**Midterm Sprint:**

**Database Project:**

**Business Scenario:**

Newfie Nook, an ecommerce platform, prides itself on championing the rich culture and craftsmanship of Newfoundland and Labrador. By offering a carefully curated selection of products, ranging from iconic local treats to unique artisanal creations, Newfie Nook celebrates the essence of the region.

Vendors, representing Newfoundlanders and Labradorians, have control over the products they list and receive timely notifications for sales. The platform fosters a seamless shopping experience for customers, who are required to create accounts to make purchases.

Each product is tied to a specific vendor, ensuring a direct connection between creators and consumers.

Customers can effortlessly add products to their carts and proceed to checkout, while vendors have the capability to manage their listings, update inventory, and monitor sales performance through dedicated accounts.

With a commitment to secure payment processing and prompt order fulfillment, Newfie Nook promotes local craftsmanship, supports NL artisans, and provides exceptional customer support services to enrich the overall shopping journey for users.

**Business Rules:**

**Product Rules:**

* Each product must have a unique Product ID.
* Product Name, Description, Price, and Category ID are mandatory fields for product listing.
* The Image field should contain a visual representation of the product.
* A product can belong to multiple categories, and each category can have multiple products.

**Product Category Rules:**

* Product ID and Category ID together must be unique in the join table – *Composite Primary Key*.
* A product can belong to multiple categories, and a category can encompass multiple products.

**Person Rules:**

* Each person must have a unique Person ID.
* Person Name, Email, Phone Number, and Person type (customer or vendor) are mandatory fields.
* Address\_ID links the Address table to the person it belongs to. Each Person can have multiple addresses (billing and mailing, for example).

**Customer Rules:**

* Each customer must have a unique Customer ID.
* Customer ID and Person ID together must be unique in the Customer entity.
* A customer is associated with a person, providing general information like name and contact details.
* Customer also comes with account information such as username, password and payment information.

**Address Rules:**

* Each address must have a unique Address ID that is linked to a unique Person\_ID.
* Street Address, City, Province, Postal Code, and Address Type are mandatory fields.

**Vendor Rules:**

* Each vendor must have a unique Vendor ID.
* Vendor ID and Person ID together must be unique in the Vendor entity.
* Vendor Name (business name), username, password, and payment information are mandatory fields.

**Order Invoice Rules:**

* Each order invoice must have a unique Invoice ID.
* Invoice ID must be unique in the Order Invoice entity, and links to foreign keys Customer\_ID, Vendor\_ID, and Product\_ID.
* Each invoice can have one customer, but multiple vendors and products.
* Date and Shipping Information are mandatory fields for order tracking.

**Join Table Rules:**

* For each join table, the combination of the respective IDs (e.g., Person\_ID and Address\_ID) must be unique.
* Join tables facilitate many-to-many relationships between entities.
* These rules help maintain data integrity, ensure uniqueness, and guide the relationships between different entities in the Newfie Nook database. They are essential for the proper functioning and consistency of the ecommerce platform.

**Entities & Attributes for Database Creation:**

* **Person:**
  + *Attributes:* Person\_ID (PK), Person\_Name, Email, Phone\_Number, Person\_Type (Vendor or Customer), Address\_ID (FK)
  + *Description:* Displays key information for individuals (Vendor and Customer) such as their name, email, phone number, and linked to their address. This is further divided into Customer and Vendor entities.
* **Address:**
  + *Attributes:* Address\_ID (PK), Street\_Address, City, Province, Postal\_Code, Address\_Type, Person\_ID (FK)
  + *Description:* Stores information related to the addresses of individuals on the Newfie Nook platform. Each address is uniquely identified by an Address ID. The attributes include the street address, city, province, postal code, and the type of address, indicating whether it is for mailing or billing purposes. This entity is crucial for managing and organizing the location details associated with users on the platform.
* **Vendors:**
  + *Attributes:* Vendor\_ID (PK), Person\_ID (FK), Vendor\_Name, Username, Password, Payment\_Info
  + *Description:* Represents the artisans and businesses from Newfoundland and Labrador who supply products to Newfie Nook. Stores login credentials for vendors, allowing them to manage their product listings, update inventory, and track sales performance.
* **Customers**:
  + *Attributes*: Customer\_ID (PK), Person\_ID (FK), Username, Password, Payment\_Info
  + *Description*: Represents customer accounts that provide login credentials (username and password, type of payment information, and links customers to their accounts, enabling them to access personalized features and track their order history.
* **Products:**
  + *Attributes*: Product\_ID (PK), Product\_Name, Description, Price, Image, Category\_ID (FK), Vendor\_ID (FK)
  + *Description*: Represents the products available for purchase on the platform. Includes both NL treats (e.g., Purity Products, Pineapple Crush, Caramel Logs) and artisanal creations (e.g., handcrafted sea glass jewelry, traditional knitwear).
* **Product\_Category:**
  + *Attributes:* Category\_ID (PK), Category\_Name
  + *Description:* Shows the products category and helps to organize the products based on what the customer is looking for.
* **Order\_Invoice:**
  + *Attributes:* Invoice\_ID (PK), Date, Shipping\_Information, Customer\_ID (FK), Vendor\_ID (FK), Product\_ID (FK)
  + *Description:* Entity represents the records of invoices generated for customer orders on the Newfie Nook platform. This entity is crucial for tracking and managing order-related financial transactions, providing a record of the purchase details and associated shipping information. It is linked to the vendor who created the products, product ID to show what product is being purchased, and customer ID to keep track of what they have purchased.
* **Join Tables:**
  + In the Newfie Nook database, several join tables are utilized to manage many-to-many relationships between various entities. These tables act as bridges, facilitating connections between different aspects of the platform. These join tables are integral components of the database design, providing the flexibility to establish and manage complex relationships between entities within the Newfie Nook ecommerce platform.
  + *Join Tables:*
    - Join\_Vendor\_Address
      * Links the address to the associated vendor, can have multiple addresses.
    - Join\_Customer\_Address
      * Links the address to the associated customer, can have multiple addresses.
    - Join\_Vendor\_Products
      * Links the vendor to the products they are selling.
    - Join\_Vendor\_Order
      * Links the vendor to the Order Invoice and their product that is being purchased.
    - Join\_Customer\_Products
      * Links the customer to the products they are purchasing.
    - Join\_Customer\_Order
      * Links the customer to the order invoice that displays the products, vendor, data, total purchase, product price, and more.
    - Join\_Product\_Order
      * Links the products being purchased to the order invoice and provides the necessary information.

**Relationships of ERD:**

To comprehensively assess the database design for Newfie Nook, an ecommerce platform promoting Newfoundland and Labrador's cultural products, a detailed examination of its structure and relationships is essential. The Entity-Relationship Diagram (ERD) provides a visual representation of the database schema, but accompanying documentation enriches the understanding. The database design for Newfie Nook adheres to the principles of normalization, with a particular focus on achieving Third Normal Form (3NF) to minimize data redundancy and maintain data integrity.

The ERD reveals primary entities such as Person, Address, Vendors, Customers, Products, Product\_Category, and Order\_Invoice, each characterized by distinct attributes and relationships.

Notably, composite keys and foreign keys facilitate relational integrity, ensuring data consistency and coherence. The documentation elaborates on the purpose of each entity, elucidating how they contribute to the platform's functionality. For instance, the Person entity encompasses both vendors and customers, underscoring their vital roles within the ecosystem.

Address details are crucial for organizing user location data, while Vendors and Customers entities encapsulate user credentials and payment information, enabling personalized experiences.

Products and Product\_Category entities categorize offerings, while Order\_Invoice records transactional data, facilitating order tracking and financial management. Additionally, join tables like Join\_Vendor\_Address and Join\_Customer\_Address bridge many-to-many relationships, enhancing database flexibility.

By combining the ERD with comprehensive documentation, database administrators can conduct thorough assessments, ensuring data quality, referential integrity, infrastructure adequacy, and optimized performance align with Newfie Nook's objectives of celebrating local craftsmanship and enhancing customer experiences.

**Legend for Newfie Nook Database ERD:**

* **Primary Keys:** The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).
* **Foreign Keys:** The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.
* **Composite Primary Keys:** A composite primary key is a primary key formed by combining one or more foreign keys from the table. These are used in join/association tables.

**A yellow key with a hole in the middle

Description automatically generated**

**Primary Keys (Gold Key) :**

* Person – Person\_ID (PK)
* Vendor – Vendor\_ID (PK)
* Customer – Customer\_ID (PK)
* Address – Address\_ID (PK)
* Order\_Invoice – Invoice\_ID (PK)
* Products – Product\_ID (PK)
* Product\_Category – Category\_ID (PK)

**A grey and white key

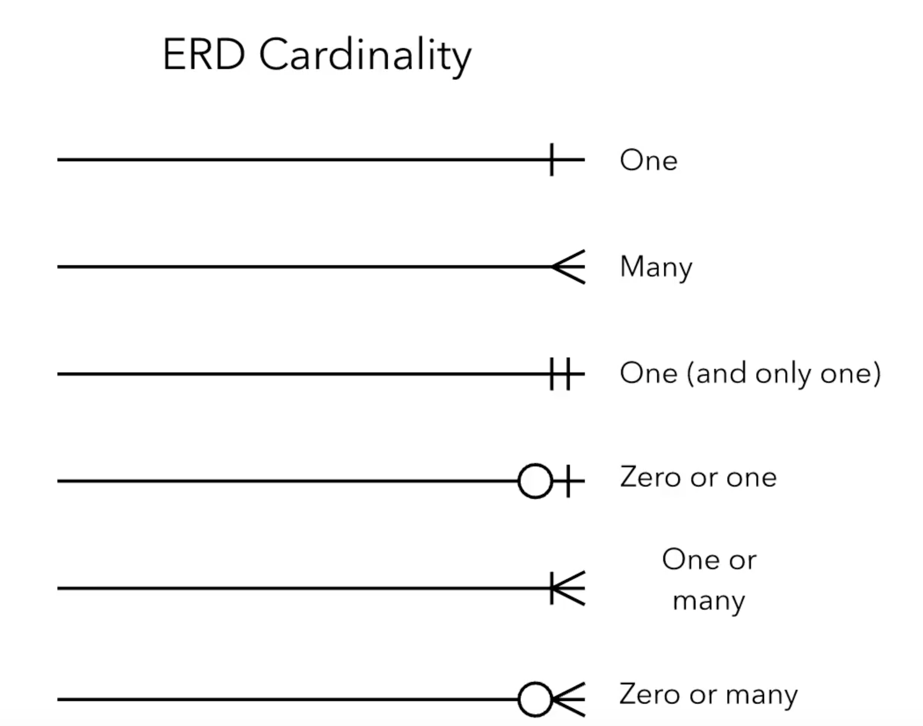
Description automatically generated**

**Foreign Keys (Silver Keys) :**

* Person – Address\_ID (FK)
* Vendor – Person\_ID (FK)
* Customer – Person\_ID (FK)
* Address – Person\_ID (FK)
* Order\_Invoice – Customer\_ID (FK), Vendor\_ID (FK), Product\_ID (FK)
* Products – Category\_ID (FK)
* Product\_Category – None

**Composite Primary Keys:**

* **Join\_Category\_Product**
  + Category\_ID, Product\_ID
* **Join\_Customer\_Address**
  + Customer\_ID, Address\_ID
* **Join\_Customer\_Order**
  + Customer\_ID, Invoice\_ID
* **Join\_Product\_Order**
  + Product\_ID, Invoice\_ID
* **Join\_Vendor\_Address**
  + Vendor\_ID, Address\_ID
* **Join\_Vendor\_Order**
  + Vendor\_ID, Invoice\_ID
* **Join\_Vendor\_Products**
  + Vendor\_ID, Product\_ID

**Cardinalities:**

**About Mock Data:**

To populate the Newfie Nook database with realistic and diverse data, the mock data generation tool Mockaroo.com was employed. The "Person" table, representing individuals associated with the business, was populated with names, emails, phone numbers, and designated roles as either customers or vendors. The "Address" table received synthetic street addresses, cities, provinces, and postal codes, with distinct types like mailing or billing.

For customer-specific data, the "Customer" table was filled with unique customer IDs, linking back to corresponding individuals in the "Person" table. Similarly, the "Vendor" table included vendor-specific details such as names and contact persons.

The "Product" table received information about various products, including names, descriptions, prices, and images from a custom JSON file. Categories for these products were established in the "Product\_Category" table, and shows the relationship between products and categories.

Furthermore, the mock data generation extended to other crucial tables, like "Customer\_Account," "Vendor\_Account," "Order\_Invoice," and related join tables. This diverse set of mock data mirrors a realistic scenario, enabling comprehensive testing and analysis of the Newfie Nook database structure and functionality. The use of Mockaroo.com ensures that the database is populated with relevant, varied, and representative information for a robust and comprehensive testing environment.

**Indexes:**

Indexes play a crucial role in database optimization, enhancing the speed and efficiency of data retrieval operations. In the context of our database design, three key types of indexes have been strategically implemented to address specific optimization needs: Foreign Key Indexing, Date-based Indexing, and Product-related Indexing. Each type of index is tailored to improve the performance of particular queries and operations related to individuals, order dates, and product-related information.

**Foreign Key Indexing:**

These indexes are created on fields serving as foreign keys. They enhance the efficiency of JOIN operations and data retrieval related to specific individuals, customers, vendors, and associated entities.

**Date-based Index:**

This index is tailored for efficient searching based on order dates. It significantly improves the performance of queries that involve filtering or sorting order records by date.

**Product-Related Indexing:**

These indexes are designed to optimize the search and filtering of products. The former enhances queries related to a person's products, while the latter improves the performance of queries involving product categorization.

**Explanation of SQL Queries**

**Example 1:** **Retrieve data about orders and their products for customers**

This query retrieves information about orders, including invoice details and product information for each order.

**Example 2:** **Using GROUP BY and aggregate functions**

This query counts the total number of products for each person, utilizing the GROUP BY clause and COUNT() function.

**Example 3: Counting the number of customers for each vendor**

This query calculates the total number of products associated with each vendor. The result shows the count of products for each vendor.

**Example 4:** **Retrieving customers and their orders**

This query retrieves customer information along with the count of orders for each customer, using LEFT JOIN to include customers with no orders.

**Example 5:** **Using WHERE clause**

This query selects all records from the Person table where the Person\_Type is 'customer'.

**Example 6:** **Using ORDER BY**

This query sorting results by Price in descending and ascending order.

**Example 7:** **Retrieving Product Information with Corresponding Person ID from Order Invoices**

This query retrieves product information along with the corresponding person ID from order invoices, using a JOIN operation.

**Example 8: Displays the count of products and their names in each category**

This query provides a comprehensive view of product counts and their corresponding names within each category. Results show as an array of product names for better readability.

**Example 9: Find vendors whose average product price exceeds 2**

This query identifies vendors whose average product price exceeds 2, using JOIN operations and the HAVING clause.

**Example 10: Using WHERE**

Retrieve data about orders and their products for customers made after a specific date and using specific method (shipping information).

**Example 11:** **Obtain information about products and their respective invoice numbers for each individual**

This query retrieves product information along with the corresponding invoice numbers for each individual, using JOIN operations.

Top of Form

Bottom of Form