Justin Hurcombe

MSIS-2503: Fundamentals of SQL

Assignment #2

2/7/24

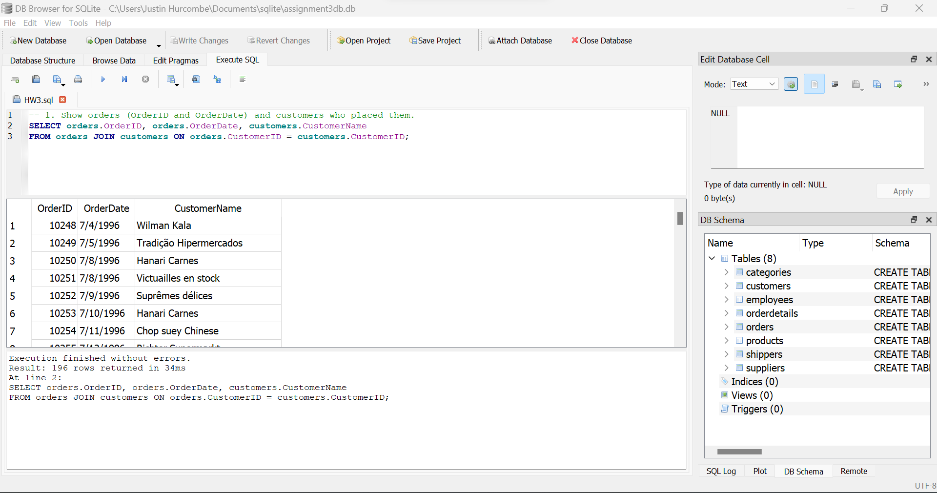
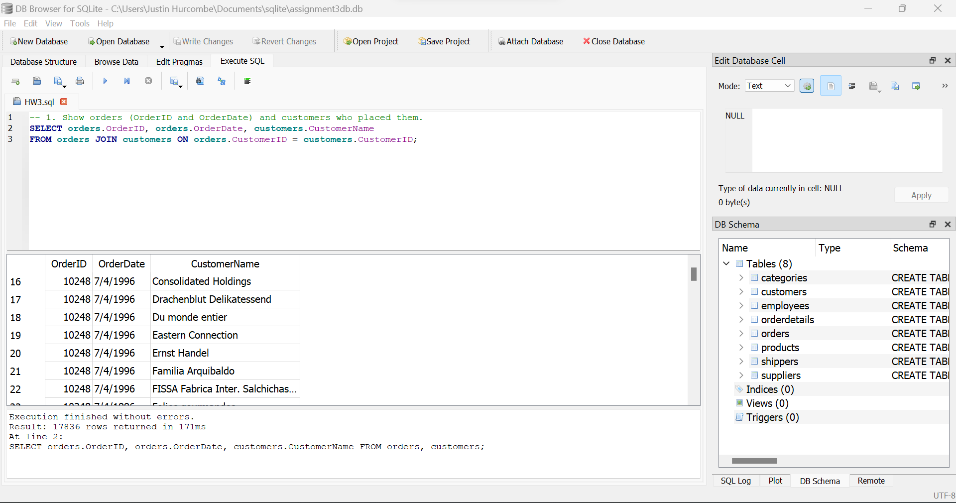
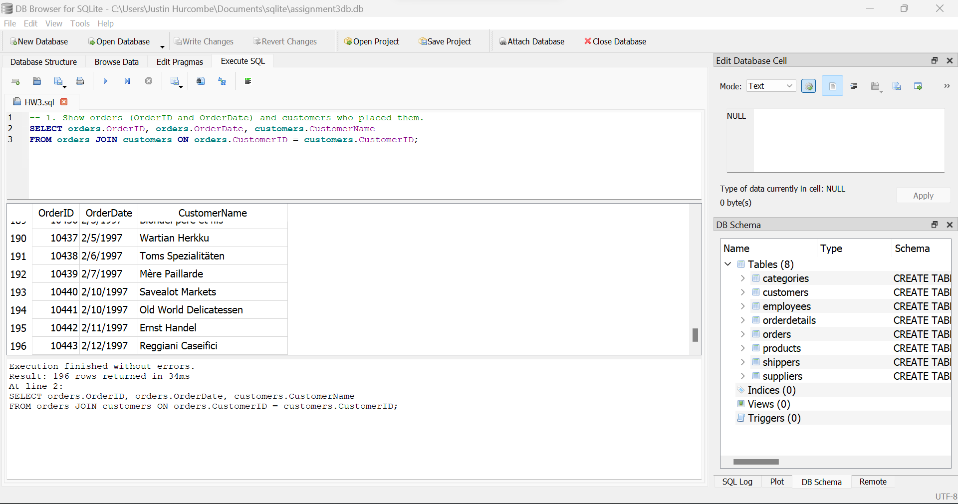
For given tables and query task, give the query statement, execute the query and show the result.

(All questions have to actually execute query and include the execution results or screen shot.)

# Question 1: Show orders (orderID and orderDate) and customers who placed them.

# SELECT orders.ORDERID, orders.OrderDate, customers.CustomerName

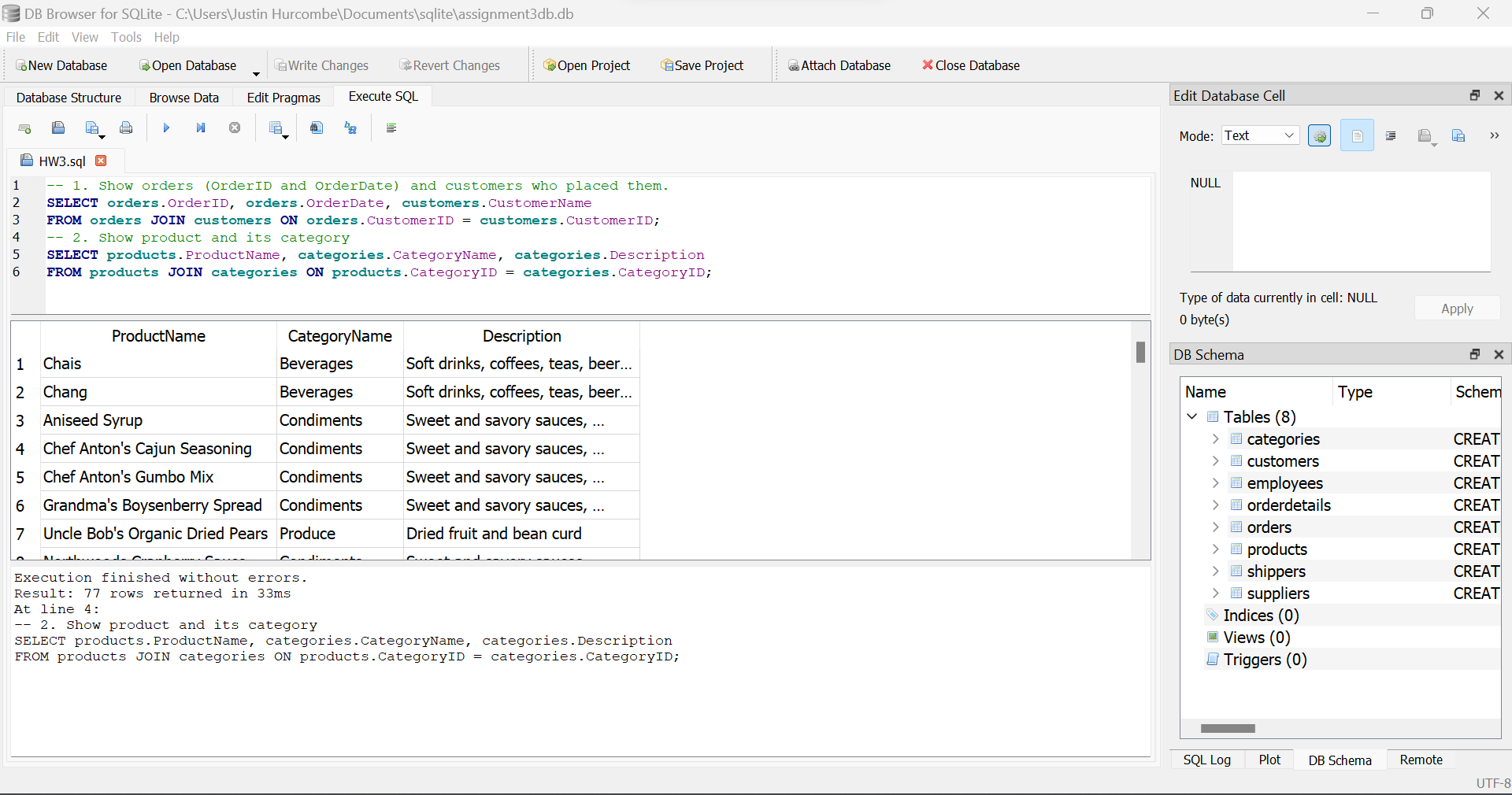
# FROM orders JOIN customers ON orders.CustomerID = customers.CustomerID;

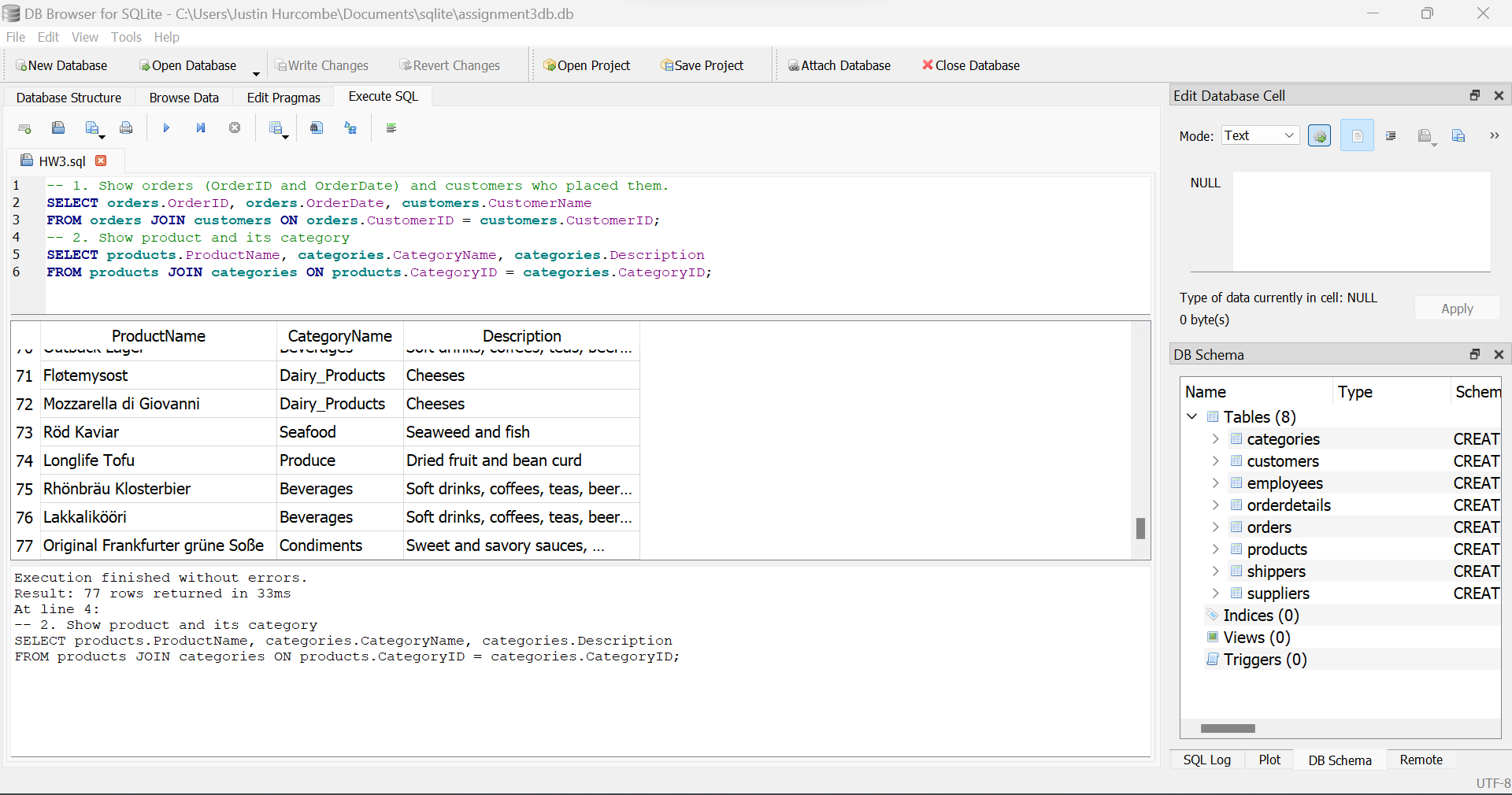
  

**Show product and its category.**

SELECT products.ProductName, categories.CategoryName, categories.Description

FROM products JOIN categories ON products.CategoryID = categories.CategoryID;

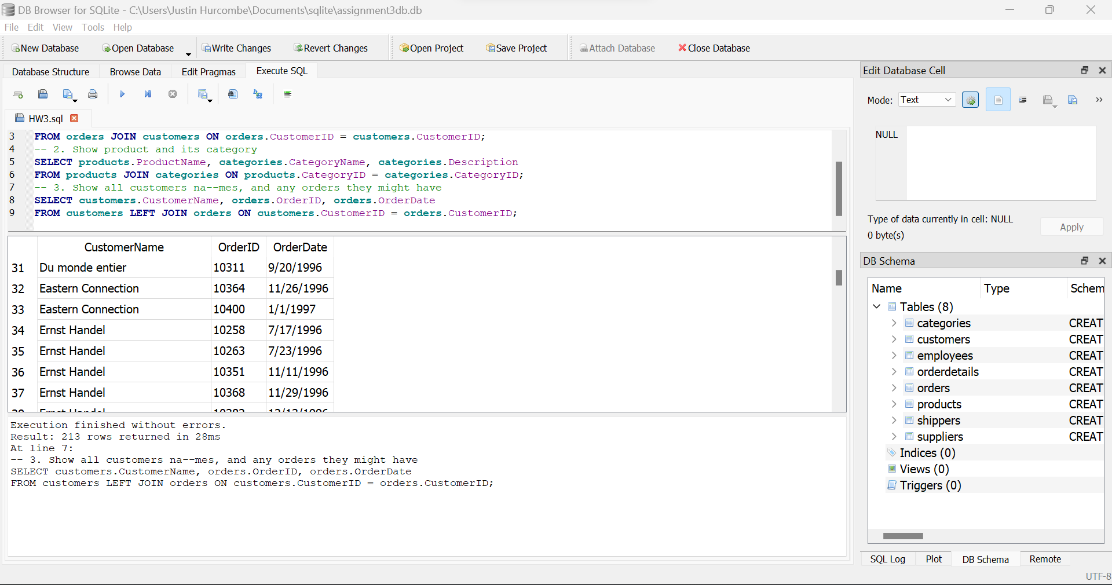
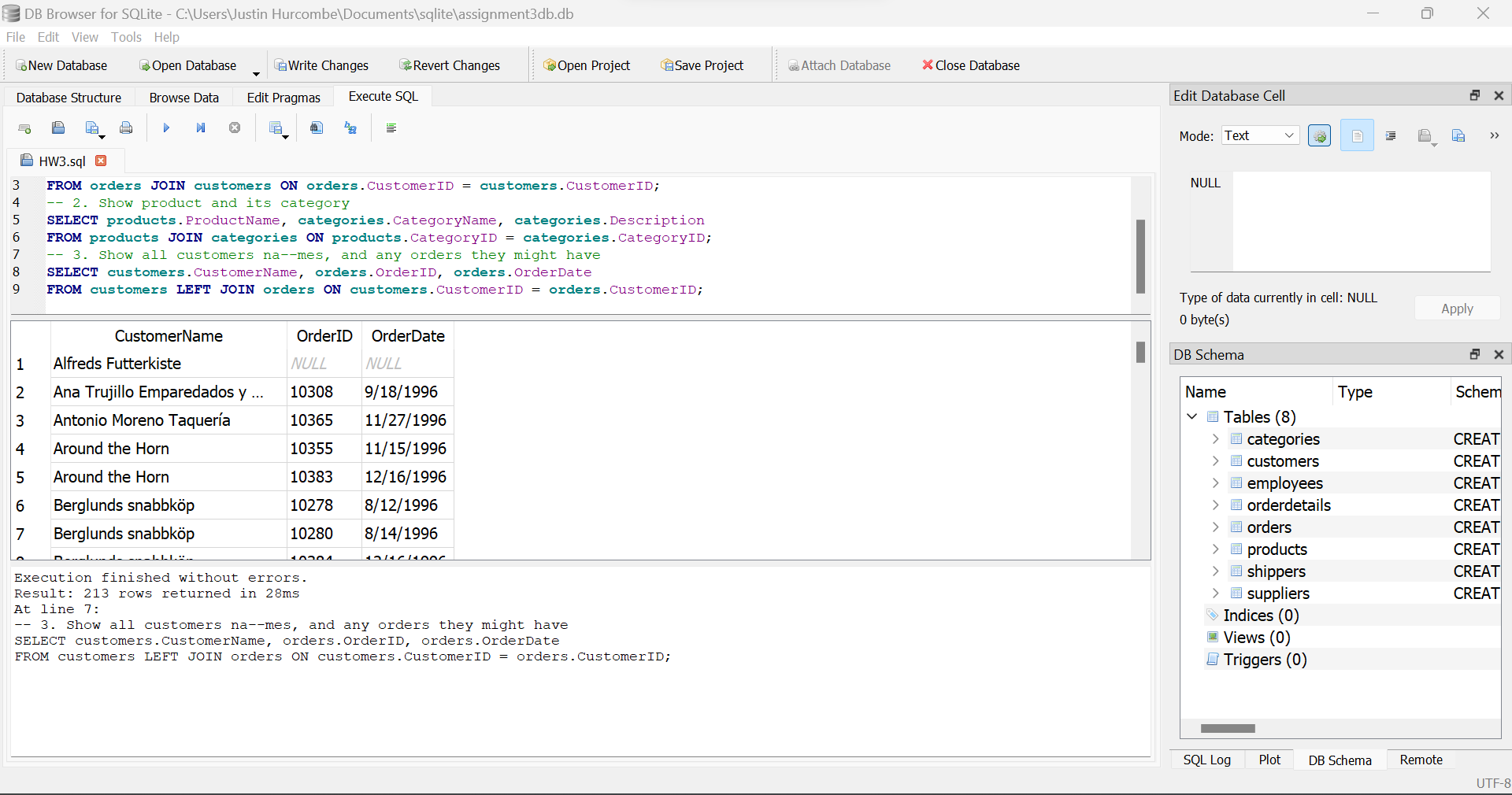


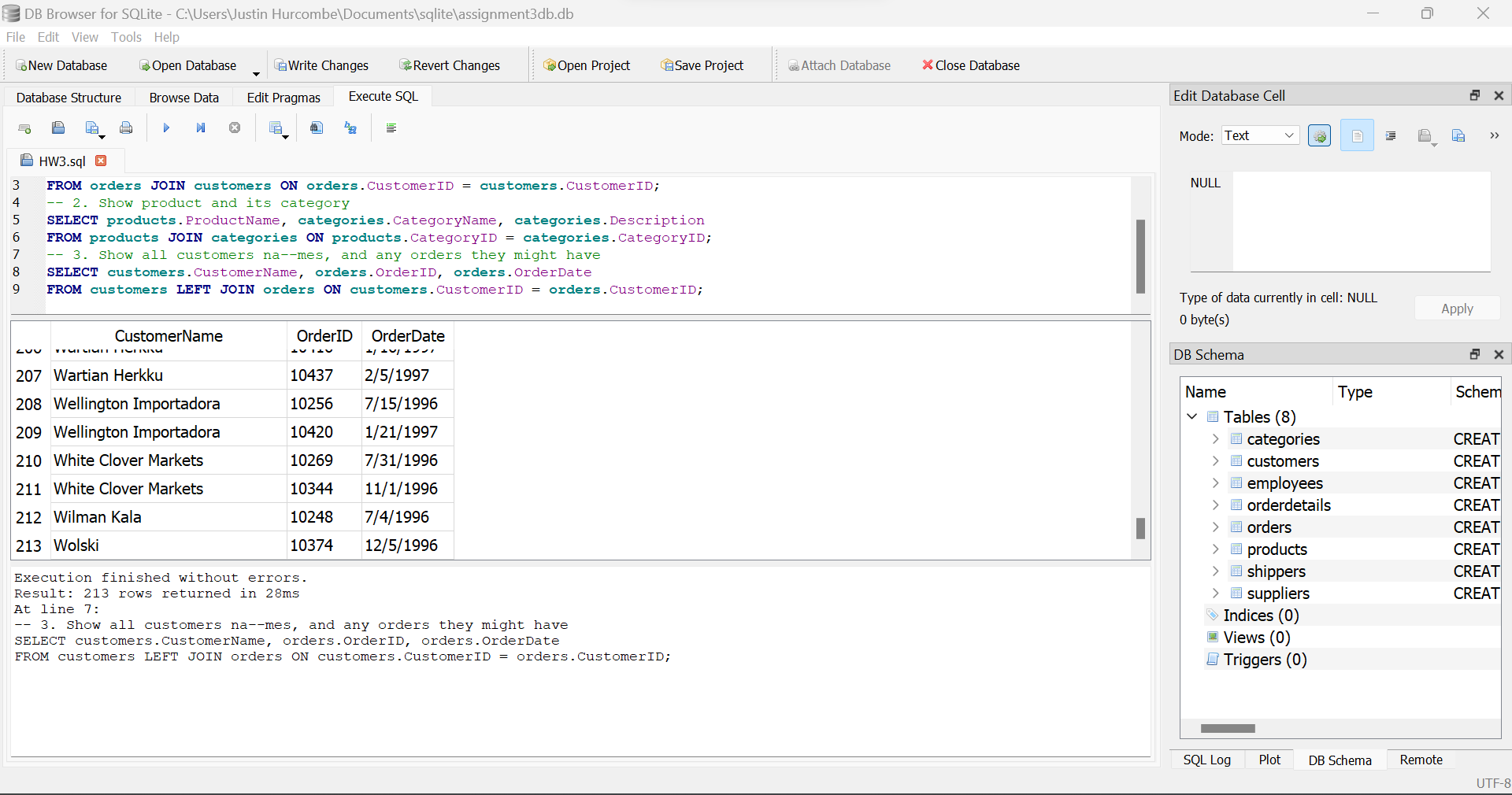


**Show all customers names , and any orders they might have.**

SELECT customers.CustomerName, orders.OrderID, orders.OrderDate

FROM customers LEFT JOIN orders ON customers.CustomerID = orders.CustomerID;

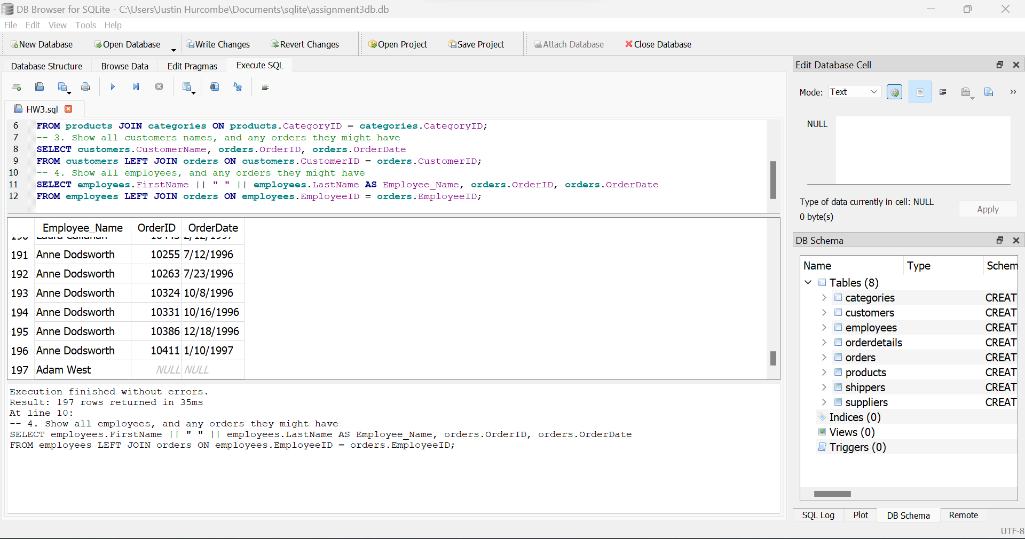
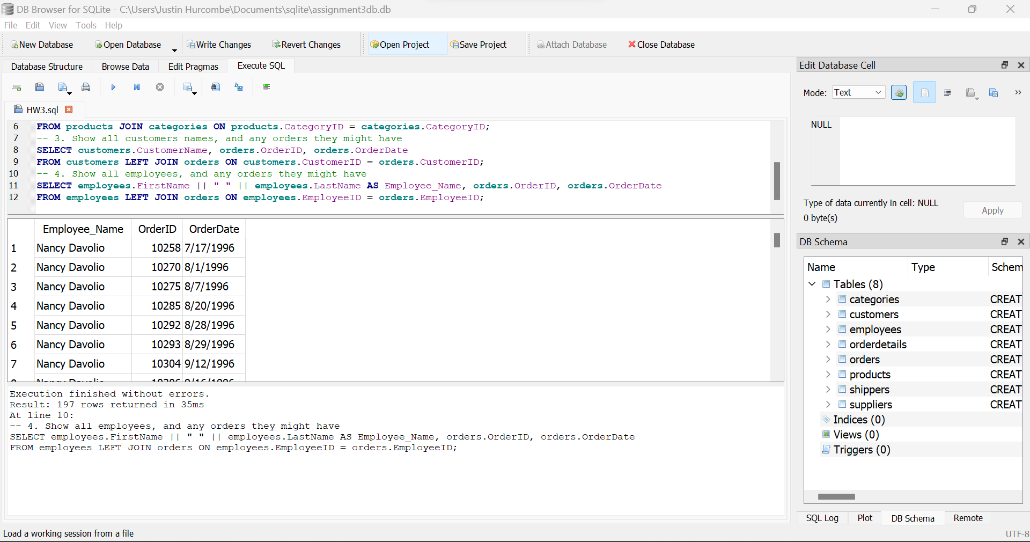


****

**Show all employees, and any orders they might have.**

SELECT employees.FirstName || “ “ || employees.LastName AS Employee\_Name, orders.OrderID, orders.OrderDate

FROM employees LEFT JOIN orders ON employees.EmployeeID = orders.EmployeeID;

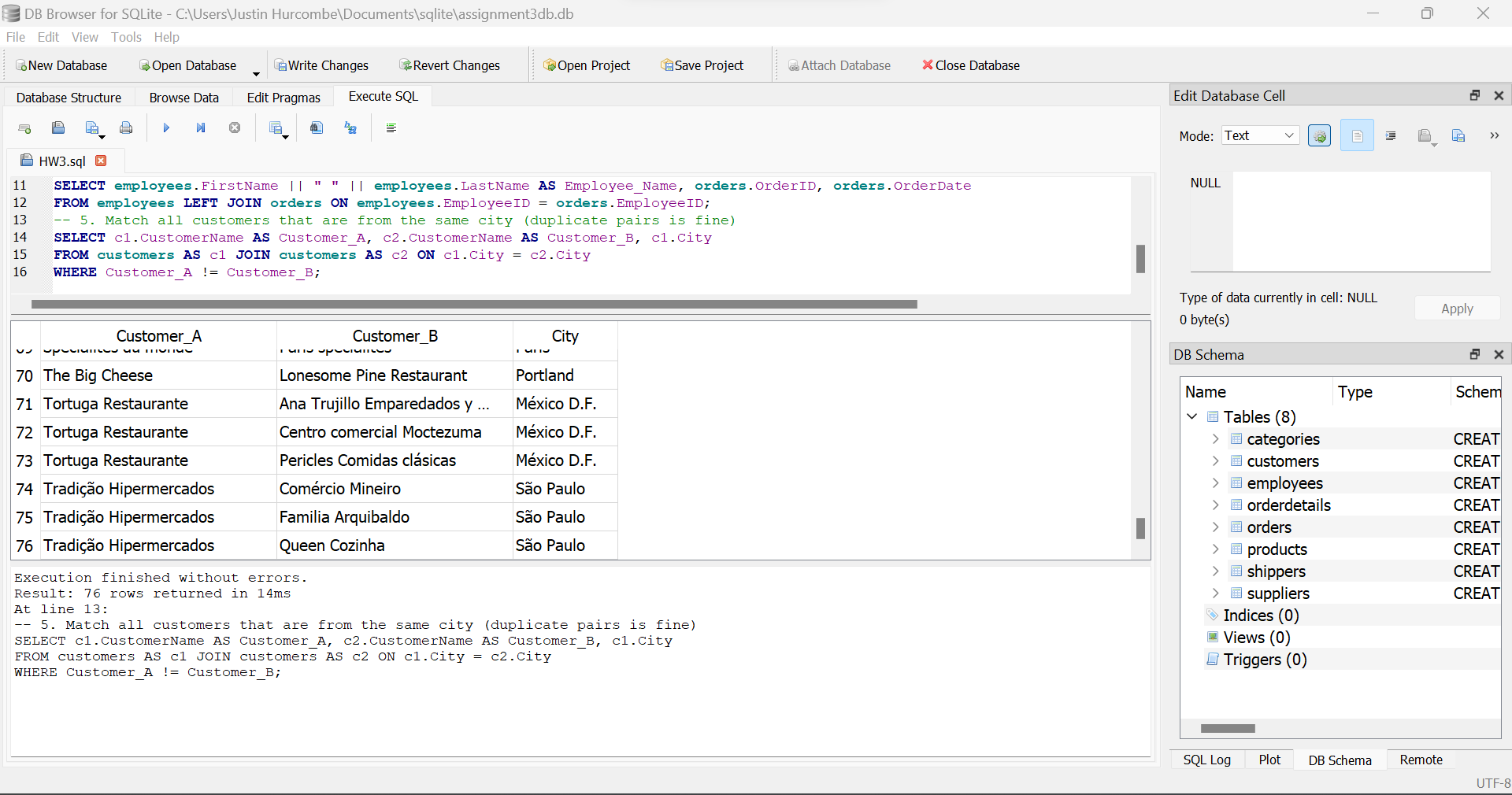
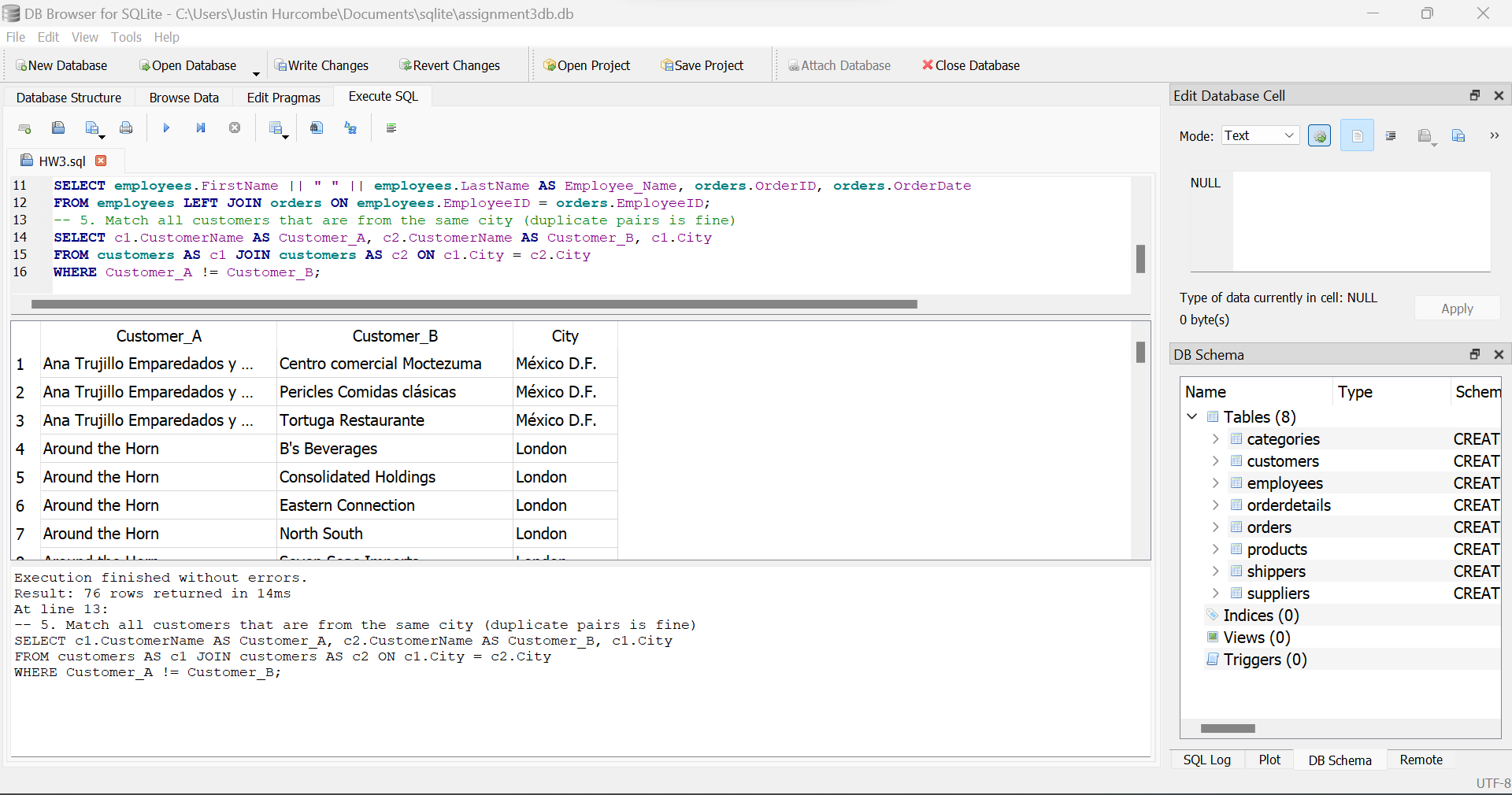


**Match all customers that are from the same city (duplicate pairs is fine, for example, (customerA, customerB, cityA) and (customerB, customerA, cityA) can be both in result).**

SELECT c1.CustomerName AS Customer\_A, c2.CustomerName AS Customer\_B, c1.city

FROM customers AS c1 JOIN customers AS c2 ON c1.City = c2.City

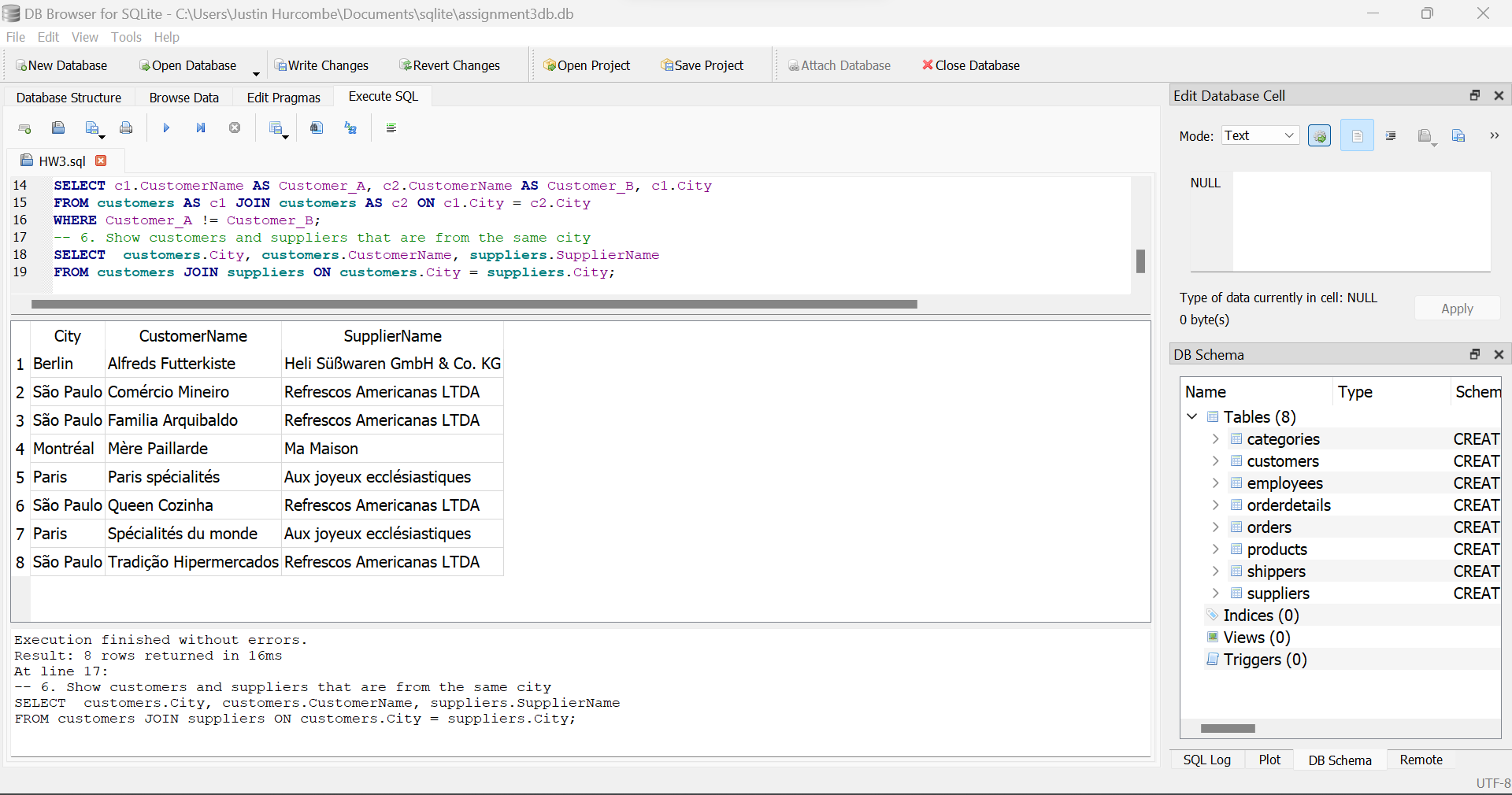
WHERE Customer\_A != Customer\_B;

****

**Show customers and suppliers that are from the same city.**

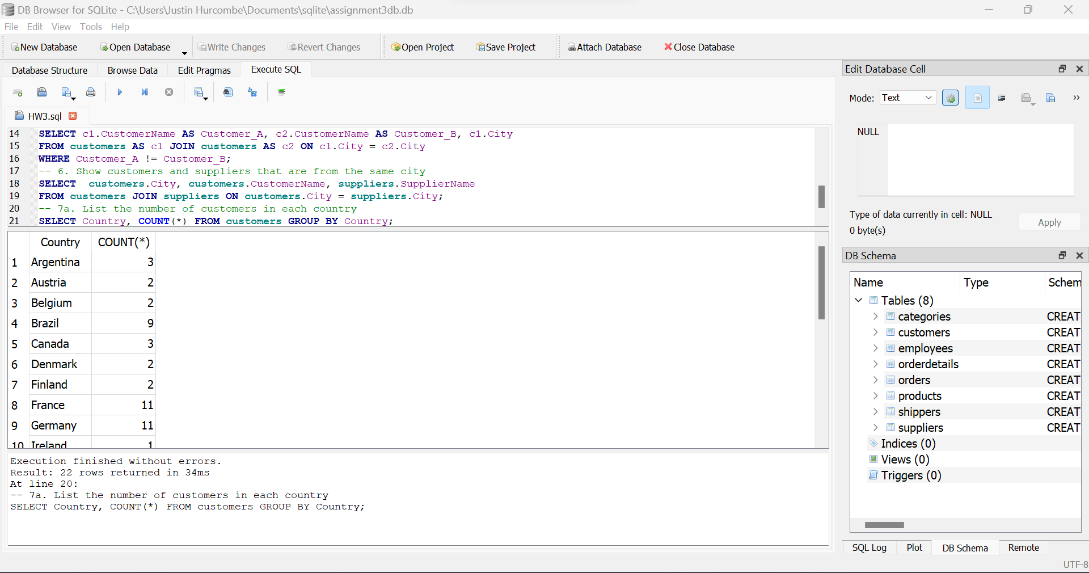
SELECT customers.City, customers.CustomerName, suppliers.SupplierName

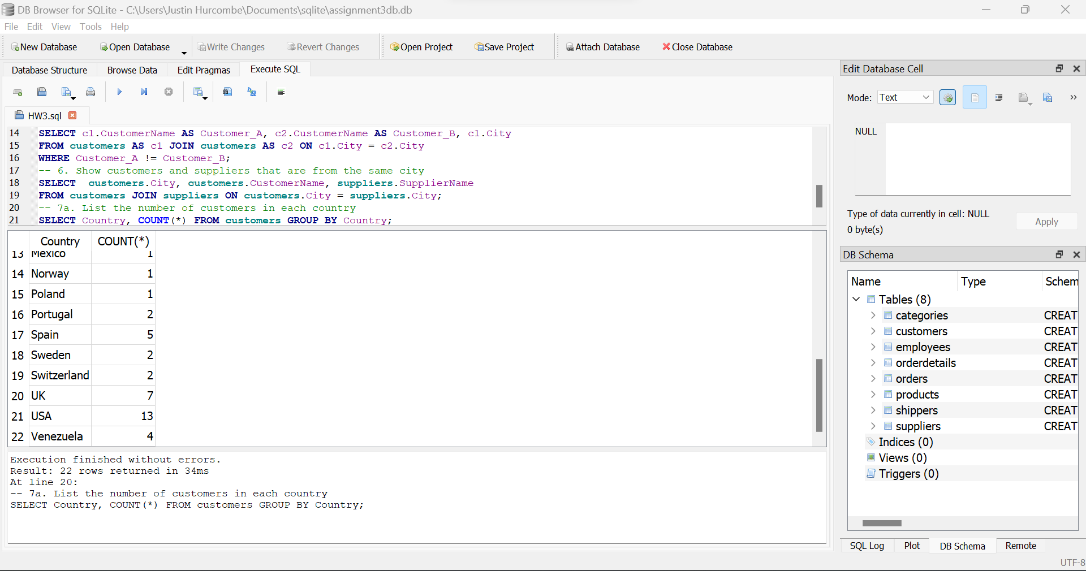
FROM customers JOIN suppliers ON customers.City = suppliers.City;

****

1. **List the number of customers in each country.**

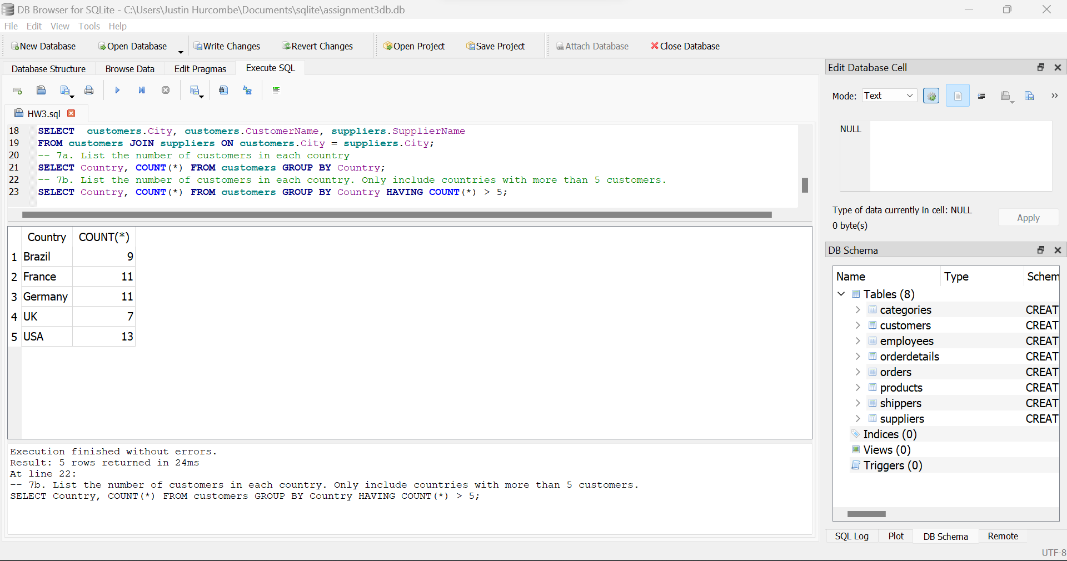
SELECT Country, COUNT(\*) FROM customers GROUP BY Country;

****

****

1. **List the number of customers in each country. Only include countries with more than 5 customers.**

SELECT Country, COUNT(\*) FROM customers GROUP BY Country HAVING COUNT(\*) > 5;

****

**Question 8**

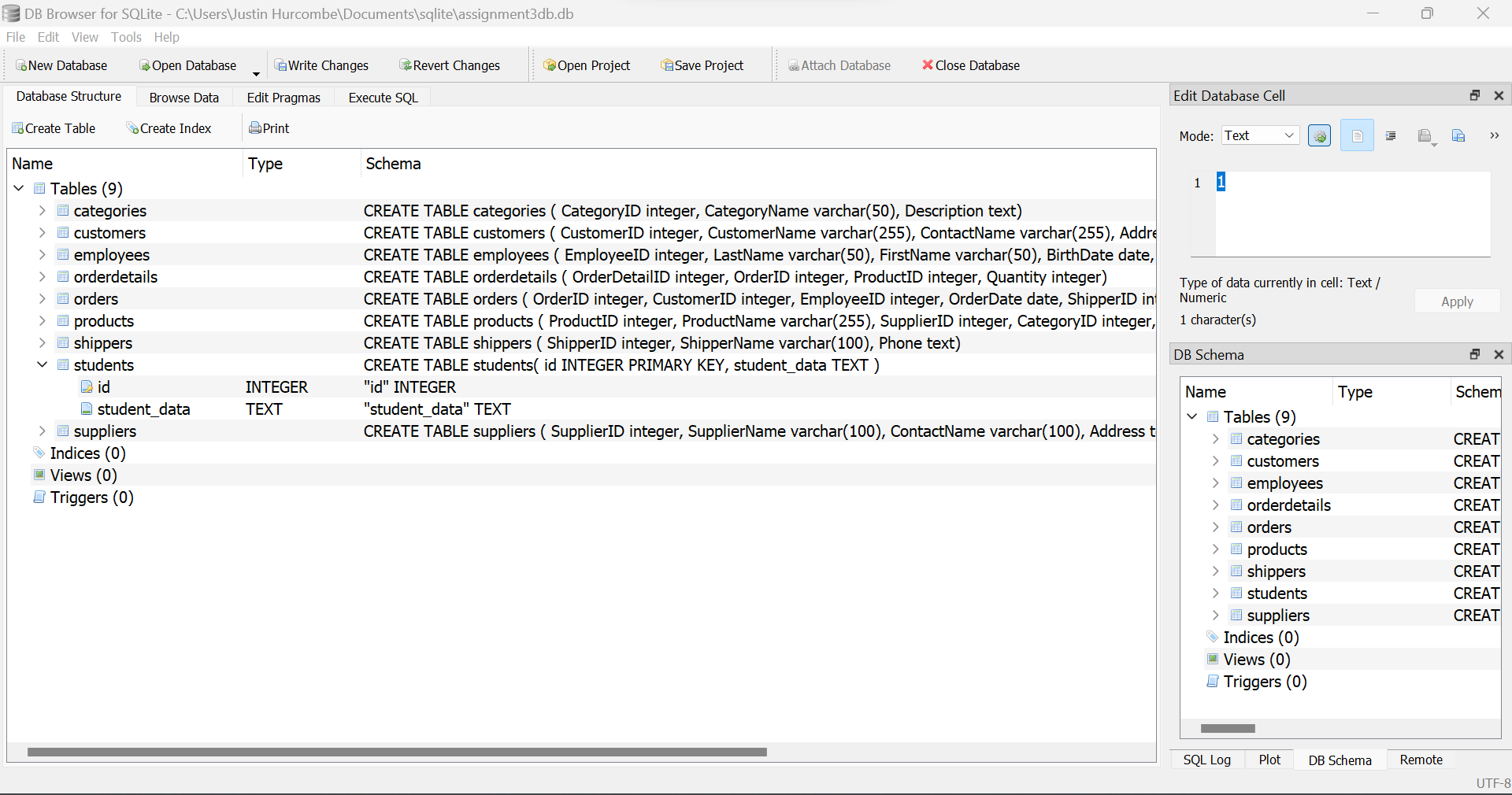
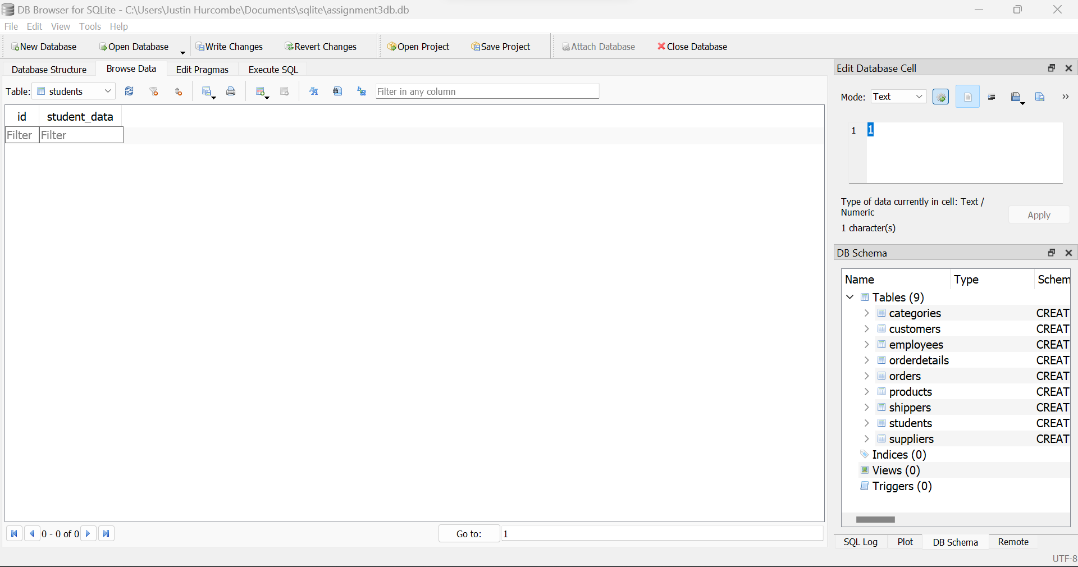
**Create table in your database**

**CREATE TABLE students(**

**Id INTEGER PRIMARY KEY,**

**Student\_data TEXT**

**);**

****

**Finish the following tasks (use SQLITE, if you are able to other database, it is also ok)**

1. **Insert your JSON object into this table**

INSERT INTO students(student\_data) VALUES(JSON(

'{

"student\_id": "S1001", "first\_name": "John", "last\_name": "Doe", "age": 20,

"major": "Computer Science", "enrolled\_courses": [

{

"course\_id": "CS101",

"course\_name": "Introduction to Computer Science", "credits": 4

},

{

"course\_id": "CS102",

"course\_name": "Data Structures and Algorithms", "credits": 4

}

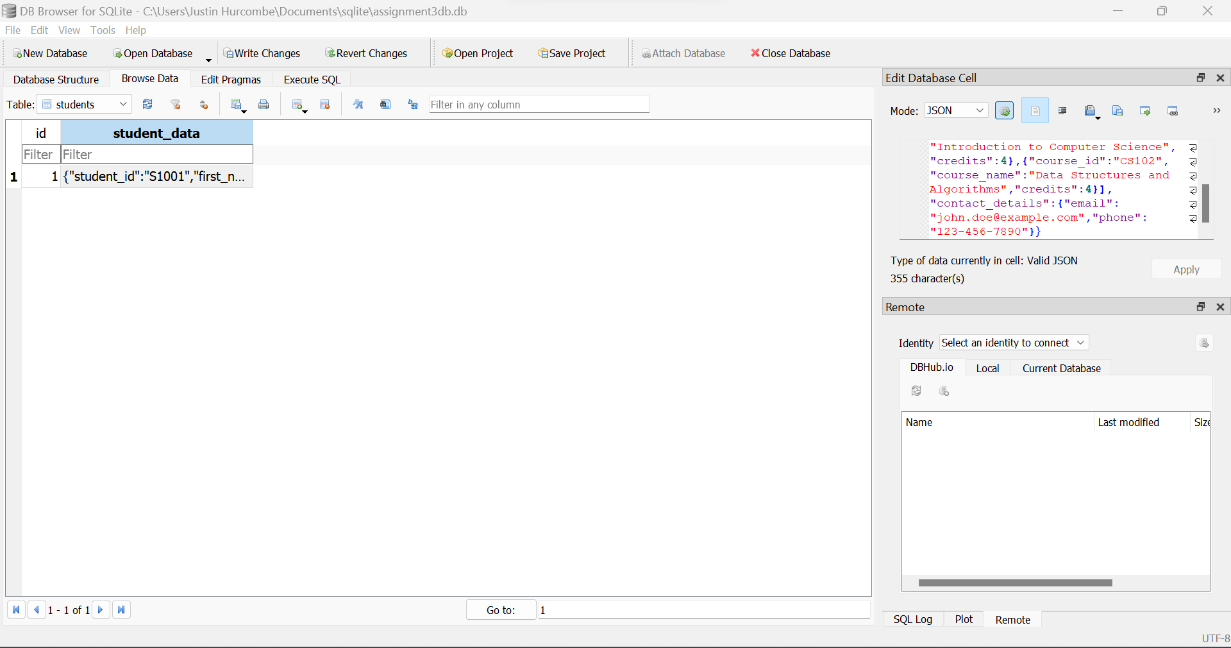
],

"contact\_details": {

"email": "john.doe@example.com", "phone": "123-456-7890"

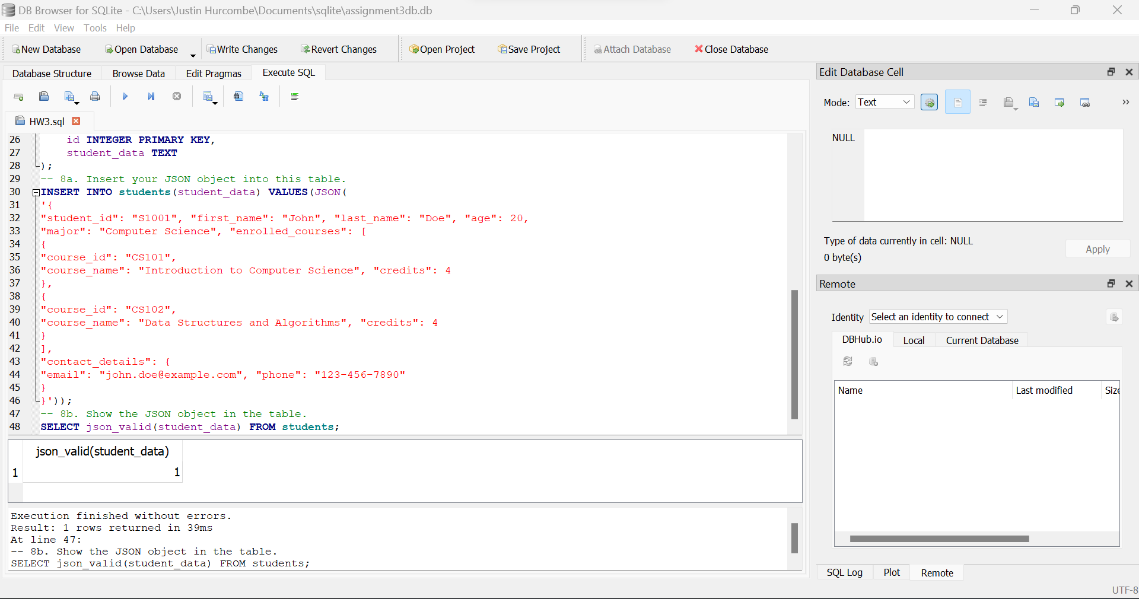
}

}'));



1. **Show the JSON object in the table**

SELECT json\_valid(studnt\_data) FROM students

****

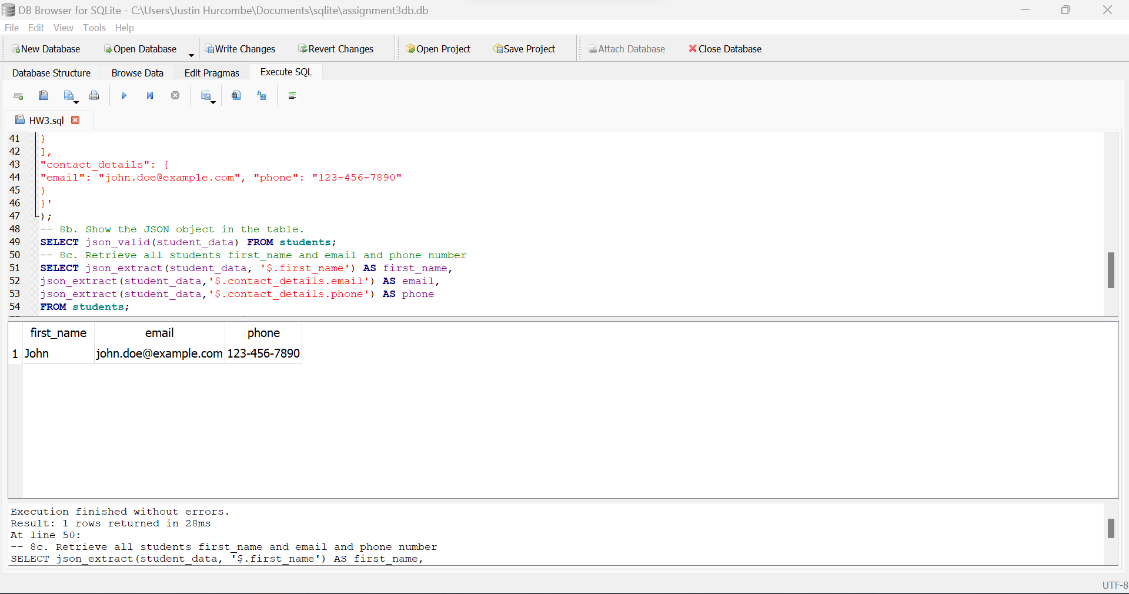
1. **Retrieve all students first\_name and email and phone number**

SELECT json\_extract(student\_data, ‘$.first\_name’) AS first\_name,

json\_extract(student\_data, ‘$.contact\_details.email’) AS email,

json\_extract(student\_data, ‘$.contact\_details.phone’) AS phone

FROM students;

****

1. **Extract all “computer science students”**

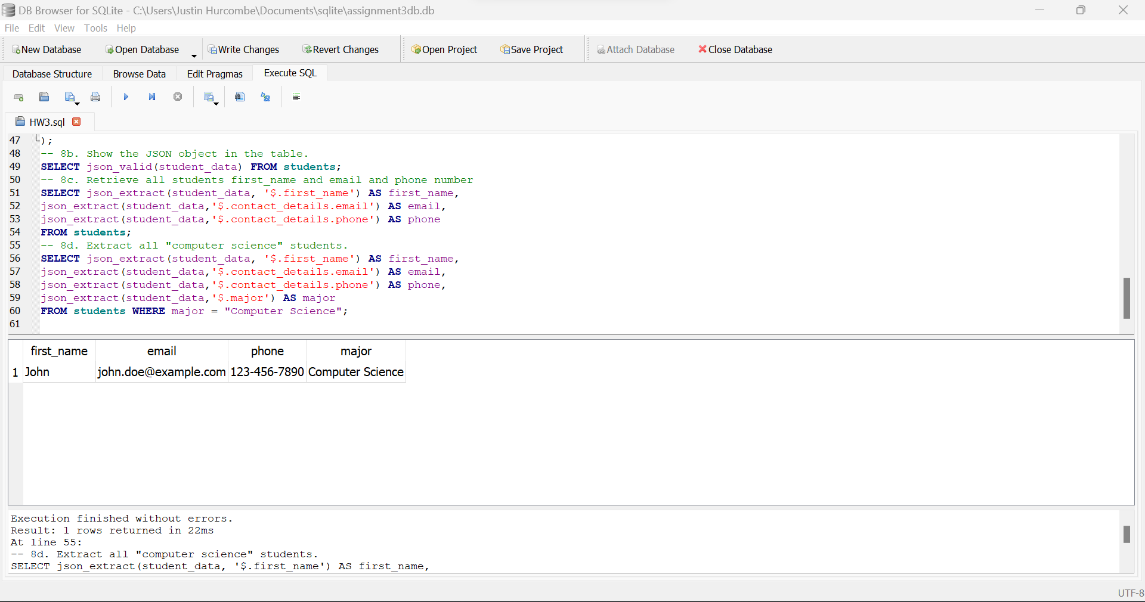
SELECT json\_extract(student\_data, '$.first\_name') AS first\_name,

json\_extract(student\_data,'$.contact\_details.email') AS email,

json\_extract(student\_data,'$.contact\_details.phone') AS phone,

json\_extract(student\_data,'$.major') AS major

FROM students WHERE major = "Computer Science";

****

1. **Display list of courses for each student**

SELECT

students.id,

json\_extract(student\_data, '$.first\_name') AS first\_name,

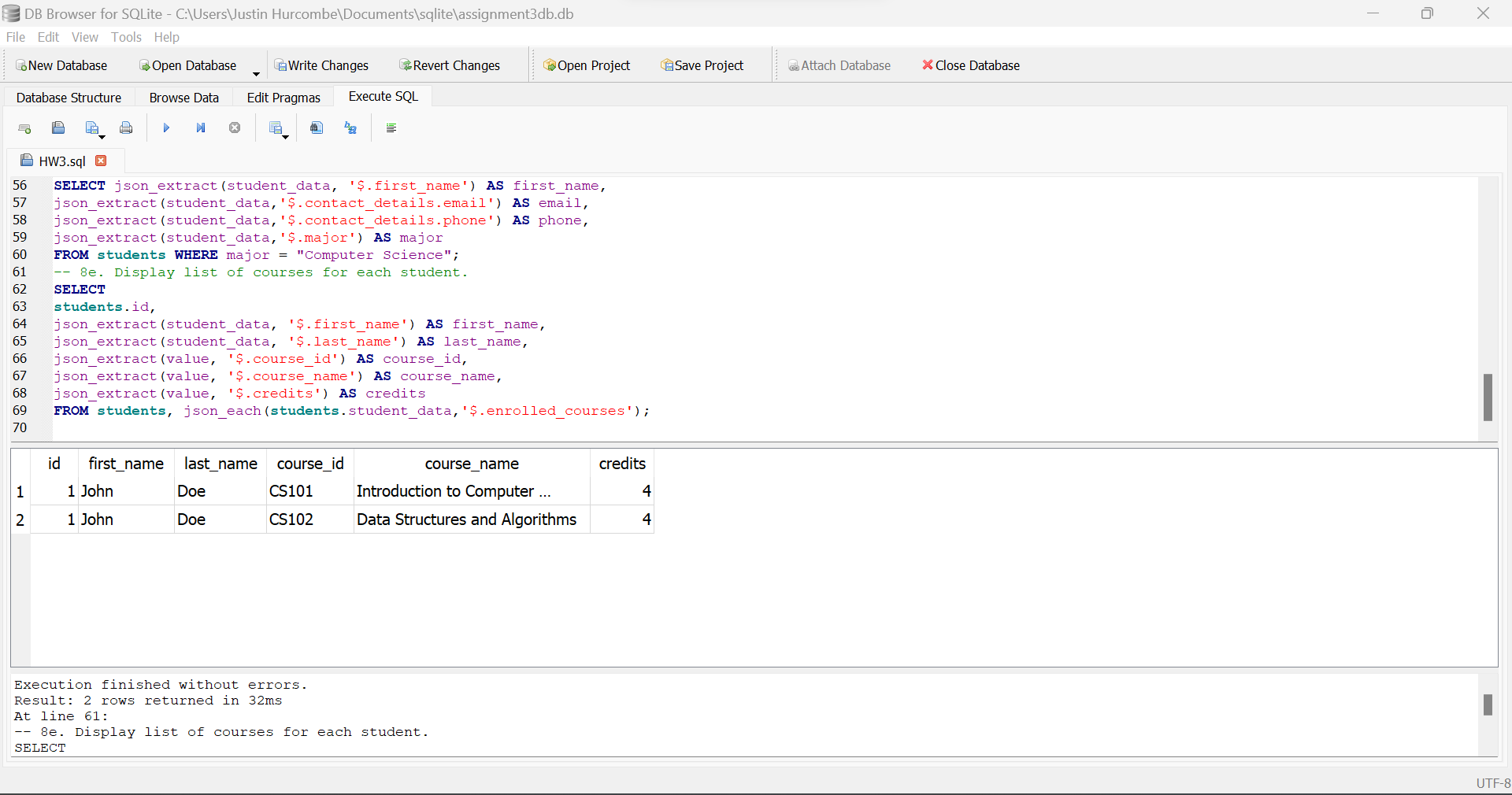
json\_extract(student\_data, '$.last\_name') AS last\_name,

json\_extract(value, '$.course\_id') AS course\_id,

json\_extract(value, '$.course\_name') AS course\_name,

json\_extract(value, '$.credits') AS credits

FROM students, json\_each(students.student\_data,'$.enrolled\_courses');



1. **Display first course for all students**

SELECT

students.id,

json\_extract(student\_data, '$.first\_name') AS first\_name,

json\_extract(student\_data, '$.last\_name') AS last\_name,

json\_extract(student\_data, '$.enrolled\_courses[0].course\_id') AS course\_id,

json\_extract(student\_data, '$.enrolled\_courses[0].course\_name') AS course\_name,

json\_extract(student\_data, '$.enrolled\_courses[0].credits') AS credits

FROM students;

