

The Database Life-Cycle
Re-engineering the Early Stages of a Startup's Database

Phase 2

Team S2-20: Ryan Huang, Joanne Tsai

1. Users

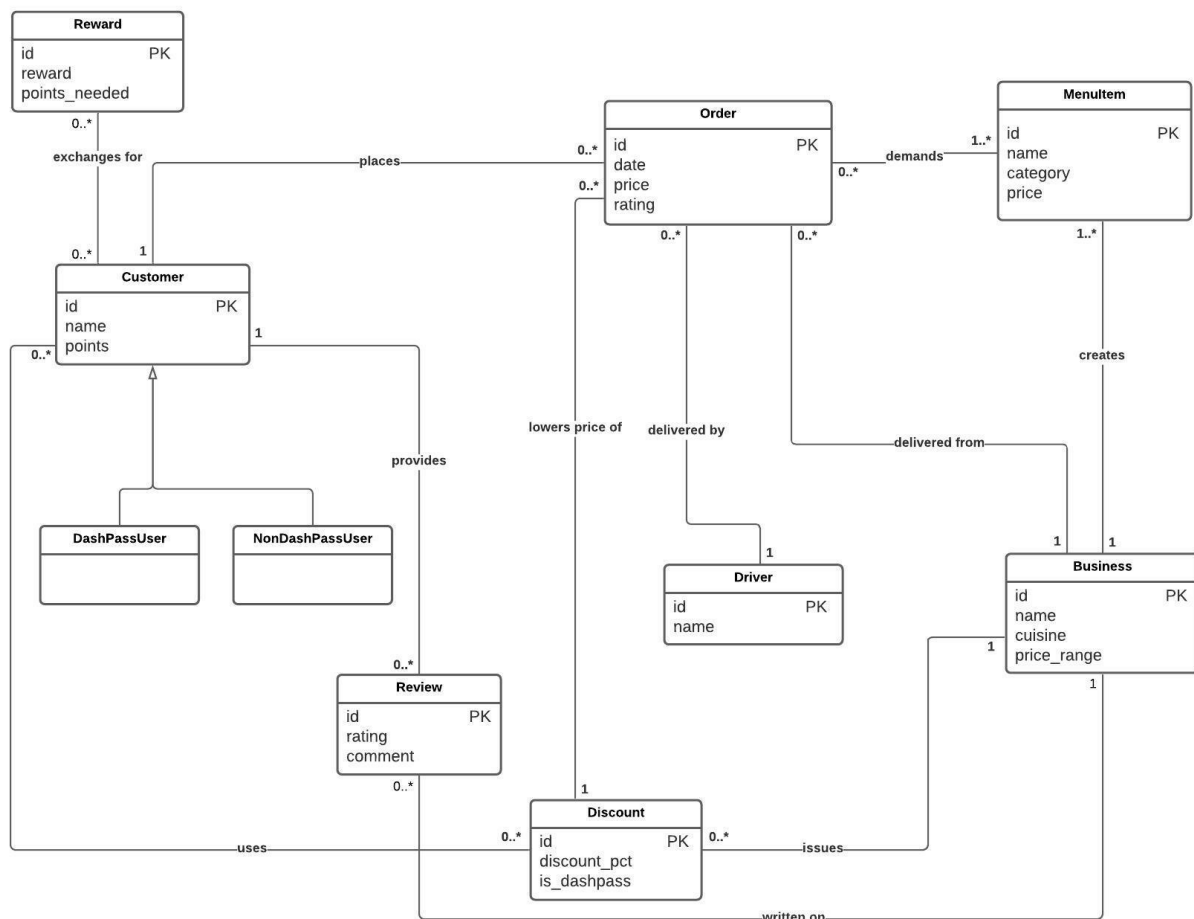
Role 1	Customer	An individual who is seeking a particular service by a business and wants the most convenient form of delivery on their behalf in order to feel confident in their payment.
Role 2	Business	A business that has a need to implement a quick, efficient, and coordinated delivery system for their services to customers.
Role 3	Driver	Independently contracted drivers on behalf of DoorDash who want to maximize their profits with maximum efficiency in the process of doing so.

2. User Stories

ID	Simple/ Complex/ Analytical	As a <role>	I want to <goal>	So that <reason>
US1	simple	customer	Filter cuisine and price to select a restaurant	I can find a place that serves the food I want at the price I can accept
US2	analytical	customer	Sort the restaurants by the most amount of reviews	I can choose an option that has been vetted which is more likely to be the most popular option
US3	complex	customer	Find a list of businesses of a particular price range that offer discounts for dashpass users	I can enjoy the product at the most cost efficient price as a loyal user of Doordash
US4	complex	business	See my total revenue in a month in a given year	I can see if my business is making profit in the short run
US5	analytical	driver	See my earnings in a given year	I can get a sense of how much income I have earned by delivering throughout the year.
US6	analytical	driver	Count how many orders I have in a day	I can see how many orders I have to make that day
US7	simple	business	Insert incoming orders into the system	I can efficiently create and deliver finished products in a timely

				manner
US8	complex	business	Get the average rating of all drivers who have driven for me	I can ensure that customers are getting the best delivery experience possible
US9	complex	customer	<i>(new functionality)</i> Check what possible rewards I can get with my current doordash points	I can use the doordash points I have gained effectively
US10	complex	business	Find customers who have left me a review	I can thank my customers who have provided feedback

3. Conceptual Model



Link to Conceptual Model:

https://lucid.app/lucidchart/2a335f8f-3aa3-4dd9-b44d-f07c9c0599b1/edit?viewport_loc=-416%2C-33%2C2855%2C1432%2C0_0&invitationId=inv_105ef378-e64d-4c93-8c73-37a0bf1f8c24

4. Relational Model

Customer(**id**, name, points)

DashPassUser(**id**)

NonDashPassUser(**id**)

Driver(**id**, name)

Business(**id**, name, price_range, cuisine)

MenuItem(**id**, name, category, price, business_id)

Order(**id**, date, price, customer_id, driver_id, business_id, discount_id, rating)¹

Review(**id**, rating, comment, customer_id, business_id)

Discount(**id**, discount_pct, is_dashpass, business_id)

Rewards(**id**, reward, points_needed)²

Delivery(business_id, drive_id)

Use(customer_id, discount_id)

Demand(order_id, menuItem_id)

Exchange(customer_id, reward_id)

5. Functional Dependencies & Normalized Schema

Table 1: Customer

Function Dependency: [id -> name, points]

Explanation: Each customer id corresponds to one customer name and one specific number of points that the customer has and is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 2: DashPassUser

Functional dependencies: [id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.

¹ *Note: "Rating" is set to default NULL since the value of rating will only be updated by the system after the delivery, i.e., when an incoming order is inserted (implemented in US8), the rating should be left blank (NULL). Also, the "rating" in this table is the customer rating for drivers, and because it is not included in any of the user stories, any driver ratings listed have been made after the order has been delivered to the customer.

² *Note: Rewards are offered by Doordash, not the businesses. Also doordash points are supposed to be gained through making orders but since that is not part of any of our user stories, the points seen in the database have already been obtained by the customer.

Table 3: NonDashPassUser

Functional dependencies: [id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.

Table 4: Driver

Function Dependency: [id -> name]

Explanation: Each driver id corresponds to one driver's name and it is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 5: Business

Functional Dependency: [id -> name, price_range, cuisine]

Explanation: Each Business id corresponds to one business name, one price range, and cuisine and is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 6: MenuItem

Functional Dependency: [id -> name, category, price]

Explanation: Each MenuItem id corresponds to one name of the meal, one category, and one price and is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 7: Order

Functional Dependency: [id -> date, price, customer_id, driver_id, business_id, rating]

Explanation: Each Order id corresponds to one date, one price, one customer id, one driver id, one business id, and one rating and is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 8: Review

Functional Dependency: [id -> rating, comment, customer_id, business_id]

Explanation: Each Review id has only one corresponding rating, comment, customer id, and business id and is the only data that can lead us to this information. There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 9: Discount

Functional Dependencies: [id -> discount_pct, is_dashpass, business_id]

Explanation: Each Discount id has only one corresponding discount percent, one is_dashpass check to see if it is offered to dashpass users only, and one business id (the business who offered the discount). There are no repeating values, partial dependencies, transitive dependencies, or bad functional dependencies and it is therefore in BCNF.

Table 10: Reward

Functional Dependencies: [id -> reward]

Explanation: Each Reward id corresponds to one specific reward and it is the only data that can lead us to this information. It is therefore in BCNF.

Table 11: Delivery

Functional dependencies: [business_id, driver_id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.

Table 12: Use

Functional dependencies: [customer_id, discount_id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.

Table 13: Demand

Functional dependencies: [order_id, menuItem_id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.

Table 14: Exchange

Functional dependencies: [customer_id, reward_id ->]

Explanation: There are trivially no bad functional dependencies and it is therefore in BCNF.