CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

LAB - 3

Banner ID: B00984406
GitLab Assignment Link: N/A

Table of Contents

Designing a banking application database with tables

Create a database called “lab3”:

CREATE DATABASE IF NOT EXISTS lab3;

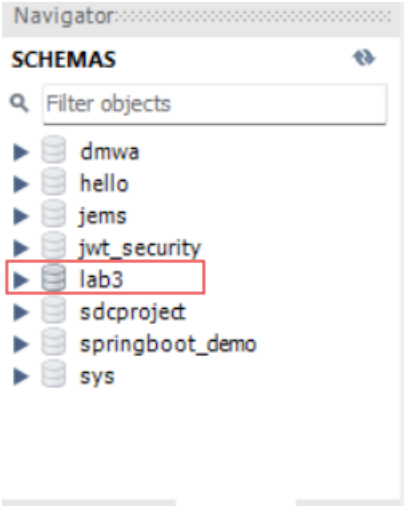


Figure 1.1: Database lab3 created

Create a table “customer_details”:

```
CREATE TABLE customer_details (
 customer_id INT,
 name VARCHAR(20),
 address VARCHAR(50),
 email VARCHAR(20),
 phone_number VARCHAR(15),
 PRIMARY KEY (customer_id)
);
```

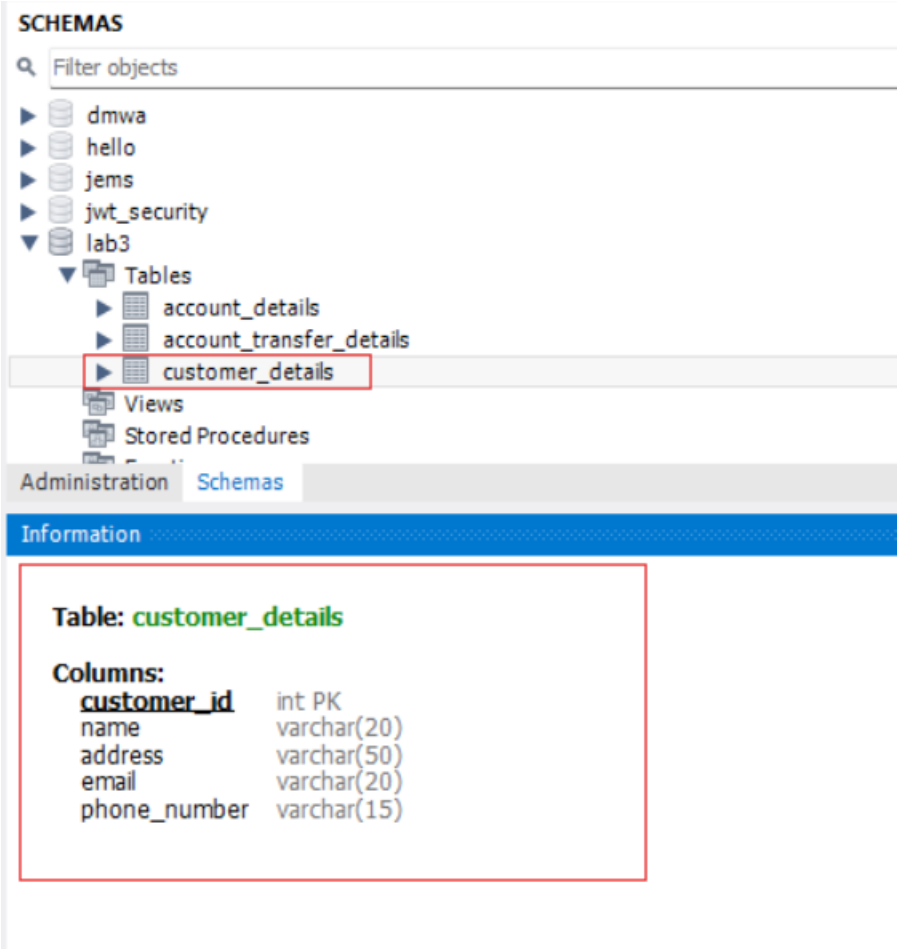


Figure 1.2: customer_details table created

Create a table “account_details”:

```
CREATE TABLE account_details(
 account_number INT,
 account_balance INT NOT NULL,
 customer_id INT,
 PRIMARY KEY (account_number),
 FOREIGN KEY (customer_id) REFERENCES customer_details(customer_id)
);
```

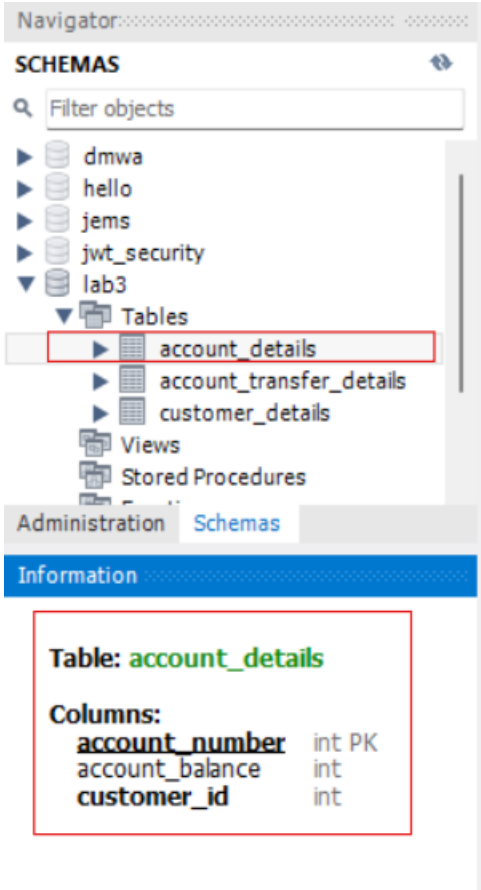


Figure 1.3: account_details table created

Create a table “customer_transfer_details”:

```
CREATE TABLE account_transfer_details (
 transfer_id INT,
 sender_account_number INT,
 receiver_account_number INT,
 transfer_date DATETIME,
 status ENUM ("waiting","accepted","declined"),
 PRIMARY KEY (transfer_id),
 FOREIGN KEY (sender_account_number) REFERENCES
account_details(account_number),
 FOREIGN KEY (receiver_account_number) REFERENCES
account_details(account_number)
);
```

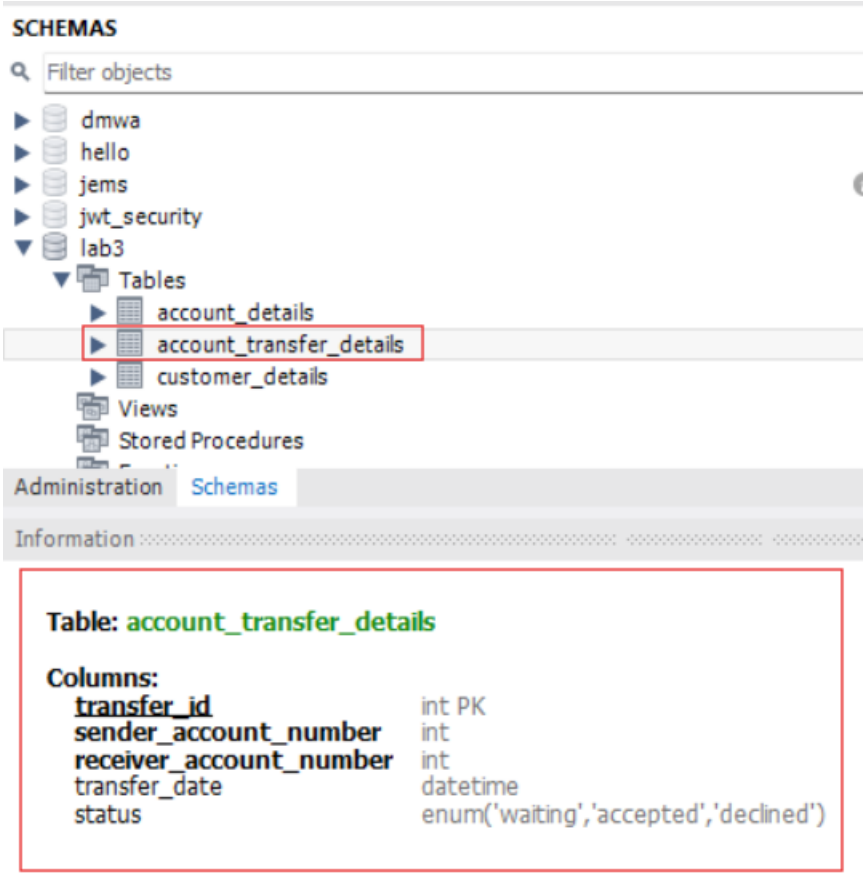


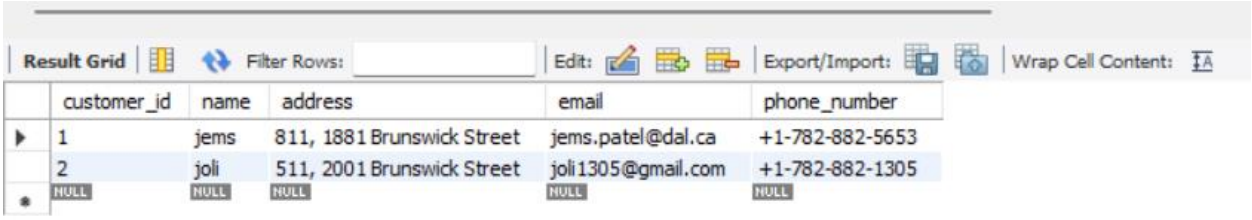
Figure 1.4: account_transfer_details table created

Inserting the sample/dummy data in tables

Inserting the data into the table “customer_details”:

INSERT INTO customer_details(customer_id, name, address, email, phone_number) VALUES (1, "jems", "811, 1881 Brunswick Street", "jems.patel@dal.ca", "+1-782-882-5653");

INSERT INTO customer_details(customer_id, name, address, email, phone_number) VALUES (2, "joli", "511, 2001 Brunswick Street", "joli1305@gmail.com", "+1-782-882-1305");



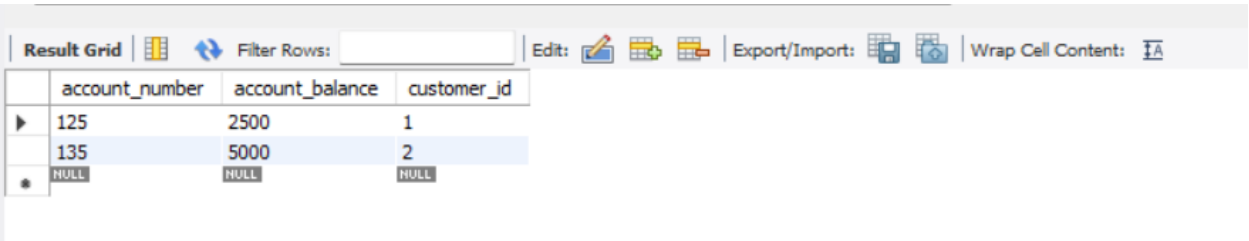
customer_id	name	address	email	phone_number
1	jems	811, 1881 Brunswick Street	jems.patel@dal.ca	+1-782-882-5653
2	joli	511, 2001 Brunswick Street	joli1305@gmail.com	+1-782-882-1305
NULL	NULL	NULL	NULL	NULL

Figure 2.1: inserting data into the customer_details table

Inserting the data into the table “account_details”:

INSERT INTO account_details(account_number, account_balance, customer_id) VALUES (125, 2500, 1);

INSERT INTO account_details(account_number, account_balance, customer_id) VALUES (135, 5000, 2);



account_number	account_balance	customer_id
125	2500	1
135	5000	2
NULL	NULL	NULL

Figure 2.2: inserting data into the account_details table

Create transactions for the below two scenarios.

3.A Scenario-1 (Transaction “Accepted” state): -

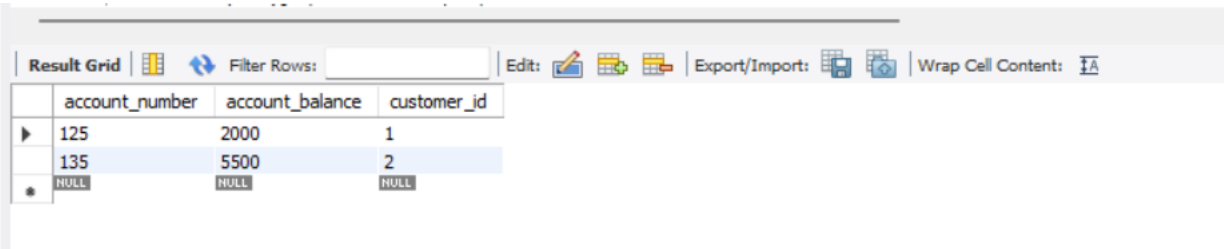
Creating the transaction:

```
SET AUTOCOMMIT=0;

START TRANSACTION;

	UPDATE account_details
		SET account_balance = account_balance - 500
			WHERE account_number = 125;

	INSERT INTO account_transfer_details (transfer_id, sender_account_number,
	receiver_account_number, transfer_date, status)
		VALUES (1, 125, 135, NOW(), 'waiting');

	-- Assuming the transaction has accepted (assumption)

	UPDATE account_details
		SET account_balance = account_balance + 500
			WHERE account_number = 135;

	UPDATE account_transfer_details
		SET status = 'accepted'
			WHERE transfer_id = 1;

COMMIT;
```

Explanation:

- Initially, I disabled auto commit and initiated the transaction. Subsequently, I deducted 500 from the sender's account balance by specifying the account number.
- Following that, I inserted a new record into the account_transfer_details table with necessary details like sender's account number, receiver's account number, transfer date, and the transaction status set to "waiting".
- Assuming the transaction passed the security verification, I increased the receiver's account balance by 500 units.
- Finally, I changed the transaction status to "accepted" and committed the transaction.

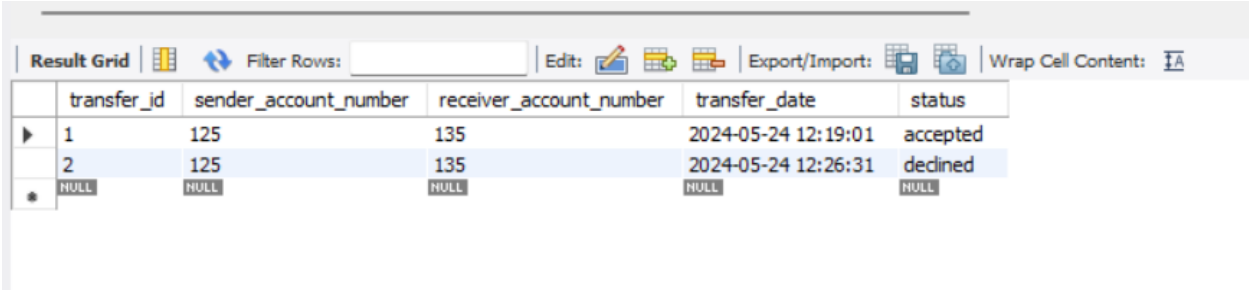
Screenshots:



account_number	account_balance	customer_id
125	2000	1
135	5500	2
NULL	NULL	NULL

Figure 3.1: account_details table after the successfully transaction

- It is observed that a total of 100 have been debited from account number 111 and credited to account number.



transfer_id	sender_account_number	receiver_account_number	transfer_date	status
1	125	135	2024-05-24 12:19:01	accepted
2	125	135	2024-05-24 12:26:31	declined
NULL	NULL	NULL	NULL	NULL

Figure 3.2: account_transfer_details table

- An additional entry has been made for the transaction, indicating it has been accepted.

3.B Scenario-2 (Transaction “Declined” state): -

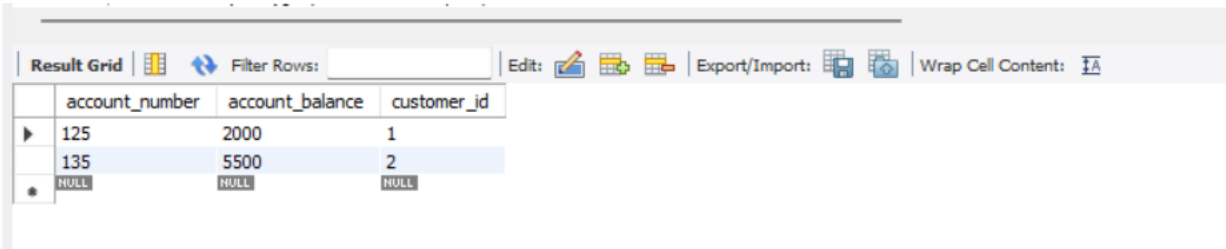
```
SET AUTOCOMMIT=0;

START TRANSACTION;

	UPDATE account_details
		SET account_balance = account_balance - 500
			WHERE account_number = 125;

	INSERT INTO account_transfer_details (transfer_id, sender_account_number,
	receiver_account_number, transfer_date, status)
		VALUES (2, 125, 135, NOW(), 'waiting');

	SAVEPOINT before_failure;

	-- Assuming the transaction has failed (assumption)

	ROLLBACK TO before_failure;

	UPDATE account_details
		SET account_balance = account_balance + 500
			WHERE account_number = 125;

	UPDATE account_transfer_details
		SET status = 'declined'
			WHERE transfer_id = 2;

COMMIT;
```

Explanation:

- Initially, I disabled auto commit and initiated the transaction. Subsequently, I deducted 500 from the sender's account balance by specifying the account number.
- Subsequent to that, I added a fresh entry into the account_transfer_details table with essential information such as the sender's account number, receiver's account number, transfer date, and the transaction status set as "waiting".
- Following this, I established a savepoint named before_failure to enable reverting to this point in case of any transaction failure.
- In the event of a transaction failure (this is an assumption), I rolled back the transaction to the savepoint before_failure, effectively reversing any modifications made post the savepoint.
- After rolling back, I restored the sender's account balance by adding back the 500 units that were previously deducted.

- Finally, I updated the transaction status to "declined" in the account_transfer_details table to indicate the failure of the transaction and then committed the transaction to finalize all changes.

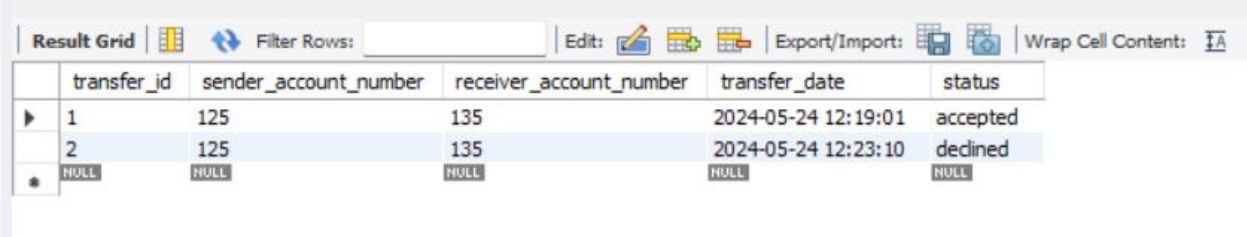
Screenshots:



account_number	account_balance	customer_id
125	2000	1
135	5500	2
NULL	NULL	NULL

Figure 3.3: account_details table after the failed transaction

- Both the sender and the receiver maintain their balance unchanged following the unsuccessful transaction.



transfer_id	sender_account_number	receiver_account_number	transfer_date	status
1	125	135	2024-05-24 12:19:01	accepted
2	125	135	2024-05-24 12:23:10	declined
NULL	NULL	NULL	NULL	NULL

Figure 3.4: account_transfer_details table

- An additional entry has been made for the transaction, indicating it has been declined.