
mechanics need databases too

Prepared by Group 14: Heather Fillerup- Software Developer, Chris Nelson- Software Developer

CS 340-400: Spr 2020

Project- Step 2 Draft

April 14, 2020

Summary

Overview

Mahinui auto shop has seen record business in the last decade, repairing 50 or more cars on any given day. The owner, Brad, has finally decided to upgrade his repair order workflow from pen and paper being passed between mechanics to web and database. With more customers coming in by the day, keeping track of records has become a nightmare. Brad is looking to create a system to track and record the progress of the car repair from diagnosis to customer pick up. The website will allow the mechanics to:

1. Create a repair order
2. Associate a car and a customer to the repair order
3. Update the statuses of the repair order until the car repair is complete or the customer decides to cancel
4. View a table that displays all of the cars currently being repaired at the shop and the progress of each repair

Database outline

customers: records details about customers who own the car (Heather)

- id: int, auto_increment, unique, not NULL, PK
- f_name: varchar, not NULL
- l_name: varchar, not NULL
- contact_no: varchar, not NULL
- email_address: varchar
- street_address: varchar, not NULL
- city: varchar, not NULL
- state: varchar, not NULL
- zip_code: int, not NULL

-
- relationship: a 1:M relationship between customers and cars is implemented with customer_id as a FK inside of cars

cars: records details about the car being repaired (Chris)

- id: int, auto_increment, unique, not NULL, PK
- customer_id: int, not NULL FK (tracks primary owner)
- license_plate_no: varchar, not NULL
- make: varchar, not NULL
- model: varchar, not NULL
- year: int, not NULL
- description: varchar
- relationship: a 1:M relationship between cars and repairs is implemented with car_id as a FK inside of repairs; a 1:M relationship between customers and cars is implemented with customer_id as a FK inside of cars

repairs: records details about the repair order being done on a car (Heather and Chris)

- id: int, auto_increment, unique, not NULL, PK
- car_id: int, not NULL, FK
- total_hours: int (+ = each status hour)
- total_parts_cost: int (+ = each parts total cost)
- total_cost: decimal (calculate based on total hours * shop rate + total parts cost)
- date_received: date
- date_completed: date
- relationship: a M:M relationship between repairs and statuses is implemented with a relationship table repairs_statuses; a M:M relationship between parts and repairs is implemented with a repairs_parts relationship; a 1:M relationship between cars and repairs is implemented with car_id as a FK inside of repairs

statuses: records detail about the various statuses of the repair order (Heather and Chris)

- id: int, auto_increment, unique, not NULL, PK
- category: varchar, not NULL (diagnosis, parts ordered, parts received, repair started, repair finished, test driven, customer paid, customer picked up)
- mechanic_id: int, not NULL, FK
- start_date: date, not NULL
- end_date: date

-
- current_status: Boolean, not NULL
 - complete: Boolean, not NULL
 - hours: int
 - relationship: a M:M relationship between repairs and statuses is implemented with a relationship table repairs_statuses; a 1:M relationship between mechanics and statuses implemented with mechanic_id as a FK inside of statuses

mechanics: records details of the mechanic working on the statuses (Chris)

- id: int, auto_increment, unique, not NULL, PK
- f_name: varchar, not NULL
- l_name: varchar, not NULL
- rate: int
- relationship: a 1:M relationship between mechanics and statuses implemented with mechanic_id as a FK inside of statuses

Shema



