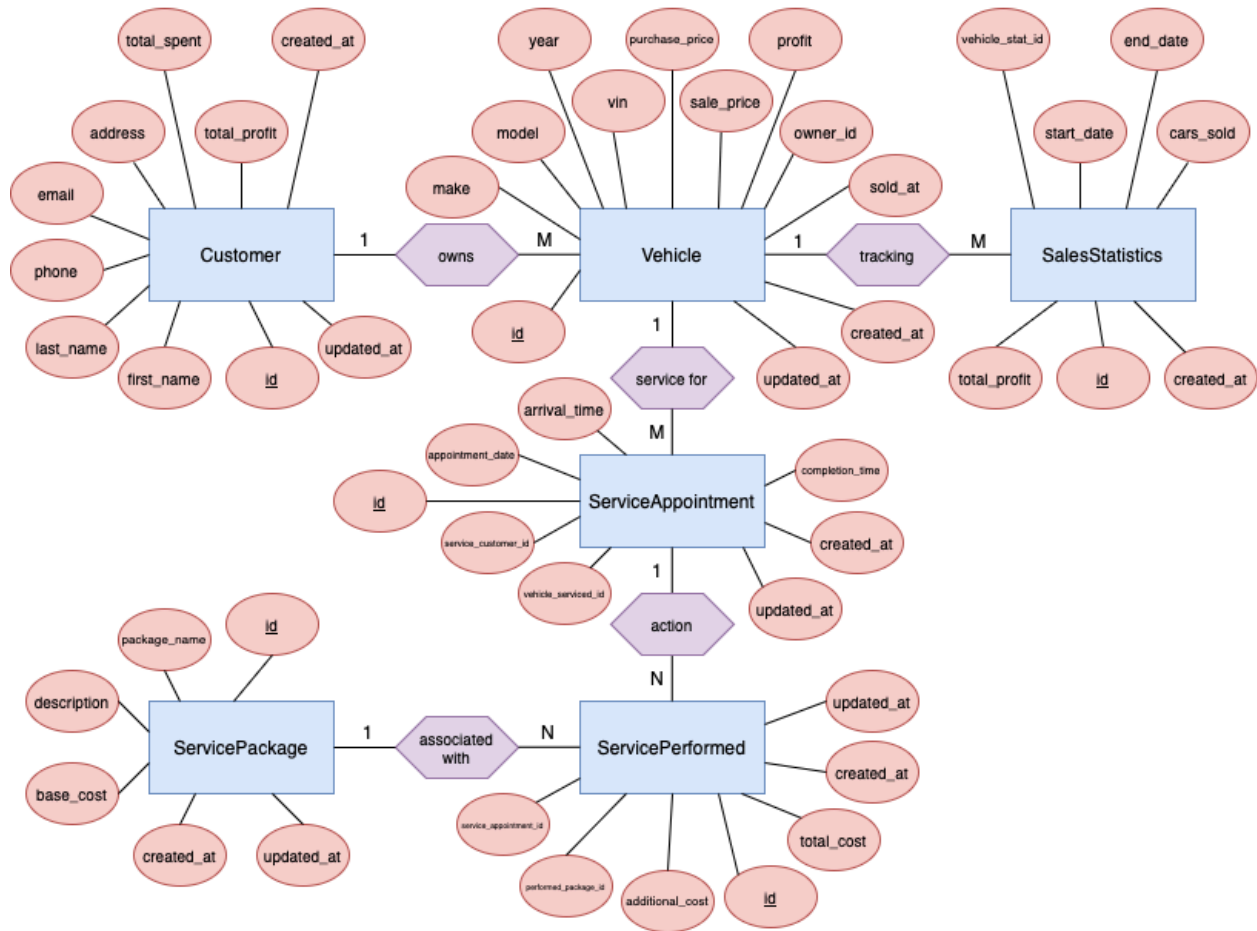Project Report: Car Dealership Service and Sales Management System

Ahmed Mashaal (am3274) & Mohamed Shabarek (mfs27)

CS631 Project

ER Diagram

Customer
- total_spent
- created_at
- address
- total_profit
- email
- phone
- last_name
- first_name
- id
- updated_at

Customer 1 —⬡ owns ⬡— M **Vehicle**

Vehicle
- year
- purchase_price
- profit
- vin
- sale_price
- model
- owner_id
- make
- sold_at
- id
- created_at
- updated_at

Vehicle 1 —⬡ tracking ⬡— M **SalesStatistics**

SalesStatistics
- vehicle_stat_id
- end_date
- start_date
- cars_sold
- total_profit
- id
- created_at

Vehicle 1 —⬡ service for ⬡— M **ServiceAppointment**

ServiceAppointment
- arrival_time
- appointment_date
- id
- completion_time
- service_customer_id
- created_at
- vehicle_serviced_id
- updated_at

ServiceAppointment 1 —⬡ action ⬡— N **ServicePerformed**

ServicePackage
- package_name
- id
- description
- base_cost
- created_at
- updated_at

ServicePackage 1 —⬡ associated with ⬡— N **ServicePerformed**

ServicePerformed
- updated_at
- created_at
- total_cost
- service_appointment_id
- performed_package_id
- additional_cost
- id

Step-by-Step Mapping

Step 1: Handling Entities
For each regular entity type in the ER schema, create a relation (table) and map each attribute.
The primary key of each entity will be the primary key of the corresponding table.

1. **Customer**
   - **Table Name**: Customer
   - **Attributes**: id (PK), first_name, last_name, phone, email, address, total_spent, total_profit, created_at, updated_at
2. **Vehicle**
   - **Table Name**: Vehicle
   - **Attributes**: id (PK), make, model, year, vin (unique), purchase_price, sale_price, profit, owner_id (FK to Customer), sold_at, created_at, updated_at
3. **ServiceAppointment**
   - **Table Name**: ServiceAppointment
   - **Attributes**: id (PK), appointment_date, arrival_time, completion_time, service_customer_id (FK to Customer), vehicle_serviced_id (FK to Vehicle), created_at, updated_at
4. **ServicePackage**
   - **Table Name**: ServicePackage
   - **Attributes**: id (PK), package_name, description, base_cost, created_at, updated_at
5. **SalesStatistics**
   - **Table Name**: SalesStatistics
   - **Attributes**: id (PK), vehicle_stat_id (FK to Vehicle), start_date, end_date, cars_sold, total_profit, created_at

---

Step 2: Mapping Weak Entities

- **ServiceAppointment**
  - **Table Name**: ServiceAppointment
  - **Attributes**: id (PK), appointment_date, arrival_time, completion_time, service_customer_id (FK to Customer), vehicle_serviced_id (FK to Vehicle), created_at, updated_at

- ○ **Weak entity dependent on:** `service_customer_id` → Foreign key references `Customer(id)` (**1 relationship**).`vehicle_serviced_id` → Foreign key references `Vehicle(id)` (**1 relationship**)
- **ServicePerformed**
  - ○ **Table Name**: `ServicePerformed`
  - ○ **Attributes**: id (PK), `service_appointment_id` (FK to `ServiceAppointment`), `performed_package_id` (FK to ServicePackage), `additional_cost`, `total_cost`, `created_at`, `updated_at`
  - ○ **Weak entity dependent on:** `service_appointment_id` → Foreign key references `ServiceAppointment(id)`(**1 relationship**).`performed_package_id` → Foreign key references `ServicePackage(id)` (**1 relationship**).

Step 3: Mapping 1:1 Relationships

1. **Customer and Vehicle (1)**
   - ○ **Foreign Key**: Add `owner_id` in the `Vehicle` table as a foreign key referencing `Customer(id)`.
   - ○ **Explanation**: A `Customer` can own multiple `Vehicles`, but each `Vehicle` belongs to exactly one `Customer`.


Step 4: Mapping 1:N Relationships

1. **Vehicle and SalesStatistics (1)**
   - ○ **Foreign Key**: Add `vehicle_stat_id` in the `SalesStatistics` table as a foreign key referencing `Vehicle(id)`.
   - ○ **Explanation**: A `Vehicle` can be associated with multiple entries in `SalesStatistics` for tracking purposes.
2. **Vehicle and ServiceAppointment (1)**
   - ○ **Foreign Key**: Add `vehicle_serviced_id` in the `ServiceAppointment` table as a foreign key referencing `Vehicle(id)`.
   - ○ **Explanation**: A `Vehicle` can have multiple `ServiceAppointments`.
3. **ServicePackage and ServicePerformed (1)**
   - ○ **Foreign Key**: Add `package_name` in the `ServicePerformed` table as a foreign key referencing `Vehicle(id)`.
   - ○ **Explanation**: A `ServicePackage` can be associated with multiple `ServicePerformed` entries

Final Set of Relations

After completing these steps, your final relational schema is:

1. **Customer**(id, `first_name`, `last_name`, `phone`, `email`, `address`, `total_spent`, `total_profit`, `created_at`, `updated_at`)
2. **Vehicle**(id, `make`, `model`, `year`, `vin`, `purchase_price`, `sale_price`, `profit`, `owner_id`(FK), `sold_at`, `created_at`, `updated_at`)
3. **ServiceAppointment**(id, `appointment_date`, `arrival_time`, `completion_time`, `service_customer_id`(FK), `vehicle_serviced_id`(FK), `created_at`, `updated_at`)
4. **ServicePackage**(id, `package_name`, `description`, `base_cost`, `created_at`, `updated_at`)
5. **ServicePerformed**(id, `service_appointment_id`(FK), `performed_package_id`(FK), `additional_cost`, `total_cost`, `created_at`, `updated_at`)
6. **SalesStatistics**(id, `vehicle_stat_id`(FK), `start_date`, `end_date`, `cars_sold`, `total_profit`, `created_at`)

Relational Schema
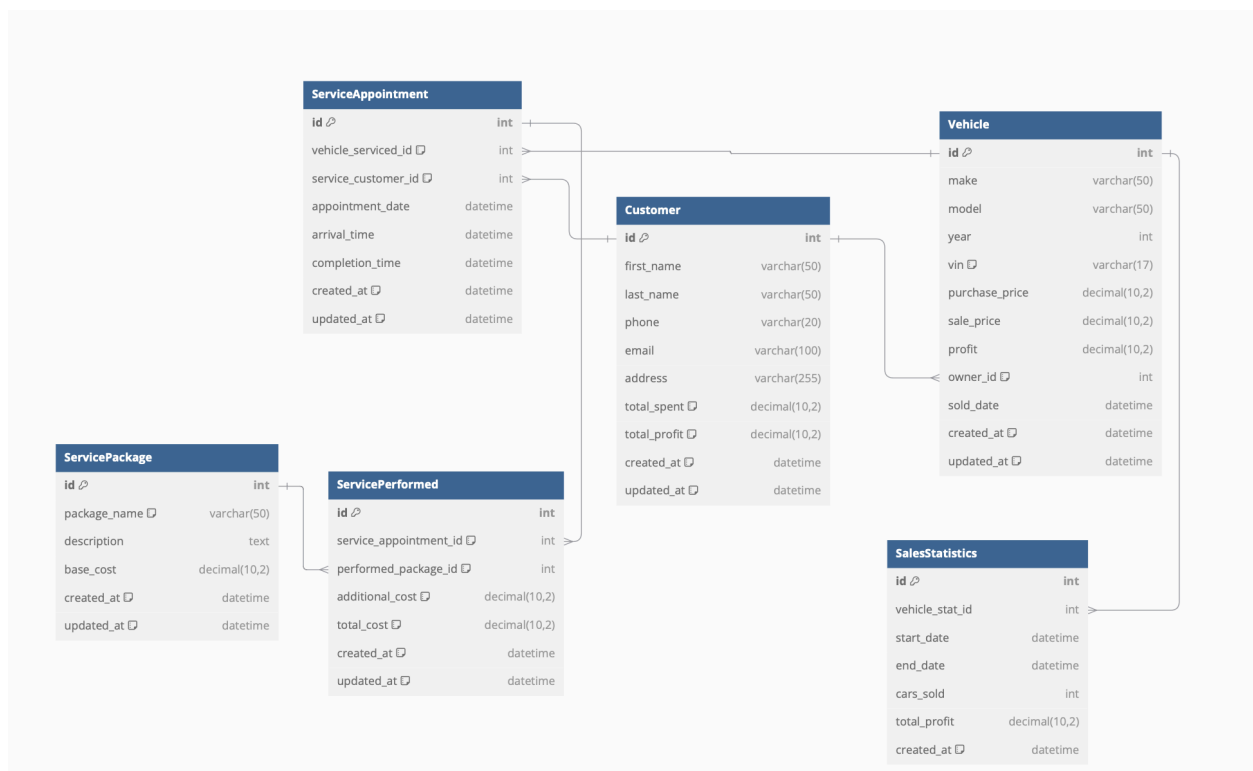
Database Relational Instances

customer table

	customer_id	first_name	last_name	phone	email	address	total_spent	total_profit	created_at	updated_at
1	1	Liam	Miller	123-234-3456	liammiller123@gmail.com	100 Main St	1000.20	505.15	2024-12-15 05:17:45	2024-12-15 05:17:45
2	2	Fiona	Smith	101-202-3003	fionasmith3@gmail.com	220 5th Ave	1256.32	127.54	2024-12-15 05:17:45	2024-12-15 05:17:45
3	3	Ivor	Watson	100-200-3000	ivorwatson400@hotmail.com	521 Broad St	5000.54	883.29	2024-12-15 05:17:45	2024-12-15 05:17:45
4	4	Gary	Monroe	010-020-0330	garymonroe1995@yahoo.com	97 Summit St	8884.23	120.65	2024-12-15 05:17:45	2024-12-15 05:17:45

vehicle table

	vehicle_id	make	model	year	vin	purchase_price	sale_price	profit	owner_id	sold_at	created_at	updated_at
1	1	Toyota	Camry	2015	34164V643948209	20243.21	28554.23	8311.02	1	2023-10-04	2024-12-15 05:17:45	2024-12-15 05:17:45
2	2	Kia	Rio	2012	12745V143578642	18554.64	19948.49	1393.85	2	2020-12-12	2024-12-15 05:17:45	2024-12-15 05:17:45
3	3	Honda	Civic	2008	21542V678904323	15221.30	27382.39	12161.09	3	2022-05-05	2024-12-15 05:17:45	2024-12-15 05:17:45

service_appointment table

	appt_id	appt_date	arrival_time	completion_time	service_customer_id	vehicle_serviced_id	created_at	updated_at
1	1	2024-11-30	09:30:00	10:30:00	1	1	2024-12-15 05:17:45	2024-12-15 05:17:45
2	2	2024-12-07	13:15:00	14:30:00	2	2	2024-12-15 05:17:45	2024-12-15 05:17:45

service_package table

	pkg_id	pkg_name	description	base_cost	created_at	updated_at
1	1	Package 1	null	150.00	2024-12-15 05:17:45	2024-12-15 05:17:45
2	2	Package 2	null	250.00	2024-12-15 05:17:45	2024-12-15 05:17:45
3	3	Package 3	null	350.00	2024-12-15 05:17:45	2024-12-15 05:17:45

sales_stats table

	stats_id	vehicle_stat_id	start_date	end_date	cars_sold	total_profit	created_at
1	1	1	2020-12-15	2023-10-05	10	12234.54	2024-12-15 05:17:45
2	2	2	2020-10-01	2023-11-11	15	14257.32	2024-12-15 05:17:45

Design Decisions and Justifications

1. **Database Normalization:**
   - Decision: Normalize the database to the third normal form to reduce redundancy.
   - Justification: Ensures data integrity and simplifies queries.
2. **PDF Generation:**
   - Decision: Use ReportLab for PDF generation.
   - Justification: It is a robust and flexible library for creating professional documents.
3. **Error Logging:**
   - Decision: Implement detailed logging.
   - Justification: Simplifies debugging and provides an audit trail for errors.
4. **Service Package Flexibility:**
   - Decision: Use a dynamic approach to calculate service costs based on car age and additional requirements.
   - Justification: Enhances scalability and user satisfaction.

User Manual

Instructions for System Usage:

Main System

Car Dealership	Add Customer Schedule Service Sales Statistics

Welcome to the Car Dealership Management System

Manage your sales, services, and statistics with ease.

Car Sale	Schedule Service	Sales Statistics
Record a car sale and print the customer bill.	Book service appointments for customers.	Analyze sales and profit data by period.
Go to Car Sale	Go to Schedule Service	View Sales Statistics

© 2024 Car Dealership Management System

Car Sales Management

1. Navigate to the Sell Car page.
2. Enter vehicle and customer details.
3. Confirm the sale to update the database and generate a bill.

Sell a Car

Vehicle ID

First Name

Last Name

Sale Price

Sale Date

12/15/2024

Submit

Car Dealership Bill

Customer Name: Ahmed Mashaal

Phone: 9982373323

Email: amm@example.com

Vehicle: Kia Rio (2012)

VIN: 12745V143578642

Sale Price: $2715162.00

Sale Date: 2024-12-04

Profit: $2696607.36

Thank you for your business!

Service Scheduling

1. Navigate to the Service page.
2. Select a service package and enter the appointment details.
3. Confirm the appointment to generate a service bill.

Car Dealership	Add Customer Schedule Service Sales Statistics

Schedule Service Appointment

Service Package Package 1 - $150.00 ⟳ Vehicle ID 2   Appointment Date 01/01/2025 Arrival Time 12:30 PM Schedule

© 2024 Car Dealership Management System

Car Dealership Service Bill

Customer Name: Ahmed Mashaal

Phone: 9982373323

Email: amm@example.com

Vehicle: Kia Rio (2012)

Appointment Date: 2024-12-09

Arrival Time: 12:23

Completion Time: N/A

Total Cost: $150.00

Thank you for your business!

Viewing Statistics

1. Navigate to the Sales Statistics page.
2. Enter the desired date range.
3. View detailed reports on sales and profits.

Sales Statistics

Start Date
12/15/2024

End Date
12/15/2024

Get Statistics

Make	Model	Cars Sold	Total Profit
Toyota	Camry	1	$193189.79
Kia	Rio	1	$2696607.36
Honda	Civic	1	$12161.09

Common Issues and Solutions

1. Issue: Missing customer or vehicle data.
   - Solution: Ensure all required fields are populated and valid.
2. Issue: PDF generation errors.
   - Solution: Confirm that the ReportLab library is installed and configured.

Challenges and Resolutions

Challenges

1. **Handling Decimal Arithmetic Errors:**
   - **Error:** `unsupported operand type(s) for +: 'decimal.Decimal' and 'float'`
   - **Resolution:** Ensured all calculations used consistent data types (Decimal for all monetary values).
2. **Missing Service Packages:**
   - **Error:** `Service package not found: None`
   - **Resolution:** Added validation to check for package existence before proceeding.
3. **TemplateNotFound Errors:**
   - **Error:** `jinja2.exceptions.TemplateNotFound: schedule_service.html`
   - **Resolution:** Renamed templates and ensured paths matched the routing logic.
4. **PDF File Not Found:**
   - **Error:** `[Errno 2] No such file or directory: 'service_bill_x.pdf'`

- ○ **Resolution:** Created a `temp/` directory for generated files and cleaned up after serving them.
5. **Divergent Branches in Git:**
   - ○ **Error:** `fatal: Need to specify how to reconcile divergent branches.`
   - ○ **Resolution:** Used `git push --force` to align the remote branch with the local repository.

Appendix: Sample Data

Customer Table

customer_id	first_name	last_name	total_spent	total_profit
1	John	Doe	15000.00	2000.00
2	Jane	Smith	20000.00	2500.00

Vehicle Table

vehicle_id	make	model	year	owner_id	profit
1	Toyota	Camry	2020	1	2000.00
2	Honda	Accord	2019	2	2500.00

Service_Appointment Table

appt_id	vehicle_serviced_id	total_cost
1	1	300.00
2	2	250.00