Michael Anilonis

Brian Dorsey

Adam Downey

Retail Store Database System

The retail store database system is designed to work for a general chain of retail stores. It includes support for handling products, employees, customers, sales, and stores. The purpose of the system is to organize a retail chain’s operations and keep track of many important interactions in the businesses day to day operations. The system also has included functionality to give users important information about the store chain’s performance over time, including product, employee, and store performance, as well as customer habits. These more advanced queries can be modified to give information of different kinds.

The database used for this system is MySQL version 14.14, and MySQL Workbench was used to make changes to the database more easily. It is hosted by Oswego’s Pi server with database name wight19. The database consists of five distinct tables; Product, Employee, Customer, Sale, and Store. The Product table holds all the products for sale by the retail chain, as well as all data about these products. The Employee table handles employees personal information and their store of employment. The Customer table handles customers personal information. The Sale table records sales made by employees and the products included in the sale. Finally, the Store table contains all data about each store, its location, and its manager.

The database is design process was started with required attributes. These attributes were stored in carefully chosen relations and normalized to prevent redundancy.

Customer (custID, firstName, lastName, phoneNum, memberSince, email, date\_of\_birth)

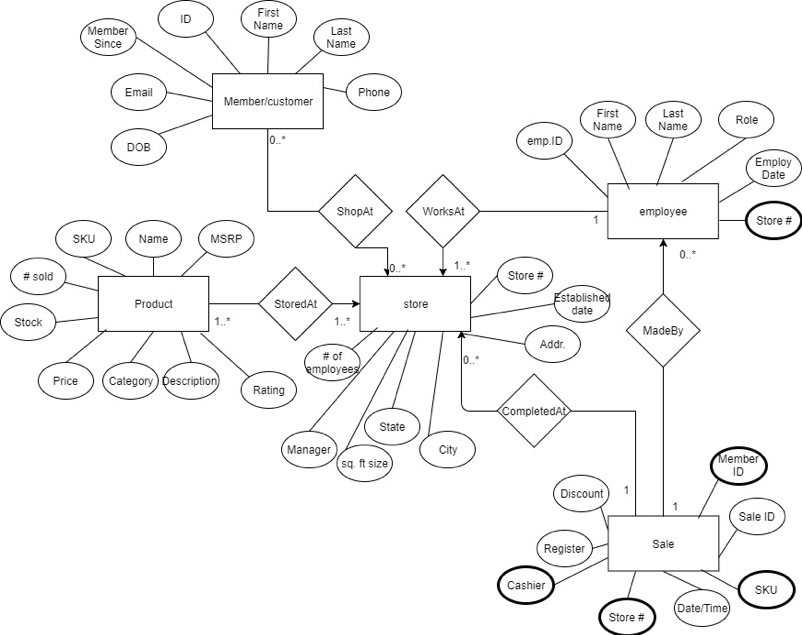
Employee (empID, firstName, lastName, role, employeDate, storeNum)

Product (msrp, prodName, sku, sold, price, category, description, rating)

Sale (saleID, cashier, storeNum, sku, member, discount, date\_of\_sale)

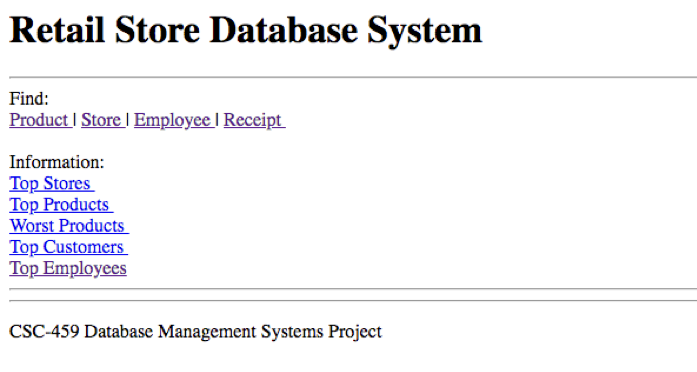
Store (storeNum, estdate, address, city, state, sqFt, managerID, numofEmp)

The entity relationship diagram of the retail database system was then built from these relations.



The user interface for the retail store database system is a web-based application written in Java with JSP pages, and is hosted by a Tomcat application server. All tables can be searched by their respective primary key from within the interface, and some tables can be searched more easily by other significant fields. Additional pages allow a user to view information about the highest and lowest performing stores, employees, and products by means of more complex queries. These queries consider the performance by net sale total or sale count. The highest and lowest spending customers can also be queried from a web page. These pages return a number of results as directed by the interface user.

The following are screenshots showing some of the functionality of the user interface.



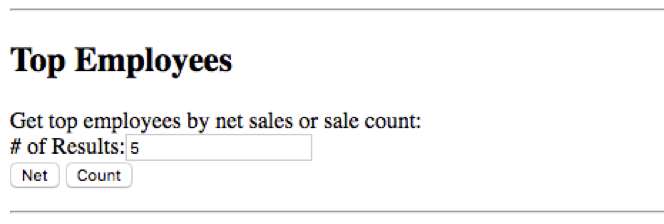
The home page presents the user with options to search a particular table or get information about a certain entity. For example, the following shows a search for a store with id 4.



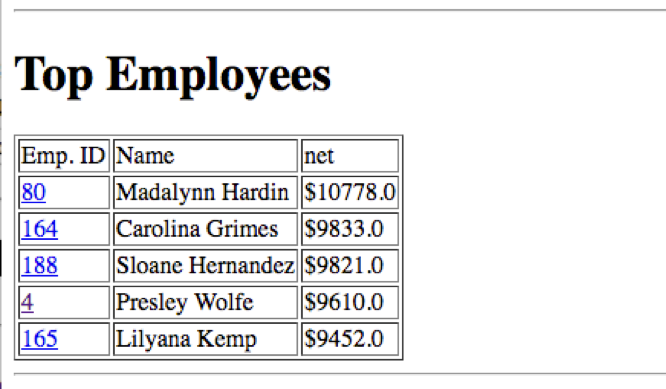
The application returns the data stored in the database pertaining to the id searched.



Users may also get information about the highest performing employees by the “Top Employees” page. The user must enter a number of records to return as well as the method to present the search. The “Net” search will return the top employees by net sales total, and the “Count” search will return them by total sales count.



A request for the top 5 employees by net sales total would return the following table.



From here the user can run another search, or choose to view a particular top employee’s information. The process of searching for other data or viewing other high and low information is the same as for employees.

The MySQL database was populated with a large amount of dummy data to give a wide range of search results and more detailed information about each store. The database contains 10 stores, 200 employees, 1000 products, and 5000 customers and sales. The population was done using Java code written to generate CSV files of random data formatted for the particular attributes. Incrementing integer ids were used for identifying most relations.

In conclusion, our retail database system attempts to simulate a real-life retail store chain company. The database was modeled for imitation of what a person might need to know about the chain as a whole. A database schema was created that would best represent a retail chain store. We were then able to implement mock data inside the database to understand possible real-life use cases of our system. After the data was implemented queries were implemented to analyze this data and return useful information regarding the store chain.