**VEHICLE INSURANCE DATABASE**

Prepared by:

**DBMS(CS310)**

**Group-1:**

Akash Chaurasiya(20bcs001) Devansh Mahant (20bcs039)

Abhishake Dubey(20bcs002) Tushar Dhotre(20bcs042)

Aditya Tiwari(20bcs007) Anup Jujagar(20bcs065)

Ankit Kumar(20bcs017) Kartik Bhamare (20bcs064)

Mahendra Puniya (20bcs088) Pruthviraj Kamble (20bcs104)

**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

**Acknowledgement**

We would like to sincerely and profusely thank Dr Pramod Yelmewad, for his able guidance and support in completing the Project. We would also like to thank Dr Uma Seshadri for her guidance and giving us the opportunity to take up this project.

- Group-1 students

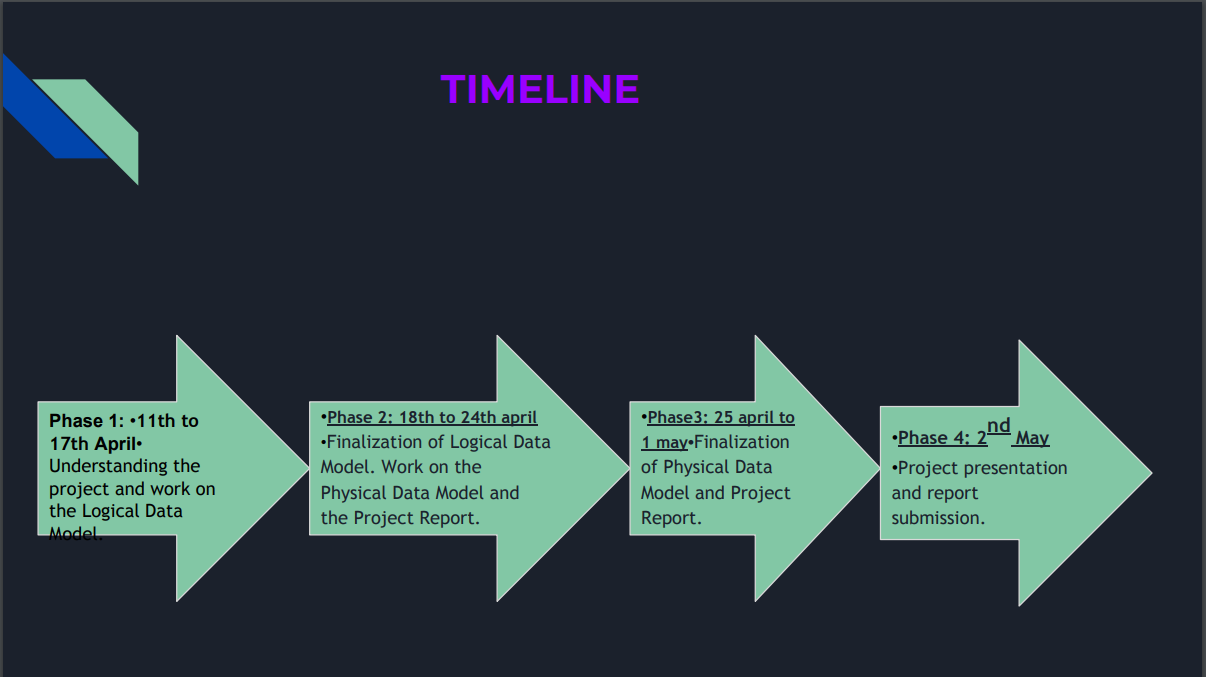
**AIM:**

This project provides an overall understanding of theoretical and practical concepts of DBMS, this project helpus in learning advanced modelling, normalization, transactional relational database design, SQL and Procedural language and SQL coding. In this project we get an experience to work on ‘MYSQL WORKBENCH’.

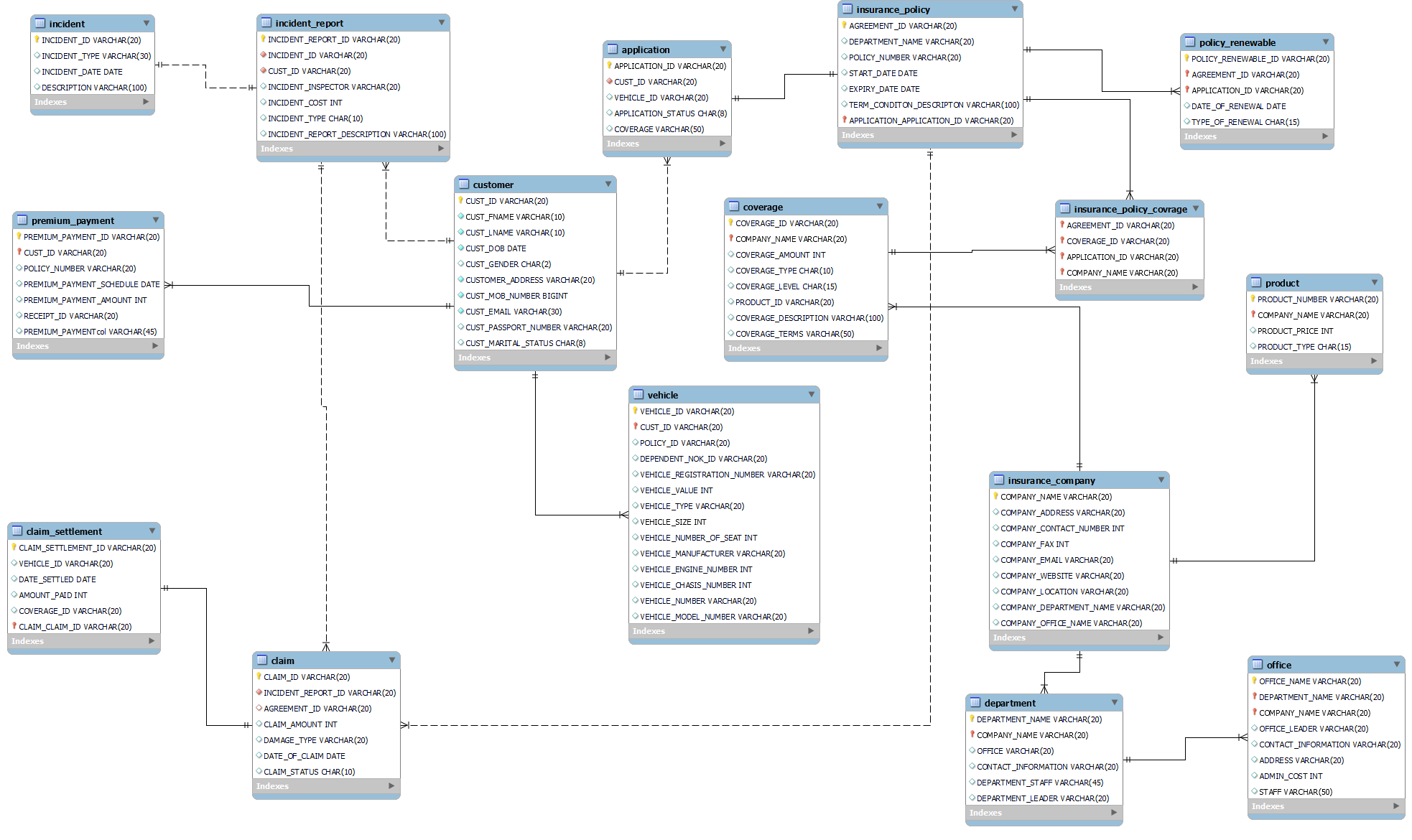
**APPROACH:**

After reading the pdf of vehicle insurance company, we tried to understand all the requirements which are needed to create an error free database. We created a conceptual data model to understand the relationship between different entities. We also created an entity relationship diagram (ER-Diagram) and created all the tables and inserted all relevant data to execute all queries.

**TIMELINE:**



**E-R DIAGRAM:**



**QUERIES:**

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

SELECT t1\_customer.CUST\_ID, t1\_customer.CUST\_FNAME, t1\_customer.CUST\_LNAME, t1\_customer.CUSTOMER\_ADDRESS,

t1\_customer.CUST\_EMAIL, t1\_customer.CUST\_MOB\_NUMBER, t1\_vehicle.VEHICLE\_ID,

t1\_vehicle.VEHICLE\_REGISTRATION\_NUMBER, t1\_vehicle.VEHICLE\_TYPE,

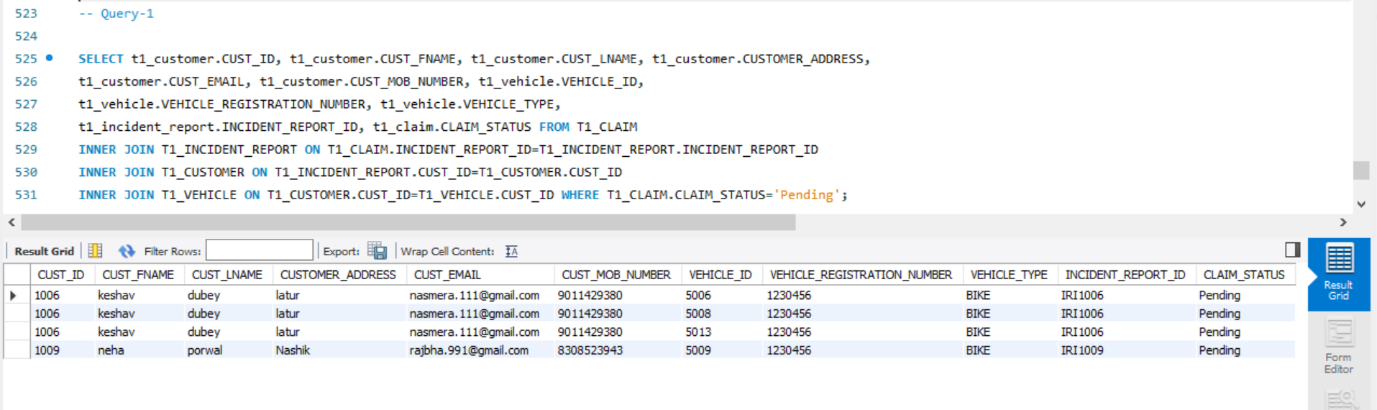
t1\_incident\_report.INCIDENT\_REPORT\_ID, t1\_claim.CLAIM\_STATUS FROM T1\_CLAIM

INNER JOIN T1\_INCIDENT\_REPORT ON T1\_CLAIM.INCIDENT\_REPORT\_ID=T1\_INCIDENT\_REPORT.INCIDENT\_REPORT\_ID

INNER JOIN T1\_CUSTOMER ON T1\_INCIDENT\_REPORT.CUST\_ID=T1\_CUSTOMER.CUST\_ID

INNER JOIN T1\_VEHICLE ON T1\_CUSTOMER.CUST\_ID=T1\_VEHICLE.CUST\_ID WHERE T1\_CLAIM.CLAIM\_STATUS='Pending';

Output :



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

SELECT SUM(t1\_customer.CUST\_ID) AS SUM\_CUST\_ID, t1\_customer.CUST\_ID,

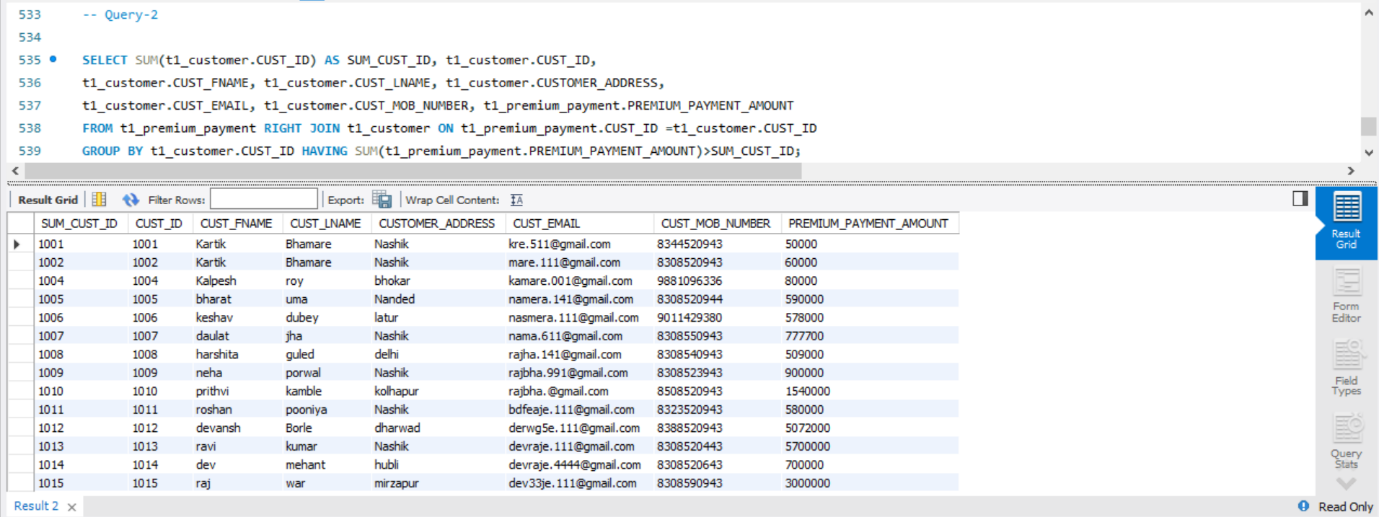
t1\_customer.CUST\_FNAME, t1\_customer.CUST\_LNAME, t1\_customer.CUSTOMER\_ADDRESS,

t1\_customer.CUST\_EMAIL, t1\_customer.CUST\_MOB\_NUMBER, t1\_premium\_payment.PREMIUM\_PAYMENT\_AMOUNT

FROM t1\_premium\_payment RIGHT JOIN t1\_customer ON t1\_premium\_payment.CUST\_ID =t1\_customer.CUST\_ID

GROUP BY t1\_customer.CUST\_ID HAVING SUM(t1\_premium\_payment.PREMIUM\_PAYMENT\_AMOUNT)>SUM\_CUST\_ID;

Output:



1. 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

SELECT \* FROM t1\_INSURANCE\_COMPANY

WHERE Company\_Name IN

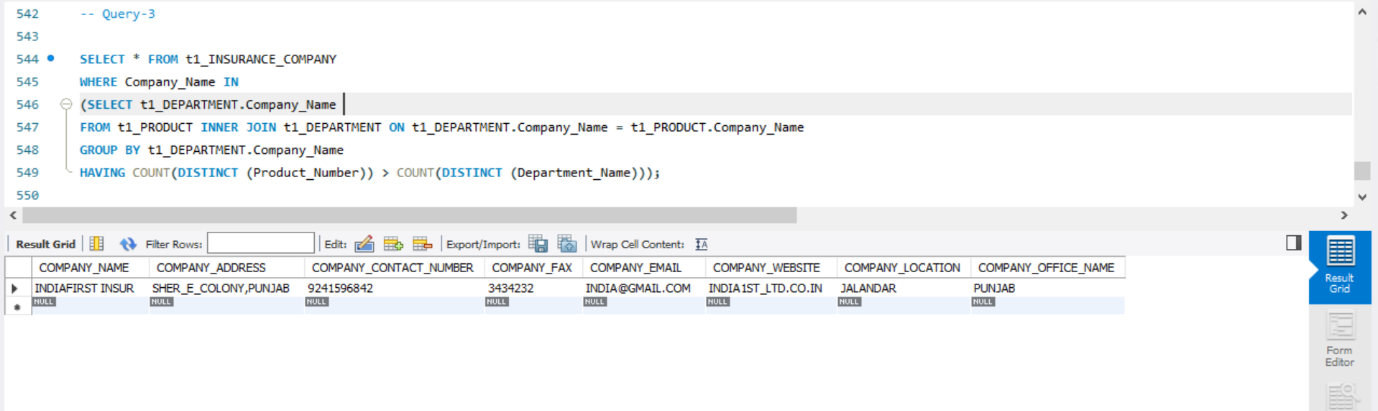
(SELECT t1\_DEPARTMENT.Company\_Name

FROM t1\_PRODUCT INNER JOIN t1\_DEPARTMENT ON t1\_DEPARTMENT.Company\_Name = t1\_PRODUCT.Company\_Name

GROUP BY t1\_DEPARTMENT.Company\_Name

HAVING COUNT(DISTINCT (Product\_Number)) > COUNT(DISTINCT (Department\_Name)));

Output:



1. 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

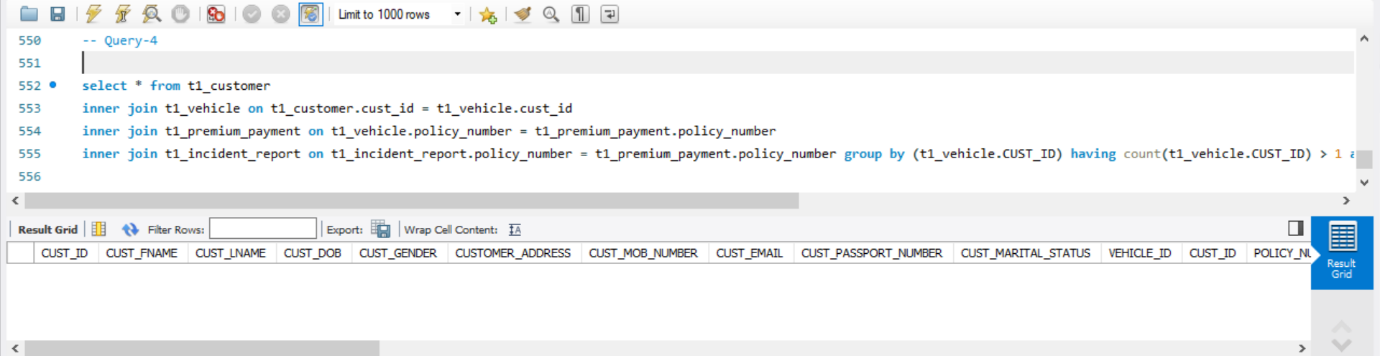
select \* from t1\_customer

inner join t1\_vehicle on t1\_customer.cust\_id = t1\_vehicle.cust\_id

inner join t1\_premium\_payment on t1\_vehicle.policy\_number = t1\_premium\_payment.policy\_number

inner join t1\_incident\_report on t1\_incident\_report.policy\_number = t1\_premium\_payment.policy\_number group by (t1\_vehicle.CUST\_ID) having count(t1\_vehicle.CUST\_ID) > 1 and t1\_premium\_payment.CUST\_ID is null ;

Output:



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

SELECT t1\_vehicle.VEHICLE\_ID,

t1\_vehicle.VEHICLE\_REGISTRATION\_NUMBER, t1\_vehicle.VEHICLE\_TYPE , t1\_vehicle.VEHICLE\_NUMBER , t1\_premium\_payment.PREMIUM\_PAYMENT\_AMOUNT

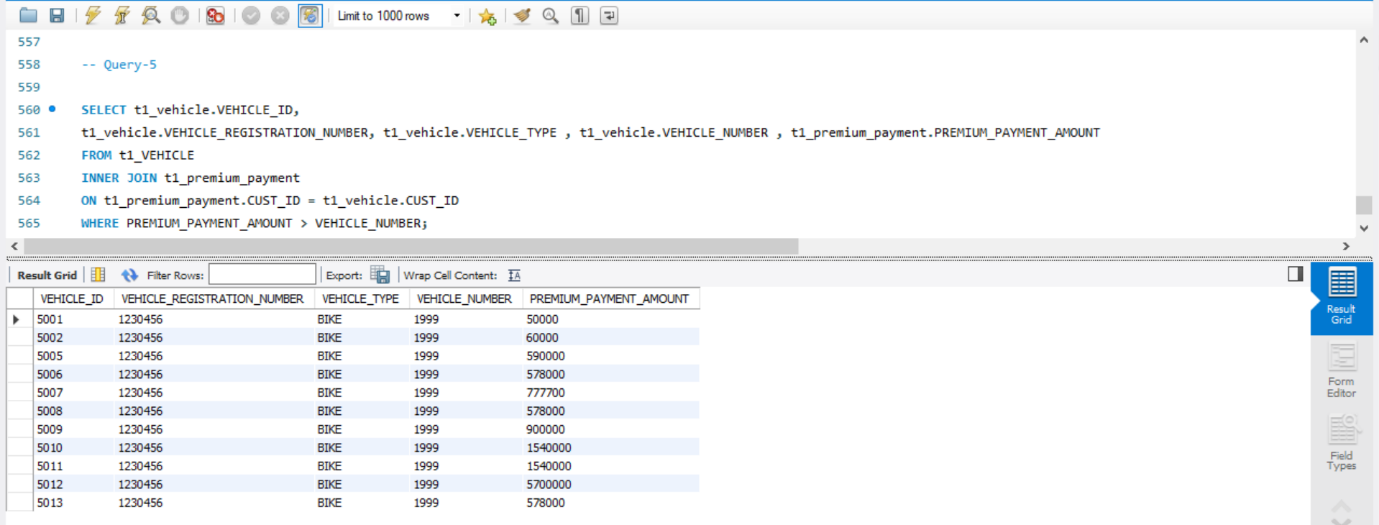
FROM t1\_VEHICLE

INNER JOIN t1\_premium\_payment

ON t1\_premium\_payment.CUST\_ID = t1\_vehicle.CUST\_ID

WHERE PREMIUM\_PAYMENT\_AMOUNT > VEHICLE\_NUMBER;

Output:



1. 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 t1\_customer.CUST\_ID, t1\_customer.CUST\_FNAME, t1\_customer.CUST\_LNAME, t1\_customer.CUST\_EMAIL,

t1\_customer.CUST\_MOB\_NUMBER, t1\_claim.CLAIM\_ID, t1\_claim\_settlement.CLAIM\_SETTLEMENT\_ID,

t1\_vehicle.VEHICLE\_ID, t1\_coverage.COVERAGE\_AMOUNT, t1\_claim.CLAIM\_AMOUNT

FROM t1\_customer JOIN t1\_incident\_report ON t1\_customer.CUST\_ID=t1\_incident\_report.CUST\_ID

JOIN t1\_claim ON t1\_incident\_report.INCIDENT\_REPORT\_ID=t1\_claim.INCIDENT\_REPORT\_ID

JOIN t1\_claim\_settlement ON t1\_claim.CLAIM\_ID=t1\_claim\_settlement.CLAIM\_ID

JOIN t1\_vehicle ON t1\_vehicle.CUST\_ID=t1\_customer.CUST\_ID

JOIN t1\_application ON t1\_customer.CUST\_ID=t1\_application.CUST\_ID

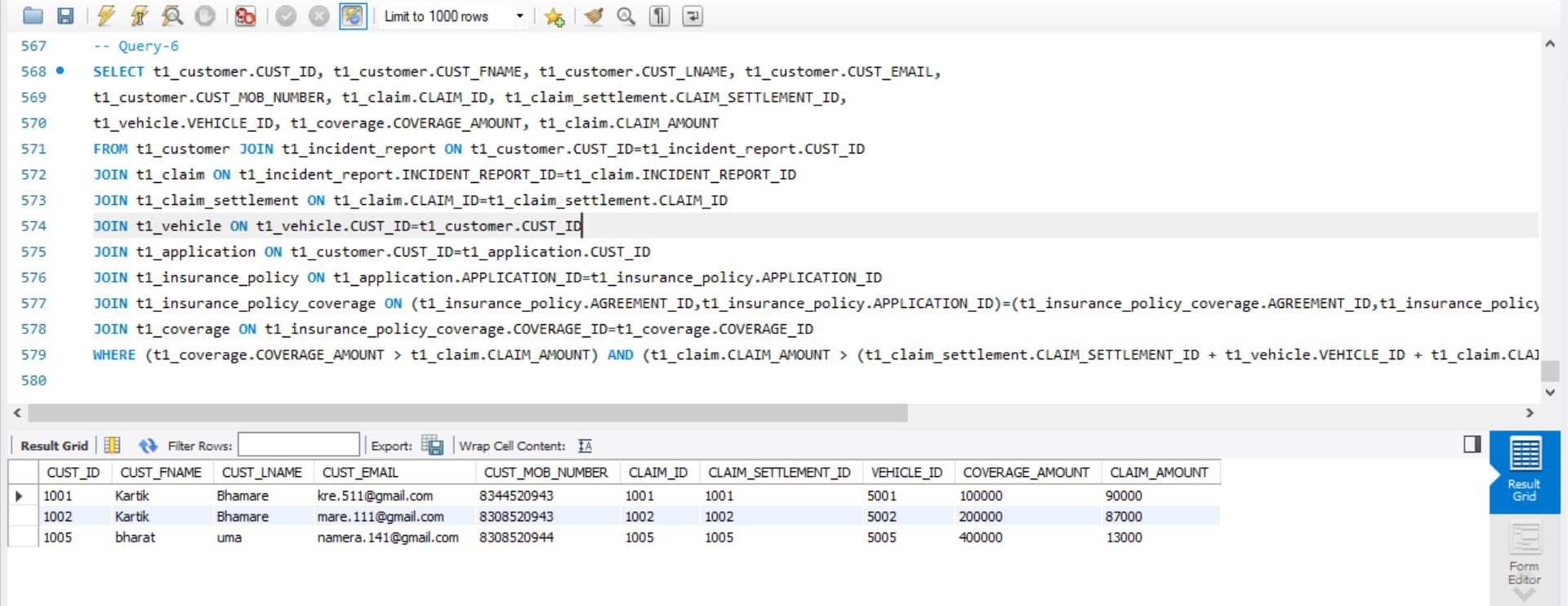
JOIN t1\_insurance\_policy ON t1\_application.APPLICATION\_ID=t1\_insurance\_policy.APPLICATION\_ID

JOIN t1\_insurance\_policy\_coverage ON (t1\_insurance\_policy.AGREEMENT\_ID,t1\_insurance\_policy.APPLICATION\_ID)=(t1\_insurance\_policy\_coverage.AGREEMENT\_ID,t1\_insurance\_policy\_coverage.APPLICATION\_ID)

JOIN t1\_coverage ON t1\_insurance\_policy\_coverage.COVERAGE\_ID=t1\_coverage.COVERAGE\_ID

WHERE (t1\_coverage.COVERAGE\_AMOUNT > t1\_claim.CLAIM\_AMOUNT) AND (t1\_claim.CLAIM\_AMOUNT > (t1\_claim\_settlement.CLAIM\_SETTLEMENT\_ID + t1\_vehicle.VEHICLE\_ID + t1\_claim.CLAIM\_ID + t1\_customer.CUST\_ID));

Output:



**CHALLENGES:**

1. **FOREIGN KEY CONSTRAINT:**

We have to change the sequence of table while creating their schema to ensure that is foreign key used has

been initialised as primary key.

1. **REWORKING THE DATABASE:**

We have to rework on database to satisfies the queries.

1. **Discrepancy in values between team members.**

**CONCLUSION:**

We have successfully created the database for vehicle insurance company and executed all the given queries.