Owen Lindsey

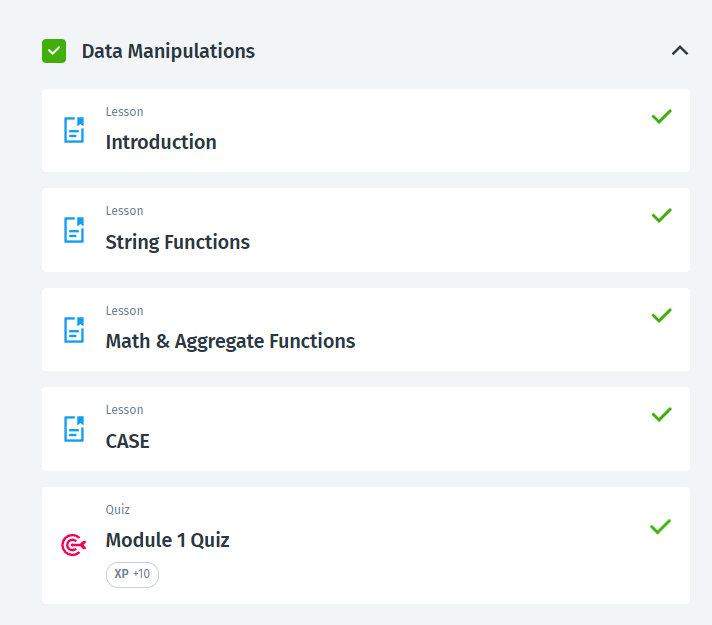
Professor Sluiter

10/22/2023

Activity 4

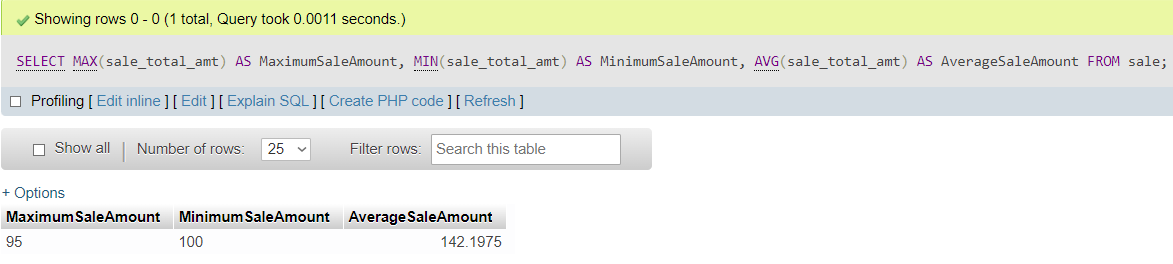
CST-345

Part 1: SoloLearn certification:



Part 2: Database Design Concepts

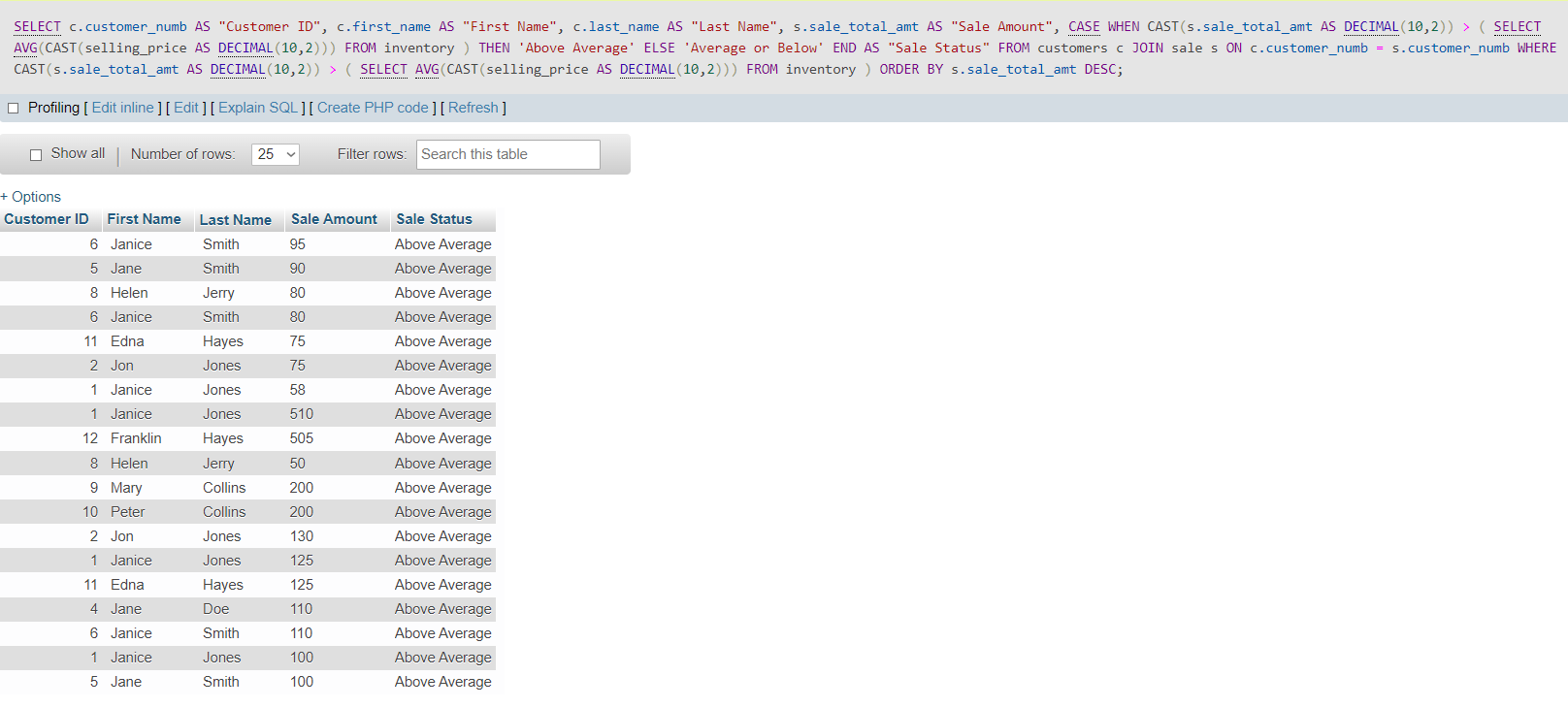
1. Execute a join statement :

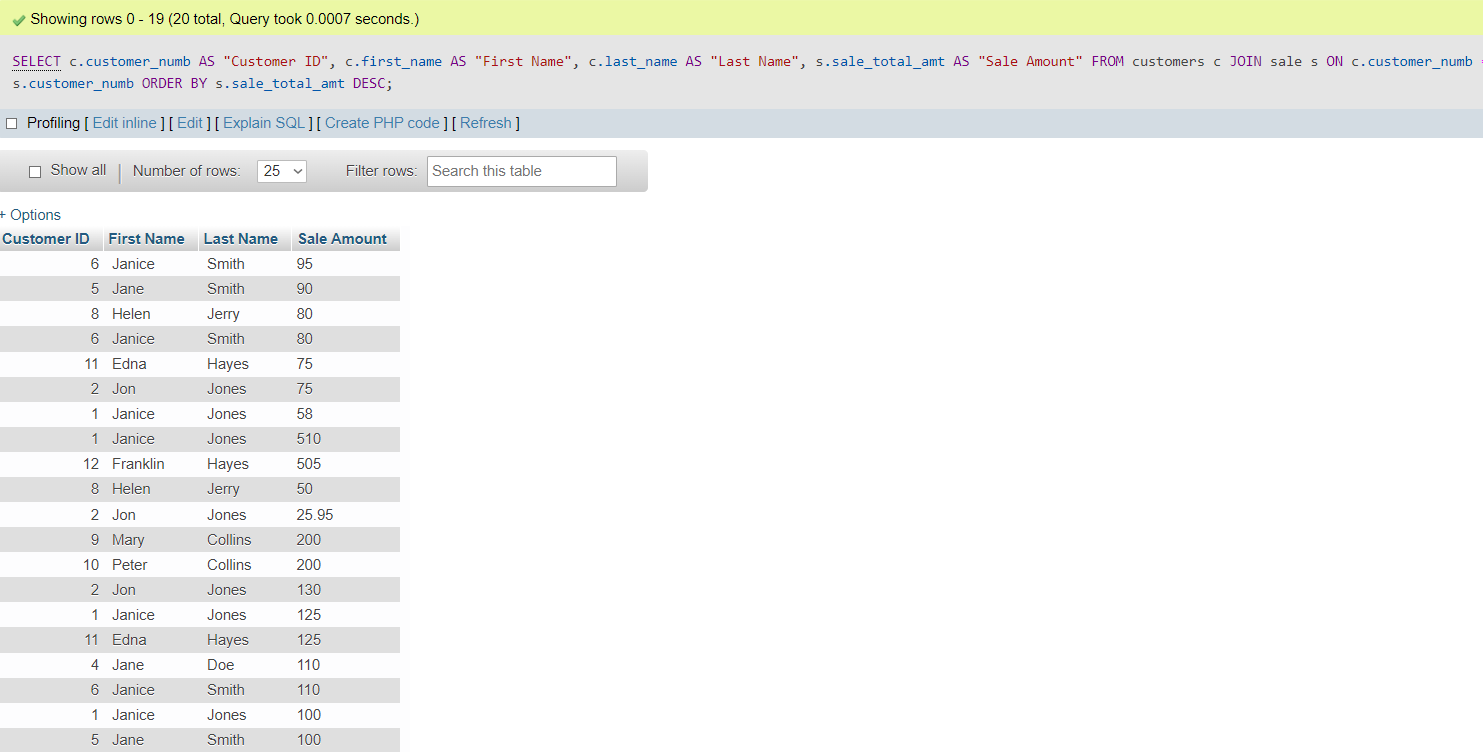


1. Start with the following select statement, which will return all of the book data for items that were sold.

A screenshot of a computer

Description automatically generated

1. Modify the following select statement, which returns all of the sales and shows the name of the customer.
2. Modify the select statement to show each customer ID and first and last name with a total amount (sum) of money that the customer has spent at the store.



1. A screenshot of a table

   Description automatically generatedModify the select statement to show each customer ID and first and last name with a total amount (sum) of money that the customer has spent at the store.

B.) The three examples of database normalization errors, the explanations for why each is designed incorrectly, and a solution for each.

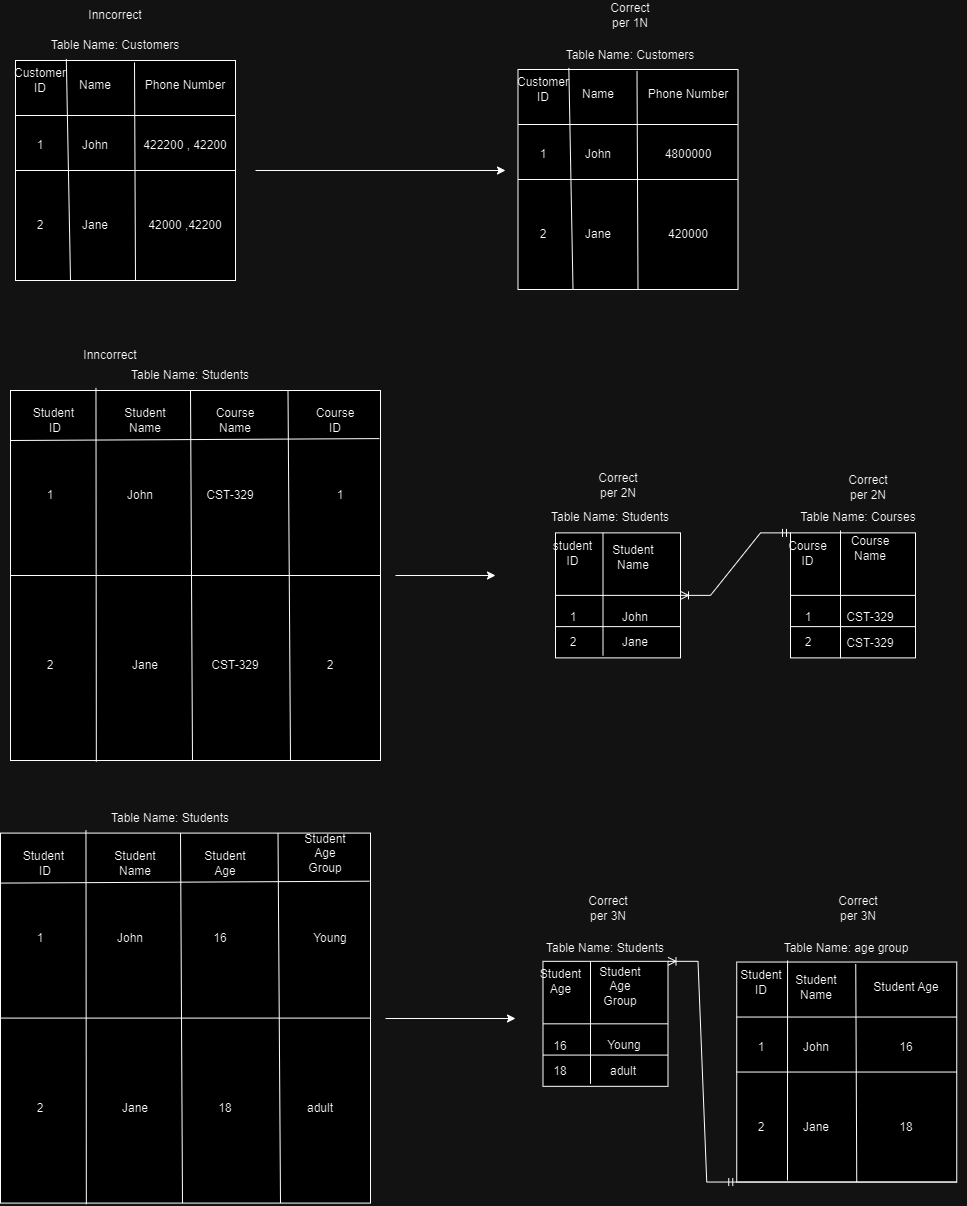


Table group a.) First Normal Form (1NF): In this fix, we addressed the violation of storing multiple values in a single column, which is a common issue when dealing with delimited values like phone numbers. By splitting the phone\_numbers column into multiple rows, we achieved 1NF compliance. This change ensures that each column contains atomic (indivisible) values, making it easier to manage and query the data.

Table group b.) Second Normal Form (2NF): The problem here was partial dependency on the primary key in a composite primary key table. To resolve this, we split the table into two: Student and Course. This separation eliminates partial dependency, as each table now has a single-column primary key. It also makes the database more maintainable and scalable, ensuring that modifications to one entity won't affect another.

Table group C.) Third Normal Form (3NF): The initial table had a transitive dependency on the primary key, where the age\_group was determined by the student\_age. To correct this, we split the table into Student and AgeGroup. This move removed the transitive dependency, making the database more flexible and preventing data anomalies. Now, any changes to student details won't affect the age group, and vice versa.

Part 3: Music app

Screenshot of populated form:

