Restaurant Database Implementation

1. Entities in Database

Entities Table:

- Table No: The number ID of the table in the restaurant (Primary Key)
- Table Status: The status of the table currently {Available or Occupied}

Entities Order:

- Order ID: The number ID of the order (Primary Key)
- Table No: The Table No of the table which the order come from *(Foreign Key)*
- Order Status: The status of the order {Pending or Confirmed}
- Total Cost: The total bill (in dollar) of the order

Entities Product:

- Product ID: The number ID of the product (Primary Key)
- Name: The name of the product
- Price: The price of the 1 product
- Category: The category which the product belongs to
- Image Link: The URL to the online image of the product

Entities Payment:

- Payment ID: The number ID of the payment (Primary Key)
- Order ID: The number ID of the order which the payment belongs to *(Foreign Key)*
- Total Cost: The total bill (in dollar) of the order
- Payment Type: The type of the payment { Online or Physical }
- Payment Status: The status of the payment currently { Pending or Confirmed }
- Payment Date: The date of the order and the payment occurred

2. Relationship between Entities

Relationship between Order and Table:

- An Order must belong to A Table
- A Table can have many Orders, or none

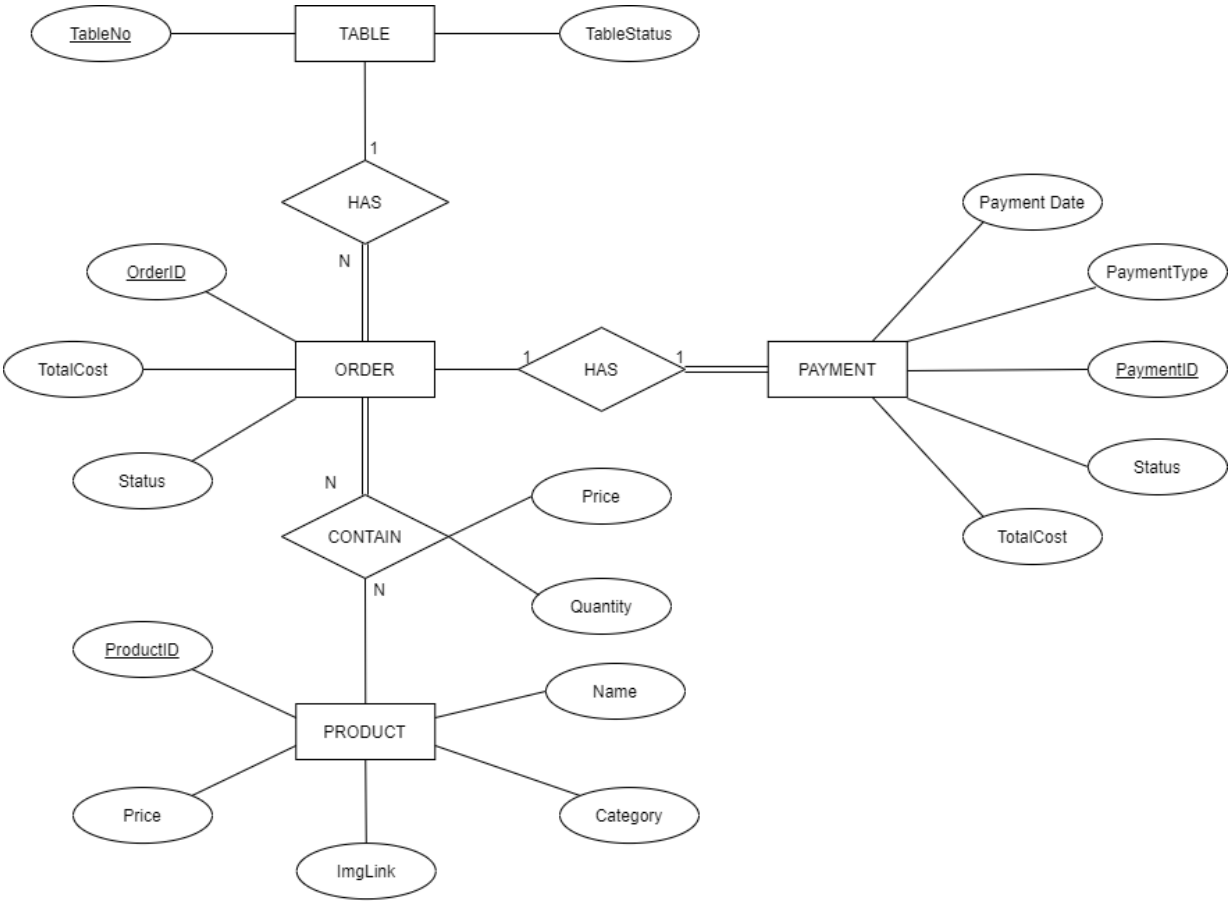
Relationship between Order and Product:

- An Order can have many Products, but at least one
- A Product can belong to many Orders, or none
- Attribute Quantity: The amount of that Product in the Order
- Attribute Price: The price of the 1 product

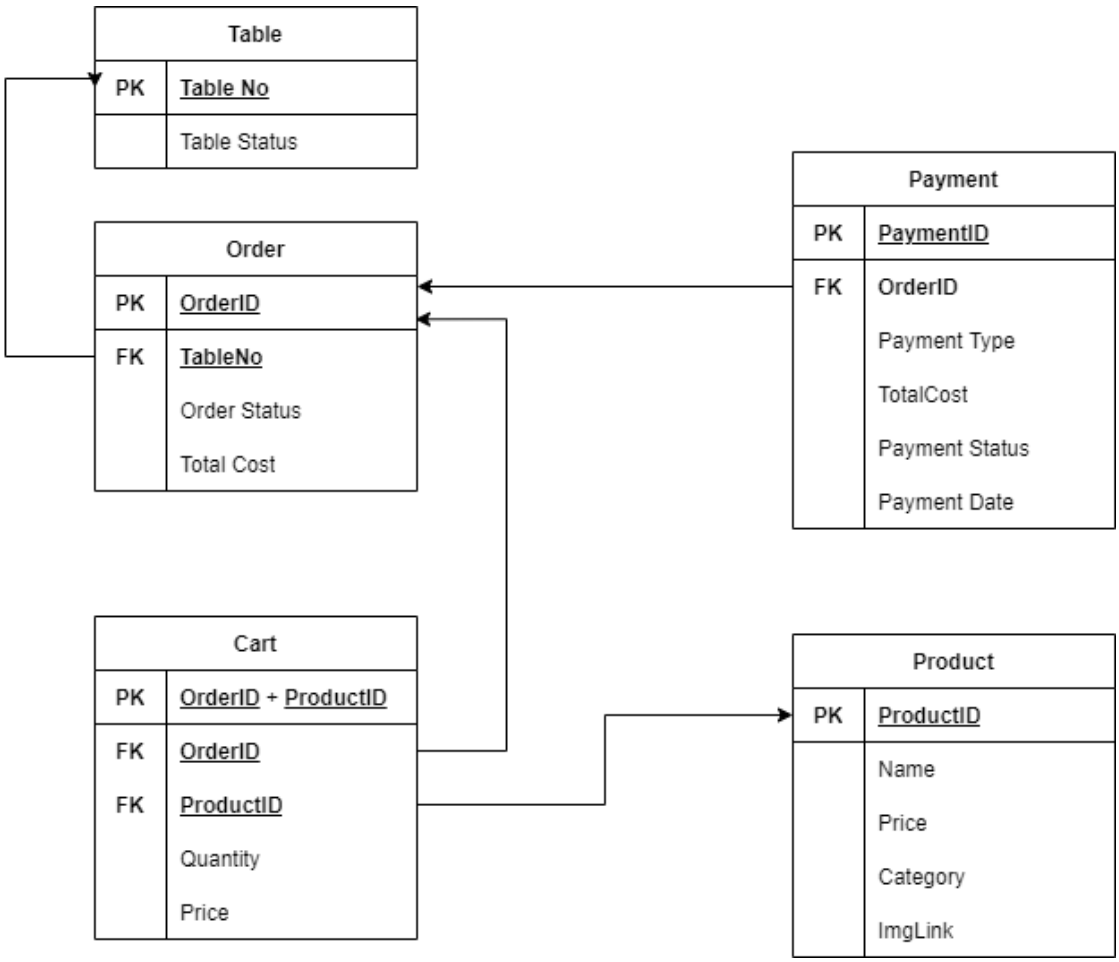
Relationship between Order and Payment:

- An Order can belong to A Payment
- A Payment must belong to An Order

3. Entities Relationship Diagram



4. Relation Diagram



Due to the relationship between Product and Order is a Many – Many Relationship, the relationship becomes a new table Cart

- Order ID: The number ID of the order *(Foreign Key)*
- Product ID: The number ID of the product *(Foreign Key)*
- Quantity: The amount of product with Product ID in the order with Order ID
- Order ID + Product ID (Primary Key)

5. SQL Implementation

```
CREATE TABLE RTable (
    TableNo             int             NOT NULL,
    TableStatus         int,
    PRIMARY KEY (TableNo)
);
```

Table [RTable] for Table

```
CREATE TABLE ROrder (
    OrderID         int             auto_increment,
    TableNo         int             NOT NULL,
    OrderStatus         int,
    TotalCost       float(10),
    PRIMARY KEY (OrderID)
);
```

Table [ROrder] for Order

* : Order Status is int type with 0 means Confirmed Status, 1 means Pending Status

```
CREATE TABLE RProduct (
    ProductID       int             auto_increment,
    Name            varchar(50)         default '',
    Price           float(10),
    Category        varchar(50),
    ImgLink         varchar(256),
    PRIMARY KEY (ProductID)
);
```

Table [RProduct] for Product

```
CREATE TABLE Cart (
    ProductID       int             NOT NULL,
    OrderID         int             NOT NULL,
    Price           float(10)       NOT NULL,
    Quantity        int             NOT NULL,
    check (Quantity > 0),
    constraint PPK PRIMARY KEY (ProductID, OrderID)
);
```

Table [Cart] for Cart

```
CREATE TABLE RPayment (
    PaymentID       int             auto_increment,
    OrderID         int             NOT NULL,
    TotalCost       float(10),
    PaymentStatus   int,
    PaymentType     int,
    PaymentDate     date,
    PRIMARY KEY (PaymentID)
);
```

Table [RPayment] for Payment

* : Payment Status is int type with 0 means Confirmed Status, 1 means Pending Status

* : Payment Type is int type with 0 means Physical Payment, 1 means Online Payment

```
alter table ROrder
    add FOREIGN KEY (TableNo)
                REFERENCES RTable(TableNo)
            ON DELETE NO ACTION
            ON UPDATE CASCADE
;

alter table Cart
    add FOREIGN KEY (ProductID)
                REFERENCES RProduct(ProductID)
            ON DELETE NO ACTION
            ON UPDATE CASCADE,
    add FOREIGN KEY (OrderID)
                REFERENCES ROrder(OrderID)
            ON DELETE NO ACTION
            ON UPDATE CASCADE
;

alter table RPayment
    add FOREIGN KEY (OrderID)
                REFERENCES ROrder(OrderID)
            ON DELETE NO ACTION
            ON UPDATE CASCADE
;
```

Alter to Tables to Create Foreign Key

6. Sample Data

In the Table [rtable], we insert 25 sample data, as 25 tables each has a table number from 1 to 25 with status as 0 (table is available)

In the Table [product], we insert sample data of menu products from Tim Hortons menu. All of these products on the menu are created and owned by Tim Hortons Inc, our group does not claim ownership over them nor use them for commercial purposes.