DAMG6210 - Data Management and Database Design

RETAIL ANALYTICS

Updated Project_02 document

Team 05			
MEMBER	Ashvini Patidar	Ritwik Giri	Tanuja Mahajan
NU ID	002747763	002873045	002221239
NU EMAIL	Patidar.as@northeastern.edu	Giri.ri@northeastern.edu	Mahajan.t@northeastern.edu

TABLE OF CONTENT

1. OVERVIEW **2**

2. ENTITY RELATIONSHIP DIAGRAM **3**

3. BUSINESS RULE **3**

4. VIEWS **4**

5. SECURITY **7**

6. DATA FLOW DIAGRAMS **8**

7. TABLES **11**

i. User

ii. Cart

iii. Cart Items

iv. Product

v. Prices

vi. Store

vii. Store Manager

1. OVERVIEW

In an increasingly digital and price-conscious world, consumers are constantly seeking ways to optimize their spending on essential items. A comprehensive price comparison platform for groceries and electronics emerges as a valuable solution to address this need. This platform is designed to provide consumers with a user-friendly, efficient, and insightful way to make informed purchasing decisions across two crucial categories of products: groceries and electronics.

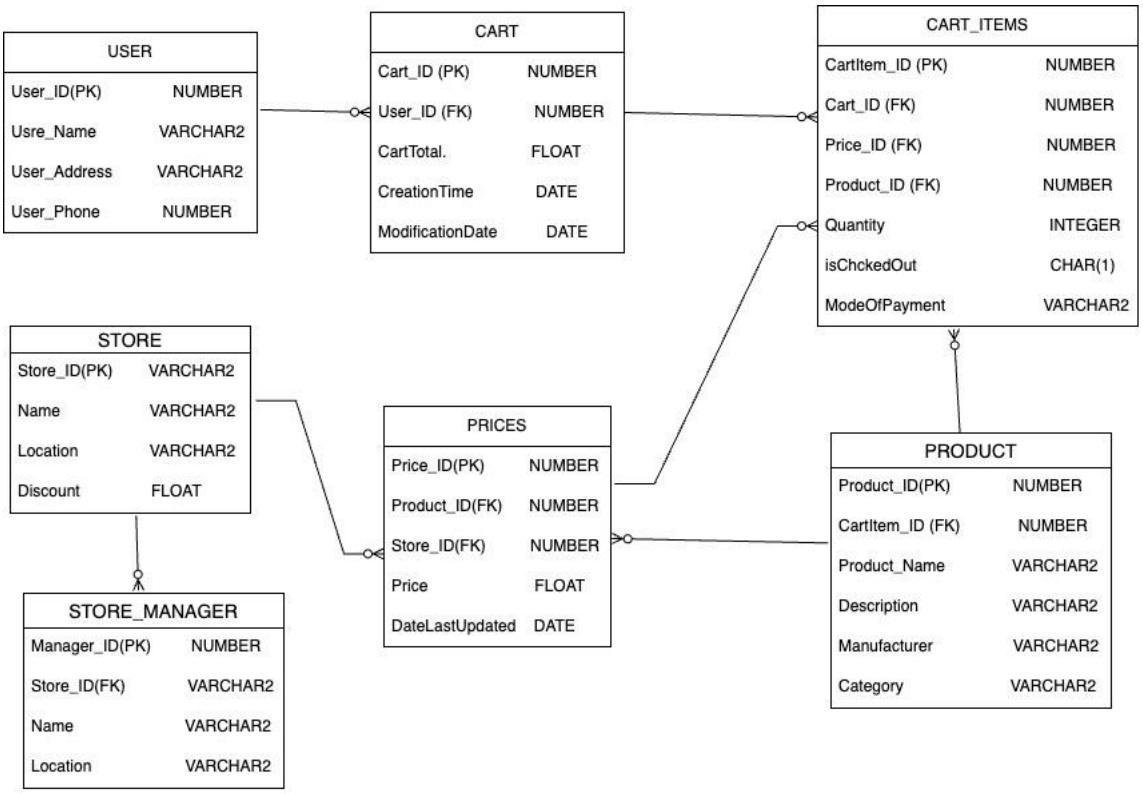
Price Comparison and User Engagement: A core feature is its ability to facilitate direct price comparisons among multiple retailers, offering transparency on discounts and promotions. Users can engage through user-generated reviews and ratings, fostering a sense of trust and community. Historical price data empowers informed purchases by displaying price trends over time.

Users derive multiple benefits from this platform. Primarily, it enables them to consistently save money by effortlessly identifying the best prices and discounts available for their desired products. This financial advantage is particularly appealing in today's cost-conscious climate. Additionally, the platform enhances shopping efficiency by centralizing all necessary information, simplifying the decision-making process and eliminating the need to navigate various websites or visit physical stores.

Moreover, users can make well-informed choices, ensuring their purchases align closely with their specific needs and preferences.Lastly, users save precious time by eliminating unnecessary trips and extensive price research, as the platform streamlines the shopping process, making it not only more efficient but also inherently more convenient.

2. ENTITY RELATIONSHIP DIAGRAM (revised)

The project has 7 main tables and relations are shown in ER diagram below :
Link: https://app.diagrams.net/#G17qoNFePKJxvi1YQfC4WaG4B99t5N5rys



3. BUSINESS RULES

Based on the Retail business requirement, we have estimated:

1. **Customer Segmentation:** Customers are categorized into two types: Consumers and Store Managers.
2. **Store Information:** Each store is represented as an entity and is associated with a Store Manager. Store Managers are responsible for overseeing the operations of their respective stores.
3. **Product Information:** Products are a central entity, each with a unique identifier. Product details such as name, category, and specifications (for electronics) are stored.
4. **Price Management:** Prices for products are managed separately as Price entities, associated with specific products. Each price entry includes the price, currency, and date of the last update. Price entries are linked to both products and stores.
5. **Cart Management:** Customers, in this case, Consumers, have a Cart entity to manage their selected items. Cart Items represent the products added to a Cart. Customers can add and remove items from their Cart as they shop.
6. **Price Comparison:** Price comparison functionality allows customers to compare product prices across different stores, making informed purchase decisions.

7. **Store Manager Responsibilities:** Store Managers oversee their store's inventory. They can also update the price of products and discount in their store to remain competitive in the market.
8. **User Interaction and Activity:** Customer interactions with the platform, such as product searches, adding items to the Cart, and making purchases.

4. VIEWS

We have envisioned 6 views to highlight the business use cases for associated tables:

Product_Details_View :

This view will be created using "Product," "Price," and "Store" tables to display detailed information about products, including their name, category, description, specifications, price, and the name of the store where they are available. Upon selecting a specific product, this view offers comprehensive insights into the chosen item, encompassing specifications, descriptions, user-generated reviews, and a comprehensive list of stores where it can be acquired.

Product_Details_View
Product_ID
Product_Name
Product_Category
Product_Description
Product_Price
Store_Name

Price_Comparison_View:

This view will be created using "Product," "Price," and "Store" tables to display a price comparison view. This view includes details about the product, its price, the store where it's available, the store's location, the currency used for pricing, and the date when the price was last updated. This view empowers **users** to conduct a direct, side-by-side examination of product prices across different retailers, facilitating informed purchasing decisions. It prominently highlights the retailer with the most advantageous offer and provides historical price data for reference.

Price_Comparison_View
Product_ID
Product_Name
Product_Category
Product_Price
Store_Name
Store_Location
Price_Last_Updated

Shopping_Cart_View:

This view will be created using "Cart, Cart_Item, Product," and "Price" tables to display the contents of a shopping cart. This view includes details such as the cart ID, cart item ID, product ID, product name, product category, cart item quantity, cart item price, cart item total, and the cart's total cost. Designed for **users**, this view offers a snapshot of the contents within their shopping cart. It showcases Cart Items, complete with product specifics, quantities, and cumulative expenses. Users can readily add, remove, or modify items within the cart.

Shopping_Cart_View
Cart_ID
CartItem_ID
Product_ID
Product_Name
Product_Category
CartItem_Quantity

CartItem_Price
CartItem_Total
Cart_Total

Store_Inventory_View:

This view will be created using "Store," "Price," and "Product" tables to display a view of store inventory. This view includes details such as store ID, store name, store location, product ID, product name, product price, and the currency used for pricing. This specialized view is accessible to Store Managers, granting them insights into the inventory levels of their respective store. It displays product stock quantities and provides the functionality to update product prices and inventory levels.

Store_Inventory_View
Store_ID
Store_Name
Store_Location
Product_ID
Product_Name
Product_Price

Price_Update_View:

This view will be created using "Product," "Price," and "Store" tables to display a view of price updates. This view includes details such as product ID, product name, product category, product price, store ID, store name, and the date when the price was last updated. **Store Managers** can employ this view to adjust the pricing of products within their store. Any changes made here are reflected in the Price entity, ensuring accurate and up-to-date price comparisons.

Price_Update_View
Product_ID
Product_Name
Product_Category
Product_Price
Store_ID
Store_Name
Price_Last_Updated

5. SECURITY

USER LEVEL ACCESS AND ROLES

Below roles will be created in database-by-database administrator:

1) **Role 1: Admin**

This is the administrator of the database who has all the controls. Admin will have access to select/ Insert/ Update/ Delete all the tables. Admin can also change/ Create passwords for Store Managers and customers.

Permissions: Admin will have access to all tables and views in the database. He can Create/ Insert/ Update/ Delete information in any of the tables in the database.

2) **ROLE 2 : Store Manager**

The store manager is responsible for overseeing a certain store. Inventory management in the store is the responsibility of the store manager. The store manager is in charge of maintaining the most recent information regarding product prices, store-specific discounts, and product discounts.

Permissions: Store manager will have Select access to all tables. Store Manager can access views : Price_Update_View, Store_Inventory_View . Store Manager can insert/ update/ delete in Store table, Product table and Price table.

3) ROLE 3 : User

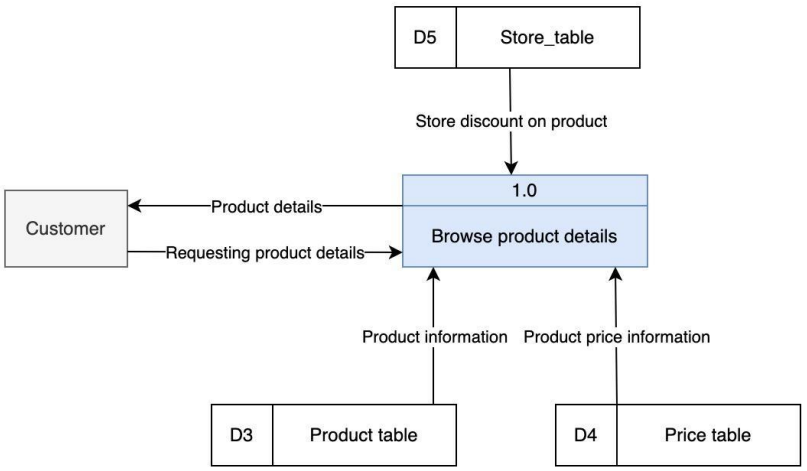
Can browse and search for products. Create and manage their shopping carts. Set up price alerts for products.

Permissions: User will select access to product and store. Can access Shopping_Cart_View, Price_Comparison_View and Product_Details_View . Users can insert/ update/ delete cart and cart Item table .

6. DATA FLOW DIAGRAMS

Customer Browsing Products

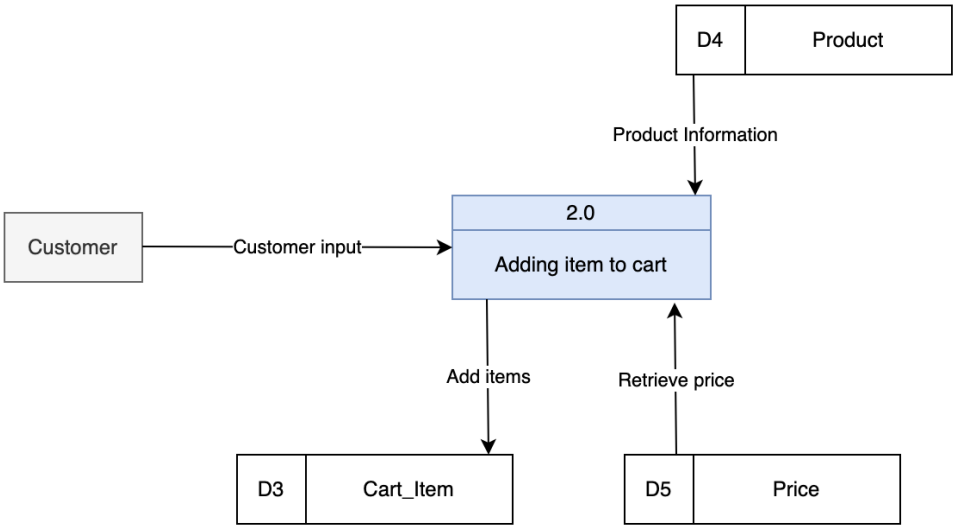
The process of product browsing encompasses interactions among customers, product data, prices, and store databases. Customers initiate requests for product details and prices, directing their queries to the respective tables containing product, store, and price information. This interaction facilitates their exploration and access to information about the products available in various stores.



Adding Items to Cart

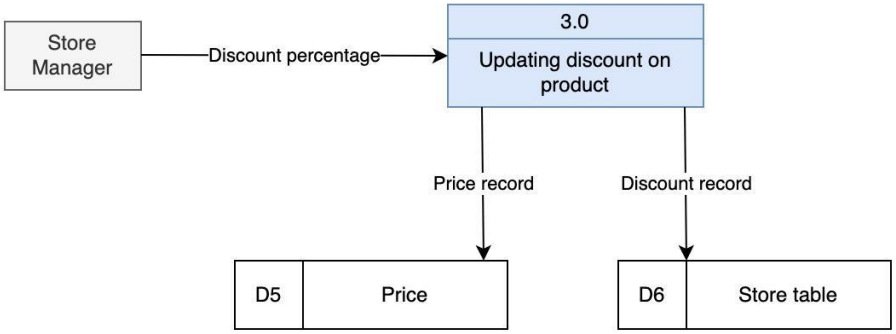
Customers, as external entities, start the process of adding items to their shopping carts. In this process, they make requests to add products, specifying the product and desired quantity. These requests trigger interactions with the Cart Item, Product, and Price data, ensuring the accurate recording of items in the cart and proper pricing. This

seamless interaction enables customers to curate their shopping experience and make informed purchasing decisions.



Store Manager Updating Store Information:

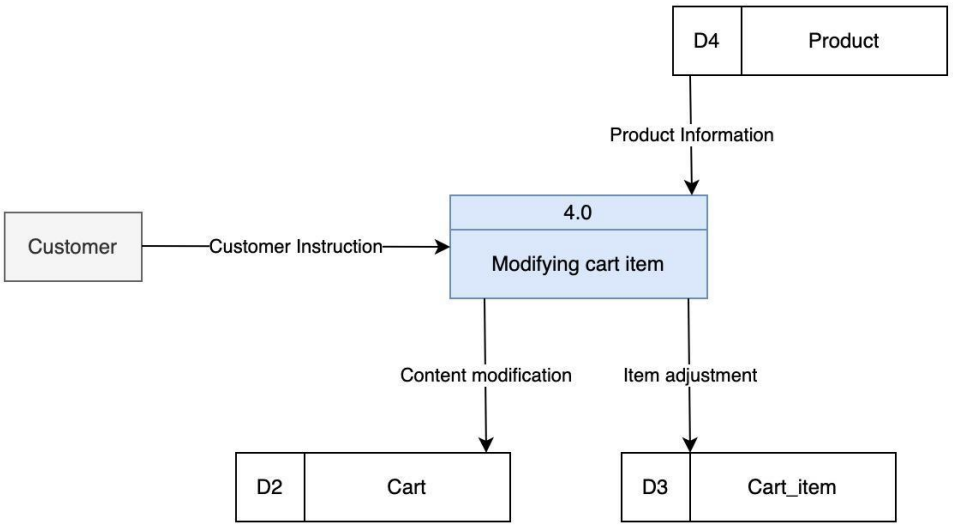
The procedure of updating store information involves interactions between Store Managers, price data, and store databases. Store Managers initiate requests to update store-specific details, particularly related to pricing and discounts. These changes are then transmitted to the "Store Data" and "Price Data". It guarantees that pricing and discount information aligns with the directives of Store Managers, ultimately promoting effective store management and ensuring customers receive precise pricing and discount details.



Modifying Cart Items

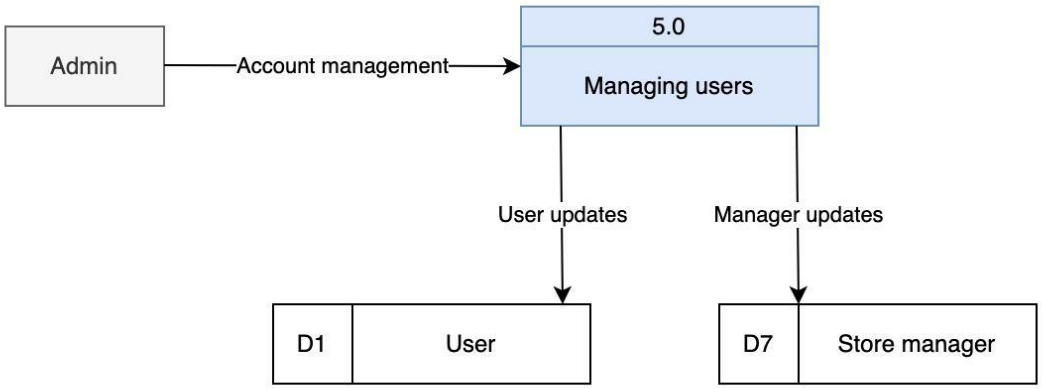
Customers initiate the process of modifying the contents of their shopping cart, allowing them to add, remove, or change item quantities. These changes may involve adding new items, removing existing ones, or altering item quantities. These customer requests are

directed to the "Manage Cart". The process involves interactions with Cart, Cart Item, and Product data to update the cart's contents, ensuring accurate representation of item quantities, and validating changes for product availability, enhancing customers' shopping flexibility.



Admin Managing Users:

Admins have authority to oversee user and Store Manager accounts. This process includes creating, updating, or deleting these accounts and managing their permissions. Admin efficiently handle administrative tasks, ensuring user and Store Manager account management is accurate and up-to-date, and their permissions are properly maintained. This data flow is essential for effective administrative control and interaction with associated data.



7. TABLES

Based on the Price comparison requirement, we have estimated a database schema of 7 tables with the addition of tables and views in the future as the scope of the project improves.

i. User:

The "User" table shows the users and their details who utilize the price comparison platform to get real-time price information facilitating informed purchasing decisions.

This table manages user information, utilizing "User_ID" as the primary key for unique identification, "User_Name" to store user names, "User_Address" to record user addresses, and "User_Phone" to store user phone numbers. This structure enables effective management and retrieval of user details within the database.

USER			
Key/Constraint	Datatype	Attribute	Description
Primary Key (PK)	Number	User_ID	A unique identifier for each user
Not Null	Varchar2	User_Name	The name of the user
Not Null	Varchar2	User_Address	The address of the user
Not Null	Number	User_Phone	The phone number of the user

ii. Cart:

The "CART" table shows information related to the shopping cart of the user.

It stores shopping cart information, utilizing "Cart_ID" as the primary key for distinct identification, "User_ID" as a foreign key referencing the associated user, total costs (CartTotal), and timestamps for cart creation and modification.

CART			
key/constraint	datatype	Attribute	Description

Primary Key(PK)	Number	Cart_ID	A unique identifier for each shopping cart
Foreign Key	Number	User_ID	A reference to the Customer who owns the cart
Not Null	Float	CartTotal	The Total of the Cart items
Not Null	Date	CreationTime	The creation time of the cart
	Date	ModificationTime	The modification time of the cart

iii. Cart Items:

The 'CART_ITEMS' table contains data about items within shopping carts, including unique CartItem_IDs, associated Cart_IDs, quantities, prices, checkout statuses, payment methods, and references to the corresponding Product_IDs and Price_IDs.

CART_ITEMS			
key/constraint	datatype	Attribute	Description
Primary Key	Number	CartItem_ID	A unique identifier for each item in the cart
Foreign Key	Number	Cart_ID	A reference to the Cart to which the item belongs
Not Null	Integer	Quantity	The quantity of the product in the cart
Foreign Key	Float	Price_ID	reference to the price associated with the product in the cart.
Not Null	Char (1)	isCheckedOut	The cart ready to be checked out or not
Not Null	Varchar2	ModeOfPayment	The method of payment for the item, e.g., credit card, cash, etc.
Foreign Key	Number	Product_ID	A reference to the Product ID of the product

iv. **Product:**

The "PRODUCT" table shows information related to various products in the cart.

It stores product information, utilizing "Product_ID" as the primary key, CartItem_ID as the Foreign key capturing "Product_Name", "Description", "Manufacturer" and Category as mandatory fields of type "varchar2," enabling effective product management and retrieval for business operations and customer interactions.

PRODUCT			
key/constraint	datatype	Attribute	Description
Primary Key	Number	Product_ID	A unique identifier for each product
Not Null	varchar2	Product_Name	The name of the product
Not Null	varchar2	Description	A brief description of the product
Not Null	varchar2	Manufacturer	The manufacturer of the product
Foreign key	Number	CartItem_ID	A reference to the CartItem_ID of the table CART_ITEMS
Not Null	Varchar2	Category	The category of the product

v. **Prices:**

The "PRICE" table manages pricing information across stores, using "Price_ID" as the primary key, "Product_ID" and "Store_ID" as foreign keys, recording product prices as "Float," and tracking price update dates for tracking the historical price changes, facilitating effective price management and market analysis.

PRICES			
key/constraint	datatype	Attribute	Description

Primary Key	Number	Price_ID	A unique identifier for each price entry
Foreign Key	Number	Product_ID	A reference to the Product associated with the price
Foreign Key	Varchar2	Store_ID	A reference to the Store where the product is available
Not Null	Float	Price	The price of the product
Not Null	Date	DateLastUpdated	The date when the price was last updated

vi. Store:

The "STORE" table shows information related to different stores.

It manages store information, with "Store_ID" as the primary key for distinct identification, capturing "Name" and "Location" as mandatory fields for easy retrieval, while "Discount" records numerical discount values, facilitating effective store management and customer interaction.

STORE			
key/constraint	datatype	Attribute	Description
Primary key	Varchar2	Store_ID	A unique identifier for each store
Not Null	Varchar2	Name	The name of the store
Not Null	Varchar2	Location	The physical location of the store
	Float	Discount	The discount offered by the store on the product

vii. Store_Manager:

The "STORE MANAGER" table tracks unique identifiers and details, including names and locations, for managers overseeing specific stores, with "Manager_ID" as the primary key and "Store_ID" as the foreign key referencing the associated store.

STORE MANAGER			
key/constraint	datatype	Attribute	Description
Primary Key	Number	Manager_ID	A unique identifier for each store manager
Foreign Key	Varchar2	Store_ID	A reference to the Store managed by the manager
Not Null	varchar2	Name	The name of the store manager
Not Null	varchar2	Location	The location of the store manager