VEHICLE INSURANCE DATABASE

DBMS (CS310)

PREPARED BY:

- B. Deepak (20BCS027)
- Chippagiri Karthik (20BCS036)
- Bhukya Jagadeesh (20BCS032)
- Ankati Vivek (20BCS016)
- Kasse Lokesh Babu (20BCS067)
- K. Nirmal Swaroop(20BCS068)
- K. Balaji (20BCS073)
- P. Vivek Yadav (20BCS105)
- Ravula Veekshith Reddy(20BCS111)
- Rangam Sai Charan(20bcs108)

Under the Guidance of-

Dr. Uma Seshadri,

Professor, HOD, Department of Computer Science

Indian Institute of Information Technology Dharwad.

Dr. Pramod Yelmewad

Assistant Professor, Department of Computer Science

Indian Institute of Information Technology Dharwad.

Dr. Supriya Nadiger

Assistant Professor, Department of Computer Science

Indian Institute of Information Technology Dharwad.

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

TABLE OF CONTENTS

ACKNOWLEDGEMENT

We would like to sincerely and profusely thank **Dr Uma Seshadri**, for her able guidance and for giving us the opportunity to take up this project. We would also like to thank **Dr. Pramod Yelmewad** and **Dr. Supriya Nadiger** for their guidance and also for giving us the opportunity to take up this project.

AIM:

Project is meant for helping our learning skills improve in understanding relational database management system (DBMS). We will learn some theoretical and practical concepts of DBMS and how they are implemented in real life.

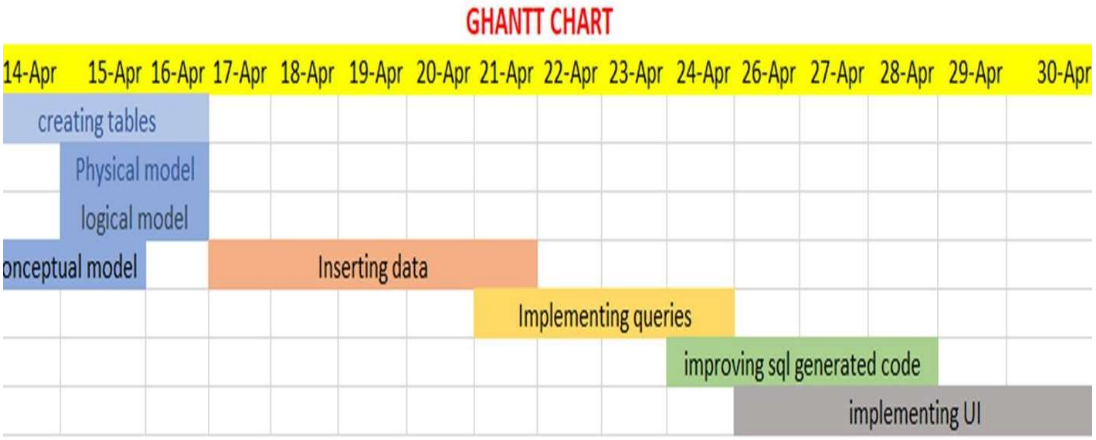
OBJECTIVE:

The main objective is to gain practical knowledge on working with a database, encountering and overcoming the problems and challenges faced during its implementation. We need to create a database for a vehicle insurance company. To understand all the relationships, functions, constraints, operators etc. which are used while using MySQL. Lastly, an important objective of this project is to be able to collaborate and work as team and bring out value for everybody.

ROLES AND RESPONSIBILITES:

STUDENT NAME	ROLE	ROLLNO	Email
B. Deepak	Data insertion, Conceptual DM,	20BCS027	20bcs027@iiitdwd.ac.in
C. Karthik	Data Insertion, Conceptual DM	20BCS036	20bcs036@iiitdwd.ac.in
Jagadeesh	Data Insertion, Physical Model	20BCS032	20bcs032@iiitdwd.ac.in
A. Vivek	Data Insertion, Queries	20BCS016	20bcs016@iiitdwd.ac.in
K. Lokesh	Data Insertion, Physical Model	20BCS067	20bcs067@iiitdwd.ac.in
K. Nirmal Swaroop	Data Insertion, Queries	20BCS068	20bcs068@iiitdwd.ac.in
K. Balaji	Data Insertion, Physical model	20BCS073	20bcs073@iiitdwd.ac.in
P. Vivek	Data Insertion, Queries	20BCS105	20bcs105@iiitdwd.ac.in
Sai Charan	Data Insertion, Queries	20BCS108	20bcs027@iiitdwd.ac.in
Veekshith	Data Insertion, Queries	20BCS111	20bcs027@iiitdwd.ac.in

TIMELINE:


GHANTT CHART

PROJECT IMPLEMENTATION:

→ Tables Creation

→ Conceptual data model (CDM)

→ Physical data model (PDM)

→ Understanding Database

→ Insertion of valid data

→ Creating functions and stored procedures

→ Writing the queries

→ Execution of queries

→ Working on the errors, if they occur while improving the data base.

TABLES CREATION:

According to database the tables which are required are created without inserting any data initially. Table with the column names and rows are created.
Here are the following tables and the names of the tables,

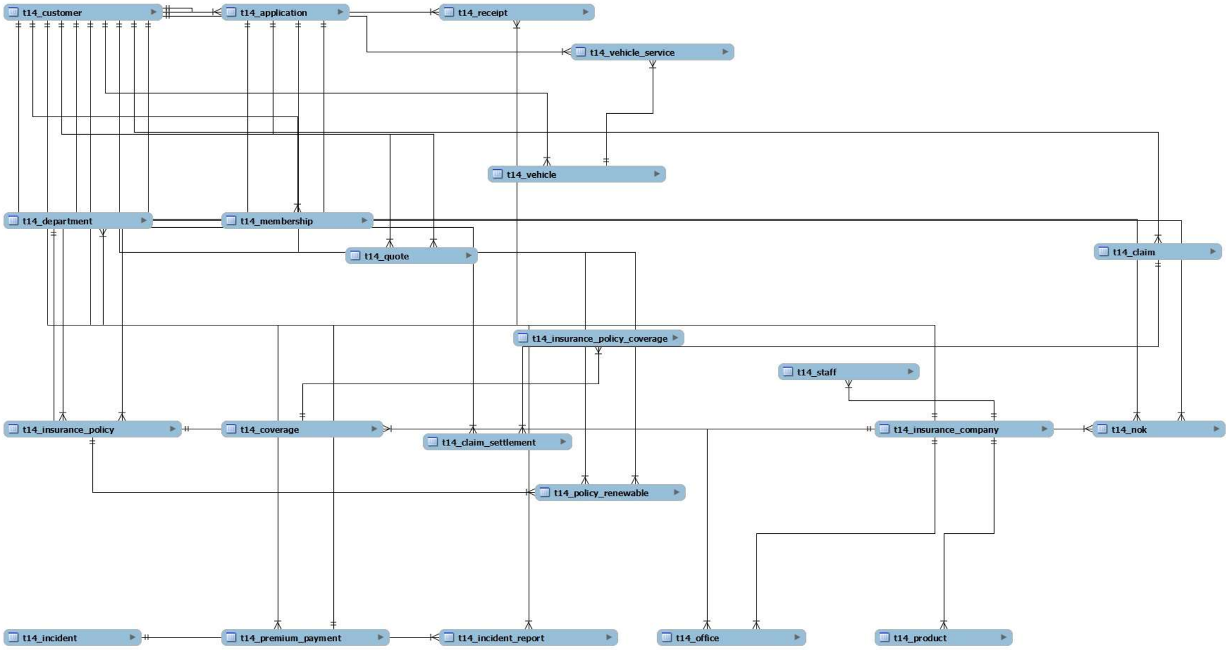
- t14_customer;
- t14_application;
- t14_quote;
- t14_insurance_policy;
- t14_premium_payment;
- t14_vehicle;
- t14_claim;
- t14_claim_settlement;
- t14_staff;
- t14_department;
- t14_office;
- t14_membership;
- t14_vehicle_service;
- t14_nok;
- t14_insurance_companies;
- t14_policy_renewable;
- t14_incident;
- t14_incident_report;
- t14_coverage;
- t14_product;
- t14_receipt;

- t14_insurance_Policy_coverage;

CONCEPTUAL DATA MODEL(CDM):

The other name for the conceptual data model is a business model. This model focuses on identifying the data used in the business but not its processing flow or physical characteristics. As the conceptual data model is of high level it usually not contains attributes in its structure. This model is used to define the relationship among the data entities but not provide information about cardinality properties.

Below is the **CDM** of vehicle insurance database.



PHYSICAL DATA MODEL (PDM):

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables. Features of a physical data model include:

→ Specification all tables and columns.

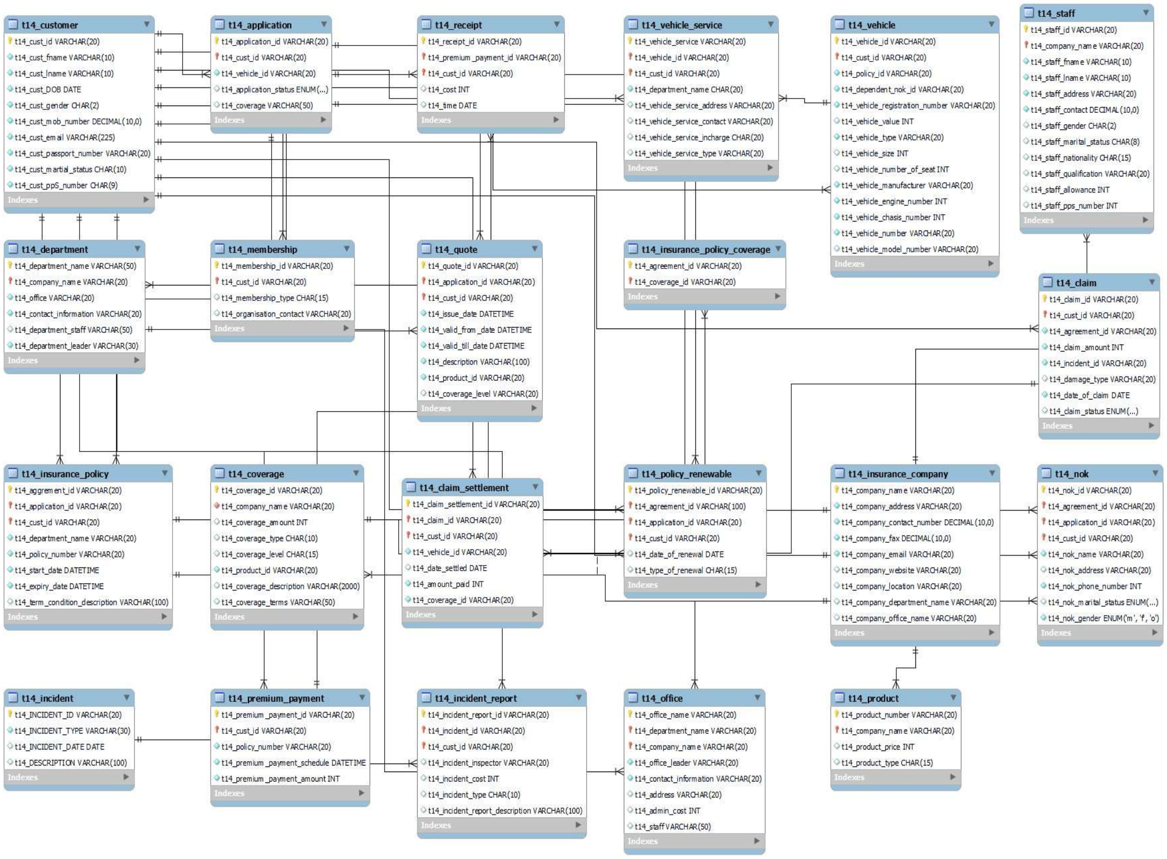
→ Foreign keys are used to identify relationships between tables.

→ Denormalization may occur based on user requirements.

→ Physical considerations may cause the physical data model to be quite different from the logical data model.

→ Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server.

Below is the PDM of Vehicle database management system.


t14_customer
t14_cust_id VARCHAR(20)
t14_cust_fname VARCHAR(10)
t14_cust_lname VARCHAR(10)
t14_cust_DOB DATE
t14_cust_gender CHAR(2)
t14_cust_mob_number DECIMAL(10,0)
t14_cust_email VARCHAR(225)
t14_cust_passport_number VARCHAR(20)
t14_cust_martial_status CHAR(10)
t14_cust_ppS_number CHAR(9)
Indexes
t14_application
t14_application_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_vehicle_id VARCHAR(20)
t14_application_status ENUM(...)
t14_coverage VARCHAR(50)
Indexes
t14_receipt
t14_receipt_id VARCHAR(20)
t14_premium_payment_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_cost INT
t14_time DATE
Indexes
t14_vehicle_service
t14_vehicle_service VARCHAR(20)
t14_vehicle_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_department_name CHAR(20)
t14_vehicle_service_address VARCHAR(20)
t14_vehicle_service_contact VARCHAR(20)
t14_vehicle_service_incharge CHAR(20)
t14_vehicle_service_type VARCHAR(20)
Indexes
t14_vehicle
t14_vehicle_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_policy_id VARCHAR(20)
t14_dependent_nok_id VARCHAR(20)
t14_vehicle_registration_number VARCHAR(20)
t14_vehicle_value INT
t14_vehicle_type VARCHAR(20)
t14_vehicle_size INT
t14_vehicle_number_of_seat INT
t14_vehicle_manufacturer VARCHAR(20)
t14_vehicle_engine_number INT
t14_vehicle_chasis_number INT
t14_vehicle_number VARCHAR(20)
t14_vehicle_model_number VARCHAR(20)
Indexes
t14_staff
t14_staff_id VARCHAR(20)
t14_company_name VARCHAR(20)
t14_staff_fname VARCHAR(10)
t14_staff_lname VARCHAR(10)
t14_staff_address VARCHAR(20)
t14_staff_contact DECIMAL(10,0)
t14_staff_gender CHAR(2)
t14_staff_marital_status CHAR(8)
t14_staff_nationality CHAR(15)
t14_staff_qualification VARCHAR(20)
t14_staff_allowance INT
t14_staff_pps_number INT
Indexes
t14_department
t14_department_name VARCHAR(50)
t14_company_name VARCHAR(20)
t14_office VARCHAR(20)
t14_contact_information VARCHAR(20)
t14_department_staff VARCHAR(50)
t14_department_leader VARCHAR(30)
Indexes
t14_membership
t14_membership_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_membership_type CHAR(15)
t14_organisation_contact VARCHAR(20)
Indexes
t14_quote
t14_quote_id VARCHAR(20)
t14_application_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_issue_date DATETIME
t14_valid_from_date DATETIME
t14_valid_till_date DATETIME
t14_description VARCHAR(100)
t14_product_id VARCHAR(20)
t14_coverage_level VARCHAR(20)
Indexes
t14_insurance_policy_coverage
t14_agreement_id VARCHAR(20)
t14_coverage_id VARCHAR(20)
Indexes
t14_claim
t14_claim_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_agreement_id VARCHAR(20)
t14_claim_amount INT
t14_incident_id VARCHAR(20)
t14_damage_type VARCHAR(20)
t14_date_of_claim DATE
t14_claim_status ENUM(...)
Indexes
t14_insurance_policy
t14_aggrement_id VARCHAR(20)
t14_application_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_department_name VARCHAR(20)
t14_policy_number VARCHAR(20)
t14_start_date DATETIME
t14_expiry_date DATETIME
t14_term_condition_description VARCHAR(100)
Indexes
t14_coverage
t14_coverage_id VARCHAR(20)
t14_company_name VARCHAR(20)
t14_coverage_amount INT
t14_coverage_type CHAR(10)
t14_coverage_level CHAR(15)
t14_product_id VARCHAR(20)
t14_coverage_description VARCHAR(2000)
t14_coverage_terms VARCHAR(50)
Indexes
t14_claim_settlement
t14_claim_settlement_id VARCHAR(20)
t14_claim_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_vehicle_id VARCHAR(20)
t14_date_settled DATE
t14_amount_paid INT
t14_coverage_id VARCHAR(20)
Indexes
t14_policy_renewable
t14_policy_renewable_id VARCHAR(20)
t14_agreement_id VARCHAR(100)
t14_application_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_date_of_renewal DATE
t14_type_of_renewal CHAR(15)
Indexes
t14_insurance_company
t14_company_name VARCHAR(20)
t14_company_address VARCHAR(20)
t14_company_contact_number DECIMAL(10,0)
t14_company_fax DECIMAL(10,0)
t14_company_email VARCHAR(20)
t14_company_website VARCHAR(20)
t14_company_location VARCHAR(20)
t14_company_department_name VARCHAR(20)
t14_company_office_name VARCHAR(20)
Indexes
t14_nok
t14_nok_id VARCHAR(20)
t14_agreement_id VARCHAR(20)
t14_application_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_nok_name VARCHAR(20)
t14_nok_address VARCHAR(20)
t14_nok_phone_number INT
t14_nok_marital_status ENUM(...)
t14_nok_gender ENUM('m', 'f', 'o')
Indexes
t14_incident
t14_INCIDENT_ID VARCHAR(20)
t14_INCIDENT_TYPE VARCHAR(30)
t14_INCIDENT_DATE DATE
t14_DESCRIPTION VARCHAR(100)
Indexes
t14_premium_payment
t14_premium_payment_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_policy_number VARCHAR(20)
t14_premium_payment_schedule DATETIME
t14_premium_payment_amount INT
Indexes
t14_incident_report
t14_incident_report_id VARCHAR(20)
t14_incident_id VARCHAR(20)
t14_cust_id VARCHAR(20)
t14_incident_inspector VARCHAR(20)
t14_incident_cost INT
t14_incident_type CHAR(10)
t14_incident_report_description VARCHAR(100)
Indexes
t14_office
t14_office_name VARCHAR(20)
t14_department_name VARCHAR(20)
t14_company_name VARCHAR(20)
t14_office_leader VARCHAR(20)
t14_contact_information VARCHAR(20)
t14_address VARCHAR(20)
t14_admin_cost INT
t14_staff VARCHAR(50)
Indexes
t14_product
t14_product_number VARCHAR(20)
t14_company_name VARCHAR(20)
t14_product_price INT
t14_product_type CHAR(15)
Indexes

Understanding Relations between Entities:

The main purpose of the Physical data model and Logical Data Model is to understand the cardinality, type of relation and relational constraints on relationship and many more. From the physical and logical data models following are the inference made

Relationship types:

One to one relation: only one mapping can be done from parent entity to child entity i.e. the foreign key in the chid table must be unique. To implement one to one relation in sql we use unique constraint on foreign key in child table.
Following are the one-to-one relationships in the database

→ Application to Quote
→ Customer to membership
→ Premium payment to Receipt
→ Claim to Claim settlement

Many to Many relation: many mappings can be done from parent entity to child entity and vice versa. To implement many to many relation in sql we use a bridged table with foreign keys from both the table and created a composite primary key to prevent the repetition of same combined attribute. Following are the Many to Many relationships in the database

→Insurance policy to coverage,
Bridged_table:t14_insurance_Policy_coverage;

One to Many relation: many mapping can be done from parent entity to child entity. To implement one to many relation in sql we just reference the foreign key in child table. Following are the Many to Many relationships in the database

→ all Remaining relations except the above relations

Creating Functions and Stored Procedures

A function is compiled and executed every time whenever it is called. A function must return a value and cannot modify the data received as parameters. In MySQL the creation of function is as followed

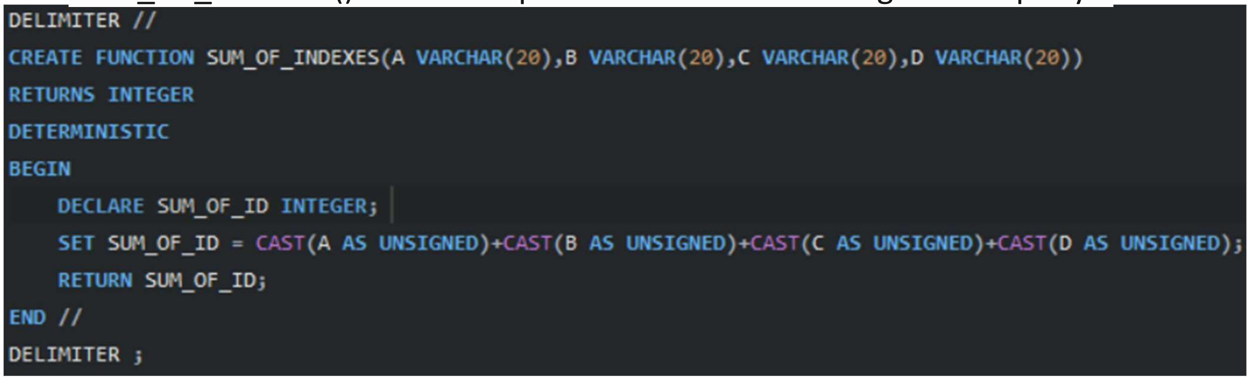
```
CREATE FUNCTION function_name [ (parameter datatype [, parameter datatype])]
RETURNS return_datatype
BEGIN
  declaration_section
  executable_section
END;
```

Stored Procedures are pre-compiled objects which are compiled for the first time and its compiled format is saved, which executes (compiled code) whenever it is called.

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype [,
parameter datatype])]
BEGIN
   Declaration_section
   Executable_section
END &&
DELIMITER ;
```

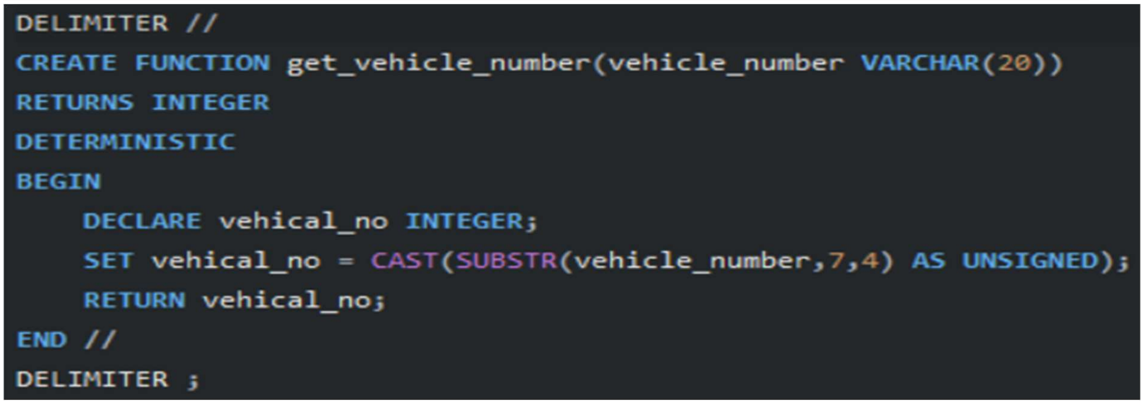
The following are the functions used in the vehicle insurance database:

SUM_OF_INDEXES() – To sum up all the indexes and designed for query 6



```
DELIMITER //
CREATE FUNCTION SUM_OF_INDEXES(A VARCHAR(20),B VARCHAR(20),C VARCHAR(20),D VARCHAR(20))
RETURNS INTEGER
DETERMINISTIC
BEGIN
    DECLARE SUM_OF_ID INTEGER;
    SET SUM_OF_ID = CAST(A AS UNSIGNED)+CAST(B AS UNSIGNED)+CAST(C AS UNSIGNED)+CAST(D AS UNSIGNED);
    RETURN SUM_OF_ID;
END //
DELIMITER ;
```

GET_VEHICLE_NUMBER () – To get vehicle number and designed for query 4



```
DELIMITER //
CREATE FUNCTION get_vehicle_number(vehicle_number VARCHAR(20))
RETURNS INTEGER
DETERMINISTIC
BEGIN
    DECLARE vehical_no INTEGER;
    SET vehical_no = CAST(SUBSTR(vehicle_number,7,4) AS UNSIGNED);
    RETURN vehical_no;
END //
DELIMITER ;
```

To get data stored procedure Get data is used and to delete data stored procedure Delete data is used.

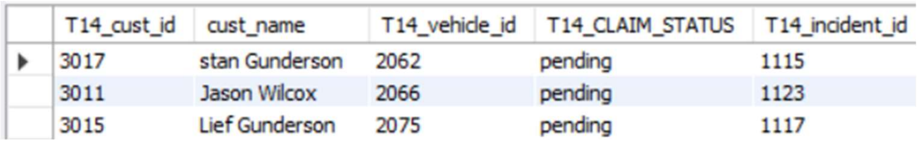
QUERIES:

Query 1:

Retrieve Customer and Vehicle details who has been involved in an incident and claim status is pending – Customer, vehicle, claim status, incident

```
SELECT c.T14_cust_id, CONCAT (T14_cust_fname, ' ', T14_cust_lname) AS
cust_name, v.T14_vehicle_id, T14_CLAIM_STATUS, T14_incident_id
FROM T14_claim cl JOIN T14_vehicle v
ON v.T14_policy_id = cl. T14_agreement_id JOIN T14_customer c ON
c.T14_cust_id = v.T14_cust_id WHERE T14_incident_id IS NOT NULL AND
T14_claim_status LIKE 'pending';
```

RESULT OF THE QUERY:



	T14_cust_id	cust_name	T14_vehicle_id	T14_CLAIM_STATUS	T14_incident_id
▶	3017	stan Gunderson	2062	pending	1115
	3011	Jason Wilcox	2066	pending	1123
	3015	Lief Gunderson	2075	pending	1117

Query 2:

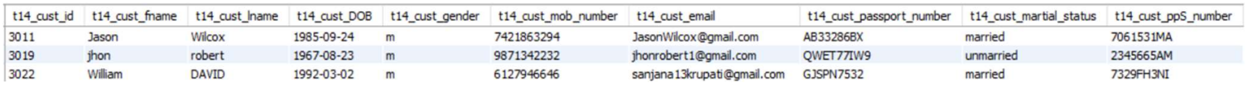
Retrieve customer details who has premium payment amount greater than the sum of all the customer Ids in the database – premium payment, customer

```
CREATE VIEW cust_sum AS SELECT SUM(T14_cust_id) FROM T14_customer;
SELECT * FROM T14_CUSTOMER WHERE T14_CUST_ID IN (SELECT T14_CUST_ID
FROM T14_PREMIUM_PAYMENT WHERE T14_premium_payment_amount >
(SELECT * FROM cust_sum));
```

(OR)

```
SELECT * FROM T14_CUSTOMER WHERE T14_CUST_ID IN (SELECT T14_CUST_ID
FROM T14_PREMIUM_PAYMENT WHERE T14_premium_payment_amount >
(SELECT SUM(T14_cust_id) FROM T14_customer));
```

RESULT OF THE QUERY:



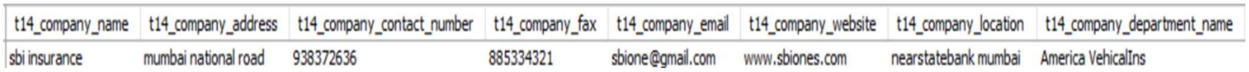
t14_cust_id	t14_cust_fname	t14_cust_lname	t14_cust_DOB	t14_cust_gender	t14_cust_mob_number	t14_cust_email	t14_cust_passport_number	t14_cust_martial_status	t14_cust_ppS_number
3011	Jason	Wilcox	1985-09-24	m	7421863294	JasonWilcox@gmail.com	AB33286BX	married	7061531MA
3019	jhon	robert	1967-08-23	m	9871342232	jhonrobert1@gmail.com	QWET77IW9	unmarried	2345665AM
3022	William	DAVID	1992-03-02	m	6127946646	sanjana13krupati@gmail.com	GJSPN7532	married	7329FH3NI

Query 3:

Retrieve Company details whose number of products is greater than departments, where the departments are located in more than one location— company, product, departments, office Query 3:

```
SELECT * FROM t14_insurance_companies WHERE t14_company_name in(select
o.t14_company_name from t14_PRODUCT p inner join t14_OFFICE o on
o.t14_Company_Name = p.t14_Company_Name  group by o.t14_Company_Name
having  Count(distinct(t14_Product_Number))
<count(distinct(t14_Department_Name)    ) and count(t14_address)>1);
```

RESULT OF THE QUERY:



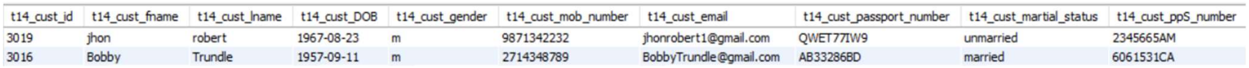
t14_company_name	t14_company_address	t14_company_contact_number	t14_company_fax	t14_company_email	t14_company_website	t14_company_location	t14_company_department_name
sbi insurance	mumbai national road	938372636	885334321	sbione@gmail.com	www.sbiones.com	nearstatebank mumbai	America VehicalIns

Query 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

```
select T14_customer.*  from T14_Customer where T14_customer.T14_cust_id IN(
SELECT c.T14_cust_id      from T14_customer c  join t14_incident_report IR on
c.T14_cust_id = IR.T14_cust_id left join t14_receipt R on c.T14_cust_id =
r.T14_cust_id where c.T14_cust_id in ( select v.T14_cust_id  from T14_vehicle V
group by T14_cust_id having count(V.T14_cust_id)>1) and R.t14_receipt_id is null
and T14_incident_type like "%accident%");
```

RESULT OF THE QUERY:



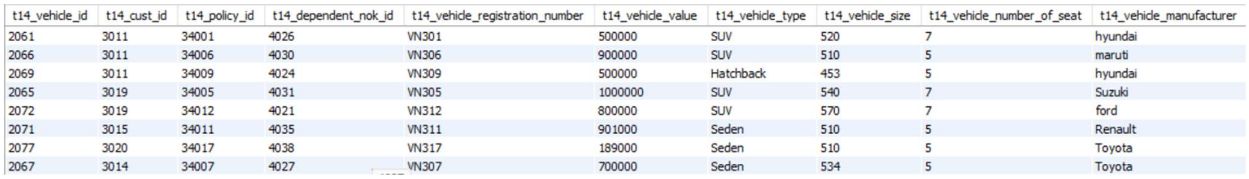
t14_cust_id	t14_cust_fname	t14_cust_lname	t14_cust_DOB	t14_cust_gender	t14_cust_mob_number	t14_cust_email	t14_cust_passport_number	t14_cust_martial_status	t14_cust_ppS_number
3019	jhon	robert	1967-08-23	m	9871342232	jhonrobert1@gmail.com	QWET77IW9	unmarried	2345665AM
3016	Bobby	Trundle	1957-09-11	m	2714348789	BobbyTrundle@gmail.com	AB33286BD	married	6061531CA

Query 5:

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

```
SELECT t14_vehicle. * FROM t14_vehicle JOIN t14_premium_payment ON
t14_vehicle. T14_cust_id = t14_premium_payment. T14_cust_id WHERE
t14_premium_payment_amount > GET_VEHICLE_NUMBER(t14_vehicle_number);
```

RESULT OF THE QUERY:



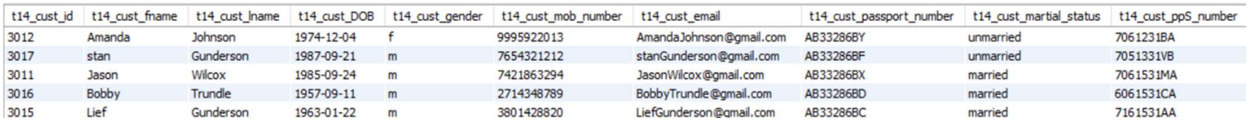
t14_vehicle_id	t14_cust_id	t14_policy_id	t14_dependent_nok_id	t14_vehicle_registration_number	t14_vehicle_value	t14_vehicle_type	t14_vehicle_size	t14_vehicle_number_of_seat	t14_vehicle_manufacturer
2061	3011	34001	4026	VN301	500000	SUV	520	7	hyundai
2066	3011	34006	4030	VN306	900000	SUV	510	5	maruti
2069	3011	34009	4024	VN309	500000	Hatchback	453	5	hyundai
2065	3019	34005	4031	VN305	1000000	SUV	540	7	Suzuki
2072	3019	34012	4021	VN312	800000	SUV	570	7	ford
2071	3015	34011	4035	VN311	901000	Seden	510	5	Renault
2077	3020	34017	4038	VN317	189000	Seden	510	5	Toyota
2067	3014	34007	4027	VN307	700000	Seden	534	5	Toyota

Query 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 T14_CUST_ID IN (SELECT
CL.T14_CUST_ID  FROM T14_COVERAGE CV JOIN
T14_INSURANCE_POLICY_COVERAGE `IPC` ON
`IPC`.T14_COVERAGE_ID=CV.T14_COVERAGE_ID  JOIN T14_INSURANCE_POLICY IP
ON `IPC`.T14_AGREEMENT_ID=IP.T14_AGGREMENT_ID JOIN T14_CUSTOMER C
ON IP.T14_CUST_ID = C.T14_CUST_ID JOIN T14_CLAIM CL ON CL.T14_CUST_ID =
C.T14_CUST_ID JOIN T14_CLAIM_SETTLEMENT CLS ON CL.T14_CLAIM_ID =
CLS.T14_CLAIM_ID  JOIN T14_VEHICLE V ON V.T14_CUST_ID = C.T14_CUST_ID
WHERE T14_COVERAGE_AMOUNT>T14_CLAIM_AMOUNT  AND
T14_CLAIM_AMOUNT>(SUM_OF_INDEXES(CL.T14_CUST_ID,CLS.T14_CLAIM_ID,C
LS.T14_CLAIM_SETTLEMENT_ID,V.T14_VEHICLE_ID)));
```

RESULT OF THE QUERY:



t14_cust_id	t14_cust_fname	t14_cust_lname	t14_cust_DOB	t14_cust_gender	t14_cust_mob_number	t14_cust_email	t14_cust_passport_number	t14_cust_martial_status	t14_cust_ppS_number
3012	Amanda	Johnson	1974-12-04	f	9995922013	AmandaJohnson@gmail.com	AB33286BY	unmarried	7061231BA
3017	stan	Gunderson	1987-09-21	m	7654321212	stanGunderson@gmail.com	AB33286BF	unmarried	7051331VB
3011	Jason	Wilcox	1985-09-24	m	7421863294	JasonWilcox@gmail.com	AB33286BX	married	7061531MA
3016	Bobby	Trundle	1957-09-11	m	2714348789	BobbyTrundle@gmail.com	AB33286BD	married	6061531CA
3015	Lief	Gunderson	1963-01-22	m	3801428820	LiefGunderson@gmail.com	AB33286BC	married	7161531AA

CONCLUSION:

A complete Car Vehicle Insurance company database is completely implemented and all the given project queries are executed and are completely working fine giving at least one line of output. Each table consists of at least 10 tuples of data. Further developments for this project can be making a user interface for this database to perform INSERT, UPDATE, DELETE operations, and Normalisation of a few tables till 4th NF. To adjust the data according to queries functions are used ad stored procedures are used to remove and select the data. Views also assist in the implementation of queries.