**EcoHaven Builders - Luxury Treehouse Homes for Nature Enthusiasts.**

Abbu Turab Syed

GitHub - <https://github.com/SyedAbbuTurab/EcoHaven-Builders>

Windows SQL Environment.

# Business Description:

Our firm, "EcoHaven Builders," is devoted to developing environmentally friendly and sustainable dwelling solutions. We specialize in building luxury treehouse residences for arboreal enthusiasts and nature lovers. These treehouse homes are intended to create a one-of-a-kind living experience in the trees, allowing our clients to interact with nature while still enjoying the conveniences of contemporary living.

Company Size and Locations:

* Estimated Company Size: 50 employees
* Locations: Our primary office is located in Chicago, and we operate in two cities, Denver and Dallas.

# Supply Chain:

## Raw Materials:

1. Sustainably Sourced Timber: We use responsibly harvested timber from certified forests.
2. Recycled Glass: Windows and glass components are made from recycled glass.
3. Solar Panels: We incorporate solar panels for eco-friendly power generation.

## Material Sources:

### Sustainably Sourced Timber:

* Supplier 1: EcoTimber Co. (Location: Austin)
* Supplier 2: GreenWood Harvesters (Location: Lombard)

### Recycled Glass:

* Supplier 1: RenewGlass Inc. (Location: Dallas)
* Supplier 2: GlassCycle Ltd. (Location: Naperville)

### Solar Panels:

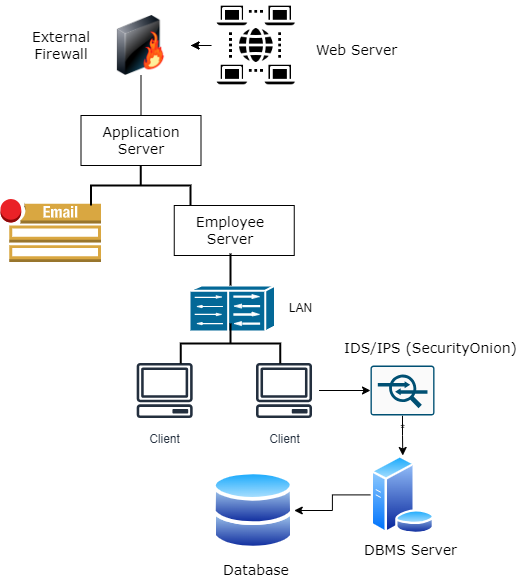
* Supplier 1: SolarTech Solutions (Location: Austin)
* Supplier 2: SunPower Renewables (Location: Florida)

# Primary Office and Operations:

Our primary office is located in Chicago, and we operate in two cities, Denver and Dallas.

# Network Architecture Model:

The network architecture features a Web Server as the client entry point, connected to an External Firewall for security. The Application Server hosts standard applications and APIs, while the Email Server handles email services. An Employee Server connects to a LAN for internal clients. An IDS/IPS Server safeguards all tiers, and a DBMS Server, protected behind the IDS/IPS layer, houses the Database, ensuring data security throughout the network.



# Part 2: Data Entities, Attributes, and Files:

## 1. Entity: Housing Design:

* Description: Housing designs are the unique architectural plans for the unconventional homes we offer to our clients. Each design has a DesignID, architect, style, and estimated cost.

| Attribute | Data Type |
| --- | --- |
| DesignID | INT |
| DesignName | VARCHAR(50) |
| Architect | VARCHAR(30) |
| DesignType | VARCHAR(20) |
| EstimatedCost | DECIMAL(10,2) |

## 2. Entity: Customer Order:

* Description: Customer orders represent the specific requests made by customers for their chosen housing designs. Order includes OrderID, CustomerID, DesignID, and OrderStatus.

| Attribute | Data Type |
| --- | --- |
| OrderID | INT |
| CustomerID | INT |
| DesignID | INT |
| OrderDate | DATE |
| OrderStatus | VARCHAR(15) |

## 3. Entity: Material Inventory:

* Description: Material inventory tracks the availability and quantity of the unconventional construction materials used in our housing projects It includes MaterialID, MaterialName, Quantity, Supplier, and UnitPrice.

| Attribute | Data Type |
| --- | --- |
| MaterialID | INT |
| MaterialName | VARCHAR(30) |
| Quantity | INT |
| Supplier | VARCHAR(40) |
| UnitPrice | DECIMAL(10,2) |

## 4. Entity: Employee:

* Description: Employees are the staff members who contribute to the various aspects of our unconventional housing business.

| Attribute | Data Type |
| --- | --- |
| EmployeeID | INT |
| FirstName | VARCHAR(20) |
| LastName | VARCHAR(20) |
| Position | VARCHAR(30) |
| Email | VARCHAR(40) |

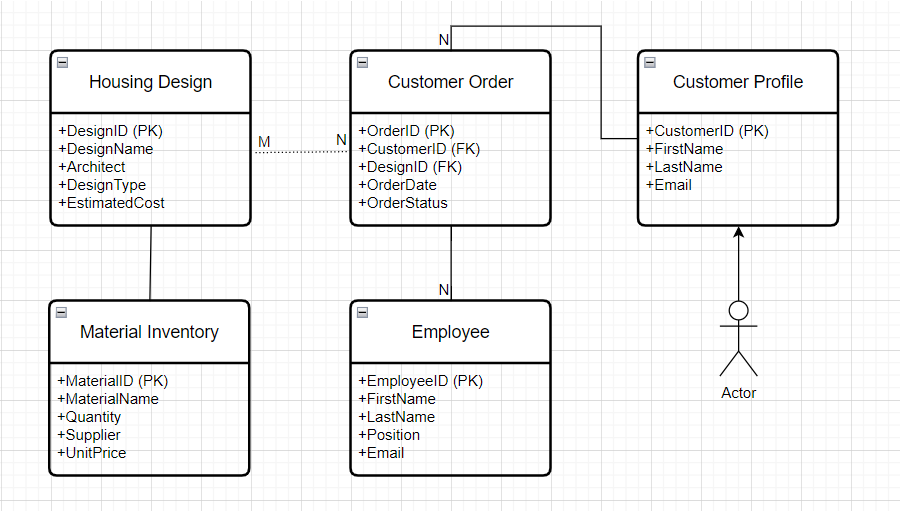
## 5. Entity: Customer Profile:

* Description: Customer profiles store information about customers who use our services and order unconventional homes.

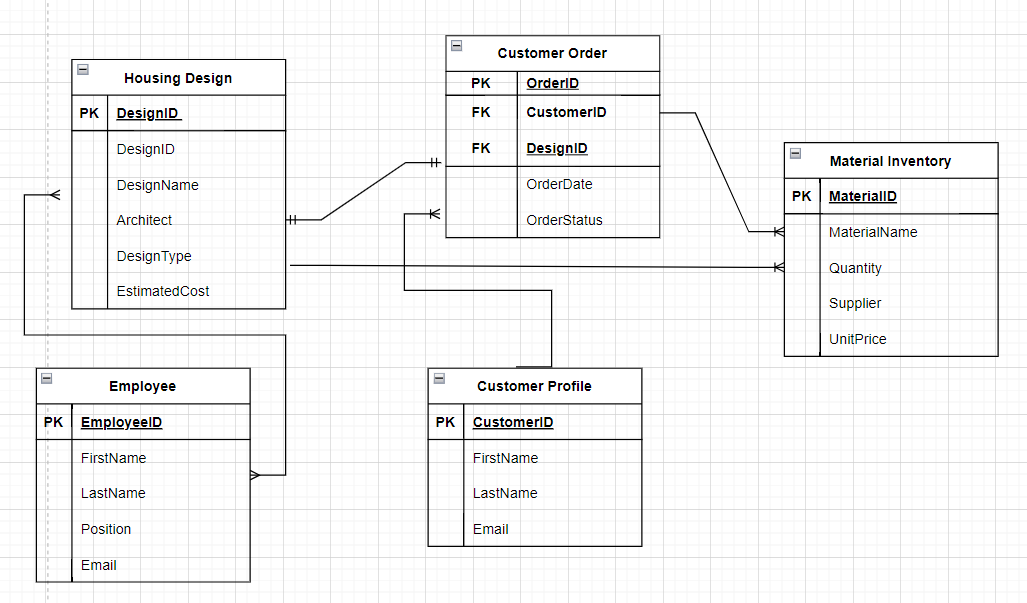
| Attribute | Data Type |
| --- | --- |
| CustomerID | INT |
| FirstName | VARCHAR(40) |
| LastName | VARCHAR(40) |
| Email | VARCHAR(50) |

# Part 3: ERD Model:

## 1. ERD using UML:



## 2. ERD using Crow’s Foot notation:



## 3. Description of the relationships:

Customer Order:

* Customer Profile: One customer profile can place many orders; This is a one-to-many relationship.
* Housing Design: One customer order can be for one housing design; This is a one-to-one relationship.

Housing Design:

* Material Inventory: One housing design can require multiple materials from the material inventory. This is a one-to-many relationship.
* Employee: Many housing designs can be assigned to many employees. This is a many-to-many relationship.

Employee:

* Housing Design/Customer Order: One employee can be assigned to multiple customer orders. This is a one-to-many relationship.

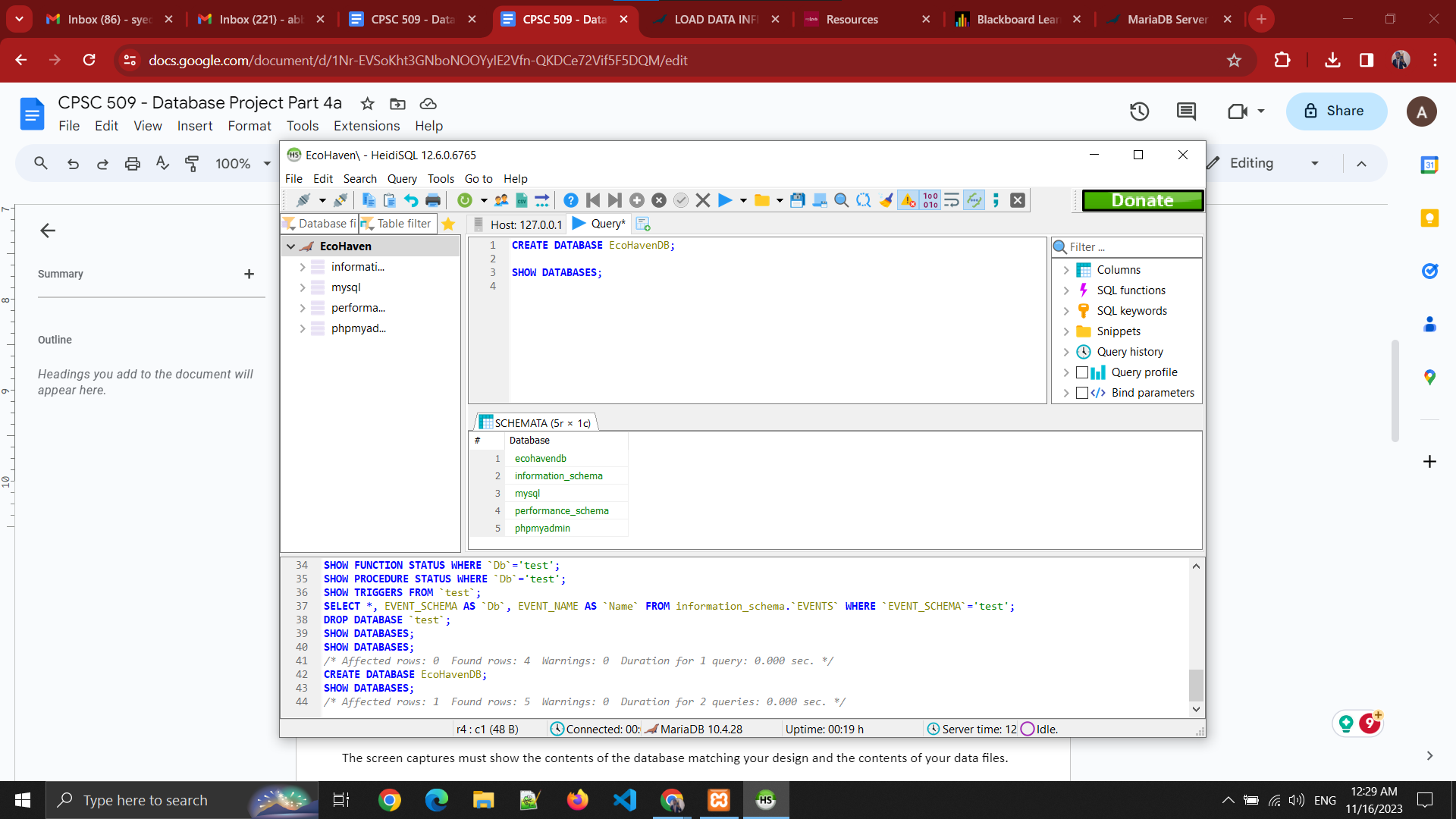
Customer Profile

* Employee: One customer profile can place many customer orders. This is a one-to-many relationship.

# Part 4(a): DDL Scripts Report:

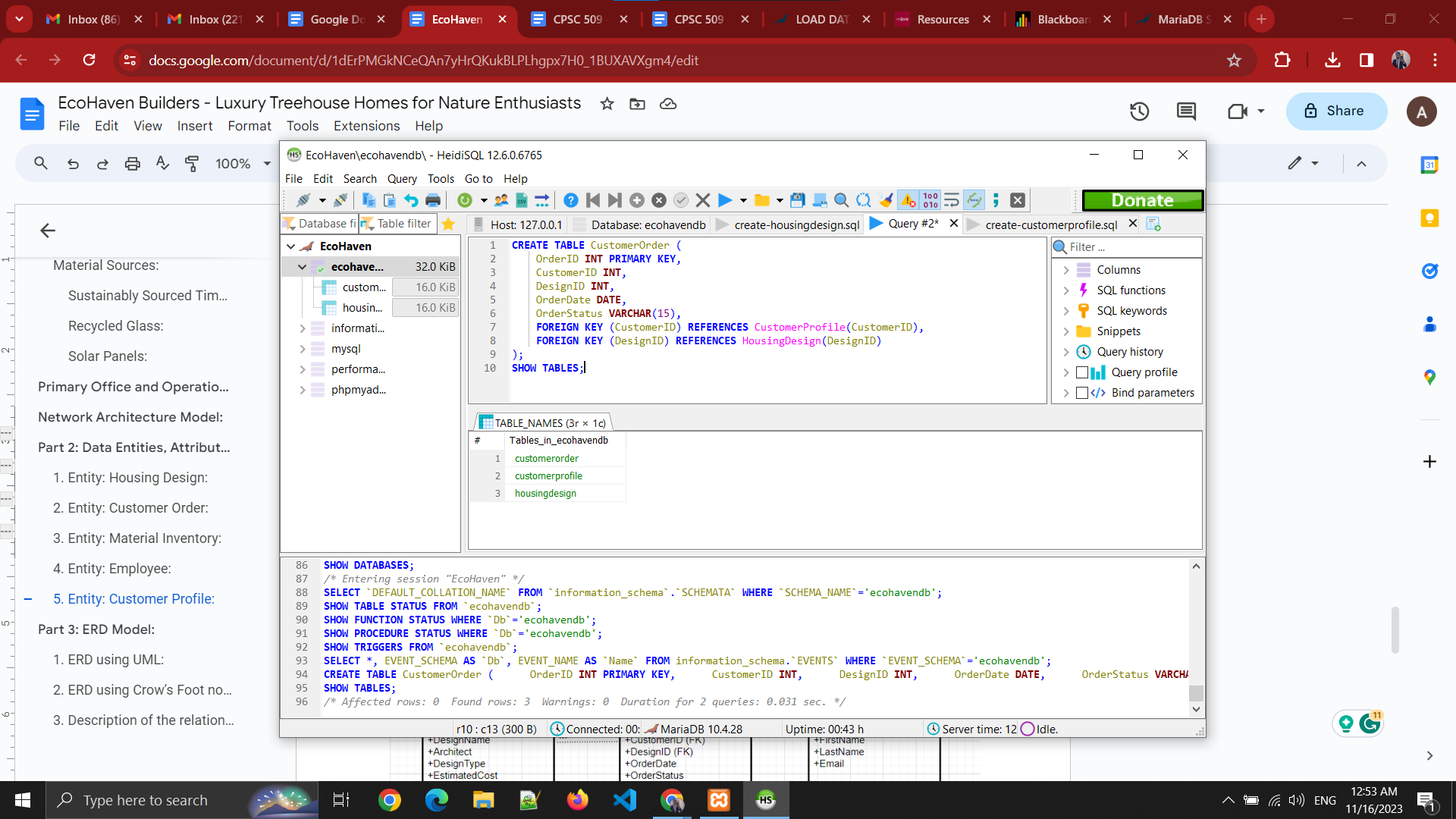
## 1. Database Creation Script:

## The below screenshot shows the script for creating a database with the name “ecohavendb”.



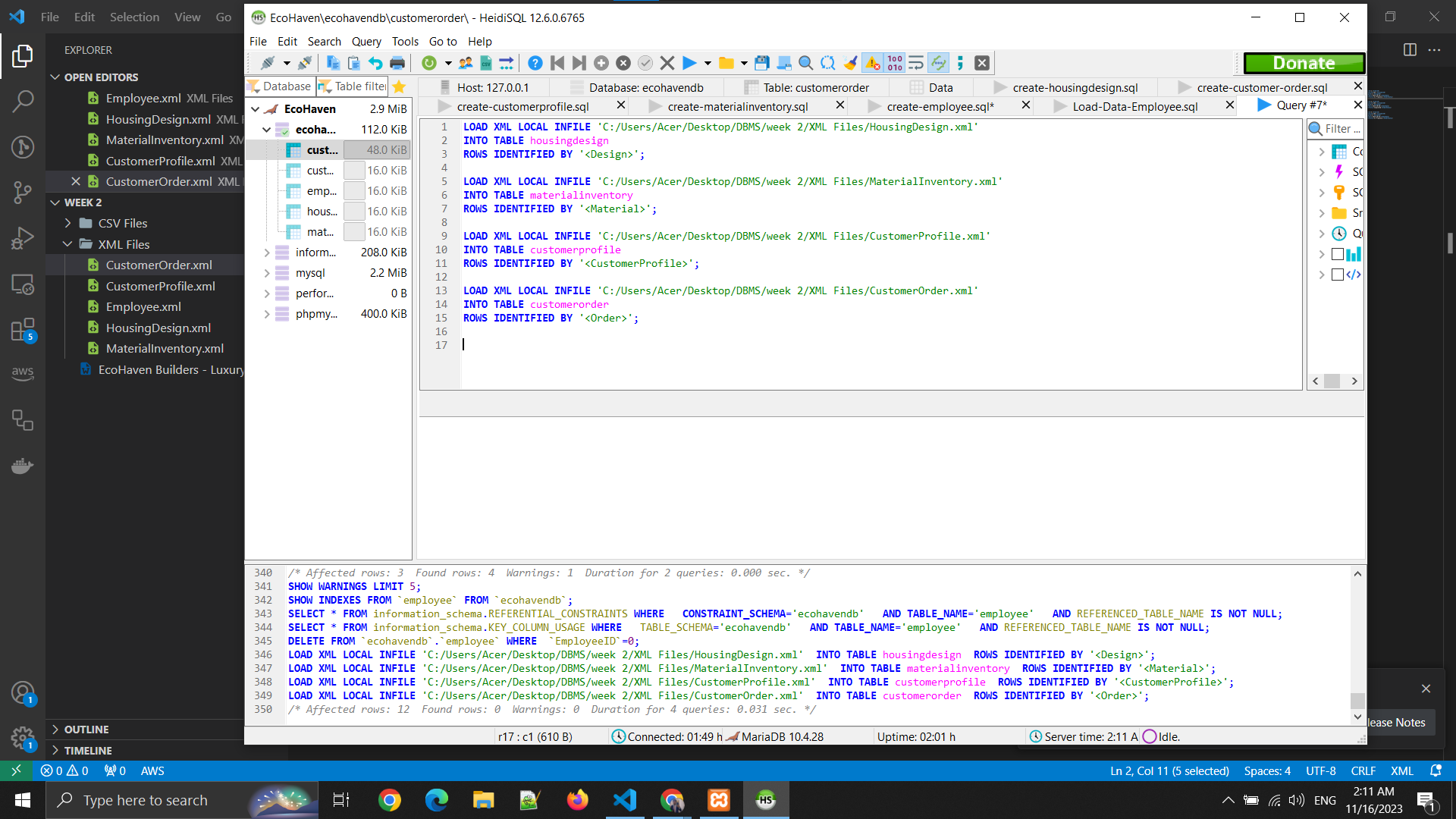
## 2. Tables Creation Script:

## The below screenshot shows the script for creating tables in the MySQL database.

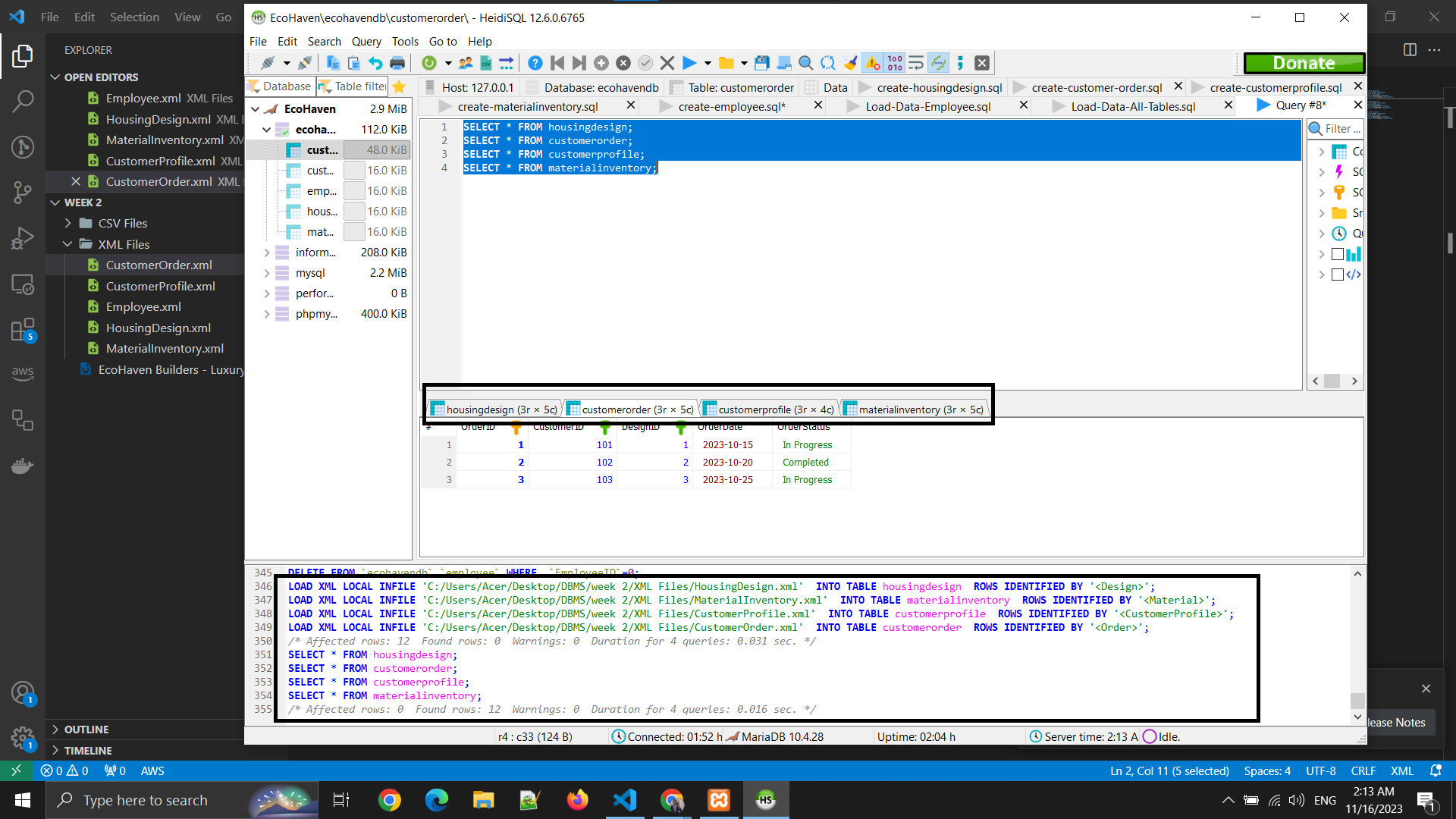


3. Load Data Into Tables:

The below screenshot shows the MySql script to load the data into the respective tables from the local XML files as specified below in the image.

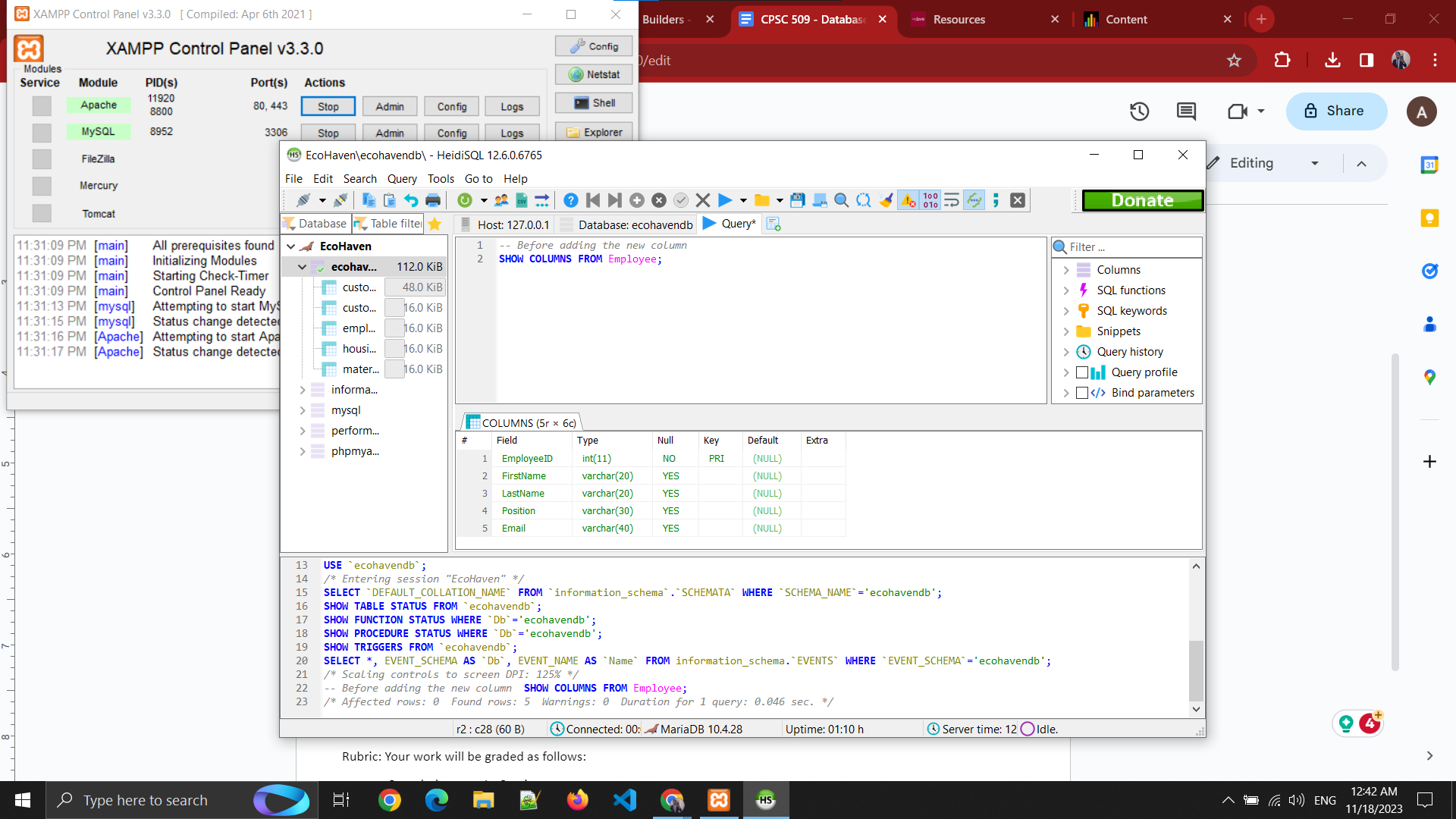
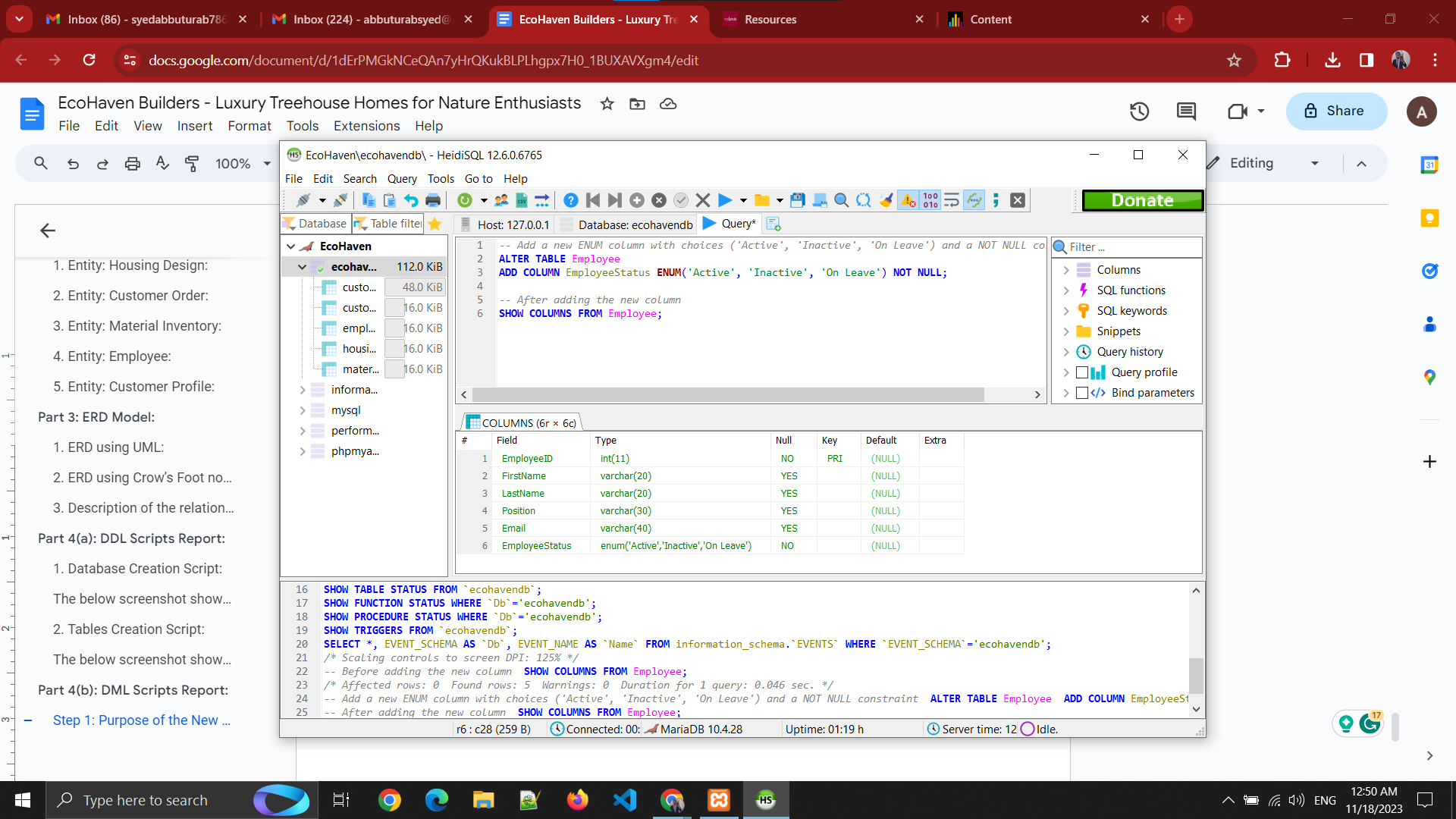


The below query shows the successful results of the load data query.



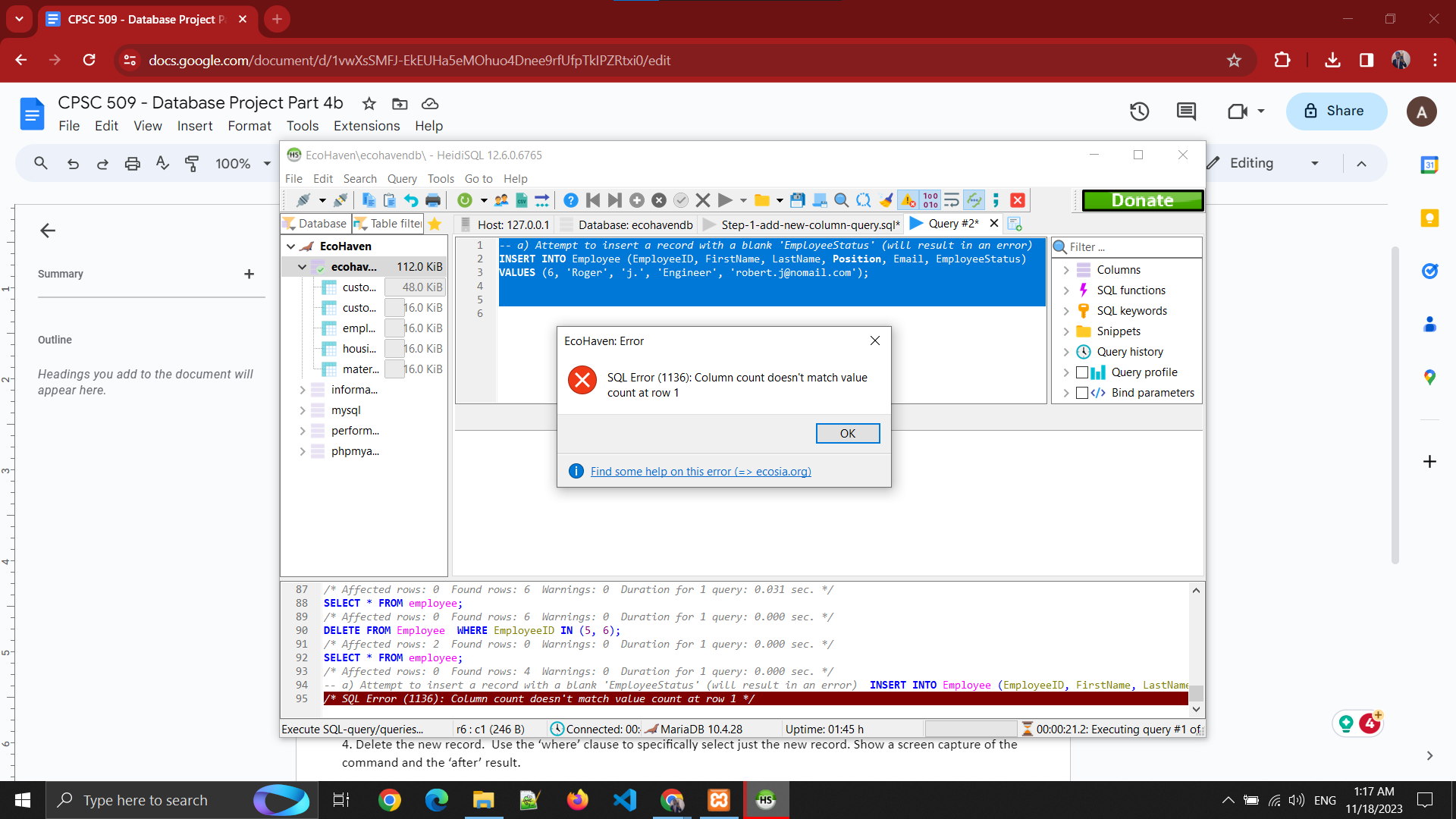
# Part 4(b): DML Scripts Report:

## Step 1: Purpose of the New Column The EmployeeStatus column serves the purpose of tracking the current employment status of each employee. The three possible values provide information about whether an employee is currently active, inactive, or on leave.

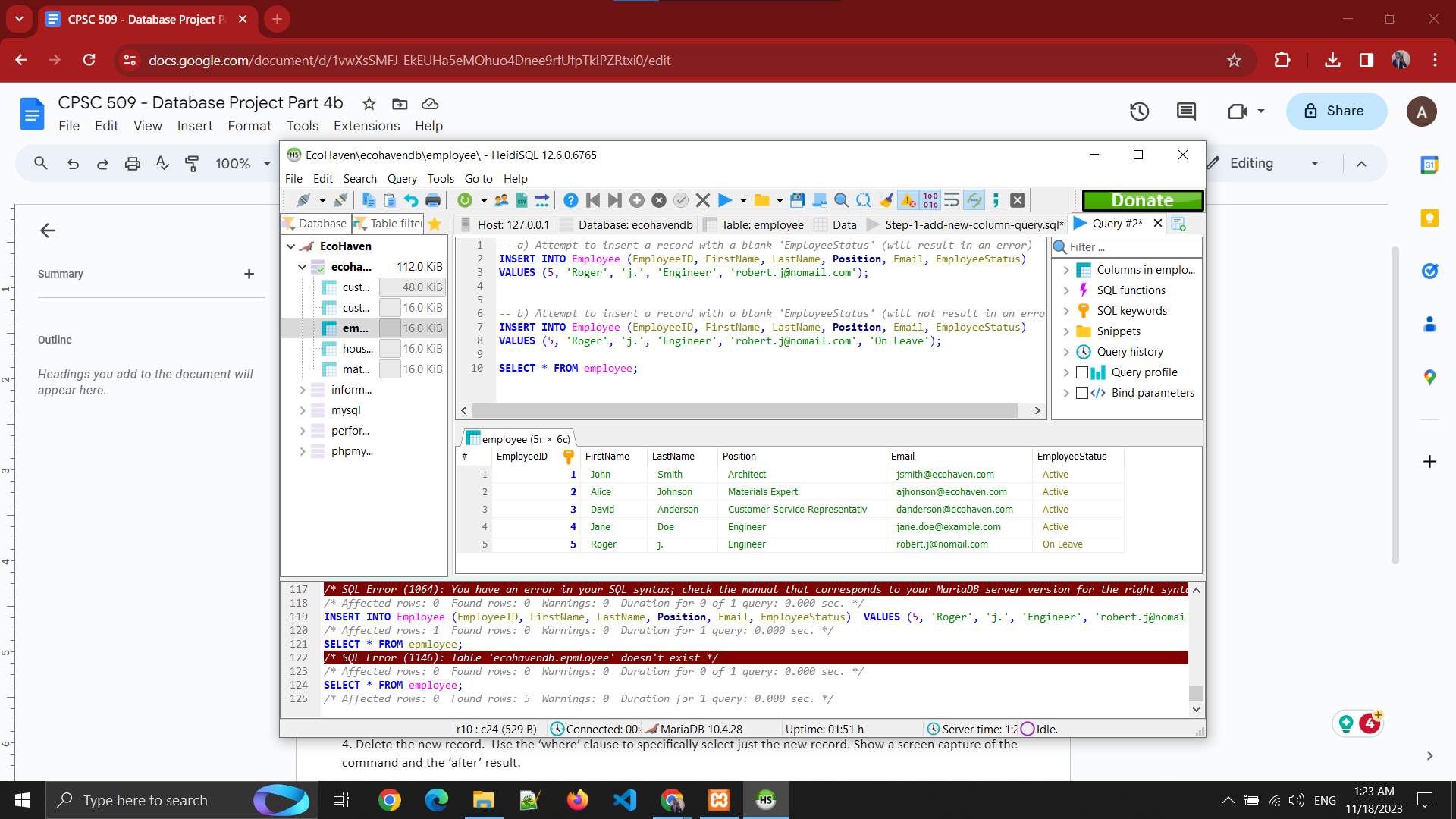


## Step 2: Insert New Record:

the system throws an error if you try to leave the required ‘enum’ value blank.



Insert query to include a valid entry so that the data works properly. See the screenshot below:



## Step 3: Changes!:

The possible changes to each column and demonstrate those changes:

**EmployeeID:** We can't modify the EmployeeID directly, especially if it's a primary key. Changing it would likely involve deleting and re-inserting the record with a new ID.

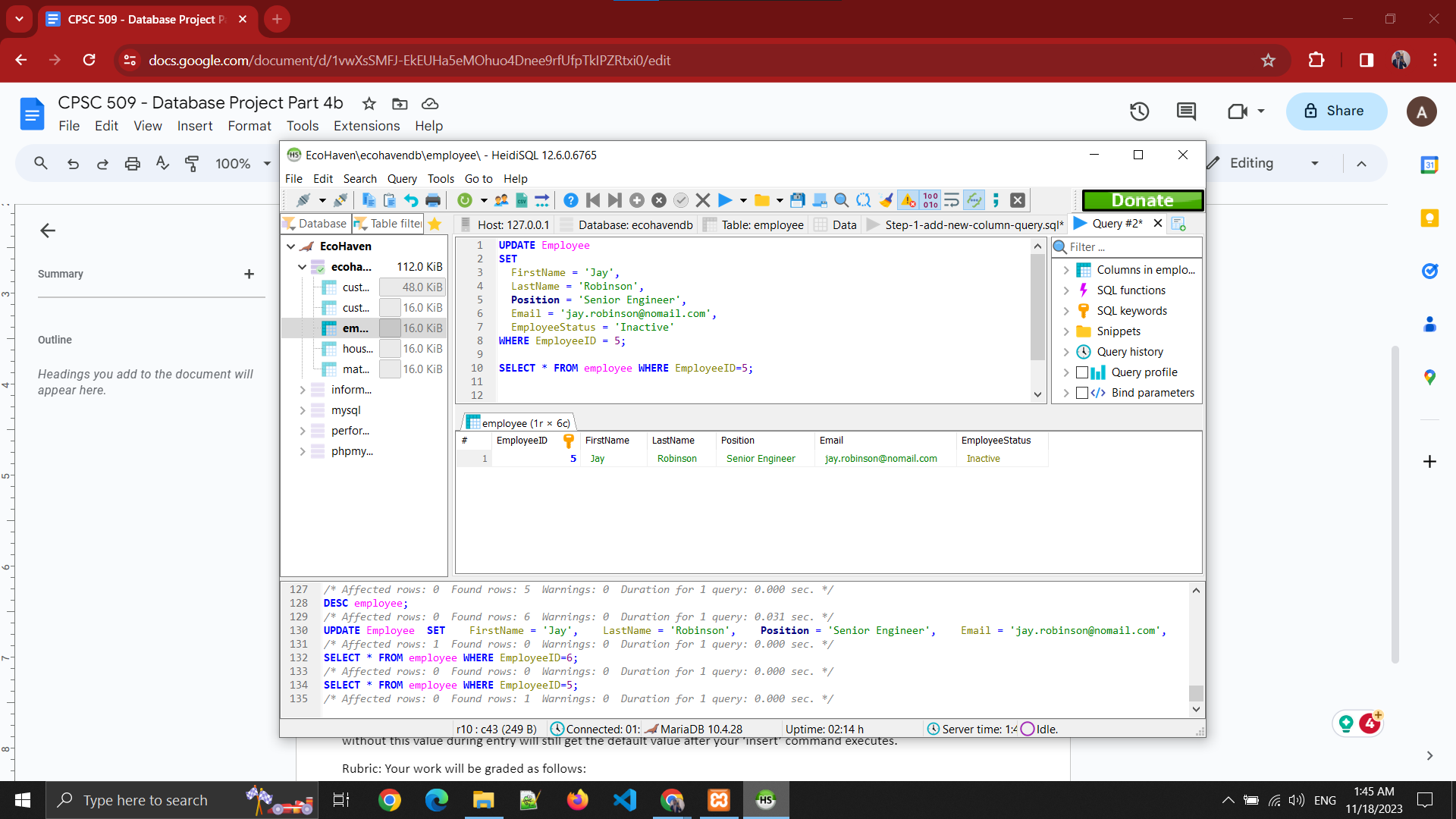
**FirstName and LastName:** We can update these fields to reflect any changes in the employee's name.

**Position:** We can update the employee's position.

**Email:** We can update the employee's email address.

**EmployeeStatus:** We can update the employee's status.

Here’s the screenshot of the changes:



## 12-Queries:

Below are the results of twelve SQL queries that cover a variety of operations:

