**Normalisation of Data**

**Assumptions to satisfy for 1NF**

* Atomicity of each field, that is, there are no repeating groups a cell
* No repeated rows, that is , a primary key is specified to prevent duplicating rows
* Data ordering does not matter
* The use of unique column names

**Database Schema Mapping for the Car Servicing System**

The schema contains tables that capture information about **customers**, **cars**, **services**, and **mechanics**. Below is the database schema, followed by the ER diagram.

**Customers Table**

|  |  |  |  |
| --- | --- | --- | --- |
| customer\_id | name | email\_address | phone\_number |

**Cars Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| reg\_no | make | model | manufacture\_date | customer\_id |

**Service Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| service\_id | drop\_off\_date | description | mileage | next\_service |

**Mechanic Table**

|  |  |  |  |
| --- | --- | --- | --- |
| mechanic\_id | name | phone\_number | grade |

**Service Mechanic Table**

|  |  |  |
| --- | --- | --- |
| service\_id | mechanic\_id | time\_spent |

**Description of tables**

**1. Customers Table**

The Customers table holds information about customers.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| customer\_id | VARCHAR (PK) | Unique identifier for each customer. |
| name | VARCHAR(255) | Customer's full name. |
| email\_address | VARCHAR(255) | Customer's email address. |
| phone\_number | VARCHAR(15) | Customer's phone number. |

**2. Cars Table**

The Cars table holds information about the cars.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| reg\_no | VARCHAR(15) (PK) | Car's registration number, Unique identifier for each car. |
| make | VARCHAR(100) | Make of the car (e.g., Toyota, Ford). |
| model | VARCHAR(100) | Model of the car. |
| manufacture\_date | DATE | Date the car was manufactured. |
| customer\_id | VARCHAR (FK) | The customer who owns the car (foreign key). |

**3. Services Table**

The Services table holds information about each service.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| service\_id | VARCHAR (PK) | Unique identifier for each service. |
| drop\_off\_datetime | DATETIME | Date and time the car was dropped off for service. |
| description | TEXT | Description of the work to be done. |
| mileage | INT | Mileage of the car at the time of service. |
| next\_service\_date | DATE | Date for the next scheduled service. |
| reg\_no | VARCHAR(15) (FK) | The car associated with the service (foreign key). |

**4. Mechanics Table**

The Mechanics table holds information about each mechanic.

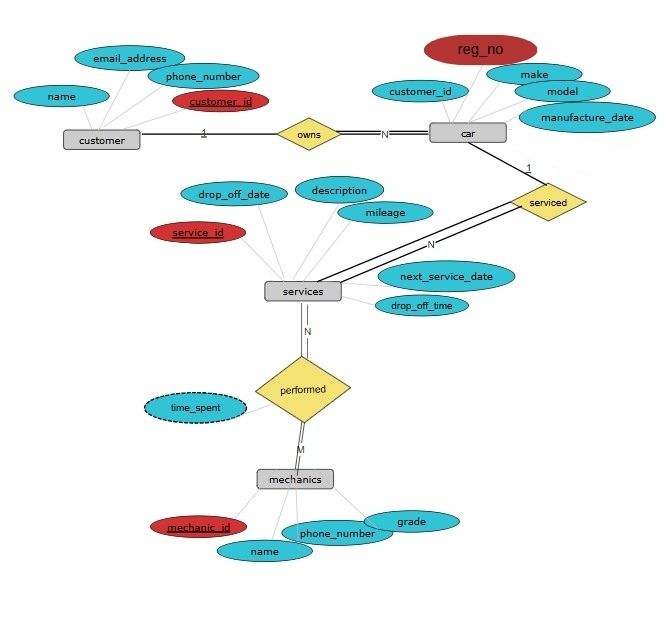
| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| mechanic\_id | INT (PK) | Unique identifier for each mechanic. |
| name | VARCHAR(255) | Mechanic's full name. |
| phone\_number | VARCHAR(15) | Mechanic's phone number. |
| grade | VARCHAR(50) | Grade of the mechanic (e.g., Junior, Senior). |

**5. Service Mechanics Table**

The Service\_Mechanics table holds information about which mechanics worked on which services, as multiple mechanics may work on one service.

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| service\_id | INT (FK) | The service ID (foreign key). |
| mechanic\_id | INT (FK) | The mechanic ID (foreign key). |
| time\_spent | INT | Time spent by the mechanic on the service (in hours). |

**ER Diagram**



Here is how the entities relate to each other:

1. **Customers** and **Cars**: A customer can own many cars, but each car belongs to only one customer. The cardinality is **1 to many (1:N)** relationship between Customers and Cars.
2. **Cars** and **Services**: A car can have many services over time, but each service is associated with exactly one car. The cardinality is **1 to many** (1:N) relationship between Cars and Services.
3. **Services** and **Mechanics**: A service can be performed by one or more mechanics, and each mechanic can work on many services. The cardinality is **Many to many** (N:M) relationship between Services and Mechanics, which is implemented via the Service\_Mechanics table.
4. **Mechanics** and **Services**: Each mechanic works on many services. The cardinality is **Many to many** (N:M) relationship between Services and Mechanics.

**ER Diagram Description**

* **Customer** (1) → (many) **Car** (one-to-many)
* **Car** (1) → (many) **Service** (one-to-many)
* **Service** (many) → (many) **Mechanic** (many-to-many through Service\_Mechanics)

**SQL Schema Code**

-- Customers table

CREATE TABLE Customers (

customer\_id VARCHAR(255) PRIMARY KEY UNIQUE, -- this constraint makes this attribute the primary key as well as unique from other entries

name VARCHAR(255),

email\_address VARCHAR(255) UNIQUE,

phone\_number VARCHAR(15)

);

-- Cars table

CREATE TABLE Cars (

registration\_no VARCHAR(15) PRIMARY KEY UNIQUE,

make VARCHAR(100),

model VARCHAR(100),

manufacture\_date DATE,

customer\_id VARCHAR (255),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id) ON DELETE CASCADE

);

-- Services table

CREATE TABLE Services (

service\_id VARCHAR(255) PRIMARY KEY,

drop\_off\_date DATE NOT NULL,

drop\_off\_time TIME NOT NULL,

description TEXT,

mileage INT,

next\_service\_date DATE,

registration\_no VARCHAR(15),

FOREIGN KEY (registration\_no) REFERENCES Cars(registration\_no) ON DELETE CASCADE

);

-- Mechanics table

CREATE TABLE Mechanics (

employee\_id VARCHAR(20) PRIMARY KEY,

name VARCHAR(255) UNIQUE,

phone\_number VARCHAR(15) UNIQUE,

grade VARCHAR(50)

);

-- Service\_Mechanics table (to handle the many-to-many relationship)

CREATE TABLE Service\_Mechanics (

service\_id VARCHAR(255),

employee\_id VARCHAR(20),

time\_spent TIME,

PRIMARY KEY (service\_id, employee\_id),

FOREIGN KEY (service\_id) REFERENCES Services(service\_id),

FOREIGN KEY (employee\_id) REFERENCES Mechanics(employee\_id),

CONSTRAINT chk\_time\_spent\_format CHECK (TIME\_TO\_SEC(time\_spent) % 60 = 0) -- Ensures seconds are always zero

);