DESIGNING A DIMENSIONAL MODEL FOR SALES ANALYSIS AND SQL QUERIES

PROJECT OVERVIEW

The goal of this project is to extract meaningful insights and performing comprehensive analysis of a car repair shop's shop operation based on a sample invoice using SQL and generate reports that can help the business optimize its operations, improve customer satisfaction, and increase profitability. This involve designing a dimensional model to analyze data based on what data you think is relevant for a sales analysis and the dimensional model should be a logical business view of data.

PART A: DIMENSIONAL MODEL REPORT

INTRODUCTION

This session of this project focuses on creating/designing a dimensional data model for analyzing the sales performance of car repair centers across western Canada using information from a sample invoice. The goal is to create a database schema that allows for efficient querying and reporting on various aspects of sales, including customer behavior, vehicle types, job performance, parts sales, and location effectiveness. The model should enable flexible analysis of sales of both services and parts by customer, vehicle brand/model/year, and shop location.

OBJECTIVES

- **Review the sample invoice**: Identify key pieces of information relevant for sales analysis.
- **Types of analysis**: Determine the type of analysis needed by the business such as sales by customer, vehicle brand/model/year, services, parts, and shop locations.
- **Identify key metrics (facts)**: Identify the facts to analyze, including service charges, parts charges, total sales, and sales tax.
- **Identify attributes (dimensions)**: Identify the dimensions to slice and dice metrics, including customer information, vehicle details, service types, part details, location information, and date/time of the transaction.
- **Create a fact table:** Create fact table to store quantitative data related to sales and include foreign keys to link to the dimension tables.
- **Create dimension tables**: Create a dimension table to store qualitative data for customers, vehicles, services, parts, locations, and dates, ensuring they support the fact table and facilitate detailed analysis.

- **Create tables for each fact and dimension** in any SQL Workbench, define primary keys, establish foreign key relationships, ensure all necessary relationships between tables, ensure referential integrity, and develop an ER diagram (Entity relationship diagram).

IMPORTANT DEFINITIONS

- **SQL (Structured Query Language):** A standard programming language used for managing and manipulating databases.
- **Database:** A structured collection of data stored electronically in a computer system.
- **Qualitative Data:** Descriptive data that can be observed but not measured, such as customer names and vehicle models.
- **Quantitative Data:** Numerical data that can be measured, such as total sales and labor hours.

DATASET OVERVIEW

The dataset for this project is derived from a sample invoice, which provides comprehensive information relevant to analyzing the sales performance of a car repair shop. This data includes details about customers, vehicles, jobs performed, parts used, and overall invoice summaries. The sample invoice serves as the primary source of data for creating a dimensional model that supports flexible analysis of sales performance across various dimensions.

Link to the sample invoice:
https://drive.google.com/file/d/1enihc1uFFW9dHPYFiBC27wpk92QodBC-/view?usp=sharing

Data Source:
The sample invoice provides the following key information:

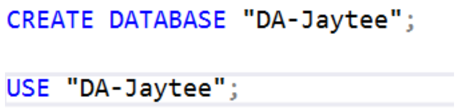
- **Customer Information:** Customer details such as name, address, and contact information.
- **Vehicle Information:** Details about the vehicle, including make, model, year, color, and mileage.
- **Job Information:** Information about the jobs performed, including job description, estimated hours, and labor rate.
- **Parts Information:** Details of parts used in the repair, including part number, part name, and unit price.
- **Invoice Information:** Summary of the transaction, including total sales, labor charges, parts charges, and sales tax.
- **Date Information:** The date of the transaction, including year, month, and day.
- **Location Information:** Information about the repair shop location, including shop name and address.

This data is structured into fact and dimension tables to create a dimensional model that supports detailed sales analysis. The tables are designed to facilitate the analysis of sales by customer, vehicle brand/model/year, service types, parts used, and shop locations across western Canada.

The dimensional model enables the business to gain insights into customer spending patterns, vehicle service trends, job performance, parts usage, and overall sales performance. This information is crucial for optimizing operations, improving customer satisfaction, and increasing profitability.

METHODOLOGY

- Tools Used:
    - Microsoft SQL Server
    - Lucidchart for ER diagram creation
- Model Naming: The dimensional data model is named `DA-Jaytee`



```
CREATE DATABASE "DA-Jaytee";

USE "DA-Jaytee";
```

DIMENSIONAL MODELING

The dimensional model is designed to facilitate the analysis of sales data. Key metrics (facts) and attributes (dimensions) have been identified to allow detailed and flexible analysis.

IMPORTANT DEFINITIONS

- **Facts:** Quantitative data used for analysis, such as service charges, parts charges, and total sales.
- **Dimensions:** Attributes that provide context to facts, such as customer information, vehicle details, and dates.
- **Dimensional Model:** It is a data structure technique optimized for data warehousing and online analytical processing (OLAP) applications. It organizes data into facts and dimensions, which makes it easier to retrieve and analyze data.
- **Data Warehouse:** It is a centralized repository that stores large volumes of data collected from different sources. It is designed for query and analysis rather than transaction processing and typically contains historical data.

STEPS TO DESIGNING THE DIMENSIONAL MODEL

The sample invoice provides key pieces of information relevant for sales analysis, including customer information, vehicle details, services performed, parts used, location of the service center, and transaction dates.

USE CASES FOR ANALYSIS:

1. Sales Performance by Customer:
    - Analyze revenue contributions from individual customers.
    - Identify high-value customers and tailor marketing strategies.
2. Sales Analysis by Vehicle:
    - Determine which vehicle makes and models require the most services.
    - Assess customer preferences and trends in vehicle servicing.
3. Sales by Job/Service Performance:
    - Evaluate profitability across different job types.

    - Identify high-demand jobs and optimize labor allocation.
4. Part Sales Trends:
    - Track which parts are sold most frequently.
    - Manage inventory effectively based on sales trends.
5. Location Performance:
    - Compare sales performance across different repair shop locations.
    - Identify underperforming locations and implement improvement strategies.
6. Time-Based Trends:
    - Analyze sales trends over various time frames (daily, monthly, yearly).
    - Identify seasonal patterns and adjust marketing efforts accordingly.

IDENTIFY FACTS AND DIMENSIONS

FACTS:

Key metrics (facts) to analyze:

- Total Sales Amount
- Total Parts Cost
- Total Labor Cost
- Quantity of Parts Sold
- Hours Worked
- Sales Tax

DIMENSIONS:

Attributes (dimensions) to slice and dice metrics:

- Customer information (name, address, contact)
- Vehicle details (make, model, year, color, mileage)
- Service types (job description, estimated hours, labor rate)
- Part details (part number, part name, unit price)
- Location information (shop name, address)
- Date/time of transaction (date, year, month, day)

DESIGN FOR THE FACTS AND DIMENSIONS TABLE

STEPS IN SQL SERVER

1. **Create Tables:** Define tables for each fact and dimension.

2. **Define Primary Keys:** Specify primary keys for each table.

3. **Establish Foreign Key Relationships:** Link the fact table to dimension tables using foreign keys.

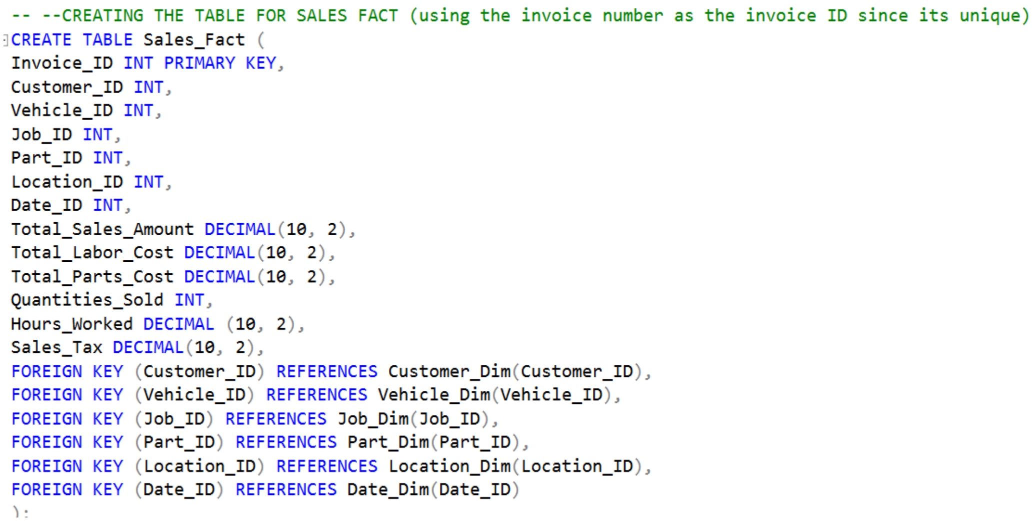
4. **Ensure Referential Integrity:** Define foreign key constraints to enforce relationships.

DESIGN FOR THE FACTS TABLE

TABLE NAME: Sales_Fact (Contains quantitative data about sales, including total amounts, costs, quantities, and time spent.)

- **Invoice_ID:** Primary Key (Used the numbers in the Invoice number; Unique identifier of the sales)
- **Customer_ID:** Foreign Key to Customer_Dim (integer, Dim stands for dimension)
- **Vehicle_ID:** Foreign Key to Vehicle_Dim (integer, Dim stands for dimension)
- **Job_ID:** Foreign Key to Job_Dim (integer, Dim stands for dimension)
- **Part_ID:** Foreign Key to Part_Dim (integer, Dim stands for dimension)
- **Location_ID:** Foreign Key to Location_Dim (integer, Dim stands for dimension)
- **Date_ID:** Foreign Key to Date_Dim (integer, Dim stands for dimension)
- **Total_Sales_Amount:** Total amount of the invoice (decimal)
- **Total_Parts_Cost:** Total cost for parts sold (decimal)
- **Total_Labor_Cost:** Total cost for labor that is job performed (decimal)
- **Quantities_Sold:** Total quantity of parts sold (integer)
- **Hours_Worked:** Total hours worked on the job (decimal)
- **Sales_Tax:** Tax added (decimal)

THE BELOW IMAGE IS THE QUERY FOR CREATING THE FACT (SALES FACT) TABLE

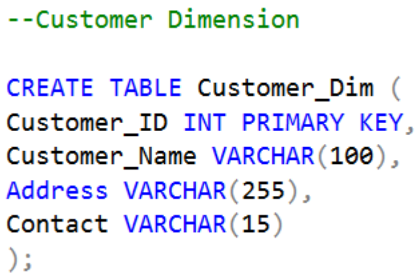


```sql
-- --CREATING THE TABLE FOR SALES FACT (using the invoice number as the invoice ID since its unique)
CREATE TABLE Sales_Fact (
Invoice_ID INT PRIMARY KEY,
Customer_ID INT,
Vehicle_ID INT,
Job_ID INT,
Part_ID INT,
Location_ID INT,
Date_ID INT,
Total_Sales_Amount DECIMAL(10, 2),
Total_Labor_Cost DECIMAL(10, 2),
Total_Parts_Cost DECIMAL(10, 2),
Quantities_Sold INT,
Hours_Worked DECIMAL (10, 2),
Sales_Tax DECIMAL(10, 2),
FOREIGN KEY (Customer_ID) REFERENCES Customer_Dim(Customer_ID),
FOREIGN KEY (Vehicle_ID) REFERENCES Vehicle_Dim(Vehicle_ID),
FOREIGN KEY (Job_ID) REFERENCES Job_Dim(Job_ID),
FOREIGN KEY (Part_ID) REFERENCES Part_Dim(Part_ID),
FOREIGN KEY (Location_ID) REFERENCES Location_Dim(Location_ID),
FOREIGN KEY (Date_ID) REFERENCES Date_Dim(Date_ID)
);
```

DESIGN FOR THE DIMENSION TABLES

1. **Customer Dimension**
   Table: Customer_Dim (Provides details about customers.)
   - **Customer_ID:** Primary Key (int) Unique identifier for the customer
   - **Customer_Name:** Customer's name (varchar (100))
   - **Address:** Customer's Address (varchar (255))
   - **Contact:** Customer's phone number (varchar (15))

THE BELOW IMAGE IS THE QUERY FOR CREATING THE CUSTOMER DIMENSION TABLE

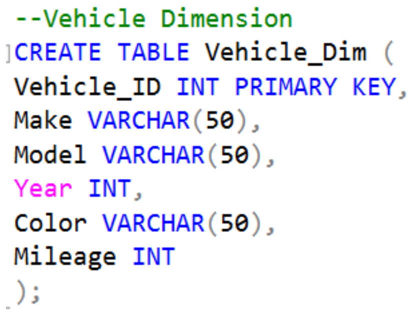


```
--Customer Dimension

CREATE TABLE Customer_Dim (
Customer_ID INT PRIMARY KEY,
Customer_Name VARCHAR(100),
Address VARCHAR(255),
Contact VARCHAR(15)
);
```

2. **Vehicle Dimension**
   Table: Vehicle_Dim (Contains information about vehicles)
   - **Vehicle_ID**: Primary Key (int) Unique identifier for the vehicle
   - **Make**: Vehicle brand or make (varchar (50))
   - **Model**: Vehicle Model (varchar (50))
   - **Year**: Year of manufacture (int)
   - **Color**: Vehicle color (varchar (50))
   - **Mileage**: Vehicle mileage (int)

THE BELOW IMAGE IS THE QUERY FOR CREATING THE VEHICLE DIMENSION TABLE



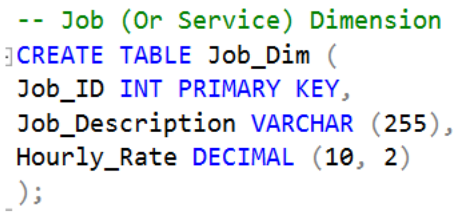
```
--Vehicle Dimension
CREATE TABLE Vehicle_Dim (
Vehicle_ID INT PRIMARY KEY,
Make VARCHAR(50),
Model VARCHAR(50),
Year INT,
Color VARCHAR(50),
Mileage INT
);
```

3. **Job Dimension**

   Table: Job_Dim (Describes the jobs performed.)

   - **Job_ID**: Primary Key (int) Unique identifier for the service
   - **Job_Description**: Description of the job/service performed (varchar (255))
   - **Hourly_Rate**: Cost per hour for the service (decimal)

THE BELOW IMAGE IS THE QUERY FOR CREATING THE JOB/SERVICE DIMENSION TABLE



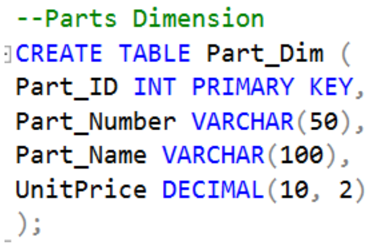
```
-- Job (Or Service) Dimension
CREATE TABLE Job_Dim (
Job_ID INT PRIMARY KEY,
Job_Description VARCHAR (255),
Hourly_Rate DECIMAL (10, 2)
);
```

4. **Part Dimension**

   Table: Part_Dim (Lists vehicle parts sold.)

   - **Part_ID**: Primary Key (int) Unique identifier for the part
   - **Part_Number**: Number of the part (varchar (50))
   - **Part_Name**: Name of the part (varchar (100))
   - **UnitPrice**: Cost per unit of the part (decimal)

THE BELOW IMAGE IS THE QUERY FOR CREATING THE PARTS DIMENSION TABLE



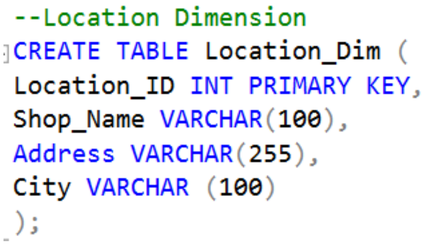
```
--Parts Dimension
CREATE TABLE Part_Dim (
Part_ID INT PRIMARY KEY,
Part_Number VARCHAR(50),
Part_Name VARCHAR(100),
UnitPrice DECIMAL(10, 2)
);
```

5. **Location Dimension**

   Table: Location_Dim (Details the locations of the repair shops)

   - **Location_ID**: Primary Key (int) Unique identifier for the location
   - **Shop_Name**: Name of the repair shop (varchar (100))
   - **Address**: Address of the repair shop (varchar (255))
   - **City**: City where the shop is located (varchar (100))

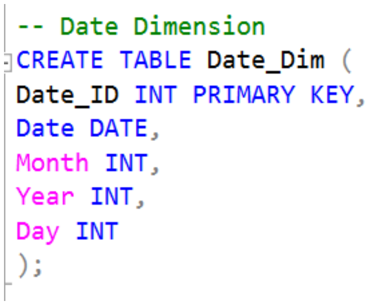
THE BELOW IMAGE IS THE QUERY FOR CREATING THE LOCATION DIMENSION TABLE



```
--Location Dimension
CREATE TABLE Location_Dim (
Location_ID INT PRIMARY KEY,
Shop_Name VARCHAR(100),
Address VARCHAR(255),
City VARCHAR (100)
);
```

6. **Date Dimension**
   Table: Date_Dim (Contains date-related information for time-based analysis; repair date)
   - **Date_ID**: Primary Key (Unique identifier for the date)
   - **Date**: The actual date (date)
   - **Year**: Year of the transaction (int)
   - **Month**: Month of the transaction (int)
   - **Day**: Day of the transaction (int)

THE BELOW IMAGE IS THE QUERY FOR CREATING THE DATE DIMENSION TABLE



```
-- Date Dimension
CREATE TABLE Date_Dim (
Date_ID INT PRIMARY KEY,
Date DATE,
Month INT,
Year INT,
Day INT
);
```

LINK TO THE SQL SCRIPT CONTAINING ALL QUERIES USED TO CREATE/DESIGN THE DIMENSION AND FACTS TABLE USIG MICROSOFT SQL SERVER:

https://drive.google.com/file/d/1BTLlrHnP-tWMeHtfBsS_fN0pMq_-TGDK/view?usp=sharing

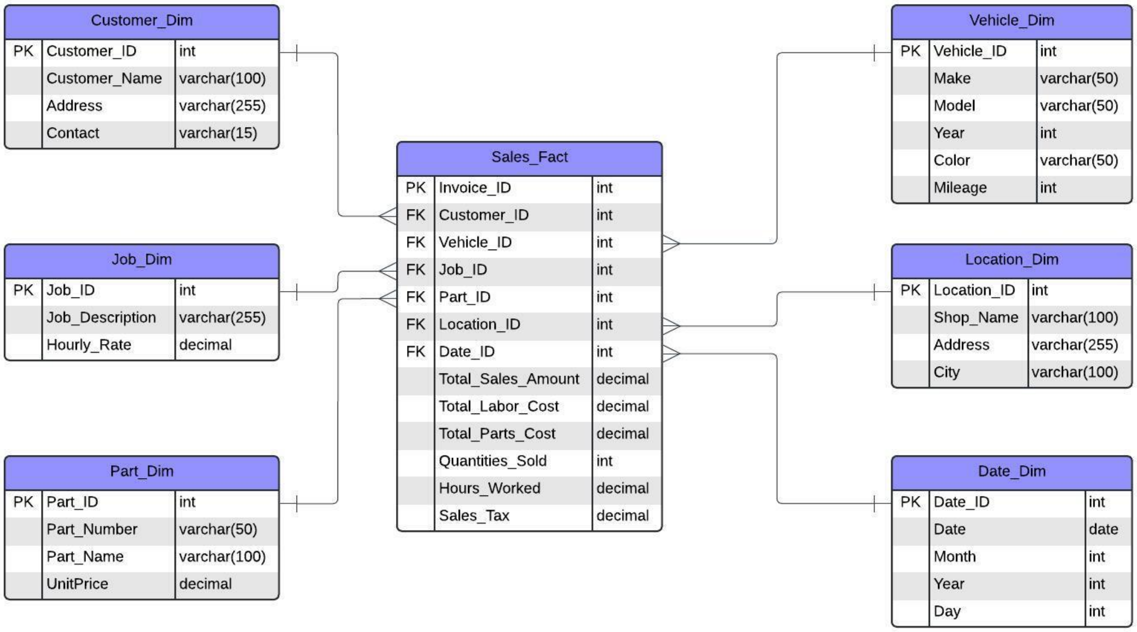
ENTITY RELATIONSHIP DIGRAM (ERD)

The ER diagram visually represents the fact and dimension tables, including their relationships. An ER diagram was created to visually represent the entities in the database and their relationships. This includes the creation of tables, defining primary and foreign keys, and ensuring referential integrity.

IMPORTANT DEFINITIONS

- **ERD (Entity-Relationship Diagram):** A visual representation of the entities in a database and the relationships between them.
- **Primary Key:** A unique identifier for a record in a database table.
- **Foreign Key:** A field in one table that uniquely identifies a row in another table, establishing a relationship between the two tables.
- **Relationship:** The connection between two or more entities based on common attributes.
- **Entities:** Objects or concepts represented in a database, such as customers, vehicles, and jobs.

The below ER diagram was created in Lucidchart



Customer_Dim		
PK	Customer_ID	int
	Customer_Name	varchar(100)
	Address	varchar(255)
	Contact	varchar(15)

Job_Dim		
PK	Job_ID	int
	Job_Description	varchar(255)
	Hourly_Rate	decimal

Part_Dim		
PK	Part_ID	int
	Part_Number	varchar(50)
	Part_Name	varchar(100)
	UnitPrice	decimal

Sales_Fact		
PK	Invoice_ID	int
FK	Customer_ID	int
FK	Vehicle_ID	int
FK	Job_ID	int
FK	Part_ID	int
FK	Location_ID	int
FK	Date_ID	int
	Total_Sales_Amount	decimal
	Total_Labor_Cost	decimal
	Total_Parts_Cost	decimal
	Quantities_Sold	int
	Hours_Worked	decimal
	Sales_Tax	decimal

Vehicle_Dim		
PK	Vehicle_ID	int
	Make	varchar(50)
	Model	varchar(50)
	Year	int
	Color	varchar(50)
	Mileage	int

Location_Dim		
PK	Location_ID	int
	Shop_Name	varchar(100)
	Address	varchar(255)
	City	varchar(100)

Date_Dim		
PK	Date_ID	int
	Date	date
	Month	int
	Year	int
	Day	int

OBSERVATIONS

1. **Fact Table (Sales_Fact):**

   - The Sales_Fact table includes all necessary foreign keys to dimension tables, which is essential for linking various data points.
   - Metrics like Total_Sales_Amount, Total_Labor_Cost, Total_Parts_Cost, Quantities_Sold, Hours_Worked, and Sales_Tax are appropriately included for analysis.

2. **Dimension Tables:**
   - Customer_Dim: Contains relevant attributes such as Customer_ID, Customer_Name, Address, and Contact.
   - Job_Dim: Includes Job_ID, Job_Description, and Hourly_Rate, which are relevant for analyzing service jobs.
   - Vehicle_Dim: Contains attributes like Make, Model, Year, Color, and Mileage, which are crucial for vehicle-related analysis.
   - Part_Dim: Has attributes for parts, including Part_Number, Part_Name, and UnitPrice.
   - Location_Dim: Contains Shop_Name, Address, and City, which are important for location-based analysis.
   - Date_Dim: Includes Date, Month, Year, and Day, allowing for time-based analysis.

Logical Explanations

- **Fact Table (Sales_Fact)**: Central table storing transactional data, including links to all dimensions for detailed analysis.
- **Dimension Tables**: Support analysis by providing context to the facts, enabling queries based on customer, vehicle, job, part, date, and location attributes.
- **Relationships**: Foreign keys in the fact table ensure that each sale is linked to relevant dimensions, maintaining data integrity and supporting comprehensive analysis.

RELATIONSHIPS

The relationships between the Sales_Fact table and each dimension table are correctly represented with foreign keys, ensuring referential integrity.

RELATIONSHIPS EXPLANATIONS BETWEEN FACTS AND DIMENSIONS

1. **Customer_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
  - From Customer_Dim to Sales_Fact:
    - **One-to-Many:** One customer can make multiple purchases.
  - From Sales_Fact to Customer_Dim:
    - **Many-to-One:** Each sale is associated with only one customer.

2. **Vehicle_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
  - From Vehicle_Dim to Sales_Fact:
    - **One-to-Many:** One vehicle can be involved in multiple service transactions.
  - From Sales_Fact to Vehicle_Dim:
    - **Many-to-One:** Each sale is linked to only one vehicle.

3. **Job_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
  - From Job_Dim to Sales_Fact:
    - **One-to-Many:** One job type can be performed multiple times across different sales.
  - From Sales_Fact to Job_Dim:
    - **Many-to-One:** Each sale corresponds to a specific job.

4. **Part_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
  - From Part_Dim to Sales_Fact:
    - **One-to-Many:** One part can be sold in multiple transactions.
  - From Sales_Fact to Part_Dim:
    - **Many-to-One:** Each sale may include one or more parts, but each part is linked to a specific transaction.

5. **Location_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
  - From Location_Dim to Sales_Fact:
    - **One-to-Many:** One location can facilitate multiple sales transactions.
  - From Sales_Fact to Location_Dim:

        - **Many-to-One:** Each sale occurs at one specific location.

6. **Date_Dim to Sales_Fact**

- **Relationship Type:** One-to-Many / Many-to-One
    - From Date_Dim to Sales_Fact:
        - **One-to-Many:** One date can have multiple sales associated with it.
    - From Sales_Fact to Date_Dim:
        - **Many-to-One:** Each sale is recorded on one specific date.

CONCLUSION

The ER diagram and detailed documentation provide a clear and logical structure for understanding and utilizing the data model. The dimensional data model developed for the Car Repair Center Sales Analysis project effectively addresses the business requirements by facilitating comprehensive and flexible analysis of sales performance across multiple dimensions, including customers, vehicles, services, parts, and locations. This model not only enhances the ability to track and evaluate key performance indicators but also supports decision-making processes by providing valuable insights into customer behavior, service efficiency, and sales trends.

The inclusion of a well-structured ER diagram and thorough documentation ensures clarity and ease of understanding, allowing stakeholders to navigate the data model with confidence. By leveraging this dimensional approach, the Car Repair Center can optimize its operations, improve customer satisfaction, and drive strategic growth, ultimately leading to a more data-driven and successful business environment.