**Title: Food Delivery Service And Database Management System**

**---------------------------------------------------------------------------------------------------------------------------**

**Abstract**

**Description**

In today’s world, there is an increasing amount of innovation that centers on delivering goods to a person’s doorstep, whether that may be groceries, clothes, furniture, etc.

Hence, we decided to build an application for a food delivery service company to help them

analyze data about their business transactions. This app provides a UI where the company can view information about customers (including their names, phone numbers, and addresses),

employees (their names and job positions), and restaurants participating in the service (their addresses and menu items). As such, the app’s intended clients are the managers of the

company, since they will likely be responsible for keeping track of current employees,

customers, and restaurant data. For example, they can query each customer’s order history and the number of orders fulfilled within a given time period. These queries will assist managers in making decisions about employee bonuses, customer discounts, etc.

For overall functionality of the application, we would have to be able to query how many hours each employee has worked, be able to create a contract with a new restaurant to join the delivery program, and be able to view the order histories of individual customers

(more functionalities are listed below). We would also take into consideration limitations such as how many hours an employee can work per day and that drivers in a particular region can only deliver food from restaurants from that region to customers in the same region

(constraints listed below).

Functionalities

1. **Transactions:**

**Employee transactions:**

* 1. Generate a list of all the employees.
  2. Show all the employees who work in a given region
  3. Generate a list of supervisors and the drivers they supervise.
  4. Create a new employee contract (input employee name, phone number, region, etc).
  5. Ratings For Driver.

**Restaurant transactions:**

1. Generate a list of all restaurants that have partnered with the delivery service.
2. Generate a list of restaurants located in a given region.
3. Generate a list of menu items and their prices from a given restaurant.
4. Ratings For Restaurants are shown in the place order screen.

**Customer transactions:**

1. Create customer account
2. Customer Sign In
3. Show all the restaurants in the same region as the customer.
4. Show the order history of a given customer.
5. **Constraints:**
   1. There is only one supervisor per region.

Enforced with an insert trigger on the Supervisors table. (check\_spvsr)

* 1. A customer can only order food from restaurants in his/her region.

Enforced with an insert trigger on the Orders table. (fromSameRegion) And also within the application.

* 1. Each driver can only complete orders that are in the same region as the restaurant and the customer, and himself/herself.

Enforced with an insert trigger on the Orders table. (fromSameRegion)

* 1. A driver can only be employed in one region

Enforced through FD empID->empRegion. User of application is also only allowed to select one option from a given drop-down list of regions.

* 1. A driver cannot work for more than 8 hours per day.

Enforced with a check() constraint during relation creation.

**----------------------------------------------------------------------------------------------------------------**

**FDs and Normalization**

**E mployees, Supervisors, Drivers Tables**

**empId** -> empName, empRegion, empPhone, empHours\_worked, salary, licenseNo, wage, supervisorId, dRating

**C ustomers Table**

**cId** -> cName, cRegion, cPhone, cAddress

**O rders Table**

**orderId** -> total, cId, time\_made, rId, empId, time\_delivered

**R estaurants Table**

**rId** -> rName, rCuisine, rRegion, rPhone, rAddress, rRating

**M enuItems table**

**rId, itemName** -> itemPrice

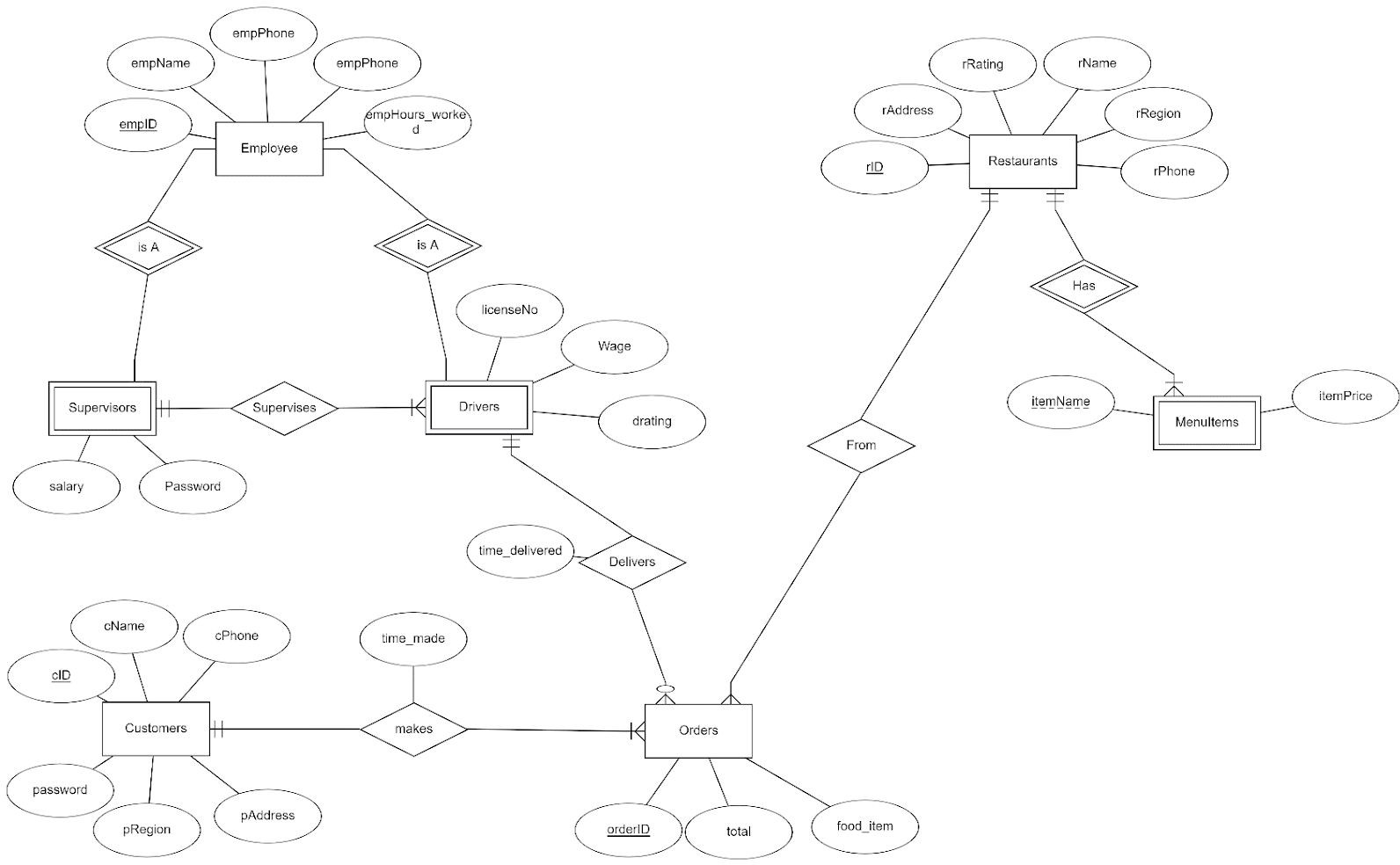
**Normalization:**

All determinants are **super keys to their respective tables**, therefore all relations are in

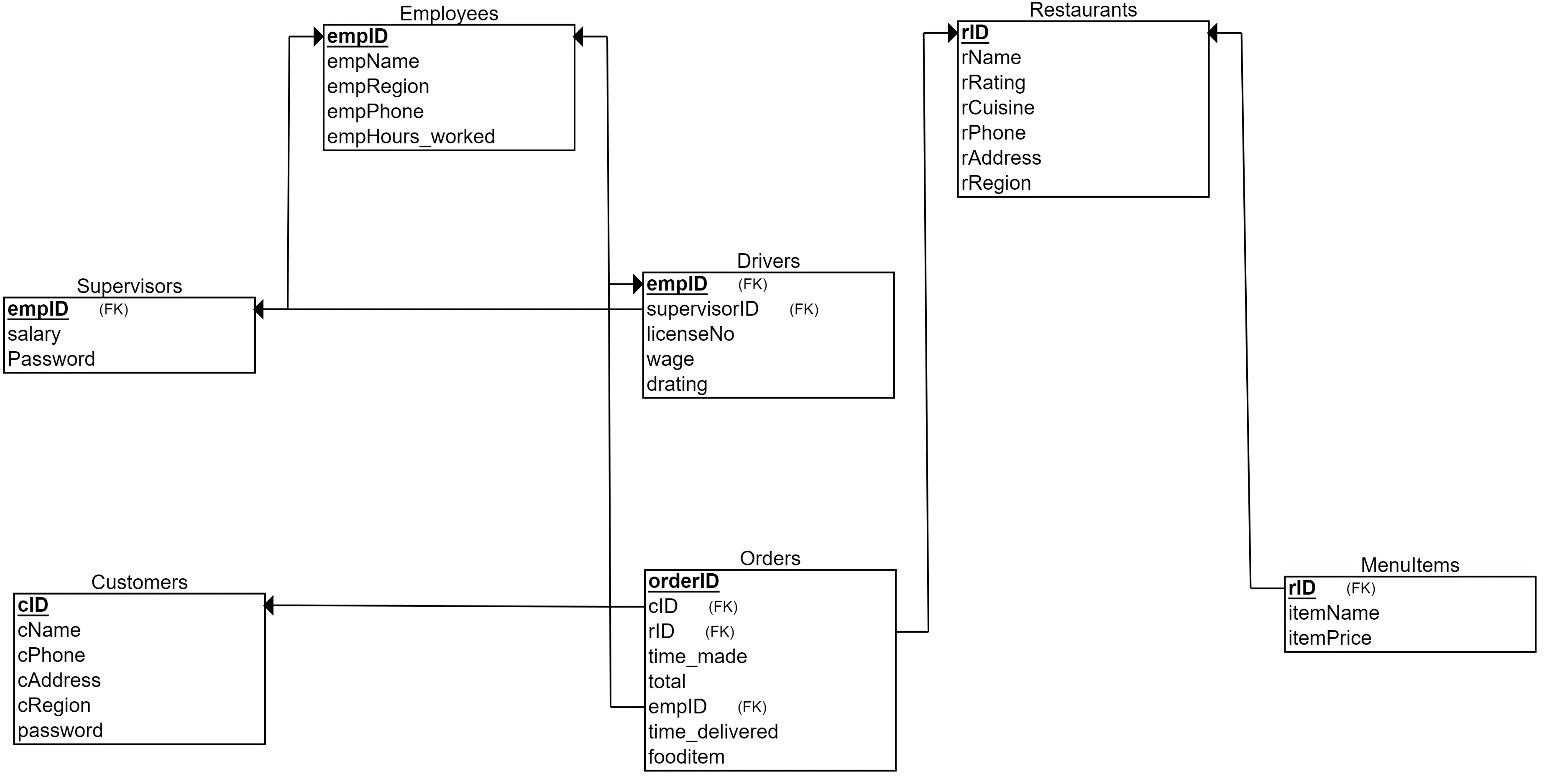
**BCNF** form.

**----------------------------------------------------------------------------------------------------------------**

**Er Diagram**



**ER to Relational Schema**



**----------------------------------------------------------------------------------------------------------------**