CSCI 3461: Databases
Term Project

Evan Farrell, Jordan Dempsey & Tyler Robinson

April 1, 2019

1 Database Diagrams

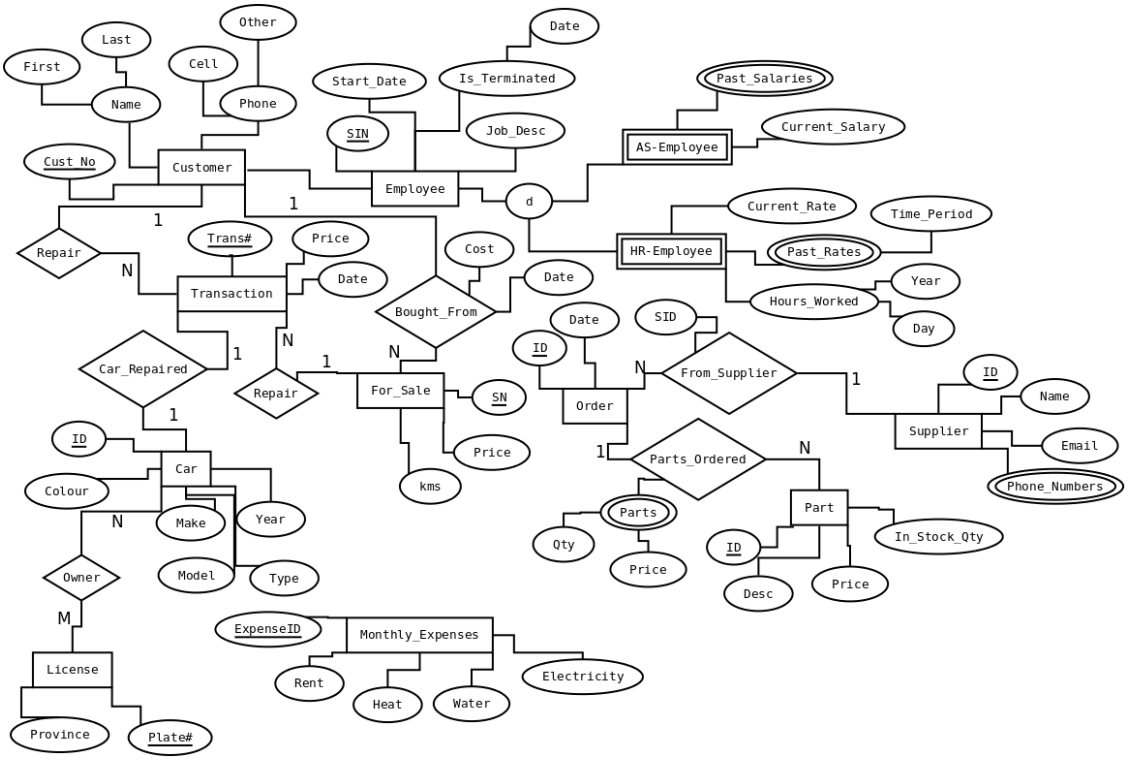

Figure 1.1: MUC EER Diagram.

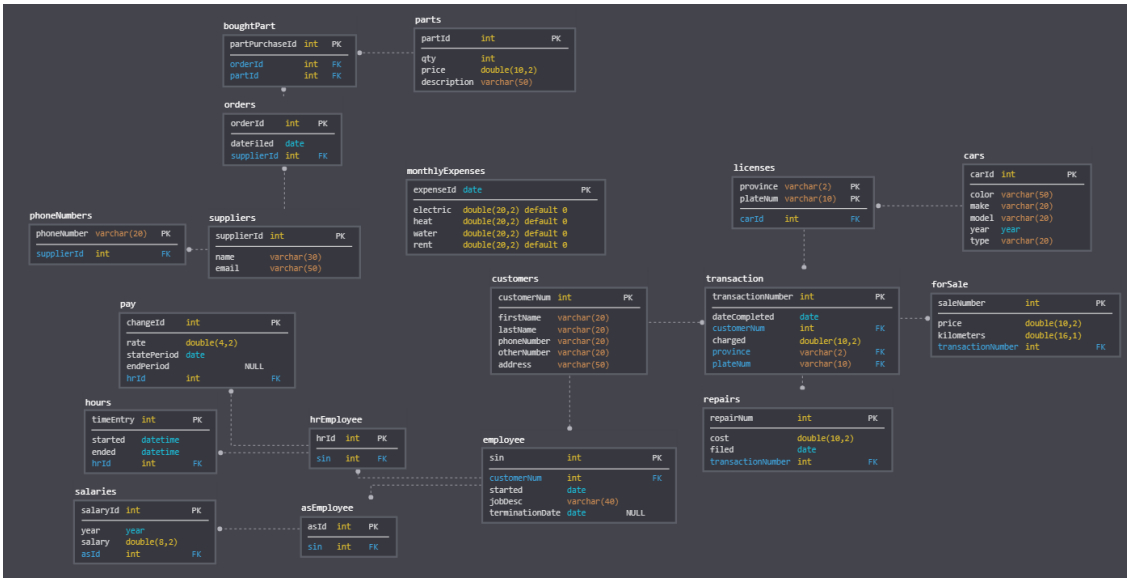

Figure 1.2: MUC Relational Database Schema.

2 Description

To keep track of it's many (happy) customers, MUC employs a large number of technologies into it's website. The web-app is written predominately in Javascript code (ECMAscript 2015) with additional PHP and CSS variants to aid in visuals and clarity.

The client-side is responsible for generating queries to the MySQL database serverside. While this is potentially the least secure way possible to perform this task, it was also the fastest for prototyping and researching.

The site is organized into a few folders[1]:

`common`	HTML shared among many pages, including the `head` section.
`css`	Exclusively CSS, completely generated.
`external`	Tools & scripts for use on the server with no access from the web.
`images`	Visual data for the website.
`sass`	SASS source files to be interpreted into raw CSS.
`report`	Source for this document.
`scripts`	PHP and JS functions to support the web-app.

Outside of these subdirectories, the files located in the home directory are all loadable webpages, the first of which being `index.php`. It's important to note that despite the PHP extension, the extent of the languages use in this case is limited to includes.

`index.php`	First page and location to view full tables.
`edit.php`	Manual management of the various tables.
`cancel.php`	Cancelling previous orders.
`order.php`	Creating and submitting new parts orders.
`supplier.php`	Adding new suppliers into the database.

Inside the scripts folder, the key files to look at would be:

- `db.php`: Routing MySQL calls to the database and returning JSON replies.
- `db.js`: Includes `sendRequest(...)`, which communicates asynchronously with `db.php` to avoid random page loads between requests.
- `helpers.js`: Includes a few common functions, including creating a table based on a database response for a query.

The remainder of the files are related directly to the page they appear on; there is no overlap between the remaining scripts.

Additional technologies/libraries used in the creation of the web-app include `jQuery`, `bootstrap` (with CSS), `popper.js` and `velocity.js`.

[1] `/external` contains the files to rebuild and autopopulate the database from scratch.