Names: Harolt Farinango, Mateo Pilaquinga

DESIGN AND IMPLEMENTATION OF A DATABASE FOR A CAR INSURANCE BUSINESS

# Objectives

Understand the complete database design cycle, from conceptualization to implementation, applied to the context of an insurance agency.

Apply theoretical database knowledge in practical scenarios, managing customers, policies, claims, and payments.

Gain practical experience in the design, implementation, and administration of a relational database system for a real insurance business.

Develop critical thinking and problem-solving skills when addressing complex relationships, constraints, and reporting needs inherent in insurance operations.

# Description

This project consists of the design and implementation of a relational database for an auto insurance company. The system manages customers, vehicles, policies, coverage, premium payments, claims, and claim payments. The database ensures referential integrity and allows for queries essential to business operations, such as policy status, pending payments, and claims history.

1. **Requirments**

**Functional Requirements:**

-Management of customers, vehicles, agents, and coverage.

-Creation and modification of policies linked to customers, vehicles, and agents.

-Recording of premium payments and claims.

-Queries on outstanding balances and totals paid per claim or policy.

-Change of policy status (active, expired, canceled, in arrears).

**Non-Functional (Quality) Requirements:**

-Portability through Docker and MySQL.

-Data integrity and business rules.

-Security, performance, and reliability.

-Maintainability and scalability.

1. **Conceptual Model (ERD)**

The conceptual model represents entities and their relationships without implementation details.

· **Customer** (CustomerID, FirstName, LastName, Address, Phone, Email)

· **Vehicle** (VehicleID, Brand, Model, Year, LicensePlate)

· **Agent** (AgentID, Name, Phone, Email)

· **Policy** (PolicyID, StartDate, EndDate, Status)

· **Coverage** (CoverageID, Name, Description)

· **PremiumPayment** (PaymentID, PaymentDate, Amount)

· **Claim** (ClaimID, ClaimDate, Description, Status)

· **ClaimPayment** (ClaimPaymentID, PaymentDate, Amount)

**Relationships and Cardinalities:**

\* Customer – owns –> Vehicle (1:N)

\*Agent – manages –> Policy (1:N)

\*Vehicle – is covered by –> Policy (1:N)

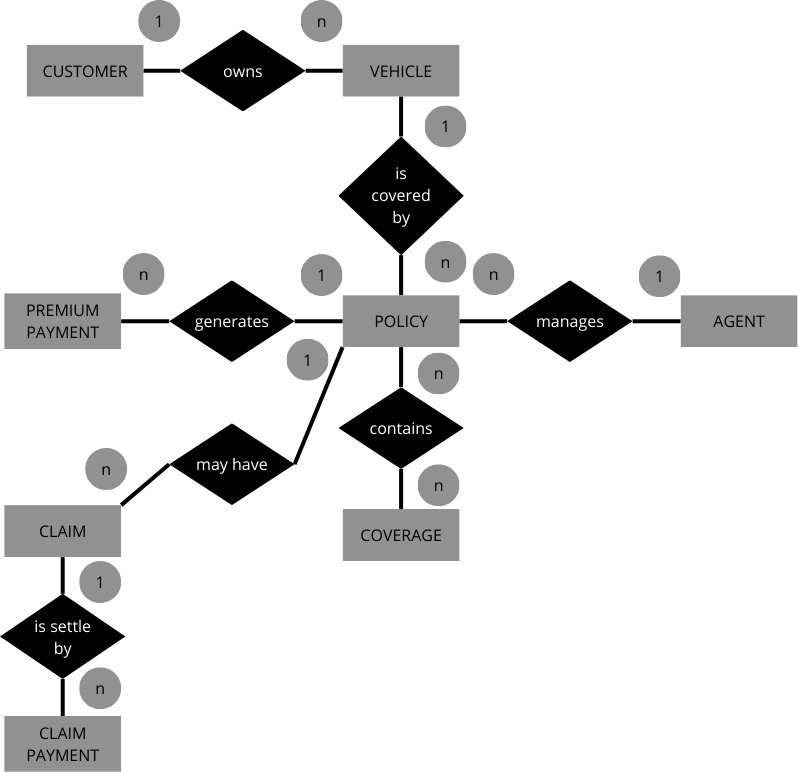
\*Policy – includes –> Coverage (N:M, via PolicyCoverage)

\*Policy – generates –> PremiumPayment (1:N)

\*Policy – may have –> Claim (1:N)

\*Claim – is settled by –> ClaimPayment (1:N)

**ERD Diagram**



1. **Logical Model**

**Customer**  
(CustomerID **PK**, FirstName, LastName, Address, Phone, Email UNIQUE)

**Vehicle**  
(VehicleID **PK**, CustomerID **FK**, Brand, Model, Year, LicensePlate UNIQUE)

**Agent**  
(AgentID **PK**, Name, Phone, Email UNIQUE)

**Policy**  
(PolicyID **PK**, CustomerID **FK**, VehicleID **FK**, AgentID **FK**, StartDate, EndDate, Status)

**Coverage**  
(CoverageID **PK**, Name, Description)

**PolicyCoverage**  
(PolicyID **FK**, CoverageID **FK**, **PK** = (PolicyID, CoverageID))

**PremiumPayment**  
(PaymentID **PK**, PolicyID **FK**, PaymentDate, Amount)

**Claim**  
(ClaimID **PK**, PolicyID **FK**, ClaimDate, Description, Status)

**ClaimPayment**  
(ClaimPaymentID **PK**, ClaimID **FK**, PaymentDate, Amount)

1. **Physical Model (SQL Script)**

CREATE DATABASE CarInsuranceDB;

USE CarInsuranceDB;

CREATE TABLE Customer (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

address VARCHAR(100),

phone VARCHAR(20),

email VARCHAR(50) UNIQUE

);

CREATE TABLE Vehicle (

vehicle\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

brand VARCHAR(50),

model VARCHAR(50),

year INT,

license\_plate VARCHAR(20) UNIQUE,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

CREATE TABLE Agent (

agent\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

phone VARCHAR(20),

email VARCHAR(50) UNIQUE

);

CREATE TABLE Policy (

policy\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

vehicle\_id INT,

agent\_id INT,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

status ENUM('active','expired','canceled','arrears'),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (vehicle\_id) REFERENCES Vehicle(vehicle\_id),

FOREIGN KEY (agent\_id) REFERENCES Agent(agent\_id)

);

CREATE TABLE Coverage (

coverage\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

description TEXT

);

CREATE TABLE PolicyCoverage (

policy\_id INT,

coverage\_id INT,

PRIMARY KEY (policy\_id, coverage\_id),

FOREIGN KEY (policy\_id) REFERENCES Policy(policy\_id),

FOREIGN KEY (coverage\_id) REFERENCES Coverage(coverage\_id)

);

CREATE TABLE PremiumPayment (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

policy\_id INT,

payment\_date DATE NOT NULL,

amount DECIMAL(10,2) NOT NULL,

FOREIGN KEY (policy\_id) REFERENCES Policy(policy\_id)

);

CREATE TABLE Claim (

claim\_id INT AUTO\_INCREMENT PRIMARY KEY,

policy\_id INT,

claim\_date DATE NOT NULL,

description TEXT,

status ENUM('open','closed','in review'),

FOREIGN KEY (policy\_id) REFERENCES Policy(policy\_id)

);

CREATE TABLE ClaimPayment (

claim\_payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

claim\_id INT,

payment\_date DATE NOT NULL,

amount DECIMAL(10,2) NOT NULL,

FOREIGN KEY (claim\_id) REFERENCES Claim(claim\_id)

)