CS310

DBMS Project Report

Vehicle Insurance

By,

Kuluru Vineeth Kumar Reddy, 18BCS043

Karthick P S,18BCS038

K.Laxmi Narayana,18BCS037

M.Venkata Kalyan Babu,18BCS049

MP.Bharath,18BCS057

B.Ragavan,18BCS016

G.Jagan Mohan Reddy,18BCS029

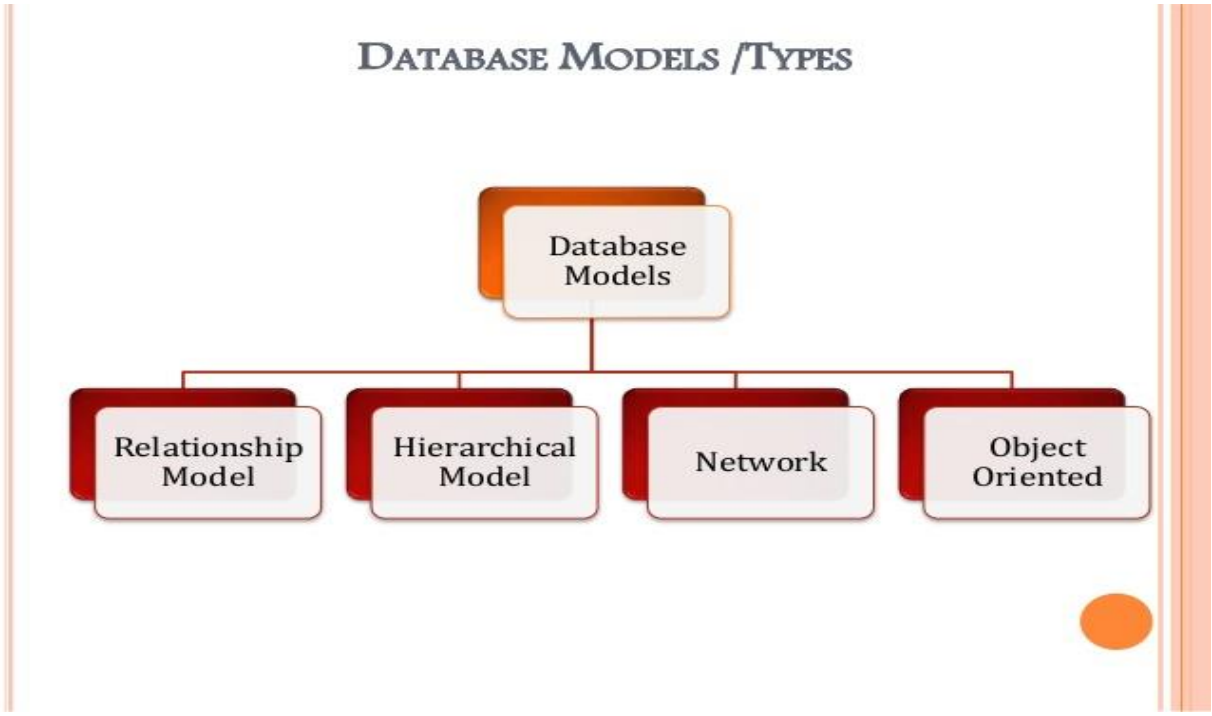
N.Trinath Reddy,18BCS059

P.Chandhan,18BCS063

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.

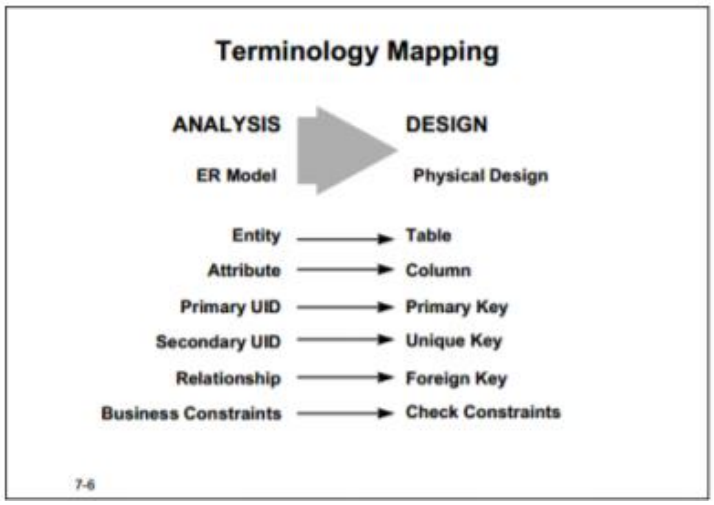
3. Third Normal Form (3NF):
A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

Normalization of Vehicle Insurance Company

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. For our assignment we created data types for Oracle

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

  Data integrity includes:

  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	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));
2	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));

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	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));
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	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);
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	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);
11	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);
12	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);
13	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);
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_XPKQU0TE_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	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);
17	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);
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	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);
21	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), CONSTRAINT T4_R_115 FOREIGN KEY (T4_CUST_ID) REFERENCES T4_CUSTOMER(T4_CUST_ID) ON UPDATE RESTRICT);
22	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);

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 / 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	

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.