

Eco Stoves Database Design

Database modelling

Assignment

Charlene Gordon | 99584891 | Dec 2017

Contents

[Introduction 2](#_Toc501015763)

[Business rules 3](#_Toc501015764)

[sample Customer Invoice 4](#_Toc501015765)

[Normalisation 5](#_Toc501015766)

[Normalisation Table 6](#_Toc501015767)

[Relationships 7](#_Toc501015768)

[EER diagrams 8](#_Toc501015769)

[Implementation of SQL 10](#_Toc501015770)

[sql queries 10](#_Toc501015771)

[Security 12](#_Toc501015772)

[Views 12](#_Toc501015773)

[Conclusion 13](#_Toc501015774)

## Introduction

ECO Stoves is a fictional company based in Ireland. They manufacture and sell stoves to customers who are mostly retailers. Currently they have no database on the sales of stoves and parts they sell. They use a paper based invoice system as they are a new company they have no system set up to handle data and make queries.

Each stove they assemble is packaged ready for install for the customer. The customer buys a stove based on the heat output they need, and what finish it has, each stove has a unique id number. The company receives a purchase order for the stove and the stove is then delivered to the customer. The company then sends an invoice for the stove. The customer then has a registered installer to fit the stove.

The customer can also order spare parts for their stove if they need them, such as glass for the door and rope seal for the door.

This document aim is to help create a database to hold all the relevant information on the products and customers. They then can record the information and update it as they need to. The database will then be able to handle queries for the company in what information they need, for example what is the most sold type of stove or to see which customers have outstanding payments.

What I will be covering in this document is an ER model for the company and a ER diagram. I will be showing the process of normalization for the tables created.

I will also create a fictional database in MySQL where I will create the ER Diagram and create a script for the tables.

I will be running queries from the database which I will show in the document

## Business rules

To design the database, I will need to perform a requirements analysis, which will identify policies and procedures for the company and will list the business rules for them. There must be consensus on what the rules mean, so that everyone who will use the database must agree in the definition and the use of these data items. For example, if you asked three people within the company to define what a ‘customer’ means you might get three different answers: To one person, a customer is the company that buys products. To a second person, the customer is the contact person for the company who buys products. To a third person, the customer is someone who might be interested in buying products. So, we must confirm meanings of terms, and confirm how the company uses these terms.

In the case of ‘Eco Stoves’ the business rules after analysis are as follows; A customer buys stock from Eco Stoves using a purchase order (at present all customers are retailers so are therefore allowed a 30-day credit to pay), Eco Stoves receives the purchase order and fulfils the order ready for delivery, the delivery is carried out by a specified courier, then an invoice is prepared with all the details as you can see in the invoice example and sent to the customer. The customer then makes a payment. Some retailers pay in full and other in installments so there may be a payment outstanding.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| sample Customer Invoice  |  | | --- | | ECO STOVES | | |  | | INVOICE | |
| Unit 6, Northern Industrial Est | |  | | **Date** | 10/05/2017 |
| Belltown, Co Waterford | |  | | **Invoice #** | 11 |
| Phone: (051) 555-116 | |  | | **For** | PO # 123456 |
| Fax: (051) 555-114 | |  | |
| [sales@ecostoves.com](mailto:sales@ecostoves.com) | |  | |  |  |
| **Bill To:** | | | | | |
| Customer Number: 123 | |  | | |  |
| Stove Suppliers | |
| Main Street | |  | | |  |
| Small Town, Co Cork | |
| (021) 342-1163 | |  | |  |  |
| **Quantity** | **Product ID & Description** | **Unit price** | | **Amount** | **Column1** |
| 3 | 05- Non-boiler stove14kw enamel | € 500.00 | | € 1,500.00 |  |
| 1 | 06- Non-boiler stove 18kw matt | € 700.00 | | € 700.00 |  |
| 4 | 10-Glass 14kw | € 10.00 | | € 40.00 |  |
|  |  |  | | € - |  |
| **Subtotal** |  |  | | **€ 2,240.00** |  |
| Make all checks payable to Eco Stoves | | **Credit** | | € 1,000.00 |  |
| If you have any questions concerning this invoice, contact Sam Smith at sam@ecostoves.com | |  |
|  | |  |  |
| **Thank you for your business!** | | **Balance due** | | **€ 1,755.20** |  |

## Normalisation

After analysis and discovering the business rules I now need to create tables so to do this I need to find the entities of each table and their attributes by looking at the data the business uses and then proceed to normalise the data. To do this, I need to make sure the following goals are achieved:

* arranging data into logical groupings such that each group describes a small part of the whole
* minimizing the amount of duplicate data stored in a database
* organizing the data such that, when you modify it, you make the change in only one place
* building a database in which you can access and manipulate the data quickly and efficiently without compromising the integrity of the data in storage

When I was normalizing the data, I had four goals: arranging data into logical groupings such that each group describes a small part of the whole; minimizing the amount of duplicate data stored in a database; organizing the data such that, when it is modified, you make the change in only one place; and building a database in which you can access and manipulate the data quickly and efficiently without compromising the integrity of the data in storage.

Snapshot of data.



## Normalisation Table

|  |  |  |  |
| --- | --- | --- | --- |
| UNF | 1NF | 2NF | ENTITY |
| CustomerNo | **OrderNo(PK)** | **CustomerNo(PK)** | **Tbl\_customer** |
| CustomerName | ProductNo | Name |  |
| Address | ProductDesc | Address |  |
| County | ProductPrice | County |  |
| Phone | OrderQTY | Phone |  |
| ProductNo | InvoiceNo |  |  |
| ProductDesc | Subtotal | **InvoiceNo (PK)** | **Tbl\_Invoice** |
| ProductPrice | Total | Subtotal |  |
| OrderNo | Payments | Total |  |
| OrderQty | OutstandingDebt | **customerNo(FK)** |  |
| SubTotal |  |  |  |
| Total | **OrderNo(PK)** | **PaymentsNo(PK)** | **Tbl\_Payments** |
| InvoiceNo | customerNo | **InvoiceNo(FK)** |  |
| Payments | Name | OutstandingDebt |  |
| OutstandingDebt | Address |  |  |
|  | County |  |  |
|  | Phone | **DeliveryNo (PK)** | **Tbl\_Deliverys** |
|  |  | deliveryDate |  |
|  |  | Courier |  |
|  |  | **CustomerNo(FK)** |  |
|  |  |  |  |
|  |  | **OrdersNo(PK)** | **Tbl\_Orders** |
|  |  | **CustomerNo(FK)** |  |
|  |  |  |  |
|  |  | ProductID | **Tbl\_Products** |
|  |  | ProductDesc |  |
|  |  | productPrice |  |
|  |  | **OrdersNo(FK)** |  |
|  |  | **DeliveryNo(FK)** |  |
|  |  | **CustomerNo(PK)** |  |
|  |  | Name |  |

## Relationships

To design a database, I had to establish the relationship between the entities;

* Customers’ 🡪 Orders; 1 customer can place an order several times
* Orders 🡪 Products; 1 Order can order many products
* Products 🡪Deliveries; Many products are packaged for 1 Delivery
* Deliveries 🡪 Customer; Many deliveries are made to 1 Customer
* Customer 🡪 Invoice; 1 customer can have many invoices
* Invoice 🡪Payments; 1 invoice can have many payments

Between the entities there may be a mutual dependency. This means that the one item cannot exist if the other item does not exist. For example, there cannot be an order if there are no customers, and there cannot be an order if there are no products.

## EER diagrams

![A close up of a map

Description generated with very high confidence]()

This was my first attempt at establishing the relationships between the entities. This is not correct as it gave a lot of duplication of the data and the relationships were not in the correct position.

![A screenshot of a cell phone

Description generated with very high confidence]()

2nd EER diagram.

## Implementation of SQL

From the eer model I created a script then I wrote an inserts script to input some data. These scripts are attatched to this report.

I had a lot of trouble with using the correct syntax and ensuring the integrity of the data was maintained.

## sql queries

![A screenshot of a cell phone

Description generated with very high confidence]()

![A screenshot of a cell phone

Description generated with high confidence]()![A screenshot of a social media post

Description generated with very high confidence]()

## Security

MySQL administration is more than running queries. You must also take precautions in security your data and managing the users who have access to this data.

MySQL uses its own encryption scheme for passwords. MySQL has a PASSWORD () function, and the same scheme is used to encrypt your MySQL administrator password.

The departments in the company are the sales office, production department and administration department.

The production department should not be able to change any data on the customer tables and on payments and invoice.

They can make changes on orders if an order changes for example.

The administration department in the case of senior administrators would oversee payments and invoice. If a new trainee were to join the company, they would not have access to update or delete any information

## Views

Views can represent a subset of the data contained in a table. Consequently, a view can limit the degree of exposure of the underlying tables to the outer world: a given user may have permission to query the view, while denied access to the rest of the base table.

Views can join and simplify multiple tables into a single virtual table.

Views can act as aggregated tables, where the database engine aggregates data (sum, average, etc.) and presents the calculated results as part of the data.

Administration would have views to calculate invoices in relation to the orders inputted by the sales department.

## Conclusion

I found this assignment very challenging and felt I did not have enough time.

I have never done programming so syntax and error messages were completely foreign to me. Due to this I wasted a lot of time writing the scripts due to error messages I couldn’t resolve. After many struggles and pulling out of hair, I achieved the scripts to run without errors after seeing my lecturer for an additional session.

This has been a huge learning curve for me I have a better understanding of databases and the importance of design.

SQL, I feel I must work more on my queries were not very complex due to time constraints.