Restaurant Delivery System Database Project: Requirements Analysis and Overview

**Zyad Fri Oumaima Kouhail Jaafar Yeffou  *Deliverable1***

**Overview :**

The Restaurant Delivery System Database Management Project is designed to streamline the process of ordering food from a variety of restaurants. It revolves around a well-structured database system that encompasses key entities like customers, delivery personnel, orders, restaurants, vehicles, meals, deliveries, and various payment methods. Customers can select their desired meals from restaurants within their city and place orders. These orders are then assigned to dedicated delivery personnel who use specific vehicles for timely delivery. The project ensures data accuracy by allowing the addition of new restaurant information as needed. Additionally, it manages customer accounts, offers different payment options (PaymentCash or PaymentOnline), and enables secure online payments through credit cards.

**Entities:**

* **Customer** The "Customer" entity represents individuals who use the restaurant delivery systeme . **Her’is a brief description and the data types for its attributes:**
* **Cid**: This is a unique identifier for each customer and it is an integer.
* **Cust\_name**: This attribute represents the name of the customer. It is of data type varchar.
* **Phone Number:** This attribute represents the customer's telephone number. It is of data type varchar.
* **adresse:** This attribute represents the customer's address. It is also of data type varchar
* **password:** This attribute stores the customer's password for authentication and security purposes. It is typically of data type varchar.
* **Email:** This attribute holds the customer's email address, which is used for communication and account-related activities. It is of data type varchar
* **deliveryPersonel** entity is designed to represent delivery personnel involved in the restaurant delivery system. **Her’is a brief description and the data types for its attributes:**
* **Pid**: This attribute is a unique identifier for each delivery personnel and is typically an integer.
* **Name**: This attribute represents the name of the delivery personnel and is of data type varchar.
* **DPCity**: This attribute indicates the city in which the delivery personnel is based or operates. It is also of data type varchar.
* **DP Phone Number:** This attribute stores the telephone number of the delivery personnel and is usually of data type varchar
* **Vehicle** entity represents the vehicles used in the restaurant delivery system**. Her’is a brief description and the data types for its attributes:**
* **Vid**: This attribute is a unique identifier for each vehicle and is typically of data type integer
* **max\_meals:** This attribute represents the maximum number of meals that the vehicle can carry at a time. It is of data type integer.
* **Speed:** This attribute indicates the speed of the vehicle, typically measured in units as kilometers per hour (km/h. The data type for this attribute is Float
* **Delivery** entity is used to represent a collection of orders made by a customer as part of a single delivery. **Her’is a brief description and the data types for its attributes:**
* **Delivery\_ID** :This attribute is a unique identifier for each delivery, and it serves as the primary key for the "Delivery" entity. It is typically of data type integer
* **Num\_of\_meals**:This attribute represents the number of orders that a delivery contains in once. It is of data type integer
* **VEHICULE\_ID**: This attribute is a foreign key that links the delivery to the specific vehicle used for the delivery. It references the "VEHICULE\_ID" attribute from the "Vehicule" entity.
* **Order** entity represents individual orders within the restaurant delivery system. **Here's an explanation of the attributes for this entity and their respective data types:**
* **Oid:** This attribute is a unique identifier for each order and is typically of data type integer or.
* **Customer\_id:** This attribute serves as a reference to the customer who placed the order. It is of data type integer and is used as a foreign key linking to the "Customer" entity.
* **Number of meals:** it indicates the number of meals that a single customer asked for in a specific order in a single delivery: It is of data type integer
* **Meals** entity represents different food items or dishes available in the restaurant delivery system. **Here's the attributes for this entity and their respective data types:**
* **menu\_id (Foreign Key):** This attribute is a foreign key that references the menu to which the meal belongs. It is of data type integer and establishes a link to the "menu\_id" in the "Restaurant" entity.

**Mname:** This attribute represents the name or title of the meal, and it is typically of data type varchar.

* **price:** This attribute holds the price of the meal and it is typically of a data type Float.
* **Restaurant** entity represents individual restaurants in the restaurant delivery system. **Here's an explanation of the attributes for this entity and their respective data types:**
* **RName:** This attribute represents the name of the restaurant and is typically of data type varchar.
* **Rid (Primary Key):** This attribute serves as the primary key for the "Restaurant" entity and is a unique identifier for each restaurant. It is typically of data type integer.
* **RCity:** This attribute indicates the city where the restaurant is located and is of data type varchar
* **payment** entity represents the payments made within the restaurant delivery system. **Here's an explanation of the attributes for this entity and their respective data types:**
* **pyID:** This attribute serves as a unique identifier for each payment and is typically of data type integer.
* **PDate:** This attribute records the date when the payment was made and is typically of data type date.
* **Payment-cash** entity represents cash payments made within the restaurant delivery system. **Here's an explanation of the attributes for this entity and their respective data types:**
* **PCPrice:** This attribute serves as a unique identifier for each cash payment and is typically of data type integer.
* **budget:** This attribute indicates the amount of money paid in cash for a specific transaction or order. It is typically of a numeric data type,Float.
* **Payment-online** entity represents online payments made within the restaurant delivery system**. Here's an explanation of the attributes for this entity and their respective data types:**
* **PoID:** This attribute serves as a unique identifier for each online payment and is typically of data type integer.
* **POPrice:** This attribute indicates the total price or amount paid online for a specific transaction or delevery. It is typically a Float.

**Relation ships:**

* Each DeliveryPersonnel is assigned to exactly one Vehicle, and each Vehicle is exclusively used by a single DeliveryPersonnel (one-to-one relationship).
* Each Vehicle is utilized to deliver at least one delivery, and each delivery is delivered by exactly one vehicle.
* Each order is assigned to at least one delivery, and each delivery consists of at least one order.
* Each order includes at least one meal, and each meal can be associated with multiple orders.
* Each meal is offered by one restaurant, and each restaurant can offer multiple meals.
* Each order is placed by a single customer (one-to-one relationship), while a customer can place orders for several meals.
* Orders are paid for using either PaymentCash or PaymentOnline methods. If PaymentOnline is chosen, the system does not store credit card information to maintain security, ensuring the confidentiality of sensitive data. If PaymentCash is selected, the DeliveryPersonnel provides change to the customer when necessary.

**Constraints:**

* **Domain Constraints.**
* **Domain constraints** specify the valid values and ranges that attributes in our database must adhere to. Here are our domain constraints that we apply to the attributes in our restaurant delivery system database:
* **Customer\_id:** This attribute should be a positive integer or a non-negative numerical value, ensuring that customer IDs are unique and non-negative (actualy for all attributes that represent an ID).
* **Phone:** Telephone numbers should conform to a specific format or pattern, such as (XXXXXXXXXX)or (XXX-XXX-XXXX), depending on our region, and should not contain non-numeric characters.
* **Meal\_price:** Meal prices should be positive numerical values, typically represented as decimal or float data types.
* **Date\_of\_payment:** This attribute should be of date or timestamp data type such as(DD-MM-YYYY), ensuring that payment dates are in a valid date and time format.
* **Budget:** The budget for cash payments should be a positive numerical value, typically represented as float data types.
* **Key Constaints**
* **Key constraints** define the uniqueness of values in our database and help ensure and maintain data integrity. Also note that key attributes (the following below) don’t have Null values.
* **DeliveryPersonnel:** The primary key " Pid" uniquely identifies each delivery personnel, ensuring that no two delivery personnel have the same identifier.
* **Vehicle:** The primary key "vehicule\_id" uniquely identifies each vehicle in our system, guaranteeing that no two vehicles share the same identifier.
* **Delivery:** The primary key "Did" uniquely identifies each delivery, ensuring that each delivery has a unique identifier.
* **Restaurant:** The primary key "Rid" uniquely identifies each restaurant, ensuring that no two restaurants have the same identifier.
* **Meal:** The combination of "name" and "Rid" (Restaurant name and identifier) forms a composite Condidate key, uniquely identifying each meal. This ensures that no two meals share the same combination of name and restaurant identifier.
* **Order:** The primary key "Order\_id" uniquely identifies each order, ensuring that no two orders share the same identifier.
* **Customer:** The primary key " Cid" uniquely identifies each customer, guaranteeing that no two customers have the same identifier.
* **Payment-cash:** The primary key "PCid" uniquely identifies each cash payment transaction, ensuring that no two cash payment transactions share the same identifier.
* **Payment-online:** The primary key "POID" uniquely identifies each online payment transaction, guaranteeing that no two online payment transactions share the same identifier
* **Referential integrity Constraints:**
* Referential integrity constraints ensure that foreign key relationships in the database maintain their integrity, meaning that the foreign key (child) values must refer to a valid primary key (parent) value:
* **vehicule\_id:** This foreign key is used to associate deliveries with specific vehicles. Referential integrity constraints should be applied to ensure that a vehicle (parent record) cannot be deleted if there are any deliveries (child records) associated with it.
* **Restaurant\_Id:** This foreign key links meals to the corresponding restaurants. Referential integrity constraints should ensure that a restaurant (parent record) cannot be deleted if there are any meals (child records) associated with it.
* **Customer\_id**: This foreign key associates orders with specific customers. Referential integrity constraints should guarantee that a customer (parent record) cannot be deleted if there are any orders (child records) associated with them
* **Participation Constraints:**
* **Participation constraints** specify the minimum and maximum number of times an entity can participate in a relationship.
* **DeliveryPersonnel-Vehicle Relationship:**
* **Minimum Participation**: Each delivery personnel must be assigned to exactly one vehicle. Therefore, the minimum participation for the delivery personnel entity in this relationship should be "mandatory," meaning every delivery personnel must be associated with a vehicle.
* **Maximum Participation**: Each vehicle is used by only one delivery personnel. Therefore, the maximum participation for the delivery personnel entity in this relationship should be "one."
* **Order-Delivery Relationship:**
* **Minimum Participation**: Each order is assigned to at least one delivery. Therefore, the minimum participation for the order entity in this relationship should be "mandatory," meaning every order must be associated with a delivery.
* **Maximum Participation:** Each delivery is delivered exactly by one vehicle. Therefore, the maximum participation for the order entity in this relationship should be "many."
* **Customer-Order Relationship:**
* **Minimum Participation:** Each order is placed by at least one customer. Therefore, the minimum participation for the customer entity in this relationship should be "mandatory," meaning every order must be associated with a customer.
* **Maximum Participation:** Each customer can place multiple orders. Therefore, the maximum participation for the customer entity in this relationship should be "many."

**Written constraints that cannot be fully expressed in an Entity-Relationship Diagram (ERD)**

* **Region Matching Constraint for Deliveries:**
* A delivery should not be canceled unless the customer is in the same region as the restaurant they ordered from.
* **Budget Constraint for Deliveries:**
* The budget for a delivery should always be greater than or equal to the total price, meaning the customer must pay the full price in a single transaction.
* **Payment Constraints For each Delivery:**
* For a single delivery, a customer is allowed to associate only one credit card with their payment method. Additionally, the payment method for a delivery must be exclusive, meaning it can only be either online or cash, but not both. In other words, a customer cannot use multiple credit cards for a single delivery, and the payment must be made using only one method, either online or in cash.