Retail Database Management

Statement:
To develop a comprehensive retail database management system that empowers managers to efficiently oversee employee operations, optimize inventory control, monitor revenue across multiple stores, and enhance customer satisfaction. By tracking sales trends and understanding customer needs, the database will enable informed decision-making and drive sustainable business growth.

Objectives:

1. Employee Management
   - Track employee details such as roles, departments, and work schedules across different store locations.
   - Facilitate performance monitoring and payroll management.
2. Inventory Control
   - Monitor stock levels and maintain optimal inventory as per product demand across multiple stores.
   - Send automated alerts when inventory levels hit the buffer line to avoid stockouts.
   - Support inventory transfers between stores when needed.
3. Revenue Tracking
   - Provide detailed revenue reports for individual stores and the business as a whole.
   - Track sales by product category to identify high-performing and low-performing items.
4. Customer Insights
   - Record and analyze customer preferences, purchase history, and feedback to identify trends.
   - Enable targeted marketing and personalized promotions to boost sales.
5. Sales Trend Analysis
   - Analyze historical sales data to predict future demand and seasonality.
   - Support managers in making data-driven decisions for sales growth and product development.
6. Business Reporting
   - Generate reports for managers to review key performance indicators (KPIs) like sales, revenue, inventory turnover, and customer satisfaction.

Technical Architecture:

1. **Technologies Used:**
   - ○ Specify the tools and technologies employed, such as MSSQL, Python, and Power BI, to build the database and analytical functionalities.
2. **Database System Layered Architecture:**
   - ○ Describe the layered structure of the database system, including the data layer, business layer, and application layer.

Main Tables and Relationship Design:

1. **Entity-Relationship Diagram (ERD):**
   - Create an ERD showcasing the relationships between entities such as Employees, Products, Customers, and Orders.
2. **Table Structure Details:**
   - Provide the structural details of each table, including column names, data types, and constraints.

Core Features and Operational Processes:

- Provide a more detailed description of the functionalities, such as how automated inventory alerts work or how personalized promotional suggestions are generated.

Testing and Validation:

1. **Testing Plan:**
   - Outline a plan to validate database queries and functionalities (e.g., ensuring inventory alerts are sent correctly or revenue reports are accurate).

Potential Future Expansions:

1. **Future Features:**
   - Suggest features that could be implemented in the future, such as a cross-platform mobile application or real-time business alert system.

GitHub URL: https://github.com/RahulReddyJanekunte/DMDD_Final_Project_Group_9

Team Members:

1. **Rahul Reddy Janekunte**
2. **Jiaqi Yu**
3. **Madhumitha Nandhikatti**
4. **Yuan Chang Lee**
5. **Jieyu Zheng**