
A Comprehensive Analysis of a Car Repair Shop's Operations Using SQL

BY

HENRY OKAM

30 July 2024

1.0. Introduction

This report presents a comprehensive data analysis of a car repair shop's operations, utilizing an extended [dataset](#) of sales receipts. Through a series of SQL queries in MySQL, this study examines various aspects of the business, including customer spending habits, vehicle characteristics, job frequency, parts utilization, and financial metrics. The analysis aims to identify patterns, trends, and areas for improvement, providing actionable recommendations to optimize operations and enhance profitability. The report outlines the data ingestion and preparation process, followed by an in-depth analysis of key business questions. It concludes with optimization recommendations to drive success in the competitive automotive repair industry.

2.0. Data Ingestion and Preparation

The data ingestion and preparation process started by creating a database schema named [CarRepairShop](#). All the provided CSV data were then loaded into the MySQL import folder to access from MySQL.

```
-- 1. CREATE A DATABASE
CREATE DATABASE CarRepairShop;
USE CarRepairShop;

-- Find out the set import folder
SHOW VARIABLES LIKE "secure_file_priv";
```

3.0. Table Creation and Data Importing

Each table was created and loaded into the [relational database](#). Each table output will be displayed as a screenshot from each table creation. All SQL queries used in the table creation and data import are well outlined in the appendix section of this report and also available [here](#).

(i) *Create Customer Table and Import Data*

	CustomerID	Name	Address	Phone
▶	1	Jennifer Robinson	126 Nairn Ave, Winnipeg, MB, R3J 3C4	204-771-0784
	2	Michael Smith	250 Broadway, Winnipeg, MB, R3C 0R5	204-555-1234
	3	Sarah Johnson	789 Main St, Winnipeg, MB, R2W 3N2	204-666-5678
	4	Emily Brown	456 Elm St, Winnipeg, MB, R3M 2S5	204-777-9101
	5	David Wilson	123 Oak St, Winnipeg, MB, R2J 3C4	204-888-1112

(ii) Create Vehicle table and import data

	VehicleID	Make	Model	Year	Color	VIN	RegNo	Mileage	OwnerName
▶	1	BMW	X5	2012	Black	CVS123456789123-115Z	BMW 123	16495	Jennifer Robinson
	2	Toyota	Corolla	2015	White	TYS678901234567-876Z	TOY 456	45000	Michael Smith
	3	Honda	Civic	2018	Blue	HCS345678901234-123X	HON 789	30000	Sarah Johnson
	4	Ford	Escape	2020	Red	FES234567890123-456Y	FOR 987	15000	Emily Brown
	5	Chevrolet	Malibu	2016	Silver	CMS456789012345-789Z	CHE 321	60000	David Wilson

(iii) Create Invoice table and import data

	InvoiceID	InvoiceDate	Subtotal	SalesTaxRate	SalesTax	TotalLabour	TotalParts	Total	CustomerID	VehicleID
*	12345	2023-09-10	969.87	13.00	207.33	625.00	969.87	1802.20	1	1
	12346	2023-09-15	325.00	13.00	42.25	325.00	250.00	617.25	2	2
	12347	2023-09-20	200.00	13.00	26.00	200.00	150.00	376.00	3	3
	12348	2023-09-25	300.00	13.00	39.00	300.00	325.00	664.00	4	4
	12349	2023-09-30	440.00	13.00	57.20	440.00	340.00	837.20	5	5

(iv) Create Job table and import data

	JobID	VehicleID	Description	Hours	Rate	Amount	InvoiceID
	1	1	Diagnose front wheel vibration	0.50	125.00	62.50	12345
	2	1	Replace front CV Axel	3.50	125.00	437.50	12345
	3	1	Balance tires	1.00	125.00	125.00	12345
	4	2	Oil change	1.00	75.00	75.00	12346
	5	2	Replace brake pads	2.00	125.00	250.00	12346
	6	3	Replace battery	1.50	100.00	150.00	12347
	7	3	Tire rotation	1.00	50.00	50.00	12347
	8	4	Transmission check	2.00	150.00	300.00	12348
	9	4	Replace air filter	0.50	50.00	25.00	12348
	10	5	Coolant flush	1.50	120.00	180.00	12349
	11	5	Replace spark plugs	2.00	130.00	260.00	12349

(v) Create Parts table and import data

	PartID	JobID	PartNumber	PartName	Quantity	UnitPrice	Amount	InvoiceID
▶	1	2	23435	CV Axel	1	876.87	876.87	12345
	2	2	7777	Shop Materials	1	45.00	45.00	12345
	3	3	W187	Wheel Weights	4	12.00	48.00	12345
	4	5	54321	Brake Pads	1	200.00	200.00	12346
	5	6	67890	Battery	1	120.00	120.00	12347
	6	7	11223	Tire Rotation Kit	1	30.00	30.00	12347
	7	8	33445	Transmission Fluid	1	100.00	100.00	12348
	8	9	99887	Air Filter	1	25.00	25.00	12348
	9	10	77654	Coolant	1	60.00	60.00	12349
	10	11	99876	Spark Plugs	4	20.00	80.00	12349

4.0. Data Cleaning

After importing all the data, the data was then cleaned, properly formatted, and indexed for efficient querying.

```
-- Check that all the data were properly imported and cleaned.
select * from vehicle;
select * from customer;
select * from invoice;
select * from job;
select * from parts;

-- Ensuring the tables are indexed for efficient querying
CREATE INDEX idx_customers_name ON customer (Name);
CREATE INDEX idx_vehicle_make_model_year ON vehicle (Make, Model, Year);
CREATE INDEX idx_job_description ON job (Description);
CREATE INDEX idx_parts_number ON parts (PartNumber);
CREATE INDEX idx_invoice_date ON invoice (InvoiceDate);
```

5.0. Data Analysis

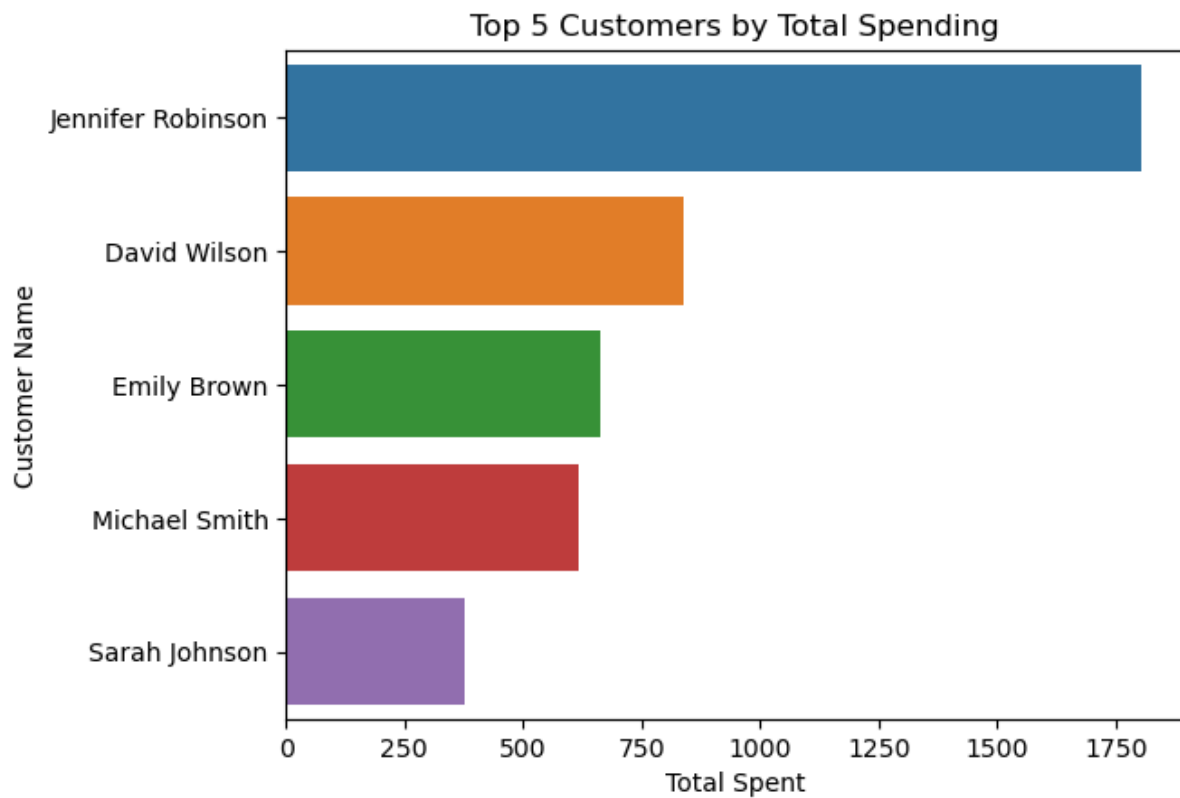
(a) Customer Analysis

(i) Identify the top 5 customers who have spent the most on vehicle repairs and parts.

SQL Output:

Result Grid			Filter Rows:
	Name	TotalSpending	
▶	Jennifer Robinson	1802.20	
	David Wilson	837.20	
	Emily Brown	664.00	
	Michael Smith	617.25	
	Sarah Johnson	376.00	

Bar Chart:



The top 5 customers are *Jennifer Robinson*, *David Wilson*, *Emily Brown*, *Michael Smith* & *Sarah Johnson*.

(ii) Determine the average spending of customers on repairs and parts.

Result Grid		Filter Rows:
	AvgRepairSpend	AvgPartsSpend
	378.000000	406.974000

Average Spend of customers on Repairs = CAD378.000

Average Spend of customers on Parts = CAD406.974

(iii) Analyse the frequency of customer visits and identify any patterns.

Result Grid				
		Filter Rows:		
		Export:		
		Wrap Cell Content:		
Name	VisitFrequency	InvoiceDate	DaysBetweenVisits	
Jennifer Robinson	1	2023-09-10	5	
Michael Smith	1	2023-09-15	5	
Sarah Johnson	1	2023-09-20	5	
Emily Brown	1	2023-09-25	5	
David Wilson	1	2023-09-30	NULL	

All customers each have single visits and have a pattern of visiting every 5 days.

(b) Vehicle Analysis

(i) Calculate the average mileage of vehicles serviced.

AverageMileage
33299.0000

The average mileage of vehicles serviced is 33299.

(ii) Identify the most common vehicle makes and models brought in for service.

Result Grid			
		Filter Rows:	
Make	Model	Frequency	
BMW	X5	1	
Chevrolet	Malibu	1	
Ford	Escape	1	
Honda	Civic	1	
Toyota	Corolla	1	

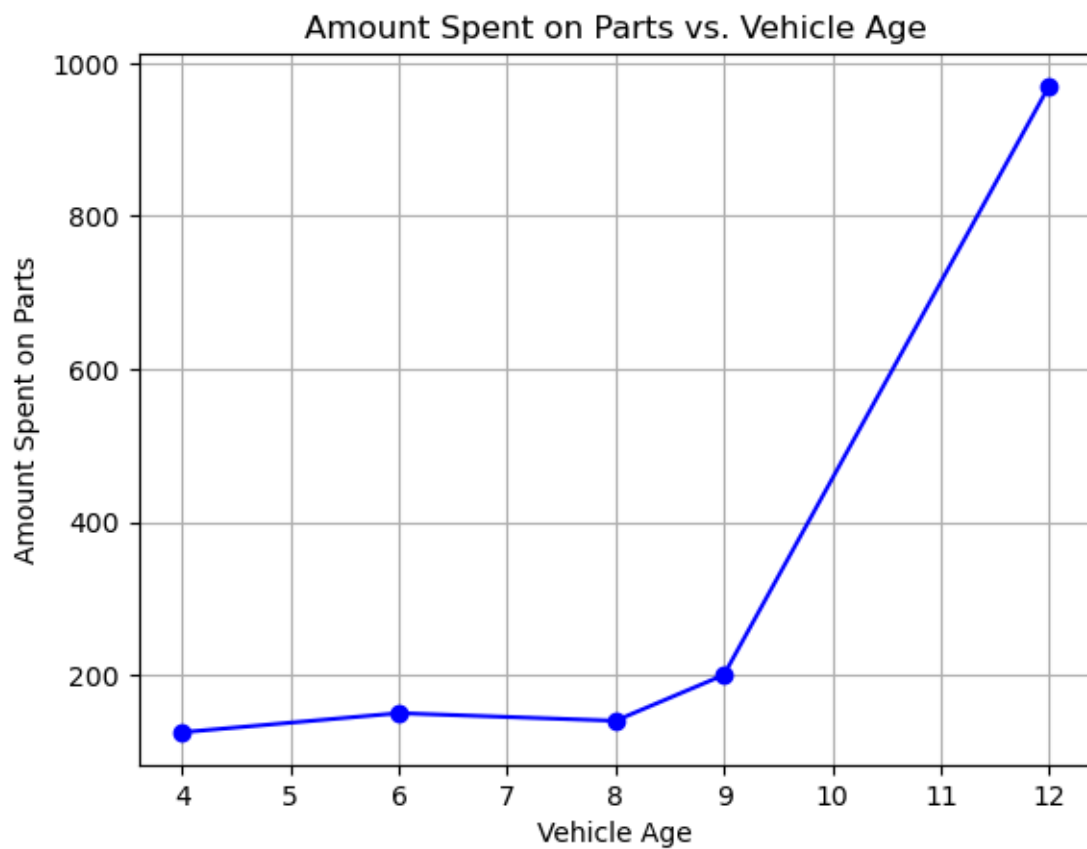
The most common vehicle makes and models are as seen in the output.

(iii) Analyse the distribution of vehicle ages and identify any trends in service requirements based on vehicle age.

SQL output:

Result Grid				
		Filter Rows:		
		Export:		
		Wrap		
Make	VehicleAge	AmountSpentOnParts	QtyParts	
Ford	4	125.00	2	
Honda	6	150.00	2	
Chevrolet	8	140.00	5	
Toyota	9	200.00	1	
BMW	12	969.87	6	

Line Chart:



The Vehicle ages are distributed between 4 – 12 years of age. From the line chart, it can be inferred that the older vehicles cost more on service requirements (parts) compared to the ones with lower ages meaning that an older vehicle will cost more to service over a younger one.

(c) Job Performance Analysis:

(i) Determine the most common types of jobs performed and their frequency.

SQL Output:

Result Grid Filter Rows:	
Description	Frequency
Balance tires	1
Coolant flush	1
Diagnose front wheel vibration	1
Oil change	1
Replace air filter	1
Replace battery	1
Replace brake pads	1
Replace front CV Axel	1
Replace spark plugs	1
Tire rotation	1
Transmission check	1

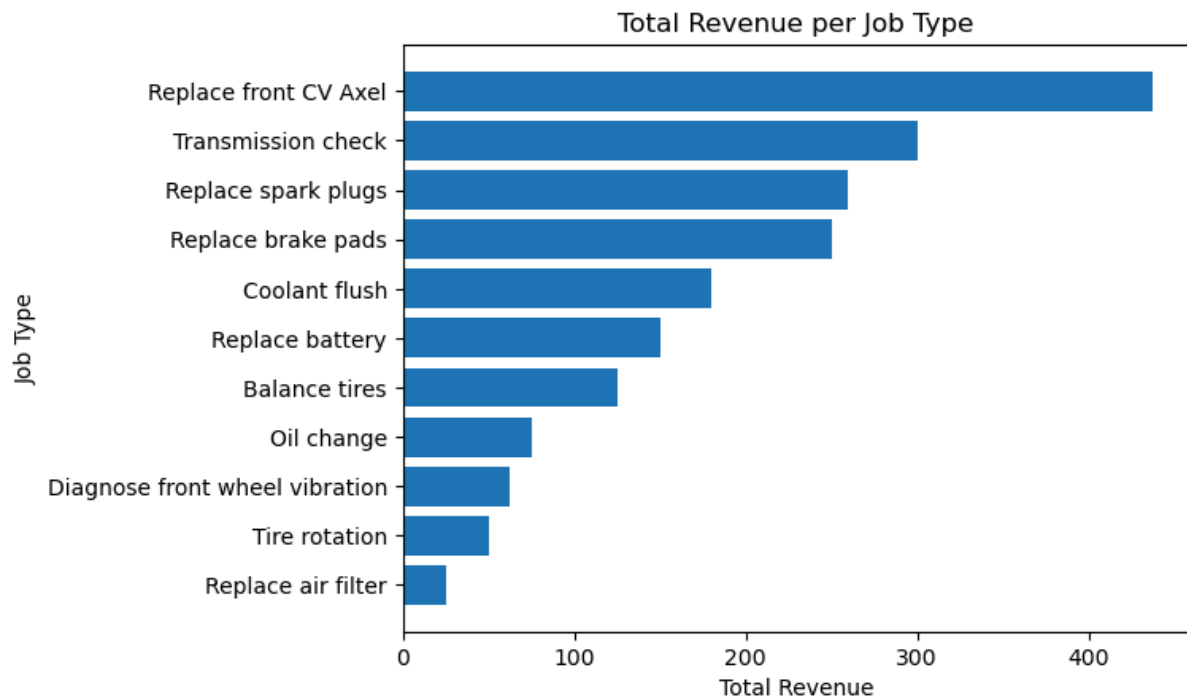
All jobs performed are unique and were performed just once based on the data available.

(ii) Calculate the total revenue generated from each type of job.

SQL output:

Description	TotalRevenue
Replace front CV Axel	437.50
Transmission check	300.00
Replace spark plugs	260.00
Replace brake pads	250.00
Coolant flush	180.00
Replace battery	150.00
Balance tires	125.00
Oil change	75.00
Diagnose front wheel vibration	62.50
Tire rotation	50.00
Replace air filter	25.00

Horizontal Bar Chart:



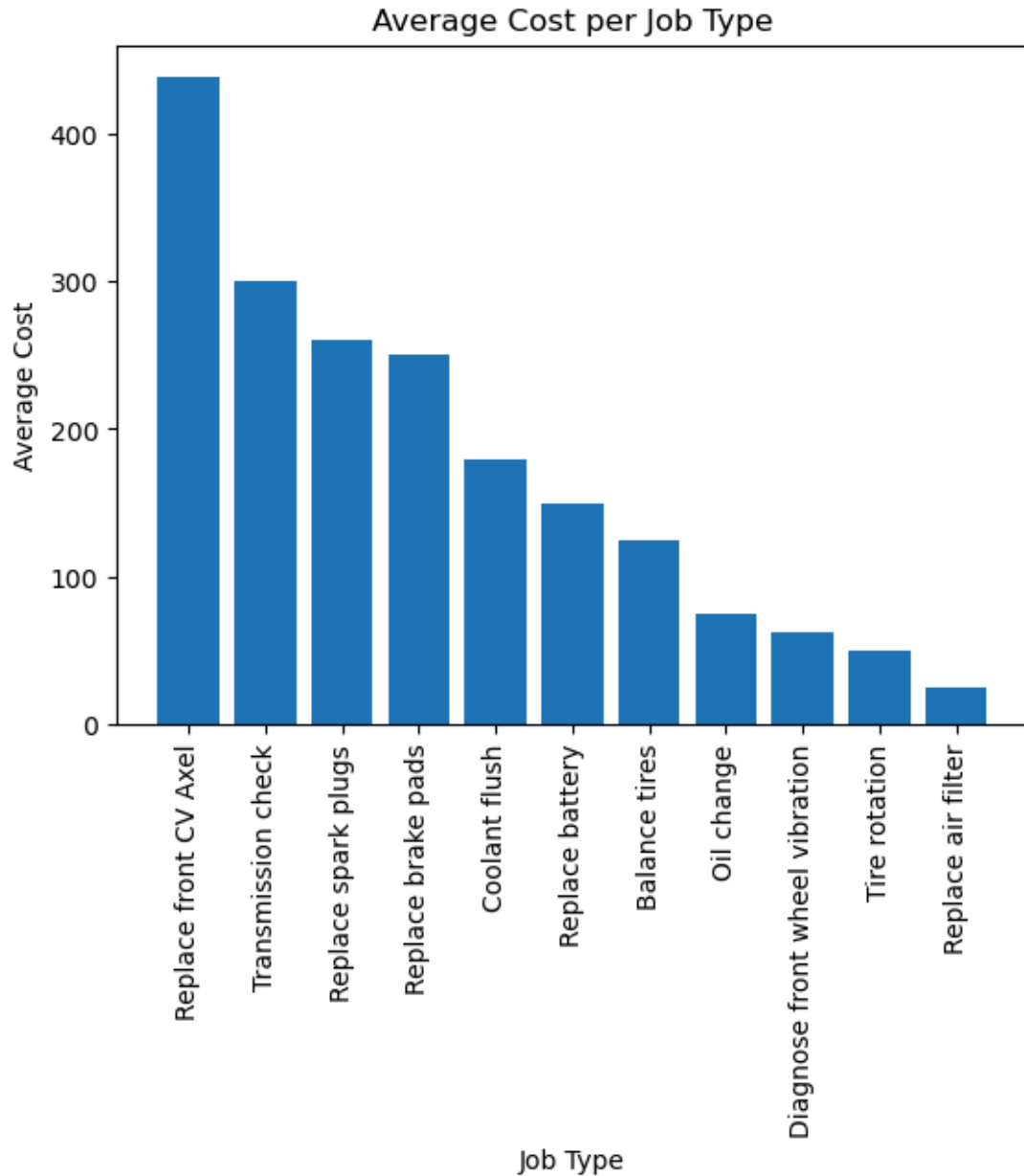
The total revenue per job displayed shows the highest revenue generated from ‘*Replace front CV Axel*’ and the lowest generated from ‘*Replace air filter*’.

(iii) Identify the jobs with the highest and lowest average costs.

SQL Output:

Description	AverageCost
Replace front CV Axel	437.500000
Transmission check	300.000000
Replace spark plugs	260.000000
Replace brake pads	250.000000
Coolant flush	180.000000
Replace battery	150.000000
Balance tires	125.000000
Oil change	75.000000
Diagnose front wheel vibration	62.500000
Tire rotation	50.000000
Replace air filter	25.000000

Bar chart:



The jobs with the highest and lowest average costs are:

- Replace front CV Axel (Highest average Cost of 437.5)
- Replace air filter (Lowest average Cost of 25.0)

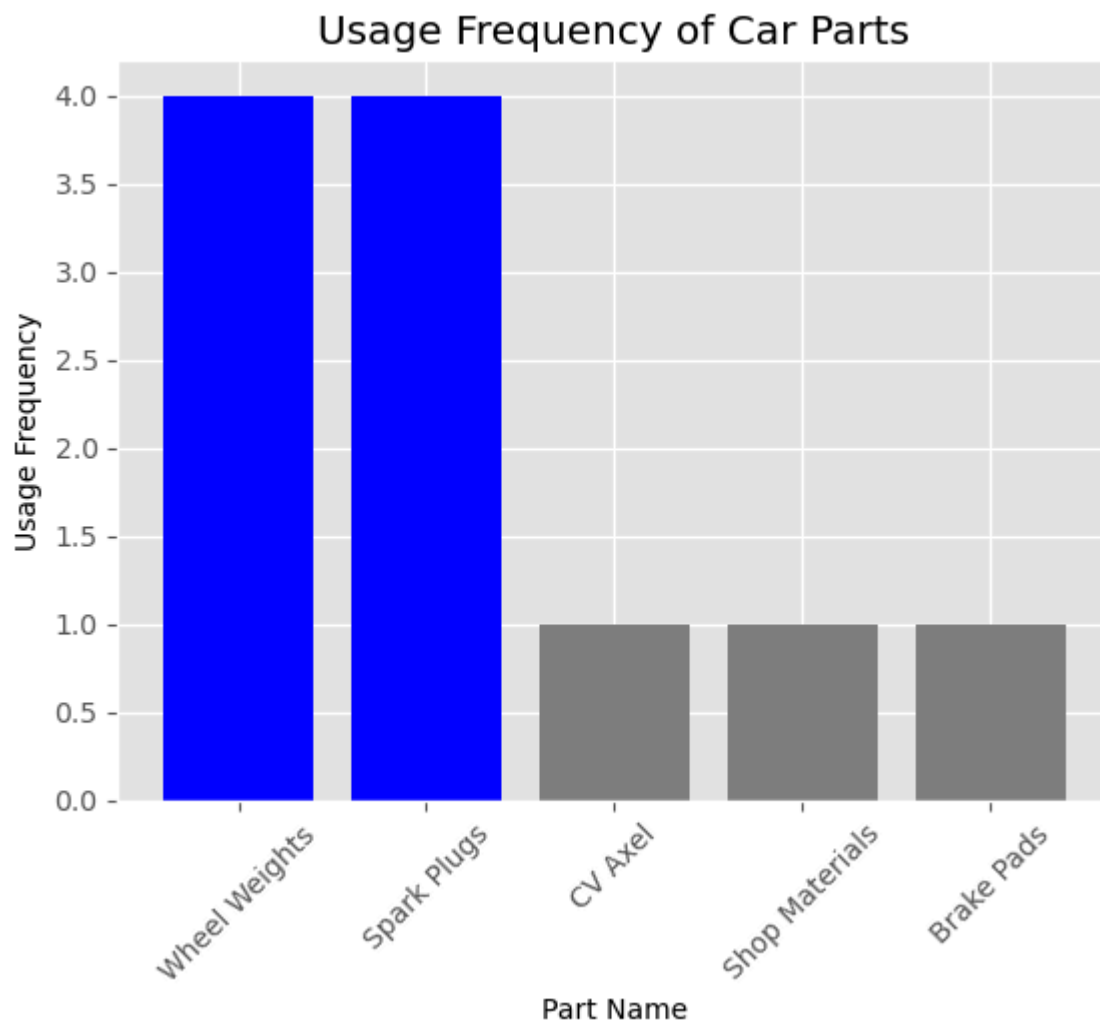
(d) Parts Usage Analysis:

(i) List the top 5 most frequently used parts and their total usage.

SQL Output:

PartName	UsageFrequency
Wheel Weights	4
Spark Plugs	4
CV Axel	1
Shop Materials	1
Brake Pads	1

Bar Chart:



The top 5 most frequently used parts are displayed in the bar chart. “Wheel weights” and “Spark Plugs” are more frequently used than other parts.

(ii) Calculate the average cost of parts used in repairs.

AveragePartCost
158.487000

The average cost of parts used in repairs is CAD158.487

(ii) *Determine the total revenue generated from parts sales.*

	TotalPartsRevenue
▶	1584.87

The total revenue generated from parts sales is CAD1584.87.

(e) Financial Analysis:

(i) *Calculate the total revenue generated from labor and parts for each month.*

	Year	Month	TotalRevenue
▶	2023	9	3924.87

Total revenue from labor and parts for each month is – CAD3924.87 in November 2023.

(ii) *Determine the overall profitability of the repair shop.*

	Profit
▶	3924.87

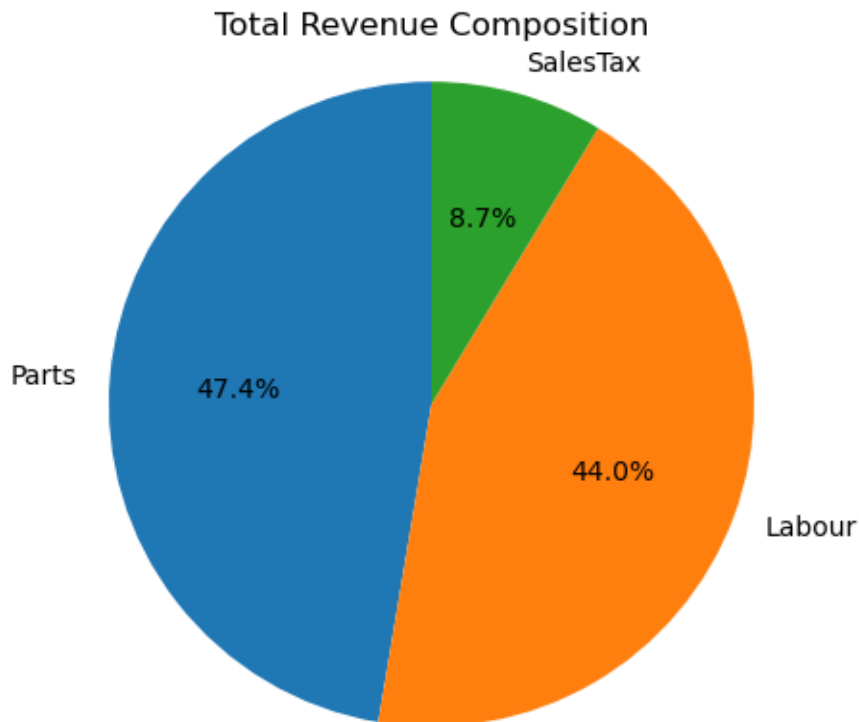
The overall profitability of the repair shop is 3924.87 which is the amount generated from both labor and parts.

(iii) *Analyse the impact of sales tax on the total revenue.*

SQL Output:

	TotalRevenue	TotalParts	TotalLabour	TotalSalesTax	Percent_tax
▶	4296.65	2034.87	1890.00	371.78	0.086528

Pie Chart:



Sales tax accounted for 8.7% of the total revenue generated by the business.

6.0. Optimization Recommendations

Based on the analysis, here are some actionable recommendations:

Identify Underperforming Services:

The analysis of job types and their frequencies can help identify underperforming services. For example, if a particular type of job has a high frequency but contributes less to the total revenue, it may indicate a need for price adjustment or marketing efforts to promote those services. However, from our analysis, no job had a frequency of more than one which shows that none of the services underperformed as there is insufficient data to rate the performance of the job services.

Suggesting Parts Inventory Management:

Parts that are frequently used should be kept in higher stock to ensure availability and reduce wait times for repairs. The analysis of frequently used parts can guide inventory management. For instance, parts like "Wheel weights" and "Spark Plugs" appear to be used often and should be kept in higher stock to meet demand.

Proposing Customer Loyalty Programs:

Customers like “Jennifer Robinson” spent a significant amount on repairs and parts. Implementing a customer loyalty program with rewards or discounts for these top-spending customers could foster loyalty and repeat business.

Recommending Scheduling Adjustments:

The analysis of job frequencies can inform scheduling adjustments. For instance, if certain jobs are performed more frequently, more resources or staff can be allocated for those specific tasks during peak hours to reduce wait times. However, there are no observed jobs that are deemed to be more frequently performed than others at the repair centers based on the data available. More data would be required to ascertain the frequency of jobs performed for the foreseeable future at the repair centers.

7.0. Appendix

7.1. SQL Scripts used to Create Database, tables, and import Data

-- 1. CREATE A DATABASE

```
CREATE DATABASE CarRepairShop;
```

```
USE CarRepairShop;
```

-- Find out the set import folder

```
SHOW VARIABLES LIKE "secure_file_priv";
```

-- 2. Create Table and Import Data

```
CREATE TABLE Customer (
```

```
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
```

```
    Name VARCHAR(100),
```

```
    Address VARCHAR(255),
```

```
    Phone VARCHAR(20)
```

```
);
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/customer.csv'
```

```
INTO TABLE Customer
```

```
FIELDS TERMINATED BY ','
```

ENCLOSED BY ''''

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(Name, Address, Phone);

-- 3. Create Vehicle Table and import data

CREATE TABLE Vehicle (

VehicleID INT auto_increment PRIMARY KEY,

Make VARCHAR(50),

Model VARCHAR(50),

Year INT,

Color VARCHAR(20),

VIN VARCHAR(50),

RegNo VARCHAR(50),

Mileage INT,

OwnerName VARCHAR(100)

);

-- Import data into Vehicle table

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehicle.csv'

INTO TABLE Vehicle

FIELDS TERMINATED BY ','

ENCLOSED BY ''''

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(Make, Model, Year, Color, VIN, RegNo, Mileage, OwnerName);

-- 4. Create the Invoice table and import data

CREATE TABLE Invoice (

InvoiceID INT PRIMARY KEY,

```

InvoiceDate DATE,
Subtotal DECIMAL(10, 2),
SalesTaxRate DECIMAL(5, 2),
SalesTax DECIMAL(10, 2),
TotalLabour DECIMAL(10, 2),
TotalParts DECIMAL(10, 2),
Total DECIMAL(10, 2),
CustomerID INT,
VehicleID INT,
FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)
);

-- Import data into invoice table
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/invoice.csv'
INTO TABLE Invoice
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

-- 5. Create Job table and import data
CREATE TABLE Job (
    JobID INT PRIMARY KEY,
    VehicleID INT,
    Description VARCHAR(255),
    Hours DECIMAL(5, 2),
    Rate DECIMAL(10, 2),
    Amount DECIMAL(10, 2),
    InvoiceID INT,

```



```
FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceID),  
FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)  
);
```

```
-- Import data into Job table
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/job.csv'  
INTO TABLE Job  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

```
-- 6. Create Part Table and import data
```

```
CREATE TABLE Parts (  
    PartID INT PRIMARY KEY,  
    JobID INT,  
    PartNumber VARCHAR(50),  
    PartName VARCHAR(255),  
    Quantity INT,  
    UnitPrice DECIMAL(10, 2),  
    Amount DECIMAL(10, 2),  
    InvoiceID INT,  
    FOREIGN KEY (JobID) REFERENCES Job(JobID),  
    FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceID)  
);
```

```
-- Import data into Part table
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/parts.csv'  
INTO TABLE Parts  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''''
```

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

-- Check that all the data were properly imported and cleaned.

select * from vehicle;

select * from customer;

select * from invoice;

select * from job;

select * from parts;

-- Ensuring the tables are indexed for efficient querying

CREATE INDEX idx_customers_name ON customer (Name);

CREATE INDEX idx_vehicle_make_model_year ON vehicle (Make, Model, Year);

CREATE INDEX idx_job_description ON job (Description);

CREATE INDEX idx_parts_number ON parts (PartNumber);

CREATE INDEX idx_invoice_date ON invoice (InvoiceDate);

7.2. SQL QUERIES FOR DATA ANALYSIS TASKS

-- A. Customer Analysis:

-- a. Identify the top 5 customers who have spent the most on vehicle repairs and parts.

SELECT Customer.Name, SUM(Invoice.Total) AS TotalSpending

FROM Invoice

JOIN Customer

ON Invoice.CustomerID = Customer.CustomerID

GROUP BY Customer.Name

ORDER BY TotalSpending DESC

LIMIT 5;

-- b. Determine the average spending of customers on repairs and parts.

```
SELECT
    AVG(Invoice.TotalLabour) AS AvgRepairSpend,
    AVG(Invoice.TotalParts) AS AvgPartsSpend
FROM Invoice;
```

-- c. Analyze the frequency of customer visits and identify any patterns.

```
SELECT
    Customer.Name,
    COUNT(Invoice.InvoiceID) AS VisitFrequency,
    InvoiceDate,
    DATEDIFF(
        LEAD(InvoiceDate) OVER (ORDER BY InvoiceDate),
        InvoiceDate
    ) AS DaysBetweenVisits
FROM Invoice
JOIN Customer ON Invoice.CustomerID = Customer.CustomerID
GROUP BY Customer.Name, InvoiceDate
ORDER BY VisitFrequency DESC, InvoiceDate;
```

-- B. Vehicle Analysis

-- a. Calculate the average mileage of vehicles serviced.

```
SELECT AVG(Vehicle.Mileage) AS AverageMileage
FROM Vehicle;
```

-- b. Identify the most common vehicle makes and models brought in for service.

```
SELECT Vehicle.Make, Vehicle.Model, COUNT(*) AS Frequency
FROM Vehicle
GROUP BY Vehicle.Make, Vehicle.Model
ORDER BY Frequency DESC
LIMIT 5;
```

-- c. Analyze the distribution of vehicle ages and identify any trends in service requirements based on vehicle age.

```
SELECT Make,  
       (YEAR(CURDATE()) - Vehicle.Year) AS VehicleAge,  
       SUM(Amount) AS AmountSpentOnParts,  
       SUM(Quantity) AS QtyParts  
FROM parts  
JOIN invoice  
ON parts.InvoiceID = invoice.InvoiceID  
JOIN vehicle  
ON invoice.VehicleID = vehicle.VehicleID  
GROUP BY Make, VehicleAge  
ORDER BY VehicleAge;
```

-- C. Job Performance Analysis

-- a. Determine the most common types of jobs performed and their frequency.

```
SELECT Job.Description, COUNT(JobID) AS Frequency,  
       ROUND(AVG(Hours),1) AS AvgHoursSpent  
FROM Job  
GROUP BY Job.Description  
ORDER BY Frequency DESC, AvgHoursSpent DESC;
```

-- b. Calculate the total revenue generated from each type of job.

```
SELECT Job.Description, SUM(Job.Amount) AS TotalRevenue  
FROM Job  
GROUP BY Job.Description  
ORDER BY TotalRevenue DESC;
```

-- c. Identify the jobs with the highest and lowest average costs.

```
SELECT Job.Description, AVG(Job.Amount) AS AverageCost
```

FROM Job
GROUP BY Job.Description
ORDER BY AverageCost DESC;

-- D. Parts Usage Analysis

-- a. List the top 5 most frequently used parts and their total usage.

SELECT Parts.PartName, SUM(Quantity) AS UsageFrequency
FROM Parts
GROUP BY Parts.PartName
ORDER BY UsageFrequency DESC
LIMIT 5;

-- b. Calculate the average cost of parts used in repairs.

SELECT AVG(Parts.Amount) AS AveragePartCost
FROM Parts;

-- c. Determine the total revenue generated from parts sales.

SELECT SUM(Parts.Amount) AS TotalPartsRevenue
FROM Parts;

-- E. Financial Analysis

-- a. Calculate the total revenue generated from labor and parts for each month.

SELECT
YEAR(Invoice.InvoiceDate) AS Year,
MONTH(Invoice.InvoiceDate) AS Month,
SUM(Invoice.TotalLabour + Invoice.TotalParts) AS TotalRevenue
FROM Invoice
GROUP BY Year, Month
ORDER BY Year, Month;

-- b. Determine the overall profitability of the repair shop.

```
SELECT SUM(Invoice.Total) - SUM(Invoice.SalesTax) AS Profit  
FROM Invoice;
```

-- c. Analyze the impact of sales tax on the total revenue.

```
SELECT  
    SUM(Total) AS TotalRevenue,  
    SUM(TotalParts) AS TotalParts,  
    SUM(TotalLabour) AS TotalLabour,  
    SUM(SalesTax) AS TotalSalesTax,  
    SUM(SalesTax) / SUM(Total) AS Percent_tax  
FROM Invoice;
```