

# Retail Inventory Management & Forecasting

## Project Overview

The **Retail Inventory Management and Forecasting** project aims to improve inventory management and demand forecasting using the **AdventureWorks dataset**. This project consists of three main components:

1. **ERD Schema:** A visual representation of the entities and relationships within the retail inventory data.
2. **SQL Tables and Queries:** The design of SQL tables to manage data and SQL queries to extract meaningful insights.
3. **Data Warehouse Design:** A structured design of the data warehouse with fact and dimension tables to support efficient data analysis.

The project uses **SQL Server** for database management, **Azure Data Factory** for ETL processes, and **Python** for machine learning-based demand forecasting.

## Key Components

### 1. ERD Schema

The **Entity-Relationship Diagram (ERD)** visually represents the structure of the retail inventory system. The ERD includes entities such as products, suppliers, sales, and inventory levels, and the relationships between them. The key relationships are as follows:

- **Products** are linked to **Suppliers**.
- **Products** are linked to **Sales** through product IDs.
- **Products** are linked to **Inventory Levels** to track stock levels.

This schema helps in understanding how data is organized and how different entities interact within the system.

## 2. SQL Tables and Queries

The **SQL tables** are designed to store and manage the data necessary for efficient inventory management and forecasting. Below are some examples:

### **SQL Tables:**

1. **Products:** Stores product details such as ProductID, ProductName, and Category.
2. **Suppliers:** Stores supplier details like SupplierID, SupplierName, and Location.
3. **Sales:** Records sales transactions with fields such as SalesID, ProductID, QuantitySold, and TotalAmount.
4. **InventoryLevels:** Tracks current inventory with fields like ProductID, StockLevel, and LastUpdated.

### **SQL Queries:**

- **Query 1: Top-Selling Products**

```
SELECT ProductName, SUM(QuantitySold) AS TotalSold  
  
FROM Sales  
  
JOIN Products ON Sales.ProductID = Products.ProductID  
  
GROUP BY ProductName  
  
ORDER BY TotalSold DESC
```

This query identifies the top-selling products based on the total quantity sold.

- **Query 2: Low Stock Alerts**

```
SELECT ProductName, InventoryLevel  
  
FROM Products
```

```
JOIN InventoryLevels ON Products.ProductID = InventoryLevels.ProductID  
  
WHERE InventoryLevel < 10
```

This query identifies products that have stock levels below a specified threshold.

- **Query 3: Monthly Revenue**

```
SELECT YEAR(SalesDate) AS Year, MONTH(SalesDate) AS Month,  
SUM(TotalAmount) AS TotalRevenue  
  
FROM Sales  
  
GROUP BY YEAR(SalesDate), MONTH(SalesDate)  
  
ORDER BY Year, Month
```

This query calculates the total revenue generated each month.

### 3. Data Warehouse Design

The **data warehouse design** is structured to support efficient data analysis by creating fact and dimension tables. The design is as follows:

- **FactSales:** A fact table that stores transactional sales data, including SalesID, ProductID, QuantitySold, and TotalRevenue.
- **DimProducts:** A dimension table that stores product details like ProductID, ProductName, and Category.
- **DimSuppliers:** A dimension table that stores supplier information like SupplierID, SupplierName, and Location.
- **DimInventoryLevel:** A dimension table that tracks inventory levels with fields like ProductID and InventoryLevel.

The **fact tables** store transactional data, while the **dimension tables** store descriptive information to allow efficient querying and reporting.

# Setup and Installation

## Prerequisites

Before running the project, ensure the following tools are installed:

1. **Python 3.7+**
2. **SQL Server** (or a compatible database for the AdventureWorks dataset)
3. **Azure Data Factory** (optional, for ETL)
4. **Azure Machine Learning** (optional, for model evaluation)

## Installation Steps

1. Install the required Python dependencies:

bash

Copy code

```
pip install -r requirements.txt
```

2. Set up your **SQL Server** with the AdventureWorks dataset.
3. Run the ETL pipeline:

bash

Copy code

```
python Python/etl_pipeline.py
```

4. Run the demand forecasting model:

bash

Copy code

```
python Python/demand_forecasting_model.py
```

## Conclusion

The **Retail Inventory Management and Forecasting** project provides a comprehensive solution for managing inventory and forecasting demand using data-driven techniques. The project is built on a solid foundation, with well-designed **ERD schemas**, **SQL tables**, and

**data warehouse architecture** to ensure efficient data management and analysis. The integration of machine learning models further enhances forecasting capabilities, making it an effective tool for retail businesses.