Project Plan

Project Title: Online Shoe Store Database Application

Project Overview:

This project aims to design and implement a robust database application for an online shoe store. The application will handle various aspects of the business, including customer management, inventory tracking, order processing, employee management, and shipping logistics.

Project Objectives:

1. Database Design:

* Create a comprehensive ER schema diagram to represent the conceptual model.
* Transform the conceptual model into a relational model in 3NF.
* Identify and define entities, attributes, relationships, and constraints.

2. SQL Script Development:

* Write DDL scripts for creating database tables, relationships, and constraints.
* Develop DML scripts to populate the database with sample data.

3. Database Implementation:

* Execute SQL scripts to create the database structure.
* Populate the database with sample data to facilitate testing and development.

4. Query Development and Testing:

* Create a set of 12 queries covering single table queries, multiple table joins, aggregate functions, subqueries, ORDER BY, and HAVING clauses.
* Test queries for accuracy, efficiency, and performance.
* Screen Shots for each Query ran located on pages 6-19

5. Documentation:

* Generate project documentation, including ER schema diagrams, relational models, and SQL scripts.
* Document the database schema, relationships, and constraints.

Conclusion:

This project plan outlines the essential tasks and objectives for successfully developing and deploying the online shoe store database application.

Conceptual Model Document (ER Schema Diagram):

Entities:

* Customers

Attributes: CustomerID (PK), FirstName, LastName, Email, Phone

* Inventory

Attributes: ItemID (PK), Brand, Model, Size, Price, QuantityInStock

Relationships: OrderID (FK)

* Orders

Attributes: OrderID (PK), CustomerID (FK), EmployeeID (FK), ShippingID (FK), OrderDate

Relationships: CustomerID (references Customers), EmployeeID (references Employees), ShippingID (references Shipping)

* Employees

Attributes: EmployeeID (PK), FirstName, LastName, Position

* Shipping

Attributes: ShippingID (PK), ShipDate, DeliveryStatus

A diagram of a flowchart

Description automatically generated

The Relational Model in Third Normal Form (3NF) for the online shoe store application, derived from the conceptual model, involves defining tables and relationships that are free from certain types of redundancy.

Customers Table:

CustomerID (Primary Key)

FirstName

LastName

Email

Phone

Employees Table:

EmployeeID (Primary Key)

FirstName

LastName

Position

Inventory Table:

ItemID (Primary Key)

Brand

Model

Size

Price

QuantityInStock

Shipping Table:

ShippingID (Primary Key)

ShipDate

DeliveryStatusOrders Table:

OrderID (Primary Key)

CustomerID (Foreign Key)

EmployeeID (Foreign Key)

ShippingID (Foreign Key)

OrderDate

Relationships:

Customers to Orders (One-to-Many):

* Each customer can place multiple orders, but each order belongs to one customer.
* Foreign Key: CustomerID in Orders refers to CustomerID in Customers.

Orders to Inventory (One-to-Many):

* An order can consist of multiple inventory items, but each item belongs to one order.

Orders to Employees (Many-to-One):

* Each order is associated with one employee who processed it, but an employee can handle multiple orders.
* Foreign Key: EmployeeID in Orders refers to EmployeeID in Employees.

Orders to Shipping (Many-to-One):

* Each order is associated with one shipping record, but a shipping record can be associated with multiple orders (for tracking purposes).
* Foreign Key: ShippingID in Orders refers to ShippingID in Shipping.

Explanations:

* The primary keys are denoted with (Primary Key), and foreign keys are marked with (Foreign Key).
* Relationships are defined based on the cardinality and business rules.
* The model is designed to minimize redundancy and maintain data integrity.

This relational model serves as a foundation for creating the corresponding SQL script that can be used to create the database schema. The relationships ensure that data is organized efficiently and without unnecessary duplication, supporting the normalization principles of a relational database.

Implementation:  
**LPorter\_cis556\_schema.sql**  
 Create Tables

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(20)

);

CREATE TABLE Inventory (

ItemID INT PRIMARY KEY,

Brand VARCHAR(50),

Model VARCHAR(50),

Size VARCHAR(10),

Price DECIMAL(10, 2),

QuantityInStock INT

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

EmployeeID INT,

ShippingID INT,

OrderDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),

FOREIGN KEY (ShippingID) REFERENCES Shipping(ShippingID)

);

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Position VARCHAR(50)

);

CREATE TABLE Shipping (

ShippingID INT PRIMARY KEY,

ShipDate DATE,

DeliveryStatus VARCHAR(50)

);

**LPorter\_cis556\_data.sql** (example sql code only. Used <https://mockaroo.com/> to create mock data)  
  
Populate Customers

INSERT INTO Customers (CustomerID, FirstName, LastName, Email, Phone)

VALUES

(1, 'Marylee', 'Brands', ' mbrands0@taobao.com, ' 410-982-2326),

(2, 'Claudian', 'Baston', ' cbaston1@aol.com, ' 389-737-8967),

Add more customers as needed;

Populate Inventory

INSERT INTO Inventory (ItemID, Brand, Model, Size, Price, QuantityInStock)

VALUES

(244175, ‘Under Armour’, ' Suede Classic ', 'L', 725.04, 4402),

(831061, ‘Puma’, ‘Air Max’, 'XL', 713.59, 9001),

Add more inventory items as needed;

Populate Employees

INSERT INTO Employees (EmployeeID, FirstName, LastName, Position)

VALUES

(601488, 'Jessie', 'Soanes', ‘Senior Financial Analyst'),

(194478, 'Garv', 'Mushet', ' Occupational Therapist'),

Add more employees as needed;

Populate Shipping

INSERT INTO Shipping (ShippingID, ShipDate, DeliveryStatus)

VALUES

(92374, ‘2023-12-04’, 'Delivered'),

(888402, ‘2023-12-04’, ‘Failed Delivery'),

Add more shipping records as needed;

Populate Orders

INSERT INTO Orders (OrderID, CustomerID, EmployeeID, ShippingID, OrderDate)

VALUES

(1, 1, 1, 1, '2023-11-30'),

(2, 2, 2, 2, '2023-11-30'),

Add more orders as needed;

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated