



NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.

Navigation

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Table: transactions

Columns:

TransactionID

AccountID

TransactionType

Amount

TransactionDate

int AI PK

int

enum('Deposit','Withdrawal')

decimal(10,2)

timestamp

SQL File 3

1 START TRANSACTION;

2 UPDATE Loans SET Status = 'Paid' WHERE LoanID = 5;

3 INSERT INTO Payments (LoanID, AmountPaid) VALUES (5, 5000);

4 COMMIT;

5

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help

Snippets

Output

Action Output

Time Action Message Duration / Fetch

27 10:30:08 COMMIT 0 row(s) affected 0.000 sec

28 10:31:38 SELECT * FROM transactions LIMIT 0, 1000 262 row(s) returned 0.000 sec / 0.000 sec

29 10:32:51 START TRANSACTION 0 row(s) affected 0.000 sec

30 10:32:51 UPDATE Loans SET Status = 'Paid' WHERE LoanID = 5 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0 0.000 sec

31 10:32:51 INSERT INTO Payments (LoanID, AmountPaid) VALUES (5, 5000) 1 row(s) affected 0.000 sec

32 10:32:51 COMMIT 0 row(s) affected 0.016 sec

Title: Marking a Loan as Paid and Logging a Payment

Description: This transaction updates the status of LoanID 5 to "Paid" and records a payment of ₱5,000 in the Payments table. The COMMIT statement ensures that both operations are permanently saved.

Navigation

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Table: transactions

Columns:

TransactionID

AccountID

TransactionType

Amount

TransactionDate

int AI PK

int

enum('Deposit','Withdrawal')

decimal(10,2)

timestamp

SQL File 3

1 SELECT * FROM payments;

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help

Snippets

Result Grid

Filter Rows

PaymentID LoanID AmountPaid PaymentDate

253 205 4388.73 2025-03-02 10:27:57

254 176 3667.81 2025-03-02 10:27:57

255 30 172.90 2025-03-02 10:27:57

256 243 4861.05 2025-03-02 10:27:57

257 199 3786.53 2025-03-02 10:27:57

258 220 4349.53 2025-03-02 10:27:57

259 219 388.07 2025-03-02 10:27:57

260 55 3891.75 2025-03-02 10:27:57

512 5 5000.00 2025-03-02 10:32:51

payments 12 x

Apply

Revert

Context Help

Snippets

Output

Action Output

Time Action Message Duration / Fetch

28 10:31:38 SELECT * FROM transactions LIMIT 0, 1000 262 row(s) returned 0.000 sec / 0.000 sec

29 10:32:51 START TRANSACTION 0 row(s) affected 0.000 sec

30 10:32:51 UPDATE Loans SET Status = 'Paid' WHERE LoanID = 5 0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0 0.000 sec

31 10:32:51 INSERT INTO Payments (LoanID, AmountPaid) VALUES (5, 5000) 1 row(s) affected 0.000 sec

32 10:32:51 COMMIT 0 row(s) affected 0.016 sec

33 10:34:32 SELECT * FROM payments LIMIT 0, 1000 261 row(s) returned 0.000 sec / 0.000 sec

Title: Confirming Loan Payment Records

Description: This query retrieves all payments, ensuring the recently recorded ₱5,000 payment for LoanID 5 is correctly reflected in the database.



NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Schema: bankingsystem

Week6_LabFiles_Charls-Emil_B

SQL File 3

Limit to 1000 rows

1 CREATE USER 'bank_clerk'@'localhost' IDENTIFIED BY 'securepassword';

2 GRANT SELECT, UPDATE ON BankingSystem.Accounts TO 'bank_clerk'@'localhost';

3

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
32	10:32:51	COMMIT	0 row(s) affected	0.016 sec
33	10:34:32	SELECT * FROM payments LIMIT 0, 1000	261 row(s) returned	0.000 sec / 0.000 sec
34	10:35:51	SELECT * FROM accounts LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec
35	10:36:31	SELECT * FROM loans LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec
36	10:37:12	CREATE USER bank_clerk@'localhost' IDENTIFIED BY 'securepassword'	0 row(s) affected	0.031 sec
37	10:37:12	GRANT SELECT, UPDATE ON BankingSystem.Accounts TO bank_clerk@'localhost'	0 row(s) affected	0.000 sec

Object Info

Session

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Schema: bankingsystem

Week6_LabFiles_Charls-Emil_B

SQL File 3

Limit to 1000 rows

1 CREATE USER 'auditor'@'localhost' IDENTIFIED BY 'readonlypass';

2 GRANT SELECT ON BankingSystem.* TO 'auditor'@'localhost';

3

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
34	10:35:51	SELECT * FROM accounts LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec
35	10:36:31	SELECT * FROM loans LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec
36	10:37:12	CREATE USER bank_clerk@'localhost' IDENTIFIED BY 'securepassword'	0 row(s) affected	0.031 sec
37	10:37:12	GRANT SELECT, UPDATE ON BankingSystem.Accounts TO bank_clerk@'localhost'	0 row(s) affected	0.000 sec
38	10:38:58	CREATE USER auditor@'localhost' IDENTIFIED BY 'readonlypass'	0 row(s) affected	0.000 sec
39	10:38:58	GRANT SELECT ON BankingSystem.* TO auditor@'localhost'	0 row(s) affected	0.000 sec

Object Info

Session

Title: Defining User Roles and Permissions in the Database

Description: This query creates two database users:

- 'bank_clerk' with permission to view and update the Accounts table.
- 'auditor' with read-only access to the entire BankingSystem database.

This ensures role-based access control for security.

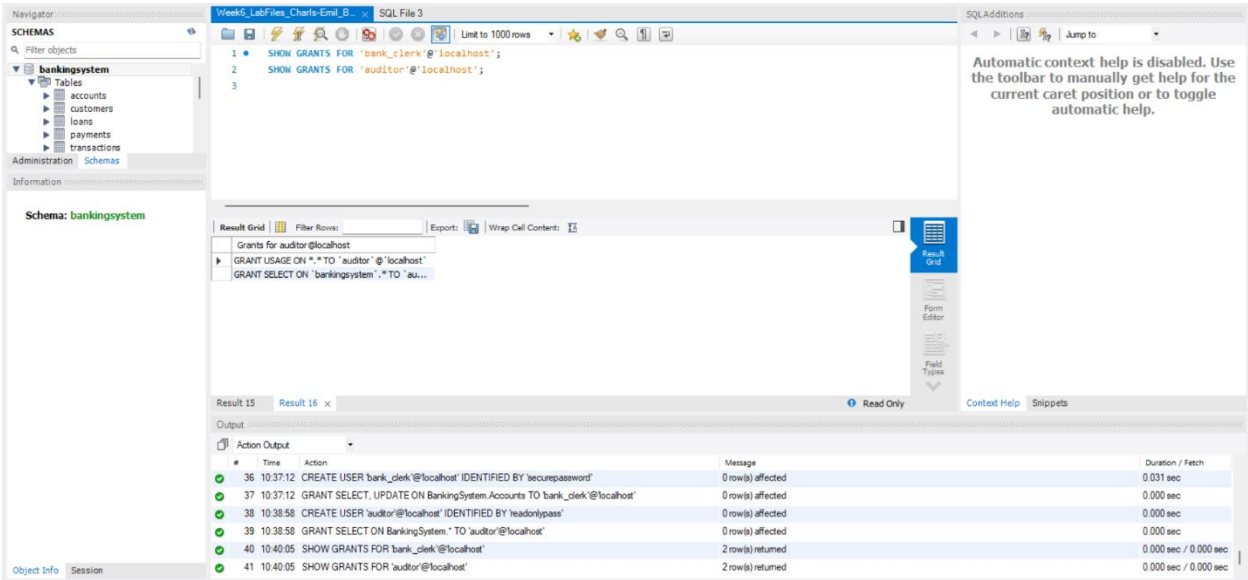


NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

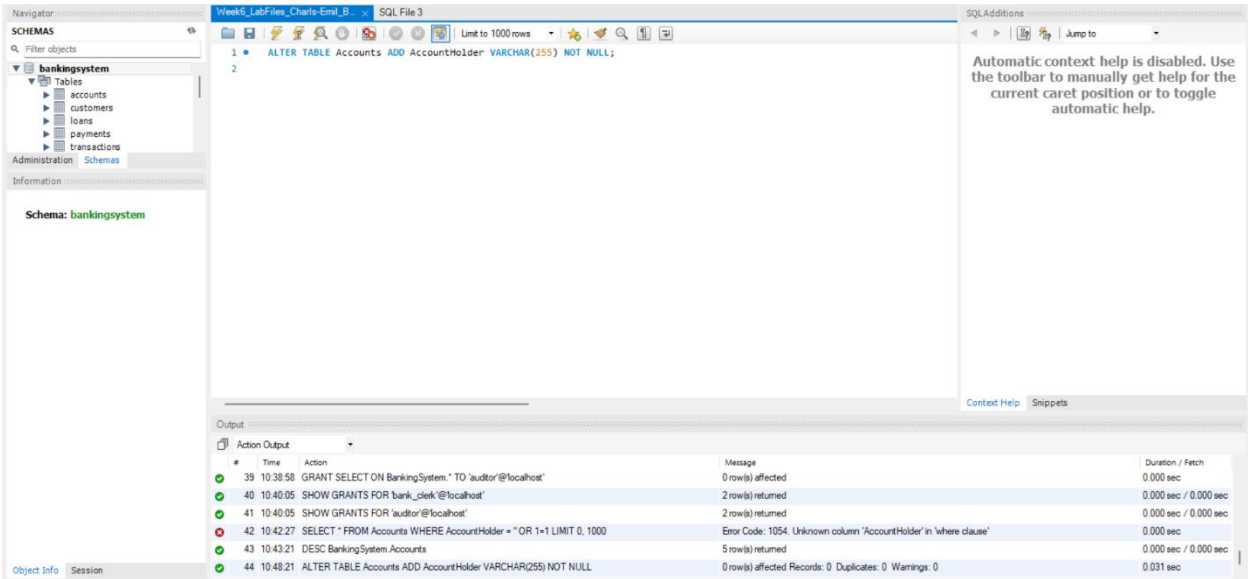
COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.



Title: Reviewing Granted Permissions for Users

Description: This query checks and verifies the assigned privileges for the 'bank_clerk' and 'auditor' users, ensuring they have the correct permissions based on their roles.



Title: Adding an Account Holder Column to Accounts Table

Description: This query modifies the Accounts table by adding an AccountHolder column to store the name of the individual associated with each account. The NOT NULL constraint ensures all accounts must have an account holder.



NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Schema: bankingsystem

Object Info

Session

Week6_LabFiles_Charls-Emil_B_

SQL File 3

Limit to 1000 rows

1 SELECT * FROM Accounts WHERE AccountHolder = '' OR 1=1;

2

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows

Export/Import

Wrap Cell Contents

Accounts 18

Apply

Revert

Context Help

Snippets

AccountID	CustomerID	AccountType	Balance	CreatedAt	AccountHolder
1	200	Checking	15379.19	2025-03-02 10:23:48	
2	224	Checking	59932.18	2025-03-02 10:23:48	
3	151	Savings	72298.83	2025-03-02 10:23:48	
4	170	Savings	43288.18	2025-03-02 10:23:48	
5	28	Savings	71785.99	2025-03-02 10:23:48	
6	184	Savings	74340.29	2025-03-02 10:23:48	
7	147	Checking	66391.62	2025-03-02 10:23:48	
8	94	Checking	92378.16	2025-03-02 10:23:48	
9	8	Checking	57660.56	2025-03-02 10:23:48	
10	54	Savings	64147.22	2025-03-02 10:23:48	

Action Output

#	Time	Action	Message	Duration / Fetch
40	10:40:05	SHOW GRANTS FOR 'bank_clerk'@'localhost'	2 row(s) returned	0.000 sec / 0.000 sec
41	10:40:05	SHOW GRANTS FOR 'auditor'@'localhost'	2 row(s) returned	0.000 sec / 0.000 sec
42	10:42:27	SELECT * FROM Accounts WHERE AccountHolder = '' OR 1=1 LIMIT 0, 1000	Error Code: 1054. Unknown column 'AccountHolder' in 'where clause'	0.000 sec
43	10:43:21	DESC Bankingsystem.Accounts	5 row(s) returned	0.000 sec / 0.000 sec
44	10:48:21	ALTER TABLE Accounts ADD AccountHolder VARCHAR(255) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
45	10:48:45	SELECT * FROM Accounts WHERE AccountHolder = '' OR 1=1 LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec

Title: Testing SQL Injection Vulnerability

Description: This query attempts to exploit a SQL injection vulnerability by using the condition OR 1=1, which would return all accounts if the database were not properly secured against such attacks.

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Schema: bankingsystem

Object Info

Session

Week6_LabFiles_Charls-Emil_B_

SQL File 3

Limit to 1000 rows

1 PREPARE stmt FROM 'SELECT * FROM Accounts WHERE AccountHolder = ?';

2 SET @holder = 'Alice Johnson';

3 EXECUTE stmt USING @holder;

4 DEALLOCATE PREPARE stmt;

5

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows

Export

Wrap Cell Contents

Result 19

Read Only

Context Help

Snippets

AccountID	CustomerID	AccountType	Balance	CreatedAt	AccountHolder
-----------	------------	-------------	---------	-----------	---------------

Action Output

#	Time	Action	Message	Duration / Fetch
44	10:48:21	ALTER TABLE Accounts ADD AccountHolder VARCHAR(255) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
45	10:48:45	SELECT * FROM Accounts WHERE AccountHolder = '' OR 1=1 LIMIT 0, 1000	260 row(s) returned	0.000 sec / 0.000 sec
46	10:50:08	PREPARE stmt FROM 'SELECT * FROM Accounts WHERE AccountHolder = ?'	0 row(s) affected Statement prepared	0.015 sec
47	10:50:08	SET @holder = 'Alice Johnson'	0 row(s) affected	0.000 sec
48	10:50:08	EXECUTE stmt USING @holder	0 row(s) returned	0.000 sec / 0.000 sec
49	10:50:08	DEALLOCATE PREPARE stmt	0 row(s) affected	0.000 sec

Title: Preventing SQL Injection with Prepared Statements

Description: This query safely retrieves accounts belonging to 'Alice Johnson' using a prepared statement, ensuring user input is securely handled to prevent SQL injection attacks.

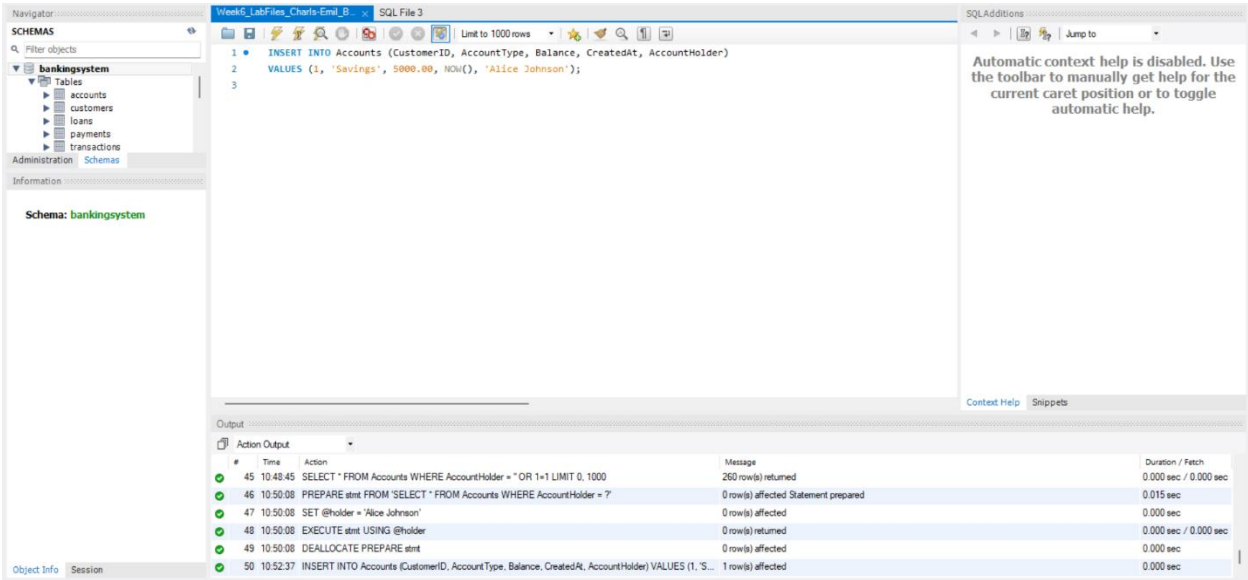


Republic of the Philippines
Bicol University Polangui
Polangui, Albay



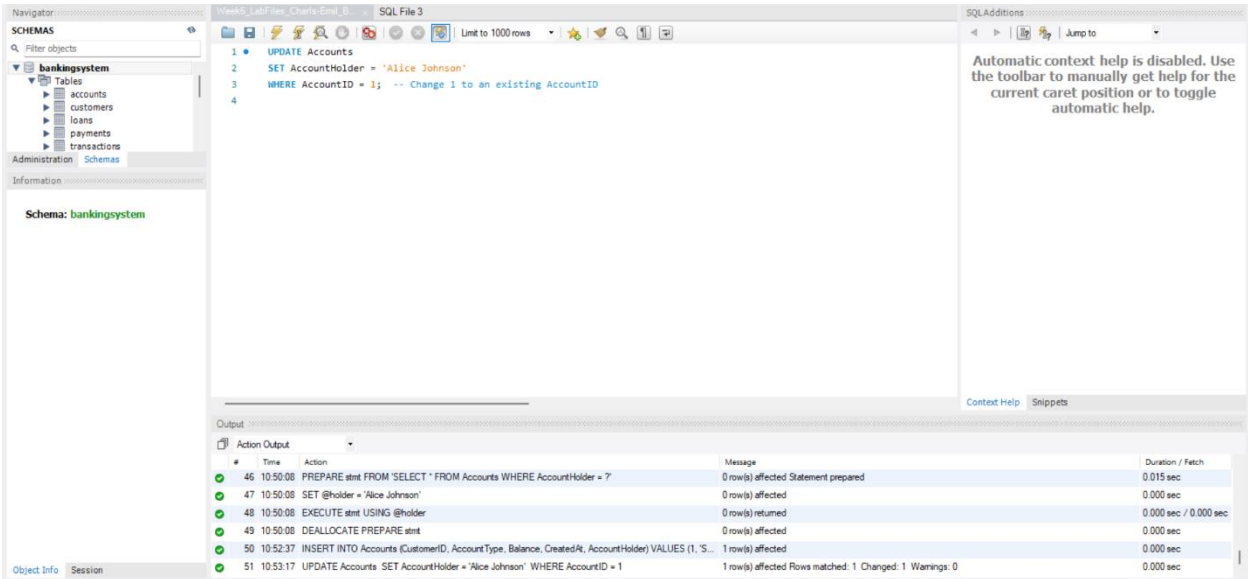
NAME: Charls Emil C. Barquin
COURSE, YEAR & SECTION: BSIS-2A

SUBJECT: Information Management 1
PROFESSOR: Mr. Red Guillermo V. Jr.



Title: Creating a New Savings Account for Alice Johnson

Description: This query inserts a new savings account for CustomerID 1, assigning 'Alice Johnson' as the account holder and setting the balance to ₱5,000.



Title: Assigning an Account Holder to an Existing Account

Description: This query updates AccountID 1 to assign 'Alice Johnson' as the account holder, ensuring account records include holder details.

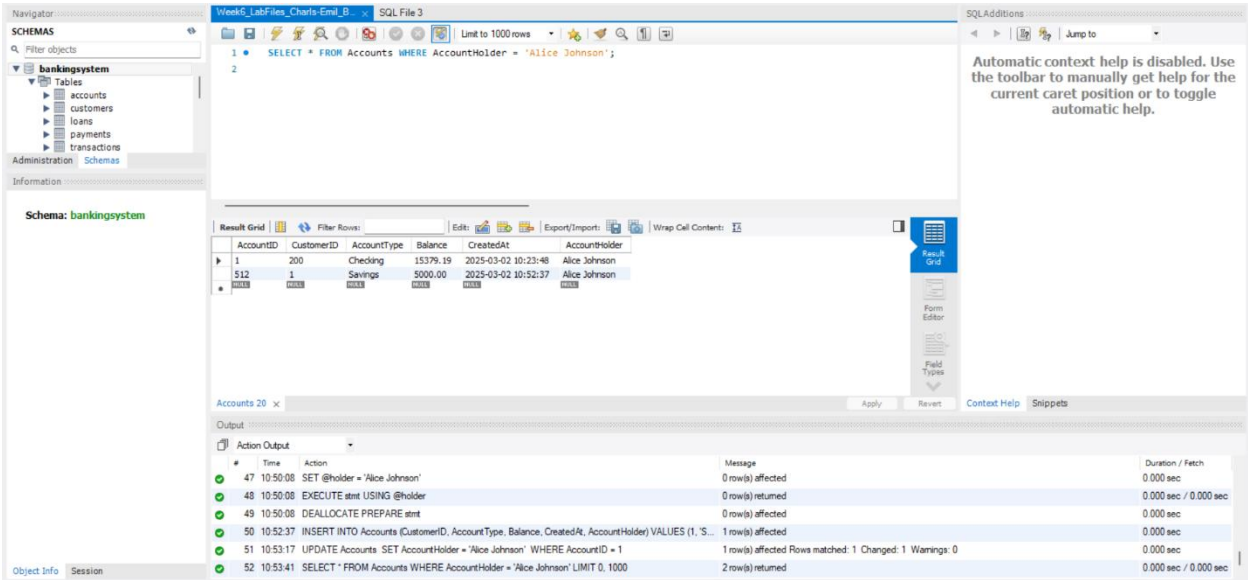


NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

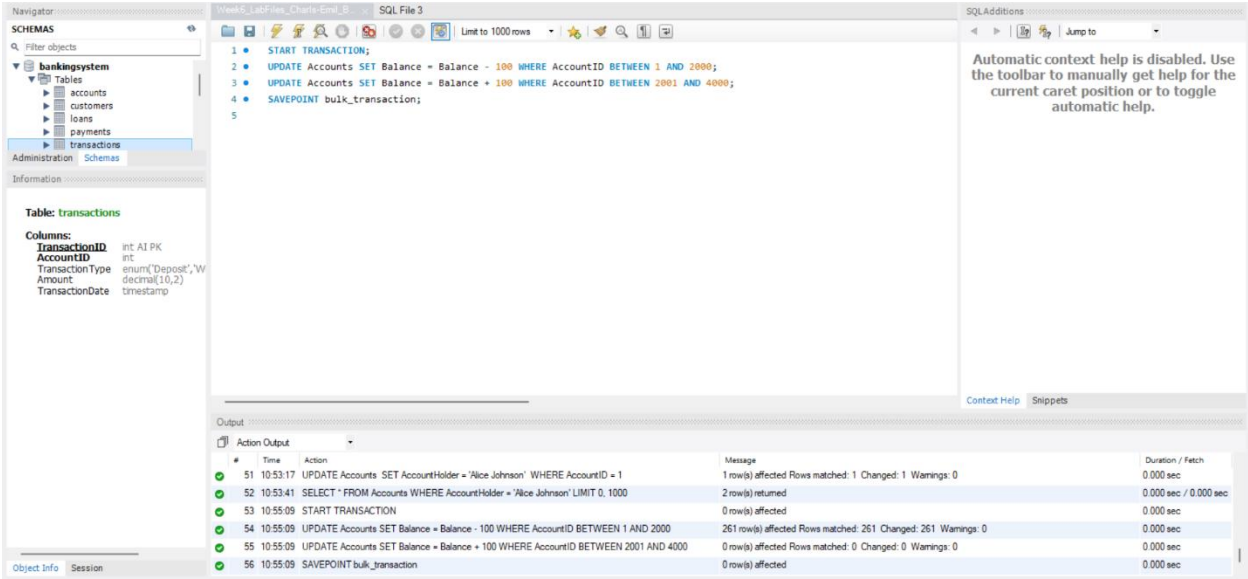
COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.



Title: Fetching Accounts Belonging to Alice Johnson

Description: This query retrieves all accounts associated with 'Alice Johnson', ensuring that account ownership details are correctly recorded.



Title: Bulk Transaction with Savepoint

Description: This transaction updates the balances of multiple accounts in bulk by deducting 100 from accounts with AccountID between 1 and 2000 and adding 100 to accounts with AccountID between 2001 and 4000. A savepoint named bulk_transaction is created to allow partial rollback if needed.



NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.

Navigator

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Table: accounts

Columns:

AccountID int AI PK

CustomerID int

AccountType enum('Savings','Checking','Loan','Credit','Other')

Balance decimal(10,2)

CreatedAt timestamp

AccountHolder varchar(255)

SQL File 3

1 SELECT * FROM Accounts WHERE AccountID BETWEEN 1 AND 5;

2

Result Grid

Filter Rows

Limit to 1000 rows

Export/Import

Wrap Cell Contents

Table: accounts

AccountID CustomerID AccountType Balance CreatedAt AccountHolder

1 200 Checking 15279.19 2025-03-02 10:23:48 Alice Johnson

2 224 Checking 59832.18 2025-03-02 10:23:48

3 151 Savings 72198.83 2025-03-02 10:23:48

4 170 Savings 43188.18 2025-03-02 10:23:48

5 28 Savings 71685.99 2025-03-02 10:23:48

Accounts 21

Apply

Revert

Context Help

Snippets

Output

Action Output

Time Action Message Duration / Fetch

52 10:53:41 SELECT * FROM Accounts WHERE AccountHolder = 'Alice Johnson' LIMIT 0, 1000 2 row(s) returned 0.000 sec / 0.000 sec

53 10:55:09 START TRANSACTION 0 row(s) affected 0.000 sec

54 10:55:09 UPDATE Accounts SET Balance = Balance - 100 WHERE AccountID BETWEEN 1 AND 2000 261 row(s) affected Rows matched: 261 Changed: 261 Warnings: 0 0.000 sec

55 10:55:09 UPDATE Accounts SET Balance = Balance + 100 WHERE AccountID BETWEEN 2001 AND 4000 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0 0.000 sec

56 10:55:09 SAVEPOINT bulk_transaction 0 row(s) affected 0.000 sec

57 10:55:53 SELECT * FROM Accounts WHERE AccountID BETWEEN 1 AND 5 LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec

Title: Checking Updated Account Balances

Description: This query retrieves and displays the balances of the first five accounts (AccountID 1 to 5) to verify the changes made in the bulk transaction before committing.

Navigator

SCHEMAS

Filter objects

bankingsystem

Tables

accounts

customers

loans

payments

transactions

Administration

Schemas

Information

Table: accounts

Columns:

AccountID int AI PK

CustomerID int

AccountType enum('Savings','Checking','Loan','Credit','Other')

Balance decimal(10,2)

CreatedAt timestamp

AccountHolder varchar(255)

SQL File 3

1 COMMIT;

2

Output

Action Output

Time Action Message Duration / Fetch

53 10:55:09 START TRANSACTION 0 row(s) affected 0.000 sec

54 10:55:09 UPDATE Accounts SET Balance = Balance - 100 WHERE AccountID BETWEEN 1 AND 2000 261 row(s) affected Rows matched: 261 Changed: 261 Warnings: 0 0.000 sec

55 10:55:09 UPDATE Accounts SET Balance = Balance + 100 WHERE AccountID BETWEEN 2001 AND 4000 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0 0.000 sec

56 10:55:09 SAVEPOINT bulk_transaction 0 row(s) affected 0.000 sec

57 10:55:53 SELECT * FROM Accounts WHERE AccountID BETWEEN 1 AND 5 LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec

58 10:57:39 COMMIT 0 row(s) affected 0.000 sec

Title: Committing Bulk Transaction

Description: This command commits the previously executed bulk transaction, permanently saving the changes made to the affected account balances.

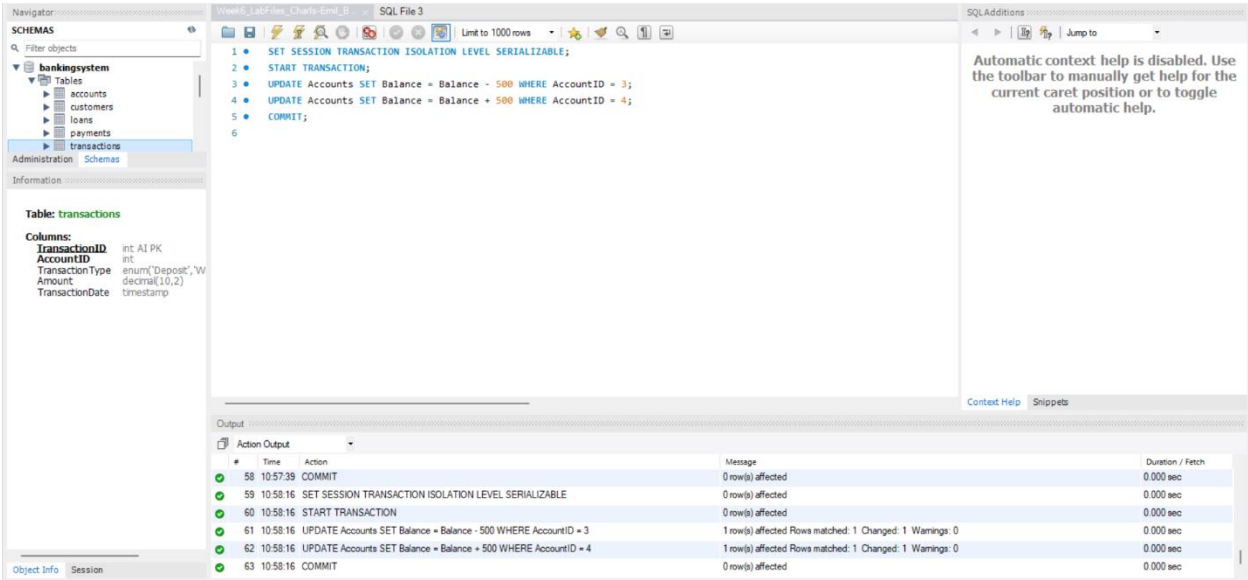


NAME: Charls Emil C. Barquin

SUBJECT: Information Management 1

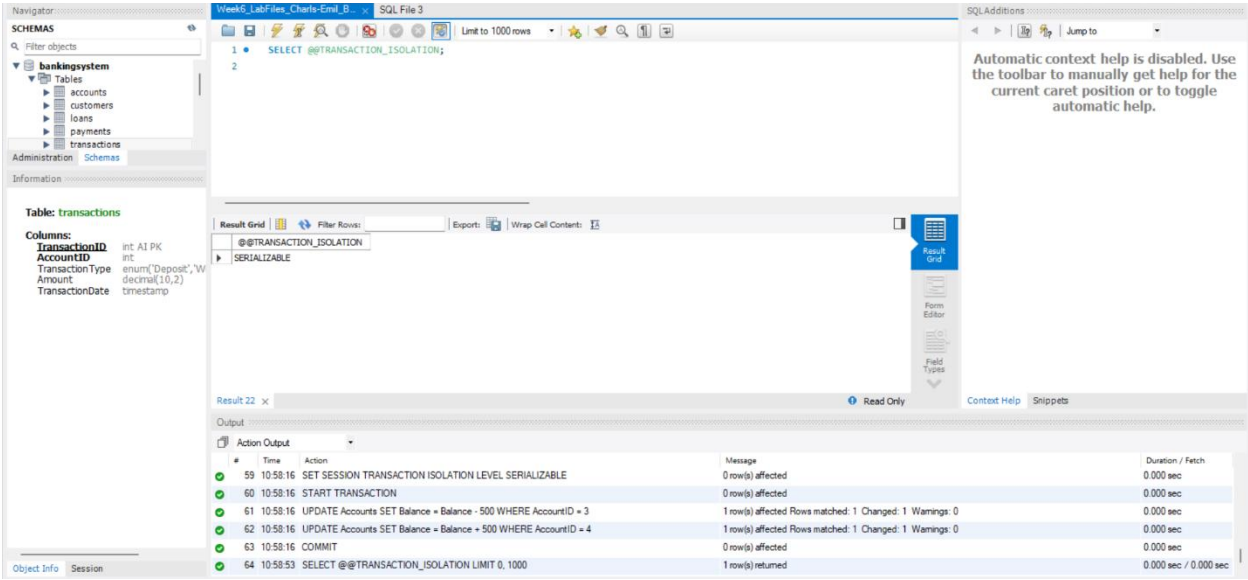
COURSE, YEAR & SECTION: BSIS-2A

PROFESSOR: Mr. Red Guillermo V. Jr.



Title: Ensuring Secure Fund Transfer with Serializable Isolation

Description: This query sequence sets the transaction isolation level to SERIALIZABLE, the strictest level, ensuring that transactions are executed in a fully isolated manner to prevent concurrency issues. It then starts a transaction to transfer ₱500 from AccountID = 3 to AccountID = 4. The balance is first deducted from Account 3 and then added to Account 4. Finally, the transaction is committed, making the changes permanent and ensuring data integrity.



Title: Checking the Current Transaction Isolation Level

Description: This query retrieves and displays the current transaction isolation level, confirming that the session is set to SERIALIZABLE as specified earlier.