Advanced Relational Database Development Report

George Pake- SID2109071



Contents

[1 Project Proposal 3](#_Toc164164893)

[1.1 Introduction 3](#_Toc164164894)

[1.2 Key database users 3](#_Toc164164895)

[1.3 Explanation of why a database-driven application is necessary for the business 4](#_Toc164164896)

[1.4 Business questions 4](#_Toc164164897)

[2 Database Design and Implementation 5](#_Toc164164898)

[2.1 Conceptual Model 5](#_Toc164164899)

[2.1.1 Initial list of entities and attributes 5](#_Toc164164900)

[2.1.2 Conceptual ERD in Chen format 6](#_Toc164164901)

[2.1.3 Draft Schema 7](#_Toc164164902)

[2.2 Discussion of assumptions 8](#_Toc164164903)

[2.3 Final Database Model 9](#_Toc164164904)

[2.3.1 Logical ERD 9](#_Toc164164905)

[2.3.2 Updated Schema 11](#_Toc164164906)

[2.4 Inserting data into tables 12](#_Toc164164907)

[3 Query design, construction and testing 14](#_Toc164164908)

[3.1 Query 1 14](#_Toc164164909)

[3.2 Query 2 15](#_Toc164164910)

[3.3 Query 3 16](#_Toc164164911)

[3.4 Query 4 16](#_Toc164164912)

[3.5 Query 5 17](#_Toc164164913)

[4 SQL Programming 18](#_Toc164164914)

[4.1 Stored Procedures 18](#_Toc164164915)

[4.1.1 Procedure 1- Search Engine 18](#_Toc164164916)

[4.1.2 Procedure 2- Customer spending 18](#_Toc164164917)

[4.1.3 Procedure 3- VAT Report for item category 19](#_Toc164164918)

[4.2 Triggers 20](#_Toc164164919)

[4.2.1 Trigger 1- Recording previous item prices when they are updated 20](#_Toc164164920)

[4.2.2 Trigger 2- Recording details of customers that ‘leave’ the business 21](#_Toc164164921)

Figures

[Figure 1- Conceptual ERD for my proposed database in Chen format 6](#_Toc164164838)

[Figure 2- Logical ERD for my proposed database 9](#_Toc164164839)

[Figure 3- Structure of customers table 9](#_Toc164164840)

[Figure 4- Structure of delivery routes table 9](#_Toc164164841)

[Figure 5- Structure of items table 10](#_Toc164164842)

[Figure 6- Structure of item categories table 10](#_Toc164164843)

[Figure 7- Structure of orders for delivery table 10](#_Toc164164844)

[Figure 8- Structure of order items table 10](#_Toc164164845)

# 1 Project Proposal

## 1.1 Introduction

The proposed database is to be used as a back end for the home delivery service of a large supermarket. Many customers can place one order, but each delivery van will contain multiple customers’ orders. Each product is categorised making it easier to find and retrieve information about it. There will be specific delivery routes depending on where the customer lives. There will be one designated route for multiple postal codes.

The Better Supermarket opened a year ago, and is based in a small but growing town with a population of 5000 people. There are 500 registered customers, and 1000 orders were placed in the last year. The town already has one supermarket, but due to the increasing amount of people moving in, there has been a need for an additional supermarket in the town. It is expected that the town’s population will almost double within the next decade.

The delivery service covers nine different postcodes across five routes. This represents approximately half of the town. The other half of the town is represented by the original supermarket. The supermarket will not to deliver to postcodes which aren’t listed within any of the listed delivery routes.

Items are kept on the shelves in the supermarket shop floor, and they are hand picked by a designed team responsible for packing customers’ orders. The team will have access to the database and shortly know what exactly to pick, and what delivery van to load the order into.

## 1.2 Key database users

The key groups of users that will interact with the database are:

* Customers- they will be creating accounts with their personal details on, such as their address and contact number, and modifying data if necessary. These will need to be accessed by delivery drivers and managers in order to plan delivery routes and drops. They also will be adding products to their orders, which will need to be picked by shop floor staff. The quantity of products available will need to be regularly updated automatically so that customers are notified if a product is out of stock.
* Product manufacturers- The most important attributes that they will need to access are the quantity sold, date sold and age ranges of the customers ordering it. They will want to know how many times their product is getting sold, what profit they are making etc. Knowing the dates that each product is sold would be useful so they know how many to produce and send to the supermarket at each time of the year. If their products is selling well then they could slightly increase the price, whereas if it isn’t then they could decrease the price. The age ranges of customers could be important as well so that they could potentially modify their product to make it more suitable for the age ranges.
* Financial managers- It is important for them to see data about every product: category, quantity sold, price etc. If a product isn’t selling well then they might take it off the shelf and replace it with a new or existing product. On the other hand, if a product is selling very well then they might order more in, or make more space for that specific product. Having knowledge of quantity sold is especially important for calculating financial information like turnover and profit margin.
* Delivery van fleet managers- The attributes that are important for them are customer addresses (postcode), and the sum of each distinct postal code. They will be provided with information that can be used to optimize delivery routes. If there are a small quantity of orders from one postal code, then only a small number of vans will need to be allocated to the corresponding route, and vice versa with a large quantity from one postal code. Additionally, they will be able to know the rough distance travelled on each route (and how frequent the route is used), meaning that dates of when to service or replace vans can be estimated.
* Stock controllers/replenishers- they will need to be notified when a product needs stocking again. To do this, they must be alerted after every product sale, so that they can keep track of how many of each item are on the shelf. By accessing such information, stock controllers can tell financial managers, or whoever is responsible for ordering in new stock, when a product is low in stock and thus have more of that item ordered in. In the assumptions section below, it is assumed that items will never be out of stock, so the role is important for fulfilment of this assumption

## 1.3 Explanation of why a database-driven application is necessary for the business

A supermarket is a large business with huge amounts of customers ordering home deliveries every day. The company’s turnover will be significantly than that of a small convenience store or similar sized retail business. Despite the supermarket receiving 1000 orders for the past calendar year, it is expected that it will grow rapidly over the years, so using an order book to track orders would be inefficient, ineffective and time consuming. A database provides a structured and organized way to store, manage, and retrieve relevant records of data. Orders can be processed smoothly and seamlessly, meaning that there is accurate and up to date information on the availability of products and statuses of customer orders. Whilst it might be appropriate to manually record data within a smaller sized business, a larger sized business where customer demand is much higher means that data must be able to be accessed and retrieved instantly, and updated in real time. Storing data in a scalable database makes it easier to analyse trends and optimize company procedures in the future. An example of this could be that the number of orders received from a few postal codes on one delivery route have doubled. To respond to this, an additional van has been added to the route, resulting in decreased waiting time which improves customer service and relations with the business.

## 1.4 Business questions

Some questions about the business that could be answered by querying the database are:

1. **How many orders had a slower than average delivery time and were delivered in December?**
2. **Who are the company’s top 3 spending customers?**
3. **Which 3 product categories generated the company the highest amount of revenue in the last year, and what percentage did each make up?**
4. **Which customers placed 12 or more orders from us, and how many did they place?**
5. **Have any customers under the age of 18 bought alcohol?**

# 2 Database Design and Implementation

## 2.1 Conceptual Model

### 2.1.1 Initial list of entities and attributes

|  |
| --- |
| **Customers** |
| Customer ID |
| Firstname |
| Lastname |
| Postcode |
| Date of Birth |
| Telephone Number |

|  |
| --- |
| **Orders** |
| Order ID |
| Customer ID |
| Date Order Placed |
| Delivery Postcode |

|  |
| --- |
| **Order Items** |
| Order Item ID |
| Order ID |
| Item ID |
| Quantity |
| Price |

|  |
| --- |
| **Deliveries** |
| Delivery ID |
| Order ID |
| Delivery Date |
| Delivery Route ID |

|  |
| --- |
| **Delivery Routes** |
| Delivery Route ID |
| Postcode |

|  |
| --- |
| **Items** |
| Item ID |
| Item Name |
| Price |
| Category ID |

|  |
| --- |
| **Item Categories** |
| Category ID |
| Category Name |

### 2.1.2 Conceptual ERD in Chen format

A diagram of a product

Description automatically generated

Figure 1- Conceptual ERD for my proposed database in Chen format

The Chen diagram in Figure 1 shows the relationships between entities to others. Only key attributes are included, with the primary key of each underlined. All of these entities become will become tables in the proposed database. Additionally, the relationship between orders and products will become an entity named “Order Item” as it is a many to many relationship.

* 1 and only 1 customer can place as many orders as they like
* 1 to many orders contain 1 to many items. E.g. some items available on the shop floor wont need to be shipped, as customers might not order them
* 1 delivery includes 1 or many orders. A delivery van cannot have 0 orders, as it then cannot be classed as a delivery
* Many deliveries are assigned to 0 or 1 delivery routes. There will be many vans on one route, but if a postcode isn’t close enough to the store, items cannot be delivered to that customer
* Many items (or products as mentioned above) belong to one category.

### 2.1.3 Draft Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customers** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Customer ID | int | Y | N | Y |  |
| Firstname | varchar(32) |  | N |  | Size of 32- caters for customers with very long names |
| Lastname | varchar(32) |  | N |  | “” |
| Postcode | varchar(8) |  | N |  | 8 digits is sufficient e.g. “AB12 CDE” |
| Date Of Birth | date |  | N |  |  |
| Telephone Number | varchar(16) |  | N | Y | No two people have the same number |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Orders** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Order ID | int | Y | Y | Y |  |
| Customer ID | int |  | Y | Y | Don’t auto increment as there could be more than one order per customer |
| Date Order Placed | date |  | Y |  |  |
| Delivery Postcode | varchar(8) |  | Y |  | Same postcode as in Customers table |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order Items** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Order Item ID | int | Y | Y | Y |  |
| Order ID | int |  | Y | Y |  |
| Item ID | int |  | Y | Y |  |
| Quantity | int |  | Y |  | You can only buy whole items, int is the required data type |
| Total Price | float(6,2) |  | Y |  | Max 6 digits total is suitable, as not selling items at the top end of the scale |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Items** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Item ID | int | Y | Y | Y |  |
| Item Name | varchar(255) |  | Y |  | Size of 255- detailed item names used. E.g. “Ben’s Original Egg Fried Microwave Rice 220g” not “Rice” |
| Price | float(6,2) |  | Y |  | See Order Items table |
| Category ID | int |  | Y |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item Categories** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Category ID | int | Y | Y | Y |  |
| Category Name | varchar(64) |  | Y | Y | Size of 64- large enough to fit in longer category names, but doesn’t need to be any larger as names won’t be a description, just a few words length max |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Deliveries** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Delivery ID | int | Y | Y | Y |  |
| Order ID | int |  | Y |  |  |
| Delivery Date | date |  | Y |  |  |
| Delivery Route ID | int |  | Y |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delivery Routes** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Delivery Route ID | int | Y | Y | Y |  |
| Delivery Postcode | varchar(4) |  | Y |  | Only the first part of the postcode is relevant (e.g. AB12). Unlike in the Customers Table, a larger size isn’t required here |

## 2.2 Discussion of assumptions

Throughout the development of the model, I have made the following assumptions:

* Every customer receives their delivery by van i.e. not on foot bike etc. They do not live within very close proximity to the store where it is unrealistic for vans to deliver to.
* The customer’s home postcode is the same as the delivery postcode. This makes it simpler to manage delivery and delivery route data, as there is less data that needs to be accessed.
* Postcode is not a unique attribute. I am assuming that there could be more than one customer placing an order to be delivered to the same address e.g. family members or work colleagues.
* The customer entity does not contain their address. By adding an address attribute, this would mean that there’d be transitional dependency between the address and the postcode, as the postcode is dependent on the address. Another table would need to be created as a result. It will be assumed that the delivery driver can navigate to the customer’s home via the use of the postcode, and if they cannot find it, they could contact them via the telephone number that they’ve provided.
* Every item in the items table is available to be sold and packed in orders. This keeps the model simpler, as by tracking whether items are available for sale requires another attribute to flag the status, and another table to track the changes.
* There will never be items out of stock. The focus is on the delivery side of the supermarket rather than the storage. In the real world, customers would be able to modify their orders and select a similar item as a replacement. By removing this aspect, the complexity of the model is reduced as the changes won’t have to be tracked.
* Items will be the same price all year round. Whilst there are usually promotions depending on the time of the year, this won’t be the case here. The reason for this decision is so that there aren’t random fluctuations in sales data and prices of orders. There will be more consistency within the database. Customers also won’t be able to get an overall discount on their orders.
* To reduce complexity, products won’t be categorized into more than one category. This would be harder when trying to retrieve data about one product. Though there won’t be any items that are uncategorized as well.
* Each order will only be delivered once i.e. not in separate trips to deliver part of the order. By doing this, it reduces complexity in the deliveries table, as the order number which is the primary key would have duplicates.

## 2.3 Final Database Model

### 2.3.1 Logical ERD

A diagram with text and a diagram

Description automatically generated with medium confidence

Figure 2- Logical ERD for my proposed database

Figure 2 above shows the Entity Relationship Diagram (ERD) for my proposed database. Crows Foot notation is used to show the different relationships between the entities. I have shown each entity and their entities, including data type and size, of which the explanations as to why I have chosen each are shown in the Database Schema below. The primary and foreign keys are noted. The differences between the Chen diagram above are that the Order Items table is now shown here, and not as a “contains” relationship.

Below are the structures for the six tables shown above in PHP My Admin.

A screenshot of a computer

Description automatically generated

Figure 3- Structure of customers table



Figure 4- Structure of delivery routes table

A screenshot of a computer

Description automatically generated

Figure 5- Structure of items table

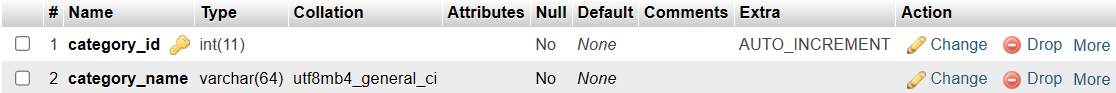


Figure 6- Structure of item categories table

A screenshot of a computer

Description automatically generated

Figure 7- Structure of orders for delivery table

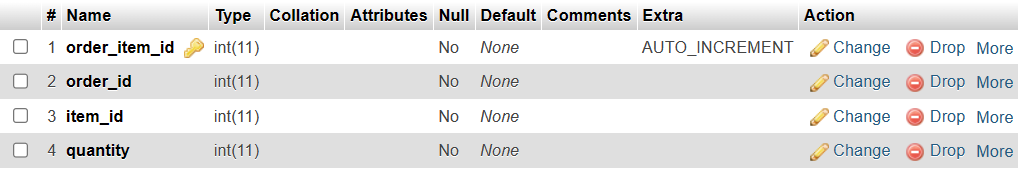


Figure 8- Structure of order items table

### 2.3.2 Updated Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Customers** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Customer ID | int | Y | N | Y |  |
| Firstname | varchar(32) |  | N |  | Size of 32- caters for customers with very long names |
| Lastname | varchar(32) |  | N |  | “” |
| Postcode | varchar(8) |  | N |  | 8 digits is sufficient e.g. “AB12 CDE” |
| Date Of Birth | date |  | N |  |  |
| Telephone Number | varchar(16) |  | N | Y | No two people have the same number |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Orders for delivery** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Order ID | int | Y | Y | Y |  |
| Customer ID | int |  | Y | Y | Don’t auto increment as there could be more than one order per customer |
| Date Order Placed | date |  | Y |  |  |
| Delivery Date | date |  | Y |  |  |
| Delivery Route ID | int |  | Y |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order Items** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Order Item ID | int | Y | Y | Y |  |
| Order ID | int |  | Y | Y |  |
| Item ID | int |  | Y | Y |  |
| Quantity | int |  | Y |  | You can only buy whole items, int is the required data type |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Items** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Item ID | int | Y | Y | Y |  |
| Item Name | varchar(64) |  | Y |  | Size of 64- no item name is longer than that. All names are basic e.g. frozen pizza rather than descriptive which would include brand, weight etc. |
| Price | float(5,2) |  | Y |  | No item is more expensive than £999.99, so this length is sufficient |
| Category ID | int |  | Y |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Item Categories** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Category ID | int | Y | Y | Y |  |
| Category Name | varchar(64) |  | Y | Y | Size of 64- large enough to fit in longer category names, but doesn’t need to be any larger as names won’t be a description, just a few words length max |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Delivery Routes** | | | | | |
| Attribute | Type | AI | NN | Unique | Comments |
| Delivery Postcode | varchar(4) |  | Y |  | Only the first part of the postcode is relevant (e.g. AB12). Unlike in the Customers Table, a larger size isn’t required here |
| Delivery Route ID | int | Y | Y | Y |  |

There have been multiple changes to the schema to ensure that all the data is in 3NF:

* The Deliveries table has been removed, and the attributes have been added to the Orders for Delivery table (New name). The primary keys would have been the same (Order ID would have been the primary key in Deliveries as it is unique, Delivery ID wasn’t needed anyway). Merging the two tables into one makes it easier to insert data, as you would have to check each record of the Orders table before inserting it into the Deliveries table e.g. checking that the date order placed in the orders table is before the delivery date in the deliveries table- it is now visually easier to check such things
* Delivery Postcode has been removed from the Orders table. As mentioned earlier, I have assumed that the home is the same as their preferred delivery address. There is transitive dependency between the Customer ID and the Delivery postcode i.e. you can previously work out what a customer’s postcode is if you know their ID.
* Total Price has been removed from the Order Items table. There is transitive dependency between it and the Item ID.
* In the Delivery Routes table, Delivery Postcode is now the primary key. This is because each delivery route includes more than one postcode, so there would be duplicate values if the Delivery Route ID was the primary key.

## 2.4 Inserting data into tables

Customers table before

A screenshot of a computer

Description automatically generated

Customers table after

A screenshot of a computer

Description automatically generated

Delivery routes table before

A screenshot of a computer

Description automatically generated

Delivery routes table after

A screenshot of a computer

Description automatically generated

Items table before

A screenshot of a computer code

Description automatically generated

Items table after

A screenshot of a computer program

Description automatically generated

Item categories table before

A screenshot of a computer

Description automatically generated

Item categories table after

A screenshot of a computer

Description automatically generated

Order items table before

A screenshot of a computer

Description automatically generated

Order items table after

A screenshot of a computer program

Description automatically generated

# 3 Query design, construction and testing

## 3.1 Query 1

**How many orders for each delivery route had a slower than average delivery time and were delivered in December?**

The query is relevant to the business as it is important for those who manage the delivery vans to analyse the performance of the fleet. For example, delivery route 3 performed poorly in December-there were 13 orders for that route which had a delivery time slower than the average. The fleet manager could note the information and either upgrade to a larger van or purchase an additional van to avoid long waiting times for deliveries.

A computer code with text

Description automatically generated

A screenshot of a computer

Description automatically generated

## 3.2 Query 2

**Who are the company’s top 3 spending customers?**

The query is relevant to the business as identifying customers who are responsible for large portions of a company’s income can help improve sales strategies. For example, you could identify the items bought from these customers and potentially adjust the prices to increase revenue. Items that are selling well, but the company are not profiting from could have their prices increased as well.

A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

## 3.3 Query 3

**Which 3 product categories generated the company the highest amount of revenue in the last year, and what percentage did each make up?**

This query is relevant to the business as store managers can identify where most of their income is coming from, and then potentially make room for more products of the relevant category. As computer accessories make up approximately half of the company’s revenue, similar products can be put up for sale. Running this query shows that computer accessories are profitable, and it would be worth spending money on importing products and expanding shelves to stock the products available for delivery.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

## 3.4 Query 4

**Which customers placed 12 or more orders from us, and how many did they place?**

This query is relevant to the business as it relates to customer loyalty. Many large companies offer reward schemes to loyal customers e.g. Tesco Clubcard. Knowing which customers are loyal means that companies can know how effective their loyalty schemes are. If there aren’t enough customers purchasing a large number of orders, then there is not a point of introducing a reward scheme.

**A screenshot of a computer code

Description automatically generated**

**A close-up of a computer screen

Description automatically generated**

## 3.5 Query 5

**Have any customers under the age of 18 bought alcohol?**

This query is relevant to the business as it is important for companies to follow legal procedures. If a customer under the age of 18 has purchased alcohol, then they have broken the law. Having knowledge of the customer’s personal details helps to identify them for prosecution, and knowing which item was purchased means that staff can restock it and know the updated quantity of the item on the shelf.

A computer code with text

Description automatically generated



# 4 SQL Programming

## 4.1 Stored Procedures

### 4.1.1 Procedure 1- Search Engine

This procedure, which represents a simple search engine, is relevant to the business as customers can use it to speed up placing their order. If they are on a budget, having parameters to input the minimum and maximum value of a product can help them. Searching for the category name rather than the name of the actual product is more useful, as similar products can be displayed if the user wasn’t satisfied with the product they originally intended to purchase.

A computer code with text

Description automatically generated



A table with text and numbers

Description automatically generated

### 4.1.2 Procedure 2- Customer spending

This procedure is relevant to the business. Staff who deal with loyalty can search about individual customers and boost relationships by offering perks such as discounts. With the database including many records of customers, staff can search for part of their name if they cannot remember the full name.

A computer code with text

Description automatically generated



A screenshot of a computer

Description automatically generated

### 4.1.3 Procedure 3- VAT Report for item category

This procedure is relevant to the business as it is important for businesses to know the value-added tax of each product. Showing this will help to calculate the total amount of due VAT. Adding a parameter to order by either the item name or price helps to identify a specific product. A large amount of results will be displayed for each category.

A computer code with many colored text

Description automatically generated with medium confidence



A table with numbers and symbols

Description automatically generated with medium confidence

## 4.2 Triggers

### 4.2.1 Trigger 1- Recording previous item prices when they are updated

This trigger is relevant to the business as it creates an audit trail of previous product prices. Providing historical data of different products helps to understand and anticipate market developments. The data can be used in discussions with suppliers, which is necessary in order to maintain a high profit margin.

SQL Code to create trigger

A computer screen shot of a computer code

Description automatically generated

Item History table before



Items table before

A screenshot of a computer

Description automatically generated

Item History table after



Items table after

A screenshot of a computer

Description automatically generated

### 4.2.2 Trigger 2- Recording details of customers that ‘leave’ the business

This trigger is relevant to the business as it is useful for management and HR staff to access data about previous employees. This is especially important whilst dealing with legal cases, such as judging whether employees were paid fairly.

SQL Code

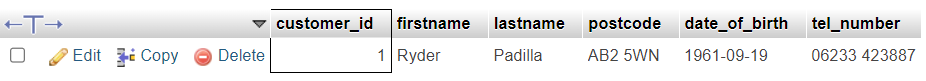
A screenshot of a computer code

Description automatically generated

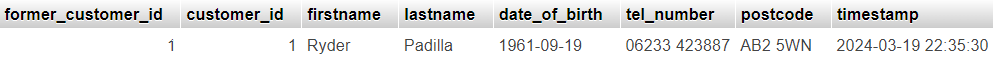
Former customers table before



Customers table before



Former customers table after



Customers table after

