Logo

Description automatically generated

**DoorDash Food Delivery Service**

**CS-6360:001 - Database Design**

**Team Number: 31**

**Team Members:**

1. **Manan Dalal (MUD200000)**
2. **Vishesh Mehta (VJM190001)**
3. **Fenil Godhani (FKG210000)**

**Overview:**

**About DoorDash:**

* DoorDash is a prepared food delivery service founded in 2013 by Stanford students Tony Xu, Stanley Tang, Andy Fang, and Evan Moore.
* A Y combinator–backed company, DoorDash is one of several technology companies that use logistics services to offer food delivery from restaurants on-demand.
* DoorDash launched in Palo Alto and, as of May 2019, had expanded to more than 4,000 cities and offers a selection of 340,000 stores across the U.S. and Canada.
* The company is currently worth more than $13 billion and is the largest third-party delivery service in the USA, surpassing Grubhub in 2019.

**Why we chose DoorDash?**

* As the company explains, DoorDash creates opportunities by empowering local businesses and in turn, generating new ways for people to earn, work and live.
* The purpose of choosing DoorDash database is to dig deeper into learning how a food delivery service as DoorDash has revolutionized the people’s eating habits and standards and has helped generate jobs.
* The business model used by DoorDash is intriguing and we wanted to learn about the data requirements of such a database to help us find some meaningful real life database storage problem solving experience.

**How does DoorDash Work?**

* DoorDash has several users that are either customers, dashers or business partners.
* Business partners are the restaurant/store owners who partner with DoorDash.
* Dashers are the people who work for DoorDash and deliver food door-to-door to the customers.
* Once the customers provide their location details, DoorDash displays all the partnered restaurants/stores in a ‘x’ mile radius from the provided location.
* A customer can then search for or filter out restaurants based on various parameters like cuisine, ratings, etc.…, add items to his/her cart and place an order.
* Once the order is prepared, a dasher picks up the contents and delivers them to the customer’s address.
* The customer must prepay and can give a review for that order, restaurant or/and the dasher.
* The customer can also apply for a refund if he/she finds the order to be unsatisfactory.
* The customer and the restaurant can also cancel an order if something goes wrong.

**Assumptions**

* Food is always available between the time a restaurant opens until it closes. That means that we would not check the quantity of the available food.
* Customers will be shown restaurants within a particular radius only.
* Customers are only allowed to order food from one restaurant at a time. That means that a single order cannot have items from multiple restaurants.

**Data/Functional Requirements**

**Customers:**

* Create/Update Login and Contact Information
* They can search for restaurants based on name, cuisine, menu items etc...
* They can add items to their cart and place orders.
* Can add instructions and specific requests within the order.
* Receive updates and track the status of their order.
* Pay, and ask for refund for their order.
* Can tip the dasher.
* View their order histories.
* Can provide feedback/reviews for the order, restaurant or/and the dasher.
* Can become a member and enjoy other benefits by buying DashPass.

**Restaurants:**

* Create a profile and can add new items/edit existing items to the menu.
* They can receive orders, update status of the order and get information of the dasher.
* They can discontinue with DoorDash with mutual consensus.
* Receive payments from the customer via DoorDash.
* Can view insightful data such as most/least items ordered, best/least rated items, user feedbacks etc.…
* Can add bank information to receive payments.

**Dashers:**

* Choose from available orders to pick up around their area/current location.
* Know when the order is available for pickup.
* Can contact customer/restaurant in case there is discrepancy.
* De-Register incase they want to discontinue.
* Can receive payments/fees from DoorDash and tips from customers.
* Can add bank/wallet information to receive payments.

**ER/EER Diagram**

**Diagram

Description automatically generated**

**Work-Flow Diagram**

**Diagram, engineering drawing

Description automatically generated**

**Mapping of ER/EER Diagram to Relational Schema**

**Initial Relational Model**

Below is the initial relational model derived from the ER/EER using the proper conversion rules.

**Text

Description automatically generated with low confidence**

**Normalizing Initial Schema Design**

**1-NF: Removing multiple values from the attribute and create another record(entry) to achieve 1-NF.**

**Example:** In the Relating USER, there can be multiple values for phone numbers and emails.

**Table

Description automatically generated**

Like the above example, we do this operation for all such values in our schema. Thus, we can say that our database schema relation is now in 2-NF

**2-NF: The 2nd Normal Form states that there must be no partial dependencies in any relation. This means that all attributes must solely depend on the entire key and not just on a subset of the key.**

* On examining all our relations, we can observe that the only the relations CART-ITEM and ORDER-ITEM have composite keys.
* But the non-prime attribute ‘quantity’ is functionally dependent on the entire key. Hence, we can say that it does not have any partial dependency.
* All other relations have only one attribute as key and hence for those relations having a partial dependency is not a possibility.

Thus, we can say that our database schema relation is now in 2-NF.

**3-NF: The 3rd Normal Form states that there must be no transitive dependencies in any relation. This means that all attributes must solely depend on the key attribute and not on some other non-key attribute.**

* We can see in the relation CUSTOMER that the attributes such as are functionally dependent ‘Street’, ‘Apt Number’, ‘City’, ‘AreaCode’, ‘State’, ’Longitude’ and Latitude are dependent on the attribute ‘AddressId’ and attributes like ‘CardExp’ and ‘NameOnCard’ are dependent on ‘CardNumber’. This is the very definition of a transitive dependency.
* Thus, we can now separate the relation CUSTOMER into three different relations such as:

R1: Customer (SSN, DashPass, Default CardNo, Default AddressId, AddressId, Street, Apt, City, State, Longitude, Latitude, AreaCode, CardNo, NameOnCard, CardExp)

Turns into the following:

R1: Customer (SSN, DashPass, Default CardNo, DefaultAddressId)

R2: Customer\_address (AddressId, Street, Apt, City, State, Longitude, Latitude, AreaCode)

R3: Credit\_Card (CardNo, NameOnCard, CardExp)

* Similarly, we apply same logic to the other relations having transitive dependencies.

Once all the transitive dependencies are resolved, we can say that our database schema relation is now in 3-NF.

**Final Relational Model**

After applying normalization, the below diagram is the normalized relational schema of our system.

**Diagram

Description automatically generated**

**SQL Statements**

**Creating Tables**

After performing normalization on the original set of tables, the following are all the tables that our system would require to function properly along with their MySQL definitions:

1. User

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Customer

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Customer Address

Graphical user interface, text, application

Description automatically generated

1. Credit Card

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Bank Account

Graphical user interface, text

Description automatically generated with medium confidence

1. Dasher

Graphical user interface, text, application

Description automatically generated

1. Restaurant Owner

Graphical user interface, text, application

Description automatically generated

1. Restaurant Address

Graphical user interface, text, application

Description automatically generated

1. Restaurant

Graphical user interface, text, application

Description automatically generated

1. Menu Item

Graphical user interface, text, application

Description automatically generated

1. Cart

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Cart Item

Graphical user interface, text

Description automatically generated

1. Discount Coupon

Graphical user interface, text, application

Description automatically generated

1. Order

Graphical user interface, text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

1. Order Item

Text

Description automatically generated with medium confidence

1. Payment

Graphical user interface, text

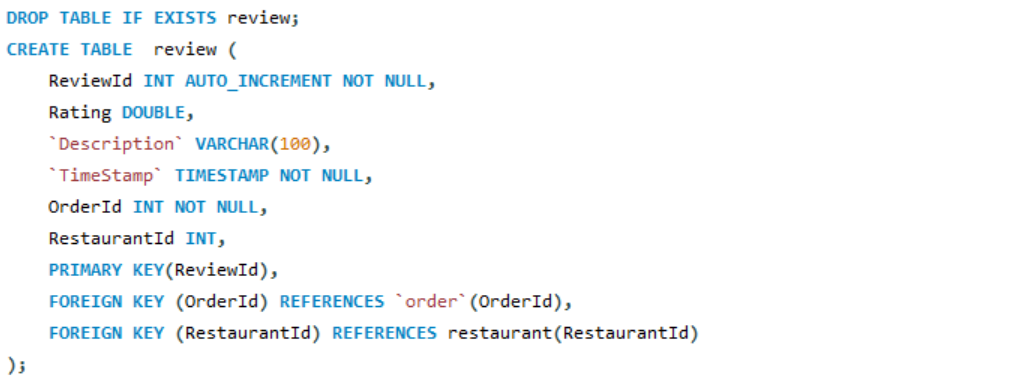
Description automatically generated

1. Refund

Graphical user interface, text, application

Description automatically generated

1. Review



1. Dasher Rating

Graphical user interface, text, application, chat or text message

Description automatically generated

**Inserting Data**

As shown below, here are some sample insertion operations on all the tables of the database system.

**A picture containing text

Description automatically generated**

**A picture containing text

Description automatically generated**

**A picture containing table

Description automatically generated**

**Insights into Data**

1. **To find average rating of all restaurants.**

Text

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated

1. **To check how many restaurant owners own more than 1 restaurant.**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, table

Description automatically generated

1. **To check how many users have a DashPass and how many don’t.**

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

1. **To find the restaurants who get the maximum orders.**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

1. **To find out how many times customers are asking for refund.**

Graphical user interface, text

Description automatically generated

Graphical user interface, application

Description automatically generated

**Views**

1. **Creating a view that shows the total number of restaurants in different cities.**

**Text

Description automatically generated with medium confidence**

**Table

Description automatically generated with medium confidence**

1. **To visualize all important information of a customer.**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface, table

Description automatically generated**

1. **To get all important details of an order.**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generatedTable

Description automatically generated**

1. **To find out how well each dasher is performing based on their average rating.**

**Text

Description automatically generated with medium confidence**

**Graphical user interface, text, application

Description automatically generated**

**PL/SQL Statements**

**Triggers**

1. **Creating a trigger that can satisfy the condition that “An order cannot be deleted if it has not been delivered or refunded”**

**Graphical user interface

Description automatically generated with medium confidence**

**Graphical user interface, application, Word

Description automatically generated**

****

1. **Creating a trigger that can satisfy the condition that “A customer can only have items from a single restaurant in his cart at a single time. If he adds an item from another restaurant, then the cart should be emptied first.”**

**Graphical user interface, text, application, email

Description automatically generated**

****

**Stored Procedures**

1. **A Customer can search for restaurants using filters like cuisine, city etc..**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

1. **DoorDash can find out the earnings of each restaurant between a particular period.**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

1. **A Restaurant can change the status of an order as it gets processed and completed.**

**Graphical user interface, text, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application

Description automatically generated**

*Before**After*