



Evaluation of Aldi's Inventory Management and Supply Chain Strategies

ISTM 6202 Team Project

Team 02

Saksham Verma

Sakshi Pandit

Falguni Shah

Vishal Vangala

Feichen Yu

Link to the PPT Slides: <https://1drv.ms/p/s!AjieO1QGFJ42giFKuQiduqYPYNF1>

Table of Contents

| | |
|---|----|
| <i>Executive Summary</i> | 1 |
| Section I: Business Problem | 2 |
| 1.1 Description of Organization and Its Problem(s)..... | 2 |
| 1.2 As-IS Business Process Map..... | 3 |
| 1.3 Conceptual Database Design (As-IS ERD)..... | 5 |
| Section II: IT-Based Solution Development | 9 |
| 2.1 Description of IT-based Solution..... | 9 |
| 2.2 To-Be Business Process Map..... | 11 |
| 2.3 Conceptual Database Design (To-Be ERD)..... | 13 |
| 2.4 Business Function to Data Entity Matrix..... | 18 |
| 2.5 Logical Database Design..... | 18 |
| Section III: Database Implementation | 19 |
| 3.1 Physical Database Design..... | 19 |
| 3.2 Create Database..... | 25 |
| 3.3 Select Queries..... | 35 |
| References | 37 |

Executive Summary

The report provides an in-depth analysis of ALDI's inventory management system, emphasizing the need for resilience and efficiency in the face of disruptions like the COVID-19 pandemic and severe weather events. It proposes a range of IT-based solutions to enhance ALDI's inventory management capabilities, ensuring seamless operations and customer satisfaction. ALDI's inventory management system is built on the "just-in-time" approach, minimizing storage costs and optimizing resource allocation. This strategy aligns with ALDI's commitment to offering quality products at competitive prices, ensuring customer satisfaction and loyalty. However, vulnerabilities in the system have been exposed by external disruptions, leading to occasional shortages and stockouts. Events like the COVID-19 pandemic and severe weather phenomena have highlighted the need for resilience and adaptability in ALDI's inventory management practices. The report proposes several IT-based solutions to address these challenges:

- Local Distributor Integration: Seamless integration with local distributors streamlines order placement and tracking, ensuring prompt replenishment of inventory.
- Self-Branded Product Lifecycle Management: A specialized module oversees the entire lifecycle of self-branded products, offering comprehensive oversight and enhancing inventory allocation and innovation opportunities.
- Historical Data Analysis and Trend Forecasting: Utilizing sophisticated analytics and machine learning algorithms enables ALDI's to anticipate demand variations and mitigate risks associated with disruptions, ensuring proactive inventory management.
- Perpetual Inventory Management: Integration of a perpetual inventory management system complements the "just-in-time" approach, enhancing operational efficiency and resilience to unforeseen events.

The proposed solutions aim to strengthen ALDI's inventory management capabilities, ensuring timely order fulfillment, and inventory replenishment. The "To-Be" business process map outlines the intended workflow, emphasizing seamless coordination between departments and external distributors. By implementing these IT-based solutions, ALDI can enhance its operational efficiency, improve customer satisfaction, and maintain its position as a leader in the retail industry amidst evolving market dynamics.

Section I: Business Problem

1.1 Description of Organization and Its Problem(s)

In 1913, the Albrecht family initiated a bakery trade in Essen-Schonnebeck, laying the foundation for a customer-centric business model. Karl and Theo Albrecht took control of ALDI in 1945, expanding it into a branch network focusing on quality products and competitive prices. In 1961, they divided the network due to differing views on cigarette sales, with Karl focusing on southern Germany (ALDI Süd) and Theo on northern Germany (ALDI Nord) (ALDI Nord). ALDI Süd and ALDI Nord coexisted in the United States. ALDI Süd operates ALDI, while ALDI Nord acquired Trader Joe's in the late 1970s (Ngo). Our team will mainly focus on ALDI Süd (abbr. ALDI) in this report. Adaptation is the key to sustained success in the dynamic retail industry landscape. Aldi relies on a sophisticated inventory management system to optimize its operations and maintain its competitive edge. The system is characterized by its "just-in-time" approach, where stock is received only when needed, minimizing storage costs and ensuring efficient resource allocation (Wanyoike-Kimiti). This strategy aligns with Aldi's overarching goal of offering high-quality products at affordable prices.

The "just-in-time" inventory management system enables the company to reduce excess inventory and associated holding costs, ultimately translating into lower consumer prices (Aldi). Aldi can improve operational efficiency and profitability by adjusting inventory based on consumer demand patterns. However, its reliance on timely supplier deliveries makes its inventory system vulnerable to disruptions like transportation delays and natural disasters. These disruptions can lead to shortages of essential goods, impacting customer satisfaction. The vulnerabilities were brought to the forefront by the COVID-19 pandemic, which brought up unprecedented issues due to lockdowns, panic buying, and disruptions in the supply chain (Behr). Aldi's inventory management was stretched due to increased demand for specific products and difficulties in manufacturing, leading to occasional shortages and stockouts in its stores.

Likewise, severe weather phenomena such as winter storm Uri intensified the difficulties Aldi's inventory management system faced (Behr). The storm caused Aldi to have inventory shortages and logistical issues due to power outages, infrastructure damage, and supply chain delays. To enhance its inventory management system's resilience, Aldi needs strategic initiatives to mitigate supply chain risks and boost operational agility. This includes diversifying suppliers, implementing backup plans, and investing in technology for better forecasting and demand planning.

Based on the analysis of these situations, our team has assumed that Aldi should utilize data analytics and predictive modeling techniques to anticipate customer demand variations and proactively adjust inventory levels. By using technology like inventory optimization software and automated replenishment systems, Aldi can enhance efficiency in inventory management, improving overall operations and customer value. While Aldi's "just-in-time" approach offers cost-saving benefits and operational effectiveness, it requires continuous adaptation to thrive in the ever-evolving retail landscape. Addressing supply chain vulnerabilities and embracing technology-driven solutions will enable Aldi to strengthen its inventory management capabilities and maintain its leadership position in the retail industry.

1.2 As-IS Business Process Map

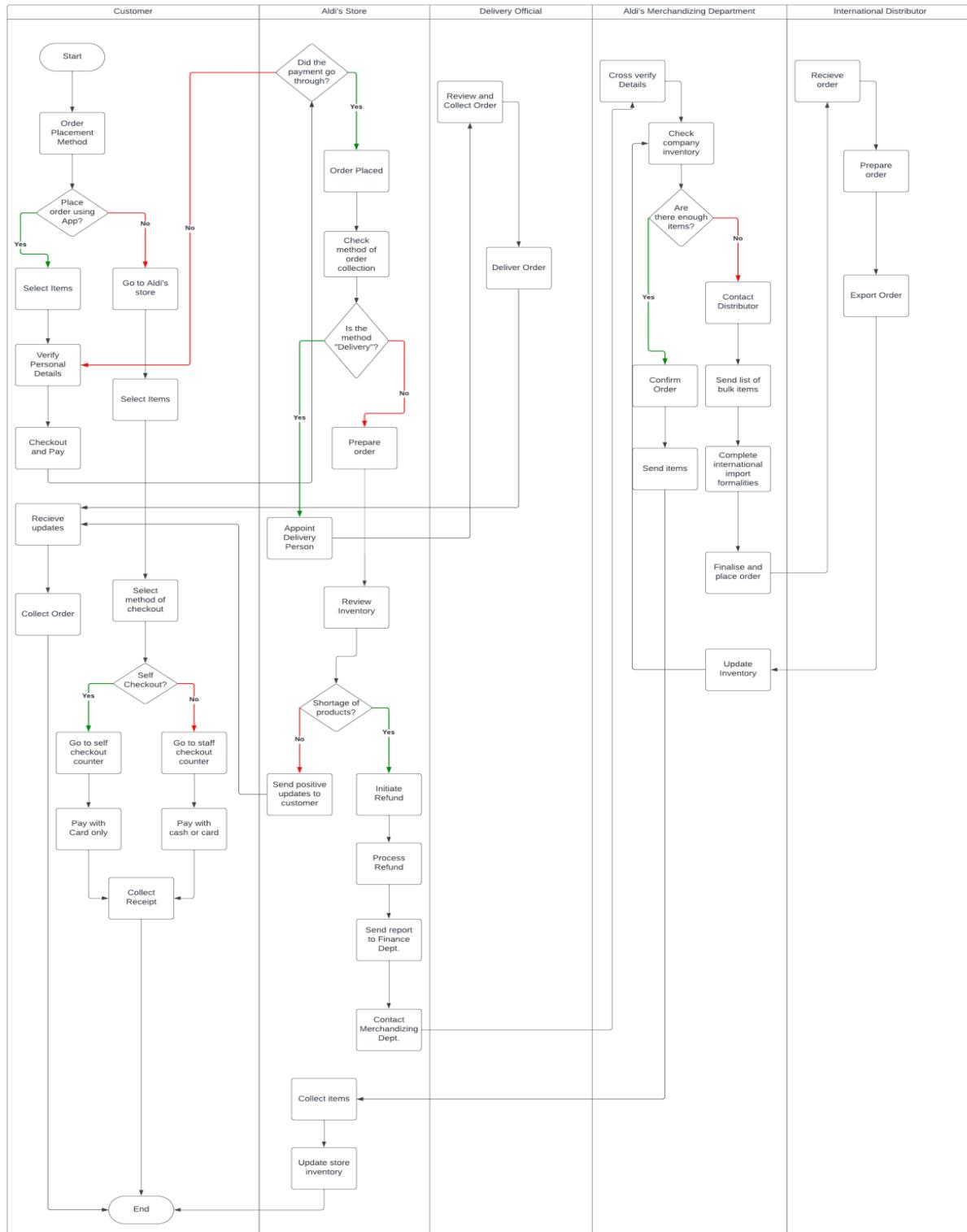


Figure 1. As-Is Business Process Map for Aldi

The business process map above outlines the current Aldi's system for handling customer orders, inventory management, and coordination between different departments and external distributors. It is divided into multiple swimlanes, each representing a different component department in the process, such as the customer, Aldi's store, delivery official, Aldi's merchandising department, and the international distributor.

1. Customer:

- Starts with the customer deciding whether to order via payment method or place the order using the app.
- If payment fails, the process ends.
- If using the app, the customer selects items, verifies personal details, checks out and pays.
- The customer then receives updates and either collects the order in-store or selects the method of collection at checkout, which could be a self-checkout or staff checkout, with the latter allowing payment by cash.

2. Aldi's Store:

- Upon an order placement, the store reviews and collects the order.
- If the collection method is delivery, the order is prepared for delivery.
- If there's an inventory shortage, positive updates are sent to the customer, or a refund is initiated and processed.
- Reports are sent to the finance department, and the merchandising department is contacted.
- Items are collected, and the store inventory is updated.

3. Delivery Official:

- The delivery person approves the order for delivery if that was the selected method.
- The order is then delivered.

4. Aldi's Merchandising Department:

- This department is involved in cross-verifying details and checking the company inventory.
- If there are not enough items, the distributor is contacted, or if sufficient, the order is confirmed.
- A list of bulk items is sent out if required.
- The department is responsible for sending items, and for international orders, completing international formalities.

- Orders are finalized and placed, and inventory is updated accordingly.

5. International Distributor:

- The international distributor receives the order, prepares it, and if it's for export, completes the required formalities before shipping the order.

Processes in this As-Is process map show the Just in Time Approach that Aldi's implements and hence is not able to stock up the items in time often. The contact with the international distributor may seem pretty simple in the process map but it is actually a very formal and time-taking process that Aldi's has to implement for restocking its merchandising department.

1.3 Conceptual Database Design (As-IS ERD)

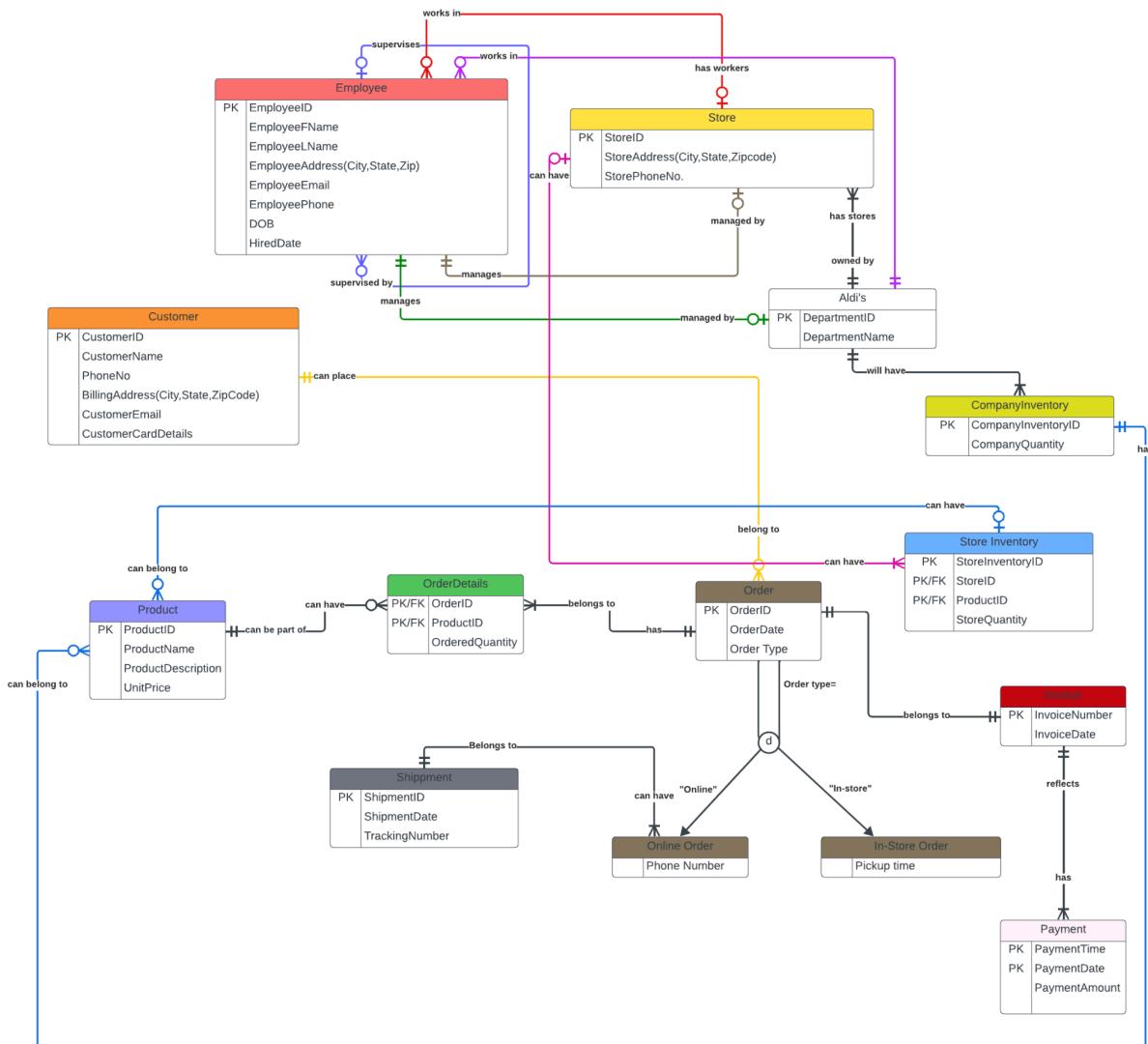


Figure 2. As-Is ERD for Aldi

Business Rules:

1. Employee - This entity stores information about employees, such as name, address, email, phone, role, etc. It has a primary key (PK) 'EmployeeID' which uniquely identifies each employee.

Relationships:

- An employee can supervise other employees.
- An employee manages a store.
- An employee works in a store.

2. Store - This entity holds data about store locations, including street address, city, state, and phone number. Its PK is 'StoreID'.

Relationships:

- A store has workers (employees).
- A store can have several departments.
- A store is managed by an employee.
- A store is owned by a department.
- A store's inventory consists of products.

3. Department - Represents different departments within the company, identified by 'DeptID' as its PK.

Relationships:

- A department is managed by an employee.
- A department owns several stores.
- A department has workers.

4. Supplier - Contains details about the suppliers including name, contact details, and address. The PK here is 'SupplierID'.

Relationships:

- A supplier contracts with a department.
- A supplier belongs to a company inventory entity.
- A supplier has products.

5. Product - Stores details of products, like name, description, and supplier information. Its PK is 'ProductID'.

Relationships:

- A product can be a part of an order.
- A product can belong to store inventory.
- A product can belong to company inventory.
- A product has a supplier.

6. OrderDetails - Keeps track of the details related to customer orders, such as quantity and the ordered products. It has a composite PK made of 'OrderID' and 'ProductID'.

Relationships:

- It belongs to an order.
- It is associated with a product.

7. Order - Stores information about each order, including shipment details. Its PK is 'OrderID'. It has two subtypes, online order and in-store order. Online order has a phone number attribute and the instore order has a pickup time attribute.

Relationships:

- An order can have multiple order details.
- An order can have a shipment.

8. Shipment - Contains details about the shipment of orders like shipment date and tracking number. The PK is 'ShipmentID'.

Relationships:

- A shipment has an order.

9. Invoice - Holds the invoice details for purchases, including invoice date and amount. 'InvoiceNumber' is the PK.

Relationships:

- An invoice has an order.

10. Payment - Records payment transactions, including payment date, amount, and type. Its PK is 'PaymentNumber'.

Relationships:

- A payment is associated with an invoice.

11. Customer - Contains customer information such as name, phone, and billing address. Its PK is 'CustomerID'.

Relationships:

- A customer can place orders.
- A customer can have several order histories.

12. StoreInventory and CompanyInventory - These entities track the inventory in each store and company-wide, respectively. Their PKs are 'StoreInventoryID' and 'CompanyInventoryID'.

Relationships:

- Store inventory belongs to a store and consists of products.
- Company inventory is associated with products and suppliers.

Section II: IT-Based Solution Development

2.1 Description of IT-based Solution

Under the current inventory management system used by ALDI, events like the COVID-19 pandemic and severe weather phenomena have highlighted vulnerabilities in the system, leading to occasional shortages and stockouts in its stores. To sustain its competitive edge and address these challenges, Aldi needs to enhance the resilience of its inventory management system while ensuring efficient resource allocation and cost savings. The IT-based solution our team has come up with the following IT solutions:-

1. Local Distributors Integration

Just-in-time supply management relies heavily on suppliers to deliver timely materials and components. Any disruptions in the supply chain, such as delayed deliveries or quality issues, can lead to production delays or shortages. Thus, ALDI has the potential to improve its IT system by integrating it with the systems of its local distributors. This enables instantaneous contact, facilitating smooth order placement, tracking, and fulfillment. By implementing automation in the ordering process and optimizing communication channels, ALDI could reduce delays and errors typically associated with manual order processing. Besides, ALDI can know the supply and demand situation of different products promptly, including the quality and quantity of the product, whether cold chain distribution is required, the cycle of the product in the warehouse, etc. This, in turn, guarantees the prompt replenishment of inventory.

2. Self-Branded Product Lifecycle Management

ALDI can create a specialized module within its management system that can effectively oversee the complete life cycle of self-branded products. This encompasses several stages, including acquiring raw materials, product development, manufacture, distribution, and sales. This module offers a comprehensive overview of every phase of the product lifecycle, enabling ALDI to manage inventory levels effectively, actively watch production progress, and conduct real-time sales performance analysis. ALDI can enhance inventory allocation, reduce waste, and uncover opportunities for product innovation and expansion by consolidating data of self-branded products. Furthermore, incorporating external modules, such as demand forecasting and supplier management, allows ALDI to effectively align inventory levels with customer demand, thereby guaranteeing a consistent provision of self-branded merchandise within its retail establishments. The increased visibility and control over the inventory of self-branded products positively impact the efficiency of inventory management and the profitability of ALDI.

3. Historical Data Analysis and Trend Forecasting

ALDI has the potential to utilize sophisticated analytics and machine learning algorithms to examine past sales data and forecast forthcoming patterns of demand. ALDI can make informed decisions about inventory levels and product assortments by analyzing

historical sales performance and recognizing seasonal patterns, trends, and consumer preferences. Simultaneously, given the circumstances faced during the COVID-19 pandemic, it is imperative to consider the concealed hazards arising from the pandemic and diverse extreme weather events. Using predictive analytics enables ALDI to proactively adjust inventory levels, improve the composition of products, and mitigate the potential risks associated with out-of-stock and overstocking.

4. Perpetual Inventory Management

Just-in-time systems require frequent, smaller deliveries with high quality. This will cause a higher transportation fee, and any minor defects will quickly disrupt Aldi's wholesale chain (Melanie). Thus, Aldi may optimize its operational efficiency by integrating a perpetual inventory management system alongside lean inventory methods. Aldi could minimize its stock to keep the warehouse fee low and still make its price competitive. But, with some stock on hand, Aldi can handle sudden events better, including natural disasters, pandemics, etc. It is still essential to implement lean concepts, which involve reducing waste, optimizing processes, and keeping inventory levels low; Aldi may enhance operational efficiency and increase inventory turnover ratios.

2.2 To-Be Business Process Map

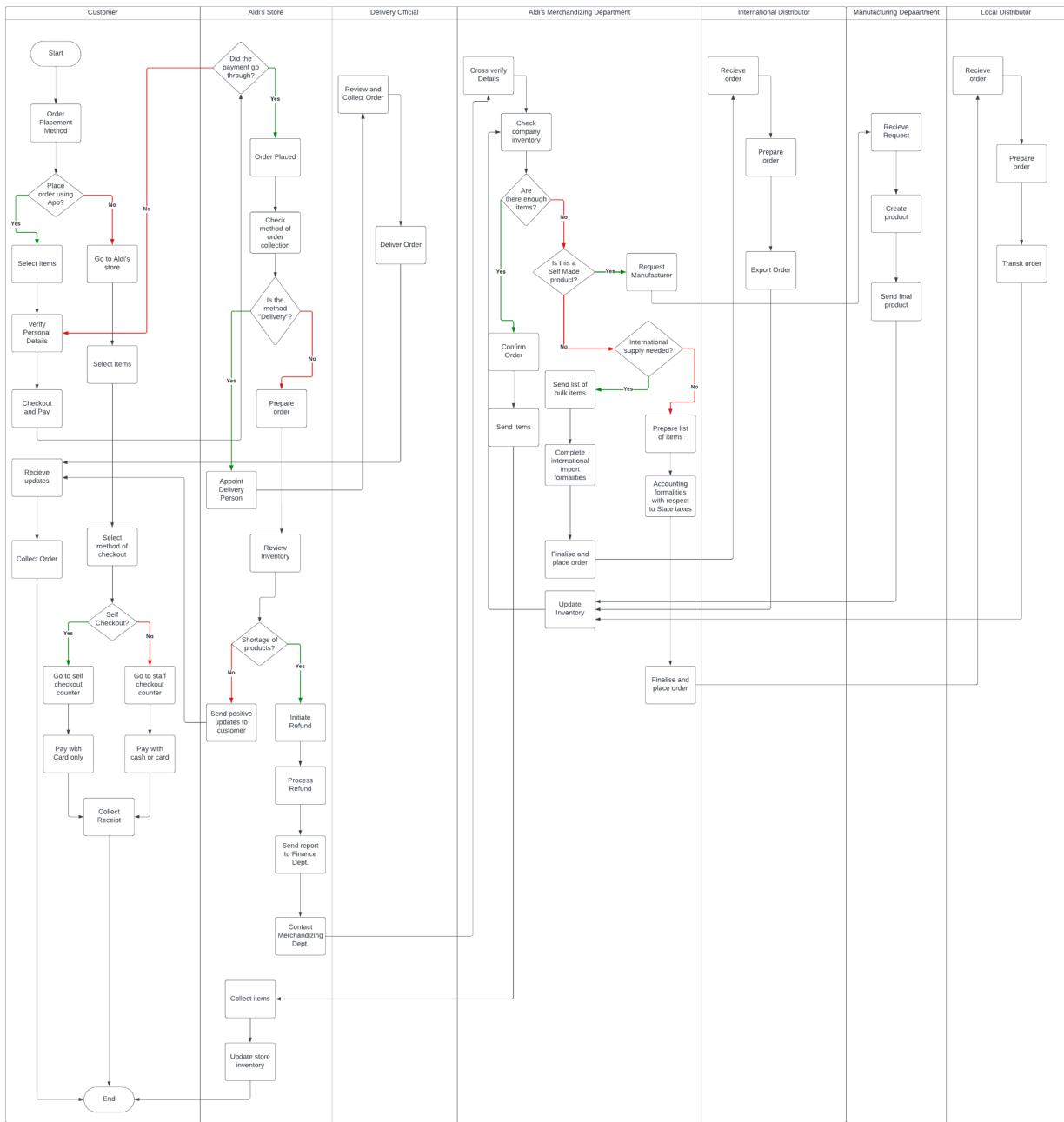


Figure 3. To-Be Business Process Map

The "To-Be" business process map, outlines the intended process for Aldi's inventory management system, for an order fulfillment workflow involving several departments of the business.

1. Customer:

- The process begins with the customer starting an order. The order can either be handled or not handled, leading to a decision point.
- If handled, the customer selects items, verifies details, chooses the checkout method (card or cash), and then either collects the order or receives a response and collects the order.
- If not handled, the flow is directed outside of the customer area and is not visible on this part of the diagram.

2. Aldi's Store:

- An order may also begin from Aldi's store, where it gets reviewed and placed.
- The credit check is done at the store level if the order is on credit.
- Depending on whether delivery is needed, the process may split. If delivery is required, a delivery person is involved; if not, the order goes to inventory management within the store.
- There is also a return process, where the item is either sent to a warranty person, goes back into inventory, or is refunded.
- Positive feedback may be sent to the customer, and the store's inventory is updated.

3. Delivery Official:

- The delivery official checks inventory and either cross-verifies with the store or goes straight to delivering the order.
- If the item isn't in stock, the flow splits to the Aldi's Merchandising Department.

4. Aldi's Merchandising Department:

- Here, the merchandising department checks inventory.
- If the item isn't available, they may request it from the manufacturer in case of a self made product.
- If it is not a self made product then the branding will be checked (if it is an international brand or a local brand)
- If an international brand, then a request may be sent to the international distributor with completing all the international documentation and formalities but if it is a local brand, then the accounting with respect to the state tax will be done and the request will be sent to the local distributor.

- Upon receiving the item, they confirm the order, prepare the box, complete consignment information, finalize and place the order, and update the inventory.

5. International Distributor:

- Receives the order, prepares it, exports it to the merchandising department who in turn updates the inventory.

6. Manufacturing Department:

- Receives the request, creates the product, transmits the order, and sends the final product out.

7. Local Distributor:

- They play a role similar to the International Distributor but focused on local distribution. They prepare and transit the order to its destination.

2.3 Conceptual Database Design (To-Be ERD)

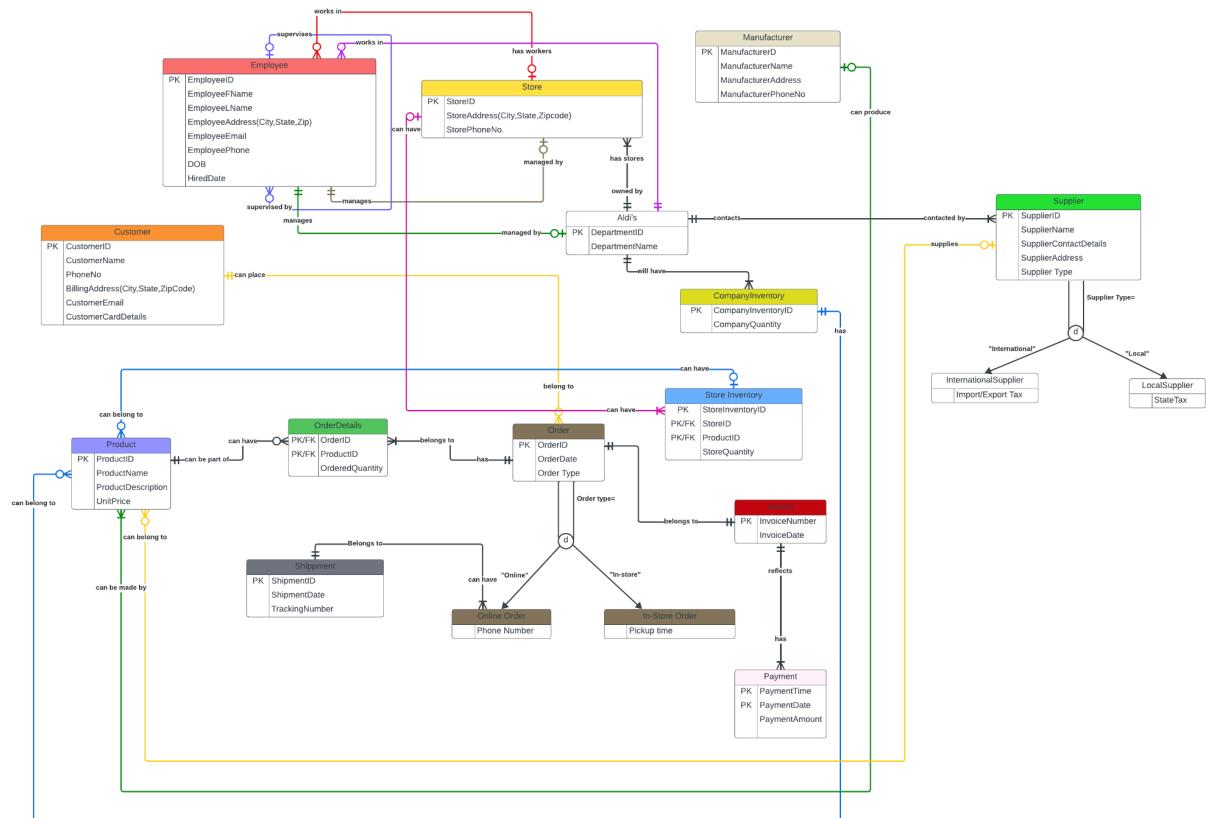


Figure 4. To-Be ERD

Business Rules:

1. Employee - This entity stores information about employees, such as name, address, email, phone, role, etc. It has a primary key (PK) 'EmployeeID' which uniquely identifies each employee.

Relationships:

- An employee can supervise other employees.
- An employee manages a store.
- An employee works in a store.
- An employee may or may not work in a manufacturing department

2. Store - This entity holds data about store locations, including street address, city, state, and phone number. Its PK is 'StoreID'.

Relationships:

- A store has workers (employees).
- A store can have several departments.
- A store is managed by an employee.
- A store is owned by a department.
- A store's inventory consists of products.

3. Department - Represents different departments within the company, identified by 'DeptID' as its PK.

Relationships:

- A department is managed by an employee.
- A department owns several stores.
- A department has workers.

4. Supplier - Contains details about the suppliers including name, contact details, and address. The PK here is 'SupplierID'. The supplier may be a local supplier or an international supplier but has to be one of the either.

Relationships:

- A supplier contracts with a department.
- A supplier belongs to a company inventory entity.

- A supplier has products.
5. Product - Stores details of products, like name, description, and supplier information. Its PK is 'ProductID'.

Relationships:

- A product can be a part of an order.
 - A product can belong to store inventory.
 - A product can belong to company inventory.
 - A product can be manufactured or be self made.
 - A product has a supplier.
6. OrderDetails - Keeps track of the details related to customer orders, such as quantity and the ordered products. It has a composite PK made of 'OrderID' and 'ProductID'.

Relationships:

- It belongs to an order.
 - It is associated with a product.
7. Order - Stores information about each order, including shipment details. Its PK is 'OrderID'. It has two subtypes, online order and in-store order. Online order has a phone number attribute and the instore order has a pickup time attribute.

Relationships:

- An order can have multiple order details.
 - An order can have a shipment.
8. Shipment - Contains details about the shipment of orders like shipment date and tracking number. The PK is 'ShipmentID'.

Relationships:

- A shipment has at least one online order.
9. Invoice - Holds the invoice details for purchases, including invoice date and amount. 'InvoiceNumber' is the PK.

Relationships:

- An invoice has an order and one payment associated with it.
10. Payment - Records payment transactions, including payment date, amount, and type. It's PK is 'PaymentNumber'.

Relationships:

- A payment is associated with an invoice.

11. Customer - Contains customer information such as name, phone, and billing address. Its PK is 'CustomerID'.

Relationships:

- A customer can place orders.
- A customer can have several order histories.

12. StoreInventory and CompanyInventory - These entities track the inventory in each store and company-wide, respectively. Their PKs are 'StoreInventoryID' and 'CompanyInventoryID'.

Relationships:

- Store inventory belongs to a store and consists of products.
- Company inventory is associated with products and suppliers.

13. Manufacturer - Contains manufacturer information for ordering self-made products from. The PK is Manufacturer ID.

Relationships:

- A manufacturing department will have at least one employee and can have many.
- A manufacturing department will create at least one product and can create many.

The 5 C's of business with respect to Aldi's inventory management system:-

Aldi's inventory management system likely adheres to principles that resonate with efficient supply chain management and retail operations. Though specific details of Aldi's internal processes aren't accessible, we can outline a general framework integrating the 5C's concept and suggesting typical business rules pertinent to their inventory management:

1. Customer Satisfaction (C1):

- Business Rule: Guarantee high-demand products are consistently available to fulfill customer expectations.
- Example: Establish minimum inventory thresholds for popular items and enact automatic reorder triggers.

2. Cost Control (C2):

- Business Rule: Minimize carrying costs by optimizing inventory levels and turnover rates.
- Example: Implement just-in-time inventory practices to diminish surplus stock and the associated holding costs.

3. Cash Flow Management (C3):

- Business Rule: Fine-tune procurement processes to optimize cash flow.
- Example: Negotiate advantageous payment terms with suppliers to defer cash outflows while sustaining inventory levels.

4. Communication (C4):

- Business Rule: Enable real-time communication between stores and central distribution centers for precise demand forecasting.
- Example: Employ RFID or barcode technology for inventory tracking and integrate with a centralized database for instant updates across the supply chain.

5. Coordination (C5):

- Business Rule: Harmonize inventory replenishment activities across stores to prevent stockouts and overstock scenarios.
- Example: Deploy centralized inventory management software to synchronize replenishment orders based on demand forecasts and inventory levels at each location.

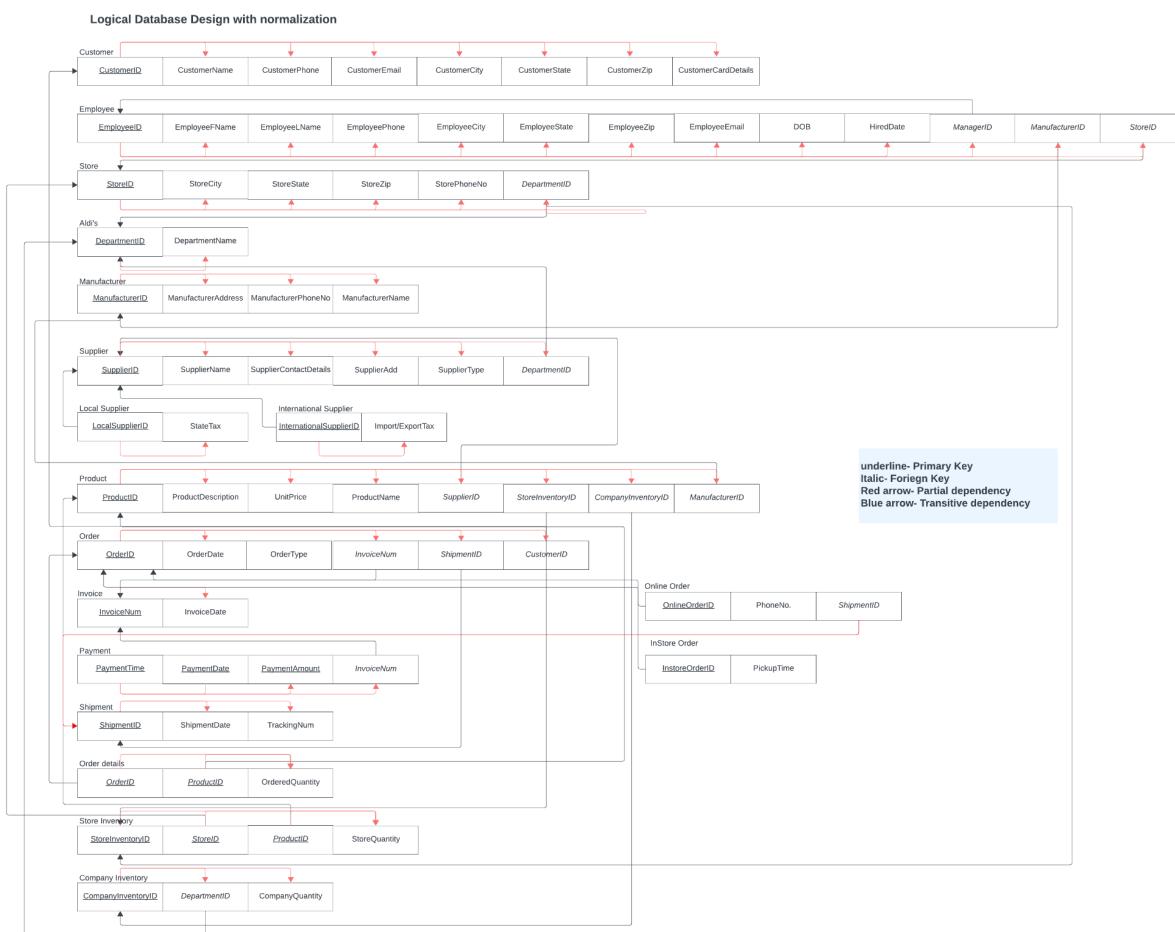
These principles, complemented by tailored business rules and technology solutions that are provided in this section may empower Aldi's to proficiently manage their inventory, satisfying customer demand, controlling costs, optimizing cash flow, enhancing communication, and ensuring coordination throughout their supply chain network.

2.4 Business Function to Data Entity Matrix

| | Data Entity | | | | | | | | | | | | | |
|------------------------|-------------|----------|-------|--------|--------------|----------|---------|-------|---------|---------|----------|---------------|-----------------|-------------------|
| Business Functions | Customer | Employee | Store | Aldi's | Manufacturer | Supplier | Product | Order | Invoice | Payment | Shipment | Order Details | Store Inventory | Company Inventory |
| Business Planning | X | X | | X | X | | | X | | | | | X | X |
| Product Development | | | | X | X | X | | | | | | | X | X |
| Inventory Management | | | | X | X | X | X | X | | | | | X | X |
| Order Fulfillment | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| Order Shipment | X | X | X | | | | | X | X | | X | X | | |
| Sales Summarization | X | X | X | | | | | X | X | X | | | X | |
| Production Operations | | | X | X | X | X | X | | | | | | X | X |
| Finance and Accounting | X | X | X | X | X | X | X | X | X | X | | | | |

X = data entity used within business function

2.5 Logical Database Design



Section III: Database Implementation

3.1 Physical Database Design

```
CREATE TABLE CUSTOMER(  
    customerID VARCHAR(50) not null,  
    customername VARCHAR(50) not null,  
    customerphone int(10) not null,  
    customeremail VARCHAR(50) not null,  
    customercity VARCHAR(50) not null,  
    customerstate VARCHAR(50) not null,  
    customerzip VARCHAR(50) not null,  
    customercarddetails VARCHAR(50) not null,  
    constraint cpk PRIMARY KEY(customerID)  
);
```

```
CREATE TABLE STORE (  
    StoreID VARCHAR(50) NOT NULL,  
    StoreCity VARCHAR(50) NOT NULL,  
    StoreState VARCHAR(50) NOT NULL,  
    StoreZip VARCHAR(50) NOT NULL,  
    StorePhoneNo INT(10) NOT NULL,  
    CONSTRAINT store_pk PRIMARY KEY (StoreID)  
);
```

```
CREATE TABLE Aldis_T(  
    DepartmentID VARCHAR(50) NOT NULL,  
    DepartmentName VARCHAR(50) NOT NULL,
```

```
StoreID VARCHAR(50) NOT NULL,  
CONSTRAINT Aldis_PK PRIMARY KEY (DepartmentID),  
CONSTRAINT Aldis_FK FOREIGN KEY(StoreID) REFERENCES STORE(StoreID)  
);
```

```
CREATE TABLE Manufacturer(  
ManufacturerID VARCHAR(50) NOT NULL,  
ManufacturerAddress VARCHAR(100) NOT NULL,  
ManufacturerPhoneNo INT(10) NOT NULL,  
ManufacturerName VARCHAR(50) NOT NULL,  
CONSTRAINT Manufacturer_PK Primary Key (ManufacturerID));
```

```
CREATE TABLE COMPANYINVENTORY(  
CompanyInventoryID VARCHAR (50) NOT NULL,  
DepartmentID VARCHAR(50) NOT NULL,  
CompanyQuantity INT(10) NOT NULL,  
CONSTRAINT cipk PRIMARY KEY (CompanyInventoryID),  
CONSTRAINT cifk FOREIGN KEY (DepartmentID) REFERENCES  
Aldis_T(DepartmentID)  
);
```

```
CREATE TABLE STOREINVENTORY (  
StoreInventoryID VARCHAR (50) NOT NULL,  
StoreID VARCHAR(50) NOT NULL,  
StoreQuantity INT(10) NOT NULL,  
CONSTRAINT sipk PRIMARY KEY(StoreInventoryID),  
CONSTRAINT sifk FOREIGN KEY(StoreID) REFERENCES store(StoreID)
```

);

```
CREATE TABLE Supplier(
    SupplierID VARCHAR(50) NOT NULL,
    SupplierName VARCHAR(50) NOT NULL,
    SupplierContactDetails INT(10) NOT NULL,
    SupplierAdd VARCHAR (50) NOT NULL,
    DepartmentID VARCHAR(50) NOT NULL,
    SupplierType ENUM('LocalSupplier' , 'InternationalSupplier') NOT NULL,
    CONSTRAINT Supplier_PK PRIMARY KEY (SupplierID),
    CONSTRAINT Supplier_FK FOREIGN KEY(DepartmentID) REFERENCES
        aldis_t(DepartmentID)
);
```

```
CREATE TABLE LocalSupplier (
    LocalSupplierID VARCHAR(50) NOT NULL,
    StateTax DECIMAL(6,2) NOT NULL,
    CONSTRAINT LocalSupplier_PK PRIMARY KEY (LocalSupplierID),
    CONSTRAINT LocalSupplier_FK Foreign Key (LocalSupplierID) REFERENCES Supplier
        (SupplierID));
```

```
CREATE TABLE InternationalSupplier(
    InternationalSupplierID VARCHAR(50) NOT NULL,
    ImportExportTax DECIMAL(6,2) NOT NULL,
    CONSTRAINT InternationalSupplier_PK PRIMARY KEY (InternationalSupplierID),
    CONSTRAINT InternationalSupplier_FK Foreign Key (InternationalSupplierID)
        REFERENCES Supplier(SupplierID));
```

```

CREATE TABLE Employee (
    employeeID VARCHAR(50) NOT NULL,
    employeeFName VARCHAR(50) NOT NULL,
    employeeLname VARCHAR(50) NOT NULL,
    employeePhone VARCHAR(20) NOT NULL,
    employeeCity VARCHAR(50) NOT NULL,
    employeeState VARCHAR(50) NOT NULL,
    employeeZip VARCHAR(10) NOT NULL,
    employeeEmail VARCHAR(100) NOT NULL,
    DOB DATE NOT NULL,
    HiredDate DATE NOT NULL,
    ManagerID VARCHAR(50),
    StoreID VARCHAR(50) NOT NULL,
    ManufacturerID VARCHAR(50) NOT NULL,
    CONSTRAINT epk PRIMARY KEY(employeeID),
    CONSTRAINT fk_ManagerID FOREIGN KEY (ManagerID) REFERENCES
Employee(employeeID),
    CONSTRAINT fk_StoreID FOREIGN KEY (StoreID) REFERENCES STORE(StoreID),
    CONSTRAINT fk_ManufacturerID FOREIGN KEY (ManufacturerID) REFERENCES
Manufacturer(ManufacturerID)
);


```

```

CREATE TABLE Product (
    productID VARCHAR(50) NOT NULL,
    productDescription VARCHAR(50) NOT NULL,
    UnitPrice DECIMAL(6,2) NOT NULL,
    ProductName VARCHAR(50) NOT NULL,
    SupplierID VARCHAR(50) NOT NULL,

```

```
StoreInventoryID VARCHAR(50) NOT NULL,  
CompanyInventoryID VARCHAR(50) NOT NULL,  
ManufacturerID VARCHAR(50) NOT NULL,  
CONSTRAINT product_pk PRIMARY KEY (productID),  
CONSTRAINT pfk_SupplierID FOREIGN KEY (SupplierID) REFERENCES  
Supplier(SupplierID),  
CONSTRAINT pfk_ManufacturerID FOREIGN KEY (ManufacturerID) REFERENCES  
Manufacturer(ManufacturerID)  
);
```

```
CREATE TABLE INVOICE(  
invoicenum VARCHAR(50) NOT NULL,  
invoicedate DATE NOT NULL,  
CONSTRAINT ipk PRIMARY KEY (invoicenum)  
);
```

```
CREATE TABLE SHIPMENT(  
shipmentID VARCHAR(50) NOT NULL,  
shipmentdate DATE NOT NULL,  
trackingnum VARCHAR(50) NOT NULL,  
CONSTRAINT spk PRIMARY KEY (shipmentID)  
);
```

```
CREATE TABLE ORDERS(  
orderID VARCHAR(50) NOT NULL,  
orderdate DATE NOT NULL,  
customerID VARCHAR(50) NOT NULL,  
invoicenum VARCHAR(50) NOT NULL,
```

```
shipmentID VARCHAR(50) NOT NULL,  
CONSTRAINT opk PRIMARY KEY (orderID),  
CONSTRAINT ofk1 FOREIGN KEY (customerID) REFERENCES CUSTOMER(customerID),  
CONSTRAINT ofk2 FOREIGN KEY (invoicenum) REFERENCES INVOICE(invoicenum),  
CONSTRAINT ofk3 FOREIGN KEY (shipmentID) REFERENCES SHIPMENT(shipmentID)  
);
```

```
CREATE TABLE Payment (  
paymentID VARCHAR(50) NOT NULL,  
PaymentDate DATE NOT NULL,  
PaymentAmount DECIMAL(8, 2) NOT NULL,  
invoicenum VARCHAR(50) NOT NULL,  
CONSTRAINT pk_payment PRIMARY KEY(paymentID,PaymentDate,PaymentAmount),  
CONSTRAINT fk_InvoiceNumber FOREIGN KEY (invoicenum) REFERENCES  
INVOICE(invoicenum)  
);
```

```
CREATE TABLE OrderDetails (  
orderID VARCHAR(50) NOT NULL,  
productID VARCHAR(50) NOT NULL,  
OrderQuantity VARCHAR(50) NOT NULL,  
CONSTRAINT odpk PRIMARY KEY(orderID,productID),  
CONSTRAINT fk_OrderID FOREIGN KEY (orderID) REFERENCES ORDERS(orderID),  
CONSTRAINT fk_ProductID FOREIGN KEY (productID) REFERENCES  
Product(productID)  
);
```

3.2 Create Database

```
INSERT INTO CUSTOMER (customerID, customername, customerphone, customeremail,  
customercity, customerstate, customerzip, customercarddetails)
```

VALUES

```
('C001', 'John Doe', 123456789, 'john@example.com', 'New York', 'NY', '10001', 'Paypal'),  
('C002', 'Jane Smith', 234568901, 'jane@example.com', 'Los Angeles', 'CA', '90001',  
'Mastercard'),  
('C003', 'Michael Johnson', 346789012, 'michael@example.com', 'Chicago', 'IL', '60601', 'Visa'),  
('C004', 'Emily Davis', 456789123, 'emily@example.com', 'Houston', 'TX', '77001', 'Visa'),  
('C005', 'William Brown', 567801234, 'william@example.com', 'Phoenix', 'AZ', '85001',  
'Mastercard'),  
('C006', 'Mary Wilson', 678901345, 'mary@example.com', 'Philadelphia', 'PA', '19101', 'Venmo'),  
('C007', 'James Taylor', 789013456, 'james@example.com', 'San Antonio', 'TX', '78201',  
'Paypal'),  
('C008', 'Jessica Martinez', 890134567, 'jessica@example.com', 'San Diego', 'CA', '92101',  
'Visa'),  
('C009', 'David Anderson', 901234578, 'david@example.com', 'Dallas', 'TX', '75201',  
'Mastercard'),  
('C010', 'Ashley Thomas', 123457890, 'ashley@example.com', 'San Jose', 'CA', '95101', 'Visa');
```

```
INSERT INTO STORE (StoreID, StoreCity, StoreState, StoreZip, StorePhoneNo)
```

VALUES

```
('S001', 'New York', 'NY', '10001', 123457890),  
('S002', 'Los Angeles', 'CA', '90001', 234678901),  
('S003', 'Chicago', 'IL', '60601', 345789012),  
('S004', 'Houston', 'TX', '77001', 456789023),  
('S005', 'Phoenix', 'AZ', '85001', 567891234),  
('S006', 'Philadelphia', 'PA', '19101', 678902345),  
('S007', 'San Antonio', 'TX', '78201', 780123456),
```

```
('S008', 'San Diego', 'CA', '92101', 890123467),  
('S009', 'Dallas', 'TX', '75201', 902345678),  
('S010', 'San Jose', 'CA', '95101', 123457890);
```

```
INSERT INTO Aldis_T (DepartmentID, DepartmentName, StoreID)
```

```
VALUES
```

```
('D001', 'Grocery', 'S001'),  
(‘D002’, ‘Produce’, ‘S002’),  
(‘D003’, ‘Dairy’, ‘S003’),  
(‘D004’, ‘Meat’, ‘S004’),  
(‘D005’, ‘Bakery’, ‘S005’),  
(‘D006’, ‘Frozen Foods’, ‘S006’),  
(‘D007’, ‘Canned Goods’, ‘S007’),  
(‘D008’, ‘Household’, ‘S008’),  
(‘D009’, ‘Health & Beauty’, ‘S009’),  
(‘D010’, ‘Pet Supplies’, ‘S010’);
```

```
INSERT INTO Supplier (SupplierID, SupplierName, SupplierContactDetails, SupplierAdd,  
DepartmentID, SupplierType)
```

```
VALUES
```

```
(‘SUP001’, ‘ABC Suppliers’, 123456790, ‘123 Supplier St, New York, NY, 87654’, ‘D001’,  
‘LocalSupplier’),  
(‘SUP002’, ‘XYZ Suppliers’, 245678901, ‘456 Supplier Ave, Cary, NC, 09760’, ‘D002’,  
‘LocalSupplier’),  
(‘SUP003’, ‘123 Supplies’, 345689012, ‘789 Supplier Blvd, Charlotte, NV, 99757’, ‘D003’,  
‘LocalSupplier’),  
(‘SUP004’, ‘456 Distributors’, 456890123, ‘321 Supplier Rd, Washington, DC, 20038’, ‘D004’,  
‘LocalSupplier’),
```

('SUP005', '789 Wholesalers', 567891234, '654 Supplier Ln, San Fransisco, CA, 98742', 'D005',
'LocalSupplier'),
(('SUP006', 'ABC&D Importers', 678902345, '987 Supplier Dr, Irvine, CA, 66833', 'D006',
'InternationalSupplier'),
(('SUP007', 'XYZ Corp', 789012456, '741 Supplier Pl, Buffalo, NY, 54279', 'D007',
'InternationalSupplier'),
(('SUP008', '1234 Inc', 890123567, '852 Supplier Ct, Yellowstone, TX, Supplierzip', 'D008',
'InternationalSupplier'),
(('SUP009', '5678 Co', 901234568, '963 Supplier Hwy, Canyon, FL, Supplierzip', 'D009',
'InternationalSupplier'),
(('SUP010', '9012 Ltd', 123456890, '147 Supplier Dr, Hawaii, HW, Supplierzip', 'D010',
'InternationalSupplier');

INSERT INTO LocalSupplier (LocalSupplierID, StateTax)

VALUES

('SUP001', 5.00),
(('SUP002', 5.50),
(('SUP003', 6.00),
(('SUP004', 6.50),
(('SUP005', 7.00);

INSERT INTO InternationalSupplier (InternationalSupplierID, ImportExportTax)

VALUES

('SUP006', 12.50),
(('SUP007', 13.00),
(('SUP008', 13.50),
(('SUP009', 14.00),
(('SUP010', 14.50);

```
INSERT INTO Manufacturer (ManufacturerID, ManufacturerAddress, ManufacturerPhoneNo,  
ManufacturerName)
```

```
VALUES
```

```
('M001', '123 Main St, Anytown, USA', 123457890, 'ABC Manufacturing'),  
('M002', '456 Elm St, Othertown, USA', 234567801, 'XYZ Industries'),  
('M003', '789 Oak St, Anothertown, USA', 346789012, '123 Company'),  
('M004', '321 Maple St, Somewhere, USA', 456780123, '456 Corporation'),  
('M005', '654 Pine St, Nowhere, USA', 567801234, '789 Enterprises'),  
('M006', '987 Cedar St, Elsewhere, USA', 678912345, 'ABC&D Ltd'),  
('M007', '741 Birch St, Anywhere, USA', 789012356, 'XYZ Corp'),  
('M008', '852 Walnut St, Somewhither, USA', 890234567, '1234 Manufacturing'),  
('M009', '963 Sycamore St, There, USA', 901234678, '5678 Industries'),  
('M010', '147 Cherry St, Everywhere, USA', 123567890, '9012 Enterprises');
```

```
INSERT INTO Employee (employeeID, employeeFName, employeeLname, employeePhone,  
employeeCity, employeeState, employeeZip, employeeEmail, DOB, HiredDate, ManagerID,  
StoreID, ManufacturerID)
```

```
VALUES
```

```
('EMP001', 'John', 'Doe', '123-456-7890', 'New York', 'NY', '10001', 'john.doe@example.com',  
'1985-01-15', '2020-05-10', NULL, 'S001', 'M001'),  
('EMP002', 'Jane', 'Smith', '234-567-8901', 'Los Angeles', 'CA', '90001',  
'jane.smith@example.com', '1990-03-20', '2021-02-15', 'EMP001', 'S002', 'M002'),  
('EMP003', 'Michael', 'Johnson', '345-678-9012', 'Chicago', 'IL', '60601',  
'michael.johnson@example.com', '1988-07-12', '2019-11-30', 'EMP001', 'S003', 'M003'),  
('EMP004', 'Emily', 'Davis', '456-789-0123', 'Houston', 'TX', '77001',  
'emily.davis@example.com', '1993-04-25', '2020-09-05', 'EMP002', 'S004', 'M004'),  
('EMP005', 'William', 'Brown', '567-890-1234', 'Phoenix', 'AZ', '85001',  
'william.brown@example.com', '1995-10-08', '2022-03-12', 'EMP002', 'S005', 'M005'),  
('EMP006', 'Mary', 'Wilson', '678-901-2345', 'Philadelphia', 'PA', '19101',  
'mary.wilson@example.com', '1987-12-18', '2021-06-20', 'EMP003', 'S006', 'M006'),
```

```

('EMP007', 'James', 'Taylor', '789-012-3456', 'San Antonio', 'TX', '78201',
'james.taylor@example.com', '1991-09-30', '2018-08-25', 'EMP003', 'S007', 'M007'),
('EMP008', 'Jessica', 'Martinez', '890-123-4567', 'San Diego', 'CA', '92101',
'jessica.martinez@example.com', '1994-02-14', '2017-10-10', 'EMP004', 'S008', 'M008'),
('EMP009', 'David', 'Anderson', '901-234-5678', 'Dallas', 'TX', '75201',
'david.anderson@example.com', '1989-06-28', '2016-12-05', 'EMP005', 'S009', 'M009'),
('EMP010', 'Ashley', 'Thomas', '123-456-7890', 'San Jose', 'CA', '95101',
'ashley.thomas@example.com', '1992-11-05', '2023-01-18', 'EMP006', 'S010', 'M010');

```

INSERT INTO Product (productID, productDescription, UnitPrice, ProductName, SupplierID, StoreInventoryID, CompanyInventoryID, ManufacturerID)

VALUES

```

('P001', 'Cereal', 3.99, 'Corn Flakes', 'SUP001', 'SI001', 'CI001', 'M001'),
('P002', 'Apples', 1.99, 'Red Delicious', 'SUP002', 'SI002', 'CI002', 'M002'),
('P003', 'Milk', 2.49, 'Whole Milk', 'SUP003', 'SI003', 'CI003', 'M003'),
('P004', 'Beef', 9.99, 'Ribeye Steak', 'SUP004', 'SI004', 'CI004', 'M004'),
('P005', 'Bread', 2.29, 'White Bread', 'SUP005', 'SI005', 'CI005', 'M005'),
('P006', 'Ice Cream', 4.49, 'Vanilla', 'SUP006', 'SI006', 'CI006', 'M006'),
('P007', 'Soup', 1.79, 'Chicken Noodle', 'SUP007', 'SI007', 'CI007', 'M007'),
('P008', 'Dish Soap', 3.99, 'Dawn', 'SUP008', 'SI008', 'CI008', 'M008'),
('P009', 'Shampoo', 5.99, 'Pantene', 'SUP009', 'SI009', 'CI009', 'M009'),
('P010', 'Dog Food', 8.49, 'Pedigree', 'SUP010', 'SI010', 'CI010', 'M010');

```

INSERT INTO INVOICE (invoicenum, invoicedate)

VALUES

```

('INV001', '2024-04-01'),
('INV002', '2024-04-02'),
('INV003', '2024-04-03'),
('INV004', '2024-04-04'),

```

```
('INV005', '2024-04-05'),  
('INV006', '2024-04-06'),  
('INV007', '2024-04-07'),  
('INV008', '2024-04-08'),  
('INV009', '2024-04-09'),  
('INV010', '2024-04-10');
```

INSERT INTO SHIPMENT (shipmentID, shipmentdate, trackingnum)

VALUES

```
('SHIP001', '2024-04-01', '1234567890'),  
(('SHIP002', '2024-04-02', '2345678901'),  
(('SHIP003', '2024-04-03', '3456789012'),  
(('SHIP004', '2024-04-04', '4567890123'),  
(('SHIP005', '2024-04-05', '5678901234'),  
(('SHIP006', '2024-04-06', '6789012345'),  
(('SHIP007', '2024-04-07', '7890123456'),  
(('SHIP008', '2024-04-08', '8901234567'),  
(('SHIP009', '2024-04-09', '9012345678'),  
(('SHIP010', '2024-04-10', '0123456789');
```

INSERT INTO ORDERS (orderID, orderdate, customerID, invoicenum, shipmentID)

VALUES

```
('ORD001', '2024-04-01', 'C001', 'INV001', 'SHIP001'),  
(('ORD002', '2024-04-02', 'C002', 'INV002', 'SHIP002'),  
(('ORD003', '2024-04-03', 'C003', 'INV003', 'SHIP003'),  
(('ORD004', '2024-04-04', 'C004', 'INV004', 'SHIP004'),  
(('ORD005', '2024-04-05', 'C005', 'INV005', 'SHIP005'),
```

```
('ORD006', '2024-04-06', 'C006', 'INV006', 'SHIP006'),  
('ORD007', '2024-04-07', 'C007', 'INV007', 'SHIP007'),  
('ORD008', '2024-04-08', 'C008', 'INV008', 'SHIP008'),  
('ORD009', '2024-04-09', 'C009', 'INV009', 'SHIP009'),  
('ORD010', '2024-04-10', 'C010', 'INV010', 'SHIP010');
```

INSERT INTO Payment (paymentID, PaymentDate, PaymentAmount, invoicenum)

VALUES

```
('PAY001', '2024-04-01', 100.00, 'INV001'),  
('PAY002', '2024-04-02', 150.00, 'INV002'),  
('PAY003', '2024-04-03', 200.00, 'INV003'),  
('PAY004', '2024-04-04', 75.00, 'INV004'),  
('PAY005', '2024-04-05', 120.00, 'INV005'),  
('PAY006', '2024-04-06', 90.00, 'INV006'),  
('PAY007', '2024-04-07', 80.00, 'INV007'),  
('PAY008', '2024-04-08', 130.00, 'INV008'),  
('PAY009', '2024-04-09', 95.00, 'INV009'),  
('PAY010', '2024-04-10', 110.00, 'INV010');
```

INSERT INTO OrderDetails (orderID, productID, OrderQuantity)

VALUES

```
('ORD001', 'P001', '2'),  
('ORD002', 'P002', '5'),  
('ORD003', 'P003', '3'),  
('ORD004', 'P004', '1'),  
('ORD005', 'P005', '4'),  
('ORD006', 'P006', '2'),
```

```
('ORD007', 'P007', '3'),  
('ORD008', 'P008', '2'),  
('ORD009', 'P009', '1'),  
('ORD010', 'P010', '4');
```

INSERT INTO COMPANYINVENTORY (CompanyInventoryID, DepartmentID, CompanyQuantity)

VALUES

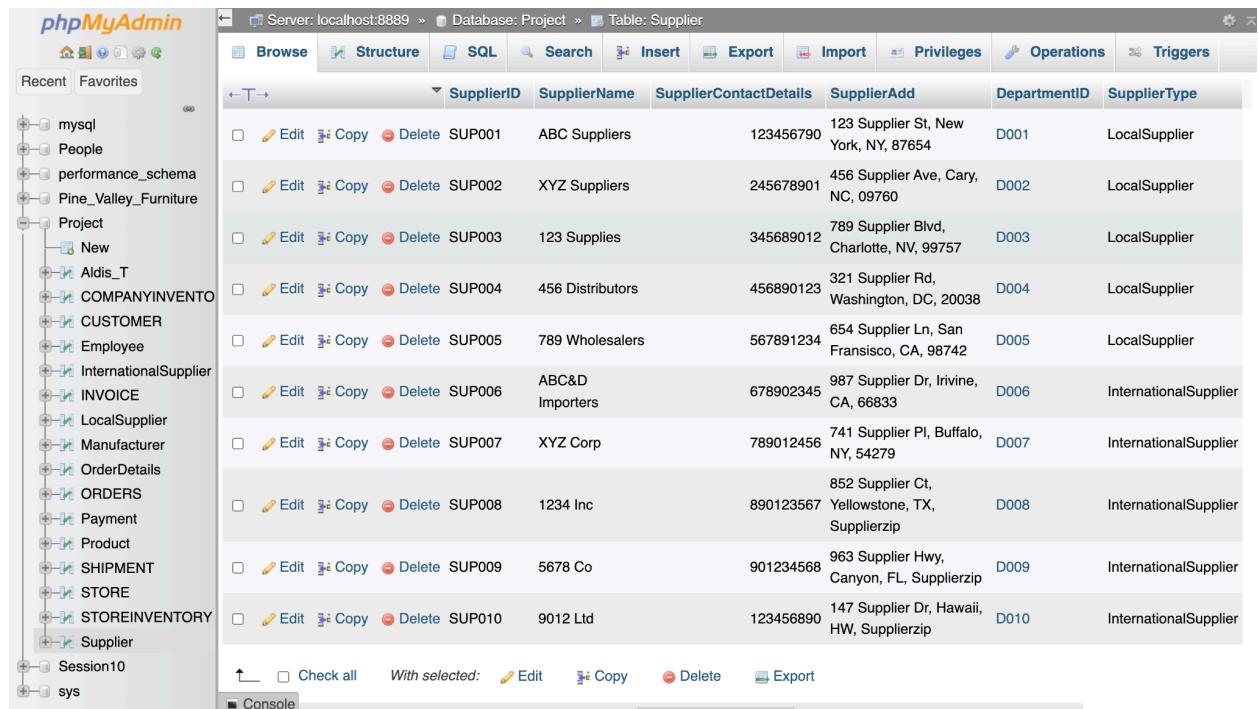
```
('CI001', 'D001', 100),  
('CI002', 'D002', 150),  
('CI003', 'D003', 200),  
('CI004', 'D004', 75),  
('CI005', 'D005', 120),  
('CI006', 'D006', 90),  
('CI007', 'D007', 80),  
('CI008', 'D008', 130),  
('CI009', 'D009', 95),  
('CI010', 'D010', 110);
```

INSERT INTO STOREINVENTORY (StoreInventoryID, StoreID, StoreQuantity)

VALUES

```
('SI001', 'S001', 50),  
('SI002', 'S002', 75),  
('SI003', 'S003', 100),  
('SI004', 'S004', 30),  
('SI005', 'S005', 60),  
('SI006', 'S006', 45),
```

('SI007', 'S007', 40),
 ('SI008', 'S008', 65),
 ('SI009', 'S009', 47),
 ('SI010', 'S010', 55);



The screenshot shows the phpMyAdmin interface for the 'Supplier' table in the 'Project' database. The table has columns: SupplierID, SupplierName, SupplierContactDetails, SupplierAdd, DepartmentID, and SupplierType. The data consists of 10 rows, each representing a supplier with a unique ID, name, address, department, and type. The 'SupplierType' column indicates whether the supplier is 'LocalSupplier' or 'InternationalSupplier'. The 'SupplierAdd' column includes detailed addresses for each supplier.

| | SupplierID | SupplierName | SupplierContactDetails | SupplierAdd | DepartmentID | SupplierType |
|--------------------------|----------------------------|------------------|------------------------|---|--------------|-----------------------|
| <input type="checkbox"/> | Edit Copy Delete SUP001 | ABC Suppliers | 123456790 | 123 Supplier St, New York, NY, 87654 | D001 | LocalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP002 | XYZ Suppliers | 245678901 | 456 Supplier Ave, Cary, NC, 09760 | D002 | LocalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP003 | 123 Supplies | 345689012 | 789 Supplier Blvd, Charlotte, NV, 99757 | D003 | LocalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP004 | 456 Distributors | 456890123 | 321 Supplier Rd, Washington, DC, 20038 | D004 | LocalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP005 | 789 Wholesalers | 567891234 | 654 Supplier Ln, San Fransisco, CA, 98742 | D005 | LocalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP006 | ABC&D Importers | 678902345 | 987 Supplier Dr, Irvine, CA, 66833 | D006 | InternationalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP007 | XYZ Corp | 789012456 | 741 Supplier Pl, Buffalo, NY, 54279 | D007 | InternationalSupplier |
| <input type="checkbox"/> | | | | 852 Supplier Ct, Yellowstone, TX, Supplierzip | D008 | InternationalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP009 | 1234 Inc | 890123567 | 963 Supplier Hwy, Canyon, FL, Supplierzip | D009 | InternationalSupplier |
| <input type="checkbox"/> | Edit Copy Delete SUP010 | 5678 Co | 901234568 | 147 Supplier Dr, Hawaii, HW, Supplierzip | D010 | InternationalSupplier |
| <input type="checkbox"/> | | | | 123456890 | | |

phpMyAdmin

Server: localhost:8889 » Database: Project » Table: Product

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 9 (10 total, Query took 0.0003 seconds.)

SELECT * FROM `Product`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

| productID | productDescription | UnitPrice | ProductName | SupplierID | StoreInventoryID | CompanyInventoryID |
|-----------|--------------------|-----------|----------------|------------|------------------|--------------------|
| P001 | Cereal | 3.99 | Corn Flakes | SUP001 | SI001 | CI001 |
| P002 | Apples | 1.99 | Red Delicious | SUP002 | SI002 | CI002 |
| P003 | Milk | 2.49 | Whole Milk | SUP003 | SI003 | CI003 |
| P004 | Beef | 9.99 | Ribeye Steak | SUP004 | SI004 | CI004 |
| P005 | Bread | 2.29 | White Bread | SUP005 | SI005 | CI005 |
| P006 | Ice Cream | 4.49 | Vanilla | SUP006 | SI006 | CI006 |
| P007 | Soup | 1.79 | Chicken Noodle | SUP007 | SI007 | CI007 |
| P008 | Dish Soap | 3.99 | Dawn | SUP008 | SI008 | CI008 |
| P009 | Shampoo | 5.99 | Pantene | SUP009 | SI009 | CI009 |
| P010 | Dog Food | 8.49 | Pedigree | SUP010 | SI010 | CI010 |

Check all With selected: Edit Copy Delete Export

Console

phpMyAdmin

Server: localhost:8889 » Database: Project » Table: ORDERS

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 9 (10 total, Query took 0.0002 seconds.)

SELECT * FROM `ORDERS`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

| orderID | orderdate | customerID | invoicenum | shipmentID |
|---------|------------|------------|------------|------------|
| ORD001 | 2024-04-01 | C001 | INV001 | SHIP001 |
| ORD002 | 2024-04-02 | C002 | INV002 | SHIP002 |
| ORD003 | 2024-04-03 | C003 | INV003 | SHIP003 |
| ORD004 | 2024-04-04 | C004 | INV004 | SHIP004 |
| ORD005 | 2024-04-05 | C005 | INV005 | SHIP005 |
| ORD006 | 2024-04-06 | C006 | INV006 | SHIP006 |
| ORD007 | 2024-04-07 | C007 | INV007 | SHIP007 |
| ORD008 | 2024-04-08 | C008 | INV008 | SHIP008 |
| ORD009 | 2024-04-09 | C009 | INV009 | SHIP009 |
| ORD010 | 2024-04-10 | C010 | INV010 | SHIP010 |

Check all With selected: Edit Copy Delete Export

Console

The screenshot shows the phpMyAdmin interface for the CUSTOMER table in the Project database. The table has 10 rows of data:

| customerID | customername | customerphone | customeremail | customercity | customerstate | customerzip |
|------------|------------------|---------------|---------------------|--------------|---------------|-------------|
| C001 | John Doe | 123456789 | john@example.com | New York | NY | 10001 |
| C002 | Jane Smith | 234568901 | jane@example.com | Los Angeles | CA | 90001 |
| C003 | Michael Johnson | 346789012 | michael@example.com | Chicago | IL | 60601 |
| C004 | Emily Davis | 456789123 | emily@example.com | Houston | TX | 77001 |
| C005 | William Brown | 567801234 | william@example.com | Phoenix | AZ | 85001 |
| C006 | Mary Wilson | 678901345 | mary@example.com | Philadelphia | PA | 19101 |
| C007 | James Taylor | 789013456 | james@example.com | San Antonio | TX | 78201 |
| C008 | Jessica Martinez | 890134567 | jessica@example.com | San Diego | CA | 92101 |
| C009 | David Anderson | 901234578 | david@example.com | Dallas | TX | 75201 |
| C010 | Ashley Thomas | 123457890 | ashley@example.com | San Jose | CA | 95101 |

3.3 Select Queries

3.3.1 Select query to find the Total quantity of Pedigree available:

```
SELECT SUM(si.StoreQuantity) AS TotalQuantity
FROM StoreInventory si
JOIN Product p ON si.StoreInventoryID = p.StoreInventoryID
WHERE p.ProductName = 'Pedigree';
```

```
SELECT SUM(si.StoreQuantity) AS TotalQuantity FROM StoreInventory si JOIN Product p ON si.StoreInventoryID =
p.StoreInventoryID WHERE p.ProductName = 'Pedigree';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

TotalQuantity

55

3.3.2 Select query to display all order details with customer, product, and payment information:

```

SELECT o.orderID, c.customername, p.ProductName, od.OrderQuantity, pay.PaymentAmount,
pay.PaymentDate

FROM ORDERS o

JOIN CUSTOMER c ON o.customerID = c.customerID

JOIN OrderDetails od ON o.orderID = od.orderID

JOIN Product p ON od.productID = p.productID

LEFT JOIN Payment pay ON o.invoicenum = pay.invoicenum;

```

| orderID | customername | ProductName | OrderQuantity | PaymentAmount | PaymentDate |
|---------|------------------|----------------|---------------|---------------|-------------|
| ORD001 | John Doe | Corn Flakes | 2 | 100.00 | 2024-04-01 |
| ORD002 | Jane Smith | Red Delicious | 5 | 150.00 | 2024-04-02 |
| ORD003 | Michael Johnson | Whole Milk | 3 | 200.00 | 2024-04-03 |
| ORD004 | Emily Davis | Ribeye Steak | 1 | 75.00 | 2024-04-04 |
| ORD005 | William Brown | White Bread | 4 | 120.00 | 2024-04-05 |
| ORD006 | Mary Wilson | Vanilla | 2 | 90.00 | 2024-04-06 |
| ORD007 | James Taylor | Chicken Noodle | 3 | 80.00 | 2024-04-07 |
| ORD008 | Jessica Martinez | Dawn | 2 | 130.00 | 2024-04-08 |
| ORD009 | David Anderson | Pantene | 1 | 95.00 | 2024-04-09 |
| ORD010 | Ashley Thomas | Pedigree | 4 | 110.00 | 2024-04-10 |

3.3.3 Select query to Retrieve product stock:

```

SELECT p.productName, si.StoreQuantity AS StoreStock, ci.CompanyQuantity AS
CompanyStock

FROM Product p

LEFT JOIN StoreInventory si ON p.StoreInventoryID = si.StoreInventoryID

LEFT JOIN COMPANYINVENTORY ci ON p.CompanyInventoryID =
ci.CompanyInventoryID;

```

| productName | StoreStock | CompanyStock |
|----------------|------------|--------------|
| Corn Flakes | 50 | 100 |
| Red Delicious | 75 | 150 |
| Whole Milk | 100 | 200 |
| Ribeye Steak | 30 | 75 |
| White Bread | 60 | 120 |
| Vanilla | 45 | 90 |
| Chicken Noodle | 40 | 80 |
| Dawn | 65 | 130 |
| Pantene | 47 | 95 |
| Pedigree | 55 | 110 |

References

- Aldi: no frills grocery shopping - Technology and Operations Management.* (2015, December 6). Technology and Operations Management.
<https://d3.harvard.edu/platform-rctom/submission/aldi-no-frills-grocery-shopping/>
- ALDI Nord - Über uns.* (n.d.).
https://web.archive.org/web/20140228080204/http://www.aldi-nord.de/aldi_ueber_uns_814.html
- Behr, F. (2021, March 19). *The reason people have been disappointed with Aldi recently.* Mashed.
<https://www.mashed.com/360837/the-reason-people-have-been-disappointed-with-aldi-recently/>
- Behr, F. (2022, January 25). *Aldi customers have noticed that “Tons” of these items aren’t in stock.* Mashed.
<https://www.mashed.com/746060/aldi-customers-have-noticed-that-tons-of-these-items-arent-in-stock/>
- Melanie. (2023, December 21). *How to Improve Inventory Management: 10 Effective Strategies.* Unleashed Software.
<https://www.unleashedsoftware.com/blog/9-critical-strategies-inventory-management>
- Ngo, H. (2023, January 27). *The real difference between the two different Aldis.* Mashed.
<https://www.mashed.com/226743/the-real-difference-between-the-two-different-aldis/>
- Wanyoike-Kimiti, P. (2023, December 16). *Secrets behind Aldi’s innovative supply chain.* Mashed. <https://www.mashed.com/1460508/aldi-supply-chain/>