

Database Design for Online Retail Store Winter 2022

Abhimanyu Bhatnagar, 2020273

Atyant Sony, 2020039

Madhava Krishna, 2020217

Ritika Nagar, 2020112

April 24, 2022

Contents

Chapter 1

Defining the Problem

1.1 Objective

The goal is to design a database management system for an online retail store, similar to Big Bazaar, Flipkart and Amazon. Our project models on Grofers (now BlinkIt), a fast grocery marketplace for consumers to purchase day-to-day goods from.

1.2 Problem Statement

The ER diagrams and the relational schema that follow are based on the following (rudimentary) problem statement:

The online retail store serves many customers. The customers are required to hold an account on the platform to be able to purchase items. They can create an account by specifying their name, email address, phone number, and address. Customers add products to a shopping cart. They apply coupons on the shopping cart; the coupons have a coupon code and an associated discount percentage.

Customers order items by checking out the items on their shopping cart. The order is placed once the transaction is confirmed. The order is delivered to the customer only, they cannot have the order delivered to other addresses. A product can belong to various categories and has specifications and a cost. Each product is obtained from a vendor which the store transacts with. After being purchased from the vendor, the products are stored in a warehouse.

Warehouse employees are responsible for packing and preparing orders. The readied order is then delivered to the customer by a delivery agent. In case of any lapses with an order, the customer complains to support staff who create a complaint number against the order. They send out the details regarding the complaint to the customer.

1.3 Stakeholders

Notable stakeholders of this problem include:

1. Customer
2. Employees
3. Suppliers
4. Management of the company: board of executives, shareholders

Other stakeholders include manufacturers, warehousing service providers

1.4 Assumptions

1. The retail store would be operated in one country only. Therefore, there isn't an option to change the country.
2. There will be a shopping cart associated with the customer's account. They won't be able to add item to cart without being logged in.
3. The delivery would be taken care of by the company itself; items will be shipped from one warehouse only (Similar to Blinkit, erstwhile Grofers).
4. Coupons would be applied on the order, not the cart. Coupon would be applied during the checkout process.
5. Coupons would be applied using the coupon code, which is a unique alphanumeric value. A coupon cannot be reused.
6. Employees will be divided into delivery partners (responsible for delivering the order), warehouse workers (tasked with preparation of orders) and service employees (responsible for conflict resolution).

7. Vendors have only one account number and will be based only in India.
8. Employees have only one email address.
9. There won't be any wishlisting features, nor any saved-items feature like the ones offered by Amazon.
10. Employees' performance would be graded on a scale from 1 to 10 (for ease of data entry) with decimal values being permissible.
11. The mode of payment would be Cash on Delivery (no alternate methods would be provided) for the sake of simplicity.

Chapter 2

ER Diagram

2.1 Entities

1. Customer

- (a) Customer ID: Primary key
- (b) Name: composite
 - i. First Name
 - ii. Last Name
- (c) Address: composite
 - i. House Number
 - ii. Locality
 - iii. City
 - iv. State
 - v. Pin Code
- (d) Email Address
- (e) Password

2. Product

- (a) Product ID: Primary key
- (b) Price
- (c) Category

- (d) Discount Percentage
- (e) GST

3. Orders

- (a) Order ID
- (b) Total_Cost
- (c) Taxes
- (d) Discount Percentage

4. Vendor

- (a) Vendor ID
- (b) Address
 - i. Plot number
 - ii. City
 - iii. State
 - iv. Pin code
- (c) Name
- (d) Phone number
- (e) Email_address
- (f) password

5. Coupon

- (a) Coupon Code: Primary key
- (b) Discount Percentage
- (c) Validity

6. Warehouse

- (a) Warehouse ID
- (b) Address: composite
 - i. Plot number
 - ii. City

- iii. State
- iv. Pin Code
- (c) Telephone Number: multivalued

7. Employee:

- (a) Employee ID: Primary key
- (b) Date of Joining
- (c) Position
- (d) Department
- (e) Email Address
- (f) Name: composite
 - i. First Name
 - ii. Last Name
- (g) Residential Address: composite
 - i. House number
 - ii. Locality
 - iii. City
 - iv. State
 - v. Pin Code
- (h) Date of Joining
- (i) Performance
- (j) Salary
- (k) Gender
- (l) Date of Birth
- (m) Age : derived
- (n) Password

Employees can be specialised into:

- Delivery Partner:
 - (a) Vehicle ID

- (b) Vehicle Type
 - Warehouse Worker: no additional attributes
 - Service Employee: no additional attributes
8. Shopping Cart (Weak entity)
- (a) Customer ID, Product ID: Primary Key
 - (b) Customer ID: Foreign Key (references Customer)
 - (c) Product ID: Foreign Key (references Product)
 - (d) Quantity
 - (e) Total cost: Derived attribute

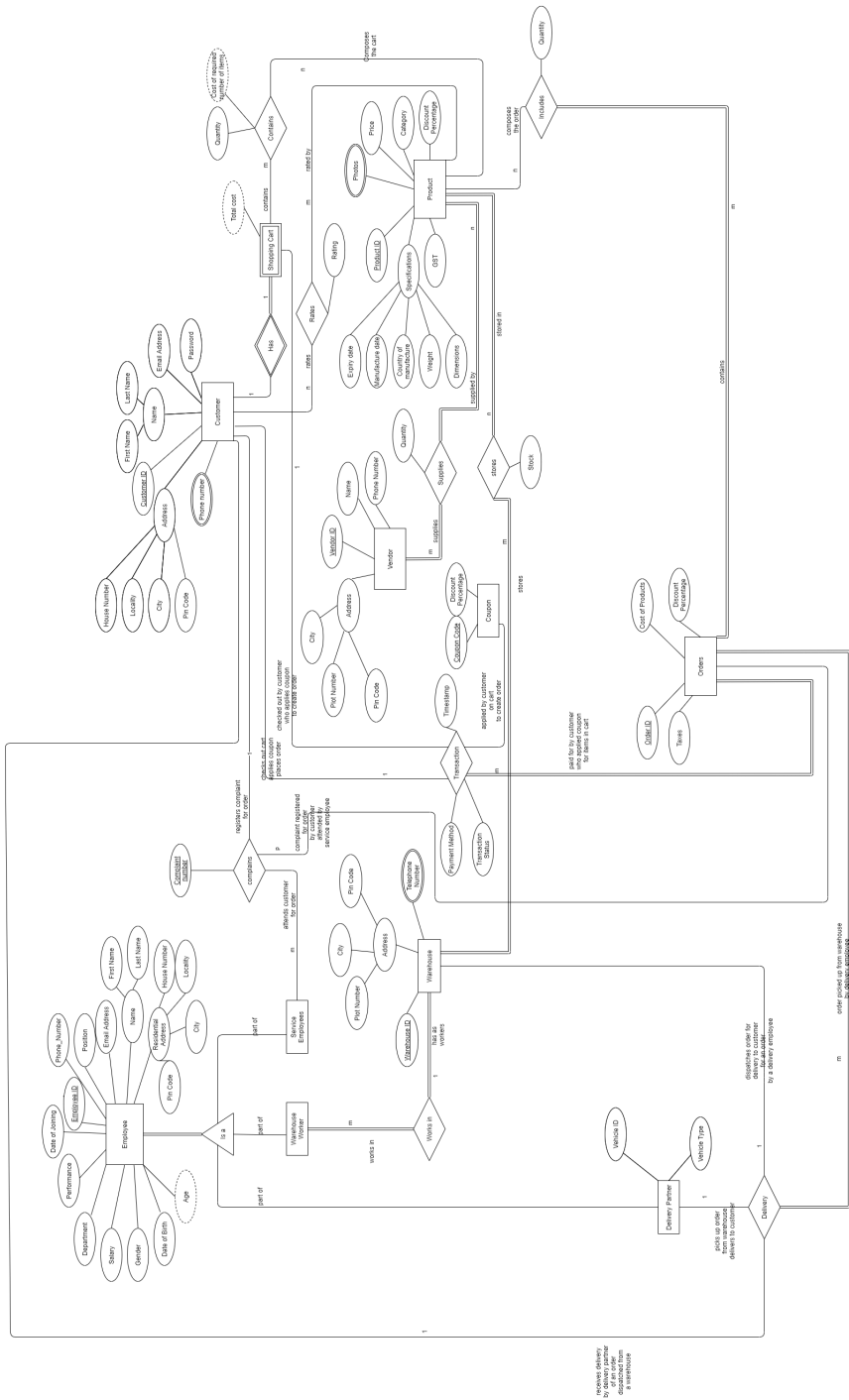
2.2 Relations

1. Supplies: between vendor and product
 - (a) Vendor ID, Product ID: Primary Key
 - (b) Vendor ID: Foreign Key (references Vendor)
 - (c) Product ID: Foreign Key (references Product)
 - (d) Quantity: attribute
2. Includes: between product and order to show which order contained what products
 - (a) Order ID, Product ID: Primary Key
 - (b) Order ID: Foreign Key (references Order)
 - (c) Product ID: Foreign Key (references Product)
 - (d) Quantity: attribute
3. Stores: between warehouse and product to show which warehouse contains what products.
 - (a) Warehouse ID, Product ID: Primary Key
 - (b) Warehouse ID: Foreign Key (references Warehouse)
 - (c) Product ID: Foreign Key (references Product)

- (d) Stocks: attribute
- 4. Delivery: Quarternary relation between warehouse, order, delivery partner and customer to represent delivery.
 - (a) Order ID: Primary Key
 - (b) Order ID: Foreign Key (references Order)
 - (c) Employee ID: Foreign Key (references Employee)
 - (d) Customer ID: Foreign Key (references Customer)
 - (e) Warehouse ID: Foreign Key (references Warehouse)
 - (f) Delivery_date
- 5. Transaction: Relation between customer and their cart, order and coupon code.
 - (a) Order ID: Primary Key
 - (b) Customer ID: Foreign Key (references Customer)
 - (c) Payment Method
 - (d) Transaction Status
 - (e) Transaction Time
 - (f) Coupon Code: Foreign Key (references Coupon)
- 6. Complains: between service employees (specialisation), customer and order to indicate a dispute.
 - (a) Complaint Number: Primary Key
 - (b) Customer ID: Foreign Key (references Customer)
 - (c) Order ID: Foreign Key (references Order)
 - (d) Service Employee ID: Foreign Key (references Employee)
 - (e) date_of_crelation
 - (f) resolved
 - (g) details
- 7. Works in: between Warehouse Worker and warehouse.

- (a) Employee ID, Warehouse ID: Primary key
- 8. Has: relation between customer and shopping cart (weak entity).
 - (a) Customer ID: Primary key
- 9. Contains: relation between shopping cart and product to indicate that shopping cart contains the item.
 - (a) Quantity
 - (b) Cost of items in the cart: Derived quantity.
- 10. Rates: between customer and product for indicating rating.
 - (a) Customer ID, Product ID: Primary key
 - (b) Rating: attribute

2.3 Diagram



Chapter 3

Relational Schema

The ER diagram was reduced by noting multiplicities and the following tables for the relational schema resulted.

1. Customer(customer ID, First_name, Last_name, House_number, Locality, City, Pincode, Email_Address, Password)
2. Product(product ID, Price, Category, Discount Percentage, GST Percentage)
3. Vendor(Vendor ID, First Name, Last Name, Plot Number, City, Pincode,email_adress,password)
4. Warehouse(Warehouse ID, Plot Number, City, Pincode)
5. Employee(Employee ID, First name, Last name, Age, Salary, Gender, Department, Performance, Position, Date of Joining, Date of Birth, Email Address, House Number, Locality, City, Pincode, Phone Number,password)
6. Delivery Partner(Employee ID, Vehicle ID, Vehicle Type)
7. Warehouse Worker(Employee ID)
8. Service Employee (Employee ID)
9. Orders(Order ID, Total Price, Taxes, Total Discount Percentage)
10. Coupon(Coupon Code, Discount Percentage)

11. Transaction(Order ID, Payment Method, Transaction Status ,Transaction Time, Customer ID, Coupon code)
Here, Customer ID is referenced from Customer table and coupon code from Coupon table. Coupon code can be null, customer id cannot.
12. Delivery (Order ID, Employee ID, Customer ID, Warehouse ID,delivery_date)
Here Employee ID is taken from the Employee table if it belongs to Delivery Partner, Customer ID from Customer table, Warehouse ID from Warehouse table.
13. Stores(Warehouse ID, Product ID, Quantity)
14. Supplies(Vendor ID, Product ID, Quantity)
15. Shopping Cart(Customer ID, Product ID, Quantity)
16. Complains(Complaint Number, Customer ID, Order ID, Service Employee ID,resolved,date_of_creation,details)
Here, Customer ID is taken from Customer table, Order ID from Orders table, Service Employee ID from Service Employees table.
17. Order Products(Order ID, Product ID, Quantity)

Note: Employee has both fields of age and DOB. This is a redundancy on our part as the data entries were designed with age as an attribute. This may be eliminated in the final version depending on whether there is need or not (number of employees should be significantly lower than the number of customers).

The SQL queries for creating the database, populating the database and the queries required are submitted under the names: 'database.creation.sql', 'database_population.sql' and 'queries.sql' in the Database folder.

Chapter 4

Changes from the Mid Evaluation

Schema

1. Dropped vendor_phone, customer_phone, warehouse_phone, product_rating, product_photos tables to reduce complexity.
2. Converted complains quarternary relation to ternary relation.
3. Added attributes email_address and password in vendor.
4. Added password attribute in employee table.
5. Auto-incremented customer, product, vendor, warehouse, orders, complains (for ease of insertion).
6. Added attribute delivery_date in delivery.
7. Removed attribute category from product, added attribute product_name.
8. Added attributes details, resolved and date_of_creation in complains.

Queries and Views

1. Added another query to account for the missing query during the mid-evaluation.

2. Created new views and roles.
3. Created triggers pertaining to the application.

Deliverables for Final Evaluation

Project Scope and Relational Schema

Scope, depth and relation schema were updated from the mid-evaluation. Details in the previous chapter.

Views and Grants

Views were created (present in views.sql file in the Database directory). All views were created taking into account the requirement by the application. Corresponding roles were created and were granted authorisation.

SQL Queries

Queries are present in queries_endsem.sql.

Most of the queries were taken from the embedded queries which were required to run the application. Do note that the data was changed in order to generate the pure SQL queries from embedded queries and hence may not be coherent.

Embedded SQL Queries

All embedded SQL queries (ranging from easy to complex) are in embedded_sql.txt

Query optimisation

Queries are automatically optimised by MySQL, however, we had tried to optimise the number of calls by searching creative ways to only select as many columns as required, and created indexes to assist with querying.

Indexing

MySQL creates indices for primary keys. In addition to those, we had tried to optimise all those functions that we were using in the application.

Attributes for which we created index tables

- email_address and password from customer
- email_address and password from vendor
- stock from stores
- quantity from shopping_cart.
- products on shopping_cart.
- product_id from stores.
- customer_id from transaction
- service_employee_id from complains
- product_id from supply

Triggers

Triggers were created (present in Trigger.sql in the Database directory). The following triggers were created:

- For generating password for employee (uses MD5 checksum to generate temporary queries).
- Removing item from cart once it goes out of stock.
- Reduce discount percentage once stock reduces below 100.

- Insert into stores whenever a new product is inserted (adding it to list before incrementing value).

NOTE: All files pertaining to queries and the submission are in the Database director.

Overview of the Sites

Introduction

Keeping in mind that customers and employees and vendors have different requirements, a decision was taken to split the application into two parts: one which took care of the Customers' side, and another which catered to Employees and Vendors.

The scope was limited in several places, in order to create a high-quality, coherent product that would work well.

4.1 Customers' Portal / Main Portal

The customers' portal allows for several features:

1. Signing up new customers.
2. Browsing products
3. Logging in
4. Adding products to cart
5. Checking out items
6. Monitoring orders and delivery dates
7. Sending complaints regarding orders.

A few critical assumptions were made:

- The payment occurs only through Cash on Delivery. This decision was taken to limit creation of another form for payment details (which were getting hard to handle).
- After complaining about an order, the customer would be contacted via email by the assigned service employee, and they would not be able to track the status of their complaints. This decision was made looking at Amazon's flow, wherein complaining results in the company employees contacting the customer.
- There is no option to return an order, or rate them, for the same reason that forms were getting overwhelming to handle.
- Cart items cannot be dropped as a single unit, the cart can only be fully cleared (in order to reduce complexity).
- In case the item stock gets to 0, it would be removed from all customers' carts (due to the trigger).
- Delivery details won't be shown.

4.2 Buiseness Portal

This site is used by employees(service employees and delivery guy) and vendor.

- Vendors' portal flow
 1. New Vendor creates his/her account
 2. He/She logins to his/her account
 3. He/she can restock the products he/she is already selling
 4. He/she can add new products
- Employees' portal flow
 1. Delivery Employees and Service Employees can login to the portal
 2. Service Employees can see Customers' Complaints and mark them as resolved.

3. Delivery guy can see the orders that he/she has to deliver and mark them as delivered.

As with the customers' portal, a few assumptions were made:

- In employees' portal, we cannot register a new employee (database admin would be responsible for that).
- Warehouse workers have no provision to login.
- Service employees would only be responsible for handling complaints(would only be marking them as resolved after sending an external email, at least on the portal).
- Delivery agents would be allowed to login and they would only be shown a list of deliveries to be made. Only a provision to mark the item as delivered would exist.
- We would not cross verify with the customers whether they recieved the order or not. The customer can complain if the order was not recieved.
- Vendors would have an option to add a new product, but would not be shown the product_id of the newly inserted item (to reduce complexity, they would still be able to visit the customer's site and get to know the product id).
- Restocking only happens on the basis of product_id.

Chapter 5

Project Development Procedure and Deliverables

5.1 ER Formulation

Before commencing with the ER diagram formulation, many brainstorming sessions were held to determine which all aspects the online retail store would target and to what degree. The result of those sessions is the *problem statement* and the list of assumptions. The problem statement is a rudimentary sketch of what all scenarios the store should be equipped to handle and what to expect from the usage scenarios of the database management system. The ER diagram and relational schema have been added in the previous chapters and the diagram should be available in the ER_Diagram folder.

5.2 Weak Entity

The shopping cart was determined to be a weak entity since it does not have any identifying attribute other than belonging to a specific customer. Hence, on careful consideration of all its attributes and the lack of a distinctive attribute, it was decided to be kept as a weak entity.

5.3 Entity Relationship Participation and Types

This was another topic on which multiple meetings had been conducted, specifically on the nature of the ternary/quarternary relations and how to decompose them. Ultimately, Delivery and Transaction were kept because of their utility and ability to define the whole scenario without conflicts with other entities and relations. The participation types (total and partial), relationship roles and cardinality constraints have been mentioned in the ER diagram. Detailed constraints have been mentioned in the SQL database creation file.

5.4 Ternary Relation

A ternary relation: complains was identified which involves an order, the customer who placed that order and one of the service employees. The cardinality is many on the order's and service employee's side and unary on the customer's side. It was identified in the following way: "A customer can complain to many service employees about many orders."

5.5 Relational Schemas

They have been included in one of the earlier chapters.

5.6 Sufficient and Valid Constraints in DDL

The DDL file has been enclosed in the Database folder under the name 'database_creation.sql'.

5.7 Data Entry

Data was generated using Mockaroo and using python scripts for the foreign key tables. The primary keys were chosen in increasing order (similar to AUTO_INCREMENT) and most of the tables (except warehouse and complains, which contain 5 and 10 rows respectively), have 50 to over a hundred entries.

5.8 Queries

Queries have been included in the Database folder under the file name 'queries.sql', which should be run after running 'database_creation.sql' and 'data_population.sql' respectively.

Chapter 6

Contributions of Team Members

Collaboration was done by the means of Microsoft Visual Studio Code's Live Share feature which allowed for real time simultaneous editing of multiple files by many participants. Communication occurred through Google Meet links. Following are the contribution of each team member in the development and ideation of the project till the mid-evaluation. Many of the responsibilities overlapped and all tasks were completed with contributions from all.

1. Abhimanyu Bhatnagar:
 - (a) Ideation of the ER diagram
 - (b) Drawing the ER diagram
 - (c) Reduction to Relational Schema
 - (d) DDL : creating tables
 - (e) Creating test data
 - (f) Part of the SQL queries
 - (g) Documentation
 - (h) Triggers
 - (i) Views and grants
 - (j) Indexing
 - (k) Embedded sql queries(Database connection)

- (l) helped with the Customer page
- (m) Atyant Sony:
 - i. Ideation of the ER diagram
 - ii. Drawing the ER diagram
 - iii. Reduction to Relational Schema
 - iv. DDL : creating tables
 - v. Creating test data
 - vi. Part of the SQL queries
 - vii. Documentation and latex formatting.
 - viii. Handled HTML forms
 - ix. Embedded sql queries
 - x. Indexing
 - xi. Made front end
- (n) Ritika Nagar:
 - i.
- (o) Madhava Krishna:
 - i. Determining requirements and setting a problem statement.
 - ii. Drawing the ER diagram
 - iii. Reduction to Relational Schema
 - iv. Part of the DDL
 - v. Creating part of the test data
 - vi. SQL query
 - vii. Documentation and latex formatting.
 - viii. Database Connection
 - ix. Customer Portal Landing Page
 - x. Part of the employees' portal
 - xi. Part of the customers' page
 - xii. Triggers
 - xiii.