

SQL Project Showcase Report

Project Title: Vehicle Sales Order Management System

Project Overview

This project focuses on designing and implementing a relational database system using SQL to manage vehicle sales operations. The database captures customer information, product (vehicle) details, shipping partners, and order transactions in a structured manner. The goal is to ensure data integrity, reduce redundancy, and enable analytical reporting.

Problem Statement

Organizations dealing with vehicle sales require a structured system to manage customers, products, logistics, and orders efficiently. Storing all information in a single table leads to redundancy, inconsistency, and difficulty in analysis. This project solves the problem by designing a normalized relational database using multiple related tables.

Database Design (ER Model)

The database consists of four main entities: `customer_t`, `product_t`, `shipper_t`, and `order_t`. Each entity represents a real-world object, and relationships are established using foreign keys. The `order_t` table acts as the central fact table connecting all master tables.

Tables and Their Purpose

customer_t: Stores customer demographic, contact, address, and payment information. Primary Key: `CUSTOMER_ID`. This table allows tracking customer behavior and order history.

product_t: Stores vehicle details such as manufacturer, model, color, year, and price. Primary Key: `PRODUCT_ID`. Separating product data avoids duplication across orders.

shipper_t: Stores shipping partner information including shipper name and contact details. Primary Key: `SHIPPER_ID`. This enables logistics tracking and shipment analysis.

order_t: Stores transactional data for each order, including customer, product, shipper, quantity, price, discounts, order dates, shipping mode, and customer feedback. Primary Key: `ORDER_ID`. Foreign Keys link this table to `customer_t`, `product_t`, and `shipper_t`.

Keys and Relationships

Primary keys uniquely identify records in each table. Foreign keys establish one-to-many relationships: one customer can place many orders, one product can appear in many orders, and one shipper can handle multiple orders. This design enforces referential integrity and supports scalable data growth.

Normalization

The database follows normalization principles up to the third normal form (3NF). Each table stores data relevant only to its entity, eliminating redundancy and preventing update, insertion, and deletion anomalies.

SQL Concepts Implemented

The project demonstrates practical use of SQL concepts including CREATE TABLE, primary and foreign keys, constraints, JOIN operations, aggregate functions (COUNT, SUM, AVG), GROUP BY and HAVING clauses, and date handling.

Business Use Cases Enabled

The database supports real-world analysis such as total sales per customer, popular vehicle models, revenue analysis, shipment tracking by shipper, and quarterly sales reporting.

Challenges Faced and Solutions

Key challenges included handling relationships between entities and avoiding data duplication. These were resolved through proper normalization, use of foreign keys, and careful table design.

Conclusion

This project demonstrates a strong understanding of relational database design, SQL fundamentals, and real-world data modeling. It showcases the ability to translate business requirements into a scalable and efficient database solution.