

# **PROJECT DOCUMENTATION**

## ***“MetCar Car Dealership ”***

**Database Systems  
COEN 280**

**Group Member**

**Rutuja Shivde  
Student Id: W0756495  
(Single Person Team)**

# TABLE OF CONTENTS

<b>TABLE OF CONTENTS.....</b>	<b>2</b>
<b>1.0 INTRODUCTION.....</b>	<b>3</b>
1.1 Requirements.....	3
<b>2.0 DESIGN.....</b>	<b>4</b>
2.1 E-R Diagram.....	4
2.2 Table Schema .....	4
2.3 Normalization .....	5
<i>2.3.1 Functional Dependencies.....</i>	<i>5</i>
<i>2.3.2 Form of Normalization.....</i>	<i>5</i>
<b>3.0 IMPLEMENTATION .....</b>	<b>8</b>
3.1 Table Descriptions and Inserted Data.....	8
3.2 Procedures Implemented .....	8
3.3 Query and Output.....	8
3.4 Limitations .....	12
3.5 Extra functionality implemented beyond the given requirement .....	12
3.6 Assumptions .....	12
<b>Appendix A: List of attached files .....</b>	<b>14</b>

## 1.0 INTRODUCTION

**Project Name:** MetCar Car Dealership

**Project Group Member:** Rutuja Shivde (Student Id: W0756495)

### **Brief Project Description:**

It's an application built for maintaining the records at MetCar Car Dealership. The system assists the management of the dealership to record new repair jobs, new customers, billing, etc by providing them an easy access to the dealership database. The system provides a nice and simple user interface for accomplishing these different tasks.

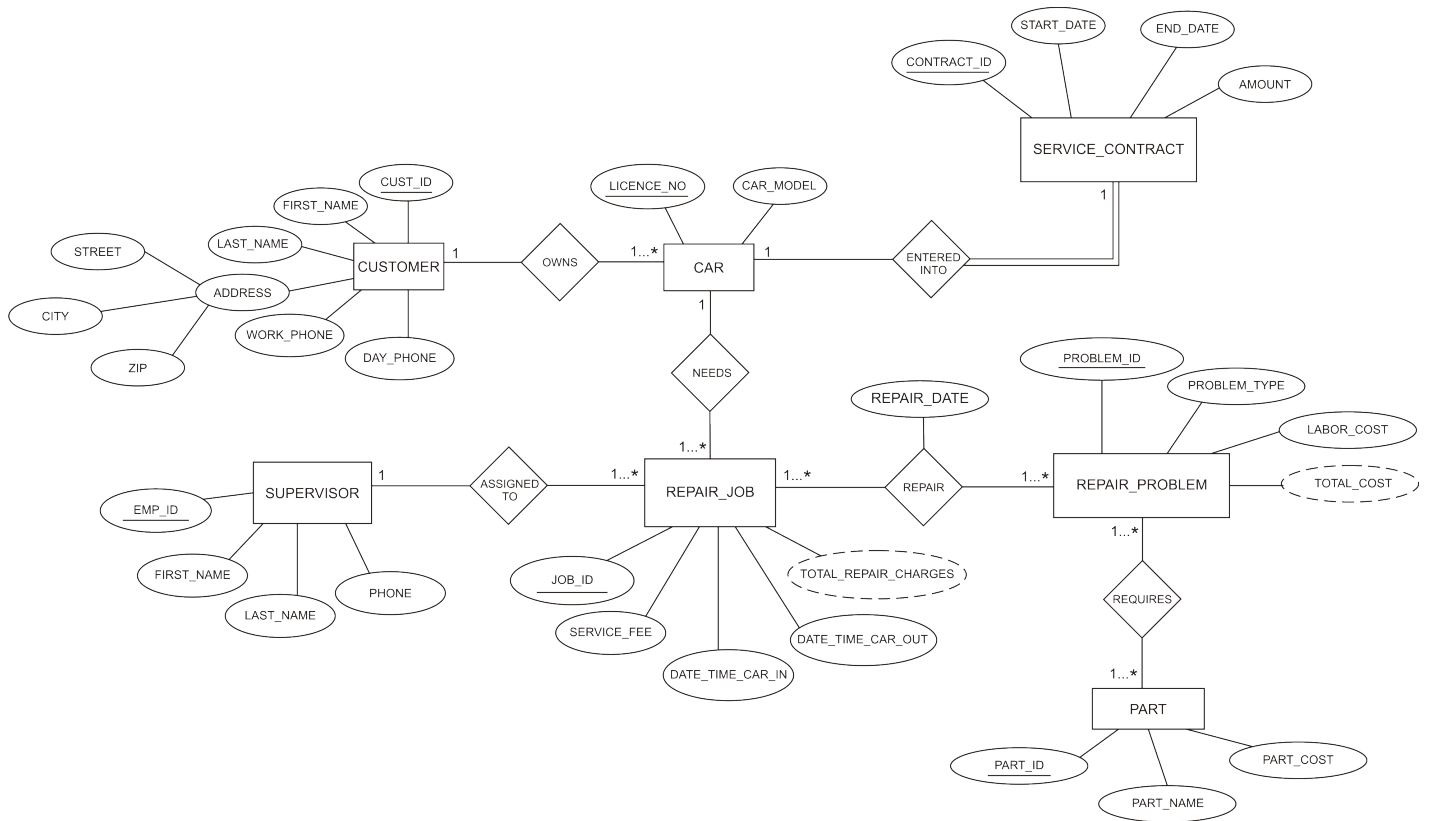
### 1.1 Requirements

- a. Maintain a detailed list of the customer records with the details of the repair jobs done. The management uses the customer records to schedule the routine maintenance dates for its regular customers.
- b. Maintain a detailed list of the customers with service contracts.
- c. Maintain a record of the repair jobs to assist the management in customer billing.
- d. All the information in the database helps the management assess their monthly and yearly revenues and the future areas for business expansion.

## 2.0 DESIGN

### 2.1 E-R Diagram

( E-R Diagram.jpeg file also attached separately.)



### 2.2 Table Schema

1. customer ( cust\_id, first\_name, last\_name, street, city, zip, work\_phone, day\_phone )
2. car ( licence no, cust\_id, car\_model )
3. service\_contract ( contract id, car\_licence, start\_date, end\_date, amount )
4. supervisor ( emp\_id, first\_name, last\_name, phone )

5. repair\_job ( job\_id, service\_fee, date\_time\_car\_in, date\_time\_car\_out, total\_repair\_charges, emp\_id, car\_licence )
6. repair\_problem ( problem\_id, problem\_type, labor\_cost, total\_cost )
7. repair ( job\_id, problem\_id, repair\_date )
8. part ( part\_id, part\_name, part\_cost )
9. requires ( problem\_id, part\_id )

## 2.3 Normalization

### 2.3.1 Functional Dependencies

#### 1. R: customer

cust\_id -> first\_name, last\_name, street, city, zip, work\_phone, day\_phone

#### 2. R: car

licence\_no -> car\_model, cust\_id

#### 3. R: service\_contract

contract\_id -> start\_date, end\_date, amount, car\_licence

#### 4. R: supervisor

emp\_id -> first\_name, last\_name, phone

#### 5. R: repair\_job

job\_id -> service\_fee, date\_time\_car\_in, date\_time\_car\_out, total\_repair\_charges, emp\_id, car\_licence

#### 6. R: repair\_problem

problem\_id -> problem\_type, labor\_cost, total\_cost

#### 7. R: repair

( job\_id, problem\_id ) -> repair\_date

#### 8. R: part

part\_id -> part\_name, part\_cost

### 2.3.2 Form of Normalization

#### 1. R: customer

Closure of determinant:

(cust\_id)+ = cust\_id, first\_name, last\_name, street, city, zip, work\_phone, day\_phone

So, cust\_id is superkey

Therefore, relation customer is in BCNF.

## **2. R: car**

Closure of determinant:

( licence\_no )+ = licence\_no, car\_model, cust\_id

So, licence\_no is superkey

Therefore, relation car is in BCNF.

## **3. R: service\_contract**

Closure of determinant:

( contract\_id )+ = contract\_id, start\_date, end\_date, amount, car\_licence

So, contract\_id is superkey

Therefore, relation service\_contract is in BCNF.

## **4. R: supervisor**

Closure of determinant:

( emp\_id )+ = emp\_id, first\_name, last\_name, phone

So, emp\_id is superkey

Therefore, relation supervisor is in BCNF.

## **5. R: repair\_job**

Closure of determinant:

( job\_id )+ = job\_id, service\_fee, date\_time\_car\_in, date\_time\_car\_out,  
total\_repair\_charges, emp\_id, car\_licence

So, job\_id is superkey

Therefore, relation repair\_job is in BCNF.

## **6. R: repair\_problem**

Closure of determinant:

$(\text{problem\_id})^+ = \text{problem\_id}, \text{problem\_type}, \text{labor\_cost}, \text{total\_cost}$

So,  $\text{problem\_id}$  is superkey

Therefore, relation  $\text{repair\_problem}$  is in BCNF.

## **7. R: repair**

Closure of determinant:

$(\text{job\_id}, \text{problem\_id})^+ = \text{job\_id}, \text{problem\_id}, \text{repair\_date}$

So,  $(\text{job\_id}, \text{problem\_id})$  is superkey

Therefore, relation  $\text{repair}$  is in BCNF.

## **8. R: part**

Closure of determinant:

$(\text{part\_id})^+ = \text{part\_id}, \text{part\_name}, \text{part\_cost}$

So,  $\text{part\_id}$  is superkey

Therefore, relation  $\text{part}$  is in BCNF.

## 3.0 IMPLEMENTATION

**Programming Environment:** PHP, HTML, CSS, Apache Web Server

**Database:** MYSQL

### 3.1 Table Descriptions and Inserted Data

( MetCarTables.sql file attached)

### 3.2 Procedures Implemented

Following **implementation files** are attached –

1. MetCarHome.html
2. MetCarNewCustomer.php
3. Invoice.php
4. MetCarExistingCustomer.php
5. MetCarBill.php
6. MetCarServiceContract.php
7. MetCarRepairJob.php
8. MetCarJobList.php
9. MetCarSupervisor.php
10. MetCarSupervisorList.php
11. MetCarBalance.php
12. MetCarShowBalance.php
13. MetCarCSS.css

MySQL **database table creation and insertion query file** –

1. MetCarMYSQL.sql

### 3.3 Query and Output

Following are the different queries implemented for the MetCar Car Dealership. The SQL queries are embedded inside the project code that is in PHP. The specific queries are located inside the files mentioned in front of each query option.

1. **Create customer invoice** – Invoice.php



**Following tables gets updated** – customer, car, service\_contract, repair\_job, repair

**Queries used in showing the output–**

```
1.  
"insert into customer values('".$custId."', '".$firstName."',  
".$lastName."', '".$street."', '".$city."', ".$zip.",  
".$workPhone."', '".$dayPhone."')"
```

```
2.  
"insert into car values('".$carLicence."', ".$custId.",  
".$carModel."')"
```

```
3.  
"insert into service_contract values('".$contractId."',  
".$carLicence."', CURDATE(), ADDDATE(CURDATE(), INTERVAL 3  
YEAR), ".$amount.")"
```

```
4.  
"insert into repair_job values ('".$jobId."', ".$serviceFee.",  
CURRENT_TIMESTAMP(), TIMESTAMPADD(DAY, 4,  
CURRENT_TIMESTAMP()), ".$totalBill.", ".$empId.",  
".$carLicence."')"
```

```
5.  
"insert into repair values('".$jobId."', ".$probId.",  
ADDDATE(CURDATE(), INTERVAL 4 DAY))"
```

## 2. Create bill for existing customer – MetCarBill.php

**Following tables are used for bill creation** – customer, car, repair\_job, repair\_problem, part, service\_contract

**Queries used in showing the output–**

```
1. "select * from customer where day_phone = '".$billPhone."'"  
2. "select * from car where cust_id = ".$custId  
3. "select * from car where cust_id = ".$custId." and  
   car_model = '".$carModel."'"  
4. "select * from repair_job where car_licence =  
   '".$licence."'"
```

5. `"select * from repair_problem where problem_id in (select problem_id from repair where job_id = ".$repairJobId.")"`
6. `"select * from part where part_id in (select part_id from requires where problem_id = ".$repairProblemId.")"`
7. `"select * from service_contract where car_licence = ".$licence."' and end_date > CURDATE()"`

3. **Show list of all current service contracts** – MetCarServiceContract.php

**Following tables are used** – service\_contract

**Query used in showing the output–**

`"select * from service_contract where end_date >= CURDATE()"`

4. **Show list of all repair jobs & their details between two specific dates** – MetCarJobList.php

**Following tables are used** – repair\_job, supervisor, car, repair\_problem

**Query used in showing the output–**

`"select *  
from repair_job  
where DATE(date_time_car_out) >= ".$startDate."' and  
DATE(date_time_car_out) <= ".$endDate."'"`

5. **Show list of all supervisors** – MetCarSupervisorList.php

**Following tables are used** – supervisor

**Query used in showing the output–**

`"select * from supervisor"`

6. **Show Supervisor name who was assigned to most jobs between two specific dates** - MetCarSupervisorList.php

**Following tables are used** – repair\_job, supervisor

**Queries used in showing the output–**

1. `"select emp_id, count(job_id)  
from repair_job  
where DATE(date_time_car_out) >= ".$startDate."' and  
DATE(date_time_car_out) <= ".$endDate."'  
group by emp_id"`

```

having count(job_id) >= all(select count(job_id)
                           from repair_job
                           where DATE(date_time_car_out) >= '". $startDate.'"
                           and DATE(date_time_car_out) <= '". $endDate.'"
                           group by emp_id)"

```

```

2. $supervisor = $row['emp_id'];
   $sql1 =
       "select * from supervisor where emp_id = ".$supervisor

```

7. **Show Supervisor name who was assigned to least jobs between two specific dates** - MetCarSupervisorList.php

**Following tables are used** – repair\_job, supervisor

**Queries used in showing the output-**

```

1. "select emp_id, count(job_id)
   from repair_job
   where DATE(date_time_car_out) >= '". $startDate.'" and
   DATE(date_time_car_out) <= '". $endDate.'"
   group by emp_id
   having count(job_id) <= all(select count(job_id)
                               from repair_job
                               where DATE(date_time_car_out) >= '". $startDate.'"
                               and DATE(date_time_car_out) <= '". $endDate.'"
                               group by emp_id )"

```

```

2. $supervisor = $row['emp_id'];
   $sql1 =
       "select * from supervisor where emp_id = ".$supervisor

```

8. **Show average no. of jobs each supervisor was assigned to between two specific dates** - MetCarSupervisorList.php

**Following tables are used** – repair\_job, supervisor

**Queries used in showing the output-**

```

"select avg(job_id)
from (select count(job_id) as job_id
      from repair_job
      where DATE(date_time_car_out) >= '". $startDate.'" and
      DATE(date_time_car_out) <= '". $endDate.'"

```

```
group by emp_id) as temp"
```

9. **Show amount generated from customer billing between two specific dates**  
– MetCarShowBalance.php

**Following tables are used** – repair\_job, service\_contract, repair\_problem

**Queries used in showing the output–**

1. "select \* from repair\_job  
where DATE(date\_time\_car\_out) >= '". \$startDate.'" and  
DATE(date\_time\_car\_out) <= '". \$endDate.'"'"
2. "select \* from service\_contract  
where car\_licence = '". \$licence
3. "select \* from repair\_problem  
where problem\_id in (select problem\_id  
from repair  
where job\_id = '". \$repairJobId."  
)"

The outputs of these queries are captured in the associated tables. The attached file MetCarTables.sql shows the outputs. Plus, the screen-shots of the forms are also attached.

### 3.4 Limitations

The MetCar Car Dealership system fulfills all the basic required functionalities. All the basic required queries are working fine without any problems.

### 3.5 Extra functionality implemented beyond the given requirement

No extra functionality is implemented. All the basic functionalities are fulfilled.

### 3.6 Assumptions

Following are some of the assumptions of the MetCar Car dealership System implemented –

1. When a new repair job is inserted into the system, the 'date\_time\_car\_in' attribute of repair-job table is set to the date/time of the moment the invoice gets created.

2. The repair job is assumed to get completed in 4 days. So, 'date\_time\_car\_out' attribute of the repair\_job table automatically gets set to the date 4 days ahead of the moment the invoice gets created.
3. There are some fixed number of repair problems, dealership deals with and the parts required for that problem are also fixed.
4. A customer can have more than one car.
5. If car is having transmission problem and it has a service contract, then he gets 20% discount.
6. The service contract is valid for 3 years from the date it's activated. The start\_date of the contract is set to the current date of the invoice creation.

## **Appendix A: List of attached files**

1. E-R Diagram.jpeg
2. Implementation Files
3. Screen Shots of the UI pages