Setting Up a Database for a Small Retail Business:

1. Understanding the Business Requirements

The first step is identifying the data the shop needs to manage. Here are some examples of what the database will store:

- **Inventory**: Products like milk, bread, and detergent, with details such as quantity in stock and price.
    - Example: Milk (10 units in stock, £1.50 each), Bread (20 units in stock, £1.00 each).
- **Sales**: Each sale records the date, the products sold, and the quantities.
    - Example: On 15 January, 2 bottles of milk and 1 loaf of bread were sold.
- **Customer Information**: Contact details for customers and their loyalty program status.
    - Example: Jane Doe, email: jane.doe@example.com, loyalty points: 50.

Users of the database:

- The **Shop Manager** needs to track stock and sales trends.
- The **Shop Staff** needs to record transactions and check loyalty points.
- The **IT Support** team will maintain the database and ensure it runs smoothly.

2. Designing the Database Schema

Think of the database as a collection of tables, each representing one type of information. For this shop, you need these tables:

- **Products Table**:
    - Example columns: Product ID, Product Name, Category, Price, Stock Level.
    - Example data:

Product ID	Product Name	Category	Price	Stock Level
1	Milk	Dairy	£1.50	10

Product ID	Product Name	Category	Price	Stock Level
2	Bread	Bakery	£1.00	20

- **Customers Table**:
  - Example columns: Customer ID, Name, Email, Loyalty Points.
  - Example data:

Customer ID	Name	Email	Loyalty Points
1	Jane Doe	jane.doe@example.com	50

- **Sales Table**:
  - Example columns: Sale ID, Product ID, Customer ID, Date, Quantity.
  - Example data:

Sale ID	Product ID	Customer ID	Date	Quantity
1	1	1	2025-01-15	2

  -

Relationships Between Tables:

- **Products and Sales**: Each sale will refer to a specific product to update stock and calculate revenue.
- **Customers and Sales**: If a customer is part of the loyalty program, their transactions should connect to their record.

3. Implementing the Database

To create the database, use a tool like **Microsoft Access** or MYSQL:

1. Open a new table.
2. Add column headers for the details (e.g., Product Name, Price, Stock Level).
3. Populate the rows with sample data as shown in the examples above.

4. Populating the Database

Input data manually using forms or spreadsheet interfaces. For example:

- In the **Products Table**, enter:
    - Row 1: Milk, Dairy, £1.50, 10.
    - Row 2: Bread, Bakery, £1.00, 20.
- In the **Customers Table**, enter:
    - Row 1: Jane Doe, jane.doe@example.com, 50 loyalty points.

Alternatively, import data from existing files (like Excel). Most database tools allow you to drag and drop spreadsheets into the system.

5. Maintaining the Database

To keep the database accurate and useful:

1. **Regular Updates**:
    - Example: After selling 2 bottles of milk, reduce the stock level from 10 to 8 in the Products Table.
2. **Monitor Data Quality**:
    - Check for errors like duplicate customer entries or missing product details.
3. **Backups**:
    - Export the database weekly as a backup file (e.g., save it as an Excel file or database file).
4. **Data Security**:
    - Restrict access so only authorised staff can edit tables. For example, the manager can update product prices, but staff can only record sales.