

CS310
DBMS
Group Assignment

VEHICLE INSURANCE COMPANY

BY

1)Kuluru Vineeth Kumar Reddy (18BCS043)

2) Karthick P S(18BCS038)

3) K. Laxmi Narayana (18BCS037)

4) M. Venkata Kalyan Babu (18BCS049)

5) MP. Bharath (18BCS057)

6) B. Ragavan (18BCS016)

7) G. Jagan Mohan Reddy (18BCS029)

8) N Trinath Reddy (18BCS059)

9) P. Chandhan (18BCS063)

PART-A:-CONCEPTUAL DATA MODEL

ABOUT A PROJECT :-

Section 1:What is the project all about?

Project title :

A Database for a Vehicle Insurance Company .

Project Definition :

Purpose:

This module provides a comprehensive discussion of, and practical experience in, advanced entity modelling; normalisation; transactional relational database design; SQL and PL/SQL coding; and generation of data backed management reports. Students gain practical experience using contemporary database modelling and design tools and technologies, and apply sound design principles for creating effective decision support solutions for realistic business scenarios.

Part A: Conceptual Data Model

Design Rules:

To design our car insurance database conceptual data model we first needed to decide what characteristics underpin the model under investigation. As a group we decided on various rules that need to be implemented in order for the model to be consistent and precise. Table 1 below illustrates these rules.

Table 1: Design rules for CDM model of car insurance database

| Design Rule | Description | Example |
|---------------|--|---|
| Rule 1 | All the individual entity types must be in capital letters. | CUSTOMER |
| Rule 2 | An underscore is used to label an entity type with more than one word. | TERMS_CONDITIONS |
| Rule 3 | Plurals are not used when labelling entity types. | APPLICATION |
| Rule 4 | No abbreviations are used when labelling entity types. | QUOTE |
| Rule 5 | Entity types must not be in Numerical. | VEHICLE |
| Rule 6 | Every entity must contain a Primary key. | Like CUSTOMER_ID for the customer Entity. |

Assumptions :

To design the Conceptual Data Model (CDM) we have a certain set of assumptions. These assumptions will help shape our model to allow consistency within our design. Table 2 presents the assumptions used in this model.

Table 2: Assumptions used in car insurance database model

| Assumption | Description |
|---------------------|---|
| Assumption 1 | Customers must have original proof Id's like aadhaar, driving license etc.. |
| Assumption 2 | Customer not having personal accidental cases before in on-road |
| Assumption 3 | The online insurance is given to customers over 18 years of age. |

| | |
|---------------------|---|
| Assumption 4 | The online insurance needs some driving history of the customer. |
| Assumption 5 | The online insurance needs to know the type of car the customer drives. |
| Assumption 6 | The online insurance needs to know about the insurance history of the customer. |

Entity Types :

All of the entity types that we feel are relevant in our CDM are illustrated in Table 3, below.

Table 3: Entity types used in car insurance database system CDM model.

| Entity Type | Description |
|-------------------------------------|--|
| T4_CUSTOMER | Records all the personal details about the customer. |
| T4_APPLICATION | Records details of the insurance coverage requested by Customer |
| T4_QUOTE | Records details of customer potential cost of the insurance product. |
| T4_INSURANCE_POLICY_COVERAGE | Records details of the insurance agreement. |
| T4_PREMIUM_PAYMENT | Records details of customer cost and payments. |
| T4_CLAIM | Records details of customer claims in case of an accident. |
| T4_CLAIM_SETTLEMENT | Records details of settlement made on claims. |
| T4_STAFF | Records details of employees. |
| T4_DEPARTMENT | Records details of the various departments. |

| | |
|-----------------------------|---|
| T4_OFFICE | Records details of different office locations. |
| T4_MEMBERSHIP | Records details of customer membership,clubs and societies. |
| T4_VEHICLE_SERVICE | Records details of different car services offered. |
| T4_NOK | Records details of the next o kin. |
| T4_COVERAGE | Records all terms and conditions in regard to the policy |
| T4_INSURANCE_POLICY | Records details of Insurance agreement. |
| T4_PRODUCT | Records details of the products offered by insurance company |
| T4_RECEIPT | Details of premium payments to customer |
| T4_INSURANCE_COMPANY | Details of the insurance organization giving the insurance cover. |
| T4_VEHICLE | Records details of Vehicle model, cost and registration. |
| T4_INCIDENT | Records details of the accident, theft, fire, etc. |
| T4_POLICY_RENEWABLE | Records details of due date of insurance policy. |
| T4_INCIDENT_REPORT | Records details of the individual incident |

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 4 below.

Table 4: CDM relationship of entities for car insurance database :

| Entity Type | Related To Entities | Relationship |
|---------------------------------------|---|---|
| T4_QUOTE | T4_APPLICATION | One to one |
| T4_APPLICATION | T4_INSURANCE_POLICY T4_CUSTOMER | One to many One to many |
| T4_CUSTOMER | T4_MEMBERSHIP T4_PREMIUM_PAYMENT T4_CLAIM T4_VEHICLE | Many to many One to many One to many One to one, one to many |
| T4_INSURANCE_POLICY | T4_DEPARTMENT T4_NOK | One to many One to many |
| T4_PREMIUM_PAYMENT | T4_RECEIPT | One to many |
| T4_CLAIM | T4_CLAIM_SETTLEMENT | One to one |
| T4_VEHICLE INSURANCE T4_DEPARTMENT | T4_DEPARTMENT T4_SERVICE | One to one, one to many One to many |
| T4_DEPARTMENT | T4_OFFICE T4_INSURANCE_COMPANY | Many to many One to many |
| T4_INSURANCE_COMPANY | T4_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 ERwin software.

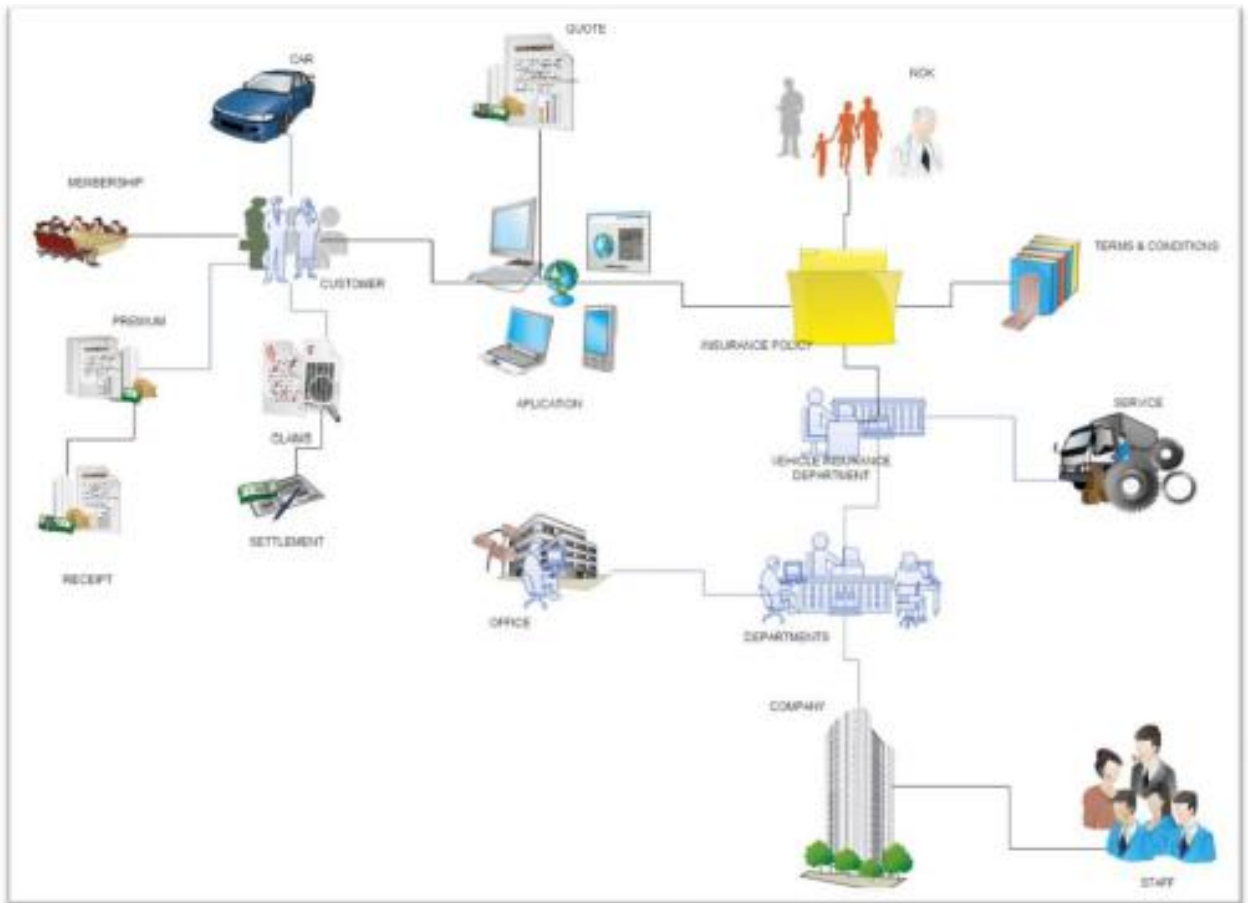
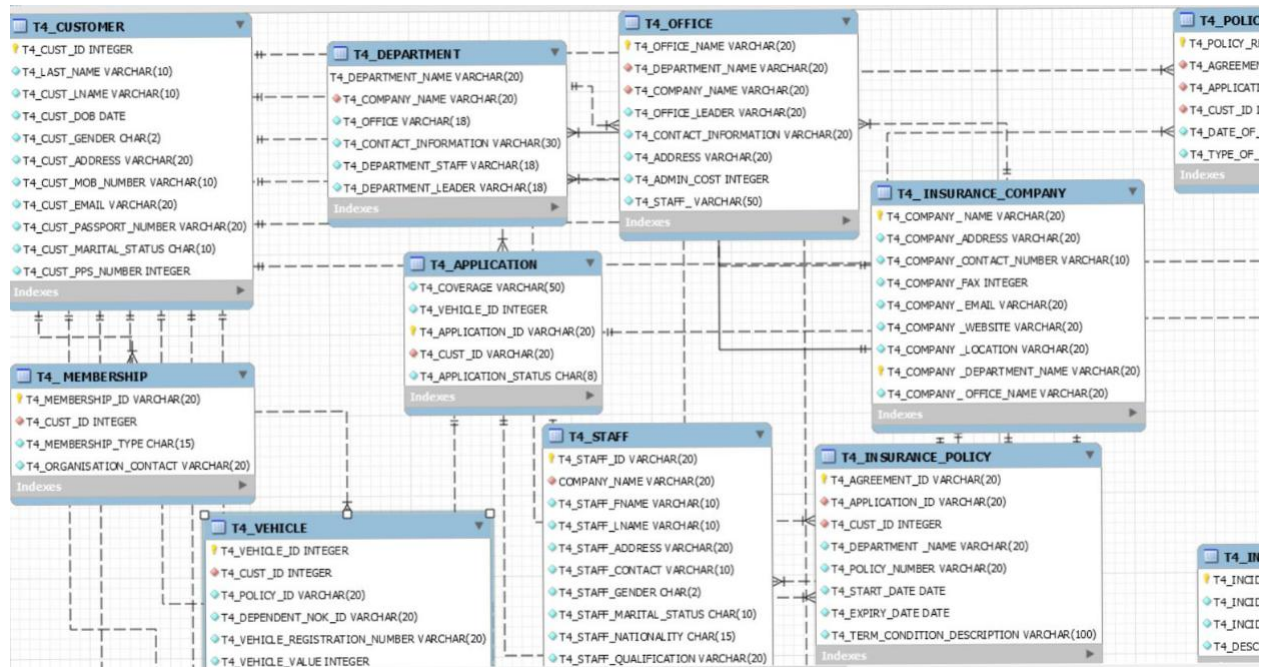
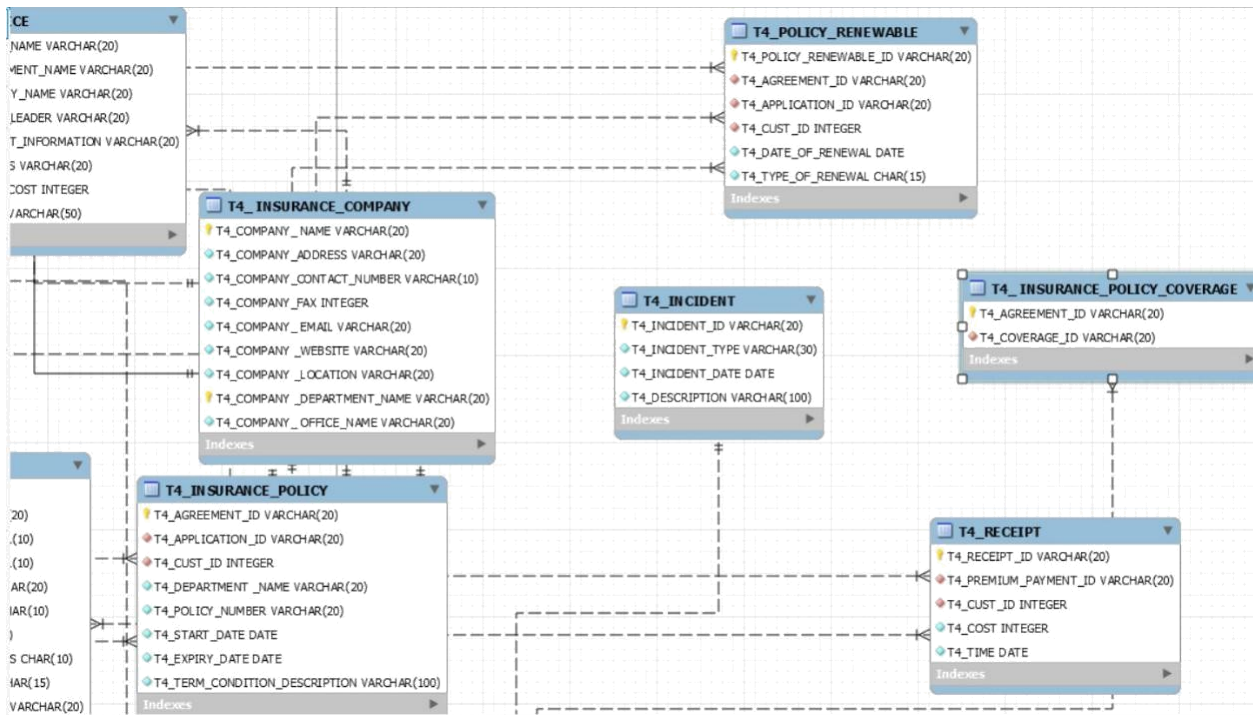
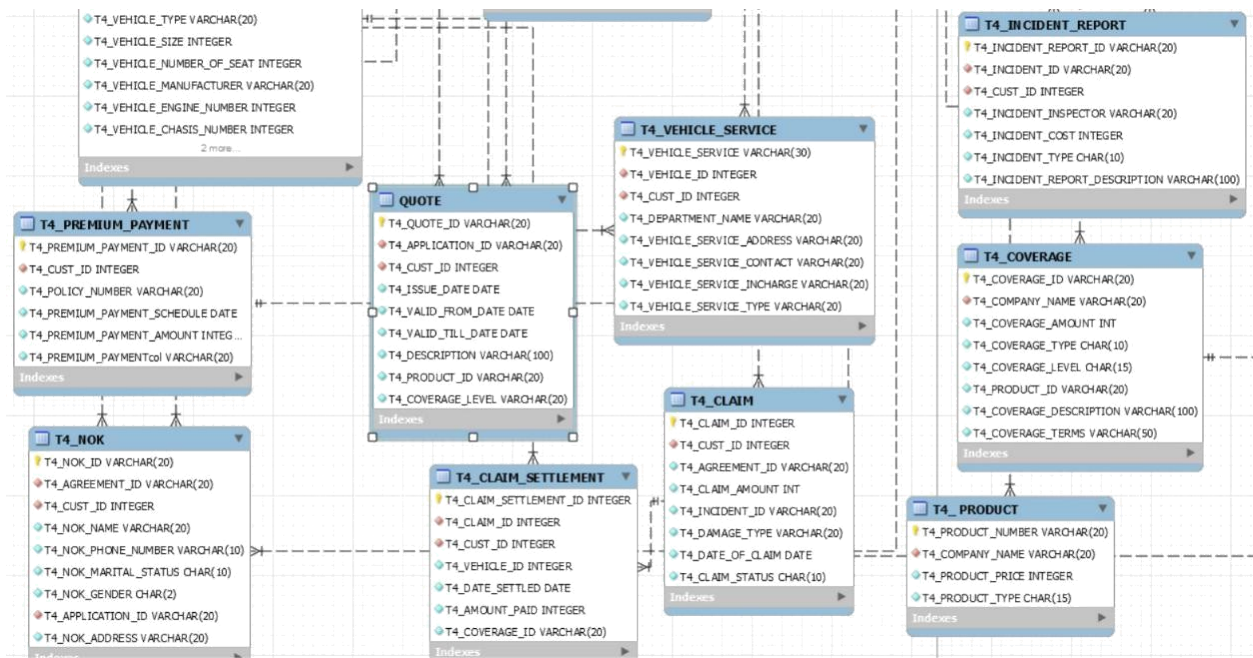


Figure 4: Conceptual data model of Car insurance(with entity types) :





PART-B:-LOGICAL DATA MODEL

Introduction

Part B of this report includes design of logical data model (LDM) for vehicle insurance company XYZ Ltd.

First we had to make some changes and improvements to the conceptual data model which are explained and presented in the last part of this report. Then we identify all the attributes in old and new entities and assign them the primary (PK) and foreign keys (FK) and make relationships with them in Erwin to make a full LDM. We identified each variable and the data types that this LDM could be used to design the database in Access or Oracle database management systems.

Introduction of terms used for constructing LDM

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.

Properties of entities can have values:

- Name
- Description of the meaning and significance.
- Weather entity is dependent or non-dependent.
- List of attributes (Car entity: year, manufacturer, model, mileage, owner, licence, book of maintenance) with properties (data type, size, is it required or not).
- The attributes (or attribute) are used to precisely identify an entity (primary key – PK, foreigner key – FK, ...).
- Constraints of individual or combined attributes values (e.g. date of issue of new policy can't be prior to renewal date of policy).
- Rules to grant permission to users or user groups to access the entity.
- Expected number of entity instances and expected growth rate.

Or additional:

- List of attributes to be indexed to optimize access time.
- List of attributes to be encrypted or compressed.
- Weather entity should become a database view or a table.
- Weather entity should become a materialized view.
- List of database triggers to be implemented for that entity.

Relations

Relationship- Designates logical association between entities, with cardinality of the participant

entities: one-to-one, one-to-many, or many-to-many relationships. Relationships can be identifying or non-identifying (identifying A-B; existence of B depends on existence of A).

Generalization/specialization– Indicates an “is a” relationship between entities. For example ,department entity is a generalization of different types of departments; at the same time vehicle Insurance department or travel insurance department is specialization of department entity.

Aggregation- is an abstraction that turns relationship between entities into an aggregate entity, rarely used. Example: “customer-insurance advisor - date” can be an aggregate entity called Appointment.

Constraints

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

Data types

When we assign attributes to entities with primary keys and foreign keys do the normalization, we identify each attribute with data type for each data management system – Access as seen in example Table 6, below.

Table 6: Example of data types in Access and DMS

| Entity type | Attributes | Data type |
|-------------|------------------|-------------|
| T4_CUSTOMER | T4_FIRST_NAME | Varchar(10) |
| | T4_LAST_NAME | Varchar(10) |
| | T4_DATE_OF_BIRTH | Date |

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

This section of the report identifies all of the attributes, data types and primary and foreign keys for our system LDM. For better overview we present the table number to the corresponding entity type, followed by the business rule of what we wish the entity type to capture, posted below in Table 7.

Table 7: Table number entity type

| Table Number | Entity Type | Business Rules |
|--------------|---------------------|--|
| 1 | T4_CUSTOMER | Records all the personal details about the customer |
| 2 | T4_APPLICATION | Records details of the insurance cover requested by the customer. |
| 3 | T4_QUOTE | Records details of customer potential cost of the insurance product. |
| 4 | T4_INSURANCE_POLICY | Records details of Insurance agreement. |

| | | |
|-----------|------------------------------|--|
| 5 | T4_PREMIUM_PAYMENT | Records details of customer cost of payments. |
| 6 | T4_VEHICLE | Records details of Vehicle model, cost and registration. |
| 7 | T4_CLAIM | Records details of customer claims in case of an incident. |
| 8 | T4_CLAIM SETTLEMENT | Records details of settlement made on claims |
| 9 | T4_STAFF | Records details of employees |
| 10 | T4_DEPARTMENT | Records details of the various departments |
| 11 | T4_OFFICE | Records details of different office locations |
| 12 | T4_MEMBERSHIP | Records details of customer membership, clubs, societies. |
| 13 | T4_VEHICLE_SERVICE | Records details of different vehicle services offered |
| 14 | T4_NOK | Records details of the next of kin |
| 15 | T4_INSURANCE_COMPANY | Details of the Insurance organization giving the insurance cover |
| 16 | T4_POLICY_RENEWABLE | Records details of due date of insurance policy |
| 17 | T4_INCIDENT | Records details of the accident, theft, fire, etc. |
| 18 | T4_INCIDENT_REPORT | Records details of the individual incident |
| 19 | T4_COVERAGE | Records all terms and conditions in regard to the policy |
| 20 | T4_PRODUCT | Records details of the products offered by insurance company |
| 21 | T4_RECEIPT | Details of premium payments to customer |
| 22 | T4_INSURANCE_POLICY_COVERAGE | It shows agreement and coverage details |

Table 8: Abbreviation table of attributes manes used in LDM.

| LDM Attributes | Column Name Abbreviations |
|-----------------------|----------------------------------|
| T4_CUST_ID | CUSTOMER_IDENTIFICATION |
| T4_CUST_FNAME | CUSTOMER_FNAME |
| T4_CUST_LNAME | CUSTOMER_LNAME |
| T4_CUST_DOB | CUSTOMER_DATEOFBIRTH |
| T4_CUST_PPS_NUMBER | CUSTOMER_PERSONALPUBLICNUMBER |
| T4_STAFF_FNAME | STAFF_FNAME |
| T4_STAFF_LNAME | STAFF_LNAME |
| T4_STAFF_PPS_NUMBER | STAFF_PERSONALPUBLICNUMBER |
| T4_ADMIN_COST | ADMINISTRATION_COST |
| T4_NOK_ID | NEXTOFKIN_IDENTIFICATION |

Table-LDM 1: T4_CUSTOMER

| Attributes | Data type | Primary Keys and Foreign keys |
|-------------------------|-------------|-------------------------------|
| T4_CUST_ID | INT | PK |
| T4_CUST_FNAME | VARCHAR(10) | |
| T4_CUST_LNAME | VARCHAR(10) | |
| T4_CUST_DOB | DATE | |
| T4_CUST_GENDER | CHAR(2) | |
| T4_CUST_ADDRESS | VARCHAR(20) | |
| T4_CUST_MOB_NUMBER | VARCHAR(10) | |
| T4_CUST_EMAIL | VARCHAR(20) | |
| T4_CUST_PASSPORT_NUMBER | VARCHAR(20) | |
| T4_CUST_MARITAL_STATUS | CHAR(10) | |
| T4_CUST_PPS_NUMBER | INT | |

EXPLANATION:-

The T4_ CUSTOMER attributes record all the essential personal details of the customer. The T4_CUST_ID is the unique primary key.

Table-LDM 2: T4_APPLICATION

| Attributes | Data type | Primary Keys and Foreign keys |
|-----------------------|-------------|-------------------------------|
| T4_APPLICATION_ID | VARCHAR(20) | PK |
| T4_CUST_ID | INT | FK |
| T4_VEHICLE_ID | INT | |
| T4_APPLICATION_STATUS | CHAR(8) | |
| T4_COVERAGE | VARCHAR(50) | |

EXPLANATION:-

The T4_APPLICATION attributes record all the essential application details of the customer. The T4_APPLICATION_ID is the unique primary key and the T4_CUST_ID is a foreign key linking the table back to the entity type T4_CUSTOMER.

Table-LDM 3: T4_QUOTE

| Attributes | DATA TYPE | PRIMARY KEYS AND FOREIGN KEYS |
|--------------------|--------------|-------------------------------|
| T4_QUOTE_ID | VARCHAR(20) | PK |
| T4_APPLICATION_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_ISSUE_DATE | DATE | |
| T4_VALID_FROM_DATE | DATE | |
| T4_VALID_TILL_DATE | DATE | |
| T4_DESCRIPTION | VARCHAR(100) | |
| T4_PRODUCT_ID | VARCHAR(20) | |
| T4_COVERAGE_LEVEL | VARCHAR(20) | |

EXPLANATION:-

The T4_QUOTE attributes record all the essential quotation details of the customer. The T4_QUOTE_ID is the unique primary key and T4_APPLICATION_ID and T4_CUST_ID is a foreign key linking the table back to the respective entities

Table-LDM 4: T4_INSURANCE_POLICY

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|--------------------------------|--------------|-------------------------------------|
| T4_AGREEMENT_ID | VARCHAR(20) | PK |
| T4_APPLICATION_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_DEPARTMENT_NAME | VARCHAR(20) | |
| T4_POLICY_NUMBER | VARCHAR(20) | |
| T4_START_DATE | DATE | |
| T4_EXPIRY_DATE | DATE | |
| T4_TERMS_CONDITION_DESCRIPTION | VARCHAR(100) | |

EXPLANATION:-

The T4_INSURANCE POLICY attributes record all the essential policy details of the customer. The T4_AGREEMENT_ID is the unique primary key and the T4_CUST_ID, and T4_APPLICATION_ID are linked to the other corresponding entities through their foreign keys.

Table-LDM 5: T4_PREMIUM_PAYMENT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-----------------------------|-------------|-------------------------------------|
| T4_PREMIUM_PAYMENT_ID | VARCHAR(20) | PK |
| T4_CUST_ID | INT | FK |
| T4_PREMIUM_PAYMENT_AMOUNT | INT | |
| T4_PREMIUM_PAYMENT_SCHEDULE | DATE | |
| T4_RECEIPT_ID | VARCHAR(20) | |
| T4_POLICY_NUMBER | VARCHAR(20) | |

EXPLANATION:-

The T4_PREMIUM_PAYMENT attributes record all the essential policy premium payments details of the customer. The T4_PREMIUM_PAYMENT_ID is the unique primary key and the T4_CUST_ID is the Foreign key linking tables to T4_CUSTOMER entities.

Table-LDM 6: T4_VEHICLE

| ATTRIBUTES | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|--------------------------------|-------------|-------------------------------|
| T4_VEHICLE_ID | INT | PK |
| T4_CUST_ID | INT | FK |
| T4_POLICY_ID | VARCHAR(20) | |
| T4_DEPENDENT_NOK_ID | VARCHAR(20) | |
| T4_VEHICLE_REGISTRATION_NUMBER | VARCHAR(20) | |
| T4_VEHICLE_VALUE | INT | |
| T4_VEHICLE_TYPE | VARCHAR(20) | |
| T4_VEHICLE_SIZE | INT | |
| T4_VEHICLE_NUMBER_OF_SEAT | INT | |
| T4_VEHICLE_MANUFACTURER | VARCHAR(20) | |
| T4_VEHICLE_ENGINE_NUMBER | INT | |
| T4_VEHICLE_CHASIS_NUMBER | INT | |
| T4_VEHICLE_NUMBER | VARCHAR(20) | |
| T4_VEHICLE_MODEL_NUMBER | VARCHAR(20) | |

EXPLANATION:-

The T4_VEHICLE attributes record all the essential T4_VEHICLE details belonging to the customer. The T4_VEHICLE_ID is the unique primary key and the T4_CUST_ID is the foreign key linking table to T4_CUSTOMER entity.

Table-LDM 7: T4_CLAIM

| ATTRIBUTES | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|------------------|-------------|-------------------------------|
| T4_CLAIM_ID | INT | PK |
| T4_CUST_ID | INT | FK |
| T4_AGREEMENT_ID | VARCHAR(20) | |
| T4_CLAIM_AMOUNT | INT | |
| T4_INCIDENT_ID | VARCHAR(20) | |
| T4_DAMAGE_TYPE | VARCHAR(20) | |
| T4_DATE_OF CLAIM | DATE | |
| T4_CLAIM_STATUS | CHAR(10) | |

EXPLANATION:-

The T4_CLAIM attributes record all the essential T4_CLAIM details of the customer in case of an incident. The T4_CLAIM_ID is the unique primary key and the T4_CUST_ID is foreign key linking table to T4_CUSTOMER entity.

Table-LDM 8: T4_CLAIM_SETTLEMENT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|------------------------|-------------|-------------------------------------|
| T4_CLAIM_SETTLEMENT_ID | INT | PK |
| T4_CUST_ID | INT | FK |
| T4_VEHICLE_ID | INT | |
| T4_DATE_SETTLED | DATE | |
| T4_AMOUNT_PAID | INT | |
| T4_COVERAGE_ID | VARCHAR(20) | |
| T4_CLAIM_ID | INT | FK |

EXPLANATION:-

The T4_CLAIM_SETTLEMENT attributes record all the essential claim settlement details of the customer after an incident. The T4_CLAIM_SETTLEMENT_ID is the unique primary key and the T4_CUST_ID and T4_CLAIM_ID are the foreign keys that link the table to the corresponding entity.

Table-LDM 9: T4_STAFF

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-------------------------|-------------|-------------------------------|
| T4_STAFF_ID | VARCHAR(20) | PK |
| T4_COMPANY_NAME | VARCHAR(20) | FK |
| T4_STAFF_FNAME | VARCHAR(10) | |
| T4_STAFF_LNAME | VARCHAR(10) | |
| T4_STAFF_ADDRESS | VARCHAR(20) | |
| T4_STAFF_CONTACT | VARCHAR(10) | |
| T4_STAFF_GENDER | CHAR(2) | |
| T4_STAFF_MARITAL_STATUS | CHAR(10) | |
| T4_STAFF_NATIONALITY | CHAR(15) | |
| T4_STAFF_QUALIFICATION | VARCHAR(20) | |
| T4_STAFF_ALLOWANCE | INT | |
| T4_STAFF_PPS_NUMBER | INT | |

EXPLANATION:-

The T4_STAFF attributes record all the essential staff details working in the insurance company. The T4_STAFF_ID is the unique primary key and the T4_COMPANY_NAME is a foreign key linking the table back to the entity type T4_COMPANY.

Table-LDM 10: T4_DEPARTMENT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|------------------------|-------------|-------------------------------|
| T4_DEPARTMENT_NAME | VARCHAR(20) | FK,PK |
| T4_COMPANY_NAME | VARCHAR(20) | FK |
| T4_OFFICE | VARCHAR(18) | |
| T4_CONTACT_INFORMATION | VARCHAR(30) | |
| T4_DEPARTMENT_STAFF | VARCHAR(18) | |
| T4_DEPARTMENT_LEADER | VARCHAR(18) | |

EXPLANATION:-

The T4_DEPARTMENT attributes record all the essential company department details within the insurance company. The T4_DEPARTMENT_NAME is the unique primary key and Foreign key and the T4_COMPANY_NAME is a foreign key linking the table back to the entity type COMPANY.

Table-LDM 11: T4_OFFICE

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|------------------------|-------------|-------------------------------|
| T4_OFFICE_NAME | VARCHAR(20) | PK |
| T4_DEPARTMENT_NAME | VARCHAR(20) | FK |
| T4_COMPANY_NAME | VARCHAR(20) | FK |
| T4_OFFICE_LEADER | VARCHAR(20) | |
| T4_CONTACT_INFORMATION | VARCHAR(20) | |
| T4_ADDRESS | VARCHAR(20) | |
| T4_ADMIN_COST | INT | |
| T4_STAFF | VARCHAR(50) | |

EXPLANATION:-

The T4_OFFICE attributes record all the essential office details within the insurance company. The T4_OFFICE_NAME is the unique primary key and the T4_DEPARTMENT_NAME and T4_COMPANY_NAME are foreign keys linking the table back to the respective entity types.

Table-LDM 12: T4_MEMBERSHIP

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-------------------------|-------------|-------------------------------|
| T4_MEMBERSHIP_ID | VARCHAR(20) | PK |
| T4_CUST_ID | INT | FK |
| T4_MEMBERSHIP_TYPE | CHAR(15) | |
| T4_ORGANISTAION_CONTACT | VARCHAR(20) | |

EXPLANATION:-

The T4_MEMBERSHIP attributes record all the essential membership details available for insured customers. The T4_MEMBERSHIP_ID is the unique primary key and the T4_CUST_ID is a foreign key linking the table back to the entity type T4_CUSTOMER.

Table-LDM 13: T4_VEHICLE_SERVICE

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-----------------------------|-------------|-------------------------------|
| T4_VEHICLE_SERVICE | VARCHAR(30) | PK |
| T4_VEHICLE_ID | INT | FK |
| T4_CUST_ID | INT | FK |
| T4_VEHICLE_SERVICE_ADDRESS | VARCHAR(20) | |
| T4_VEHICLE_SERVICE_CONTACT | VARCHAR(20) | |
| T4_VEHICLE_SERVICE_ | VARCHAR(20) | |
| T4_VEHICLE_SERVICE_INCHARGE | VARCHAR(20) | |
| T4_VEHICLE_SERVICE_TYPE | VARCHAR(20) | |
| T4_COMPANY_NAME | VARCHAR(20) | |

EXPLANATION:-

The T4_VEHICLE_SERVICE attributes record all the essential vehicle services offered to insured customer details. The T4_VEHICLE_SERVICE is the unique primary key and the T4_CUST_ID and T4_VEHICLE_ID are linked to the other corresponding entities.

Table-LDM 14: T4_NOK

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|---------------------------|-------------|-------------------------------|
| T4_NOK_ID | VARCHAR(20) | PK |
| T4_AGREEMENT_ID | VARCHAR(20) | FK |
| T4_APPLICTAION_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_NOK_NAME | VARCHAR(20) | |
| T4_NOK_ADRESS | VARCHAR(20) | |
| T4_NOK_PHONE_NUMBER | VARCHAR(10) | |
| T4_NOK_GENDER | CHAR(2) | |
| T4_NOK_MARITAL_STATU S | CHAR(10) | |

EXPLANATION:-

The T4_NOK attributes record information on the next of kin details. T4_NOK_ID is the unique primary key here. T4_AGREEMENT_ID, T4_APPLICATION_ID, and T4_CUST_ID are foreign keys linking back information to their respective entities.

Table-LDM 15: T4_INSURANCE_COMPANY

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|----------------------------|-------------|-------------------------------|
| T4_COMPANY_NAME | VARCHAR(20) | PK |
| T4_COMAPNY_DEPARTMENT_NAME | VARCHAR(20) | PK |
| T4_COMPANY_ADRESS | VARCHAR(20) | |
| T4_COMPANY_CONTACT_NUMBER | VARCHAR(10) | |
| T4_COMPANY_FAX | INT | |
| T4_COMPANY_EMAIL | VARCHAR(20) | |
| T4_COMPANY_WEBSITE | VARCHAR(20) | |
| T4_COMPANY_LOCTAION | VARCHAR(20) | |
| T4_COMPANY_OFFICE_NAME | VARCHAR(20) | |

EXPLANATION:-

The T4_INSURANCE COMPANY attributes record all the essential company details of the customer. The T4_COMPANY_NAME and T4_COMPANY_DEPARTMENT_NAME are the primary keys.

Table-LDM 16: T4_POLICY_RENEWABLE

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|------------------------|-------------|-------------------------------------|
| T4_POLICY_RENEWABLE_ID | VARCHAR(20) | PK |
| T4_AGREEMENT_ID | VARCHAR(20) | FK |
| T4_APPLICATION_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_DATE_OF_RENEWAL | DATE | |
| T4_TYPE_OF_RENEWAL | CHAR(15) | |

EXPLANATION:-

The T4_POLICY_RENEWABLE attributes record all the essential policy renewal details of the insured customer. The T4_POLICY_RENEWABLE_ID is the unique primary key and the T4_AGREEMENT_ID, T4_APPLICATION_ID and T4_CUST_ID are foreign keys linking the table back to the respective entities.

Table-LDM 17: T4_INCIDENT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|---------------|--------------|-------------------------------|
| INCIDENT_ID | VARCHAR(20) | PK |
| INCIDENT_TYPE | VARCHAR(30) | |
| INCIDENT_DATE | DATE | |
| DESCRIPTION | VARCHAR(100) | |

EXPLANATION:-

The T4_INCIDENT attributes record all the essential incident details such as Accident and theft on the insured customer vehicle. The T4_INCIDENT_ID is the unique primary key.

Table-LDM 18: T4_INCIDENT_REPORT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|--------------------------------|--------------|-------------------------------|
| T4_INCIDENT_REPORT_ID | VARCHAR(20) | PK |
| T4_INCIDENT_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_INCIDENT_TYEP | CHAR(10) | |
| T4_INCIDENT_INSPECTOR | VARCHAR(20) | |
| T4_INCIDENT_COST | INT | |
| T4_INCIDENT_REPORT_DESCRIPTION | VARCHAR(100) | |

EXPLANATION:-

The T4_ INCIDENT_REPORT_ID attributes record all the essential incident occurrences on the customer vehicle. The T4_INCIDENT_REPORT_ID is the unique primary key and the T4_CUST_ID, AND T4_INCIDENT_ID are foreign keys linking the table back to their respective entity types.

Table-LDM 19: T4_COVERAGE

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-------------------------|--------------|-------------------------------|
| T4_COVERAGE_ID | VARCHAR(20) | PK |
| T4_COMPANY_NAME | VARCHAR(20) | FK |
| T4_COVERAGE_AMOUNT | INT | |
| T4_COVERAGE_TYPE | CHAR(15) | |
| T4_COVERAGE_LEVEL | CHAR(10) | |
| T4_PRODUCT_ID | VARCHAR(20) | |
| T4_COVERAGE_DESCRIPTION | VARCHAR(100) | |
| T4_COVERAGE_TERMS | VARCHAR(50) | |

EXPLANATION:-

The T4_COVERAGE attributes record all the essential coverage details of the insurance policy to the customer. The T4_COVERAGE_ID is the unique primary key and the T4_COMPANY_NAME is a foreign key linking the table back to the entity type T4_COMPANY.

Table-LDM 20: T4_PRODUCT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-------------------|-------------|-------------------------------|
| T4_PRODUCT_NUMBER | VARCHAR(20) | PK |
| T4_COMPANY_NAME | VARCHAR(20) | FK |
| T4_PRODUCT_TYPE | CHAR(15) | |
| T4_PRODUCT_PRICE | INT | |

EXPLANATION:-

The T4_PRODUCT attributes record all the essential company products details offered by the Insurance company. The T4_PRODUCT_NUMBER is the unique primary key and T4_COMPANY_NAME is the foreign key linking table to T4_COMPANY entities.

Table-LDM 21: T4_RECEIPT

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-----------------------|-------------|-------------------------------------|
| T4_RECIEPT_ID | VARCHAR(20) | PK |
| T4_PREMIUM_PAYMENT_ID | VARCHAR(20) | FK |
| T4_CUST_ID | INT | FK |
| T4_TIME | DATE | |
| T4_COST | INT | |

EXPLANATION:-

The T4_RECEIPT attributes record all the essential payments done by T4_CUSTOMERS to the Insurance company. The T4_RECEIPT_ID is the unique primary key and T4_PREMIUM_PAYMENT_ID and T4_CUST_ID are Foreign keys linking tables to their respective entities.

Table-LDM 22: T4_INSURANCE_POLICY_COVERAGE

| Attributes | Data type | PRIMARY KEYS AND FOREIGN KEYS |
|-----------------|-------------|-------------------------------------|
| T4_AGREEMENT_ID | VARCHAR(20) | PK |
| T4_COVERAGE_ID | VARCHAR(20) | FK |

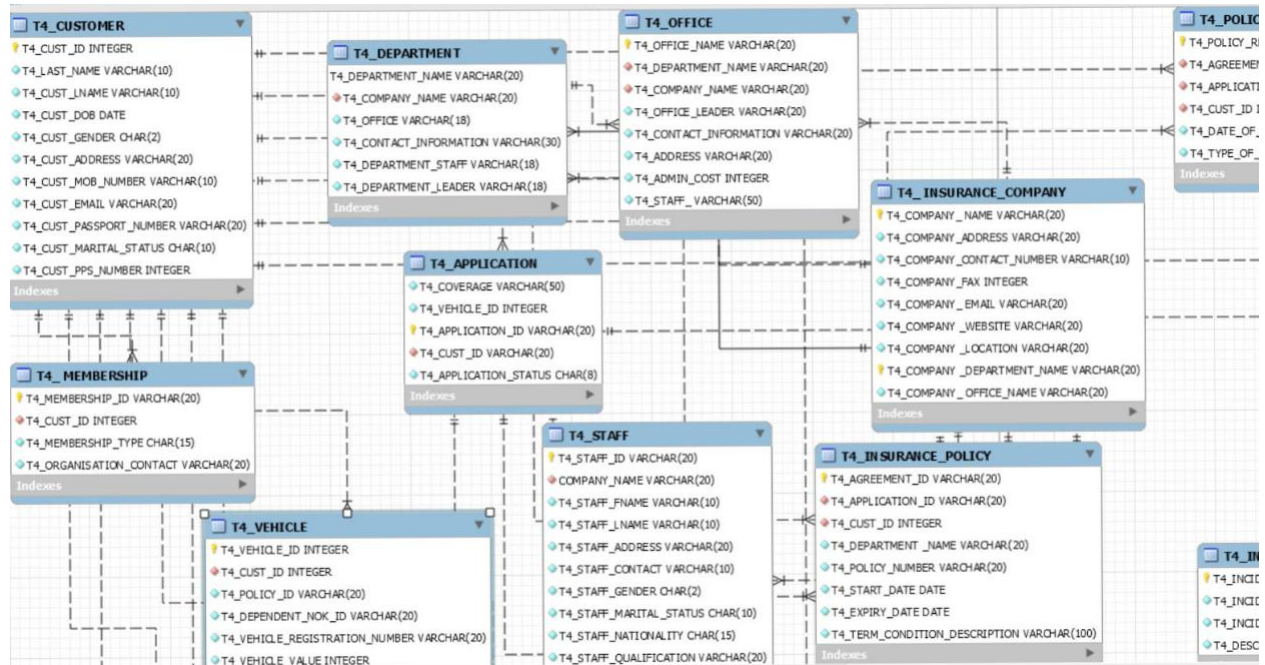
EXPLANATION:-

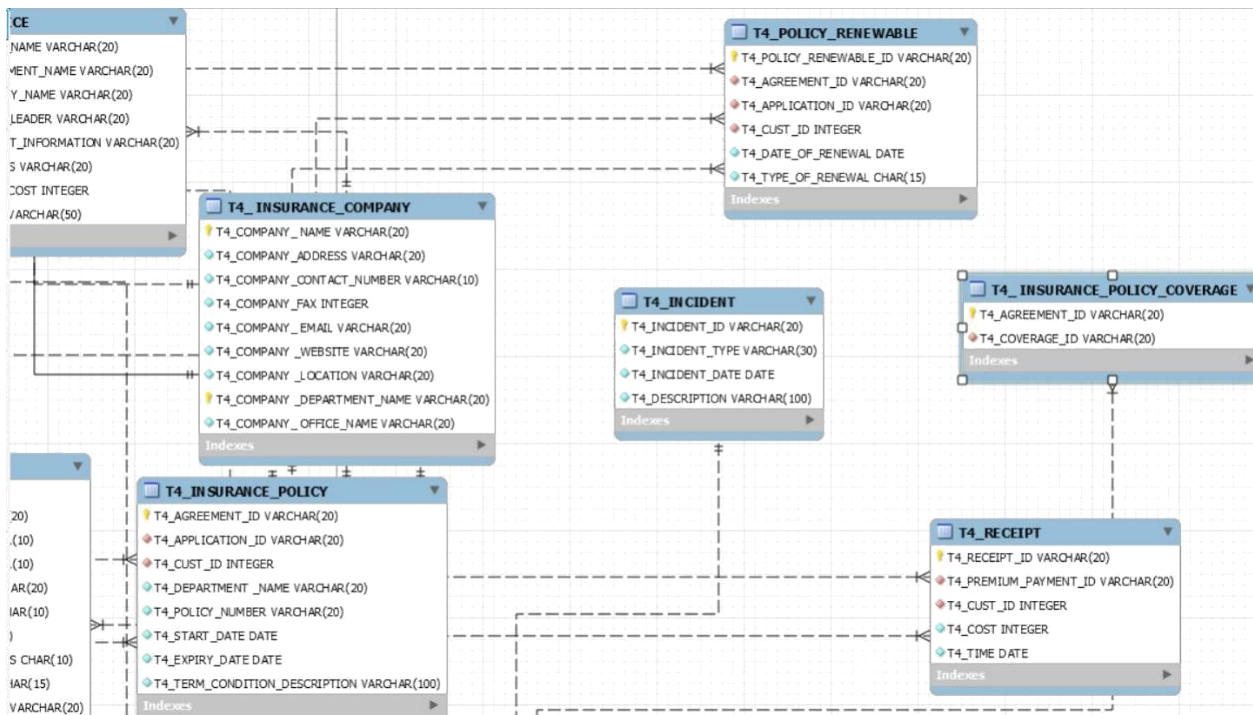
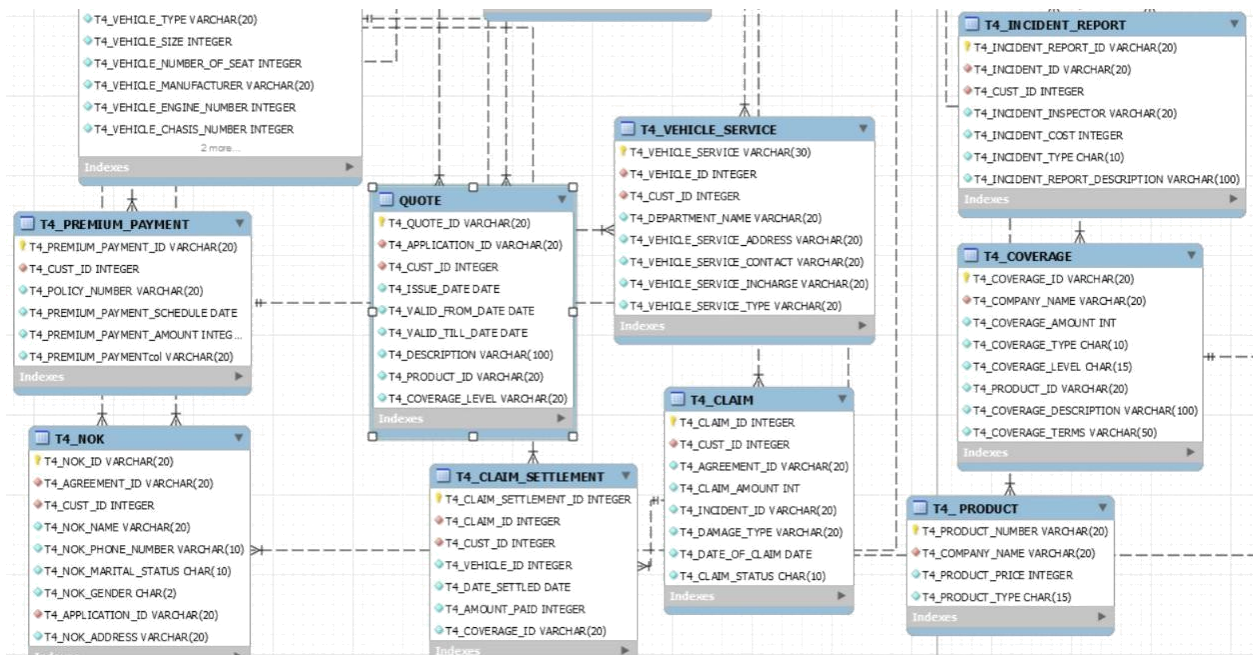
The T4_INSURANCE_POLICY_COVERAGE records details of the Vehicle policy that entails Terms Conditions of the Contract. T4_AGREEMENT_ID is the unique primary key and T4_COVERAGE_ID is the Foreign key linking this table to T4_COVERAGE Entity.

Graphical presentation of LDM

The Logical Data Model (LDM) that we have designed for this part of report in graphical Figure-LDM 1. It has all the entity types, attributes and relationships that are valid and pertinent in designing our online vehicle insurance database system.

Figure-LDM 1: Logical data model of Vehicle insurance for AVIVA Ltd.

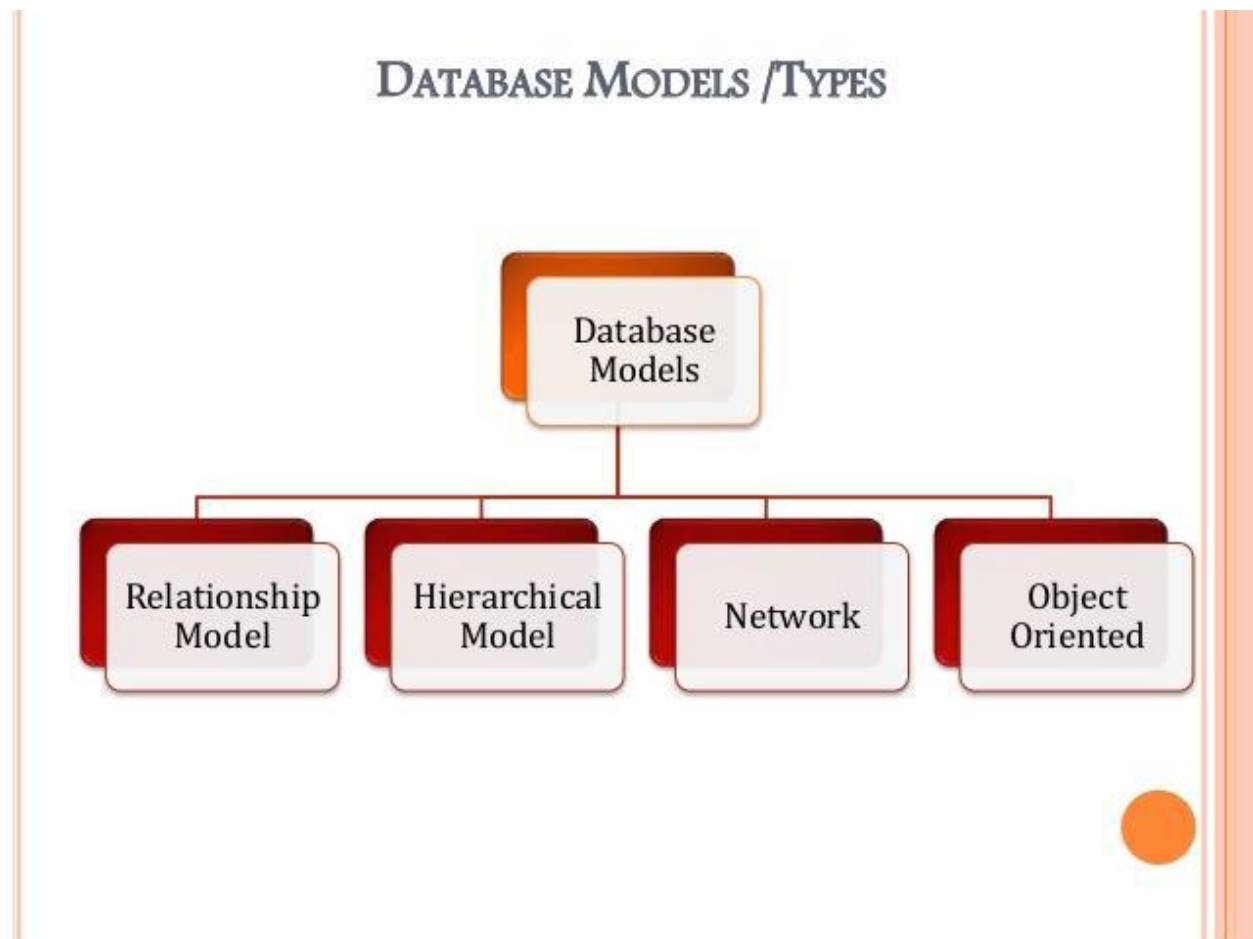




Part C:Physical Data Model and Database Design

Introduction

When building an entity relationship (ER) model we tend to use it to later build different physical models of database types. Therefore physical data model is used to implement into different technical software and hardware environments that is due to current state of technology and is changing as technologies change.



Normalization

When building an entity relationship (ER) model we tend to use it to later build different physical models of database types. Therefore physical data model is used to implement into different technical software and hardware environments that is due to current state of technology and is changing as technologies change.

1. First Normal Form (1NF):

If a relation contain composite or multi-valued attribute, it violates first normal form, or a relation is in first normal form if it does not contain any **composite** or **multi-valued attribute**. A relation is in first normal form if every attribute in that relation is singled valued attribute.

2. Second Normal Form (2NF):

Second Normal Form (2NF) is based on the concept of full functional dependency. To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency.

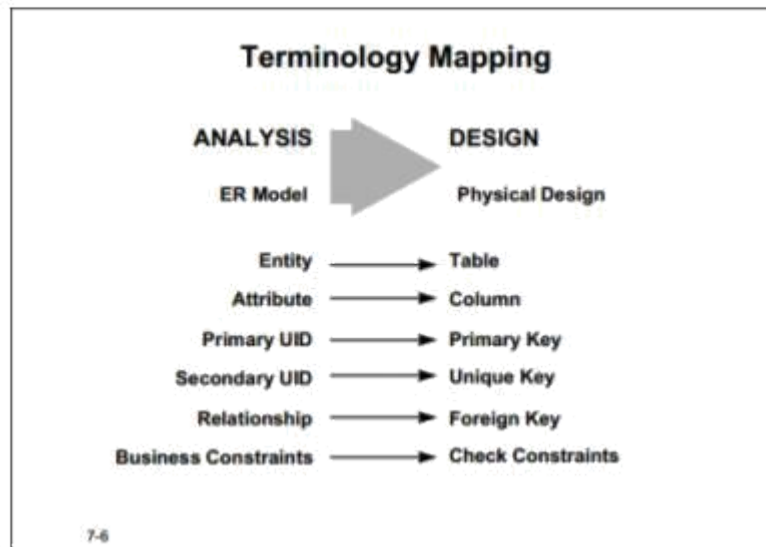
3. Third Normal Form (3NF):

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

| Normal form | Table |
|-----------------------------|--|
| First normal form (1NF) | |
| Second normal form (2NF) | DEPARTMENT OFFICE VEHICLE NOK INSURANCE_POLICY CLAIM CLAIM_SETTLEMENT PREMIUM_PAYMENT QUOTE INCIDENT_REPORT POLICY_RENEWABLE |
| Third normal form (3NF) | CUSTOMER RECEIPT APPLICATION STAFF INSURANCE_COMPANY MEMBERSHIP PRODUCT COVERAGE VEHICLE_SERVICE INCIDENT |

Terminology

The relational data model consists of mathematical rules, that later translate its syntax to physical model, but there are not always correspondent to syntax of physical model. Therefore you have to keep track of them and find how to implement them in correct way. This change also means change of terminology.



RDBMS

As we start doing to build relational data management system (RDBMS) we need a large number of parameters to obtain a correct adapted physical model. You must be aware that there is no absolute truth here. Some of most important points of creating physical models for RDBMS (Speelpenning, et al., 2001):

- Expected volume of tables, the hardware characteristics (CPU speed, memory size, number of disks and corresponding space), the architecture-client/server or three size, the network bandwidth, speed and operating systems are important determinants.
- User experience second big issue (response time, the GUI and frequency of use of modules).
- Depending which version you use as some elements may or may not exist.

PHYSICAL MODEL

When creating physical models we create tables or clusters and we must write specifications of internal data type for each its columns. These types define generic domain of values that each column can contain.

Some concerns using data types building a physical model:

- Data types can have a narrow focus (number, date).
- Some are general purpose data types (various character data types).
- Data types can allow for variable length or not. Choosing a large fixed length for a column to only store a few bytes per row, makes a large table. This may affect performance specially if stored on multiple blocks, resulting in great number of I/O's and so affecting performance.
- Large data object types are not advised to be used in where clause as they are only retrievable against other columns

The Data types used in our project:

- **CHAR (size)** these are fixed-length character data of length-sized bytes. Maximum size is 2000 bytes.

Typical use: for official International Currency Codes which are a fixed three characters in length such as USD, FFR.

- **VARCHAR (size)** Variable-length character string having maximum length-sized bytes. Maximum size is 4000, and minimum is 1. This is the most commonly-used data type and you should use it if you are not sure which one to use.

Typical use: for storing individual ASCII text lines of unlimited length ASCII texts on which you need to be able to search using a wildcard.

- **DATE** Valid date range from January 1, 4712 BC to December 31, 4712 AD. A date data type also contains time components. You should use it only when you know the full date

including day, month, and year. The time component is often set to 00:00 (midnight) in normal use of dates.

Typical use: any date where the full date is known.

- **INTEGER**: The INTEGER datatype is usually referred to as NUMBER(38). Its precision can range from 1 to 38.

DATABASE SECURITY

Database security is described by the following aspects :

- Data independence

Data independence is an important method ensures data security; it can be divided into logical independence and physical independence. Physical independence means applications and data are store independent of each other data is managed by DBMS and application not need understand it, application just need processing the data when the physical storage of data changing application without changing.

- Data Security

1. Isolation protect important files in database.
2. Using authorization rules, such as access control method and accounts, passwords permissions control.
3. Data Encryption

- Data Integrity

1. Data validation: ensure clean, correct and useful data.
2. Data consistency: Different users are using the same data should be identical

3. Data correctness: The input value of the data should be consistent with data in database

- **Concurrency Control**

Concurrency Control means the database is a shared resource for multi-use, When multiple user concurrent access to data, in the database will have multiple transactions simultaneously access the same data. If not controlled for concurrent operation may be cause incorrect to read and store data, destroy the consistency of the database. Locking mechanism of the database can effectively protect the database achieve concurrency control.

The database provides three different types of locks:

1. DML lock: DML locks used to protect data integrity; DML locks mainly include TM lock and TX lock. TM lock called table lock, TX called transaction locks or row locks.

2. DDL lock: DDL lock protects the structure of the database objects. In the DDL operation is automatically added DDL lock to the object, to protect these objects will not be modified by other sessions.

3. Internal locks and Latches: Protect the internal structure of the database.

- **Recovery**

When the database fails DBMS need to find faults and fix problems, thus preventing data corruption. Moreover database should regularly back up and establish a spare machine, makes the database can be restored as quickly as possible from the fault.

PDM Code

| Table No | PDM Code |
|-------------|----------|
|-------------|----------|

| | |
|---|--|
| 1 | <pre>CREATE TABLE IF NOT EXISTS T4_CUSTOMER (T4_CUST_ID INTEGER NOT NULL , T4_CUST_FNAME VARCHAR(10) NOT NULL , T4_CUST_LNAME VARCHAR(10) NOT NULL , T4_CUST_DOB DATE NOT NULL , T4_CUST_GENDER CHAR(2) NOT NULL , T4_CUST_ADDRESS VARCHAR(20) NOT NULL , T4_CUST_MOB_NUMBER VARCHAR(10) NOT NULL , T4_CUST_EMAIL VARCHAR(20) NOT NULL , T4_CUST_PASSPORT_NUMBER VARCHAR(20) NOT NULL , T4_CUST_MARITAL_STATUS CHAR(10) NOT NULL , T4_CUST_PPS_NUMBER INTEGER NOT NULL , CONSTRAINT T4_XPKCUSTOMER_1 PRIMARY KEY (T4_CUST_ID));</pre> |
|---|--|

| | |
|---|--|
| 2 | <pre>CREATE TABLE IF NOT EXISTS T4_INCIDENT (T4_INCIDENT_ID VARCHAR(20) NOT NULL , T4_INCIDENT_TYPE VARCHAR(30) NOT NULL , T4_INCIDENT_DATE DATE NOT NULL , T4_DESCRIPTION VARCHAR(100) NOT NULL , CONSTRAINT T4_XPKINCIDENT_17 PRIMARY KEY (T4_INCIDENT_ID));</pre> |
|---|--|

3

```
CREATE TABLE IF NOT EXISTS T4_INCIDENT_REPORT
(
    T4_INCIDENT_REPORT_ID VARCHAR(20) NOT NULL ,
    T4_INCIDENT_TYPE CHAR(10) NOT NULL ,
    T4_INCIDENT_INSPECTOR VARCHAR(20) NOT NULL ,
    T4_INCIDENT_COST INTEGER NOT NULL ,
    T4_INCIDENT_REPORT_DESCRIPTION VARCHAR(100) NOT
NULL ,
    T4_INCIDENT_ID VARCHAR(20) NOT NULL ,
    T4_CUST_ID INTEGER NOT NULL ,

    CONSTRAINT T4_XPKINCIDENT_REPORT_18 PRIMARY KEY
(T4_INCIDENT_REPORT_ID),

    CONSTRAINT T4_R_83 FOREIGN KEY (T4_INCIDENT_ID)
REFERENCES T4_INCIDENT(T4_INCIDENT_ID) ON DELETE
CASCADE,

    CONSTRAINT T4_R_86 FOREIGN KEY (T4_CUST_ID)
REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE
CASCADE

);
```

| | |
|---|--|
| 4 | <pre>CREATE TABLE IF NOT EXISTS T4_INSURANCE_COMPANY (T4_COMPANY_NAME VARCHAR(20) NOT NULL , T4_COMPANY_ADDRESS VARCHAR(20) NOT NULL , T4_COMPANY_CONTACT_NUMBER VARCHAR(10) NOT NULL , T4_COMPANY_FAX INTEGER NOT NULL , T4_COMPANY_EMAIL VARCHAR(20) NOT NULL , T4_COMPANY_WEBSITE VARCHAR(20) NOT NULL , T4_COMPANY_LOCATION VARCHAR(20) NOT NULL , T4_COMPANY_DEPARTMENT_NAME VARCHAR(20) NOT NULL , T4_COMPANY_OFFICE_NAME VARCHAR(20) NOT NULL , CONSTRAINT T4_XPKINSURANCE_COMPANY_15 PRIMARY KEY(T4_COMPANY_NAME,T4_COMPANY_DEPARTMENT_NAME));</pre> |
|---|--|

5

CREATE TABLE IF NOT EXISTS T4_DEPARTMENT

(

T4_DEPARTMENT_NAME VARCHAR(20) NOT NULL ,

T4_COMPANY_NAME VARCHAR(20) NOT NULL ,

T4_OFFICE VARCHAR(18) NOT NULL ,

T4_CONTACT_INFORMATION VARCHAR(30) NOT NULL,

T4_DEPARTMENT_STAFF VARCHAR(18) NOT NULL ,

T4_DEPARTMENT_LEADER VARCHAR(18) NOT NULL ,

CONSTRAINT T4_XPKDEPARTMENT PRIMARY
KEY (T4_DEPARTMENT_NAME),

CONSTRAINT T4_R_56 FOREIGN KEY
(T4_COMPANY_NAME,T4_DEPARTMENT_NAME) REFERENCES
T4_INSURANCE_COMPANY
(T4_COMPANY_NAME,T4_COMPANY_DEPARTMENT_NAME)
ON UPDATE CASCADE

);

6

```
CREATE TABLE IF NOT EXISTS T4_VEHICLE
(
    T4_VEHICLE_ID INTEGER NOT NULL ,
    T4_POLICY_ID VARCHAR(20) NOT NULL ,
    T4_DEPENDENT_NOK_ID VARCHAR(20) NOT NULL ,
    T4_VEHICLE_REGISTRATION_NUMBER VARCHAR(20) NOT
NULL ,
    T4_VEHICLE_VALUE INTEGER NOT NULL ,
    T4_VEHICLE_TYPE VARCHAR(20) NOT NULL ,
    T4_VEHICLE_SIZE INTEGER NOT NULL ,
    T4_VEHICLE_NUMBER_OF_SEAT INTEGER NOT NULL ,
    T4_VEHICLE_MANUFACTURER VARCHAR(20) NOT NULL ,
    T4_VEHICLE_ENGINE_NUMBER INTEGER NOT NULL ,
    T4_VEHICLE_CHASIS_NUMBER INTEGER NOT NULL ,
    T4_VEHICLE_NUMBER VARCHAR(20) NOT NULL ,
    T4_VEHICLE_MODEL_NUMBER VARCHAR(20) NOT NULL ,
    T4_CUST_ID INTEGER NOT NULL ,

    CONSTRAINT T4_XPKVEHICLE_6 PRIMARY
KEY (T4_VEHICLE_ID),

    CONSTRAINT T4_R_92 FOREIGN KEY (T4_CUST_ID)
REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE
CASCADE

);
```

7

```
CREATE TABLE IF NOT EXISTS T4_VEHICLE_SERVICE
(
    T4_VEHICLE_SERVICE VARCHAR(30) NOT NULL ,
    T4_VEHICLE_ID INTEGER NOT NULL ,
    T4_CUST_ID INTEGER NOT NULL ,
    T4_DEPARTMENT_NAME VARCHAR(20) NOT NULL ,
    T4_VEHICLE_SERVICE_ADDRESS VARCHAR(20) NOT NULL
    ,
    T4_VEHICLE_SERVICE_CONTACT VARCHAR(20) NOT NULL
    ,
    T4_VEHICLE_SERVICE_INCHARGE VARCHAR(20) NOT NULL
    ,
    T4_VEHICLE_SERVICE_TYPE VARCHAR(20) NOT NULL ,
    T4_COMPANY_NAME VARCHAR(20) NOT NULL ,

    CONSTRAINT T4_XPKVEHICLE_SERVICE PRIMARY
    KEY(T4_VEHICLE_SERVICE),

    CONSTRAINT T4_R_50 FOREIGN KEY (T4_VEHICLE_ID)
    REFERENCES T4_VEHICLE(T4_VEHICLE_ID) ON DELETE
    CASCADE,

    CONSTRAINT T4_R_51 FOREIGN KEY (T4_CUST_ID)
    REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE
    RESTRICT
);
```

| | |
|---|---|
| 8 | <pre>CREATE TABLE IF NOT EXISTS T4_PREMIUM_PAYMENT (T4_PREMIUM_PAYMENT_ID VARCHAR(20) NOT NULL , T4_POLICY_NUMBER VARCHAR(20) NOT NULL , T4_PREMIUM_PAYMENT_AMOUNT INTEGER NOT NULL DEFAULT 0 , T4_PREMIUM_PAYMENT_SCHEDULE DATE NOT NULL , T4_RECEIPT_ID VARCHAR(20) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKPREMIUM_PAYMENT_5 PRIMARY KEY(T4_PREMIUM_PAYMENT_ID), CONSTRAINT T4_R_85 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE CASCADE);</pre> |
|---|---|

9

```
CREATE TABLE IF NOT EXISTS T4_RECEIPT
(
    T4_RECEIPT_ID VARCHAR(20) NOT NULL ,
    T4_TIME DATE NOT NULL ,
    T4_COST INTEGER NOT NULL ,
    T4_PREMIUM_PAYMENT_ID VARCHAR(20) NOT NULL ,
    T4_CUST_ID INTEGER NOT NULL ,

    CONSTRAINT T4_XPKRECEIPT_21
PRIMARY KEY(T4_RECEIPT_ID),

    CONSTRAINT T4_R_84 FOREIGN KEY
(T4_PREMIUM_PAYMENT_ID) REFERENCES
T4_PREMIUM_PAYMENT (T4_PREMIUM_PAYMENT_ID) ON
UPDATE CASCADE ON DELETE RESTRICT,

    CONSTRAINT T4_R_88 FOREIGN KEY (T4_CUST_ID)
REFERENCES T4_CUSTOMER ( T4_CUST_ID) ON UPDATE
RESTRICT ON DELETE CASCADE

);
```


| | |
|----|---|
| 10 | <pre>CREATE TABLE IF NOT EXISTS T4_APPLICATION (T4_APPLICATION_ID VARCHAR(20) NOT NULL , T4_VEHICLE_ID INTEGER NOT NULL , T4_APPLICATION_STATUS CHAR(8) NOT NULL , T4_COVERAGE VARCHAR(50) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKAPPLICATION_2 PRIMARY KEY(T4_APPLICATION_ID), CONSTRAINT T4_R_93 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON UPDATE RESTRICT);</pre> |
|----|---|

| | |
|----|---|
| 11 | <pre>CREATE TABLE IF NOT EXISTS T4_INSURANCE_POLICY (T4_AGREEMENT_ID VARCHAR(20) NOT NULL , T4_DEPARTMENT_NAME VARCHAR(20) NOT NULL , T4_POLICY_NUMBER VARCHAR(20) NOT NULL , T4_START_DATE DATE NOT NULL , T4_EXPIRY_DATE DATE NOT NULL , T4_TERM_CONDITION_DESCRIPTION VARCHAR(100) NOT NULL , T4_APPLICATION_ID VARCHAR(20) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKINSURANCE_POLICY_4 PRIMARY KEY(T4_AGREEMENT_ID), CONSTRAINT T4_R_95 FOREIGN KEY (T4_APPLICATION_ID) REFERENCES T4_APPLICATION (T4_APPLICATION_ID) ON DELETE CASCADE , CONSTRAINT T4_R_96 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER (T4_CUST_ID) ON UPDATE RESTRICT);</pre> |
|----|---|

| | |
|----|--|
| 12 | <pre>CREATE TABLE IF NOT EXISTS T4_POLICY_RENEWABLE (T4_POLICY_RENEWABLE_ID VARCHAR(20) NOT NULL , T4_DATE_OF_RENEWAL DATE NOT NULL , T4_TYPE_OF_RENEWAL CHAR(15) NOT NULL , T4_AGREEMENT_ID VARCHAR(20) NOT NULL , T4_APPLICATION_ID VARCHAR(20) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKPOLICY_RENEWABLE_16 PRIMARY KEY(T4_POLICY_RENEWABLE_ID), CONSTRAINT T4_R_101 FOREIGN KEY (T4_AGREEMENT_ID)REFERENCES T4_INSURANCE_POLICY (T4_AGREEMENT_ID) ON DELETE CASCADE, CONSTRAINT T4_R_102 FOREIGN KEY (T4_APPLICATION_ID)REFERENCES T4_APPLICATION(T4_APPLICATION_ID) ON UPDATE CASCADE, CONSTRAINT T4_R_103 FOREIGN KEY (T4_CUST_ID)REFERENCES T4_CUSTOMER (T4_CUST_ID) ON UPDATE RESTRICT);</pre> |
|----|--|

| | |
|----|---|
| 13 | <pre>CREATE TABLE IF NOT EXISTS T4_MEMBERSHIP (T4_MEMBERSHIP_ID VARCHAR(20) NOT NULL , T4_MEMBERSHIP_TYPE CHAR(15) NOT NULL , T4_ORGANISATION_CONTACT VARCHAR(20) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKMEMBERSHIP_12 PRIMARY KEY(T4_MEMBERSHIP_ID), CONSTRAINT T4_R_91 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE CASCADE);</pre> |
|----|---|

14

```
CREATE TABLE IF NOT EXISTS T4_QUOTE
(
    T4_QUOTE_ID VARCHAR(20) NOT NULL ,
    T4_ISSUE_DATE DATE NOT NULL ,
    T4_VALID_FROM_DATE DATE NOT NULL ,
    T4_VALID_TILL_DATE DATE NOT NULL ,
    T4_DESCRIPTION VARCHAR(100) NOT NULL ,
    T4_PRODUCT_ID VARCHAR(20) NOT NULL ,
    T4_COVERAGE_LEVEL VARCHAR(20) NOT NULL ,
    T4_APPLICATION_ID VARCHAR(20) NOT NULL ,
    T4_CUST_ID INTEGER NOT NULL ,

    CONSTRAINT T4_XPKQUOTE_3 PRIMARY KEY
(T4_QUOTE_ID),

    CONSTRAINT T4_R_71 FOREIGN KEY (T4_APPLICATION_ID)
REFERENCES T4_APPLICATION (T4_APPLICATION_ID) ON
DELETE CASCADE,

    CONSTRAINT T4_R_72 FOREIGN KEY (T4_CUST_ID)
REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE
CASCADE

);
```

15

CREATE TABLE IF NOT EXISTS T4_STAFF

(

T4_STAFF_ID VARCHAR(20) NOT NULL ,

T4_STAFF_FNAME VARCHAR(10) NOT NULL ,

T4_STAFF_LNAME VARCHAR(10) NOT NULL ,

T4_STAFF_ADDRESS VARCHAR(20) NOT NULL ,

T4_STAFF_CONTACT VARCHAR(10) NOT NULL ,

T4_STAFF_GENDER CHAR(2) NOT NULL ,

T4_STAFF_MARITAL_STATUS CHAR(10) NOT NULL ,

T4_STAFF_NATIONALITY CHAR(15) NOT NULL ,

T4_STAFF_QUALIFICATION VARCHAR(20) NOT NULL ,

T4_STAFF_ALLOWANCE INTEGER NOT NULL ,

T4_STAFF_PPS_NUMBER INTEGER NOT NULL ,

T4_COMPANY_NAME VARCHAR(20) NOT NULL ,

CONSTRAINT T4_XPKSTAFF_9 PRIMARY KEY
(T4_STAFF_ID),

CONSTRAINT T4_R_105 FOREIGN KEY
(T4_COMPANY_NAME) REFERENCES T4_INSURANCE_COMPANY
(T4_COMPANY_NAME) ON UPDATE CASCADE

);

| | |
|----|---|
| 16 | <pre>CREATE TABLE IF NOT EXISTS T4_NOK (T4_NOK_ID VARCHAR(20) NOT NULL , T4_NOK_NAME VARCHAR(20) NOT NULL , T4_NOK_ADDRESS VARCHAR(20) NOT NULL , T4_NOK_PHONE_NUMBER VARCHAR(10) NOT NULL , T4_NOK_GENDER CHAR(2) NOT NULL , T4_NOK_MARITAL_STATUS CHAR(10) NOT NULL , T4_AGREEMENT_ID VARCHAR(20) NOT NULL , T4_APPLICATION_ID VARCHAR(20) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKNOK_14 PRIMARY KEY (T4_NOK_ID), CONSTRAINT R_73 FOREIGN KEY (T4_AGREEMENT_ID) REFERENCES T4_INSURANCE_POLICY (T4_AGREEMENT_ID) ON DELETE CASCADE, CONSTRAINT R_74 FOREIGN KEY (T4_APPLICATION_ID) REFERENCES T4_APPLICATION(T4_APPLICATION_ID) ON DELETE CASCADE, CONSTRAINT R_75 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON UPDATE RESTRICT ON DELETE CASCADE);</pre> |
|----|---|

| | |
|----|--|
| 17 | <pre>CREATE TABLE IF NOT EXISTS T4_PRODUCT (T4_PRODUCT_PRICE INTEGER NOT NULL , T4_PRODUCT_TYPE CHAR(15) NOT NULL , T4_PRODUCT_NUMBER VARCHAR(20) NOT NULL , T4_COMPANY_NAME VARCHAR(20) NOT NULL , CONSTRAINT T4_XPKPRODUCT_20 PRIMARY KEY (T4_PRODUCT_NUMBER), CONSTRAINT T4_R_107 FOREIGN KEY (T4_COMPANY_NAME) REFERENCES T4_INSURANCE_COMPANY (T4_COMPANY_NAME) ON UPDATE CASCADE);</pre> |
|----|--|

18

CREATE TABLE IF NOT EXISTS T4_OFFICE

(

T4_OFFICE_NAME VARCHAR(20) NOT NULL ,

T4_OFFICE_LEADER VARCHAR(20) NOT NULL ,

T4_CONTACT_INFORMATION VARCHAR(20) NOT NULL ,

T4_ADDRESS VARCHAR(20) NOT NULL ,

T4_ADMIN_COST INTEGER NOT NULL ,

T4_STAFF VARCHAR(50) NOT NULL ,

T4_DEPARTMENT_NAME VARCHAR(20) NOT NULL ,

T4_COMPANY_NAME VARCHAR(20) NOT NULL ,

CONSTRAINT T4_XPKOFFICE_11 PRIMARY
KEY (T4_OFFICE_NAME),

CONSTRAINT T4_R_110 FOREIGN KEY
(T4_DEPARTMENT_NAME) REFERENCES T4_DEPARTMENT
(T4_DEPARTMENT_NAME) ON DELETE CASCADE,

CONSTRAINT T4_R_111 FOREIGN KEY
(T4_COMPANY_NAME) REFERENCES T4_INSURANCE_COMPANY
(T4_COMPANY_NAME) ON UPDATE CASCADE

);

19

```
CREATE TABLE IF NOT EXISTS T4_COVERAGE
(
    T4_COVERAGE_ID VARCHAR(20) NOT NULL ,
    T4_COVERAGE_AMOUNT INTEGER(20) NOT NULL ,
    T4_COVERAGE_TYPE CHAR(10) NOT NULL ,
    T4_COVERAGE_LEVEL CHAR(15) NOT NULL ,
    T4_PRODUCT_ID VARCHAR(20) NOT NULL ,
    T4_COVERAGE_DESCRIPTION VARCHAR(100) NOT NULL ,
    T4_COVEARGE_TERMS VARCHAR(50) NOT NULL ,
    T4_COMPANY_NAME VARCHAR(20) NOT NULL ,
    CONSTRAINT T4_XPKCOVERAGE_19 PRIMARY
KEY (T4_COVERAGE_ID),
    CONSTRAINT T4_R_112 FOREIGN KEY
(T4_COMPANY_NAME) REFERENCES T4_INSURANCE_COMPANY
(T4_COMPANY_NAME) ON UPDATE CASCADE
);
```

| | |
|----|--|
| 20 | <pre> CREATE TABLE IF NOT EXISTS T4_INSURANCE_POLICY_COVERAGE (T4_AGREEMENT_ID VARCHAR(20) NOT NULL , T4_COVERAGE_ID VARCHAR(20)NOT NULL , CONSTRAINT T4_XPKINSURANCE_POLICY_4 PRIMARY KEY(T4_AGREEMENT_ID), CONSTRAINT T4_R_98 FOREIGN KEY (T4_COVERAGE_ID)REFERENCES T4_COVERAGE(T4_COVERAGE_ID) ON DELETE CASCADE); </pre> |
| 21 | <pre> CREATE TABLE IF NOT EXISTS T4_CLAIM (T4_CLAIM_ID INTEGER NOT NULL , T4_AGREEMENT_ID VARCHAR(20) NOT NULL , T4_CLAIM_AMOUNT INTEGER(20) NOT NULL , T4_INCIDENT_ID VARCHAR(20) NOT NULL , T4_DAMAGE_TYPE VARCHAR(20) NOT NULL , T4_DATE_OF_CLAIM DATE NOT NULL , T4_CLAIM_STATUS CHAR(10) NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKCLAIM_7 PRIMARY KEY (T4_CLAIM_ID), </pre> |

```
CONSTRAINT T4_R_115 FOREIGN KEY (T4_CUST_ID)
REFERENCES T4_CUSTOMER(T4_CUST_ID) ON
UPDATE RESTRICT
```

```
);
```

| | |
|----|---|
| 22 | <pre> CREATE TABLE IF NOT EXISTS T4_CLAIM_SETTLEMENT (T4_CLAIM_SETTLEMENT_ID INTEGER NOT NULL , T4_VEHICLE_ID INTEGER NOT NULL , T4_DATE_SETTLED DATE NOT NULL , T4_AMOUNT_PAID INTEGER NOT NULL , T4_COVERAGE_ID VARCHAR(20) NOT NULL , T4_CLAIM_ID INTEGER NOT NULL , T4_CUST_ID INTEGER NOT NULL , CONSTRAINT T4_XPKCLAIM_SETTLEMENT_8 PRIMARY KEY(T4_CLAIM_SETTLEMENT_ID), CONSTRAINT T4_R_120 FOREIGN KEY (T4_CLAIM_ID) REFERENCES T4_CLAIM(T4_CLAIM_ID) ON DELETE CASCADE, CONSTRAINT T4_R_121 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON DELETE CASCADE); </pre> |
|----|---|

Who has access to certain parts of database ?

“A CRUD matrix is a table showing the functions in an application containing SQL statement affecting parts of a database” (Williams, 2001). It is a great way to show us what kind interaction appears between user and tables in database. This analysis of possible user scenarios and shows us tables that are very used and those that are not used, and it brings us the view how database is burdened with possible – bottleneck in system performance (Williams, 2001).

Using four SQL statements:

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

| MODULES | Customer | Manger of insurance | Insurance agent | Accountant | HR department | Damage inspector | Database administrator | Finance department | |
|-------------------|----------|---------------------|-----------------|------------|---------------|------------------|------------------------|--------------------|--|
| ENTITIES | | | | | | | | | |
| 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 | |

Conclusion:

We have a class project created database (DB) with all documentations and reports included. Our goal was to create DB for Online vehicle insurance company with code . There were some big and small challenges but we succeeded in making a functional DB. We started to build conceptual data model (CDM) we continued with logical data model (LDM) and then we made physical data model (PDM) all in Erwin software program.