**The University of Michigan - Dearborn**

**Department of Computer and Information Science**

**Dr. Raed Almomani CIS421 Database Management Systems Winter2023**

**Relational Database Project**

**Due: April 7, 2023 at 11:59PM**

**Presentation Days: April 10 and April 12**

**Total points: 50**

**1-The team members:**

**-Noureddine Ouelhaci**

**-Zeinab Sabra**

**-Hira Sharif**

**2-The application background description:**

A book-selling business needs to keep track of its inventory, customers, orders, and reviews. They require a system "Database" that can manage and organize this information in an efficient and effective manner. For inventory, they need to keep track of information such as the book title, author, publisher, publication date, price, and quantity available.

They also need a system that stores customer information such as their name, address, email, password, credit card details "credit card number- expiry date- cvs number-card holder name", and phone

number. Each customer can place one or more orders and write multiple reviews. Each item should be linked to a book and include the quantity ordered and the price at the time of the purchase.

For orders, they need to track the order date and total cost of each transaction. Each order can have one or more items, and each item can be associated with one order. Additionally, they need a system to track customer reviews of books, including the book ID, customer ID, rating, and review text. Each review should be associated with one book.

## 3-The database requirements:

### a- Tables:

**The inventory data includes book title, author, publisher, publication date, price, and quantity available:**

The database should have a "Books" table/entity that stores information about books available for sale, including book_ID, book_title, author, publisher, publication_date, price, and quantity in stock.

**Customer data should include name, address, email, password, credit card details, and phone number:**

The database should have a "Customers" table/entity that stores information about customers, including customer_ID, name, address, email, password, credit_card_number, expiry_date, cvs_number, cardholder_name, and phonenumber.

**The order data includes the order date, total cost, and item details:**

The database should have an "Orders" table/entity that tracks customer orders and is linked to both the "Customers" and "Books" entities. This table should store information about the order_date, total_cost of each transaction, customer_ID, and order_ID.

The database should have an "Order Details" table/entity that stores information about the books, including the book_ID, quantity, and price, Order_ID, Order_Detail_ID, and is linked to the "Orders" and "Books" entities.

**The review data should include book ID, customer ID, rating, and review text:**

The database should have a "Reviews" table/entity that stores information about customer reviews of books and is linked to both the "Customers" and "Books" entities. This table should contain information such as the Review_ID, book_ID, customer_ID, rating, and review_text.

**b- Relationships:**

Each customer can place one or more orders and write multiple reviews.

Each item should be linked to a book and include the quantity ordered and the price at the time of the purchase.

Each order can have one or more items, and each item can be associated with one order.

Each review should be associated with one book.

**c- Database Management:**

The database should enforce referential integrity to ensure that each order is associated with a specific customer and that each item is linked to the correct book.

The database should have appropriate indexes to optimize performance, such as indexes on book title, author, and customer name.

The database should have appropriate security measures in place to protect customer data and prevent unauthorized access to the system.

## 4-The ER diagram:

**Relations explanations:**

-in the context of the ER diagram for the book-selling business, "Each customer can place one or more orders" means that there is a one-to-many relationship between the "Customers" table and the "Orders" table. This relationship can be represented in the ER diagram by creating a foreign key in the "Orders" table that references the primary key in the "Customers" table.

This relationship allows the book-selling business to keep track of which customer placed each order and enables them to retrieve a customer's order history. For example, if a customer wants to see all the orders they have placed with the business, the database

can be queried to retrieve all orders associated with that customer's ID.

- In the context of the ER diagram for the book-selling business, "Each order can have one or more items, and each item can be associated with one order" means that there is a one-to-many relationship between the "Orders" table and the "Order Details" table. This relationship can be represented in the ER diagram by creating a foreign key in the "Order Details" table that references the primary key in the "Orders" table.

This relationship allows the book-selling business to keep track of the books that are associated with each order. For example, if an order consists of three books, there would be three records in the "Order Details" table, each linked to the same order by the order's ID. This allows the book-selling business to retrieve all books associated with a specific order, and enables them to calculate the total cost of the order based on the prices of each book.

- In the context of the ER diagram for the book-selling business, "Each review should be associated with one book" means that there is a one-to-many relationship between the "Books" table and the "Reviews" table. This relationship can be represented in the ER diagram by creating a foreign key in the "Reviews" table that references the primary key in the "Books" table.

This relationship allows the book-selling business to keep track of which book a review is associated with. For example, if a customer writes a review for a particular book, the database can be queried to retrieve all reviews associated with that book's ID.

This enables the book-selling business to display reviews for each book on their website or in their store, allowing customers to make informed purchasing decisions.

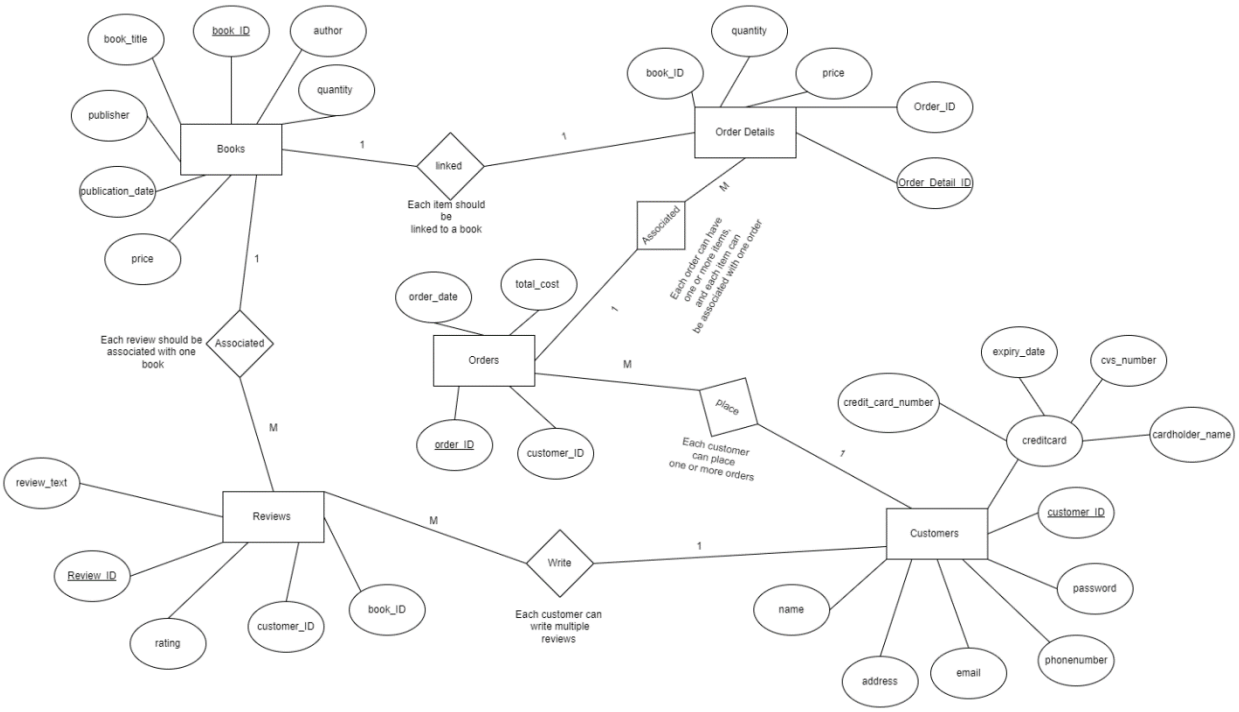
- In the context of the ER diagram for the book-selling business, "Each customer can write multiple reviews" means that there is a one-to-many relationship between the "Customers" table and the "Reviews" table. This relationship can be represented in the ER diagram by creating a foreign key in the "Reviews" table that references the primary key in the "Customers" table.

This relationship allows the book-selling business to keep track of which customer wrote each review. For example, if a customer writes multiple reviews, each review would be associated with that customer's ID in the "Reviews" table. This enables the book-selling business to identify which customers have written reviews, and allows them to contact customers for follow-up or promotional purposes.

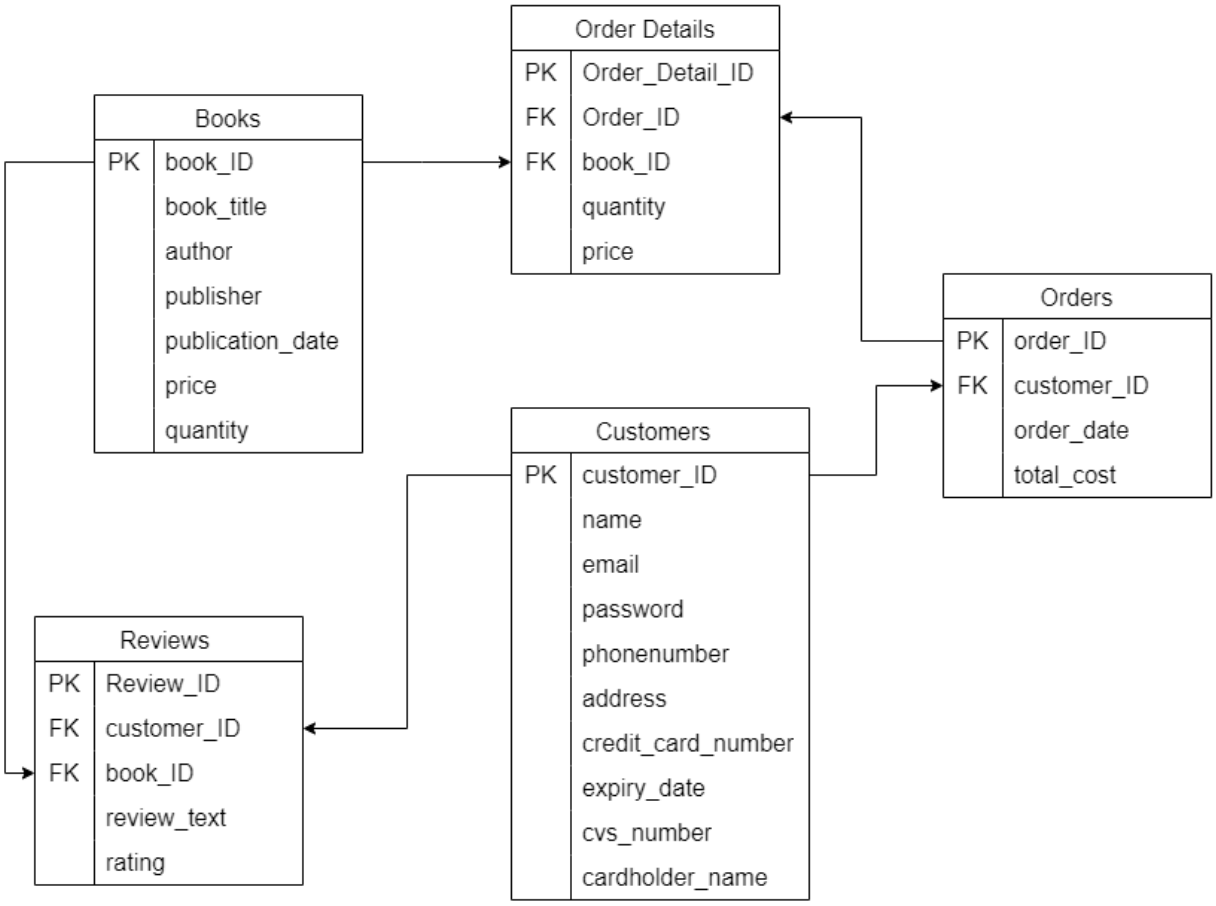
- In the context of the ER diagram for the book-selling business, "Each item should be linked to a book" means that there is a one-to-one relationship between the "Order Details" table and the "Books" table. This relationship can be represented in the ER diagram by creating a foreign key in the "Order Details" table that references the primary key in the "Books" table.

This relationship allows the book-selling business to keep track of which book each item is associated with. For example, if a customer orders an item, the database can be queried to retrieve the book associated with that item's ID "Order Details ID". This

enables the book-selling business to ensure that the correct book is shipped to the customer, and to maintain accurate inventory records.



## 5-The relational database schema:




**SQL DDL statements used to define the database in SQLITE:**

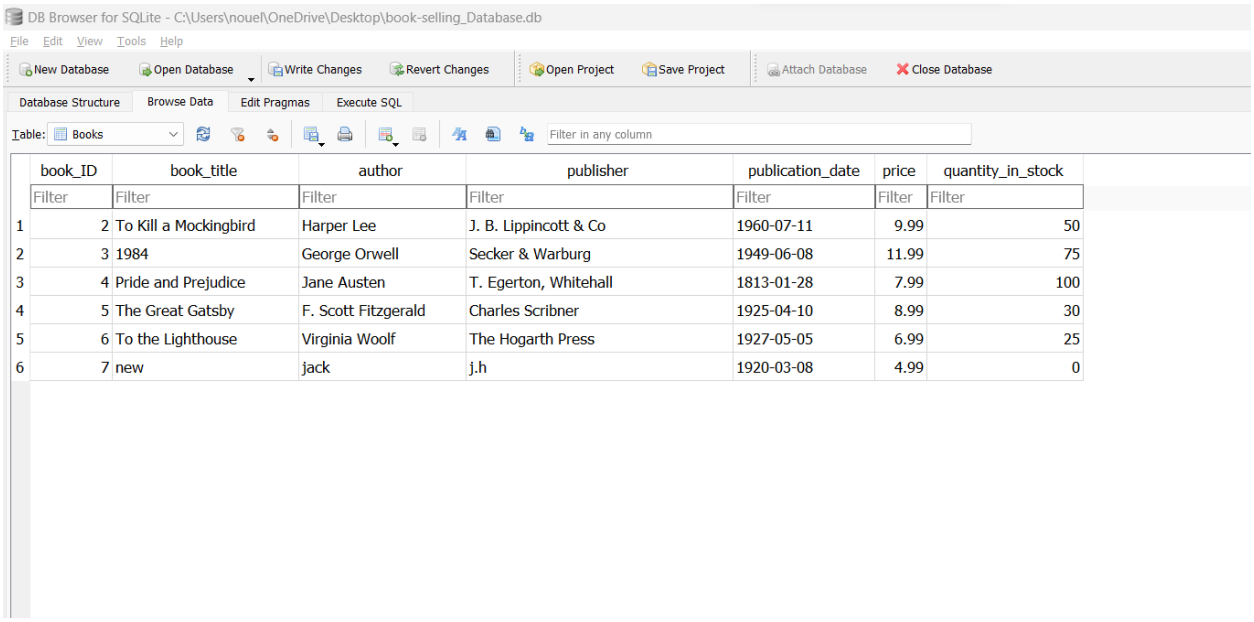
```
CREATE TABLE "Books " (
        "book_ID "    INTEGER NOT NULL,
        "book_title "  TEXT NOT NULL UNIQUE,
        "author "      TEXT NOT NULL,
        "publisher "  TEXT NOT NULL,
        "publication_date" TEXT NOT NULL,
        "price "        REAL NOT NULL,
        "quantity_in_stock "        INTEGER NOT NULL,
        PRIMARY KEY("book_ID " AUTOINCREMENT)
```

```
);
CREATE TABLE "Customers" (
        "customer_ID "      INTEGER NOT NULL,
        "name"        TEXT NOT NULL,
        "email "        TEXT NOT NULL UNIQUE,
        "password " TEXT NOT NULL,
        "phonenumber "    TEXT NOT NULL UNIQUE,
        "address "    TEXT NOT NULL,
        "credit_card_number "    INTEGER NOT NULL,
        "expiry_date "        TEXT NOT NULL,
        "cvs_number "       INTEGER NOT NULL,
        "cardholder_name "        TEXT NOT NULL,
        PRIMARY KEY("customer_ID " AUTOINCREMENT)
);
CREATE TABLE "Order_Details" (
        "Order_Detail_ID "INTEGER NOT NULL,
        "Order_ID "  INTEGER NOT NULL,
        "book_ID"    INTEGER NOT NULL,
        "quantity "   INTEGER NOT NULL,
        "price "        REAL NOT NULL,
        PRIMARY KEY("Order_Detail_ID " AUTOINCREMENT),
        FOREIGN KEY("Order_ID ") REFERENCES "Orders "("order_ID "),
        FOREIGN KEY("book_ID") REFERENCES "Books "("book_ID ")
);
CREATE TABLE "Orders " (
```

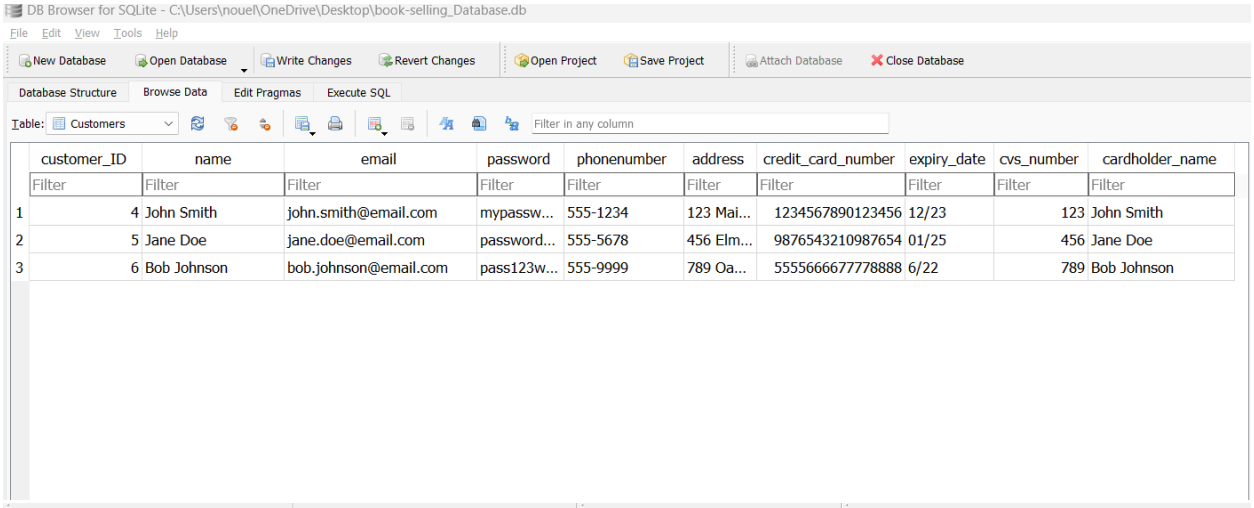
```
        "order_ID "  INTEGER NOT NULL,
        "customer_ID "      INTEGER NOT NULL,
        "order_date "       TEXT NOT NULL,
        "total_cost" REAL NOT NULL,
        FOREIGN KEY("customer_ID ") REFERENCES "Customers"("customer_ID "),
        PRIMARY KEY("order_ID " AUTOINCREMENT)
);
CREATE TABLE "Reviews " (
        "Review_ID "        INTEGER NOT NULL,
        "customer_ID "      INTEGER NOT NULL,
        "book_ID "  INTEGER NOT NULL,
        "review_text"       TEXT,
        "rating "       INTEGER,
        PRIMARY KEY("Review_ID " AUTOINCREMENT),
        FOREIGN KEY("customer_ID ") REFERENCES "Customers"("customer_ID "),
        FOREIGN KEY("book_ID ") REFERENCES "Books "("book_ID ")
```

);

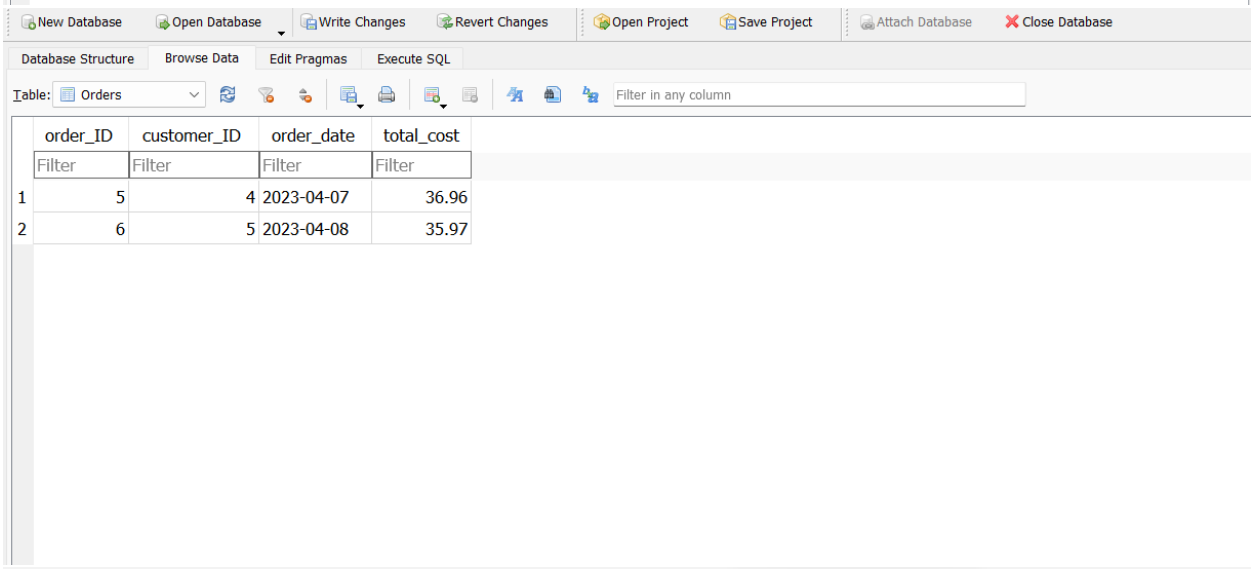
## 6-The sample database:



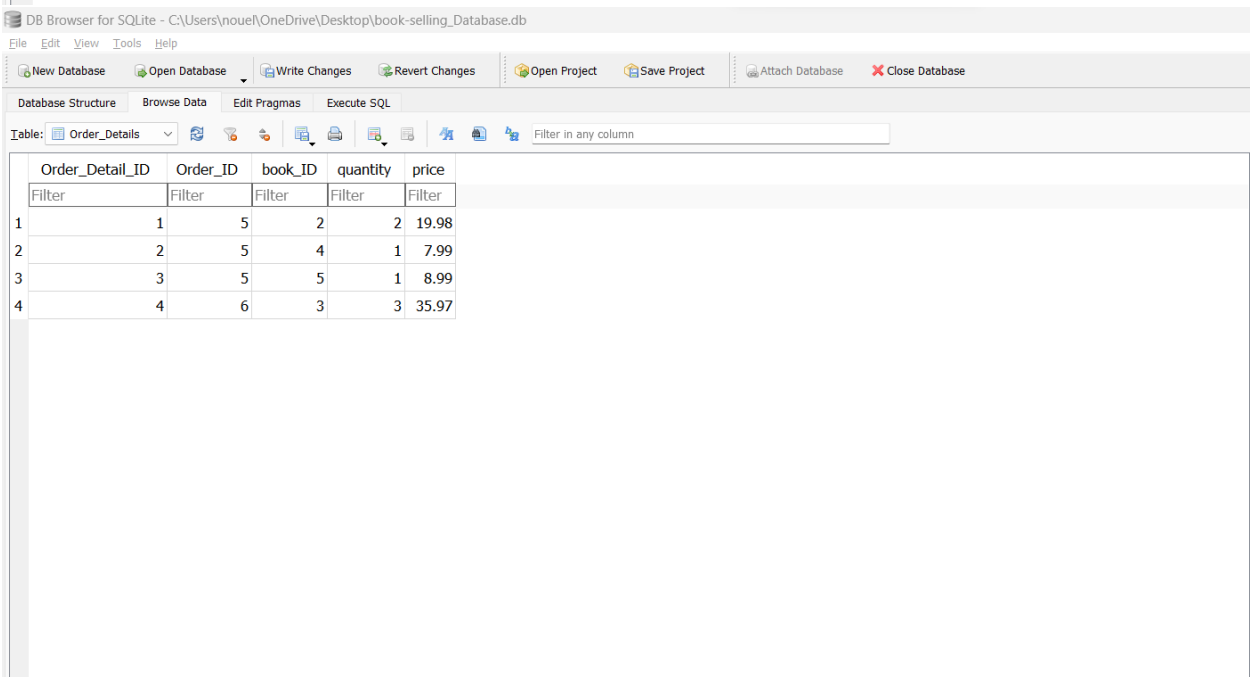

| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1984 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |



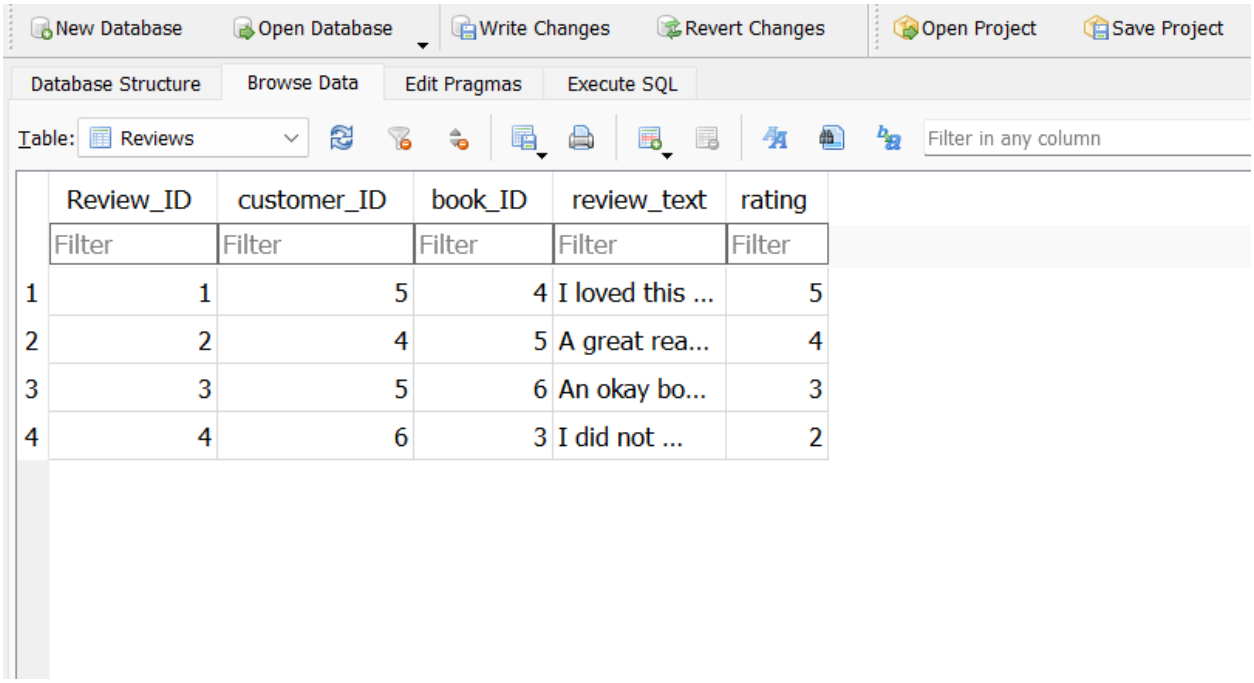

| | customer_ID | name | email | password | phonenumber | address | credit_card_number | expiry_date | cvs_number | cardholder_name |
|---|---|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 4 | John Smith | john.smith@email.com | mypassw... | 555-1234 | 123 Mai... | 1234567890123456 | 12/23 | 123 | John Smith |
| 2 | 5 | Jane Doe | jane.doe@email.com | password... | 555-5678 | 456 Elm... | 9876543210987654 | 01/25 | 456 | Jane Doe |
| 3 | 6 | Bob Johnson | bob.johnson@email.com | pass123w... | 555-9999 | 789 Oa... | 5555666677778888 | 6/22 | 789 | Bob Johnson |




| | order_ID | customer_ID | order_date | total_cost |
|---|---|---|---|---|
| | Filter | Filter | Filter | Filter |
| 1 | 5 | 4 | 2023-04-07 | 36.96 |
| 2 | 6 | 5 | 2023-04-08 | 35.97 |




| | Order_Detail_ID | Order_ID | book_ID | quantity | price |
|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 5 | 2 | 2 | 19.98 |
| 2 | 2 | 5 | 4 | 1 | 7.99 |
| 3 | 3 | 5 | 5 | 1 | 8.99 |
| 4 | 4 | 6 | 3 | 3 | 35.97 |




| | Review_ID | customer_ID | book_ID | review_text | rating |
|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 5 | 4 | I loved this ... | 5 |
| 2 | 2 | 4 | 5 | A great rea... | 4 |
| 3 | 3 | 5 | 6 | An okay bo... | 3 |
| 4 | 4 | 6 | 3 | I did not ... | 2 |

## 7-The SQL statements:

INSERT INTO "main"."Customers"

("name", "email ", "password ", "phonenumber ", "address ", "credit_card_number ", "expiry_date ", "cvs_number ", "cardholder_name ")

VALUES ('John Smith', 'john.smith@email.com', 'mypassword', '555-1234', '123 Main St, Anytown USA', 1234567890123456, '12/23', 123, 'John Smith');

INSERT INTO "main"."Customers"

("name", "email ", "password ", "phonenumber ", "address ", "credit_card_number ", "expiry_date ", "cvs_number ", "cardholder_name ")

VALUES ('Jane Doe', 'jane.doe@email.com', 'password123', '555-5678', '456 Elm St, Anycity USA', 9876543210987654, '01/25', 456, 'Jane Doe');

INSERT INTO "main"."Customers"

("name", "email ", "password ", "phonenumber ", "address ", "credit_card_number ", "expiry_date ", "cvs_number ", "cardholder_name ")

VALUES ('Bob Johnson', 'bob.johnson@email.com', 'pass123word', '555-9999', '789 Oak St, Anyville USA', 5555666677778888, '6/22', 789, 'Bob Johnson');

INSERT INTO "main"."Books "

("book_title ", "author ", "publisher ", "publication_date", "price ", "quantity_in_stock ")

VALUES ('To Kill a Mockingbird', 'Harper Lee', 'J. B. Lippincott & Co', '1960-07-11', 9.99, 50);

INSERT INTO "main"."Books "

("book_title ", "author ", "publisher ", "publication_date", "price ", "quantity_in_stock ")

VALUES ('1984', 'George Orwell', 'Secker & Warburg', '1949-06-08', 11.99, 75);

INSERT INTO "main"."Books "

("book_title ", "author ", "publisher ", "publication_date", "price ", "quantity_in_stock ")

VALUES ('Pride and Prejudice', 'Jane Austen', 'T. Egerton, Whitehall', '1813-01-28', 7.99, 100);

INSERT INTO "main"."Books "

("book_title ", "author ", "publisher ", "publication_date", "price ", "quantity_in_stock ")

VALUES ('The Great Gatsby', 'F. Scott Fitzgerald', 'Charles Scribner', '1925-04-10', 8.99, 30);

INSERT INTO "main"."Books "

("book_title ", "author ", "publisher ", "publication_date", "price ", "quantity_in_stock ")

VALUES ('To the Lighthouse', 'Virginia Woolf', 'The Hogarth Press', '1927-05-05', 6.99, 25);

INSERT INTO "main"."Orders "

("customer_ID ", "order_date ", "total_cost")

VALUES (4, '2023-04-07', 36.96);

INSERT INTO "main"."Orders "

("customer_ID ", "order_date ", "total_cost")

VALUES (5, '2023-04-08', 35.97);

INSERT INTO "main"."Order_Details"

("Order_ID ", "book_ID", "quantity ", "price ")

VALUES (5, 2, 2, 19.98);

INSERT INTO "main"."Order_Details"

("Order_ID ", "book_ID", "quantity ", "price ")

VALUES (5, 4, 1, 7.99);

INSERT INTO "main"."Order_Details"

("Order_ID ", "book_ID", "quantity ", "price ")

VALUES (5, 5, 1, 8.99);

```
INSERT INTO "main"."Order_Details"
("Order_ID ", "book_ID", "quantity ", "price ")
VALUES (6, 3, 3, 35.97);

INSERT INTO "main"."Reviews "
("customer_ID ", "book_ID ", "review_text", "rating ")
VALUES (5, 4, 'I loved this book! The characters were so well developed', 5);

INSERT INTO "main"."Reviews "
("customer_ID ", "book_ID ", "review_text", "rating ")
VALUES (4, 5, 'A great read, I couldn''t put it down', 4);

INSERT INTO "main"."Reviews "
("customer_ID ", "book_ID ", "review_text", "rating ")
VALUES (5, 6, 'An okay book, not my favorite but it was interesting', 3);

INSERT INTO "main"."Reviews "
("customer_ID ", "book_ID ", "review_text", "rating ")
VALUES (6, 3, 'I did not enjoy this book, the plot was slow and the characters were boring.', 2);
```

**The following are examples of the different types of the SQL statements:**

For testing purposes I created a new table because I didn't want to delete an important table from the database.

a-Create Table:

```
CREATE TABLE "New" (
	"new_ID"	INTEGER NOT NULL,
	"firstattribute"	TEXT NOT NULL,
	"secattribute"	TEXT NOT NULL UNIQUE,
	PRIMARY KEY("new_ID" AUTOINCREMENT)
);
```

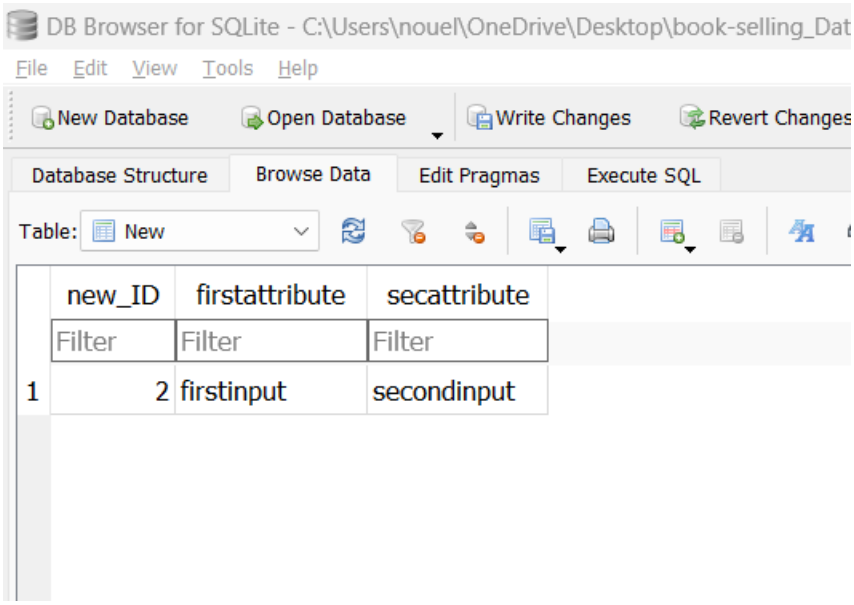
b-Insert:

```
INSERT INTO "main"."New"
("firstattribute", "secattribute")
VALUES ('firstinput', 'secondinput');
```

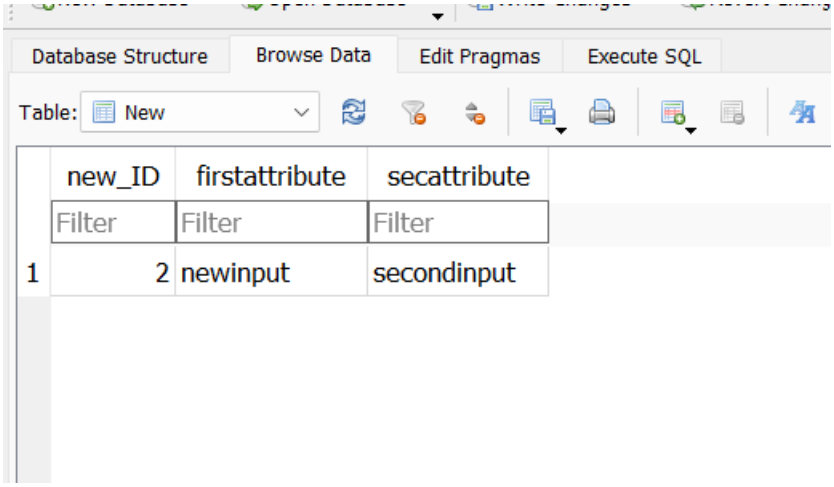
c-Update Row:

```
UPDATE "main"."New" SET "firstattribute" = "newinput" WHERE "new_ID" = 2;
```

Before:

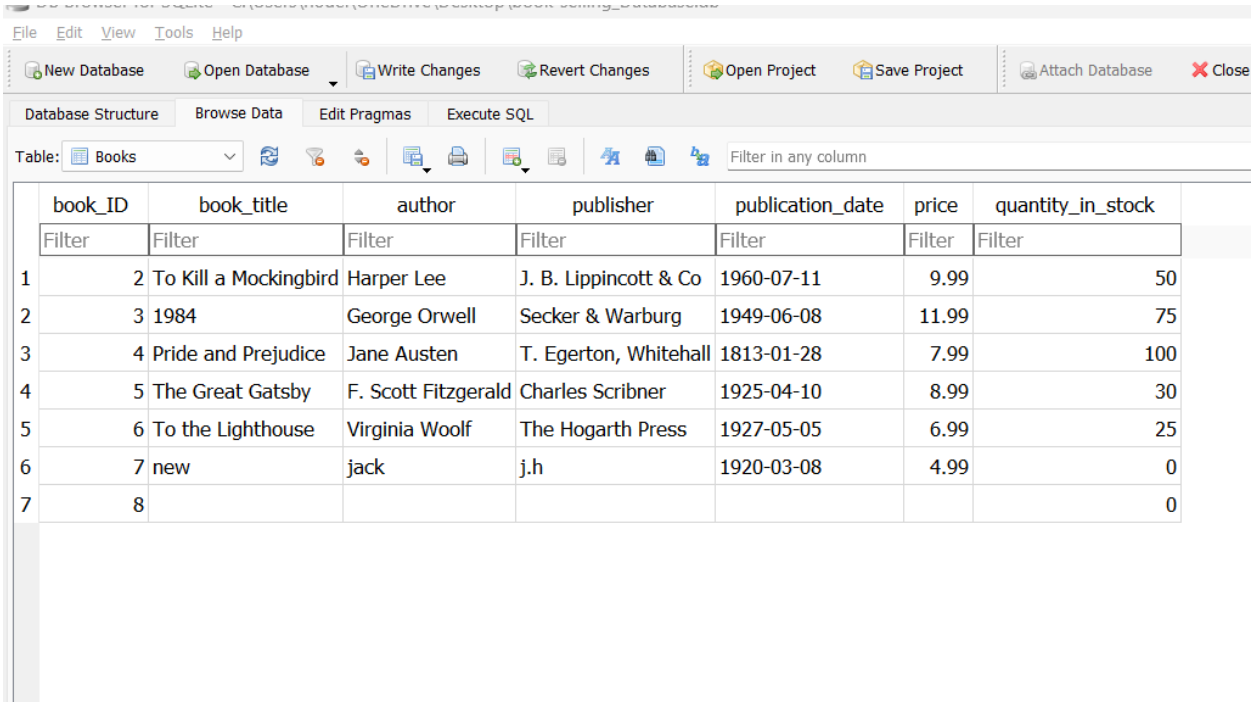



After:



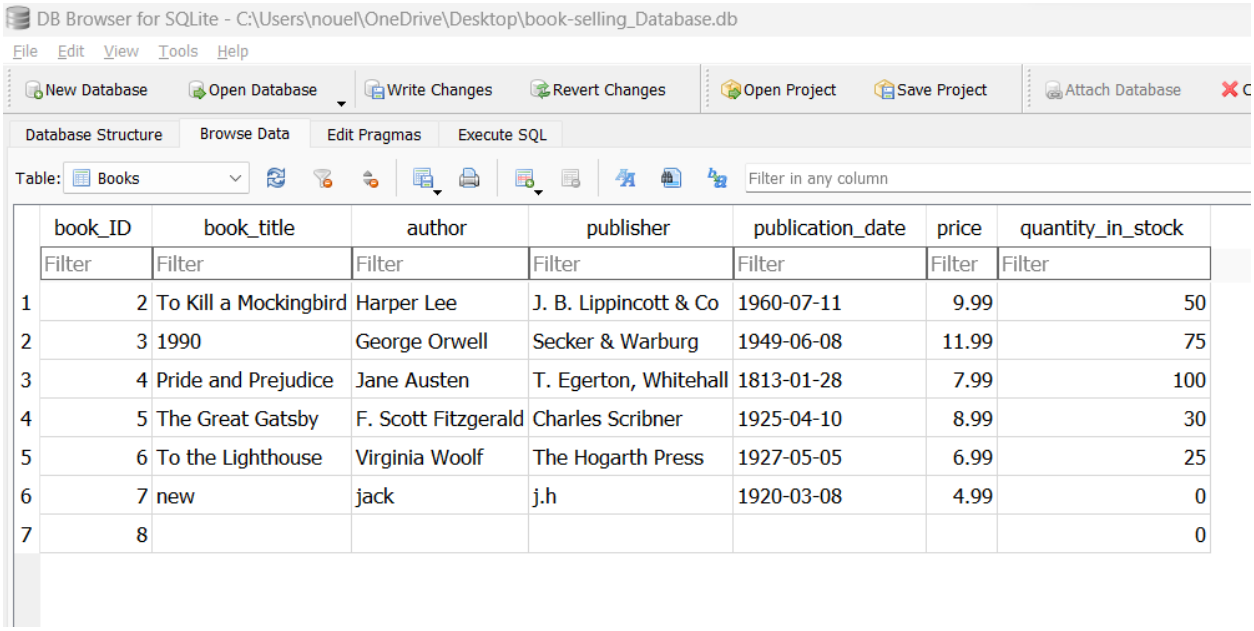

UPDATE "main"."Books " SET "book_title " = "1990" WHERE "book_ID " = 3;

Before:




| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1984 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 7 | 8 | | | | | | 0 |

After:

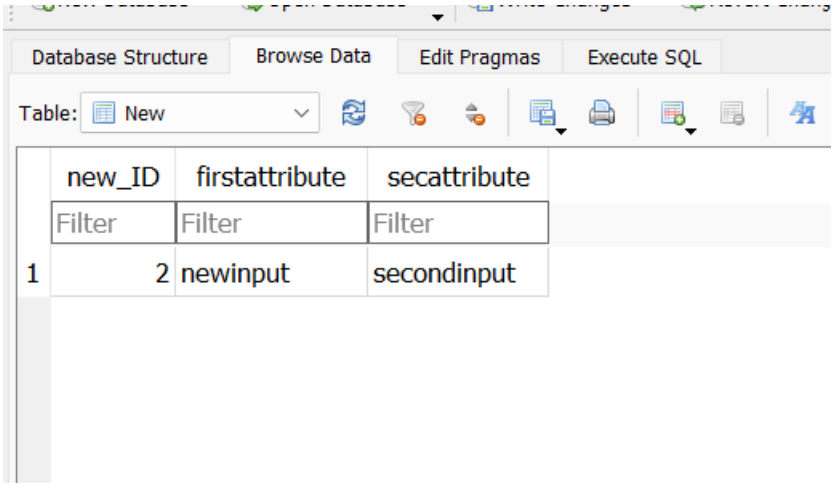



| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 7 | 8 | | | | | | 0 |

d-Delete:

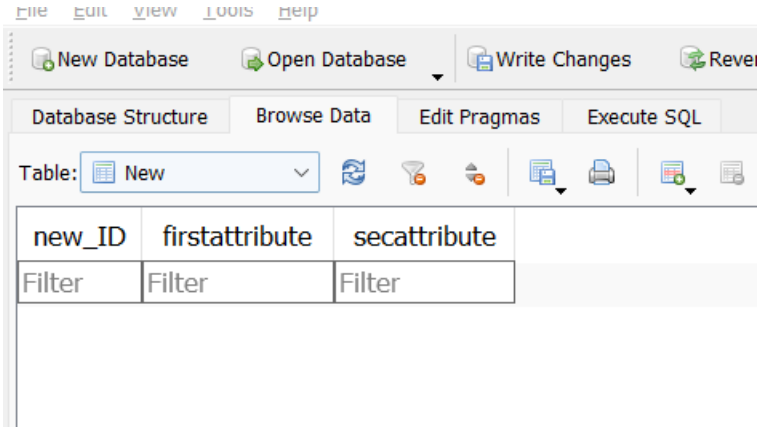
DELETE FROM "main"."New" WHERE "new_ID" = 2;

Before:



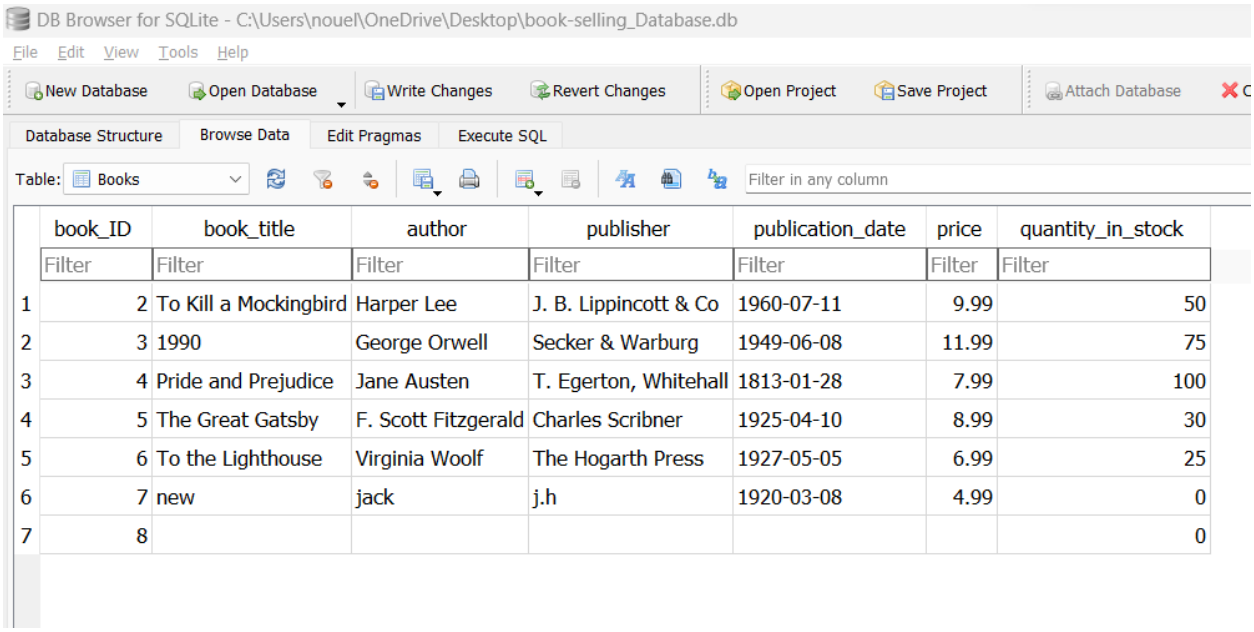

| | new_ID | firstattribute | secattribute |
|---|---|---|---|
| | Filter | Filter | Filter |
| 1 | 2 | newinput | secondinput |

After:



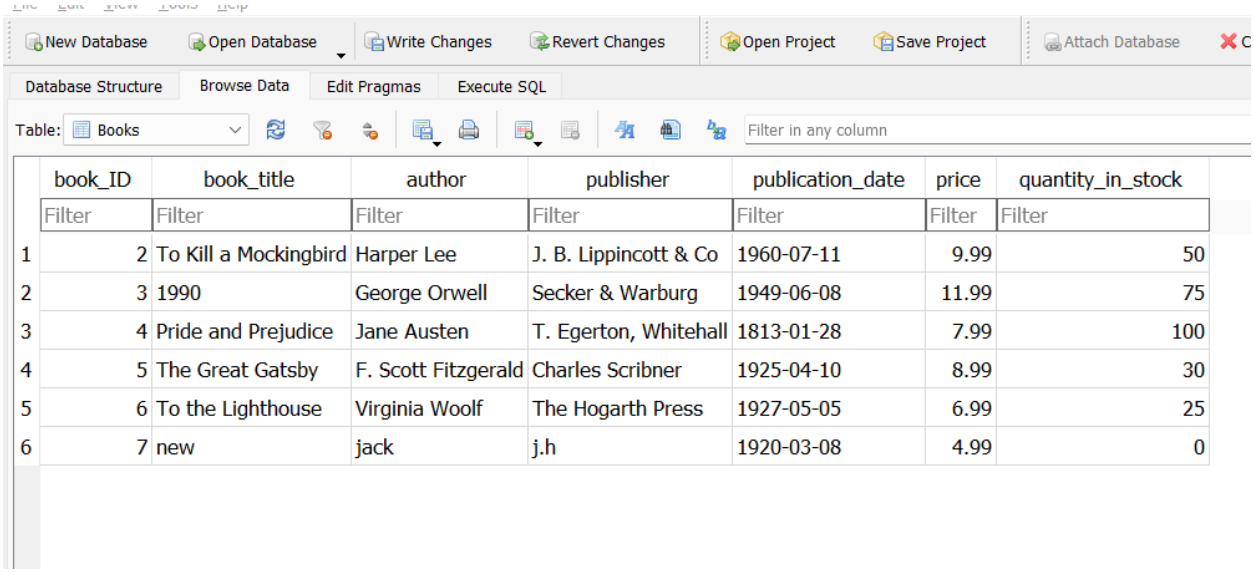

DELETE FROM "main"."Books " WHERE "book_ID " = 8;

Before:




| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 7 | 8 | | | | | | 0 |

After:

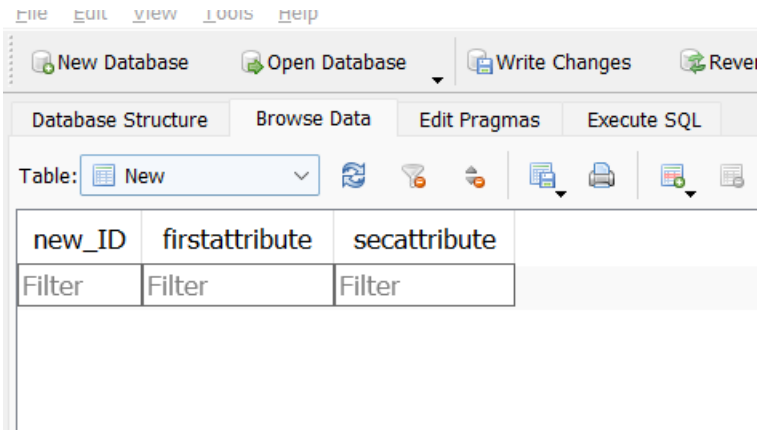



| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |

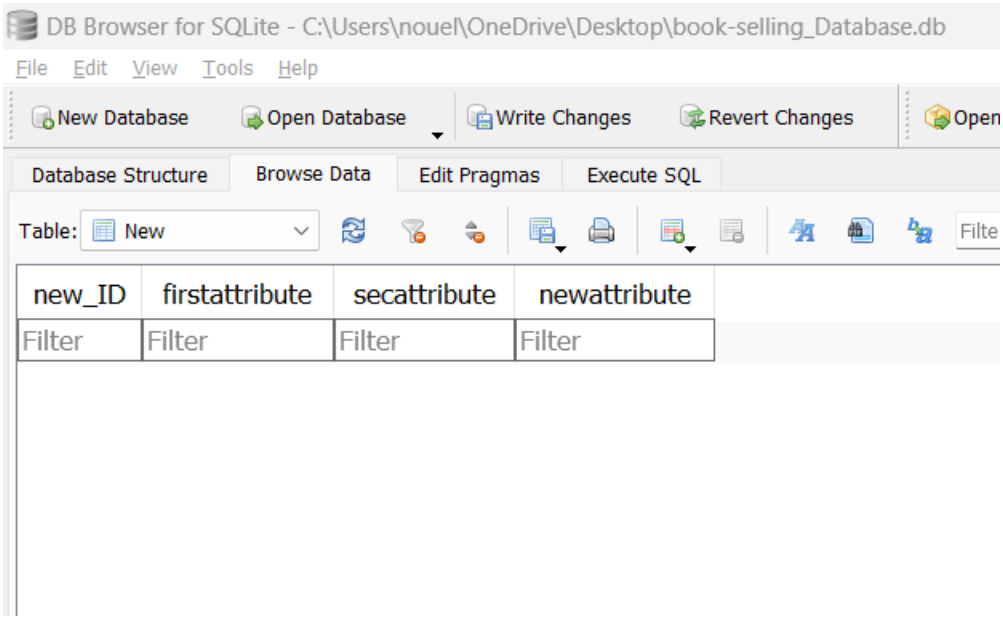
e-Alter add column:

ALTER TABLE "main"."New" ADD COLUMN "newattribute" Text;

Before:

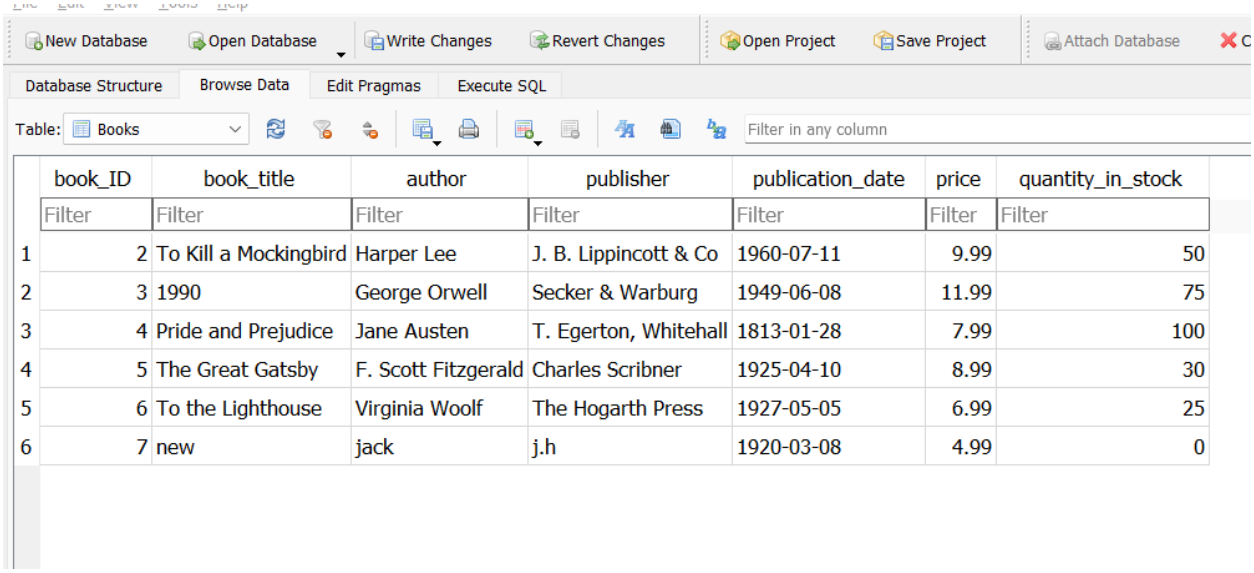


After:



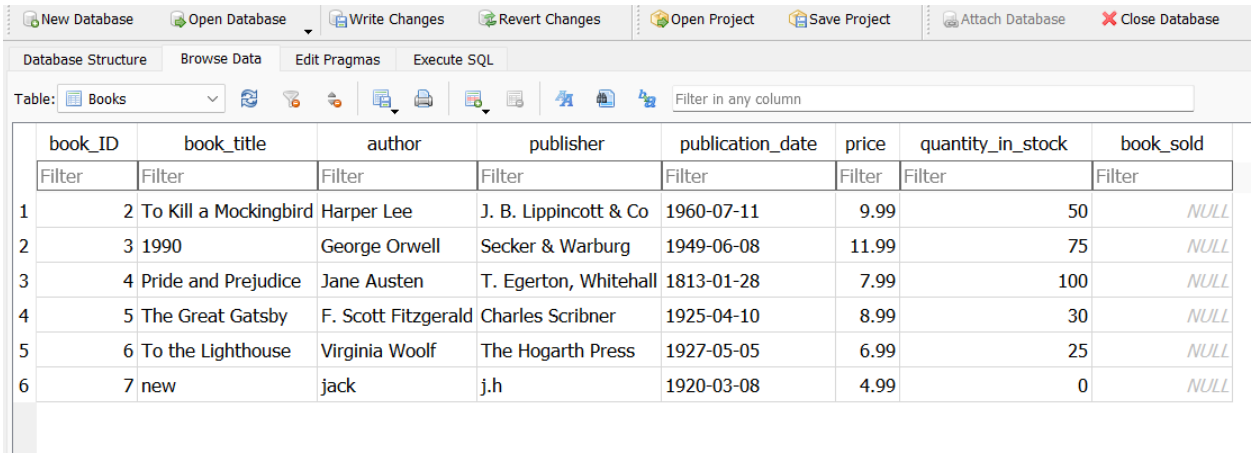

ALTER TABLE "main"."Books " ADD COLUMN "book_sold" INTEGER;

Before:




| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 2 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |

After:

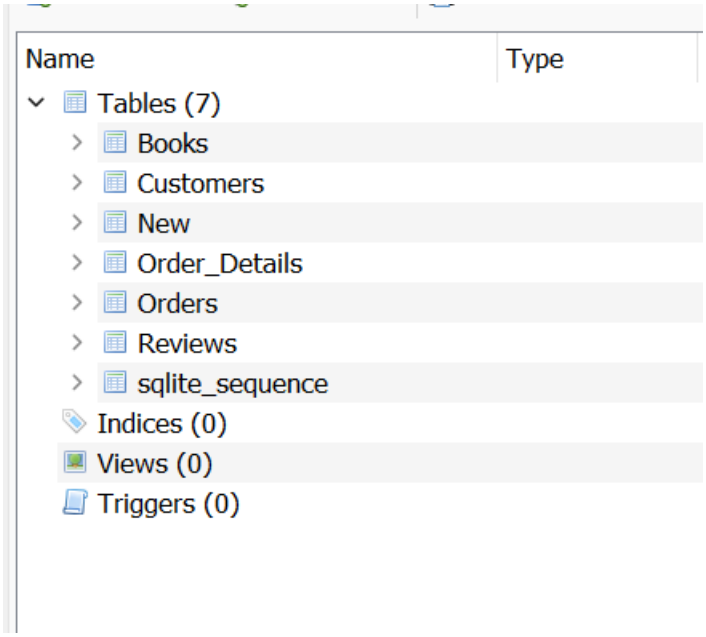



| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock | book_sold |
|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 | NULL |
| 2 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 | NULL |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 | NULL |
| 4 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 | NULL |
| 5 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 | NULL |
| 6 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 | NULL |

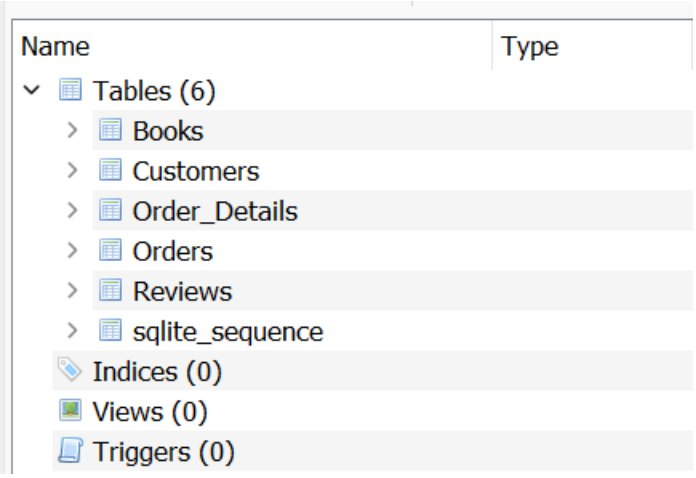
f-Drop:

DROP TABLE "main"."New";

Before:



After:



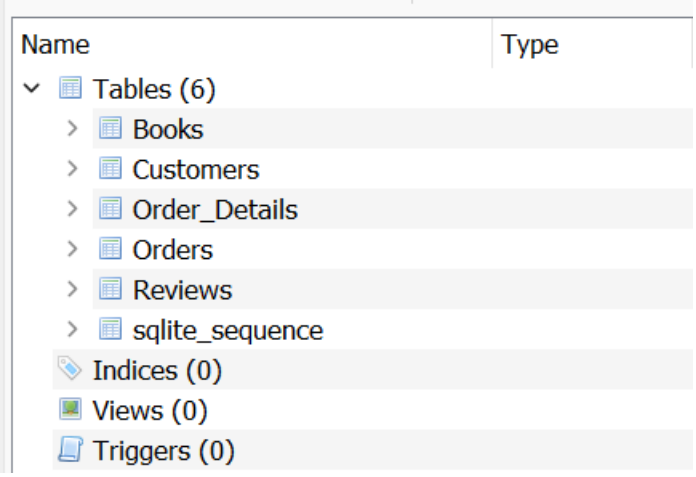

g-Create Index:

CREATE INDEX "author_index" ON "Books " ("author ");

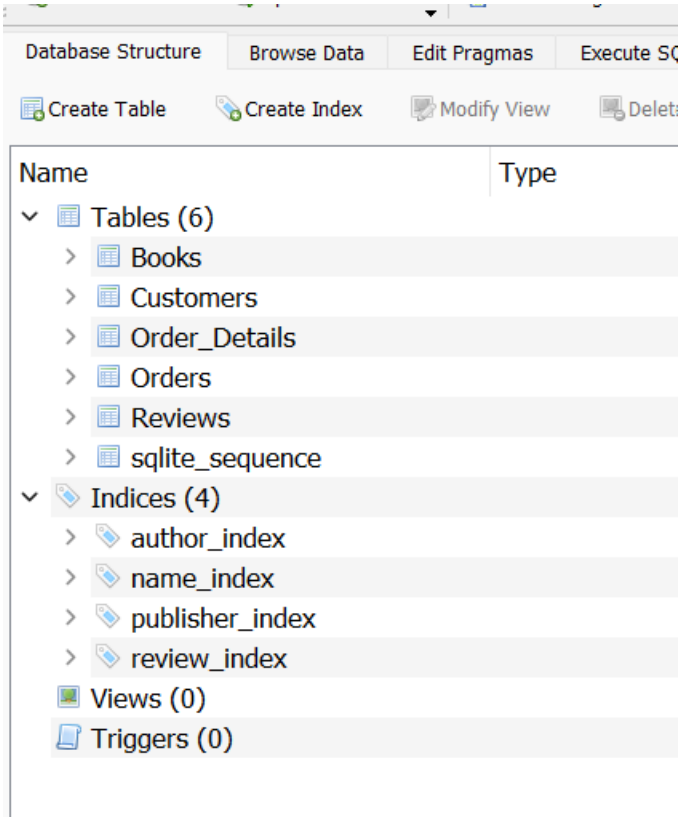
CREATE INDEX "name_index" ON "Customers" ("name");

CREATE INDEX "review_index" ON "Reviews " ("review_text");

Before:



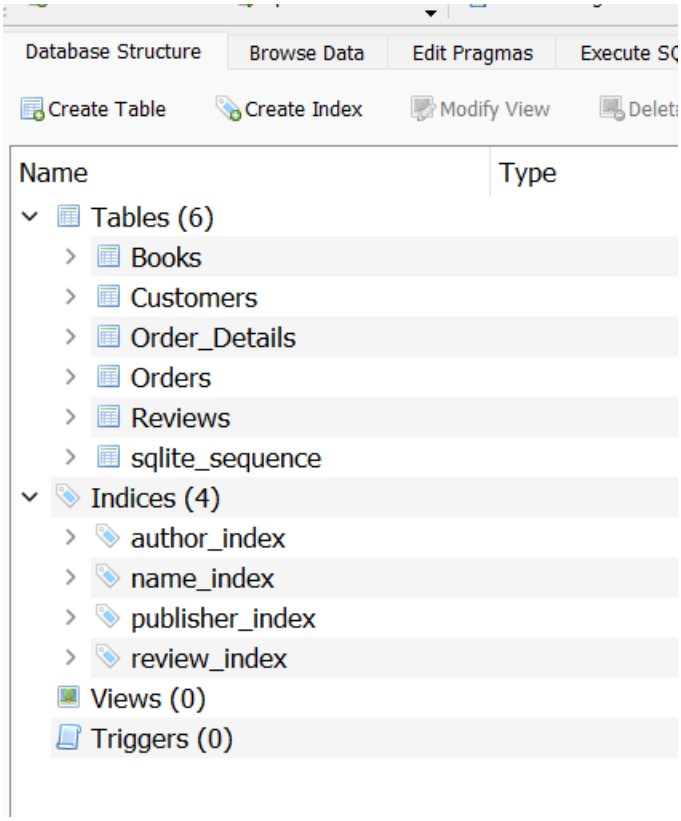

After:



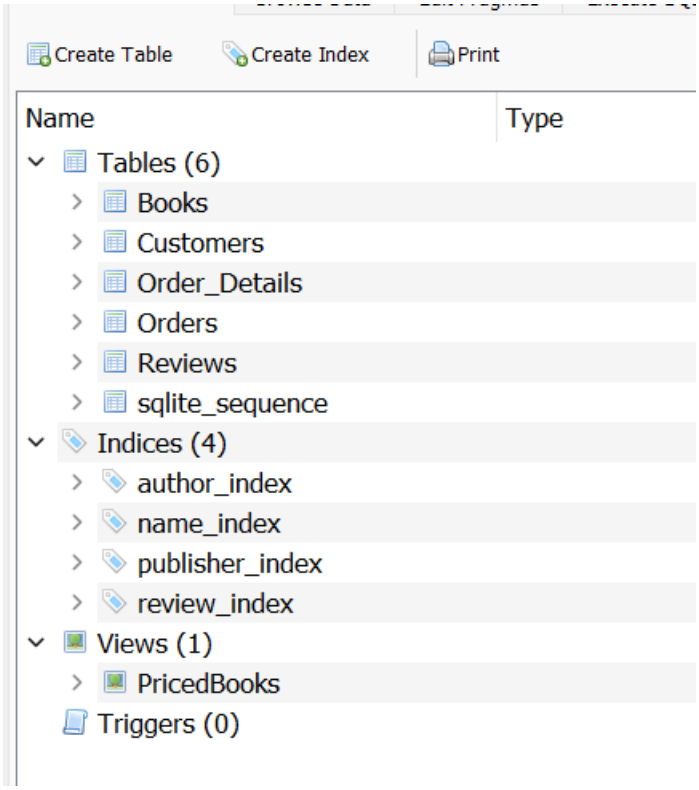

h-Create View:

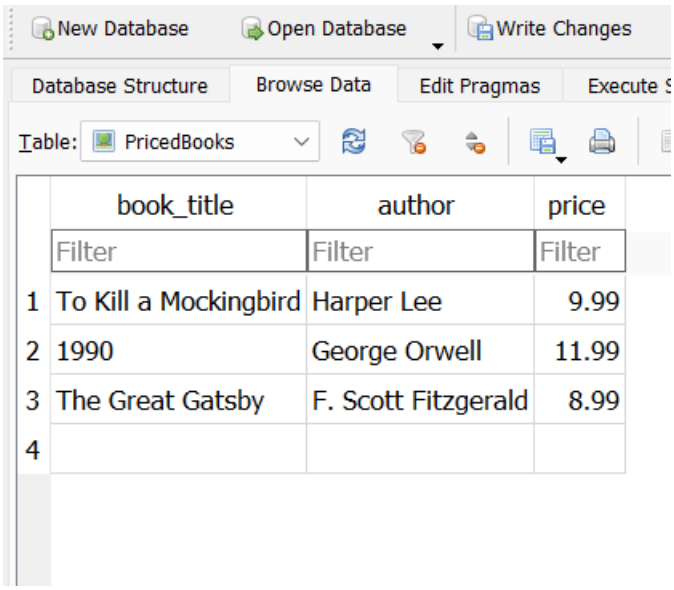
```
CREATE VIEW "PricedBooks" AS

SELECT "book_title ", "author ", "price "

FROM "Books "

WHERE "price " >= 8;
```

Before:


Database Structure
Browse Data
Edit Pragmas
Execute SQ
Create Table
Create Index
Modify View
Delet
Name
Type
Tables (6)
Books
Customers
Order_Details
Orders
Reviews
sqlite_sequence
Indices (4)
author_index
name_index
publisher_index
review_index
Views (0)
Triggers (0)

After:


Create Table
Create Index
Print
Name
Type
Tables (6)
Books
Customers
Order_Details
Orders
Reviews
sqlite_sequence
Indices (4)
author_index
name_index
publisher_index
review_index
Views (1)
PricedBooks
Triggers (0)

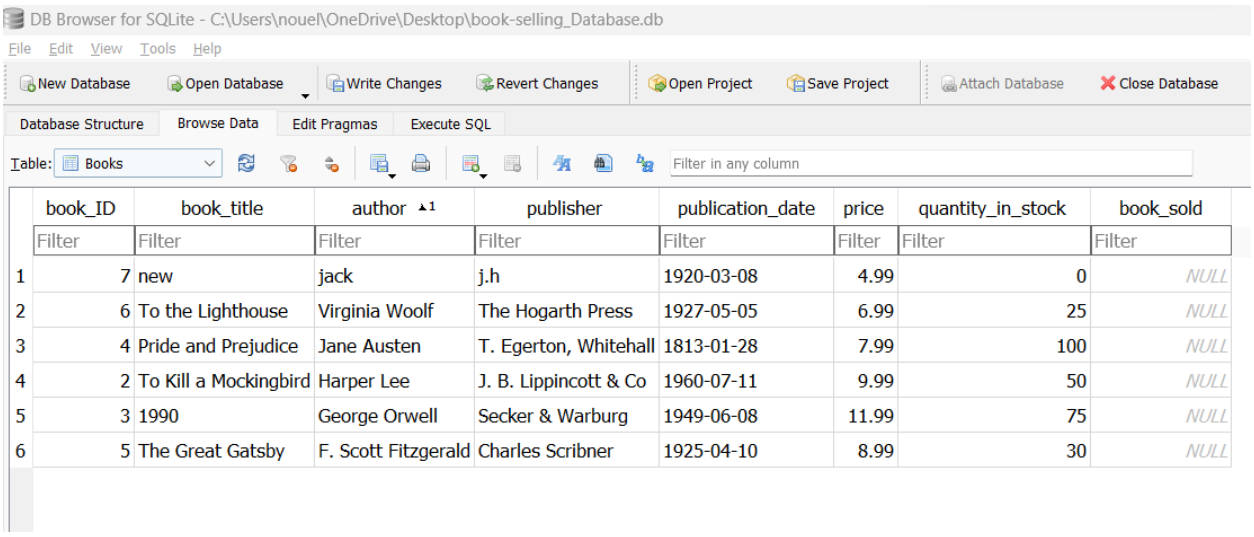



| | book_title | author | price |
|---|---|---|---|
| | Filter | Filter | Filter |
| 1 | To Kill a Mockingbird | Harper Lee | 9.99 |
| 2 | 1990 | George Orwell | 11.99 |
| 3 | The Great Gatsby | F. Scott Fitzgerald | 8.99 |
| 4 | | | |

i-Alter Delete a Column:

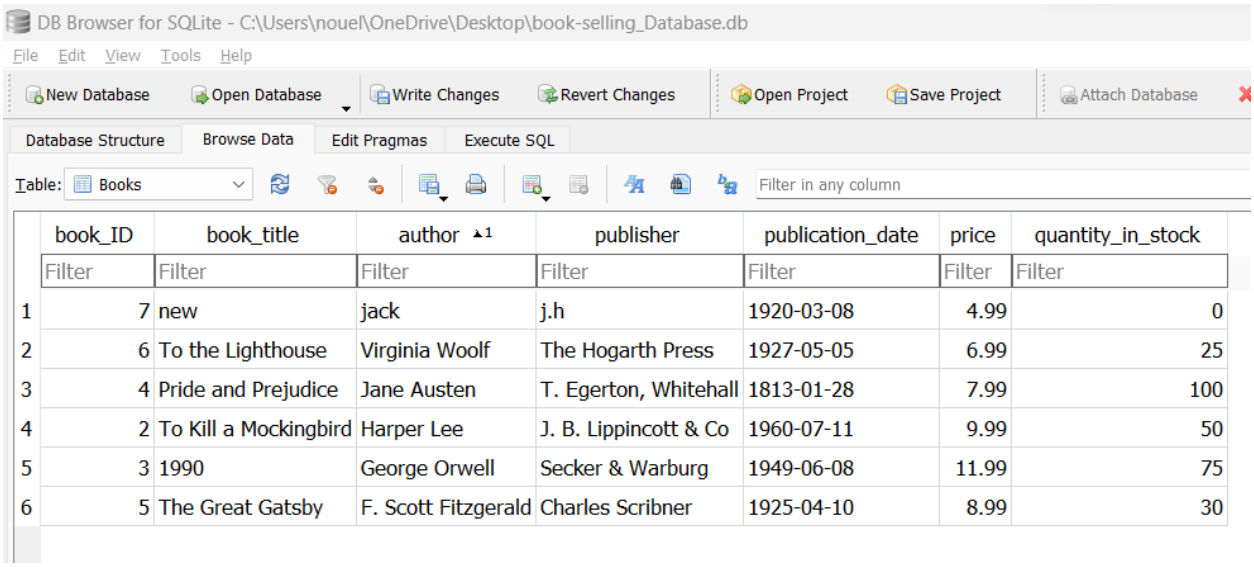
ALTER TABLE "Books " DROP COLUMN "book_sold";

Before:




| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock | book_sold |
|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 | NULL |
| 2 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 | NULL |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 | NULL |
| 4 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 | NULL |
| 5 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 | NULL |
| 6 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 | NULL |

After:

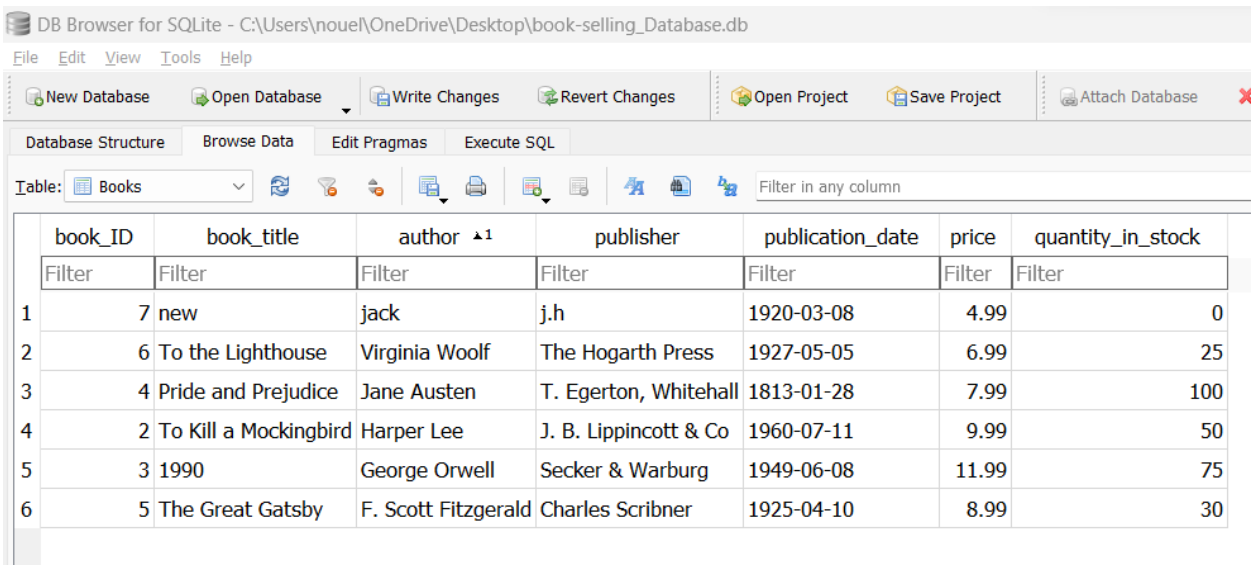



| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 2 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 5 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 6 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |

j-Alter Update Column name:

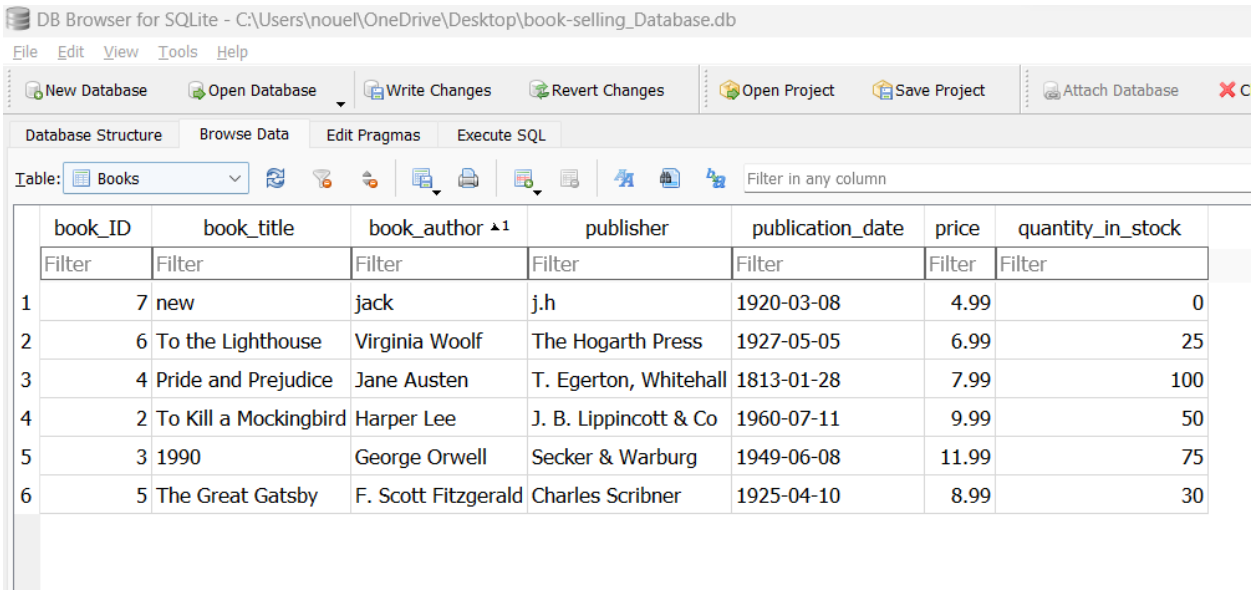
ALTER TABLE "Books " RENAME COLUMN "author " TO "book_author";

Before:




| | book_ID | book_title | author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 2 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 5 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 6 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |

After:

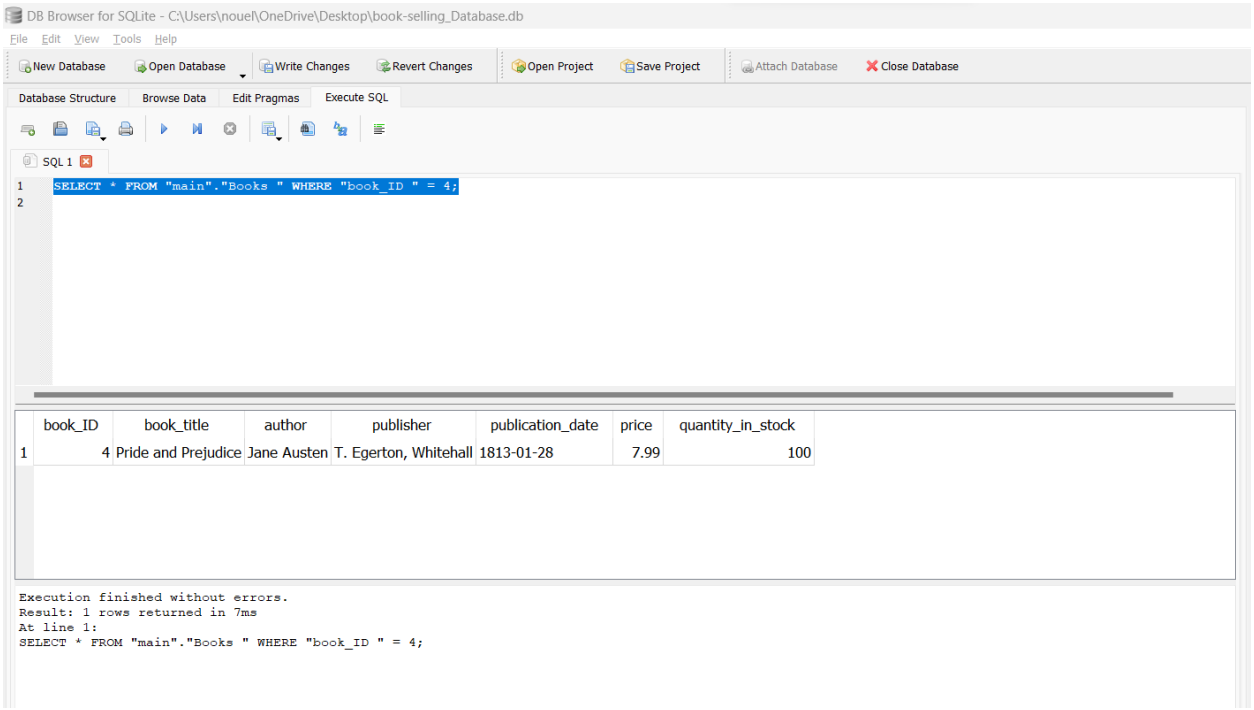



| | book_ID | book_title | book_author | publisher | publication_date | price | quantity_in_stock |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 7 | new | jack | j.h | 1920-03-08 | 4.99 | 0 |
| 2 | 6 | To the Lighthouse | Virginia Woolf | The Hogarth Press | 1927-05-05 | 6.99 | 25 |
| 3 | 4 | Pride and Prejudice | Jane Austen | T. Egerton, Whitehall | 1813-01-28 | 7.99 | 100 |
| 4 | 2 | To Kill a Mockingbird | Harper Lee | J. B. Lippincott & Co | 1960-07-11 | 9.99 | 50 |
| 5 | 3 | 1990 | George Orwell | Secker & Warburg | 1949-06-08 | 11.99 | 75 |
| 6 | 5 | The Great Gatsby | F. Scott Fitzgerald | Charles Scribner | 1925-04-10 | 8.99 | 30 |

## 8-The query:

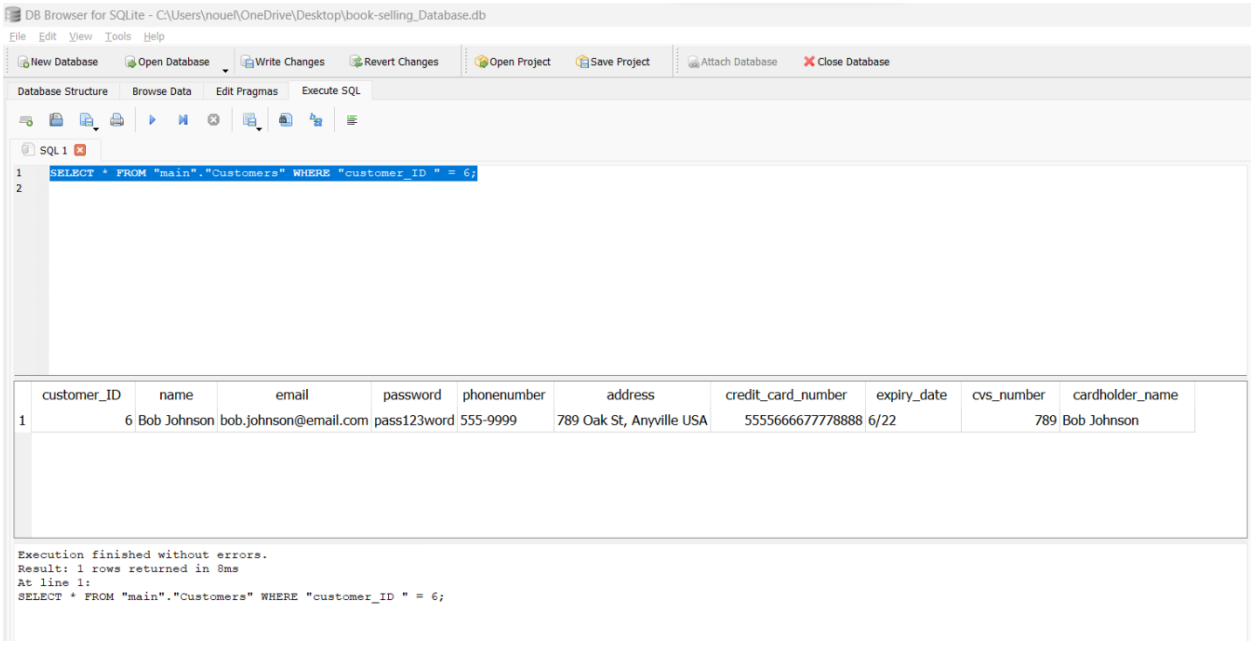
**1- Query to retrieve all information about a specific book, including its title, author, publisher, publication date, price, and quantity in stock:**

SELECT * FROM "main"."Books " WHERE "book_ID " = 4;

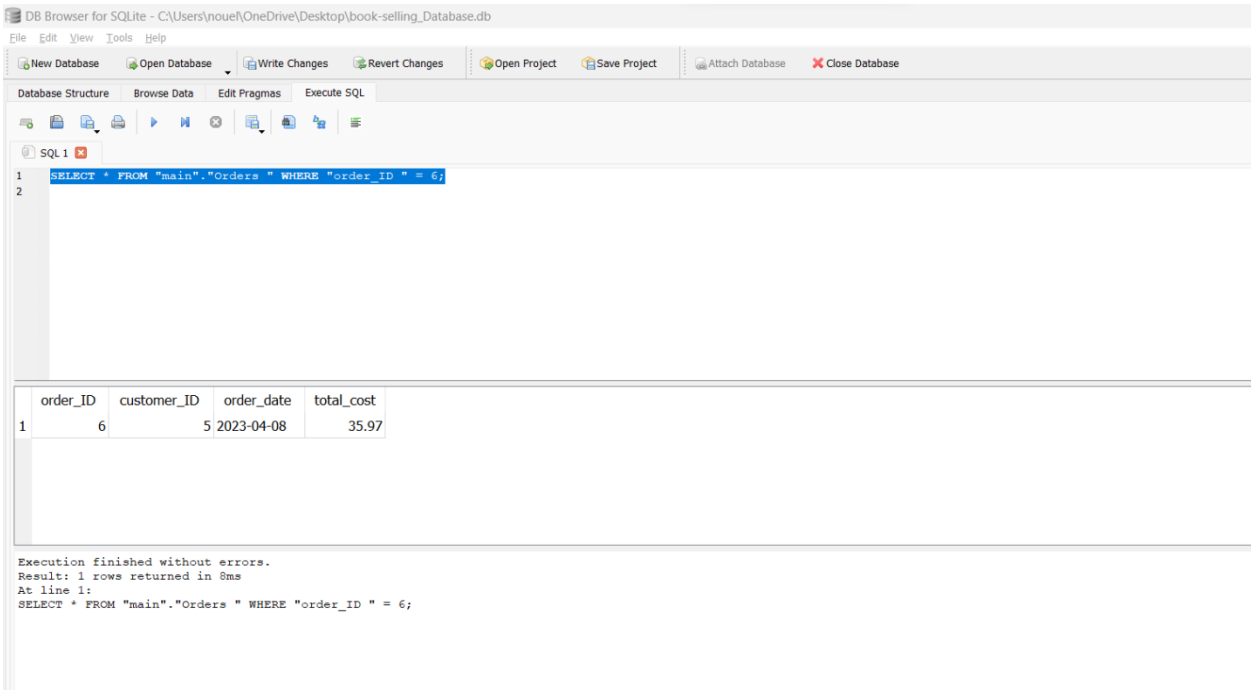



2- Query to retrieve all information about a specific customer, including their name, address, email, credit card number, and phone number:

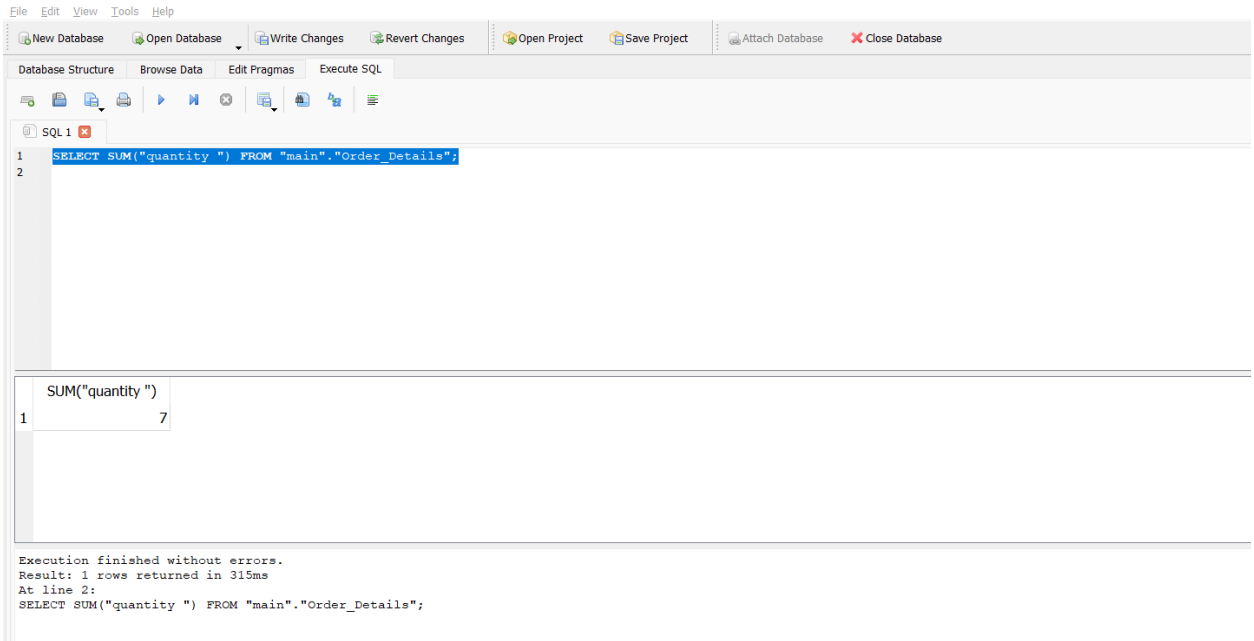
SELECT * FROM "main"."Customers" WHERE "customer_ID " = 6;



3- Query to retrieve all information about a specific order, including the order date, total cost, and the customer who placed the order:
SELECT * FROM "main"."Orders " WHERE "order_ID " = 6;

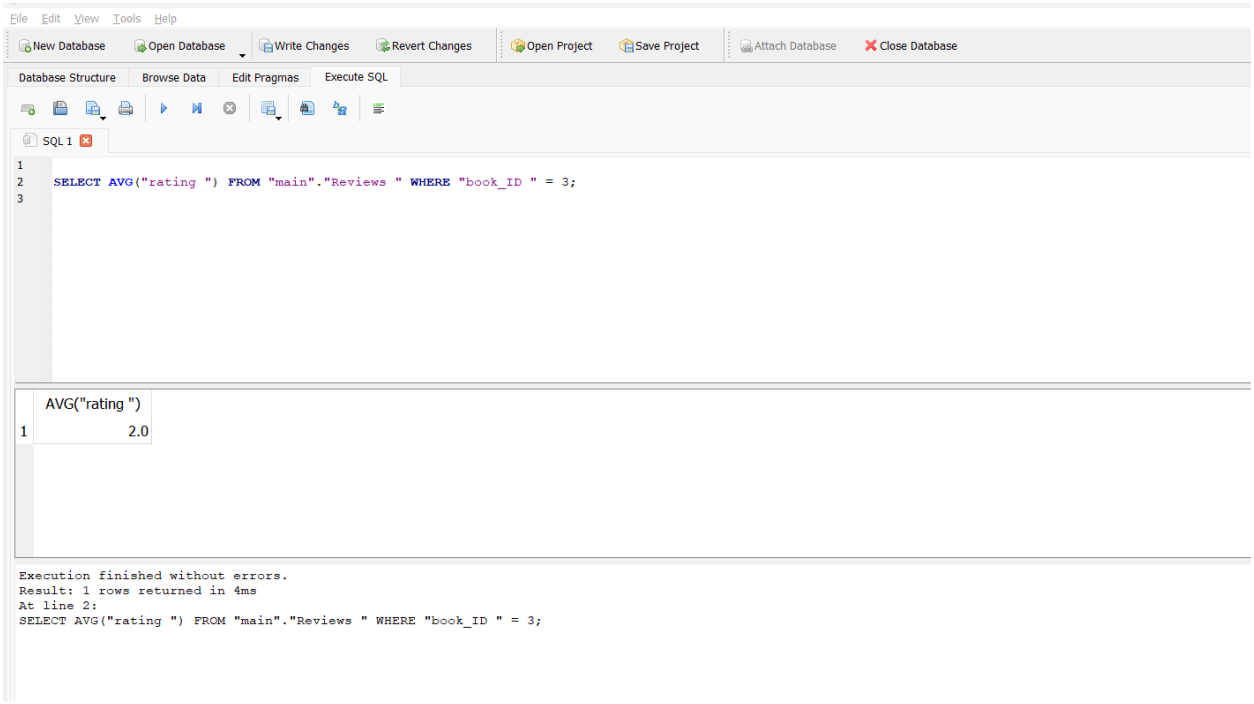



4- Query to retrieve the total number of books in the inventory:
SELECT SUM("quantity ") FROM "main"."Order_Details";



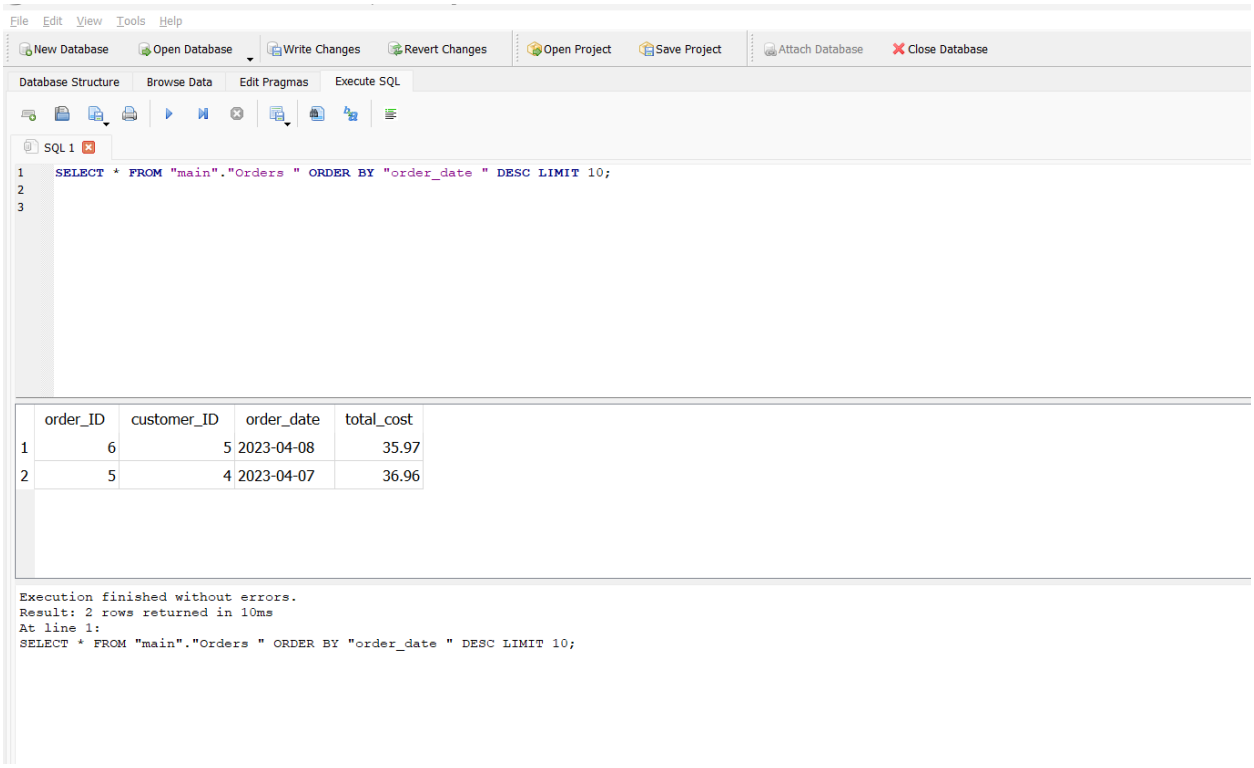

5- Query to retrieve the average rating for a specific book:

SELECT AVG("rating ") FROM "main"."Reviews " WHERE "book_ID " = 3;



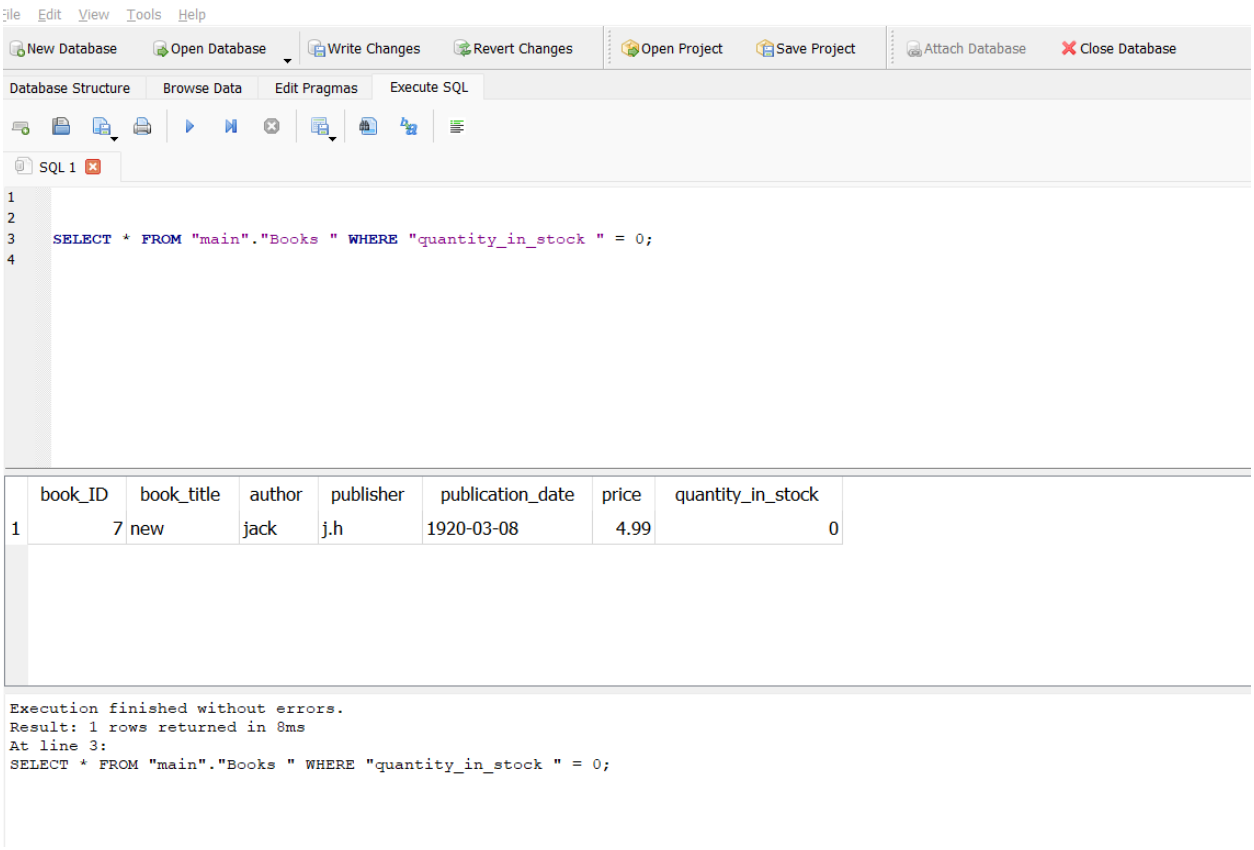

6 - Query to retrieve the most recent orders, sorted by order date in descending order:

SELECT * FROM "main"."Orders " ORDER BY "order_date " DESC LIMIT 10;



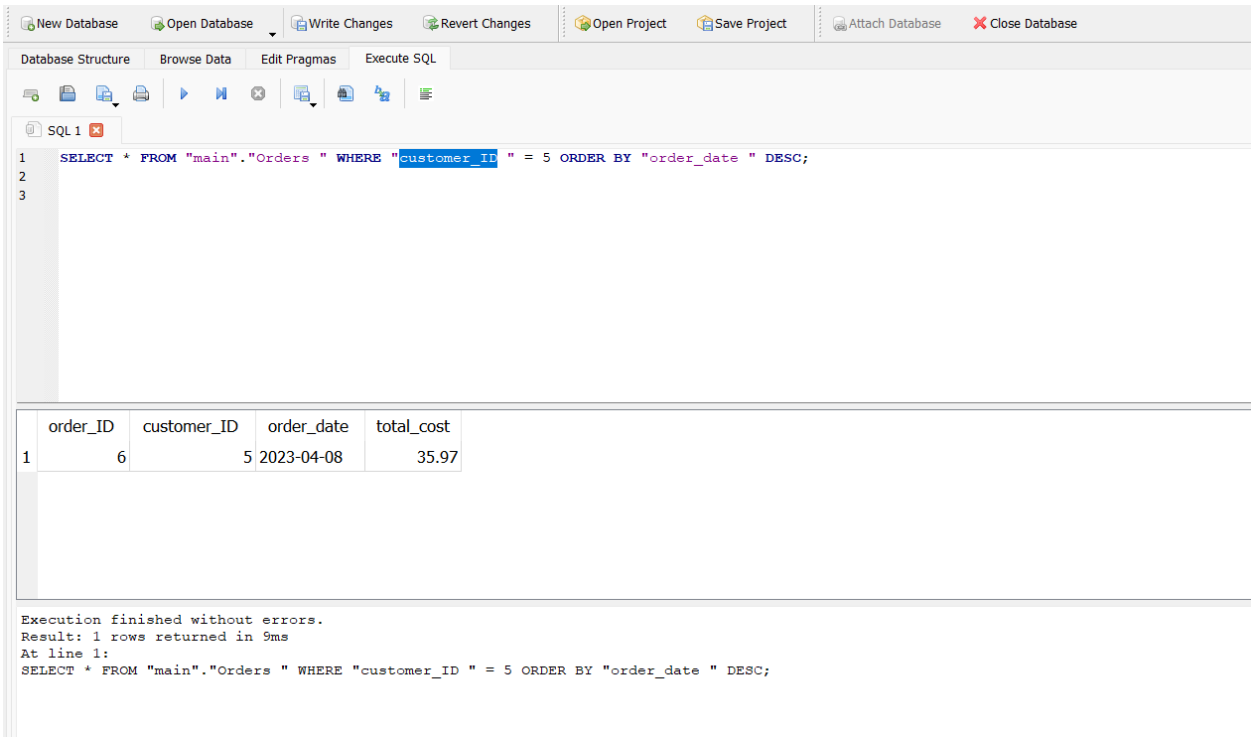

7- Query to retrieve all books that are out of stock:

SELECT * FROM "main"."Books " WHERE "quantity_in_stock " = 0;




8- Retrieve all orders made by a specific customer, sorted by order date in descending order:

SELECT * FROM "main"."Orders " WHERE "customer_ID " = 5 ORDER BY "order_date " DESC;

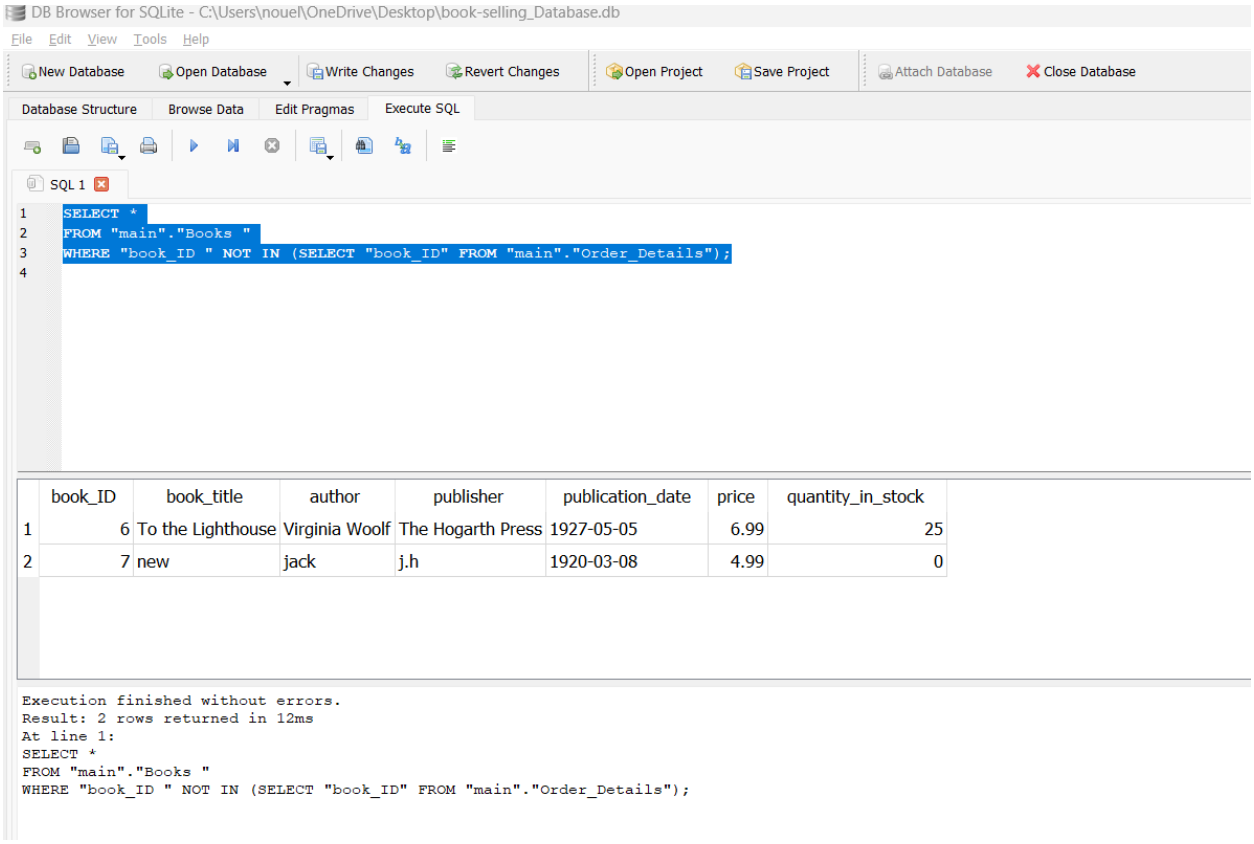



9- Show the books that have not been ordered yet:

SELECT *

FROM "main"."Books "

WHERE "book_ID " NOT IN (SELECT "book_ID" FROM "main"."Order_Details");

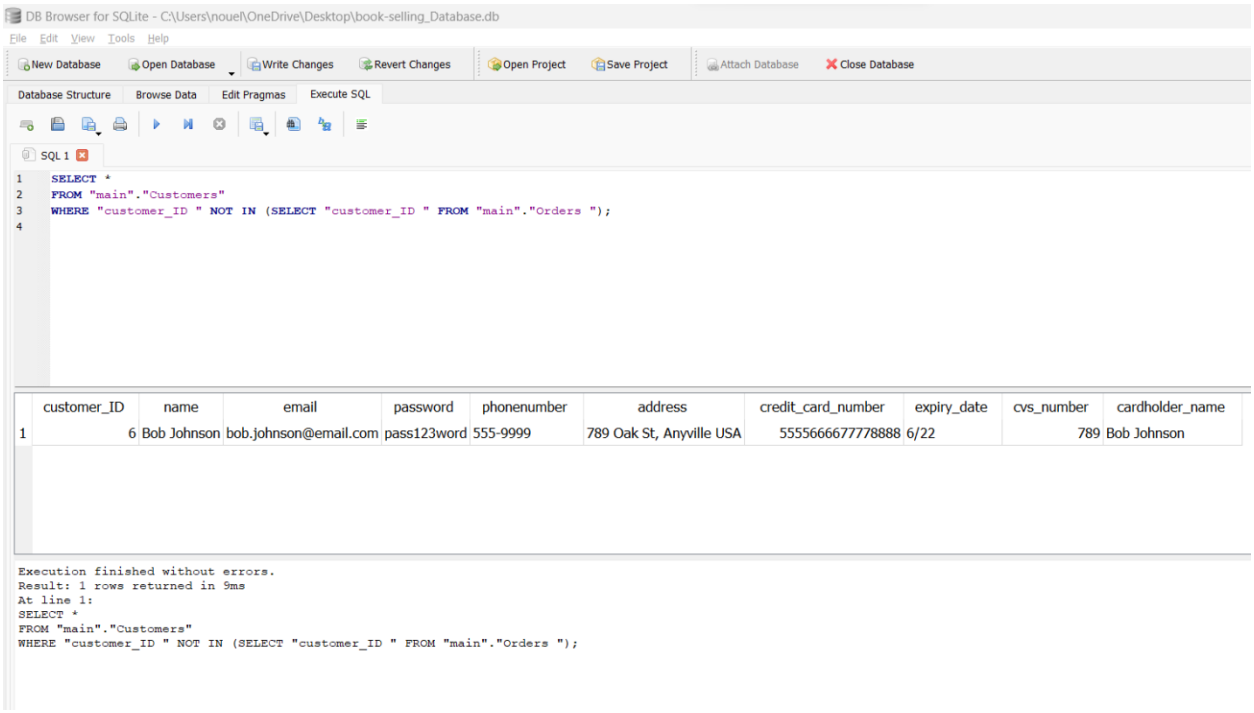



10- Show the customers who have not made any orders:

SELECT *

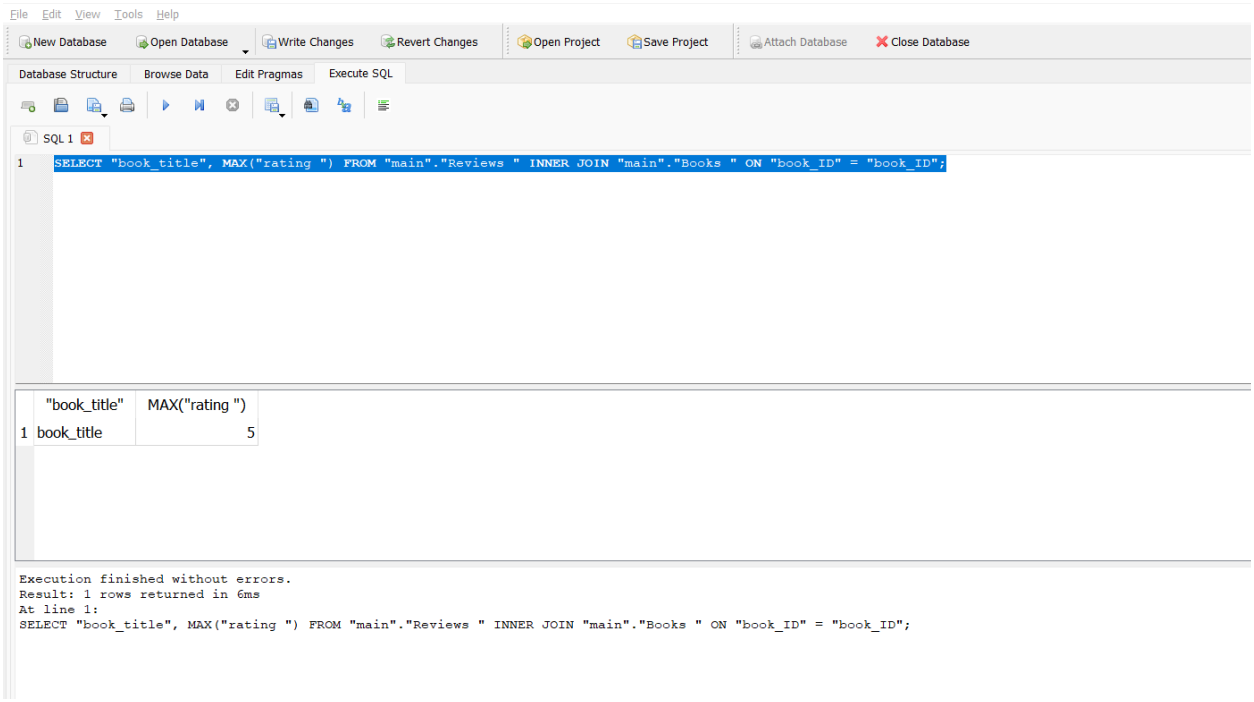
FROM "main"."Customers"

WHERE "customer_ID " NOT IN (SELECT "customer_ID " FROM "main"."Orders ");




11- Select the book with the highest rating:

SELECT "book_title", MAX("rating ") FROM "main"."Reviews " INNER JOIN "main"."Books " ON "book_ID" = "book_ID";

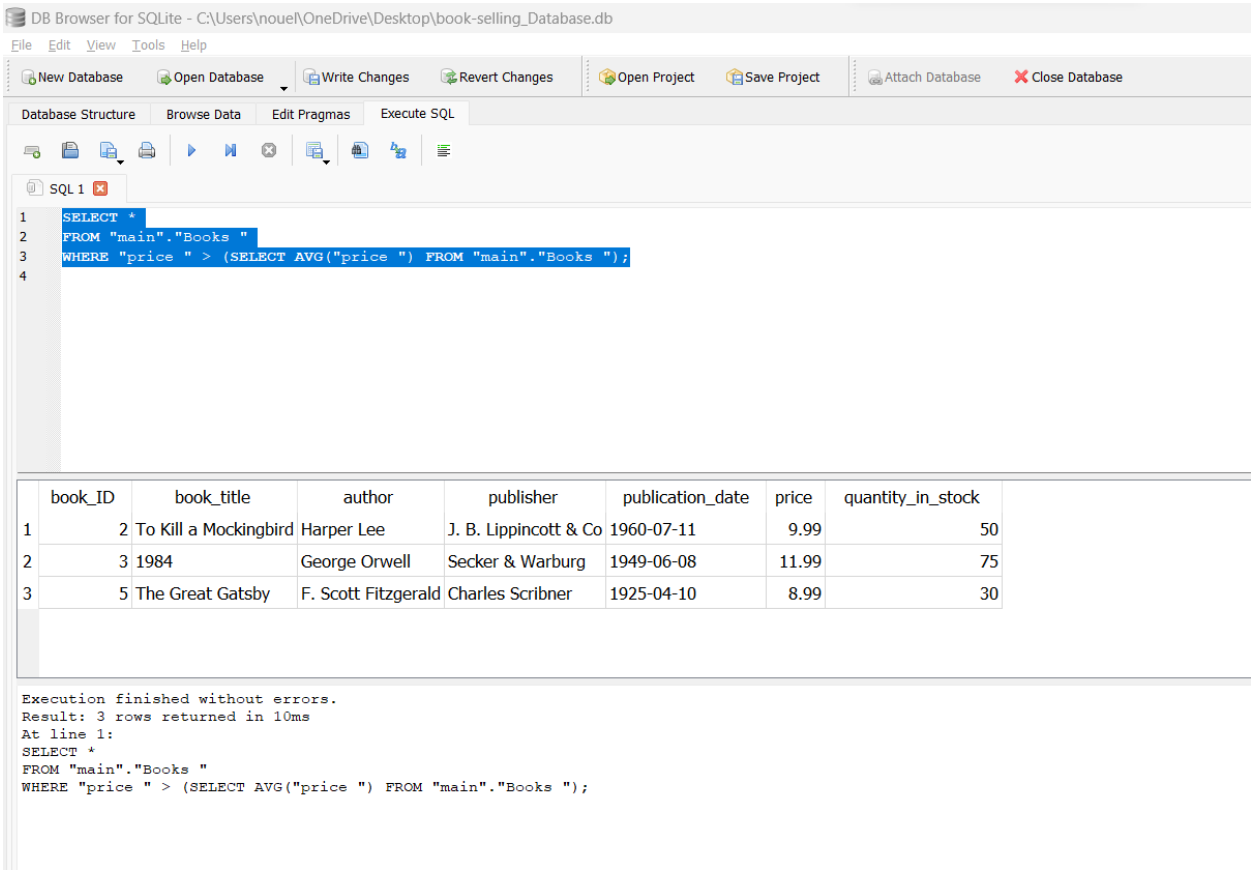



12- This query selects all books from the Books table where the price is greater than the average price of all books in the table. The subquery (SELECT AVG(price)

FROM Books) calculates the average price of all books in the table:
SELECT *

FROM "main"."Books "

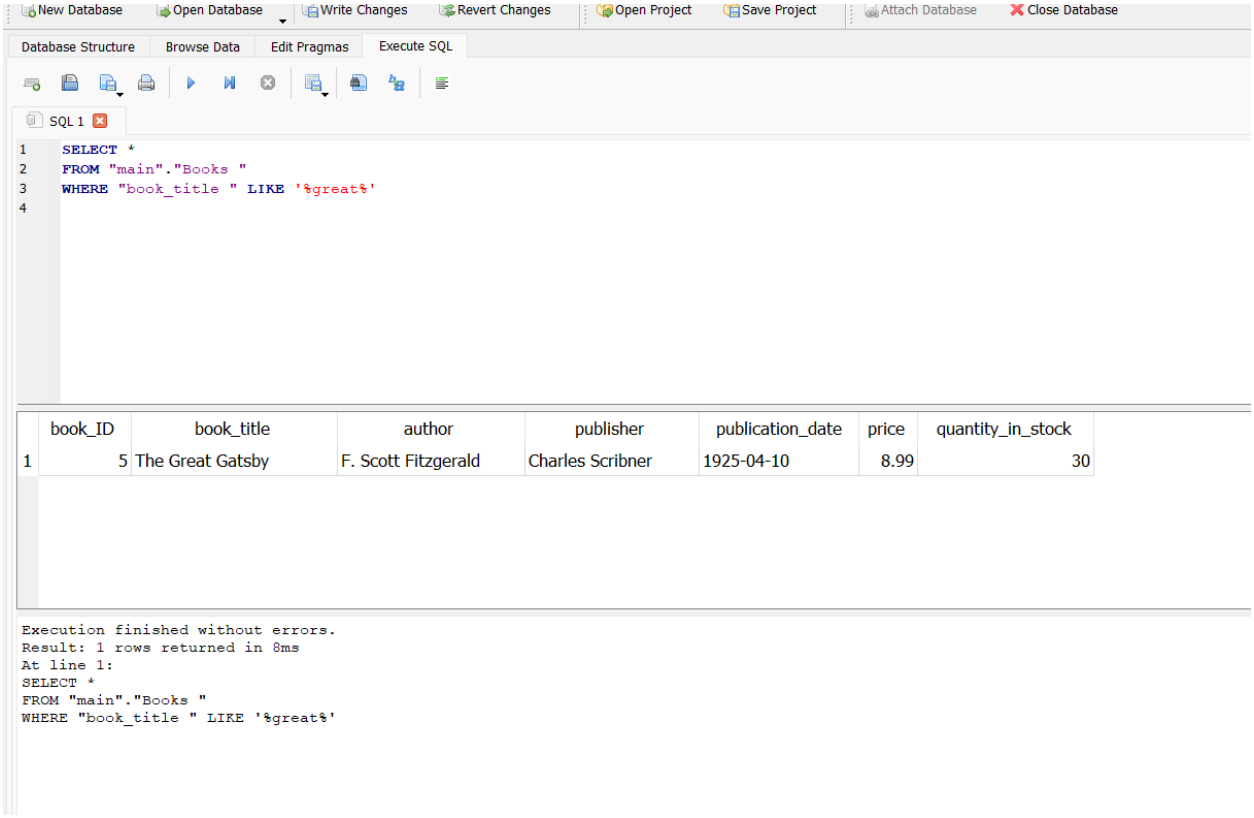
WHERE "price " > (SELECT AVG("price ") FROM "main"."Books ");




13- This query selects all books from the Books table where the book title contains the word 'Great':
SELECT *

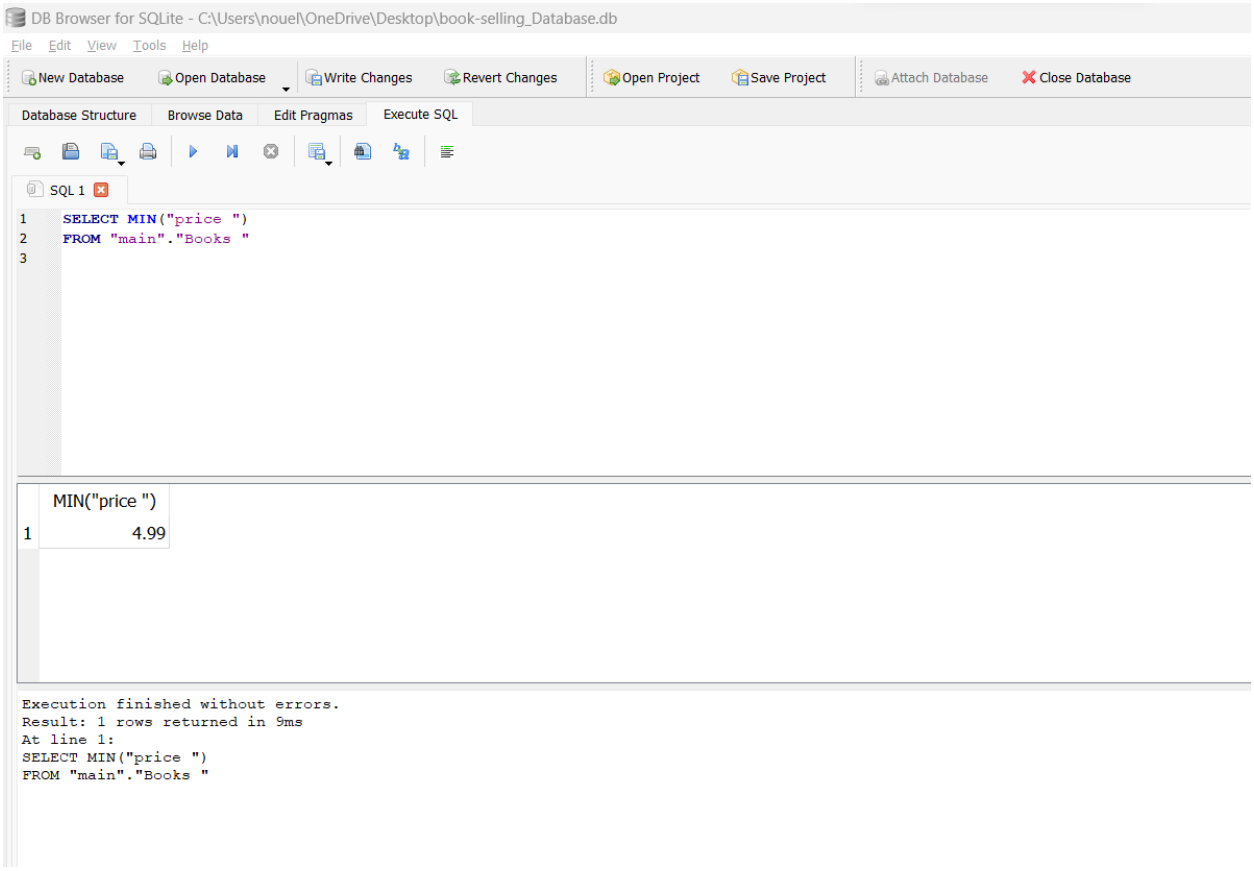
FROM "main"."Books "

WHERE "book_title " LIKE '%great%'



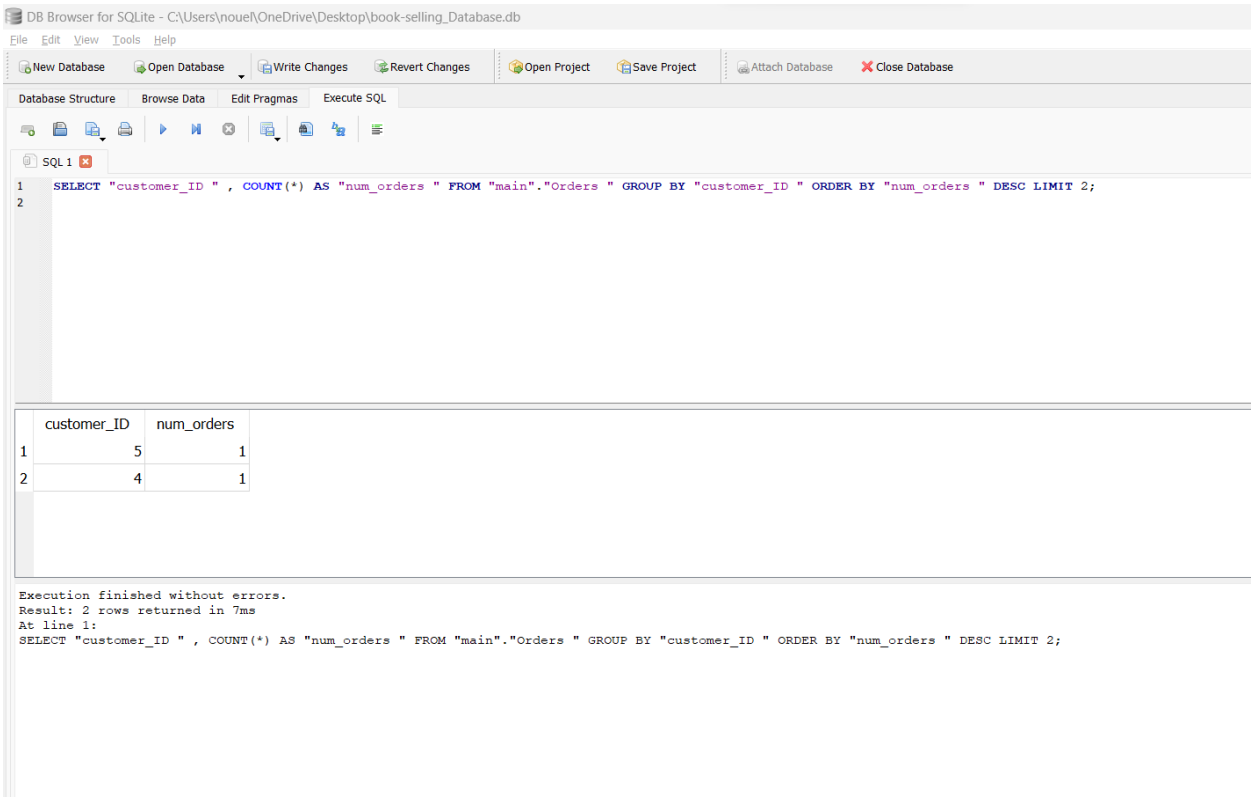

14- This query selects the lowest price from the Books table:
SELECT MIN("price ")

FROM "main"."Books "




15-Find the customer who has placed the most orders:

SELECT "customer_ID " , COUNT(*) AS "num_orders " FROM "main"."Orders "
GROUP BY "customer_ID " ORDER BY "num_orders " DESC LIMIT 2;




**9-Team Contribution:**

Every team member collaborated seamlessly on all aspects of the project, utilizing a single shared computer to collectively generate valuable contributions to every stage of the process. This included in-depth analysis of the problem, determining database requirements, creating an ER diagram, designing a relational database schema, generating a sample database, constructing SQL statements, and developing complex queries. Every team member was fully engaged and offered unique perspectives, resulting in a highly effective and comprehensive final product.

# Note:

In SQLite, the "Date" data type is not available. After conducting some research, it was found that an alternative can be achieved by using the "TEXT-INTEGER-REAL" data type, as the input format can be controlled through the application used by the users. As an alternative, we utilized the "Text" data type in our implementation.