-05-30	current
6	6	deposit	1200	2023-06-30	savings
7	7	withdrawal	500	2023-07-30	zero balance
8	8	transfer	1000	2023-08-30	current
9	9	deposit	2000	2023-09-30	savings
10	10	withdrawal	300	2023-10-30	current

Output:

#	Time	Action	Message	Duration / Fetch
1	11:18:39	SELECT a.account_id, c.first_name, c.last_name, a.amount AS current_balance, a.amount * 0.05 AS interest_accrued F...	4 row(s) returned	0.000 sec / 0.000 sec
2	11:20:19	SELECT * FROM Customers WHERE city <> 'New York' LIMIT 0,1000	Error Code: 1054. Unknown column 'city' in 'where clause'	0.000 sec
3	11:23:28	SELECT t.transaction_id, t.account_id, t.transaction_type, t.amount, t.transaction_date, a.account_type FROM Transact...	10 row(s) returned	0.000 sec / 0.000 sec

Object Info: No object selected

Output: Read Only Context Help Snippets

System tray: ENG IN 11:23 AM 12/10/2023

Q6. Write a SQL query to Get a list of customers along with their account details.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
793 FROM Transactions t
794 JOIN Accounts a ON t.account_id = a.account_id;
795
796 • SELECT c.customer_id,c.first_name,c.last_name,c.DOB,c.email,c.phone,a.account_id,a.account_type,a.amount AS current_balance
797 FROM Customers c
798 LEFT JOIN Accounts a ON c.customer_id = a.customer_id;
799
800
801 • create database CarRentalSystem;
802 • use CarRentalSystem;
```

The results grid displays 10 rows of customer information, including their first name, last name, DOB, email, phone number, account ID, account type, and current balance. The columns are labeled: customer_id, first_name, last_name, DOB, email, phone, account_id, account_type, and current_balance.

customer_id	first_name	last_name	DOB	email	phone	account_id	account_type	current_balance
1	John	Doe	1990-05-15	john.doe@example.com	1234567890	1	savings	5000
2	Alice	Smith	1985-08-22	alice.smith@example.com	9876543210	2	current	7500
3	Bob	Johnson	1992-04-10	bob.johnson@example.com	5555555555	3	zero balance	2500
4	Emily	Williams	1988-12-01	emily.williams@example.com	4567890123	4	savings	10000
5	Michael	Brown	1995-06-30	michael.brown@example.com	7890123456	5	current	5000
6	Sophia	Miller	1997-09-18	sophia.miller@example.com	2345678901	6	savings	8000
7	Daniel	Davis	1993-03-05	daniel.davis@example.com	8901234567	7	zero balance	3000
8	Olivia	Jones	1989-11-20	olivia.jones@example.com	3210987654	8	current	6000
9	David	Taylor	1991-07-12	david.taylor@example.com	6789012345	9	savings	12000
10	Grace	Thomas	1994-02-28	grace.thomas@example.com	5432109876	10	current	4000

Q7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
800
801 • SELECT t.transaction_id,t.transaction_type,t.amount,t.transaction_date,c.customer_id,c.first_name,c.last_name,c.DOB,c.e
802 FROM Transactions t
803 JOIN Accounts a ON t.account_id = a.account_id
804 JOIN Customers c ON a.customer_id = c.customer_id
805 WHERE t.account_id = 1;
806
807
808 • create database CarRentalSystem;
809 • use CarRentalSystem;
```

The results grid displays 1 row of transaction details for account ID 1, showing a deposit of 1000 on 2023-01-30. The columns are labeled: transaction_id, transaction_type, amount, transaction_date, customer_id, first_name, last_name, DOB, email, and phone.

transaction_id	transaction_type	amount	transaction_date	customer_id	first_name	last_name	DOB	email	phone
1	deposit	1000	2023-01-30	1	John	Doe	1990-05-15	john.doe@example.com	1234567890

Q8. Write a SQL query to Identify customers who have more than one account.

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the Query Editor:

```
804 JOIN Customers c ON a.customer_id = c.customer_id
805 WHERE t.account_id = 1;
807 • SELECT c.customer_id, c.first_name, c.last_name, COUNT(a.account_id) AS account_count
808 FROM Customers c
809 JOIN Accounts a ON c.customer_id = a.customer_id
810 GROUP BY c.customer_id, c.first_name, c.last_name
811 HAVING COUNT(a.account_id) > 1;
```

The result grid shows the following data:

customer_id	first_name	last_name	account_count

On the right side, there is a vertical toolbar titled "SQLAdditions" with icons for back, forward, search, and jump. Below the toolbar, a message reads: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." A sidebar on the left contains sections for MANAGEMENT, INSTANCE, and PERFORMANCE, each with various sub-options. The Administration section is currently selected. At the bottom, the Result History panel shows two recent actions:

#	Time	Action	Message	Duration / Fetch
1	11:27:25	SELECT t.transaction_id,t.transaction_type,t.amount,t.transaction_date,c.customer_id,c.first_name,c.last_name,c.DOB...	1 row(s) returned	0.000 sec / 0.000 sec
2	11:28:57	SELECT c.customer_id,c.first_name,c.last_name,COUNT(a.account_id) AS account_count FROM Customers c JOIN	0 row(s) returned	0.000 sec / 0.000 sec

Q9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

The screenshot shows the MySQL Workbench interface. The main window displays a query editor with the following SQL code:

```
FROM Customers
ORDER BY DOB DESC
LIMIT 1;
```

The results grid shows one row of data:

	Sum(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END)
1	3600

The 'Information' panel on the left indicates 'No object selected'. The bottom status bar shows the duration of the query execution as 0.000 sec / 0.000 sec.

Q10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
811 HAVING COUNT(a.account_id) > 1;
812
813 • SELECT a.account_id,a.account_type,AVG(a.amount) AS average_daily_balance
814   FROM Accounts a
815   JOIN Transactions t ON a.account_id = t.account_id
816   WHERE t.transaction_date BETWEEN '2023-10-01' AND '2023-10-31'
817   GROUP BY a.account_id, a.account_type;
818
819
820 • create database CarRentalSystem;
```
- Result Grid:** Shows the output of the query:

account_id	account_type	average_daily_balance
10	current	4000
- Output Window:** Displays the execution log:

#	Time	Action	Message	Duration / Fetch
1	11:27:25	SELECT t.transaction_id,t.transaction_type,t.amount,t.transaction_date,c.customer_id,c.first_name,c.last_name,c.DOB,c...	1 row(s) returned	0.000 sec / 0.000 sec
2	11:28:57	SELECT c.customer_id,c.first_name,c.last_name,COUNT(a.account_id) AS account_count FROM Customers c JOIN ...	0 row(s) returned	0.000 sec / 0.000 sec
3	11:31:28	SELECT a.account_id,a.account_type,AVG(a.amount) AS average_daily_balance FROM Accounts a JOIN Transaction...	1 row(s) returned	0.032 sec / 0.000 sec

Q11. Calculate the total balance for each account type.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
571   SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE -amount END)
572   FROM
573     Transactions
574   WHERE
575     transaction_type IN ('deposit', 'withdrawal');
576
577 • SELECT
```
- Result Grid:** Shows the output of the query:

account_type	SUM(amount)
savings	35000
current	23800
zero balance	5500
- Output Window:** Displays the execution log:

#	Time	Action	Message	Duration / Fetch
1	18:25:58	SELECT account_type, SUM(amount) FROM Accounts GROUP BY account_type LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Q12. Identify accounts with the highest number of transactions order by descending order.

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator... Query 1 ...
MANAGEMENT
    Server Status
    Client Connectic...
    Users and Privile...
    Status and Syst...
    Data Export
    Data Import/Res...
INSTANCE
    Startup / Shutdo...
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Rep...
Administration
Information
No object selected
Result 114 x
Output:
Action Output
# Time Action
1 11:34:07 SELECT a.account_id, COUNT(t.transaction_id) AS transaction_count FROM Accounts a LEFT JOIN Transactions t ON a.account_id = t.account_id 10 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
Result Grid | Filter Rows: Export: Wrap Cell Content: 
account_id transaction_count
1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
SQLAdditions: < > | Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

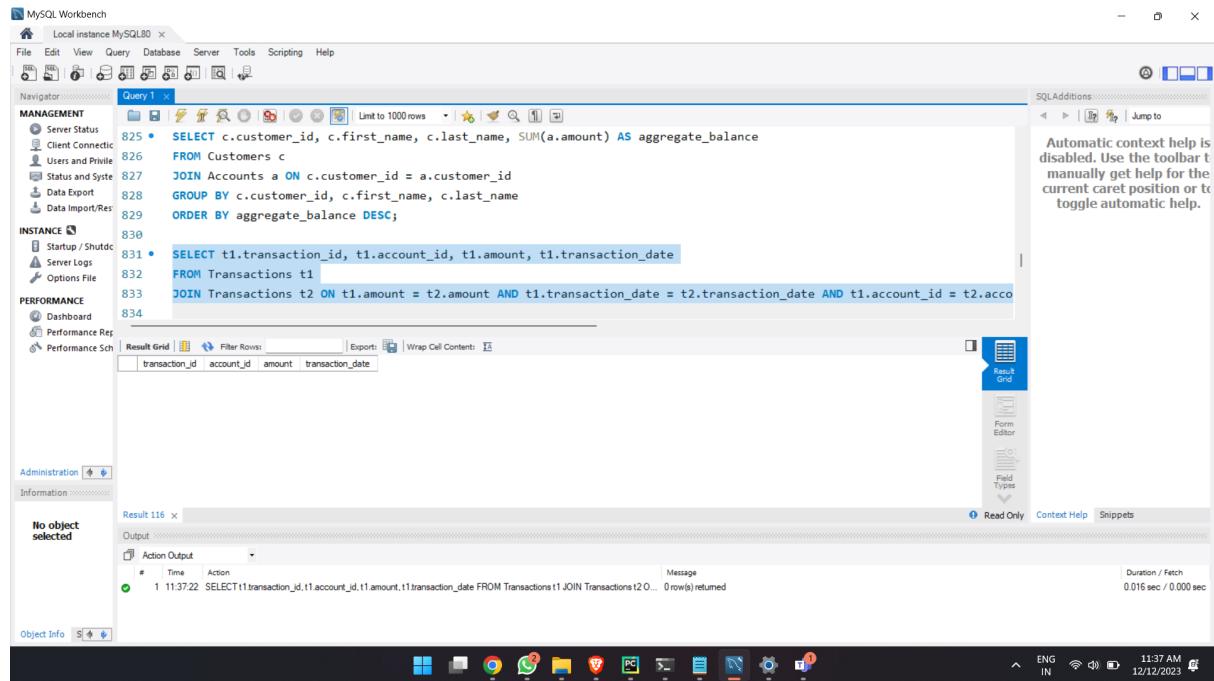
Q13. List customers with high aggregate account balances, along with their account types.

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator... Query 1 ...
MANAGEMENT
    Server Status
    Client Connectic...
    Users and Privile...
    Status and Syst...
    Data Export
    Data Import/Res...
INSTANCE
    Startup / Shutdo...
    Server Logs
    Options File
PERFORMANCE
    Dashboard
    Performance Rep...
Administration
Information
No object selected
Result 115 x
Output:
Action Output
# Time Action
1 11:35:39 SELECT c.customer_id, c.first_name, c.last_name, SUM(a.amount) AS aggregate_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 aggregate_balance DESC; 10 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
Result Grid | Filter Rows: Export: Wrap Cell Content: 
customer_id first_name last_name aggregate_balance
9 David Taylor 12000
4 Emily Williams 10000
6 Sophia Miller 8000
2 Alice Smith 7500
8 Olivia Jones 6000
1 John Doe 5000
5 Michael Brown 5000
10 Grace Thomas 4000
7 Daniel Davis 3000
3 Bob Johnson 2500
SQLAdditions: < > | Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

Q14. Identify and list duplicate transactions based on transaction amount, date, and account.



The screenshot shows the MySQL Workbench interface. In the top-left corner, the title bar reads "MySQL Workbench Local instance MySQL80". The main area contains two tabs: "Query1" and "Result 116".

Query1:

```
825 • SELECT c.customer_id, c.first_name, c.last_name, SUM(a.amount) AS aggregate_balance
  FROM Customers
  JOIN Accounts a ON c.customer_id = a.customer_id
 GROUP BY c.customer_id, c.first_name, c.last_name
 ORDER BY aggregate_balance DESC;
826
827
828
829
830
831 • SELECT t1.transaction_id, t1.account_id, t1.amount, t1.transaction_date
  FROM Transactions t1
  JOIN Transactions t2 ON t1.amount = t2.amount AND t1.transaction_date = t2.transaction_date AND t1.account_id = t2.account_id
832
833
834
```

Result 116:

transaction_id	account_id	amount	transaction_date
1	1	100	2023-12-12 11:37:22

The results show one row with transaction_id 1, account_id 1, amount 100, and transaction_date 2023-12-12 11:37:22.

TASK 4: Subquery and its type:

Q1. Retrieve the customer(s) with the highest account balance.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
834 •   SE Execute the selected portion of the script or everything, if there is no selection
835 •     a.account_id, a.amount AS account_balance
836
837 FROM Customers c
838 JOIN Accounts a ON c.customer_id = a.customer_id
839 WHERE a.amount = (SELECT MAX(amount) FROM Accounts);
840
841
842 •      create database CarRentalSystem;
843 •      use CarRentalSystem;
```

The result grid shows one row of data:

customer_id	first_name	last_name	account_id	account_balance
9	David	Taylor	9	12000

The status bar at the bottom right indicates the time is 11:40 AM and the date is 12/12/2023.

Q2. Calculate the average account balance for customers who have more than one account.

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query is:

```
838 WHERE a.amount = (SELECT MAX(amount) FROM Accounts);
839
840 •     SELECT c.customer_id, c.first_name, c.last_name, AVG(a.amount) AS average_account_balance
841 FROM Customers c
842 JOIN Accounts a ON c.customer_id = a.customer_id
843 WHERE c.customer_id IN (SELECT customer_id FROM Accounts GROUP BY customer_id HAVING COUNT(account_id) > 1)
844 GROUP BY c.customer_id, c.first_name, c.last_name;
845
846
847 •      create database CarRentalSystem;
```

The result grid shows one row of data:

customer_id	first_name	last_name	average_account_balance
9	David	Taylor	12000

The status bar at the bottom right indicates the time is 11:42 AM and the date is 12/12/2023.

Q3. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

846 •  SELECT *
847   FROM Accounts a
848     JOIN Transactions t ON a.account_id = t.account_id
849 WHERE t.amount > (SELECT AVG(amount) FROM Transactions);
      
```
- Result Grid:** Displays a table with 9 rows of data from the Transactions table. The columns are account_id, customer_id, account_type, amount, transaction_date, transaction_id, account_id, transaction_type, and amount. The data includes various account types like 'zero balance', 'current', and 'savings' with transaction amounts ranging from 1500 to 12000.
- Output Window:** Shows the execution log with three entries:

#	Time	Action	Message	Duration / Fetch
1	11:40:35	SELECT c.customer_id, c.first_name, c.last_name, a.account_id, a.amount AS account_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, a.account_id, a.amount ORDER BY account_balance DESC LIMIT 1	1 row(s) returned	0.046 sec / 0.000 sec
2	11:42:30	SELECT c.customer_id, c.first_name, c.last_name, AVG(a.amount) AS average_account_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	0 row(s) returned	0.000 sec / 0.000 sec
3	11:44:18	SELECT * FROM Accounts a JOIN Transactions t ON a.account_id = t.account_id WHERE t.amount > (SELECT AVG(amount) FROM Transactions)	4 row(s) returned	0.000 sec / 0.000 sec

Q4. Identify customers who have no recorded transactions.

There are no such customers.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:


```

851
852
853 •  SELECT c.customer_id, c.first_name, c.last_name
854   FROM Customers c
855     LEFT JOIN Accounts a ON c.customer_id = a.customer_id
856     LEFT JOIN Transactions t ON a.account_id = t.account_id
857 WHERE t.transaction_id IS NULL;
      
```
- Result Grid:** Displays a table with 0 rows, indicating no customers have no recorded transactions.
- Output Window:** Shows the execution log with four entries:

#	Time	Action	Message	Duration / Fetch
2	11:42:30	SELECT c.customer_id, c.first_name, c.last_name, AVG(a.amount) AS average_account_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	0 row(s) returned	0.000 sec / 0.000 sec
3	11:44:18	SELECT * FROM Accounts a JOIN Transactions t ON a.account_id = t.account_id WHERE t.amount > (SELECT AVG(amount) FROM Transactions)	4 row(s) returned	0.000 sec / 0.000 sec
4	11:46:08	SELECT c.customer_id, c.first_name, c.last_name FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.customer_id WHERE a.account_id IS NULL	0 row(s) returned	0.046 sec / 0.000 sec

Q5. Calculate the total balance of accounts with no recorded transactions.

There are no such customer so there is no average.

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
851
852
853 • SELECT c.customer_id, c.first_name, c.last_name
854   FROM Customers c
855   LEFT JOIN Accounts a ON c.customer_id = a.customer_id
856   LEFT JOIN Transactions t ON a.account_id = t.account_id
857   WHERE t.transaction_id IS NULL;
```

The results grid shows three columns: customer_id, first_name, and last_name. There are no rows returned.

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
2	11:42:30	SELECT c.customer_id, c.first_name, c.last_name, AVG(a.amount) AS average_account_balance FROM Customers c ...	0 row(s) returned	0.000 sec / 0.000 sec
3	11:44:18	SELECT * FROM Accounts a JOIN Transactions t ON a.account_id = t.account_id WHERE t.amount > (SELECT AVG(...)	4 row(s) returned	0.000 sec / 0.000 sec
4	11:46:08	SELECT c.customer_id, c.first_name, c.last_name FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.c...	0 row(s) returned	0.046 sec / 0.000 sec

Q6. Retrieve transactions for accounts with the lowest balance.

The screenshot shows the MySQL Workbench interface with a query editor containing the following SQL code:

```
856
857
858 • LEFT JOIN Transactions t ON a.account_id = t.account_id
859 WHERE t.transaction_id IS NULL;
860
861 • SELECT t.transaction_id,t.account_id,t.transaction_type,t.amount, t.transaction_date
862   FROM Transactions t
863   JOIN Accounts a ON t.account_id = a.account_id
864   WHERE a.amount = (SELECT MIN(amount) FROM Accounts);
865
```

The results grid shows five columns: transaction_id, account_id, transaction_type, amount, and transaction_date. One row is returned, showing a deposit of 1500 on 2023-03-30.

The output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
3	11:44:18	SELECT * FROM Accounts a JOIN Transactions t ON a.account_id = t.account_id WHERE t.amount > (SELECT AVG(...)	4 row(s) returned	0.000 sec / 0.000 sec
4	11:46:08	SELECT c.customer_id, c.first_name, c.last_name FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.c...	0 row(s) returned	0.046 sec / 0.000 sec
5	11:49:09	SELECT t.transaction_id,t.account_id,t.transaction_type,t.amount, t.transaction_date FROM Transactions t JOIN Acco...	1 row(s) returned	0.016 sec / 0.000 sec

Q7. Identify customers who have accounts of multiple types.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT (Server Status, Client Connectic, Users and Privilec, Status and Syste, Data Export, Data Import/Res), INSTANCE (Startup / Shuttde, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Rep, Performance Sch).
- Query Editor:** Query 1 (SQL code)

```
863
864 •   SELECT c.customer_id, c.first_name, c.last_name
865     FROM Customers c
866       JOIN Accounts a ON c.customer_id = a.customer_id GROUP BY c.customer_id, c.first_name, c.last_name HAVING COUNT(DISTINCT
867
868
869
870
871
872 •   create database CarRentalSystem;
```
- Result Grid:** customer_id, first_name, last_name
- Output Window:** Result 122 (Action Output, Duration / Fetch)

#	Time	Action	Message	Duration / Fetch
4	11:46:08	SELECT c.customer_id, c.first_name, c.last_name FROM Customers c LEFT JOIN Accounts a ON c.customer_id = a.c...	0 row(s) returned	0.046 sec / 0.000 sec
5	11:49:09	SELECT t.transaction_id,t.account_id,t.transaction_type,t.amount,t.transaction_date FROM Transactions t JOIN Acco...	1 row(s) returned	0.016 sec / 0.000 sec
6	11:50:54	SELECT c.customer_id,c.first_name,c.last_name FROM Customers c JOIN Accounts a ON c.customer_id = a.custome...	0 row(s) returned	0.031 sec / 0.000 sec
- System Bar:** ENG IN 11:53 AM 12/12/2023

Q8. Calculate the percentage of each account type out of the total number of accounts.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** MANAGEMENT (Server Status, Client Connectic, Users and Privilec, Status and Syste, Data Export, Data Import/Res), INSTANCE (Startup / Shuttde, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Rep, Performance Sch).
- Query Editor:** Query 1 (SQL code)

```
866     JOIN Accounts a ON c.customer_id = a.customer_id GROUP BY c.customer_id, c.first_name, c.last_name HAVING COUNT(DISTINC
867
868
869 •   SELECT account_type,COUNT(account_id) AS account_count,(COUNT(account_id) * 100.0 / (SELECT COUNT(*) FROM Accounts)) AS
870     FROM Accounts
871     GROUP BY account_type;
872
873
874
875 •   create database CarRentalSystem;
```
- Result Grid:** account_type, account_count, percentage
- Output Window:** Result 123 (Action Output, Duration / Fetch)

account_type	account_count	percentage
savings	4	40.0000
current	4	40.0000
zero balance	2	20.0000
- System Bar:** ENG IN 11:53 AM 12/12/2023

Q9. Retrieve all transactions for a customer with a given customer_id.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
871 • GROUP BY account_type;
872
873 • SELECT *
874   FROM Transactions t
875   JOIN Accounts a ON t.account_id = a.account_id
876   WHERE a.customer_id = 5;
877
878
879 • create database CarRentalSystem;
880 • use CarRentalSystem;
```
- Result Grid:** Shows the results of the query:

transaction_id	account_id	transaction_type	amount	transaction_date	account_id	customer_id	account_type	amount	transaction_date
5	5	transfer	2000	2023-05-30	5	5	current	5000	2023-05-05
- Output Window:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
7	11:53:08	SELECT account_type,COUNT(account_id) AS account_count,(COUNT(account_id) * 100 / (SELECT COUNT(*) F...) 3 row(s) returned		0.062 sec / 0.000 sec
8	11:54:23	SELECT * FROM Transactions t JOIN Accounts a ON t.account_id = a.account_id WHERE a.customer_id = 'you_cu...' Error Code: 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server ver...		0.000 sec
9	11:54:39	SELECT * FROM Transactions t JOIN Accounts a ON t.account_id = a.account_id WHERE a.customer_id = 5 LIMIT ... 1 row(s) returned		0.000 sec / 0.000 sec

Q10. Calculate the total balance for each account type, including a subquery within the SELECT clause.

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the following SQL code:

```
877
878
879 • SELECT account_type,(SELECT SUM(amount) FROM Accounts a) AS total_balance
880   FROM Transactions t
881   JOIN Accounts a ON t.account_id = a.account_id
882   GROUP BY account_type;
883
884
885 • create database CarRentalSystem;
886 • use CarRentalSystem;
```
- Result Grid:** Shows the results of the query:

account_type	total_balance
saving	63000
current	63000
zero balance	63000
- Output Window:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
10	11:56:07	SELECT account_type,(SELECT SUM(amount) FROM Accounts a WHERE a.account_type = t.account_type) AS tota... Error Code: 1054: Unknown column 't.account_type' in 'where clause'		0.000 sec
11	11:56:49	SELECT account_type,(SELECT SUM(amount) FROM Accounts a WHERE a.account_type = t.account_type) AS tota... Error Code: 1054: Unknown column 't.account_type' in 'where clause'		0.000 sec
12	11:57:33	SELECT account_type,(SELECT SUM(amount) FROM Accounts a) AS total_balance FROM Transactions t JOIN Acco... 3 row(s) returned		0.000 sec / 0.000 sec