DS108-05-06 - SQL Final Project

Welcome to the Final Project for the SQL course! Great job making it this far! This Final Project will be different from the Hands-On projects you have previously seen in a couple of different ways. For the Final Project, you will be putting together the numerous topics you have learned into one large project. It is designed to mimic real problems which you may face in your career, so it may be a challenge for you and will also take several hours. Take this project step-by-step and be aware that the project description below is written to be a bit less specific than previous Hands-Ons. The Final Project is supposed to challenge you to do some problem solving to figure out how to accomplish a task. You can always review past lessons or use a Google search if needed. Please read through the following setup instructions before you start the project. Good luck!

Setup

This Final Project is structured into three parts, and each part may ask you to run multiple queries. After each query, please take a screenshot of the MySQL Workbench output and add it to a Word document (or an equivalent) and name this file `SQL-FinalProject`. This way, you will be able to submit your answers to each part all at once.

For this project, you will be creating two tables with the second table referencing the first table. You will need to do this with a foreign key. You will then join these tables together and create a view.

Good luck!

Additional Info! For an example of what is expected for the final, watch this workshop: SQL Final Project.

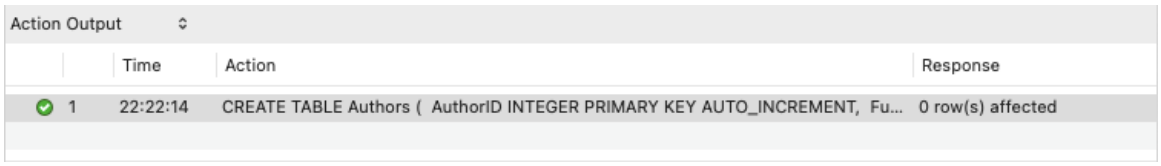
To Submit: Be sure to zip and submit your `SQL-FinalProject` Word document when finished! You will not be able to re-submit, so be sure the screenshots to each part are located within this document. ___

Part 1

1.1 Run a query that creates a table named `Authors`

Run a query that creates a table named `Authors` that has the following columns: AuthorID, FullName, BirthCountry. - AuthorID is the primary key and auto increments.

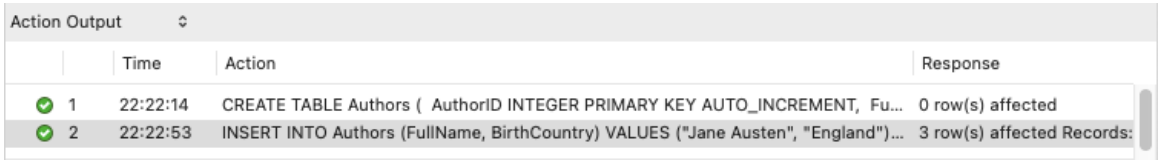
```
/*
AuthorID INTEGER PRIMARY KEY AUTO_INCREMENT
FullName VARCHAR(50)
BirthCountry VARCHAR(50)
*/

CREATE TABLE Authors (
    AuthorID INTEGER PRIMARY KEY AUTO_INCREMENT,
    FullName VARCHAR(50),
    BirthCountry VARCHAR(50)
)
```



1.2 Add the following to Authors table

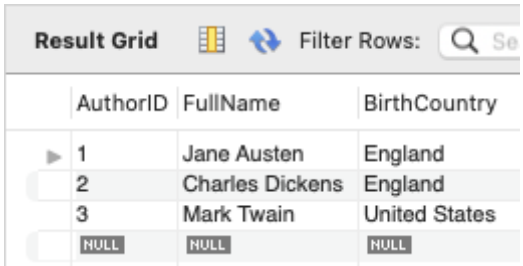
FullName	BirthCountry
Jane Austen	England
Charles Dickens	England
Mark Twain	United States

```
INSERT INTO Authors (FullName, BirthCountry)
VALUES
("Jane Austen", "England"),
("Charles Dickens", "England"),
("Mark Twain", "United States");
```



1.3 Run a query to see all of the authors within the database.

```
SELECT * FROM Authors;
```

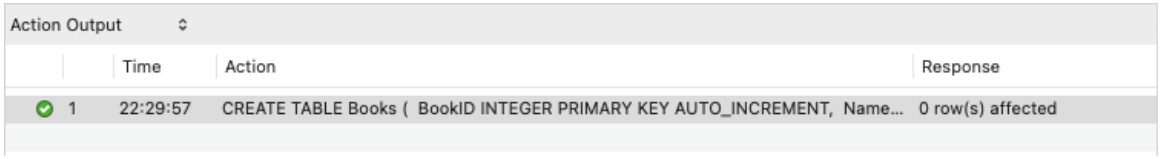


Part 2

2.1 Run a query that creates a table named `Books`

Run a query that creates a table named `Books` that has the following columns: BookID, Name, AuthorID. - BookID is the primary key and auto increments. - AuthorID is a foreign key that referenced the Authors table on the AuthorID column.

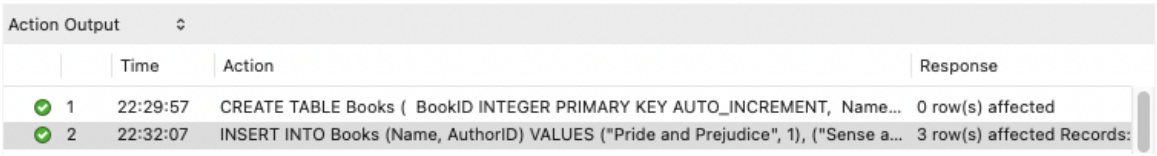
```
CREATE TABLE Books (
    BookID INTEGER PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(50) NOT NULL,
    AuthorID VARCHAR(50) NOT NULL,
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);
```



2.2 Add the following the Books table

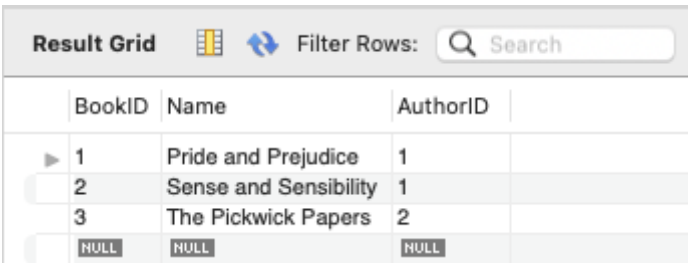
Name	AuthorID
Pride and Prejudice	1
Sense and Sensibility	1
The Pickwick Papers	2

```
INSERT INTO Books (Name, AuthorID)
VALUES
("Pride and Prejudice", 1),
("Sense and Sensibility", 1),
("The Pickwick Papers", 2);
```



2.3 Run a query to see all of the books within the database.

```
SELECT * FROM Books;
```

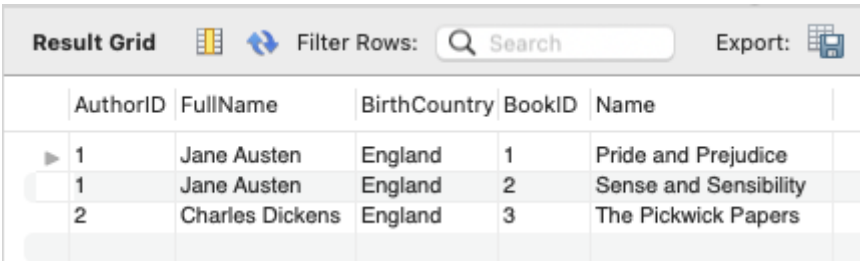


Part 3

3.1 Run a query that joins the Authors and Books tables

Run a query that joins the Authors and Books table together using the `AuthorID` foreign key.

```
/* AuthorID is key for overlap */

SELECT * FROM Authors
JOIN Books USING (AuthorID);
```

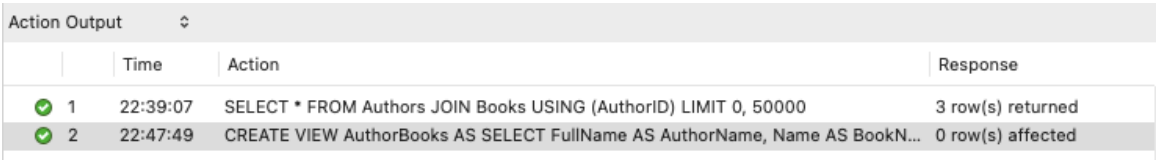


AuthorID	FullName	BirthCountry	BookID	Name
1	Jane Austen	England	1	Pride and Prejudice
1	Jane Austen	England	2	Sense and Sensibility
2	Charles Dickens	England	3	The Pickwick Papers

3.2 Create a view called `AuthorBooks`

Next, create a view named `AuthorBooks` using the join query created in step 1 adding the following parameters: - Show only the Authors full name and book name. - Rename the column name results using the `AS` keyword. - The Authors FullName should display as AuthorName. - The Books Name should display as BookName. - Order the results alphabetically by the authors full name.

```
CREATE VIEW AuthorBooks AS
SELECT FullName AS AuthorName, Name AS BookName
FROM Authors
JOIN Books USING (AuthorID)
ORDER BY AuthorName;
```



3.3 Run a query to see the view

Lastly, run a query to see the view you just created.

```
SELECT * FROM AuthorBooks;
```