Final Project Report

Group 10: Andrew Gould, Yuelin Xie

Introduction:

Our scenario is equivalent to the inventory, employee, and customer management software that is required to operate most retail stores. By keeping a database full of all of these details, we can provide many different insights about the state of the store, like what needs to be ordered, what customers are buying, and where to allocate marketing budgets.

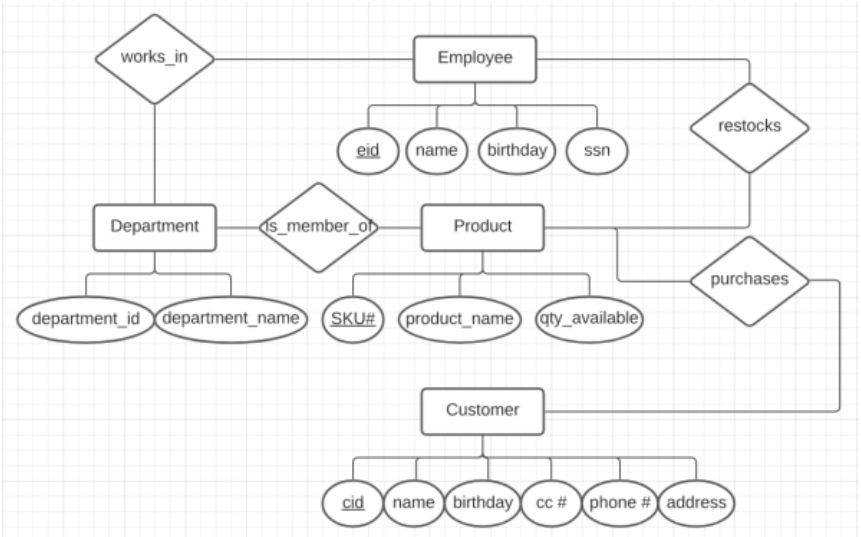
Requirements analysis (be brief):

Data: customer's name, customer's order ID, customer's address, customer's phone number, customer's credit card number, employee's name, employee's ID, employee's SSN, employee's birthday, employee's department, product's name, product's SKU#, product quantity available, department ID, department name.

Constraints: Each employee has an unique id. Each product has its own SKU number. A unique identifier for a user is their id. Each employee serves a different department.

Operations: Different products in different departments. Employees' job is to restock the product, and each of them works in a different department. Customers purchase the products.

Conceptual design: ER-diagram and constraints



Logical design: convert our ER diagram to relational schemas.

DEPARTMENT(did, name)
PRODUCT(SKU, name, quantity, department_id)
CUSTOMER(cid, name, birthday, cc_number)
EMPLOYEE(eid, name, birthday, ssn, department_id)

Architecture & Implementation:

We used php and mySQL to build this project on linux. For the php part, we first used to write several php front-end files to display the basic framework on the web page, such as product name, fillable text box and submit button. Then in the product part of php, we added findProduct functionality and addProducts functionality. Then we add the function to add the employee to the Employee table from EmployeeInfo.php. We then added the ability to add transactions from the website, and recent transactions with transaction IDs. Finally, we updated table styling for the view products page.

Testing:

We tested our project using two methods. For the first method we used xampp to test our project. Second test we used nginx on the EECS personal page server.

Our website of code:
https://github.com/yuelinxie/647_Project