NerdLabs

POWERING YOUR TECHNOLOGY NEEDS, ONE PART AT A TIME.

DBMS | CSE 202

Team 46

Sameer Gupta (2021093) • Prateek Kumar (2021081)

Contents

Project Scope

A Comprehensive Database Management System for an E-commerce Retail Store for Computer Parts.

The system will be designed to efficiently manage information associated with an online computer retail store, including customer information, inventory, sales, financial data, and statistics.

The users can authenticate themselves on the application, after which they can search for the components they wish to purchase and filter them according to needs. The users can look for other products compatible with the currently visited products. Moreover, they can see their purchase history and provide product ratings and reviews.

The store manager(s) can add or remove products, provide offers, and specify product quantity and availability in warehouses. They can also look at store statistics which will give information on inventory, registered users, revenue, etc.

Requirements

Functional

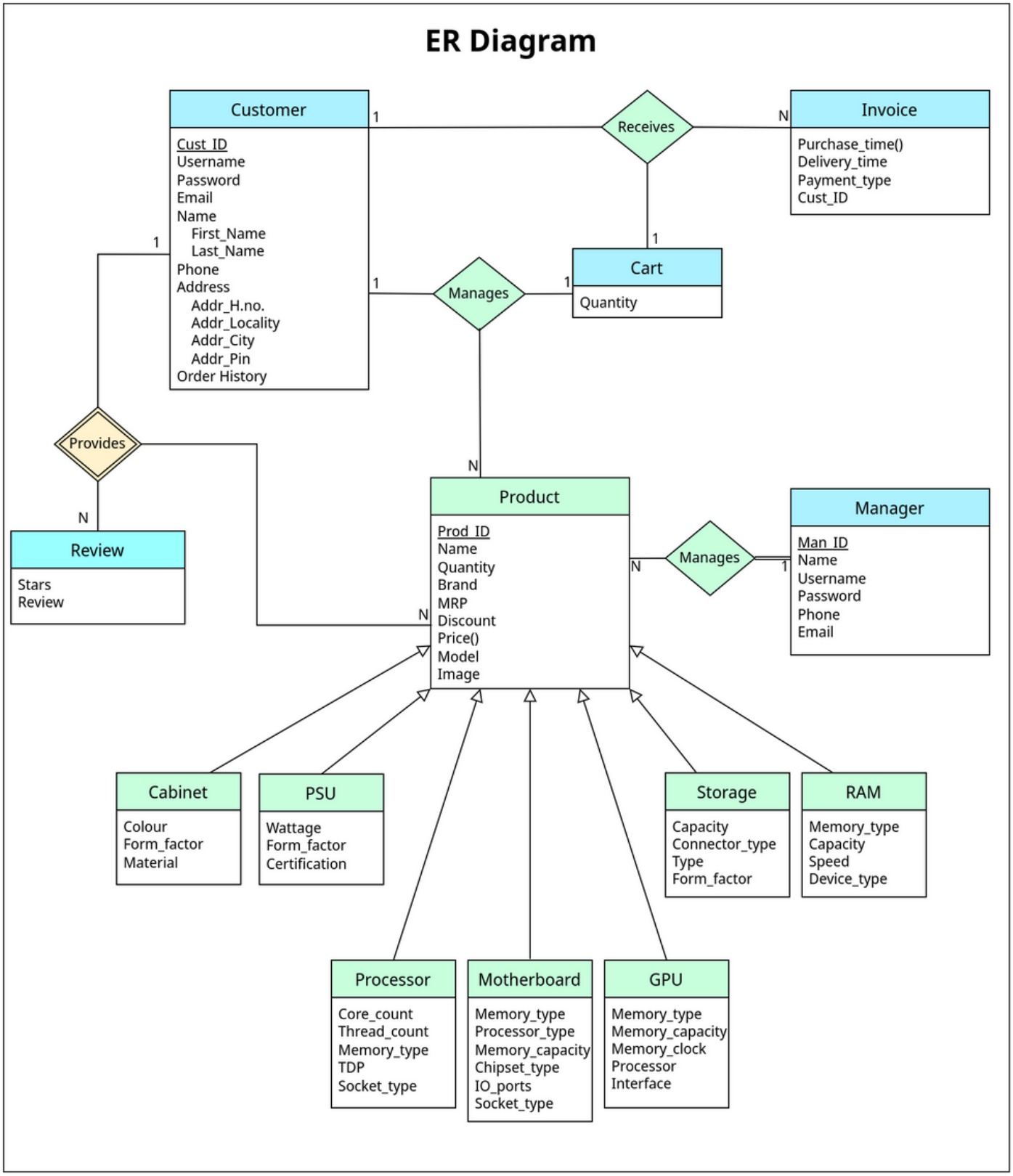
- **User registration and login**: New users should be able to register on the application and then log in with their credentials.
- **Product searching and filtering**: Products can be searched by their name, type, brand, etc., and the results can be further filtered based on prices, ratings or reviews, etc.
- **Shopping cart management**: Users can update their shopping cart and get details like amount, delivery time, etc.
- **Update inventory and catalog**: Admin can update the number of products available, add or remove products from the record, and edit product details.
- **Provide product offers**: Admin can add discounts and offers on products.
- **View store statistics**: Admin can view specific information about their store like revenue generated, product sales, inventory information, etc.

Requirements

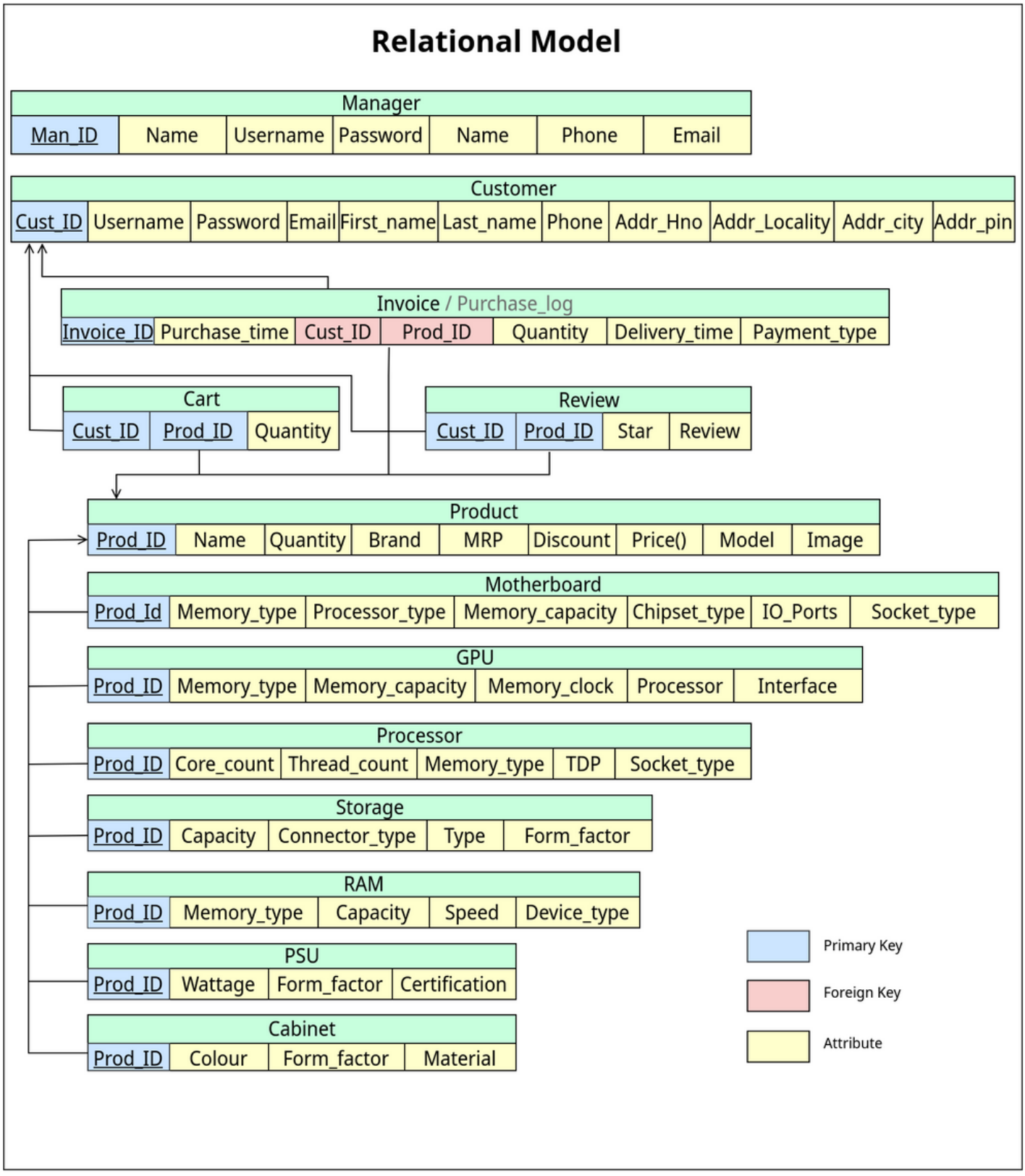
Technical

- The interface for the web application will be free-flowing and easy to grasp.
- Upon visiting the platform, the user can sign up or log in to their account.
- Once authenticated, they will be presented with the highest-rated products, or those on special discounts, in the store.
- They would also be able to choose amongst themselves or search for any particular product or category.
- Apart from that, they can filter out the results according to price ranges, ratings, brand names, etc.
- On the database side, we will keep track of registered users and the products available having various defining attributes, such as brand name, price, features, ratings, warranty, etc.
- Selected items would be placed into a cart which can be updated at any point of their online journey, after which they would be able to checkout.

ER Diagram


ER Diagram
Customer
Cust_ID
Username
Password
Email
Name
First_Name
Last_Name
Phone
Address
Addr_H.no.
Addr_Locality
Addr_City
Addr_Pin
Order History
Receives
Invoice
Purchase_time()
Delivery_time
Payment_type
Cust_ID
Cart
Quantity
Manages
Provides
Review
Stars
Review
Product
Prod_ID
Name
Quantity
Brand
MRP
Discount
Price()
Model
Image
Manages
Manager
Man_ID
Name
Username
Password
Phone
Email
Cabinet
Colour
Form_factor
Material
PSU
Wattage
Form_factor
Certification
Storage
Capacity
Connector_type
Type
Form_factor
RAM
Memory_type
Capacity
Speed
Device_type
Processor
Core_count
Thread_count
Memory_type
TDP
Socket_type
Motherboard
Memory_type
Processor_type
Memory_capacity
Chipset_type
IO_ports
Socket_type
GPU
Memory_type
Memory_capacity
Memory_clock
Processor
Interface

Relational Model



Database Schema

Manager

Man_ID	int
Name	varchar
Username	varchar
Password	varchar
Phone	int
Email	varchar

Invoice

Invoice_ID	int
Purchase_time	DATETIME
Cust_ID	int
Prod_ID	int
Quantity	int
Delivery_time	DATETIME
Payment_type	varchar

Database Schema

Customer

Cust_ID	int
Username	varchar
Password	varchar
Email	varchar
First_name	varchar
Last_name	varchar
Phone	int
Addr_Hno	varchar
Addr_Locality	varchar
Addr_city	varchar
Addr_pin	int

Cart

Cust_ID	int
Cust_ID	int
Prod_ID	int

Database Schema

Review

Cust_ID	int
Prod_ID	int
Star	int
Review	varchar

Product

Prod_ID	int
Name	varchar
Quantity	int
Brand	varchar
MRP	int
Discount	int
Price()	int
Model	int
Image	varchar

Database Schema

Cabinet

Prod_ID	int
Color	varchar
Form_Factor	varchar
Material	varchar

Motherboard

Prod_ID	int
Memory_type	varchar
Processor_type	varchar
Memory_capacity	int
Chipset_type	varchar
IO_ports	int
Socket_type	varchar

Database Schema

GPU

Prod_ID	int
Memory_type	varchar
Memory_capacity	int
Memory_clock	int
Memory_capacity	varchar
Memory_clock	varchar

Processor

Prod_ID	int
Core_count	int
Thread_count	int
Memory_type	varchar
TDP	int
Socket_type	varchar

Database Schema

Storage

Prod_ID	int
Memory_type	varchar
Memory_capacity	varchar
Memory_clock	varchar
Memory_capacity	varchar
Memory_clock	int

RAM

Prod_ID	int
Memory_type	varchar
Capacity	int
Speed	varchar
Device_type	varchar

PSU

Prod_ID	int
Memory_type	varchar
Prod_ID	int
Memory_type	varchar

Sample Queries

Complex Search Queries

1. Selecting Compatible Processors for a particular motherboard.
2. View Inventory Information
3. View Cart Information
4. Select Customers that live in a particular PIN range, and have purchased at least 1 product in the last 15 days.
5. Select top 100 products that were purchased in the last 10 days.
6. Display products that have been sold in a given time period.
7. Display products delivered in PIN code >= 110050 and had the least delivery time.
8. Display products of a specific brand having more than 4 stars.
9. Show date wise daily revenue of last month.
10. Increase prices of High end Laptop RAMs.

Adding and Updation (Constraints)

1. Adding Zero Amount of products in cart.
2. Adding a non-existing product in cart.
3. Adding new customer/ Manager with an existing username/ email/ phone.
4. Updating Customer Details.
5. Adding new attributes for product.

OLAP

1. From Customer Cart, get product count, per category.
2. Revenue per Date.
3. Purchases by customers, per Date.
4. Get inventory information per product type.

Triggers

1. Update the Quantity of a product when added to cart.
2. Update the quantity of a product when removed from cart.
3. Decrease discount of product using a predetermined formula using current quantity, when added to invoice.
4. Update the quantity of a product when it is added to invoice.

Transactions

1. Adding, updating and deleting products in store.
2. Customer orders invoice generation.
3. Registering a new customer in store.
4. Product's full details fetch operation.

User Guide

Running the application

- Inside the application folder, make sure app.py exists, then type: **flask run.**
- Application begins to run at:
  127.0.0.1 : 5000

Functionalities

* *Marked functions are for admin only.*

Home Page

Contains "featured" product cards.

Data: Contains OLAP queries.
Search: Displays product information based on product name.
Categories: Contains product information based on categories.
Cart: View customer cart.
Login: Customer login.
Register: New customer registration.
/admin: Admin Login.

Data

Contains OLAP queries and
*database search.

Query 1, 2, 3, 4: As described above.

Cart

Contains customer cart, along with deletion of product from cart.

Triggers operate when items are added to or removed from cart.

Checkout: Confirms order and checkouts to invoice.

Product Page

Contains information about products, including Reviews by Customers.

Add to Cart: Adds product to the customer's cart.

Admin Exclusive

Add Product Page

/admin/add/<category>
Admin can add product to any category.

Update Product Page

/admin/update/<prod_id>
Admin can update product information based on the product id.

Delete Product Page

/admin/delete
Admin can delete products using product ids.