

Lincoln Murray

Mazenod College

A SQLite Database Assignment

Rock Scalers

For all your climbing needs

# Part 1

## Outline

The purpose of this software/Database is to provide a backend for a rock climbing store that sells products for a mark up from other stores. In order to provide a low work profit. This backend will be based in SQLite and will hold a list of products, their respective suppliers, manufacturers and customer they’ve been sold too. There are many steps required to complete this task, To start off it is required to conceptualize the plan/idea for the task, As previously mentioned this task is a rock climbing store. Secondly it is required to confirm that the data is able to be collected, so a piece of python code to scrape the data from various websites was created and the data integrity was verified. After that the plan needs to be set in stone by creating this planning document. The investigation part first then the design then final edits. Finally the Database needs to be developed, insert queries for the data first then the rest of the queries for the front end and lastly compiling it all together in a flask based python back end. Then the entire project can be finalized, this includes final edits in the document, front end, and any other final touches. You can see this visually laid out in the timeline below:

## Timeline

Description

Requirements

* A SQLite Database
  + The storage of relevant data of customers.
    - Orders
    - Personal Data
      * Address
        + Billing
        + Delivery
      * Card details
      * Name
  + The storage of relevant data of products
    - Manufacturers
      * Established date
      * Established locations
    - Suppliers
      * URL
    - Products
      * Price from different suppliers
      * Purchase URL
      * Description
* A Flask based python frontend.
  + The ability for a customer to order products.
  + Visually appealing
  + Display products
    - Automatically find cheapest price from suppliers

Ethical, Legal and Security issues

The Australian privacy principles

Australian Privacy Principle 1—open and transparent management of personal information:

The customers will be alerted how it will only be used within the store.

Australian Privacy Principle 2—anonymity and pseudonymity:

This principle will be covered by providing the ability for a guest account to be used if required.

Australian Privacy Principle 3—collection of solicited personal information:

Only required information will be stored such as card information, delivery and billing information and their name.

Australian Privacy Principle 4—dealing with unsolicited personal information:

All data will be collected with consent from the customers.

Australian Privacy Principle 5—notification of the collection of personal information:

The users will be alerted about what data is stored by the website.

Australian Privacy Principle 6—use or disclosure of personal information:

The customer will be alerted to the use of personal information within the site and how it will be only used within this product.

Australian Privacy Principle 7—direct marketing:

The customer may consent to direct marketing requiring the use of their name and a contact method, this service will be operated privately.

Australian Privacy Principle 8—cross‑border disclosure of personal information:

All Data will be stored within an Australian server and will not cross the border to a foreign entity.

Australian Privacy Principle 9—adoption, use or disclosure of government related identifiers:

No government identifiers will be used to identify any information.

Australian Privacy Principle 10—quality of personal information:

The quality of the personal information collected is up to the customer, as they’re inputting it directly, this means that any personal information is first hand and should be of high quality with great accuracy.

Australian Privacy Principle 11—security of personal information:

The personal information will be secure as it will be stored locally on a secure server with a high quality firewall, and any data transmitted will be encrypted

Australian Privacy Principle 12—access to personal information:

Users of the store may send a request for all records their personal data to be sent to them.

Australian Privacy Principle 13—correction of personal information:

Users of the store may also request for incorrect data to be corrected as soon as possible.

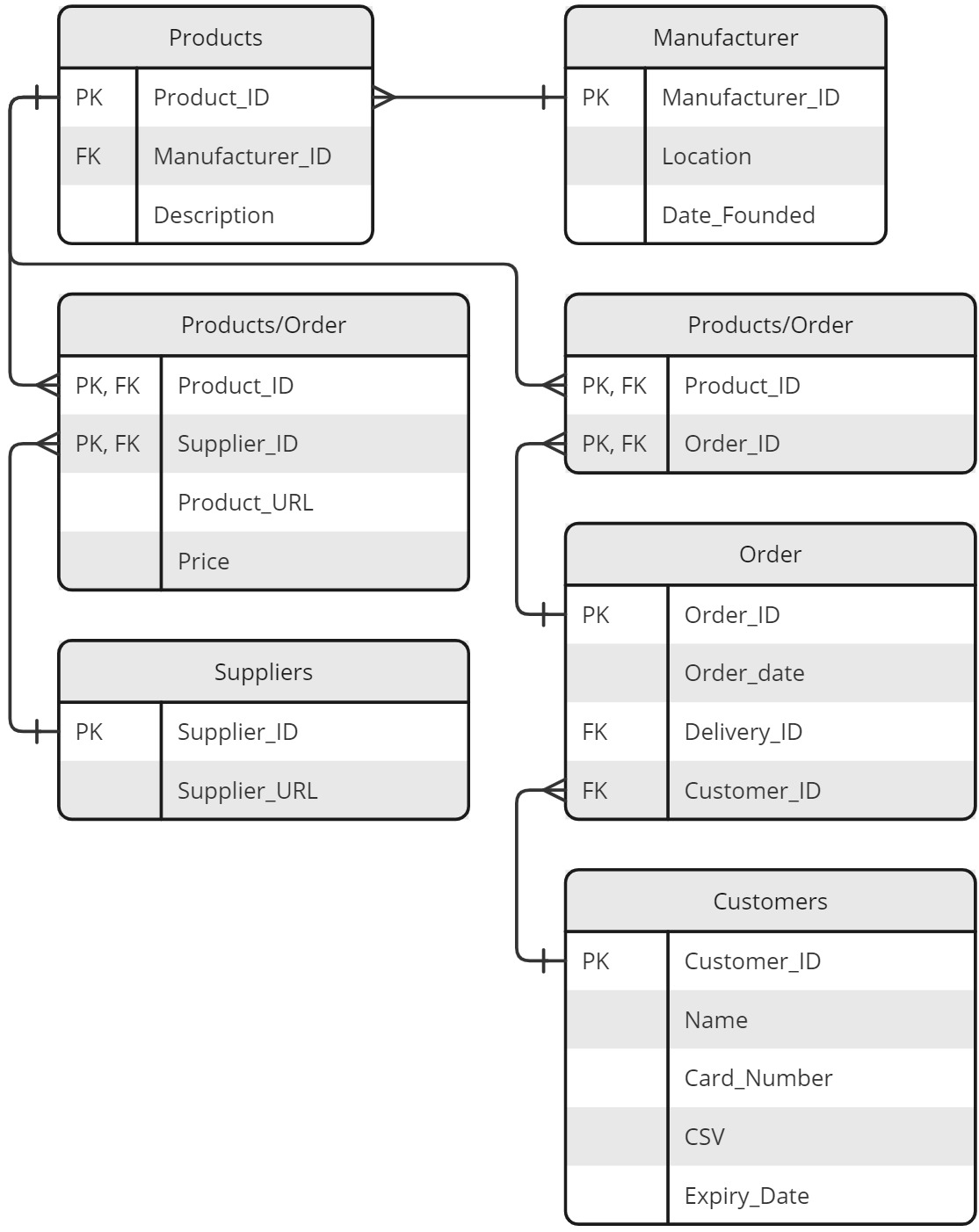
In addition to the Australian privacy principles Users may also request to have the record of their personal information deleted, excluding records of a sale. This means that once the payment has been processed their customer details will be deleted.

Quality of Data

There are limited factors impacting the quality of data as it is all collected first hand. This is due to the nature of the database, some impacting factors include the data from the stores data is scraped from, Some of these such as spelling mistakes are not that critical and will not impact the store. Other factors such as faulty prices are critical but if the store the data is scraped from is wrong then it should not impact the profit margin of the store. Under the assumption that they honor their mistake.

## Design

Entity Relationship Diagram



Relational Notation

Products(ProductID, ManufacturerID (FK), Description) Manufacturers(ManufacturerID, Location, DateFounded)

Product-Supplier(ProductID (FK), SupplierID (FK), ProductURL, Price)

Suppliers(SupplierID, SupplierURL)

Product-Order(ProductID (FK), OrderID (FK))

Orders(OrderID, CustomerID (FK), DeliveryAddress, BillingAddress, DeliveryRequest, Total)

Customers(CustomerID, Name)

Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Element Name | Data Type | Description | Constraints |
| ProductID | Integer | A Unique identify for each product. | Must be unique, >= 0 and not null |
| ManufacturerID | Integer | A Unique identify for each manufacturer. | Must be unique, >= 0 and not null |
| Description | String | The descriptor for each product. | No constraints |
| Location | String | General location of the manufacturer companies origin. | Max Length 50 Characters |
| DateFounded | Integer | Year the company was founded | Must be > 0, <= Current Year and not null |
| SupplierID | Integer | A Unique identify for each supplier. | Must be unique, >= 0 and not null |
| ProductURL | String | The URL to order the product. | Max Length 100 Characters and not null |
| Price | Float | The price of the product | Must be > 0 and not null |
| SupplierURL | String | The URL to view the suppliers website | Max Length 100 Characters and not null |
| OrderID | Integer | A Unique identify for each order. | Must be unique, >= 0 and not null |
| CustomerID | Integer | A Unique identify for each customer. | Must be unique, >= 0 and not null |
| DeliveryAddress | String | The Address for the order to be delivered to. | Max Length 200 Characters and not null |
| BillingAddress | String | The Billing address required by law to invoice the purchase to. | Max Length 200 Characters |
| DeliveryRequest | String | Any special requests for the delivery such as ATL. | Max length 500 Characters |
| Total | Float | The price of the order. | Must be >= 0 and not null |
| Name | String | The Customer name used for shipping and addressing them. | Max Length 50 Characters |

Outline of Queries

Queries required for this project include but are not limited to:

Create Orders

Create Customers

Select Products

Select Product in an order

Add/Remove Items from an order

Insert all products

# Evaluation

## Reflect

My software solution to part 1 meets the criteria i outlined in part one reasonably well, storing all the required data in the backend. Although it lacks the display of this data in the frontend solution. The Flask based python frontend allows for customers to add or remove products to or from an order, is reasonably visually appealing through its minimalist layout and displays all the products stored within the SQLite Database. Where it does lack is in finding the cheapest price from suppliers as currently the Database only has one suppliers per product, although this would be done through the use of an aggregate function such as MIN().

The Structure of the finished Database is quite different to the initial ER Diagram outlined. Such as storing the Delivery address instead of the Delivery ID in the orders table and the billing address in the Customers table. This is one of the extra features implemented for individuals that don't want the same billing and delivery address.

## Bugs/Limitations

Check the issues tab on GitHub

## Retrospective

In order to optimize the process for next time there are a few things I would do differently, to start off I would also plan out the software component of the project, in order to create a smoother cleaner set of code and UI, another change I would make is to sketch out the UI clearly in order to avoid making critical decisions while developing. While Things that work well include creating the ER Diagram before developing the Database and using python3 as well as BS4 to scrape the data.

# References

ADDITIONAL SYLLABUS SUPPORT BOOKLET. (2024, February 6).

<https://senior-secondary.scsa.wa.edu.au/__data/assets/pdf_file/0003/1090875/Year-11_12_Computer-Science_ATAR_Additional-syllabus-support-booklet.PDF>

Black Diamond Equipment Australia | Climbing Equipment & Outdoor Gear. (2024). Black Diamond Australia.

<https://www.blackdiamondequipment.com.au/>

COMPUTER SCIENCE ATAR COURSE Year 12 syllabus. (2024).

<https://senior-secondary.scsa.wa.edu.au/__data/assets/pdf_file/0008/1017782/Computer-Science-Y12-Syllabus-ATAR-20192.PDF>

HTML Tables generator – TablesGenerator.com. (n.d.).

[Www.tablesgenerator.com](http://Www.tablesgenerator.com)

[https://www.tablesgenerator.com/html\_tables#](https://www.tablesgenerator.com/html_tables)

ilovepdf.com. (2019). iLovePDF | Online PDF tools for PDF lovers. ILovePDF - Online Tools for PDF.

<https://www.ilovepdf.com/>

La Sportiva Spa. (2024). La Sportiva Spa.

<https://www.lasportiva.com/au>

md2pdf - Markdown to PDF. (2024). Netlify.app.

<https://md2pdf.netlify.app/>

MyBib. (2018, July 13). MyBib bibliography generator. MyBib.com.

<https://www.mybib.com>

SQL Online Compiler - for Data Science. (n.d.). Sqliteonline.com.

<https://sqliteonline.com/>