**TABLE OF CONTENTS**

**Introduction………………………………………………………………………………………………2**

**What is considered a good Database Design?…………………………………………………….2**

**Common Terminologies in Database Design………………………………………………………3**

**Determining the Objective of the Database ………………………………………………………..4**

**Data collection analysis and organizing the necessary information………………………….4**

**Dividing the information into Tables …………………………………………………….………….5**

**Defining the Preliminary Table List (Subjects)…………………………………………………….6**

**Reviewing and refining the Design …………………………………………………………………..7**

**Defining Business Rules for the Database…………………………………………………………7**

**Conclusion……………………………………………………………………………………………….9**

**Introduction**

FreshCo is a modern, intuitive grocery delivery service that prioritizes freshness, convenience, and sustainability. We partner with local farmers and suppliers to bring you the best products while supporting the community. This repository houses the code, documentation, and resources to enhance and expand FreshCo's capabilities. To support its operations, FreshCo requires a robust database design that ensures data integrity, performance, scalability, and security. This case study explores the steps FreshCo took to achieve an optimal database design.

**What is Considered a Good Database Design?**

A good database design is characterized by:

* **Data Integrity and Consistency**: Ensures accurate and reliable data through constraints and normalization.
* **Performance Optimization**: Provides fast query responses and efficient data retrieval.
* **Scalability**: Can grow with increasing data volumes and user demands.
* **Security**: Protects sensitive information and complies with data protection regulations.
* **Usability and Maintainability**: Easy to understand, update, and manage.

**Common Terminologies in Database Design**

* Primary Key: A unique identifier for a record in a table.
* Foreign Key: A field in one table that uniquely identifies a row in another table.
* Normalization: The process of organizing data to reduce redundancy and improve data integrity.
* Index: A database object that improves the speed of data retrieval.
* Schema: The structure of a database, including tables, fields, and relationships.
* Table: A table is the chief structure of a database, representing specific subjects, objects, or events. They help organize data in the database.
* Field: A field in a database is a single piece of data or a specific attribute within a table. It stores actual data.
* Record: A record (row) in a database is a complete set of related fields representing one item or entity.
* Keys: Keys in a database are attributes or combinations of attributes used to identify and establish relationships between tables uniquely.
* Data: The value you store in a table. Data itself is meaningless.
* Information: This is the organized and processed data stored in tables. Information is meaningful.
* Primary Key: A primary key in a database is a unique identifier for each record in a table.
* Foreign Key: A foreign key in a database is a field that links to the primary key of another table to establish a relationship between the two tables.
* Views: Views in a database are virtual tables created by querying and combining data from other tables.
* Relationships: Relationships in a database connect tables through keys to organize and manage related data.
* Data Integrity: Data integrity in a database refers to the accuracy and consistency of stored data.
* Nulls: Nulls in a database represent missing or undefined values in a field.
* ERD (Entity Relationship Diagram): An ERD (Entity Relationship Diagram) in a database is a visual representation of the relationships between entities (tables) in a database.
* Structured Query Language (SQL): Structured Query Language (SQL) is a programming language used to manage and manipulate relational databases.

**Determining the Objective of the Database (Interviews)**

To determine the objectives of the database, FreshCo conducted interviews with key stakeholders, including store managers, supply chain coordinators, and IT staff. The main objectives identified were:

* Efficiently manage product inventory.
* Track customer orders and deliveries.
* Provide real-time sales reports.
* Ensure data security and regulatory compliance.

**Data Collection Analysis and Organizing the Necessary Information**

FreshCo gathered data from various sources, including existing spreadsheets, sales reports, and customer feedback forms. The collected data was analyzed to identify essential information, such as:

* Product details (name, category, price, stock levels).
* Customer information (name, contact details, order history).
* Order details (order ID, products ordered, quantities, total cost).
* Delivery information (delivery ID, order ID, status, estimated delivery time).

**Dividing the Information into Tables — Defining the Preliminary Table List (Subjects)**

Dividing information into tables is a crucial step in database design. It ensures that data is organized logically and can be accessed efficiently. Here's how FreshCo approached this process:

### **Identifying Major Entities**

The first step was to identify the major entities (subjects) that the database needed to store information about. These entities were derived from the data analysis phase. FreshCo identified the following major entities:

* **Customers**: Individuals who shop at FreshCo.
* **Products**: Items sold by FreshCo.
* **Orders**: Transactions made by customers.
* **Order Details**: Specifics of the products included in each order.
* **Deliveries**: Information about the delivery of orders.

### **Creating Tables for Each Entity**

Each identified entity was translated into a table. Here are the details for each table:

#### **Customers Table**

The **Customers** table stores all relevant information about FreshCo's customers.

* **Fields**:
* CustomerID (Primary Key): A unique identifier for each customer.
* FirstName: The customer's first name.
* LastName: The customer's last name.
* Email: The customer's email address, which is unique for each customer.
* Phone Number: The customer's phone number.
* Address: The customer's address.

#### **Products Table**

The **Products** table contains information about the items available for sale at FreshCo.

* **Fields**:
  + ProductID (Primary Key): A unique identifier for each product.
  + ProductName: The name of the product.
  + Category: The category to which the product belongs (e.g., dairy, bakery, produce).
  + Price: The price of the product.
  + Stock: The quantity of the product available in stock.

#### **Orders Table**

The **Orders** table records transactions made by customers.

* **Fields**:
  + OrderID (Primary Key): A unique identifier for each order.
  + CustomerID (Foreign Key): Links to the CustomerID in the **Customers** table to identify who placed the order.
  + OrderDate: The date and time when the order was placed.
  + Total Amount: The total amount of the order.

#### **Order Details Table**

The **Order Details** table provides details about the products included in each order.

* **Fields**:
  + OrderDetailID (Primary Key): A unique identifier for each order detail record.
  + OrderID (Foreign Key): Links to the OrderID in the **Orders** table to identify the order.
  + ProductID (Foreign Key): Links to the ProductID in the **Products** table to identify the product.
  + Quantity: The quantity of the product ordered.
  + Price: The price of the product at the time of the order.

#### **Deliveries Table**

The **Deliveries** table tracks the status and details of order deliveries.

* **Fields**:
* Delivery ID (Primary Key): A unique identifier for each delivery record.
* OrderID (Foreign Key): Links to the OrderID in the **Orders** table to identify the order being delivered.
* Delivery Status: The status of the delivery (e.g., pending, shipped, delivered).
* Estimated Delivery Time: The estimated time of delivery.

**Reviewing and Refining the Design**

After establishing the initial design, FreshCo reviewed and refined it:

* **Normalization**: Ensured the database was normalized to the Third Normal Form (3NF) to eliminate redundancy and improve data integrity.
* **Indexing**: Added indexes to frequently queried fields, such as Email in the Customers table and OrderDate in the Orders table.
* **Validation**: Conducted extensive testing to ensure the database met performance and integrity requirements.

**Defining Business Rules for the Database**

FreshCo defined several business rules to ensure the database operated effectively:

* **Unique Email Constraint**: To ensure each customer can be uniquely identified and contacted, FreshCo enforces a unique email constraint in the Customers table.
* **Stock Management**: FreshCo implemented triggers to automatically update product stock levels when orders are placed or canceled. This helps in maintaining accurate inventory levels and avoiding over-selling products.
* **Order Validation**: Orders cannot be placed if the requested products are out of stock. This ensures customer satisfaction and prevents overselling.
* **Data Security**: FreshCo takes data security seriously, especially for sensitive customer information. Encryption is used for fields such as customer phone numbers and addresses to protect them from unauthorized access.

**Conclusion**

By following best practices in database design, FreshCo successfully created a robust, scalable, and secure database that met its operational needs. The well-structured schema ensured data integrity, optimized performance, and supported future growth.

Key takeaways include the importance of maintaining data integrity, organizing data effectively, and establishing clear relationships between tables. By following these guidelines, you can create a database that not only stores information efficiently but also enhances business operations and decision-making processes.

This case study highlights the importance of thoughtful database design in achieving business objectives and enhancing overall efficiency.

[(25) Denso DANIEL | LinkedIn](https://www.linkedin.com/in/denso-daniel-ba2145137/)