

YouTube Certified Database - Project Step 6

Project Group 121

Frontend URL: <http://flip2.engr.oregonstate.edu:17742/>

Project Title:

Repair Shop Database

Team Members:

- Brian Spier
- Shannon Bell

Executive Summary:

- The first major changes Shannon and I made after starting this project were updates to the database table attributes. We were made aware that some of our attributes were incorrectly labeled, for example we labeled some of the foreign keys as primary keys. We also labeled some of the foreign key attributes as unique which just does not work.
- The second major change we made was adding an intersect table for the Vehicles and Customers' M:M relationship. At first we were trying to define the relationship between customers and vehicles as 1:1 but that was incorrect.
- During the step 5 draft Shannon and I were having a hard time getting the backend to communicate with the frontend using React and made the decision to switch over to handlebars.
- We had to make a second intersection table between Transactions and Repairs late in development in order to enable multiple repairs, due to a change in how we thought about using the database. This change allowed us to add the cost attribute to the repairs table. The new table also forced us to change our DDL, affected handlebar pages, app.js and our repairs, update, add, and delete pages.
- Lastly we updated our database outline, ERD and schema to match the new attribute changes made and reflect the new table addition.

Overview:

YouTube Certified is a small vehicle repair shop with a single location. YouTube Certified is looking to increase their efficiency and is in need of a database to efficiently track vehicles, customers, transactions, repairs and employees. With the new database YouTube Certified hopes to speed up the repair process by 20% by cutting down the time it takes to determine what services are needed based on service history and making transactions quick and efficient. By cataloging repairs on vehicles and which employee performed the repair YouTube certified will be able to quickly access what repair needs to be performed and which employee has the most experience with that vehicle.

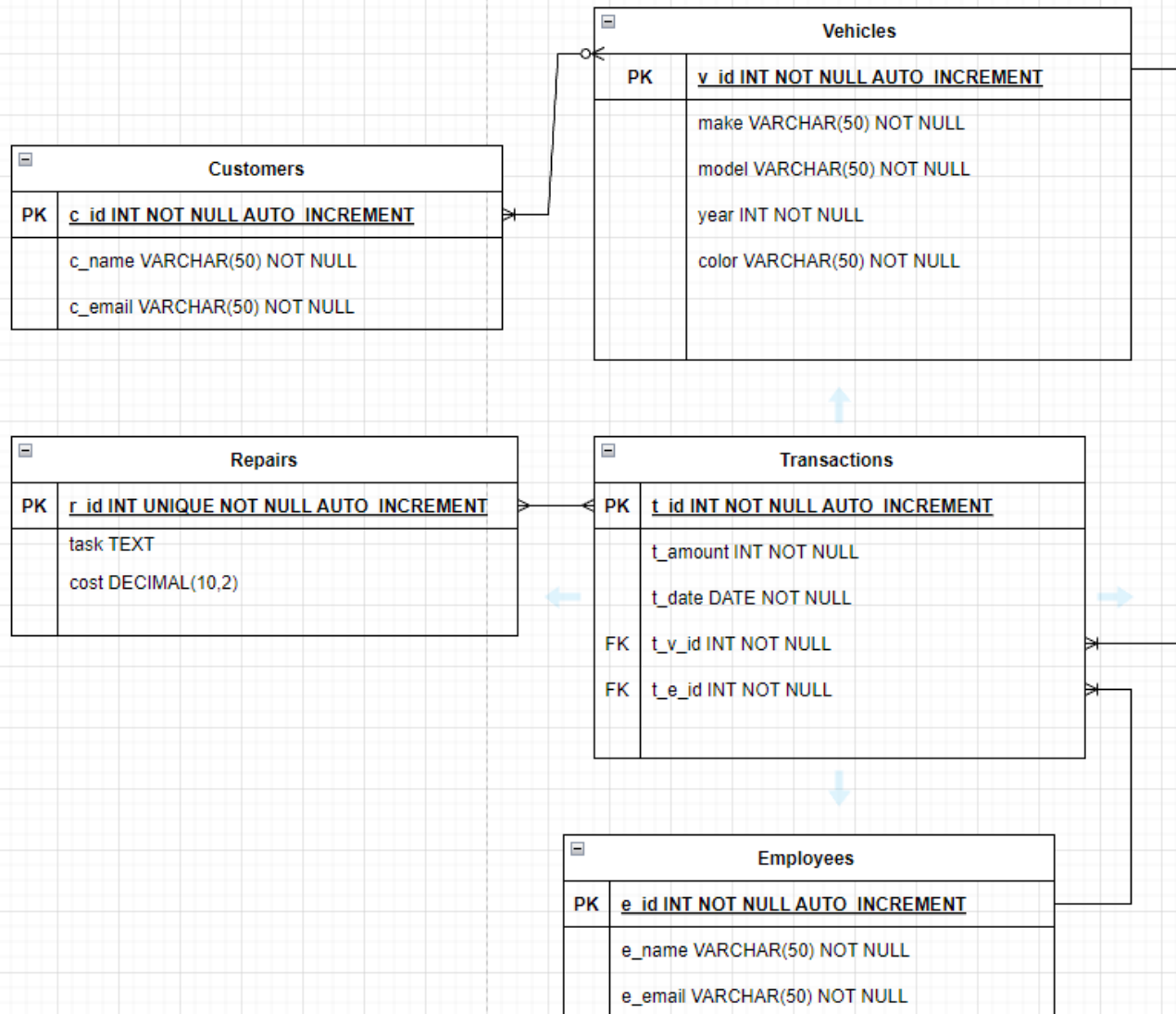
Database Outline:

- Employees: Records information regarding employees of the repair shop.
 - e_id: INT UNIQUE NOT NULL AUTO_INCREMENT PK,
 - e_name: VARCHAR(50) NOT NULL,
 - e_email: VARCHAR(50) NOT NULL,
 - Relationships:
 - 1:M relation between Employees and Transactions with e_id as FK inside of Transactions.
- Customers: Details the information about customers who have left their vehicle/s to be repaired.
 - c_id: INT UNIQUE NOT NULL AUTO_INCREMENT PK,
 - c_name: VARCHAR(50) NOT NULL,
 - c_email: VARCHAR(50) NOT NULL,
 - Relationships
 - M(mandatory):M(optional) relationship between Customers and Vehicles implemented with v_id as FK inside of Customer_Vehicle intersection table as 1:M.
- Vehicles: Records information regarding vehicles undergoing repairs.

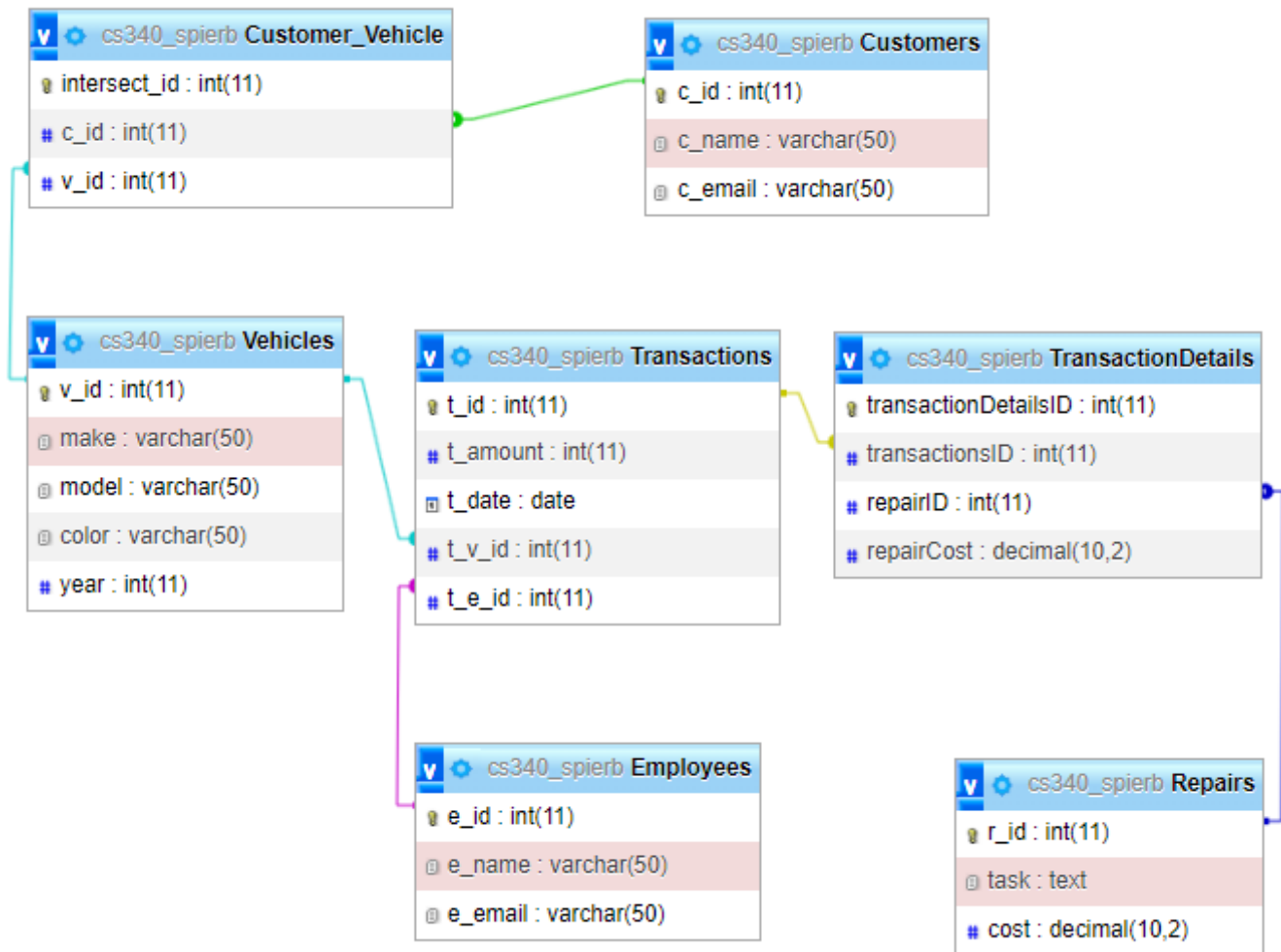
- v_id: INT UNIQUE AUTO_INCREMENT PK,
- make: VARCHAR(50) NOT NULL,
- model: VARCHAR(50) NOT NULL,
- color: VARCHAR(50), not NULL
- year: INT NOT NULL,
- Relationships
 - M(optional):M(mandatory) relationship between Vehicles and Customers implemented with v_id as FK inside of Customer_Vehicle as 1:M.
- Customer_Vehicle: Records which vehicles belong to which customers
 - intersect_id: INT UNIQUE NOT NULL AUTO_INCREMENT PK,
 - c_id: INT NOT NULL FK,
 - v_id INT FK,
 - 1:M relation between Customers and Customer Vehicle implemented with c_id as a FK inside of Customer_Vehicle
 - 1:M relation between Vehicles and Customer Vehicle implemented with v_id as a FK inside of Customer_Vehicle
- Repairs: Stores a catalog of repairs that can be performed and the cost of that repair.
 - r_id: int, increment, unique, not NULL, PK
 - task: VARCHAR(255), not NULL
 - cost: DECIMAL(10, 2),
 - Relationships:
 - M:M relationship between Repairs and Transactions implemented with r_id as FK within Transaction_Details as 1:M.
- Transactions: Logs each repair as they are done, tracking what vehicle was repaired and by which employee. The date and invoice amount are tracked.
 - t_id: INT UNIQUE AUTO_INCREMENT PK,
 - intersect_id: int, unique, not NULL, FK
 - amount: int, NOT NULL
 - date: DATETIME
 - t_r_id: int, not NULL, FK
 - t_e_id: int, not NULL, FK

- Relationships:
 - M:1 relationship between Transactions and Vehicles with v_id as FK within Transactions.
 - M:1 relationship between Transactions and Employees with e_id as a FK within Transactions.
 - M:M relationship between Transactions and Repairs implemented with t_id as FK within Transactions_Details as 1:M.
- Transaction_Details: Records the transaction this transaction detail belongs to, this also records the repair that was done and how much it cost.
 - transactionDetailsID: INT NOT NULL AUTO_INCREMENT PK,
 - transactionsID: int FK,
 - repairID: int FK,
 - repairCost: DECIMAL(10,2),
 - Relationships:
 - M:1 relationship between Transaction_Details and Transactions with t_id as FK within Transaction_Details
 - M:1 relationship between Transactions_Details and Repairs with r_id as FK within Transaction_Details

ER Diagram:



Schema:



Sample Data:

Customers

c_id	c_name	c_email
1	Charle Day	cday@hello.com
2	Jim Smith	jsmith@hello.com
3	Tom Brown	tombrown68@hello.com

Vehicles

v id	make	model	color	year
1	Toyota	Tacoma	white	1997
2	Honda	Accord	gold	2003
3	Subaru	BRZ	blue	2013

Customer_Vehicles

intersect ID	c id	v id
1	1	2
2	2	1
3	3	3

Employees

e id	e name	e email
1	Jim Bob	jimmybob@hello.com
2	Al Turnator	alturnator@hello.com
3	Stephen Duga	dugaduga@hello.com

Repairs

r id	task	cost
1	Oil Change	50.00
2	Brake Change	200.00
3	Tune Up	350.00

Transactions

t id	t amount	t date	t intersect id	t e id
1	50	2023-01-29	1	3

2	200	2022-10-15	3	1
3	350	2022-02-05	2	2

Transaction Details

transactionDetailsID	transactionsID	repairID	repairCost
1	1	1	50.00
2	3	3	350
2	2	2	200

Screen Captures:

Transactions Page: This page has create and read CRUD operations. The create operation allows the user to create a transaction by inputting the transaction amount, date, vehicle id and employee id. This page also reads and returns all of the transactions in the database and allows for searching specific transactions by transaction id.

←

→

↺

Not secure | flip2.engr.oregonstate.edu:17742

Repair Shop Log

[Transactions](#)
[Employees](#)
[Customers](#)
[Vehicles](#)
[Customer Vehicle Repairs](#)
[Transaction Details](#)

Transactions List

This page lists the log of repairs that have been done in the shop.

Transaction ID	Transaction Amount	Transaction Date	Vehicle ID	Employee ID
1	50	Sun Jan 29 2023 00:00:00 GMT-0800 (Pacific Standard Time)	1	3
2	200	Sat Oct 15 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	3	1
12	400	Fri Mar 17 2023 00:00:00 GMT-0700 (Pacific Daylight Time)	1	3

Adding New Transaction

To add a new Transaction, please enter their information below and click Submit.

Transaction Total
Transaction Date
Vehicle ID:
Employee ID:

Search for Transaction with the Transaction ID

Search Transaction by ID using the field below.

Search by Transaction ID:

Employees Page: This page has create, read, update and delete CRUD operations. The create operation allows the users to add new employees to the database by inputting the employee name and their email. The employees are displayed with the read operation and the employees can also be searched by id. This page also allows for employees' emails to be updated by selecting their name. Finally this page gives the option to delete all employees who are not listed in a transaction.

[←](#) [→](#) [↻](#) [⚠ Not secure](#) | flip3.engr.oregonstate.edu:17742/Employees

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle Repairs](#) [Transaction Details](#)

Employees List

This page lists all the employees currently within the database.

Employee ID	Employee Name	Employee Email	Delete
1	Jim Bob	jbb@hello.com	<button>Delete</button>
3	Stephen Duga	bt2@hello.com	<button>Delete</button>
32	Jeff String	jstring@hello.com	<button>Delete</button>

Adding New Employee

To add a new employee, please enter their information below and click Submit.

Name: Email: Submit

Search Employees by ID

Search employees by ID using the field below. Partial IDs work.

Search by Employee ID: Submit Reset

Updating An Employee's Email Address

To update an employee's email, select their name from the drop-down menu on the left and enter their new email address on the right

Name: Email: Submit

Customers Page: This page has create, read, update and delete CRUD operations. This page is similar to the employee page and allows customers to be added to the database. This page also allows customers to be updated and deleted. Clicking the delete button will remove the customer from the database. Updating customers will allow a user to change the customer's email in the database.

[←](#) [→](#) [↻](#) [⚠ Not secure](#) | [flip2.engr.oregonstate.edu:17742/Customers](#)

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle Repairs](#) [Transaction Details](#)

Customers List

This page lists all the customers currently within the database.

Customer ID	Customer Name	Customer Email	Delete
1	Charle Day	cday@hello.com	<button>Delete</button>
2	Jim Smith	jsmith@hello.com	<button>Delete</button>
5	brian	b@gmail.com	<button>Delete</button>
7	Al Turnator	bt2@hello.com	<button>Delete</button>
8	Al	jimbob@hello.com	<button>Delete</button>
9	timmy	timmy@hello.com	<button>Delete</button>
10	Big Todd	bt2@hello.com	<button>Delete</button>

Adding New Customer

To add a new customer, please enter their information below and click Submit.

Name: Email: Submit

Search Customers by ID

Search customers by ID using the field below. Partial IDs work.

Search by Customer ID: Submit Reset

Updating An Customer's Email Address

To update an customer's email, select their name from the drop-down menu on the left and enter their new email address on the right

Name: Email: Submit

Vehicles Page: This page has create, read, update and delete CRUD operations. This page allows users to add new vehicles to the database by inputting the cars make, model, color and year. This page also displays all the vehicles listed in the database, and allows users to search a vehicle with the vehicle id. The delete button will remove the vehicle from the database.

[←](#) [→](#) [↻](#) [⚠ Not secure](#) | [flip2.engr.oregonstate.edu:17742/Vehicles](#)

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle Repairs](#) [Transaction Details](#)

Vehicles List

This page lists all the vehicles currently within the database.

Vehicle ID	Make	Model	Color	Year	Delete
1	Toyota	Tacoma	white	1997	Delete
3	scion	frs	blue	2015	Delete
8	Nissan	350z	blue	2013	Delete
9	Nissan	350z	blue	300	Delete

Adding New Vehicle

To add a new vehicle, please enter their information below and click Submit.

Make: Model: Color: Year: [Submit](#)

Search Vehicles by ID

Search vehicles by ID using the field below. Partial IDs work.

Search by Vehicle ID: [Submit](#) [Reset](#)

Updating A Vehicles Information

To update a vehicle's information, select their ID from the drop-down menu on the left and enter the new info on the right

ID: Make: Model: Color: Year: [Submit](#)

Customer Vehicle Page: This page has create and read CRUD operations. This page is the intersection page for customers and vehicles. The page displays which customer owns which vehicle. A new entry requires the user to input the customer and vehicle associated with them.

Project Step 6 (Portfolio Assignm

Activity 2 - Connect webapp to c

College of Engineering: TEACH

← → ↺ ⚠ Not secure | flip2.engr.oregonstate.edu:17742/Customer_Vehicle

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle](#) [Repairs](#) [Transaction Details](#)

Customer_Vehicle List

This page lists who owns which vehicles that are currently within the database.

Customer Veehicle ID	Customer ID	Vehicle ID	Delete
2	2	1	<button>Delete</button>

Adding New Customer Vehicle Relation

Please choose a Customer ID and Vehicle ID to add to the Customer Vehicle list and click submit

Customer: Vehicle:

Repairs Page: This page has create, read, update and delete CRUD operations. Adding a new repair requires the user to input the task description and the cost of the task. This page displays all repairs in the database. Repair tasks and costs can be updated by selecting the repair id and inputting the desired changes. Lastly all repairs can be individually searched by repair id

Project Step 6 (Portfolio Assignm

Activity 2 - Connect webapp to c

College of Engineering: TEACH

classmysql

← → ↻ ⚠ Not secure | flip2.engr.oregonstate.edu:17742/Repairs

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle](#) [Repairs](#) [Transaction Details](#)

Repairs List

This page lists all the repairs that can be selected from in the database.

Repair ID	Task	Cost	Delete
1	Oil Change	50	<button>Delete</button>
2	Brake Change	200	<button>Delete</button>
3	Tune Up	350	<button>Delete</button>
4	Tire balance	100	<button>Delete</button>

Adding New Repair

To add a new repair offered, please enter their information below and click Submit.

Task: Cost: Submit

Search Repairs by ID

Search repairs by ID using the field below. Partial IDs work.

Search by Repair ID: Submit Reset

Updating a Repair Task

To update the task of a repair, select the ID from the drop-down menu on the left and enter the new description on the right

ID: Task: Cost: Submit

Transactions Details: This page has create and read CRUD operations. This page allows users to search individual transactions by transaction id. Users can input new transactions by inputting the transaction id and repair. Transactions details can be individually searched with the transaction id, we chose to search by transaction id because it will display all transactions associated with that transaction id.

← → ↻ ⚠ Not secure | flip2.engr.oregonstate.edu:17742/Transaction_Details

Repair Shop Log

[Transactions](#) [Employees](#) [Customers](#) [Vehicles](#) [Customer Vehicle](#) [Repairs](#) [Transaction Details](#)

Transaction Details

This page lists all the Transactions Details currently within the database.

Transaction Details ID	Transaction ID	Repair ID	Repair Cost
1	1	1	50
2	2	3	350
3	1	2	200
5	12	2	200
6	12	2	200
7	1	3	350
8	2	3	350
10	1	1	50

Adding New Transaction Details

To add new transaction details please enter the information below and click Submit.

Transaction ID:

Select a Transaction ID ▼

 Repair:

Select a Repair ▼

Submit

Reset

Search for Transaction Details with the Transaction ID

Search Transaction Details by ID using the field below.

Search by Transaction ID:

Submit

Reset