ROBERT GORDON UNIVERSITY ABERDEEN

INFORMATICS INSTITUTE OF TECHNOLOGY

BSc (Hons) Artificial Intelligence and Data Science

Module: CM1603 Database Systems

Individual Coursework Report

Module Leader: Ms. Dileeka Alwis

RGU Student ID: 2330910

IIT Student ID: 20230374

Student Name: Setunge Mudalige Don Chirath Shamika Setunge

Acknowledgment

I would like to express my profound gratitude to Mrs. Dileeka Alwis for her tremendous collaboration and support in making my project achieve such a noteworthy outcome. Not to mention, I would want to thank my friends and supporters, without whom the work could not have been completed effectively and on time.

Table of content

Table of Figures

Section 1

Entities

- Customer
- OnlineOrder
- Item
- Supplier
- Delivery
- BankTransferPayment
- OnlinePayment
- Book
- Stationery

Primary Keys

- Customer: customerId
- OnlineOrder: orderId
- Item: itemCode
- Supplier: supplierId
- Delivery: deliveryId
- BankTransferPayment: bankPaymentId
- OnlinePayment: onlinePaymentId

Non-Key Attributes

- Customer: name (f_name, l_name), eMail, address (stNo, street, city, postal_code), tellNo[1..3], password
- OnlineOrder: orderDate, orderStatus, totalPrice
- Item: price, stockLevel, reorderLevel
- Supplier: supplierType, name, eMail, address (stNo, street, city, postal_code), tellNo[1..3]
- Delivery: deliveryDate, deliveryAddress (stNo, street, city, postal_code), deliveryStatus
- BankTransferPayment: amount, paymentDate, bankName, accNo, accHolderName
- OnlinePayment: amount, paymentDate, transactionId, cardType, cardNumber, expiryDate, cvv
- Book: ISBN, title, genre, category, author, publisher, yearOfPublication
- Stationery: type, brand, color, size

Relationship Between Entities

- Customer
  - Customer *places* OnlineOrder
  - Customer *makes* OnlinePayment
  - Customer *makes* BankTransferPayment
  - Customer *receives* Delivery
- OnlineOrder
  - OnlineOrder *is_for* Item
  - OnlineOrder *Leads_to* Delivery
- Supplier
  - Supplier *supplies* Item
- OnlinePayment
  - OnlinePayment *confirms* OnlineOrder
- BankTransferPayment
  - BankTransferPayment *confirms* OnlineOrder

Multiplicity Constraints

- Each customer may place one or many online orders, while ensuring that each order is linked to exclusively to a single customer.
- Each customer may make one or many online payments, while ensuring that each online payment is linked to exclusively to a single customer.
- Each customer may make one or many bank transfer payments, while ensuring that each bank transfer payment is linked to exclusively to a single customer.
- An online payment can only confirm one online order, while each online order must be confirmed by one or many online payments.
- A bank transfer payment can only confirm one online order, while each online order must be confirmed by one or many bank transfer payments
- An online order is for one or many items, while each item may include in one or many online orders.
- Each supplier must supply at least a single item or supplier can supply many items; each item is supplied by one or many suppliers.
- At the end each online order may leads to a delivery, while each delivery is linked to exclusively to a single online order. This shop offer ‘store pickup’ facility that why online orders may leads to a delivery.

- Each customer may receive one or many deliveries, while ensuring that each delivery is linked to exclusively to a single customer.

Assumptions

- Customer may place one or many online orders and to confirms that one or many online orders customer may make one or many online or bank transfer payments.
- The Book Haven store supports for partial payments, allowing customers to pay for an online order in instalments. Due to expansive items but store doesn't release any items until store receives the full payment.
- If customer doesn't place an online order, then customer won't make payments and won't receive any deliveries.
- It is expected that a consumer will stick to a single payment method when confirming an online order.
- Items are specialized as books and stationeries, this online shop only sales books and stationeries.
- Only reading books are considered as books, while writing books are considered as stationeries.
- Supplier table entity supplier type attribute ensures whether the supplier is individual person or a company.
- In its operational boundaries, this online store only offers domestic shipping. The portal only serves local customers and engages with local suppliers.
- Customers send their online orders to different delivery address not only their private home address.
- orderStatus in OnlineOrder has these different statuses ('pending', 'processing', 'out for delivery', 'delivered', 'store pickup').
- deliveryStatus in Delivery has these different statuses ('pending', 'in transit', 'delivered').

EERD (Extended Entity Relationship Diagram)

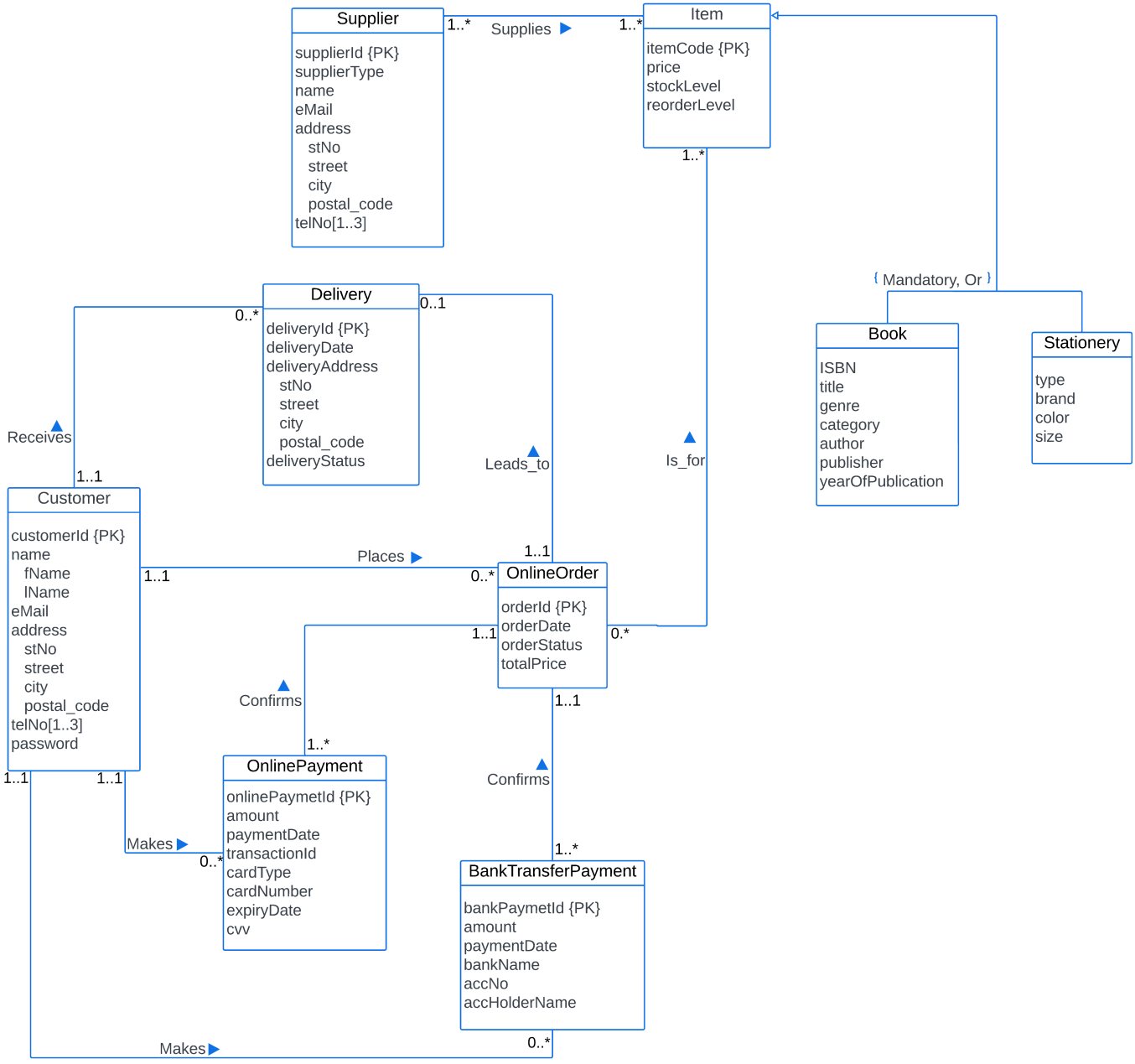


Figure 1 Extended Entity Relationship Diagram

Section 2

Relational Schema Diagram

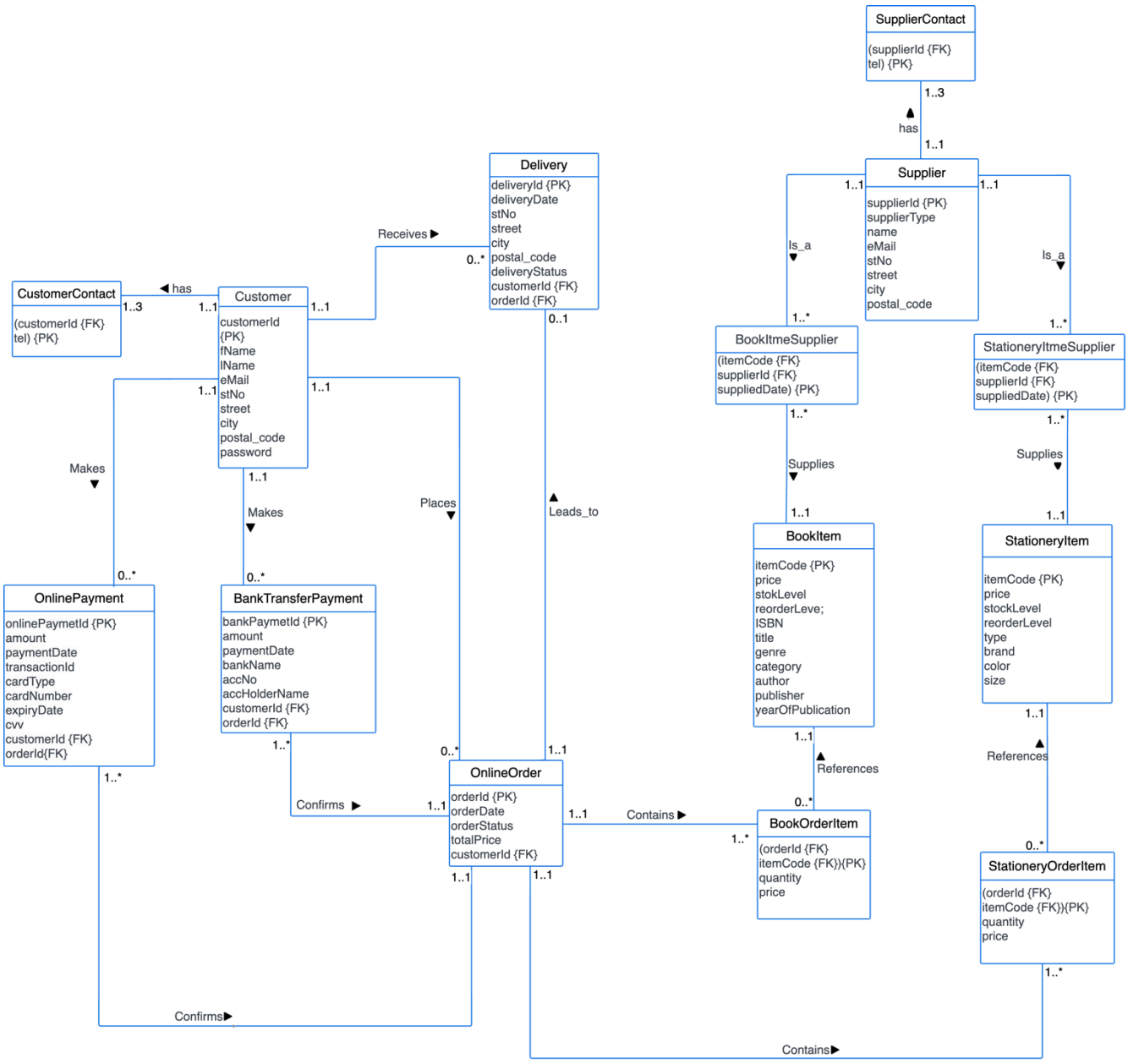


Figure 2 Relational Schema Diagram

Relational Schema Diagram Assumptions

- Created a new table called CustomerContact and assumed that the telNo in Customer table was a multi-valued attribute.
- Created a new table called SupplierContact and assumed that the telNo in Supplier was a multi-valued attribute.
- Created a new two table called BookItem and StationeryItem, assumed that the Item table is specialized as Book and Stationery.
- Created a new table called StationeryItemSupplier and assumed a many-to-many multiplicity constraint between Supplier and Item.
- Created a new table called BookItemSupplier and assumed a many-to-many multiplicity constraint between Supplier and Item.
- Created a new table called BookOrderItem and assumed a many-to-many multiplicity constraint between OnlineOrder and Item. Add quantity and price attributes to BookOrderItem.
- Created a new table called StationeryOrderItem and assumed a many-to-many multiplicity constraint between OnlineOrder and Item. Add quantity and price to StationeryOrderItem.
- customerId was added as a foreign key in OnlinePayment, assuming that the multiplicity constraint between Customer and OnlinePayment was one-to-many.
- customerId was added as a foreign key in BankTransferPayment, assuming that the multiplicity constraint between Customer and BankTransferPayment was one-to-many.
- customerId was added as a foreign key in OnlineOrder, assuming that the multiplicity constraint between Customer and OnlineOrder was one-to-many.
- customerId was added as a foreign key in Delivery, assuming that the multiplicity constraint between Customer and Delivery was one-to-many.
- orderId was added as a foreign key in OnlinePayment, assuming that the multiplicity constraint between OnlineOrder and OnlinePayment was one-to-many.
- orderId was added as a foreign key in BankTransferPayment, assuming that the multiplicity constraint between OnlineOrder and BankTransferPayment was one-to-many.
- orderId was added as a foreign key in Delivery, assuming that the multiplicity constraint between OnlineOrder and Delivery was one-to-many.

Section 3

Table Creation and Population of Data

Creating The Database

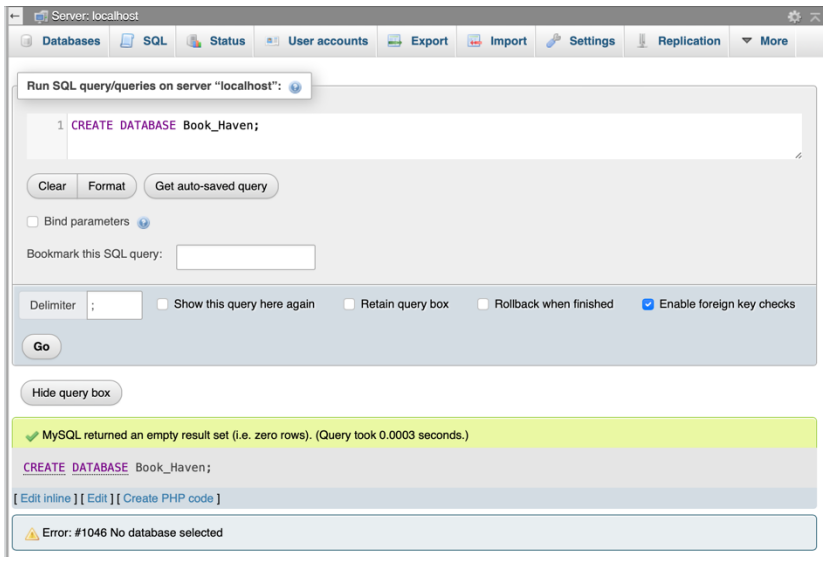


Figure 3 Query to create the Database.

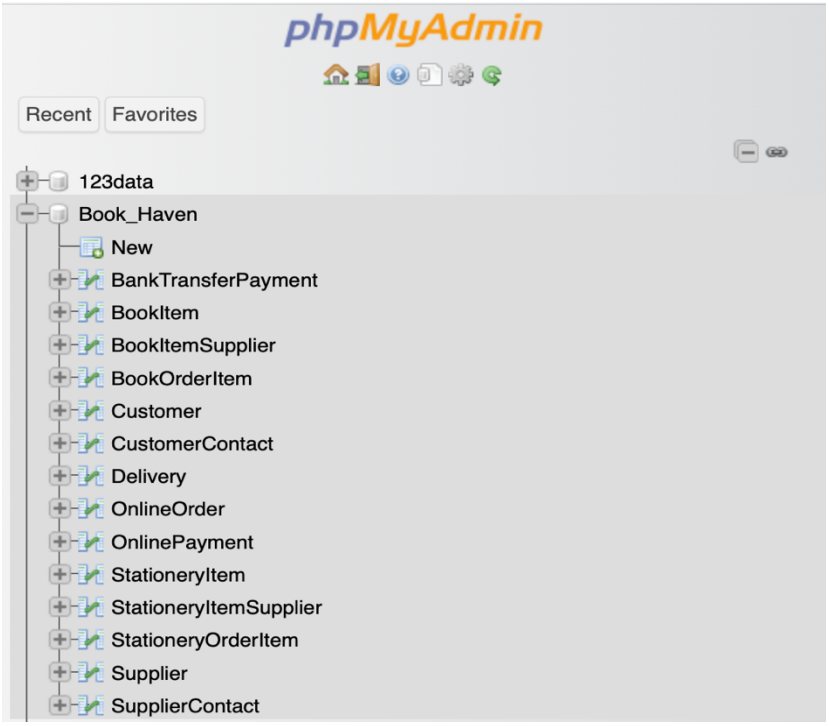


Figure 4 Created Database.

Table Creation

Customer Table

- Use AUTO_INCREMENT function to increment customerId number.

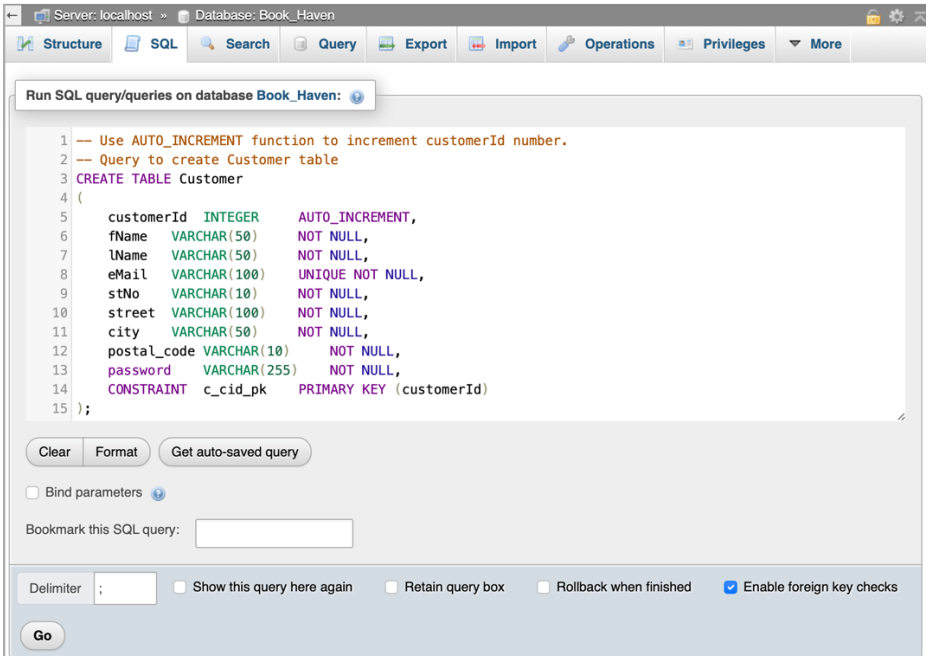


Figure 5 Query to create Customer table.

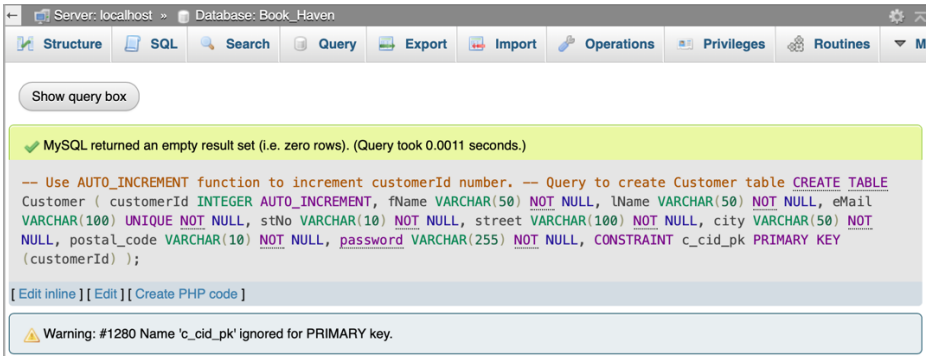


Figure 6 Customer table created successfully.

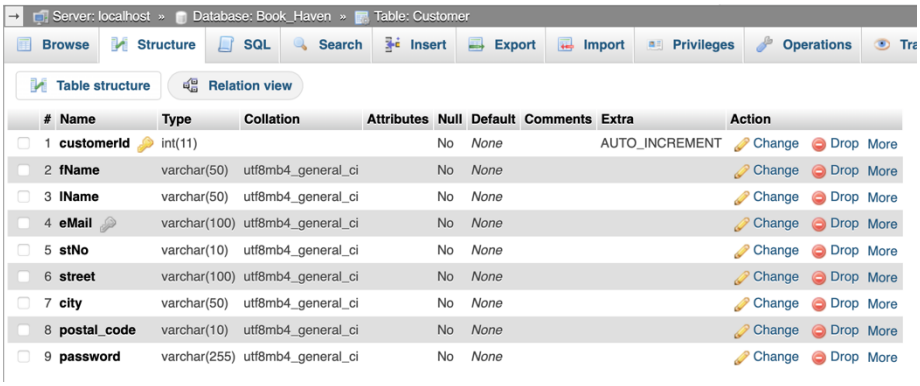


Figure 7 Customer table column specifications

CustomerContact Table

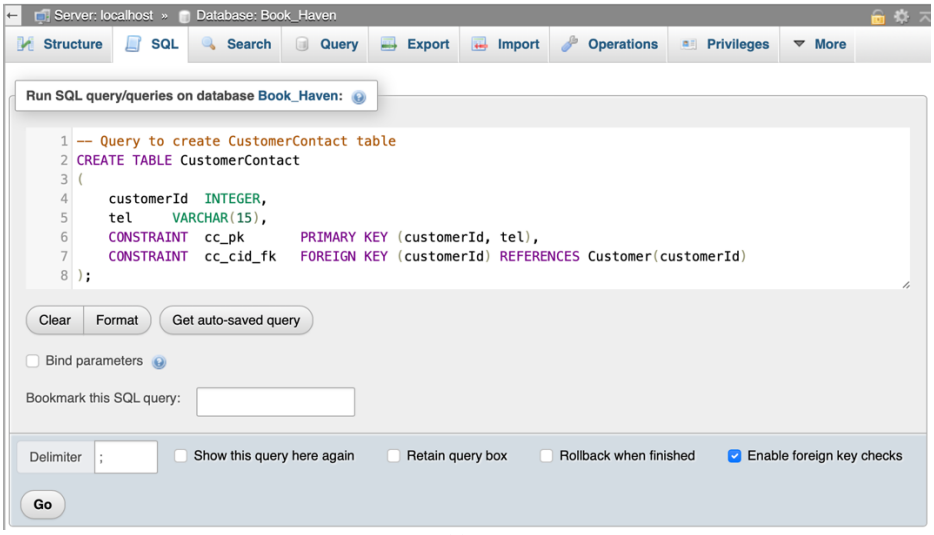


Figure 8 Query to create CustomerContact table.

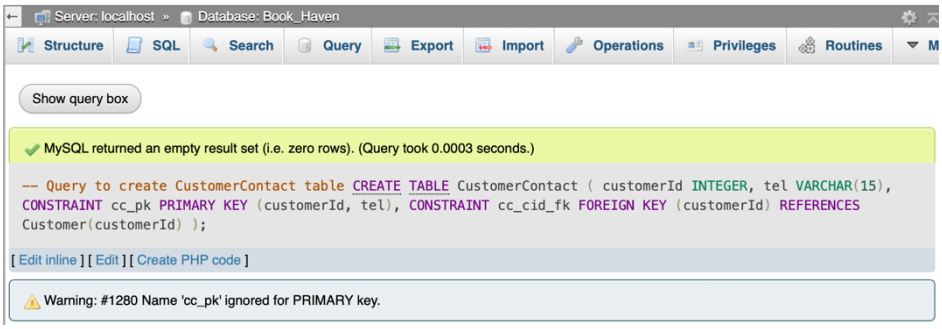


Figure 9 CustomerContact table created successfully.

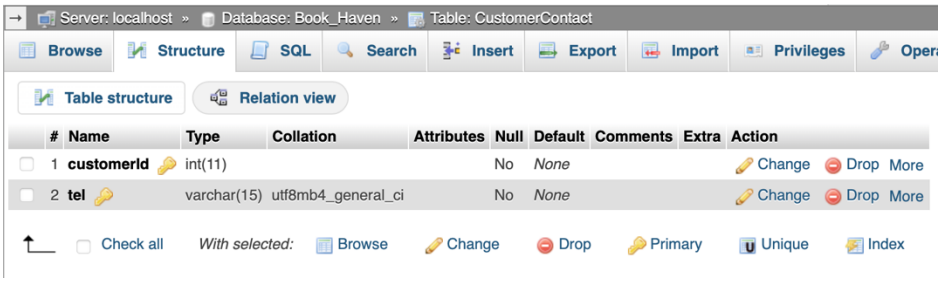


Figure 10 CustomerContact table Column Specifications.

OnlineOrder Table

- Use CHECK function to check whether the orderStatus only in ('pending', 'processing', 'out for delivery', 'delivered', 'store pickup').

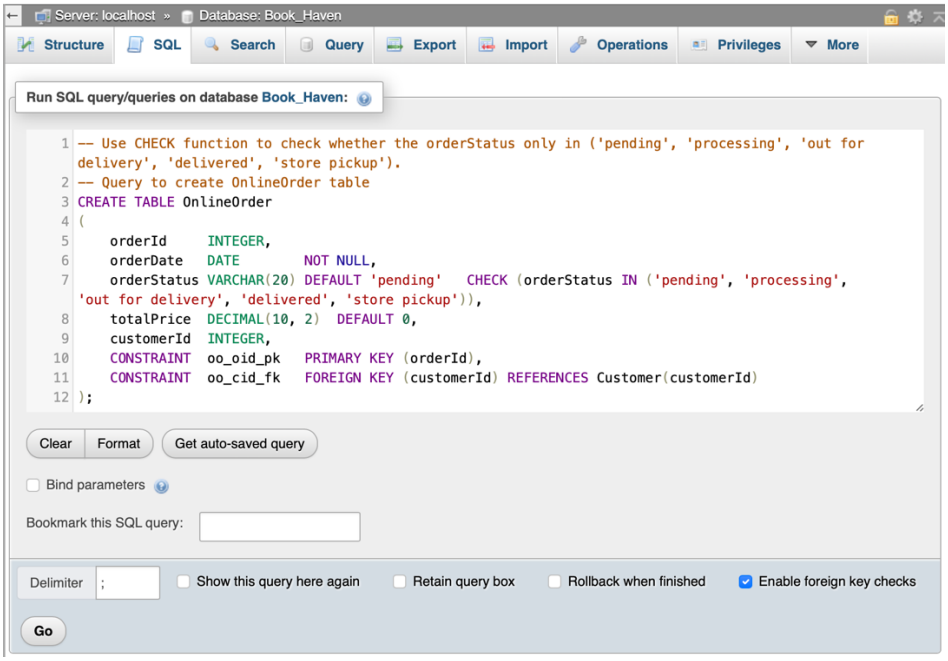


Figure 11 Query to create OnlineOrder table.

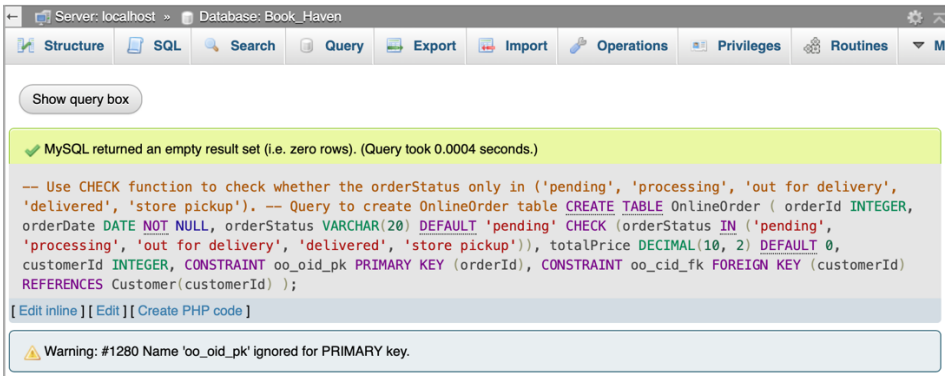


Figure 12 OnlineOrder table created successfully.

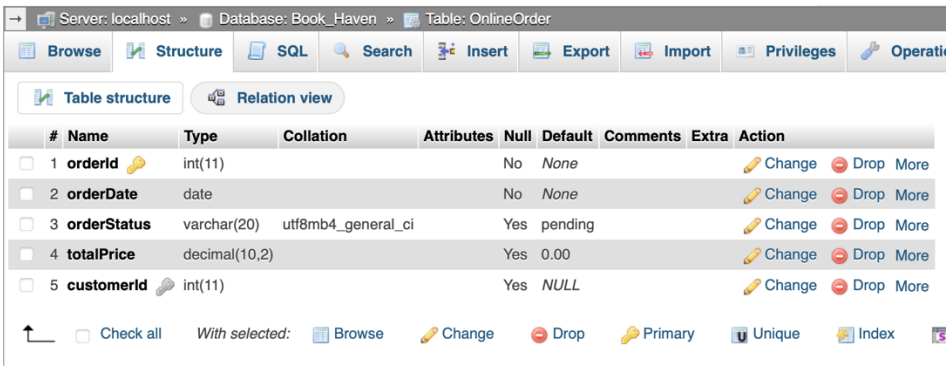


Figure 13 OnlineOrder table Column Specifications.

OnlinePayment Table

- Use AUTO_INCREMENT function to increment onlinePaymentId number.

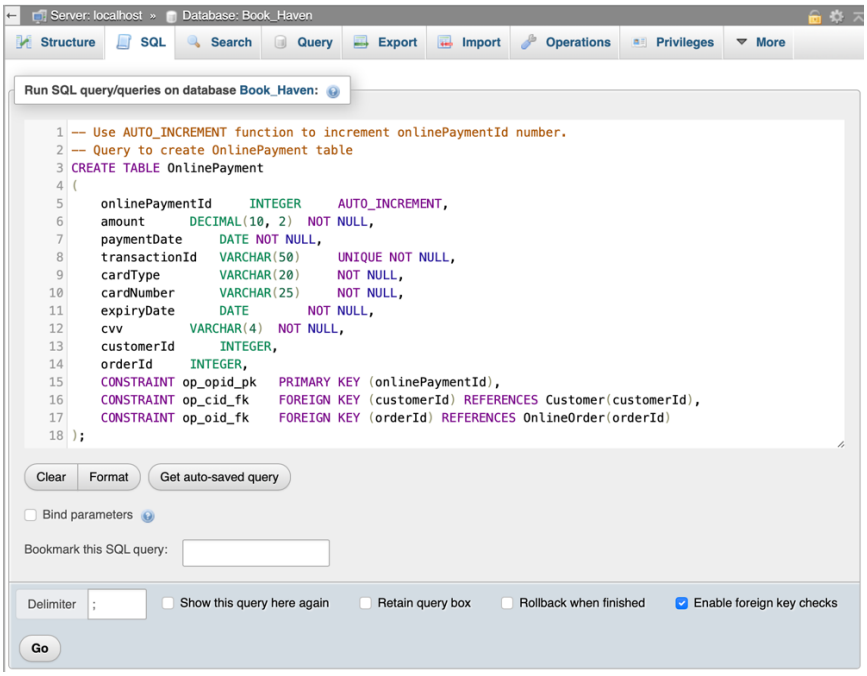


Figure 14 Query to create OnlinePayment table.

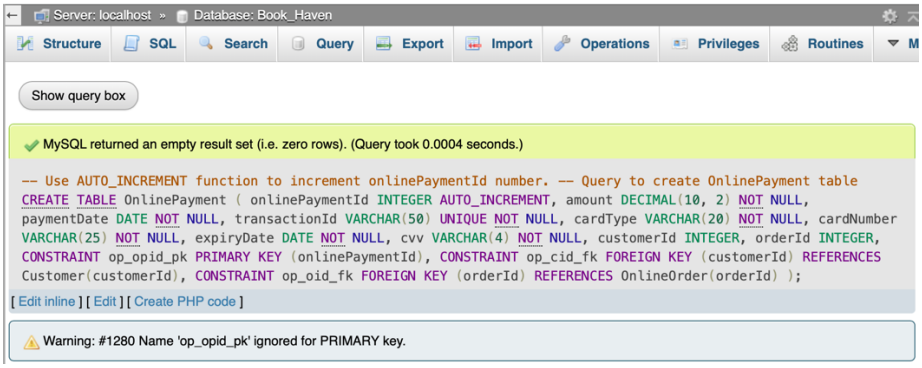


Figure 15 OnlinePayment table created successfully.

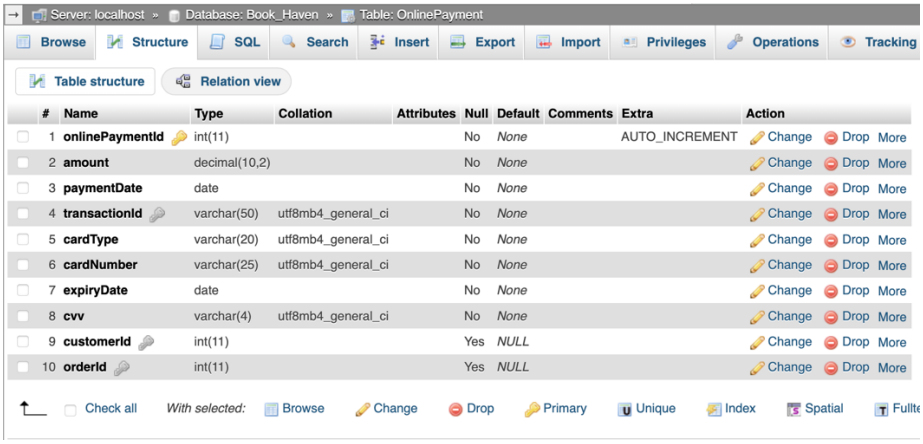


Figure 16 OnlinePayment table Column Specifications.

BankTransferPayment Table

- Use AUTO_INCREMENT function to increment bankTransferPaymentId number.

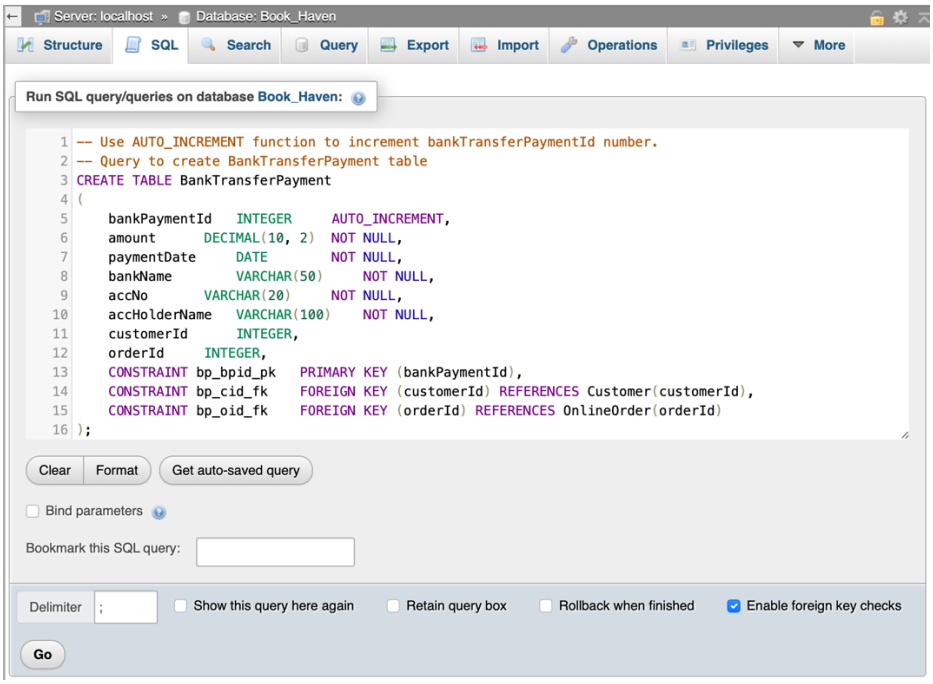


Figure 17 Query to create BankTransferPayment table.

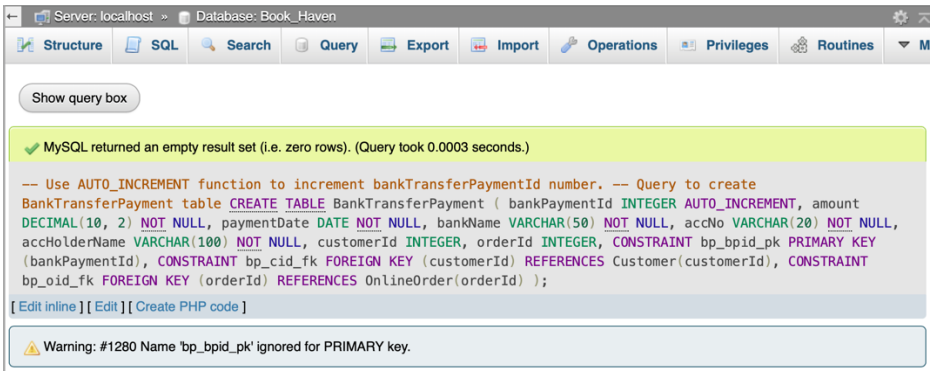


Figure 18 BankTransferPayment table created successfully.

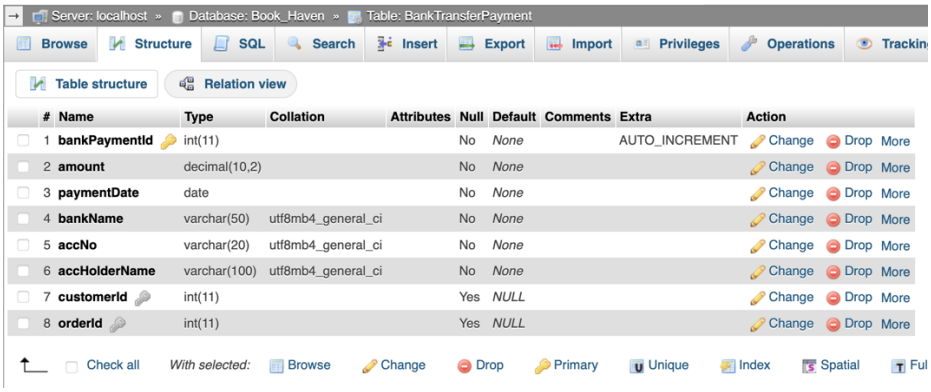


Figure 19 BankTransferPayment table Column Specifications.

Delivery Table

- Use AUTO_INCREMENT function to increment bankTransferPaymentId number. Use CHECK function to check whether the orderStatus only in ('pending', 'in transit', 'delivered').

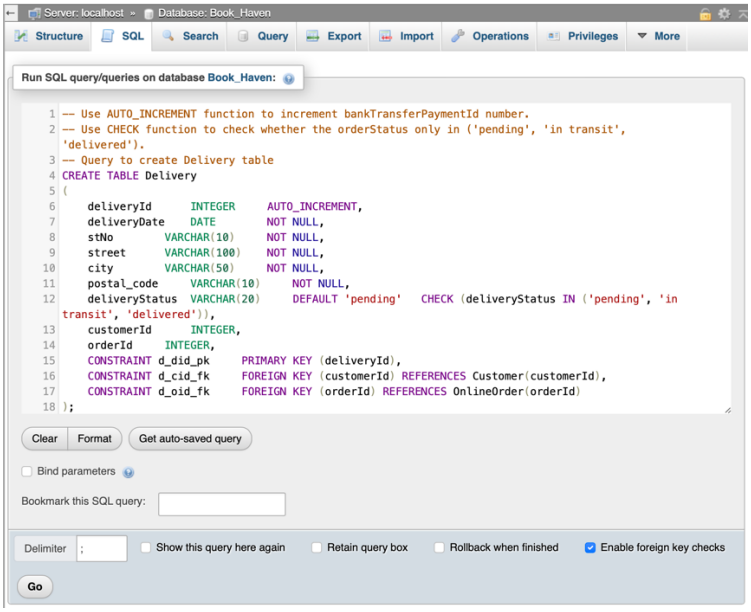


Figure 20 Query to create Delivery table.

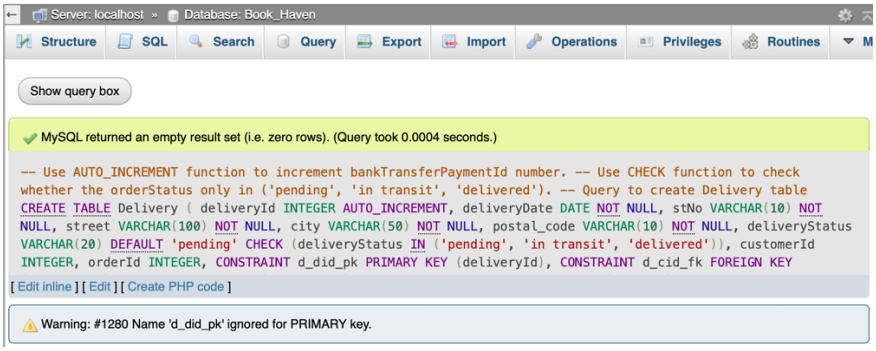


Figure 21 Delivery table created successfully.

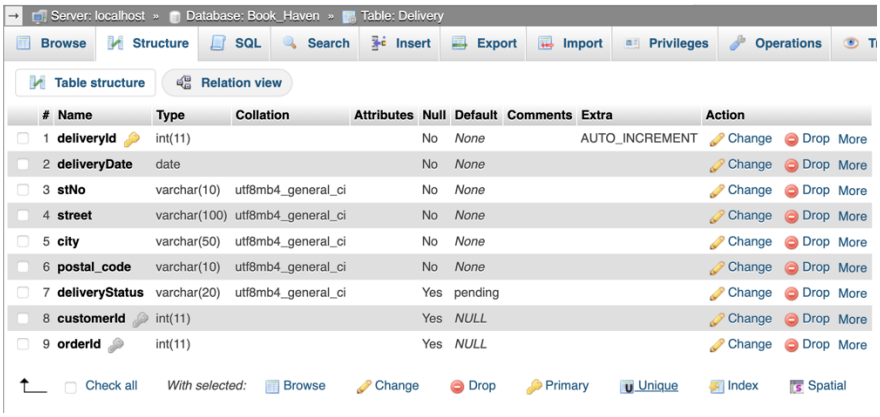


Figure 22 Delivery table Column Specifications.

Supplier Table

- Use AUTO_INCREMENT function to increment supplierId number.
  Use CHECK function to check whether the supplierType only in ('Individual', 'Company').

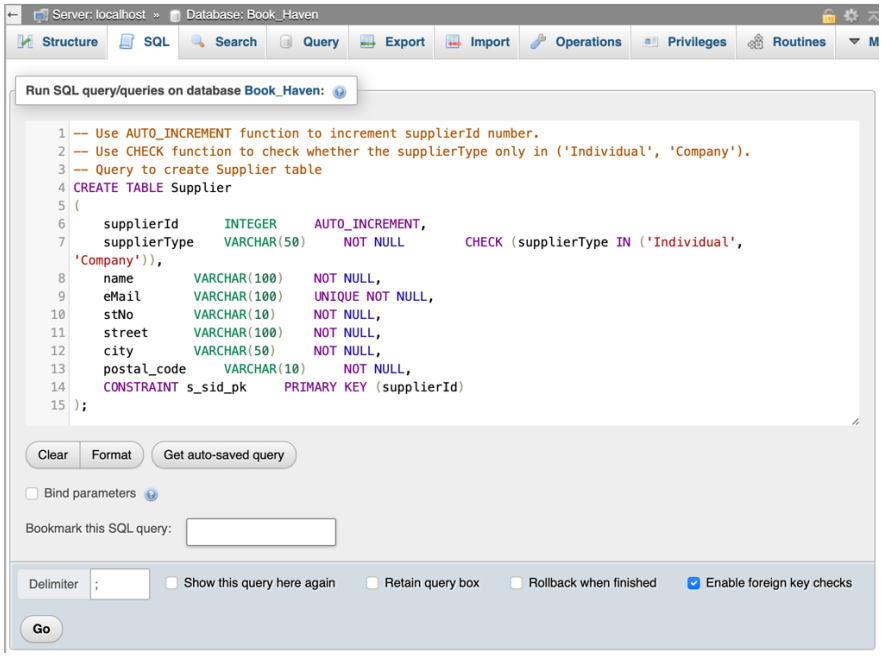


Figure 23 Query to create Supplier table.

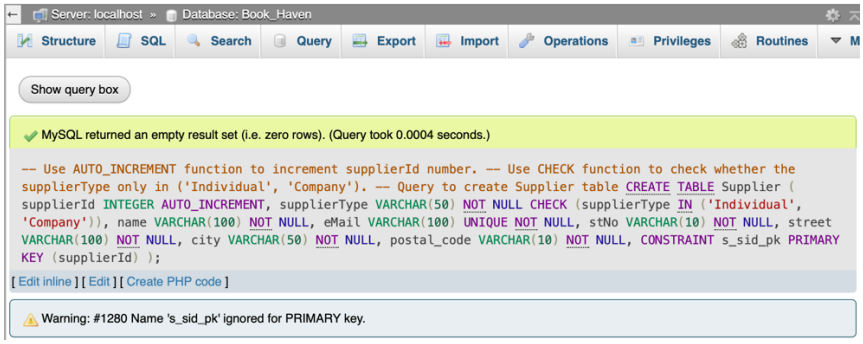


Figure 24 Supplier table created successfully.

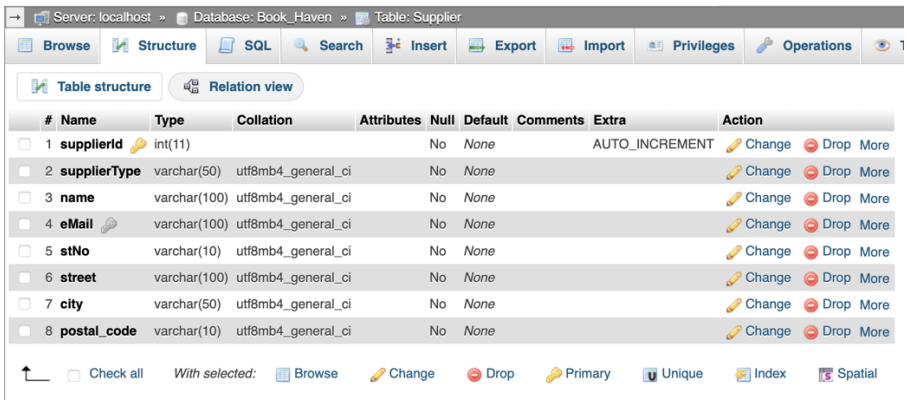


Figure 25 Supplier table Column Specifications.

SupplierContact Table

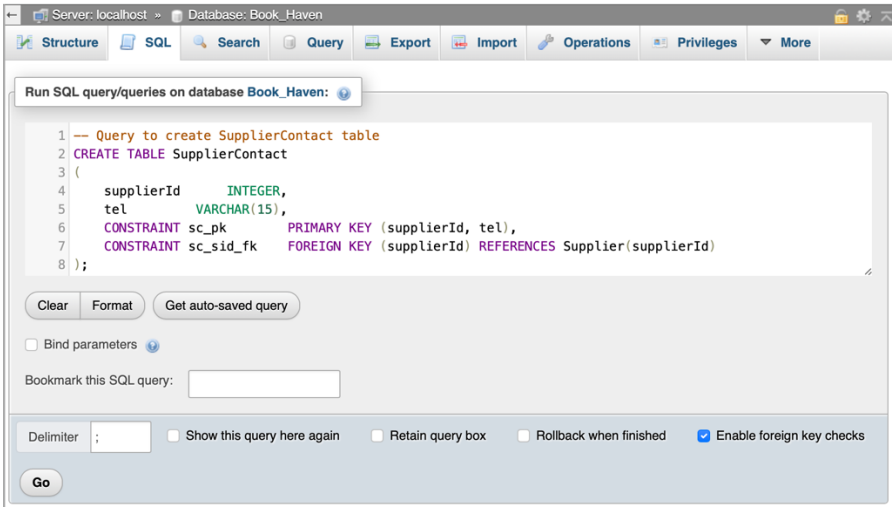


Figure 26 Query to create SupplierContact table.

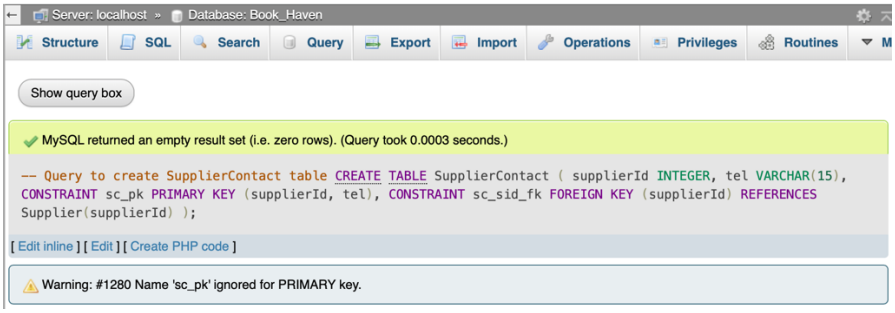


Figure 27 SupplierContact table created successfully.

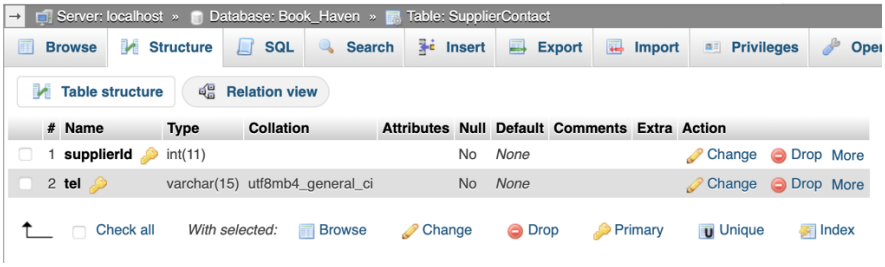


Figure 28 SupplierContact table Column Specifications.

BookItem Table

- Use AUTO_INCREMENT function to increment itemCode number.

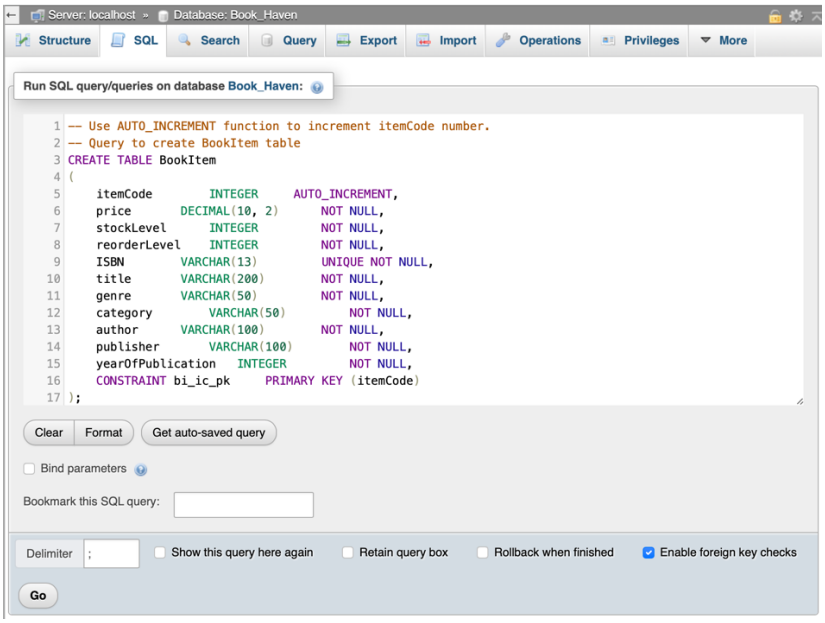


Figure 29 Query to create BookItem table.

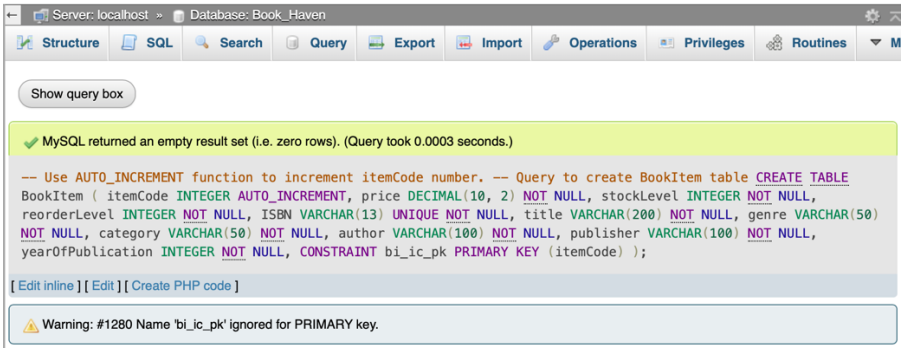


Figure 30 BookItem table created successfully.

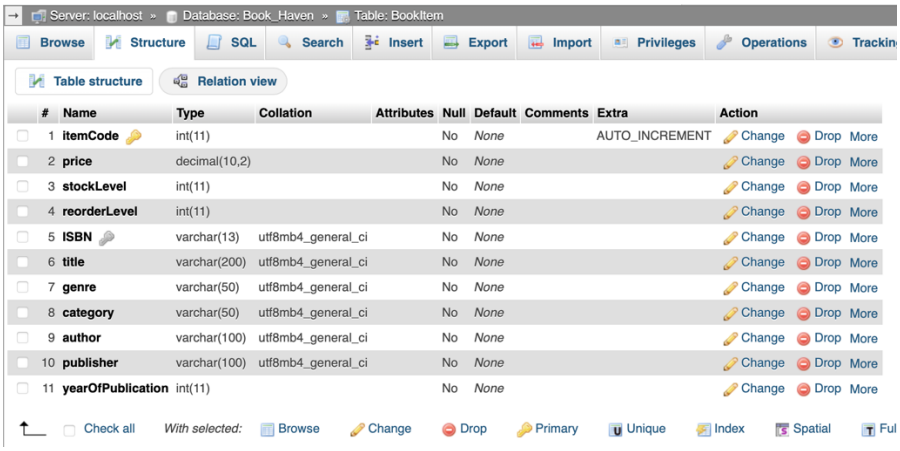


Figure 31 BookItem table Column Specifications.

- This database can only holds up to 999 books because stationery itemCode starts from 1000.

StationeryItem Table

- Use AUTO_INCREMENT function to increment itemCode number starting from 1000.

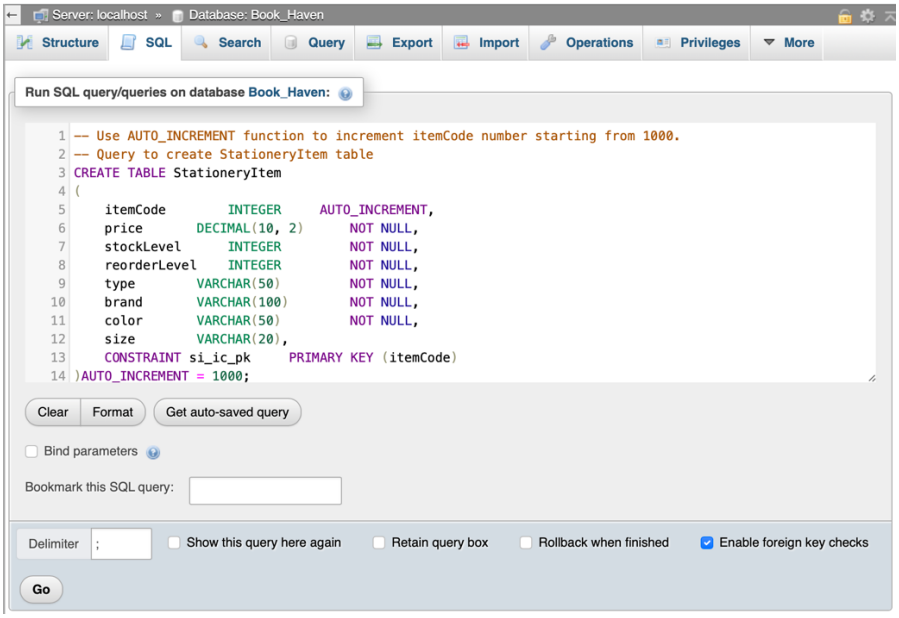


Figure 32 Query to create StationeryItem table.

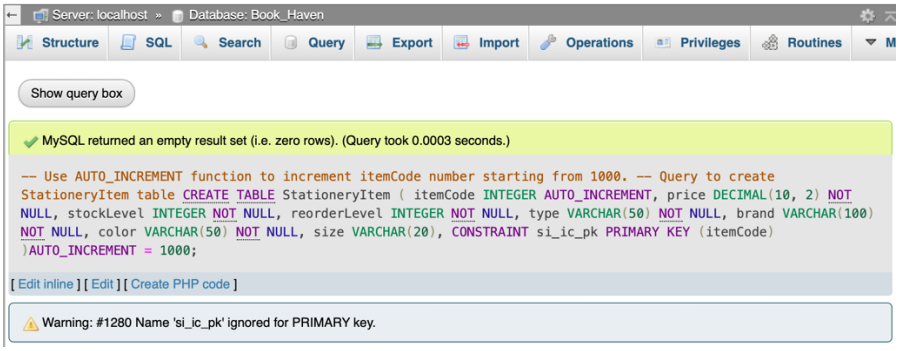


Figure 33 StationeryItem table created successfully.

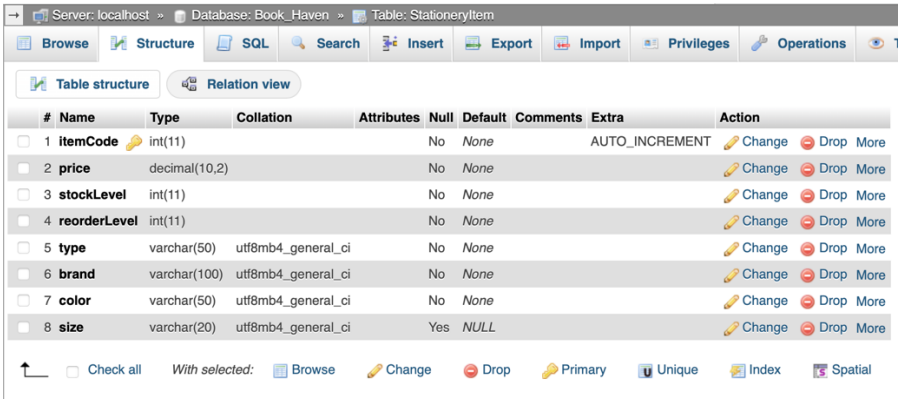


Figure 34 StationeryItem table Column Specifications.

BookItemSupplier Table

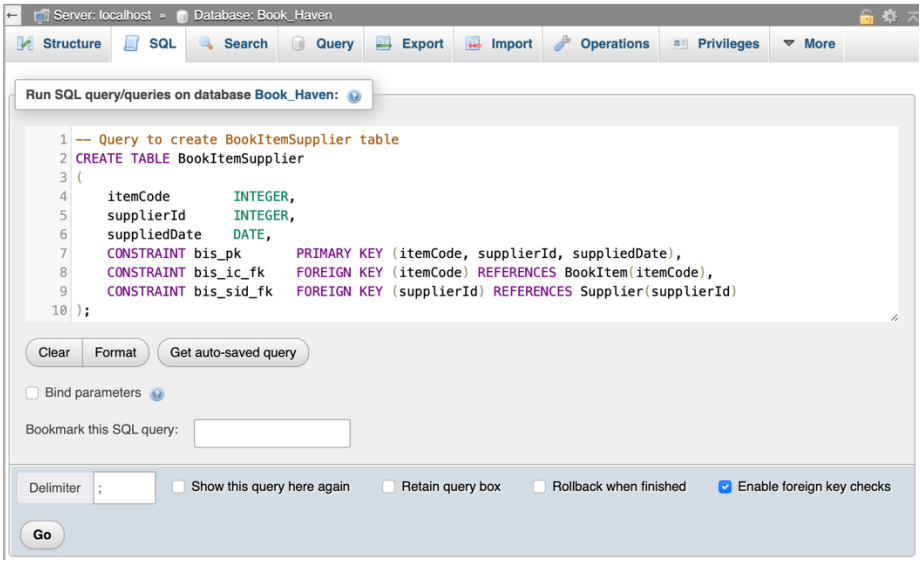


Figure 35 Query to create BookItemSupplier table.

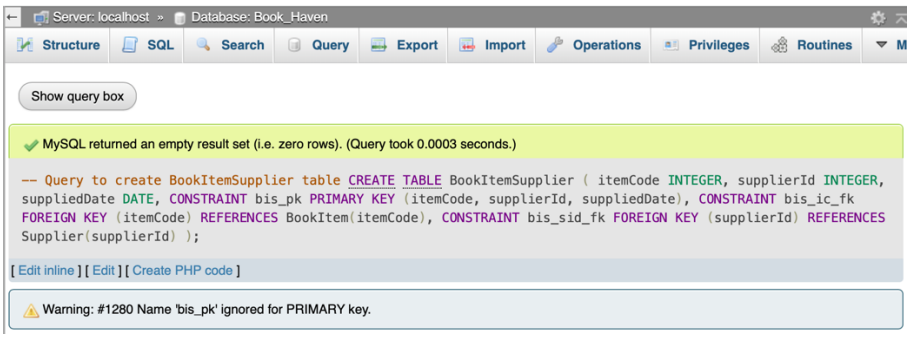


Figure 36 BookItemSupplier table created successfully.

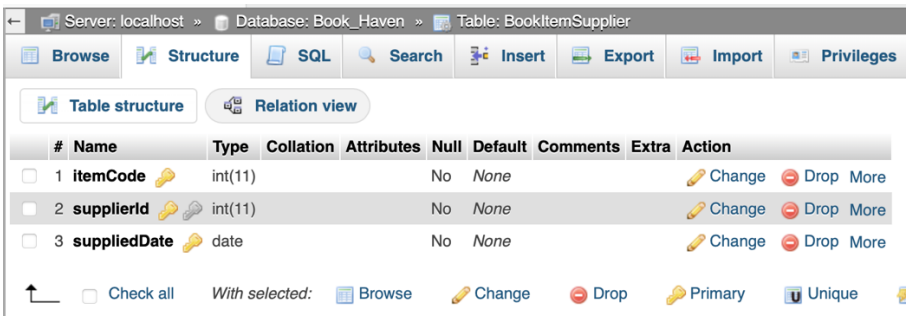


Figure 37 BookItemSupplier table Column Specifications.

StationeryItemSupplier Table

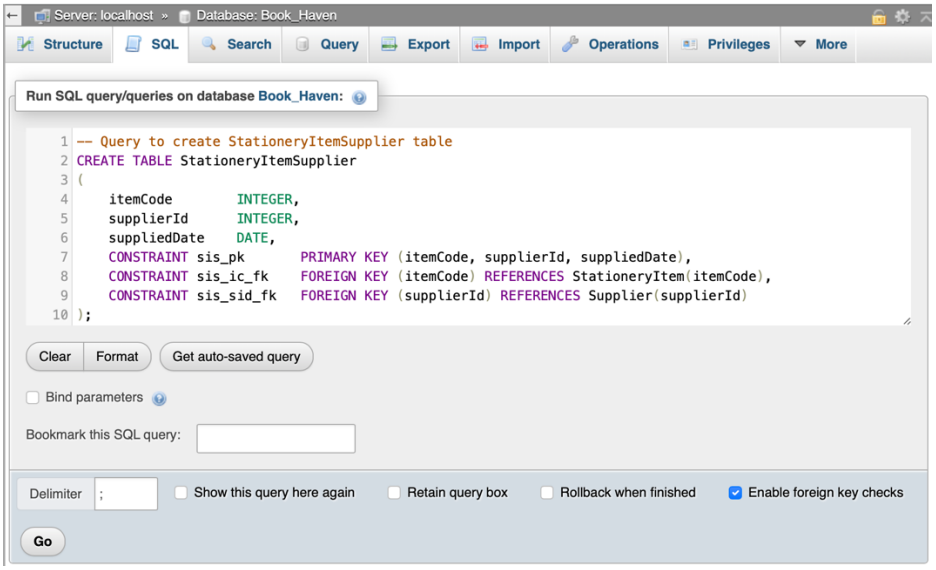


Figure 38 Query to create StationeryItemSupplier table.

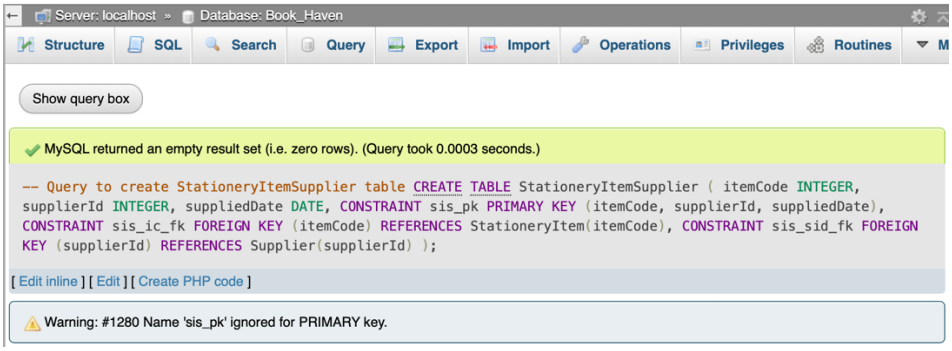


Figure 39 StationeryItemSupplier table created successfully.

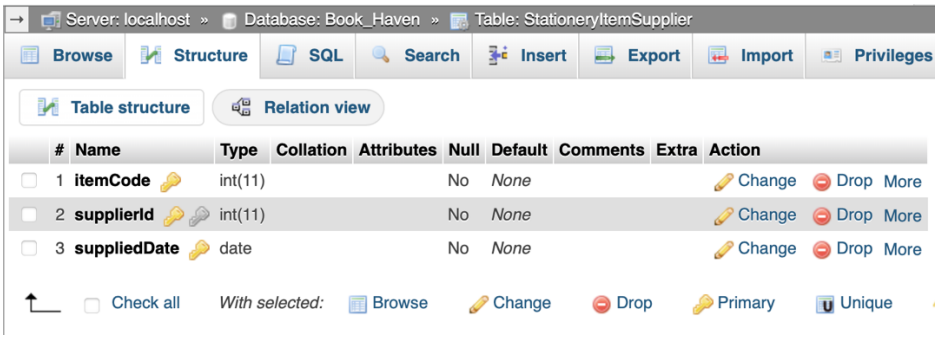


Figure 40 StationeryItemSupplier table Column Specifications.

BookOrderItem Table

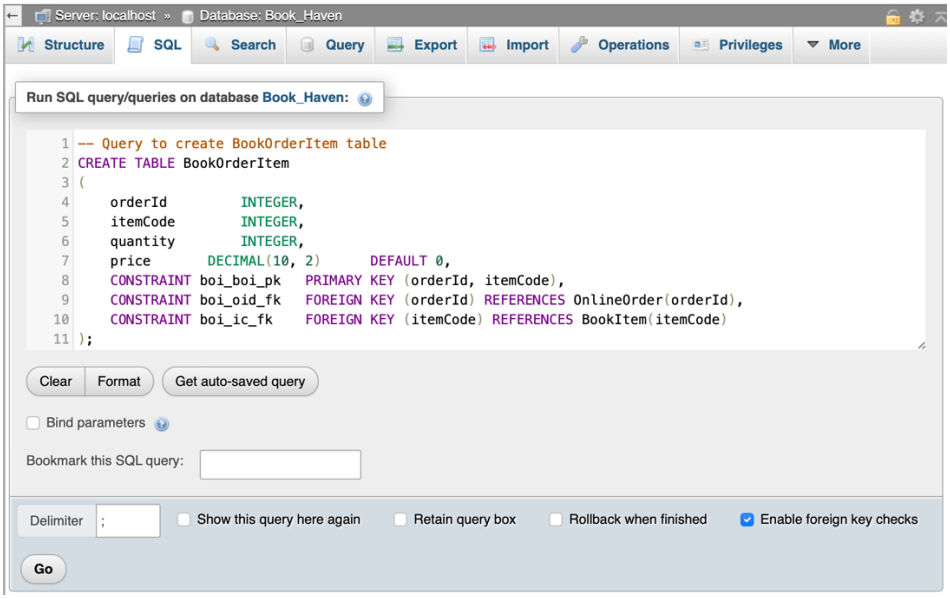


Figure 41 Query to create BookOrderItem table.

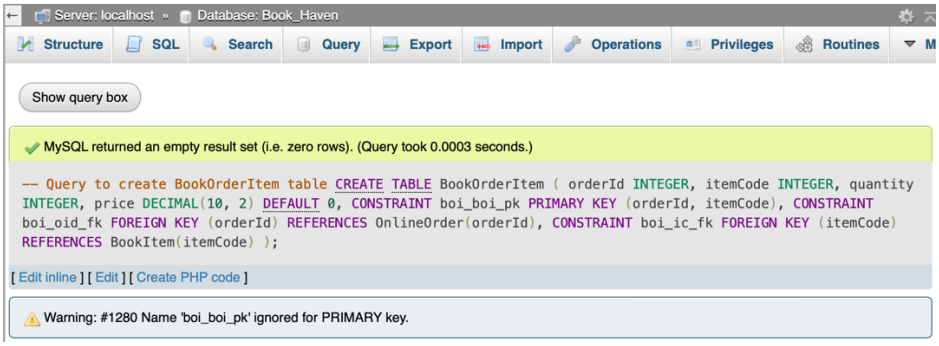


Figure 42 BookOrderItem table created successfully.

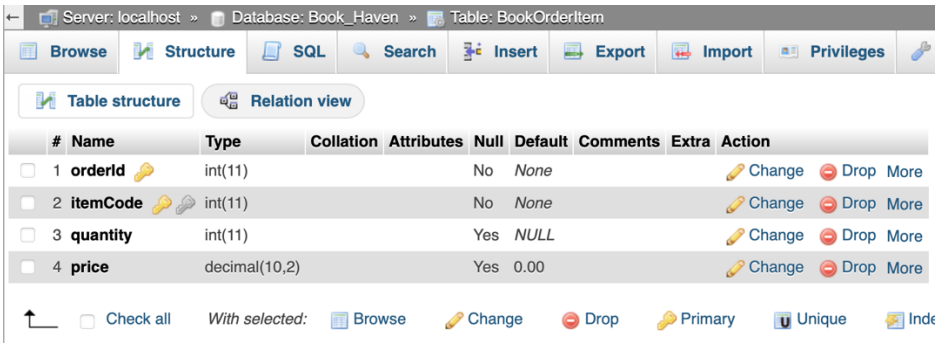


Figure 43 BookOrderItem table Column Specifications.

StationeryOrderItem Table

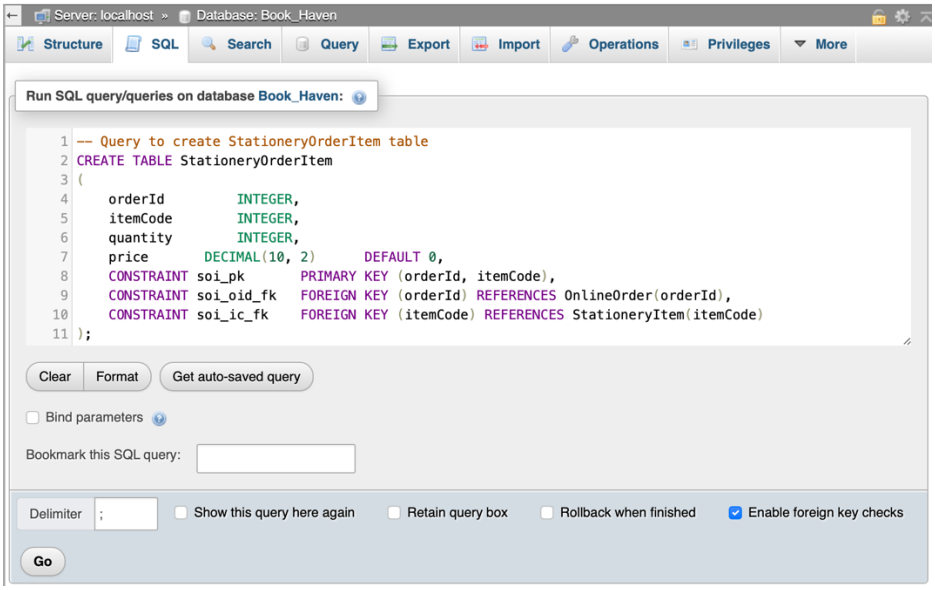


Figure 44 Query to create StationeryOrderItem table.

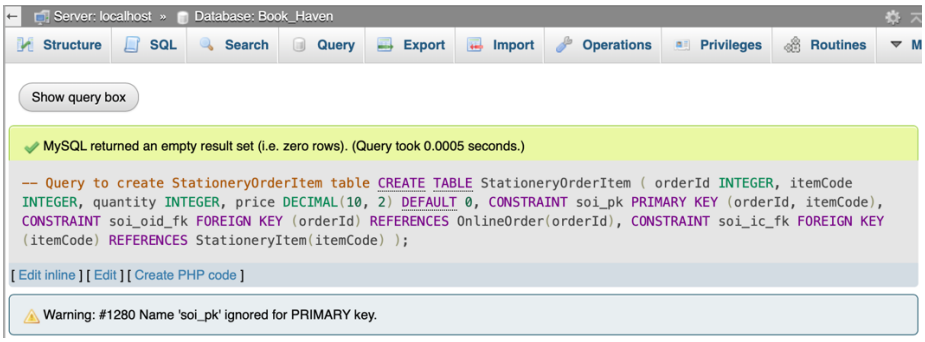


Figure 45 StationeryOrderItem table created successfully.

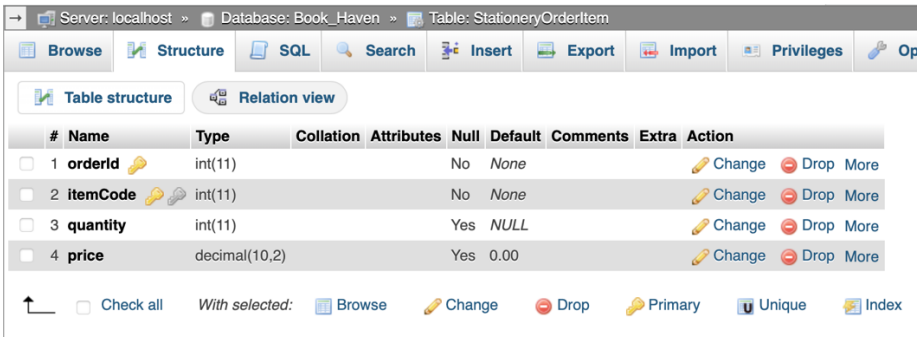


Figure 46 StationeryOrderItem table Column Specifications.

Inserting data to each table

A crucial component in the 'OnlineOrder' table named 'orderId' keeps this database arranged nicely. It links to other tables such as OnlinePayment, BankTransferPayment, BookOrderItem, StationeryOrderItem and Delivery in addition to serving as the primary identifier. As a result, information in 'OnlineOrder' must come before any linked tables. Because it is foreign key for OnlinePayment, BankTransferPayment, BookOrderItem, StationeryOrderItem and Delivery and foreign keys can't be null

Data entry proceeds in an easily understood way: begin with the Supplier→SupplierContact, BookItem, StationeryItem,and Customer→CustomerContact tables, then fill the intermediate BookItemSupplier and StationeryItemSupplier, then proceed through OnlineOrder → BookOrderItem or StationeryOrderItem→ OnlinePayment or BankTransferPayment→ Delivery.

Triggers automate certain activities to improve efficiency. The price in BookOrderItem and StationeryOrderItem is calculated by multiplying the quantity by the price relevant to the itemCode. Aditionally calculating the sum of the price in either BookOrderItem or StationeryOrderItem relevant to a single orderId to update the totalPrice in the OnlineOrder Table.

In summary, triggers automate crucial procedures for an efficient database structure, and this database is thoroughly structured to ensure that data is entered in an organized manner.

Customer Table

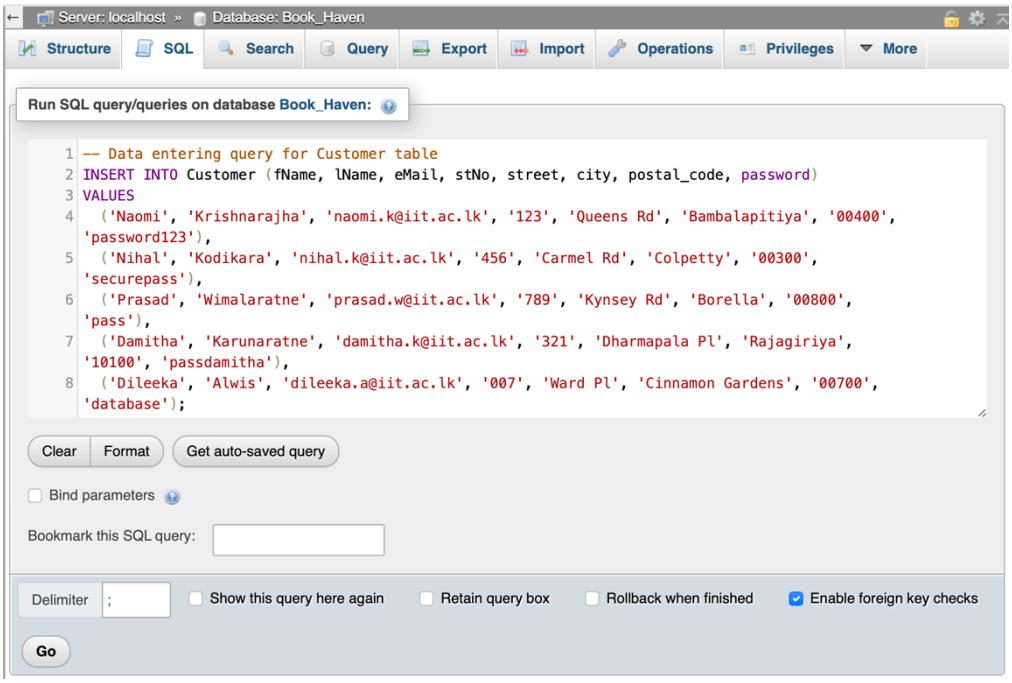


Figure 47 Data entering query for Customer table.

Figure 48 Data entered successfully to Customer table.

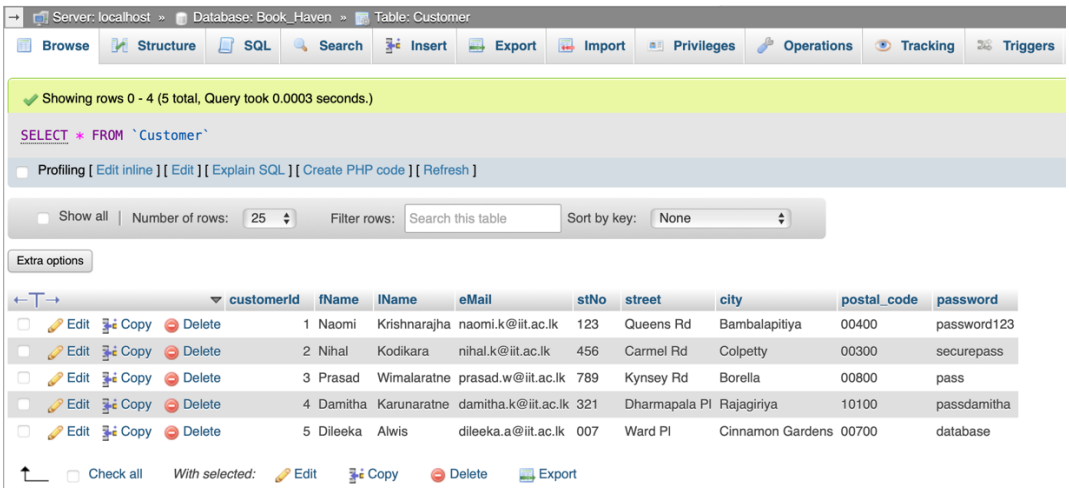


Figure 49 Data entered Customer table.

CustomerContact Table

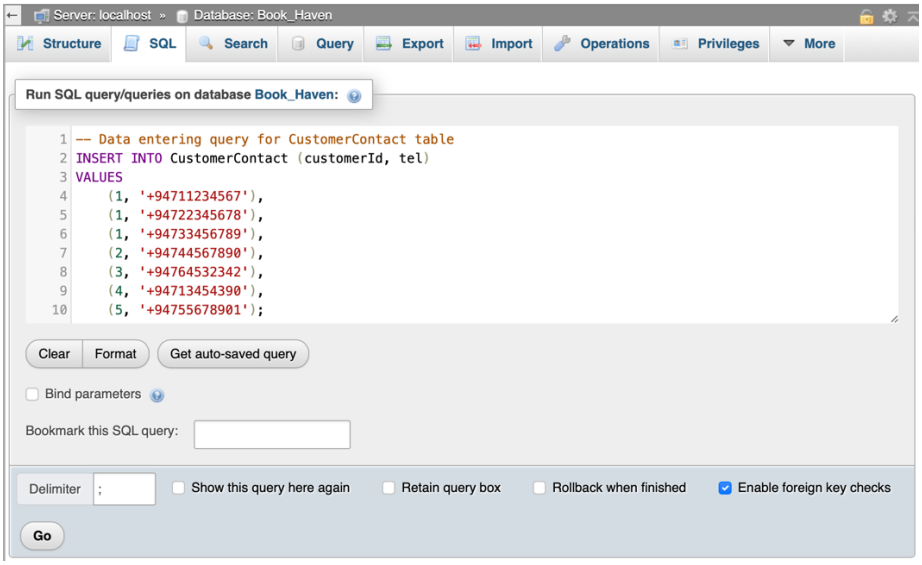


Figure 50 Data entering query for CustomerContact table.

Figure 51 Data entered successfully to CustomerContact table.

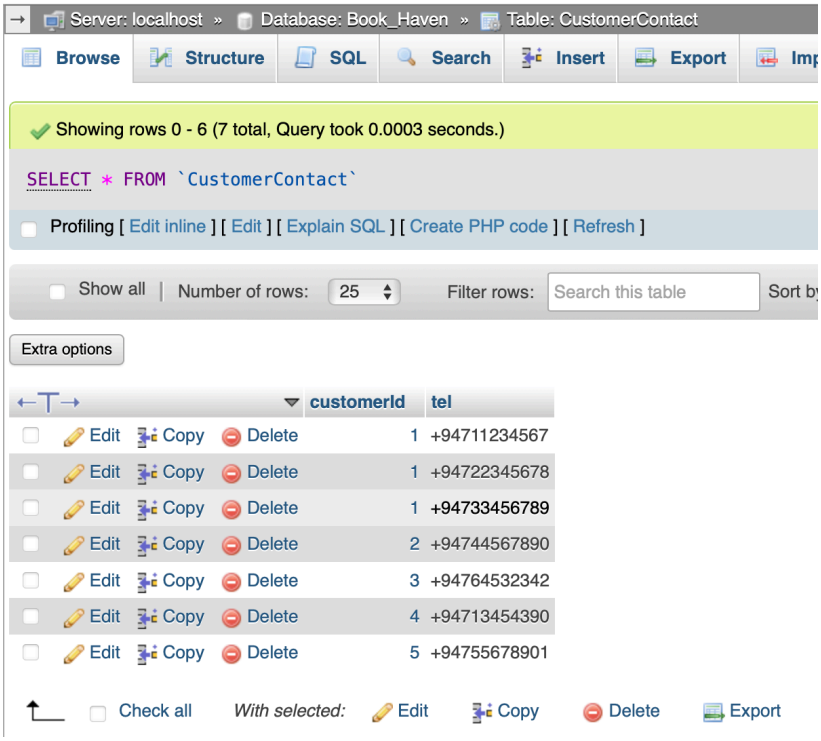


Figure 52 Data entered CustomerContact table.

BookItem Table

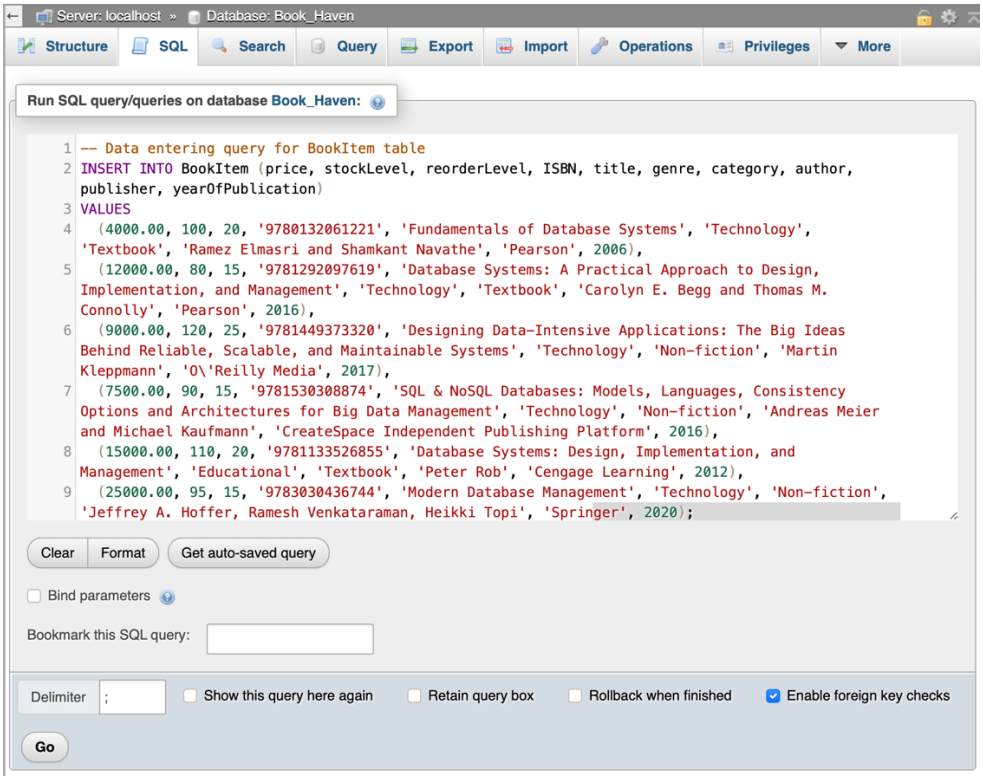


Figure 53 Data entering query for BookItem table.

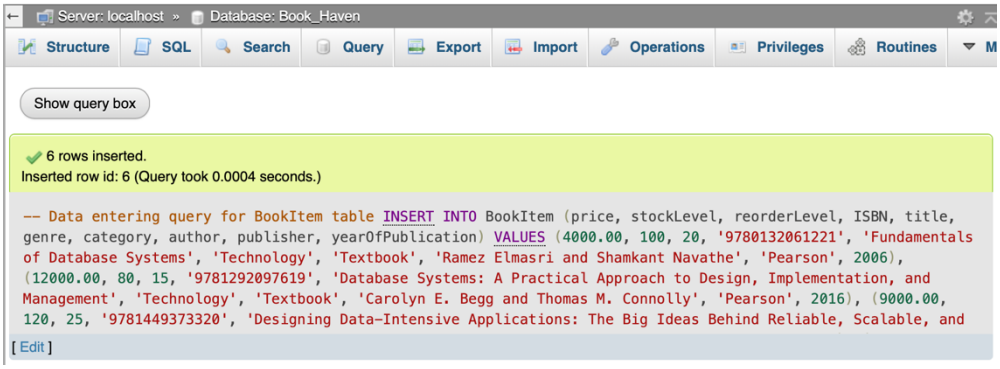


Figure 54 Data entered successfully to BookItem table.

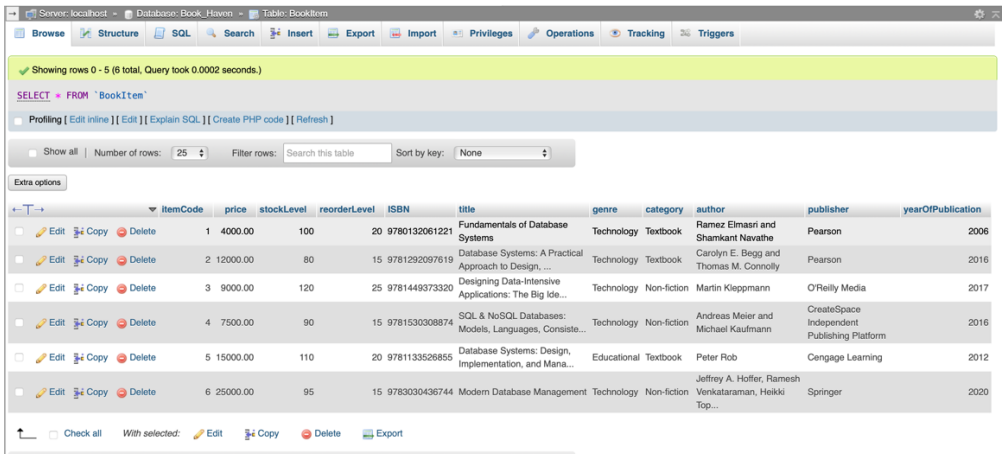


Figure 55 Data entered BookItem tabl.

StationeryItem Table

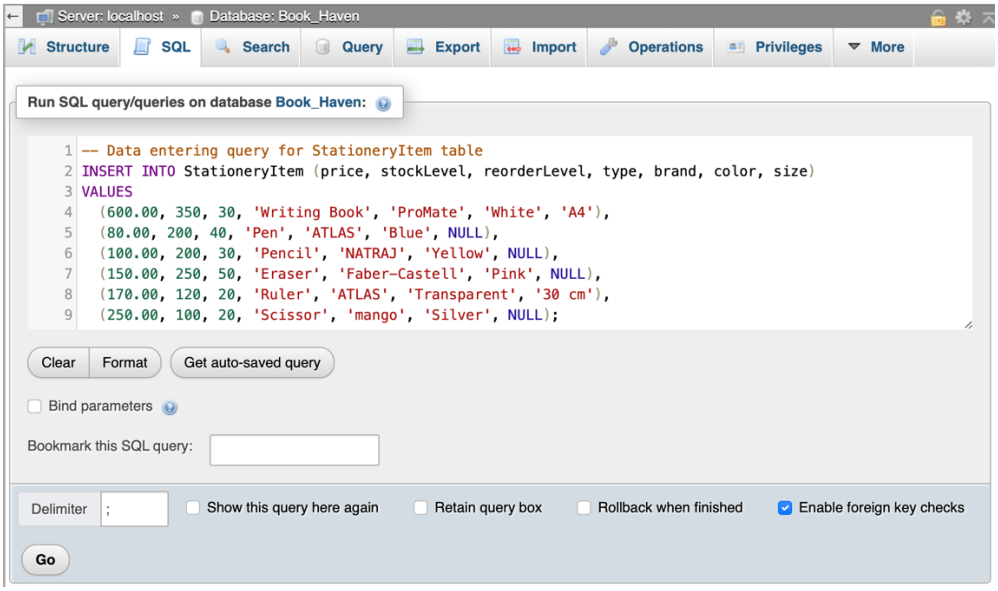


Figure 56 Data entering query for StationeryItem table.

Figure 57 Data entered successfully to StationeryItem table.

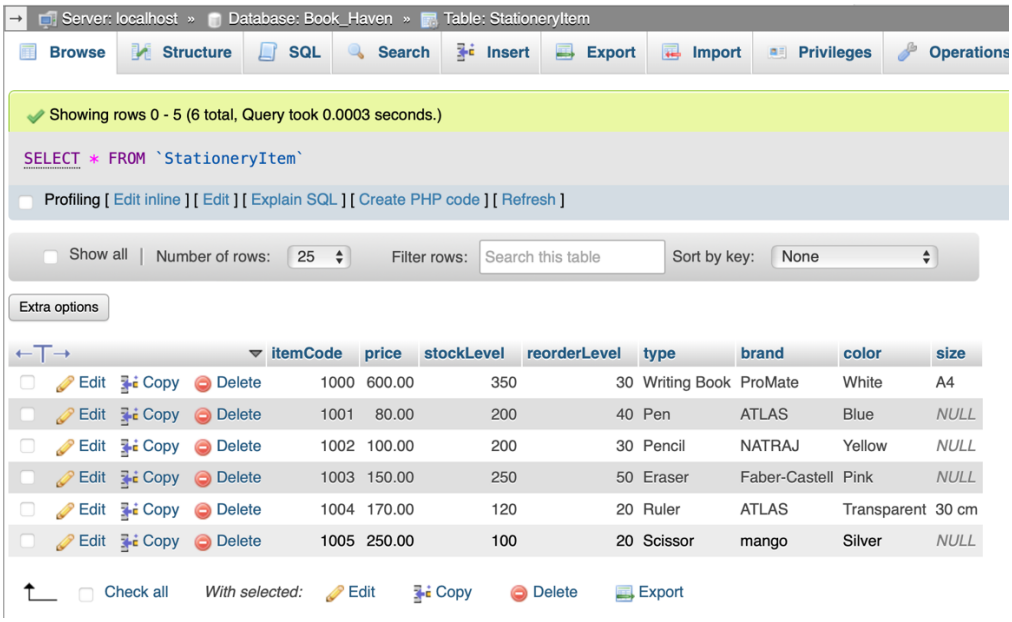


Figure 58 Data entered StationeryItem table.

Supplier Table

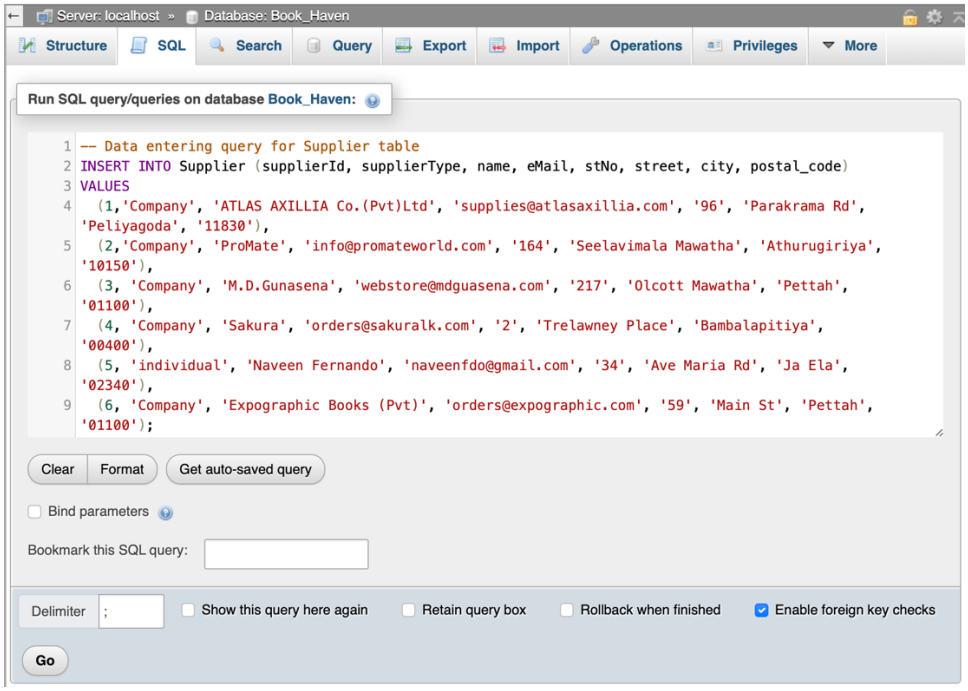


Figure 59 Data entering query for Supplier table.

Figure 60 Data entered successfully to Supplier table.

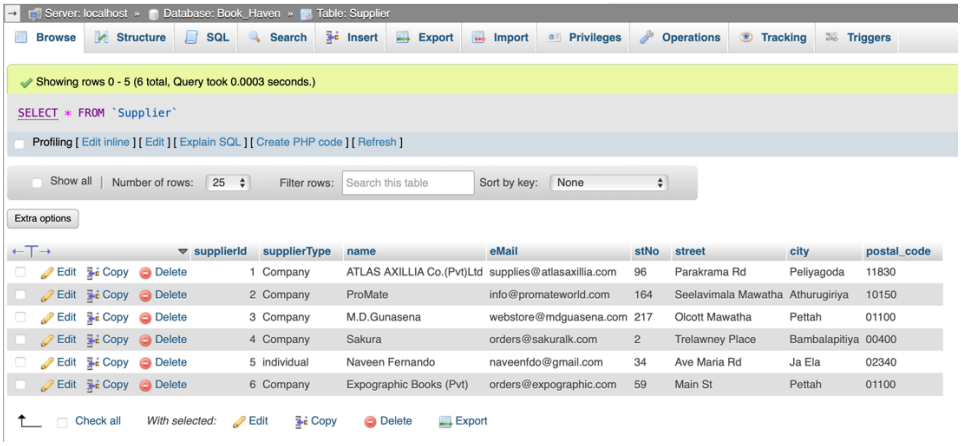


Figure 61 Data entered Supplier table.

SupplierContact Table

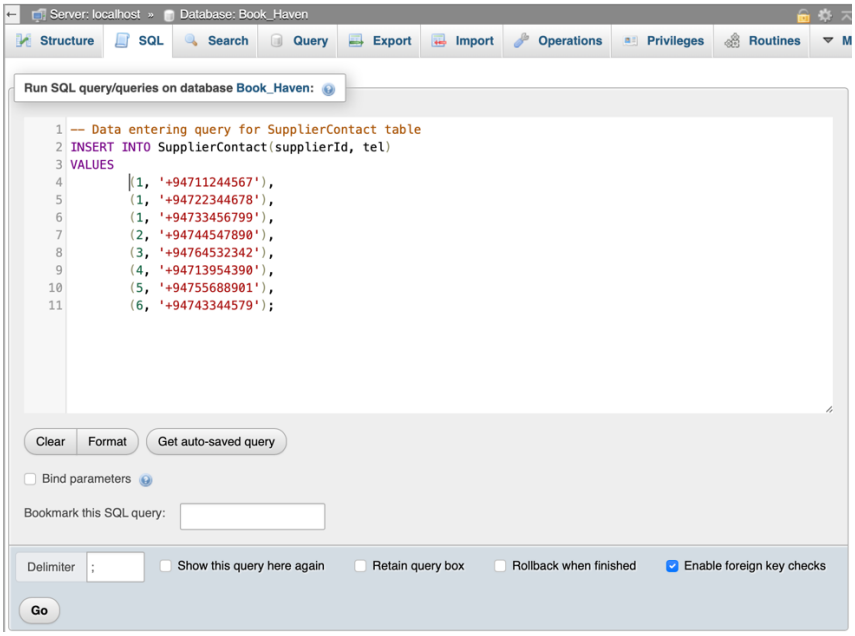


Figure 62 Data entering query for SupplierContact table.

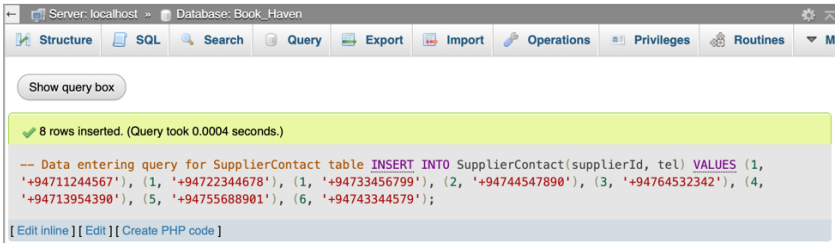


Figure 63 Data entered successfully to SupplierContact table.

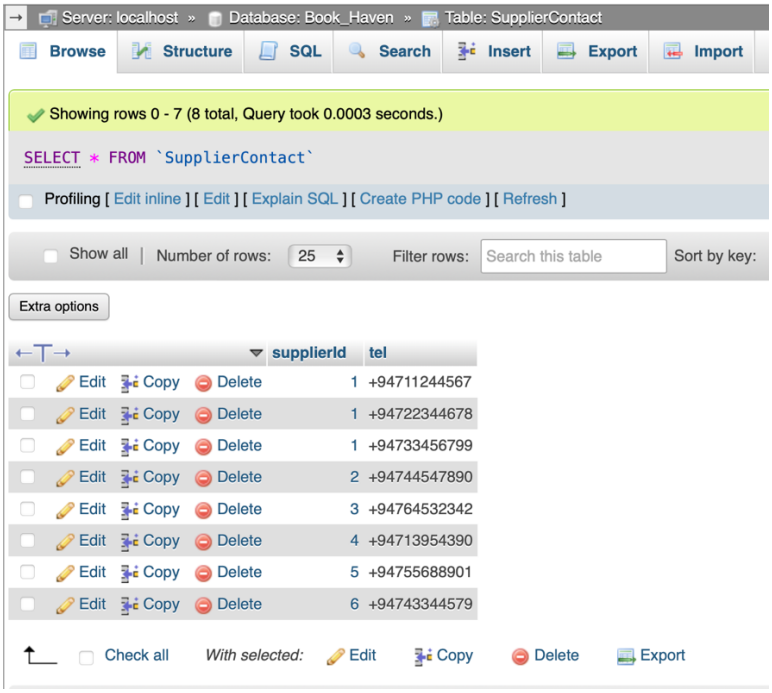


Figure 64 Data entered SupplierContact table.

BookItemSupplier Table

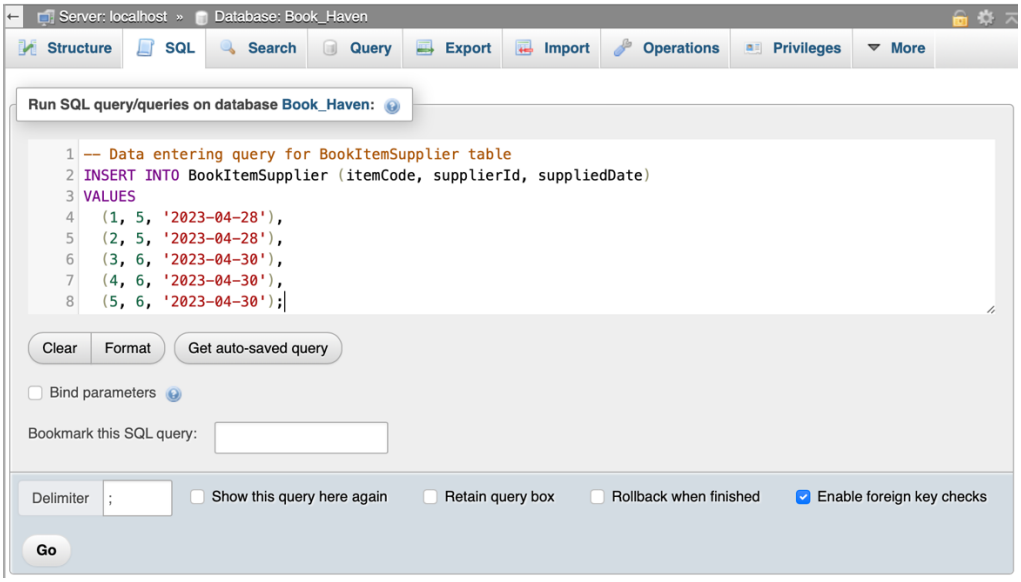


Figure 65 Data entering query for BookItemSupplier table.

Figure 66 Data entered successfully to BookItemSupplier table.

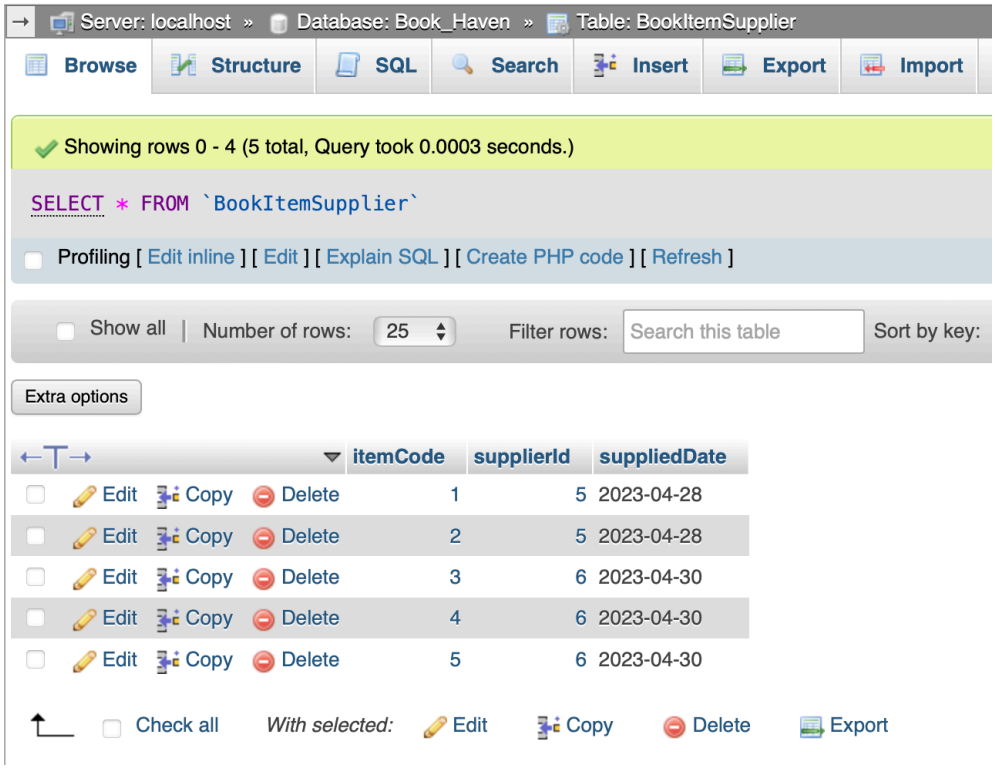


Figure 67 Data entered BookItemSupplier table.

StationeryItemSupplier Table

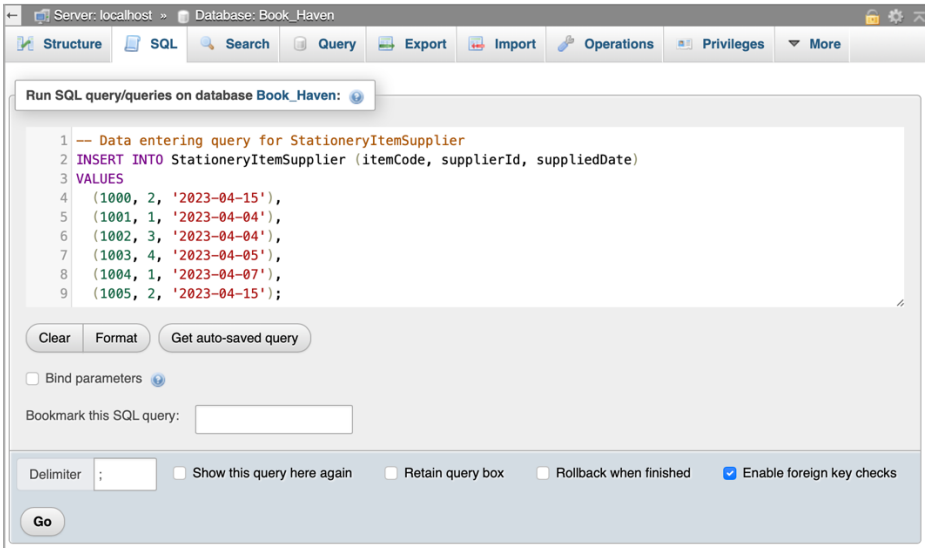


Figure 68 Data entering query for StationeryItemSupplier table.

Figure 69 Data entered successfully to StationeryItemSupplier table.d

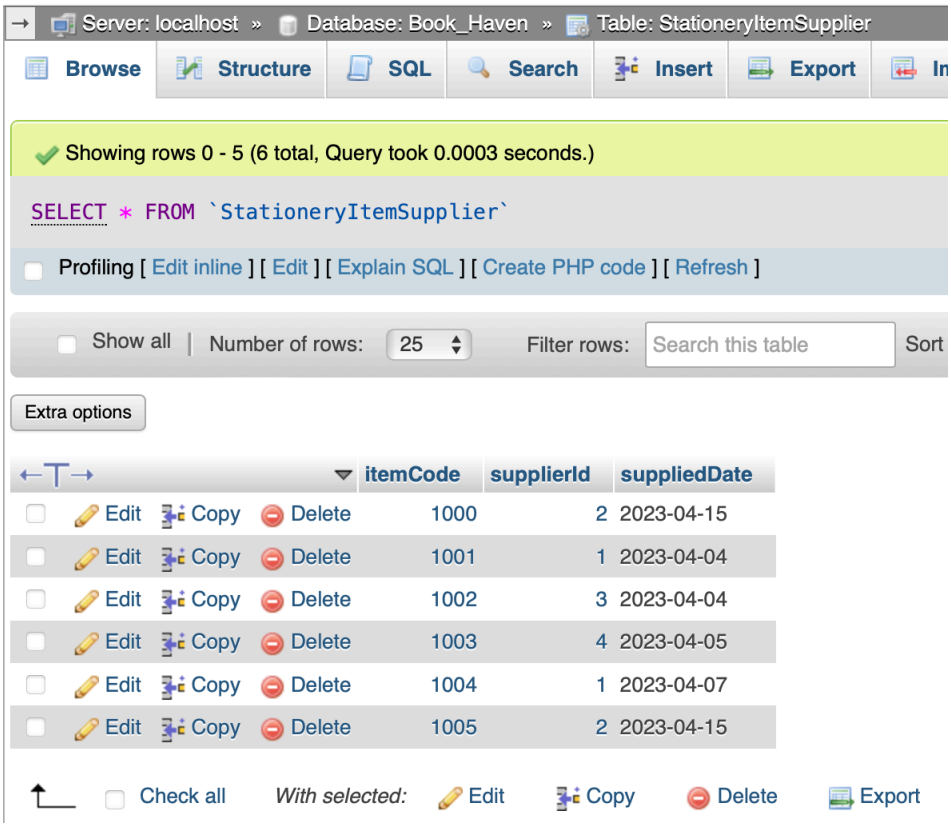
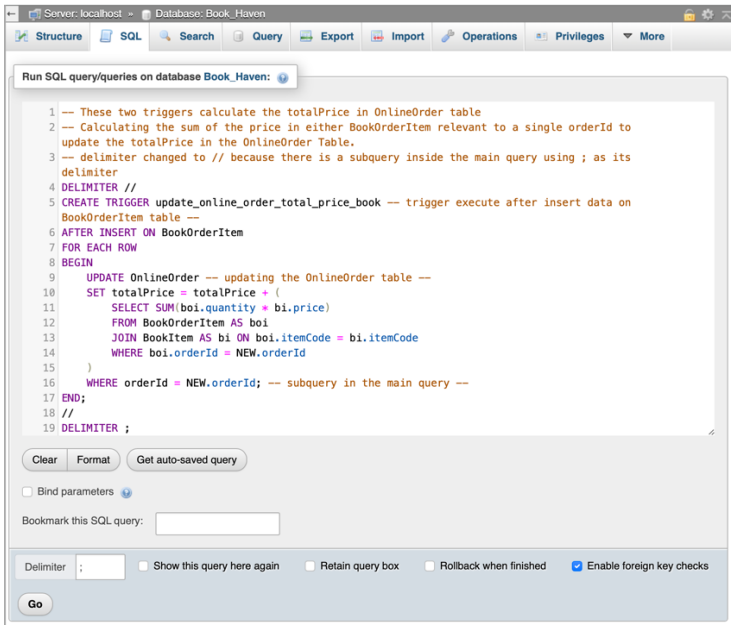


Figure 70 Data entered StationeryItemSupplier table.

Trigger 1

These two triggers calculate the totalPrice in OnlineOrder table.



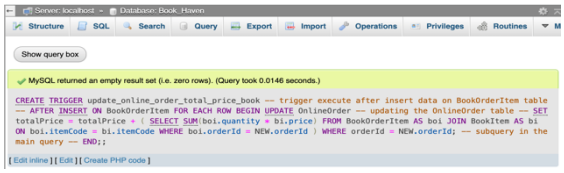


Figure 71 Trigger that update the totalPrice in OnlineOrder table(book)

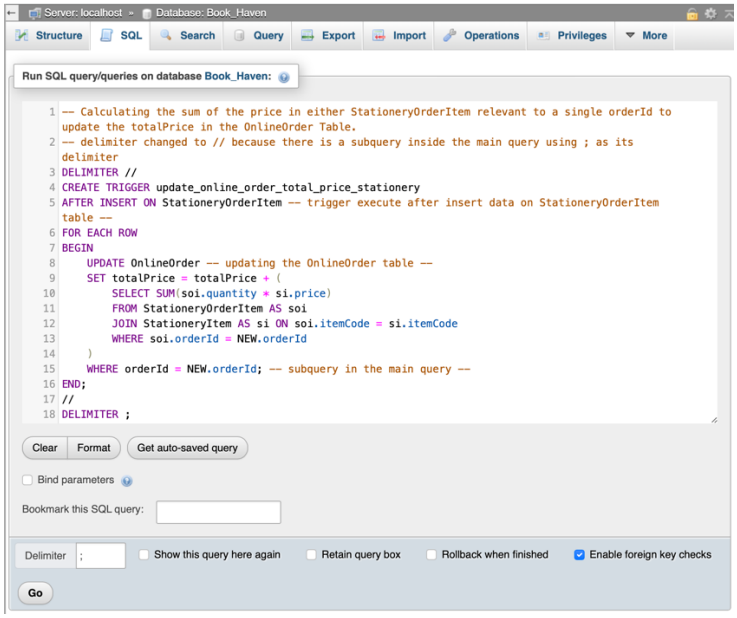


Figure 72 Trigger that update the totalPrice in OnlineOrder table(stationery)

OnlineOrder Table

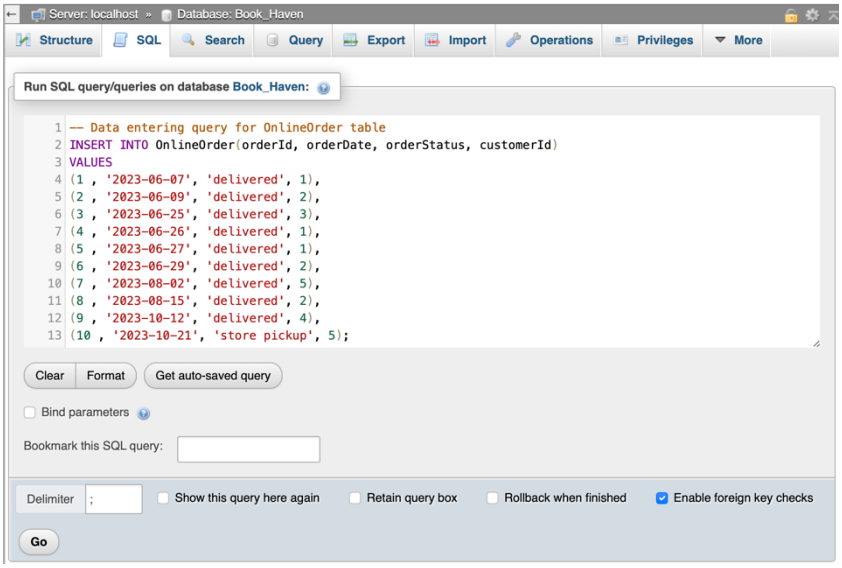


Figure 73 Data entering query for OnlineOrder table.

Figure 74 Data entered successfully to OnlineOrder table.

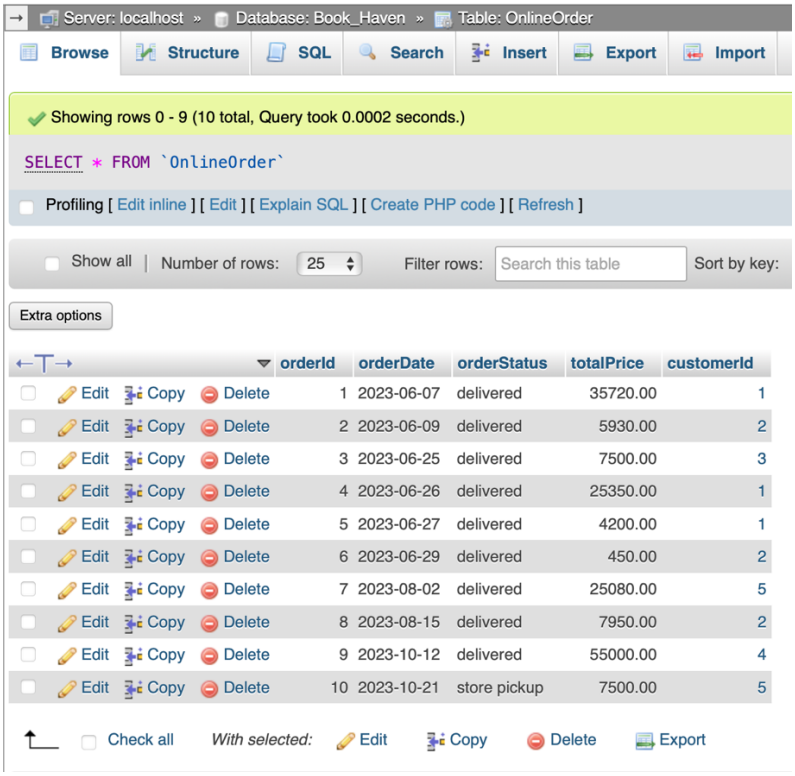
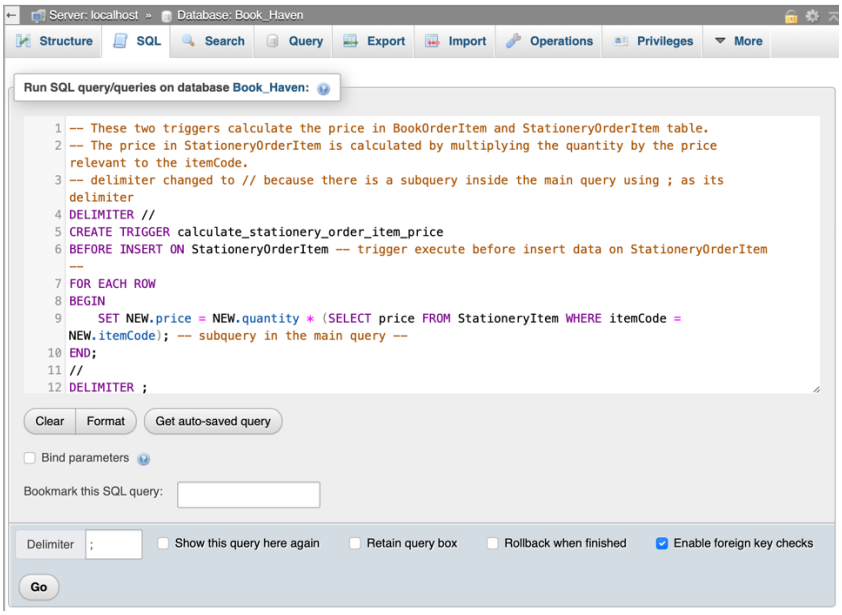


Figure 75 Data entered OnlineOrder table.

Trigger 2

These two triggers calculate the price in BookOrderItem and StationeryOrderItem table.



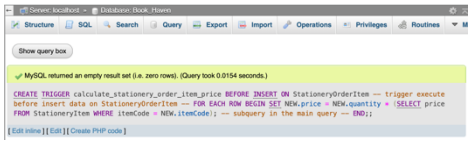
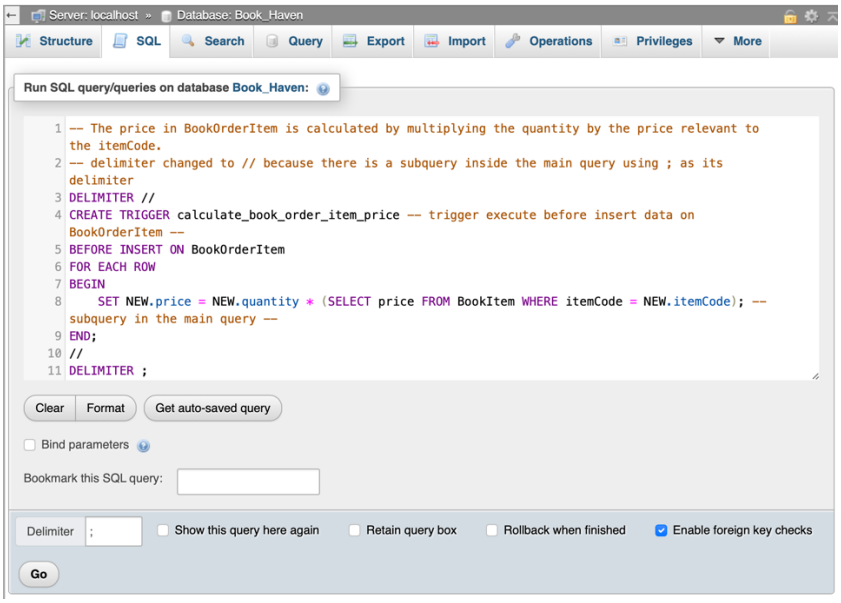


Figure 76 Trigger that calculate the price in StationeryOrderItem table.



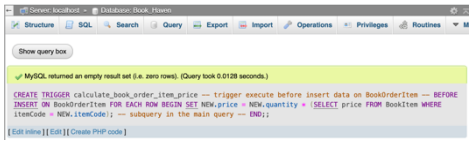


Figure 77 Trigger that calculate the price in BookOrderItem table.

BookOrderItem Table

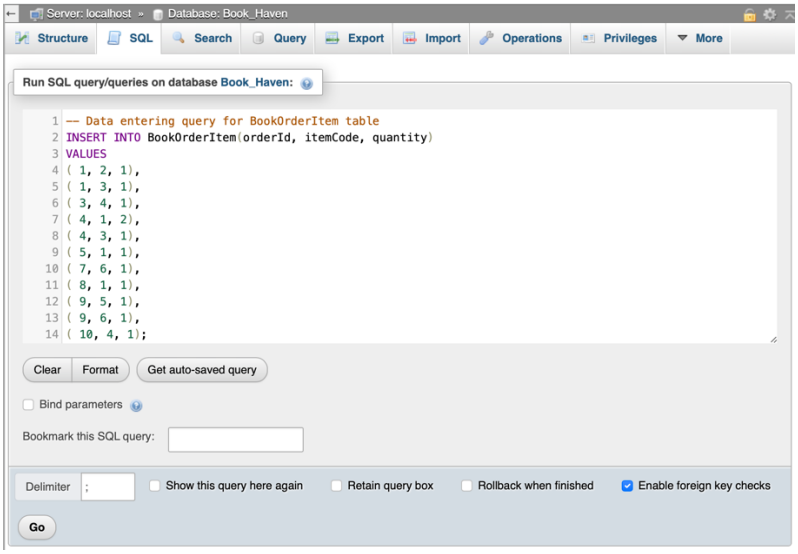


Figure 78 Data entering query for BookOrderItem table.

Figure 79 Data entered successfully to BookOrderItem table.

`

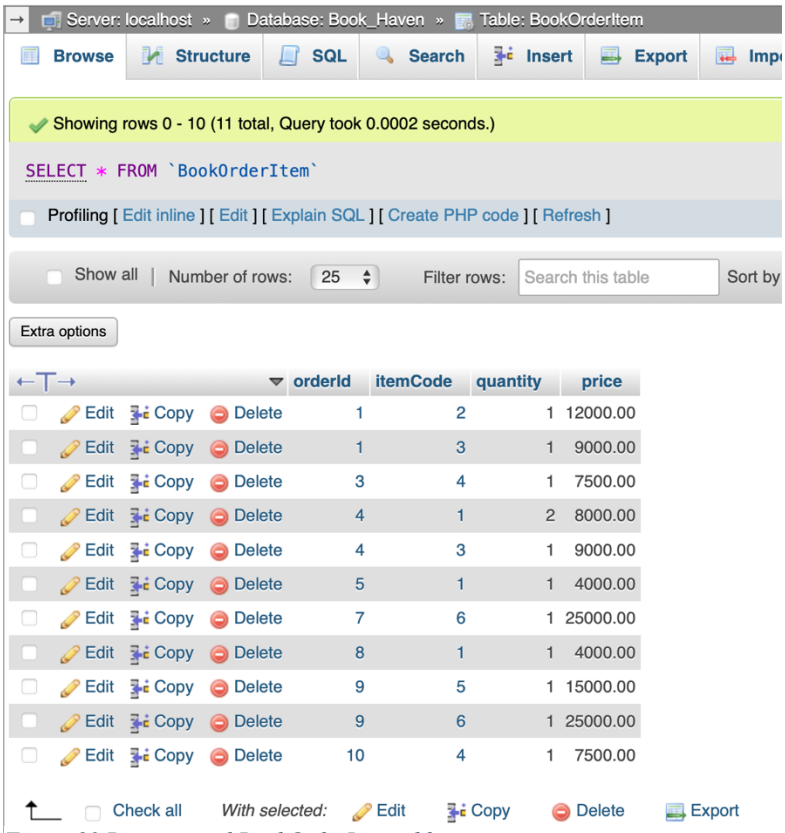


Figure 80 Data entered BookOrderItem table.

StationeryOrderItem Table

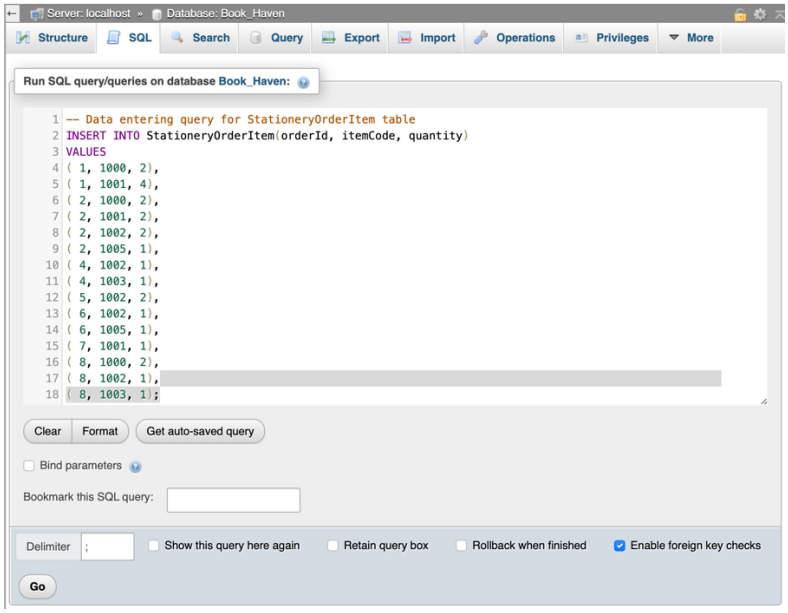


Figure 81 Data entering query for StationeryOrderItem table.

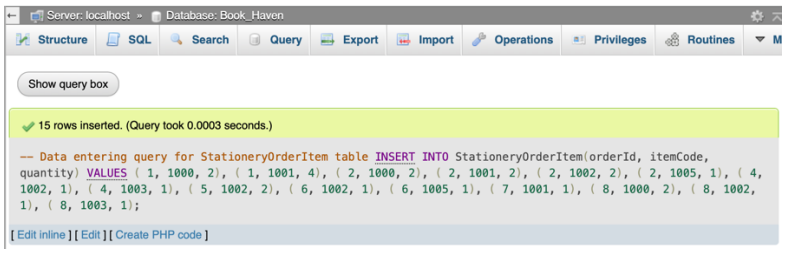


Figure 82 Data entered successfully to StationeryOrderItem table.

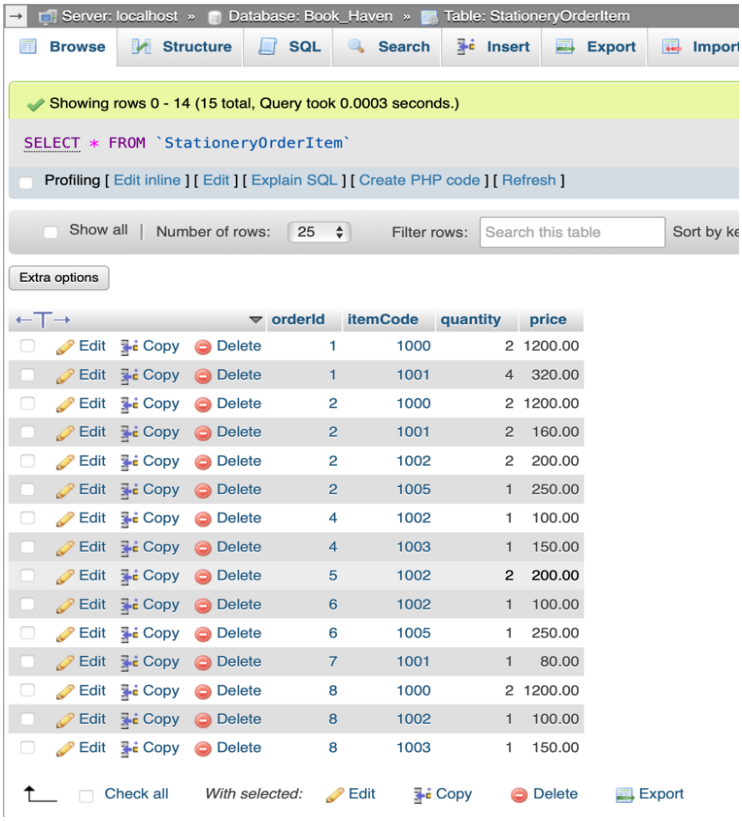


Figure 83Data entered StationeryOrderItem table.

OnlinePayment Table

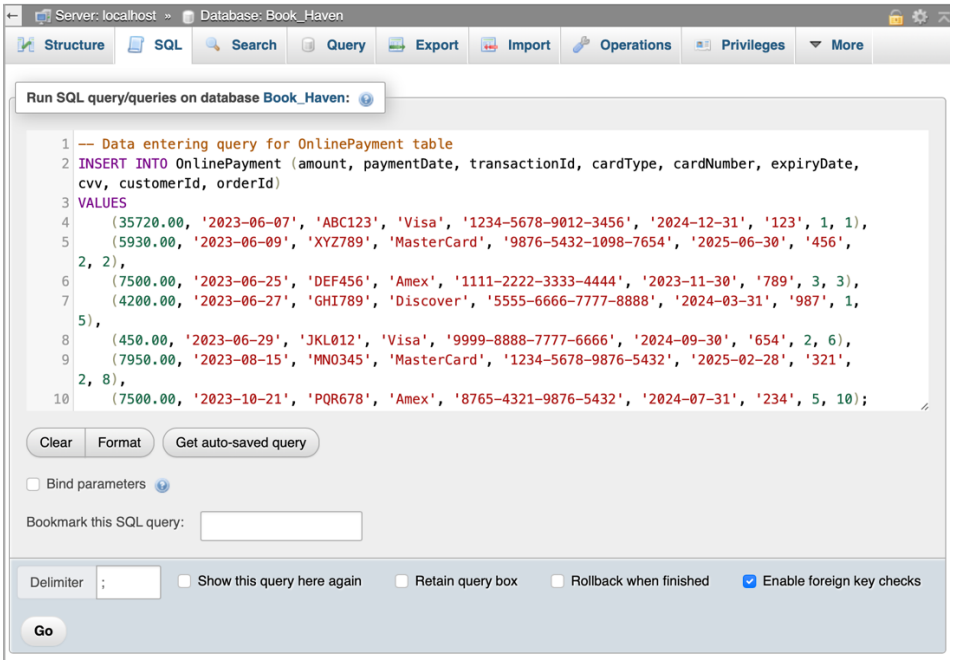


Figure 84 Data entering query for OnlinePayment table.

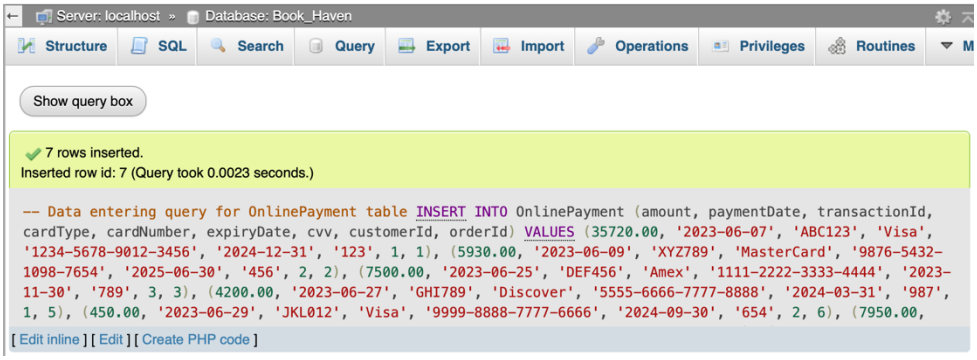


Figure 85 Data entered successfully to OnlinePayment table.

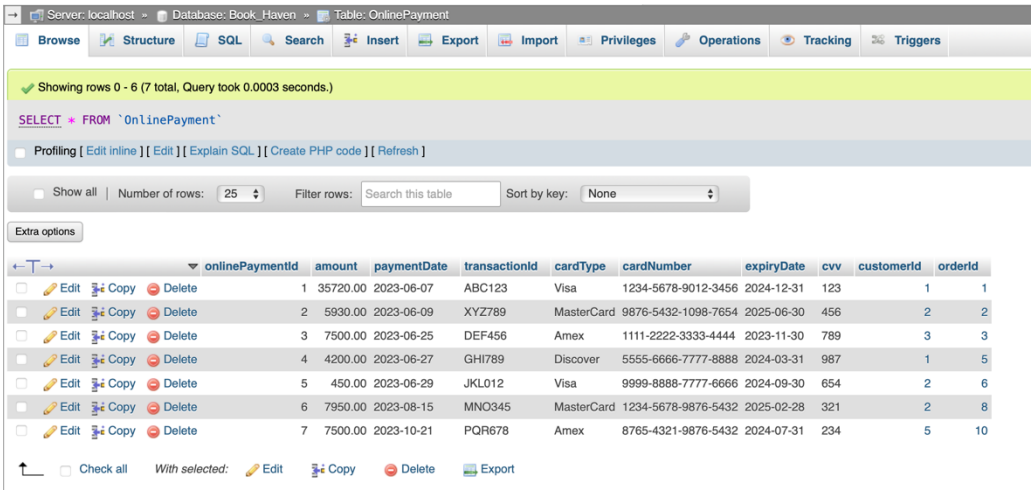


Figure 86 Data entered OnlinePayment table.

BankTransferPayment Table

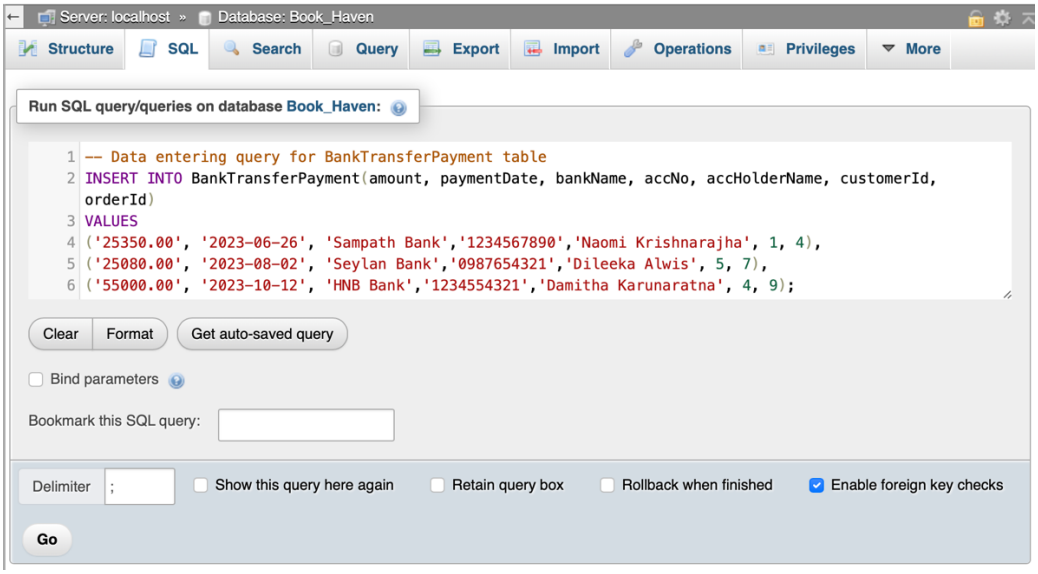


Figure 87 Data entering query for BankTransferPayment table.

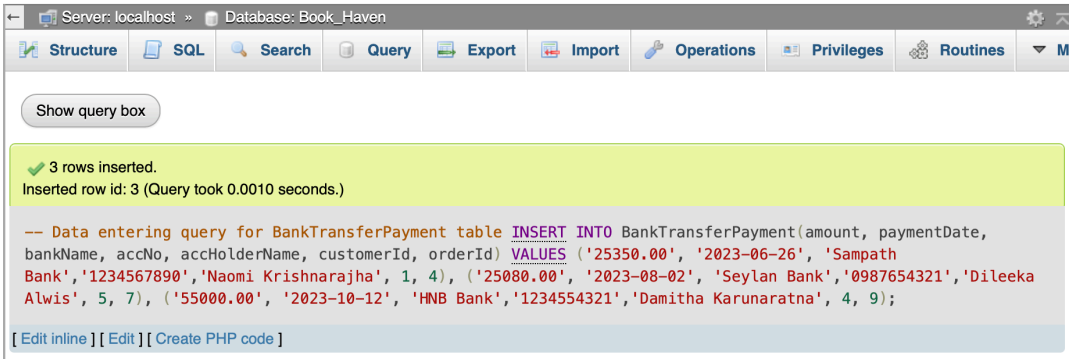


Figure 88 Data entered successfully to BankTransferPayment table.

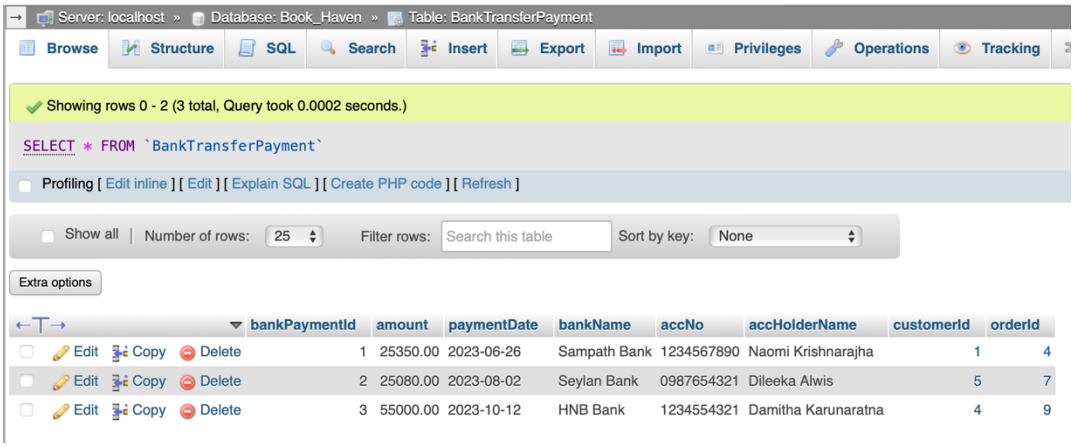


Figure 89 Data entered BankTransferPayment table.

Delivery Table

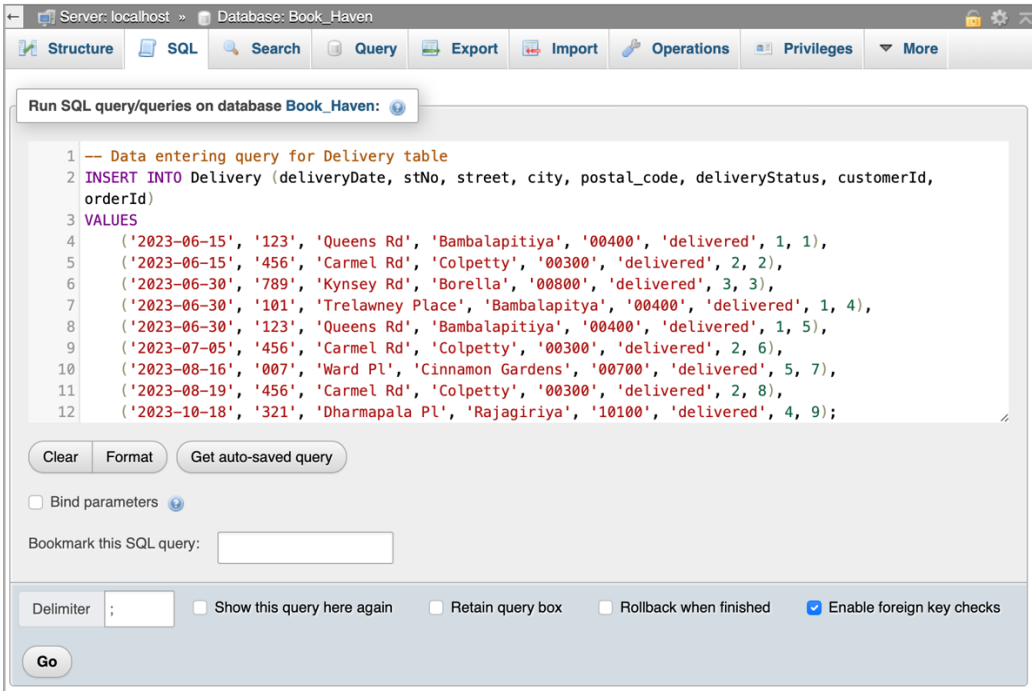


Figure 90 Data entering query for Delivery table.

Figure 91 Data entered successfully to Delivery table.

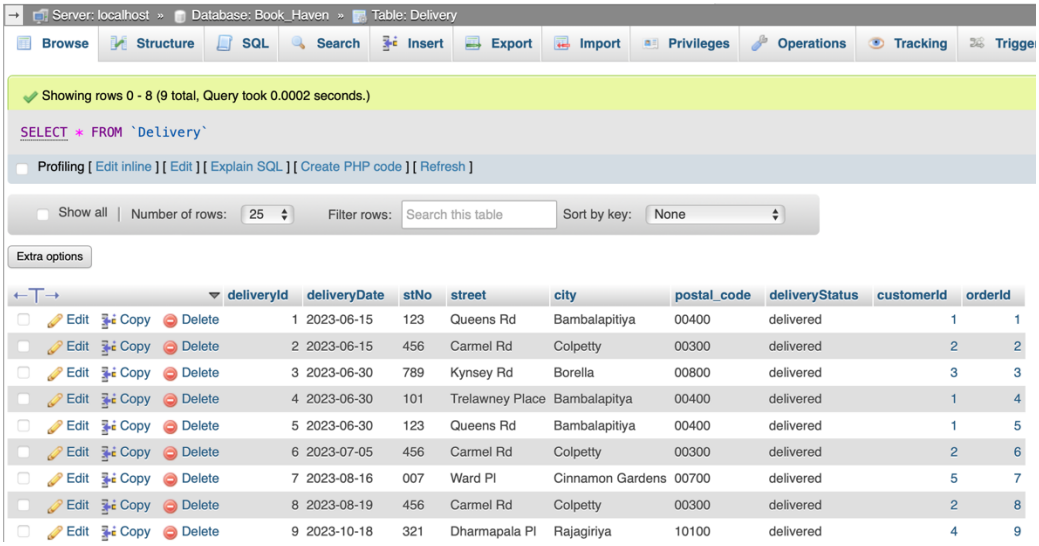


Figure 92 Data entered Delivery table.

Database Diagram

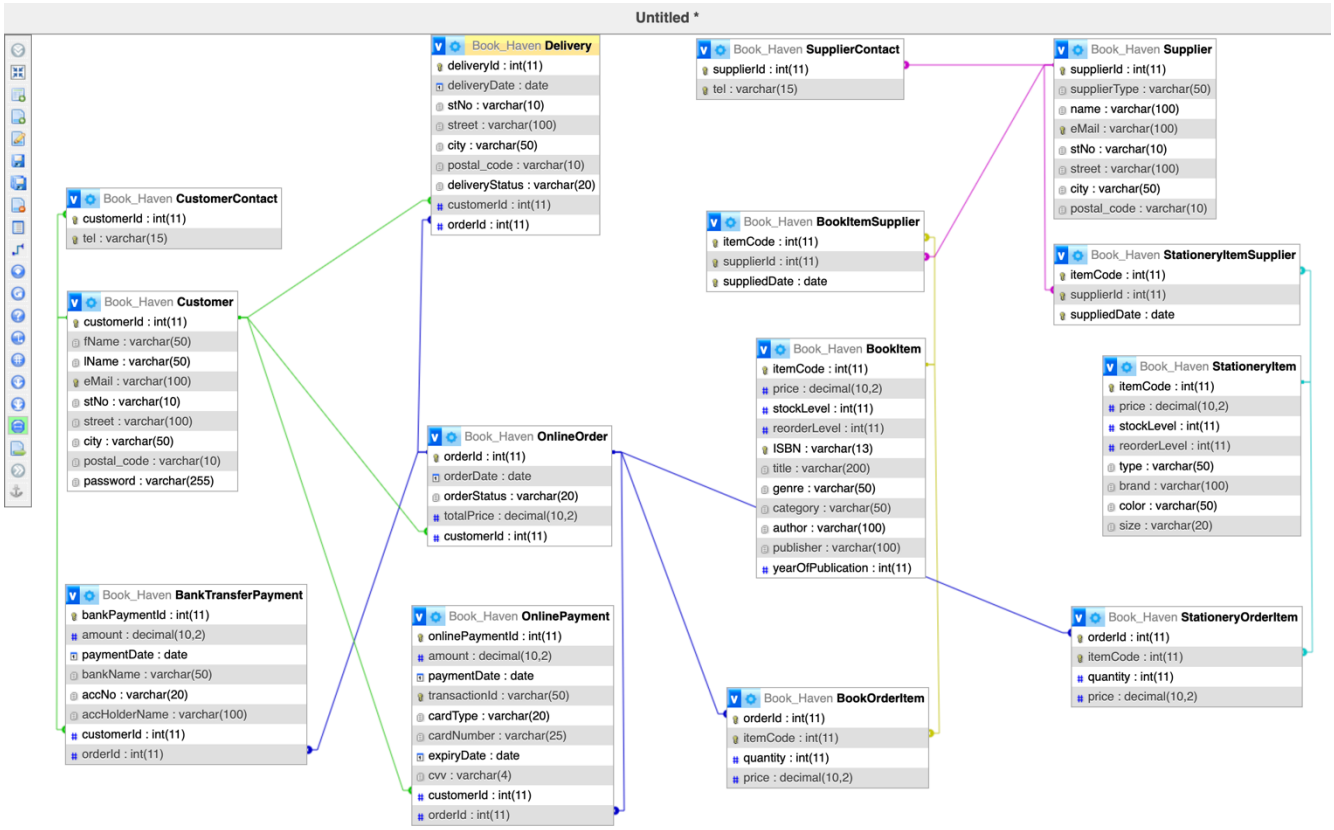


Figure 93 Database diagram

Section 4

Data manipulation with SQL

a)

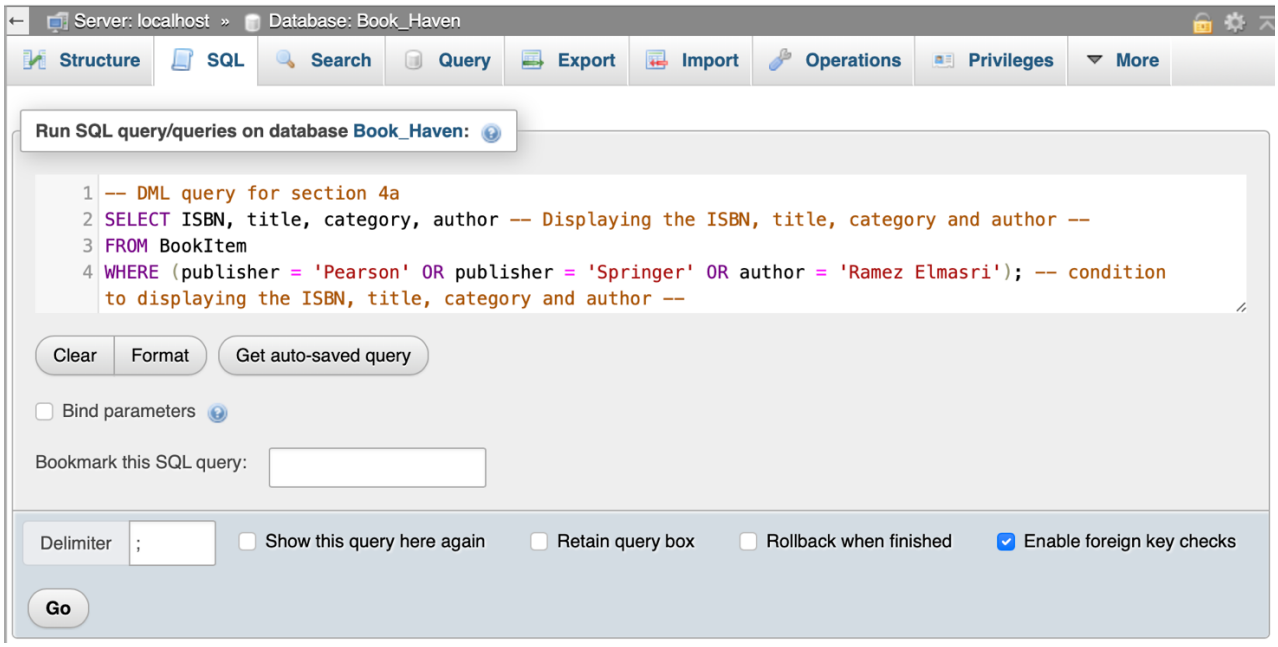


Figure 94 DML query for question a)

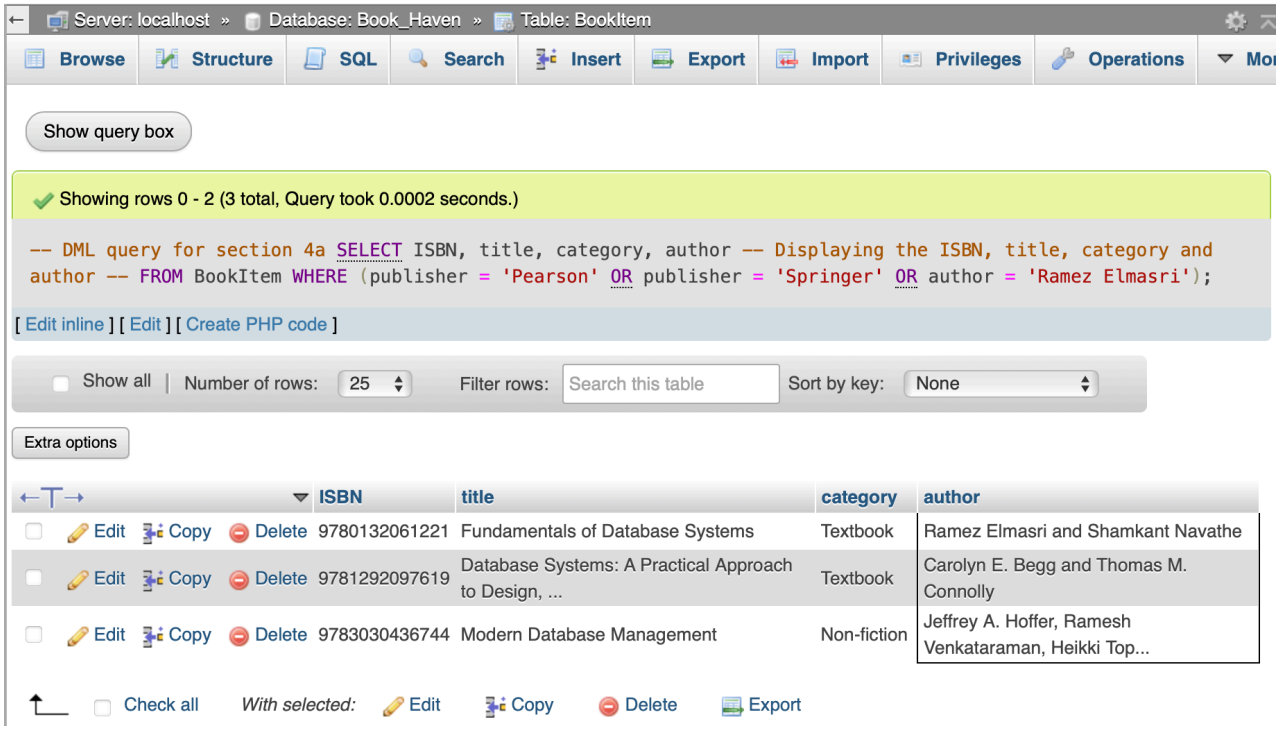


Figure 95 results for question a)

b)

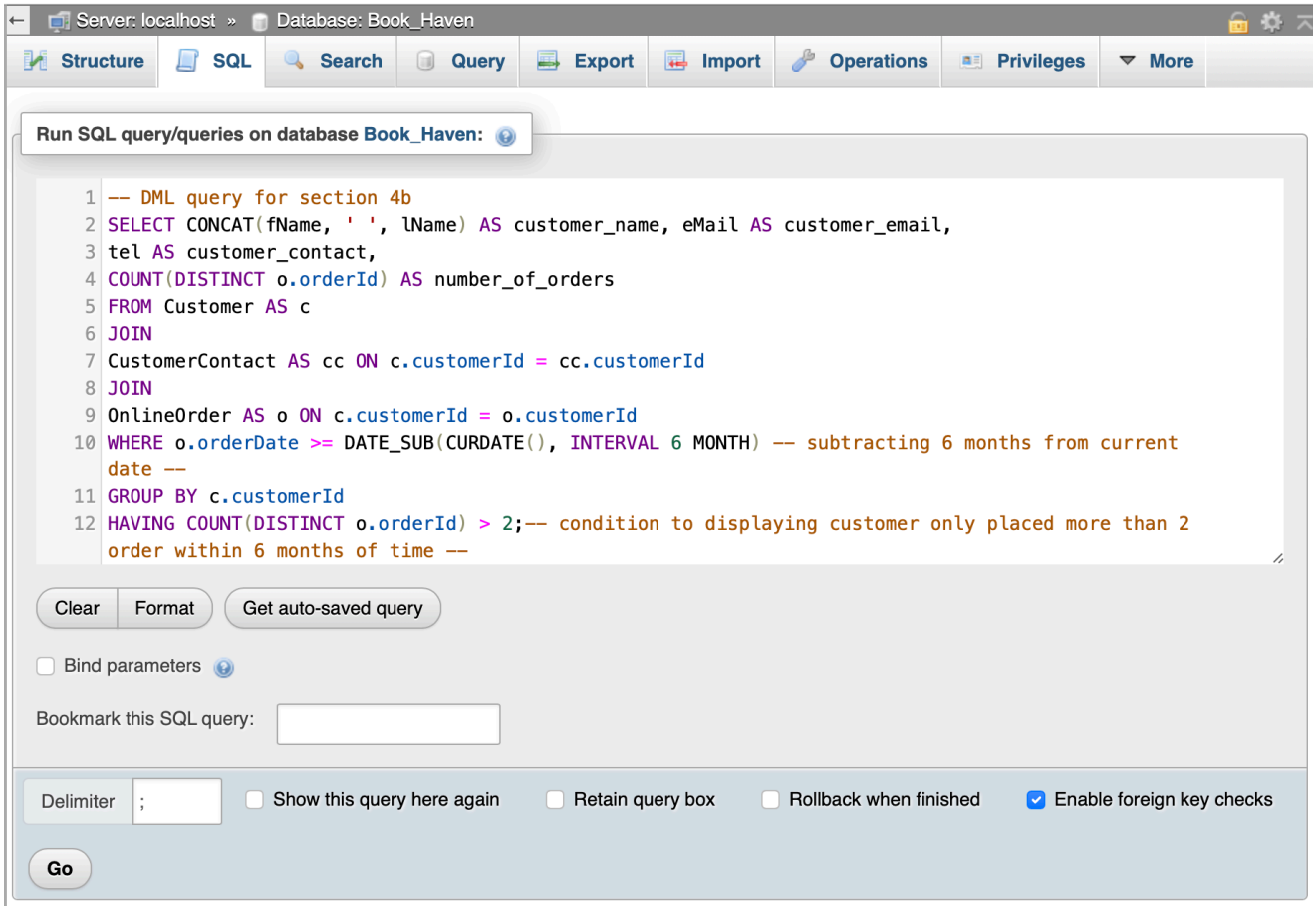


Figure 96 DML query for question b)

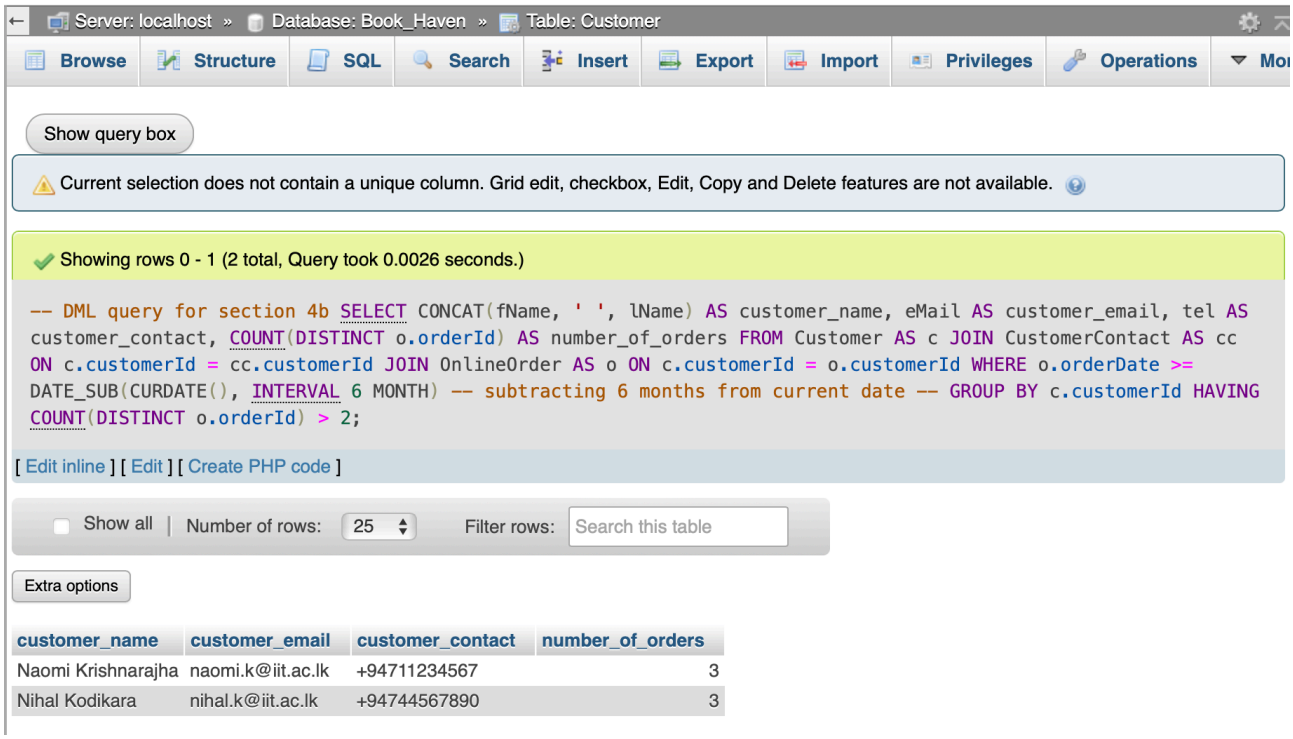


Figure 97 results for question b)

Reference list

- Vertabelo Data Modeler. (2014). *How to Create a Database Model From Scratch*. [online] Available at: https://vertabelo.com/blog/how-to-create-a-database-model-from-scratch/ [Accessed 1 Dec. 2023].

- Vertabelo Data Modeler. (2015). *Tips for Better Database Design*. [online] Available at: https://vertabelo.com/blog/9-tips-for-better-database-design/ [Accessed 27 Nov. 2023].

- Connolly, T.M. and Begg, C. (2015). *Database systems: a practical approach to design, implementation and management*. Harlow: Pearson Education Limited.

- w3schools (2019). *SQL Tutorial*. [online] W3schools.com. Available at: https://www.w3schools.com/sql/ .

Triggers

- GeeksforGeeks. (2018). *SQL | Triggers*. [online] Available at: https://www.geeksforgeeks.org/sql-triggers/?ref=ml_lbp [Accessed 27 Nov. 2023].

- www.youtube.com. (n.d.). *Triggers | SQL | Tutorial 20*. [online] Available at: https://www.youtube.com/watch?v=gpthfJnvzY8 [Accessed 27 Nov. 2023].

- www.youtube.com. (n.d.). *Triggers In SQL | Triggers In Database | SQL Triggers Tutorial For Beginners | Edureka*. [online] Available at: https://www.youtube.com/watch?v=f6VWSlnHGCE [Accessed 27 Nov. 2023].