Library Database Business Requirements Document (BRD)

1. Introduction

The purpose of this document is to outline the business requirements for the development of a library management database system. This system aims to facilitate the library's operations, including book borrowing, genre and author management, staff responsibilities, and fine calculations for overdue books. The document specifies functional and non-functional requirements, assumptions, and constraints to ensure the system meets the library's needs effectively.

2. Business Goals

- To enable efficient borrowing and returning of books by members.
- To maintain accurate and up-to-date information about books, authors, genres, and loans.
- To ensure proper management of fines for overdue books.
- To enforce rules for membership registration and staff responsibilities.
- To support multi-author and multi-genre book classifications.

3. Functional Requirements

1. **Membership Management**:

    a. Members can borrow any number of books or none at all. Borrowing is optional and unrestricted in quantity.
    b. Members must provide a valid email or phone number to register.
2. **Book and Genre Management**:
    a. Each genre must have at least one associated book. Genres without books will be automatically removed from the database.
    b. Books can belong to multiple genres, and each genre can have any number of books.
3. **Author Management**:
    a. Each author must have at least one associated book. Authors without books will be automatically removed from the database.
    b. Books can be written by multiple authors, and each author can write multiple books.
4. **Loan Management**:
    a. A book can only be borrowed by one member at a time. Multiple members cannot borrow the same copy of a book simultaneously.
    b. Each loan is managed by one staff member.
    c. Members must return books within 14 days of the loan date.
5. **Fine Management**:
    a. Fines for overdue books are calculated at a fixed rate of $12 per day starting from the 15th day.
6. **Staff Management**:

a. Staff members can manage at least one loan and at most a defined maximum number of loans.
b. Each staff member must have a hire date and a valid phone number.

4. Assumptions and Constraints

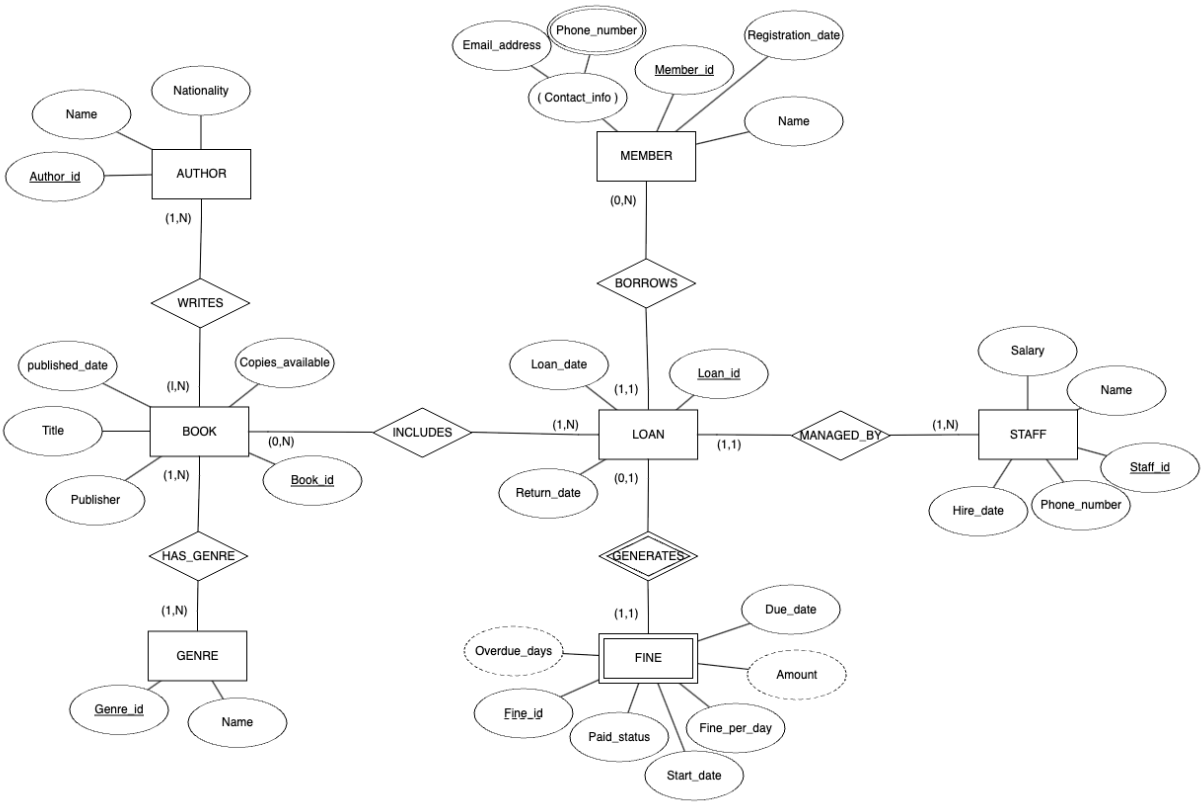
- Assumptions:
  - Members will return borrowed books in good condition.
  - Staff will only manage loans within their work hours.
  - Genres and authors will only be removed automatically when no books are associated with them.
- Constraints:
  - The library has a fixed fine rate of $12 per day for overdue books.
  - Books must be returned within 14 days to avoid fines.
  - Each staff member has a predefined maximum number of loans they can manage at one time.

6. Notes

- **Book**: An item available for borrowing.
- **Member**: A registered user who can borrow books from the library.
- **Loan**: A record of a book borrowed by a member.
- **Genre**: A category or classification of books (Fiction, Non-Fiction).
- **Author**: The writer or contributor of a book.
- **Fine**: A monetary penalty for overdue books.

- **Staff Member**: An employee responsible for managing loans and other library operations.

ERD:

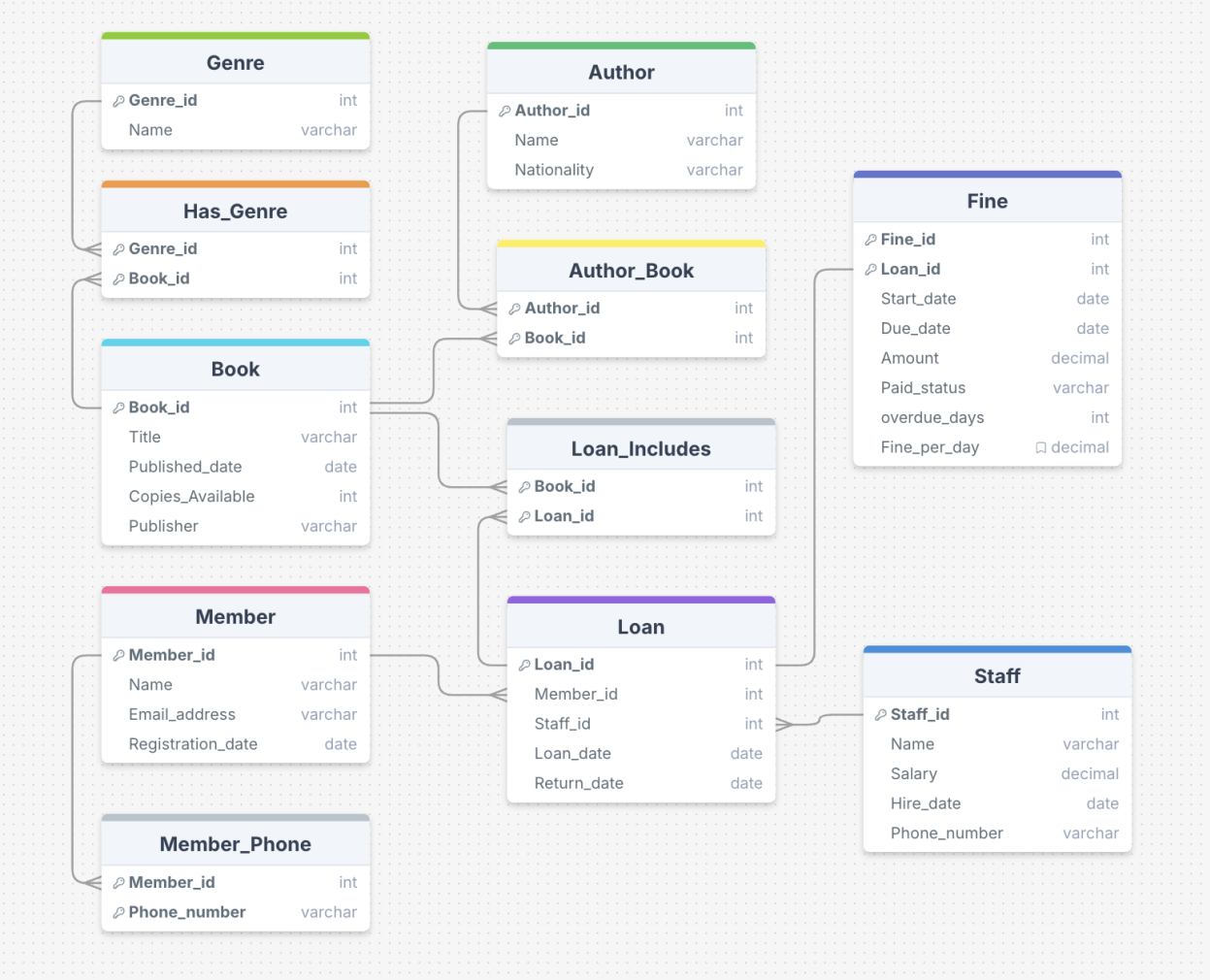


Database Design Document

Introduction:

The purpose of this document is to provide a detailed design for the Library Database system, which is intended to support the library's operational needs. The database will manage information about members, books, authors, genres, staff, loans, and fines. And will explain the relationship between them.

Schema:

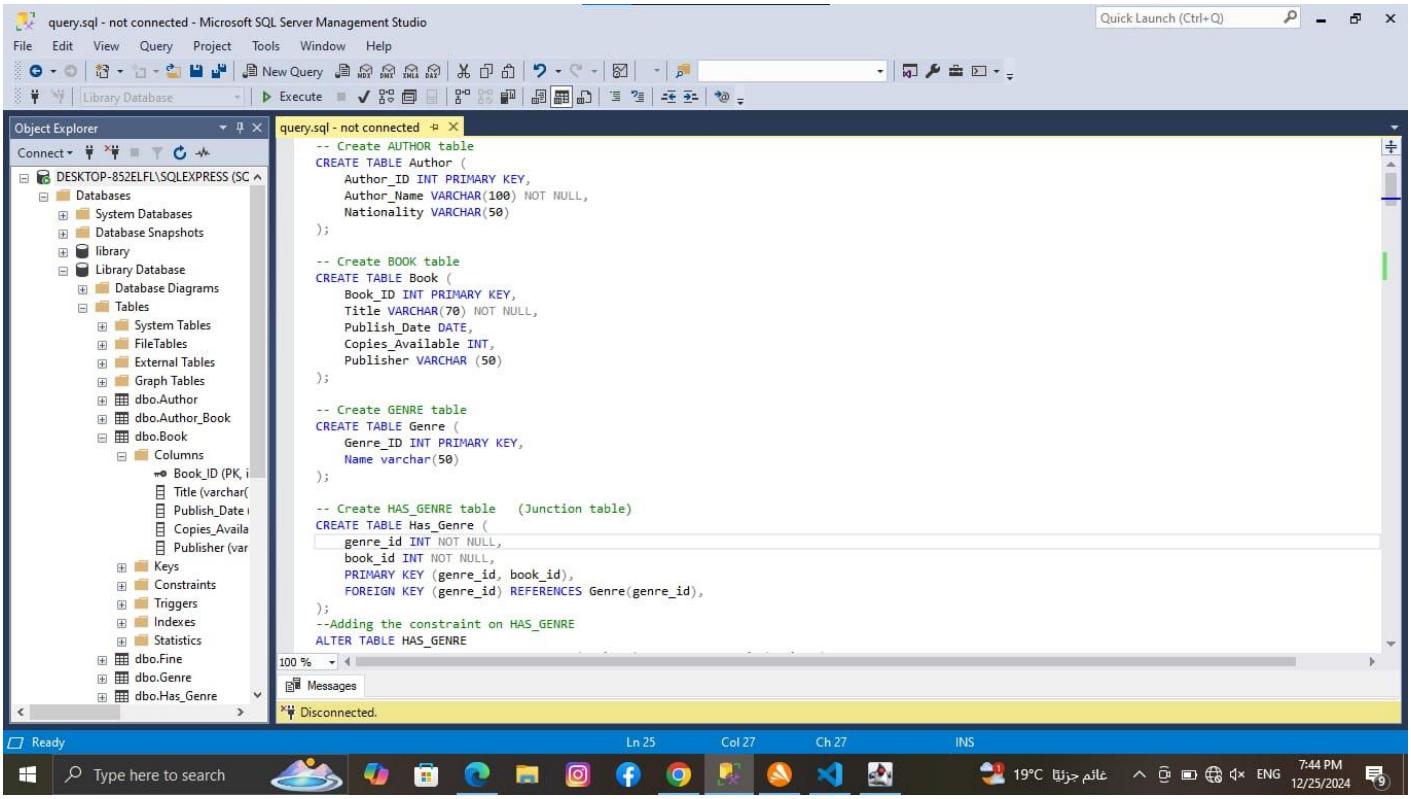


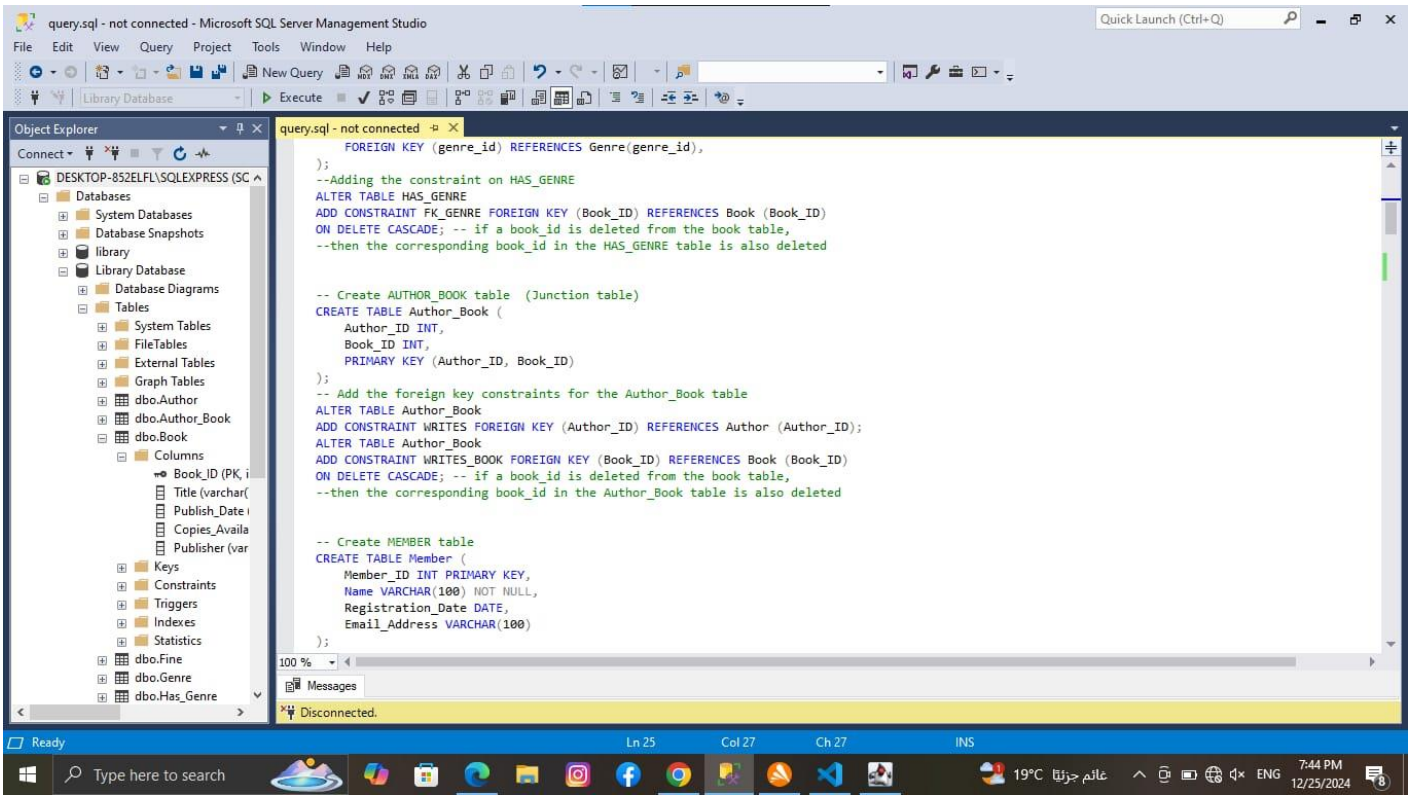
Assumptions:

1. *The "phone number" is a multi-valued attribute. Therefore, a separated table is created to store "member id" and "phone number", where "member id" and "phone number" is a composite primary key allowing a single member to have multiple phone numbers without worrying about duplications that might violate key constraints.*
2. *Since the relationship between the entity "genre" and the entity "book" is many-to-many, a junction table called "has genre" is created to hold the "book id" and "genre id" as foreign keys and they are also a composite primary key to manage the relationship between the two entities without violating key constraints.*
3. *Since the relationship between the entity "book" and the entity "author" is many-to-many, a junction table called "Author_Book" is created to hold the "book id" and "author id" as foreign keys and they are also a composite primary key to manage the relationship between the two entities without violating key constraints.*
4. *Since the relationship between the entity "book" and the entity "loan" is many-to-many, a junction table called "Loan_Includes" is created to hold the "book id" and "loan id" as foreign keys and they are also a composite primary key to manage the relationship between the two entities without violating key constraints.*

5. *A constraint called "ON DELETE CASCADE" was added to the foreign key in the "**Author_Book**" table, which is a junction table between the "**Book**" and "**Author**" tables. This ensures that when a book is deleted from the "**Book**" table, all corresponding records in the "**Author_Book**" table with the matching Book_ID will also be deleted, maintaining data integrity.*
6. *Similarly, the same "ON DELETE CASCADE" constraint was applied to the foreign key in the "**Has_Genre**" table, ensuring that if a book is deleted from the "**Book**" table, all corresponding records in the "**Has_Genre**" table with the matching "Book_ID" will also be deleted.*

SQL Script



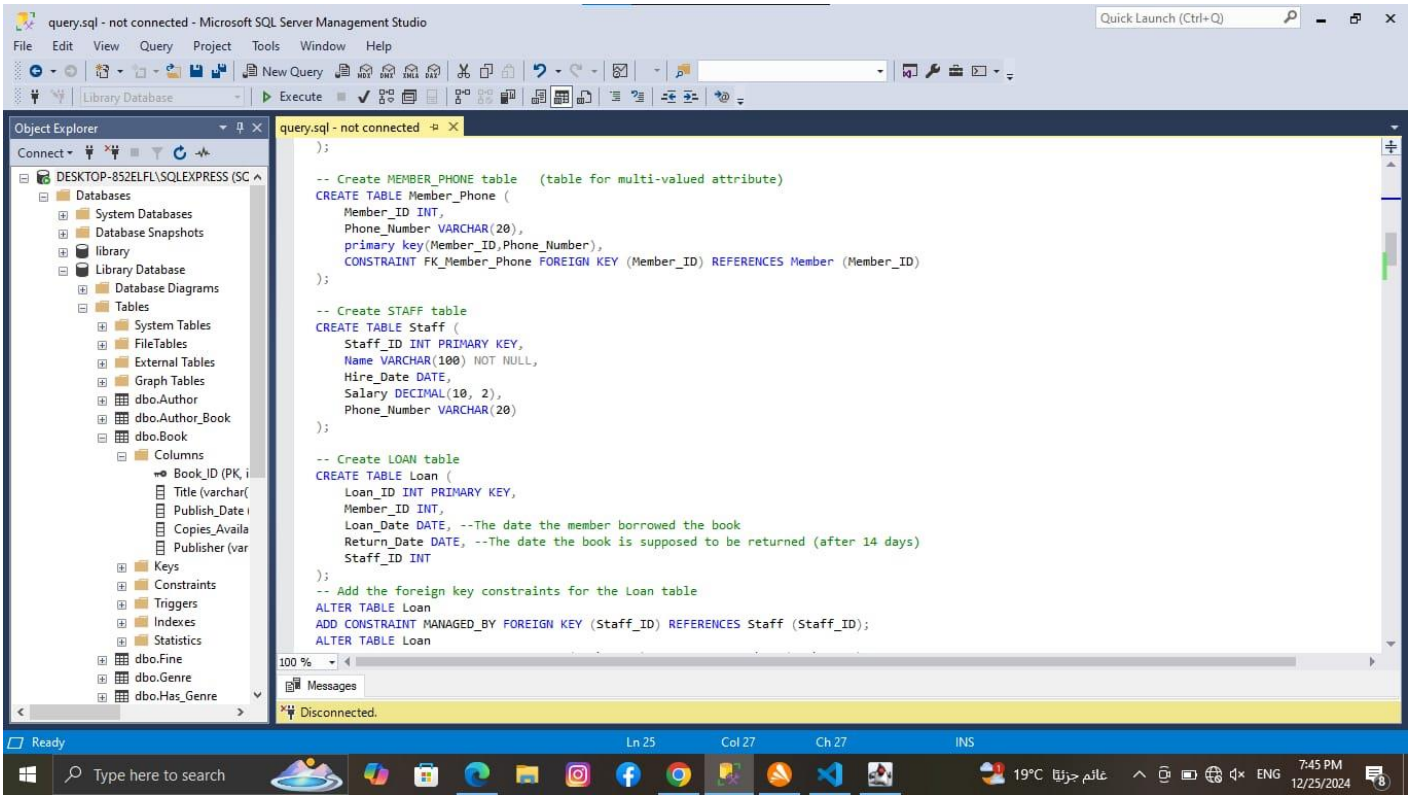


We began by creating the necessary tables for the library database using the (CREATE TABLE) command. First, we defined the (Author) table with the

attributes (Author_ID, Author_Name, and Nationality), and set (Author_ID) as the primary key. Next, we followed a similar process to define the remaining tables.

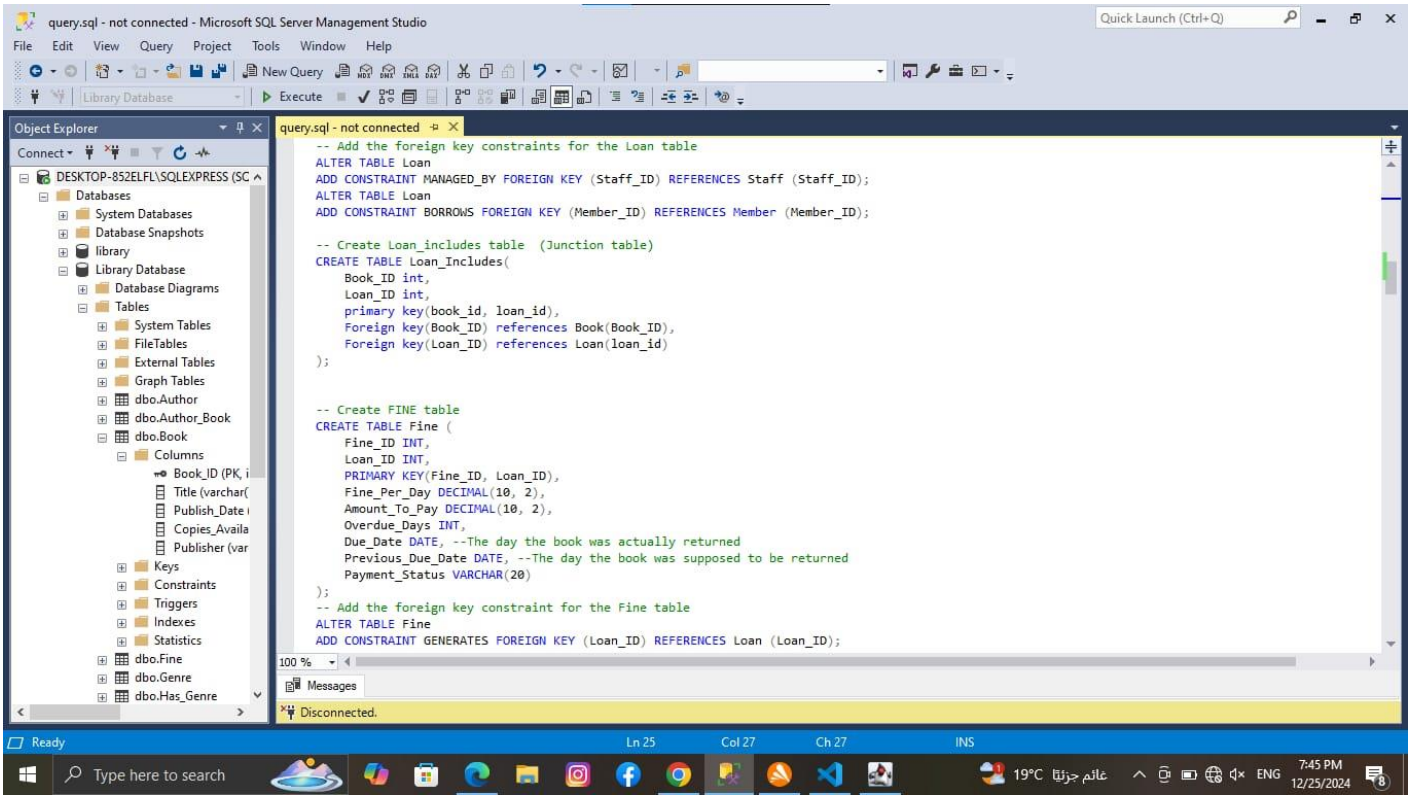
Since the relationship between the Genre and the Book table is many to many, we created the junction table (Has_Genre) to link the tables with a composite key of both their primary keys (genre_id) and (book_id) and we added a foreign key constraint to link the genre_id in the Has_Genre table to the genre_id in the Genre table using (FOREIGN KEY_REFERENCES) clause. Then, we added a foreign key constraint to the HAS_GENRE table to link the Book_ID attribute from the HAS_GENRE table to the Book_ID in the Book table, and we added the ON DELETE CASCADE here as well to ensure that if a book is deleted from the Book table, all corresponding entries in the Book_Genre table with that Book_ID will also be automatically deleted. Preventing having any genres that don't have any corresponding books.z

Since there is a many-to-many relationship between the Author and Book tables, we created the Author_Book table as a junction table. This table has a composite key consisting of the primary keys from both the Author and Book tables, to link authors to the books they have written. Then, we added foreign key constraints to the Author_Book table: 1. WRITES constraint: This links the Author_ID attribute from the Author_Book table to the Author_ID in the Author table to make sure that each record in the Author_Book table refers to a specific author. 2. WRITES_BOOK constraint: This links the Book_ID attribute from the Author_Book table to the Book_ID in the Book table. Then we used the ON DELETE CASCADE command so that if a book is deleted from the Book table, all corresponding records in the Author_Book table with that Book_ID will be automatically deleted. This helps maintain referential integrity by removing the relationship between authors and books when the book is removed.



Since the phone_number is a multivalued attribute in the Member table, we created the Member_Phone table to allow members to have multiple phone numbers associated with their accounts. We then set the PRIMARY KEY to be a composite key of both the Member_ID and the phone_number. We also added a foreign key constraint to reference the Member_ID from the Member table, establishing a relationship between the two tables.

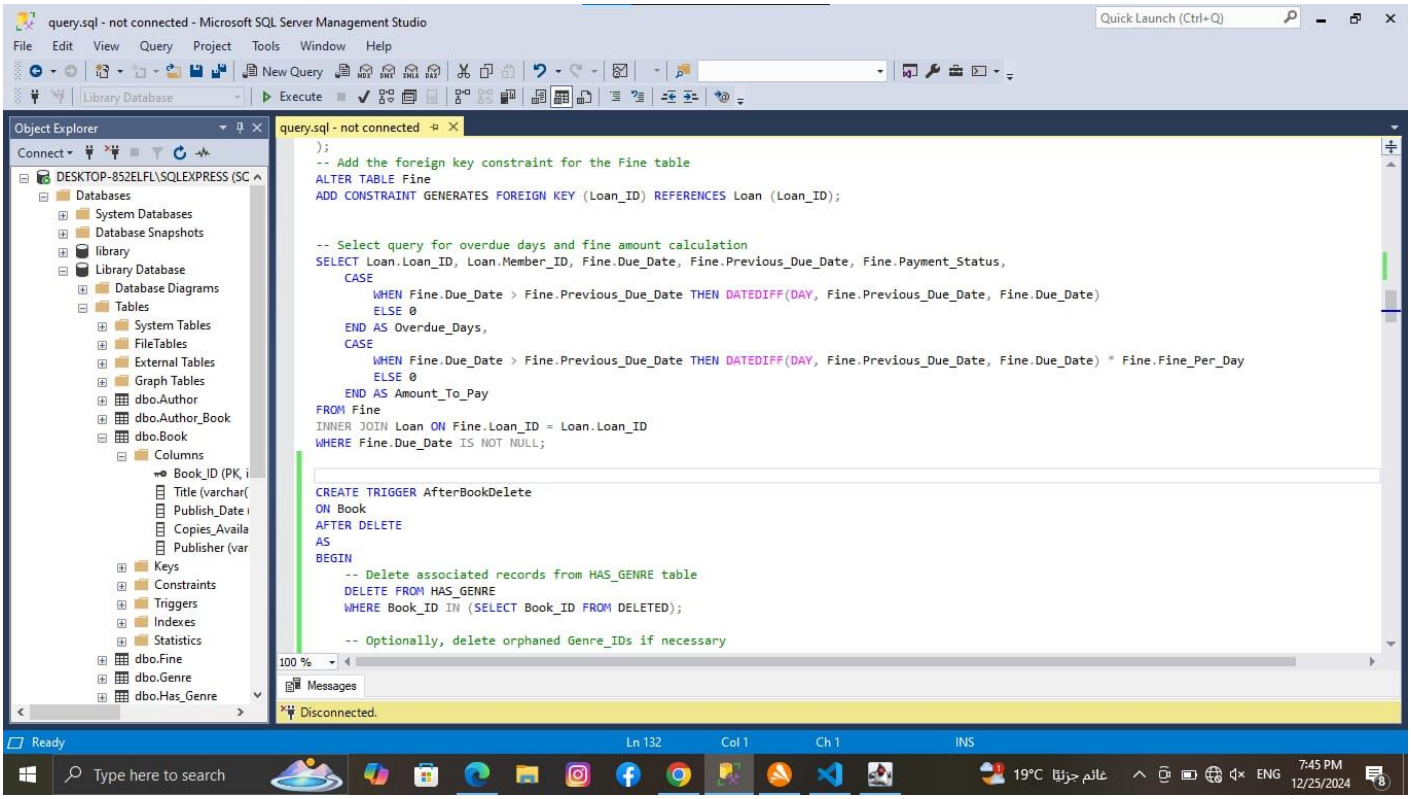
Here, we created the Loan table and set the PRIMARY KEY to be the Loan_ID, then we added the foreign-key constraints to the table to create those relationships: 1. MANAGED_BY constraint: This links the Staff_ID from the Staff table to the Staff_ID in the Loan table. It establishes that each loan is managed by a staff member. 2. BORROWS constraint: This links the Member_ID attribute from the Member table to the Member_ID in the Loan table to make sure that every loan is associated with a specific member.



We created the Loan_Includes table as a junction table to handle the many-to-many relationship between the Book and Loan tables. Which allows us to track which books are included in which loans.

Since the Fine table represents a weak entity, and its identifying entity is Loan, we created the Fine table with a composite primary key (Fine_ID, Loan_ID). Which makes sure that each Fine record is uniquely identified by the combination of both the Fine_ID and the Loan_ID.

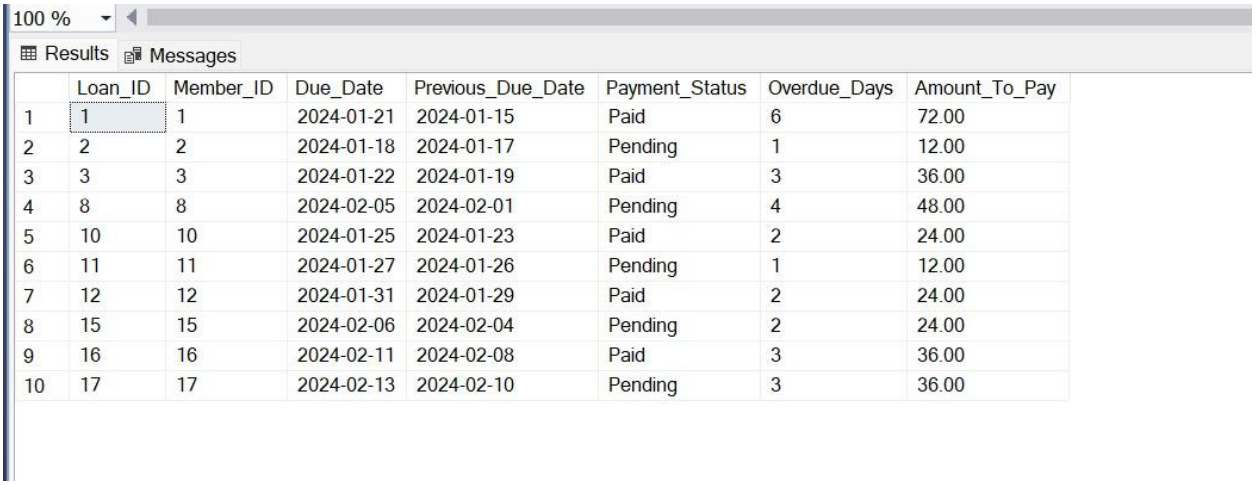
Then we added the foreign key (GENERATES) constraint which links the Loan_ID attribute from the Loan table to the Loan_ID in the Fine table to ensure that each fine is associated with a specific loan.



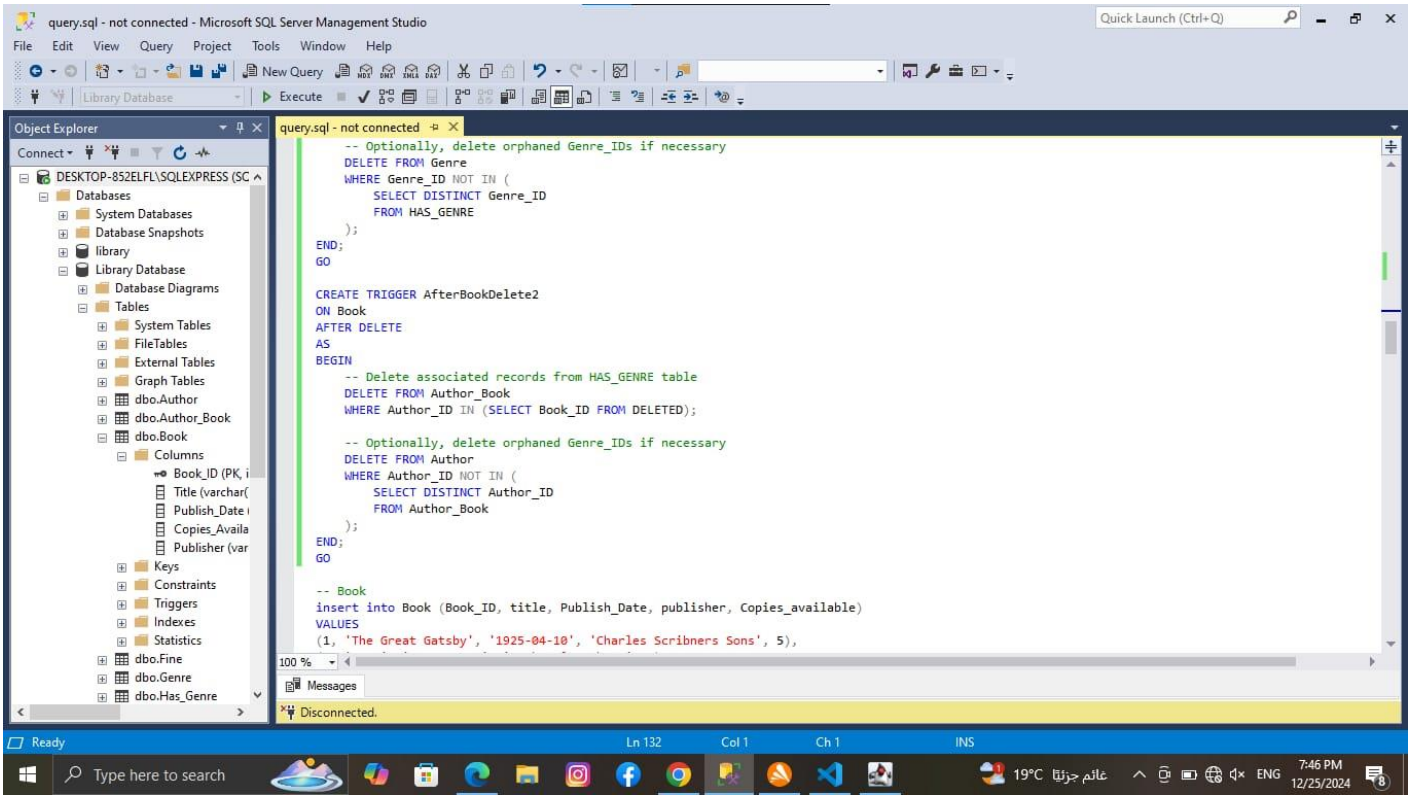
Since the Overdue_Days and Amount_To_Pay attributes in the Fine table are derived, we calculated them using the following query. First, we retrieved the Loan_ID and Member_ID from the Loan table, along with the Due_Date, Previous_Due_Date, and Payment_Status from the Fine table using a SELECT statement. We then used a CASE-WHEN clause to apply a condition: if the Due_Date (the actual date the book was returned) is later than the Previous_Due_Date (the expected return date, which is 14 days after the loan date), it indicates that the book was returned late. In this case, we calculate the Overdue_Days by using the DATEDIFF function, which subtracts the Previous_Due_Date from the Due_Date to determine the number of overdue days. This value is stored in the Overdue_Days attribute. Finally, we calculate the Amount_To_Pay by multiplying the Fine_Per_Day by the Overdue_Days, determining the total fine amount for the member to pay. Then we used the INNER JOIN to combine the Fine table with the Loan table using the Loan_ID attribute as the common link. We ensured that only records with matching Loan_ID values in both tables were included in the results, as INNER JOIN only includes rows that have corresponding matches in both tables. Then we added a WHERE condition to make sure we only

include fines where the Due_Date is not null, meaning the book has actually been returned.

And this is the output of the query:



	Loan_ID	Member_ID	Due_Date	Previous_Due_Date	Payment_Status	Overdue_Days	Amount_To_Pay
1	1	1	2024-01-21	2024-01-15	Paid	6	72.00
2	2	2	2024-01-18	2024-01-17	Pending	1	12.00
3	3	3	2024-01-22	2024-01-19	Paid	3	36.00
4	8	8	2024-02-05	2024-02-01	Pending	4	48.00
5	10	10	2024-01-25	2024-01-23	Paid	2	24.00
6	11	11	2024-01-27	2024-01-26	Pending	1	12.00
7	12	12	2024-01-31	2024-01-29	Paid	2	24.00
8	15	15	2024-02-06	2024-02-04	Pending	2	24.00
9	16	16	2024-02-11	2024-02-08	Paid	3	36.00
10	17	17	2024-02-13	2024-02-10	Pending	3	36.00



we created a trigger named AfterBookDelete. This trigger ensures that the database maintains integrity when a record is deleted from the Book table.

The trigger is activated immediately after a DELETE operation occurs on the Book table.

It automatically removes related records from the HAS_GENRE table to prevent invalid references (enforce referential integrity) and clean up the Genre table by deleting any genres that are no longer associated with any books.

When a book is deleted from the Book table, its Book_ID is temporarily stored in the virtual DELETED table. The DELETE statement removes all records from the HAS_GENRE table where the Book_ID matches any Book_ID in the DELETED table.

This ensures that the HAS_GENRE table does not contain references to books that no longer exist.

This second step ensures the Genre table does not contain "orphaned" genres (genres not associated with any books). The DELETE statement removes all genres from the Genre table where the Genre_ID is not found in the HAS_GENRE table.

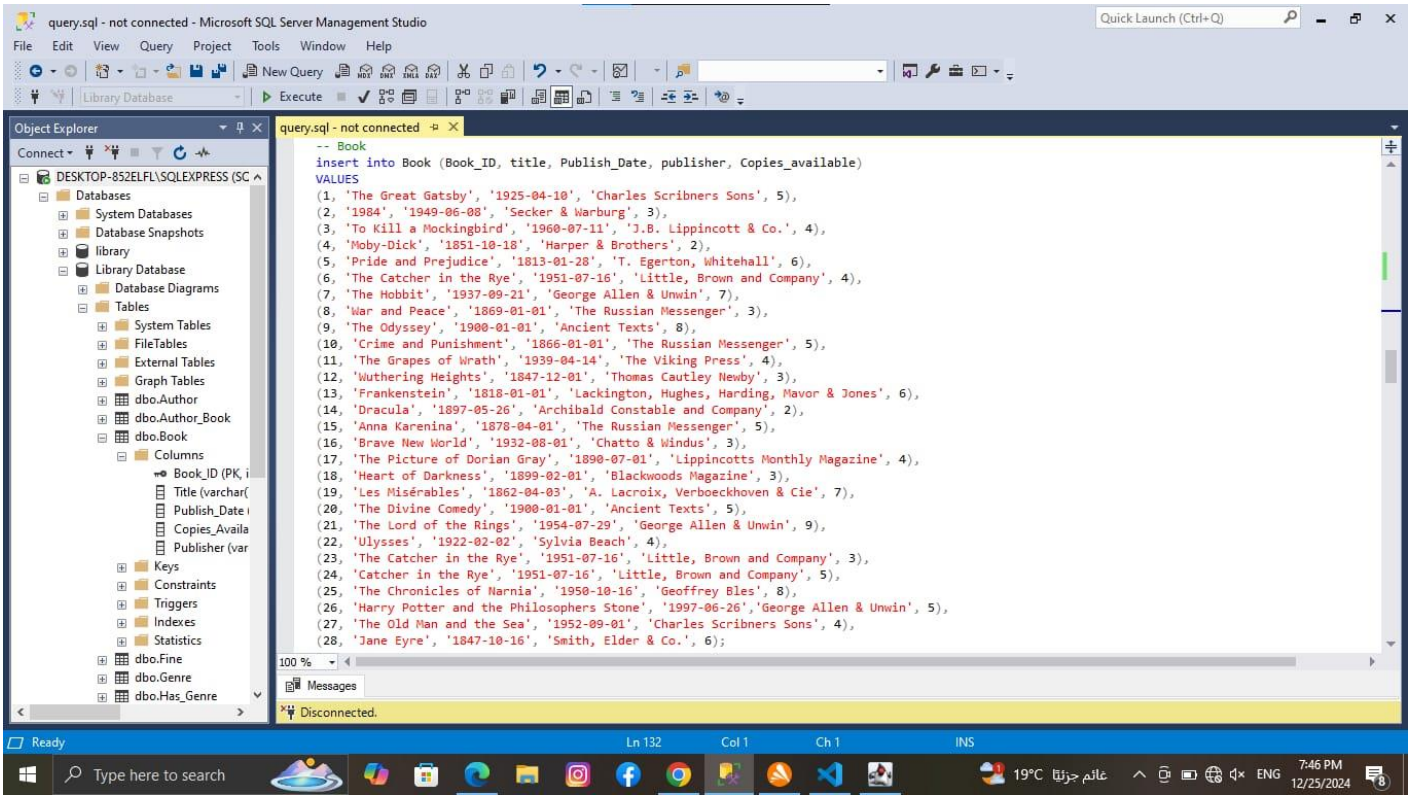
We created another trigger named AfterBookDelete2, which is designed to maintain the integrity of the relationships between the Book, Author_Book, and Author tables in a database. This trigger is executed automatically after a record is deleted from the Book table.

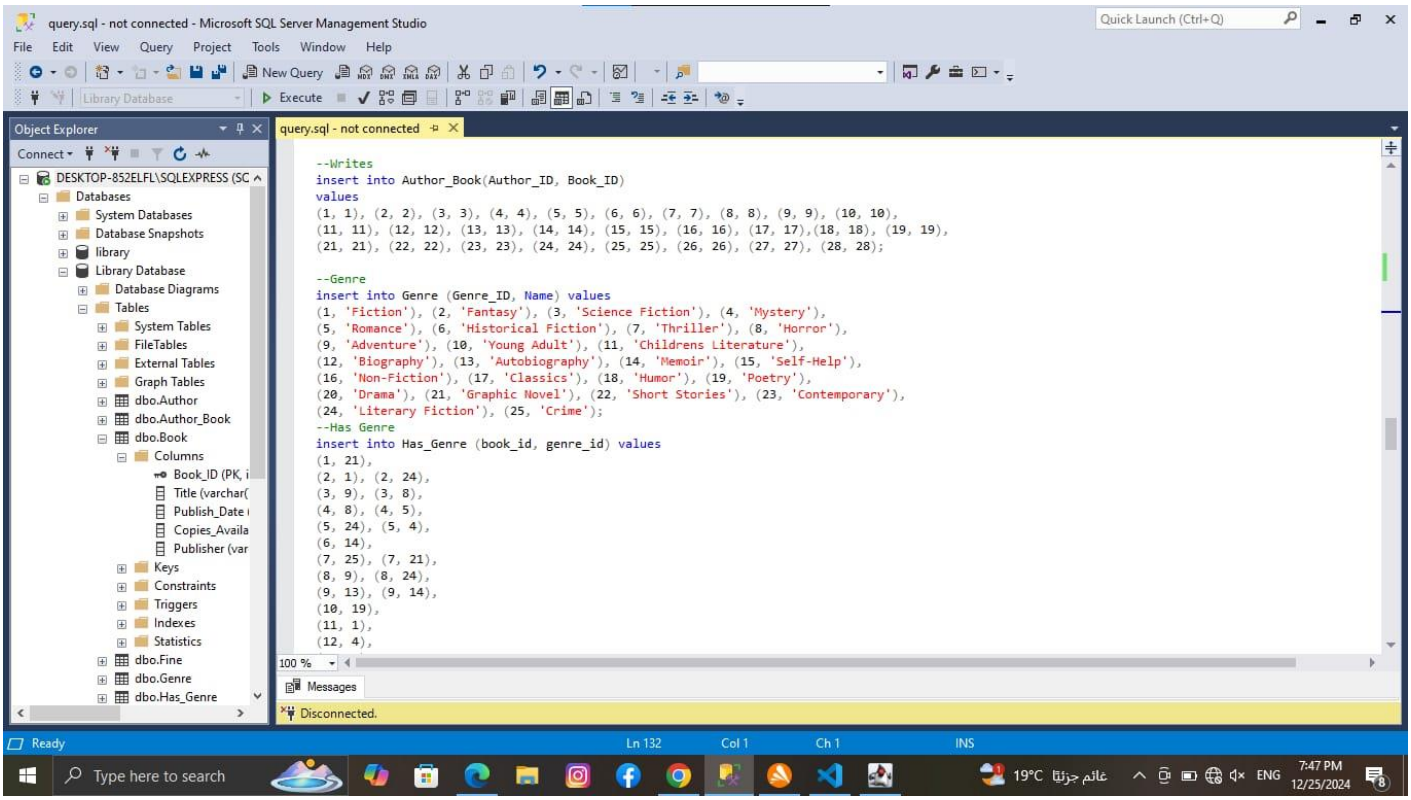
This trigger is activated immediately after a DELETE operation occurs on the Book table.it automatically removes related records from the Author_Book table when a book is deleted and cleans up the Author table by deleting any authors who are no longer associated with any books.

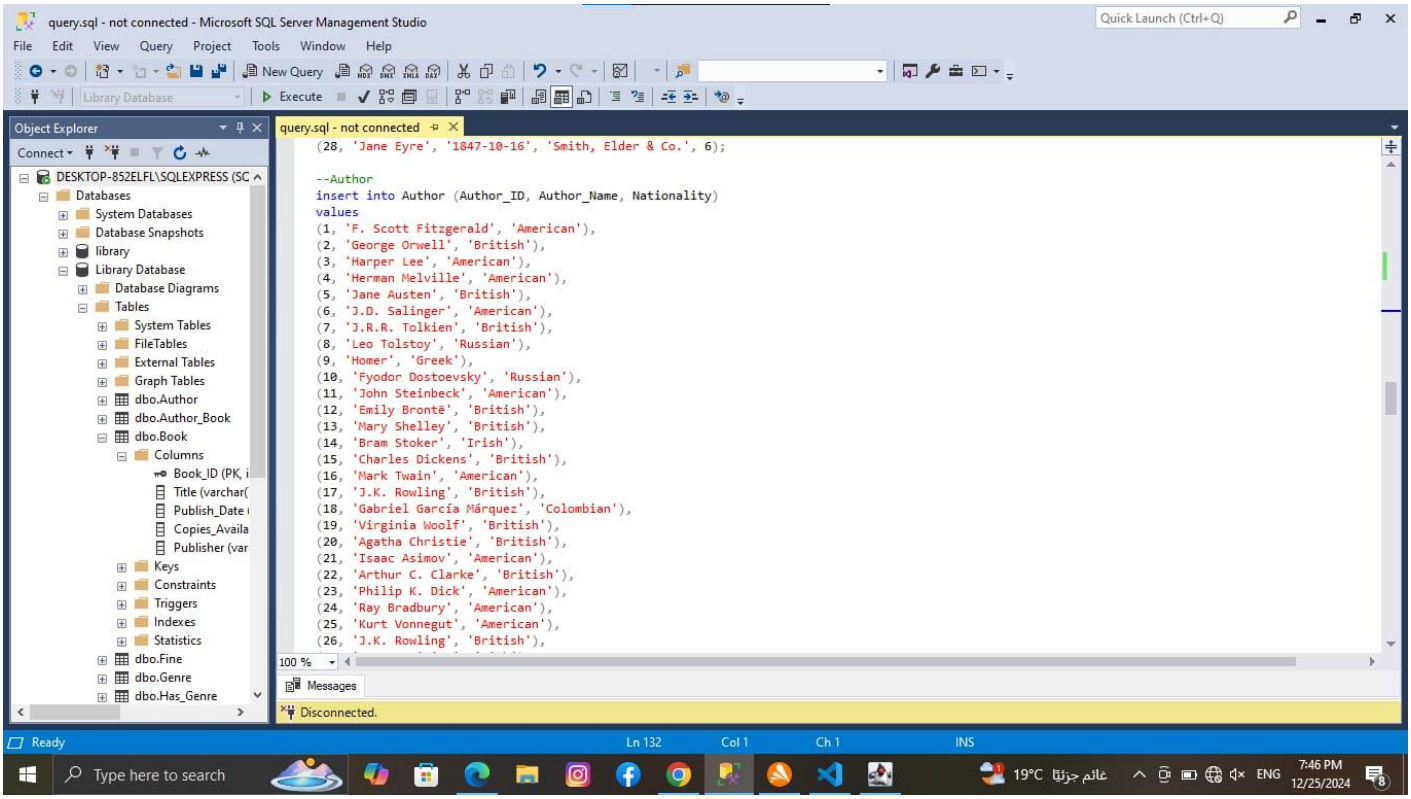
The DELETED virtual table temporarily holds the Book_ID values of the books that were deleted.

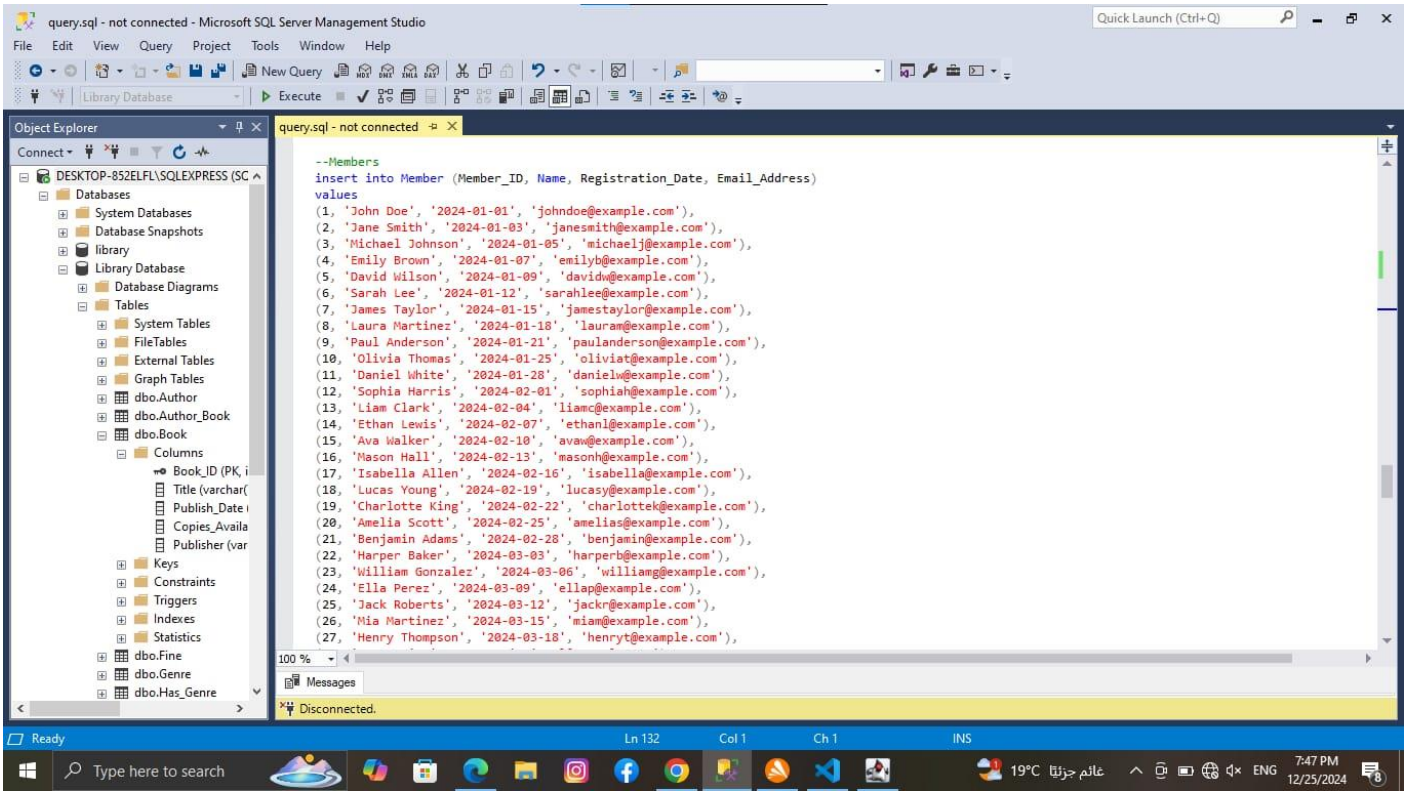
The DELETE statement removes all rows from the Author_Book table where the Author_ID matches any Book_ID in the DELETED table. This ensures the Author_Book table does not contain references to books that no longer exist.
This second step ensures the Author table does not contain "orphaned" authors (authors who are not associated with any books).
The DELETE statement removes all authors from the Author table where the Author_ID is not found in the Author_Book table.

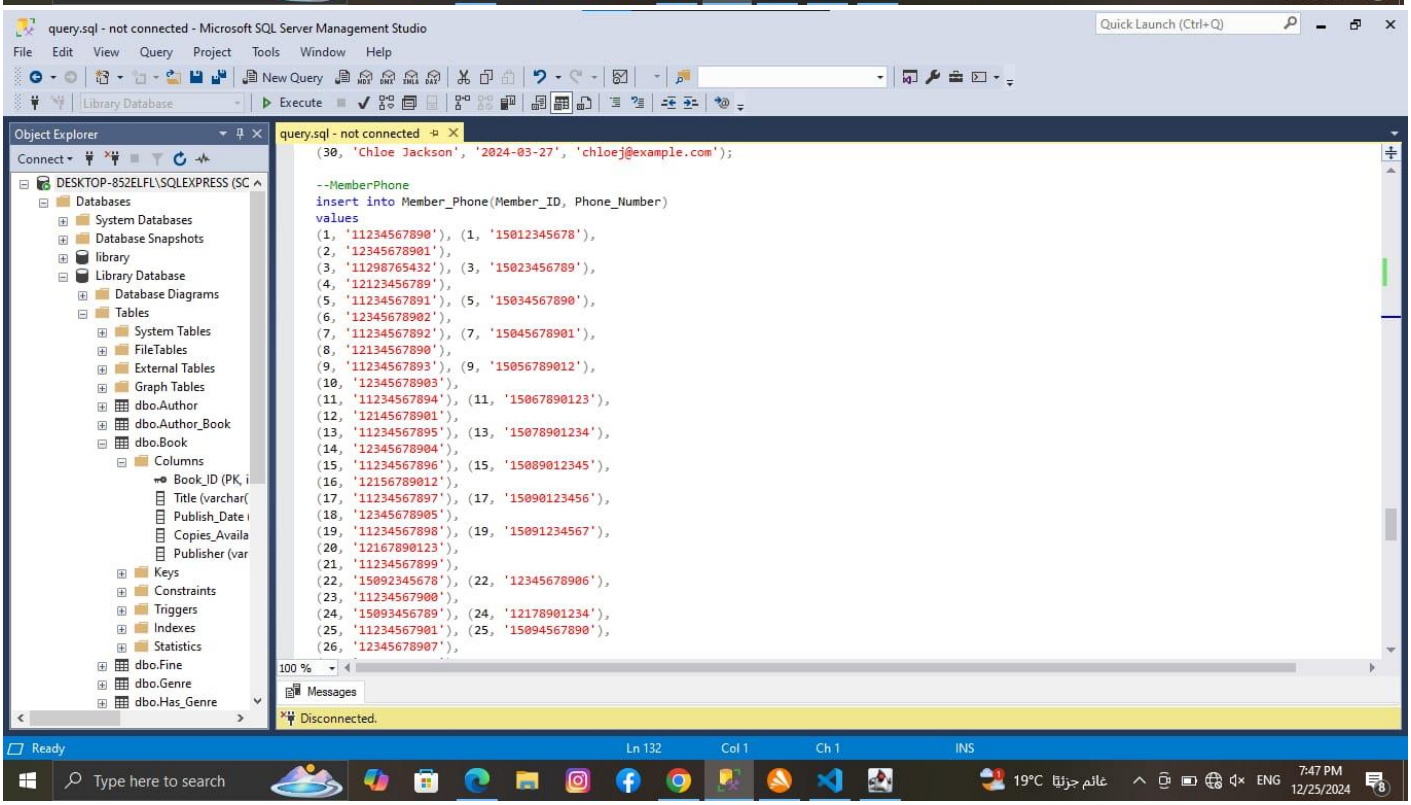
Here are all the insertions we made in every table:

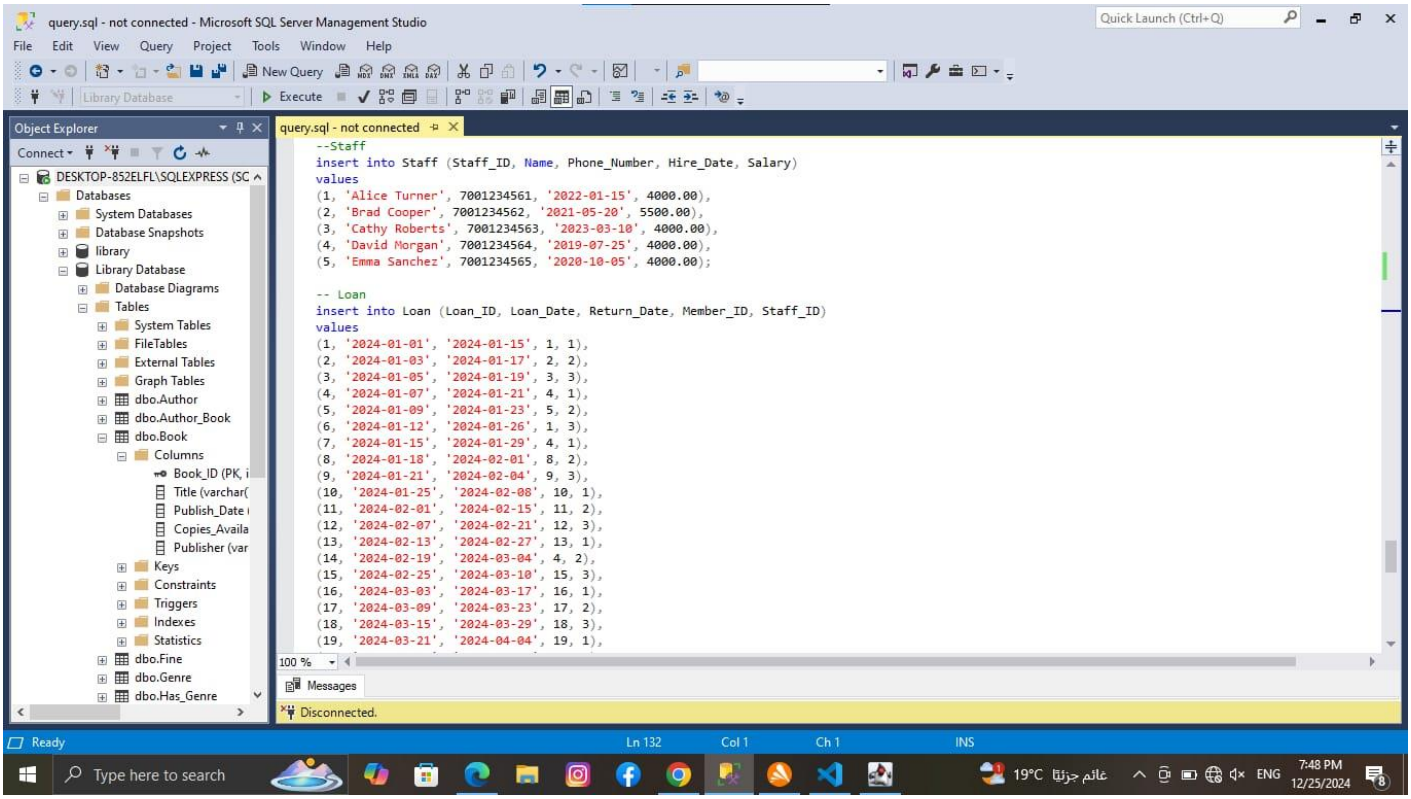

query.sql - not connected - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
-- Book
insert into Book (Book_ID, title, Publish_Date, publisher, Copies_available)
VALUES
(1, 'The Great Gatsby', '1925-04-10', 'Charles Scribners Sons', 5),
(2, '1984', '1949-06-08', 'Secker & Warburg', 3),
(3, 'To Kill a Mockingbird', '1960-07-11', 'J.B. Lippincott & Co.', 4),
(4, 'Moby-Dick', '1851-10-18', 'Harper & Brothers', 2),
(5, 'Pride and Prejudice', '1813-01-28', 'T. Egerton, Whitehall', 6),
(6, 'The Catcher in the Rye', '1951-07-16', 'Little, Brown and Company', 4),
(7, 'The Hobbit', '1937-09-21', 'George Allen & Unwin', 7),
(8, 'War and Peace', '1869-01-01', 'The Russian Messenger', 3),
(9, 'The Odyssey', '1900-01-01', 'Ancient Texts', 8),
(10, 'Crime and Punishment', '1866-01-01', 'The Russian Messenger', 5),
(11, 'The Grapes of Wrath', '1939-04-14', 'The Viking Press', 4),
(12, 'Wuthering Heights', '1847-12-01', 'Thomas Cautley Newby', 3),
(13, 'Frankenstein', '1818-01-01', 'Lackington, Hughes, Harding, Mavor & Jones', 6),
(14, 'Dracula', '1897-05-26', 'Archibald Constable and Company', 2),
(15, 'Anna Karenina', '1878-04-01', 'The Russian Messenger', 5),
(16, 'Brave New World', '1932-08-01', 'Chatto & Windus', 3),
(17, 'The Picture of Dorian Gray', '1890-07-01', 'Lippincotts Monthly Magazine', 4),
(18, 'Heart of Darkness', '1899-02-01', 'Blackwoods Magazine', 3),
(19, 'Les Misérables', '1862-04-03', 'A. Lacroix, Verboeckhoven & Cie', 7),
(20, 'The Divine Comedy', '1900-01-01', 'Ancient Texts', 5),
(21, 'The Lord of the Rings', '1954-07-29', 'George Allen & Unwin', 9),
(22, 'Ulysses', '1922-02-02', 'Sylvia Beach', 4),
(23, 'The Catcher in the Rye', '1951-07-16', 'Little, Brown and Company', 3),
(24, 'Catcher in the Rye', '1951-07-16', 'Little, Brown and Company', 5),
(25, 'The Chronicles of Narnia', '1950-10-16', 'Geoffrey Bles', 8),
(26, 'Harry Potter and the Philosophers Stone', '1997-06-26','George Allen & Unwin', 5),
(27, 'The Old Man and the Sea', '1952-09-01', 'Charles Scribners Sons', 4),
(28, 'Jane Eyre', '1847-10-16', 'Smith, Elder & Co.', 6);
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:46 PM
12/25/2024

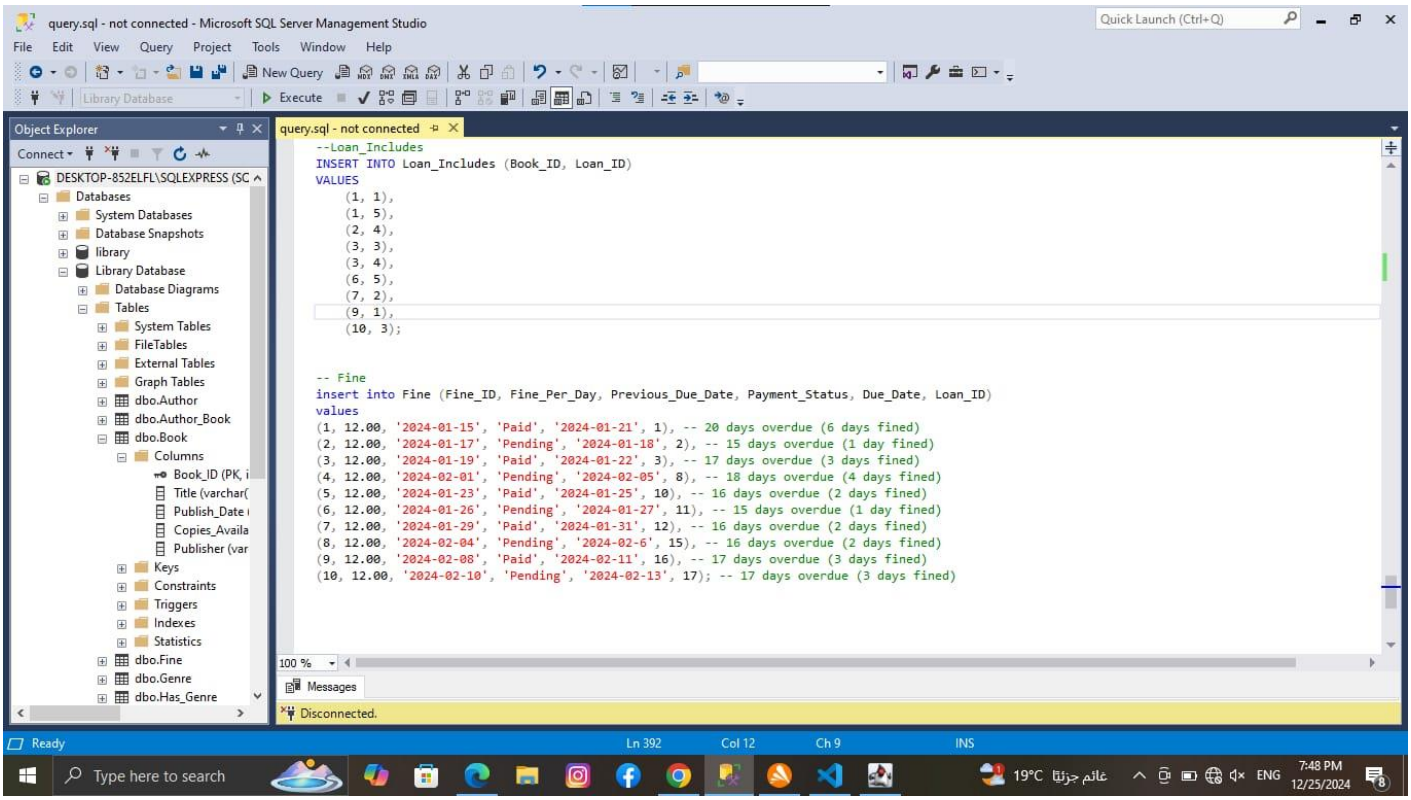

query.sql - not connected - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
--Writes
insert into Author_Book(Author_ID, Book_ID)
values
(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8), (9, 9), (10, 10),
(11, 11), (12, 12), (13, 13), (14, 14), (15, 15), (16, 16), (17, 17),(18, 18), (19, 19),
(21, 21), (22, 22), (23, 23), (24, 24), (25, 25), (26, 26), (27, 27), (28, 28);
--Genre
insert into Genre (Genre_ID, Name) values
(1, 'Fiction'), (2, 'Fantasy'), (3, 'Science Fiction'), (4, 'Mystery'),
(5, 'Romance'), (6, 'Historical Fiction'), (7, 'Thriller'), (8, 'Horror'),
(9, 'Adventure'), (10, 'Young Adult'), (11, 'Childrens Literature'),
(12, 'Biography'), (13, 'Autobiography'), (14, 'Memoir'), (15, 'Self-Help'),
(16, 'Non-Fiction'), (17, 'Classics'), (18, 'Humor'), (19, 'Poetry'),
(20, 'Drama'), (21, 'Graphic Novel'), (22, 'Short Stories'), (23, 'Contemporary'),
(24, 'Literary Fiction'), (25, 'Crime');
--Has Genre
insert into Has_Genre (book_id, genre_id) values
(1, 21),
(2, 1), (2, 24),
(3, 9), (3, 8),
(4, 8), (4, 5),
(5, 24), (5, 4),
(6, 14),
(7, 25), (7, 21),
(8, 9), (8, 24),
(9, 13), (9, 14),
(10, 19),
(11, 1),
(12, 4),
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:47 PM
12/25/2024


query.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date (
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
(28, 'Jane Eyre', '1847-10-16', 'Smith, Elder & Co.', 6);
--Author
insert into Author (Author_ID, Author_Name, Nationality)
values
(1, 'F. Scott Fitzgerald', 'American'),
(2, 'George Orwell', 'British'),
(3, 'Harper Lee', 'American'),
(4, 'Herman Melville', 'American'),
(5, 'Jane Austen', 'British'),
(6, 'J.D. Salinger', 'American'),
(7, 'J.R.R. Tolkien', 'British'),
(8, 'Leo Tolstoy', 'Russian'),
(9, 'Homer', 'Greek'),
(10, 'Fyodor Dostoevsky', 'Russian'),
(11, 'John Steinbeck', 'American'),
(12, 'Emily Brontë', 'British'),
(13, 'Mary Shelley', 'British'),
(14, 'Bram Stoker', 'Irish'),
(15, 'Charles Dickens', 'British'),
(16, 'Mark Twain', 'American'),
(17, 'J.K. Rowling', 'British'),
(18, 'Gabriel García Márquez', 'Colombian'),
(19, 'Virginia Woolf', 'British'),
(20, 'Agatha Christie', 'British'),
(21, 'Isaac Asimov', 'American'),
(22, 'Arthur C. Clarke', 'British'),
(23, 'Philip K. Dick', 'American'),
(24, 'Ray Bradbury', 'American'),
(25, 'Kurt Vonnegut', 'American'),
(26, 'J.K. Rowling', 'British'),
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:46 PM
12/25/2024

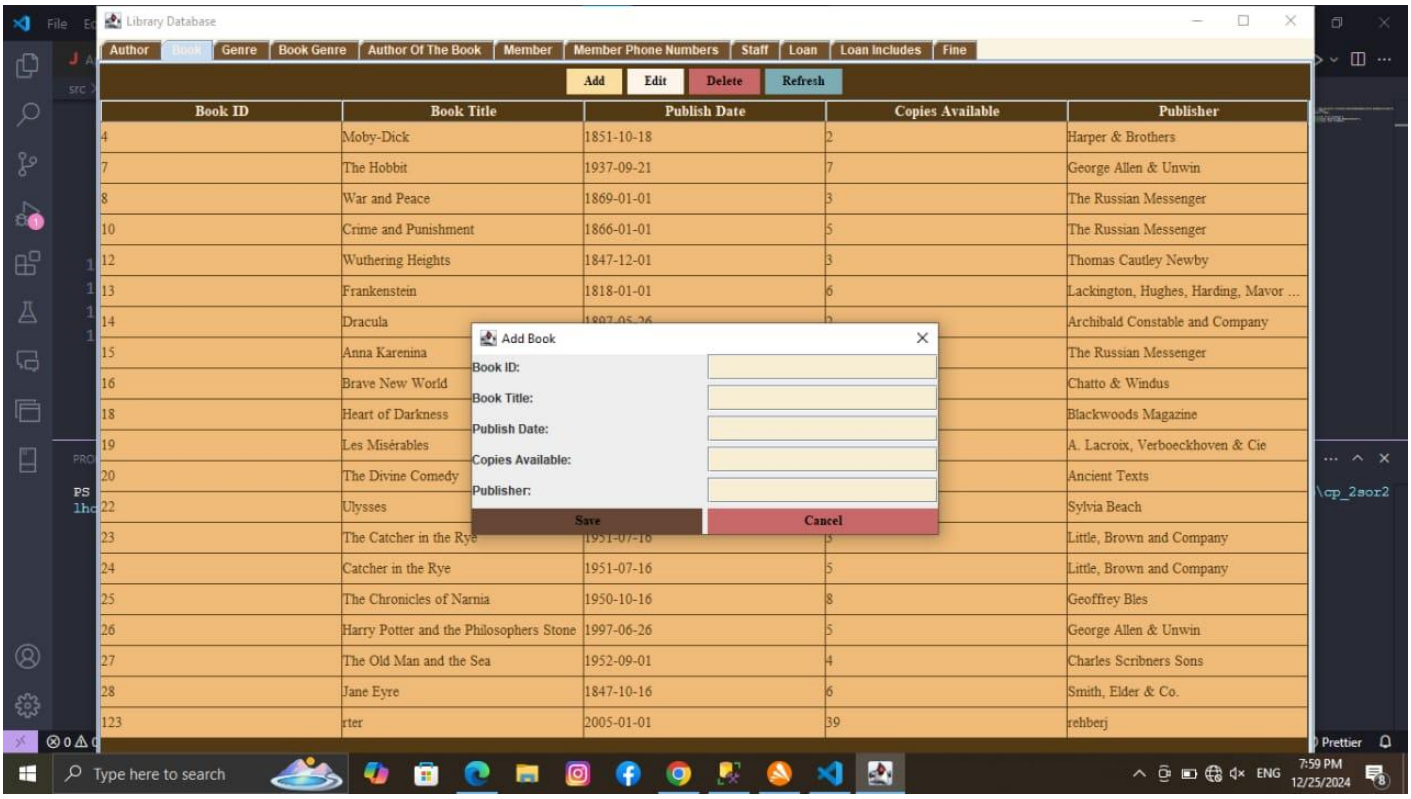

query.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
--Members
insert into Member (Member_ID, Name, Registration_Date, Email_Address)
values
(1, 'John Doe', '2024-01-01', 'johndoe@example.com'),
(2, 'Jane Smith', '2024-01-03', 'janesmith@example.com'),
(3, 'Michael Johnson', '2024-01-05', 'michaelj@example.com'),
(4, 'Emily Brown', '2024-01-07', 'emilyb@example.com'),
(5, 'David Wilson', '2024-01-09', 'davidw@example.com'),
(6, 'Sarah Lee', '2024-01-12', 'sarahlee@example.com'),
(7, 'James Taylor', '2024-01-15', 'jamestaylor@example.com'),
(8, 'Laura Martinez', '2024-01-18', 'lauram@example.com'),
(9, 'Paul Anderson', '2024-01-21', 'paulanderson@example.com'),
(10, 'Olivia Thomas', '2024-01-25', 'oliviat@example.com'),
(11, 'Daniel White', '2024-01-28', 'danielw@example.com'),
(12, 'Sophia Harris', '2024-02-01', 'sophiah@example.com'),
(13, 'Liam Clark', '2024-02-04', 'liamc@example.com'),
(14, 'Ethan Lewis', '2024-02-07', 'ethanl@example.com'),
(15, 'Ava Walker', '2024-02-10', 'avaw@example.com'),
(16, 'Mason Hall', '2024-02-13', 'masonh@example.com'),
(17, 'Isabella Allen', '2024-02-16', 'isabella@example.com'),
(18, 'Lucas Young', '2024-02-19', 'lucasy@example.com'),
(19, 'Charlotte King', '2024-02-22', 'charlottek@example.com'),
(20, 'Amelia Scott', '2024-02-25', 'amelias@example.com'),
(21, 'Benjamin Adams', '2024-02-28', 'benjamin@example.com'),
(22, 'Harper Baker', '2024-03-03', 'harperb@example.com'),
(23, 'William Gonzalez', '2024-03-06', 'williamg@example.com'),
(24, 'Ella Perez', '2024-03-09', 'ellap@example.com'),
(25, 'Jack Roberts', '2024-03-12', 'jackr@example.com'),
(26, 'Mia Martinez', '2024-03-15', 'miam@example.com'),
(27, 'Henry Thompson', '2024-03-18', 'henryt@example.com'),
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:47 PM
12/25/2024


query.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
(30, 'Chloe Jackson', '2024-03-27', 'chloej@example.com');
--MemberPhone
insert into Member_Phone(Member_ID, Phone_Number)
values
(1, '11234567890'), (1, '15012345678'),
(2, '12345678901'),
(3, '11298765432'), (3, '15023456789'),
(4, '12123456789'),
(5, '11234567891'), (5, '15034567890'),
(6, '12345678902'),
(7, '11234567892'), (7, '15045678901'),
(8, '12134567890'),
(9, '11234567893'), (9, '15056789012'),
(10, '12345678903'),
(11, '11234567894'), (11, '15067890123'),
(12, '12145678901'),
(13, '11234567895'), (13, '15078901234'),
(14, '12345678904'),
(15, '11234567896'), (15, '15089012345'),
(16, '12156789012'),
(17, '11234567897'), (17, '15090123456'),
(18, '12345678905'),
(19, '11234567898'), (19, '15091234567'),
(20, '12167890123'),
(21, '11234567899'),
(22, '15092345678'), (22, '12345678906'),
(23, '11234567900'),
(24, '15093456789'), (24, '12178901234'),
(25, '11234567901'), (25, '15094567890'),
(26, '12345678907'),
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:47 PM
12/25/2024

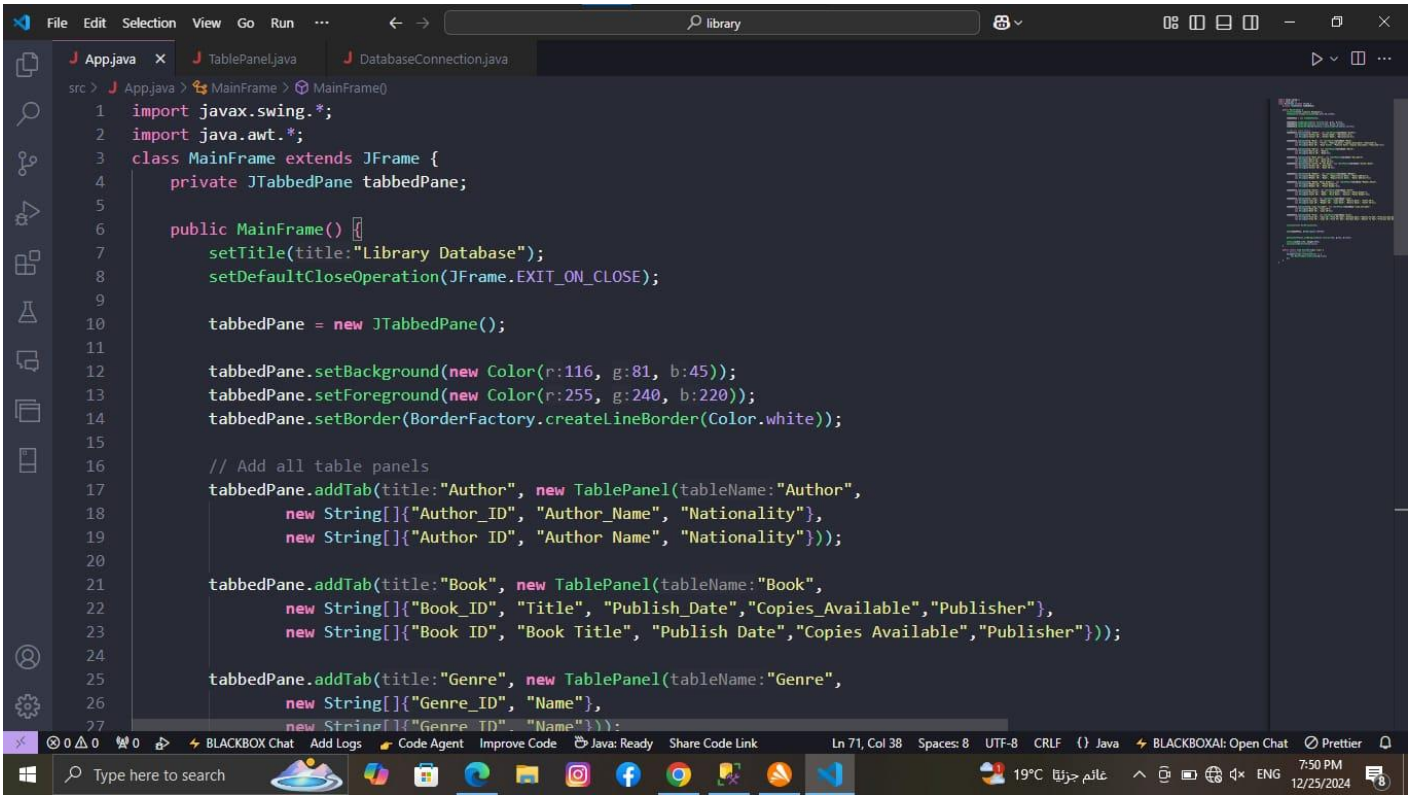

query.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
--Staff
insert into Staff (Staff_ID, Name, Phone_Number, Hire_Date, Salary)
values
(1, 'Alice Turner', 7001234561, '2022-01-15', 4000.00),
(2, 'Brad Cooper', 7001234562, '2021-05-20', 5500.00),
(3, 'Cathy Roberts', 7001234563, '2023-03-10', 4000.00),
(4, 'David Morgan', 7001234564, '2019-07-25', 4000.00),
(5, 'Emma Sanchez', 7001234565, '2020-10-05', 4000.00);
-- Loan
insert into Loan (Loan_ID, Loan_Date, Return_Date, Member_ID, Staff_ID)
values
(1, '2024-01-01', '2024-01-15', 1, 1),
(2, '2024-01-03', '2024-01-17', 2, 2),
(3, '2024-01-05', '2024-01-19', 3, 3),
(4, '2024-01-07', '2024-01-21', 4, 1),
(5, '2024-01-09', '2024-01-23', 5, 2),
(6, '2024-01-12', '2024-01-26', 1, 3),
(7, '2024-01-15', '2024-01-29', 4, 1),
(8, '2024-01-18', '2024-02-01', 8, 2),
(9, '2024-01-21', '2024-02-04', 9, 3),
(10, '2024-01-25', '2024-02-08', 10, 1),
(11, '2024-02-01', '2024-02-15', 11, 2),
(12, '2024-02-07', '2024-02-21', 12, 3),
(13, '2024-02-13', '2024-02-27', 13, 1),
(14, '2024-02-19', '2024-03-04', 4, 2),
(15, '2024-02-25', '2024-03-10', 15, 3),
(16, '2024-03-03', '2024-03-17', 16, 1),
(17, '2024-03-09', '2024-03-23', 17, 2),
(18, '2024-03-15', '2024-03-29', 18, 3),
(19, '2024-03-21', '2024-04-04', 19, 1),
100 %
Messages
Disconnected.
Ready
Ln 132
Col 1
Ch 1
INS
Type here to search
19°C
ENG
7:48 PM
12/25/2024

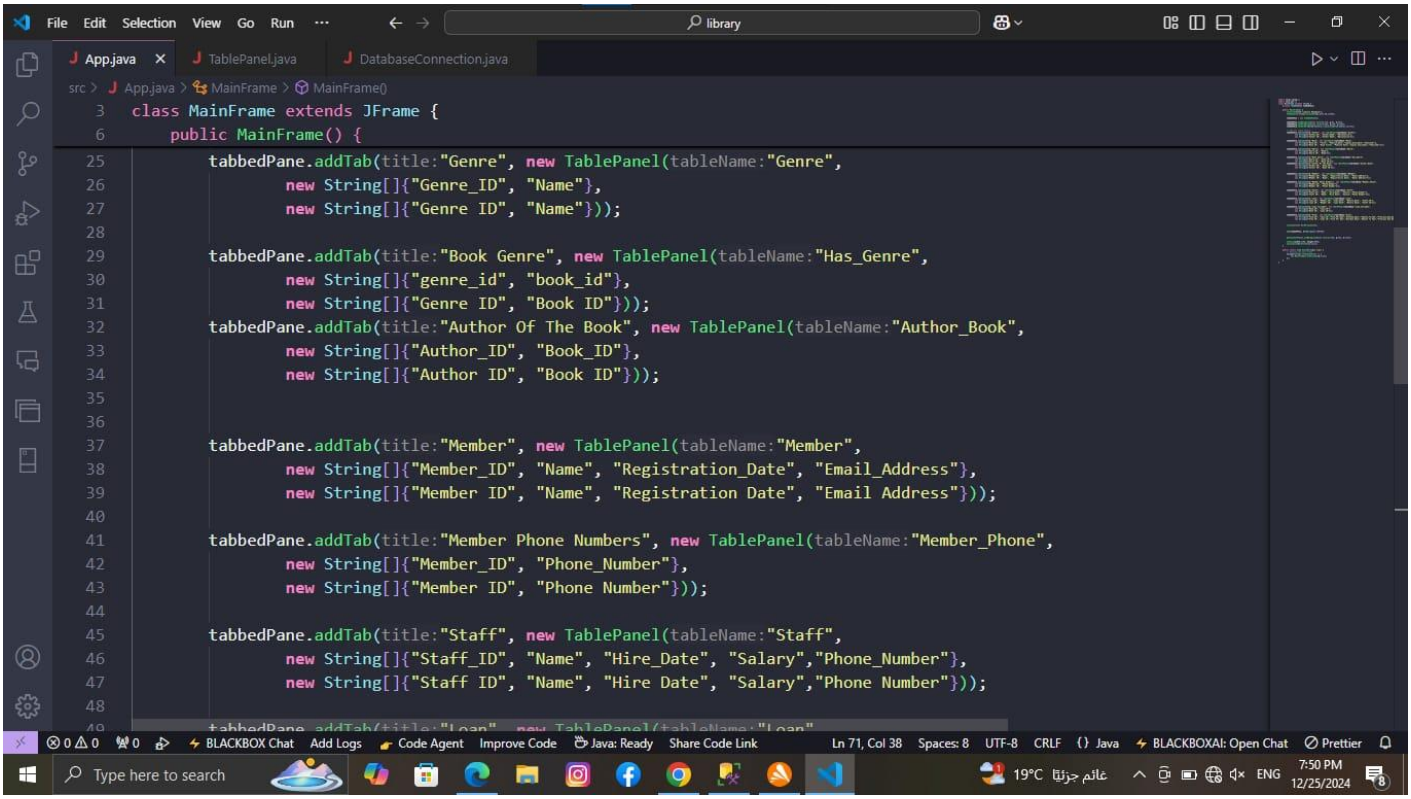

query.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-852ELFL\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
library
Library Database
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Author
dbo.Author_Book
dbo.Book
Columns
Book_ID (PK, i
Title (varchar(
Publish_Date
Copies_Availa
Publisher (var
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Fine
dbo.Genre
dbo.Has_Genre
query.sql - not connected
--Loan_Includes
INSERT INTO Loan_Includes (Book_ID, Loan_ID)
VALUES
(1, 1),
(1, 5),
(2, 4),
(3, 3),
(3, 4),
(6, 5),
(7, 2),
(9, 1),
(10, 3);
-- Fine
insert into Fine (Fine_ID, Fine_Per_Day, Previous_Due_Date, Payment_Status, Due_Date, Loan_ID)
values
(1, 12.00, '2024-01-15', 'Paid', '2024-01-21', 1), -- 20 days overdue (6 days fined)
(2, 12.00, '2024-01-17', 'Pending', '2024-01-18', 2), -- 15 days overdue (1 day fined)
(3, 12.00, '2024-01-19', 'Paid', '2024-01-22', 3), -- 17 days overdue (3 days fined)
(4, 12.00, '2024-02-01', 'Pending', '2024-02-05', 8), -- 18 days overdue (4 days fined)
(5, 12.00, '2024-01-23', 'Paid', '2024-01-25', 10), -- 16 days overdue (2 days fined)
(6, 12.00, '2024-01-26', 'Pending', '2024-01-27', 11), -- 15 days overdue (1 day fined)
(7, 12.00, '2024-01-29', 'Paid', '2024-01-31', 12), -- 16 days overdue (2 days fined)
(8, 12.00, '2024-02-04', 'Pending', '2024-02-6', 15), -- 16 days overdue (2 days fined)
(9, 12.00, '2024-02-08', 'Paid', '2024-02-11', 16), -- 17 days overdue (3 days fined)
(10, 12.00, '2024-02-10', 'Pending', '2024-02-13', 17); -- 17 days overdue (3 days fined)
100 %
Messages
Disconnected.
Ready
Ln 392
Col 12
Ch 9
INS
Type here to search
19°C
ENG
7:48 PM
12/25/2024

GUI

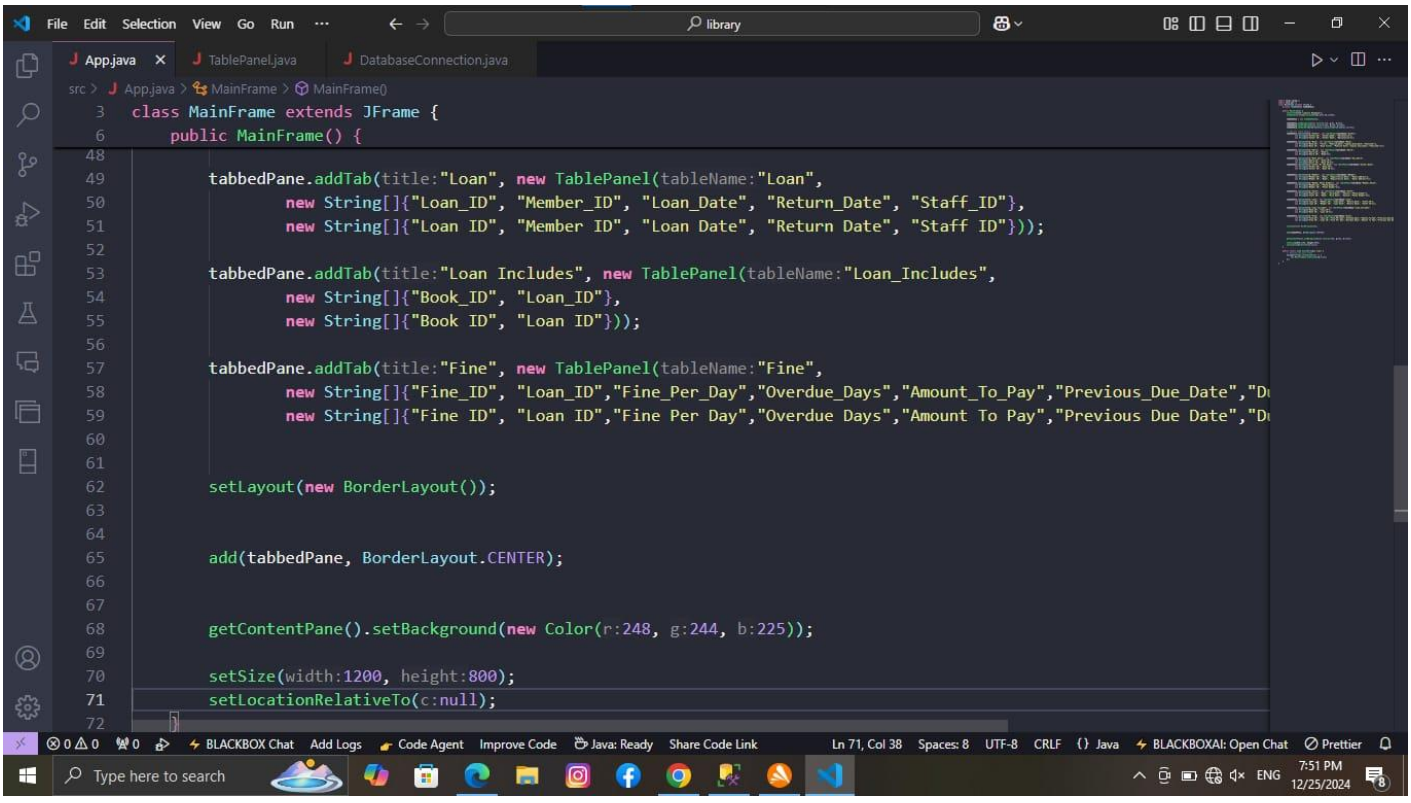

Library Database
Author
Book
Genre
Book Genre
Author Of The Book
Member
Member Phone Numbers
Staff
Loan
Loan Includes
Fine
Add
Edit
Delete
Refresh
Book ID
Book Title
Publish Date
Copies Available
Publisher
4 Moby-Dick 1851-10-18 2 Harper & Brothers
7 The Hobbit 1937-09-21 7 George Allen & Unwin
8 War and Peace 1869-01-01 3 The Russian Messenger
10 Crime and Punishment 1866-01-01 5 The Russian Messenger
12 Wuthering Heights 1847-12-01 3 Thomas Cautley Newby
13 Frankenstein 1818-01-01 6 Lackington, Hughes, Harding, Mavor ...
14 Dracula Archibald Constable and Company
15 Anna Karenina The Russian Messenger
16 Brave New World Chatto & Windus
18 Heart of Darkness Blackwoods Magazine
19 Les Misérables A. Lacroix, Verboeckhoven & Cie
20 The Divine Comedy Ancient Texts
22 Ulysses Sylvia Beach
23 The Catcher in the Rye 1951-07-16 Little, Brown and Company
24 Catcher in the Rye 1951-07-16 5 Little, Brown and Company
25 The Chronicles of Narnia 1950-10-16 8 Geoffrey Bles
26 Harry Potter and the Philosophers Stone 1997-06-26 5 George Allen & Unwin
27 The Old Man and the Sea 1952-09-01 4 Charles Scribners Sons
28 Jane Eyre 1847-10-16 6 Smith, Elder & Co.
123 rter 2005-01-01 39 rehberj
Add Book
Book ID:
Book Title:
Publish Date:
Copies Available:
Publisher:
Save
Cancel
Type here to search
ENG
7:59 PM
12/25/2024

GUI CODE

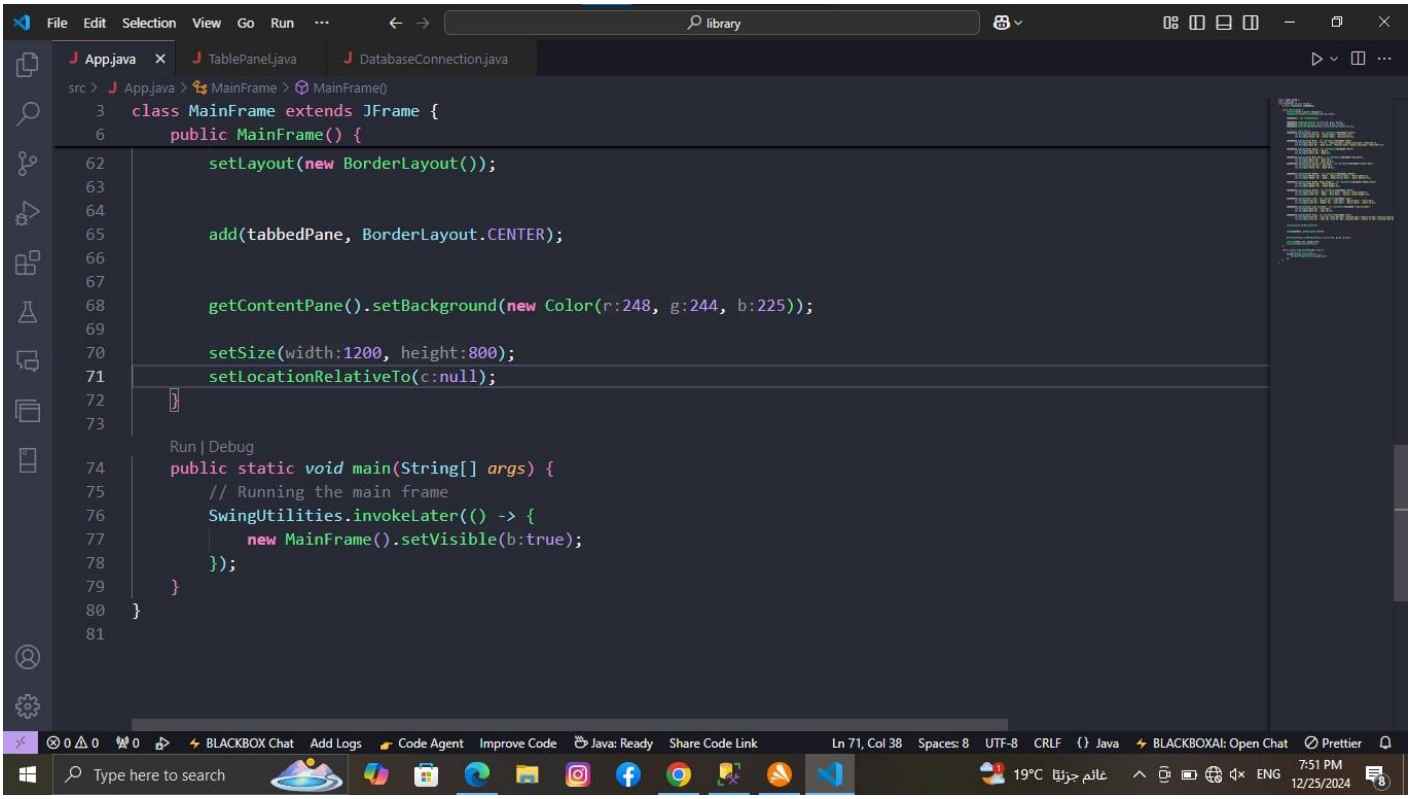

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > App.java > MainFrame > MainFrame()
1 import javax.swing.*;
2 import java.awt.*;
3 class MainFrame extends JFrame {
4 private JTabbedPane tabbedPane;
5
6 public MainFrame() {
7 setTitle(title:"Library Database");
8 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9
10 tabbedPane = new JTabbedPane();
11
12 tabbedPane.setBackground(new Color(r:116, g:81, b:45));
13 tabbedPane.setForeground(new Color(r:255, g:240, b:220));
14 tabbedPane.setBorder(BorderFactory.createLineBorder(Color.white));
15
16 // Add all table panels
17 tabbedPane.addTab(title:"Author", new TablePanel(tableName:"Author",
18 new String[]{"Author_ID", "Author_Name", "Nationality"},
19 new String[]{"Author ID", "Author Name", "Nationality"}));
20
21 tabbedPane.addTab(title:"Book", new TablePanel(tableName:"Book",
22 new String[]{"Book_ID", "Title", "Publish_Date","Copies_Available","Publisher"},
23 new String[]{"Book ID", "Book Title", "Publish Date","Copies Available","Publisher"}));
24
25 tabbedPane.addTab(title:"Genre", new TablePanel(tableName:"Genre",
26 new String[]{"Genre_ID", "Name"},
BLACKBOX Chat
Add Logs
Code Agent
Improve Code
Java: Ready
Share Code Link
Ln 71, Col 38
Spaces: 8
UTF-8
CRLF
Java
BLACKBOXAI: Open Chat
Prettier
Type here to search
19°C
ENG
7:50 PM
12/25/2024



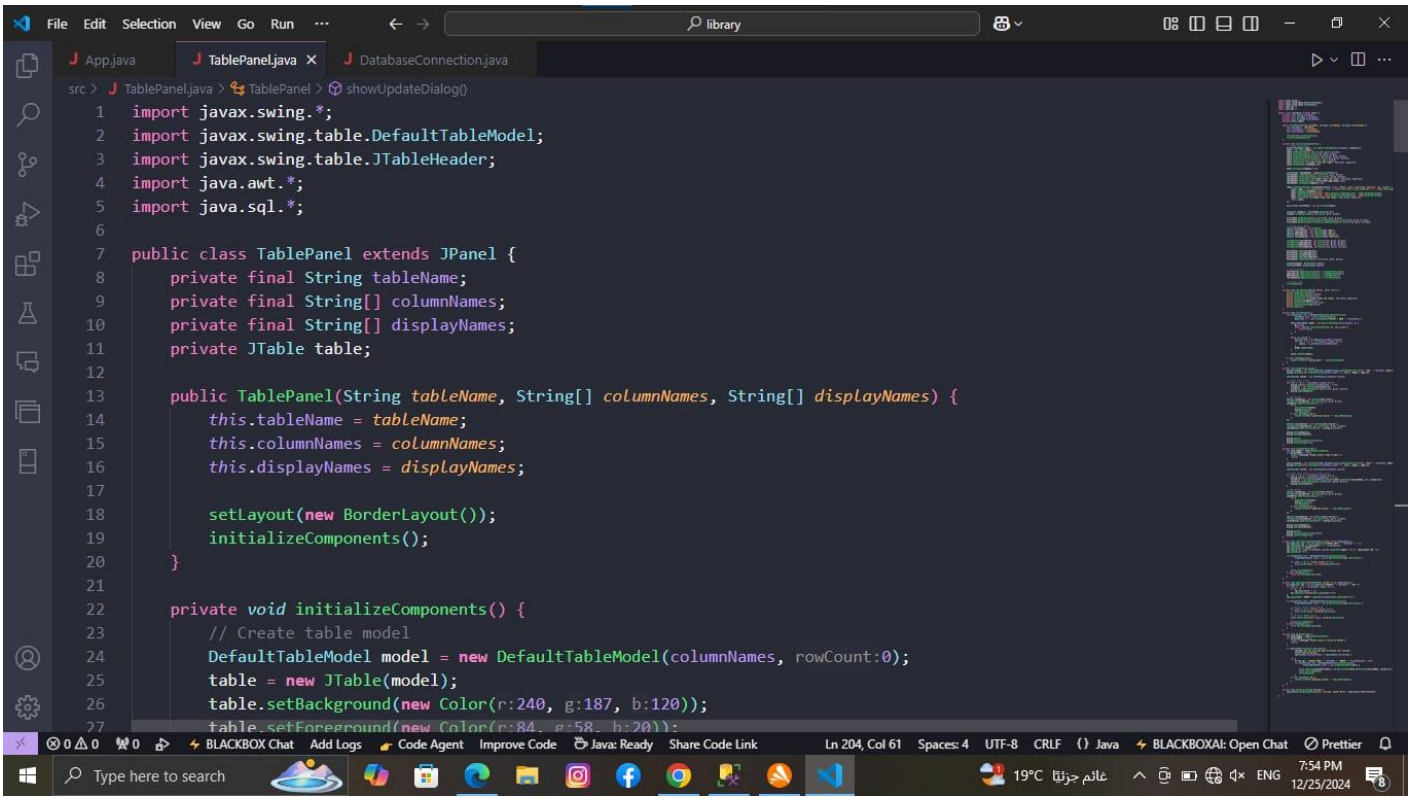
```
class MainFrame extends JFrame {
    public MainFrame() {
        tabbedPane.addTab(title:"Genre", new TablePanel(tableName:"Genre",
                new String[]{"Genre_ID", "Name"},
                new String[]{"Genre ID", "Name"}));

        tabbedPane.addTab(title:"Book Genre", new TablePanel(tableName:"Has_Genre",
                new String[]{"genre_id", "book_id"},
                new String[]{"Genre ID", "Book ID"}));
        tabbedPane.addTab(title:"Author Of The Book", new TablePanel(tableName:"Author_Book",
                new String[]{"Author_ID", "Book_ID"},
                new String[]{"Author ID", "Book ID"}));


        tabbedPane.addTab(title:"Member", new TablePanel(tableName:"Member",
                new String[]{"Member_ID", "Name", "Registration_Date", "Email_Address"},
                new String[]{"Member ID", "Name", "Registration Date", "Email Address"}));

        tabbedPane.addTab(title:"Member Phone Numbers", new TablePanel(tableName:"Member_Phone",
                new String[]{"Member_ID", "Phone_Number"},
                new String[]{"Member ID", "Phone Number"}));

        tabbedPane.addTab(title:"Staff", new TablePanel(tableName:"Staff",
                new String[]{"Staff_ID", "Name", "Hire_Date", "Salary","Phone_Number"},
                new String[]{"Staff ID", "Name", "Hire Date", "Salary","Phone Number"}));
```



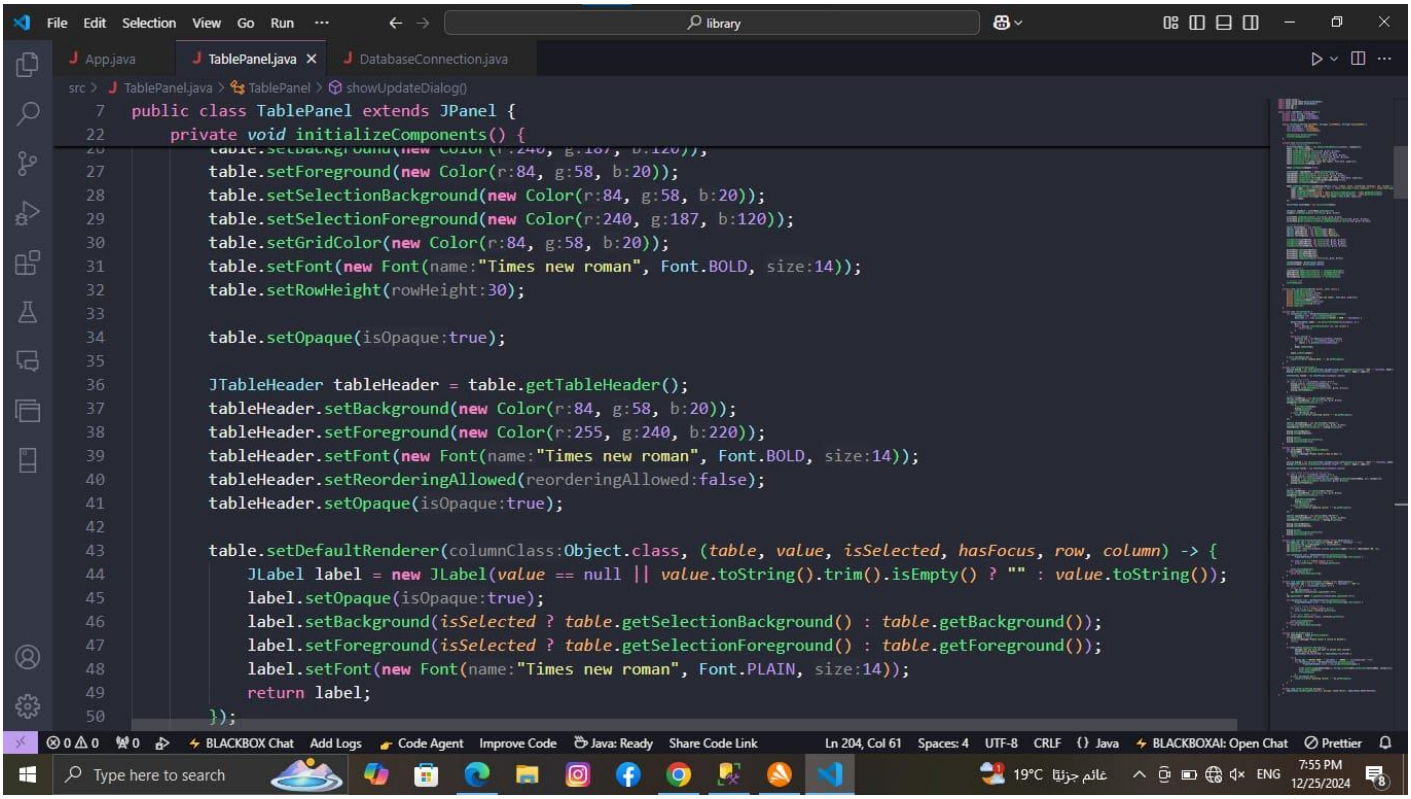
```
class MainFrame extends JFrame {
    public MainFrame() {

        tabbedPane.addTab(title:"Loan", new TablePanel(tableName:"Loan",
                new String[]{"Loan_ID", "Member_ID", "Loan_Date", "Return_Date", "Staff_ID"},
                new String[]{"Loan ID", "Member ID", "Loan Date", "Return Date", "Staff ID"}));

        tabbedPane.addTab(title:"Loan Includes", new TablePanel(tableName:"Loan_Includes",
                new String[]{"Book_ID", "Loan_ID"},
                new String[]{"Book ID", "Loan ID"}));

        tabbedPane.addTab(title:"Fine", new TablePanel(tableName:"Fine",
                new String[]{"Fine_ID", "Loan_ID","Fine_Per_Day","Overdue_Days","Amount_To_Pay","Previous_Due_Date","D
                new String[]{"Fine ID", "Loan ID","Fine Per Day","Overdue Days","Amount To Pay","Previous Due Date","D


        setLayout(new BorderLayout());


        add(tabbedPane, BorderLayout.CENTER);


        getContentPane().setBackground(new Color(r:248, g:244, b:225));

        setSize(width:1200, height:800);
        setLocationRelativeTo(c:null);
    }
```

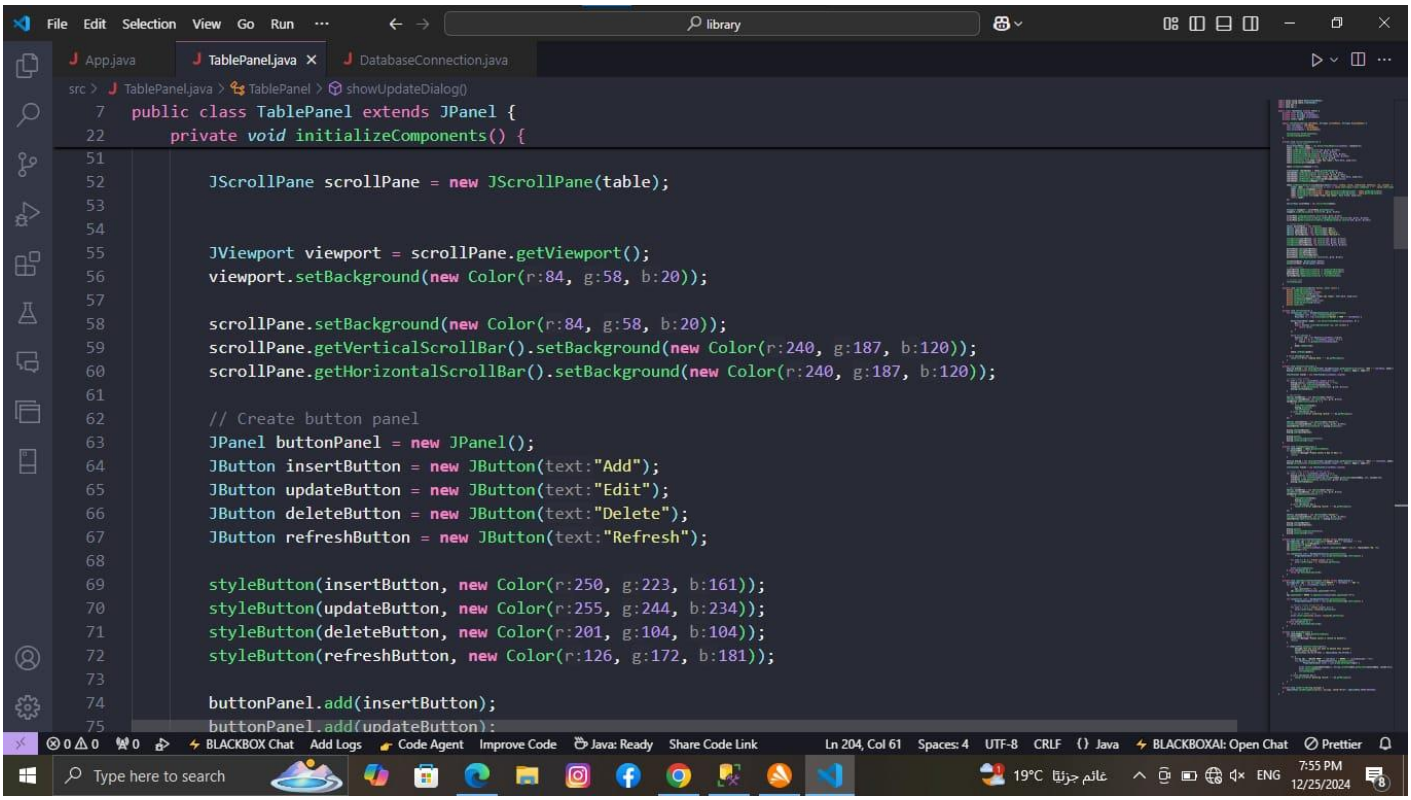


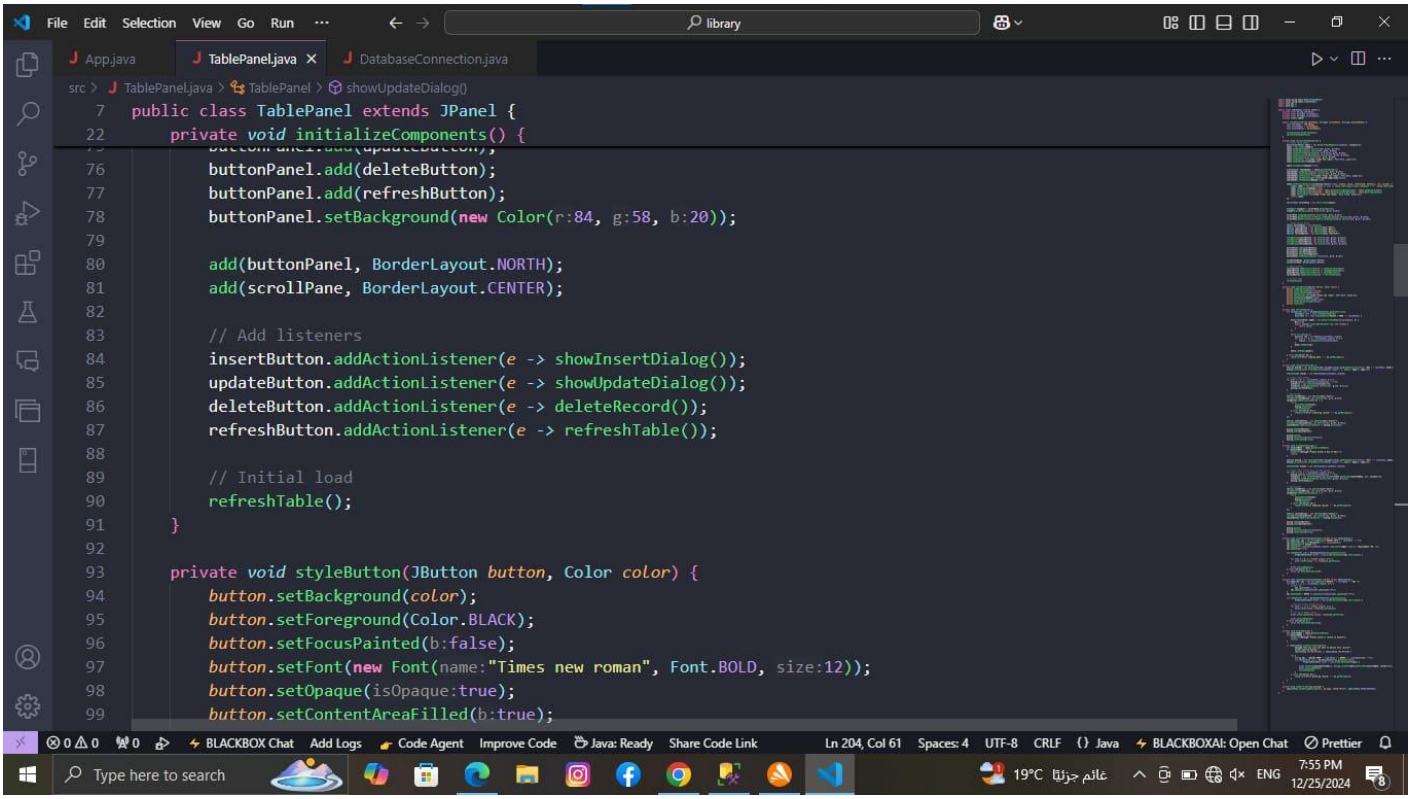
```
class MainFrame extends JFrame {
    public MainFrame() {
        setLayout(new BorderLayout());


        add(tabbedPane, BorderLayout.CENTER);


        getContentPane().setBackground(new Color(r:248, g:244, b:225));

        setSize(width:1200, height:800);
        setLocationRelativeTo(c:null);
    }

    Run | Debug
    public static void main(String[] args) {
        // Running the main frame
        SwingUtilities.invokeLater(() -> {
            new MainFrame().setVisible(b:true);
        });
    }
}
```

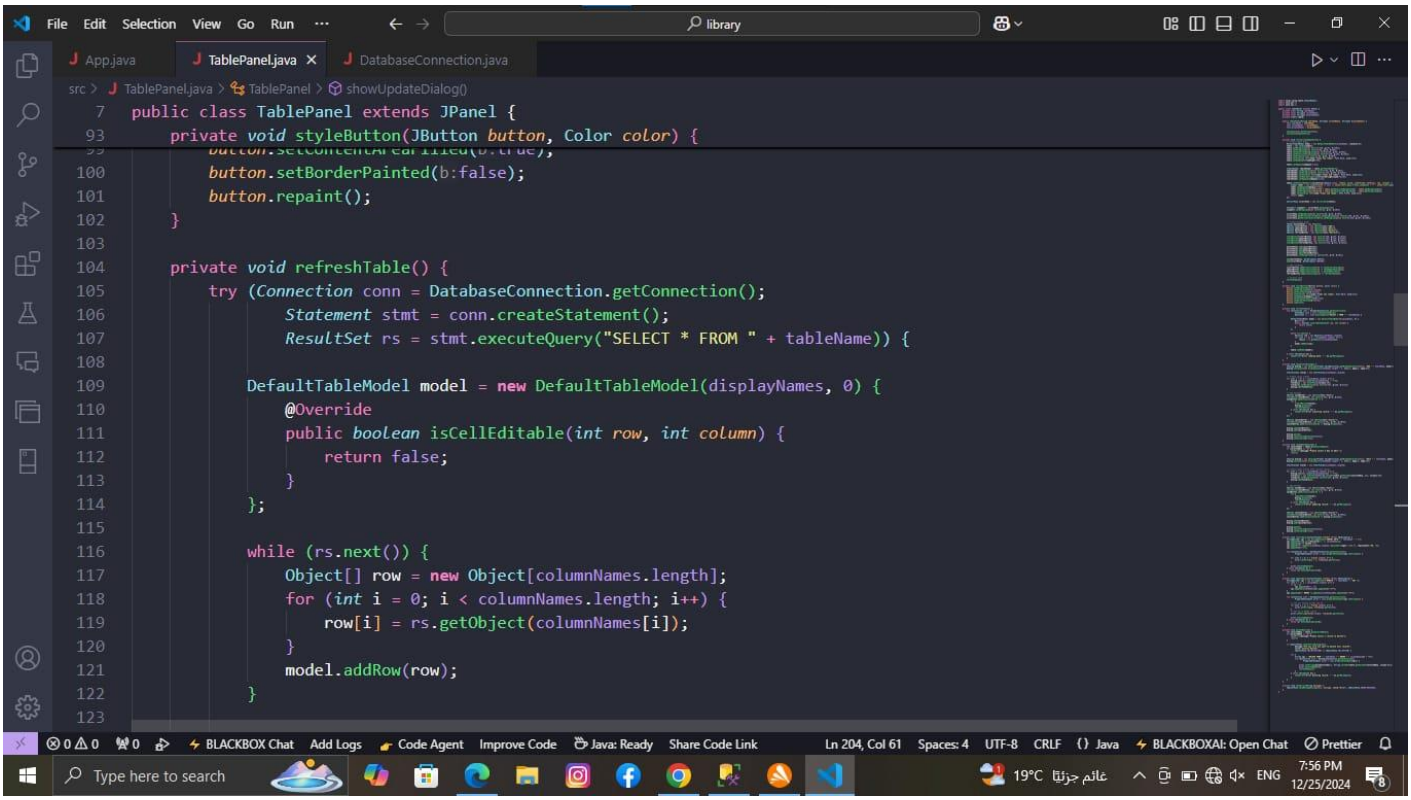


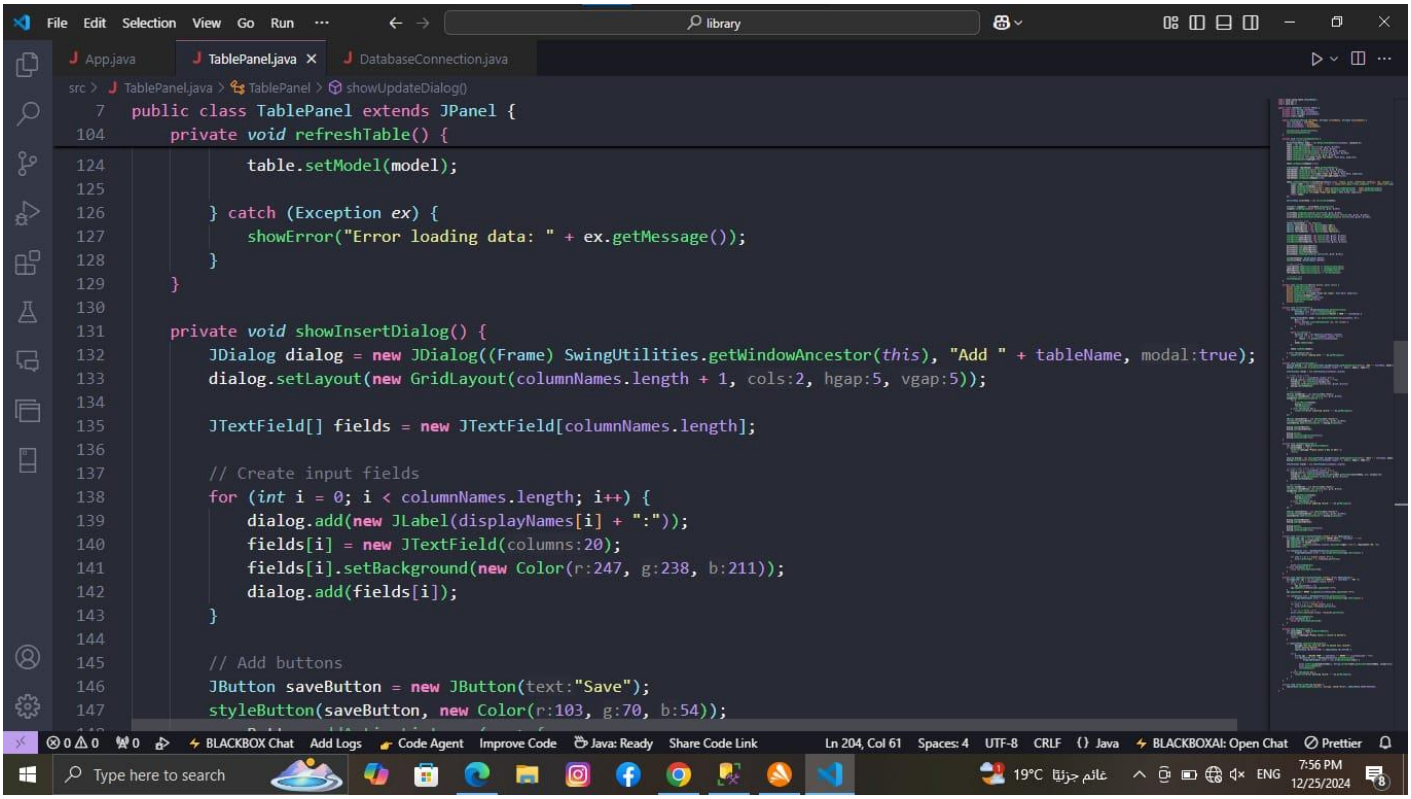
```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.sql.*;

public class TablePanel extends JPanel {
    private final String tableName;
    private final String[] columnNames;
    private final String[] displayNames;
    private JTable table;

    public TablePanel(String tableName, String[] columnNames, String[] displayNames) {
        this.tableName = tableName;
        this.columnNames = columnNames;
        this.displayNames = displayNames;

        setLayout(new BorderLayout());
        initializeComponents();
    }

    private void initializeComponents() {
        // Create table model
        DefaultTableModel model = new DefaultTableModel(columnNames, rowCount:0);
        table = new JTable(model);
        table.setBackground(new Color(r:240, g:187, b:120));
        table.setForeground(new Color(r:84, g:58, b:20));
```

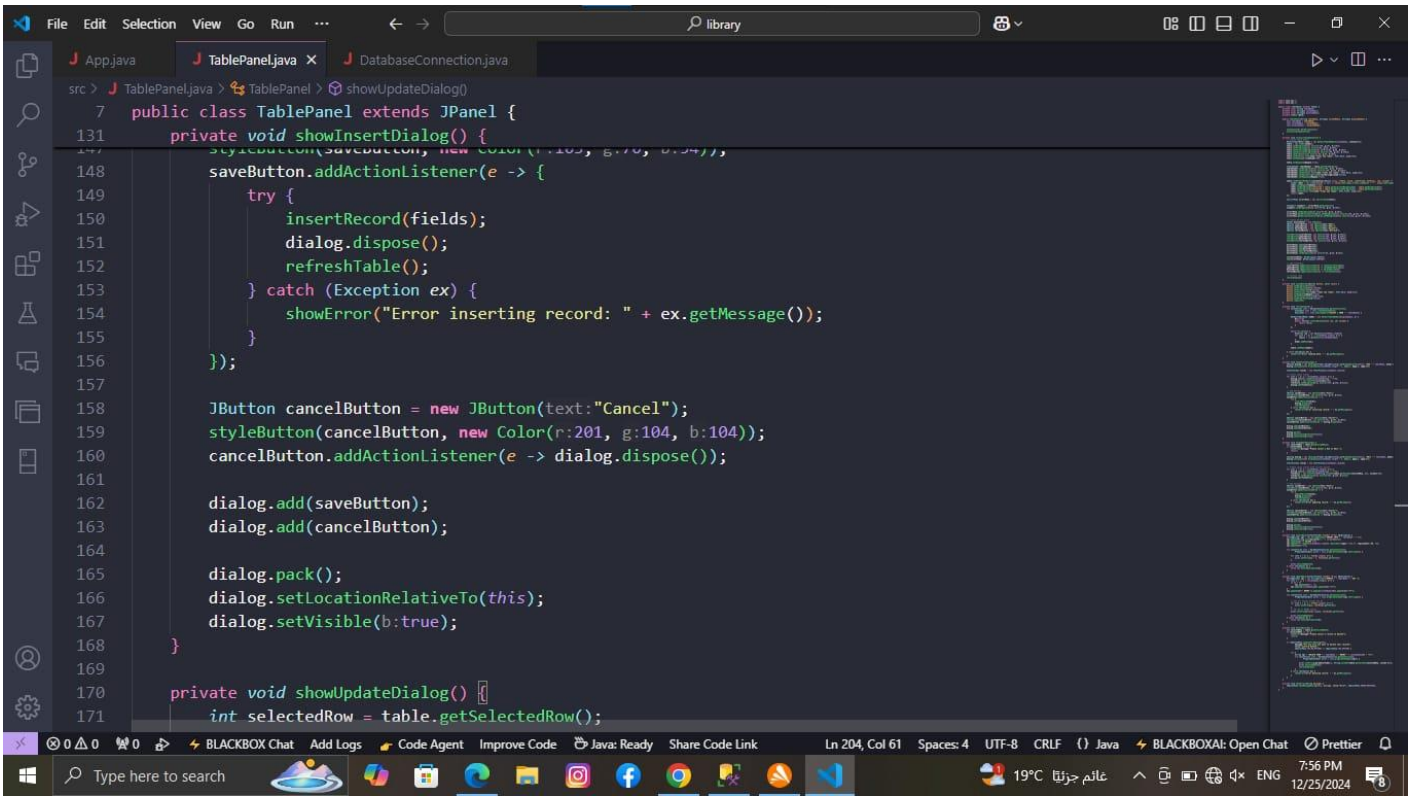

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void initializeComponents() {
table.setForeground(new Color(r:84, g:58, b:20));
table.setSelectionBackground(new Color(r:84, g:58, b:20));
table.setSelectionForeground(new Color(r:240, g:187, b:120));
table.setGridColor(new Color(r:84, g:58, b:20));
table.setFont(new Font(name:"Times new roman", Font.BOLD, size:14));
table.setRowHeight(rowHeight:30);
table.setOpaque(isOpaque:true);
JTableHeader tableHeader = table.getTableHeader();
tableHeader.setBackground(new Color(r:84, g:58, b:20));
tableHeader.setForeground(new Color(r:255, g:240, b:220));
tableHeader.setFont(new Font(name:"Times new roman", Font.BOLD, size:14));
tableHeader.setReorderingAllowed(reorderingAllowed:false);
tableHeader.setOpaque(isOpaque:true);
table.setDefaultRenderer(columnClass:Object.class, (table, value, isSelected, hasFocus, row, column) -> {
JLabel label = new JLabel(value == null || value.toString().trim().isEmpty() ? "" : value.toString());
label.setOpaque(isOpaque:true);
label.setBackground(isSelected ? table.getSelectionBackground() : table.getBackground());
label.setForeground(isSelected ? table.getSelectionForeground() : table.getForeground());
label.setFont(new Font(name:"Times new roman", Font.PLAIN, size:14));
return label;
});
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:55 PM
12/25/2024

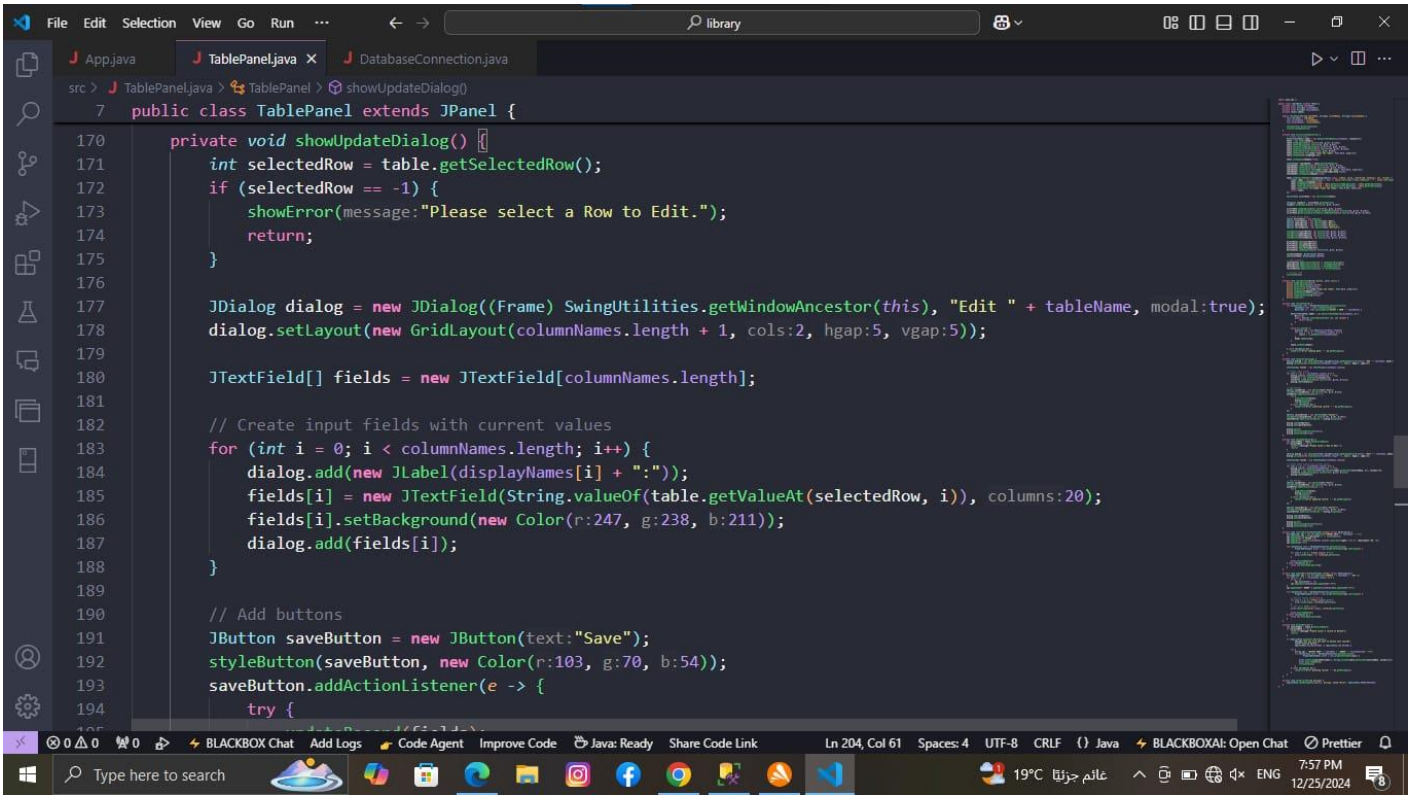

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void initializeComponents() {
JScrollPane scrollPane = new JScrollPane(table);
JViewport viewport = scrollPane.getViewport();
viewport.setBackground(new Color(r:84, g:58, b:20));
scrollPane.setBackground(new Color(r:84, g:58, b:20));
scrollPane.getVerticalScrollBar().setBackground(new Color(r:240, g:187, b:120));
scrollPane.getHorizontalScrollBar().setBackground(new Color(r:240, g:187, b:120));
// Create button panel
JPanel buttonPanel = new JPanel();
JButton insertButton = new JButton(text:"Add");
JButton updateButton = new JButton(text:"Edit");
JButton deleteButton = new JButton(text:"Delete");
JButton refreshButton = new JButton(text:"Refresh");
styleButton(insertButton, new Color(r:250, g:223, b:161));
styleButton(updateButton, new Color(r:255, g:244, b:234));
styleButton(deleteButton, new Color(r:201, g:104, b:104));
styleButton(refreshButton, new Color(r:126, g:172, b:181));
buttonPanel.add(insertButton);
buttonPanel.add(updateButton);
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:55 PM
12/25/2024

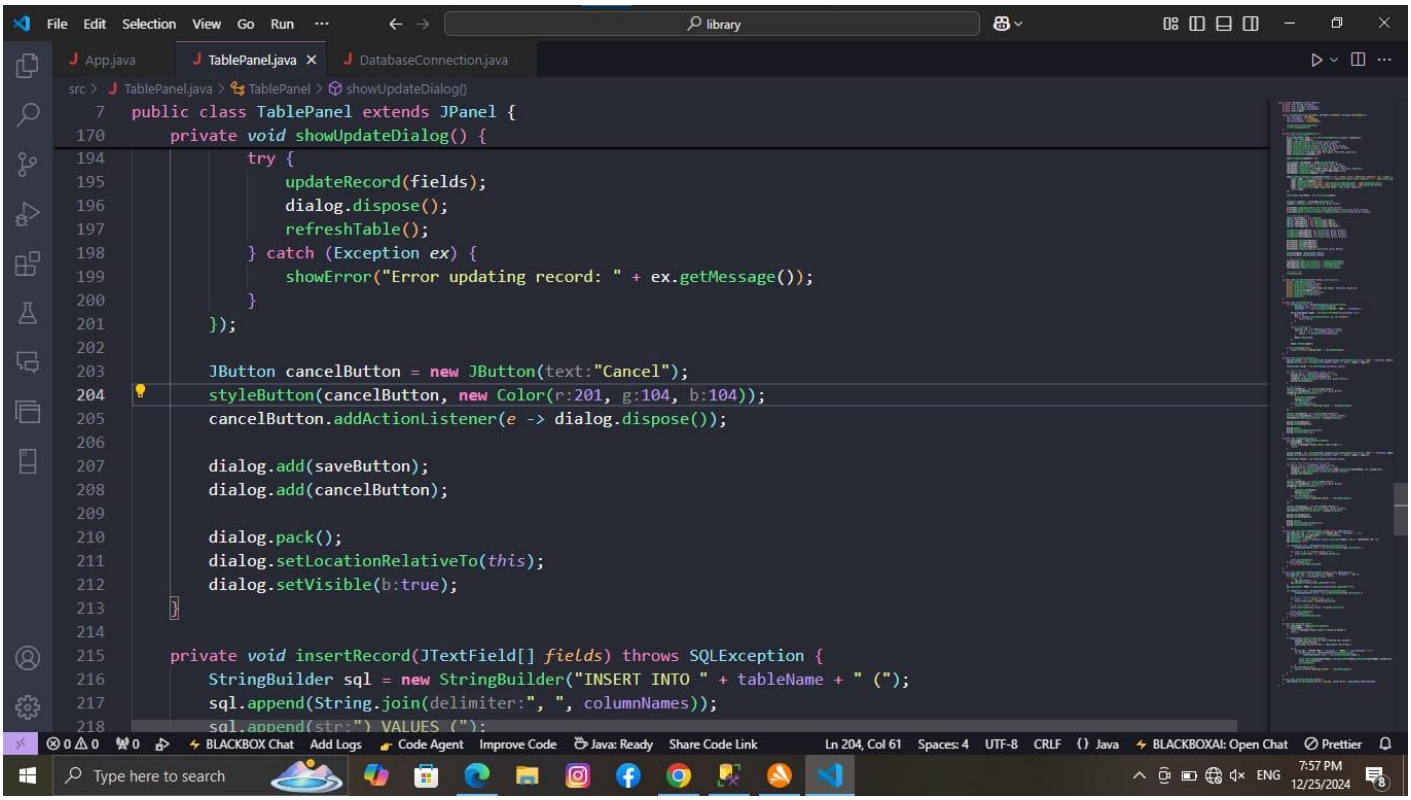

File Edit Selection View Go Run ···
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void initializeComponents() {
buttonPanel.add(deleteButton);
buttonPanel.add(refreshButton);
buttonPanel.setBackground(new Color(r:84, g:58, b:20));
add(buttonPanel, BorderLayout.NORTH);
add(scrollPane, BorderLayout.CENTER);
// Add listeners
insertButton.addActionListener(e -> showInsertDialog());
updateButton.addActionListener(e -> showUpdateDialog());
deleteButton.addActionListener(e -> deleteRecord());
refreshButton.addActionListener(e -> refreshTable());
// Initial load
refreshTable();
}
private void styleButton(JButton button, Color color) {
button.setBackground(color);
button.setForeground(Color.BLACK);
button.setFocusPainted(b:false);
button.setFont(new Font(name:"Times new roman", Font.BOLD, size:12));
button.setOpaque(isOpaque:true);
button.setContentAreaFilled(b:true);
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:55 PM
12/25/2024

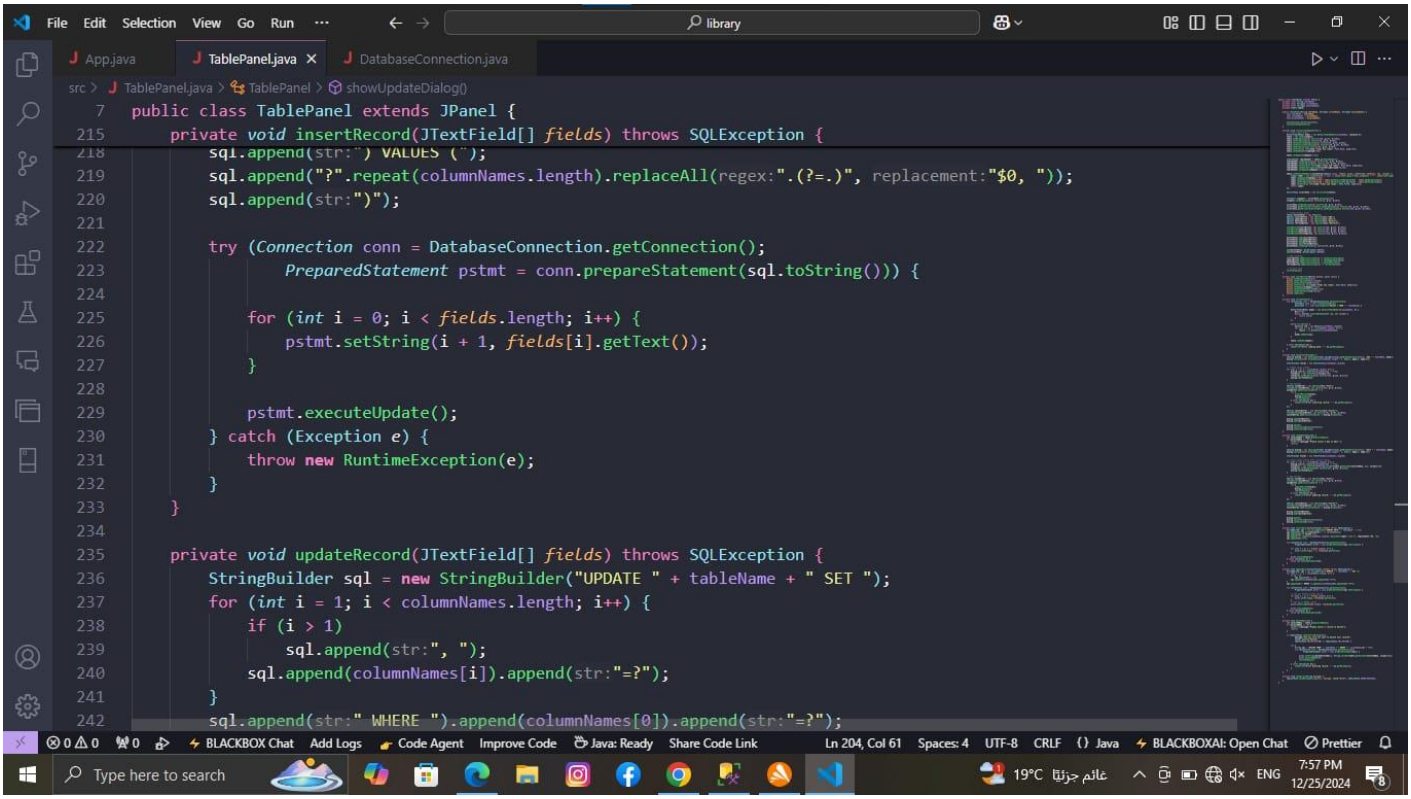

File Edit Selection View Go Run ···
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void styleButton(JButton button, Color color) {
button.setBorderPainted(b:false);
button.repaint();
}
private void refreshTable() {
try (Connection conn = DatabaseConnection.getConnection();
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM " + tableName)) {
DefaultTableModel model = new DefaultTableModel(displayNames, 0) {
@Override
public boolean isCellEditable(int row, int column) {
return false;
}
};
while (rs.next()) {
Object[] row = new Object[columnNames.length];
for (int i = 0; i < columnNames.length; i++) {
row[i] = rs.getObject(columnNames[i]);
}
model.addRow(row);
}
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:56 PM
12/25/2024

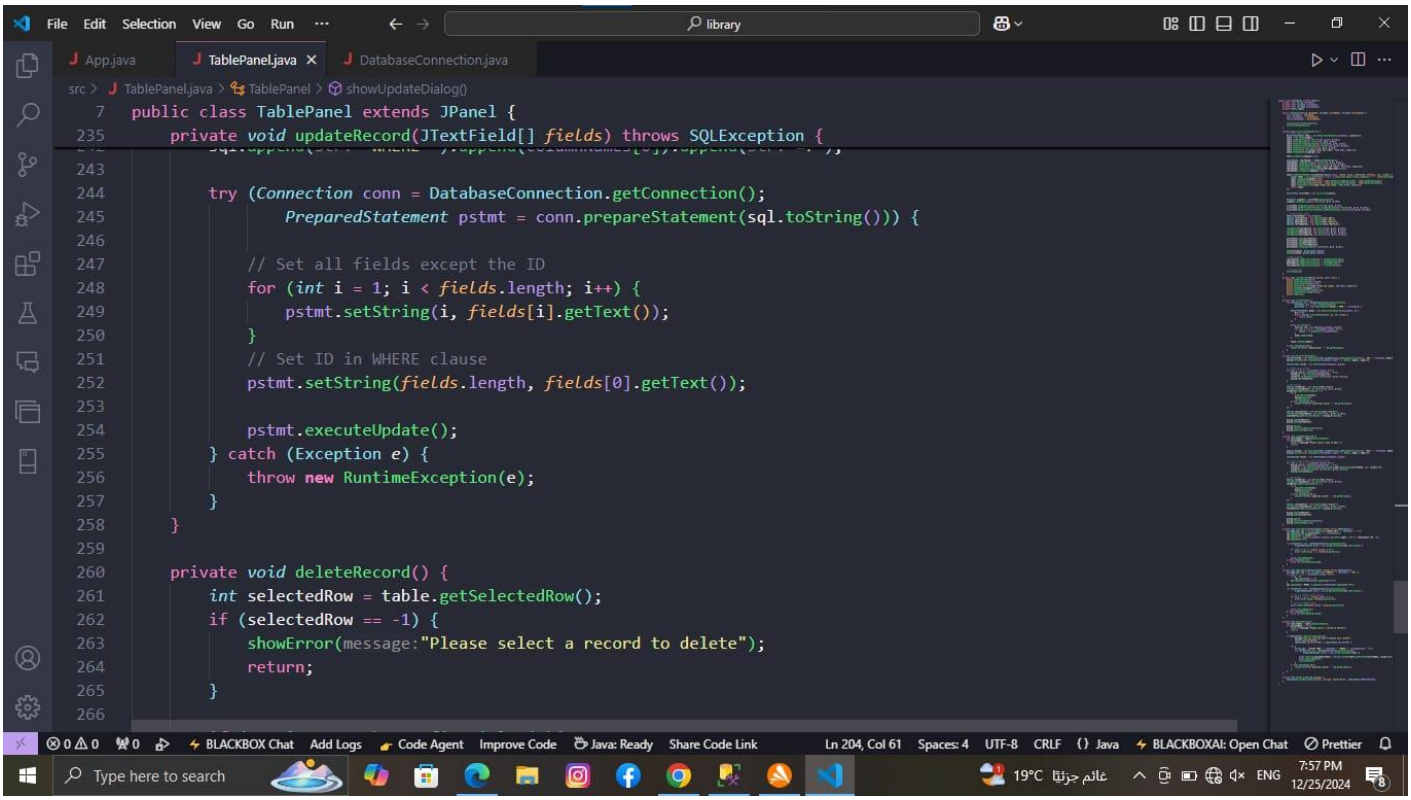

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void refreshTable() {
table.setModel(model);
} catch (Exception ex) {
showError("Error loading data: " + ex.getMessage());
}
}
private void showInsertDialog() {
JDialog dialog = new JDialog((Frame) SwingUtilities.getWindowAncestor(this), "Add " + tableName, modal:true);
dialog.setLayout(new GridLayout(columnNames.length + 1, cols:2, hgap:5, vgap:5));
JTextField[] fields = new JTextField[columnNames.length];
// Create input fields
for (int i = 0; i < columnNames.length; i++) {
dialog.add(new JLabel(displayNames[i] + ":"));
fields[i] = new JTextField(columns:20);
fields[i].setBackground(new Color(r:247, g:238, b:211));
dialog.add(fields[i]);
}
// Add buttons
JButton saveButton = new JButton(text:"Save");
styleButton(saveButton, new Color(r:103, g:70, b:54));
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:56 PM
12/25/2024

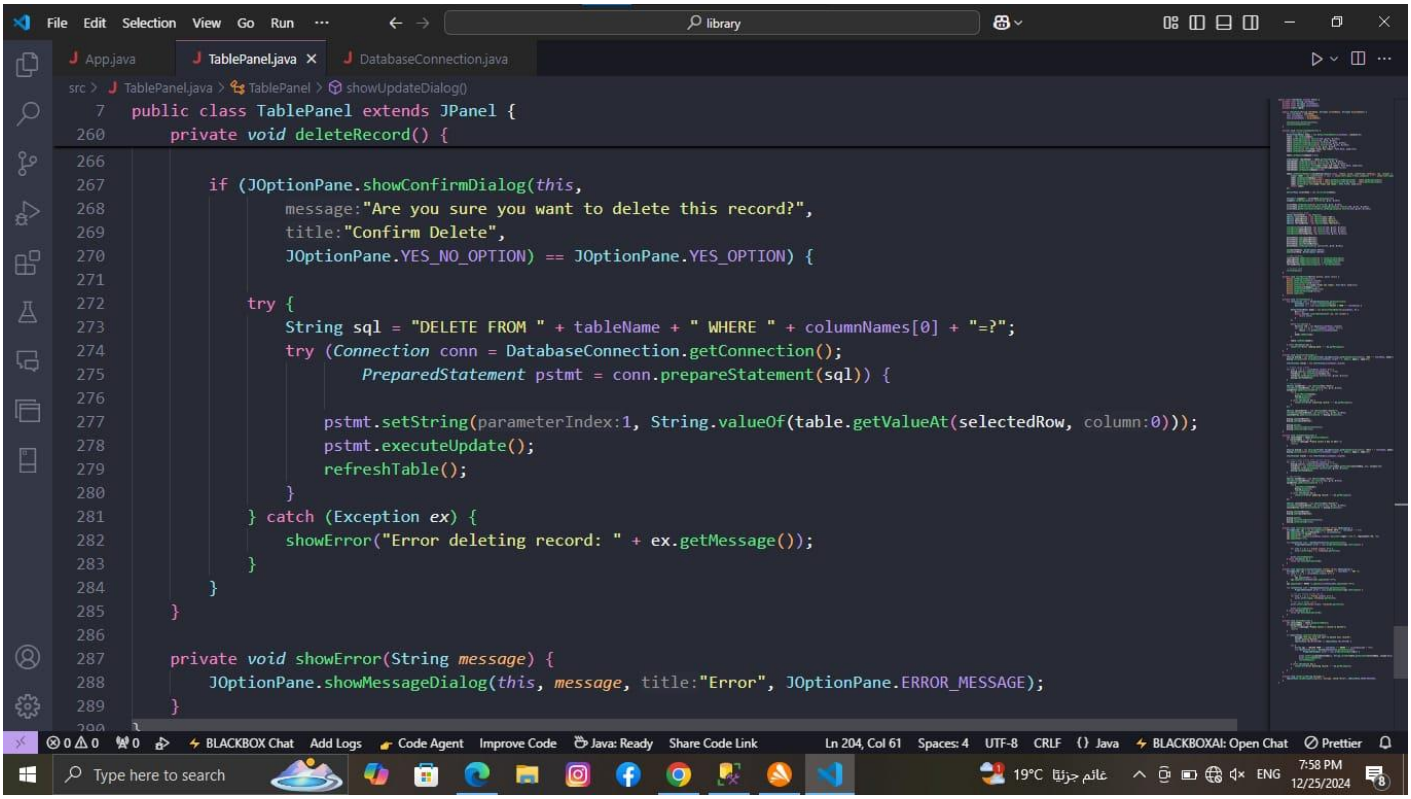

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void showInsertDialog() {
saveButton.addActionListener(e -> {
try {
insertRecord(fields);
dialog.dispose();
refreshTable();
} catch (Exception ex) {
showError("Error inserting record: " + ex.getMessage());
}
});
JButton cancelButton = new JButton(text:"Cancel");
styleButton(cancelButton, new Color(r:201, g:104, b:104));
cancelButton.addActionListener(e -> dialog.dispose());
dialog.add(saveButton);
dialog.add(cancelButton);
dialog.pack();
dialog.setLocationRelativeTo(this);
dialog.setVisible(b:true);
}
private void showUpdateDialog() {
int selectedRow = table.getSelectedRow();
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:56 PM
12/25/2024

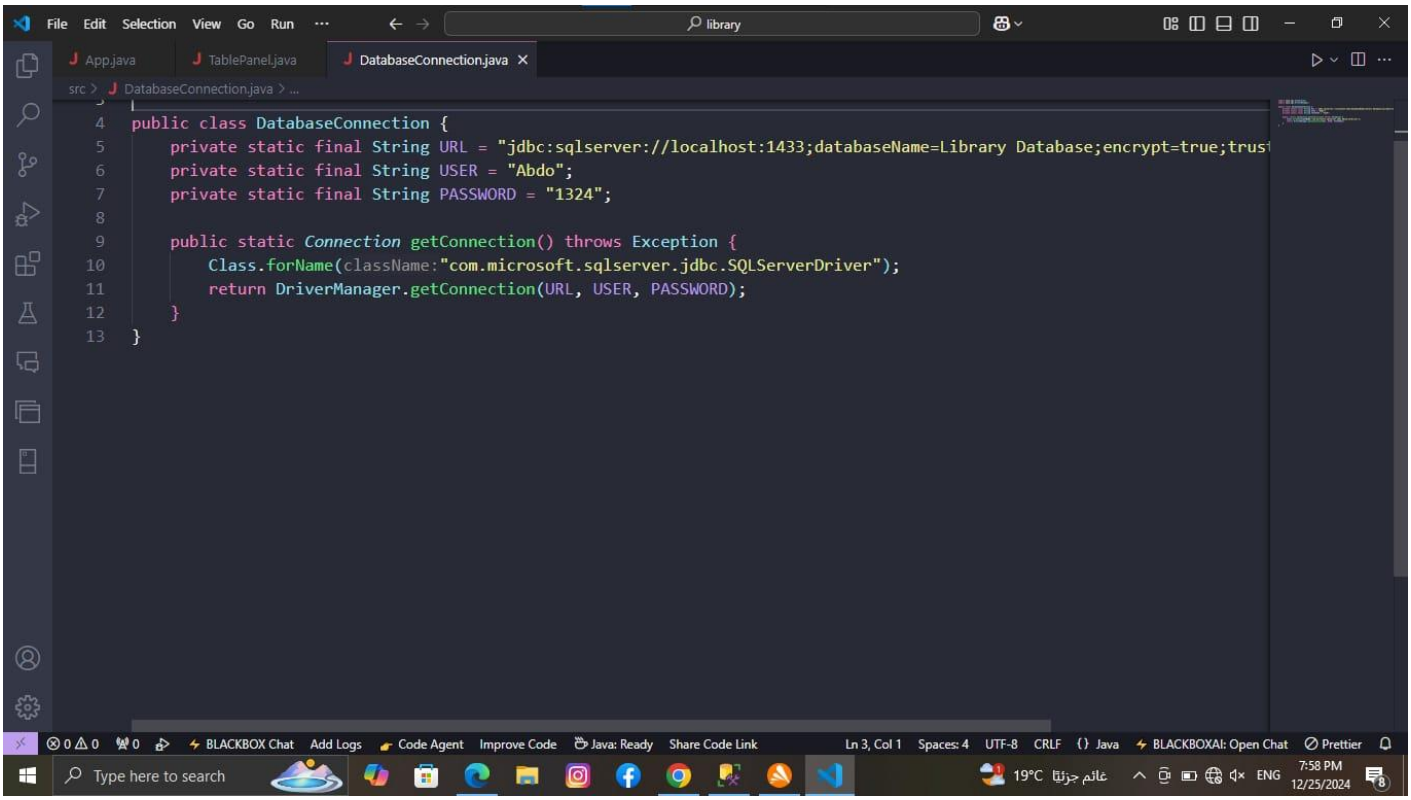

File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void showUpdateDialog() {
int selectedRow = table.getSelectedRow();
if (selectedRow == -1) {
showError(message:"Please select a Row to Edit.");
return;
}
JDialog dialog = new JDialog((Frame) SwingUtilities.getWindowAncestor(this), "Edit " + tableName, modal:true);
dialog.setLayout(new GridLayout(columnNames.length + 1, cols:2, hgap:5, vgap:5));
JTextField[] fields = new JTextField[columnNames.length];
// Create input fields with current values
for (int i = 0; i < columnNames.length; i++) {
dialog.add(new JLabel(displayNames[i] + ":"));
fields[i] = new JTextField(String.valueOf(table.getValueAt(selectedRow, i)), columns:20);
fields[i].setBackground(new Color(r:247, g:238, b:211));
dialog.add(fields[i]);
}
// Add buttons
JButton saveButton = new JButton(text:"Save");
styleButton(saveButton, new Color(r:103, g:70, b:54));
saveButton.addActionListener(e -> {
try {
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:57 PM
12/25/2024


File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void showUpdateDialog() {
try {
updateRecord(fields);
dialog.dispose();
refreshTable();
} catch (Exception ex) {
showError("Error updating record: " + ex.getMessage());
}
});
JButton cancelButton = new JButton(text:"Cancel");
styleButton(cancelButton, new Color(r:201, g:104, b:104));
cancelButton.addActionListener(e -> dialog.dispose());
dialog.add(saveButton);
dialog.add(cancelButton);
dialog.pack();
dialog.setLocationRelativeTo(this);
dialog.setVisible(b:true);
}
private void insertRecord(JTextField[] fields) throws SQLException {
StringBuilder sql = new StringBuilder("INSERT INTO " + tableName + " (");
sql.append(String.join(delimiter:", ", columnNames));
sql.append(str:") VALUES (");
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
ENG
7:57 PM
12/25/2024


File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void insertRecord(JTextField[] fields) throws SQLException {
sql.append(") VALUES (");
sql.append("?".repeat(columnNames.length).replaceAll(".(?=.)", "$0, "));
sql.append(")");
try (Connection conn = DatabaseConnection.getConnection();
PreparedStatement pstmt = conn.prepareStatement(sql.toString())) {
for (int i = 0; i < fields.length; i++) {
pstmt.setString(i + 1, fields[i].getText());
}
pstmt.executeUpdate();
} catch (Exception e) {
throw new RuntimeException(e);
}
}
private void updateRecord(JTextField[] fields) throws SQLException {
StringBuilder sql = new StringBuilder("UPDATE " + tableName + " SET ");
for (int i = 1; i < columnNames.length; i++) {
if (i > 1)
sql.append(", ");
sql.append(columnNames[i]).append("=?");
}
sql.append(" WHERE ").append(columnNames[0]).append("=?");
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:57 PM
12/25/2024


File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void updateRecord(JTextField[] fields) throws SQLException {
try (Connection conn = DatabaseConnection.getConnection();
PreparedStatement pstmt = conn.prepareStatement(sql.toString())) {
// Set all fields except the ID
for (int i = 1; i < fields.length; i++) {
pstmt.setString(i, fields[i].getText());
}
// Set ID in WHERE clause
pstmt.setString(fields.length, fields[0].getText());
pstmt.executeUpdate();
} catch (Exception e) {
throw new RuntimeException(e);
}
}
private void deleteRecord() {
int selectedRow = table.getSelectedRow();
if (selectedRow == -1) {
showError("Please select a record to delete");
return;
}
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:57 PM
12/25/2024


File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > TablePanel.java > TablePanel > showUpdateDialog()
public class TablePanel extends JPanel {
private void deleteRecord() {
if (JOptionPane.showConfirmDialog(this,
message:"Are you sure you want to delete this record?",
title:"Confirm Delete",
JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
try {
String sql = "DELETE FROM " + tableName + " WHERE " + columnNames[0] + "=?";
try (Connection conn = DatabaseConnection.getConnection();
PreparedStatement pstmt = conn.prepareStatement(sql)) {
pstmt.setString(parameterIndex:1, String.valueOf(table.getValueAt(selectedRow, column:0)));
pstmt.executeUpdate();
refreshTable();
} catch (Exception ex) {
showError("Error deleting record: " + ex.getMessage());
}
}
}
private void showError(String message) {
JOptionPane.showMessageDialog(this, message, title:"Error", JOptionPane.ERROR_MESSAGE);
}
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 204, Col 61 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:58 PM
12/25/2024


File Edit Selection View Go Run
library
App.java
TablePanel.java
DatabaseConnection.java
src > DatabaseConnection.java > ...
public class DatabaseConnection {
private static final String URL = "jdbc:sqlserver://localhost:1433;databaseName=Library Database;encrypt=true;trust
private static final String USER = "Abdo";
private static final String PASSWORD = "1324";
public static Connection getConnection() throws Exception {
Class.forName(className:"com.microsoft.sqlserver.jdbc.SQLServerDriver");
return DriverManager.getConnection(URL, USER, PASSWORD);
}
}
BLACKBOX Chat Add Logs Code Agent Improve Code Java: Ready Share Code Link
Ln 3, Col 1 Spaces: 4 UTF-8 CRLF Java BLACKBOXAI: Open Chat Prettier
Type here to search
19°C
ENG
7:58 PM
12/25/2024