

Ski & Snowboard Rental Shop



Crissy Leach & David Vehapetian

Project Motivation

We wanted to make an easy method for ski & snowboard rental stores to keep track of and maintain inventory. We were motivated to do this because we noticed that many of these type of stores do it manually and it takes too long and is prone to error, so this would be faster and more reliable.





System Description

Our system makes finding and keeping track of rental equipment easier and more efficient because we store information for each piece of equipment, what store that piece of equipment is in, when it was rented and the rental length, if it is available, and what rental package it was contained in. Our system also makes it easier to keep track of equipment repairs because we store information on each repair including which item was repaired and which employee was responsible for the repair.

System Architecture

Database Management System: SQLite

Front-end Language: Java

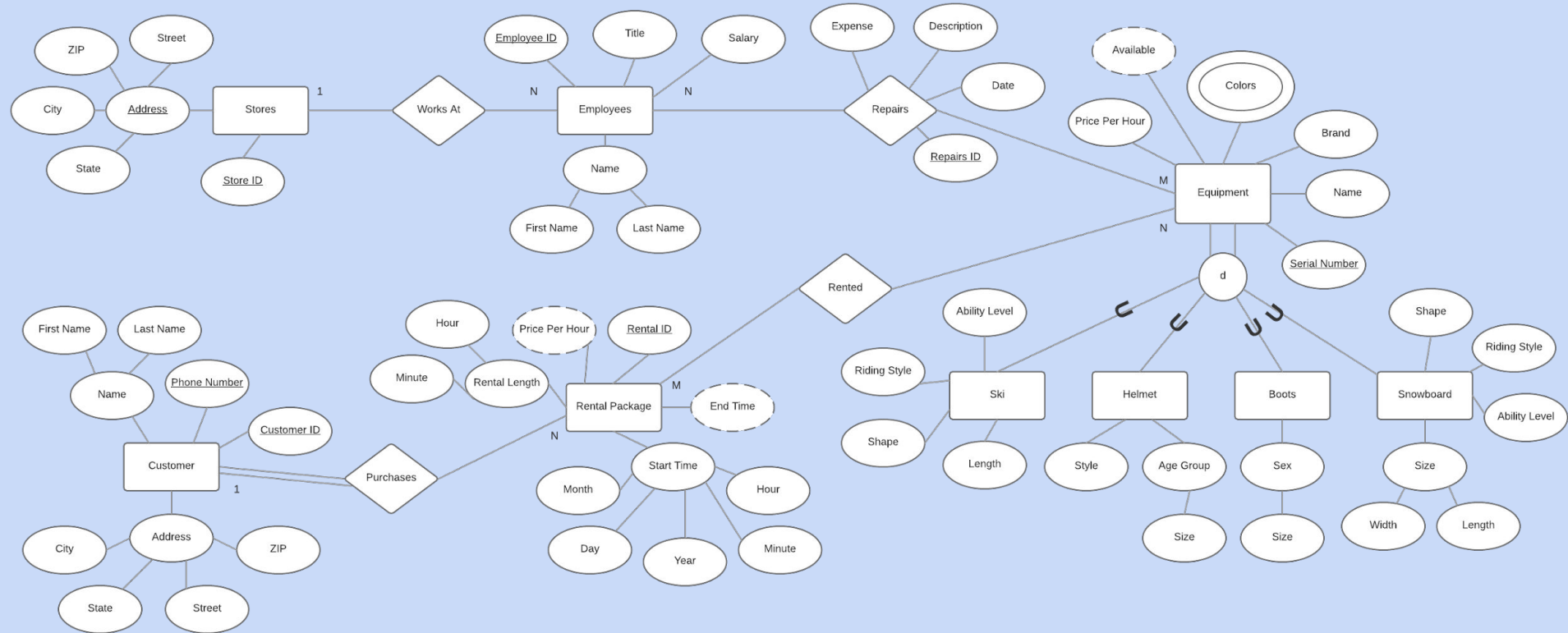
Why?

We chose to use SQLite and Java for our application because we were already familiar with both and it would be easy to convert to a web application if needed.

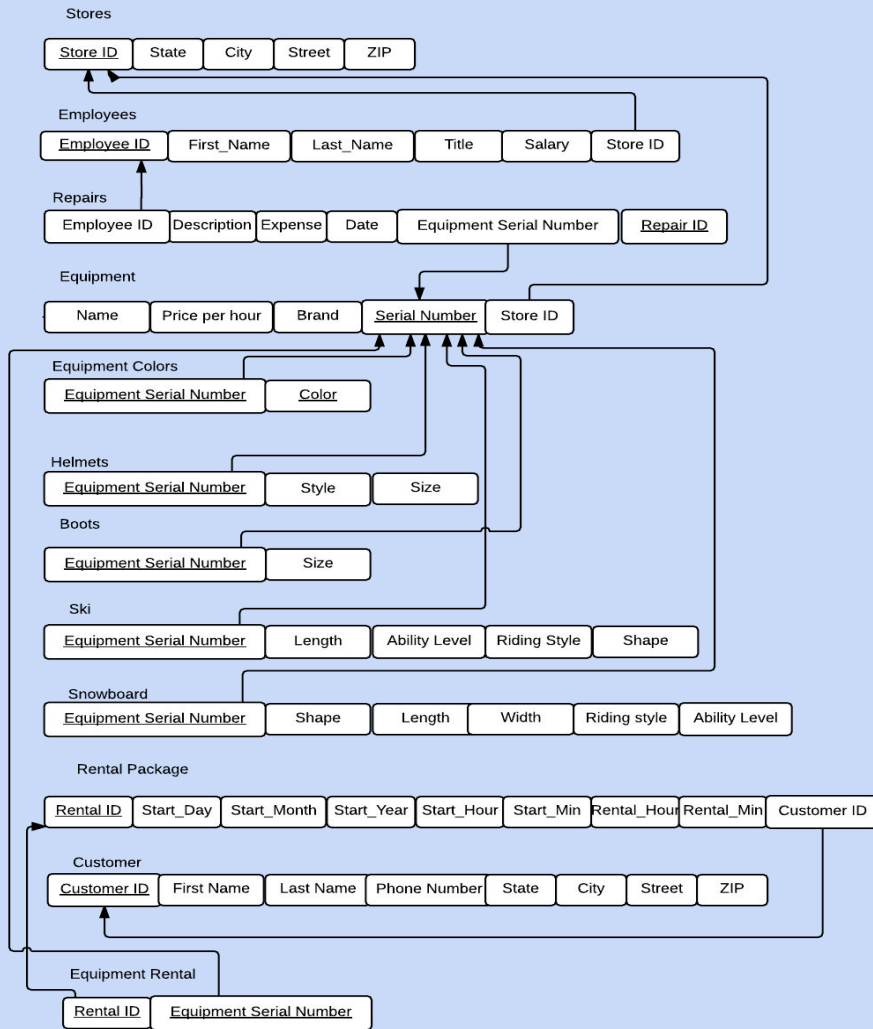
How does everything talk?

Java will open the database and ask it questions given the parameter that suits your needs. Then it will store the answer and relay it to the user.

ERD



Relational Model



Source of the Data

- The data in our rental stores is completely fabricated.
- Data about equipment is based on typical data stored at ski & snowboard rental stores.



Physical Design

Index: We did not include any indexes because we did not think it was necessary for improving query performance. There are no attributes that would be queried for frequently that have a large number of tuples. Because the indexes take up extra space (linear with the number of rows) and extra time (creation & maintenance), we felt that an index would be unnecessary.

Denormalization: We did not need to denormalize any of the tables because we did not want any redundancy and converting to 1/2NF was not necessary for optimized performance.

Materialized View: We did not need any materialized views because there wasn't any data that we constantly needed to query for that would take a long time.

Query #1

Description

Find what employees that work in the stores in a specific state (user input) that have repaired above \$50 worth of expenses.

"src/things/stuffs/RentalShop.db" 1 MA

```
Employees who made at least $50 worth of repairs:
```

```
Name          Total Repair Cost
```

```
|
```

```
Jane Doe      $65
```

Query #2

Description

Find all beginner level skis in the stores in a specific state (user input) that have a price per hour of less than \$50.

"

src/things/stuffs/RentalShop.db" 2 CA

These are the beginner level skis in given state having a price per hour less than \$50

| State | Equipment type | Equipment brand | Price per hour |
|-------|----------------|-----------------|----------------|
| CA | ski | atomic | \$45 |

Query #3

Description

Find all equipment rented in specified state (user input) within the year 2016.

"

src/things/stuffs/RentalShop.db" 3 NH

```
These are the equipment rented in given state.  
State   Equipment type   Equipment brand  
  
NH      helmet          lucky bums  
NH      boots           k2
```

Query #4

Description

Find which stores that are in the same state (user input) as a customer have snowboards that are under \$50.

"src/things/stuffs/RentalShop.db" 4 MA

These are the snowboards in your state that are less than \$50 per hour to rent.

| State | Price |
|-------|-------|
|-------|-------|

| | |
|----|------|
| MA | \$20 |
|----|------|

| | |
|----|------|
| MA | \$40 |
|----|------|

Query #5

Description

Find all snowboards that are red in a specific state (user input).

"src/things/stuffs/RentalShop.db" 5 MA

These are all the snowboards that are red in stores in your given state.

| State | Brand |
|-------|----------|
| MA | system |
| MA | system |
| MA | symbolic |

Live Demonstration!