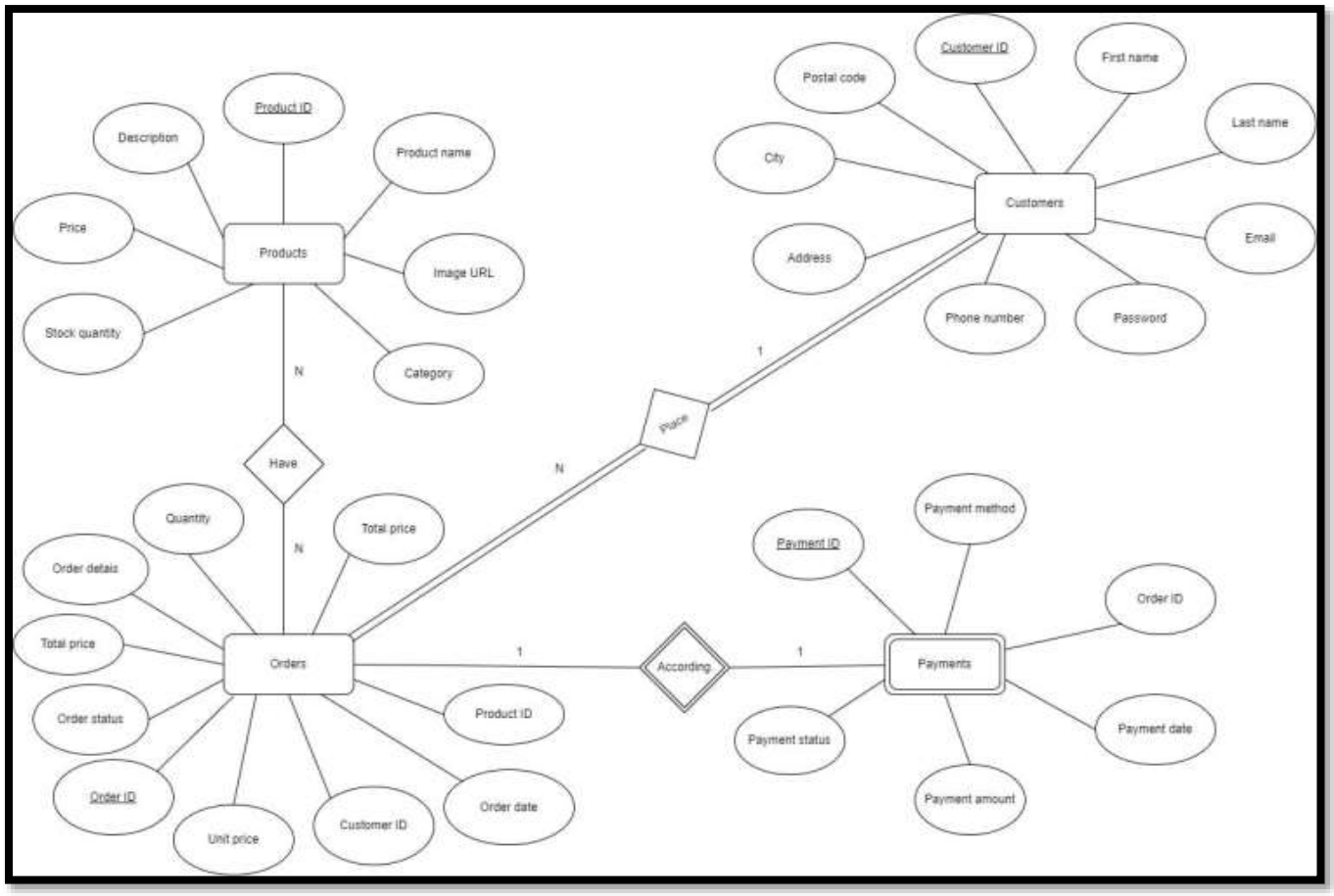
DEVELOPMENT OF THE ER DIAGRAM

GROUP MEMBERS:

E/19/166 – Jayathunga W.W.K.

E/19/324 – Rathnayake R.M.B.D.K.

ER Diagram


Product ID
Description
Product name
Price
Products
Image URL
Stock quantity
Category
N
Have
N
Quantity
Total price
Order details
Total price
Orders
Order status
Order ID
Unit price
Customer ID
Order date
Product ID
1
N
Place
1
Customer ID
Postal code
First name
Last name
City
Customers
Email
Address
Phone number
Password
1
According
1
Payment ID
Payment method
Order ID
Payments
Payment status
Payment date
Payment amount

Description of the Above ER Diagram

- The relationship between **Product** and **Order** is many-to-many, meaning that a product can be in many orders and an order can have many products. The relationship between **Customer** and **Order** is also one-to-many, meaning that a customer can have many orders, but an order can only have one customer. The relationship between **Order** and **Payment** is one-to-one, meaning that an order can only have one payment and a payment can only be for one order.

- The payment entity is a weak entity. The **Payment** entity is dependent on the **Order** entity, as it cannot exist without an associated order. A payment must be made for an order, and therefore the **Payment** entity's existence is dependent on the **Order** entity.

- The relationship between the **Payment** entity and the **Order** entity is an identifying relationship. The **Payment** entity's primary key includes the **order_id** attribute from the **Order** entity, and a payment cannot exist without an associated order.

- There is a total participation relationship between the **Order** entity and the **Customer** entity. In this case, each **Order** entity instance must be associated with a **Customer** entity instance, and every **Customer** entity instance must have at least one associated **Order** entity instance.

➢ **Product** entity and **Order** entity have a many-to-many relationship:

  - A **Product** entity instance can be associated with many **Order** entity instances.

- An **Order** entity instance can be associated with many **Product** entity instances.

- **Customer** entity and **Order** entity have a one-to-many relationship:
  - A **Customer** entity instance can be associated with many **Order** entity instances.
  - An **Order** entity instance can be associated with only one **Customer** entity instance.

- **Order** entity and **Payment** entity have a one-to-one relationship:
  - An **Order** entity instance can be associated with only one **Payment** entity instance.
  - A **Payment** entity instance can be associated with only one **Order** entity instance.

- There is no direct relationship between **Product** and **Customer** entities. That is because a **Customer** entity can place an **Order** for one or more **Product** entities. The **Order** entity holds information about the relationship between the **Customer** entity and the **Product** entity or entities they ordered. The relationship between these entities is mediated by the **Order** entity. Therefore, the **Order** entity serves as a bridge between the **Product** and **Customer** entities. It allows for a many-to-many relationship between **Product** and **Customer** through the one-to-many relationship between **Order** and each of the **Product** and **Customer** entities.

- There is an indirect relationship between **Payment** and **Customer** entities, which is mediated by the **Order** entity. That is the **Customer** entity can place an **Order** for one or more **Product** entities, and then make a **Payment** for that order. The **Payment** entity holds information about the payment made by the **Customer** entity. Therefore, the **Order** entity serves as a bridge between the **Payment** and **Customer** entities. It allows for a one-to-many relationship between **Payment** and **Customer** through the one-to-one relationship between **Payment** and **Order** entities.

Clear Picture of ER Diagram