



INDIAN INSTITUTE OF  
INFORMATION  
TECHNOLOGY

## **Database Management System Project**

**TOPIC: Database Design for a Vehicle Insurance Company.**

**Guided by:**

- i) Uma.S
- ii) Supriya.N

# Presented By:

Page | 2

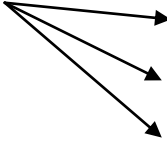
## **PART A**

### **Conceptual Data Model**

- 
- 1.Meghana.N(18bcs053)
  - 2. Varun.M.H(18bcs108)

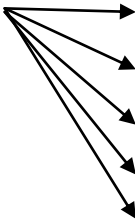
## **PART B**

### **Logical Data Model**

- 
- 1.Bharat Bhushan(18bcs026)
  - 2. K.Chaitanya (18bcs045)
  - 3. Sushanth.B.P (18bcs102)

## **PART C**

### **Physical Data Model**

- 
- 1. Anuj C.S(18bcs009)
  - 2.Daneshwari .Y(18bcs022)
  - 3.Neha.T(18bcs060)
  - 4.Rahul.S(18bcs075)
  - 5.Trishul (18bcs104)

## **Table of content**

<b>1.</b>	<b>Conceptional Data Modelling.....</b>	<b>4</b>
	Entity types .....	5
	Entity Relationships .....	7
	Graphical Representation .....	8
<b>2.</b>	<b>Logical Data Modelling</b>	
	Introduction .....	9
	LDM Tables .....	11
	Graphical Representation .....	26
<b>3.</b>	<b>Physical Data Modelling .....</b>	<b>27</b>
	Normalisation .....	28
<b>4.</b>	<b>Testing Queries .....</b>	<b>29</b>
<b>5.</b>	<b>Grant and Privileges .....</b>	<b>32</b>

## 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.

<u>Rules</u>	<u>Description</u>	<u>Example</u>
Rule 1	All individual entity types must be in capital letters	G1_COSTUMER
Rule 2	An underscore is used to label an entity type with more than one word	G1_INSURANCE_POLICY
Rule 3	Plurals are not used when labelling entity types	G1_STAFF
Rule 4	No abbreviations are used when labelling entity types	G1_QUOTE
Rule 5	The name of every entity type should start with a G1_ to represent Group1	G1_APPLICATION

## Assumptions

To design the Conceptual Data Model (CDM) we certain set of assumptions. These assumptions willhelp shape our model to allow consistency within our design.

<u>Assumptions</u>	<u>Description</u>
Assumption 1	Customer must be a permanent international driving licence
Assumption 2	The online insurance has no physical high-street presence
Assumption 3	The online insurance is given to customers over 18 years of age
Assumption 4	The online insurance needs some driving history of customer
Assumption 5	The online insurance needs to know type of car customer drives
Assumption 6	The online insurance needs to know about insurance history of customer

## Entity types

<u>Entity type</u>	<u>Description</u>
G1_CUSTOMER	Records all the personal details about the customer
G1_APPLICATION	Records details of the insurance cover requested by Customer
G1_QUOTE	Records details of customer potential cost of the insurance product
G1_INSURANCE_POLICY	Records details of Insurance agreement
G1_PREMIUM	Records details of customer payments
G1_VEHICLE	Records details of Vehicle model, cost and registration
G1_CLAIM	Records details of customer claims in case of an incident
G1_SETTLEMENT	Records details of settlement made on claims
G1_STAFF	Records details of employees
G1_DEPARTMENT	Records details of the various departments
G1_OFFICE	Records details of different office locations
G1_MEMBERSHIP	Records details of customer membership
G1_SERVICE	Records details of different car services offered

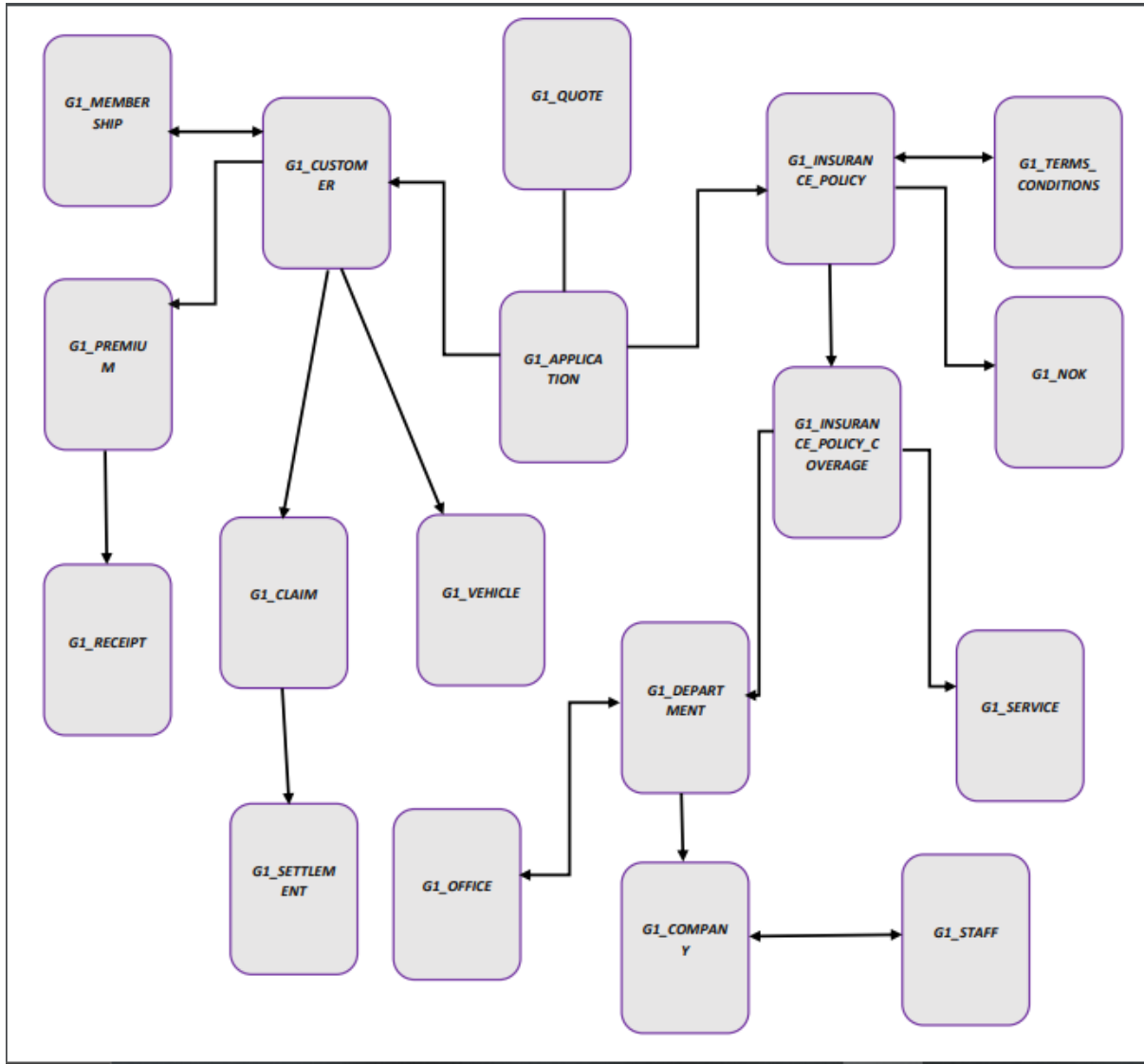
G1_NOK	Records details of the next o kin
G1_TERMS_CONDITIONS	Records all terms and conditions in regard to the policy
G1_INSURANCE_POLICY_COVER AGE	Records details of vehicle insurance cover
G1_RECEIPT	Records details of Receipt of Premiums
G1_COMPANY	Details of the Insurance organization giving the insurance cover

## Entity types related to Entities' Relationship

Page | 7

<u>Entity Type</u>	<u>Related to Entities</u>	<u>Relationship</u>
G1_QUOTE	G1_APPLICATION	One to one
G1_APPLICATION	G1_INSURANCE_POLICY G1_COSTUMER	One to many One to many
G1_COSTUMER	G1_MEMBERSHIP G1_PREMIUM G1_CLAIM G1_VEHICLE	Many to many one to many one to many one to many
G1_INSURANCE_POLICY	G1_VEHICLE_INSURANCE _DEPARTMENT G1_TERMS_CONDITIONS G1_NOK	one to many  Many to many  one to many
G1_PREMIUM	G1_RECEIPT	one to many
G1_CLAIM	G1_SETTLEMENT	one to many
G1_INSURANCE _POLICY_COVERAGE	G1_DEPARMENT  G1_SERVICE	one to many  one to many
G1_DEPARTMENT	G1_OFFICE G1_COMPANY	Many to many one to many
G1_COMPANY	G1_STAFF	Many to many

## Graphical presentation of CDM





# Part B: Logical Data Model

## Introduction

Part B of this report includes design of logical data model for vehicle insurance company ABC Ltd.

We need some changes and improvements to the conceptual data model in part A of this report. Identify all the attributes in old and new entities. Assigned the primary (PK) and foreign keys (FK) and made relationships with them in ER diagram to make a full LDM. Identifying useful variables and data types for this logical data model using the MS SQL workbench.

## Introduction of terms used for constructing LDM

### Elements

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: 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.

## 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 .

**Table 6: Example of data types in MYSQL DMS**

Entity type	Attributes	Data type
G1_CUSTOMER	G1_FIRST_NAME	varchar (size)
	G1_LAST_NAME	varchar (size)
	G1_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 understanding table number and rules are been given below.

Page | 11 **Table 7: Table number entity type**

Table Number	Entity Type	Business Rules
1	G1_CUSTOMER	Records all the personal details about the customer
2	G1_APPLICATION	Records details of the insurance cover requested by customer
3	G1_QUOTE	Records details of customer potential cost of the insurance product
4	G1_INSURANCE_POLICY	Records details of Insurance agreement
5	G1_PREMIUM	Records details of customer cost of payments
6	G1_VEHICLE	Records details of Vehicle model, cost and registration
7	G1_CLAIM	Records details of customer claims in case of an incident
8	G1_SETTLEMENT	Records details of settlement made on claims
9	G1_STAFF	Records details of employees
10	G1_DEPARTMENT	Records details of the various departments
11	G1_OFFICE	Records details of different office locations
12	G1_MEMBERSHIP	Records details of customer membership, clubs, societies
13	G1_SERVICE	Records details of different vehicle services offered
14	G1_NOK	Records details of the next of kin
15	G1_COMPANY	Details of the Insurance organization giving the insurance cover
16	G1_TERMS_CONDITIONS	Records all terms and conditions in regard to the policy
17	G1_RECEIPT	Details of premium payments to customer
18	G1_VEHICLE_INSURANCE _DEPARTMENT	Records details of vehicle insurance cover

Table 8: Abbreviation table of attributes names used in LDM

LDM Attributes	Column Names Abbreviations
G1_CUST_ID	CUSTOMER_IDENTIFICATION
G1_CUST_FNAME	CUSTOMER_FIRST_NAME
G1_CUST_LNAME	CUSTOMER_LAST_NAME
G1_CUST_DOB	CUSTOMER_DATEOFBIRTH
G1_CUST_PPS_NUMBER	CUSTOMER_PERSONALPUBLICSERVICE_NUMBER
G1_STAFF_FNAME	STAFF_FIRSTNAME
G1_STAFF_LNAME	STAFF_LASTNAME
G1_STAFF_PPS_NUMBER	STAFF_PERSONALPUBLICSERVICE_NUMBER
G1_ADMIN_COST	ADMINISTRATION_COST
G1_NOK_ID	NEXTOFKIN_IDENTIFICATION

**Table-LDM 1: G1\_CUSTOMER**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_CUST_ID	INT	PK	The CUSTOMER attributes record all the essential personal details of the customer. The CUST_ID is the unique primary key.
G1_CUST_FNAME	VARCHAR (10)		
G1_CUST_LNAME	VARCHAR (10)		
G1_CUST_DOB	DATE		
G1_CUST_GENDER	ENUM('M','F')		
G1_CUST_ADDRESS	VARCHAR2(20)		
G1_CUST_MOB_NUMBER	INT		
G1_CUST_EMAIL	VARCHAR (20)		
G1_CUST_PASSPORT_NUMBER	VARCHAR (20)		
G1_CUST_MARITAL_STATUS	VARCHAR (20)		
G1_CUST_PPS_NUMBER	INT		

**Table-LDM 2: G1\_Application**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_APPLICATION_ID	INT	PK	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.
G1_CUST_ID	INT	FK	
G1_VEHICLE_ID	INT		
G1_COVERAGE	VARCHAR (80)		

Table-LDM 3: G1\_QUOTE

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_QUOTE_ID	INT	PK	The QUOTE attributes record all the essential quotation details of the customer. The QUOTE_ID is the unique primary key and APPLICATION_ID and is a foreign key linking the table back to the respective entities
G1_APPLICATION_ID	INT	FK	
G1_ISSUE_DATE	DATE		
G1_VALID_FROM_DATE	DATE		
G1_VALID_TILL_DATE	DATE		
G1_DESCRIPTION	VARCHAR (200)		
G1_PRODUCT_ID	VARCHAR (20)		
G1_COVERAGE_LEVEL	VARCHAR (20)		

Table-LDM 4: G1\_INSURANCE\_POLICY

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_AGREEMENT_ID	INT	PK	The INSURANCE POLICY attributes record all the essential policy details of the customer. The AGREEMENT_ID is the unique primary key , and APPLICATION_ID are linked to the other corresponding entities through their foreign keys.
G1_APPLICATION_ID	INT	FK	
G1_DEPARTMENT_NAME	VARCHAR (20)		
G1_POLICY_NUMBER	INT		
G1_START_DATE	DATE		
G1_EXPIRY_DATE	DATE		
G1_TERM_CONDITION_DESCRIPTION	VARCHAR(500)		

Table-LDM 5: G1\_PREMIUM

Attributes		Data Type	Primary and Foreign Keys	Explanation
G1_PREMIUM_PAYMENT_ID	INT		PK	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 table to CUSTOMER entity.
G1_CUST_ID	INT		FK	
G1_POLICY_NUMBER	INT			
G1_PREMIUM_PAYMENT_SCHEDULE	DATE			
G1_PREMIUM_PAYMENT_AMOUNT	INT			

**Table-LDM 6: G1\_ VEHICLE**

Attributes	Data Type	Primary and ForeignKeys	Explanation
G1_VEHICLE_ID	INT	PK	The VEHICLE attributes record all the essential VEHICLE details belonging to the customer. The VEHICLE_ID is the unique primary key is the foreign key linking table to CUSTOMER entity.
G1_CUST_ID	INT	FK	
G1_POLICY_ID	INT		
G1_DEPENDENT_NOK_ID	INT		
G1_VEHICLE_REGISTRATION_NUMBER	INT		
G1_VEHICLE_VALUE	INT		
G1_VEHICLE_TYPE	VARCHAR (20)		
G1_VEHICLE_SIZE	INT		
G1_VEHICLE_NUMBER_OF_SEAT	INT		
G1_VEHICLE_MANUFACTURER	VARCHAR (20)		
G1_VEHICLE_ENGINE_NUMBER	VARCHAR (20)		
G1_VEHICLE_CHASIS_NUMBER	VARCHAR (20)		
G1_VEHICLE_NUMBER	VARCHAR (20)		
G1_VEHICLE_MODEL_NUMBER	VARCHAR (20)		



Table-LDM 7: G1\_CLAIM

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_CLAIM_ID	INT	P K	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 table to CUSTOMER entity.
G1_CUST_ID	INT	FK	
G1_CLAIM_AMOUNT	INT		
G1_INCIDENT_ID	INT		
G1_DAMAGE_TYPE	VARCHAR (20)		
G1_DATE_OF_CLAIM	DATE		
G1_CLAIM_STATUS	VARCHAR (20)		

Table-LDM 8: G1\_SETTLEMENT

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_CLAIM_SETTLEMENT_ID	INT	P K	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 CLAIM_ID are the foreign keys that link the table to the corresponding entity.
G1_CLAIM_ID	INT	FK	
G1_DATE_SETTLED	DATE		
G1_AMOUNT_PAID	INT		

**Table-LDM 9: G1\_STAFF**

Attributes	Data Type Oracle	Primary and Foreign Keys	Explanation
<b>G1_STAFF_ID</b>	INT	PK	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.
<b>G1_STAFF_FNAME</b>	VARCHAR (20)		
<b>G1_STAFF_LNAME</b>	VARCHAR (20)		
<b>G1_STAFF_ADDRESS</b>	VARCHAR (20)		
<b>G1_STAFF_CONTACT</b>	INT		
<b>G1_STAFF_GENDER</b>	ENUM('M','F')		
<b>G1_STAFF_MARITAL_STAT US</b>	ENUM('Single','Ma rried','Divorced','Wi dow')		
<b>G1_STAFF_NATIONALITY</b>	VARCHAR (20)		
<b>G1_STAFF_QUALIFICATION</b>	VARCHAR (20)		
<b>G1_STAFF_ALLOWANCE</b>	INT		
<b>G1_STAFF_PPS_NUMBER</b>	INT		

Table-LDM 10: G1\_ DEPARTMENT

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_DEPARTMENT_NAME	VARCHAR (20)	P K	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.
G1_IPC_ID	INT	FK	
G1_OFFICE	VARCHAR (200)		
G1_CONTACT_INFORMATION	VARCHAR (200)		
G1_DEPARTMENT_STAFF	VARCHAR (200)		
G1_DEPARTMENT_LEADER	VARCHAR (200)		

Table-LDM 11: G1\_OFFICE

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_OFFICE_NAME	VARCHAR (20)	PK	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.
G1_OFFICE_LEADER	VARCHAR (20)		
G1_CONTACT_INFORMATION	VARCHAR (20)		
G1_ADDRESS	VARCHAR (200)		
G1_ADMIN_COST	INT		
G1_STAFF	VARCHAR (200)		

**Table-LDM 12: G1\_ MEMBERSHIP**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_MEMBERSHIP_ID	INT	PK	The MEMBERSHIP attributes record all the essential membership details available for insured customer. The MEMBERSHIP_ID is the unique primary key and the CUST_ID is a foreign key linking the table back to the entity type CUSTOMER.
G1_CUST_ID	INT	FK	
G1_MEMBERSHIP_TYPE	VARCHAR (20)		
G1_ORGANISATION_CONTACT	INT		

**Table-LDM 13: G1\_ SERVICE**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_VEHICLE_SERVICE	VARCHAR (20)	PK	The VEHICLE_SERVICE attributes record all the essential vehicle services offered to insured customer details. The VEHICLE_SERVICE is the unique primary key and ipc_ID are linked to the other corresponding entities.
G1_IPC_ID	INT	FK	
G1_VEHICLE_SERVICE_ADDRESSES	VARCHAR (200)		
G1_VEHICLE_SERVICE_CONTACT	INT		
G1_VEHICLE_SERVICE_INCHARGE	VARCHAR (20)		
G1_VEHICLE_SERVICE_TYPE	VARCHAR (20)		

Table-LDM 14: G1\_NOK

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_NOK_ID	INT	PK	The NOK attributes record information on the next of kin details. NOK_ID is the unique primary key here. AGREEMENT_ID are foreign keys linking back information to their respective entities.
G1_AGREEMENT_ID	INT	FK	
G1_NOK_NAME	VARCHAR (20)		
G1_NOK_ADDRESS	VARCHAR (20)		
G1_NOK_PHONE_NUMBER	INT		
G1_NOK_MARITAL_STATUS	ENUM('Single', 'Married', 'Divorced', 'Widow')		
G1_NOK_GENDER	ENUM('M', 'F')		

**Table-LDM 15: G1\_COMPANY**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_COMPANY_NAME	VARCHAR (20)	PK	The INSURANCE COMPANY attributes record all the essential company details of the customer. The COMPANY_ID is the unique primary key
G1_DEPARTMENT_NAME	VARCHAR (20)	P K	
G1_COMPANY_ADDRESS	INT		
C G1_COMPANY_CONTACT_NUMBER	INT		
G1_COMPANY_FAX	VARCHAR (20)		
G1_COMPANY_EMAIL	VARCHAR (20)		
G1_COMPANY_WEBSITE	VARCHAR (20)		
G1_COMPANY_LOCATION	VARCHAR (20)		
G1_COMPANY DEPARTMENT_NAME	VARCHAR (20)		
G1_COMPANY_OFFICE_NAME			

Table-LDM 16: G1\_TERMS\_CONDITIONS

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_COVERAGE_ID	INT	PK	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.
G1_COMPANY_NAME	VARCHAR (20)	FK	
G1_COVERAGE_AMOUNT	INT		
G1_COVERAGE_TYPE	VARCHAR (20)		
G1_COVERAGE_LEVEL	VARCHAR (20)		
G1_PRODUCT_ID	INT		
G1_COVERAGE_DESCRIPTION	VARCHAR (20)		
G1_COVERAGE_TERMS	VARCHAR (200)		



**Table-LDM 17: G1\_RECEIPT**

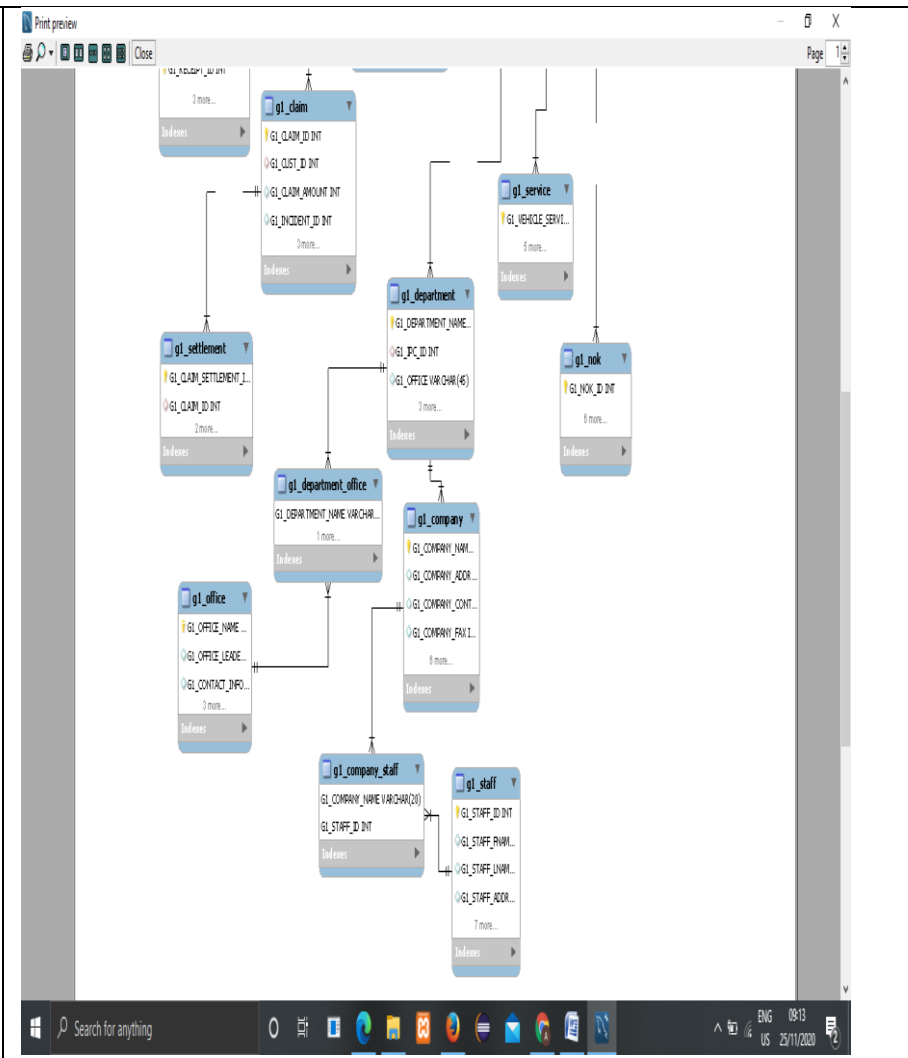
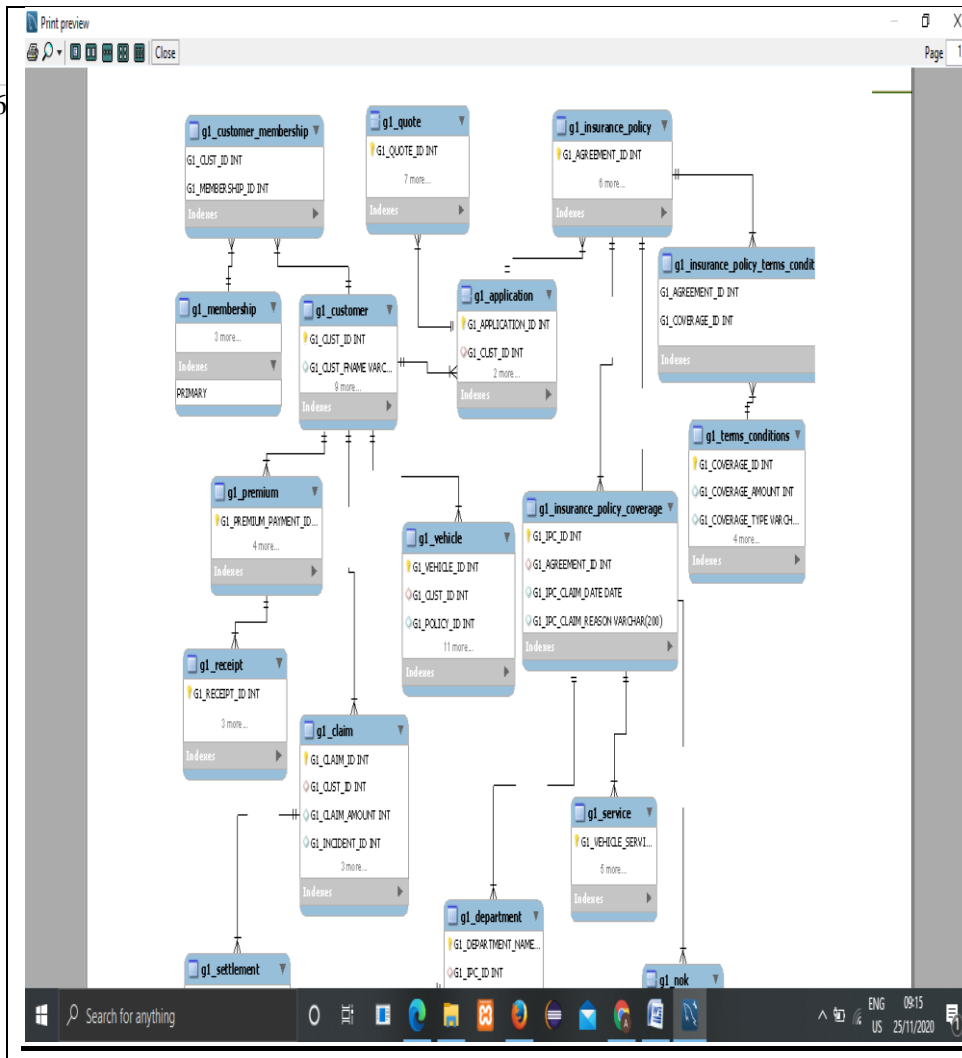
Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_RECEIPT_ID	INT	PK	The RECEIPT attributes record all the essential payments done by CUSTOMERS to Insurance company. The RECEIPT_ID is the unique primary key and PREMIUM_PAYMENT_ID and are Foreign keys linking table to their respective entities.
G1_PREMIUM_PAYMENT_ID	INT	FK	
G1_COST	INT		
G1_TIME	DATE		

**Table-LDM 18: G1\_INSURANCE\_POLICY\_COVERAGE**

Attributes	Data Type	Primary and Foreign Keys	Explanation
G1_IPC_ID	INT	PK	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
G1_AGREEMENT_ID	VARCHAR (20)	FK	
G1_IPC_CLAIM_DATE	DATE		
G1_IPC_CLAIM_REASON	VARCHAR(200)		

# Graphical representation of LDM

Page | 26



## **PHYSICAL DATA MODEL (PART C)**

### 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.

### 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

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.

### 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.

# NORMALIZATION

It is a relationship database concept and is done in process of building ER. If the correct entity model is being build will confirm to the rules of normalization. Each rule has corresponding data model interpretation, which can be used to validate placement of attributes in ER model.

- It is a process which allows user to remove redundant data within the database.
- **Normalization** is the process of splitting relations into well structured relations that allow users to insert, delete, and update tuples without introducing database.
- Normalization is a multi-step process beginning with an “unnormalized” relation
- Process involves restructuring the tables to successively meeting higher forms of Normalization.
- A properly normalized database should have the following characteristics
- Scalar/Atomic values in each fields
- Absence of redundancy.
- Minimal use of null values.
- Minimal loss of information.

## Testing Queries

### First query

Query 1

Limit to 1000 rows

```
1 •
2 • ner join G1_Vehicle as V on C.G1_Cust_Id=V.G1_Cust_Id where C.G1_Cust_Id in (select G1_Cust_Id from G1_Claim where G1_Claim_Status = 'pending');
```

Result Grid

Filter Rows:  Export: Wrap Cell Content:

	G1_Cust_FName	G1_Cust_DOB	G1_Cust_Address	G1_Cust_Mob_Number	G1_Vehide_Registration_Number	G1_Vehide_Type
▶	neha	2001-03-15	chitradurga	639596	3	sedan
	bharath	2000-06-03	shimogga	855180	8	sedan
	meghana	2000-08-29	manglore	328798	11	sedan

Result 1

Read Only

Result Grid

Form Editor

Field Types

## 2<sup>nd</sup> Query

Query 1

Limit to 1000 rows

```
1
2
3
4 •  ess,C.G1_Cust_Mob_Number,V.G1_Vehicle_Registration_Number,V.G1_Vehicle_Type from G1_Customer as C inner join G1_Vehicle as V on C.G1_Cust_Id=V.G1_Cust_Id;
5
6
7 •  Premium on G1_Customer.G1_Cust_Id=G1_Premium.G1_Cust_Id where G1_Premium.G1_Premium_Payment_Amount > (select Sum(G1_Cust_Id) from G1_Customer);
```

Result Grid

Filter Rows: | Export: | Wrap Cell Content: |

	G1_CUST_ID	G1_CUST_FNAME	G1_CUST_LNAME	G1_CUST_DOB	G1_CUST_GENDER	G1_CUST_ADDRESS	G1_CUST_EMAIL	G1_CUST_MOB_NUMBER	G1_CUST_PASSPORT
▶	1	rahul	s	2000-05-08	M	banglore	rahulshanthkumar@gmail.com	779591	DA26010
	2	anuj	c.s	2000-06-26	M	ranibennur	anujcs@gmail.com	974561	CA06260
	3	neha	t	2001-03-15	F	chitradurga	nehat@gmail.com	639596	TN26150
	4	varun	a	2000-10-14	M	kazipet	varuna@gmail.com	145591	AV26111
	5	sushanth	p	2000-10-29	M	mysore	sushanthp@gmail.com	739521	PS26220
	6	trishul	k.s	2000-01-05	M	dharwad	trishulks@gmail.com	742501	ST41010
	7	chaitanya	k.v.s	2001-05-12	M	ungole	chaitanyakvs@gmail.com	989256	KC26028
	8	bharath	g.k	2000-06-03	M	shimogga	bharathgk@gmail.com	855180	GK26069
	9	daneshwari	m.y	2000-02-05	F	davengere	daneshwarimy@gmail.com	983145	MD11884
	10	meghana	n	2000-08-29	F	manglore	meghanan@gmail.com	328798	NM29081

Result 2 ×

Read Only

### Third Query

Table\_Creation Value\_Insertion Project\_Query\* x

Limit to 1000 rows

```
3  /* First Query */
4  • select C.G1_Cust_FName,C.G1_Cust_DOB,C.G1_Cust_Address,C.G1_Cust_Mob_Number,V.G1_Vehicle_Registration_Number,V.G1_Vehicle_Type from G1_Customers
5
6  /* Second Query */
7  • select G1_Customer.* from G1_Customer inner join G1_Premium on G1_Customer.G1_Cust_Id=G1_Premium.G1_Cust_Id where G1_Premium.G1_Premium_Payer
8
9  /* Fourth Query */
10 • SELECT * FROM G1_CUSTOMER WHERE G1_CUST_ID IN (SELECT G1_CUST_ID FROM G1_VEHICLE GROUP BY(G1_CUST_ID) HAVING count(G1_VEHICLE_ID) >=2 AND G1_C
11 /* Fifth Query */
12 • SELECT V1 * FROM G1_VEHICLE V1 INNER JOIN G1_Premium P1 ON V1.G1_CUST_ID= P1.G1_CUST_ID AND V1.G1_VEHICLE_NUMBER
```

Result Grid

Filter Rows: Edit: Export/Import: Wrap Cell Content:

	G1_CUST_ID	G1_CUST_FNAME	G1_CUST_LNAME	G1_CUST_DOB	G1_CUST_GENDER	G1_CUST_ADDRESS	G1_CUST_EMAIL	G1_CUST_MOB_NUMBER	G1_CUST_PASSPORT_NUMBER
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

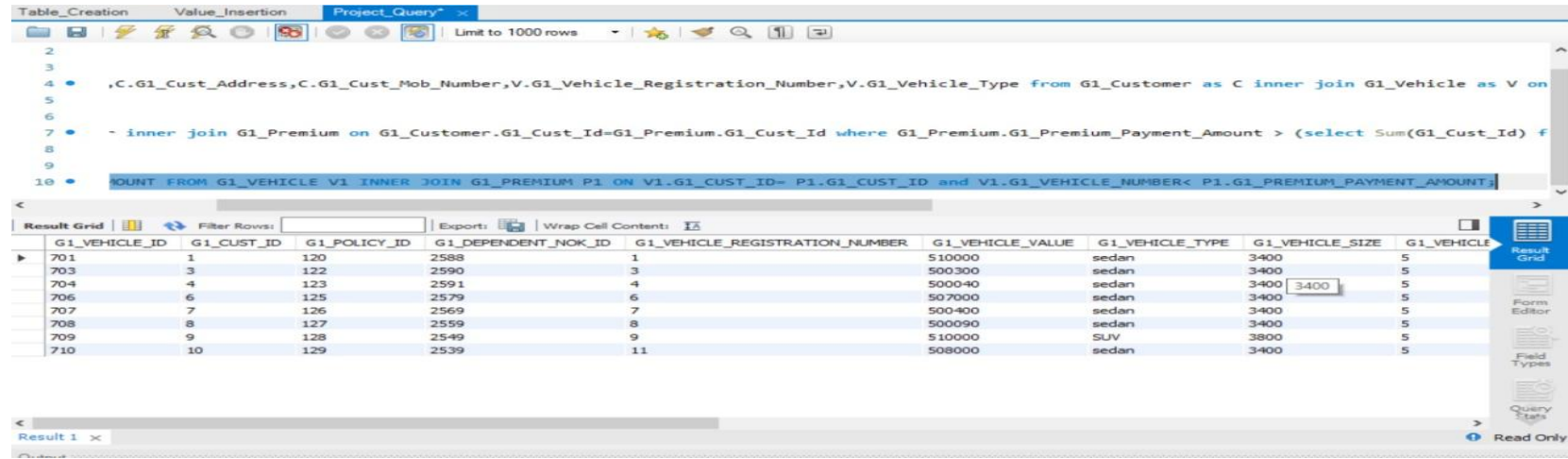
Field Types

Query Stats

\_CUSTOMER 1 x

Apply Revert

## Fourth Query



The screenshot shows a database query editor with a tab labeled "Project\_Query". The SQL query is as follows:

```
SELECT C.G1_Cust_Address, C.G1_Cust_Mob_Number, V.G1_Vehicle_Registration_Number, V.G1_Vehicle_Type FROM G1_Customer AS C INNER JOIN G1_Vehicle AS V ON V.G1_Vehicle_ID = C.G1_Vehicle_ID  
- inner join G1_Premium on G1_Customer.G1_Cust_Id=G1_Premium.G1_Cust_Id where G1_Premium.G1_Premium_Payment_Amount > (select Sum(G1_Cust_Id) f  
OUNT FROM G1_VEHICLE V1 INNER JOIN G1_PREMIUM P1 ON V1.G1_CUST_ID= P1.G1_CUST_ID and V1.G1_VEHICLE_NUMBER< P1.G1_PREMIUM_PAYMENT_AMOUNT;
```

Below the query, the "Result Grid" displays the following data:

G1_VEHICLE_ID	G1_CUST_ID	G1_POLICY_ID	G1_DEPENDENT_NOK_ID	G1_VEHICLE_REGISTRATION_NUMBER	G1_VEHICLE_VALUE	G1_VEHICLE_TYPE	G1_VEHICLE_SIZE	G1_VEHICLE
701	1	120	2588	1	510000	sedan	3400	5
703	3	122	2590	3	500300	sedan	3400	5
704	4	123	2591	4	500040	sedan	3400	5
706	6	125	2579	6	507000	sedan	3400	5
707	7	126	2569	7	500400	sedan	3400	5
708	8	127	2559	8	500090	sedan	3400	5
709	9	128	2549	9	510000	SUV	3800	5
710	10	129	2539	11	508000	sedan	3400	5

## Grant And Privilege

A **privilege** is a right to execute a particular type of SQL statement or to access another user's object. Some examples of privileges include the right to:

- Connect to the database (create a session)
- Create a table
- Select rows from another user's table
- Execute another user's stored procedure



You grant privileges to users so these users can accomplish tasks required for their job. You should grant a privilege only to a user who absolutely requires the privilege to accomplish necessary work. Excessive granting of unnecessary privileges can compromise security. A user can receive a privilege in two different ways:

Page | 33

- You can grant privileges to users explicitly. For example, you can explicitly grant the privilege to insert records into the `employees` table to the user `SCOTT`.
- You can also grant privileges to a role (a named group of privileges), and then grant the role to one or more users. For example, you can grant the privileges to select, insert, update, and delete records from the `employees` table to the role named `clerk`, which in turn you can grant to the users `scott` and `brian`.

Because roles allow for easier and better management of privileges, you should normally grant privileges to roles and not to specific users.

There are two distinct categories of privileges:

- System privileges
- Schema object privileges