

**CS310**

**DBMS**

**Group Assignment**

**Vehicle Insurance Company**

# TABLE OF CONTENTS

AN INSURANCE COMPANY  
DATABASE

<b>I. PROJECT INITIATION DOCUMENT</b>	<b>4</b>
<i>About a project</i>	
<i>Section 1: What is the project all about?</i>	
<b>II. PART A: CONCEPTUAL DATA MODEL (10%)</b>	<b>4</b>
<i>Design Rules</i>	<b>5</b>
<i>Assumptions</i>	<b>6</b>
<i>Entity Types</i>	<b>7</b>
<i>Relationships in CDM</i>	<b>8</b>
<i>Applying Relationships to Entities</i>	<b>8</b>
<i>Graphical presentation of CDM</i>	<b>9</b>
<b>III. PART B: LOGICAL DATA MODEL (15%)</b>	
<i>Introduction</i>	
<i>Introduction of terms used for constructing LDM</i>	<b>11</b>
<i>Elements</i>	
<i>Relations</i>	
<i>Constraints</i>	<b>12</b>

# TABLE OF CONTENTS

AN INSURANCE COMPANY  
DATABASE

## III. PART B: LOGICAL DATA MODEL (15%)

<i>Data Types</i>	<b>12</b>
<i>Business Rules, Attributes, Data types and Primary/Foreign Keys</i>	<b>13</b>
<i>Graphical presentation of LDM</i>	<b>29</b>
<i>Modifications to CDM</i>	

## IV. PART C: PHYSICAL DATA MODEL AND DATABASE DESIGN (25%)

<i>Introduction</i>	<b>30</b>
<i>Changes made from LDM to PDM (MS Access-Oracle)</i>	<b>37</b>
<i>PDM for vehicle insurance service.</i>	<b>38</b>
<i>Who has access to certain parts of database</i>	<b>49</b>
<i>Debrief – Discussed lessons learned</i>	<b>49</b>
<i>Conclusion</i>	<b>50</b>

## VI. QUERIES

**51**



# PROJECT TEAM NAMES

**PROJECT : VEHICLE INSURANCE COMPANY**

NAMES	REG NO
1) Anjuru Lokesh	18BCS006
2) A.S.V.Aditya	18BCS011
3) D.Geetha Krishna	18BCS025
4) CH.N.V.Avinash	18BCS021
5) G. Pranav	18BCS028
6) S. Kalyan	18BCS084
7) M. Sai	18BCS050
8) G. Nikhil	18BCS030
9) K. Sai Ganesh	18BCS044

## ABOUT A PROJECT

### Section 1: What is the project all about?

#### Project title:

A database for a Vehicle Insurance Company

#### Background:

**Aim :** The aim of the Project is to develop the data base on 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 (10%)

#### Design Rules :

Firstly, the Conceptual Data Model focuses focus on entities and their relationships used to design our car insurance database the particular team decides the various rules to be implemented in the project database and also needed to decide what characteristics underpin the model under investigation. The Decided rules as Follows below:

**Table 14\_1 : Design rules for CDM model of car insurance database**

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

## Assumptions

The team members follows the certain set of assumptions to design the CDM. These assumptions will helps the shape our model to allow consistency within our design and these assumptions may results the customer personal details must to apply insurance in this model. Those Assumptions as follows below:

**Tabel 14\_2 : Assumptions used in car insurance database model**

<b>Assumption</b>	<b>Description</b>
<b>Assumption 1</b>	Customer must have original proof ids like driving license and aadhar etc.,
<b>Assumption 2</b>	Customer not having personal accidental cases before in on-road.
<b>Assumption 3</b>	The online insurance is given to customers over 18 years of age
<b>Assumption 4</b>	The online insurance needs some driving history of customer
<b>Assumption 5</b>	The online insurance needs to know the type of car the customer drives
<b>Assumption 6</b>	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 below.

**Table 14\_3: Entity types used in car insurance data base system CDM model.**

<b>Entity Type</b>	<b>Description</b>
CUSTOMER	Records all the personal details about the customer
APPLICATION	Records details of the insurance coverage requested by 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 claims 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 o kin
TERMS_CONDITIONS	Records all terms and conditions in regard to 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 below.

**Table 14\_4 : 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 ERwin software.

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

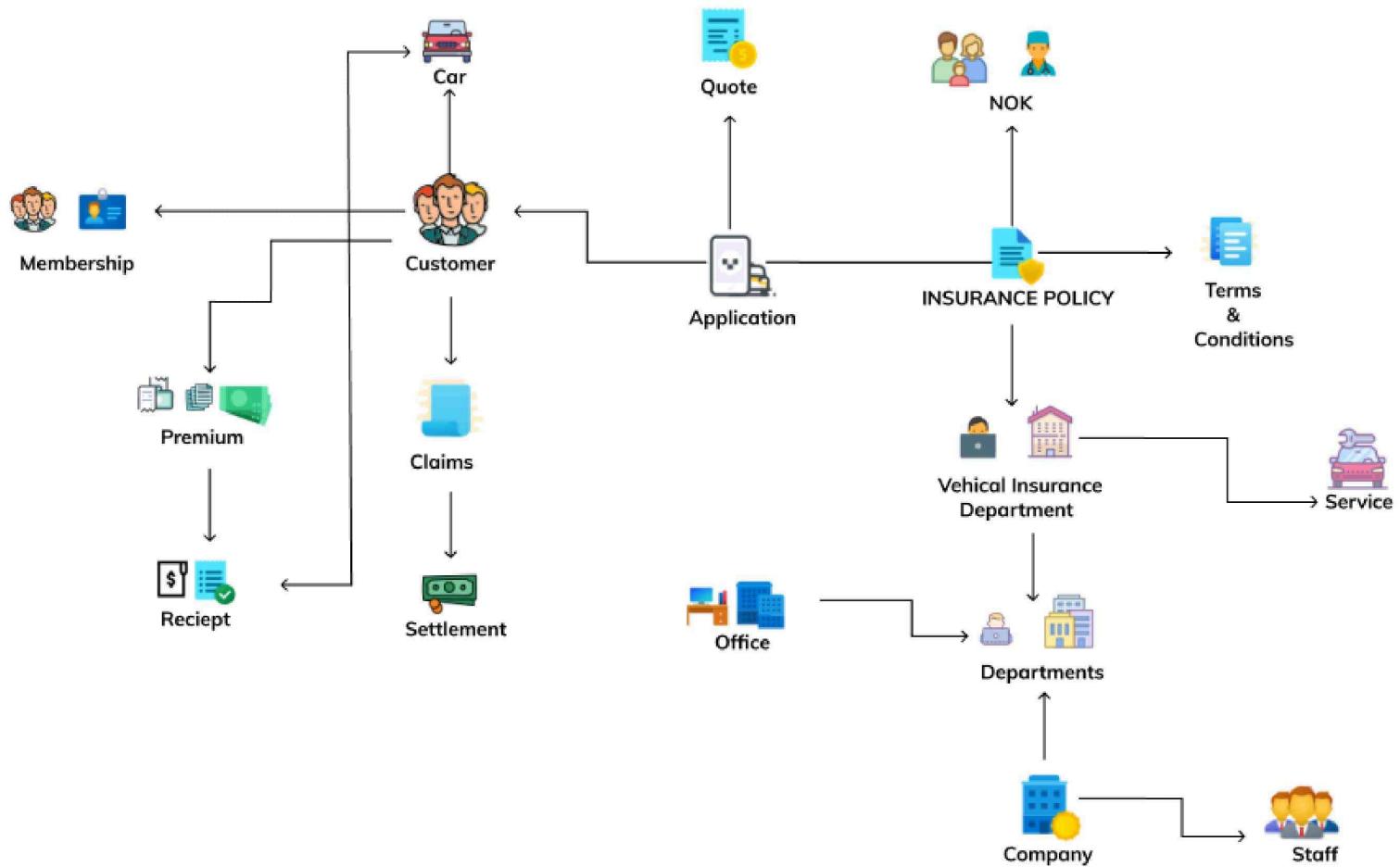
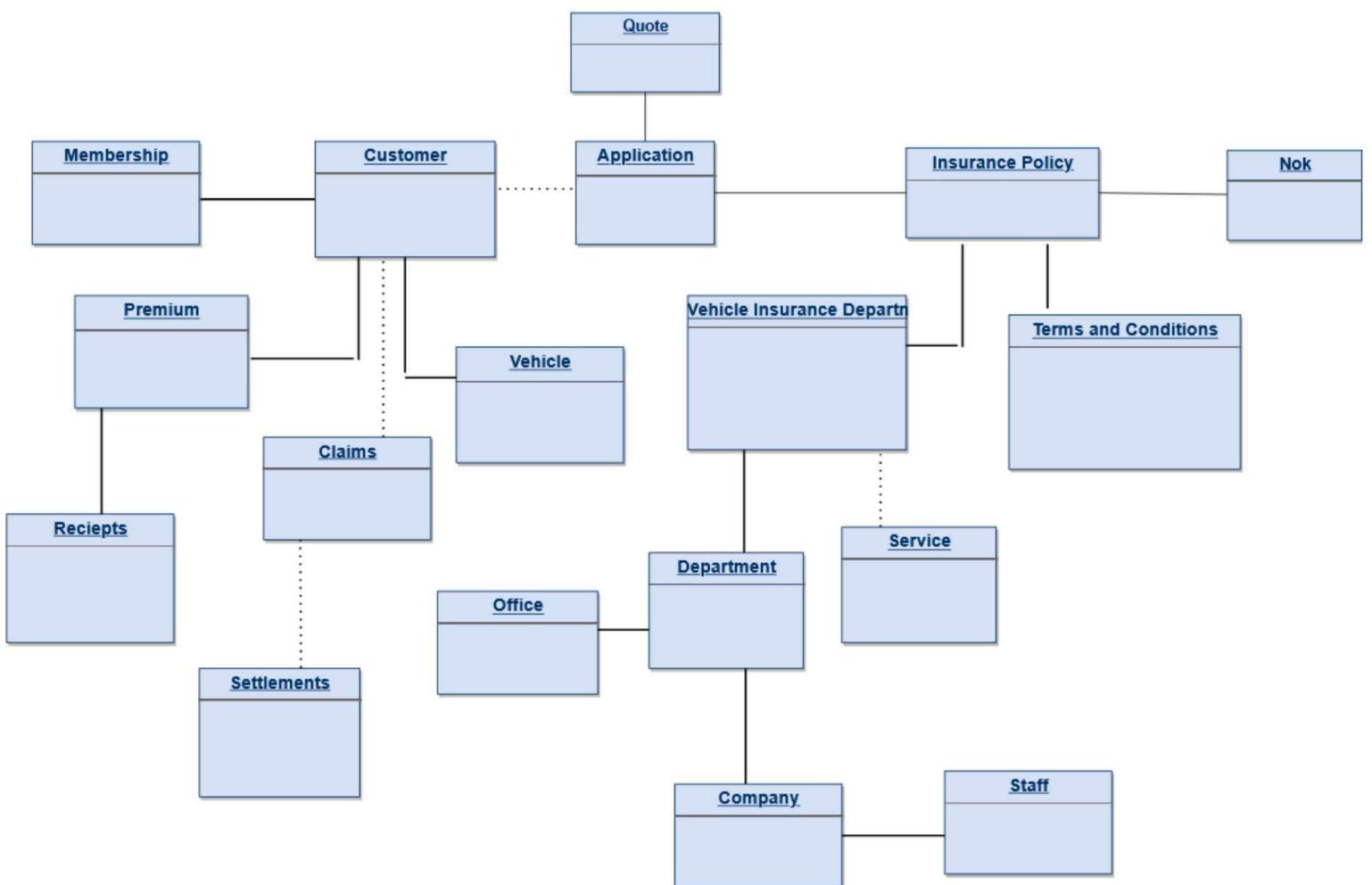


Fig 4: Conceptual data model of Car insurance (with entity types)



## Part B: Logical Data Model:

### Introduction:

Part B of this report includes the design of a 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
- Whether 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
- Whether entity should become a database view or a table
- Whether 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 :-

Aggregation is an abstraction that turns relationships 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 models that are based on dependencies between entities and their attributes. Normalization is used with the goal objective to avoid duplication of information in order to safeguard 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 and Oracle as seen in example Table 6, below,

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

Entity type	Attributes	Data type Access	Data type Oracle
CUSTOMER	FIRST_NAME	TEXT	varchar2(size)
	LAST_NAME	TEXT	varchar2(size)
	DATE_OF_BIRTH	DATE/TIME	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	CUSTOMER	Records all the personal details of the customer
2	APPLICATION	Records details of the insurance cover requested by customer
3	QUOTE	Records details of customer potential cost of the insurance product
4	INSURANCE_POLICY	Records details of 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 claims 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 due date of 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 in regard to the policy
20	PRODUCT	Records details of the products offered by insurance company
21	RECEIPT	Details of premium payments to customer
22	INSURANCE_POLICY_COVERAG E	It shows agreement and coverage details

**Table 8: Abbreviation table of attributes names used in LDM**

LDM Attributes	Column Names Abbreviations
CUST_ID	CUSTOMER_IDENTIFICATION
CUST_FNAME	CUSTOMER_FIRST_NAME
CUST_LNAME	CUSTOMER_LAST_NAME
CUST_DOB	CUSTOMER_DATEOFBIRTH
CUST_PPS_NUMBER	CUSTOMER_PERSONALPUBLICSERVICE_NUMBER
STAFF_FNAME	STAFF_FIRSTNAME
STAFF_LNAME	STAFF_LASTNAME
STAFF_PPS_NUMBER	STAFF_PERSONALPUBLICSERVICE_NUMBER
ADMIN_COST	ADMINISTRATION_COST
NOK_ID	NEXTOFKIN_IDENTIFICATION

**Table\_14-LDM 1: CUSTOMER:**

Attributes	Data types Access	Data type Oracle	Primary and foreign keys
CUST_ID	TEXT	VARCHAR2(20)	PK
CUST_FNAME	TEXT	VARCHAR2(10)	
CUST_LNAME	TEXT	VARCHAR2(10)	
CUST_DOB	DATE	DATE	
CUST_GENDER	TEXT	CHAR(2)	
CUST_ADDRESS	TEXT	VARCHAR2(20)	
CUST_MOB_NUMBER	NUMBER	INTEGER	
CUST_EMAIL	TEXT	VARCHAR2(20)	
CUST_PASSPORT_NUMBER	TEXT	VARCHAR2(20)	
CUST_MARITAL_STATUS	TEXT	CHAR(8)	
CUST_PPS_NUMBER	NUMBER	INTEGER	

#### Explanation

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

## Table\_14-LDM 2:APPLICATION:

Attributes	Data types Access	Data type Oracle	Primary and foreign keys
APPLICATION_ID	TEXT	VARCHAR2(20)	PK
CUST_ID	TEXT	VARCHAR2(20)	FK
VEHICLE_ID	TEXT	VARCHAR2(20)	
APPLICATION_STATUS	TEXT	CHAR(8)	
COVERAGE	TEXT	VARCHAR2(20)	

### Explanation

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.

## Table\_14-LDM 3: QUOTE:

ATTRIBUTES	Data types Access	Data type Oracle	Primary and foreign keys
QUOTE_ID	TEXT	VARCHAR2(20)	PK
APPLICATION_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
ISSUE-DATE	DATE/TIME	DATE	
VALID_FROM_DATE	DATE/TIME	DATE	
VALID_TILL_DATE	DATE/TIME	DATE	
DESCRIPTION	TEXT	VARCHAR2(20)	
PRODUCT_ID	TEXT	VARCHAR2(20)	
COVERAGE_LEVEL	TEXT	VARCHAR2(20)	

## EXPLANATION

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 is a foreign key linking the table back to the respective entities

## Table\_14-LDM 4:INSURANCE\_POLICY:

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
AGREEMENT_ID	Text	VARCHAR2(20)	PK
APPLICATION_ID	Text	VARCHAR2(20)	FK
CUST_ID	Text	VARCHAR2(20)	FK
DEPARTMENT_NAME	Text	DATE	
POLICY_NUMBER	Text	DATE	
START_DATE	DATE/TIME	VARCHAR2(100)	
EXPIRY_DATE	DATE/TIME	VARCHAR2(20)	
TERM_CONDITION_DESCRIPTION	Text	VARCHAR2(20)	

## Explanation

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 are linked to the other corresponding entities through their foreign keys.

## Table\_14-LDM 5:PREMIUM\_PAYMENT:

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
PREMIUM_PAYMENT_ID	TEXT	VARCHAR2(20)	PK
CUST_ID	TEXT	VARCHAR2(20)	FK
POLICY_NUMBER	TEXT	VARCHAR2(20)	
PREMIUM_PAYMENT_SCHEDULE	DATE/TIME	DATE	
PREMIUM_PAYMENT_AMOUNT	NUMBER	INTEGER	
RECEIPT_ID	TEXT	VARCHAR2(20)	

### EXPLANATION

The PREMIUM\_PAYMENT attributes record all the essential policy premium payments details of the customer. The PREM\_PAYMENT\_ID is the unique primary key and the CUST\_ID is the Foreign key linking tables to CUSTOMER entities.

## Table\_14-LDM 6: VEHICLE:

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
VEHICLE_ID	TEXT	VARCHAR2(20)	PK
CUST_ID	TEXT	VARCHAR2(20)	FK
POLICY_ID	TEXT	VARCHAR2(20)	
DEPENDENT_NOK_ID	TEXT	VARCHAR2(20)	
VEHICLE_REGISTRATION_NUMBER	TEXT	VARCHAR2(20)	
VEHICLE_VALUE	NUMBER	INTEGER	
VEHICLE_TYPE	TEXT	VARCHAR2(20)	

VEHICLE_SIZE	NUMBER	INTEGER	
VEHICLE_NUMBER_OF_SEAT	NUMBER	INTEGER	
VEHICLE_MANUFACTURER	TEXT	VARCHAR2(20)	
VEHICLE_ENGINE_NUMBER	NUMBER	INTEGER	
VEHICLE_CHASIS_NUMBER	NUMBER	INTEGER	
VEHICLE_NUMBER	TEXT	VARCHAR2(20)	
VEHICLE_MODEL_NUMBER	TEXT	VARCHAR2(20)	

### EXPLANATION

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 table to CUSTOMER entities.

**Table\_14-LDM 7: CLAIM**

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
CLAIM_ID	TEXT	VARCHAR2(20)	PK
CUST_ID	TEXT	VARCHAR2(20)	FK
AGREEMENT_ID	TEXT	VARCHAR2(20)	
CLAIM_AMOUNT	NUMBER	INTEGER	
INCIDENT_ID	TEXT	VARCHAR2(20)	
DAMAGE_TYPE	TEXT	VARCHAR2(20)	
DATE_OF_CLAIM	DATE/TIME	DATE	
CLAIM_STATUS	TEXT	CHAR(10)	

### EXPLANATION

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 foreign key linking tables to CUSTOMER entities.

## Table\_14-LDM 8: CLAIM\_SETTLEMENT:

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
CLAIM_SETTELEMENT_ID	TEXT	VARCHAR2(20)	PK
CLAIM_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
VEHICLE_ID	TEXT	VARCHAR2(20)	
DATA_SETTLED	DATE/TIME	DATE	
AMOUNT_PAID	NUMBER	INTEGER	
COVERAGE_ID	TEXT	VARCHAR2(20)	

### EXPLANATION

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.

## Table\_14 -LDM 9: STAFF:

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
STAFF_ID	TEXT	VARCHAR2(20)	PK
COMPANY_NAME	TEXT	VARCHAR2(20)	FK
STAFF_FNAME	TEXT	VARCHAR2(10)	
STAFF_LNAME	TEXT	VARCHAR2(10)	
STAFF_ADDRESS	TEXT	VARCHAR2(20)	
STAFF_CONTACT	NUMBER	INTEGER	
STAFF_GENDER	TEXT	CHAR(2)	

STAFF_MARTIAL_STATUS	TEXT	CHAR(8)	
STAFF_NATIONALITY	TEXT	CHAR(15)	
STAFF_QUALIFICATION	TEXT	VARCHAR2(20)	
STAFF_ALLOWANCE	NUMBER	INTEGER	
STAFF_PPS_NUMBER	NUMBER	INTEGER	

### EXPLANATION

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.

**Table\_14-LDM 10: DEPARTMENT:**

ATTRIBUTES	Data type Access	Data type Oracle	Primary and foreign keys
DEPARTMENT_NAME	TEXT	VARCHAR2(20)	PK
COMPANY	TEXT	VARCHAR2(20)	FK
OFFICE	TEXT	VARCHAR2(20)	
CONTACT_INFORMATION	TEXT	VARCHAR2(20)	
DEPARTMENT_STAFF	TEXT	VARCHAR2(20)	
DEPARTMENT_LEADER	TEXT	VARCHAR2(20)	

### EXPLANATION

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.

**Table\_14-LDM 11: OFFICE:**

<b>ATTRIBUTES</b>	<b>Data type Access</b>	<b>Data type Oracle</b>	<b>Primary and foreign keys</b>
OFFICE_NAME	TEXT	VARCHAR2(20)	PK
DEPARTMENT_NAME	TEXT	VARCHAR2(20)	FK
COMPANY_NAME	TEXT	VARCHAR2(20)	FK
OFFICE_LEADER	TEXT	VARCHAR2(20)	
CONTACT_INFORMATION	TEXT	VARCHAR2(20)	
ADDRESS	TEXT	VARCHAR2(20)	
ADMIN_COST	NUMBER	INTEGER	
STAFF	TEXT	VARCHAR2(50)	

**EXPLANATION**

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.

**Table\_14-LDM 12: MEMBERSHIP:**

<b>ATTRIBUTES</b>	<b>Data type Access</b>	<b>Data type Oracle</b>	<b>Primary and foreign keys</b>
MEMBERSHIP_ID	TEXT	VARCHAR2(20)	PK
CUST_ID	TEXT	VARCHAR2(20)	FK
MEMBERSHIP_TYPE	TEXT	CHAR(15)	
ORGANISATION_CONTACT	TEXT	VARCHAR2(20)	

**EXPLANATION**

The MEMBERSHIP attributes record all the essential membership details available for insured customers. 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.

## Table\_14-LDM 13: VEHICLE\_SERVICE:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
VEHICLE_SERVICE	TEXT	VARCHAR2(20)	PK
VEHICLE_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
DEPARTMENT-NAME	TEXT	CHAR(20)	
VEHICLE_SERVICE_ADDRSS	TEXT	VARCHAR2(20)	
VEHICLE_SERVICE_CONTACT	TEXT	VARCHAR2(20)	
VEHICLE_SERVICE_INCHARGE	TEXT	CHAR(20)	
VEHICLE_SERVICE_TYPE	TEXT	VARCHAR2(20)	

### EXPLANATION

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.

## Table\_14-LDM 14:NOK:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
NOK_ID	TEXT	VARCHAR2(20)	PK
AGREEMENT_ID	TEXT	VARCHAR2(20)	FK
APPLICATION-ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK

NOK_NAME	TEXT	VARCHAR2(20)	
NOK_ADDRESS	TEXT	VARCHAR2(20)	
NOK_PHONE_NUMBER	NUMBER	INTEGER	
NOK_MARTIAL-STATUS	TEXT	CHAR(8)	
NOK_GENDER	TEXT	CHAR(2)	

### Explanation

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.

**Table\_14-LDM 15: INSURANCE\_COMPANY:**

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
COMPANY_NAME	TEXT	VARCHAR2(20)	PK
COMPANY_ADDRESS	TEXT	VARCHAR2(20)	
COMPANY_CONTACT_NUMBER	NUMBER	INTEGER	
COMPANY_FAX	NUMBER	VARCHAR2(20)	
COMPANY_EMAIL	TEXT	VARCHAR2(20)	
COMPANY_WEBSITE	TEXT	VARCHAR2(20)	
COMPANY_LOCATION	TEXT	VARCHAR2(20)	
COMPANY_DEPARTMENT_NAME	TEXT	VARCHAR2(20)	
COMPANY_OFFICE_NAME	TEXT	VARCHAR2(20)	

### Explanation

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

## Table\_14-LDM 16: POLICY\_RENEWABLE:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
POLICY_RENEWABLE_ID	TEXT	VARCHAR2(20)	PK
AGREEMENT_ID	TEXT	VARCHAR2(20)	FK
APPLICATION_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
DATE_OF_RENEWAL	DATE	DATE	
TYPE_OF_RENEWAL	TEXT	CHAR(15)	

### Explanation

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.

## Table\_14-LDM 17: INCIDENT:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
INCIDENT_ID	TEXT	VARCHAR2(20)	PK
INCIDENT_TYPE	TEXT	VARCHAR2(30)	
INCIDENT_DATE	DATE	DATE	
DESCRIPTION	TEXT	VARCHAR2(100)	

### Explanation

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.

### Table\_14-LDM 18: INCIDENT\_REPORT:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
INCIDENT_REPORT_ID	TEXT	VARCHAR2(20)	PK
INCIDENT_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
INCIDENT_INSPECTOR	TEXT	VARCHAR2(20)	
INCIDENT_COST	NUMBER	INTEGER	
INCIDENT_TYPE	TEXT	CHAR(10)	
INCIDENT_REPORT_DESCRIPTION	TEXT	VARCHAR2(100)	

#### Explanation

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.

### Table\_14-LDM 19: COVERAGE:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
COVERAGE_ID	TEXT	VARCHAR2(20)	PK
COMPANY_NAME	TEXT	VARCHAR2(20)	FK
COVERAGE_AMOUNT	NUMBER	INTEGER	
COVERAGE_TYPE	TEXT	CHAR(10)	
COVERAGE_LEVEL	TEXT	CHAR(15)	

PRODUCT_ID	TEXT	VARCHAR2(20)	
COVERAGE_DESCRIPTION	TEXT	VARCHAR2(100)	
COVERAGE_TERMS	TEXT	VARCHAR2(50)	

### Explanation

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.

**Table\_14-LDM 20: PRODUCT:**

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
PRODUCT_NUMBER	TEXT	VARCHAR2(20)	PK
COMPANY_NAME	TEXT	VARCHAR2(20)	FK
PRODUCT_PRICE	NUMBER	INTEGER	
PRODUCT_TYPE	TEXT	CHAR(15)	

### Explanation

The PRODUCT attributes record all the essential company products details offered by the Insurance company. The PRODUCT\_NUMBER is the unique primary key and COMPANY\_NAME is the foreign key linking table to COMPANY entities.

## Table\_14-LDM 21: RECEIPT:

Attributes	Data type Access	Data type Oracle	Primary and foreign keys
RECEIPT_ID	TEXT	VARCHAR2(20)	PK
PREMIUM_PAYMENT_ID	TEXT	VARCHAR2(20)	FK
CUST_ID	TEXT	VARCHAR2(20)	FK
COST	NUMBER	INTEGER	
TIME	DATE	DATE	

### Explanation

The RECEIPT attributes record all the essential payments done by CUSTOMERS to insurance company. The RECEIPT\_ID is the unique primary key and PREMIUM\_PAYMENT\_ID and CUST\_ID are Foreign keys linking tables to their respective entities.

## Table\_14-LDM 22: INSURANCE\_POLICY\_COVERAGE:

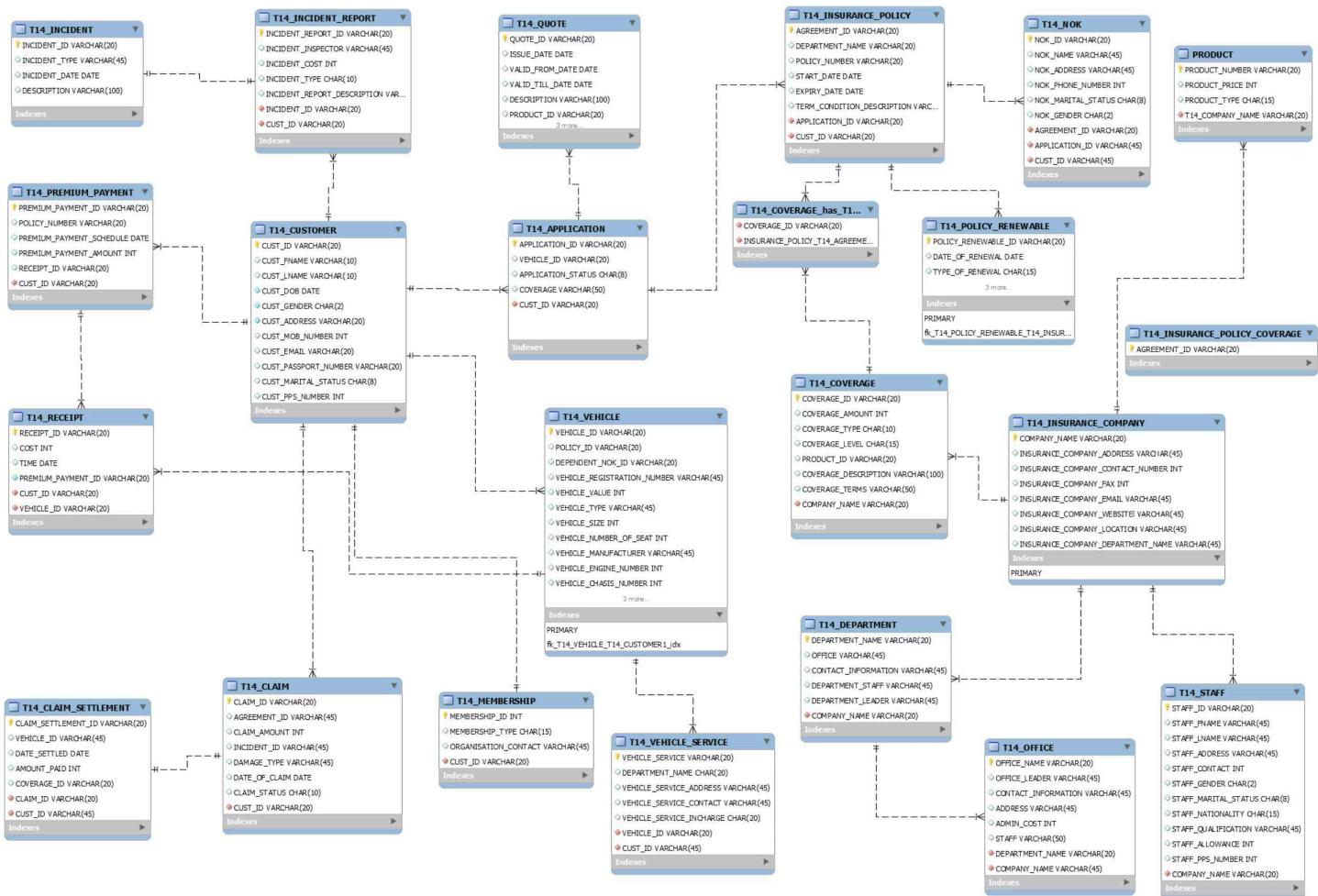
Attributes	Data type Access	Data type Oracle	Primary and foreign keys
AGREEMENT_ID	TEXT	VARCHAR2(20)	PK
COVERAGE_ID	TEXT	VARCHAR2(20)	FK

### Explanation

The INSURANCE\_POLICY\_COVERAGE records details of the Vehicle policy that entails Terms Conditions of the Contract. AGREEMENT\_ID is the unique primary key and COVERAGE\_ID is the Foreign key linking this table to 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



## Part C: Physical Data Model and Database Design (25%)

### Introduction:

When building an entity relationship (ER) model we tend to use it to later build different physical models of database types, see Figure 5. 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.

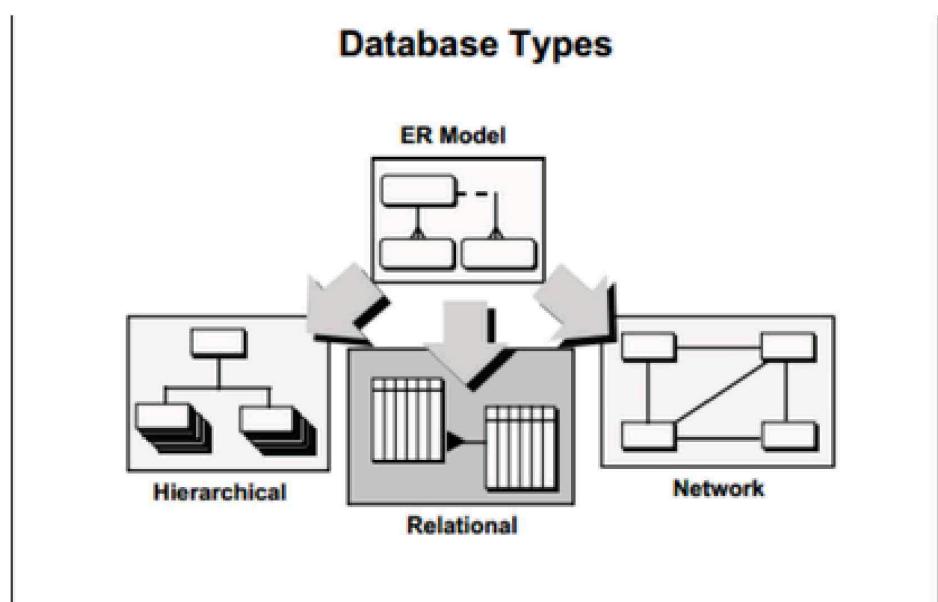


Figure 5: Database types (Source: Data Modelling and Relational Database Design (Speelpenning , et al., 2001))

### NORMALIZATION

It is a relationship database concept and is done in process of building ER. If the correct entity model is being build will conform to the rules of normalization. Each rule has corresponding data model interpretation, which can be used to validate placement of attributes in ER model, see Figure 6 (Speelpenning , et al., 2001).

## Normalization Rules

Normal Form Rule	Description
First Normal Form	All attributes are single valued.
Second Normal Form (2NF)	An attribute must be dependent upon entity's entire unique identifier.
Third Normal Form (3NF)	No non-UID attribute can be dependent on another non-UID attribute.

**"A normalized entity-relationship data model automatically translates into a normalized relational database design"**

**"Third normal form is the generally accepted goal for a database design that eliminated redundancy"**

Figure 6: Normalization rules (Source: Data Modelling and Relational Database Design (Speelpenning , et al., 2001))

Table 9: 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, see Figure 7. This change also means change of terminology (Speelpenning , et al., 2001).

## 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. Example to get best performance in Oracle RDBMS, see Figure 8. 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 of oracle 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, see Figure 9.

These are some concerns using data types in Oracle RDBMS building a physical model (Speelpenning , et al., 2001).

- 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 and MS Access RDBMS.

## Most Commonly-Used Oracle Data Types

Here you can see most used Oracle data types (Speelpenning , et al., 2001):

- 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.

- VARCHAR2 (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. It replaces the old Oracle version 6 CHAR data type.

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.

- NUMBER This data type is used for numerical values, with or without a decimal, of virtually unlimited size. This data type is used for data on which calculation or sorting should be possible. Avoid its use for numbers like a phone number, where the value does not have anymeaning.

Typical use: amount of money, quantities, generated unique key values.

- 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.

- LONG Character data of variable length up to 2 gigabytes. Obsolete since Oracle8. Was used for ASCII text files where you do not need to search using the wildcard or substring functionality. Use CLOB data type instead.

Typical use: for storing the source code of HTML pages.

- LONG RAW Raw binary data of variable length up to 2 gigabytes. Obsolete since Oracle8. Was used for large object types where the database should not try to interpret the data. Use BLOB data type instead.

Typical use: images or video clips.

- CLOB Character large object type. Replaces LONG. Major difference: a table can have more than one CLOB column where there was only one LONG allowed. Maximum size is 4gigabytes.

Typical use: see LONG.

- BLOB Character large object type replaces LONG RAW. Major difference: a table can have more than one BLOB column where there was only one LONGRAW allowed. Maximum size is 4 gigabytes.

Typical use: see LONG RAW.

- BFILE Contains a locator to a large binary file stored outside the database to enable byte stream I/O access to external LOBs residing on the database server.

Typical use: movies

## Most Commonly-Used MS Access Data Types

Type of Data	Description	Size
Short Text	Text or combinations of text and numbers, including numbers that do not require calculating (e.g. phone numbers).	Up to 255 characters.
Long Text	Lengthy text or combinations of text and numbers.	Up to 63, 999 characters.
Number	Numeric data used in mathematical calculations.	1, 2, 4, or 8 bytes (16 bytes if set to Replication ID).
Date/Time	Date and time values for the years 100 through 9999.	8 bytes
Currency	Currency values and numeric data used in mathematical calculations involving data with one to four decimal places.	8 bytes
AutoNumber	A unique sequential (incremented by 1) number or random number assigned by Microsoft Access whenever a new record is added to a table.	4 bytes (16 bytes if set to Replication ID).
Yes/No	Yes and No values and fields that contain only one of two values (Yes/No, True/False, or On/Off).	1 bit.
Hyperlink	Text or combinations of text and numbers stored as text and used as a hyperlink address.	Up to 8,192 (each part of a Hyperlink data type can contain up to 2048 characters).

## 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.

In oracle there are three different system privileges:

DBA: have all the privileges, it is the highest system privileges and only the DBA can create the database structure.

RESOURCE: Users with resource privilege can create an entity in database, but it cannot to create and change database structure

CONNECT: Connect privilege is the least privilege of database, in oracle 10G connect privilege only can login database and create session.

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.

## Oracle 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.

## Changes made from LDM to PDM (MS Access - Oracle)

We made some syntax changes to LDM model for MS Access and Oracle database. Change from the use of "text" for MS Access and changed "VARCHAR" to "VARCHAR2" or "CHAR". We added another entity INSURANCE\_POLICY\_COVERAGE as to better performance of database model and normalization. As we had all entities and attributes finished we made also data type definition report, with description of entities, attributes within each entities, attributes definitions and data types for MS Access and Oracle database.

## PDM for vehicle insurance service

We have prepared two codes for creation of database in Oracle SQL and MS Access, see Table 10. With ERwin Physycal data model see Appendix 7 and data type identification report see Appendix 4.

**Table 10: PDM code for Oracle and MS Access**

**TABLE INCIDENT :**

```
CREATE TABLE t14 INCIDENT
(
Incident_Id varchar(25) NOT NULL ,
Incident_Type VARCHAR(30) NULL ,
Incident_Date DATE NOT NULL ,
Description VARCHAR(100) NULL ,
CONSTRAINT XPKINCIDENT_17 PRIMARY KEY (Incident_Id)
);
CREATE UNIQUE INDEX XPKINCIDENT_17 ON t14 INCIDENT
(Incident_Id ASC);
```

**TABLE CUSTOMER :**

```
CREATE TABLE t14 CUSTOMER
(
Cust_Id VARCHAR(20) NOT NULL ,
Cust_FName VARCHAR(10) NOT NULL ,
Cust_LName VARCHAR(10) NOT NULL ,
Cust_DOB DATE NOT NULL ,
Cust_Gender CHAR(2) NOT NULL ,
Cust_Address VARCHAR(20) NOT NULL ,
Cust_MOB_Number INTEGER NOT NULL ,
Cust_Email VARCHAR(20) NULL ,
Cust_Passport_Number VARCHAR(20) NULL ,
Cust_Marital_Status CHAR(8) NULL ,
Cust_PPS_Number INTEGER NULL ,
CONSTRAINT XPKCUSTOMER_1 PRIMARY KEY (Cust_Id)
);
CREATE UNIQUE INDEX XPKCUSTOMER_1 ON t14 CUSTOMER
(Cust_Id ASC);
```

## INCIDENT\_REPORT :

```

CREATE TABLE t14 INCIDENT_REPORT
(
    Incident_Report_Id VARCHAR(20) NOT NULL ,
    Incident_Type CHAR(10) NULL ,
    Incident_Inspector VARCHAR(20) NULL ,
    Incident_Cost INTEGER NULL ,
    Incident_Report_Description VARCHAR(100) NULL ,
    Incident_Id VARCHAR(20) NOT NULL ,
    Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKINCIDENT_REPORT_18 PRIMARY KEY
    (Incident_Report_Id,Incident_Id,Cust_Id),
    CONSTRAINT R_83 FOREIGN KEY (Incident_Id) REFERENCES
    t14 INCIDENT (Incident_Id),
    CONSTRAINT R_86 FOREIGN KEY (Cust_Id) REFERENCES
    t14 CUSTOMER (Cust_Id)
);
CREATE UNIQUE INDEX XPKINCIDENT_REPORT_18 ON
t14 INCIDENT REPORT (Incident_Report_Id ASC,Incident_Id
ASC,Cust_Id ASC);

```

## INSURANCE\_COMPANY :

```

CREATE TABLE t14 INSURANCE_COMPANY
(
    Company_Name VARCHAR(20) NOT NULL ,
    Company_Address VARCHAR(20) NULL ,
    Company_Contact_Number INTEGER NULL ,
    Company_Fax INTEGER NULL ,
    Company_Email VARCHAR(20) NULL ,
    Company_Website VARCHAR(20) NULL ,
    Company_Location VARCHAR(20) NULL ,
    Company_Department_Name VARCHAR(20) NULL ,
    Company_Office_Name VARCHAR(20) NULL ,
    CONSTRAINT XPKINSURANCE_COMPANY_15 PRIMARY
    KEY(Company_Name)
);
CREATE UNIQUE INDEX XPKt14 INSURANCE COMPANY_15 ON
t14 INSURANCE COMPANY(Company_Name ASC);

```

## DEPARTMENT :

```

CREATE TABLE t14_DEPARTMENT
(
Department_Name VARCHAR(20) NOT NULL ,
Department_ID CHAR(18) NOT NULL ,
Department_Staff CHAR(18) NULL ,
Department_Offices CHAR(18) NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKDEPARTMENT PRIMARY
KEY(Department_Name,Department_ID,Company_Name),
CONSTRAINT R_56 FOREIGN KEY (Company_Name)REFERENCES
t14_insurance_company( Company_Name)
);
CREATE UNIQUE INDEX XPKDEPARTMENT ON t14_DEPARTMENT
(Department_Name ASC,Department_ID ASC,Company_Name ASC);

```

## VEHICLE\_SERVICE :

```

CREATE TABLE t14_VEHICLE_SERVICE
(
Vehicle_Service_Company_Name VARCHAR(20) NOT NULL ,
Vehicle_Service_Address VARCHAR(20) NULL ,
Vehicle_Service_Contact VARCHAR(20) NULL ,
Vehicle_Service_Incharge VARCHAR(20) NULL ,
Vehicle_Service_Type VARCHAR(20) NULL ,
Department_Name VARCHAR(20) NOT NULL ,
Vehicle_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKVEHICLE_SERVICE PRIMARY
KEY(Vehicle_Service_Company_Name),
CONSTRAINT R_50 FOREIGN KEY (Department_Name) REFERENCES
t14_department (Department_Name),
FOREIGN KEY ( Vehicle_Id) REFERENCES t14_Vehicle(Vehicle_Id)
);
drop table t14_VEHICLE_SERVICE;
CREATE UNIQUE INDEX XPKVEHICLE_SERVICE ON
t14_VEHICLE_SERVICE
(Vehicle_Service_Company_Name ASC,Department_Name ASC);

```

## VEHICLE :

```

CREATE TABLE t14_VEHICLE
(
Vehicle_Id VARCHAR(20) NOT NULL ,
Policy_Id VARCHAR(20) NULL ,
Dependent_NOK_Id VARCHAR(20) NULL ,
Vehicle_Registration_Number VARCHAR(20) NOT NULL ,
Vehicle_Value INTEGER NULL ,
Vehicle_Type VARCHAR(20) NOT NULL ,
Vehicle_Size INTEGER NULL ,
Vehicle_Number_Of_Seat INTEGER NULL ,
Vehicle_Manufacturer VARCHAR(20) NULL ,
Vehicle_Engine_Number INTEGER NULL ,
Vehicle_Chassis_Number INTEGER NULL ,
Vehicle_Number VARCHAR(20) NULL ,
Vehicle_Model_Number VARCHAR(20) NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKVEHICLE_6 PRIMARY KEY (Vehicle_Id,Cust_Id),
CONSTRAINT R_92 FOREIGN KEY (Cust_Id) REFERENCES
t14_CUSTOMER(Cust_Id)
);
CREATE UNIQUE INDEX XPKVEHICLE_6 ON t14_VEHICLE (Vehicle_Id ASC,Cust_Id
ASC);

```

## PREMIUM\_PAYMENT :

```

CREATE TABLE t14_PREMIUM_PAYMENT
(
Premium_Payment_Id VARCHAR(20) NOT NULL ,
Policy_Number VARCHAR(20) NOT NULL ,
Premium_Payment_Amount INTEGER NOT NULL ,
Premium_Payment_Schedule DATE NOT NULL ,
Receipt_Id VARCHAR(20) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKPREMIUM_PAYMENT_5 PRIMARY
KEY(Premium_Payment_Id,Cust_Id),
CONSTRAINT R_85 FOREIGN KEY (Cust_Id) REFERENCES
t14_CUSTOMER(Cust_Id)
);
CREATE UNIQUE INDEX XPKPREMIUM_PAYMENT_5 ON
t14_PREMIUM_PAYMENT(Premium_Payment_Id ASC,Cust_Id ASC);

```

## RECEIPT :

```

CREATE TABLE t14_RECEIPT
(
Receipt_Id VARCHAR(20) NOT NULL ,
Time DATE NOT NULL ,
Cost INTEGER NOT NULL ,
Premium_Payment_Id VARCHAR(20) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
Vehicle_Id VARCHAR(20) NOT NULL,
CONSTRAINT XPKRECEIPT_21 PRIMARY
KEY(Receipt_Id,Premium_Payment_Id,Cust_Id),
CONSTRAINT R_84 FOREIGN KEY (Premium_Payment_Id, Cust_Id)
REFERENCES t14_PREMIUM_PAYMENT (Premium_Payment_Id, Cust_Id),
FOREIGN KEY (Vehicle_Id)
REFERENCES t14_vehicle(Vehicle_Id)
);
CREATE UNIQUE INDEX XPKRECEIPT_21 ON t14_RECEIPT
(Receipt_Id ASC,Premium_Payment_Id ASC,Cust_Id ASC);

```

## APPLICATION :

```

CREATE TABLE t14_APPLICATION
(
Application_Id VARCHAR(20) NOT NULL ,
Vehicle_Id VARCHAR(20) NOT NULL ,
Application_Status CHAR(8) NOT NULL ,
Coverage VARCHAR(50) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKAPPLICATION_2 PRIMARY KEY
(Application_Id,Cust_Id),
CONSTRAINT R_93 FOREIGN KEY (Cust_Id) REFERENCES
t14_CUSTOMER(Cust_Id)
);
CREATE UNIQUE INDEX XPKAPPLICATION_2 ON
t14_APPLICATION(Application_Id ASC,Cust_Id ASC);

```

## INSURANCE\_POLICY :

```

CREATE TABLE t14_INSURANCE_POLICY
(
    Agreement_id VARCHAR(20) NOT NULL ,
    Department_Name VARCHAR(20) NULL ,
    Policy_Number VARCHAR(20) NULL ,
    Start_Date DATE NULL ,
    Expiry_Date DATE NULL ,
    Term_Condition_Description VARCHAR(100) NULL ,
    Application_Id VARCHAR(20) NOT NULL ,
    Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKINSURANCE_POLICY_4 PRIMARY
    KEY(Agreement_id,Application_Id,Cust_Id),
    CONSTRAINT R_95 FOREIGN KEY (Application_Id, Cust_Id) REFERENCES
    t14_APPLICATION (Application_Id, Cust_Id)
);
CREATE UNIQUE INDEX XPKINSURANCE_POLICY_4 ON
t14_INSURANCE_POLICY
(Agreement_id ASC,Application_Id ASC,Cust_Id ASC);

```

## POLICY\_RENEWABLE :

```

CREATE TABLE t14_POLICY_RENEWABLE
(
    Policy_Renewable_Id VARCHAR(20) NOT NULL ,
    Date_Of_Renewal DATE NOT NULL ,
    Type_Of_Renewal CHAR(15) NOT NULL ,
    Agreement_id VARCHAR(20) NOT NULL ,
    Application_Id VARCHAR(20) NOT NULL ,
    Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKPOLICY_RENEWABLE_16 PRIMARY
    KEY(Policy_Renewable_Id,Agreement_id,Application_Id,Cust_Id),
    CONSTRAINT R_101 FOREIGN KEY (Agreement_id, Application_Id,
    Cust_Id)
    REFERENCES t14_INSURANCE_POLICY (Agreement_id, Application_Id,
    Cust_Id)
);
CREATE UNIQUE INDEX XPKPOLICY_RENEWABLE_16 ON
t14_POLICY_RENEWABLE
(Policy_Renewable_Id ASC,Agreement_id ASC,Application_Id
ASC,Cust_Id ASC);

```

## MEMBERSHIP :

```

CREATE TABLE t14_MEMBERSHIP
(
    Membership_Id VARCHAR(20) NOT NULL ,
    Membership_Type CHAR(15) NOT NULL ,
    Organisation_Contact VARCHAR(20) NULL ,
    Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKMEMBERSHIP_12 PRIMARY
    KEY(Membership_Id,Cust_Id),
    CONSTRAINT R_91 FOREIGN KEY (Cust_Id) REFERENCES
    t14_CUSTOMER(Cust_Id)
);
CREATE UNIQUE INDEX XPKMEMBERSHIP_12 ON
t14_MEMBERSHIP(Membership_Id ASC,Cust_Id ASC);

```

## QUOTE :

```

CREATE TABLE t14_QUOTE
(
    Quote_Id VARCHAR(20) NOT NULL ,
    Issue_Date DATE NOT NULL ,
    Valid_From_Date DATE NOT NULL ,
    Valid_Till_Date DATE NOT NULL ,
    Description VARCHAR(100) NULL ,
    Product_Id VARCHAR(20) NOT NULL ,
    Coverage_Level VARCHAR(20) NOT NULL ,
    Application_Id VARCHAR(20) NOT NULL ,
    Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKQUOTE_3 PRIMARY
    KEY(Quote_Id,Application_Id,Cust_Id),
    CONSTRAINT R_94 FOREIGN KEY (Application_Id, Cust_Id)
    REFERENCES
    t14_APPLICATION (Application_Id, Cust_Id)
);
CREATE UNIQUE INDEX XPKQUOTE_3 ON t14_QUOTE
(Quote_Id ASC,Application_Id ASC,Cust_Id ASC);

```

## STAFF :

```

CREATE TABLE t14_STAFF
(
Staff_Id VARCHAR(20) NOT NULL ,
Staff_Fname VARCHAR(10) NULL ,
Staff_LName VARCHAR(10) NULL ,
Staff_Adress VARCHAR(20) NULL ,
Staff_Contact INTEGER NULL ,
Staff_Gender CHAR(2) NULL ,
Staff_Marital_Status CHAR(8) NULL ,
Staff_Nationality CHAR(15) NULL ,
Staff_Qualification VARCHAR(20) NULL ,
Staff_Allowance INTEGER NULL ,
Staff_PPS_Number INTEGER NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKSTAFF_9 PRIMARY KEY (Staff_Id,Company_Name),
CONSTRAINT R_105 FOREIGN KEY (Company_Name) REFERENCES
t14_INSURANCE_COMPANY (Company_Name)
);
CREATE UNIQUE INDEX XPKSTAFF_9 ON t14_STAFF(Staff_Id
ASC,Company_Name ASC);

```

## NOK :

```

CREATE TABLE t14_NOK
(
Nok_Id VARCHAR(20) NOT NULL ,
Nok_Name VARCHAR(20) NULL ,
Nok_Address VARCHAR(20) NULL ,
Nok_Phone_Number INTEGER NULL ,
Nok_Gender CHAR(2) NULL ,
Nok_Marital_Status CHAR(8) NULL ,
Agreement_id VARCHAR(20) NOT NULL ,
Application_Id VARCHAR(20) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKNOK_14 PRIMARY
KEY(Nok_Id,Agreement_id,Application_Id,Cust_Id),
CONSTRAINT R_99 FOREIGN KEY (Agreement_id, Application_Id, Cust_Id)
REFERENCES t14_INSURANCE_POLICY (Agreement_id, Application_Id, Cust_Id)
);
CREATE UNIQUE INDEX XPKNOK_14 ON t14_NOK
(Nok_Id ASC,Agreement_id ASC,Application_Id ASC);

```

## PRODUCT :

```

CREATE TABLE t14_PRODUCT
(
Product_Price INTEGER NULL ,
Product_Type CHAR(15) NULL ,
Product_Number VARCHAR(20) NOT NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKPRODUCT_20 PRIMARY KEY
(Product_Number,Company_Name),
CONSTRAINT R_107 FOREIGN KEY (Company_Name) REFERENCES
t14_INSURANCE_COMPANY (Company_Name)
);
CREATE UNIQUE INDEX XPKPRODUCT_20 ON t14_PRODUCT
(Product_Number ASC,Company_Name ASC);

```

## OFFICE :

```

CREATE TABLE t14_OFFICE
(
Office_Name VARCHAR(20) NOT NULL ,
Office_Leader VARCHAR(20) NOT NULL ,
Contact_Information VARCHAR(20) NOT NULL ,
Address VARCHAR(20) NOT NULL ,
Admin_Cost INTEGER NULL ,
Staff VARCHAR(50) NULL ,
Department_Name VARCHAR(20) NOT NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKOFFICE_11 PRIMARY KEY
(Office_Name,Department_Name,Company_Name),
CONSTRAINT R_104 FOREIGN KEY (Department_Name) REFERENCES
t14_DEPARTMENT (Department_Name) ,
FOREIGN KEY (Company_Name) REFERENCES
t14_INSURANCE_COMPANY (Company_Name)
);
CREATE UNIQUE INDEX XPKOFFICE_11 ON t14_OFFICE
(Office_Name ASC,Department_Name ASC,Company_Name ASC);

```

## COVERAGE :

```

CREATE TABLE t14_COVERAGE
(
Coverage_Id VARCHAR(20) NOT NULL ,
Coverage_Amount INTEGER NOT NULL ,
Coverage_Type CHAR(10) NOT NULL ,
Coverage_Level CHAR(15) NOT NULL ,
Product_Id VARCHAR(20) NOT NULL ,
Coverage_Description VARCHAR(100) NULL ,
Covearge_Terms VARCHAR(50) NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKCOVERAGE_19 PRIMARY KEY
(Coverage_Id,Company_Name),
CONSTRAINT R_102 FOREIGN KEY (Company_Name) REFERENCES
t14_INSURANCE_COMPANY (Company_Name)
);
CREATE UNIQUE INDEX XPKCOVERAGE_19 ON t14_COVERAGE
(Coverage_Id ASC,Company_Name ASC);

```

## INSURANCE\_POLICY\_COVERAGE :

```

CREATE TABLE t14_INSURANCE_POLICY_COVERAGE
(
Agreement_id VARCHAR(20) NOT NULL ,
Application_Id VARCHAR(20) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
Coverage_Id VARCHAR(20) NOT NULL ,
Company_Name VARCHAR(20) NOT NULL ,
CONSTRAINT XPKINSURANCE_POLICY_4_COVERAGE PRIMARY KEY
(Agreement_id,Application_Id,Cust_Id,Coverage_Id,Company_Name),
CONSTRAINT R_97 FOREIGN KEY (Agreement_id, Application_Id, Cust_Id)
REFERENCES t14_INSURANCE_POLICY (Agreement_id, Application_Id,
Cust_Id),
CONSTRAINT R_98 FOREIGN KEY (Coverage_Id, Company_Name)
REFERENCES t14_COVERAGE (Coverage_Id, Company_Name)
);
CREATE UNIQUE INDEX XPKINSURANCE_POLICY_4_COVERAGE ON
t14_INSURANCE_POLICY_COVERAGE
(Agreement_id ASC,Application_Id ASC,Cust_Id ASC,Coverage_Id
ASC,Company_Name ASC);

```

## CLAIM :

```

CREATE TABLE t14_CLAIM
(
Claim_Id VARCHAR(20) NOT NULL ,
Agreement_Id VARCHAR(20) NOT NULL ,
Claim_Amount INTEGER NOT NULL ,
Incident_Id VARCHAR(20) NOT NULL ,
Damage_Type VARCHAR(20) NOT NULL ,
Date_Of_Claim DATE NOT NULL ,
Claim_Status CHAR(10) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKCLAIM_7 PRIMARY KEY (Claim_Id,Cust_Id),
CONSTRAINT R_88 FOREIGN KEY (Cust_Id) REFERENCES
t14_CUSTOMER (Cust_Id)
);
CREATE UNIQUE INDEX XPKCLAIM_7 ON t14_CLAIM
(Claim_Id ASC,Cust_Id ASC);

```

## CLAIM\_SETTLEMENT :

```

CREATE TABLE t14_CLAIM_SETTLEMENT
(
Claim_Settlement_Id VARCHAR(20) NOT NULL ,
Vehicle_Id VARCHAR(20) NOT NULL ,
Date_Settled DATE NOT NULL ,
Amount_Paid INTEGER NOT NULL ,
Coverage_Id VARCHAR(20) NOT NULL ,
Claim_Id VARCHAR(20) NOT NULL ,
Cust_Id VARCHAR(20) NOT NULL ,
CONSTRAINT XPKCLAIM_SETTLEMENT_8 PRIMARY KEY
(Claim_Settlement_Id,Claim_Id,Cust_Id),
CONSTRAINT R_90 FOREIGN KEY (Claim_Id, Cust_Id) REFERENCES
t14_CLAIM
(Claim_Id, Cust_Id)
);
CREATE UNIQUE INDEX XPKCLAIM_SETTLEMENT_8 ON
t14_CLAIM_SETTLEMENT
(Claim_Settlement_Id ASC,Claim_Id ASC,Cust_Id ASC);

```

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

More you see in Table 11.

## Conclusion

We have as class project created database (DB) with all documentations and reports included. Our goal was to create DB for Online vehicle insurance company with code generated for Oracle and MS Access.

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. From physical data model we created a code to be run in Oracle and MS Access data base management system (DBMS). For better understanding for a reader and for our learning we included some theory in each faze we done and documented in project initial document (PID) with reports of progress and work being done.

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

ENTITIES	MODULES		Database administrator
	Customer	Insurance agent	
CUSTOMER	CR	CRUD	R
APPLICATION	R	CRD	CRD
QUOTE	R	CRUD	R
INSURANCE_POLICY	R	CRUD	CRD
PREMIUM_PAYMENT	-	CRUD	RU
VEHICLE	-	CRUD	RU
CLAIM	-	CRUD	RU
CLAIM_SETTLEMENT	R	CR	RU
STAFF	-	R	RU
DEPARTMENT	-	R	RU
OFFICE	R	R	RU
MEMBERSHIP	CR	CRD	RU
VEHICLE_SERVICE	R	CRU	RU
NOK	R	CRUD	RU
INSURANCE_COMPANY	R	R	RU
POLICY_RENEWABLE	R	CRU	RU
INCIDENT	-	R	RU
INCIDENT_REPORT	R	R	RU
COVERAGE	R	CRD	RU
PRODUCT	R	R	RU
RECEIPT	R	CRD	RU

## GRANT CODE

```
CREATE USER lokesh  
IDENTIFIED BY 'loleh123@';
```

```
CREATE USER geetha  
IDENTIFIED BY '1234';
```

```
CREATE USER avinash  
IDENTIFIED BY '1234';
```

```
GRANT select  
ON insurance_company.t14_nok  
TO lokesh;
```

```
grant select,insert  
on insurance_company.t14_customer  
to lokesh;
```

```
GRANT select  
ON insurance_company.t14_insurance_policy  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_claim_settlement  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_coverage  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_receipt  
TO lokesh;
```

```
GRANT select,insert,update,delete  
ON insurance_company.t14_customer  
TO geetha;
```

```
GRANT select,insert  
ON insurance_company.t14_claim_settlement  
TO geetha;
```

```
GRANT select,insert,delete  
ON insurance_company.t14_coverage  
TO geetha;
```

```
GRANT select,insert,update,delete  
ON insurance_company.t14_insurance_policy  
TO geetha;
```

```
GRANT select,insert,delete  
ON insurance_company.t14_receipt  
TO geetha;  
  
GRANT select  
ON insurance_company.t14_customer  
TO avinash;  
  
GRANT select,insert,delete  
ON insurance_company.t14_insurance_policy  
TO avinash;  
  
GRANT select,update  
ON insurance_company.t14_claim_settlement  
TO avinash;  
  
GRANT select,update  
ON insurance_company.t14_coverage  
TO avinash;  
  
GRANT select,update  
ON insurance_company.t14_receipt  
TO avinash;  
  
show grants for avinash;  
  
show grants for geetha;  
  
GRANT select  
ON insurance_company.t14_qu0te  
TO lokesh;  
GRANT select  
ON insurance_company.t14_application  
TO lokesh;  
GRANT select  
ON insurance_company.t14_office  
TO lokesh;  
  
GRANT select,insert  
ON insurance_company.t14_membership  
TO lokesh;  
  
GRANT select  
ON insurance_company.t14_vehicle_service  
TO lokesh;  
  
GRANT select  
ON insurance_company.t14_nok  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_insurance_company  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_policy_renewable  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_incident_report  
TO lokesh;
```

```
GRANT select  
ON insurance_company.t14_product  
TO lokesh;
```

```
show grants for lokesh;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_application  
TO geetha;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_qu0te  
TO geetha;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_premium_payment  
TO geetha;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_vehicle  
TO geetha;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_claim  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_staff  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_department  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_office  
TO geetha;
```

```
GRANT select,insert,delete  
ON insurance_company.t14_membership  
TO geetha;
```

```
GRANT select,insert,update  
ON insurance_company.t14_vehicle_service  
TO geetha;
```

```
GRANT select,insert,delete,update  
ON insurance_company.t14_nok  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_insurance_company  
TO geetha;
```

```
GRANT select,insert,update  
ON insurance_company.t14_policy_renewable  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_incident  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_incident_report  
TO geetha;
```

```
GRANT select  
ON insurance_company.t14_product  
TO geetha;
```

```
show grants for geetha;  
GRANT select,insert,delete  
ON insurance_company.t14_application  
TO avinash;
```

```
GRANT select  
ON insurance_company.t14_qu0te  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_premium_payment  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_vehicle  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_claim  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_staff  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_department  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_office  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_membership  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_vehicle_service  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_nok  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_insurance_company  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_policy_renewable  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_incident  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_incident_report  
TO avinash;
```

```
GRANT select,update  
ON insurance_company.t14_product  
TO avinash;
```

```
show grants for avinash;
```

## Queries :

**1. Retrieve Customer and Vehicle details who has been involved in an incident and claim status is pending**

```
select * from t14_customer inner join t14_vehicle on
t14_vehicle.Cust_Id = t14_customer.Cust_Id where
t14_customer.Cust_Id in
(select t14_claim.Cust_Id from
t14_claim,t14_incident_report where
t14_claim.Cust_Id=t14_incident_report.Cust_Id and
t14_claim.Claim_Status = 'PROCESSING' );
```

Cust_Id	Cust_FName	Cust_LName	Cust_DOB	Cust_Gender	Cust_Address	Cust_MOB_Number	Cust_Email	Cust_Passport_Number	Cust_Marital_Status	Cust_PPS_Number	Vehicle_Id	Policy_Id	Dependent_NOK_Id	Vehicle_Registration_Number
17133306	Lasya	Sahith	1976-01-24	F	24-6,Patna,Bihar	99883355	lasyasahuthidindigal	NULL	Married	NULL	741358	62658	9889	BR 07 YA 7740
17133310	Meruva	Akhila	1992-05-26	F	S45-1,Ladakh,J&K	90143214	akhilameruva	NULL	Married	NULL	152471	62662	9893	JK 01 OG 8504
17133301	Meruva	Venkat	1999-01-30	M	S-1,Dwaraka,Gj	68453214	venkatmeruva	NULL	Single	NULL	741963	62653	9884	GJ 14 DP 8510
17133303	Grandi	Surya	2000-09-13	M	S-541,Ludhiana,Pb	95321464	suryagrandi	NULL	Single	NULL	749852	62655	9886	PJ 09 VI 1400

Vehicle_Value	Vehicle_Type	Vehicle_Size	Vehicle_Number_Of_Seat	Vehicle_Manufacturer	Vehicle_Engine_Number	Vehicle_Chassis_Number	Vehicle_Number	Vehicle_Model_Number	Cust_Id
852014	car	180	6	Wagnor	954782	210453014	810457	EK46UE	17133306
100008	auto	120	4	Piago	970180	546987320	620410	BY698U	17133310
100202	car	180	5	BMW	520639	964175236	951742	VI65C5	17133301
100020	van	280	10	Tata	203541	789412565	753149	BF478U	17133303

2. Retrieve customer details who has premium payment amount greater than the sum of all the customerIds in the database

```
select * from t14_customer where t14_customer.Cust_Id
in
(select Cust_Id from t14_premium_payment where
t14_premium_payment.Premium_Payment_Amount>(sel
ect cast((select sum(Cust_Id) from t14_customer) as
unsigned)) );
```

Cust_Id	Cust_FName	Cust_LName	Cust_DOB	Cust_Gender	Cust_Address	Cust_MOB_Number	Cust_Email	Cust_Passport_Number	Cust_Marital_Status	Cust_PPS_Number
17133311	JRD	Tata	1904-07-29	M	5-1,Bombay,MH	85201401	jrdtata	NULL	Married	NULL
17133312	M	Gopinath	1935-11-13	M	845-25,UP	96325187	mgopinath	NULL	Married	NULL
17133313	Vijay	Maliya	1965-09-05	M	8-25,MP	65470892	vijaymalia	NULL	Married	NULL

3. Retrieve Company details whose number of products is greater than departments, where the departments are located in more than one location

**insert into temp select**

```
t14_insurance_company.Company_Name, count(t14_department.Company_Name), (select count(t14_product.Company_Name) from t14_product
where t14_product.Company_Name = t14_insurance_company.Company_Name group by t14_product.Company_Name)
from t14_insurance_company inner join t14_department on
t14_insurance_company.Company_Name = t14_department.Company_Name group by t14_department.Company_Name;
select * from t14_insurance_company where
Company_Name in (select name from temp where nod < nop
and nod > 1 );
delete from temp;
```

## Assumption

we assumed that different departments in a insurance company located in different locations

Company_Name	Company_Address	Company_Contact_Number	Company_Fax	Company_Email	Company_Website	Company_Location	Company_Department_Name	Company_Office_Name
KOTAK	MUMBAI-4000	78785783	35784	kotak@gmail.com	www.kotak.com	MUMBAI	VEHICLE	MUMBAI BRANCH
RELIANCE	MUMBAI-23877	83444839	43457	reliance@gmai.com	www.reliance.com	MUMBAI	VEHICLE	MAIN BRANCH
ROYAL	DELHI-600097	83487378	448558	royal@gmail.com	www.royal.com	DELHI	VEHICLE	DELHI BRANCH
SBI	MUMBAI-400644	843875	483984	sbi@gmail.com	www.sbi.com	MUMBAI	VEHICLE	MUMBAI BRANCH

4. Select Customers who have more than one Vehicle, where the premium for one of the Vehicles is not paid and it is involved in accident

```

insert into temp1
select
t14_customer.Cust_Id, count(t14_vehicle.Cust_Id), (select
count(t14_receipt.Cust_Id) from t14_receipt where
t14_receipt.Cust_Id=t14_customer.Cust_Id group by
t14_receipt.Cust_Id) from t14_customer
inner join
t14_vehicle on t14_customer.Cust_Id = t14_vehicle.Cust_Id
group by t14_vehicle.Cust_Id;
select * from t14_customer where Cust_Id in
(select name from temp1 where nov>nop and nov > 1);
delete from temp1;

```

Cust_Id	count(t14_vehicle.Cust_Id)	(select count(t14_receipt.Cust_Id) from t14_receipt where t14_receipt.Cust_Id=t14_customer.Cust_Id group by t14_receipt.Cust_Id)
17133293	1	1
17133294	1	1
17133296	1	1
17133297	3	1
17133298	1	1
17133299	1	1
17133300	1	1
17133301	1	1
17133302	3	1
17133303	1	1
17133304	1	NULL
17133305	1	1
17133306	1	1
17133307	1	1
17133308	1	1
17133309	3	1
17133310	1	1
17133311	3	1
17133312	3	1
17133313	3	1

## 5. Select all vehicles which have premium more than its vehicle number.

```
select * from t14_vehicle,t14_receipt,
t14_premium_payment where t14_vehicle.Vehicle_Id
=t14_receipt.Vehicle_Id and
t14_receipt.Premium_Payment_Id =
t14_premium_payment.Premium_Payment_Id and
t14_premium_payment.Premium_Payment_Amount >
t14_vehicle.Vehicle_Number;
```

Vehicle_Id	Policy_Id	Dependent_NOK_Id	Vehicle_Registration_Number	Vehicle_Value	Vehicle_Type	Vehicle_Size	Vehicle_Number_Of_Seat	Vehicle_Manufacturer	Vehicle_Engine_Number	Vehicle_Chassis_Number	Vehicle_Number
258963	62672	98103	DecAir08	410258	Aeroplane	4200	70	Deccan Air	788001	654123987	841234
961978	62676	98107	KingIn07	741259	Aeroplane	4200	70	Kingfisher	963148	74108521	963215
154512	62669	98100	AirIn45	110474	Aeroplane	4200	70	Air India	789656	998965412	786452

## 6. Retrieve Customer details whose Claim Amount is less than Coverage Amount and Claim Amount is greater than Sum of (CLAIM\_SETTLEMENT\_ID, VEHICLE\_ID, CLAIM\_ID, CUST\_ID )

```

select * from t14_customer where Cust_Id in
(select Cust_Id from t14_application
where Application_Id in ( select Application_Id from
t14_insurance_policy_coverage where
Coverage_Id in( select t14_coverage.Coverage_Id from
t14_coverage,t14_claim ,t14_claim_settlement where
t14_coverage.Coverage_Amount > t14_claim.Claim_Amount
and t14_claim.Claim_Amount >
(t14_claim_settlement.Claim_Id+t14_claim_settlement.Claim_Settlement_Id+t14_claim_settlement.Cust_Id+t14_claim_settlement.Vehicle_Id))));
```

Cust_Id	Cust_FName	Cust_LName	Cust_DOB	Cust_Gender	Cust_Address	Cust_MOB_Number	Cust_Email	Cust_Passport_Number	Cust_Marital_Status	Cust_PPS_Number
17133303	Grandi	Surya	2000-09-13	M	5-541,Ludhiana,Pb	95321464	suryagrandi	NULL	Single	NULL
17133301	Meruva	Venkat	1999-01-30	M	5-1,Dwaraka,Gj	68453214	venkatmeruva	NULL	Single	NULL
17133305	Datla	Raju	1979-12-23	M	55-1,Im,Manipur	84543214	rajudatla156	NULL	Married	NULL
17133294	Nikitha	kallakuri	1999-05-28	F	36-1,Batnavilli,Ap	91865432	nikithakk	NULL	single	NULL
17133309	Anitha	s	2001-10-30	F	55-1,Agra,UP	94543234	ankithasch	NULL	Single	NULL
17133297	Dudi	Sruthi	1990-06-19	F	6-5,Banglore,KN	87814637	sruthi1642	NULL	Married	NULL
17133308	Sri	Tejaswini	1987-01-22	F	25-51,Dhimapur,NL	94531214	tejaswiveluduti	NULL	Married	NULL
17133306	Lasya	Sahithi	1976-01-24	F	24-6,Patna,Bihar	99883355	lasyasahithidindigal	NULL	Married	NULL
17133293	Godina	Pranav	2000-03-09	M	36-5,Kadapa,Ap	81878149	godinapranav	NULL	single	NULL
17133310	Meruva	Akhila	1992-05-26	F	545-1,Ladakh,J&K	90143214	akhilameruva	NULL	Married	NULL
17133298	Lolla	kalyani	2002-07-09	F	354-51,Puri,Odisha	87564926	kalyanilolla	NULL	single	NULL
17133296	Golla	Bhanu	1998-03-06	M	3-5,chennai,TN	88149467	Bhanu456	NULL	single	NULL



**THANK  
YOU**