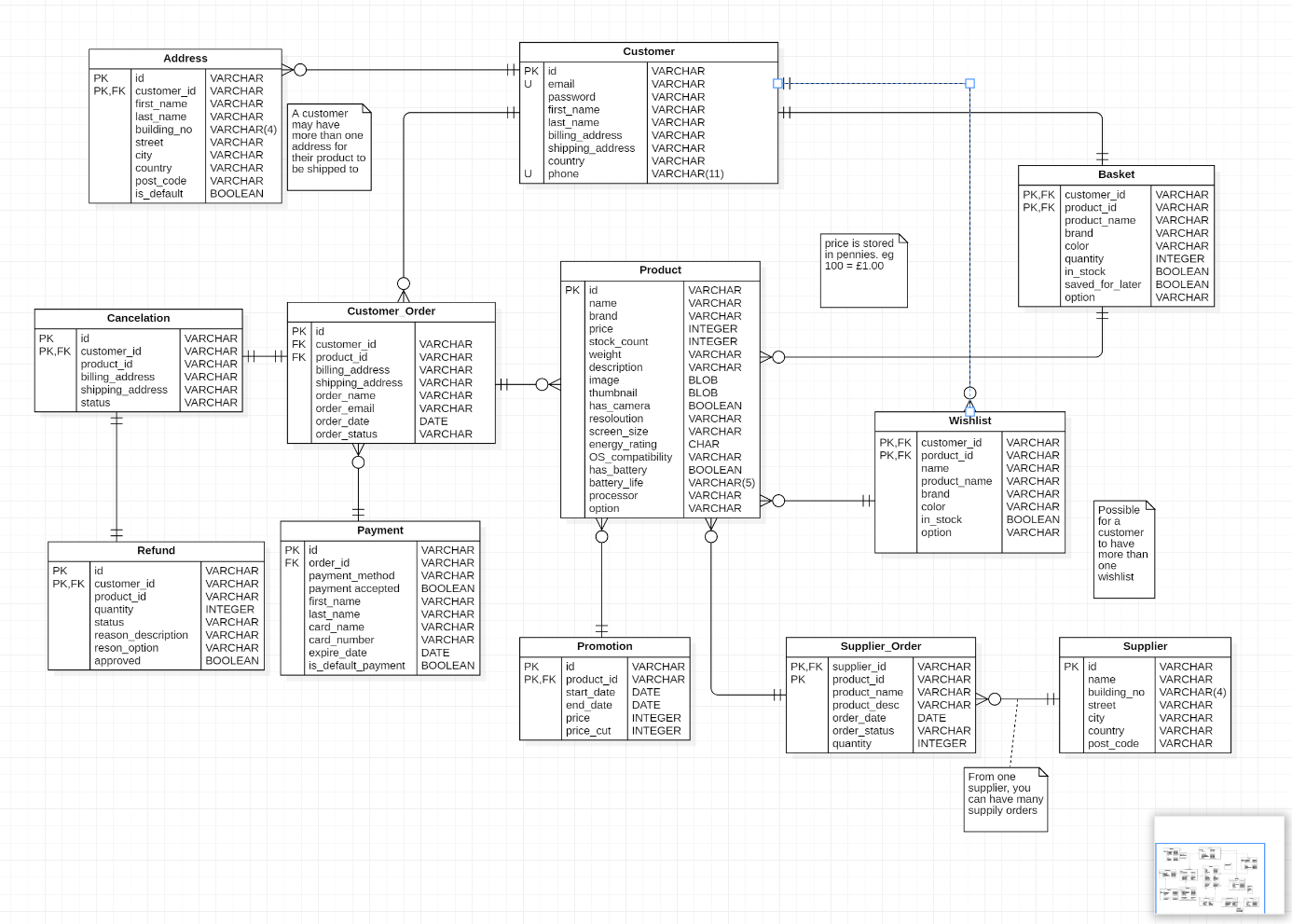
# Database

## Introduction

For this part of the report, I will go into detail about the database fits into the project. The database plays a crucial part within the project as it is responsible for holding the information about the customer details

## Entity Relationship Diagram

The image below is the entity relationship diagram, which illustrates the entities that will be used within the database and the relationship they have with other entities. Each entity has attributes which shows what data they hold and the type.



## Tables

### Customer & Address

As soon as a customer signs up to website the Customer and Address table get populated with data, each customer is given a unique id which is in both the Customer and Address tables. This id is a primary key in the Customer table. The address table has it’s own id and a composite primary key using the id and the customer\_id, because it is possible that one customer may have more than one address or two customers may live at the same address. To know which address to deliver to, the Address table has an is\_default attribute to tell which address an order should be delivered to.

### Product & Promotion

The Product table contains all the information on products that are received from the supplier and contains the most attributes. Not all of the attributes will contain data because it is very unlikely that a product will meet all the parameters (e.g a processor doesn’t have a battery life). All products will be uniquely identified with the id attribute which is also the primary key. Weather the product is in stock and the amount of stock will be shown on the website. If a product is not in stock then an estimated delivery date will be shown on the website.

For products that have a promotion applied to them are apart of the Promotion table. A composite primary key is made using the id and the product\_id attributes. All promotions have a start\_date and an end\_date for the beginning and end of a promotion. This makes it easier it apply a promotion if a promotion is seasonal. The price in the Promotion table is determined by the price in the Product table and the price\_cut in the Promotion table.

### Basket & Wishlist

Each customer can have one basket in the same way a customer enters a supermarket and picks up a basket or a trolly to do their shopping. The Basket table has a composite primary key composed of customer\_id and product\_id. The saved\_for\_later attribute is for items that are not added to the total order if the customer decided they wanted to purchase an item at a later date and not look for it on the website again.

I is possible to a customer to have more than one wish list. The Wishlist table has a composite primary key made from the id and the customer\_id attributes. Each wish list can be named by the customer if they chose to. Products that are in the wish list can have a description and a quantity to let other customers that use the website know how much to buy.

### Customer Order

Because each customer can have multiple orders, it makes sense for the Customer\_Order table to have an id attribute, this id attributes acts as a unique identifier to easily identify a customers order. For the Customer\_Order table to get information about the customer and the product, the Customer\_Order table has the attributes customer\_id and product\_id which act as foreign keys that relate to the Customer table and the Product table.

### Payment

Customers can have multiple payment methods for an order, however only one payment method will be used for each customer’s order. Payment can be differentiated by the id attribute and information about the customer can be accessed via the order\_id attribute which is a foreign key. When a customer is about to pay for an item, they can choose which way they would like to pay for their order. The customer can also set a default payment method if they want to speed up the payment process. If the desired payment method has passed the expire\_date attribute the payment is rejected.

### Cancelation & Refund

### Supplier & Supplier Order

## Database Management System

For the database management system, I created a simple command line application in python for performing basic queries. When a customer phones the customer service line about a problem with their account, customer services can solve the problem without having full access to SQL queries, any serious problems can be passed to a administrator to be solved

## Functionality