PIZZACO Database Managment Software

Developed by Aryan Merchant, Yvan Ngah

Georgia Highlands College

Overview

Pizzaco currently uses a paper-based record system, which may lead to several challenges such as difficulties in tracking customer orders and inaccurate inventory management. Additionally, this system restricts the scalability of the database, highlighting the need for a more robust and efficient database system. John has sought out a team of database architects to design a system that works around his current business model.

Objective

The objective of this project is to develop a centralized and secure database system for PIZZACO that efficiently manages customer orders, reduces manual efforts, maintains accurate records, and ultimately results in cost savings and increased profitability.

Methodology (DDL)

Customer(CustNum, CustFName, CustLName, Phone, Email, DeliveryStreetAddr, State, City, ZCode)

Employee(EmpID, EmpLName, EmpFName, EmpPhone, EmpEmail, EmpStreetAddr, EmpState, EmpCity, EmpZCode, EmpType, EmpHourRate)

Orders(OrderNum, OrderDte, Delivered, CustID, DeliveryEmp, OrderEmp)
FK: CustID → Customer
DeliveryEmp → Employee
OrderEmp → Employee

Item(ItemID, ItemName, ItemCat, Size, ItemCost, ItemPrice)

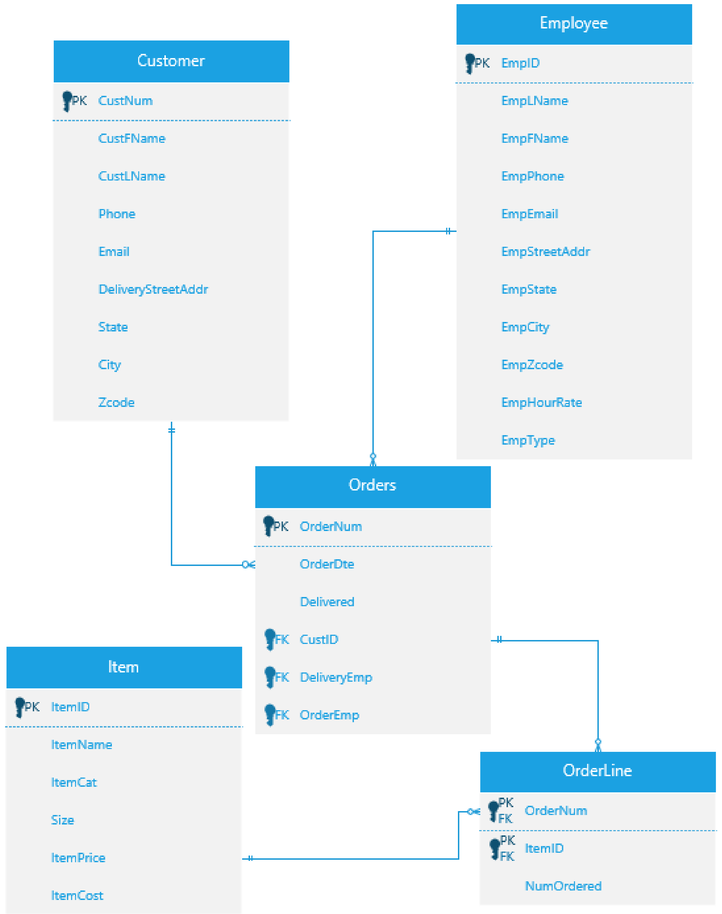
OrderLine(OrderNum, ItemID, NumOrdered)
FK: OrderID → Orders
ItemID → Item

Design Concepts

- MySQL
- MongoDB
- Oracle Database
- MS Access
- Java
- Python
- R
- PHP

The team decided to use MySQL and Java to develop the database and software, based on their flexibility and cost-effectiveness for the Pizzaco database.

ERD of PIZZACO Database



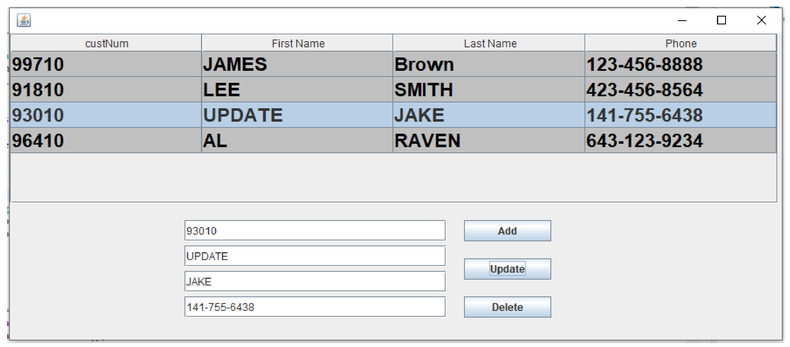
Conclusion and Future Direction

The creation of a centralized and secure database system for PIZZACO will enable the store to improve its efficiency, maintain accurate records of customer orders, and ultimately result in cost savings and increased profitability.

Future direction:

- As John's business expands, he could consider incorporating an inventory and supplier table into his database system, allowing him to streamline data and avoid redundancy.
- To help secure the database, John could implement security measures such as password protection, user access control, data encryption, regular data backups, limited access to authorized personnel, and a system of regular password updates.
- John could improve the user interface by creating a user-friendly design with clear instructions for data entry and retrieval, incorporating graphical representations of data, and offering options for customization and personalization.

GUI Software of The PIZZACO Database



Scan for Source Code!