



CSCI 4140
Advanced Database Systems

Assignment 1

May 22nd, 2021

Submitted By:

Adama Camara (B00757421)

Arup Paul (B00827772)

Clarizze Santos (B00730508)

Sijia Han (B00734818)

Table of Contents

<i>Project Overview</i>	<i>3</i>
<i>Entity Relationship Diagram</i>	<i>4</i>
Entities.....	4
Database design:	4
Business rules.....	4
Description of entities.....	5
Relationships.....	6
<i>Hosted API overview</i>	<i>6</i>
<i>Tech Stack and Software</i>	<i>7</i>
<i>Project Repository</i>	<i>8</i>
<i>Running Project Code/Script</i>	<i>8</i>
<i>Reference:</i>	<i>9</i>

Project Overview

In this project we aim to create a client-agent web application that provides a user interface to agents for managing incoming orders and updating the availability of parts in the warehouse. Further development of the project will be enclosed in the future assignments. Below are the details of our project's database management system.

Entity Relationship Diagram

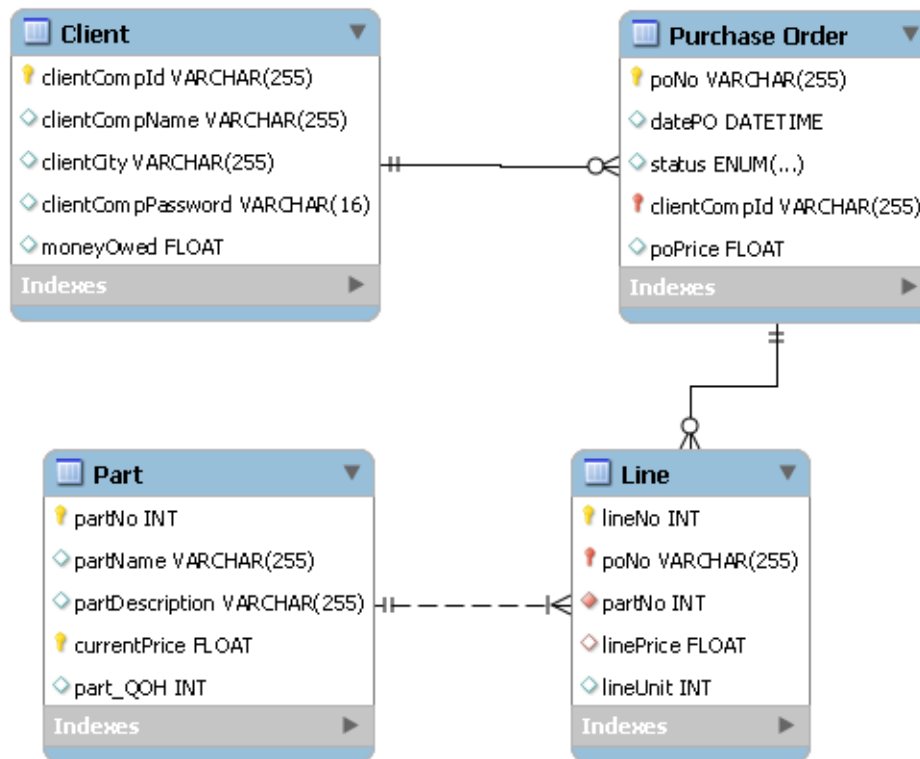


Figure 1. Group 1 Entity Relationship Diagram. [1]

Entities

Entity	Relationship	Connectivity	Entity
Client	places	1:M	Purchase Order
Purchase Order	contains	1:M	Line
Line	refers	1:1	Part

Figure 2. Database Entities.

Database design:

Create a detailed narrative of the organization's description of operations.

The organization hired our team to develop a database with a front-end application for their agents to manage incoming orders and adding/updating availability of parts that are stored in the company warehouse. The application will also have a client front-end where placing of order would be made and client will be able to check their order status and their balance.

Business rules

From the description of the business rules, every purchase from a client will create a purchase order, no matter the availability of parts in the warehouse. The agent will oversee completion of purchase order and confirm the order by making sure the parts being requested by the client are available.

Description of entities

Client: The client table is holding information on the client unique ID which was decided not to be an INT auto_increment, but a varchar() with uuid generated from the library **react-uuid**. The table also holds the client city, their password, and the money they owed.

Part: The Part table is holding information on the parts that are being stored at the company warehouse. Each part is identified by their partNo which is the primary key set as an INT auto_increment. The part table also hold information on the name, description, quantity on hand, and current price.

Line: The Line table act as a temporary cart for each purchase order. It holds information about the selected part (partNo), and link to a purchase order by holding information on the poNo (purchase order unique ID). It also holds the current price of the selected part under attribute called 'linePrice' which is a foreign key referring to the table Part. The number of selected parts is determined by the attribute "lineUnit" which is an INT that will be used to calculate the total price of a purchase order.

Purchase order: Each Purchase order will be identified by their poNo which is the primary key. The table is linked to Client through the client 'clientCompId'. The table also hold information about the order status, the date at which it was ordered and the price of the purchase order.

Our initial ER diagram was composed of 3 tables: Client, Part and Purchase Order.

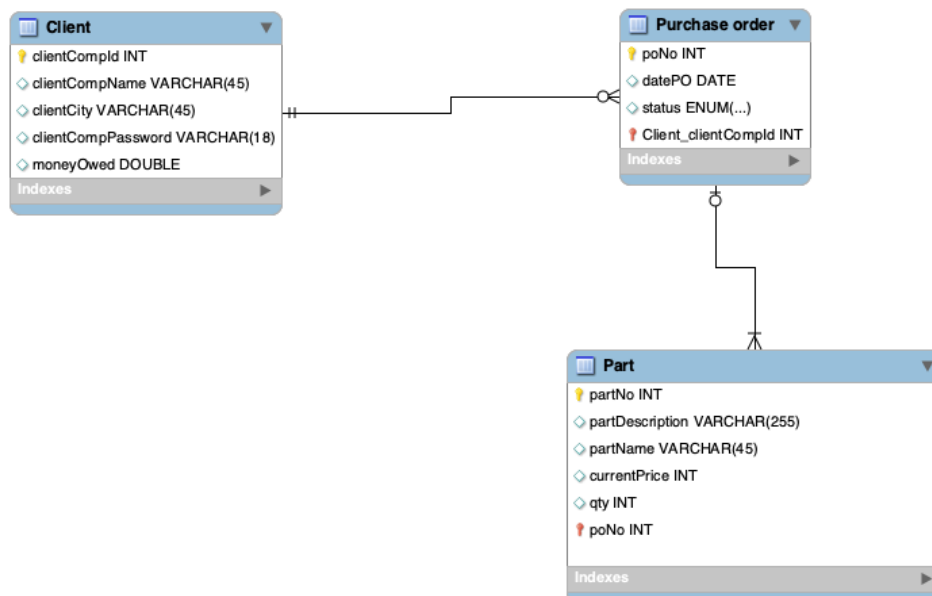


Figure 3. Initial Entity Relationship Diagram of Group1. [1]

Based on our initial understanding of the project description, the team was not sure about the functionality of agent being able to change price of parts. Our initial database design was setup

in a manner that changing the price of a part would automatically change the price stored in the purchase order table. We also figured out that the instructions given was missing information such as the Line table that we created in our latest ER diagram. The Line table will allow client to increase the unit of a selected part.

The team also decided in making the “partNo” and “currentPrice” as a composite primary key, as making the “currentPrice” a primary key allow for the table Line to have access to the price without needing to input the value manually during insertion of value in Line.

Relationships

Client (1) ---places--- Purchase Order (0,M)

We decided to have a 1:M relationship between these two entities because from the business description, each client can place multiple purchases order or no order.

Purchase Order (1) ---has--- Line (0,M)

Each purchase order can have 0 to multiple line that refers to a specific part.

Line(1) ---refers to--- Part(1)

Each line refers to one part.

Hosted API overview

Link: <https://csci4140-group1.herokuapp.com/>

Endpoints:

Sample values:

_clientCompId: 3bb6de1477974ed28bb5a

_poNo: 4292f2d43134491fa186

- Client:
 - All client [GET]: /clients
 - Client by Id [GET]: /clients/clientCompId
 - Add client [POST]: /addClient
- Part:
 - All part [GET]: /parts
 - Part by Id [GET]: /parts/1
 - Add part [POST]: /addPart

- Purchase order:
 - All purchase order [GET]: /pos
 - Purchase order by Id [GET]: /pos/poNo
 - Process order [GET]: /pos/process/poNo
 - Return order [GET]: /pos/return/poNo
- Line:
 - Line by Id [GET]: /lines/1

Tech Stack and Software

We plan to use the model-view-controller design pattern for the overall the architecture of our project. We will have a front-end, where it will contain all the design and information structure of the website we will be creating and a backend that is connected to the database that stores all the information requested by the user. Below are further details of the different frameworks that we will be using to establish our project.

Frontend

Our team decided to use HTML, CSS, and a bit of ReactJS for our front-end. We chose these for our project because all the team members have adequate experience with HTML and CSS and using ReactJS complements along with the other frameworks we will use for our backend framework and the database. We expect a minimal learning curve with using ReactJS, since we have few functionalities needed to connect the frontend to the backend. To enhance further the layout and design of our project, we will also be using Bootstrap framework.

Backend

We plan on using **NodeJS** and boilerplate the back end by generating an Express app. The Express app will be connected to the tester local machine **MySQL**. The front-end will be using **axios** to make the http requests, and the library **React-uuid** to generate random and unique identifiers for the creation of purchase order.

Tools used

In terms of IDEs and software, we will be using Virtual Studio Code as most team members are familiar with using it. We also used MySQL Workbench to generate the ER Diagram shown above. The codes for the back end and front-end will be stored under the GitLab link provided in the next section.

Project Repository

GitLab Repository: https://git.cs.dal.ca/apaul/4140s21_group1.git

Running Project Code/Script

Required Software to run Group 1's SQL Script:

- MySQL Workbench [3]

Method 1:

1. Open the Final_DB_Model.mwb using MySQL WorkBench from the submitted zip folder.
2. Click on File>Export>Forward Engineer SQL script

Method 2:

1. Download the G1-Assignment1.sql from our zip file or from our GitLab repository:
https://git.cs.dal.ca/apaul/4140s21_group1.git
2. Open your MySQL Workbench.
3. Connect your MySQL Workbench to a local database server.
4. Create a new schema in the connected server and give it a name say "4140a1g1".
5. Go to File>Open SQL Script>Select G1-Assignment.sql
6. At line 1, Enter the SQL command: use 4140a1g1;
7. Execute the script.
8. Once the scripts show executed successfully, right-click on the schema name "4140a1g1" on the right Navigator>Schemas panel and select "Refresh All".
9. The database is ready, and its tables and columns are visible by expanding the arrows under the Navigator panel.

Reference:

- [1] “MySQL Workbench,” *MySQL*. [Online]. Available: <https://www.mysql.com/products/workbench/>. [Accessed: 15-May-2021].
- [2] “Heroku,” Cloud Application Platform. [Online]. Available: <https://www.heroku.com/>. [Accessed: 22-May-2021]. The team used a free hosting service provided by Heroku for previewing the database.
- [3] “Tutorial –MySQL Workbench.” Faculty of Computer Science, Dalhousie University CSCI 4140 Summer 2021, Halifax, 03-May-2021.