SQL Database E-Commerce

By **Uzoma Maduakolam** & **Felix Berendes**

07.08.2023

Table of Contents

[Article I. Tables overview: 3](#_Toc142305697)

[Section 1.01 Customers 3](#_Toc142305698)

[Section 1.02 Payment Details 3](#_Toc142305699)

[Section 1.03 Supply 3](#_Toc142305700)

[Section 1.04 Products 3](#_Toc142305701)

[Section 1.05 Orders 3](#_Toc142305702)

[Section 1.06 Listings 3](#_Toc142305703)

[Section 1.07 Reviews 3](#_Toc142305704)

[Section 1.08 Refunds 4](#_Toc142305705)

[Section 1.09 Category 4](#_Toc142305706)

[Article II. Visual Model 5](#_Toc142305707)

[Article III. Objective 6](#_Toc142305708)

[Section 3.01 Regarding listings: 6](#_Toc142305709)

[Article IV. Tables Described 7](#_Toc142305710)

[Section 4.01 Table: Customers 7](#_Toc142305711)

[Section 4.02 Table: Payment Details 7](#_Toc142305712)

[Section 4.03 Table: Supply 7](#_Toc142305713)

[Section 4.04 Table: Products 7](#_Toc142305714)

[Section 4.05 Table: Orders 7](#_Toc142305715)

[Section 4.06 Table: Listings 7](#_Toc142305716)

[Section 4.07 Table: Reviews 7](#_Toc142305717)

[Section 4.08 Table: Refunds 8](#_Toc142305718)

[Section 4.09 Table: Categories 8](#_Toc142305719)

[Article V. Populating 9](#_Toc142305720)

[Screens used to insert data may be inspected below. 9](#_Toc142305721)

[Section 5.01 Inserting Products 9](#_Toc142305722)

[Section 5.02 Inserting Orders 9](#_Toc142305723)

[Section 5.03 Inserting Listings 9](#_Toc142305724)

[Section 5.04 Inserting Customers 10](#_Toc142305725)

[Section 5.05 Inserting Supply 10](#_Toc142305726)

[Section 5.06 Inserting Payment Details 10](#_Toc142305727)

[Section 5.07 Inserting Reviews 10](#_Toc142305728)

[Section 5.08 Inserting Refunds 10](#_Toc142305729)

# Tables overview:

## **Customers**

* + Customer ID (PK)
  + First name (VARCHAR)
  + Last name (VARCHAR)
  + Address (VARCHAR)
  + E-Mail (VARCHAR)
  + Password (VARCHAR)

## **Payment Details**

* + Customer[id]
  + Payment Method (VARCHAR)
  + Currency (VARCHAR)
  + CreatedDate (DATE)
  + UpdatedDate (DATE)

## **Supply**

* + Supply ID
  + Product[id]
  + Amount in stock (INT)

## **Products**

* + Product ID (PK)
  + Manufacturer (VARCHAR)
  + Item name (VARCHAR)
  + Retail price (DECIMAL)
  + Category [id]

## **Orders**

* + Order ID (PK)
  + Product[id]
  + Customer[id]
  + Date (DATE)
  + Status (VARCHAR)

## **Listings**

* + Listing ID (PK)
  + Product[id]
  + Price (DECIMAL)
  + Discount (INT)
  + Review[id]

## **Reviews**

* + Review ID (PK)
  + Customer[id]
  + Product[id]
  + Rating (INT)

## **Refunds**

* + Refund ID (PK)
  + Order[id]
  + Product[id]
  + Return reason (TEXT)
  + Return date (DATE)
  + Refund amount (DECIMAL)
  + Refund status (VARCHAR)

## **Category**

* + Category ID (PK)
  + Category name (VARCHAR)

|  |
| --- |
| **Key:** |
| PK Primary key -> integer |
| [\_] Reference to foreign key |
| VARCHAR Data type String |
| DATE Data type Date |
| TEXT Data type single & multi-byte characters |
| DECIMAL Data type ranged decimal number |
| ID Identifier per Table (integer) |

# Visual Model

A diagram of a company

Description automatically generated

# Objective

The objective of this schema is to provide a modular structure for an e-commerce service. Customer data is used throughout the tables to track the author of reviews, client of an order and receiver of an order.

The system is designed to allow an application to track the availability of a product vs a listing of that product via the supply table.

## Regarding **listings**:

Each listing contains a product. A product is supplied with a suggested retail price, which may however differ from the price one wants to charge the customer.

By using the Listings table, the same product can be offered at different prices depending on e.g. the category it is sold under or the country from where the customer accesses the store.

# Tables Described

## **Table: Customers**

Function: store customer information

Reason: using customer to place orders, give reviews

## **Table: Payment Details**

Function: store customers payment details

Reason: Payment details are separated to enhance data protection and store sensitive details separately.

## **Table: Supply**

Function: Store units of products available for purchase

Reason: The service must check availability of product before completing an order.

## **Table: Products**

Function: Store information about a single product.

Reason: Each product contains data relevant for completing an order and finding it in search.

## **Table: Orders**

Function: store current and past orders and their status.

Reason: The application/service likely wants to query this to display an overview of orders.

## **Table: Listings**

Function: A listing is an offer for the purchase of a product, i.e. “Listings” is the table finally displayed on the store front end page when the customer is shopping with all relevant information of the corresponding product. See section x for a detailed explanation.

Reason: see section 3.01 .

## **Table: Reviews**

Function: Hold reviews for *listed* products to display before placing an order.

Reason: Reviews are displayed for a listing, not a product. This separation allows the same product to be reviewed differently depending on where it is listed and for what price.

## **Table: Refunds**

Function: Hold refund processes for customers and track their status.

Reason: Refunds contain the order to be returned which link to the corresponding customer and contain the status of each process.

## **Table: Categories**

Function: Organise categories for products.

Reason: Categories are associated with a product which may be used for functions such as filters or organizational efforts.

# Populating

The schema was populated with semi-realistic data. An inaccuracy to be noted is the password stored in the Customer table, which would be inserted after being hashed by a hashing function and algorithm (e.g. SHA or MD5). Lacking the realistic scenario of a user entering a real password for registration, dummy text was hashed and used as a place holder. The model is not affected negatively by this, because the password is only used for authorization at login, which does not occur when inspecting the database.

## Screens used to insert data may be inspected below.

## Inserting Products

A screenshot of a computer

Description automatically generated

## Inserting Orders

A screen shot of a black screen

Description automatically generated

## Inserting Listings

A screen shot of a computer

Description automatically generated

## Inserting Customers

A screen shot of a computer screen

Description automatically generated

## Inserting Supply

A screen shot of a computer

Description automatically generated

## Inserting Payment Details

A screen shot of a computer

Description automatically generated

## Inserting Reviews

A screen shot of a computer

Description automatically generated

## Inserting Refunds

A screenshot of a computer screen

Description automatically generated