Table of Contents

[1. Clearly explain why each entity needs to be included. For example if you have a staff entity explain how this benefits the overall dataset. 3](#_Toc151387836)

[2. Identify the relevant entities of the dataset with their respective attributes, entity types (strong or weak) and primary keys. 5](#_Toc151387837)

[3. Identify the relationships, cardinalities and constraints with supporting business rules and assumptions. 5](#_Toc151387838)

[4. Draw an ER diagram for the system depicting the entities, relationships, cardinalities, participations using your preferred ERD notation. 5](#_Toc151387839)

[5. Convert the entities in the conceptual design into logical dataset(relational model). Use a logical layout of entities which is easily understood by any reader. Ensure that tables are in the correct format to allow for the best possible representation of data. 5](#_Toc151387840)

Task One - Database Design

# Clearly explain why each entity needs to be included. For example if you have a staff entity explain how this benefits the overall dataset.

For my lawnmower sales office database system, the following entities will be included:

* **Customer**: Customers are crucial for any business.

This entity allows the system to store and manage information about the customers.

* Address: The Address entity stores addresses information. Including it in the database allows the reuse of address details across entities like Customer and potentially others, allowing data consistency and avoiding redundancy.
* Product: Products represent the items being sold, in this case, lawnmowers.

This entity allows us to store details such as model, manufacturer, type, price.

This is essential to manage inventory, track sales, and provide accurate product information to customers.

* **Stock**: Stock management is crucial to track inventory levels and ensure products are available for customers. The Stock entity allows the system to store and manage information about the quantity of each product in stock.
* **Order**: The Order entity is essential to tack and manage orders.

Connects customers with the products they purchase, along with relevant details such as order date and total amount. This allows the business to fulfil orders, and generate invoices.

* OrderItem: allows a detailed breakdown of each order.

It specifies the quantity and subtotal for each product within every order. It also serves as an intermediate table for normalisation.

* **Staff**: Staff members play a vital role in managing day-to-day operations, as customers, they are also a vital party of any business.

Staff entity allows information about staff members, such as their roles, contact details, and performance, to be stored on the database. This aids in human resource management and assigning responsibilities, for example.

* Role: The Role entity stores information about different staff. Adding it to the database allows for clear associations between staff members and their roles, assisting human resource management and assigning responsibilities.
* Supplier: Suppliers provide the lawnmowers that the sales office sells. This allows tracking and managing suppliers, including contact information and products supplied. This information is helpful for inventory management reordering;
* Payment: tracks the financial aspect of transactions. It includes details such as payment date, amount, and method. This information is essential for financial record-keeping, reconciliation, and ensuring that payments are properly recorded and processed.
* Store: Multiple locations or stores may exist for a lawnmower sales office. Including a Store entity helps manage information about different physical locations, facilitating inventory management.
* **Delivery**: The Delivery entity tracks the process of delivering products to customers. It includes information such as delivery date and status. This is crucial for logistics management, order fulfilment, and providing customers with accurate delivery information.

# Identify the relevant entities of the dataset with their respective attributes, entity types (strong or weak) and primary keys.

Customer:

Attributes:

* CustomerID (Primary Key)
* FirstName
* LastName
* ContactNumber
* Email
* AddressID (Foreign Key)

Entity Type: Strong

Address

Attributes:

* AddressID (Primary Key)
* Street
* City
* Country
* PostCode

Entity Type: Strong

**Product**:

Attributes:

* ProductID (Primary Key)
* SupplierID (Foreign Key)
* Model
* Manufacturer
* Type
* Price

Entity Type: Strong

Order:

Attributes:

* OrderID (Primary Key)
* CustomerID (Foreign Key)
* AddressID (Foreign Key)
* PaymentID (Foreign Key)
* StoreID (Foreign Key)
* OrderDate
* OrderTotalAmount

Entity Type: Strong

OrderItem

Attributes:

* OrderItemID (Primary Key)
* OrderID (Foreign Key)
* ProductID (Foreign Key)
* Quantity
* Subtotal

Entity Type: Weak (Dependent on Order)

Staff

Attributes:

* StaffID (Primary Key)
* StoreID (Foreign Key)
* FirstName
* LastName
* RoleID (Foreign Key)
* ContactNumber
* AddressID (Foreign Key)

Entity Type: Strong

Role

Attributes:

* RoleID (Primary Key)
* RoleName

Entity Type: Strong

Supplier

Attributes:

* SupplierID (Primary Key)
* Name
* ContactNumber

Entity Type: Strong

Payment

Attributes:

* PaymentID (Primary Key)
* OrderID (Foreign Key)
* PaymentDate
* Amount
* PaymentMethodID (Foreign Key)

Entity Type: Weak (Dependent on Order)

PaymentMethod:

Attributes:

* PaymentMethodID (Primary Key),
* MethodName

Entity Type: Strong

Store:

Attributes:

* StoreID (Primary Key)
* Location
* Manager
* ContactNumber

Entity Type: Strong

Delivery:

Attributes:

* DeliveryID (Primary Key)
* OrderID (Foreign Key)
* DeliveryDate
* Status

Entity Type: Weak (Dependent on Order)

**Stock**:

Attributes:

* StockID (Primary Key)
* StoreID (Foreign Key)
* ProductID (Foreign Key)
* StockLevel

Entity Type: Weak (Dependent on Store and Product)

# Identify the relationships, cardinalities and constraints with supporting business rules and assumptions.

1. Customer and Address:

Relationship: One-to-One

Cardinality: One Customer has One Address.

Constraints: A customer can have only one address, and each address is associated with only one customer.

Business Rule: Each customer should have a valid address for contact and correspondence.

1. Customer and Order:

Relationship: One-to-Many

Cardinality: One Customer can place Many Orders.

Constraints: An order is placed by one customer, but a customer can place multiple orders.

Business Rule: Customers can make multiple purchases, and each purchase is recorded as a separate order.

1. Order and OrderItem:

Relationship: One-to-Many

Cardinality: One Order can have Many OrderItems.

Constraints: An order can consist of multiple items, and each order item belongs to one order.

Business Rule: Each order can contain multiple items (products), and each item is associated with a specific order.

1. Store and Order:

Relationship: One-To-Many

Cardinality: One Store can be associated with Many Orders.

Constraints: Each order is associated with one store, but a store can have multiple orders.

Business Rule: Each order is placed at a specific store, and a store can have multiple orders.

1. Staff and Store:

Relationship: Many-to-One

Cardinality: Many Staff members can be associated with One Store.

Constraints: Multiple staff members can work at the same store, but each staff member is associated with only one store.

Business Rule: Staff members work in specific stores within the organization.

1. Store and Address:

Relationship: One-to-One

Cardinality: One Store is associated with One Address.

Constraints: Each store has a single address, and each address is specific to one store.

Business Rule: Each store is uniquely identified by its address, ensuring a one-to-one relationship between a store and its location.

1. Order and Address:

Relationship: Many-to-One

Cardinality: Many Orders can be associated with One Address.

Constraints: Multiple orders can share the same address, but each address is specific to one or more orders.

Business Rule: Many orders can have the same address, reflecting scenarios where multiple orders are shipped to the same location or share a common billing address, or a customer re-orders from the store.

1. Staff and Address:

Relationship: One-to-One

Cardinality: One Staff has One Address.

Constraints: A staff can have only one address, and each address is associated with only one staff.

Business Rule: Each customer should have a valid address for contact and correspondence.

1. Order and Payment:

Relationship: One-to-One

Cardinality: One Order corresponds to One Payment.

Constraints: Each order has a corresponding payment, and each payment is related to a specific order.

Business Rule: Payments are directly tied to specific orders.

1. Payment and PaymentMethod:

Relationship: Many-to-One

Cardinality: Many Payments can be associated with One Payment Method.

Constraints: Multiple payments can use the same payment method, but each payment method is specific to one or more payments.

Business Rule: Multiple payments can share the same payment method, reflecting scenarios where customers use a common payment method for multiple transactions.

1. Stock and Product:

Relationship: One-to-One

Cardinality: One Stock entry corresponds to One Product.

Constraints: Each stock entry is associated with a specific product, and each product has a corresponding stock entry.

Business Rule: Stock information is specific to each product, ensuring a one-to-one relationship between the Stock and Product entities.

1. Product and Supplier:

Relationship: Many-to-One

Cardinality: Many Products can be supplied by One Supplier.

Constraints: Multiple products can be supplied by the same supplier, but each product is supplied by one supplier.

Business Rule: Each product is associated with a specific supplier.

1. Order and Delivery:

Relationship: One-to-One

Cardinality: One Order corresponds to One Delivery.

Constraints: Each order has a corresponding delivery, and each delivery is related to a specific order.

Business Rule: Each order should be associated with a delivery for fulfilment.

1. Stock and Store:

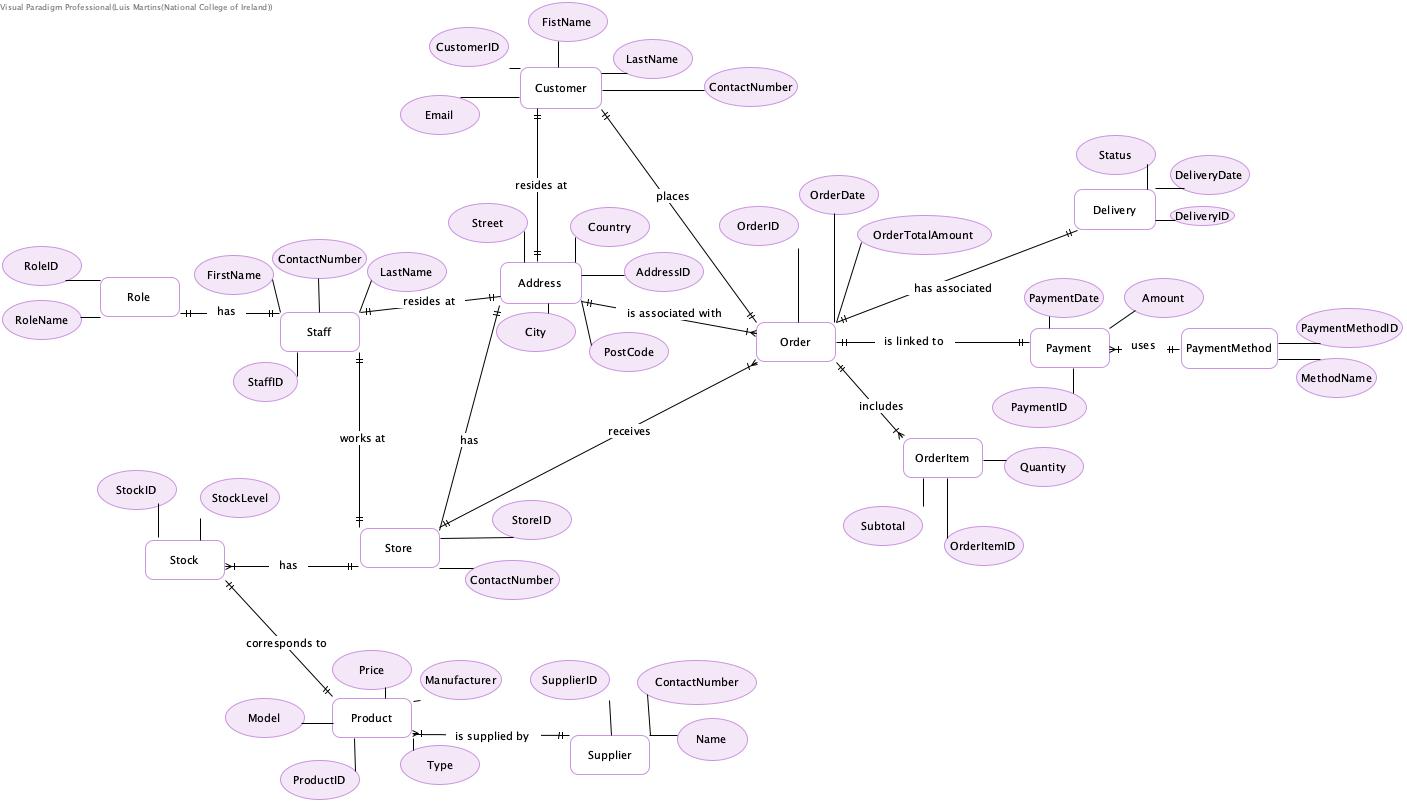
Relationship: Many-to-One

Cardinality: Many Stock entries can belong to One Store.

Constraints: Multiple stock entries can be associated with one store, but each stock entry is specific to one store.

Business Rule: Stock information is managed separately for each store, and multiple stock entries can be associated with a single store, representing the availability of different products in that store.

# Draw an ER diagram for the system depicting the entities, relationships, cardinalities, participations using your preferred ERD notation.



# Convert the entities in the conceptual design into logical dataset(relational model). Use a logical layout of entities which is easily understood by any reader. Ensure that tables are in the correct format to allow for the best possible representation of data.