Project report:
Vidur Goel team 80
2021364

In this Online retail store end to end based database application application is managed by the admins. There are some number of admins in this application and each admin will have their unique email id and they will have their Full name registered as well as a unique username and password to log in to the application in order to manage the whole application and exercising it's authorities and given priviliges in the application. Admin is responsible for adding and managing the following entities:

- Category
- Product
- Coupon
- Dealers (who supplies the products)
- Delivery Boy (who delivers the orders to their delivery address).

There is an m to n relationship (i.e, many to many relationship) between admin and category i.e., any number of categories can be added by different admins or same admin as well as a category can be managed by different admins. Similarly, there is also m to n relationship between:

- admin and products
- admin and dealer
- admin and delivery boy
- admin and coupon

All the different categories of category entity must have been created and managed by the admin. Similarly, each product, each coupon, each dealer, each delivery boy of their respectively entity must have been added and managed by admin only i.e., there has to be atleast one admin responsible for their creation as well as management.

Each category will be assigned their unique category_id assigned to them which will act as their primary key as well as application maintains each category's name as well as description/info in database. Application also assign a unique product_id to each of the products created and also maintains each product's name, cost, brand name it belongs to as well as quantity of that product currently in the database available.

There is a belongs to relationship between the category as well as product entity i.e., each product belongs to at least one category as well as a category has at least one product associated to them and this is m to n relationship between them i.e, each product must be belonging to at least one of the category as well as each category must have at least one of the product belonging to that category.
There is also a weak entity named the product_feedback which has a “has” relationship as well as one to many relationship with the product entity as each product can have many or even no product reviews associated with it but each product review must have only one product with which it has a relationship with.
Application manages each product feedback’s unique feedback id assigned to it which acts a primary key and also manages review, rating , likes and date on which it is published.
Application also assigns the unique customer id to each customer who will sign up in the application which would basically acts as a primary key and application also manages each customer’s name, unique username, password which is used to log into the application which customers enters while signing up and each customer will also give their email id, contact number , address while signing up which will become the attributes of that customer. In our application customer entity have select relationship with the category which will be m to n relationship and in order to add the product into the cart customer need to browse the category and then product.
Each customer will have 1 unique cart assigned to them i.e., a one to one relationship between the cart and customer and each cart will have a unique cart_id assigned to them also cart will contain the necessary products information that customer has added and cart also have “consists of” relationship with the products which is one to many i.e, each cart can have multiple products with different quantities but each product can be assigned to only one cart or no cart at all.
Admin manages the delivery boy entity which has a unique delivey boy id assigned to it which will acts as a primary key and also manages name, unique username , password used to log in application also manages information about the avg rating, contact number and email id as an attribute of that delivery boy.
Delivery boy has 1 to many relationship with the customer such that many customers can rate the delivery guy but one customer can only rate the delivery guy assigned to it.
After adding all the products that customer needs into the cart there is a cart to order relationship between the cart and order which is one to one as 1 cart

can result in one order and order can be made by one cart only and for each order there has to be a cart. Thus application creates the order entity which have unique order id assigned to them and also have delivery address, total cost, delivery boy id (Foreign key) , Products id (Foreign Key), expected delivery date, Track your order, order satus (Delivered or not) as the attributes and foreign keys it also has a relationship with a weak identity named as the tracking details which contains track_id as a primary key and delivery boy name, contact number and location as its attributes. Order also has a 1 to 1 relationship with the bill entity . Bill entity have a unique bill_id as a primary key with it and also has is paid(yes or not), mode of payment,order summary as its attributes.
Admin also adds and manages coupons for the customers to be applied on the cart i.e, they have 1 to 1 relationship with the cart and assumed that only one or no coupon could be applied to the cart. Coupon entity will have a unique coupon id associated with it and also have coupon expiry date, minimum order value, percentage discount, maximum discount upto, terms and conditions, coupon's code attributes associated with it which are information entered by the admin. Coupon also has n to 1 relationship with the customer I.e, customer may many coupons but a coupon can have only one customer to which it is allotted.
Admin also add and manage different dealers. Each dealer will have a unique dealer id allotted to them which will acts as a primary key to them. They will also have name, username, password, address of operations, contact number and email id as their attributes.
Dealer will have sell relationship with the products and an m to n relationship also as each product can be sold by different sellers in different/same regions and each dealer can sell many products. But each product must have at least one dealer that is selling that product.

Functional Requirements:

As we know in order to make a good fully functional e2e database application we need to design a good back-end database management system that would be managing all the data, and as we know according to the textbook definition:

A good database management system provides efficient, reliable, convenient and safe multi-user storage of and access to massive amounts of persistent data.

Therefore, our e2e database application will be having following functionalities:

- Data Integration (massive data management) :

As the name suggests our application should be able to integrate the data from various sources such as customer data, seller's data, inventory data, sales data, and many more into our database in a systematic manner.

- Data Security and Compliance (secure data):
  A security measure must always there be ready to be implemented to protect the valuable data of customers and dealers/sellers from unauthorized access and manipulation. It should also be in compliance with all relevant rules and regulations.

- Payments:
  Application should also have ability to integrate various payment gateways as well as able to handle massive transactions.

- Backup and recovery :
  A backup of the database must always be there maintained so that in case of any emergency we would be able to recover the data.

- Order and Inventory management:
  Application should be able to keep track of orders, shipping, delivery as well as stock level of products.

- Product Catalog Management:
  Application should be able to store, delete as well as modify products belonging to different categories and able to modify their attributes on time.

- A smooth login/sign in, sign up, log out/sign out must be there for new/old users in our application.

- The ability to search and navigate within our application should be as efficient and fast as it possibly could be.

- There should be various sorting/filtering methods available within our app in order to sort products based on the user needs

- Application should be able to track, update different promotions/discounts that would be available on the product by sellers/admin on timely manner. And customers should be able to browse and apply them easily.

- Data modeling, data warehousing principles should be applied to the database of our application.

- Application should be able to analyze the customer data, their sales data and predicts some trends and make decisions and marketing campaigns based on that data.