Database for an e-food Delivery System

Panourgias Christos Sklavos Ioannis

Contents

1 Introduction

An e-food delivery system is a convenient and efficient way for customers to order food from their favorite restaurants or other food vendors. The system allows customers to browse menus, place orders, and track the status of their orders in real-time. It also provides drivers with the information they need to deliver the orders to the customers.

The e-food delivery system database is a key component of the system, as it stores all of the information related to the orders, customers, restaurants, and drivers. It is constantly updated in real-time to ensure that the information it contains is accurate and up-to-date.

Overall, the e-food delivery system database is an essential tool for managing and tracking orders, and it is an integral part of the e-food delivery system. It is a valuable resource for both customers and e-food vendors, and it helps to ensure that orders are delivered efficiently and accurately.

2 Sample Queries

1. Names of all customers who have placed an order with a restaurant owned by a specific person in the past week and used a coupon.
2. What is the total number of each menu item that has been ordered from restaurants owned by a specific person, along with the average rating for each item.
3. Obtain all menu items with "chicken" in their name and an average rating of at least 4.0.
4. Names of all drivers who are available to take on new orders, including their average rating.
5. What is the total number of orders placed by each customer at restaurants owned by a specific person, along with the average rating received by each customer.
6. Names of all customers who have placed an order within the past week and used a coupon.
7. Show all orders currently in transit, along with the driver's rating for each order.
8. What is the payment method and status for a specific order, as well as the applied coupon code (if any).
9. What is the delivery address for a specific order, along with the ratings for the associated restaurant and driver.
10. What is the total number of orders placed by each restaurant, along with the average rating for each restaurant.

3 Data Requirements

3.1 Entities

- **CUSTOMER:** This entity represents the customers of the e-food delivery service.
- **RESTAURANT:** This entity represents the restaurant from which the customers might choose to buy food from.
- **MENU_ITEM:** This entity represents the items that a restaurant can have in it's menu.
- **ORDERS:** This entity holds information on the orders placed by the customers on a restaurant.
- **DRIVER:** This entity represents the drivers that deliver the food after an order has been placed on a restaurant by a customer.
- **PAYMENT:** This entity holds information on the payments that are made by the customer for an order that the customer has placed on a restaurant.
- **COUPON:** This entity represents the coupons that are used for a discount on the orders placed by the customers.
- **RATING:** This entity holds information on the ratings and reviews made by the customers on the restaurants, the drivers, or the menu items.
- **OWNER:** This entity represents the owner of a restaurant or restaurants.

3.2 Attributes and Constraints

Note: The constraints that are listed for every attribute below, cannot be modeled by the ER diagram.

- **CUSTOMER**
  - **customer_username VARCHAR(100):** The username of the customer.

Constraints
* Must be unique
* Must contain only letters and numbers
* Must not be null

– **fname VARCHAR(50):** The first name of the customer.

Constraints
* Must not be null * Must contain only letter characters

– **lname VARCHAR(50):** The last name of the customer.

Constraints
* Must not be null * Must contain only letter characters

– **email VARCHAR(100):** The email of the customer.

Constraints
* Must not be null * Must be unique * Must have a "@" followed by a domain

– **phone VARCHAR(50):** The phone number of the customer.

Constraints
* Must not be null * Must be unique * Must consist of only numbers * Must have a fixed size (specified by the country that the phone of the customer is registered to)

– **address VARCHAR(255):** The physical address of the customer.

Constraints
* Must not be null * Must consist of only numbers and letters * Must contain the name of the street and the number of the building

- **RESTAURANT**
  – **rest_vat VARCHAR(100):** The VAT number of the restaurant.

Constraints
* Must not be null * Must be unique * Must consist of only numbers * Must have a fixed size (depending on the country that the restaurant is registered to)

  – **name VARCHAR(100):** The name of the restaurant

Constraints
* Must not be null * Must be unique

- **location VARCHAR(255):** The physical address of the restaurant

  Constraints

  * Must not be null
  * Must have the name of the street and the number of the building
  * Must consist of only numbers and letters

- **cuisine_type VARCHAR(100):** The type of cuisine that the restaurant has.

  Constraints

  * Must not be null
  * Must consist of only letters

- **MENU_ITEM**
  - **item_barcode VARCHAR(100):** The barcode of the item.

    Constraints

    * Must not be null
    * Must consist of only numbers
    * Must be unique

– **rest_vat VARCHAR(100):** The VAT of the restaurant that owns the item.

Constraints
* Must not be null * Must be unique * Must consist of only numbers * Must have a fixed size (depending on the country that the restaurant is registered to)

– **name VARCHAR(100):** The name of the item.

Constraints
* Must not be null

– **price DECIMAL(4,2):** The price of the item.

Constraints
* Must not be null * Must be a positive float number

– **availability INT:** The number of instances of the item that exist in stock.

Constraints
* Must be positive integer **NOTE:** if the value is not added then the default value will be 0.

- **ORDERS**
  - **receipt_num VARCHAR(100):** The receipt number of the order.

    Constraints

    * must not be null
    * Must be unique
    * Must consist of only numbers

  - **customer_username VARCHAR(100):** The username of the customer that placed the order.

    Constraints

    * Must consist of only letters and numbers

  - **rest_vat VARCHAR(100):** The VAT of the restaurant that the order was placed on.

    Constraints

    * Must not be null
    * Must have a fixed size (depends on the country that the restaurant is registered to)
    * Must consist of only numbers

– **coupon_id VARCHAR(100):** The id of the coupon that the costumer used.

Constraints
* Must consist of only numbers

– **delivery_address VARCHAR(100):** The address that the order has to be delivered to (can be different from the address of the customer)

Constraints
* Must not be null * Must consist of only numbers and letters * Must contain the street name and the number of the building

– **total_cost DECIMAL(10,2):** The total cost of the order.

Constraints
* Must not be null * Must be a positive number

- **order_date DATETIME:** The date that the order was placed on the restaurant by the costumer.

  Constraints

  * Must not be null
  * Must contain the month, day and year.
  * Must be of the form Day-Month-Year

- **status DECIMAL(2,1):** The status of the order. Where Cancelled = -1, On the way = 0, Delivered = 1

  Constraints

  * Must not be null
  * Must be a number in the set {-1,1,0}

- **DRIVER**
  - **ssn VARCHAR(100):** The social security number of the driver.

    Constraints

    * Must not be null
    * Must consist of only numbers
    * Must have a fixed size (depending on the country)

  - **fname VARCHAR(50):** The first name of the driver.

    Constraints

    * Must not be null
    * Must consist of only letter characters

– **lname VARCHAR(50):** The last name of the driver.

Constraints
* Must not be null * Must consist of only letter characters

– **phone VARCHAR(50):** The phone number of the driver.

Constraints
* Must not be null * Must be unique * Must consist of only numbers * Must have fixed size (depending on the country)

– **address VARCHAR(255):** The physical address of the driver.

Constraints
* Must not be null * Must consist of only letter and digit characters * Must contain the street name and building number

– **status BINARY(1):** The status of the driver.

Constraints
* Must not be null * Must be a number in the set {0,1} Where available = 1, not available = 0

– **gender BINARY(1):** The gender of the driver.

Constraints
* Must not be null * Must be a number in {0,1} Where male = 0 and female = 1

– **birth_date DATETIME:** The birth date of the driver

Constraints
* Must not be null * Must be a date in the form Day-Month-Year * Must be in limited range (that depends on the regulations of the country and common sense)

- **PAYMENT**
  – **trans_id VARCHAR(100):** The transaction id of the transaction between the customer and the restaurant.

Constraints
* Must not be null * Must be unique * Must consist of only numbers

– **receipt_num VARCHAR(100):** The receipt number of the order.

Constraints
* Must not be null * Must be unique * Must consist of only numbers

– **payment_method VARCHAR(100):** The method that the customer chooses to pay for the order.

Constraints
* Must not be null * Must be in a limited range of options that are provided by the restaurant or the e-food service

– **payment_status VARCHAR(100):** The status that represents the state that the payment process is at.

Constraints
* Must not be null * Must be a number in {1, 0, -1} Where completed = 1, in progress = 0 and cancelled = -1

– **trans_date DATETIME:** The date that the transaction takes place.

Constraints
* Must not be null * Must be a valid date (not a date of the previous dates) * Must be of the form Day-Month-Year

- **COUPON**
  - **coupon_id VARCHAR(100):** The id of the coupon that is used by the costumer.

Constraints
* Must not be null
* Must be unique

  - **coupon_code VARCHAR(100):** The code of the coupon that is used in order for the discount to be activated.

Constraints
* Must not be null

  - **discount DECIMAL(4,2):** The percentage of discount that will be applied.

Constraints
* Must not be null
* Must be a positive float

  - **expiration_date DATE:** The date that the coupon expires.

Constraints
* Must not be null
* Must be a date in the form Day-Month-Year

- **RATING**

Note: A constraint of the entity is that at least one foreign key must not be null (except the customers which will always not be null).

– **rating_id VARCHAR(100):** The id of the rating.

Constraints
* Must not be null * Must be unique * Must consist of only numbers

– **customer_username VARCHAR(100):** The username of the customer that submits the rating.

Constraints
* Must consist of letters and numbers only

– **rest_vat VARCHAR(100):** The VAT number of the retaurant that is being rated.

Constraints
* Must only consist of numbers * Must have a fixed size (depends on the country)

– **driver_ssn VARCHAR(100):** The ssn of the driver that is being rated.

Constraints
* Must consist of only numbers * Must have a fixed size (depends on the country)

– **item_barcode VARCHAR(100):** The barcode of the item that is being rated.

Constraints
* Must consist of only numbers

- **rating DECIMAL(2,1):** The rating that is submitted by the costumer.

Constraints
* Must not be null * Must consist of only one integer in the range {1,2,3,4,5}

- **review TEXT(200):** The review that the customer decides to write along with the rating.

Constraints
* This attribute has no constraints

- **OWNER**
  - **ssn VARCHAR(100):** The ssn of the owner.

Constraints
* Must not be null * Must consist of only numbers * Must be unique * Must be of fixed size (depends on the country)

– **fname VARCHAR(50):** The first name of the owner.

Constraints
* Must not be null * Must consist of only letter characters

– **lname VARCHAR(50):** The last name of the owner.

Constraints
* Must not be null * Must consist of only letter characters

– **phone VARCHAR(50):** The phone number of the owner.

Constraints
* Must not be null * Must consist of only numbers * Must be of fixed size (depending on the country) * Must be unique

– **address VARCHAR(255):** The physical address of the owner.

Constraints
* Must not be null * Must have the name of the street and the number of the building

– **gender BINARY(1):** The gender of the owner.

Constraints
* Must not be null * Must be an integer in the set {0,1} Where male = 0 and female = 1

– **birth_date DATETIME:** The birth date of the owner.

Constraints
* Must not be null * Must be a valid date (depends on the regulations of the country for workers) * Must be a date in the form Day-Month-Year

3.3 Relationships

3.3.1 One-to-One Relationships

- **PAYMENT-ORDERS:** A payment can correspond to only one order and an order can have only one payment. :

3.3.2 One-to-Many Relationships

- **RESTAURANT-MENU_ITEM:** A Restaurant can have many menu items, but a menu item can belong to only one restaurant.
- **RESTAURANT-ORDERS:** A restaurant can have many orders but an order can only correspond to one restaurant.
- **RESTAURANT-RATING:** A restaurant can have many ratings but a rating can correspond to only one restaurant.
- **MENU_ITEM-RATING:** A menu item can have many ratings but a rating can correspond to only one menu item.
- **DRIVER-RATING:** A driver can have many ratings but a rating can correspond to only one driver.

- **CUSTOMER-RATING:** A costumer can give many ratings but a rating can only be given by one costumer.
- **COSTUMER-ORDERS:** A customer can place many orders but an order can be placed by only one costumer.
- **COUPON-ORDERS:** A coupon can correspond to many orders but an order can take only one coupon.

3.3.3 Many-to-Many Relationships

- **OWNER-RESTAURANT ≔ OWNED:** The owner can own many restaurants and a restaurant can have many owners.

  - **rest_vat VARCHAR(100):** The VAT of the restaurant that is owned by the owner.

    Constraints

    * Must not be null
    * Must consist of only numbers
    * Must be of a fixed size (depends on the country)

  - **owner_ssn INT:** The ssn of the owner.

    Constraints

    * Must not be null
    * Must consist of only numbers
    * Must be of fixed size (depends on the country)

– **start_date DATETIME:** The date that the owner started owning the restaurant.

Constraints
* Must not be null * Must be a valid date * Must be a date of the form Day-Month-Year

– **hours DECIMAL(10,2):**The monthly hours that the owner has to spend for the restaurant's duties.

Constraints
* Must not be null * Must be a positive float

– **investment DECIMAL(10,2):** The amount of money that the owner has invested to the restaurant.

Constraints
* Must not be null * Must be a positive float

- **DRIVER-ORDERS ≔ DELIVERS** The driver can deliver many orders and the orders can be delivered by many drivers.

  - **receipt_num VARCHAR(100):** The receipt number of the order transaction between the customer and the retaurant

    Constraints

    * Must not be null
    * Must be unique
    * Must consist of only numbers

  - **driver_ssn VARCHAR(100):** The ssn of the driver that delivers the order.

    Constraints

    * Must not be null
    * Must have a fixed size (depends on the country)
    * Must consist of only numbers

  - **time_start TIME:** The time that the order started.

    Constraints

    * Must not be null
    * Must be valid time in the form Hours-Minutes-Seconds

  - **time_end TIME:** The time that the order was delivered.

    Constraints

    * Must not be null
    * Must be valid time in the form Hours-Minutes-Seconds

- **distance DECIMAL(10,2):** The distance between the restaurant and the order's address.

Constraints
* Must not be null * Must be a positive float

- **tips DECIMAL(10,2):** The tips that the customer left for the driver.

Constraints
* Must not be null (if no tips are added then the default value is 0) * Must be zero or a positive float

- **ORDERS-MENU_ITEM ≔ INCLUDES**
  - **item_barcode VARCHAR(100):** The barcode of the item in the order.

Constraints
* Must not be null * Must consist of only numbers * Must be unique

  - **receipt_num VARCHAR(100):** The receipt number of the order that the item is in.

Constraints
* Must not be null * Must be unique * Must consist of only numbers

– **quantity INT:** The quantity of the item in the orders description.

Constraints
* Must not be null
* Must be a positive integer

3.4 Data Integrity

3.4.1 Entity Integrity

Data integrity is enforced since every entity has a primary key. This ensures the uniqueness of each row of every table. Below the entities along with their primary keys are listed.

- **CUSTOMER: customer_username**
- **RESTAURANT: rest_vat**
- **MENU_ITEM: item_barcod**
- **RATING: rating_id**
- **DRIVER: driver_ssn ("ssn" on the table)**
- **ORDERS: receipt_num**
- **PAYMENT: trans_id**
- **COUPON: coupon_id**
- **OWNER: owner_ssn ("ssn" on the table)**

3.4.2 Referential Integrity

In the e-food delivery database referential integrity is enforced using foreign keys. This ensures consistency of relationships between the tables in the database. Below the relationships along with the foreign keys are listed.

Note: The tables that represent the Many-to-Many relationships have both of the primary keys of the entities that participate in the relationship as foreign keys.

- **PAYMENT-ORDERS: {∅, trans_id}**
- **RESTAURANT-MENU_ITEM: {∅, rest_vat}**

- **RESTAURANT-ORDERS: {∅, rest_vat}**
- **RESTAURANT-RATING: {∅, rest_vat}**
- **MENU_ITEM-RATING: {∅, item_barcode}**
- **DRIVER-RATING: {∅, driver_ssn}**
- **CUSTOMER-RATING: {∅, customer_username}**
- **COSTUMER-ORDERS: {∅, costumer_username}**
- **COUPON-ORDERS: {∅, coupon_id}**
- **OWNER-RESTAURANT(OWNED): { rest_vat, owner_ssn}**
- **DRIVER-ORDERS(DELIVERS):{ receipt_num, driver_ssn}**
- **ORDERS-MENU_ITEM(INCLUDES):{ item_barcode, receipt_num}**

3.4.3 Domain Integrity

The Domain integrity is enforced through the constraints that have been specified for every attribute in section 3.2 . This ensures the validity of the data in every table of the e-food delivery database.

4 Deletion Constraints

4.1 Standalone Entities

- **RATING:** If a rating is deleted, no other entities are affected.
- **PAYMENT:** An order needs to have a payment. Therefore if a row in the PAYMENT table is deleted, a trigger will be executed and the corresponding row in the ORDERS table will be deleted.

4.2 Derived Entities

The rows of the following entities can only be deleted if a parent of their foreign keys is deleted.

- **OWNED**
- **DELIVERS**
- **INCLUDES**

4.3 Referential Entities

4.3.1 CUSTOMER:

If a row in the CUSTOMER entity is deleted, the following rules hold

- **RATING:** The rating of the customer will remain and the foreign key customer_username will be set to NULL. This happens so that the ratings don't get affected.
- **ORDERS:** The orders of the customer will remain and the username will be set to NULL so that the history of the orders does not get affected.

4.3.2 RESTAURANT

When a restaurant row is deleted, the following rules hold

- **RATING :**Since the restaurant does not exist then there is no reason for a rating to exist, therefore all the restaurants ratings will be deleted.
- **MENU_ITEM:** All the items it owns will also be deleted from the database.
- **ORDERS:** All the orders that were placed on the restaurant will be deleted too.
- **OWNED:** The row that represents the owned relationship between the restaurant and the owner will be deleted.

4.3.3 ORDERS

When a row from the ORDERS entity is deleted, the following rules hold

- **PAYMENT:** The payment of the order is deleted.
- **INCLUDES:** The items that the order includes are deleted.
- **DELIVERS:** The delivery of the order is deleted.

4.3.4 DRIVER

When a row from the DRIVER entity is deleted, the following rules hold

- **RATING:** The ratings of the driver will be deleted.
- **DELIVERS:** The orders that the driver delivered will be deleted.

4.3.5 OWNER

When a row from the OWNER entity is deleted, the following rules hold

- **OWNED:** The row that represents the owners ownership of the restaurant will be deleted.

4.3.6 COUPON

When a row from the COUPON entity is deleted, the following rules hold

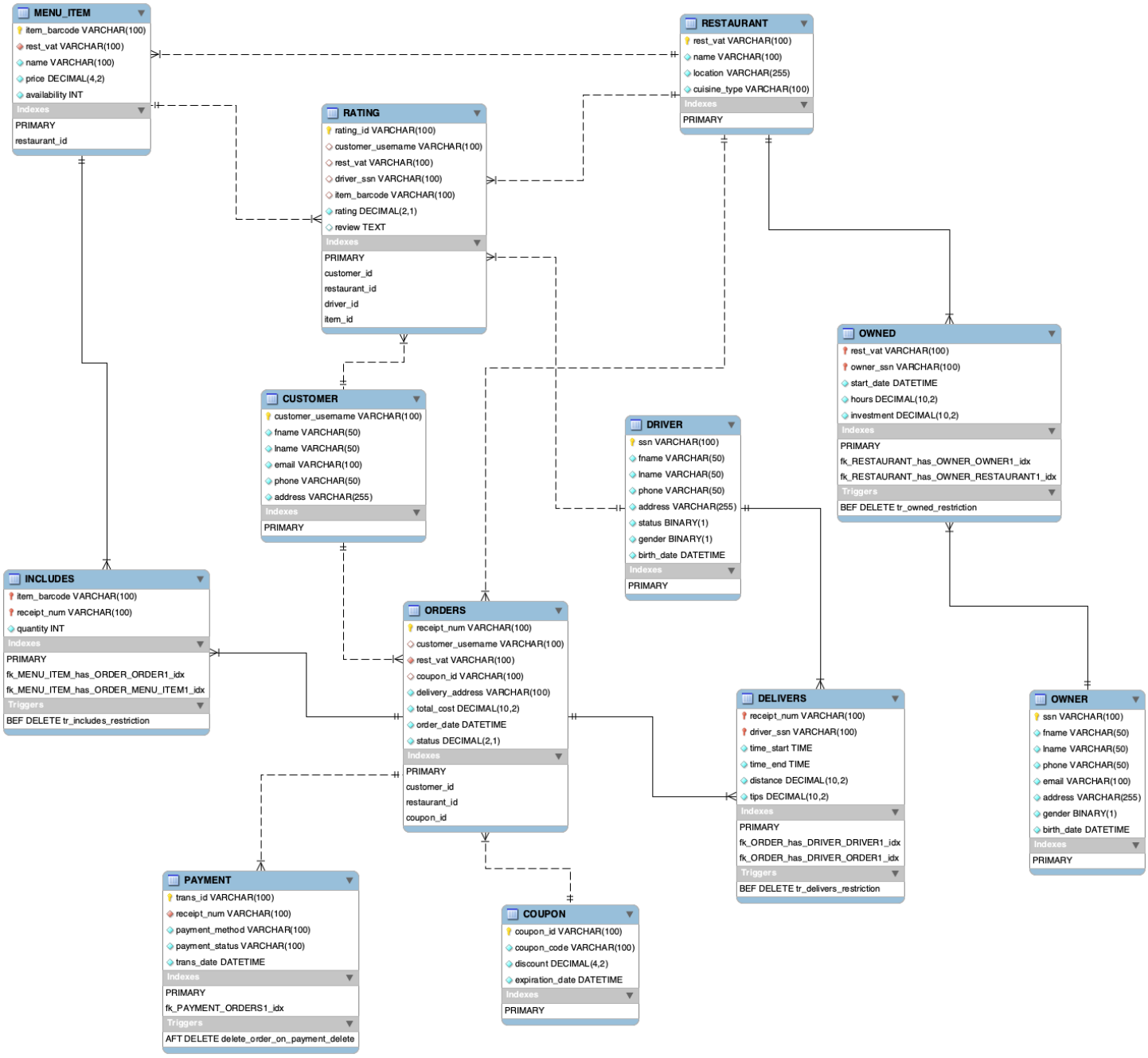
- **ORDERS:** The coupon_id attribute in the ORDERS entity will be set to NULL

4.3.7 MENU_ITEM

When a row in MENU_ITEM is deleted the following rules hold.

- **INCLUDES:** The INCLUDES row that corresponds to the item gets deleted, but the corresponding ORDERS row does not get affected.
- **RATING:** . Since the item will not exist, there is no reason for the ratings of the item to remain, therefore the ratings for the item will also be deleted.

5 Entity-Relationship Diagram



6 Relational Schema