**CS4347 Database System Final Project Deliverable 1**

**Car Rental System**

Travis Johnson

Aaron White

Adetoye Efunnuga

Danny Choi

Ethan Payne

Zijia Ding

**Task Delegation:**

Travis Johnson - comparing databases

Aaron White - implementing sql

Adetoye Efunnuga -implement sql

Danny Choi - implement sql

Ethan Payne - implement diagrams

Zijia Ding - implement sql

\*designing the database was a groupwide effort

Adetoye worked on why we chose this project

Rental System

. Rental System

· Vehicles

· Paperwork or Documents

· Office

· Employee

· Lessee

. Insurance

Vehicles

· Type: standard, luxury

· Place: hotel, airport, dealership

· Insurance: minimum, premium

· Size: sedan, truck

Office:

· City: dallas

· State: Texas

· Branch Number: 20315

· Manager SSN: 123567894

Employee:

· Employee SSN: 123456789

· Manager SSN: 123456788

· FName: Fred

· LName: Fredderson

· Age: 26

· Role: Accountant

Lessee

· Age: integer 21+

· First Name: Jim

· Last Name: Jimmerson

· payment method: debit or credit

· card number: 1111 2222 3333 4444

· Driver’s License number: 123456789

· Email: jj@email.com

· phone number: 123-456-789

· Insurance Provider: AllState

Insurance

. Type: full coverage including roadside assistance

. plan id

. price: 20

. Duration: 3 days

It is easier to gather ideas on the topic and branch off or expand more. This project is practical because it is used constantly in our current society. The system goes in-depth into various systems; it contains other sub-systems which interact effectively with each other; for example, insurance is a separate system under the rental system which is provided to customers.

Our motivation for choosing this topic is based off our personal experience with the system itself. We all have rented either a car, book, game etc and are familiar with how the process work. Our goal from this project is to understand what goes on within the rental system and how it’s so effective.

This system can be incorporated into any company, and slightly altered to fit into any company that offers rental services. This system can also be applied to companies that rent other types of vehicles like boats, rvs, jet skis etc; also, this system become valuable when we travel from one place to another for either business or vacation.

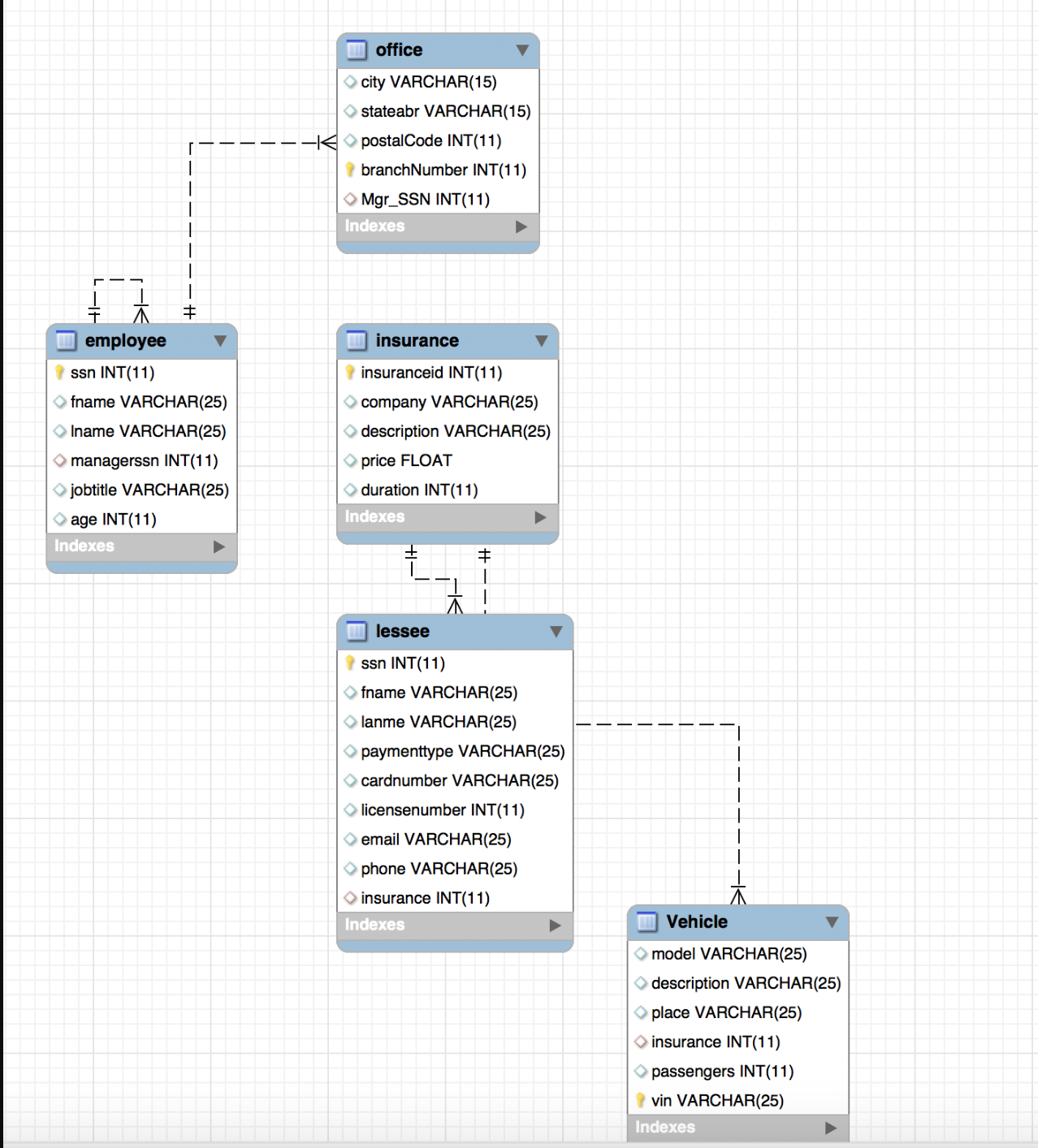
**Comparison:**

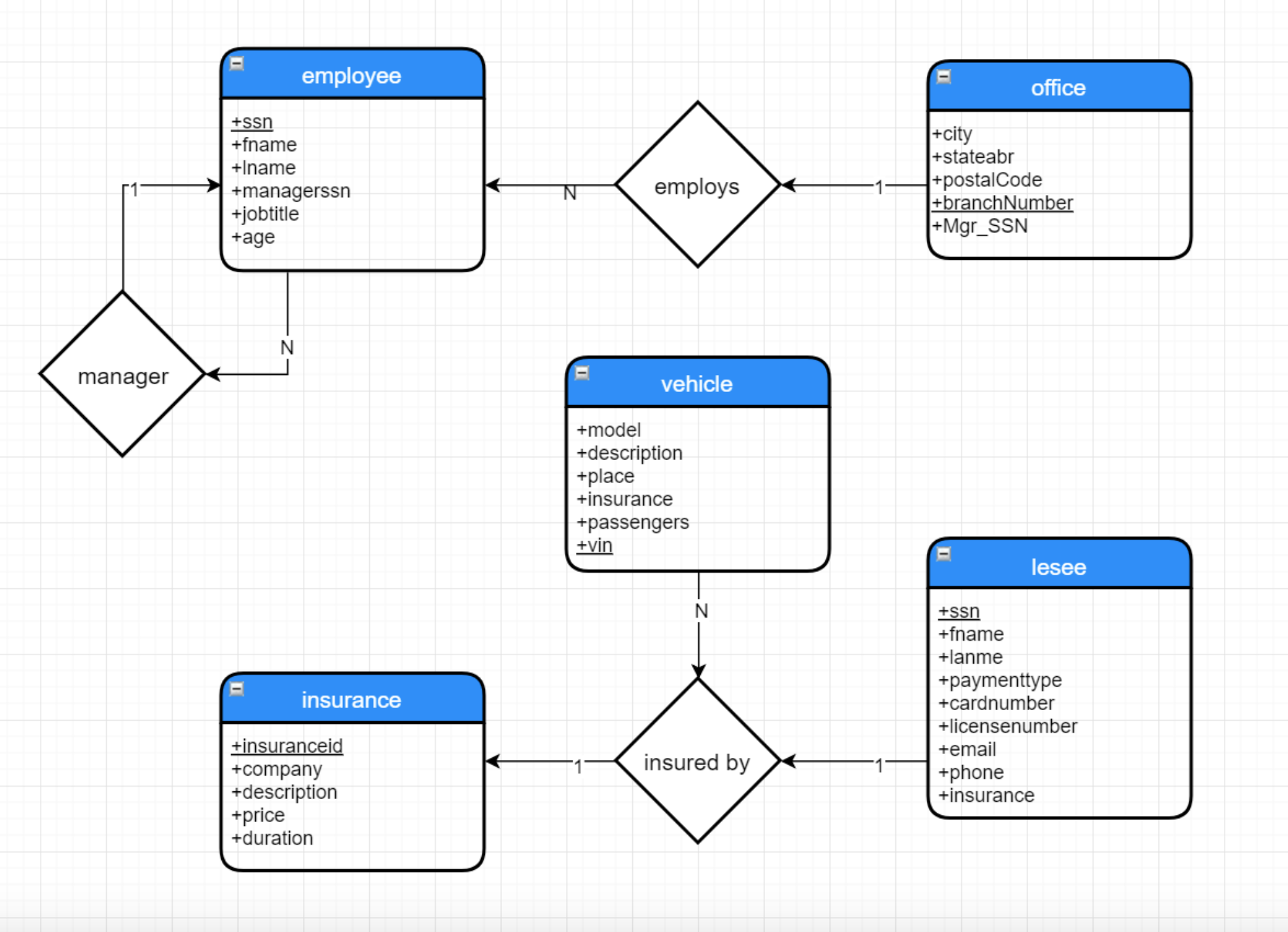
Our database is meant to be a full skeleton that is able to be modified easily and conveniently for car rental businesses. It contains the foundation that will be present in all car rental company databases. It is important that a skeleton doesn’t contain too much because companies can differ even in the same industry.

The first database ours will be compared to is from Vertabelo and the biggest difference is that it is much more spread out and less dense. It also contains relations for things such as equipment which is not necessary for our skeleton, or for many companies. One example of the density difference between our database structures is how insurance was handled. We only have one dense table while the database we are comparing it to has two. We chose having fewer, but denser, tables because it would be easier to understand how everything in the database relates amongst each other when there are fewer pieces.

The second database ours will be compared to is from Telerik. The biggest difference between our databases is that Telerik has an even more dense database. However, it is similar to the Vertabelo database in the way that it has a bunch of unnecessary attributes. Telerik handled the way they stored their extra attributes in a different way and also left out attributes that our table had, specifically the insurance attributes. However, we wouldn’t want to include some of their extra attributes, such as air conditioning, in a full diagram because we think that all cars should come with them, so they shouldn’t need a field for any notes.

Although we consider our diagram to be a good skeleton, where companies don’t have to remove attributes/ tables, or add tables, the differences between these three tables clearly show that there isn’t one correct answer.





**CS4347 Database System Final Project Deliverable II**

**Car Rental System**

Travis Johnson

Aaron White

Adetoye Efunnuga

Danny Choi

Ethan Payne

Zijia Ding

**Task Delegation:**

Ethan and Travis reviewed the tables to make sure that they were all normalized

Adetoye and Danny worked on the dependency diagram

Aaron and Zijia worked on creating the GUI program

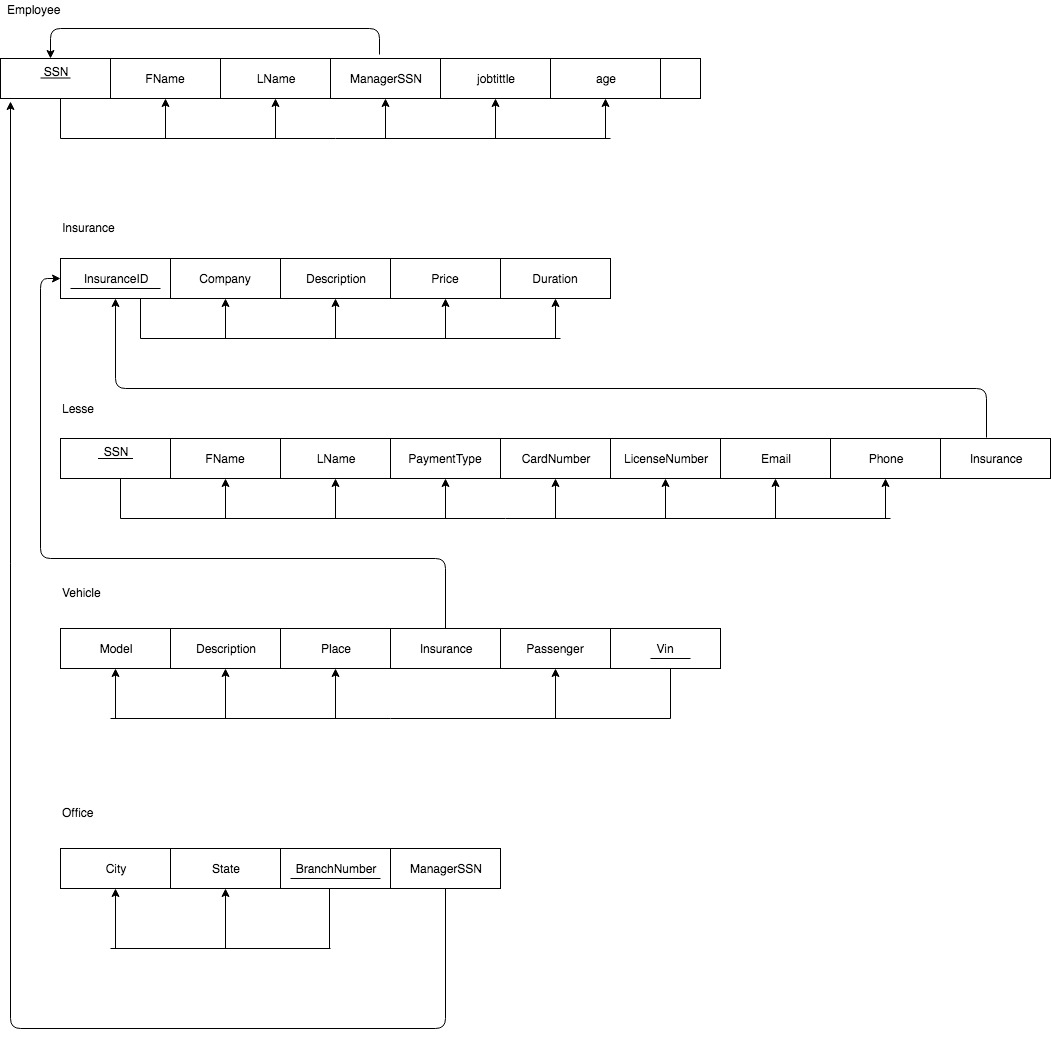
Adetoye worked on the creating the view

Travis, Danny, Adetoye, Aaron, and Zijia reviewed the report for error

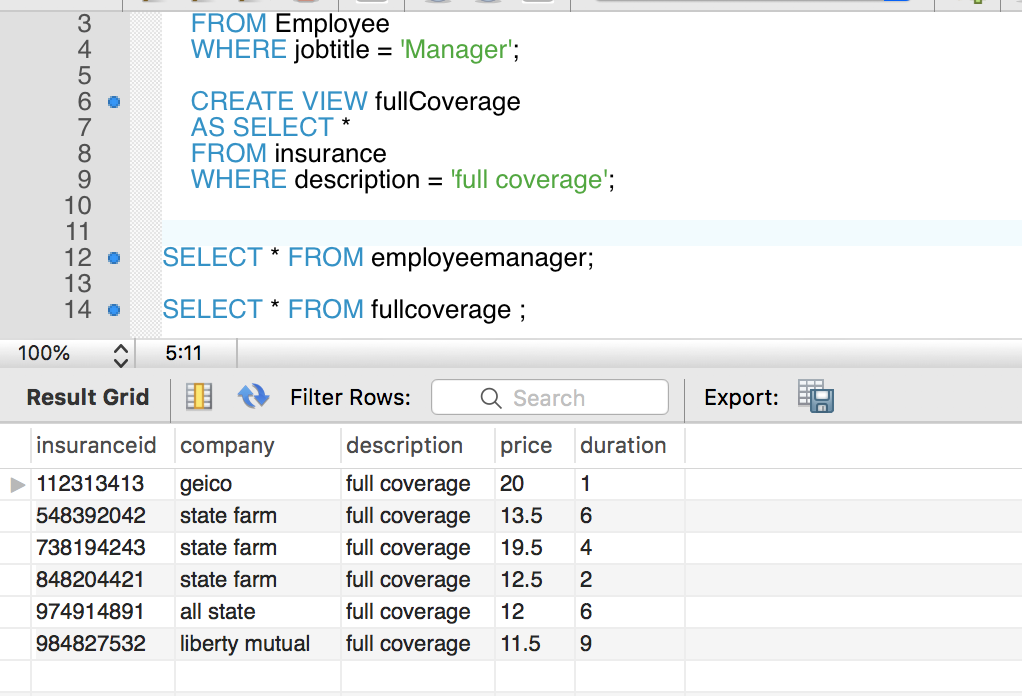
**Normalization:**

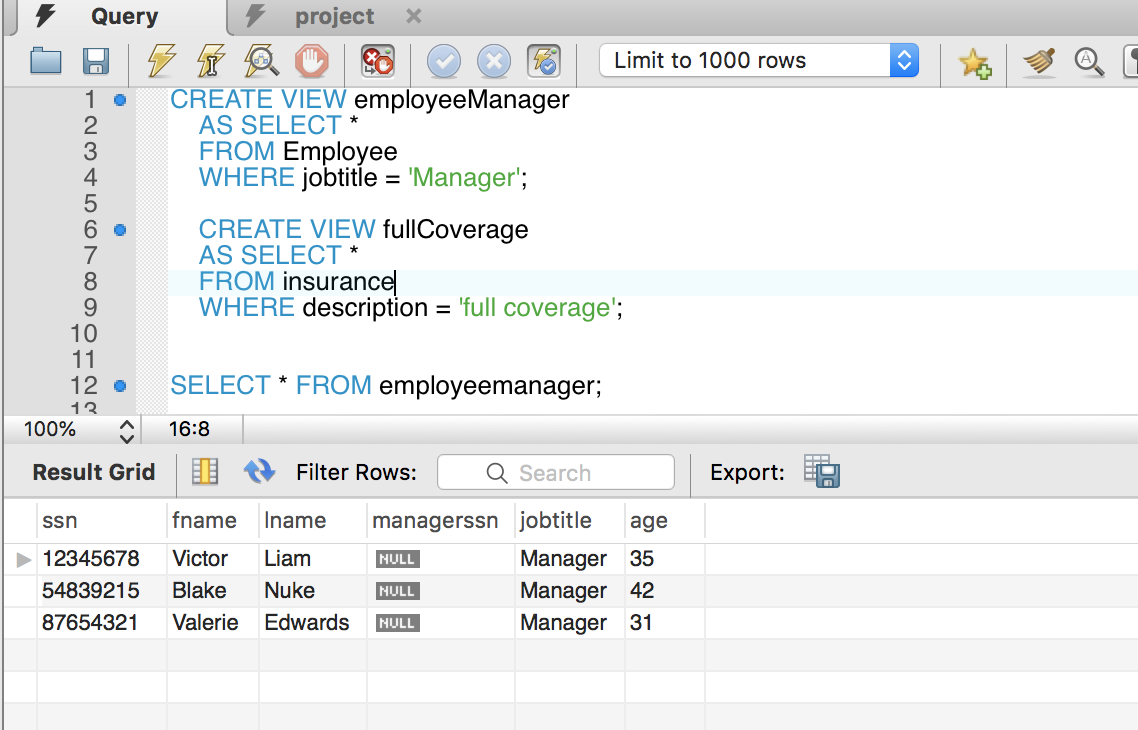
No further normalization was needed because all our table were already normalized in the initial stage. We also didn’t make any changes to our initial design because we feel like the initial design fulfils all of our intentions for the project.

**Dependency Diagram:**

****

**Create View:**





CREATE VIEW employeeManager

AS SELECT \*

FROM Employee

WHERE jobtitle = 'Manager';

CREATE VIEW fullCoverage

AS SELECT \*

FROM insurance

WHERE description = 'full coverage';

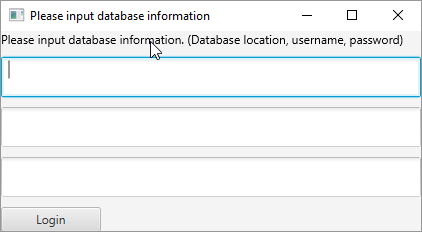
SELECT \* FROM employeemanager;

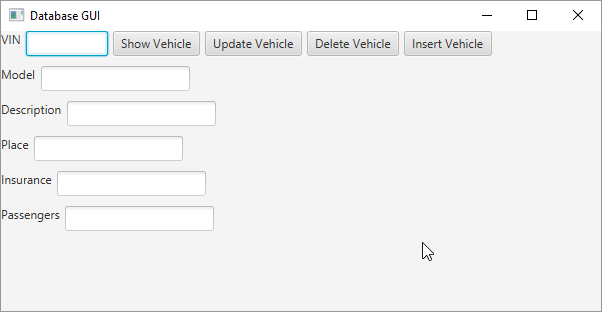
SELECT \* FROM fullcoverage ;

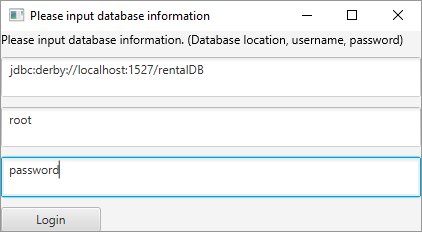
The first create view “employeeManager” displays all employees who are managers based off their job title.

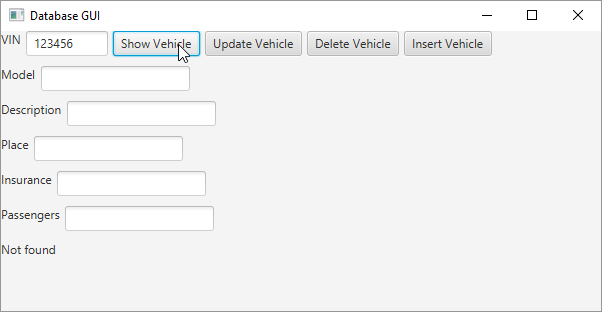
The second create view “fullCoverage” displays from the insurance table all insurance that are only full coverage.

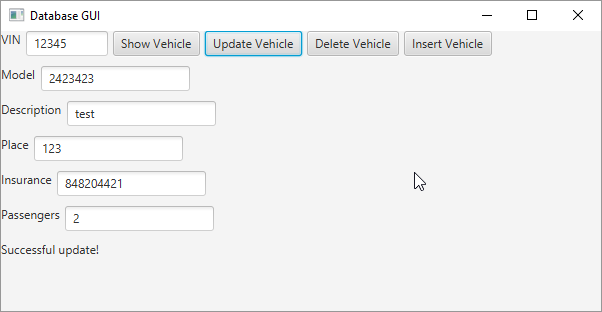
**Implementation using Java:**

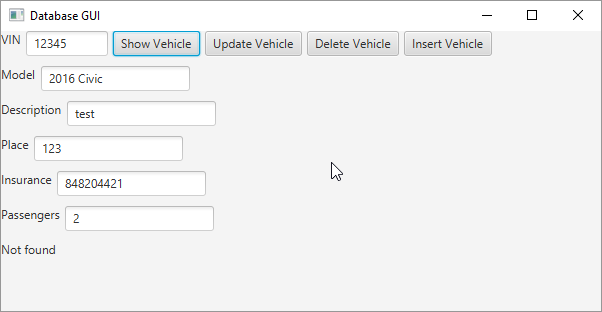
****

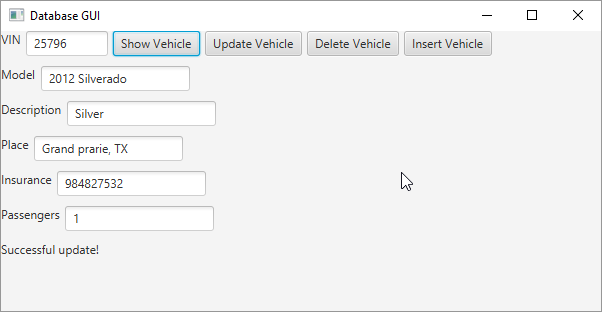
****

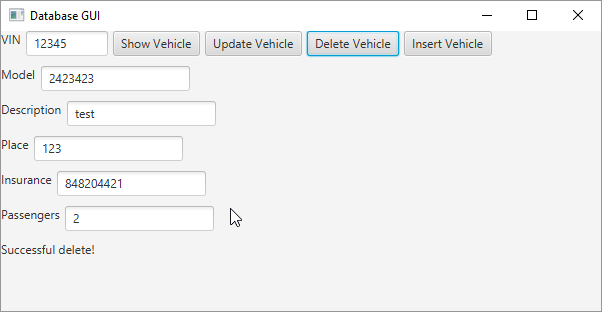
****

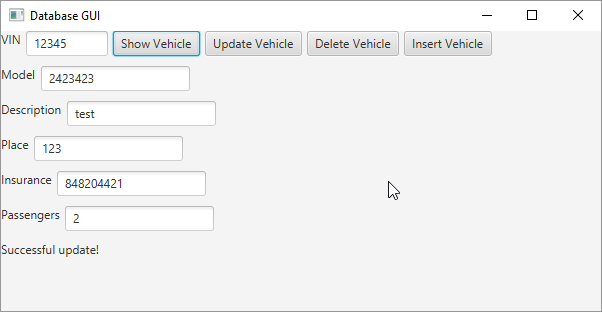
****

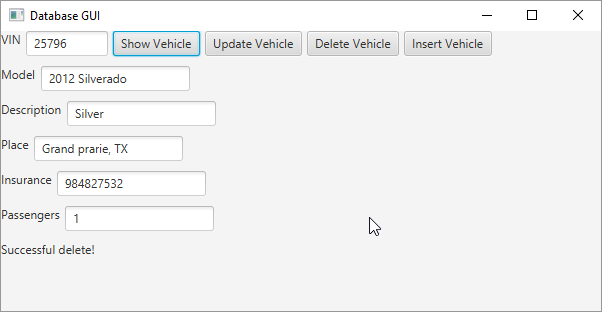
****

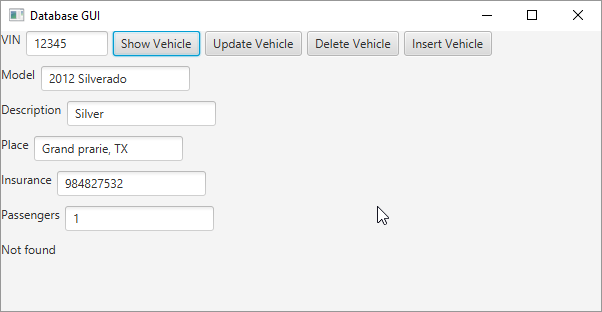
****

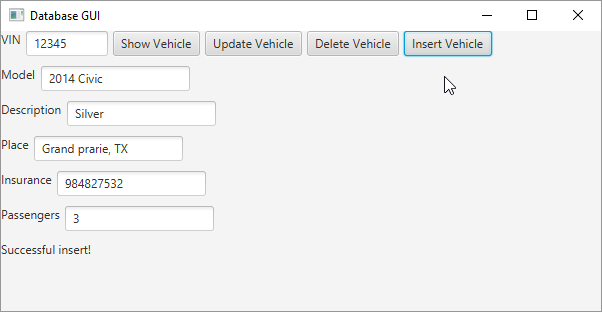
****

****

****

****

****

****

Project was demoed in class on 7/24/2018

Works Cited

S. Kher, “Renting out Cars Is as Simple as Driving: A Data Model for a Car Rental Company,”

*vertabelo*, 29-Apr-2016. [Online]. Available:

http://www.vertabelo.com/blog/technical-articles/renting-out-cars-is-as-simple-as-drivin

-a-data-model-for-a-car-rental-company. [Accessed: 26-Jun-2018].

“Telerik OpenAccess Classic,” *Configure Fiddler*. [Online]. Available:

http://docs.telerik.com/help/openaccess-classic/quickstart-carrental-sample-database.html. [Accessed: 29-Jun-2018].