# South Dakota School of Mines & Technology

# Database Management Systems, Spring 2022

CSC 484-M01

## Course Project

**Due**Friday, April 29th , 2022 at **11:59 PM** (Mountain Time).

**Reference(s)**None

**Software Required**None

**Deliverable:**

Submit a zipped file containing the following:

* Detailed report document (.pdf or .docx)
* Application files

Write a detailed report on your project that must provide the following information

* Functional dependencies of your relations
* Normalization at least up to 3NF
* SQL statements
* Application Implementation details

# South Dakota School of Mines & Technology

# Database Management Systems, Spring 2022

CSC 484-M01

Final Project Report

Mathew Clutter

Adam Kraus

**1. INTRODUCTION**

This document provides a detailed report on CSC 484 Final Project. The project was undertaken by Mathew Clutter and Adam Kraus.

Mathew Clutter and Adam Kraus worked together heavily to complete this project together. These students worked together to draft the initial requirements document, generate the ER diagrams describing the database setup, and ensure that the resulting tables are properly normalized. While many of the required tasks for this project were completed together, each individual student also worked on some tasks individually. Adam worked to seed the database and start the Python application, and Mathew focused on creating this report and documentation. While each student worked individually on some tasks, this project was largely a collaborative undertaking.

This application is going to be used for a warehouse management system. It could be used for managing products, orders, and other applicable information. This project consists of a total of 6 tables. These tables include customers, products, orders, order items, employees, and injury reports. The tables are related according to the following ER diagrams:

Diagram

Description automatically generated

Diagram

Description automatically generated

The first of these diagrams offers a higher level overview of each table and how it relates to other tables, while the second diagram shows all of the connections, along with the primary and foreign keys in each table. The second diagram was generated by the MySql Workbench application, from the database schema definition. For more information on the layout and format of this database application, please see the initial project proposal document.[[1]](#footnote-2)

All questions, and complaints, can be directed to this (these) individual(s).

Mathew Clutter: mathew.clutter@mines.sdsmt.edu

Adam Kraus: adam.kraus@mines.sdsmt.edu

**2. FUNCTIONAL DEPENDENCIES**

Before the process of normalization could begin, our team worked to identify the functional dependencies present within the set of tables proposed.

Starting with the customers table, it can be seen that the first name, last name, email, phone number, and address are all functionally dependent upon the customer ID. For each customer ID, there is only one value of the set containing the first name, last name, email, phone number and address fields. Thus, if the customer ID is known, the values for all of the other fields can be determined. Because of this relationship, the customer ID was selected to be the primary key of the customers table.

Next, in the employee table, the employees first name, last name, position, phone number, address, salary, and their current employment status can all be determined by the employee ID. The employee ID is the only attribute that can be used to uniquely identify any tuple in the relation. Thus, the employee ID was chosen to be the primary key of the employee table.

Within the injury report table, the employee ID, injury date, and description can all be determined from the injury ID. The injury ID is the only attribute that can uniquely identify the employee ID, injury date, and description within this table. Thus, the employee ID, injury date, and description are functionally dependent upon the injury ID within this table. Thus, the injury ID was chosen as the primary key within this table.

For the products table, the product name, price, and number remaining are all functionally dependent upon the product ID. The product ID is the only attribute that can uniquely identify the combination of product name, price and number remaining. Thus, the product ID was chosen as the primary key of this relation.

For the orders table, the customer ID, employee ID, placed on date, fulfilled on date, and notes can only be uniquely identified by the the order ID. Thus, the customer ID was chosen as the primary key of the orders relation.

Finally, for the order items table, the number fulfilled and number ordered can be uniquely determined by the set containing the order ID and product ID. As it takes a combination of the order ID and product ID to uniquely identify the number fulfilled and number ordered, the number fulfilled and number ordered are functionally dependent upon the combination of the order and product IDs. Because of this relationship, the order ID and the product ID were chosen together to form a composite primary key.

**3. NORMALIZATION**

In order for the tables to be in third normal form (3NF), they must fulfill the requirements to be in first normal form, and second normal form, along with the additional requirements to fullfill third normal form. In order to be in first normal form, the intersection of each row and column must contain one, and only one value. For second normal form, first normal form must be achieved, along with every non-primary-key attribute being fully functionally dependent on the primary key. Finally, for third normal form, the relation must be in first and second normal form, along with there being no non-primary-key attribute transitively dependent upon the primary key. As constructed, our tables fulfill the requirement of being in third normal form.

Each table is in first normal form, as every table only contains atomic values (that is, each column only contains one entry per row). Additionally, there are not any repeated columns in any table. This combination ensures that each table fulfills the requirements to be in first normal form, as each intersection of row and column contains only one value.

Additionally, each table is also in second normal form. For the customers, employees, injury reports, products, and orders relations, there is only one primary key (the primary key is a single attribute, not a composite key). Thus, since the primary key in each of these tables is able to uniquely identify the rest of the values in each tuple, the non primary key attributes are not dependent on a subset of the primary key, only the primary key itself. For the order items table, showing second normal form is not quite as clear, as the primary key in the order items table is a composite key consisting of the order ID and the product ID. However, the non-key attributes in this table require the values for both the order and product ID in order to be uniquely identified. It is possible for different orders to have the same product ID as another order; thus, the entry in the order item table cannot be uniquely determined by only the product ID. Similarly, for one order ID, there can be numerous product IDs associated with that singular order. Thus, in order to uniquely identify the number ordered and the number fulfilled for any order item, the combination of the product ID and the order ID is needed. As the non-key attributes are not dependent upon a subset of the primary key, this table is in second normal form.

Finally, each of the tables are in third normal form. There are no transitive functional dependencies present in any of the tables. Each table contains the minimum amount of information required to be uniquely identifiable, with minimal data duplication. In every table, the non-key attributes are only dependent upon the primary key. There are no tables where the primary key can be used to uniquely identify another attribute, which can then be used to uniquely identify another attribute within the relation. Each attribute can only be determined by the primary key itself. This ensures that each relation is in third normal form.

**4. APPLICATION IMPLEMENTATION DETAILS**

[ In this section describe the following:

* The type of application: Web, Console, Mobile App, Desktop App
* Language(s) that was/were used to implement the application
* Modules/Packages that you utilized
* Application design and design decisions
* Code snippets of how your app connects to your database
* Any other relevant application

]

**5. WORKFLOW/DEMO**

[Describe how your application works starting from the main menu/page.]

**6. CONCLUSION**

[Describe:

* what you learned during the program execution.
* any problems/issues that you run into and how you resolved it/them
* how the project/app can be improved in future.

]

1. Note that the injury report table is a more recent addition, that was not present in the initial proposal. [↑](#footnote-ref-2)