INST327: Database Design and Modeling 0203

Project Final Submission - December 12, 2022

Team 03: Sanjana Sankar, James Igwenagu, Ryan Flint, Ildreed Mbami, David Chun

INST327: Project Final Submission

Introduction:

Stores are important businesses that often have a lot to manage. They employ many people and often have vast inventories that need to be managed with databases. Some of our group members have worked in stores, and as such, we know that a lot goes into successfully operating the different business transactions that occur in a store. As a result, we are interested in creating a database that can manage a store's payroll, positions, employees, items, and inventory. This would help the restaurant run more smoothly and make it easier for employers and employees alike to manage their pay, hours, and positions. Having worked in retail, we have the experience to know that they use similar databases to keep track of their employees and sales, so this is a useful and practical application of databases that we have seen in professional environments. Finally, it is a universal acknowledgement that everyone wants to be paid properly for the time they work, and this database can help with that.

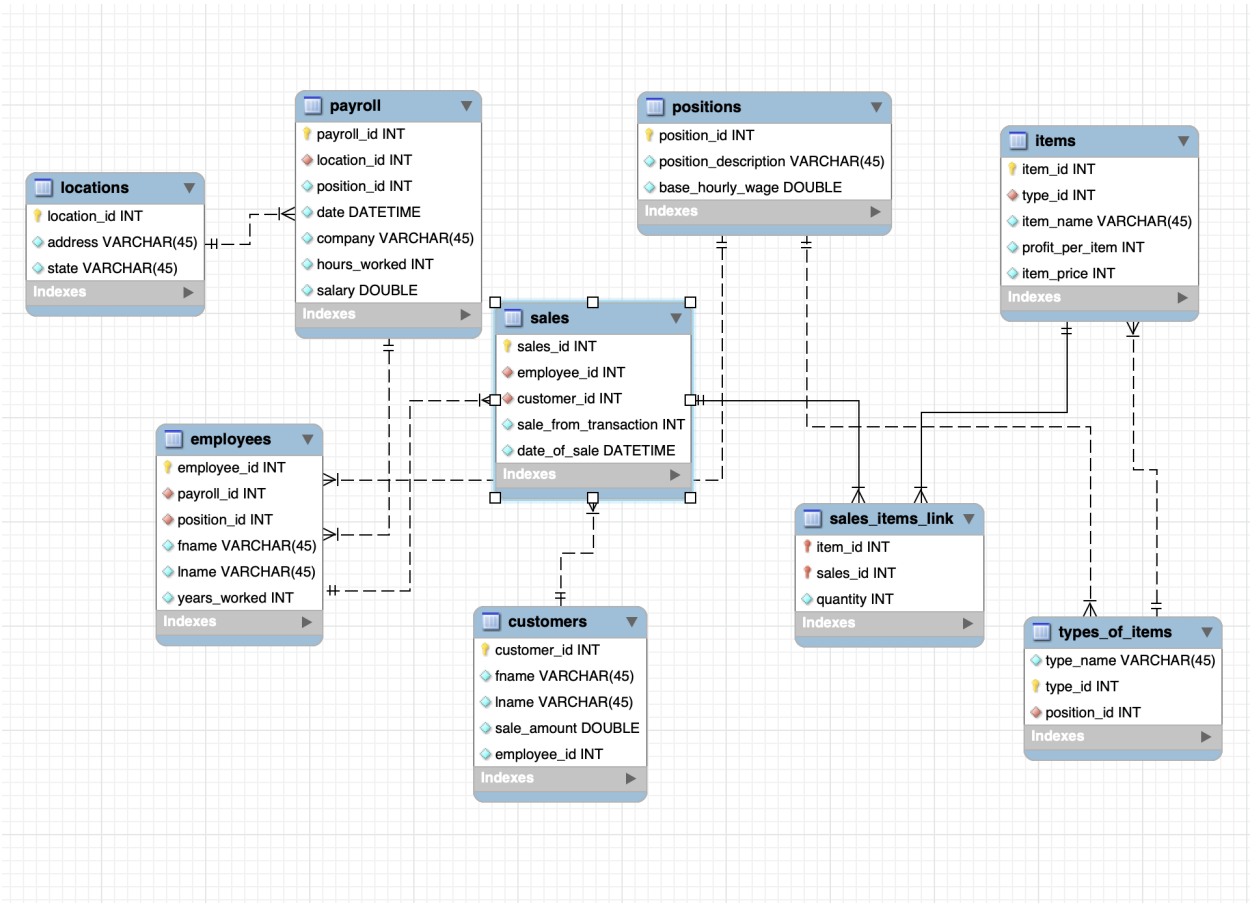
A business must know what food and items they have in their inventory and what they are running low on. Otherwise, they won't know when to make an order for more supplies to replenish inventory. This database can assist with that as well as much more.

Database Description:

Our database will track employees, what position they have in the store, whether they have multiple roles, how many hours they have worked at each position, how much they are paid hourly for each job, the items they sell, the customers they sell to, as well as the various purchases and sales that occur in the store. Our database will also include tables for items in the inventory, including the different types of items and the quantity of each, as well as the cost of the items and how much the store makes on them. This way, a manager can quickly tell which items are running low, which items are expensive for the store to purchase, which items make the store the most money, and which items sell the fastest. Every table in the database is connected with each other through foreign keys or composite keys, forming different relationships among each other to replicate an advanced database.

This database will be of great use to store managers who are looking to manage their inventories and employees. An in-depth analysis of this database will allow someone to evaluate different business strategies in order to improve business performance while also keeping track of those working for them in an organized manner.

Logical Design:



For the ERD part of this project, we worked with TAs and the instructional team to come up with recommendations for tables and different keys that would pair best with our existing tables. The different tables are quite self-explanatory, with different keys indicating the various pieces of information that the database holds. The payroll table keeps track of the payroll that is run every two weeks. The locations table keeps track of the location of the store where the transactions are taking place; the positions table keeps track of the positions of employees in the

store; the employee table keeps track of the employees working for the store; the sales table keeps track of the sales being conducted by the company; the customers table keeps track of the customers that make a purchase at the store; the items table keeps track of the items that are held; the types of things table keeps track of the different kinds of objects; and the sales items link table connects items and sales so that people can make multiple purchases within a single sale.

Physical Database:

As stated in the database description, our database will keep track of employees, the positions they hold in the business, whether they hold multiple roles, the number of hours they have put in at each one, their hourly pay for each one, the goods they sell, the people they sell to, as well as the various purchases and sales that take place there. In order to test our database, sample data is required. Sample data is necessary to test our database. Since we can't collect personnel records and hours from actual establishments, we made up our own restaurant data.

Additionally, there are other ways to generate sample data for the MySQL workbench database. We created the data using an Excel spreadsheet as our method. The fabricated contextual and systemically accurate sample data was entered into the appropriate cells after the requisite columns and rows had been created in the spreadsheet. The information was then exported from the spreadsheet and imported into the MySQL database as a file.

When it comes to exporting the data from Excel, we did so by using the "Export" function to save the information as a CSV (Comma Separated Values) file. The LOAD DATA INFILE statement can then be used to import this file into the MySQL database. The path to the CSV file and the table into which the data is to be imported can both be specified using this statement. For instance, the data from a CSV file called "sample data.csv" could be imported using the following statement into a table called "sample table":

[LOAD DATA INFILE 'sample_data.csv' INTO TABLE sample_table;]

The data can then be utilized for testing, just like any other sample data in the database, once it has been imported into the MySQL database. We discovered that it was quicker and simpler to generate substantial volumes of sample data that can be utilized to test and enhance our database by utilizing Excel to create, export, and import the data.

Views/Queries:

View_Name	Req A	Req B	Req C	Req D	Req E
`sale_transaction_view`	X			X	
`low_position_employee_sales`	X	X			
`employee_most_hours_under_10_years`	X	X			
`customer_sales`	X		X		
`hours_worked_common_state`	X	X	X		X

What each query displays:

- **Sale_transaction_view:** Shows the employee id and connects it with the sale transactions they made as well as the profit made for each transaction.
- **Low_position_employee_sales:** shows the employee name with their position and their sale transactions but only for employees that make under $15 an hour.
- **Employee_most_hours_view:** Shows the employees and their hours worked alongside their salaries, ordered by hours worked. This only includes employees who have worked less than 10 years for the store.

- **Customer_sales:** Groups sale transactions by customer to show how much each customer has spent at the store in total and places orders by total sales per customer.
- **Hours_worked_common_state**: Shows employees hours worked, ordered by hours worked, and only includes employees from the most common state for employees.

Changes from early designs:

Some changes from our project proposal involved cutting certain tables out of the design. In the early design we included an accounts table, a profit table, and a beginning inventory table. We excluded the accounts table because we didn't want to complicate our design, which was the same reasoning behind the exclusion of a Vendor Contacts Table. We also did not think that a vendor contacts table was necessary to include in our design since the database is made primarily for smaller businesses. There's also the fact that we can't assume a smaller business will have an unlimited budget.

We excluded the profits table and incorporated some of the columns we had into the Items table. The reason that we made this change is because initially, we thought that keeping the Items table and the Profits table separate was necessary for keeping the database in 2nd and 3rd normal form. However, we found that if we tracked the profit made by item instead , the profits table is not necessary. We cut the accounts table for a similar reasoning to cut the Profits table. Instead of using an accounts table, we could simply just implement some of its elements into the sales and the sales items link table. In addition, we wanted to keep our database more simple and discovered that having several different accounts to keep track of money in a small business just wasn't necessary.

One important change that we had from our early proposal design was the addition of the locations table. At first glance, adding a locations table doesn't seem necessary, until you realize that it's necessary to keep the database in the second normal form. The address and the state is something that's not at all related to a payroll, and thus including those two columns in the payroll table would cause a partial dependency. The other columns would not cause a partial dependency because date, company, hours worked, and salary are elements that are directly related to the payroll, and thus we did not add new tables for them.

Database Ethics Considerations:

As far as data privacy goes, we plan on making sure that the employee's information or privacy isn't leaked in the event of a data breach. Employees, for example, could be DOXed

inadvertently. In order to avoid these problems, we devised a method of identifying individuals. Employees don't have any of their information divulged and can receive their pay without any bias attached as well. This table makes it so that only non-personal information such as position, wage, and hours worked is tracked.

The items that we sell at our store are those that are within fair use and won't get us in legal trouble. So for things such as basic ingredients, they fall within fair use since you can't copyright ingredients. However, for menu item names, we have to make sure to either use generic terms for items such as "stationary" or "groceries." Instead, we will try our best to use specific names like "wooden pencil" or "apples." By using generic terms to describe our menu items, we abide by the fair use rule and can't be sued for it.

The Database Ethics presentation that was shown to us during the lecture acted as a positive encouragement to be more inclusive and diverse in the creation of our database. It was a little difficult to do this as we were working with data that is difficult to be biased in the first place (store inventory), but regardless, we did make sure that parts of our database where we could incorporate elements of diversity, we did.

Lessons Learned and Future Work:

Future work on this project would include more queries involving more advanced functions that combine more tables that we would be able to incorporate into an inventory based business. We would work on creating more logical tables as well as keys to include some more complexity which in turn would potentially open a gate to include more diversity into our system.

Upon completion of this project, we learned a lot about databases and working with them. The experience gained from working on this project is equivalent to a real world opportunity in which we would potentially work for an organization that would ask us to create a database similar to the one we have made. The logical, physical, as well as the CRUD queries have successfully prepared us for working with databases either with companies or for a personal project that we can decide to work on in the future.