Supply Chain and Logistics Data Model Explanation

Overview

The Supply Chain and Logistics Management System database has been designed to organize, track, and analyze business operations from order placement to product delivery.
The system manages data about departments, categories, products, customers, orders, and individual order line items. The goal is to ensure smooth operations across multiple business functions, from product classification to customer transactions, while maintaining data integrity, normalization, and analytical efficiency.

Core Entities (5) and Junction Entity (1)

1. Departments

The Departments entity represents the top-level business divisions of the organization, such as Electronics, Apparel, or Fitness.
Each department groups related categories of products to provide an organized structure for reporting and management.

- Primary Key: department_id
- Attributes: department_name
- Relationship: One department can have many categories (1:M).
  → Each category must belong to exactly one department.

2. Categories

The Categories entity further classifies products under specific departments. It enables product segmentation based on functionality or target market (e.g., “Mobile Devices,” “Sporting Goods”).

- Primary Key: category_id
- Foreign Key: department_id → references Departments
- Attributes: category_name
- Relationship:
  - Many-to-One (M:1) with Departments

    - One-to-Many (1:M) with Products

Each category can include multiple products, but every product must belong to one category.

3. Products

The Products entity contains detailed information about the items available for sale or shipment. It includes attributes such as product name, price, and description, enabling price tracking, performance analysis, and category-based sales insights.

- Primary Key: product_card_id
- Foreign Key: product_category_id → references Categories
- Attributes: product_name, product_price, product_status, product_description, product_image
- Relationship:
    - Many-to-One (M:1) with Categories
    - One-to-Many (1:M) with Order_Items

Products are linked to categories for classification and to order items to record which products were sold.

4. Customers

The Customers entity stores data about buyers who place orders. It includes personal and regional details, allowing the company to identify customer segments, regional sales patterns, and delivery locations.

- Primary Key: customer_id
- Attributes: customer_fname, customer_lname, customer_email, customer_city, customer_state, customer_country, customer_zipcode, market, region
- Relationship: One-to-Many (1:M) with Orders

Each customer can place multiple orders, but each order belongs to one customer.

5. Orders

The Orders entity represents transactions made by customers. It records details such as order date, shipping method, delivery status, and performance metrics like profit or benefit per order. It forms the backbone of the transactional system, linking customers to products through order items.

- Primary Key: order_id
- Foreign Key: customer_id → references Customers

- Attributes:
  order_date, order_status, order_city, order_state, order_country, order_zipcode, order_region, shipping_date, shipping_mode, delivery_status, late_delivery_risk, days_for_shipping_real, days_for_shipment_scheduled, benefit_per_order, sales_per_customer, order_profit_per_order
- Relationship:
  - Many-to-One (M:1) with Customers
  - One-to-Many (1:M) with Order_Items

6. Order_Items (Junction Entity)

The Order_Items table is the junction entity that resolves the many-to-many relationship between Orders and Products.
Each row in this table represents one line item in an order, specifying which product was ordered, in what quantity, and at what price.

- Primary Key: order_item_id
- Foreign Keys:
  - order_id → references Orders
  - product_card_id → references Products
- Attributes:
  order_item_cardprod_id, order_item_quantity, order_item_product_price, order_item_discount, order_item_discount_rate, order_item_total, sales, profit, order_item_profit_ratio
- Relationship:
  - Many-to-One (M:1) with Orders
  - Many-to-One (M:1) with Products
- Purpose:
  - Enables accurate tracking of individual product sales within each order.
  - Contains financial metrics that support profitability and performance reporting.

Q - Why Orders ↔ Products is Many-to-Many:

In a business like SwiftLogix (my supply chain system):

- A single order often includes multiple products (for example, one customer orders a smartwatch, headphones, and a yoga mat together).

- A single product can appear in multiple orders (for example, “Smartwatch” is purchased by many customers in different orders).

So, both sides can have multiple associations, this is the definition of a many-to-many (M:N) relationship.

For Example:

Products Table

product_card_id	**product_name**	**product_price**
101	Smartwatch	327.75
102	Yoga Mat	45.50
103	Wireless Earbuds	89.99

Orders Table

order_id	**customer_id**	**order_date**	**shipping_mode**
5001	200	2025-10-12	Standard Class
5002	201	2025-10-14	Second Class
5003	202	2025-10-15	Same Day

Q - What if we directly connected Orders ↔ Products:

We’d have to add multiple product IDs into one order record, like:

order_id	**product_card_ids**
5001	101,102,103

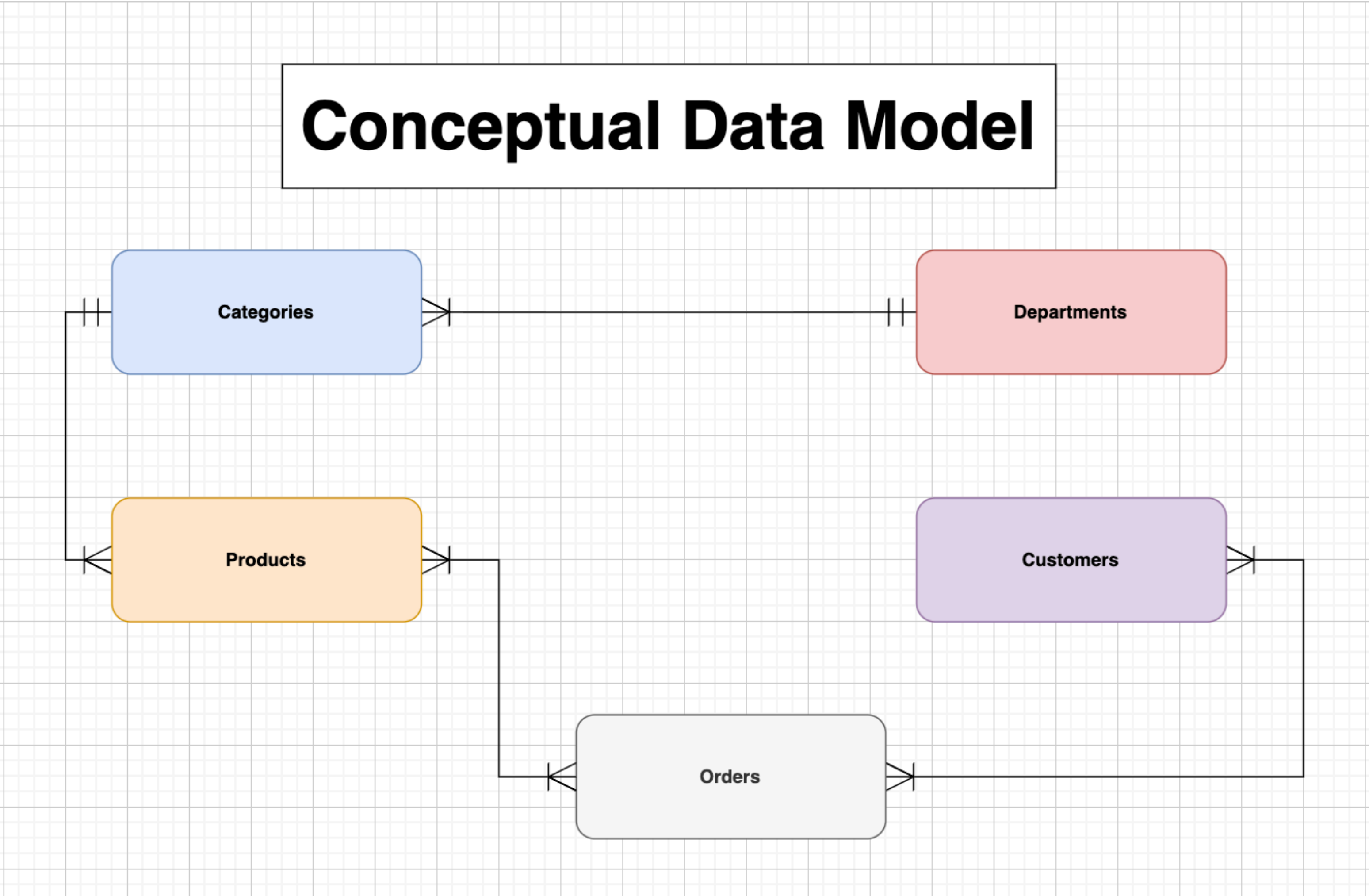
But relational databases don’t allow multi-valued attributes, this violates First Normal Form (1NF). Also, if “Smartwatch” appears again in order 5002, I’d be duplicating the product data, which breaks data integrity and 3NF. So, we use a Junction Table: “Order_Items”

The junction table (Order_Items) transforms the many-to-many into two one-to-many relationships:

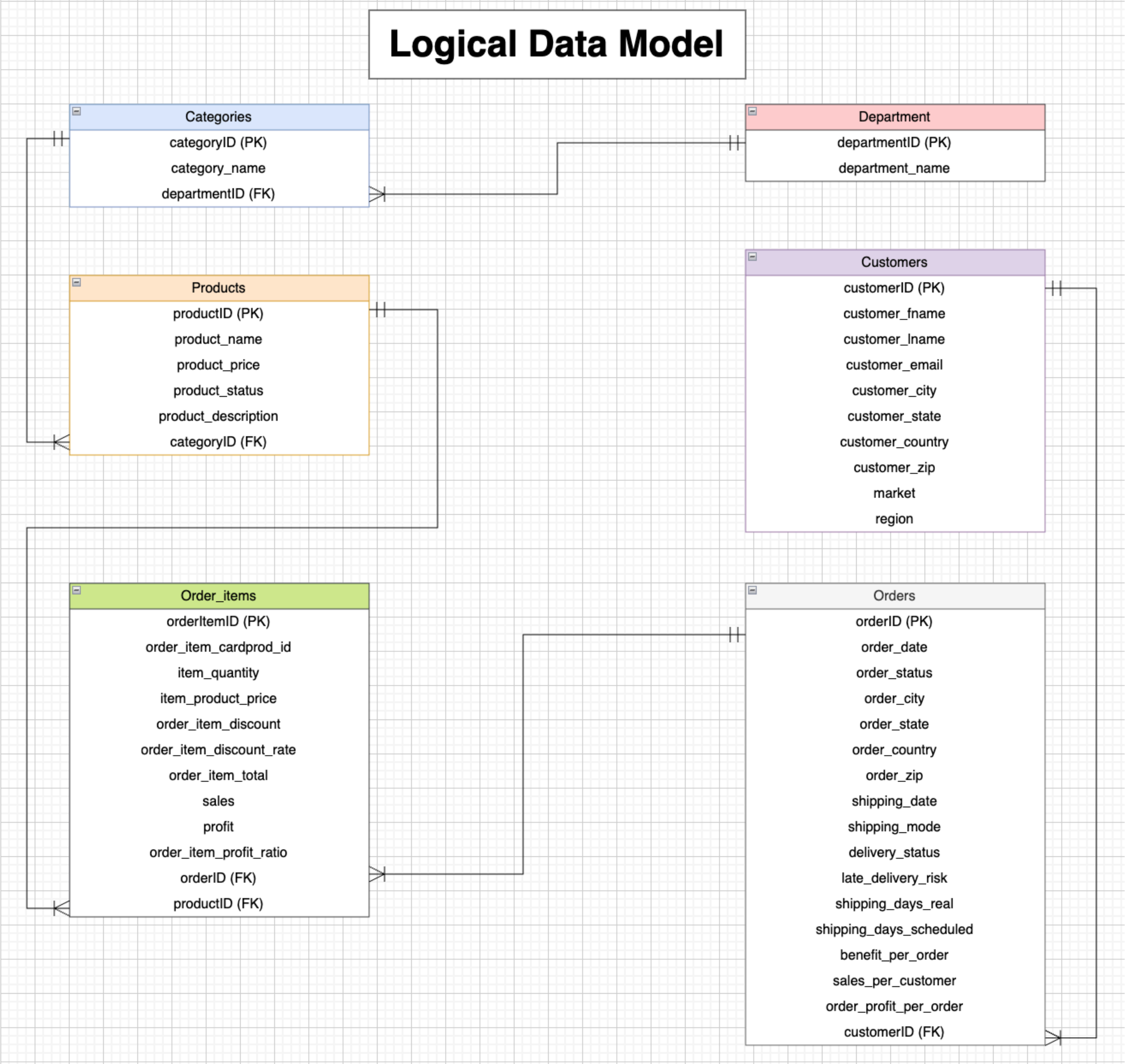
- Orders (1) → (M) Order_Items

- Products (1) → (M) Order_Items

Conceptual Data Model



Logical Data Model



Logical Data Model

Categories
categoryID (PK)
category_name
departmentID (FK)

Department
departmentID (PK)
department_name

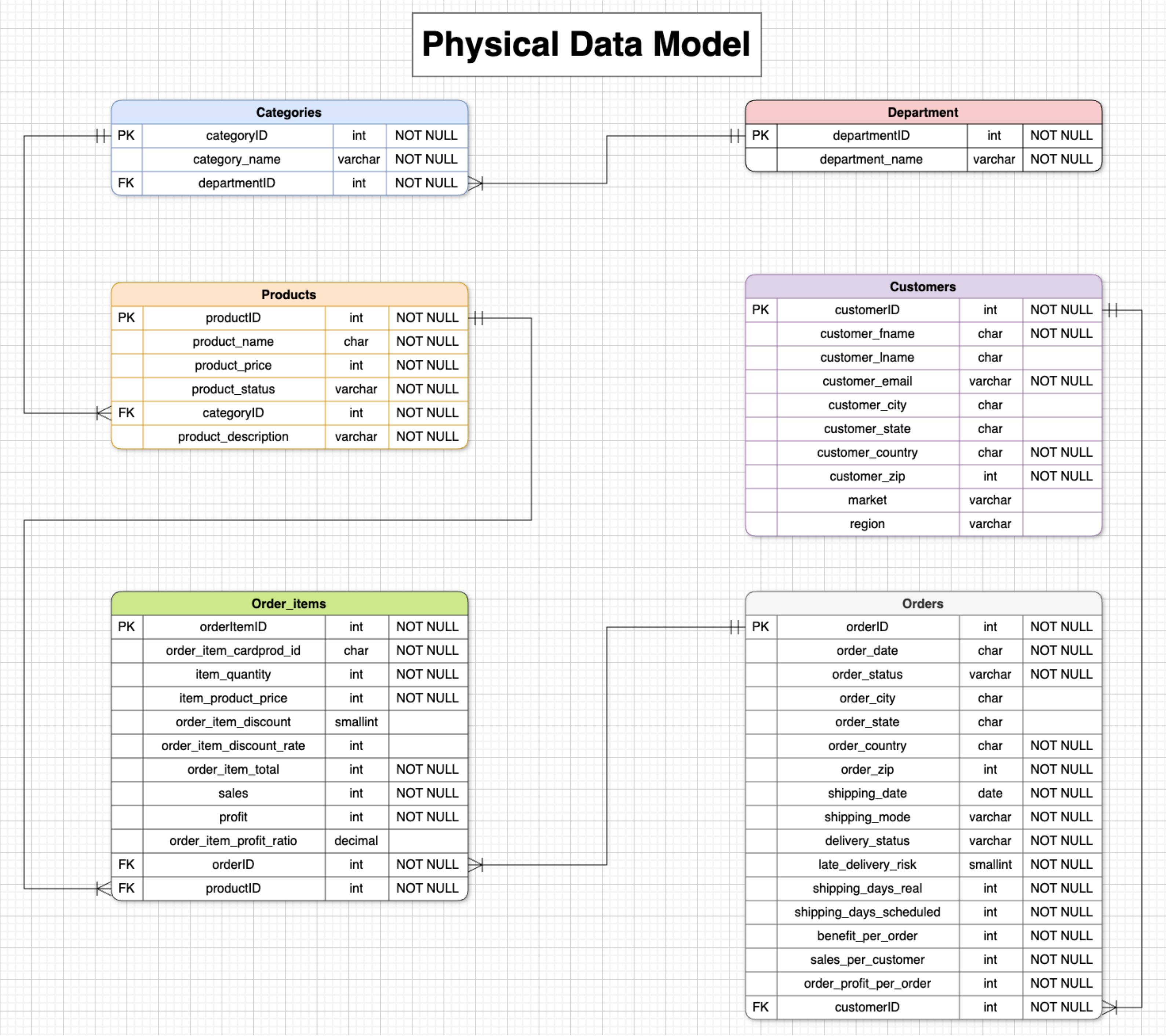
Products
productID (PK)
product_name
product_price
product_status
product_description
categoryID (FK)

Customers
customerID (PK)
customer_fname
customer_lname
customer_email
customer_city
customer_state
customer_country
customer_zip
market
region

Order_items
orderItemID (PK)
order_item_cardprod_id
item_quantity
item_product_price
order_item_discount
order_item_discount_rate
order_item_total
sales
profit
order_item_profit_ratio
orderID (FK)
productID (FK)

Orders
orderID (PK)
order_date
order_status
order_city
order_state
order_country
order_zip
shipping_date
shipping_mode
delivery_status
late_delivery_risk
shipping_days_real
shipping_days_scheduled
benefit_per_order
sales_per_customer
order_profit_per_order
customerID (FK)

Physical Data Model



Physical Data Model

Categories

PK	categoryID	int	NOT NULL
	category_name	varchar	NOT NULL
FK	departmentID	int	NOT NULL

Department

PK	departmentID	int	NOT NULL
	department_name	varchar	NOT NULL

Products

PK	productID	int	NOT NULL
	product_name	char	NOT NULL
	product_price	int	NOT NULL
	product_status	varchar	NOT NULL
FK	categoryID	int	NOT NULL
	product_description	varchar	NOT NULL

Customers

PK	customerID	int	NOT NULL
	customer_fname	char	NOT NULL
	customer_lname	char	
	customer_email	varchar	NOT NULL
	customer_city	char	
	customer_state	char	
	customer_country	char	NOT NULL
	customer_zip	int	NOT NULL
	market	varchar	
	region	varchar	

Order_items

PK	orderItemID	int	NOT NULL
	order_item_cardprod_id	char	NOT NULL
	item_quantity	int	NOT NULL
	item_product_price	int	NOT NULL
	order_item_discount	smallint	
	order_item_discount_rate	int	
	order_item_total	int	NOT NULL
	sales	int	NOT NULL
	profit	int	NOT NULL
	order_item_profit_ratio	decimal	
FK	orderID	int	NOT NULL
FK	productID	int	NOT NULL

Orders

PK	orderID	int	NOT NULL
	order_date	char	NOT NULL
	order_status	varchar	NOT NULL
	order_city	char	
	order_state	char	
	order_country	char	NOT NULL
	order_zip	int	NOT NULL
	shipping_date	date	NOT NULL
	shipping_mode	varchar	NOT NULL
	delivery_status	varchar	NOT NULL
	late_delivery_risk	smallint	NOT NULL
	shipping_days_real	int	NOT NULL
	shipping_days_scheduled	int	NOT NULL
	benefit_per_order	int	NOT NULL
	sales_per_customer	int	NOT NULL
	order_profit_per_order	int	NOT NULL
FK	customerID	int	NOT NULL