

# Food Delivery Management System

Version 1.3

**Business Requirements Document (BRD)**

**Version and Approvals**

|  |  |
| --- | --- |
| **Version History** | |
| **Version#** | **Date** | **Revised By** | **Updates** |
| 1.0 | 03/19/21 | Hitesh Chawla | Created first draft with project description |
| 1.1 | 03/19/21 | Hitesh Chawla | Added ER Diagram |
| 1.2 | 04/11/21 | Hitesh Chawla | Added logical diagram and DDL |
| 1.3 | 04/26/21 | Hitesh Chawla | Added insert and select statements |

**Table of Contents**

Table of Contents

[Food Delivery Management System 1](#_Toc70366869)

[Project Overview 5](#_Toc70366870)

[Project Overview and Background 5](#_Toc70366871)

[Business Rule 5](#_Toc70366872)

[ER Diagram 7](#_Toc70366873)

[Entity Relationship Diagram Explanation 8](#_Toc70366874)

[Logical Database Design 9](#_Toc70366875)

[Physical Database Design 9](#_Toc70366876)

[Insert Statement 9](#_Toc70366877)

**Project Details**

|  |  |
| --- | --- |
| **Project Name** | Food Delivery Management System |
| **Project Type** | Relational Database |
| **Project Start Date** | 02/01/2021 |
| **Project End Date** | 05/15/2021 |
| **Primary Driver** | Hitesh Chawla |

# Project Overview

### Project Overview and Background

A Food Delivery Management System (FDMS) involves various operations at different stages, right from when a customer orders food from a food outlet until the customer's feedback is received. We plan to design a database that would be a platform to order, deliver or pick up food from various restaurants to your doorstep.

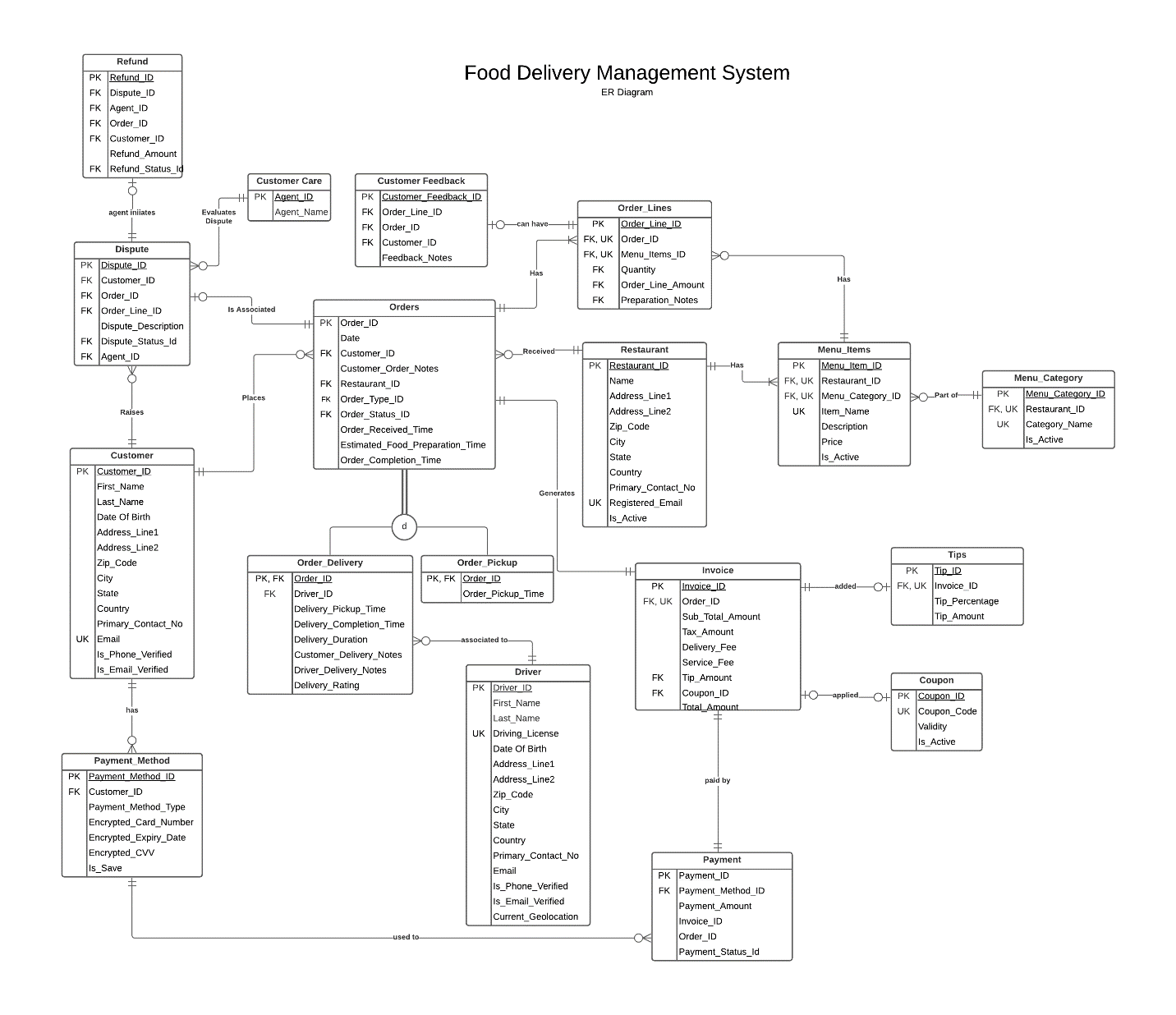
FDMS stores information related, but not limited to, customers, restaurants, available food items, orders, payment history, feedback, deliveries, and dispute details. The database will give us insights through reports to have information about sales at a given point of time, onboarded restaurants, delivery/pickup schedules, dispute handling, and provide updates on refunds. 

# Business Rules

Below are the business rules for Food Delivery Management System –

* A given customer can place zero or many orders.
* A given order can be placed by exactly one customer.
* A given order can be either of type “Delivery” or “Pick up” but not both.
* A given order can have one or many order lines.
* A given order line can be of exactly one order.
* A given order can be received by exactly one restaurant.
* A given restaurant can receive zero or many orders.
* A given menu item can be a part of zero or many order lines.
* A given order line can be made from exactly one menu item.
* A given restaurant can have one or many menu items.
* A menu item comes from exactly one restaurant.
* A given menu item is part of exactly one category
* A given category can have zero or many menu items
* A given order can generate exactly one invoice.
* A given invoice can be generated against exactly one order.
* A given invoice can have exactly one payment.
* A given payment can be made against exactly one invoice.
* A given delivery can be associated with exactly one driver.
* A given driver can have zero or many deliveries.
* A given coupon can be applied against zero or one invoice.
* A given invoice can have zero or one coupons applied against it.
* A given invoice may have a tip.
* A given tip can be associated with exactly one invoice
* A customer can have zero or many methods of payment
* A given payment method is associated to exactly one customer
* A given payment method may or may not be used for a payment
* A given payment is using exactly one payment method
* A customer can raise zero or many disputes
* A given dispute must be raised by exactly one customer
* A given dispute is associated to exactly one order
* A given order may or may not have a dispute
* A given dispute is evaluated by exactly one customer care representative.
* A given customer care representative may evaluate many disputes.
* A given customer care representative may or may not initiate refund for a given dispute
* A given refund will be initiated by exactly one customer
* A given food item/order line may have one feedback.
* A given feedback can be given for exactly one item/order line.

# ER Diagram



# Entity Relationship Diagram Explanation

Below are the different entries used to build Food Delivery Management Systems and their brief business description and value –

1. **Customer** – The customer entity holds the basis details related to customers such as address, email, phone number, etc. The Customer\_Id acts as primary key while email has a unique constraint on it
2. **Restaurant** – The restaurant entity holds the basis details related to restaurant such as address, email, phone number, name of the restaurant, etc. The Restaurant\_Id acts was primary key
3. **Customer Care** – The entity holds the employee id/agent it and name of the customer care representative
4. **Driver** – The entity holds the details of drivers such as name, driving license, address, etc. The Driver\_Id is the primary key while driver license has a unique constraint on it. It also holds the current geo location of driver to track the delivery
5. **Coupon** – The entity holds coupons details such as code, percentage of discount. Coupon\_Id is the primary key while coupon code has a unique constraint on it
6. **Payment\_Method –** The entity holds the payment method of customer such as credit card number, expiry date, if the payment method needs to be saved.
7. **Payment –** The entity holds the values for payment status, payment method used by the customer against an order. The Payment\_Id acts as primary key
8. **Menu\_Category –** Menu\_Category holds different menus each restaurant caters to such as desserts, drinks, pasta, burger, South Indian, Mediterranean, etc. The combination of Restarant\_Id and Menu\_Name have a unique constraint on it
9. **Menu\_Items –** Menu\_Items holds menu items/food served by each restaurant alongside a brief description of food items, category, and price. The combination of Restarant\_Id, Menu\_Category\_Id, and Item\_Name have a unique constraint on it
10. **Orders –** The order table will contain details about orders placed by customers from a specific restaurant along with food preparation time and order status
11. **Order\_Delivery –** TheOrder\_Delivery is the sub type of order and keeps the information related to all the deliveries. Order\_Id from order table serves as the primary key.
12. **Order\_Pickup –** TheOrder\_Pickup is the sub type of order and keeps the information related to all the pick-ups. Order\_Id from order table serves as the primary key.
13. **Order\_Lines –** Order\_Lines hold the individual order items
14. **Invoice** – Invoice holds the billing details like total amount, taxes and service charges of an order
15. **Tips**—Tip table holds the optional tip that can be given to driver against an order
16. **Customer\_Feedback**— Customer\_Feedback holds the feedback of customers against every item he/she has ordered
17. **Dispute**—The entity will keep record of disputes created by customers against any order or order item
18. **Refund**—Refund entity holds the details of any refunds approved by the customer care representative against an order

# Logical Database Design

The attached spreadsheet lists the logical database design of entities, their attributes, keys, and constraints. The tables in logical database design are in 3NF form.



# Physical Database Design

The attached SQL scripts contains the query to create the physical database structure and constraints associated to the tables for FDMS.



# Insert Statement

The attached SQL script contains insert statement and few select statement for initial data analytics for FDMS.

