Dimensional Modeling and Analysis for Car Repair Shop

Part 1: Sample Invoice Review, Creating Dimensional Model and generating ER- Diagram

1.0 Introduction

To perform a comprehensive analysis of Latino Garage, a car repair shop's operations, a sample invoice was reviewed. The analysis involves extracting meaningful insights from the available data, which includes information about customers, vehicles, jobs performed, parts used, and invoice summaries. A relational database was created. The extracted insights can help Latino Garage optimize its operations, improve customer satisfaction, and increase profitability. To achieve this, we design a dimensional model that provides a structured and efficient way to analyze the data.

Objectives

- Assess the sales performance of the car repair shop, focusing on both services and parts.
- Segment and analyze sales by customer demographics.
- Examine sales data based on vehicle characteristics such as make, model, and year.
- Evaluate the performance and revenue generated by different services offered.
- Track and analyze parts sales and inventory.
- Perform geographic analysis of sales by different shop locations.
- Identify trends and patterns in sales over time.

2.0 Sample Invoice Review

- The invoice was thoroughly reviewed to find key pieces of information relevant for sales analysis. This included:
  - ❖ Customer information

Customer Details					
Customer Name	**Street**	**City**	**State**	**Zip**	**Phone**
Jennifer Robinson	126 Nairn Ave	Winnipeg	MB	R3J 3C4	204-771-0784

  - ❖ Vehicle information

Vehicle Details					
VIN	**Make**	**Model**	**Year**	**Color**	**Mileage**
CVS123456789123-115Z	BMW	X5	2012	Black	16495

❖ Job performed details

Jobs Performed			
Job Description	**Hours**	**Rate**	**Amount**
Diagnose front wheel vibration	0.50	125.00	62.50
Replace front CV Axel	3.50	125.00	437.50
Balance tires	1.00	125.00	125.00

❖ Parts used details

Parts Purchased				
Part Number	**Part Name**	**Quantity**	**Unit Price**	**Amount**
23435	CV Axel	1	876.87	876.87
7777	Shop Materials	1	45.00	45.00
W187	Wheel Weights	4	12.00	48.00

❖ Invoice summary

Invoice Summary								
InvoiceID	**Invoice Date**	**Subtotal**	**Sales Rate**	**Tax**	**Sales Tax**	**Total Labour**	**Total Parts**	**Total**
12345	2023-09-10	969.87	13		207.33	625.00	969.87	1,802.20

3.0 Dimensional Modelling

Dimensional modeling is a data warehousing and business intelligence technique that organizes data into two main components: fact and dimension tables. **Fact tables** represent quantitative data, such as sales or costs, while **Dimensions tables** store descriptive attributes that provide context for the facts. This structure facilitates efficient data analysis and reporting. A MySQL database named DA_AbenaGyasi was created to house the dimensional model.

```
1      -- Create Database
2 •    CREATE DATABASE DA_AbenaGyasi;
3 •    USE DA_AbenaGyasi;
```

Key metrics (facts) such as labor cost, parts charges, total sales and profit were identified. Dimensions such as customer information, vehicle details, service types, part details, location information, and transaction dates were determined to provide context for these metrics.

Facts:
Total sales (subtotal + sales tax)
Labor cost
Parts cost
Profit

Dimensions:
Customer (customer details)
Vehicle (vehicle details)
Invoice (invoice details)
Job (job performed details)
Parts (parts used details)
Location (shop location)
Date (transaction date)

Tables were created for each dimension and fact, with primary and foreign key relationships established to ensure referential integrity and enable detailed analysis.

Benefits of the Dimensional Approach

Adopting a dimensional approach aims to:

- Simplify data querying and analysis. The model allows for easy and intuitive querying, enabling users to quickly extract meaningful insights.
- Enhance data accessibility. Organized data in dimension tables makes it easier for users to access and understand the information.
- Improve query performance. Optimized for read operations, the dimensional model ensures faster query performance, which is essential for timely decision-making.
- Facilitate data-driven decision making by providing a clear and structured view of the data, the model supports informed decision-making processes

4.0 SQL Creation Scripts

```sql
-- Create Factsales Table
CREATE TABLE FactSales (
    FactSalesID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT,
    VehicleID INT,
    JobID INT,
    PartID INT,
    InvoiceID INT,
    DateID INT,
    LocationID INT,
    TotalSales DECIMAL(10,2),
    LaborCost DECIMAL(10,2),
    PartsCost DECIMAL(10,2),
    Profit DECIMAL(10,2),
    FOREIGN KEY (CustomerID) REFERENCES CustomerDim(CustomerID),
    FOREIGN KEY (VehicleID) REFERENCES VehicleDim(VehicleID),
    FOREIGN KEY (JobID) REFERENCES JobDim(JobID),
    FOREIGN KEY (PartID) REFERENCES PartDim(PartID),
    FOREIGN KEY (InvoiceID) REFERENCES InvoiceDim(InvoiceID),
    FOREIGN KEY (DateID) REFERENCES Date(DateID),
    FOREIGN KEY (LocationID) REFERENCES LocationDim(LocationID)
);
```

Fig 1. Screenshots of Fact (Factsales) Table

```sql
-- Create Customer Table
CREATE TABLE CustomerDim (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(40),
    Street VARCHAR(20),
    City VARCHAR(10),
    State VARCHAR(5),
    Zip VARCHAR(10),
    Phone VARCHAR(15)
);

-- Create Vehicle Table
CREATE TABLE VehicleDim (
    VehicleID INT PRIMARY KEY,
    VIN VARCHAR(50),
    Make VARCHAR(50),
    Model VARCHAR(50),
    Year INT,
    Color VARCHAR(20),
    Mileage INT,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES CustomerDim(CustomerID)
);
```

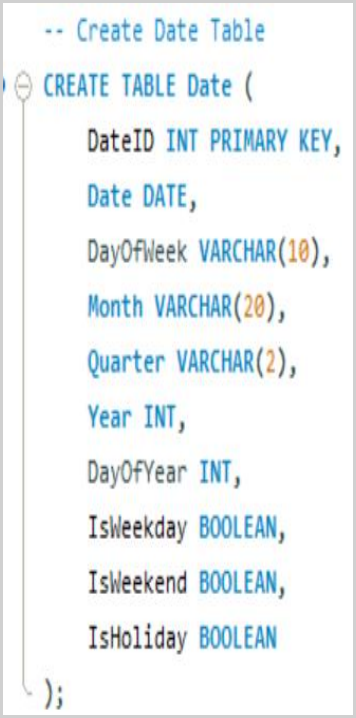
Fig 2. Screenshots of Customer and Vehicle Dimension Tables

```sql
-- Create Invoice Table
CREATE TABLE InvoiceDim (
    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 CustomerDim(CustomerID),
    FOREIGN KEY (VehicleID) REFERENCES VehicleDim(VehicleID)
);

-- Create Job Table
CREATE TABLE JobDim (
    JobID INT PRIMARY KEY,
    VehicleID INT,
    Description VARCHAR(255),
    Hours DECIMAL(5,2),
    Rate DECIMAL(5,2),
    Amount DECIMAL(10,2),
    InvoiceID INT,
    FOREIGN KEY (VehicleID) REFERENCES VehicleDim(VehicleID),
    FOREIGN KEY (InvoiceID) REFERENCES InvoiceDim(InvoiceID)
);
```

Fig 3. Screenshots of Invoice and Job Dimension Tables

```sql
-- Create Part Table
CREATE TABLE PartDim (
    PartID INT PRIMARY KEY,
    JobID INT,
    PartNumber VARCHAR(50),
    PartName VARCHAR(50),
    Quantity INT,
    UnitPrice DECIMAL(10,2),
    Amount DECIMAL(10,2),
    InvoiceID INT,
    FOREIGN KEY (JobID) REFERENCES JobDim(JobID),
    FOREIGN KEY (InvoiceID) REFERENCES InvoiceDim(InvoiceID)
);

-- Create Location Table
CREATE TABLE LocationDim (
    LocationID INT PRIMARY KEY,
    ShopName VARCHAR(50),
    Street VARCHAR(20),
    City VARCHAR(50),
    State VARCHAR(5),
    Zip VARCHAR(10),
    Phone VARCHAR(15)
);
```

Fig 4. Screenshots of Parts and Location Dimension Tables



```
-- Create Date Table
CREATE TABLE Date (
    DateID INT PRIMARY KEY,
    Date DATE,
    DayOfWeek VARCHAR(10),
    Month VARCHAR(20),
    Quarter VARCHAR(2),
    Year INT,
    DayOfYear INT,
    IsWeekday BOOLEAN,
    IsWeekend BOOLEAN,
    IsHoliday BOOLEAN
);
```

```
DELIMITER $$
CREATE PROCEDURE PopulateDateDimension(StartDate DATE, EndDate DATE)
BEGIN
    DECLARE vDate DATE;
    DECLARE vDateID INT DEFAULT 1;
    SET vDate = StartDate;

    WHILE vDate <= EndDate DO
        INSERT INTO Date (DateID, Date, DayOfWeek, Month, Quarter, Year, DayOfYear, IsWeekday, IsWeekend, IsHoliday)
        VALUES (
            vDateID,
            vDate,
            DAYNAME(vDate),
            MONTHNAME(vDate),
            QUARTER(vDate),
            YEAR(vDate),
            DAYOFYEAR(vDate),
            CASE WHEN WEEKDAY(vDate) BETWEEN 0 AND 4 THEN 1 ELSE 0 END,
            CASE WHEN WEEKDAY(vDate) IN (5, 6) THEN 1 ELSE 0 END,
            0 -- Placeholder for holiday logic
        );
        SET vDate = DATE_ADD(vDate, INTERVAL 1 DAY);
        SET vDateID = vDateID + 1;
    END WHILE;
END$$
DELIMITER ;

CALL PopulateDateDimension('2023-09-01', '2024-12-31');
```

Fig 5 and 6. Screenshots of Date Dimension Table.

Database Schema Overview

A visual inspection of the database schema DA_AbenaGyasi confirms the successful creation of the necessary tables for the sales analysis project.

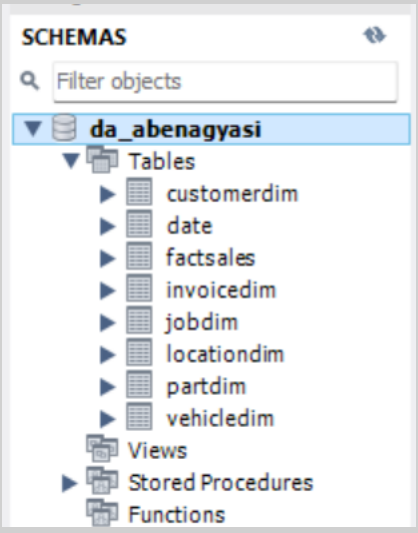


Fig 7. Visual Inspection of Database DA_AbenaGyasi

The schema includes several dimension tables and a central fact table, structured as follows: Dimension Tables: CustomerDim, DateDim, InvoiceDim, JobDim, LocationDim, PartsDim, VehicleDim

Fact Table: FactSales

This setup supports comprehensive analyses, allowing us to explore customer behaviors, vehicle details, job performances, parts utilization, and financial metrics. Each table is linked appropriately through foreign keys, facilitating efficient queries and robust data insights.

This sanity check of the database tables ensures all necessary dimensions and facts are in place for the upcoming detailed sales analysis tasks.

5.0 Entity – Relationship Diagram (ER Diagram)

An Entity-Relationship (ER) diagram is a graphical representation of data and the relationships between them within an information system. It is a visual tool used in database design to model the structure of a database.

Key components of an ER diagram:

Entities: These represent real-world objects or concepts about which information needs to be stored. (e.g., Customer, Vehicle, Invoice)

Attributes: These are the properties or characteristics of an entity. (e.g., CustomerName, VehicleID, InvoiceDate)

Relationships: These define how entities are connected. (e.g., a Customer can have many Vehicles)

An ER diagram was created using Reverse Engineer in MySQL Workbench. The ER diagram showed the relationships between the SalesFact table and the dimension tables (CustomerDim, VehicleDim, InvoiceDim, JobDim, PartsDim, LocationDim, DateDim).

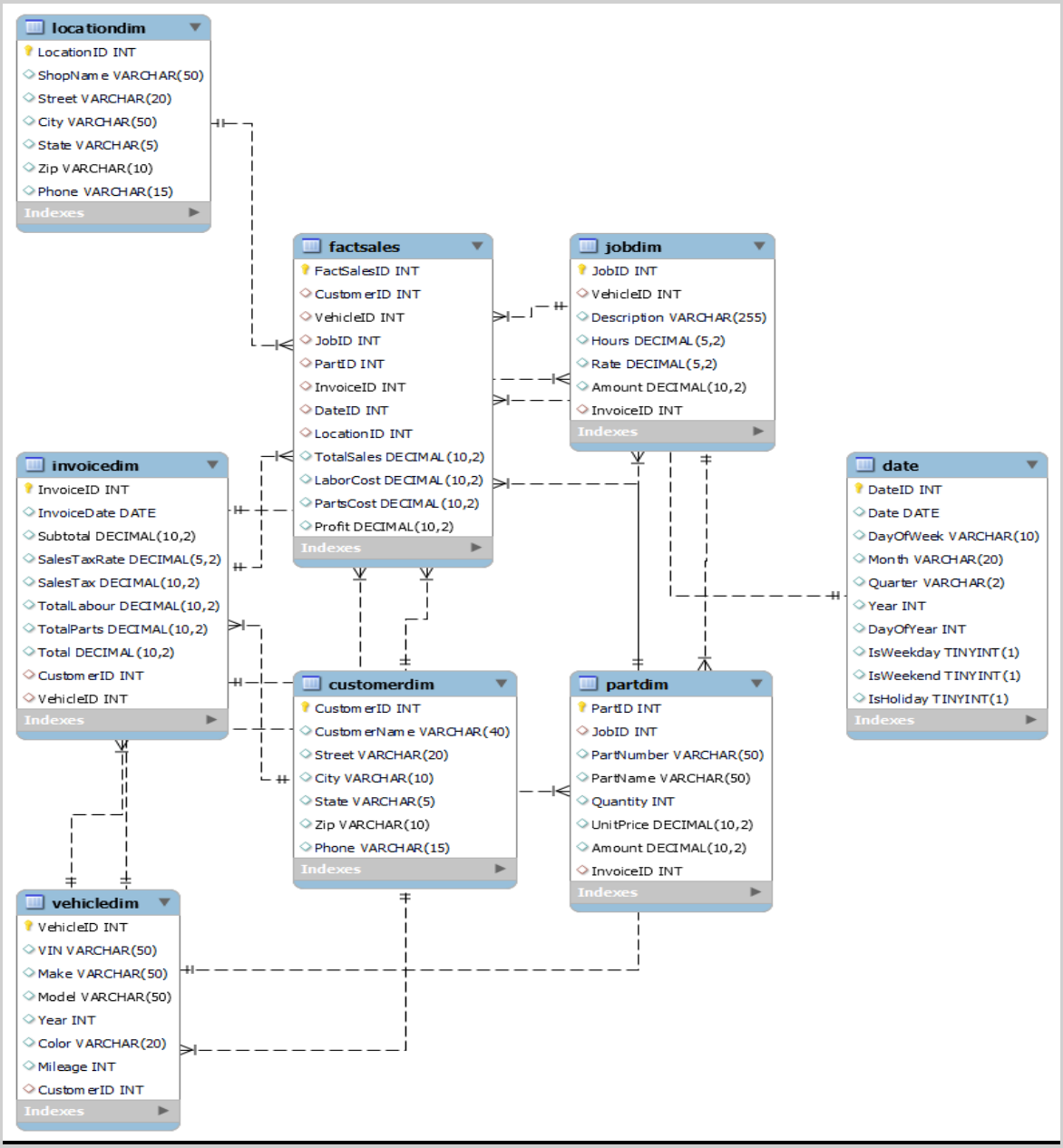


Fig 8. Entity- Relationship Diagram.

Table Descriptions and Logical Explanations for Table Creation

1. Customer Dimension Table (CustomerDim)

- Customer ID: (Primary Key), Unique identifier for each customer.
- Customer Name: Name of the Customer.
- Address: (Street, City, State, Zip code), Residential address of the customer.
- Contact Number: Phone number for contacting the customer.

The *CustomerDim* table stores information about the customers. This allows the business to analyze sales based on customer demographics and individual customer behaviors.

2. Vehicle Dimension Table (VehicleDim)

- Vehicle ID: (Primary Key), Unique identifier for each vehicle.
- Make: Manufacturer of the vehicle.
- Model: Model of the vehicle.
- Year: Year the vehicle was manufactured.
- Color: Color of the vehicle
- VIN: Vehicle Identification Number.
- Mileage: The number of miles the vehicle has travelled at the time of service.
- Customer ID: This establishes a foreign key relationship between the `CustomerID` field in the `VehicleDim` table and the `CustomerID` field in the `CustomerDim` table. It ensures that each vehicle is associated with a valid customer from the `CustomerDim` table.

The `VehicleDim` table stores detailed information about the vehicles serviced. This dimension helps in analyzing sales by different vehicle characteristics such as make, model, year, and mileage.

3. Job Dimension Table (JobDim)

- Job ID: (Primary Key), Unique identifier for each job performed.
- Description: Description of the job.
- Hours Spent: Time spent on the job.
- Rate: Hourly rate for the job.
- Amount: Total amount charged for the job
- Vehicle ID: Identifier of the vehicle related to the job and acts as a foreign key linking to the `VehicleDim` table. It helps in tracking job performance and costs for each vehicle, facilitating vehicle-specific analysis.
- Invoice ID: Identifier of the invoice associated with the job and acts as a foreign key linking to the `InvoiceDim` table. It helps in correlating job performance and charges with the overall invoice, enabling comprehensive analysis of each transaction and its components.

The `JobDim` table captures details about the jobs performed on the vehicles. It enables the analysis of sales performance and revenue by different types of services offered.

4. Invoice Dimension Table (InvoiceDim)

- Invoice ID: The primary key for the table, uniquely identifying each invoice.
- Invoice Date: The date when the invoice was issued. It allows for analysis of sales over time.
- Subtotal: This field stores the subtotal amount of the invoice before taxes and additional charges, providing a measure of the base revenue.
- Sales Tax Rate: This field stores the sales tax rate applied to the invoice, essential for calculating the sales tax amount
- Sales Tax: The total amount of sales tax charged on the invoice, reflecting the tax component of the transaction
- Total Labor: The total labor charges included in the invoice, allowing for analysis of labor-related revenue.
- Total Parts: The total cost of parts included in the invoice, helping in the analysis of parts-related revenue.
- Total: The total amount of the invoice after including the subtotal, sales tax, labor charges, and parts cost, representing the final amount payable by the customer.
- CustomerID: Identifier of the customer to whom the invoice was issued, acting as a foreign key linking to the `CustomerDim` table.
- VehicleID: Identifier of the vehicle related to the invoice, acting as a foreign key linking to the `VehicleDim` table.

The `InvoiceDim` table is designed to store detailed information about each invoice issued by the car repair shop. It provides a basis for analyzing sales performance, including labor and parts revenue and enable time-based analysis by capturing the invoice date.

5. Parts Dimension Table (PartsDim)

- Part ID: (Primary Key), Unique identifier for each part.
- Part Number: Code or number identifying the part.
- Part Name: Name or description of the part.
- Quantity: Number of units used.
- Unit Price: Price per unit of the part.
- Amount: Total amount charged for the part.
- Job ID: It acts as a foreign key linking to the `JobDim` table. It helps in analyzing the cost and usage of parts in relation to specific services provided, enabling job-specific parts analysis.
- Invoice ID: It acts as a foreign key linking to the `InvoiceDim` table.

The `PartsDim` table records information about the parts used in the services. This helps in tracking parts sales and inventory. It helps in connecting parts usage and charges with the entire invoice, facilitating comprehensive analysis of each transaction and its components, including parts used.

6. Location Dimension Table (LocationDim)

- LocationID: (Primary Key), Unique identifier for each location.
- ShopName: Name of the repair shop.
- Address: (Street, City, State, Zip code), Address of the shop.
- ContactNumber: Phone number for the shop.

The `LocationDim` table stores information about the shop locations. This allows analysis of sales by different geographic locations.

7. Date Dimension Table (DateDim)

- Date ID: (Primary Key)
- Date: Actual Date.
- Day Of Week: Stores the name of the day of the week
- Month: Stores the name of the month
- Quarter: Stores the quarter of the year
- Year: Stores the year (e.g., 2023, 2024). It is essential for analyzing long-term trends and year-over-year comparisons
- Day Of Year: Day number within the year (e.g., 1 for January 2023)
- Is Weekday: Field indicated whether the date is a weekday (true) or not (false)
- Is Weekend: Field shows whether the date is a weekend (true) or not (false)
- Is Holiday: Field indicates whether the date is a public holiday (true) or not (false). It is useful for analyzing the impact of holidays on sales performance.

The `DateDim` table is a crucial component of the sales analysis dimensional model, providing detailed temporal context for analyzing sales data. Its attributes allow for comprehensive and granular analysis of sales trends, patterns, and performance across different time dimensions. This table supports informed decision-making by offering valuable insights into how time-related factors impact sales.

8. Fact Sales Table (FactSales)

- FactSales ID: (Primary Key), This is the primary key for the table, uniquely identifying each sales record.
- Customer ID: Stores the identifier of the customer involved in the sales transaction. It acts as a foreign key linking to the `CustomerDim` table
- Vehicle ID: This field stores the identifier of the vehicle involved in the sales transaction. It acts as a foreign key linking to the `VehicleDim` table
- Job ID: This field stores the identifier of the job associated with the sales transaction. It acts as a foreign key linking to the `JobDim` table.
- Part ID: This field stores the identifier of the part involved in the sales transaction. It acts as a foreign key linking to the `PartsDim` table.
- Invoice ID: This field stores the identifier of the invoice associated with the sales transaction. It acts as a foreign key linking to the `InvoiceDim` table.
- Date ID: This field stores the identifier of the date related to the sales transaction. It acts as a foreign key linking to the `DateDim` table.
- Location ID: This field stores the identifier of the location where the sales transaction occurred. It acts as a foreign key linking to the `LocationDim` table.
- Total Sales: This field captures the total sales amount for the transaction, including all charges and taxes.
- Labor Cost: This field stores the total labor cost associated with the transaction.
- Parts Cost: This field stores the total cost of parts used in the transaction.
- Profit: This field stores the total cost of parts used in the transaction.
- FOREIGN KEY (CustomerID) REFERENCES CustomerDim (CustomerID): Ensures that each sales record is associated with a valid customer from the CustomerDim table.
- FOREIGN KEY (VehicleID) REFERENCES VehicleDim (VehicleID): Ensures that each sales record is associated with a valid vehicle from the VehicleDim table.
- FOREIGN KEY (JobID) REFERENCES JobDim (JobID): Ensures that each sales record is associated with a valid job from the JobDim table.
- FOREIGN KEY (PartID) REFERENCES PartsDim (PartID): Ensures that each sales record is associated with valid parts from the PartsDim table.
- FOREIGN KEY (InvoiceID) REFERENCES InvoiceDim (InvoiceID): Ensures that each sales record is associated with a valid invoice from the InvoiceDim table.
- FOREIGN KEY (DateID) REFERENCES DateDim (DateID): Ensures that each sales record is associated with a valid date from the DateDim table.
- FOREIGN KEY (LocationID) REFERENCES LocationDim (LocationID): Ensures that each sales record is associated with a valid location from the LocationDim table.

The `FactSales` table captures the essential metrics related to sales transactions, linking them to relevant dimensions such as customer, vehicle, job, parts, invoice, date, and location. This structure enables comprehensive and detailed analysis of sales performance, costs, and profitability, providing valuable insights to optimize operations and drive business decisions.

6.0 Conclusion

This report has established a foundational framework for analyzing the operations of a car repair shop through the implementation of a dimensional data model. By structuring data into dimensions and facts, we have created a robust platform for understanding complex business processes.

The dimensional model, encompassing customer, vehicle, job, part, invoice, date, and location information, provides a comprehensive view of the business. This structure eases efficient data analysis and supports informed decision-making.

To fully realize the potential of this dimensional model, further steps, including subsequent analysis of customer behavior, vehicle performance, job profitability, parts usage, and financial metrics will yield valuable insights into the shop's operations. These insights can be used to optimize processes, enhance customer satisfaction, and drive overall business growth.

Also, data population, in-depth analysis, and visualization, are necessary. By undertaking these actions, the car repair shop can gain a competitive advantage through data-driven decision making.