

Table of Contents:

Jaunty Jalopies Datatypes

[Datatypes](#)

Jaunty Jalopies Constraints

[Business Logic Constraints](#)

Task Decomposition with Abstract Code:

[Search page/Display Main Menu](#)

[View Vehicle Details](#)

[Login](#)

[Sales order](#)

[Repair](#)

[Sales by Color](#)

[Sales by Type](#)

[Sales by Manufacturer](#)

[Gross Customer Income](#)

[View Drill-Down Customers](#)

[Repairs by Manufacturer/Type/Model](#)

[View Drill-Down Manufacturer](#)

[Below Cost Sales](#)

[Average Time in Inventory](#)

[Parts Statistics](#)

[Monthly Sales](#)

[Monthly Sales Drill-Down](#)

Data Types:**PrivilegedUser**

Attribute	Data Type	Nullable
username	String	Not Null
password	String	Not Null
first_name	String	Not Null
last_name	String	Not Null
role	String	Not Null

IndividualCustomer

Attribute	Data Type	Nullable
address	String	Not Null
phone_number	String	Not Null
email_address	String	Null
first_name	String	Not Null
last_name	String	Not Null
driver_license_number	String	Not Null

BusinessCustomer

Attribute	Data Type	Nullable
address	String	Not Null
phone_number	String	Not Null
email_address	String	Null
tin	String	Not Null
name	String	Not Null
primary_contact_name	String	Not Null
primary_contact_title	String	Not Null

Vehicle

Attribute	Data Type	Nullable
manufacturer_name	String	Not Null
model_name	String	Not Null
model_year	Integer	Not Null
invoice_price	Float	Not Null
color	List<string>	Not Null
description	String	Null
vin	String	Not Null
type	String	Not Null
number_of_doors	Integer	Null
roof_type	String	Null
back_seat_count	Integer	Null
cargo_capacity	Float	Null
cargo_cover_type	String	Null
number_of_rear_axles	Integer	Null
has_driver_sideback_door	Boolean	Null
number_of_cup_holders	Integer	Null
drive_train_type	String	Null

Repair

Attribute	Data Type	Nullable
description	String	Not Null
completion_date	Date	Null
odometer_reading	Float	Not Null
labor_charges	Float	Null
start_date	Date	Not Null

Part

Attribute	Data Type	Nullable
part_number	String	Not Null
vendor_name	String	Not Null
quantity	Integer	Not Null
price	Float	Not Null

Sale

Attribute	Data Types	Nullable
purchase_date	Date	Not Null
sold_price	Float	Not Null
sales_person	String	Not Null
list_price	Float	Not Null

Business Logic Constraints:

Vehicle

1. Model years cannot exceed the current year plus one.

Customer

N/A

Sales

1. List price is initially 125% of invoice price, which can be negotiated later by customers.
2. Buyer (customer) can purchase several vehicles, and if purchased at same time, should be handled as separate sales transactions.

Repair

1. It cannot be assumed that a repair for a vehicle is for the customer that originally bought it.
2. The date a repair is completed cannot be before start_date.
3. Vehicle will never have more than one repair starting on the same date.
4. A repair must be completed before a new one can be started.
5. Total cost of repair is the sum of cost of all parts used for repair and labor charges.

Anonymous Access

1. Allow for searching vehicles without login.
2. Use dropdown menus for fields other than keyword or list price - just allow for selecting single value and only return unsold vehicles.
3. Results sorted by VIN in ascending order.

Privileged Access

N/A

Inventory Clerks

1. After an inventory clerk logs in, given access to the “Add Vehicle” button or link, that will allow them to add new vehicles.

Salespeople

1. After logging in, on the search page, with the same layout as a public search, start with the added option to search by VIN. Upon loading the detail page, salespeople see the same details as customers with an added button or link to sell vehicles that will load the sales order form.
2. If the sold price is less than or equal to 95% of invoice price, the system will eject the same.

Service Writer

1. After login, service writers have a link or button to open the repair form where enter VIN.
2. If no repairs are open, the repair form will allow the user to start a new repair.
3. If a vehicle has unfinished repair, repair form will only allow for updating labor charges, adding parts, or completing repair. Updates to labor charges cannot be less than previous value.

Managers

1. Have view-only access to all information along with reports.
2. After logging in, start on a search screen similar to inventory clerks and salespeople.

Roland Around

1. Complete access and able to view all information and reports and do any activity.
2. Sales order form allows Roland to enter sold prices that are less than or equal to 95% of the invoice price.
3. The repair form will allow Roland to update the labor charges on a repair to a value less than their previous value.

Reports

Sales by Color

1. If a vehicle has multiple colors, it should be classified as multiple and not included in the count of vehicles that have only one color.
2. If color has no sales, it should be "0".

Sales by Type

1. List vehicles sold by type in previous 30 days, previous year, and over all time, starting from last available sale date.
2. If type has no sales, it should be "0".

Sales by Manufacturer

1. If a manufacturer doesn't not have any sales, it should be included on the report.

Gross Customer Income

1. Report will list for top 15 customers the gross, not net, income received from them via vehicle sales and/or repairs, including any repairs in progress.
2. List of customers will be by gross income descending and by last/sale date descending.
3. Repairs list should be sorted by start date descending, end date descending, and VIN ascending - however any incomplete repairs should be listed before completed ones with the same sorting criteria.

Repairs by Manufacturer/Type/Model

1. Manufacturers who's vehicles do not have any repairs should be listed on this report, and the report should be sorted by manufacturer name ascending.

Below Cost Sales

1. For a sale whose ratio is less than or equal to 95%, the background of that row should be highlighted red.
2. Sales should be listed by sales date descending and ratio descending.

Average Time in Inventory

1. If a vehicle type has no sales history, the report should display N/A for that vehicle type.

Parts Statistics

N/A

Monthly Sales

1. Summary page with lists for all sales transactions, by year and month, the total number of vehicles sold, the total sales income, the total net income (sold price less invoice price), and the sold price/invoice ratio as a percentage.
2. If a year or month does not have sales data, it can be excluded.
3. When ratio is greater than or equal to 125%, its row should be highlighted green.
4. If ratio is less than or equal to 110%, highlight yellow.
5. Results ordered by year and month descending with the most recent year and month as the first result.
6. From each year/month result, a drill down report must be accessible. Based on the sales data for that year and month, drill down will display the top performing salesperson, by showing salesperson's first and last name, number of vehicles they sold in that year and month and their total sales for that year and month.
7. Top sales person determined by sorting total vehicles descending followed by total sales descending.

Tasks Decompositions and Abstract Code:

Task

Screen/Page

entity

input field

Button

'\$session variable'

Relationship

Searching for vehicles/Display Main Menu

Task Decomp



Lock Types: Lookup on vehicle table.

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 250 times a day

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- (Initial state of the application and upon **Return to Main Menu** button clicks.)
- Clear all input fields
- Disable additional features unless the user is already login('\$UserType' is not null or anonymous).
- Show **search** and **login** button on the Main Menu screen
- Query for information of counts of **Vehicle** using Vehicle.VIN, show this information as the total number of vehicles available for purchase in the system in the Main Menu screen
- Find available choices of searching input fields (Vehicle type, using **Vehicle.type**, Manufacturer, using **Vehicle.Manufacturer**, Model year, using **Vehicle.Model** and **Vehicle.Year**, Color, using **Vehicle.Color**) and fill available choices in the dropdowns for input fields in the Main Menu screen
- All users are able to fill in **List price** and **Keyword** input fields. Users also are able to select a **vehicle type/Manufacturer/Model/Year/Color/VIN(if enabled)** from the dropdown list in each input field in the Main Menu screen
- Upon:
 - Click the **search** button:
 - Read **Type, manufacturer, model, year, color, list price, keyword, VIN(if enabled)** input fields from the Main Menu screen

- if data is valid and if there are **Vehicle** that matches the search criteria:
 - Display the matching **Vehicle** as a list sorted by **Vehicle.VIN** in ascending order on **Main Menu screen**
 - Session variable '\$vehiclevin' = Vehicle.VIN
 - Users are allowed to select an individual result from the list. If the user selects a **Vehicle**:
 - Jump to the **View Vehicle Details** task.
- Otherwise, display the message "Sorry, it looks like we don't have that in stock!" on the **Main Menu screen**
- Click the **login** button: Jump to the **Login** task:
 - If '\$UserType' returned:
 - '\$UserType'=='Inventory clerks', enable **Add Vehicle** button and **new vehicle form** on **Main Menu screen**
 - '\$UserType'=='Salespeople', enable input field **VIN** in searching criteria on **Main Menu screen**
 - '\$UserType'=='Service Writer', enable **repair form** button on **Main Menu screen**
 - '\$UserType'=='Managers', enable **report** button, input field **VIN** in searching criteria, option to filter by sold vehicles, unsold vehicles, or all vehicles, Dropdown on **Main Menu screen**
 - '\$UserType'=='Roland Around', enable all the features on **Main Menu screen**
- Click the **Add Vehicle** button:
 - User will fill the **VIN**, **vehicle type**, **invoice price**, etc., along with the **date** it was added to inventory in **new vehicle form** on **Main Menu screen**.
 - Read those values from input fields and if Data is valid and **VIN** does not already exist as a Vehicle.VIN:
 - Inert new **Vehicle** instance with those values, then clear any success/error message, display a success message, '\$vehiclevin' = **VIN** and call the **View Vehicle Details** task.
- Click the **repair form** button: Jump to the **Repair** task
- Click the **report** button: Read the choice from the report dropdown menu on **Main Menu screen** then call the corresponding report task

View Vehicle Details

Task Decomp



Lock Types: Lookup vehicle from vehicle table, read only operation.

Number of Locks: No locks needed.

Enabling Conditions: none

Frequency: about 100 per day.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- User selected on Vehicle from the list on the **Main Menu screen**
- Display **Detail Page** Screen
- Enable link to **sell the vehicle** on **Detail Page** Screen if '\$UserType'=='Salespeople'
- Query for information about the **Vehicle** and it's details using '\$vehiclevin' from the HTTP Session/Cookie.:
 - Display '\$vehiclevin' on the **Detail Page**
 - Find and display the **Vehicle.Type** on the **Detail Page**
 - Find and display the attributes of the **Vehicle.Type** on the **Detail Page**
 - Find and display the **Vehicle.Model** and **Vehicle.Year** on the **Detail Page**
 - Find and display the **Vehicle.Manufacturer** on the **Detail Page**
 - Find and display the **Vehicle.Color** on the **Detail Page**
 - Find **Vehicle.InvoicePrice** and display invoice price times 125% as list price on the **Detail Page**
 - Find and display the **Vehicle.VDescription** on the **Detail Page**
 - If '\$UserType'=='Inventory clerks':
 - Find and display the **Vehicle.invoice_price** on the **Detail Page**
- Upon:
 - Click **sell the vehicle** link: Call the **sales order** task.
 - Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Login

Task Decomp



Lock Types: Read only on user table

Number of Locks: No locks needed.

Enabling Conditions: none

Frequency: around 20 logins a day

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

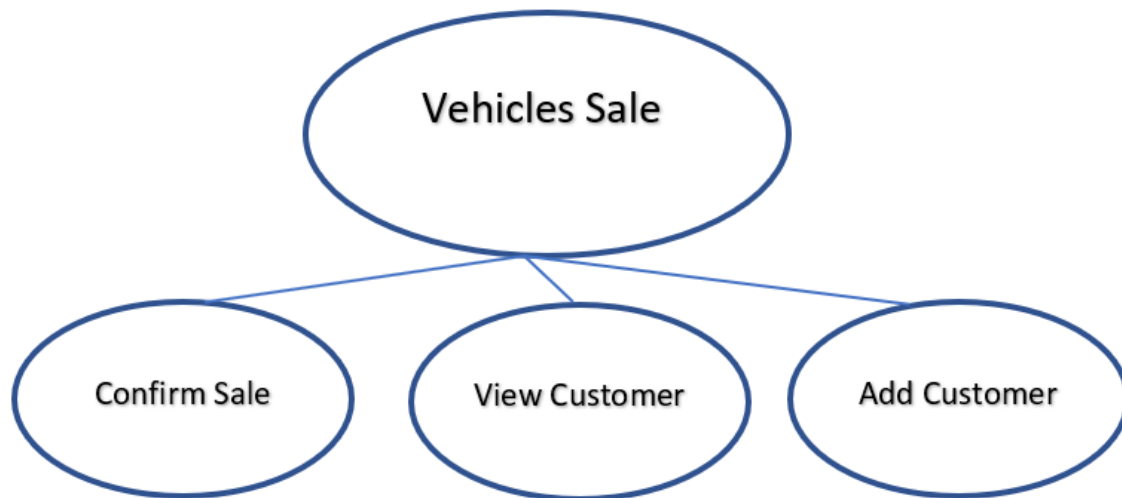
Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- User click **login** button on the Main Menu screen
- Display Login form Screen
- User enters username, password input fields.
- If data validation is successful for both username and password, then:
 - When **Enter** button is clicked:
 - If User record is found but password not match **User**.password:
 - Go back to Login form, with error message.
 - else:
 - Find the User using combination of User.username and User.password
 - Find the User.Type of the User and Store login information as session variable '\$UserType'
 - Call the Display Main Menu task with '\$UserType'.
 - Else username and password input fields are invalid, display Login form, with error message.

Sales order

Task Decomp



Lock Types: Insert on sale table, read/insert on customer table

Number of Locks: 1 or 2

Enabling Conditions: Successful login, salesperson privilege, sale of a vehicle

Frequency: less than 20 times a day

Consistency (ACID): Not critical, vehicle and customer can be viewed while it is being sold

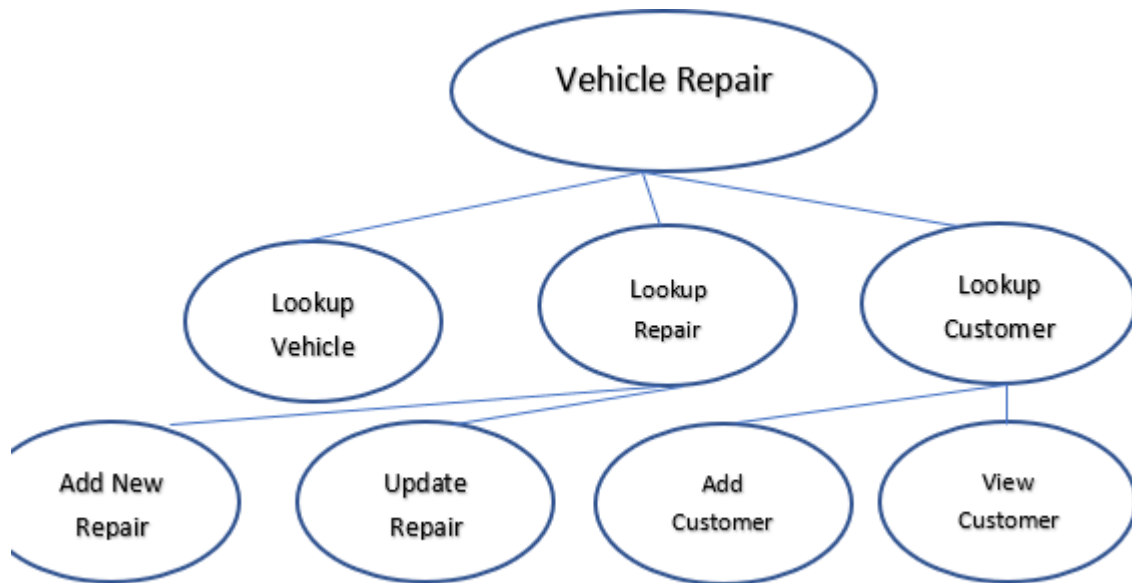
Subtasks: Confirm Sale, Lookup customer and Add customer, mother task is needed to control subtasks

Abstract Code:

- user click **sell the vehicle** on **Detail Page**
- **Sales order Form** is Displayed.
- user will fill the customer profile inputs field showed on **Sales order Form**:
 - if customer is an individual: fill their **first** and **last names**, along with their **driver's license number**
 - if customer is a business: fill the business' **tax identification number** and **business name**, along with the **name of a primary contact** and their **title**
- user will fill the transaction detail field including: **Vehicle's VIN**, **sold price**, **sold date**
- upon:
 - click the **lookup** button on **Sales order Form**: run the **lookup customer** task by query customer with **driver's license number** or **tax identification number**
 - if a customer is not found:
 - Read customer profile input fields, and call **Add customer** task.
 - click the confirm sale button on **Sales order Form**: run the **confirm sale** task by query vehicle's invoice price by reading and using **Vehicle's VIN**
 - if **sold price** is less than or equal to 95% of **Vehicle**.invoice_price:
 - Display error message of rejecting sale.
 - Otherwise, insert new **sale** instance with those values including **sold price**, **sold date**, customer **driver's license number** or **tax identification number**, **Vehicle's VIN**, **SalesPerson's Name**. Display a success message.
 - Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Repair

Task Decomp



Lock Types: Read, Insert, Update on Repair; Read on vehicle; Read on vehicle

Number of Locks: 3

Enabling Conditions: Trigger by successful login, service clerk privilege

Frequency: less than 20 times a day

Consistency (ACID): Customer and vehicle must exist before repair could be carried out

Subtasks: Mother Task is required to control all sub tasks

Abstract Code

- User click **repair form** button on the **Main Menu screen**
- Partial **repair form** is displayed.
- User will fill the **VIN** input field.
- If data validation is successful for **VIN**, then:
 - When **Enter** button is clicked: run **search vin** task.
 - If the **VIN** does not match a **Vehicle.VIN** in the database:
 - Display an error message
 - Otherwise, the rest of the **repair form** will be displayed:
 - Run **View Vehicle Details** task, the results will be displayed on **repair form screen**
 - Check if the **Vehicle** is associated with a **repair** order.
 - If no repairs are open for the **Vehicle**:
 - display **add repair button**
 - run **add repair** task after user click **add repair** button
 - Insert new **repair** order instance with **odometer reading** input field filled by user
 - click the **lookup** button on **Sales order Form**: run the **lookup customer** task by query customer with **driver's license number** or **tax identification number**
 - if a customer is not found:
 - Read customer profile input fields, and call **Add customer** task. add the

- customer to the system for repair order
 - otherwise, add the customer to the system for repair order
- User are allowed to fill the inputs field of **labor charge** and **parts**:
 - input field **quantity**, **vendor**, **part number**, **price** can be filled by user.
 - If Data is valid and user click **add parts** button, insert new parts instance with those values with the service writer's name, display a success message, Otherwise, display an appropriate error message. run **add part task** to add the part to current **repair** order
 - if **add labor charge** button is clicked, add the labor charge to current **repair** order
- Otherwise, upon:
 - click **updating labor charges** button: read input field labor charges and run **update repair** task
 - click **adding parts** button: read input field for parts and run **add part** task
 - click **completing** button: add **repair.completion_date** to current **repair** as current date
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Sales by Color

Task Decomp



Lock Types: Read only on Sales and Vehicle

Number of Locks: No locks needed.

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately once a month.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale.PurchaseDate** and corresponding sold **Vehicle.color** using **Vehicle.VIN**. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale.PurchaseDate**) and group the count by different colors. Put count and color data in a table:
 - Each color is one row.
 - Columns are the count of sales.
 - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
 - If a color does not have any sales, it is shown with a value of "0".
- Display the table in the **Sales by Color report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Sales by Type

Task Decomp



Lock Types: Read only on Sales and Vehicle

Number of Locks: No locks needed.

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately once a month.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale.PurchaseDate** and corresponding sold **Vehicle.type** using **Vehicle.VIN**. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale.PurchaseDate**) and group the count by different types. Put count and type data in a table:
 - Each type is one row.
 - Columns are the count of sales.
 - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
 - If a type does not have any sales, it is shown with a value of "0".
- Display the table in the **Sales by Type report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Sales by Manufacturer

Task Decomp



Lock Types: Read only on Sales and Vehicle

Number of Locks: No locks needed.

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately once a month.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale.PurchaseDate** and corresponding sold **Vehicle.Manufacturer** using **Vehicle.VIN**. **Sale** and **Vehicle** are related by **transfers ownership of** relationship. Count the number of **Sale** based on the different periods (**Sale.PurchaseDate**) and group the count by different Manufacturers. Put count and Manufacturer data in a table:
 - Each Manufacturer is one row.
 - Columns are the count of sales.
 - Columns including sales in the previous 30 days, sales in the previous year, sales overall time.
 - If a type does not have any sales, it will not be put on the table.
- Display the table in the **Sales by Type report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Gross Customer Income

Task Decomp



Lock Types: Lookup sales, customer, vehicle and repair tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

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

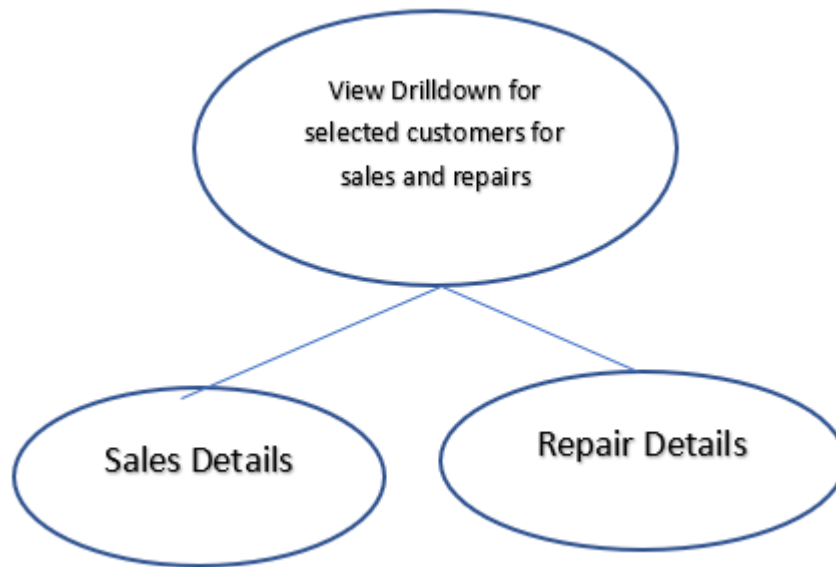
Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about all **Sale** and **Repair**, group **Sale** and **Repair** the by **Customer's** driver's license number or tax identification number. And sum **Sale.sold_price** and **Repair** total cost as gross income for each **Customer** ID. Both **Sale** and **Repair** have relationship with **Customer**. Sort Customer ID by gross income and keep the largest 15 one. Find and Place all following data for each one of 15 Customer in a list:
 - **Customer's** name
 - The date of the first sale or repair start date
 - the date of the most recent sale or repair start date
 - The number of sales
 - The number of repairs
 - the Gross income
- The list of **customers** will be by gross income descending and by last sale/repair start date descending.
- Display the list in the **Gross Customer Income report**
- Users are able to select one **customer's name** in the list. If User clicks one of the **customer's name**: Jump to **View Drill-Down** task with selected **Customer**. and their **sale** and **repair** which can be get from the table that group **sale** and **repair** by customer.
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

View Drill-Down Customers

Task Decomp



Lock Types: Lookup sales, customer, vehicle and repair tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

Consistency (ACID): Not critical, order is not critical. Sales or repair details can be obtained in any order

Subtasks: Sales details and repair details. Mother Task is not needed.

Abstract Code

- User selected a customers from the list on the **Gross Customer Income report**
- Retrieve the Customer ID, Sales, repairs from the **Gross Customer Income** task.
- Find and place all following data in a list, each row is for one **Sale**:
 - **Sale**.SoldDate
 - **Sale**.SoldPrice
 - Find the **Vehicle** associated with **Sale**, and get the **Vehicle**.VIN
 - **Vehicle**.manufacturer
 - **Vehicle**.model
 - **Sale**.salespersonname
- The listing should be sorted by sale date descending and VIN ascending
- Display the list in the **section for vehicle sales** on the **Drill-Down Screen**
- Find and place all following data in a list, each row is for one **Repair**:
 - **repair**.start_date
 - **repair**.complete_date if available
 - Find the **Vehicle** associated with **Repair**, and get the **Vehicle**.VIN
 - **repair**.odometer
 - **repair**.labor cost
 - parts cost
 - total cost
 - the service writer who opened the repair
- This listing should be sorted by start date descending, end date descending, and VIN ascending; however, any incomplete repairs should be listed before completed ones with the same sorting criteria.

- Display the list in the **section for repairs** on the **Drill-Down Screen**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Repairs by Manufacturer/Type/Model

Task Decomp



Lock Types: Lookup on Vehicle and repair tables;

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: : Trigger by successful login and appropriate privileges

Frequency: 1-3 times a month

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

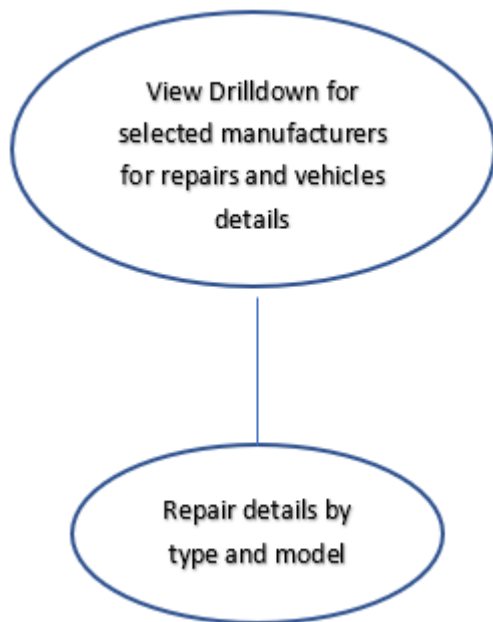
Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Repair**, Count the number of **Repair**, the sum of all parts cost, the sum of all labor cost, and the sum of total repair costs, including any repairs in progress for each **Vehicle.Manufacturer**. The **Vehicle.Manufacturer** is found by using **Vehicle.VIN** associate with each **repair**.
- Populate these data in a list where each row is for one **Vehicle.Manufacturer**, Manufacturers whose vehicles do not have any repairs should be listed on this list, and the list should be sorted by manufacturer name ascending.
- Display the list on **Repairs by Manufacturer/Type/Model screen**
- Users are able to select one manufacturer's name from the list
 - Run **Drill-down** task with the manufacturer's name

View Drill-Down Manufacturer

Task Decomposition



Lock Types: Lookup on Vehicle and repair tables;

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

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

Subtasks: Repair information by type, and further by model. Mother Task is not needed.

Abstract Code

- User select a manufacturer from the list on the **Repairs by Manufacturer/Type/Model screen**
- Retrieve the manufacturer, **Vehicle**, **repairs** from the **Repairs by Manufacturer/Type/Model** task.
- Find and place all following data in a list, each row is for one **Vehicle**.Type:
 - repair count
 - parts costs
 - labor costs
 - total costs
- Find and place all following data in a list, each row is for one **Vehicle**.model:
 - repair count
 - parts costs
 - labor costs
 - total costs
- Lists are sorted by repair count descending(by vehicle type sorted first, and then detail rows sorted).
- Display the list on the **Drill-Down Screen**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.

Below Cost Sales

Task Decomp



Lock Types: Lookup on Vehicle, sales and user tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

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

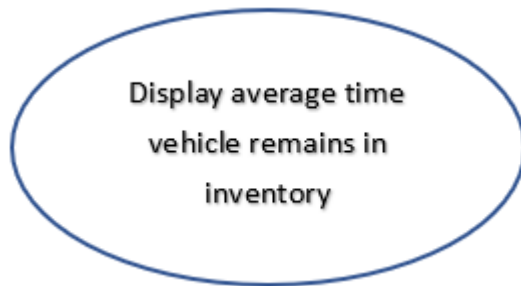
Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale** that **Sale.invoice_price** > **Sale.sold_price**. Find and place all following data in a list, each row is for one **Sale**:
 - **Sale.completedate**
 - **Sale.invoice price**
 - **Sale.sold price**
 - sold price/invoice price ratio as a percentage
 - **customer.name**. Retrieve customer's name by **customer** ID associate with **Sale**.
 - **Sale.salesperson's name**
- For a sale whose ratio is less than or equal to 95%, the background of that row should be highlighted red. Sales should be listed by sales date descending and ratio descending.
- Display the table in the **Below Cost Sales report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Average Time in Inventory

Task Decomp



Lock Types: Lookup on Vehicle and tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Sale**, find the **Sale**.complete date. Find the Vehicle.Date using VehicleVIN associated with Sale. Calculate the difference between **Sale**.complete date and **Vehicle**.Date as the amount of time a vehicle remains in inventory group by Vehicle.Type. Calculate and Put the average amount of time a vehicle remains in inventory in a list, each row is for one Vehicle.Type
- If a **vehicle**.type has no sales history, the report should display "N/A" for that **vehicle**.type.
- Display the table in the **Average Time in Inventory report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Parts Statistics

Task Decomp



Lock Types: Lookup on Parts table

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately 1-3 times a month.

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about each **Part**, find the **Part.price** and **Part.quantity** group by **Part.vendorname**. Calculate the total cost and total quantity of part for each **Part.vendorname**. Then put the vendor's name, the number of parts supplied by that vendor, and the total dollar amount spent on parts in a list.
- Display the list in the **Parts Statistics report**
- Click **Return to Main Menu** button: Call the **Display Main Menu** task.

Monthly Sales

Task Decomp



Lock Types: Lookup on sales and vehicle tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately once a month

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- Called from the **Main Menu screen**
- Query for information about **Sale**.
 - Group the Sale by year and month based on **Sale.Date** then calculate the count and sum for each group.
 - create a list which has:
 - the total number of vehicles sold, the total sales income, the total net income (calculate by using soldprice - invoice price), and the sold price/invoice price ratio as a percentage (such as 125%) for each year and month based on **Sale.Date**.
 - If a year or month does not have sales data, it can be excluded from this report.
 - When the ratio for a month is greater than or equal to 125%, its row should be highlighted with a green background. If the ratio is less than or equal to 110%, it should be highlighted with a yellow background.
 - The results will be ordered by year and month descending, with the most recent year and month as the first result.
 - Display the list on **Monthly Sales screen**
- Users are able to select one manufacturer's name from the list:
 - Run **Drill-down** task with the manufacturer's name

Monthly Sales Drill-Down

Task Decomp



Lock Types: Lookup on sales, user and vehicle tables

Number of Locks: No locks needed. All operations are read only

Enabling Conditions: Trigger by successful login and appropriate privileges

Frequency: Approximately once a month

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

Subtasks: Mother Task is not needed. No decomposition needed

Abstract Code

- User select a month/year from the list on the **Monthly Sales screen**
- Retrieve the **sales** in select month/year group from the **Monthly Sales** task.
- Group the **sale** by **sale.salesperson** then calculates the total vehicles and total sales for each **sale.salesperson**. sort the **Sale.salesperson** by total vehicles descending followed by total sales descending. The first **sale.salesperson** is the top salesperson
- Display the top salesperson on the **Monthly Sales Drill-Down Screen**
- Click ***Return to Main Menu*** button: Call the **Display Main Menu** task.