

Table of Contents:

Burdell's Ramblin' Wrecks Data Types

[Data Types](#)

Burdell's Ramblin' Wrecks Constraints

[Business Logic Constraints](#)

Burdell's Ramblin' Wrecks Data Types

[Login](#)

[Add Vehicle Information](#)

[Record Sales Transaction](#)

[Public Search](#)

[Registered Users Search](#)

[Generate Reports](#)

[Add Repair](#)

[View Vehicle Information](#)

Data Types:

Based on PostgreSQL data types.

Vehicle

Attribute	Data Type	Nullable
Color	Array(varchar(n))	Not Null
VIN	char(17)	Not Null
Mileage	integer	Not Null
Type	varchar(n)	Not Null
Model Name	varchar(n)	Not Null
Model Year	char(4)	Not Null
Condition	varchar(n)	Not Null
Description	text	Null
Manufacturer	varchar(n)	Not Null
Purchase Price	float	Not Null
Purchase Date	date	Not Null
Seller	varchar(n)	Not Null
Inventory Clerk	varchar(n)	Not Null
Sales Price	float	Null
Sales Date	date	Null
Buyer	varchar(n)	Null
Salesperson	varchar(n)	Null

Customer

Attribute	Data Type	Nullable
Email	varchar(n)	Null
Street	varchar(n)	Not Null
City	varchar(n)	Not Null
State	varchar(n)	Not Null
Zip Code	char(5)	Not Null
Phone number	varchar(n)	Not Null
First name	varchar(n)	Null
Last name	varchar(n)	Null
DL Number	varchar(n)	Null
Business name	varchar(n)	Null
Tax ID	integer	Null
Primary Contact Name	varchar(n)	Null
Primary Contact Title	varchar(n)	Null

Repairs

Attribute	Data Type	Nullable
VIN	varchar(17)	Not Null
Status	varchar(n)	Not Null
Vendor name	varchar(n)	Not Null
Description	text	Not Null
Start Date	date	Not Null
End Date	date	Not Null
NHTSA Recall Number	varchar(n)	Null
Total Cost	float	Not Null

Recalls

Attribute	Data Type	Nullable
NHTSA Recall Number	varchar(n)	Not Null
Recall Description	text	Not Null
Associated Manufacturer	varchar(n)	Not Null

Vendors

Attribute	Data Type	Nullable
Vendor name	varchar(n)	Not Null
Street	varchar(n)	Not Null
City	varchar(n)	Not Null
State	varchar(n)	Not Null
Zip Code	char(5)	Not Null
Phone number	varchar(n)	Not Null

User

Attribute	Data Type	Nullable
First Name	varchar(n)	Not Null
Last Name	varchar(n)	Not Null
Username	varchar(n)	Not Null
Password	varchar(n)	Not Null
Category	varchar(n)	Not Null

Manufacturers

Attribute	Data Type	Nullable
Manufacturer Name	varchar(n)	Not Null

Colors

Attribute	Data Type	Nullable
Name of Color	varchar(n)	Not Null

Vehicle Types

Attribute	Data Type	Nullable
Vehicle Type	varchar(n)	Not Null

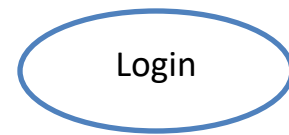
Business Logic Constraints:

Burdell's Ramblin' Wrecks

- Privileged access is given at the database level, not the application level.
- Logging in as a user enables additional features.
- Looking up or adding customers is available to all employees, but only when recording a purchase or sales transaction.
- Inventory clerks add the vehicle to inventory, along with customer and repair information.
- Salespeople can search inventory, enter customer information and record sales transactions.
- Managers can view inventory, purchase history, sales transactions, repair history, and reports.
- The business owner has access to everything.
- A user can only have one of the permission sets mentioned above.
- The list of vehicle types may be updated.
- The list of manufacturers may be updated.
- The list of vehicle colors is not expected to change.
- The entered model year cannot exceed the current year plus one.
- The model year must include century digits.
- Purchase prices are determined by the KBB value for the given vehicle and condition.
- Vehicles cannot be publicly searched or sold if under repair.
- The sales price is calculated as 125% of the original purchase price combined with 110% of any repair costs associated with the vehicle.
- If a buyer purchases multiple vehicles at the same time, they are recorded as separate sales transactions.
- Repair dates cannot overlap.
- The sales price of a vehicle cannot be changed; bargaining is not allowed.

Login

Task Decomp



Lock Types: Read-only on Users table

Number of Locks: Single

Enabling Conditions: None

Frequency: Several per day

Consistency (ACID): not critical, order is not critical

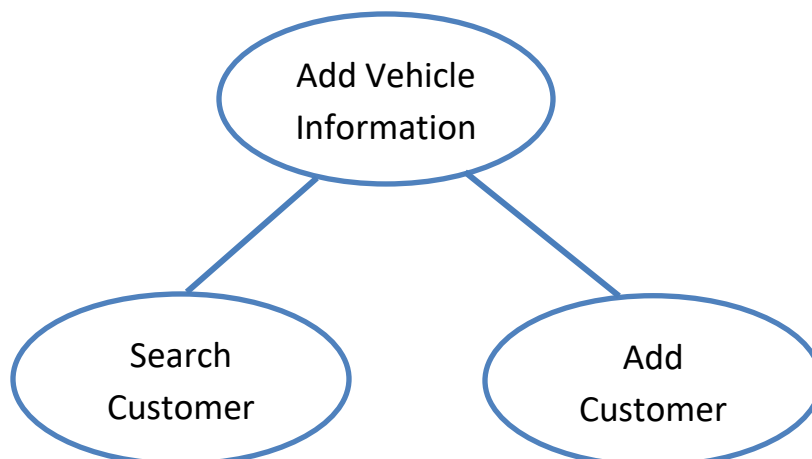
Subtasks: Mother task is not needed. No decomposition needed.

Abstract Code

- User clicks **login** button on public-facing search screen.
- User is brought to login form.
- User enters *email*, *password* input fields.
- If data validation is successful for both *username* and *password* input fields, then:
 - When **Enter** button is clicked:
 - If User record is found but password does not match:
 - Go back to Login form, with error message.
 - Else:
 - Return to public-facing search screen with appropriate report statistics based on user category (e.g. number of vehicles in repair).
- Else *username* and *password* input fields are invalid, display Login form, with error message.

Add Vehicle Information

Task Decomp



Lock Types: Write on Vehicles table, Read/Write on Customers table

Number of Locks: Several different schema constructs needed

Enabling Conditions: User with appropriate permission login and adds vehicle

Frequency: Low

Consistency (ACID): not critical, car cannot be sold until entered.

Subtasks: Mother task required. Search is done before adding customer.

Abstract Code

- User must have appropriate permission to add vehicle.
- User clicks **Add Vehicle** button and is brought to Add Vehicle form.
- User enters customer name in *seller* input field and clicks **Search** button.
- When **Search** button is clicked:
 - If correct customer record is found:
 - User selects customer record and seller is linked to vehicle.
 - Else:
 - User is shown “No Match” and prompted to add customer information.
- If user needs to add customer information:
 - User clicks **Add Customer** button.
 - User fills out Customer Information form including all required fields:
 - *Street*
 - *City*
 - *State*
 - *Postal Code*
 - *Phone Number*
 - *Email* (Optional)
 - If individual:
 - *First Name*
 - *Last Name*
 - *Driver’s License Number*
 - If business:
 - *Tax ID Number*
 - *Business Name*
 - *Primary Contact Name*
 - *Primary Contact Title*
 - User clicks **Submit** button.
 - If data validation is successful for all required fields:

- Customer is added to the system and linked to the vehicle.
- Else:
 - Fields that are invalid show an error message.
- Once customer is selected, user fills out **Vehicle Information** form.
- User enters the *VIN, vehicle type, condition, manufacturer, model year, model name, color(s), description, mileage, Blue Book value, and date of purchase* fields.
- User clicks **Submit** button.
- If data validation is successful for all required fields:
 - User is brought to the detail page for the vehicle.
- Else:
 - Invalid fields show an error message.

Record Sales Transaction

Task Decomp



Lock Types: Read/Write on Vehicles table, read-only on Customers table

Number of Locks: Single

Enabling Conditions: User with appropriate permission login and searches for vehicle to record sale

Frequency: Low

Consistency (ACID): Critical, two people cannot sell the same car

Subtasks: Mother task required. Search is done before adding customer.

Abstract Code

- User must have appropriate permission to sell a vehicle.
- User searches for the vehicle to be sold using the **Registered Search** task.
- User selects correct vehicle and is taken to the vehicle detail page.
- User clicks **Sell Vehicle** button and is brought to the **Sales Order** form.
- User enters customer name in *buyer* input field and clicks **Search** button.
- When **Search** button is clicked:
 - If correct customer record is found:
 - User selects customer record and buyer is linked to the sale.
 - Else:
 - User is shown “No Match” and prompted to add customer information.
- If user needs to add customer information, they will complete the **Add Customer** task like in the **Add Vehicle Information** task.
- Once customer is selected and linked to the sale, user fills in the *sales date* input field.
 - If *sales date* input field is valid (not in the future and a correct date):
 - The sale is recorded, and user is brought back to the **Registered User Search** form.
 - If *sales date* input field is invalid:
 - User is prompted to enter a valid date.

Public Search

Task Decomp

Lock Types: Read-only on Vehicles table

Number of Locks: Single

Enabling Conditions: None

Frequency: Many per day

Consistency (ACID): not critical, order is not critical

Subtasks: Mother task is not needed. No decomposition needed.



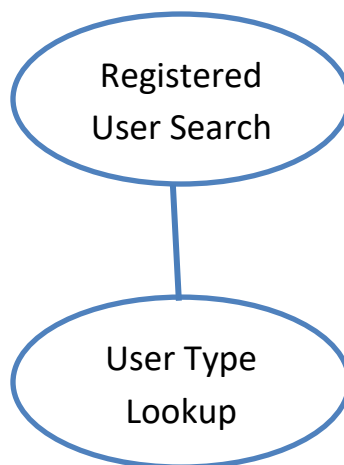
Abstract Code

- User is shown the **Public Search** form.
- User selects the search criteria field from the following options:
 - Vehicle type
 - Manufacturer

- Model year
- Color
- Keyword
- User enters *search term* input field and clicks **Search** button.
- If unsold vehicles match the *search term* input field under the *search criteria* input field and are not under repair:
 - User is shown the following attributes for each result:
 - VIN
 - Vehicle type
 - Model Year
 - Manufacturer
 - Model
 - Color(s)
 - Mileage
 - Sales Price
- Else:
 - User is displayed the following message: “Sorry, it looks like we don’t have that in stock!”
- User can select an individual result. They would be taken to the vehicle detail page and complete the **View Vehicle Information** task.

Registered Users Search

Task Decomposition



Lock Types: Read-only on Vehicles and Users tables

Number of Locks: Single

Enabling Conditions: User logs in and initiates a search

Frequency: Many per day

Consistency (ACID): not critical

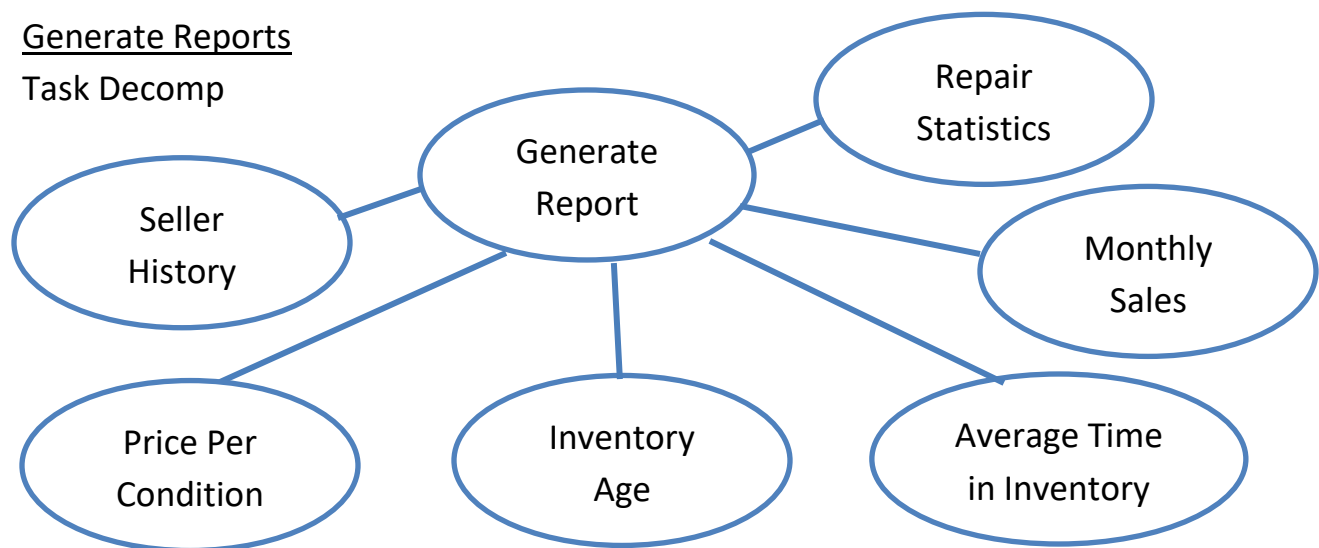
Subtasks: Mother task required. User category must be looked up.

Abstract Code

- User must first complete the **Login** task.
- User initiates a search like the **Public Search** task. However, the VIN is added to the *search criteria* input field.
- User searches for a vehicle like the **Public Search** task.
- If user is Inventory Clerk:
 - Results will show unsold vehicles including those under repair.
- If user is Salesperson:
 - Results will show unsold vehicles not under repair.
- If user is Manager or Owner:
 - Results will show sold and unsold vehicles both under repair and not.
 - User can filter results by sold/unsold vehicles using the **Filter** button on the search results page.
- User can select an individual result. They would be taken to the vehicle detail page and complete the **View Vehicle Information** task.

Generate Reports

Task Decomp



Lock Types: Read-only on all tables

Number of Locks: Several different schemas needed

Enabling Conditions: User with appropriate permission login and runs report

Frequency: Low

Consistency (ACID): not critical

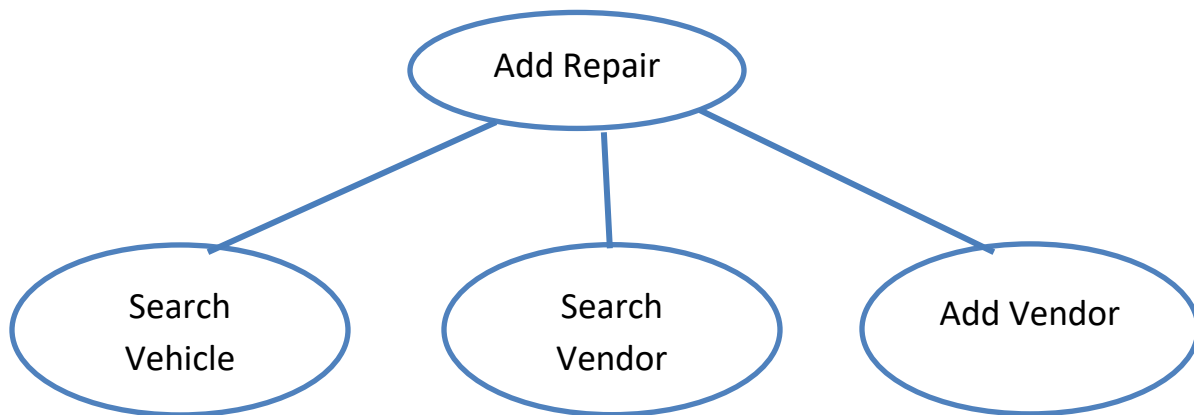
Subtasks: Mother task required.

Abstract Code

- User with appropriate permission selects *desired report* dropdown input field from initial search page.
- User clicks **Run Report** button.
- User is brought to report page matching the *desired report* field.
- Based on the report selected, user is shown different information.

Add Repair

Task Decomp



Lock Types: Read/Write on Vehicles, Vendors, Repairs, Recalls tables

Number of Locks: Several different schemas needed

Enabling Conditions: User with appropriate permission login, searches vehicle, and adds repair

Frequency: Low

Consistency (ACID): not critical, order is not critical

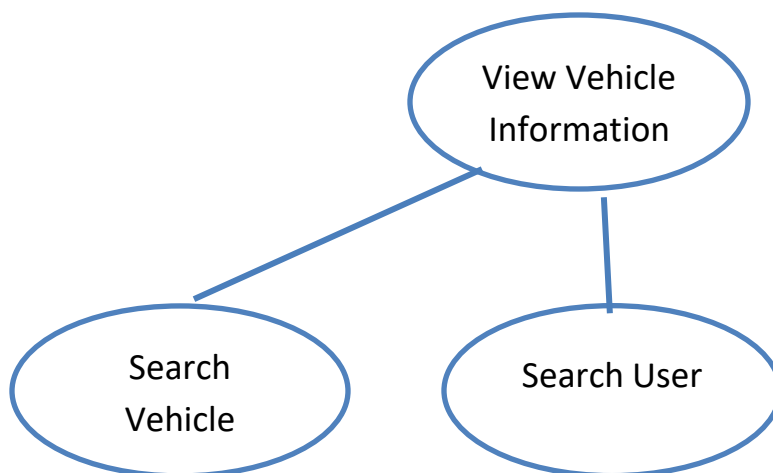
Subtasks: Mother task required

Abstract Code

- User completes the **Registered User Search** task to find the vehicle in question.
- On the vehicle detail page, user clicks the **Add Repair** button and is shown the **Add Repair** form.
- User enters all required information in the input fields.
 - User searches for an existing vendor for the repair.
 - If a vendor does not exist:
 - User enters vendor information using the **Add Vendor** form.
 - If a repair is linked to a recall:
 - User searches for an existing recall.
 - If a recall does not exist:
 - User enter recall information in the **Add Repair** form.
- User clicks the **Submit** button.
- If data validation succeeds for all required fields:
 - User is shown the updated vehicle detail page.
 - Repair is marked as “pending”. User will need to update the repair to “in progress”.
- Else:
 - Invalid fields are shown with an error message.

View Vehicle Information

Task Decomposition



Lock Types: Read-only on Vehicles, Repairs, Customers tables

Number of Locks: Multiple schemas needed.

Enabling Conditions: User searches for vehicle

Frequency: Many per day

Consistency (ACID): not critical, order is not critical

Subtasks: Mother task needed.

Abstract Code

- User completes either the **Public Search** or **Registered User Search** task.
- Once the user selects the desired vehicle, they are taken to the vehicle detail page.
- Based on the user type, the vehicle detail page will look different:
 - If a Public Search:
 - Vehicle detail page will show VIN, vehicle type, Model Year, Manufacturer, color(s), mileage, sales price, and the description of the car.
 - If an Inventory Clerk:
 - Vehicle detail page will show public information plus fields for the original purchase price and the total of all repair costs.
 - If a Salesperson:
 - Vehicle detail page will show public information plus a ***Sell Vehicle*** button.
 - If a Manager or Owner:
 - Vehicle detail page will show all information for the car including all of the seller's contact information (everything except their driver's license or tax ID number), the name (first and last) of the inventory clerk that purchased the car, the original purchase price, the purchase date, the total cost of repairs, and a repairs section. if the car has been sold, the buyer's contact information (everything except their driver's license or tax ID number), sales date, and the salesperson's name (first and last) will be displayed.
 -