**Database Study for a Fast-Food restaurant**

**A cheeseburger and fries on a plate with a drink

AI-generated content may be incorrect.**

**Contents**

[1. Detailed Business Rules Derived from Company Operations 4](#_Toc194678480)

[2. Entity-Relationship (ER) Diagram 5](#_Toc194678481)

[2.1 ERD Details 5](#_Toc194678482)

[2.2 ENTITY RELATIONSHIP DIAGRAM 7](#_Toc194678483)

[3 . Logical Model 7](#_Toc194678484)

**Database Design for Yum-Yum Burgers**

## 1. Detailed Business Rules Derived from Company Operations

**Scope:**

A fast-food restaurant called Yum-Yum Burgers, requires a structured system to manage its daily operations efficiently. The database will store essential information about menu items, employees, suppliers, customer orders, and inventory levels. The objective is to ensure smooth order processing, track stock levels to prevent shortages, and manage employee work schedules effectively. The system will also track sales trends and customer preferences to optimize business operations. This is needed to ensure that Yum-Yum Burgers can provide fast, efficient and delicious food to its customers continually.

**Order Processing Rules:**

* A customer order must contain at least one menu item.
* Orders cannot be modified once they enter the preparation phase.
* Orders must be completed within the standard preparation time (e.g., 10 minutes for fast food).
* Different menu items have different preparation times.
* Payment must be completed before an order is handed to the customer.

**Cashier Rules:**

* Cashiers must log all sales in the database.
* Discounts and promotions must be applied according to company policy.

**Kitchen Rules:**

* Kitchen staff must always follow food safety guidelines.
* Inventory levels must be updated when ingredients are used.
* Only authorized kitchen staff can prepare food orders.

**Supplier Rules:**

* Suppliers must deliver fresh ingredients based on pre-set schedules.
* Low stock levels automatically trigger a reorder request.

**Delivery Staff Rules:**

* Delivery drivers must confirm order pickup and delivery in the system.
* Late deliveries beyond the expected time frame must be reported.
* Delivery staff must verify the customer’s details before handing over an order.

**Employee Management Rules:**

* Employees must clock in and out for each shift.
* Employees cannot work beyond the maximum allowed shift hours.

## 2. Entity-Relationship (ER) Diagram

(Include an ER diagram visual representation here)

Entities and Relationships:

### 2.1 ERD Details

**Weak Entities**

* Inventory can be considered weak since it relies on Supplier\_ID for some of its data.
* Delivery could also be seen as weak since it requires Order\_ID and Driver\_ID to exist.

**Bridge Entities**

* Delivery acts as a bridge between Order and Employee (Driver).
* OrderMenuDetail acts as a bridge between Order and Menu\_Item.

**Strong Relationships**

* The Order and Customer relationship is strong since an order cannot exist without a customer.
* Employee and Delivery have a strong relationship because a driver must be assigned to a delivery.
* The Order and Employee relationship is strong since an order cannot exist without a KitchenStaff.
* The Payment and Employee relationship is strong since an order cannot exist without a Cashier.

**Weak Relationships**

* The Inventory and Supplier relationship is weak because Inventory depends on Supplier\_ID but does not uniquely define the supplier.

**Mandatory Relationships**

* Order to Customer: An order must belong to a customer.
* Payment to Order: A payment must be linked to an order.

**Super Entities**

* Employee
* Payment

**Subtype Entities**

* Cashier
* Driver
* Chef
* Card
* Cash

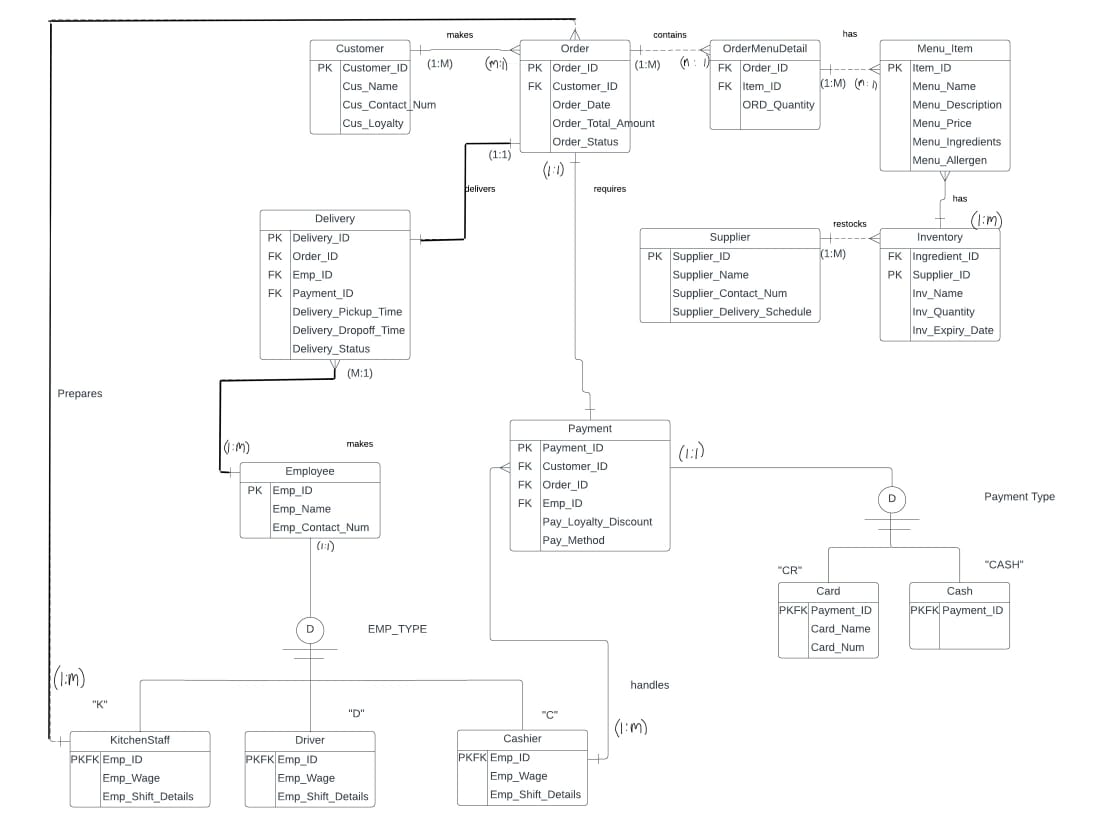
**Derived Attribute**

* The total amount of an order is not stored directly but calculated from the sum of menu item prices and their quantities in Order\_Menu.

**Composite Key**

* The primary key consists of two attributes (Order\_ID, Item\_ID), ensuring uniqueness for each menu item within a specific order.

### 2.2 ENTITY RELATIONSHIP DIAGRAM



## 3 . Logical Model

Customer (Customer\_ID, Cus\_Name, Cus\_Contact\_Num, Cus\_Loyalty)

Order (Order\_ID, *Customer\_ID*, Order\_Date, Order\_Status, Order\_Total\_Amount)

Menu Item (Item\_ID, Menu\_Name, Menu\_Description, Menu\_Price, Menu\_Ingredients, Menu\_Allergen)

Order\_Menu (Order\_ID, *Item\_ID*, ORD\_Quantity)

Inventory (Ingredient\_ID, *Supplier\_ID*, Inv\_Name, Inv\_Quantity, Inv\_Expiry\_Date)

Supplier (Supplier\_ID, Supplier\_Name, Supplier\_Contac\_Num, Supplier\_Delivery\_Schedule)

Employee (Emp\_ID, Emp\_Name, Emp\_Role, Emp\_Contact)

Delivery (Delivery\_ID, *Order\_ID*, *Emp\_ID*, Delivery\_Pickup\_Time, Delivery\_Dropoff\_Time, Delivery\_Status)

Payment (Payment\_ID, *Order\_ID*, Pay\_Method, Pay\_Loyalty\_Discount)

Cash(*Payment\_ID*, Card\_Name, Card\_Num)

Card(*Payment\_ID*)

Driver (*Emp\_ID*, Emp\_Wage, Emp\_Shift\_Details)

KitchenStaff (*Emp\_ID*, Emp\_Wage, Emp\_Shift\_Details)

Cashier (*Emp\_ID*, Emp\_Wage, Emp\_Shift\_Details)