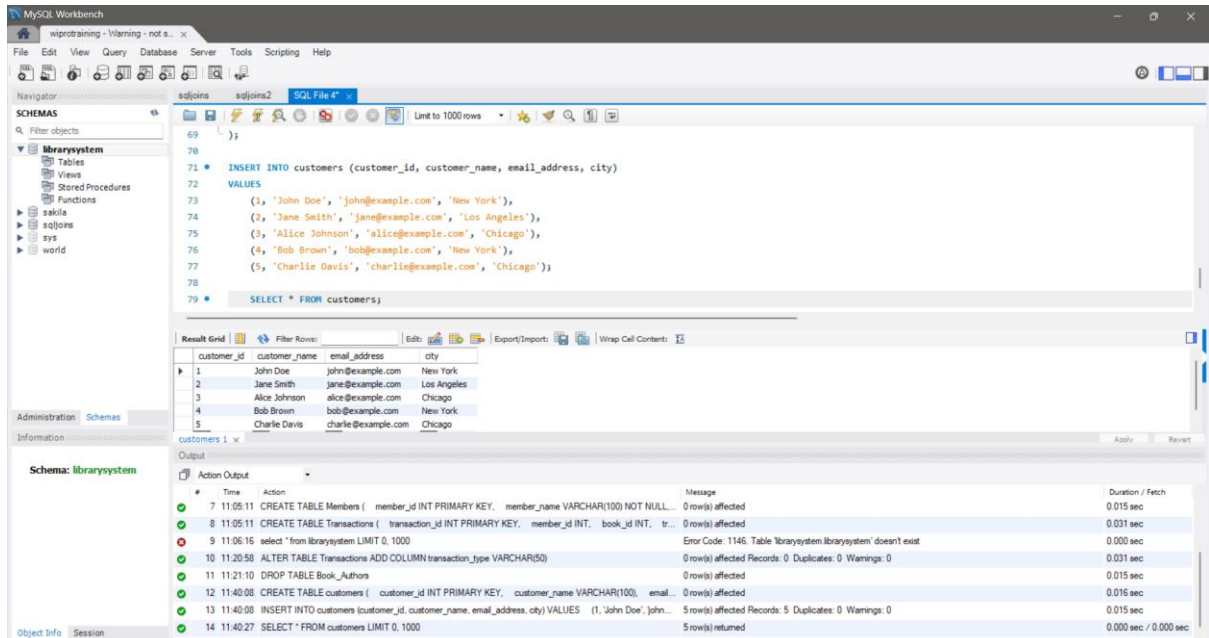


SQL Day 2 Assignments:

Assignment 1: Write a SELECT query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following code:

```
69
70
71 INSERT INTO customers (customer_id, customer_name, email_address, city)
72 VALUES
73 (1, 'John Doe', 'john@example.com', 'New York'),
74 (2, 'Jane Smith', 'jane@example.com', 'Los Angeles'),
75 (3, 'Alice Johnson', 'alice@example.com', 'Chicago'),
76 (4, 'Bob Brown', 'bob@example.com', 'New York'),
77 (5, 'Charlie Davis', 'charlie@example.com', 'Chicago');
78
79 SELECT * FROM customers;
```

The Results window shows the output of the query:

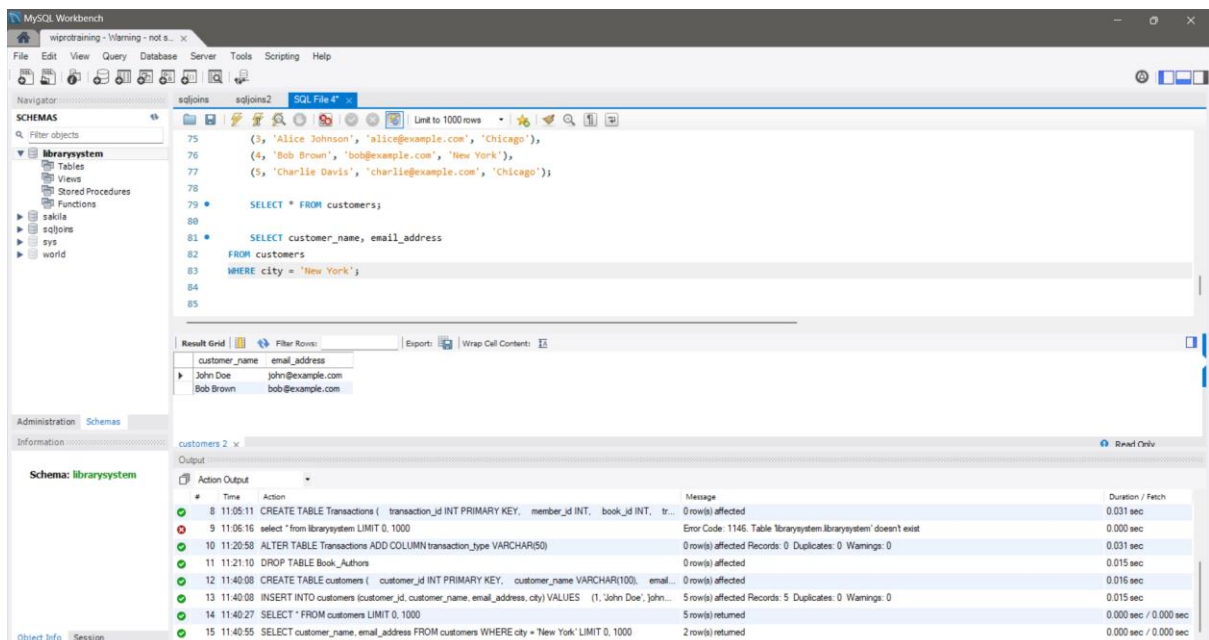
customer_id	customer_name	email_address	city
1	John Doe	john@example.com	New York
2	Jane Smith	jane@example.com	Los Angeles
3	Alice Johnson	alice@example.com	Chicago
4	Bob Brown	bob@example.com	New York
5	Charlie Davis	charlie@example.com	Chicago

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
7	11:05:11	CREATE TABLE Members (member_id INT PRIMARY KEY, member_name VARCHAR(100) NOT NULL...	0 row(s) affected	0.015 sec
8	11:05:11	CREATE TABLE Transactions (transaction_id INT PRIMARY KEY, member_id INT, book_id INT, tr...	0 row(s) affected	0.031 sec
9	11:06:16	select * from librarysystem LIMIT 0, 1000	Error Code: 1146: Table 'librarysystem.librarysystem' doesn't exist	0.000 sec
10	11:20:58	ALTER TABLE Transactions ADD COLUMN transaction_type VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
11	11:21:10	DROP TABLE Book_Authors	0 row(s) affected	0.015 sec
12	11:40:08	CREATE TABLE customers (customer_id INT PRIMARY KEY, customer_name VARCHAR(100), email...	0 row(s) affected	0.016 sec
13	11:40:08	INSERT INTO customers (customer_id, customer_name, email_address, city) VALUES (1, 'John Doe', John...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
14	11:40:27	SELECT * FROM customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Here's the SELECT query to retrieve all columns from the 'customers'

To modify it to return only the customer name and email address for customers in a specific city, let's assume the city is 'New York':



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following code:

```
75 (3, 'Alice Johnson', 'alice@example.com', 'Chicago'),
76 (4, 'Bob Brown', 'bob@example.com', 'New York'),
77 (5, 'Charlie Davis', 'charlie@example.com', 'Chicago');
78
79 SELECT * FROM customers;
80
81 SELECT customer_name, email_address
82 FROM customers
83 WHERE city = 'New York';
84
85
```

The Results window shows the output of the query:

customer_name	email_address
John Doe	john@example.com
Bob Brown	bob@example.com

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
8	11:05:11	CREATE TABLE Transactions (transaction_id INT PRIMARY KEY, member_id INT, book_id INT, tr...	0 row(s) affected	0.031 sec
9	11:06:16	select * from librarysystem LIMIT 0, 1000	Error Code: 1146: Table 'librarysystem.librarysystem' doesn't exist	0.000 sec
10	11:20:58	ALTER TABLE Transactions ADD COLUMN transaction_type VARCHAR(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
11	11:21:10	DROP TABLE Book_Authors	0 row(s) affected	0.015 sec
12	11:40:08	CREATE TABLE customers (customer_id INT PRIMARY KEY, customer_name VARCHAR(100), email...	0 row(s) affected	0.016 sec
13	11:40:08	INSERT INTO customers (customer_id, customer_name, email_address, city) VALUES (1, 'John Doe', John...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
14	11:40:27	SELECT * FROM customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
15	11:40:55	SELECT customer_name, email_address FROM customers WHERE city = 'New York' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

This query selects the 'customer_name' and 'email_address' columns from the 'customers' table, but only for customers located in the city of 'New York'. Adjust the city name as needed to retrieve data for customers in a different city.

Assignment 3: Explain the ACID properties of a transaction in your own words.

Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

Understanding ACID Properties:

ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. These properties ensure the reliability, integrity, and consistency of database transactions.

1. **Atomicity:** Atomicity guarantees that a transaction is treated as a single unit of work. It either completes entirely or is rolled back to its original state if any part of it fails. In other words, all changes made by the transaction are applied together or not at all, preventing partial updates that could leave the database in an inconsistent state.
2. **Consistency:** Consistency ensures that the database remains in a valid state before and after the transaction. Transactions must adhere to all defined constraints, rules, and relationships within the database. Any changes made by a transaction must maintain the overall integrity and correctness of the database schema.
3. **Isolation:** Isolation refers to the degree to which the changes made by one transaction are visible to other transactions executing concurrently. Isolation prevents interference between concurrent transactions, ensuring that each transaction sees a consistent snapshot of the database and operates independently of other transactions.
4. **Durability:** Durability guarantees that once a transaction is committed, its changes are permanently saved and will not be lost, even in the event of a system failure or crash. The database system must ensure that committed

transactions are durable and remain intact, providing reliability and data persistence.

Execution of the SQL commands:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following commands:

```
54
55 -- Begin the transaction
56 • START TRANSACTION;
57
58 -- Set the isolation level to SERIALIZABLE
59 • SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
60
61 -- Retrieve and lock the row to be updated
62 • SELECT * FROM orders WHERE order_id = 1 FOR UPDATE;
63
```

The Result Grid shows the following data:

order_id	customer_id	order_date	total_amount
1	1	2024-05-01	100.00

The Action Output pane shows the execution log:

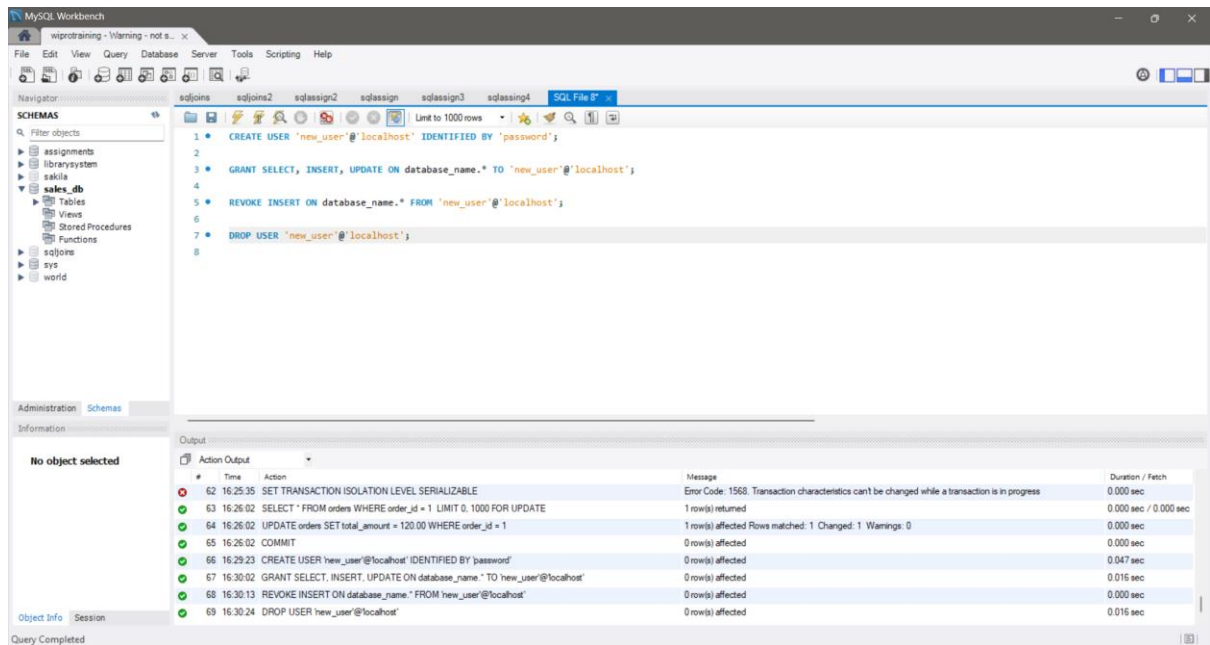
#	Time	Action	Message	Duration / Fetch
58	16:25:11	START TRANSACTION	0 row(s) affected	0.000 sec
59	16:25:11	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	Error Code: 1568, Transaction characteristics can't be changed while a transaction is in progress	0.000 sec
60	16:25:24	START TRANSACTION	0 row(s) affected	0.000 sec
61	16:25:29	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	Error Code: 1568, Transaction characteristics can't be changed while a transaction is in progress	0.000 sec
62	16:25:35	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	Error Code: 1568, Transaction characteristics can't be changed while a transaction is in progress	0.000 sec
63	16:26:02	SELECT * FROM orders WHERE order_id = 1 LIMIT 0, 1000 FOR UPDATE	1 row(s) returned	0.000 sec / 0.000 sec
64	16:26:02	UPDATE orders SET total_amount = 120.00 WHERE order_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
65	16:26:02	COMMIT	0 row(s) affected	0.000 sec

In this example, we start a transaction, set the isolation level to SERIALIZABLE, retrieve and lock the row to be updated using the SELECT...FOR UPDATE statement, update the total amount of the order, and finally commit the transaction.

To demonstrate different isolation levels, you can modify the SET TRANSACTION ISOLATION LEVEL statement to set the isolation level to READ UNCOMMITTED, READ COMMITTED, or REPEATABLE READ, and observe how each level affects the concurrency control and visibility of changes made by concurrent transactions.

Assignment 6: Create a new database user with specific privileges using the CREATE USER and GRANT commands. Then, write a script to REVOKE certain privileges and DROP the user.

Execution of the query:



The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view containing 'assignments', 'librarysystem', 'sakila', 'sales_db', 'Tables', 'Views', 'Stored Procedures', 'Functions', 'sqljoins', 'sys', and 'world'. The main editor window shows a script with the following SQL commands:

```
1 CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'password';
2
3 GRANT SELECT, INSERT, UPDATE ON database_name.* TO 'new_user'@'localhost';
4
5 REVOKE INSERT ON database_name.* FROM 'new_user'@'localhost';
6
7 DROP USER 'new_user'@'localhost';
8
```

The bottom panel shows the 'Output' tab with a table of execution results:

#	Time	Action	Message	Duration / Fetch
62	16:25:35	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE	Error Code: 1568. Transaction characteristics can't be changed while a transaction is in progress	0.000 sec
63	16:26:02	SELECT * FROM orders WHERE order_id = 1 LIMIT 0, 1000 FOR UPDATE	1 row(s) returned	0.000 sec / 0.000 sec
64	16:26:02	UPDATE orders SET total_amount = 120.00 WHERE order_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
65	16:26:02	COMMIT	0 row(s) affected	0.000 sec
66	16:29:23	CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'password'	0 row(s) affected	0.047 sec
67	16:30:02	GRANT SELECT, INSERT, UPDATE ON database_name.* TO 'new_user'@'localhost'	0 row(s) affected	0.016 sec
68	16:30:13	REVOKE INSERT ON database_name.* FROM 'new_user'@'localhost'	0 row(s) affected	0.000 sec
69	16:30:24	DROP USER 'new_user'@'localhost'	0 row(s) affected	0.016 sec

The status bar at the bottom indicates 'Query Completed'.

Assignment 7: Prepare a series of SQL statements to INSERT new records into the library tables, UPDATE existing records with new information, and DELETE records based on specific criteria. Include BULK INSERT operations to load data from an external source.

Execution of the query:

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

```

19 CREATE TABLE members (
20     member_id INT AUTO_INCREMENT PRIMARY KEY,
21     member_name VARCHAR(255) NOT NULL,
22     email VARCHAR(255)
23 );
24
25 -- Insert authors into the 'authors' table
26 INSERT INTO authors (author_name, birth_year) VALUES
27 ('F. Scott Fitzgerald', 1896),
28 ('George Orwell', 1903),
29 ('Jane Austen', 1775);
30
31 -- Insert books into the 'books' table
32 INSERT INTO books (title, author_id, published_year, category) VALUES
33 ('The Great Gatsby', 1, 1925, 'Classic'),
34 ('1984', 2, 1949, 'Dystopian'),
35 ('Pride and Prejudice', 3, 1813, 'Romance');
36
37 -- Update the published year of a book
38 UPDATE books
39 SET published_year = 2021
40 WHERE title = 'The Great Gatsby';
41
42 -- Update the email of a member
43 UPDATE members
44 SET email = 'johndoe@example.com'
45 WHERE member_name = 'John Doe';
46
47

```

Output:

#	Time	Action	Message	Duration / Fetch
71	16:33:50	CREATE TABLE books (book_id INT AUTO_INCREMENT PRIMARY KEY, title VARCHAR(255) NOT N...	Error Code: 1024. Failed to open the referenced table 'authors'	0.000 sec
72	16:34:03	CREATE TABLE authors (author_id INT AUTO_INCREMENT PRIMARY KEY, author_name VARCHAR(255) NOT N...	0 row(s) affected	0.016 sec
73	16:34:14	CREATE TABLE books (book_id INT AUTO_INCREMENT PRIMARY KEY, title VARCHAR(255) NOT N...	0 row(s) affected	0.031 sec
74	16:34:17	CREATE TABLE members (member_id INT AUTO_INCREMENT PRIMARY KEY, member_name VARC...	0 row(s) affected	0.016 sec
75	16:34:34	INSERT INTO books (title, author_id, published_year, category) VALUES ('The Great Gatsby', 1, 1925, 'Classic')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('sales_db`.`books`, CONSTR...	0.000 sec
76	16:34:49	INSERT INTO books (title, author_id, published_year, category) VALUES ('The Great Gatsby', 1, 1925, 'Classic')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('sales_db`.`books`, CONSTR...	0.016 sec
77	16:35:32	INSERT INTO authors (author_name, birth_year) VALUES ('F. Scott Fitzgerald', 1896), ('George Orwell', 1903), ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.016 sec
78	16:35:43	INSERT INTO books (title, author_id, published_year, category) VALUES ('The Great Gatsby', 1, 1925, 'Classic', ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec