UNIVERSITY OF GHANA, LEGON

DEPARTMENT OF COMPUTER ENGINEERING
SCHOOL OF ENGINEERING
SEMESTER 1 2022/2023 ACADEMIC YEAR

DATABASE END-OF-SEMESTER PROJECT
DESIGNING A DATABASE FOR AN INSURANCE COMPANY

Course Code /Title: CPEN 211 Database System Design
Credits: 3

CPEN 211 GROUP 2

MICHELLE OWUSU - 10957340
BENTIL B. REXFORD - 10946257
MENSAH NYANYO HUBERT - 10976127
EVANS ACHEAMPONG - 10987644
ANANE GEORGE NYARKO - 10947340
APPIAH YAW FRIMPONG - 10987818
DERY-KUUZUME SANDRA - 10986424

Insurance Management System Project

Introduction

Insurance is a critical aspect of financial management, and insurance companies need to manage data and customer information effectively to provide quality services. The Insurance Management System project aims to develop an online application that automates work procedures for an insurance company. The proposed system is designed to manage data and records, including customer information, policy details, and transactions. This report provides a detailed description of the project, including the system's requirements, design, implementation, and testing.

Abstract

An organization must have accurate and reliable data for effective decision-making. To this end, the organization maintains records on the various facets of maintaining relationships among them. Such related data are called a database. A database system is an integrated collection of related files, along with details of the interpretation of the data contained therein. Basically, the database system is nothing more than a computer-based record-keeping system, i.e., a system whose overall purpose is to record and maintain information/data.

For our project, we would be designing a database system for an insurance company/agency which would aid in storing, managing, retrieving, and making changes in the records of the agency. This vacation, we chanced on a couple of insurance companies and agencies which needed software, and in effect a database, for their clients. Most of them required software that would have the records of all their clients, client details, details of insurance policies, and other necessary data the insurer needs. We realized that most of these agencies use notebooks and pens to manually enter their businesses thus making it difficult to keep records of every one of their clients and hence, the design of a database system would relegate manual record-keeping.

The main aim of this project is to develop an online application for the insurance company to atomize work procedures. Using this system, agents and policyholders can know details about present policies, policy specifications, terms and conditions on a stated policy, and policy registration by the customers. Agent's commission is based upon customer policy

registration and target agent achieves for every month or year. This system maintains information about branch managers who can deal with agents and customers.
In the existing system, a manual procedure where records are used to manage data is employed. This is a time-taking process and requires much manpower. Calculations and calculating commissions and dues are done manually.

Insurance

Definition: Insurance refers to a contractual arrangement in which one party, i.e. insurance company or the insurer, agrees to compensate the loss or damage sustained to another party, i.e. the insured, by paying a definite amount, in exchange for an adequate consideration called a premium.

It is often represented by an insurance policy, wherein the insured gets **financial protection** from the insurer against losses due to the occurrence of **any event which is not under the control of the insured**.

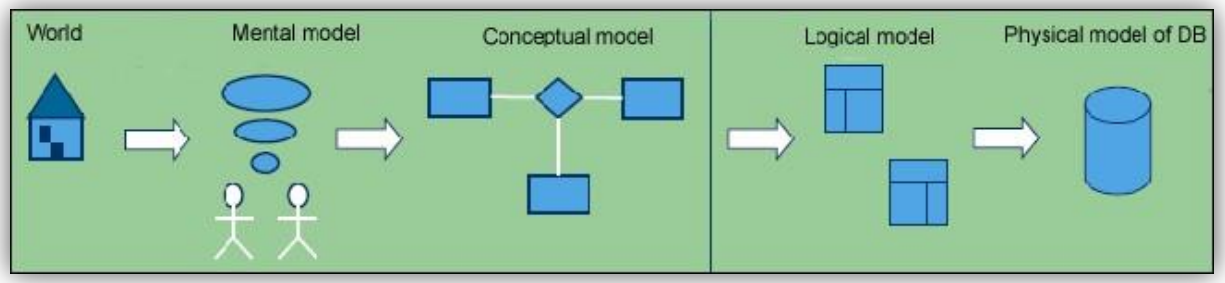
Car Insurance

Vehicle insurance covers cars, motorcycles, trucks, and all the other vehicles running on the road. This insurance is meant for giving protection against any physical damage or bodily injury that the vehicle suffers from recklessness or an accident. All the cost incurred to repair the vehicle is met by the insurance company.

Data Base Modelling

Data modeling is an activity that makes the physical world digital and then stores it in a database. As seen in Figure 1, the process of how data from the real world is processed or represented in a database model is shown.

Figure 1: Process of database building



To provide a comprehensive understanding and overview of the data model, it is common to present it graphically in entity-relationship diagrams (ERDs). The data model defines the structure of data entities and their relationships. For instance, in our vehicle insurance case, entities include Customers, Insurance Policies, and Employees, with the customer having various attributes such as customer number, address, status, and policy number.

To differentiate between various stages of organization, qualifiers are utilized. For example, the conceptual model emphasizes the entities, their relationships, and the characteristics embedded in the problem, making it ideal for communication with stakeholders.

The logical model is a step towards data management technology, specifically relation databases, and is subject to normalization.
The physical model is an implementation of data entities and incorporates optimizations such as partitioning or merging entities, duplicating data, and creating identification keys and indexes.

Figure 2 displays all three models - conceptual, logical, and physical - for a better understanding of what they look like, although the physical data model is not typically viewed as an architectural model.

Figure 2: Graphical presentation of Conceptual model (Fig.2), Logical model (Fig.3), and Physical model (Fig.4)

Figure 2: Conceptual Data Model – First Draft

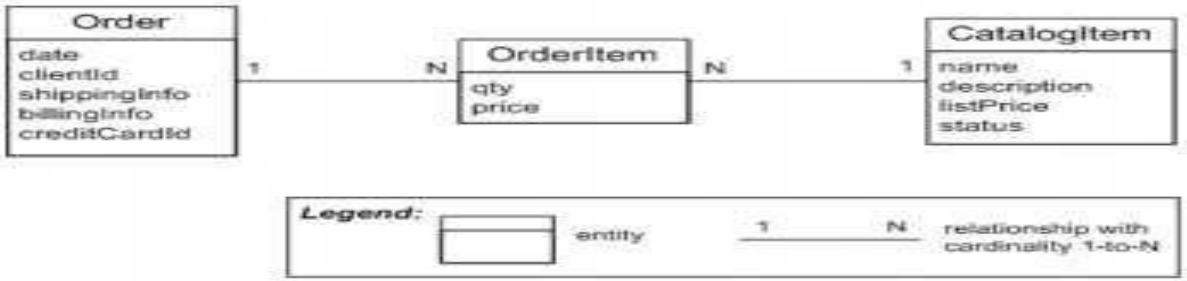


Figure 3: Logical Data Model

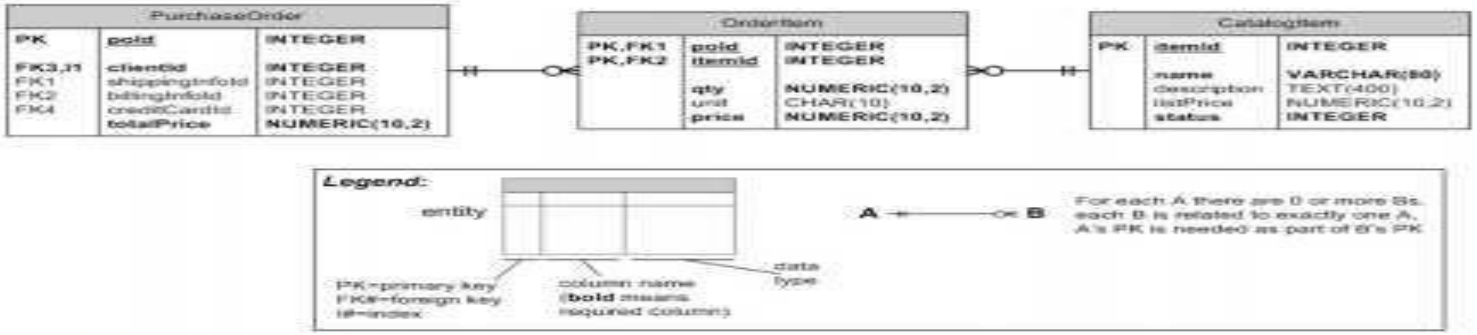


Figure 4: Physical Data Model

Part A: Conceptual Data Model

Entity Types

All the entity types that we feel are relevant to our CDM are illustrated in Table 4, below.

Table 4: Entity types used in car insurance database system CDM model.

Entity Type	Description
CUSTOMER	Records all the personal details about the customer
APPLICATION	Records details of the insurance coverage requested by the customer
QUOTE	Records details of customer potential cost of the insurance product
INSURANCE POLICY	Records details of the Insurance Agreement
PREMIUM	Records details of customer payments
VEHICLE	Records details of Vehicle model, cost, and registration
CLAIMS	Records details of customer claim in case of an incident
SETTLEMENTS	Records details of settlement made on claims
STAFF	Records details of employees
DEPARTMENT	Records details of the various departments
OFFICE	Records details of different office locations
MEMBERSHIP	Records details of customer membership
SERVICE	Records details of different car services offered
NOK	Records details of the next of kin
TERMS_CONDITIONS	Records all terms and conditions concerning the policy
VEHICLE INSURANCE DEPARTMENT	Records details of vehicle insurance cover
RECEIPT	Records details of Receipt of Premiums
COMPANY	Details of the Insurance organization giving the insurance cover

Relationships in CDM

Applying Relationships to Entities

To apply relationships to our entity types we formed certain assumptions to simplify and determine connections between entity types. These assumptions and explanations are illustrated in Table 5 below.

Table 5: CDM relationship of entities for car insurance database

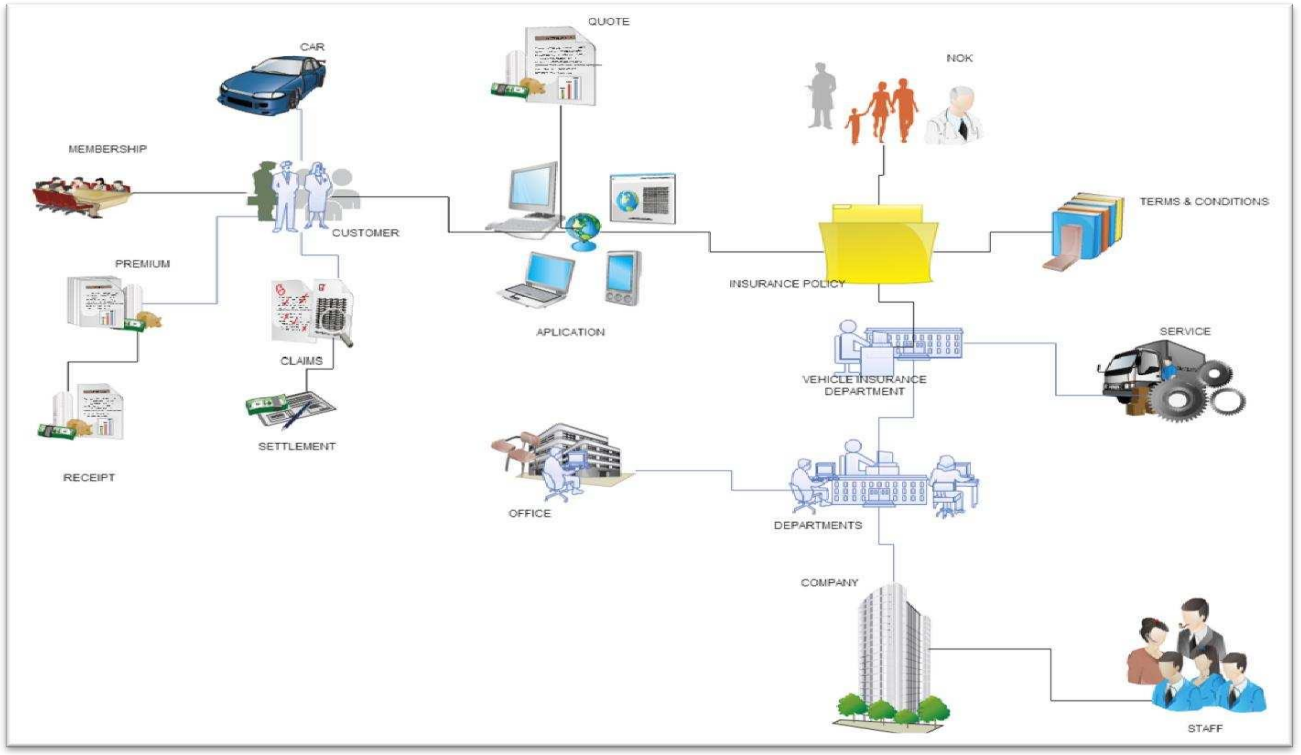
Entity type	Related To Entities	Relationship
QUOTE	APPLICATION	**one to one**
APPLICATION	INSURANCE POLICY CUSTOMER	**one to many** **one to many**
CUSTOMER	MEMBERSHIP PREMIUM CLAIMS VEHICLE	**many to many** **one to many** **one to many** **one-to-one, one to many**
INSURANCE POLICY	VEHICLE INSURANCE DEPARTMENT TERM AND CONDITION NOK	**one to many** **many to many** **one to many**
PREMIUM	RECEIPT	**one to many**
CLAIMS	SETTLEMENT	**one to one**
VEHICLE INSURANCE DEPARTMENT	DEPARTMENT SERVICE	**One-to-one, one to many** **one to many**

DEPARTMENT	OFFICE COMPANY	**many to many** **one to many**
COMPANY	STAFF	**many to many**

Graphical presentation of CDM

The Conceptual Data Model that will be used as a starting point in designing our online car insurance database system can be seen in Figure 3 (with no entities relationships) and Figure 4 (with entities relationships), done in PostgreSQL software.

Figure 3: Conceptual Data Model for Car insurance - character graphic presentation (no entities relationships)



Part B: Logical Data Model

Introduction of terms used for constructing LDM.

To build the LDM, we used different terms as described below:

Elements

Elements in the data model are named entities. This is any distinguishable object that presents part of the database. It can be related to any object in the real world

such as a car, a customer (person), a policy, a company, etc. with respective attributes that are relevant to the software system.

Relations – relations are simply tables.

Relationship - Designates logical association between entities, with the cardinality of the participant entities: one-to-one, one-to-many, or many-to-many relationships.

Constraints

The database normalization technique is used to impose restrictions on a data model that is based on dependencies between entities and their attributes. Normalization is used with the goal objective to avoid duplication of information to safeguard the consistency (integrity) of the data.

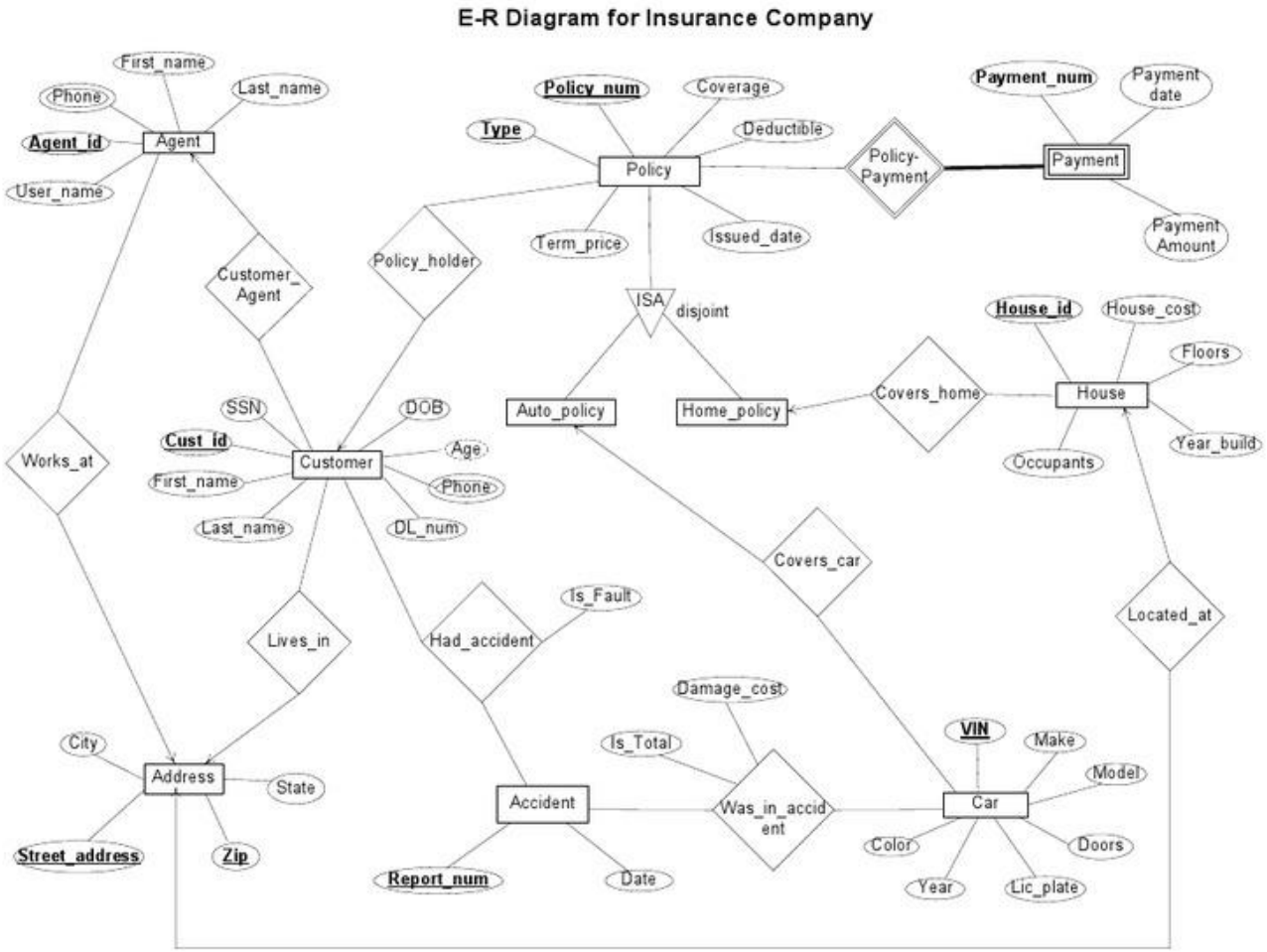
Data types

When we assign attributes to entities with primary keys and foreign keys to do the normalization, we identify each attribute with data type for each data management system – SQL examples below.

Table 6: Example of data types in Access and Oracle DMS

Entity type	Attributes	Data type Access	Data type SQL
CUSTOMER	FIRST_NAME	Text	varchar(size)
	LAST_NAME	Text	varchar(size)
	DATE_OF_BIRTH	Date/Time	date

ER Diagram



Business Rules, Attributes, Data types, and Primary/Foreign Keys

In this section of the report, we outline the attributes, data types, and primary and foreign keys for our Logical Data Model (LDM) system. To provide a clear overview, we link each table number to its corresponding entity type, and we provide the business rule for what we aim to capture with the entity type, which is presented in Table 7 below.

Table 7: Table number entity type

Table Number	Entity Type	Business Rules
1	CUSTOMER	Records all the personal details about the customer
2	APPLICATION	Records details of the insurance coverage requested by the customer
3	QUOTE	Records details of customer potential cost of the insurance product
4	INSURANCE_POLICY	Records details of the Insurance Agreement
5	PREMIUM_PAYMENT	Records details of customer cost of payments
6	VEHICLE	Records details of Vehicle model, cost, and registration
7	CLAIM	Records details of customer claim in case of an incident
8	CLAIM_SETTLEMENT	Records details of settlement made on claims
9	STAFF	Records details of employees

10	DEPARTMENT	Records details of the various departments
11	OFFICE	Records details of different office locations
12	MEMBERSHIP	Records details of customer membership, clubs, societies
13	VEHICLE_SERVICE	Records details of different vehicle services offered
14	NOK	Records details of the next of kin
15	INSURANCE_COMPANY	Details of the Insurance organization giving the insurance cover
16	POLICY_RENEWABLE	Records details of the due date of the insurance policy
17	INCIDENT	Records details of the accident, theft, fire, etc.
18	INCIDENT_REPORT	Records details of the individual incident
19	COVERAGE	Records all terms and conditions regarding the policy
20	PRODUCT	Records details of the products offered by the insurance company
21	RECEIPT	Details of premium payments to the customer
22	INSURANCE_POLICY_COVERAGE	It shows agreement and coverage details

The CUSTOMER attributes record all the essential personal details of the customer. The CUST_ID is the unique primary key.

The APPLICATION attributes record all the essential application details of the customer. The APPLICATION_ID is the unique primary key and the CUST_ID is a foreign key linking the table back to the entity type CUSTOMER.

The QUOTE attributes record all the essential quotation details of the customer. The QUOTE_ID is the unique primary key and APPLICATION_ID and CUST_ID are foreign keys linking the table back to the respective entities.

The INSURANCE POLICY attributes record all the essential policy details of the customer. The AGREEMENT_ID is the unique primary key and the CUST_ID, and APPLICATION_ID is linked to the other corresponding entities through their foreign keys.

The PREMIUM_PAYMENT attributes record all the essential policy premium payment details of the customer. The PREM_PAYMENT_ID is the unique primary key and the CUST_ID is the Foreign key.

The VEHICLE attributes record all the essential VEHICLE details belonging to the customer. The VEHICLE_ID is the unique primary key and the CUST_ID is the foreign key linking the table to the CUSTOMER entity.

The CLAIM attributes record all the essential CLAIM details of the customer in case of an incident. The CLAIM_ID is the unique primary key and the CUST_ID is the foreign key linking the table to the CUSTOMER entity.

The CLAIM_SETTLEMENT attributes record all the essential claim settlement details of the customer after an incident. The CLAIM_SETTLEMENT_ID is the unique primary key and the CUST_ID and CLAIM_ID are the foreign keys that link the table to the corresponding entity.

The STAFF attributes record all the essential staff details working in the insurance company. The STAFF_ID is the unique primary key and the COMPANY_NAME is a foreign key linking the table back to the entity type COMPANY.

The DEPARTMENT attributes record all the essential company department details within the insurance company. The DEPARTMENT_NAME is the unique primary key and the COMPANY_NAME is a foreign key linking the table back to the entity type COMPANY.

The OFFICE attributes record all the essential office details within the insurance company. The OFFICE_NAME is the unique primary key and the DEPARTMENT_NAME and COMPANY_NAME are foreign keys linking the table back to the respective entity types.

The MEMBERSHIP attributes record all the essential membership details available for the customer. The MEMBERSHIP_ID is the unique primary key and the CUST_ID is a foreign key linking the table back to the entity type CUSTOMER.

The VEHICLE_SERVICE attributes record all the essential vehicle services offered to insured customer details. The VEHICLE_SERVICE is the unique primary key and the CUST_ID and VEHICLE_ID are linked to the other corresponding entities.

The NOK attributes record information on the next of kin details. NOK_ID is the unique primary key here. AGREEMENT_ID, APPLICATION_ID, and CUST_ID are foreign keys linking back information to their respective entities.

The INSURANCE COMPANY attributes record all the essential company details of the customer. The COMPANY_ID is the unique primary key

The POLICY RENEWABLE attributes record all the essential policy renewal details of the insured customer. The POLICY_RENEWABLE_ID is the unique primary key and the AGREEMENT_ID, APPLICATION_ID, and CUST_ID are foreign keys linking the table back to the respective entities.

The INCIDENT attributes record all the essential incident details such as Accident and theft on the insured customer vehicle. The INCIDENT_ID is the unique primary key.

The INCIDENT_REPORT_ID attributes record all the essential incident occurrences on the customer vehicle. The INCIDENT_REPORT_ID is the unique primary key and the CUST_ID, AND INCIDENT_ID are foreign keys linking the table back to their respective entity types.

The COVERAGE attributes record all the essential coverage details of the insurance policy to the customer. The COVERAGE_ID is the unique primary key and the COMPANY_NAME is a foreign key linking the table back to the entity type COMPANY.

The PRODUCT attributes record all the essential company product details offered by the Insurance company. The PRODUCT_NUMBER is the unique primary key and COMPANY_NAME is the foreign key linking the table to the COMPANY entity.

The RECEIPT attributes record all the essential payments done by CUSTOMERS to the Insurance company. The RECEIPT_ID is the unique primary key and PREMIUM_PAYMENT_ID and CUST_ID are Foreign keys linking the table to their respective entities.

The INSURANCE_POLICY_COVERAGE records details of the Vehicle policy that entails the Terms and Conditions of the Contract. AGREEMENT_ID is the unique primary key and COVERAGE_ID is the foreign key linking this table to COVERAGE Entity.

Who has access to certain parts of the database?

A CRUD table displays the various functions within an application, which involve SQL statements that impact different sections of a database. This type of table provides an excellent way to illustrate the interactions that take place between users and tables within the database.

By analyzing potential user scenarios, we can determine which tables are frequently used and which are not. Additionally, this approach enables us to identify potential bottlenecks that may affect system performance, which is crucial for assessing the overall burden placed on the database.

Using four SQL statements:

- Create – INSERT – to store new data.
- Read – SELECT – to retrieve data.
- Update – UPDATE – to change or modify data.
- Delete – DELETE – delete or remove data.

Table 11: CRUD Matrix of database (Create - C, Read-R, Update-U, and Delete-D)

MODULES / ENTITIES	Customer	Manger of insurance	Insurance agent	Accountant	HR department	Damage inspector	Database administrator	Finance department	
CUSTOMER	CR	R	CRUD	R	-	-	R	-	
APPLICATION	R	R	CRD	R	R	-	CRD	R	
QUOTE	R	R	CRUD	R	R	-	R	R	
INSURANCE_POLICY	R	R	CRUD	R	-	R	CRD	R	
PREMIUM_PAYMENT	-	-	CRUD	CRUD	-	-	RU	CRUD	
VEHICLE	-	R	CRUD	-	-	-	RU	-	
CLAIM	-	R	CRUD	CRD	-	-	RU	CRUD	
CLAIM_SETTLEMENT	R	CRUD	CR	-	-	-	RU	CRUD	
STAFF	-	CR	R	R	CRUD	-	RU	-	
DEPARTMENT	-	R	R	-	CRUD	-	RU	R	
OFFICE	R	R	R	R	CRUD	-	RU	R	
MEMBERSHIP	CR	R	CRD	-	-	-	RU	R	
VEHICLE_SERVICE	R	CRD	CRU	-	-	-	RU	R	
NOK	R	R	CRUD	-	-	-	RU	R	
INSURANCE_COMPANY	R	R	R	R	R	-	RU	R	
POLICY_RENEWABLE	R	CRUD	CRU	-	-	-	RU	-	
INCIDENT	-	CRD	R	-	-	R	RU	R	
INCIDENT_REPORT	R	CRD	R	R	-	CRUD	RU	R	
COVERAGE	R	R	CRD	R	-	-	RU	R	
PRODUCT	R	CRUD	R	R	R	R	RU	R	
RECEIPT	R	CRUD	CRD	CRUD	-	-	RU	CRUD	

Project Requirements

The Insurance Management System project has several requirements that the system must meet. The project's primary objective is to automate manual calculations and provide easy retrieval of stored data. The system must provide secure and user-friendly access to the database for managers, agents, and customers. The system should also provide accurate calculations of commissions and dues based on agents' performance. Additionally, the system must provide descriptive reports to help management make informed decisions. Other requirements include the ability to manage customer details, policy information, and transactions accurately.

Drawbacks of the existing system

The existing system has no security measures against logging in and no checks are made for authorized users.

The end-user has to remember a lot of commands to make efficient use of the system.

The system does not have any descriptive report and thus does not help management in decision-making.

Proposed system

The proposed system is to computerize the agent and client department for various purposes.

The main objective is to automate manual calculations and easy retrieval of the stored data.

1. The database could include the following tables

1. Customer Info, with fields for ID, Last Name, First Name, Address, City, State, Zip, Phone, Credit Card Number, Exp Date, Multi Policy Discount, DOB.

2. Auto, with fields for Auto Policy Number, ID, Make, Model, Year, Liability Amount, UM, UIM, Med Pay, Premium Amount, Collision Damage Amount, Named Insured, Additional Driver.

3. Homeowners, with fields for Homeowners Policy Number, ID, Liability Amount, Property Damage Amount, Premium.

4. Renters, with fields for Renters Policy Number, ID, Liability Amount, Property Damage Amount, Premium.

5. Life Ins, with fields for Life Ins Policy Number, ID, Premium amount, Benefit Amount, Beneficiary.

The Customer Information table would contain basic customer contacts, some biographical information, and payment information.

The Auto table would contain information related to a customer's automobile policy.

Also needed would be transaction tables for transaction-related data:

1. Payments, with fields for Payment Transaction ID, ID, Due Date, Amount Due, Amount Paid, and Paid on Time. 2. Claims, with fields for Claim Transaction ID, ID, Amount of Claim, and Claim Approved. The database schema should avoid data redundancy, maintain data integrity, and organize data in a logical manner.

System Implementation

The Insurance Management System project is implemented using web-based technologies. The system's front end is designed using HTML, CSS, and JavaScript, while the back end is designed using PHP and MySQL. The system's user interface is designed to be simple and user-friendly, enabling managers, agents, and customers to access the system easily.

System Testing

The Insurance Management System project undergoes various testing stages to ensure that it meets the project's requirements. The testing stages include unit testing, integration testing, and system testing. Unit testing involves testing individual components of the system, while integration testing involves testing the system's integration with other systems. System testing involves testing the system as a whole to ensure that it meets the project's requirements. The system's testing phase is critical to ensure that the system is reliable, efficient, and meets the project's objectives.

Conclusion

As a group, we collaborated on a project to develop a comprehensive database for an online vehicle insurance company using SQL code. Our objective was to create a functional DB that includes all documentation and reports.

Throughout the project, we encountered various challenges, both significant and minor, but we ultimately succeeded in creating a functional DB. We began by constructing a conceptual data model (CDM), followed by a logical data model (LDM), and ultimately a physical data model (PDM), all utilizing PostgreSQL. We then utilized the PDM to generate code that can be executed in a SQL database management system (DBMS).

To enhance comprehension for readers and improve our learning, we incorporated theoretical elements into each phase and recorded our progress and work completed in the project's initial document (PID) and accompanying reports.