|  |
| --- |
| Description of the Database  **Warehouse of the food products (confectionery)** |
|  |

Contents

[1 Business Description 3](#_Toc3394)

[1.1 Business background 3](#_Toc14840)

[1.2 Problems. Current Situation 3](#_Toc2565)

[1.3 the Benefits of implementing a database. Project Vision 3](#_Toc2301)

[2 OLTP Model description 3](#_Toc10668)

[2.1 Definitions & Acronyms 3](#_Toc28171)

[2.2 Logical Scheme 4](#_Toc28693)

[2.3 Objects 5](#_Toc30004)

[2.4 SCRIPT run instructions 8](#_Toc6672)

[3 OLAP Model description 9](#_Toc19711)

[3.1 Definitions and Objects 9](#_Toc21787)

[3.2 Logical Scheme 12](#_Toc2496)

[3.3 sCRIPT run instructions 13](#_Toc13786)

# 

# Business Description

## Business background

The data model is crafted to support diverse supply chain operations encompassing product management, supplier interactions, shipment logistics, and inventory arrangements. This model is pivotal in streamlining processes such as order fulfillment, supplier management, inventory tracking, and logistics coordination across various warehouses and carriers.

## Problems. Current Situation

The present data model faces challenges related to complexity, scalability, data redundancy and integrity, flexibility, and security which can impact its efficiency and reliability. Addressing these issues is critical for maintaining a robust and efficient operation.

## the Benefits of implementing a database. Project Vision

Implementing the database will centralize and streamline inventory, shipment, and supplier management, enhancing informed decision-making, operational efficiency, accountability, scalability, and security. This strategic move aims to support sustainable business growth and improved supply chain management.

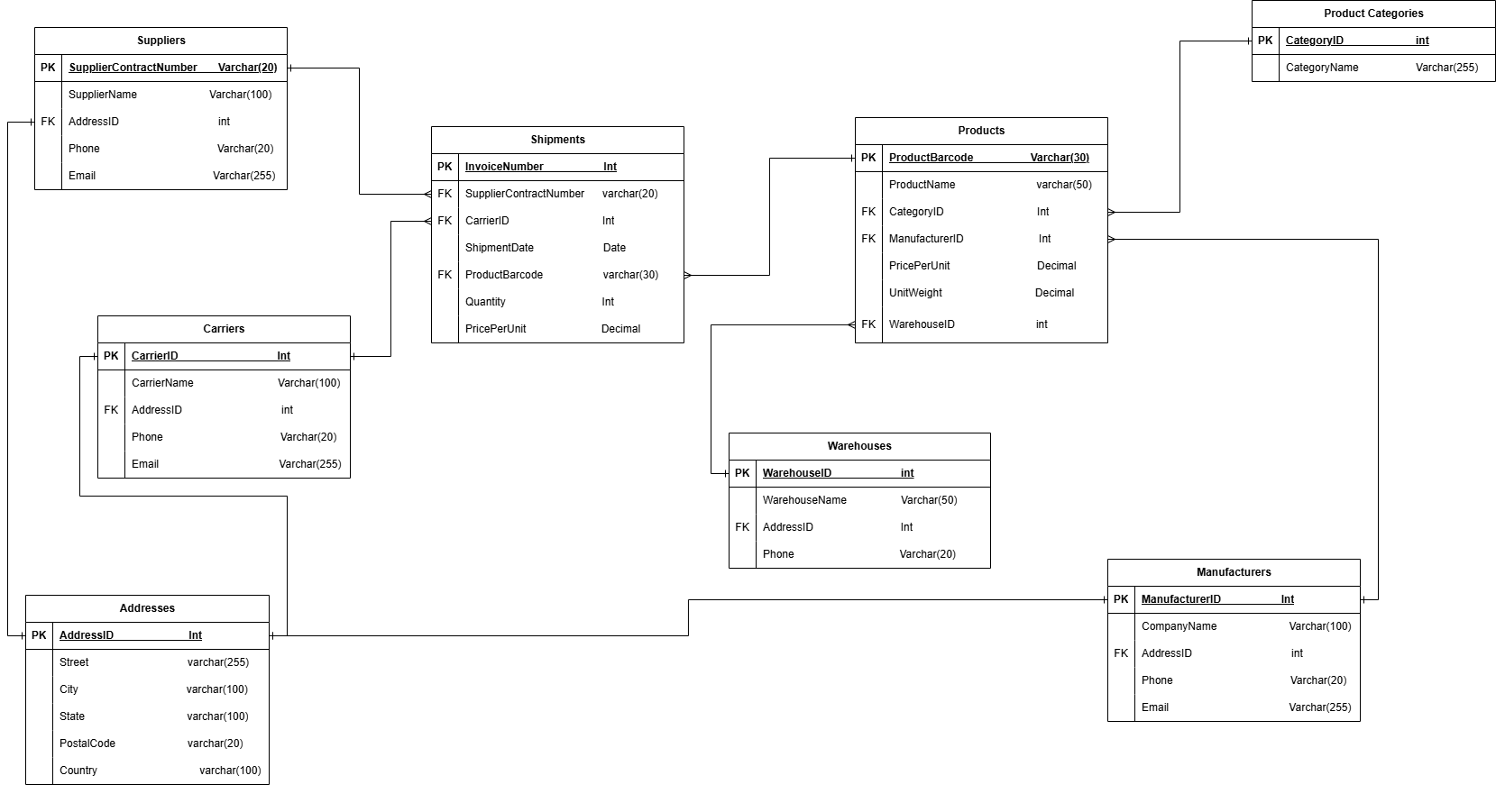
# OLTP Model description

## Definitions & Acronyms

The definitions and acronyms for the proposed data model:

1. Product (PR): An item that is manufactured or refined for sale.
2. Product Barcode (PBC): A unique identifier for a product, typically represented by a series of lines and spaces readable by machines.
3. Supplier (SU): A business or individual providing goods or services to another entity, typically involved in manufacturing or wholesale.
4. Supplier Contract Number (SCN): A unique identifier for a contract between a supplier and the business, specifying terms, conditions, and obligations.
5. Shipment (SH): The process of transporting goods from one place to another, typically from the supplier to the warehouse or directly to customers.
6. Carrier (CR): An entity or company that undertakes the transportation of goods typically via land, sea, or air.
7. Carrier ID (CID): A unique identifier for a carrier, used to track and manage logistic arrangements and performance.
8. Address (AD): A precise location, typically used for delivery or billing purposes, involving street, city, state, postal code, and country.
9. Manufacturer (MF): The producer of goods, often responsible for the initial stages of the supply chain.
10. Warehouse (WH): A facility for storing goods, often a crucial hub in the supply chain managing multiple product lines and inventory levels.
11. Warehouse ID (WHID): A unique identifier for each warehouse to distinguish among various storage locations within the business network.

## Logical Scheme



## Objects

1. **Product Categories table description:**

This table stores information about product categories that group products by similar characteristics or uses.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Product Categories | CategoryID | Unique identifier for the category, PK | Int |
| CategoryName | Name of the product category | Varchar(255) |

* The CategoryID serves as the primary key to uniquely identify each product category.

1. **The Products table description:**

This table stores detailed information about each product, including associations to its category, manufacturer, and storage warehouse.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Products | ProductBarcode | Unique barcode for the product, PK | Varchar(30) |
| ProductName | Name of the product | Varchar(255) |
| CategoryID | Identifier for the category, FK | Int |
| ManufacturerID | Identifier for the manufacturer, FK | Int |
| PricePerUnit | Price per unit of the product | Decimal |
| UnitWeight | Weight of a single unit | Decimal |
| WarehouseID | Identifier for the warehouse, FK | Int |

* The ProductBarcode serves as the primary key.
* The CategoryID, ManufacturerID, and WarehouseID fields are foreign keys referencing the Product Categories, Manufacturers, and Warehouses tables respectively.

1. **The Suppliers table description:**

This table stores information about suppliers who provide products.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Suppliers | SupplierContractNumber | Unique contract number, PK | Varchar(20) |
| SupplierName | Name of the supplier company | Varchar(255) |
| AddressID | Address identifier, FK | Int |
| Phone | Contact phone number | Varchar(20) |
| Email | Contact email address | Varchar(255) |

* The SupplierContractNumber serves as the primary key.
* The AddressID field is a foreign key referencing the Addresses table.

1. **The Shipments table description:**

This table holds information about each shipment, linking products with their distribution details.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Shipments | InvoiceNumber | Unique identifier for the shipment, PK | Int |
| SupplierContractNumber | Identifier for the supplier, FK | Varchar(20) |
| CarrierID | Identifier for the carrier, FK | Int |
| ShipmentDate | Date of the shipment | Date |
| ProductBarcode | Barcode of the product being shipped, FK | Varchar(30) |
| Quantity | Number of units being shipped | Int |
| PricePerUnit | Price per unit of the product | Decimal |

* The InvoiceNumber serves as the primary key.
* The SupplierContractNumber, CarrierID, and ProductBarcode fields are foreign keys referencing the Suppliers, Carriers, and Products tables respectively.

1. **The Manufacturers table description:**

This table stores information about manufacturers who produce the products.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Manufacturers | ManufacturerID | Unique identifier, PK | Int |
| ManufacturerName | Name of the manufacturing company | Varchar(100) |
| AddressID | Address identifier, FK | Int |
| Phone | Contact phone number | Varchar(20) |
| Email | Contact email address | Varchar(255) |

* The ManufacturerID serves as the primary key.
* The AddressID field is a foreign key referencing the Addresses table.

1. **The Addresses table description:**

This table holds precise location details for warehouses, suppliers, manufacturers, and carriers.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Addresses | AddressID | Unique identifier for the address, PK | Int |
| Street | Street address | Varchar(255) |
| City | City where the address is located | Varchar(100) |
| State | State where the address is located | Varchar(100) |
| PostalCode | Postal code of the address | Varchar(20) |
| Country | Country where the address is located | Varchar(100) |

* The AddressID serves as the primary key to uniquely identify each address.

1. **The Warehouses table description:**

This table stores information about the facilities used for storing products.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Warehouses | WarehouseID | Unique identifier for the warehouse, PK | Int |
| WarehouseName | Name of the warehouse | Varchar(50) |
| AddressID | Address identifier for the warehouse, FK | Int |
| Phone | Contact phone number of the warehouse | Varchar(20) |

* The WarehouseID serves as the primary key.
* The AddressID field is a foreign key referencing the Addresses table.

1. **The Carriers  table description:**

This table holds details about the carriers used to transport products from suppliers to warehouses and customers.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Carriers | CarrierID | Unique identifier for the carrier, PK | Int |
| CarrierName | Name of the carrier | Varchar(100) |
| AddressID | Address identifier for the carrier, FK | Int |
| Phone | Contact phone number | Varchar(20) |
| Email | Contact email address | Varchar(255) |

* The CarrierID serves as the primary key.
* The AddressID field is a foreign key referencing the Addresses table.

## SCRIPT run instructions

1. Create a new Database “warehouse”:

CREATE DATABASE "warehouse"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LOCALE\_PROVIDER = 'libc'

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

1. To create the tables - Run the script **“Warehouse\_OLTP.sql”** from the directory “script” on the created database “warehouse”.
2. To fill in the tables:

* Copy all files from the directory “csv” to the temporary directory - C:/tmp/
* Run the script “Warehouse\_ETL1.sql” on the database “warehouse”.

1. To see the insights about some metrics - Run the script **“Warehouse\_I**nsights\_**OLTP.sql”** from the directory “script” on the created database “warehouse”.

# OLAP Model description

## Definitions and Objects

The definitions for the proposed data model:

**1. FactWarehouseStores:**

The `FactWarehouseStores` table is designed to analyze and account for the goods received and stored at various warehouses. This table provides aggregated data about shipments received over time, categorized by various dimensions such as product, warehouse, category, and supplier.

Here are the specific facts and dimensions captured:

* InvoiceNumber (Primary Key): The unique invoice number of each shipment received at the warehouse, linked to the `Shipments.InvoiceNumber`.
* WarehouseID: The identifier for the warehouse where the goods are received, linked to `Warehouses.WarehouseID`. This helps in analyzing the distribution of goods across different warehouses.
* ProductBarcode: The barcode of the product received, linked to `Products.ProductBarcode`. This allows for product-specific analysis.
* CategoryID: The category identifier of the product, linked to `ProductCategories.CategoryID`. This categorization helps in analyzing the types of products received.
* SupplierContractNumber: The contract number of the supplier from whom the goods are received, linked to `Suppliers.SupplierContractNumber`. This dimension aids in supplier-specific analysis.
* ShipmentDate: The date the goods were shipped, linked directly with `Shipments.ShipmentDate`. This allows for time-based analysis down to the day level.
* QuantityShipped: The quantity of the goods received in units. This measure is essential for managing logistics and understanding shipment volumes.
* PricePerUnit: The current price of the unit.
* ValueShipped: The total monetary value of the good received from the suppliers. This fact helps in financial analysis and planning, especially in assessing the economic impact of various suppliers on supply costs.

**2. FactManufacturerShipments:**

The `FactManufacturerReceipts` table tracks the goods received from manufacturers, giving insights into the flow of goods into the warehouses from different producers. This table records shipments by their invoice number and details their manufacturer, product specifics, and financial and physical metrics.

Here are the details:

* InvoiceNumber (Primary Key): The unique invoice number of each manufacturer's shipment, linked to the `Shipments.InvoiceNumber`.
* ManufacturerID: The identifier of the manufacturer from whom the goods are received, linked to `Manufacturers.ManufacturerID`. This is crucial for analyzing the performance and reliability of different manufacturers.
* ProductBarcode: The barcode of the product received from manufacturers, linked to `Products.ProductBarcode`. It helps in detailed product tracking and inventory management.
* ShipmentDate: The date the goods were shipped from the manufacturers, linked directly with `Shipments.ShipmentDate`. This supports temporal analysis for better supply chain planning.
* QuantityShipped: The quantity of the goods received in units. This measure is essential for managing logistics and understanding shipment volumes.
* PricePerUnit: The current price of the unit.
* ValueShipped: The total monetary value of the good received from the manufacturers. This fact helps in financial analysis and planning, especially in assessing the economic impact of various manufacturers on supply costs.

Both these tables serve as a cornerstone for detailed analytics regarding the logistics and supply chain operations of a business. They offer insights into the temporal distribution, financial implications, and physical handling of goods received, providing a comprehensive view for strategic decision-making.

For a robust and functional BI environment, several dimension tables are essential to support the fact tables I mentioned earlier (FactWarehouseReceipts and FactManufacturerReceipts). These dimension tables help to contextualize the data points within the fact tables, allowing for multi-dimensional analysis. Here is a description of each needed dimension table:

**1. DimProduct:**

The DimProduct table gathers all necessary attributes related to products, facilitating detailed product-level analysis within the business.

* ProductBarcode (Primary Key): The unique barcode associated with each product. This identifier allows each product to be distinctly tracked and analyzed.
* ProductName: The name of the product which helps in easy identification and reporting.
* CategoryID: A Foreign Key that links to the DimCategory table. This relationship allows for aggregated analysis based on product categories.
* ManufacturerID: A Foreign Key that links to the DimManufacturer table for evaluating the products by their manufacturers.
* PricePerUnit: The price per unit of the product which assists in financial calculations and profit analysis.
* UnitWeight: The weight of a single unit of the product, important for logistics and inventory management.

**2. DimCategory:**

This table categorizes products, which is essential for segmenting product-related data analysis.

* CategoryID (Primary Key): The unique identifier for each category of products.
* CategoryName: The descriptive name of the category, which enhances the readability and usability of reports.

**3. DimManufacturer:**

The DimManufacturer table keeps track of product manufacturers, crucial for supply chain and vendor performance analytics.

* ManufacturerID (Primary Key): The identifier for each manufacturer.
* ManufacturerName: The name of the manufacturer. This dimension is vital for tracing products back to their producers and analyzing manufacturer-related metrics.

**4. DimWarehouse:**

The DimWarehouse table stores details about warehouses, supporting logistics and inventory management analyses.

* WarehouseID (Primary Key): The unique identifier for each warehouse.
* WarehouseName: The name of each warehouse, useful for operational and logistic reports.

**5. DimTime:**

This is a standard dimension in many BI systems that deals with time-related data for enabling trend analysis and time-based filtering.

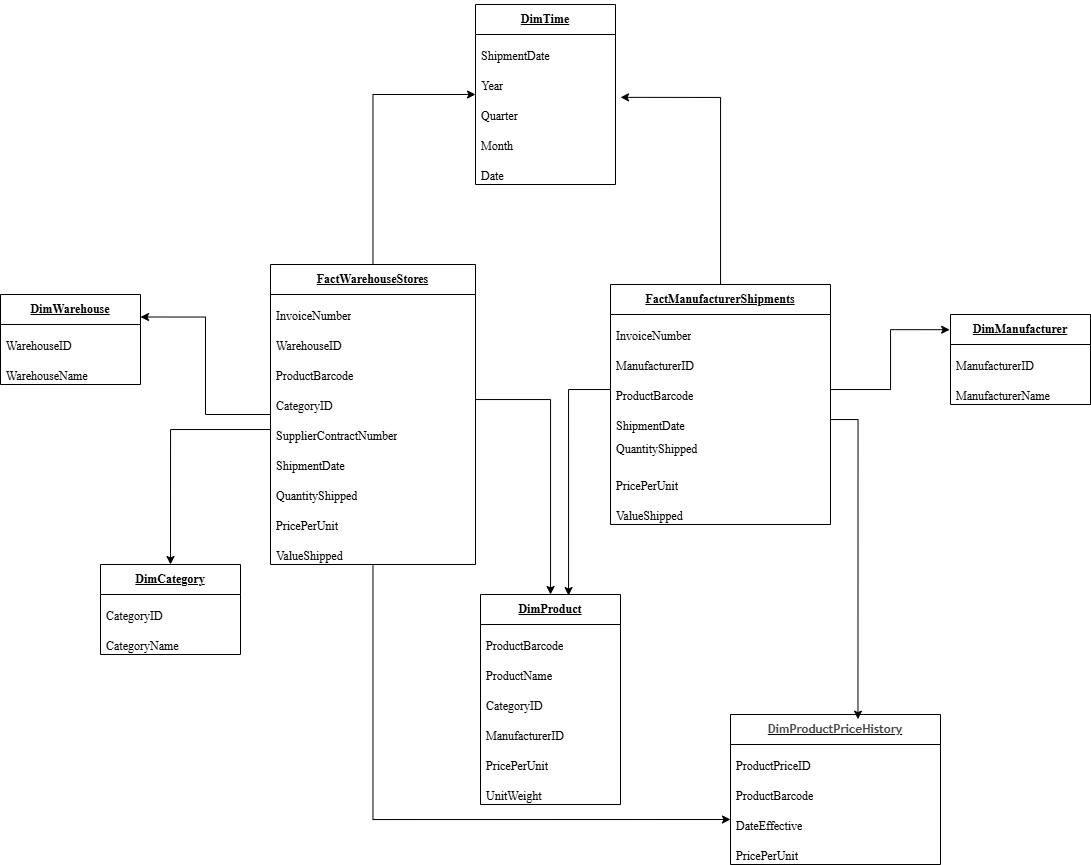
* ShipmentDate (Primary Key): A unique key representing each date, formatted as YYYYMMDD.
* Year: The year extracted from the date, useful for annual trend analysis.
* Quarter: The quarter of the year, facilitating quarterly performance reviews and summaries.
* Month: The month extracted from the date, aiding in monthly analysis and reporting.
* Date: The full date, essential for daily granularity in reporting and analysis.locations within the business network.

1. **DimProductPriceHistory**

This table will store historical data about product prices, allowing for analysis of price changes over time.

* ProductPriceID (Primary Key): A unique identifier for each record of a price change.
* ProductBarcode: The product code (Foreign Key, linked to DimProduct.ProductBarcode). This connection identifies the product whose price has changed.
* DateEffective: The date from which the given price is effective. This allows for studying how product prices have changed over time.
* PricePerUnit: The unit price of the product on this date. This key metric will be analyzed.

## Logical Scheme



## sCRIPT run instructions

1. Create a new Database “WarehouseDWH”:

CREATE DATABASE "WarehouseDWH"

WITH

OWNER = postgres

ENCODING = 'UTF8'

LOCALE\_PROVIDER = 'libc'

CONNECTION LIMIT = -1

IS\_TEMPLATE = False;

1. To create the tables - Run the script **“Warehouse\_OLAP.sql”** from the directory “script” on the created database “WarehouseDWH”.
2. To fill in the tables - Run the script “Warehouse\_ETL2.sql” on the database “WarehouseDWH”.
3. To see the insights about some metrics - Run the script **“Warehouse\_I**nsights\_**OLAP.sql”** from the directory “script” on the created database “WarehouseDWH”.