

# BSc (Hons) Artificial Intelligence and Data Science

**Module: CM1603 Database Systems**

**Individual Coursework Report**

**Module Leader: Ms. Dileeka Alwis**

RGU Student ID : 2506755

IIT Student ID : 20240281

Student Name : W.D. Nisitha Nimsara

## 1. Abstract

This report focuses on creating a Database for Slip'n Snack Café. The report includes 5 major sections that help to create the database from scratch. Section 1 is Conceptual Extended Entity Relationship Diagram(EERD). This section contains EERD with the entities, relationships, attributes, multiplicities with assumptions. Section 2 is Logical Schema (Relational Schema). This section contains the relational Schema for the above EERD with appropriate assumptions. Section 3 is Data normalization. This section contains justification of the data normalization processes up to 3NF identified from the above relational scheme. Section 4 is Table Creation and Population of Data. This section includes the complete SQL queries used to implement the above relational schema in MySQL and SQL queries that insert the data into database. Section 5 is Data Manipulation with SQL. The section contains suitable SQL queries with screenshots that fulfill the required tasks.

## 2. Acknowledgement

I am profoundly grateful to the IIT and RGU for creating the Database system module for Artificial intelligence and Data science undergraduate degree program and IIT management for creating a helpful environment for us to improve our academic and general skills under the direction of reputable and experienced lecturer and Tutor.

First and foremost, I would like to express my gratitude to Mrs. Dileeka Alwis, Module leader, for providing us with guidance in creating a complete database from the scratch. I would also like to extend my appreciation to Ms. Asini Pathmila Silva, Assistant Lecturer, for her constructive feedback and encouragement throughout this time. Their guidance and input have been crucial in shaping the direction and improving the quality of our work. Finally, I want to express my special thanks to my batchmates and friends, for providing technical advice to me and sharing the knowledge they have. Which has been fundamental in helping me to reach this milestone.

### 3. Table of Contents

1. Abstract .....	2
2. Acknowledgement .....	3
3. Table of Contents .....	4
4. List of figures .....	6
5.1 Section 1 - Conceptual Extended Entity Relationship Diagram (EERD) .....	7
5.2 Assumptions for Conceptual EERD .....	8
6.1 Section 2 - Logical ERD (Relational Schema).....	10
6.2 Assumptions for Logical Entity Relationship Diagram .....	11
7.1 Section 3 - Data Normalization (up to 3NF) .....	12
7.2 Final Relational Schema.....	16
8.1 Section 4: Table Creation .....	17
8.1.1 Create database Cafe' Management.....	17
8.1.2 Create Table Customer.....	18
8.1.3 Create CustTeleNumber Table .....	19
8.1.4 Create OrderDetail Table .....	20
8.1.5 Create Payment Table .....	21
8.1.6 Create CardPayment Table.....	22
8.1.7 Create Employee Table .....	23
8.1.8 Create EmpTeleNumber Table .....	24
8.1.9 Create DeliveryStaff Table .....	25
8.1.10 Create Vehicle Table.....	26
8.1.11 Create Assignment Table.....	27
8.1.12 Create Inventory Table.....	28
8.1.13 Create Item Table .....	29
8.1.14 Create OrderedProduct Table .....	30
8.1.15 Create Supplier Table .....	31
8.1.16 Create SupTeleNumber Table .....	32
8.1.17 Create SourceDetail Table .....	33
8.2 Population of Data .....	34
8.2.1 Insert data into Customer Table .....	34
8.2.2 Insert data into CustTeleNumbr table .....	35
8.2.3 Insert data into OrderDetail table.....	36
8.2.4 Insert data into Payment table.....	37
8.2.5 Insert data into CardPayment table .....	38

8.2.6 Insert data into Employee table.....	39
8.2.7 Insert data into EmpTeleNumber table .....	40
8.2.8 Insert data into DeliveryStaff table.....	41
8.2.9 Insert data into Vehicle table .....	42
8.2.10 Insert data into Assignment table .....	43
8.2.11 Insert data into Inventory table .....	44
8.2.12 Insert data into Item table.....	45
8.2.13 Insert data into OrderedProduct table.....	46
8.2.14 Insert data into Supplier table.....	47
8.2.15 Insert data into SupTeleNumber table .....	48
8.2.16 Insert data into SourceDetail table .....	49
8.3 Designer Diagram.....	50
9. Section 5: Data Manipulation with SQL.....	51
9.1 Question - 1 .....	51
9.2 Question - 2 .....	52
9.3 Question - 3 .....	53
10. Reference.....	54

## 4. List of figures

Figure 1 - Extended Entity Relationship Diagram .....	7
Figure 2 - Logical Entity Relationship Diagram.....	10
Figure 3 - Cafe' Management Database Creation .....	17
Figure 4 - Customer Table Creation .....	18
Figure 5 - CustTeleNumber Table Creation.....	19
Figure 6 - OrderDetail Table Creation .....	20
Figure 7 - Payment Table Creation .....	21
Figure 8 - CardPayment Table Creation.....	22
Figure 9 - Employee Table Creation.....	23
Figure 10 - EmpTeleNumber Table Creation.....	24
Figure 11 - DeliveryStaff Table Creation .....	25
Figure 12 - Vehicle Table Creation.....	26
Figure 13 - Assignment Table Creation.....	27
Figure 14 - Inventory Table Creation.....	28
Figure 15 - Item Table Creation .....	29
Figure 16 - OrderedProduct Table Creation .....	30
Figure 17 - Supplier Table Creation .....	31
Figure 18 - SupTeleNumber Table Creation.....	32
Figure 19 - SourceDetail Table Creation .....	33
Figure 20 - insert data in to Customer Table .....	34
Figure 21 - insert data in to CustTeleNumber Table .....	35
Figure 22 - insert data in to OrderDetail Table.....	36
Figure 23 - insert data in to Payment Table.....	37
Figure 24 - insert data in to CardPayment Table .....	38
Figure 25 - insert data in to Employee Table .....	39
Figure 26 - insert data in to EmpTeleNumber Table .....	40
Figure 27 - insert data in to DeliveryStaff Table.....	41
Figure 28 - insert data in to Vehicle Table .....	42
Figure 29 - insert data in to Assignment Table .....	43
Figure 30 - insert data in to Inventory Table .....	44
Figure 31 - insert data in to Item Table.....	45
Figure 32 - insert data in to OrderedProduct Table .....	46
Figure 33 - insert data in to Supplier Table.....	47
Figure 34 - insert data in to SupTeleNumber Table .....	48
Figure 35 - insert data in to SourceDetail Table .....	49
Figure 36 - Designer Diagram .....	50
Figure 37 - Question 1 Output.....	51
Figure 38 - Question 2 Output.....	52
Figure 39 - Question 3 Output.....	53

## 5.1 Section 1 - Conceptual Extended Entity Relationship Diagram (EERD)

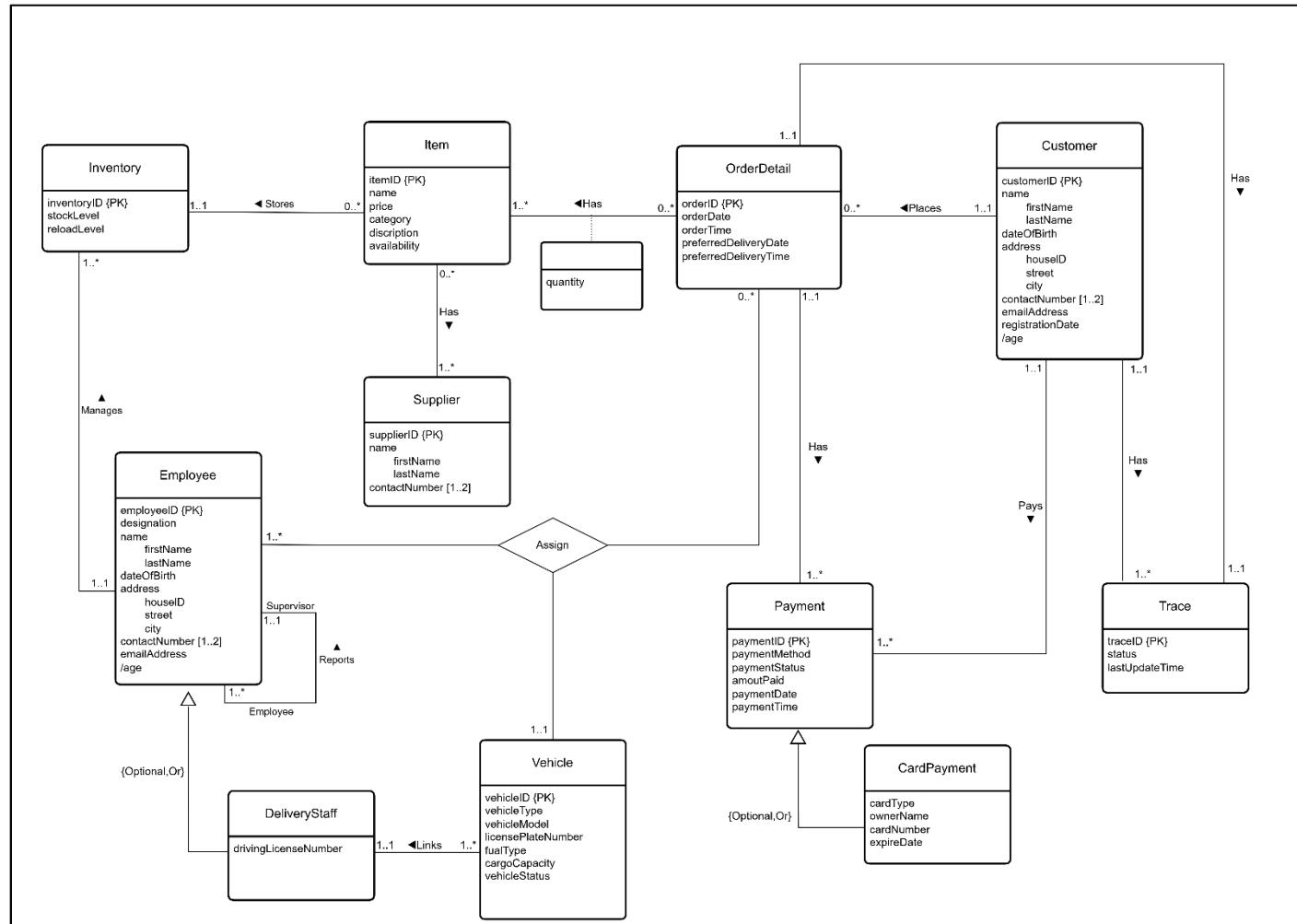


Figure 1 - Extended Entity Relationship Diagram

## 5.2 Assumptions for Conceptual EERD

- Each Customer has a unique customerID.
- Customer details include firstName, lastName, dateOfBirth, houseID, street, city, contactNumbers, emailAddress, registrationDate, and age.
- Each Customer can have minimum 1 contactNumber and maximum 2 contactNumbers.
- Age derived from dateOfBirth.
- Each Customer may place 0 or many OrderDetails, but each OrderDetail must place by exactly 1 Customer.
- Each OrderDetail uniquely identified by using orderID.
- OrderDetails include orderDate, orderTime, preferredDeliveryDate, preferredDeliveryTime.
- Each OrderDetails must have 1 or many Items, but each Item may have 0 or many OrderDetails.
- This relationship must store quantity of each Item in an Order.
- Each Item uniquely identified by using itemID.
- Item details include name, price, category, description, availability.
- Each Item must store in exactly one inventory, but each inventory may have 0 or many Items.
- Inventory has a unique inventoryID.
- Each inventory details include stockLevel, reloadLevel.
- Each Item must have 1 or many Suppliers, and each supplier may have 0 or many items.
- Supplier has a unique supplierID.
- Supplier details include firstName, lastName and contactNumbers.
- Each Supplier can have minimum 1 contactNumber and maximum 2 contactNumbers.
- Each OrderDetail must have 1 or may Payments, and each Payment must have exactly 1 OrderDetail.
- Payment has a unique paymentID.
- Payment details include paymentMethod, paymentStatus, amountPaid, paymentDate, paymentTime.
- Payment has different paymentMethods.
- CardPayment is one of subclass of Payment.
- CardPayment details include cardType, ownerName, cardNumber, expireDate.

- Each Customer can pay 1 or many Payments, but each Payment must be paid by exactly 1 Customer.
- Each OrderDetail must have exactly 1 Trace, and each Trace must have exactly 1 OrderDetail.
- Trace has a unique traceID.
- Trace details include status, lastUpdateTime.
- Each Employee has a unique employeeID.
- Employee details include designation, firstName, lastName, dateOfBirth, houseID, street, city, contactNumbers, emailAddress, and age.
- Each Employee can have minimum 1 contactNumber and maximum 2 contactNumbers.
- Age derived from dateOfBirth.
- Each employee must report to 1 Supervisor, and each Supervisor receives report from 1 or many Employees.
- Each Employee must belong to one of employee type.
- DeliveryStaff is one of subclass of Employees.
- DeliveryStaff details include drivinglicenseNumber.
- Each manager, who is an employee, is responsible for managing an inventory.
- Each manager must manage 1 or many inventories, and each inventory must be managed by exactly 1 manager.
- Each OrderDetail is assigned to 1 Vehicle and 1 Steward employee.
- Each Steward employee is responsible for assigning Vehicle to 1 or many OrderDetails.
- Each Delivery employee links with 1 or many vehicles.
- Each Vehicle links with 1 delivery employee.
- Each Vehicle has a unique vehicleID.
- Vehicle details include vehicleType, vehicleModel, licensePlateNumber, fuelType, cargoCapacity, vehicleStatus.

## 6.1 Section 2 - Logical ERD (Relational Schema)

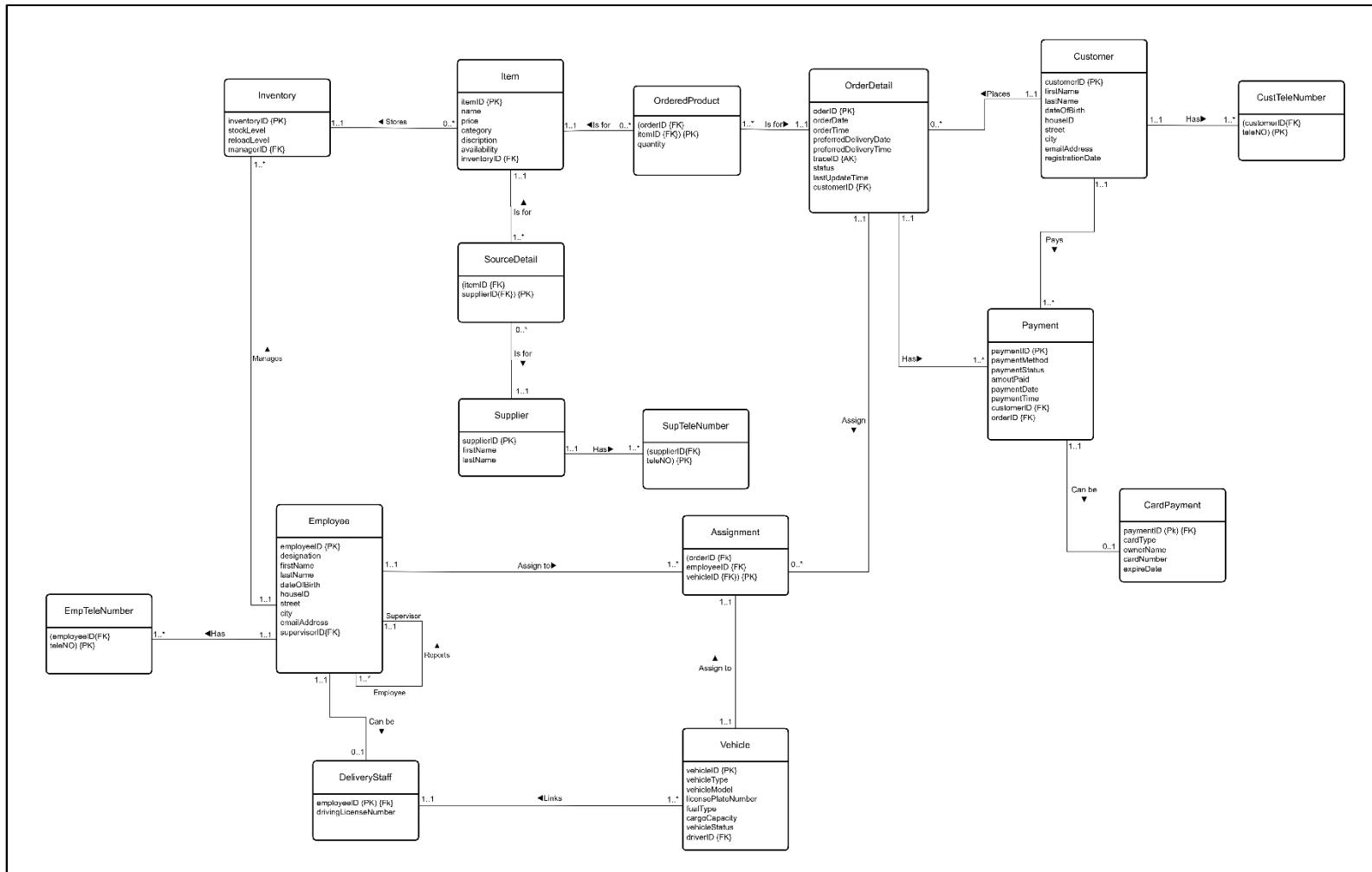


Figure 2 - Logical Entity Relationship Diagram

## 6.2 Assumptions for Logical Entity Relationship Diagram

- Customers can store multiple telephone numbers when converting EERD to logical diagram, new entity was created as (CustTeleNumber).
- Suppliers can store multiple telephone numbers, when converting EERD to logical diagram, new entity was created as (SupTeleNumber).
- Employees can store multiple telephone numbers, when converting EERD to logical diagram new entity was created as (EmpTelepNumber).
- Between Order and Item contain relationship attribute as Quantity, when converting EERD to logical diagram new child entity was created as (OrderedProduct).
- OrderedProduct has unique key{PK} that combination of both parent entities primary keys (orderID{Fk}+itemID{FK}) and attribute to store the quantity as (Quantity).
- Converting EERD to logical diagram, new child entity was created between Item and Supplier as (SourceDetail).
- SourceDetail has unique key{PK} } that combination of both parent entities primary keys (itemID{Fk}+supplierID{FK}).
- In EERD, OrderDetail, Vehicle and Employee have ternary relationship, when converting to logical design a new child entity was created as (Assignment).
- Assignment had unique key{PK} that combination of all parent entities primary keys (orderID{FK}+employeeID{FK}+vehicleID{FK}).
- CardPayment is a subclass of payment when it converted to logical design, new child entity was created as (CardPayment) and it includes all the card details.
- CardPayment has primary key of parent entity as foreign key.
- Employees have different employee types (sub classes), but only Delivery staff require to store driving license number.
- To store driving license numbers, new child was created as (DeliveryStaff).
- DeliveryStaff have primary key of parent entity as foreign key and drivingLicenseNumber.

## 7.1 Section 3 - Data Normalization (up to 3NF)

### Customer table

FD1- **customerID** → {firstName, lastName, dateOfBirth, houseID, street, city, emailAddress, registrationDate}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **customerID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

### custTeleNumber Table

FD1- {**customerID**, **teleNo**}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of **customerID** and **teleNo** and no other attributes can be seen meaning there are no partial dependencies so table can be assumed to be in 2NF.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

### OrderDetail Table

FD1 - **orderID** → {orderDate, orderTime, customerContactNumber, emailAddress, deliveryAddress, preferredDeliveryDate, preferredDeliveryTime, traceID, status, lastUpdateTime, *customerID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **orderID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Payment Table

FD1 – **paymentID** → {paymentMethod, paymentStatus, amountPaid, paymentDate, paymentTime, *customerID*, *orderID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **paymentID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## CardPayment Table

FD1 – **paymentID** → {cardType, ownerName, cardNumber, expireDate}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **paymentID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## OrderedProduct Table

FD1 – { **orderID**, **itemID** } → quantity

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of **orderID** and **itemID** and quantity attribute depending on it. so, it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF

## Item Table

FD1 – **itemID** → {Name, price, category, description, availability, *inventoryID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **itemID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Inventory Table

FD1 – **inventoryID** → { stockLevel, reloadLevel, *managerID* }

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **inventoryID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## SourceDetail Table

FD1 – {*itemID*,*supplierID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of **itemID** and **supplierID** and no other attributes can be seen meaning there are no partial dependencies so table can be assumed to be in 2NF.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Supplier Table

FD1 – **supplierID** → {firstName, lastName}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **supplierID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## SupTeleNumber Table

FD1- {*supplierID*, **teleNo**}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of **supplierID** and **teleNo** and no other attributes can be seen meaning there are no partial dependencies so table can be assumed to be in 2NF.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Employee Table

FD1 – **employeeID** → {designation, firstName, lastName, dateOfBirth, houseID, street, city, emailAddress, *supervisorID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **employeeID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## EmpTelephoneNo Table

FD1- {**employeeID**, **teleNo**}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of **employeeID** and **teleNo** and no other attributes can be seen meaning there are no partial dependencies so table can be assumed to be in 2NF.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## DeliveryStaff Table

FD1 – **employeeID** → drivingLicenseNumber

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **employeeID** and drivingLicenseNumber attribute depending on it. so, it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Vehicle Table

FD1 – **vehicleID** → {vehicleType, vehicleModel, licensePlateNumber, fuelType, cargoCapacity, vehicleStatus, *driverID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the **vehicleID** and all attributes depend on it, so it can be seen as a fully functional dependency, meaning that the table can be assumed to be in 2NF since there are no partial dependencies.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## Assignment Table

FD1 – {*orderID, employeeID, vehicleID*}

- Since the repeating attributes are not visible in this schema, it can be assumed that each field contains atomic values, so the table can be assumed to be in 1NF.
- The primary key is the combination of ***orderID, employeeID*** and ***vehicleID*** and no other attributes can be seen meaning there are no partial dependencies so table can be assumed to be in 2NF.
- Since no transitive dependency can be seen in the schema, it can be assumed that the table is in 3NF.

## 7.2 Final Relational Schema

1. Customer (***customerID***, firstName, lastName, dateOfBirth, houseID, street, city, emailAddress, registrationDate)
2. CustTeleNumber (***customerID, teleNo***)
3. OrderDetail (***orderID***, orderDate, orderTime, preferredDeliveryDate, preferredDeliveryTime, traceID, status, lastUpdateTime, *customerID*)
4. Payment (***paymentID***, paymentMethod, paymentStatus, amountPaid, paymentDate, paymentTime, *customerID, orderID*)
5. CardPayment (***paymentID***, cardType, ownerName, cardNumber, expireDate)
6. Inventory (***inventoryID***, stockLevel, reloadLevel, *managerID*)
7. Item (***itemID***, Name, price, category, description, availability, *inventoryID*)
8. OrderedProduct (*orderID, itemID*, quantity)
9. Employee (***employeeID***, designation, firstName, lastName, dateOfBirth, houseID, street, city, emailAddress, *supervisorID*)
10. EmpTelephoneNo (***employeeID, teleNo***)
11. DeliveryStaff (***employeeID***, drivingLicenseNumber)
12. SourceDetail (***itemID, supplierID***)
13. Supplier (***supplierID***, firstName, lastName)
14. SupTelephoneNo (***supplierID, teleNo***)
15. Vehicle (***vehicleID***, vehicleType, vehicleModel, licensePlateNumber, fuelType, cargoCapacity, vehicleStatus, *driverID*)
16. Assignment (***orderID, employeeID, vehicleID***)

## 8.1 Section 4: Table Creation

### 8.1.1 Create database Cafe' Management

CREATE DATABASE CafeManagement;

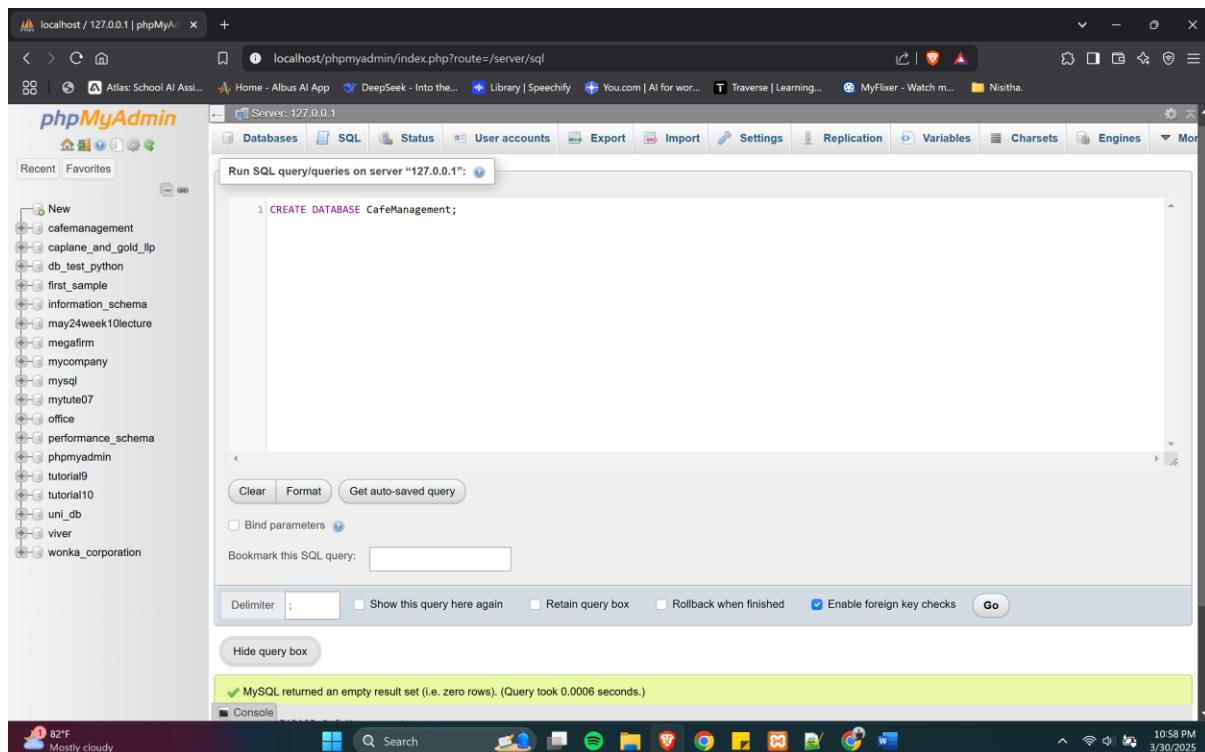


Figure 3 - Cafe' Management Database Creation

## 8.1.2 Create Table Customer

```
CREATE TABLE Customer (
    customerID char(4),
    firstName varchar(15) not null,
    lastName varchar(20) not null,
    dateOfBirth date not null,
    houseID varchar(15) not null,
    street varchar(20) not null,
    city varchar(20) not null,
    emailAddress varchar(35) unique,
    registrationDate date not null,
    CONSTRAINT c_cust_pk PRIMARY key (customerID)
);
```

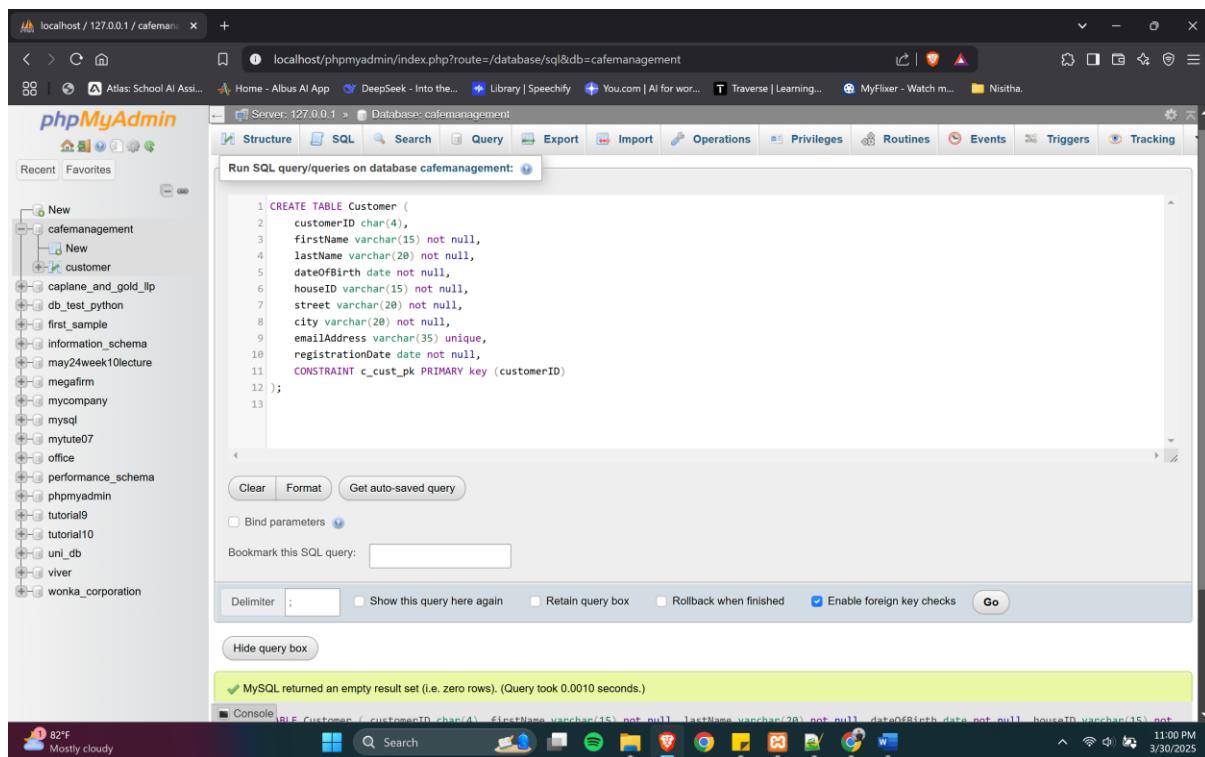


Figure 4 - Customer Table Creation

### 8.1.3 Create CustTeleNumber Table

```
CREATE TABLE Cust_Tele_Number (
    customerID char(4),
    telephoneNumber char(10),
    CONSTRAINT ct_custtele_fk FOREIGN key (customerID) REFERENCES
customer(customerID),
    CONSTRAINT ct_custTele_pk PRIMARY key (customerID , telephoneNumber)
);
```

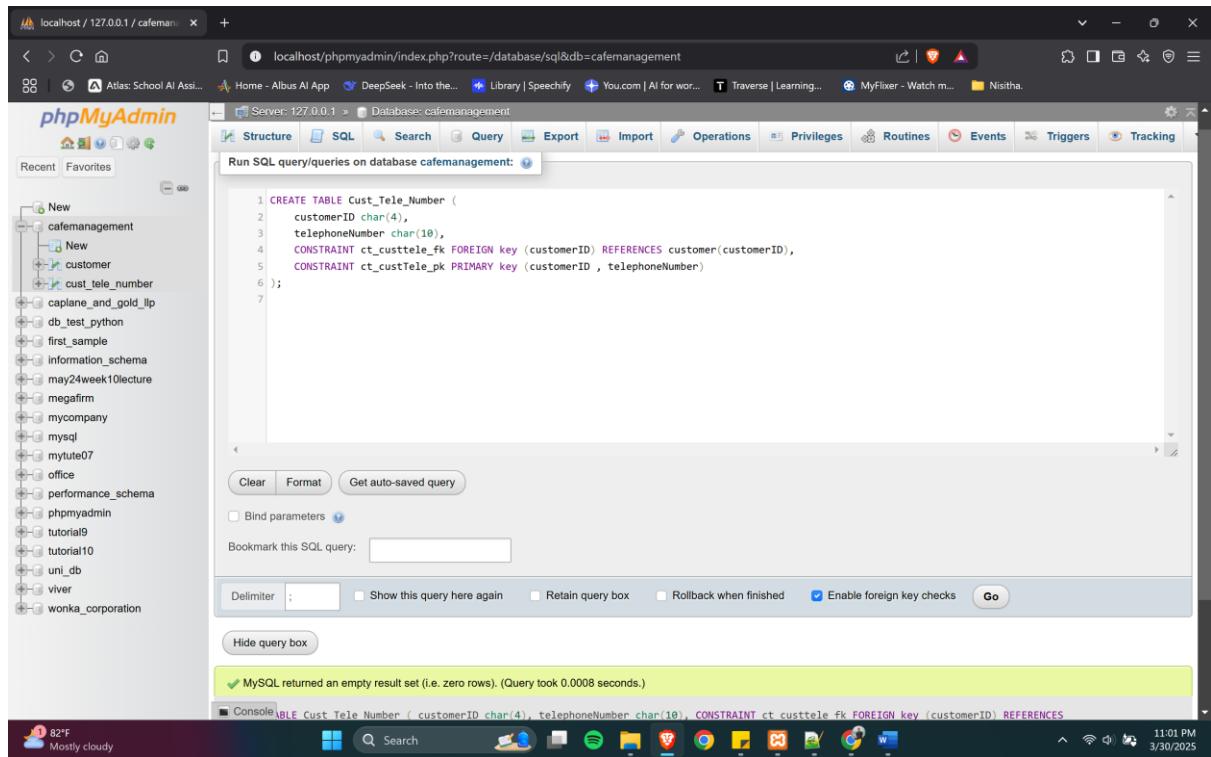


Figure 5 - CustTeleNumber Table Creation

## 8.1.4 Create OrderDetail Table

```
CREATE TABLE order_Detail (
    orderID char(4),
    orderDate date not null,
    orderTime time not null,
    preferredDeliveryDate date not null,
    preferredDeliveryTime time not null,
    traceID char(4) UNIQUE not null,
    orderStatus varchar(20) not null,
    customerID char(4) not null,
    CONSTRAINT o_order_pk PRIMARY key (orderID),
    CONSTRAINT o_cust_fk FOREIGN key (customerID) REFERENCES
customer(customerID)
);
```

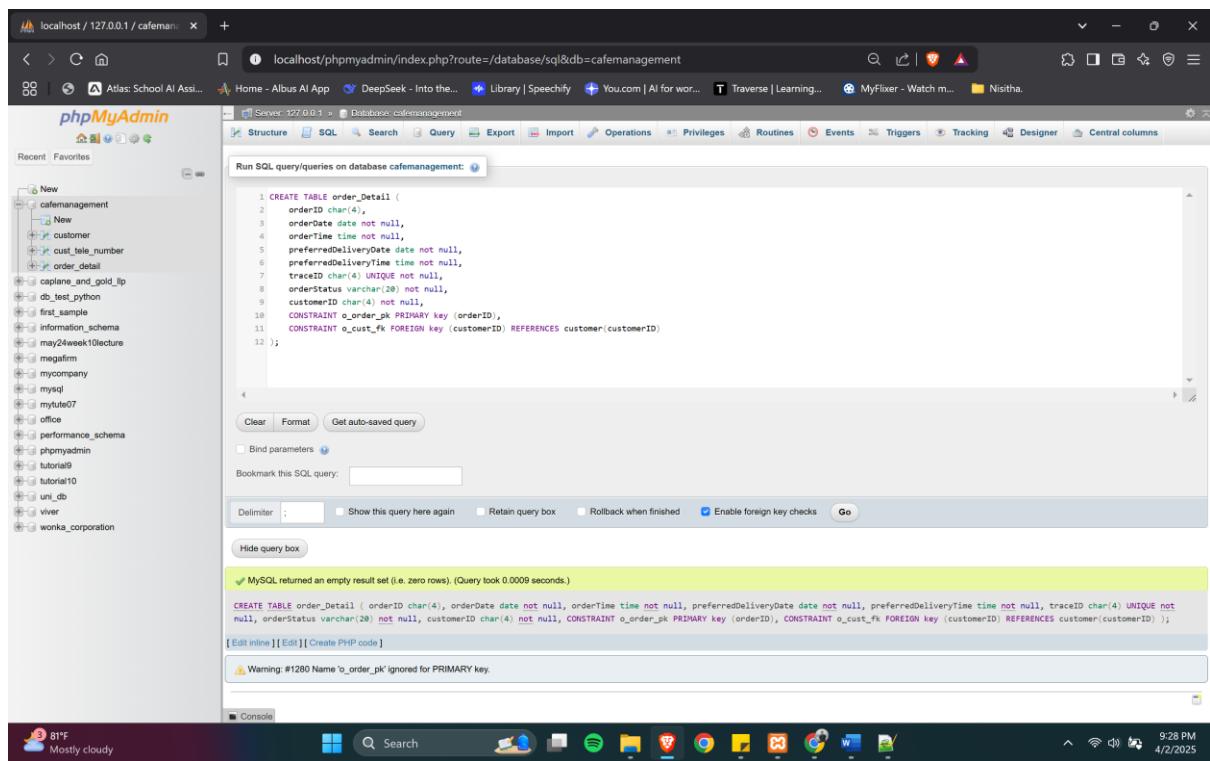


Figure 6 - OrderDetail Table Creation

### 8.1.5 Create Payment Table

```
CREATE TABLE payment (
    paymentID char(4),
    paymentMethod varchar(35) not null,
    paymentStatus varchar(20) not null,
    amountPaid float(8,2) not null,
    paymentDate date not null,
    paymenttime time not null,
    customerID char(4) not null,
    orderID char(4) not null,
    CONSTRAINT p_paym_pk PRIMARY key (paymentID),
    CONSTRAINT p_cust_fk FOREIGN key (customerID) REFERENCES
customer(customerID),
    CONSTRAINT p_order_fk FOREIGN key (orderID) REFERENCES
order_detail(orderID)
);
```

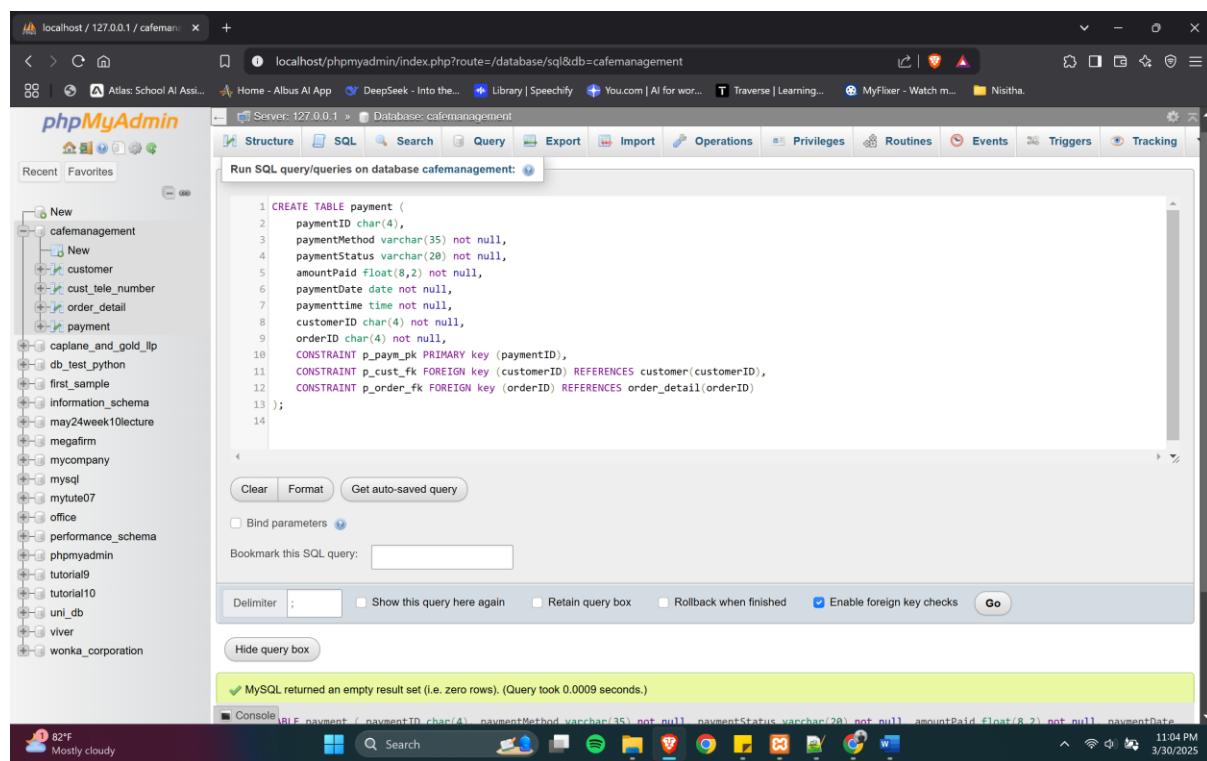


Figure 7 - Payment Table Creation

## 8.1.6 Create CardPayment Table

```
CREATE TABLE Card_Payment (
    paymentID char(4),
    cardtype varchar(10) not null,
    ownerName varchar(35) not null,
    cardNumber char(19) not null,
    expireDate varchar(5) not null,
    CONSTRAINT cp_payid_fk FOREIGN key (paymentID) REFERENCES
    payment(paymentID),
    CONSTRAINT cp_cardpy_pk PRIMARY key (paymentID)
);
```

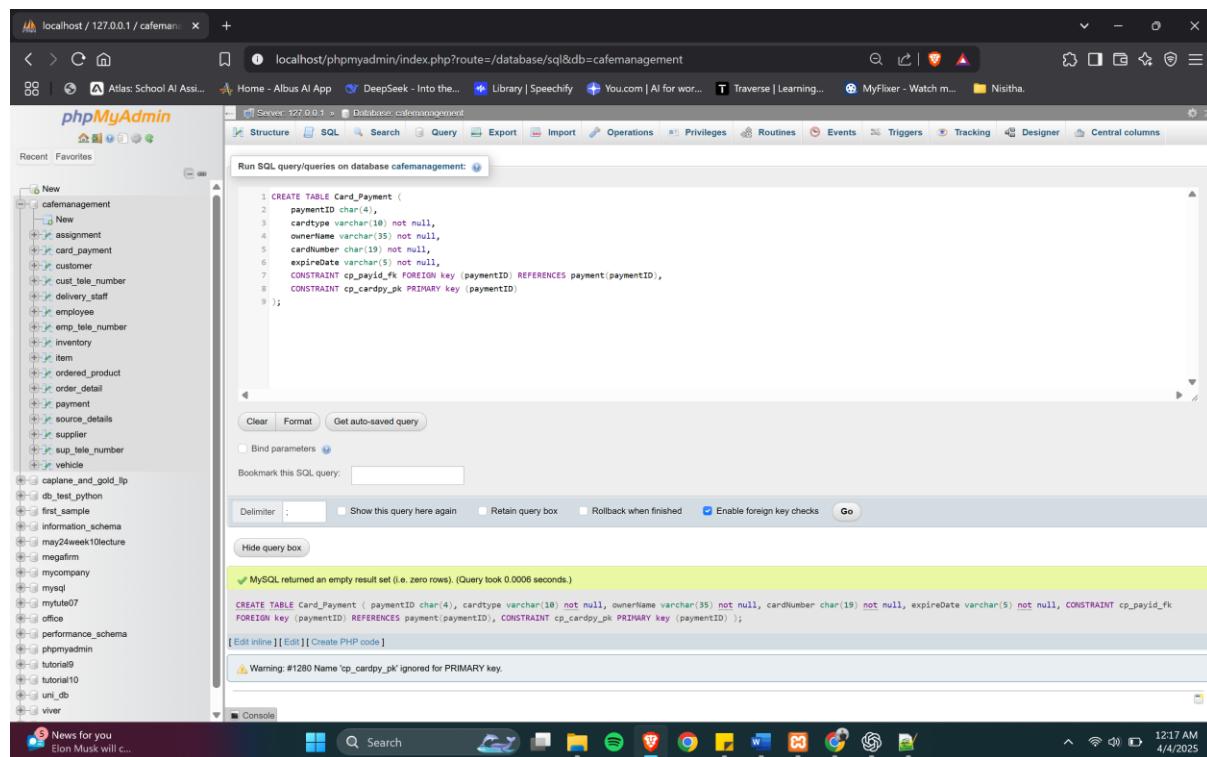


Figure 8 - CardPayment Table Creation

### 8.1.7 Create Employee Table

```
CREATE TABLE Employee (
    employeeID char(4),
    designation varchar(25) not null,
    firstName varchar(15) not null,
    lastName varchar(20) not null,
    dateOfBirth date not null,
    houseID varchar(15) not null,
    street varchar(20) not null,
    city varchar(20) not null,
    emailAddress varchar(35) UNIQUE,
    supervisorID char(4),
    CONSTRAINT e_empid_pk PRIMARY key (employeeID),
    CONSTRAINT e_supid_fk FOREIGN key (supervisorID) REFERENCES
Employee(employeeID)
);
```

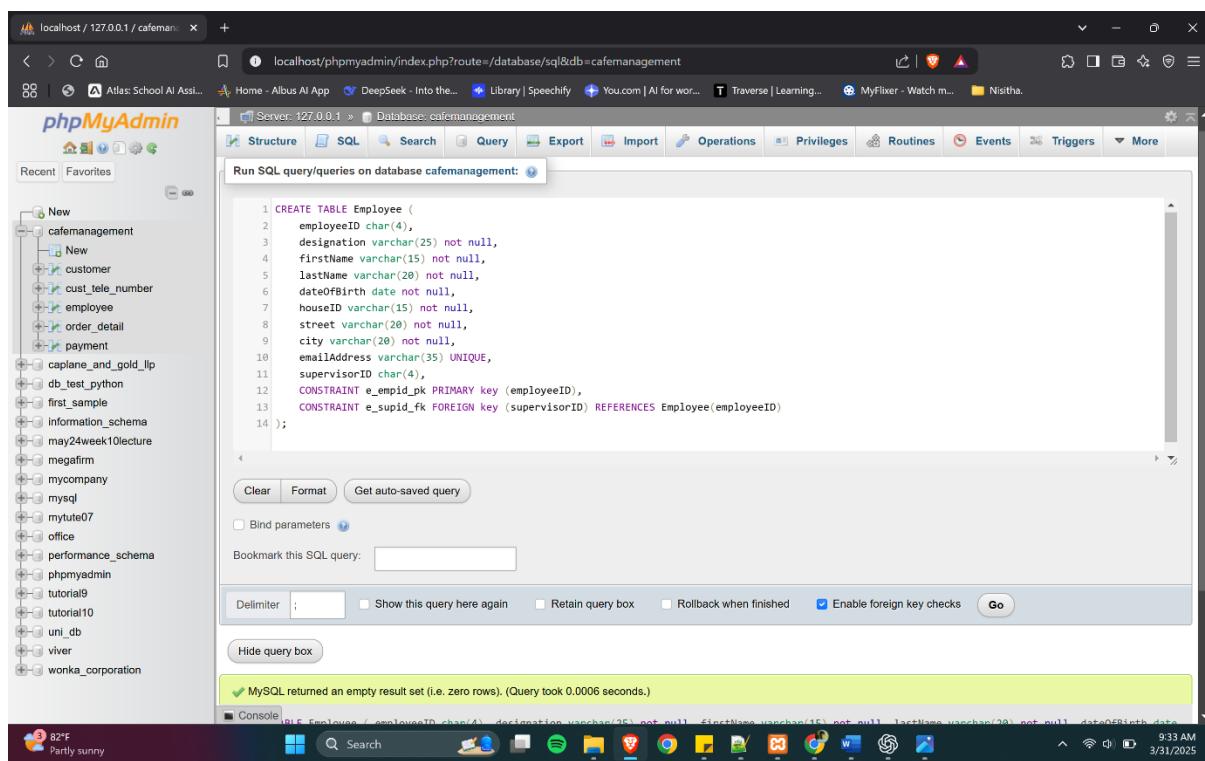


Figure 9 - Employee Table Creation

## 8.1.8 Create EmpTeleNumber Table

```
CREATE TABLE Emp_Tele_Number (
    employeeID char(4),
    telephoneNumber char(10),
    CONSTRAINT et_emptele_fk FOREIGN key (employeeID) REFERENCES
employee(employeeID),
    CONSTRAINT et_empTele_pk PRIMARY key (employeeID , telephoneNumber)
);
```

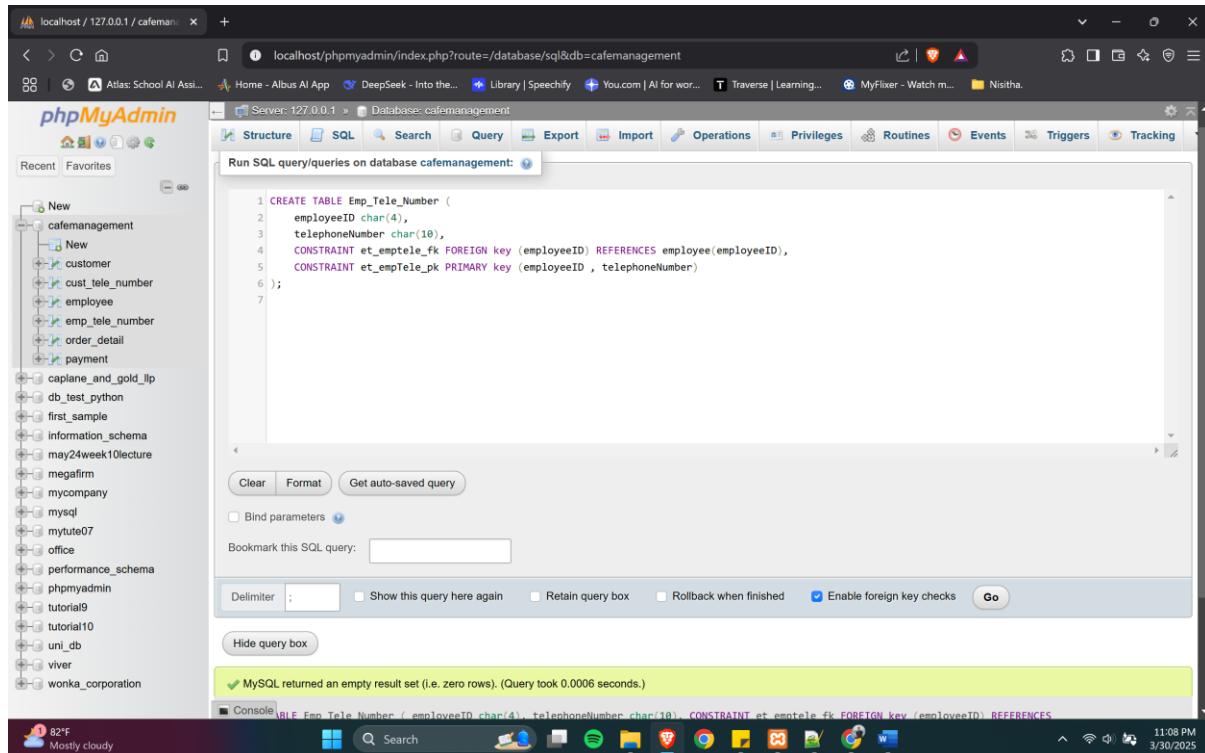


Figure 10 - EmpTeleNumber Table Creation

## 8.1.9 Create DeliveryStaff Table

```
CREATE TABLE delivery_Staff (
    employeeID char(4),
    drivingLicenseNumber char(10) not null UNIQUE,
    CONSTRAINT dest_empid_fk FOREIGN key (employeeID) REFERENCES
employee(employeeID),
    CONSTRAINT dest_destid_pk PRIMARY key (employeeID)
);
```

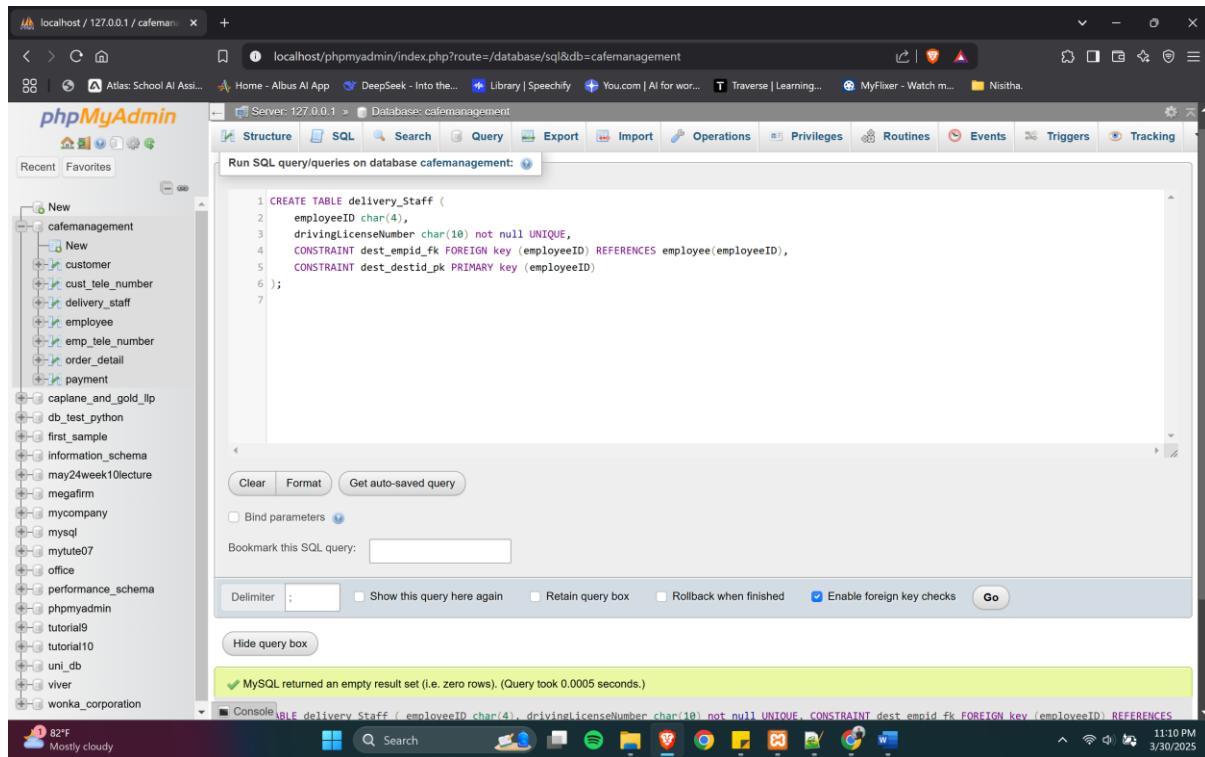


Figure 11 - DeliveryStaff Table Creation

### 8.1.10 Create Vehicle Table

```
CREATE TABLE Vehicle (
    vehicleID char(4),
    vehicleType varchar(15) not null,
    vehicleModel varchar(50) not null,
    licensePlateNumber varchar(12) not null UNIQUE,
    fuelType varchar(15) not null,
    cargoCapacity float(9,2) not null,
    vehicleStatus varchar(20) not null,
    driverID char(4) not null,
    CONSTRAINT ve_volid_pk PRIMARY KEY (vehicleID),
    CONSTRAINT ve_drid_fk FOREIGN key (driverID) REFERENCES Employee(employeeID)
);
```

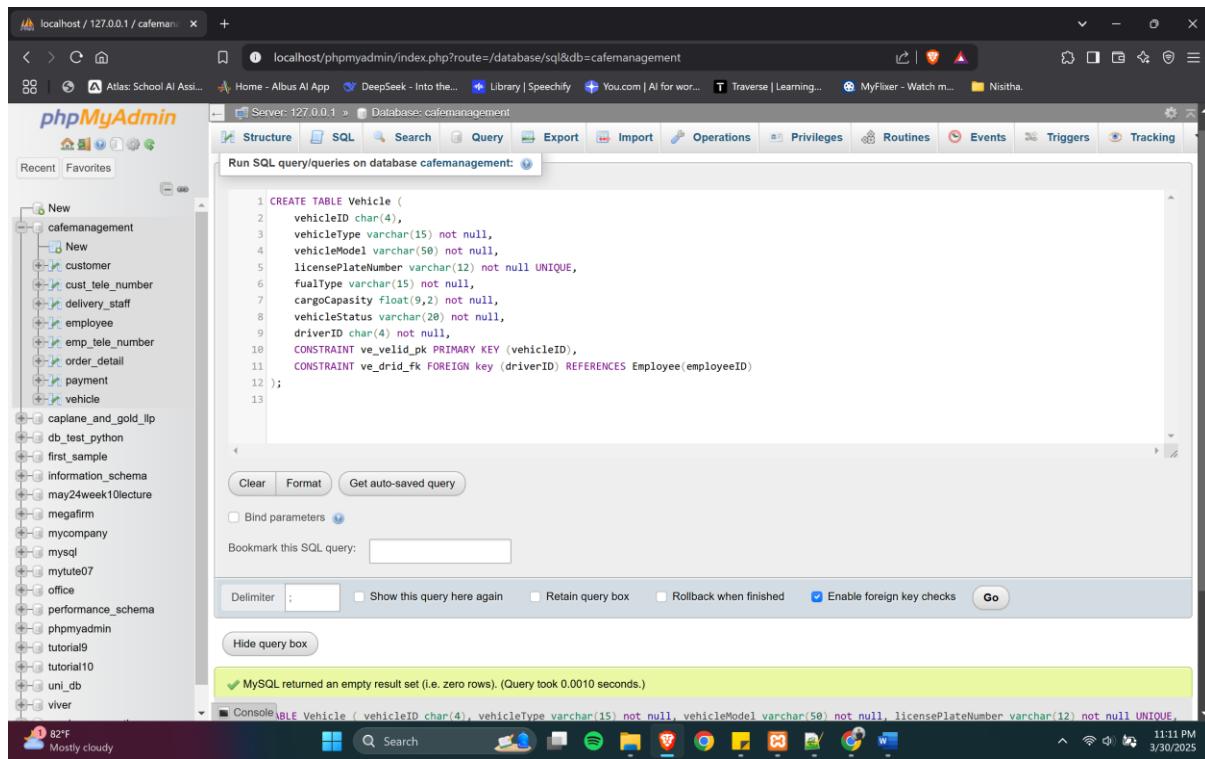


Figure 12 - Vehicle Table Creation

### 8.1.11 Create Assignment Table

```
CREATE TABLE Assignment (
    orderID char(4),
    StewardID char(4),
    vehicleID char(4),
    CONSTRAINT as_ordid_fk FOREIGN key (orderID) REFERENCES
    order_detail(orderID),
    CONSTRAINT as.empid_fk FOREIGN key (stewardID) REFERENCES
    employee(employeeID),
    CONSTRAINT as_vehid_fk FOREIGN key (vehicleID) REFERENCES
    vehicle(vehicleID),
    CONSTRAINT as_asiggid_pk PRIMARY key (orderID ,StewardID, vehicleID)
);
```

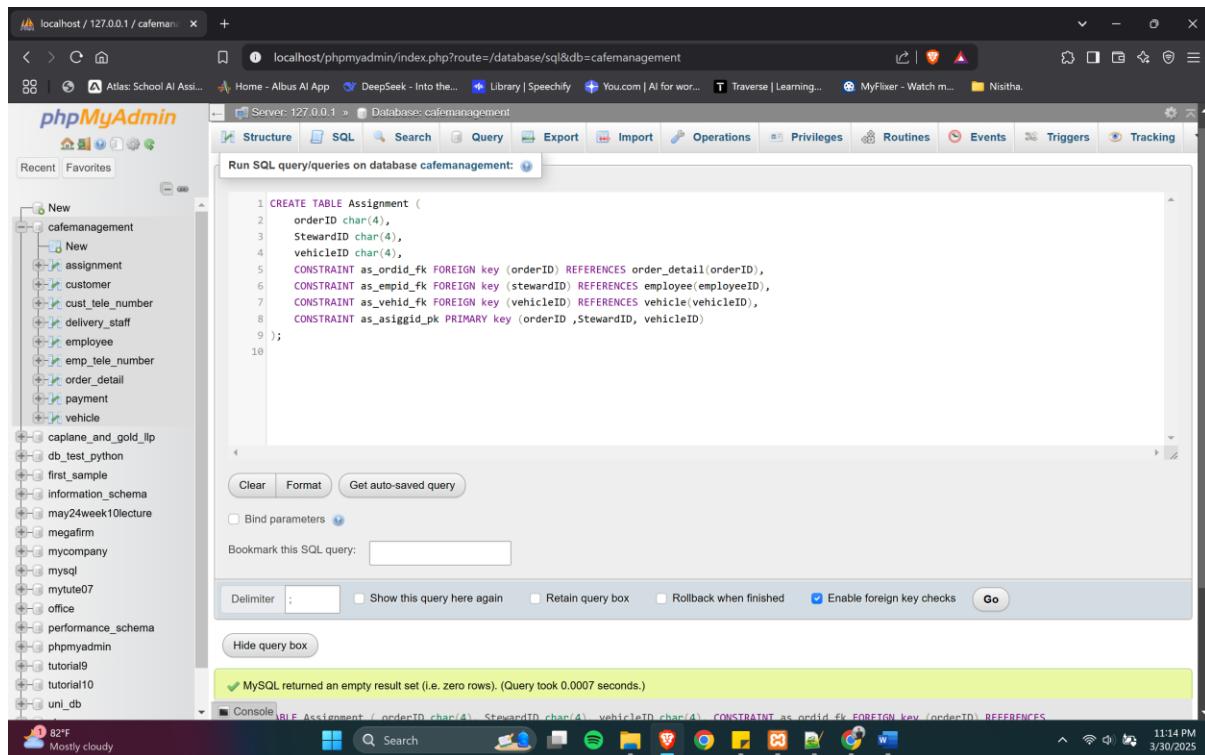


Figure 13 - Assignment Table Creation

### 8.1.12 Create Inventory Table

```
CREATE TABLE Inventory (
    inventoryID char(4),
    stocklevel float(10,2) not null,
    reloadLevel float(8,2) not null,
    managerID char(4) not null,
    CONSTRAINT in_invnid_pk PRIMARY key (inventoryID),
    CONSTRAINT in_manid_fk FOREIGN key (managerID) REFERENCES
Employee(employeeID)
);
```

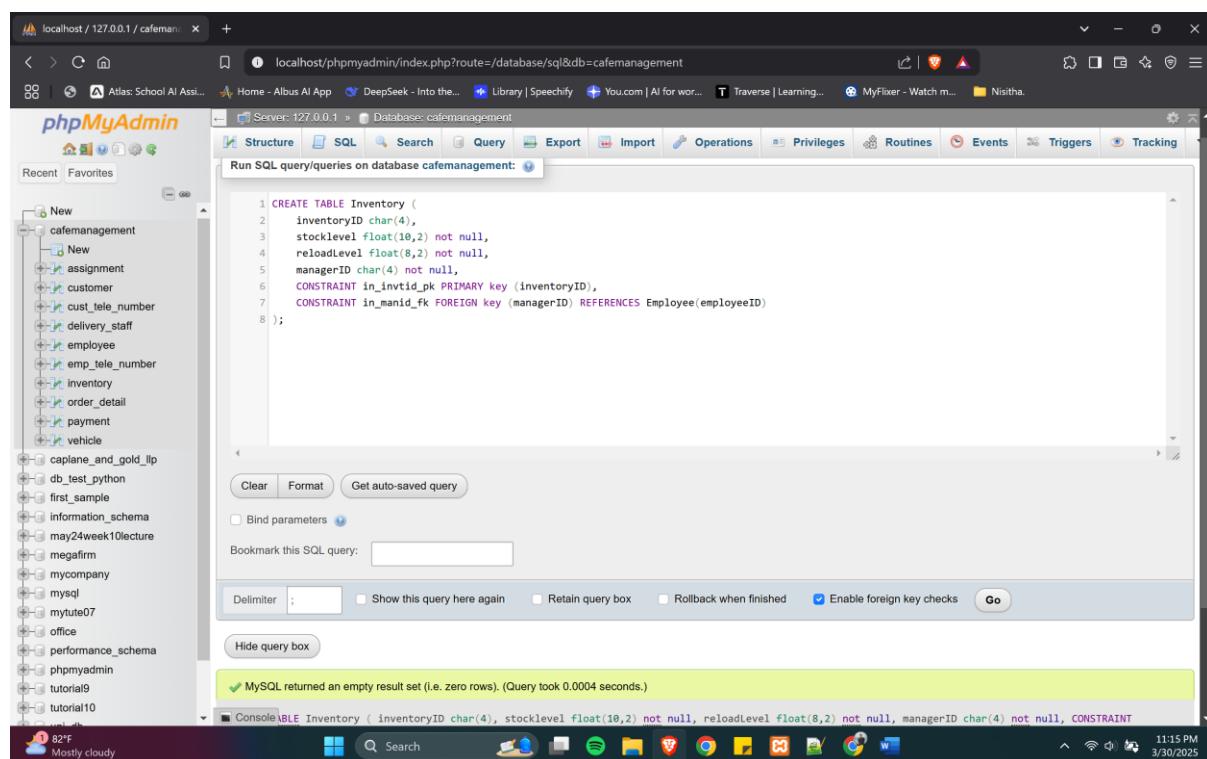


Figure 14 - Inventory Table Creation

### 8.1.13 Create Item Table

```
CREATE TABLE Item (
    itemID char(4),
    name varchar(20) not null,
    price float(8,2) not null,
    category varchar(20) not null,
    description varchar(150) not null,
    availability varchar(20) not null,
    inventoryID char(4) not null,
    CONSTRAINT it_itmid_pk PRIMARY key (itemID),
    CONSTRAINT it_inventid_fk FOREIGN key (inventoryID) REFERENCES
    inventory(inventoryID)
);
```

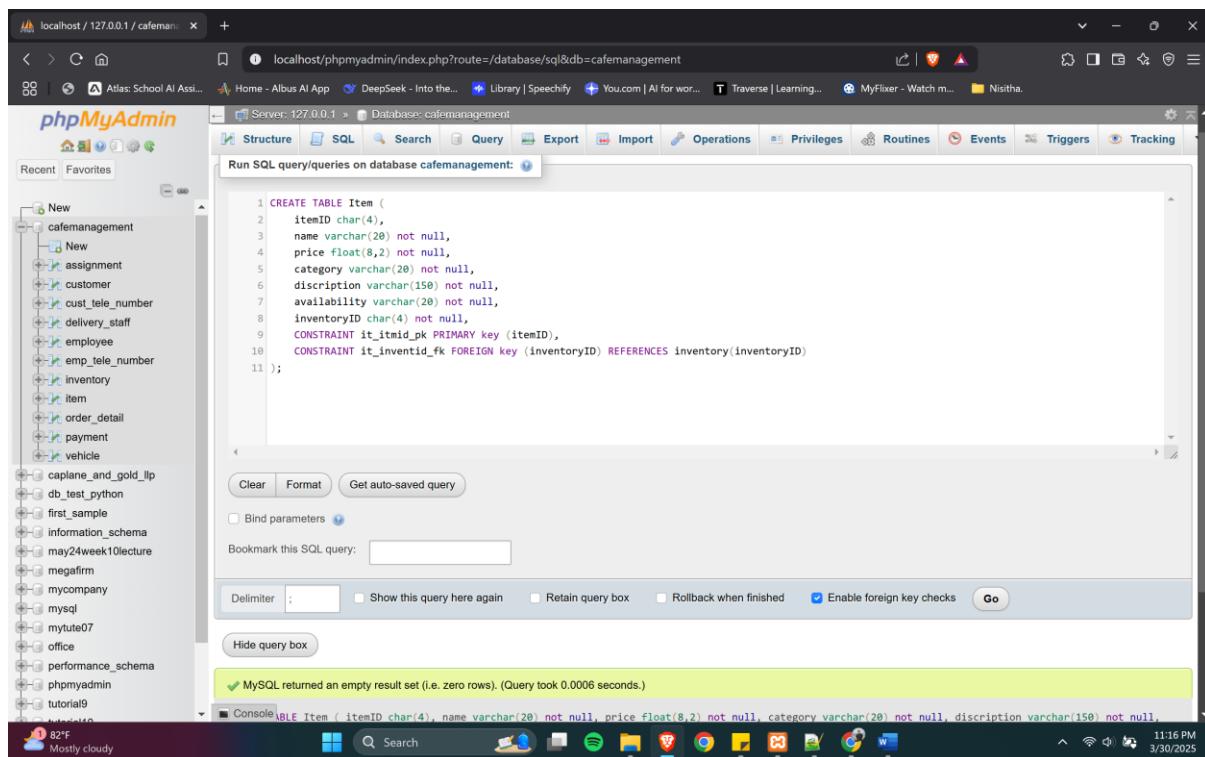


Figure 15 - Item Table Creation

### 8.1.14 Create OrderedProduct Table

```
CREATE TABLE Ordered_Product (
    orderID char(4),
    itemID char(4),
    quantity int(5) NOT null,
    CONSTRAINT op_ordid_fk FOREIGN key (orderID) REFERENCES
    order_detail(orderID),
    CONSTRAINT op_itmid_fk FOREIGN key (itemID) REFERENCES item(itemID),
    CONSTRAINT op_ordprod_pk PRIMARY key (orderID , itemID)
);
```

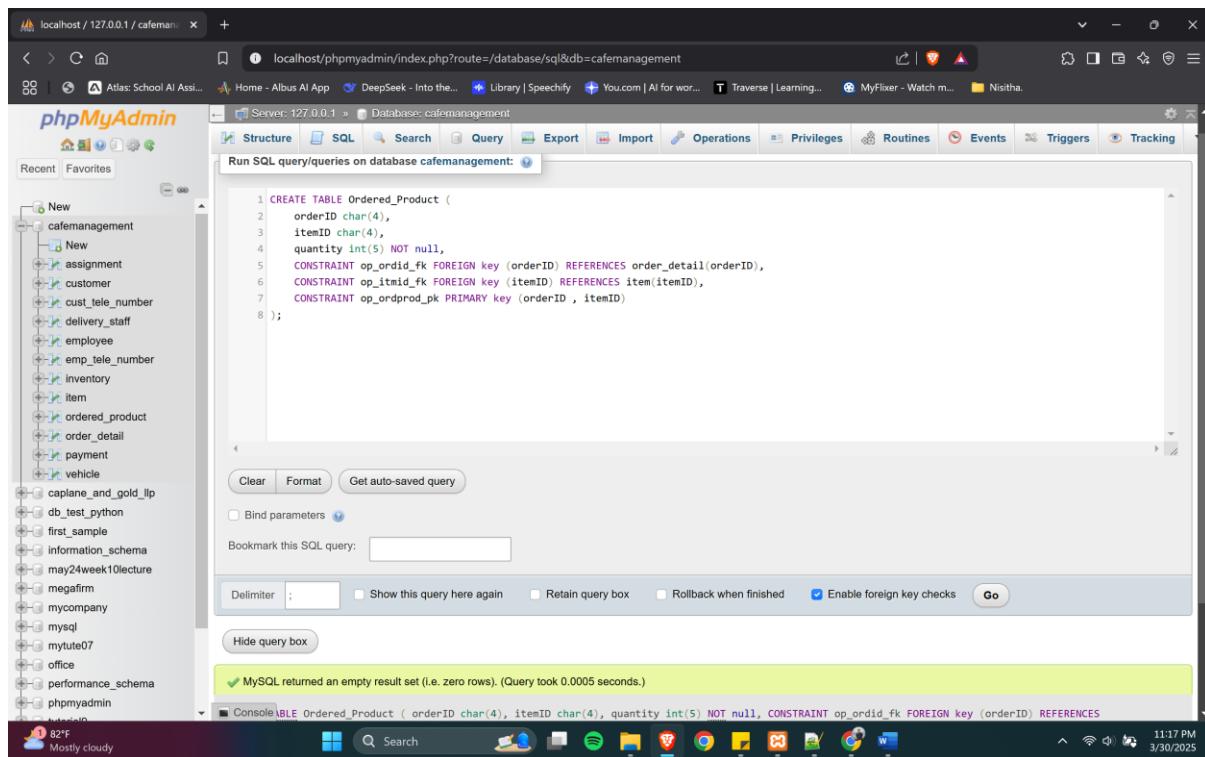


Figure 16 - OrderedProduct Table Creation

### 8.1.15 Create Supplier Table

```
CREATE TABLE Supplier (
    supplierID char(4),
    firstName varchar(15) not null,
    lastName varchar(20) not null,
    CONSTRAINT s_supid_pk PRIMARY key (supplierID)
);
```

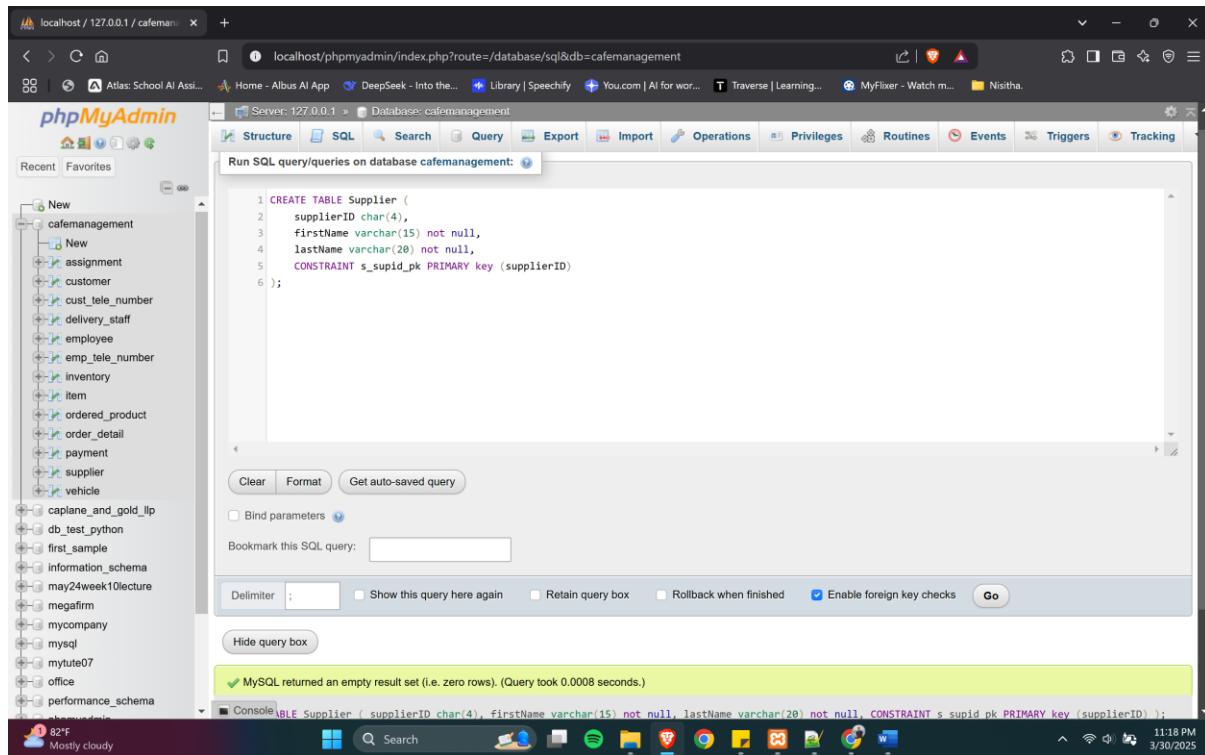


Figure 17 - Supplier Table Creation

### 8.1.16 Create SupTeleNumber Table

```
CREATE TABLE Sup_Tele_Number (
    supplierID char(4),
    telephoneNumber char(10),
    CONSTRAINT stn_sypid_fk FOREIGN key (supplierID) REFERENCES
    supplier(supplierID),
    CONSTRAINT stn_supTele_pk PRIMARY key (supplierID , telephoneNumber)
);
```

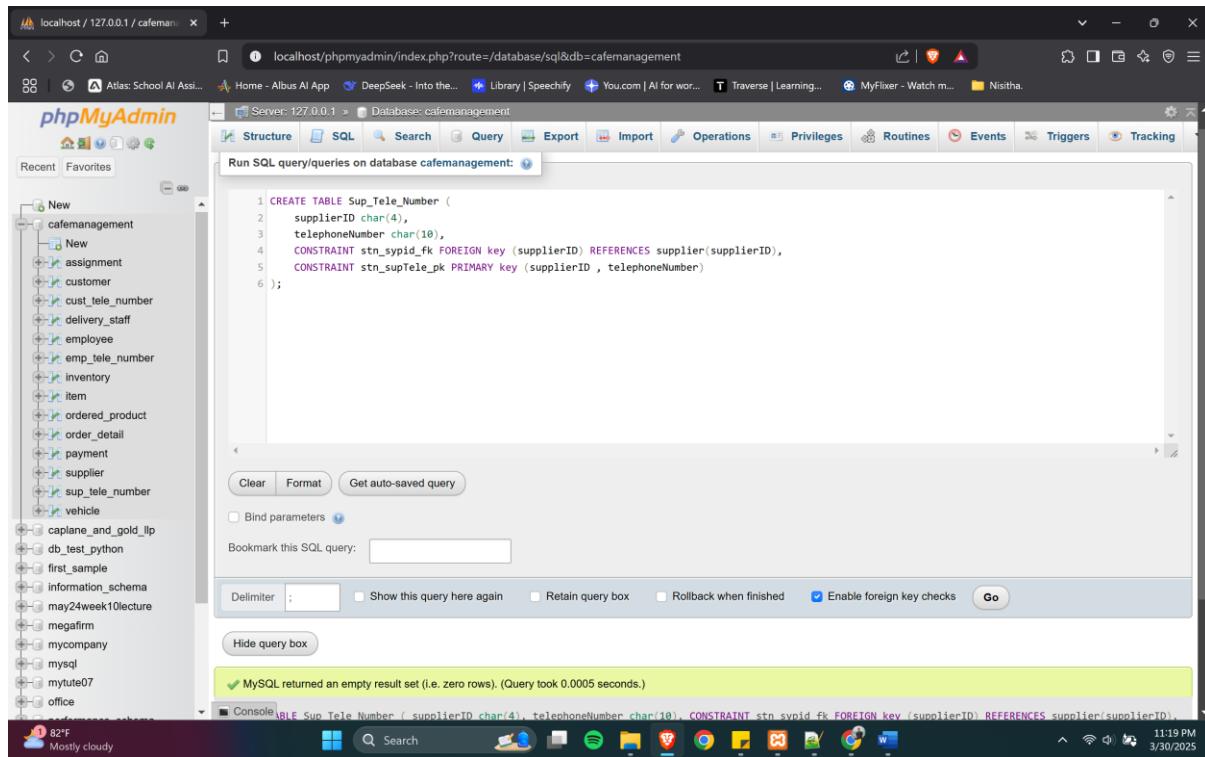


Figure 18 - SupTeleNumber Table Creation

### 8.1.17 Create SourceDetail Table

```
CREATE TABLE source_detail (
    itemID char(4),
    supplierID char(4),
    CONSTRAINT sode_itmid_fk FOREIGN key (itemID) REFERENCES item(itemID),
    CONSTRAINT sode_supid_fk FOREIGN key (supplierID) REFERENCES
    supplier(supplierID),
    CONSTRAINT sode_sordelid_pk PRIMARY key (itemID ,supplierID)
);
```

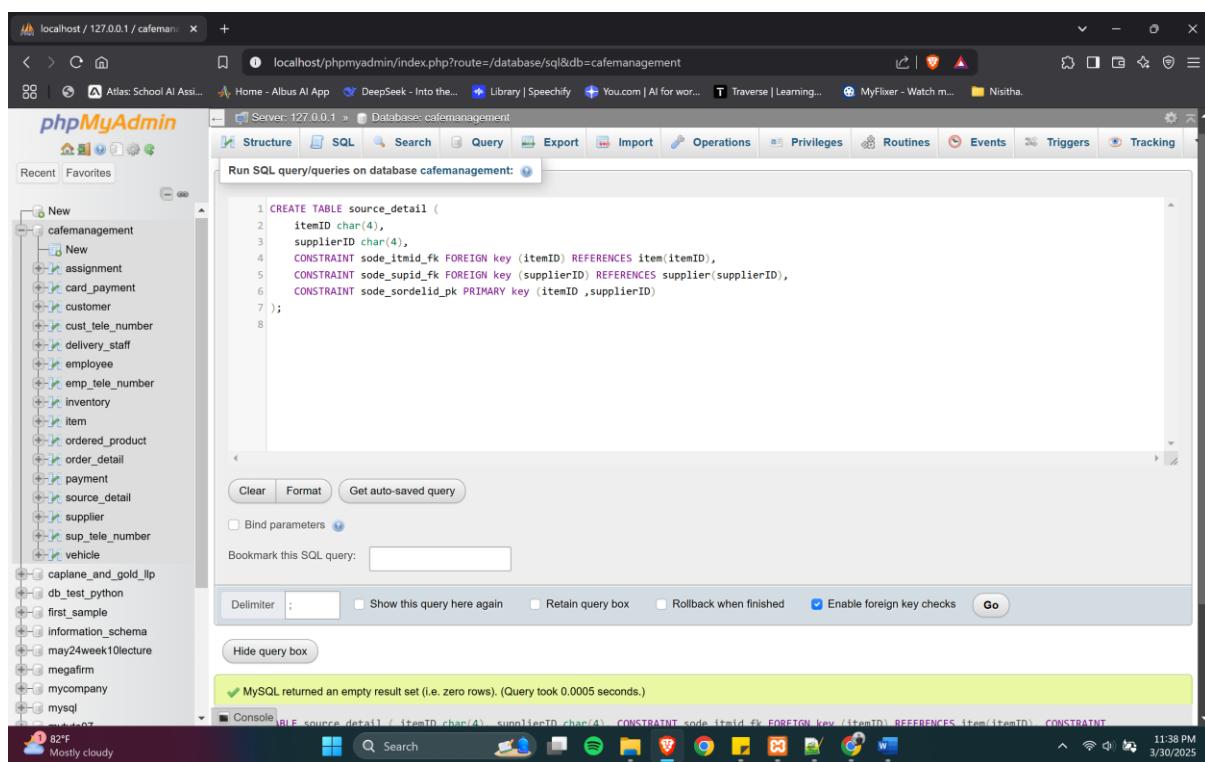


Figure 19 - SourceDetail Table Creation

## 8.2 Population of Data

### 8.2.1 Insert data into Customer Table

```
INSERT into customer
VALUES
('C001', 'Bruce', 'Wayne', '1975-02-19', 'H101', 'Gotham St', 'Gotham', 'bruce.wayne@example.com', '2023-01-15'),
('C002', 'Clark', 'Kent', '1980-06-18', 'H102', 'Metropolis Ave', 'Metropolis', 'clark.kent@example.com', '2023-02-10'),
('C003', 'Diana', 'Prince', '1985-03-22', 'H103', 'Amazon Rd', 'Themyscira', 'diana.prince@example.com', '2023-03-05'),
('C004', 'Barry', 'Allen', '1992-07-12', 'H104', 'Central Blvd', 'Central City', 'barry.allen@example.com', '2023-04-20'),
('C005', 'Hal', 'Jordan', '1983-11-11', 'H105', 'Oa St', 'Coast City', 'hal.jordan@example.com', '2023-05-30'),
('C006', 'Arthur', 'Curry', '1988-09-05', 'H106', 'Atlantis Ln', 'Atlantis', 'arthur.curry@example.com', '2023-06-25'),
('C007', 'Victor', 'Stone', '1995-12-02', 'H107', 'Titan Tower', 'Jump City', 'victor.stone@example.com', '2023-07-10'),
('C008', 'Oliver', 'Queen', '1981-08-29', 'H108', 'Star St', 'Star City', 'oliver.queen@example.com', '2023-08-15'),
('C009', 'Selina', 'Kyle', '1987-04-17', 'H109', 'Shadow Alley', 'Gotham', 'selina.kyle@example.com', '2023-09-05'),
('C010', 'Billy', 'Batson', '2005-10-14', 'H110', 'Rock of Eternity', 'Fawcett City', 'billy.batson@example.com', '2023-10-20')
; 
```

		customerID	firstName	lastName	dateOfBirth	houseID	street	city	emailAddress	registrationDate					
<input type="checkbox"/>		Edit		Copy		Delete	C001	Bruce	Wayne	1975-02-19	H101	Gotham St	Gotham	bruce.wayne@example.com	2023-01-15
<input type="checkbox"/>		Edit		Copy		Delete	C002	Clark	Kent	1980-06-18	H102	Metropolis Ave	Metropolis	clark.kent@example.com	2023-02-10
<input type="checkbox"/>		Edit		Copy		Delete	C003	Diana	Prince	1985-03-22	H103	Amazon Rd	Themyscira	diana.prince@example.com	2023-03-05
<input type="checkbox"/>		Edit		Copy		Delete	C004	Barry	Allen	1992-07-12	H104	Central Blvd	Central City	barry.allen@example.com	2023-04-20
<input type="checkbox"/>		Edit		Copy		Delete	C005	Hal	Jordan	1983-11-11	H105	Oa St	Coast City	hal.jordan@example.com	2023-05-30
<input type="checkbox"/>		Edit		Copy		Delete	C006	Arthur	Curry	1988-09-05	H106	Atlantis Ln	Atlantis	arthur.curry@example.com	2023-06-25
<input type="checkbox"/>		Edit		Copy		Delete	C007	Victor	Stone	1995-12-02	H107	Titan Tower	Jump City	victor.stone@example.com	2023-07-10
<input type="checkbox"/>		Edit		Copy		Delete	C008	Oliver	Queen	1981-08-29	H108	Star St	Star City	oliver.queen@example.com	2023-08-15
<input type="checkbox"/>		Edit		Copy		Delete	C009	Selina	Kyle	1987-04-17	H109	Shadow Alley	Gotham	selina.kyle@example.com	2023-09-05
<input type="checkbox"/>		Edit		Copy		Delete	C010	Billy	Batson	2005-10-14	H110	Rock of Eternity	Fawcett City	billy.batson@example.com	2023-10-20

Figure 20 - insert data in to Customer Table

### 8.2.2 Insert data into CustTeleNumbr table

```
INSERT into cust_tele_number  
VALUES  
('C001', '0771111111'),  
('C002', '0742222222'),  
('C002', '0763333333'),  
('C001', '0701231232'),  
('C005', '0773666663'),  
('C006', '0752323236'),  
('C007', '0773548979'),  
('C008', '0785456453'),  
('C009', '0724999876')  
;
```

		customerID	telephoneNumber
<input type="checkbox"/>	Edit  Copy  Delete	C001	0701231232
<input type="checkbox"/>	Edit  Copy  Delete	C001	0771111111
<input type="checkbox"/>	Edit  Copy  Delete	C002	0742222222
<input type="checkbox"/>	Edit  Copy  Delete	C002	0763333333
<input type="checkbox"/>	Edit  Copy  Delete	C005	0773666663
<input type="checkbox"/>	Edit  Copy  Delete	C006	0752323236
<input type="checkbox"/>	Edit  Copy  Delete	C007	0773548979
<input type="checkbox"/>	Edit  Copy  Delete	C008	0785456453
<input type="checkbox"/>	Edit  Copy  Delete	C009	0724999876

Figure 21 - insert data in to CustTeleNumber Table

### 8.2.3 Insert data into OrderDetail table

```

INSERT INTO order_detail
VALUES
('O001', '2025-02-20', '14:30:00', '2025-03-22', '10:00:00', 'T001', 'Pending', 'C001'),
('O002', '2025-02-21', '09:15:00', '2025-03-23', '12:30:00', 'T002', 'Completed', 'C003'),
('O003', '2025-03-22', '18:45:00', '2025-03-24', '14:00:00', 'T003', 'Processing', 'C005'),
('O004', '2025-03-23', '11:00:00', '2025-03-25', '16:30:00', 'T004', 'Completed', 'C002'),
('O005', '2025-03-24', '20:20:00', '2025-03-26', '09:45:00', 'T005', 'Completed', 'C004'),
('O006', '2025-03-21', '16:45:00', '2025-03-26', '12:30:00', 'T006', 'Processing', 'C006'),
('O007', '2025-03-22', '10:15:00', '2025-03-27', '09:00:00', 'T007', 'Completed', 'C009'),
('O008', '2025-03-23', '18:00:00', '2025-03-27', '15:45:00', 'T008', 'Completed', 'C007'),
('O009', '2025-03-24', '09:30:00', '2025-03-29', '08:15:00', 'T009', 'Processing', 'C008'),
('O010', '2025-03-25', '13:20:00', '2025-03-29', '11:00:00', 'T010', 'Completed', 'C010'),
('O011', '2025-03-26', '08:45:00', '2025-03-30', '10:30:00', 'T011', 'Pending', 'C002'),
('O012', '2025-03-27', '15:10:00', '2025-03-31', '12:45:00', 'T012', 'Completed', 'C005'),
('O013', '2025-03-28', '11:55:00', '2025-04-01', '14:15:00', 'T013', 'Processing', 'C001'),
('O014', '2025-03-29', '19:30:00', '2025-04-02', '16:00:00', 'T014', 'Completed', 'C004'),
('O015', '2025-03-30', '07:20:00', '2025-04-03', '09:30:00', 'T015', 'Cancelled', 'C003'),
('O016', '2025-03-31', '17:40:00', '2025-04-04', '13:00:00', 'T016', 'Completed', 'C006'),
('O017', '2025-04-01', '14:25:00', '2025-04-05', '08:45:00', 'T017', 'Completed', 'C007'),
('O018', '2025-04-02', '09:50:00', '2025-04-06', '16:30:00', 'T018', 'Completed', 'C008'),
('O019', '2025-04-03', '12:10:00', '2025-04-07', '15:20:00', 'T019', 'Pending', 'C009'),
('O020', '2025-04-04', '18:35:00', '2025-04-08', '11:15:00', 'T020', 'Completed', 'C010')
;

```

	orderID	orderDate	orderTime	preferredDeliveryDate	preferredDeliveryTime	traceID	orderStatus	customerID
	Edit	Copy	Delete					
<input type="checkbox"/>	O001	2025-02-20	14:30:00	2025-03-22	10:00:00	T001	Pending	C001
<input type="checkbox"/>	O002	2025-02-21	09:15:00	2025-03-23	12:30:00	T002	Completed	C003
<input type="checkbox"/>	O003	2025-03-22	18:45:00	2025-03-24	14:00:00	T003	Processing	C005
<input type="checkbox"/>	O004	2025-03-23	11:00:00	2025-03-25	16:30:00	T004	Completed	C002
<input type="checkbox"/>	O005	2025-03-24	20:20:00	2025-03-26	09:45:00	T005	Completed	C004
<input type="checkbox"/>	O006	2025-03-21	16:45:00	2025-03-26	12:30:00	T006	Processing	C006
<input type="checkbox"/>	O007	2025-03-22	10:15:00	2025-03-27	09:00:00	T007	Completed	C009
<input type="checkbox"/>	O008	2025-03-23	18:00:00	2025-03-27	15:45:00	T008	Completed	C007
<input type="checkbox"/>	O009	2025-03-24	09:30:00	2025-03-29	08:15:00	T009	Processing	C008
<input type="checkbox"/>	O010	2025-03-25	13:20:00	2025-03-29	11:00:00	T010	Completed	C010
<input type="checkbox"/>	O011	2025-03-26	08:45:00	2025-03-30	10:30:00	T011	Pending	C002
<input type="checkbox"/>	O012	2025-03-27	15:10:00	2025-03-31	12:45:00	T012	Completed	C005
<input type="checkbox"/>	O013	2025-03-28	11:55:00	2025-04-01	14:15:00	T013	Processing	C001
<input type="checkbox"/>	O014	2025-03-29	19:30:00	2025-04-02	16:00:00	T014	Completed	C004
<input type="checkbox"/>	O015	2025-03-30	07:20:00	2025-04-03	09:30:00	T015	Cancelled	C003
<input type="checkbox"/>	O016	2025-03-31	17:40:00	2025-04-04	13:00:00	T016	Completed	C006
<input type="checkbox"/>	O017	2025-04-01	14:25:00	2025-04-05	08:45:00	T017	Completed	C007
<input type="checkbox"/>	O018	2025-04-02	09:50:00	2025-04-06	16:30:00	T018	Completed	C008
<input type="checkbox"/>	O019	2025-04-03	12:10:00	2025-04-07	15:20:00	T019	Pending	C009
<input type="checkbox"/>	O020	2025-04-04	18:35:00	2025-04-08	11:15:00	T020	Completed	C010

Figure 22 - insert data in to OrderDetail Table

## 8.2.4 Insert data into Payment table

INSERT INTO payment

VALUES

```
('P001', 'Credit Card', 'Completed', 28.25, '2025-03-20', '14:35:00', 'C003', 'O002'),  
('P002', 'Cash on delivery', 'Completed', 27, '2025-03-21', '09:20:00', 'C002', 'O004'),  
('P003', 'Debit Card', 'Completed', 13.5, '2025-03-22', '18:50:00', 'C004', 'O005'),  
('P004', 'Cash on delivery', 'Completed', 15.8, '2025-03-23', '11:05:00', 'C009', 'O007'),  
('P005', 'Credit Card', 'Completed', 13.15, '2025-03-24', '20:25:00', 'C007', 'O008'),  
('P006', 'Debit Card', 'Completed', 16, '2025-03-25', '16:45:00', 'C010', 'O010'),  
('P007', 'Cash on delivery', 'Completed', 30, '2025-03-26', '10:30:00', 'C005', 'O012'),  
('P008', 'Debit Card', 'Completed', 17, '2025-03-27', '18:10:00', 'C004', 'O014'),  
('P009', 'credit Card', 'Refunded', 14.1, '2025-03-28', '09:45:00', 'C003', 'O015'),  
('P010', 'Cash on delivery', 'Completed', 14.5, '2025-03-29', '13:50:00', 'C006', 'O016'),  
('P011', 'Credit Card', 'Completed', 30.3, '2025-03-30', '08:20:00', 'C007', 'O017'),  
('P012', 'Cash on delivery', 'Completed', 12.9, '2025-03-31', '12:15:00', 'C008', 'O018'),  
('P013', 'Cash on delivery', 'Completed', 10, '2025-04-01', '02:12:34', 'C010', 'O020')  
;
```

	paymentID	paymentMethod	paymentStatus	amountPaid	paymentDate	paymenttime	customerID	orderID
<input type="checkbox"/>	P001	Credit Card	Completed	28.25	2025-03-20	14:35:00	C003	O002
<input type="checkbox"/>	P002	Cash on delivery	Completed	27.00	2025-03-21	09:20:00	C002	O004
<input type="checkbox"/>	P003	Debit Card	Completed	13.50	2025-03-22	18:50:00	C004	O005
<input type="checkbox"/>	P004	Cash on delivery	Completed	15.80	2025-03-23	11:05:00	C009	O007
<input type="checkbox"/>	P005	Credit Card	Completed	13.15	2025-03-24	20:25:00	C007	O008
<input type="checkbox"/>	P006	Debit Card	Completed	16.00	2025-03-25	16:45:00	C010	O010
<input type="checkbox"/>	P007	Cash on delivery	Completed	30.00	2025-03-26	10:30:00	C005	O012
<input type="checkbox"/>	P008	Credit Card	Completed	17.00	2025-03-27	18:10:00	C004	O014
<input type="checkbox"/>	P009	credit Card	Refunded	14.10	2025-03-28	09:45:00	C003	O015
<input type="checkbox"/>	P010	Cash on delivery	Completed	14.50	2025-03-29	13:50:00	C006	O016
<input type="checkbox"/>	P011	Credit Card	Completed	30.30	2025-03-30	08:20:00	C007	O017
<input type="checkbox"/>	P012	Cash on delivery	Completed	12.90	2025-03-31	12:15:00	C008	O018
<input type="checkbox"/>	P013	Cash on delivery	Completed	10.00	2025-04-01	02:12:34	C010	O020

Figure 23 - insert data in to Payment Table

### 8.2.5 Insert data into CardPayment table

```
insert into Card_Payment  
VALUES  
('P001', 'Visa', 'Diana Prince', '4111 1111 1111 1111', '12/27'),  
('P003', 'MasterCard', 'Barry Allen', '5500 0000 0000 0004', '06/26'),  
('P005', 'Visa', 'Victor Stone', '3400 0000 0000 0096', '09/28'),  
('P006', 'MasterCard', 'Billy Batson', '1234 6543 0988 3345', '12/06'),  
('P008', 'MasterCard', 'Barry Allen', '5500 0000 0000 0004', '06/26'),  
('P009', 'Visa', 'Diana Prince', '4111 1111 1111 1111', '12/27'),  
('P011', 'Visa', 'Victor Stone', '3400 0000 0000 0096', '09/28')  
;
```

	paymentID	cardtype	ownerName	cardNumber	expireDate
<input type="checkbox"/>	Edit  Copy  Delete P001	Visa	Diana Prince	4111 1111 1111 1111	12/27
<input type="checkbox"/>	Edit  Copy  Delete P003	MasterCard	Barry Allen	5500 0000 0000 0004	06/26
<input type="checkbox"/>	Edit  Copy  Delete P005	Visa	Victor Stone	3400 0000 0000 0096	09/28
<input type="checkbox"/>	Edit  Copy  Delete P006	MasterCard	Billy Batson	1234 6543 0988 3345	12/06
<input type="checkbox"/>	Edit  Copy  Delete P008	MasterCard	Barry Allen	5500 0000 0000 0004	06/26
<input type="checkbox"/>	Edit  Copy  Delete P009	Visa	Diana Prince	4111 1111 1111 1111	12/27
<input type="checkbox"/>	Edit  Copy  Delete P011	Visa	Victor Stone	3400 0000 0000 0096	09/28

Figure 24 - insert data in to CardPayment Table

## 8.2.6 Insert data into Employee table

```
insert into Employee
VALUES
('E001', 'Manager', 'Tony', 'Stark', '1970-05-29', '201', 'Avengers Tower', 'New York',
'tony.stark@example.com', NULL),
('E002', 'Manager', 'Steve', 'Rogers', '1918-07-04', '202', 'Brooklyn St', 'New York',
'steve.rogers@example.com', Null),
('E003', 'Steward', 'Nicola', 'Romanoff', '1984-11-22', '203', 'Red Room Blvd', 'Moscow',
'Nicola.romanoff@example.com', 'E001'),
('E004', 'Driver', 'Peter', 'Parker', '2001-08-10', '204', 'Queens Ave', 'New York',
'peter.parker@example.com', 'E001'),
('E005', 'Steward', 'Bruce', 'Banner', '1969-12-18', '205', 'Gamma St', 'Dayton',
'bruce.banner@example.com', 'E002'),
('E006', 'Steward', 'John', 'Odinson', '1500-01-01', '106', 'Asgardian Rd', 'Asgard',
'john.odinson@example.com', 'E002'),
('E007', 'Steward', 'Peter', 'Barton', '1980-01-07', '107', 'Sharpshooter Ln', 'Iowa',
'peter.barton@example.com', 'E001'),
('E008', 'Driver', 'Wanda', 'Maximoff', '1989-02-10', '208', 'Hex St', 'Sokovia',
'wanda.maximoff@example.com', 'E002'),
('E009', 'Driver', 'T'Challa', 'Udaku', '1980-05-21', '209', 'Royal Palace', 'Wakanda',
'tchalla.udaku@example.com', 'E006'),
('E010', 'Driver', 'Stephen', 'Strange', '1975-11-18', '210', 'Sanctum St', 'New York',
'stephen.strange@example.com', 'E006')
;
```

	<input type="checkbox"/> Edit <input type="button" value="Copy"/> <input type="button" value="Delete"/>	employeeID	designation	firstName	lastName	dateOfBirth	houseID	street	city	emailAddress	supervisorID
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E001	Manager	Tony	Stark	1970-05-29	201	Avengers Tower	New York	tony.stark@example.com	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E002	Manager	Steve	Rogers	1918-07-04	202	Brooklyn St	New York	steve.rogers@example.com	NULL
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E003	Steward	Nicola	Romanoff	1984-11-22	203	Red Room Blvd	Moscow	Nicola.romanoff@example.com	E001
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E004	Driver	Peter	Parker	2001-08-10	204	Queens Ave	New York	peter.parker@example.com	E001
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E005	Steward	Bruce	Banner	1969-12-18	205	Gamma St	Dayton	bruce.banner@example.com	E002
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E006	Steward	John	Odinson	1500-01-01	106	Asgardian Rd	Asgard	john.odinson@example.com	E002
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E007	Steward	Peter	Barton	1980-01-07	107	Sharpshooter Ln	Iowa	peter.barton@example.com	E001
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E008	Driver	Wanda	Maximoff	1989-02-10	208	Hex St	Sokovia	wanda.maximoff@example.com	E002
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E009	Driver	T'Challa	Udaku	1980-05-21	209	Royal Palace	Wakanda	tchalla.udaku@example.com	E006
<input type="checkbox"/>	<input type="button" value="Edit"/> <input type="button" value="Copy"/> <input type="button" value="Delete"/>	E010	Driver	Stephen	Strange	1975-11-18	210	Sanctum St	New York	stephen.strange@example.com	E006

Figure 25 - insert data in to Employee Table

### 8.2.7 Insert data into EmpTeleNumber table

```
insert into Emp_Tele_Number
VALUES
('E001', '0701234567'),
('E002', '0777654321'),
('E003', '0762334455'),
('E004', '0746778899'),
('E005', '0757889900'),
('E006', '0773445566'),
('E007', '0754556677'),
('E008', '0725667788'),
('E001', '0716778890'),
('E002', '0747889901'),
('E009', '0768765487'),
('E010', '0712345634')
; 
```

	employeeID	telephoneNumber
<input type="checkbox"/>	Edit  Copy  Delete E001	0701234567
<input type="checkbox"/>	Edit  Copy  Delete E001	0716778890
<input type="checkbox"/>	Edit  Copy  Delete E002	0747889901
<input type="checkbox"/>	Edit  Copy  Delete E002	0777654321
<input type="checkbox"/>	Edit  Copy  Delete E003	0762334455
<input type="checkbox"/>	Edit  Copy  Delete E004	0746778899
<input type="checkbox"/>	Edit  Copy  Delete E005	0757889900
<input type="checkbox"/>	Edit  Copy  Delete E006	0773445566
<input type="checkbox"/>	Edit  Copy  Delete E007	0754556677
<input type="checkbox"/>	Edit  Copy  Delete E008	0725667788
<input type="checkbox"/>	Edit  Copy  Delete E009	0768765487
<input type="checkbox"/>	Edit  Copy  Delete E010	0712345634

Figure 26 - insert data in to EmpTeleNumber Table

### 8.2.8 Insert data into DeliveryStaff table

```
insert into delivery_Staff  
VALUES  
(E004', 'DL12345678'),  
(E008', 'DL87654321'),  
(E009', 'DL11223344'),  
(E010', 'DL87656735')  
;
```

	<input type="button" value="←"/>	<input type="button" value="→"/>		employeeID	drivingLicenseNumber
<input type="checkbox"/>				E009	DL11223344
<input type="checkbox"/>				E004	DL12345678
<input type="checkbox"/>				E008	DL87654321
<input type="checkbox"/>				E010	DL87656735

Figure 27 - insert data in to DeliveryStaff Table

## 8.2.9 Insert data into Vehicle table

insert into Vehicle

VALUES

```
('V001', 'Bike', 'Honda CB125F', 'ABC123XYZ', 'Petrol', 15.00, 'Active', 'E004'),  
('V002', 'Minivan', 'Toyota HiAce', 'DEF456LMN', 'Diesel', 1200.50, 'Active', 'E004'),  
('V003', 'Truck', 'Ford Transit', 'GHI789OPQ', 'Diesel', 2500.75, 'Inactive', 'E010'),  
('V004', 'Bike', 'Yamaha YBR125', 'JKL012RST', 'Petrol', 18.00, 'Active', 'E008'),  
('V005', 'Minivan', 'Mercedes-Benz Vito', 'MNO345UVW', 'Electric', 1100.00, 'Under Maintenance',  
'E008'),  
('V006', 'Scooter', 'Vespa Primavera', 'PQR678XYZ', 'Petrol', 12.50, 'Active', 'E009')  
;
```

	vehicleID	vehicleType	vehicleModel	licensePlateNumber	fuelType	cargoCapacity	vehicleStatus	driverID	
<input type="checkbox"/>	Edit  Copy  Delete	V001	Bike	Honda CB125F	ABC123XYZ	Petrol	15.00	Active	E004
<input type="checkbox"/>	Edit  Copy  Delete	V002	Minivan	Toyota HiAce	DEF456LMN	Diesel	1200.50	Active	E004
<input type="checkbox"/>	Edit  Copy  Delete	V003	Truck	Ford Transit	GHI789OPQ	Diesel	2500.75	Inactive	E010
<input type="checkbox"/>	Edit  Copy  Delete	V004	Bike	Yamaha YBR125	JKL012RST	Petrol	18.00	Active	E008
<input type="checkbox"/>	Edit  Copy  Delete	V005	Minivan	Mercedes-Benz Vito	MNO345UVW	Electric	1100.00	Under Maintenance	E008
<input type="checkbox"/>	Edit  Copy  Delete	V006	Scooter	Vespa Primavera	PQR678XYZ	Petrol	12.50	Active	E009

Figure 28 - insert data in to Vehicle Table

### 8.2.10 Insert data into Assignment table

insert into Assignment

VALUES

```
('O001', 'E003', 'V001'),  
('O002', 'E005', 'V002'),  
('O003', 'E006', 'V003'),  
('O004', 'E007', 'V004'),  
('O005', 'E003', 'V005'),  
('O006', 'E006', 'V006'),  
('O007', 'E005', 'V002'),  
('O008', 'E007', 'V003'),  
('O009', 'E003', 'V005'),  
('O010', 'E005', 'V001'),  
('O011', 'E007', 'V004'),  
('O012', 'E006', 'V006'),  
('O013', 'E007', 'V002'),  
('O014', 'E005', 'V003'),  
('O015', 'E003', 'V005'),  
('O016', 'E006', 'V002'),  
('O017', 'E007', 'V004'),  
('O018', 'E005', 'V001'),  
('O019', 'E003', 'V006'),  
('O020', 'E006', 'V003')  
;
```

	orderID	StewardID	vehicleID
<input type="checkbox"/>  Edit	O001	E003	V001
<input type="checkbox"/>  Edit	O002	E005	V002
<input type="checkbox"/>  Edit	O003	E006	V003
<input type="checkbox"/>  Edit	O004	E007	V004
<input type="checkbox"/>  Edit	O005	E003	V005
<input type="checkbox"/>  Edit	O006	E006	V006
<input type="checkbox"/>  Edit	O007	E005	V002
<input type="checkbox"/>  Edit	O008	E007	V003
<input type="checkbox"/>  Edit	O009	E003	V005
<input type="checkbox"/>  Edit	O010	E005	V001
<input type="checkbox"/>  Edit	O011	E007	V004
<input type="checkbox"/>  Edit	O012	E006	V006
<input type="checkbox"/>  Edit	O013	E007	V002
<input type="checkbox"/>  Edit	O014	E005	V003
<input type="checkbox"/>  Edit	O015	E003	V005
<input type="checkbox"/>  Edit	O016	E006	V002
<input type="checkbox"/>  Edit	O017	E007	V004
<input type="checkbox"/>  Edit	O018	E005	V001
<input type="checkbox"/>  Edit	O019	E003	V006
<input type="checkbox"/>  Edit	O020	E006	V003

Figure 29 - insert data in to Assignment Table

### 8.2.11 Insert data into Inventory table

```
insert into Inventory  
VALUES  
(IN01', 150.75, 50.00, 'E001'),  
(IN02', 320.50, 100.00, 'E001'),  
(IN03', 75.25, 30.00, 'E002'),  
(IN04', 200.00, 80.00, 'E002'),  
(IN05', 90.60, 40.00, 'E002')  
;
```

	inventoryID	stocklevel	reloadLevel	managerID
<input type="checkbox"/>	IN01	150.75	50.00	E001
<input type="checkbox"/>	IN02	320.50	100.00	E001
<input type="checkbox"/>	IN03	75.25	30.00	E002
<input type="checkbox"/>	IN04	200.00	80.00	E002
<input type="checkbox"/>	IN05	90.60	40.00	E002

Figure 30 - insert data in to Inventory Table

### 8.2.12 Insert data into Item table

insert into Item

VALUES

```
(I001, 'Espresso', 3.50, 'Beverage', 'Strong and rich coffee shot.', 'Available', 'IN01'),
(I002, 'Cappuccino', 4.20, 'Beverage', 'Espresso with steamed milk and foam.', 'Available', 'IN01'),
(I003, 'Blueberry Muffin', 2.80, 'Snack', 'Soft muffin with fresh blueberries.', 'Available', 'IN02'),
(I004, 'Chocolate Cake', 5.50, 'Cake', 'Rich and moist chocolate cake.', 'Available', 'IN03'),
(I005, 'Tiramisu', 6.75, 'Dessert', 'Classic Italian coffee-flavored dessert.', 'Available', 'IN02'),
(I006, 'Cheesecake', 6.00, 'Dessert', 'Creamy cheesecake with a biscuit base.', 'Out of Stock', 'IN02'),
(I007, 'Macarons', 4.50, 'Specialty', 'Colorful and delicate almond meringue cookies.', 'Available',
'IN02'),
(I008, 'Chai Latte', 4.00, 'Beverage', 'Spiced tea with steamed milk.', 'Available', 'IN03'),
(I009, 'Americano', 3.00, 'Beverage', 'Espresso diluted with hot water.', 'Available', 'IN04'),
(I010, 'Croissant', 2.50, 'Snack', 'Flaky and buttery French pastry.', 'Available', 'IN04'),
(I011, 'Red Velvet Cake', 5.80, 'Cake', 'Moist red velvet cake with cream cheese frosting.',
'Available', 'IN05'),
(I012, 'Brownie', 3.25, 'Dessert', 'Rich and fudgy chocolate brownie.', 'Available', 'IN04'),
(I013, 'Matcha Latte', 4.75, 'Beverage', 'Green tea latte with steamed milk.', 'Out of Stock', 'IN05')
;
```

	<input type="checkbox"/> Edit  Copy  Delete	itemID	name	price	category	description	availability	inventoryID
<input type="checkbox"/>	Edit  Copy  Delete	I001	Espresso	3.50	Beverage	Strong and rich coffee shot.	Available	IN01
<input type="checkbox"/>	Edit  Copy  Delete	I002	Cappuccino	4.20	Beverage	Espresso with steamed milk and foam.	Available	IN01
<input type="checkbox"/>	Edit  Copy  Delete	I003	Blueberry Muffin	2.80	Snack	Soft muffin with fresh blueberries.	Available	IN02
<input type="checkbox"/>	Edit  Copy  Delete	I004	Chocolate Cake	5.50	Cake	Rich and moist chocolate cake.	Available	IN03
<input type="checkbox"/>	Edit  Copy  Delete	I005	Tiramisu	6.75	Dessert	Classic Italian coffee-flavored dessert.	Available	IN02
<input type="checkbox"/>	Edit  Copy  Delete	I006	Cheesecake	6.00	Dessert	Creamy cheesecake with a biscuit base.	Out of Stock	IN02
<input type="checkbox"/>	Edit  Copy  Delete	I007	Macarons	4.50	Specialty	Colorful and delicate almond meringue cookies.	Available	IN02
<input type="checkbox"/>	Edit  Copy  Delete	I008	Chai Latte	4.00	Beverage	Spiced tea with steamed milk.	Available	IN03
<input type="checkbox"/>	Edit  Copy  Delete	I009	Americano	3.00	Beverage	Espresso diluted with hot water.	Available	IN04
<input type="checkbox"/>	Edit  Copy  Delete	I010	Croissant	2.50	Snack	Flaky and buttery French pastry.	Available	IN04
<input type="checkbox"/>	Edit  Copy  Delete	I011	Red Velvet Cake	5.80	Cake	Moist red velvet cake with cream cheese frosting.	Available	IN05
<input type="checkbox"/>	Edit  Copy  Delete	I012	Brownie	3.25	Dessert	Rich and fudgy chocolate brownie.	Available	IN04
<input type="checkbox"/>	Edit  Copy  Delete	I013	Matcha Latte	4.75	Beverage	Green tea latte with steamed milk.	Out of Stock	IN05

Figure 31 - insert data in to Item Table

### 8.2.13 Insert data into OrderedProduct table

```
INSERT INTO ordered_product
```

```
VALUES
```

```
('O001', 'I001', 2),
('O001', 'I003', 1),
('O002', 'I005', 3),
('O002', 'I008', 2),
('O003', 'I002', 1),
('O003', 'I006', 2),
('O004', 'I007', 4),
('O004', 'I009', 3),
('O005', 'I004', 2),
('O005', 'I010', 1),
('O006', 'I005', 2),
('O006', 'I011', 3),
('O007', 'I003', 1),
('O007', 'I012', 4),
('O008', 'I002', 2),
('O008', 'I013', 1),
('O009', 'I006', 3),
('O009', 'I007', 2),
('O010', 'I008', 1),
('O010', 'I009', 4),
('O011', 'I010', 2),
('O011', 'I011', 1),
('O012', 'I004', 3),
('O012', 'I005', 2),
('O013', 'I006', 1),
('O013', 'I007', 4),
('O014', 'I008', 2),
('O014', 'I009', 3),
('O015', 'I010', 1),
('O015', 'I011', 2),
('O016', 'I012', 3),
('O016', 'I013', 1),
('O017', 'I002', 4),
('O017', 'I005', 2),
('O018', 'I003', 3),
('O018', 'I007', 1),
('O019', 'I006', 2),
('O019', 'I010', 4),
('O020', 'I008', 1),
('O020', 'I009', 2)
```

```
;
```

	<input type="button" value="←"/>	<input type="button" value="→"/>	orderID	itemID	quantity
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O001	I001	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O001	I001	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O002	I005	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O002	I008	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O003	I002	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O003	I003	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O003	I006	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O004	I007	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O004	I009	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O005	I004	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O005	I010	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O006	I005	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O006	I011	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O007	I003	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O007	I012	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O008	I002	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O008	I013	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O009	I006	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O009	I007	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O010	I008	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O010	I009	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O011	I010	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O011	I011	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O012	I004	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O012	I005	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O013	I006	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O013	I007	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O014	I008	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O014	I009	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O015	I010	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O015	I011	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O016	I012	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O016	I013	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O017	I002	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O017	I005	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O018	I003	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O018	I007	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O019	I006	2
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O019	I010	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O020	I008	1
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	O020	I009	2

Figure 32 - insert data in to OrderedProduct Table

### 8.2.14 Insert data into Supplier table

```
insert into supplier  
VALUES  
('S001', 'Alex', 'Luther'),  
('S002', 'Joe', 'Kerr'),  
('S003', 'Harley', 'Quinnley'),  
('S004', 'Selina', 'Keller'),  
('S005', 'Ozzie', 'Cobble'),  
('S006', 'Eddie', 'Nash'),  
('S007', 'Sam', 'Wilson'),  
('S008', 'Ray', 'Gould')  
;
```

			supplierID	firstName	lastName
<input type="checkbox"/>	 Edit	 Copy	 Delete	S001	Alex Luther
<input type="checkbox"/>	 Edit	 Copy	 Delete	S002	Joe Kerr
<input type="checkbox"/>	 Edit	 Copy	 Delete	S003	Harley Quinnley
<input type="checkbox"/>	 Edit	 Copy	 Delete	S004	Selina Keller
<input type="checkbox"/>	 Edit	 Copy	 Delete	S005	Ozzie Cobble
<input type="checkbox"/>	 Edit	 Copy	 Delete	S006	Eddie Nash
<input type="checkbox"/>	 Edit	 Copy	 Delete	S007	Sam Wilson
<input type="checkbox"/>	 Edit	 Copy	 Delete	S008	Ray Gould

Figure 33 - insert data in to Supplier Table

### 8.2.15 Insert data into SupTeleNumber table

```
insert into Sup_Tele_Number  
VALUES  
('S001', '0416543210'),  
('S002', '0435432109'),  
('S003', '0914321098'),  
('S004', '0413210987'),  
('S005', '0912109876'),  
('S006', '0419878533'),  
('S007', '0914534563'),  
('S008', '0417878787')  
;
```

			supplierID	telephoneNumber
<input type="checkbox"/>	 Edit	 Copy	 Delete	S001 0416543210
<input type="checkbox"/>	 Edit	 Copy	 Delete	S002 0435432109
<input type="checkbox"/>	 Edit	 Copy	 Delete	S003 0914321098
<input type="checkbox"/>	 Edit	 Copy	 Delete	S004 0413210987
<input type="checkbox"/>	 Edit	 Copy	 Delete	S005 0912109876
<input type="checkbox"/>	 Edit	 Copy	 Delete	S006 0419878533
<input type="checkbox"/>	 Edit	 Copy	 Delete	S007 0914534563
<input type="checkbox"/>	 Edit	 Copy	 Delete	S008 0417878787

Figure 34 - insert data in to SupTeleNumber Table

### 8.2.16 Insert data into SourceDetail table

```
insert into source_details
```

```
VALUES
```

```
('I001', 'S001'),  
('I002', 'S002'),  
('I003', 'S003'),  
('I004', 'S001'),  
('I005', 'S004'),  
('I006', 'S005'),  
('I007', 'S002'),  
('I008', 'S006'),  
('I009', 'S004'),  
('I010', 'S007'),  
('I011', 'S008'),  
('I012', 'S008'),  
('I013', 'S003')
```

```
;
```

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	itemID	supplierID
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I001	S001
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I002	S002
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I003	S003
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I004	S001
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I005	S004
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I006	S005
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I007	S002
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I008	S006
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I009	S004
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I010	S007
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I011	S008
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I012	S008
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	I013	S003

Figure 35 - insert data in to SourceDetail Table

## 8.3 Designer Diagram

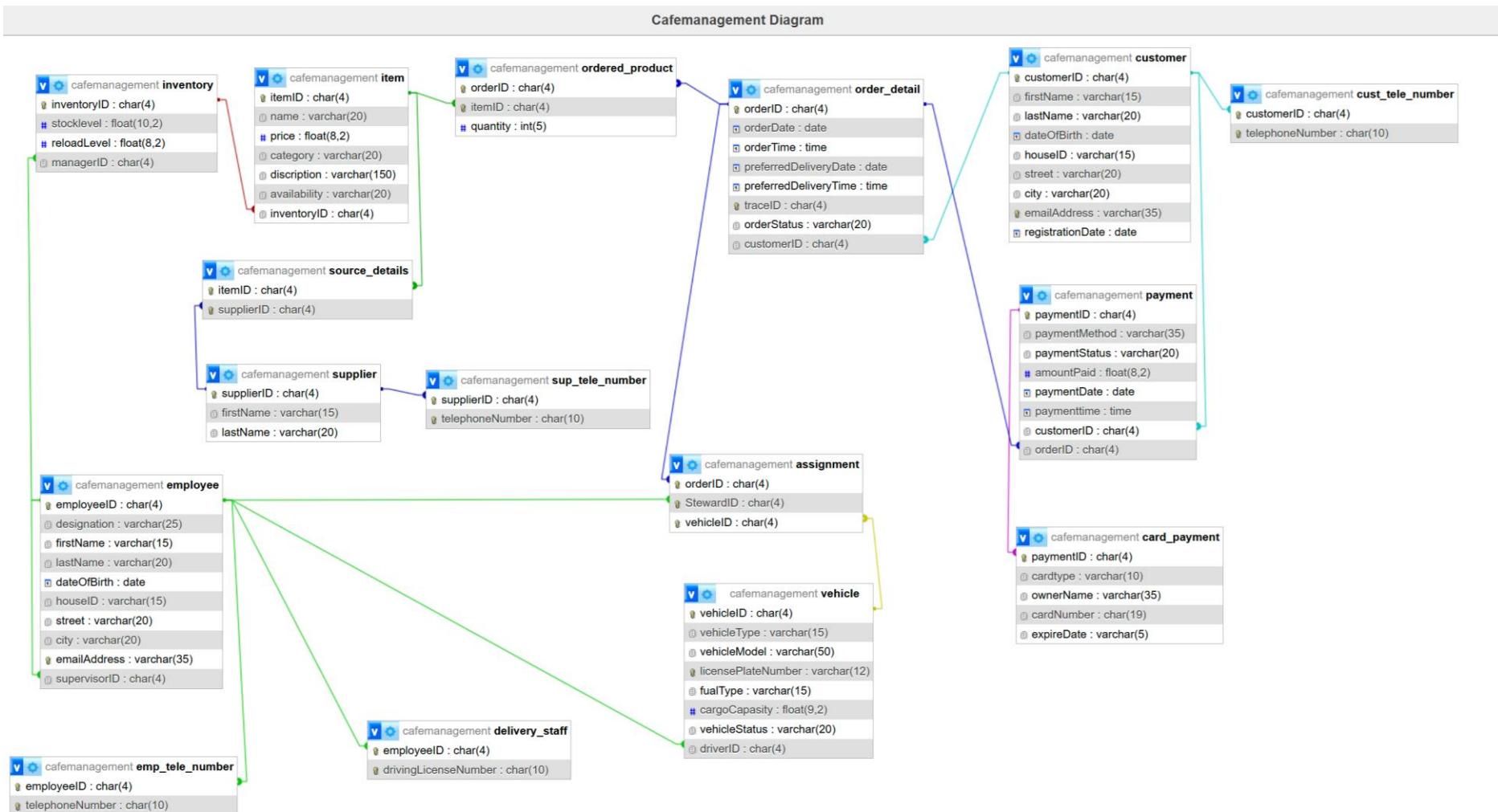


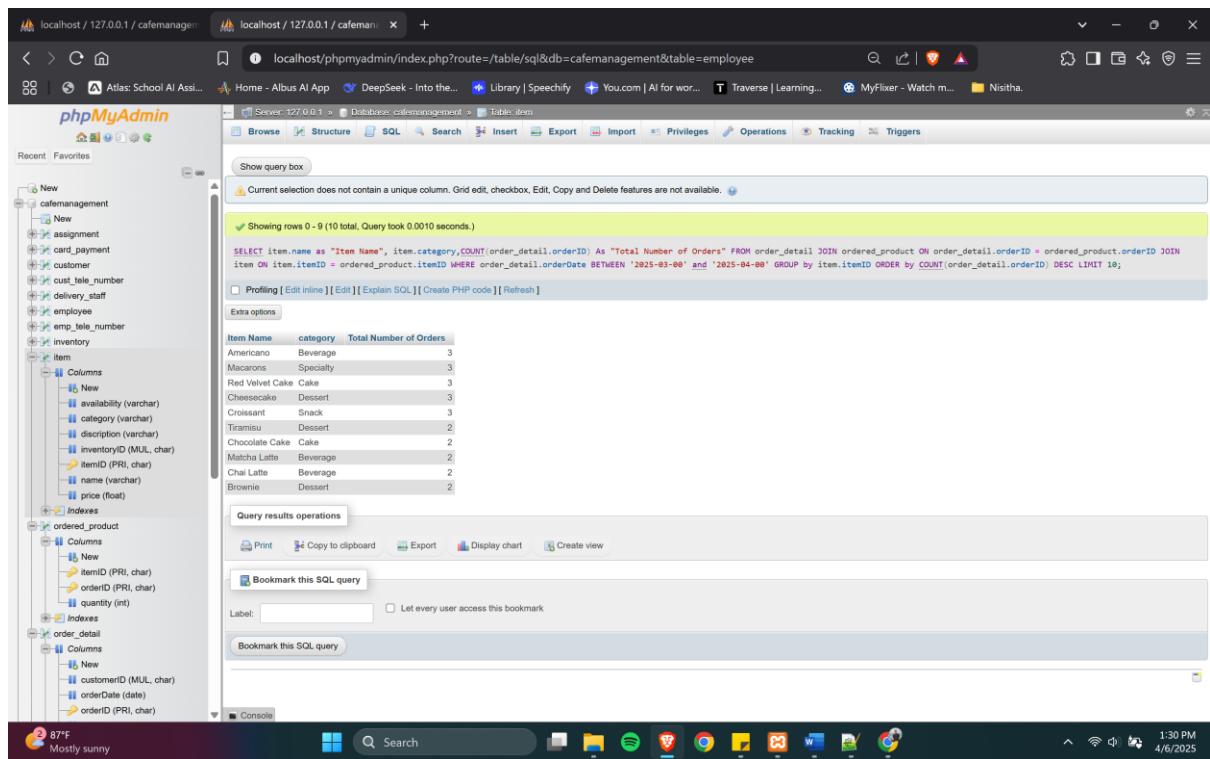
Figure 36 - Designer Diagram

## 9. Section 5: Data Manipulation with SQL

### 9.1 Question - 1

Retrieve the top 10 most frequently ordered items in March 2025, displaying their item name, category, and the total number of orders placed during this period.

```
SELECT item.name as "Item Name", item.category,COUNT(order_detail.orderID) As "Total Number of Orders"
FROM order_detail JOIN ordered_product
ON order_detail.orderID = ordered_product.orderID
JOIN item
ON item.itemID = ordered_product.itemID
WHERE order_detail.orderDate BETWEEN '2025-03-00' and '2025-04-00'
GROUP by item.itemID
ORDER by COUNT(order_detail.orderID) DESC
LIMIT 10;
```



The screenshot shows the phpMyAdmin interface for a database named 'cafemanagement'. The left sidebar lists tables: 'cafemanagement' (with 'New', 'assignment', 'card\_payment', 'customer', 'cust\_tele\_number', 'delivery\_staff', 'employee', 'emp\_tele\_number', 'inventory'), 'item' (with 'Columns' like 'availability', 'category', 'description', 'inventoryID', 'itemName', 'price', and 'Indexes'), and 'ordered\_product' (with 'Columns' like 'itemID', 'orderID', and 'Indexes'). The main area displays the results of the executed SQL query:

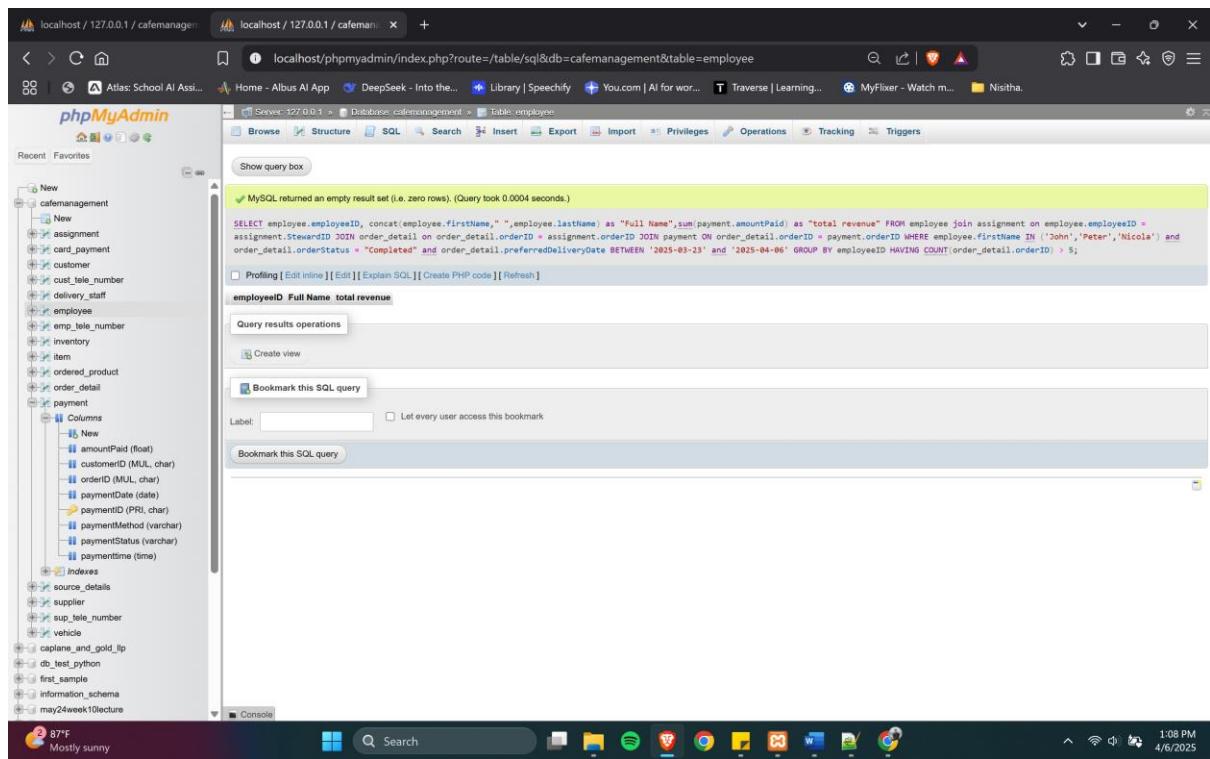
Item Name	category	Total Number of Orders
Americano	Beverage	3
Macarons	Specialty	3
Red Velvet Cake	Cake	3
Cheesecake	Dessert	3
Croissant	Snack	3
Tiramisu	Dessert	2
Chocolate Cake	Cake	2
Mocha Latte	Beverage	2
Chai Latte	Beverage	2
Brownie	Dessert	2

Figure 37 - Question 1 Output

## 9.2 Question - 2

Retrieve the total revenue generated by each steward, along with their staff ID and full name, for stewards with the first name 'John', 'Peter', and 'Nicola', who have successfully completed more than 5 deliveries within the past 2 weeks.

```
SELECT employee.employeeID, concat(employee.firstName, " ", employee.lastName) as "Full Name", sum(payment.amountPaid) as "total revenue"
FROM employee join assignment
on employee.employeeID = assignment.StewardID
JOIN order_detail
on order_detail.orderID = assignment.orderID
JOIN payment
ON order_detail.orderID = payment.orderID
WHERE employee.firstName IN ('John','Peter','Nicola') and order_detail.orderStatus = "Completed"
and order_detail.preferredDeliveryDate BETWEEN '2025-03-23' and '2025-04-06'
GROUP BY employeeID
HAVING COUNT(order_detail.orderID) > 5;
```



The screenshot shows the phpMyAdmin interface on a Windows desktop. The left sidebar displays the database structure for 'cafemanagement'. The main window shows the results of a SQL query:

```
employeeID Full Name total revenue
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)
```

The query executed was:

```
SELECT employee.employeeID, concat(employee.firstName, " ", employee.lastName) as "Full Name", sum(payment.amountPaid) as "total revenue" FROM employee join assignment on employee.employeeID = assignment.StewardID JOIN order_detail on order_detail.orderID = assignment.orderID JOIN payment ON order_detail.orderID = payment.orderID WHERE employee.firstName IN ('John','Peter','Nicola') and order_detail.orderStatus = "Completed" and order_detail.preferredDeliveryDate BETWEEN '2025-03-23' and '2025-04-06' GROUP BY employeeID HAVING COUNT(order_detail.orderID) > 5;
```

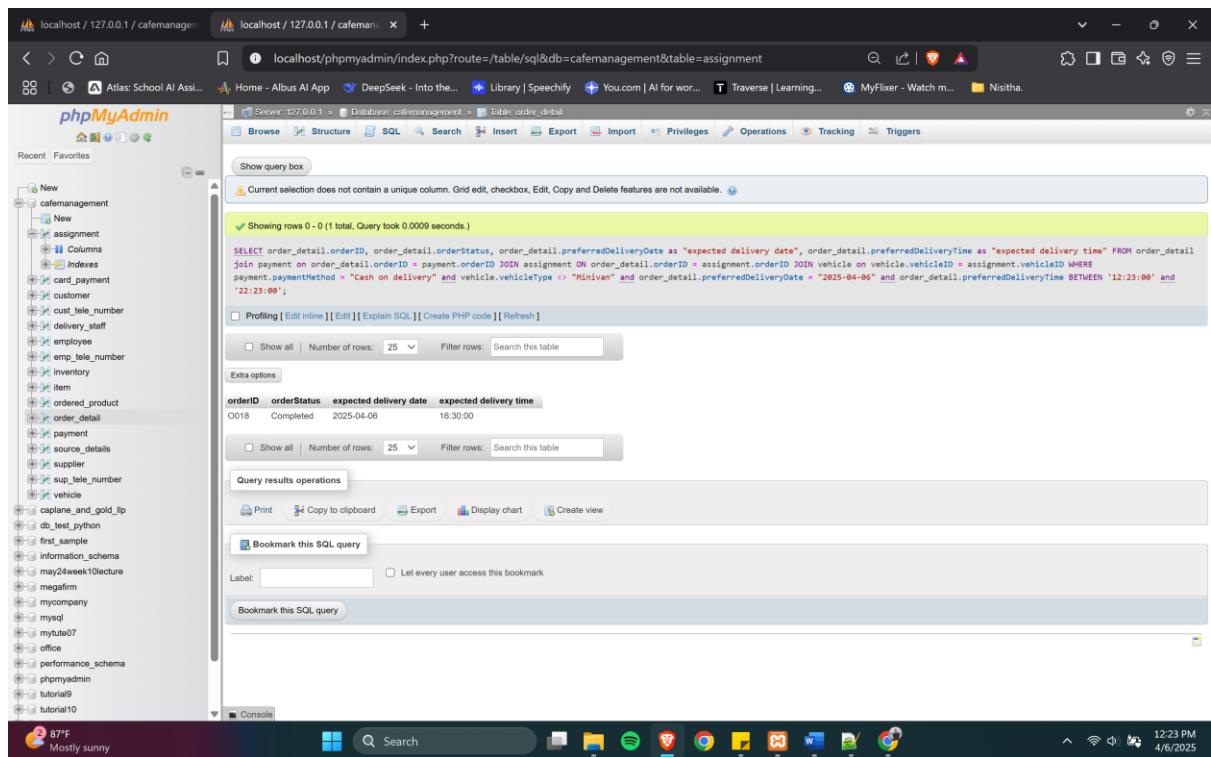
The results table is empty, indicating no matches were found. The system status bar at the bottom shows it's 1:08 PM on 4/6/2025.

Figure 38 - Question 2 Output

### 9.3 Question - 3

Retrieve the order ID, order status, and expected delivery date and time for all cash on delivery orders that are scheduled to be delivered within the next 10 hours and are assigned to vehicles other than minivans.

```
SELECT order_detail.orderID, order_detail.orderStatus, order_detail.preferredDeliveryDate as "expected delivery date", order_detail.preferredDeliveryTime as "expected delivery time"
FROM order_detail join payment
on order_detail.orderID = payment.orderID
JOIN assignment
ON order_detail.orderID = assignment.orderID
JOIN vehicle
on vehicle.vehicleID = assignment.vehicleID
WHERE payment.paymentMethod = "Cash on delivery" and vehicle.vehicleType <> "Minivan" and
concat(order_detail.preferredDeliveryDate, ' ', order_detail.preferredDeliveryTime) BETWEEN
concat("2025-04-06", " ", '12:23:00') and concat("2025-04-06", " ", '22:23:00');
```



The screenshot shows the phpMyAdmin interface with the following details:

- Database:** cafemanagement
- Table:** order\_detail
- Query:**

```
SELECT order_detail.orderID, order_detail.orderStatus, order_detail.preferredDeliveryDate as "expected delivery date", order_detail.preferredDeliveryTime as "expected delivery time"
FROM order_detail join payment
on order_detail.orderID = payment.orderID
JOIN assignment
ON order_detail.orderID = assignment.orderID
JOIN vehicle
on vehicle.vehicleID = assignment.vehicleID
WHERE payment.paymentMethod = "Cash on delivery" and vehicle.vehicleType <> "Minivan" and
concat(order_detail.preferredDeliveryDate, ' ', order_detail.preferredDeliveryTime) BETWEEN
concat("2025-04-06", " ", '12:23:00') and concat("2025-04-06", " ", '22:23:00');
```
- Results:**

orderID	orderStatus	expected delivery date	expected delivery time
0018	Completed	2025-04-06	16:30:00

Figure 39 - Question 3 Output

## 10. Reference

Elmasri, R. and Navathe, B. (2015) *Fundamentals of Database Systems*. 7th edition. England: Pearson.

Connolly, T. and Begg, C. (2015) *Database Systems - A Practical Approach to Design, Implementation and ManagementSystems*. 6th edition. England: Pearson.