A project report on

A DATABASE FOR A VEHICLE INSURANCE COMPANY

Under the Guidance of

Dr. Uma Sheshadri

Dr. Pramod Yelmewad

Dr. Supriya Nadiger

Course Name – Database Management Systems

Course Code – CS310

Course Instructors – **Dr. Uma Sheshadri**

Dr. Pramod Yelmewad

Team Members

Team No.2 – Pioneers

20BCS087-Mudiga Bharadwaz

20BCS125-Somisetty Sai Praneeth

20BCS126-Soumya Ranjan Sahu

20BCS128-Sree Deva Krupananda

20BCS129-Sri Hari L

20BCS137-Vineeth S R

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

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 for allowing us to take up this project.

Team No.2 – Pioneers

AIM:

This project provides an overall understanding of the theoretical and practical concepts of DBMS. This project helps us in learning advanced modeling, normalization, transactional relational database design, SQL and Procedural language, and SQL coding. In this project, we got the experience to work on MySQL Workbench.

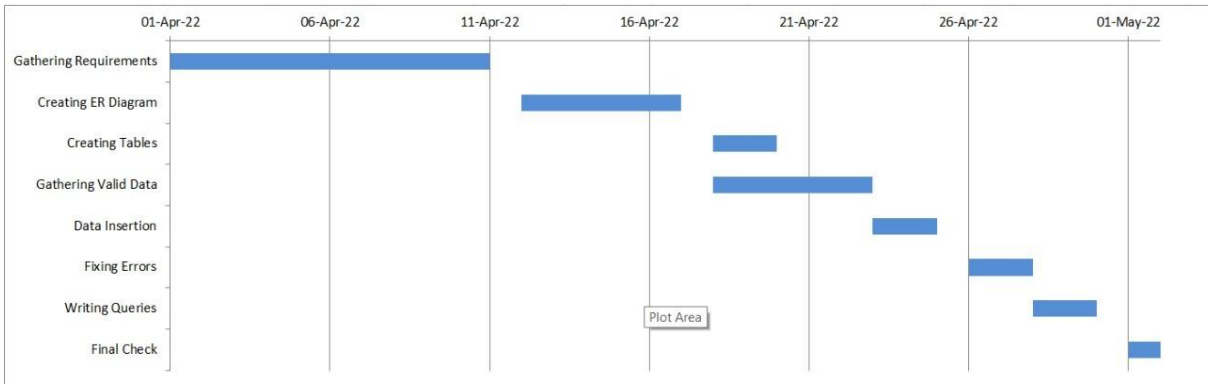
APPROACH:

After reading the document on database design for a vehicle insurance company, we tried to understand all the requirements which are needed to create an error-free database. We also created a physical data model (PDM) for all the 22 tables and inserted relevant data to execute all queries.

PROCEDURE:

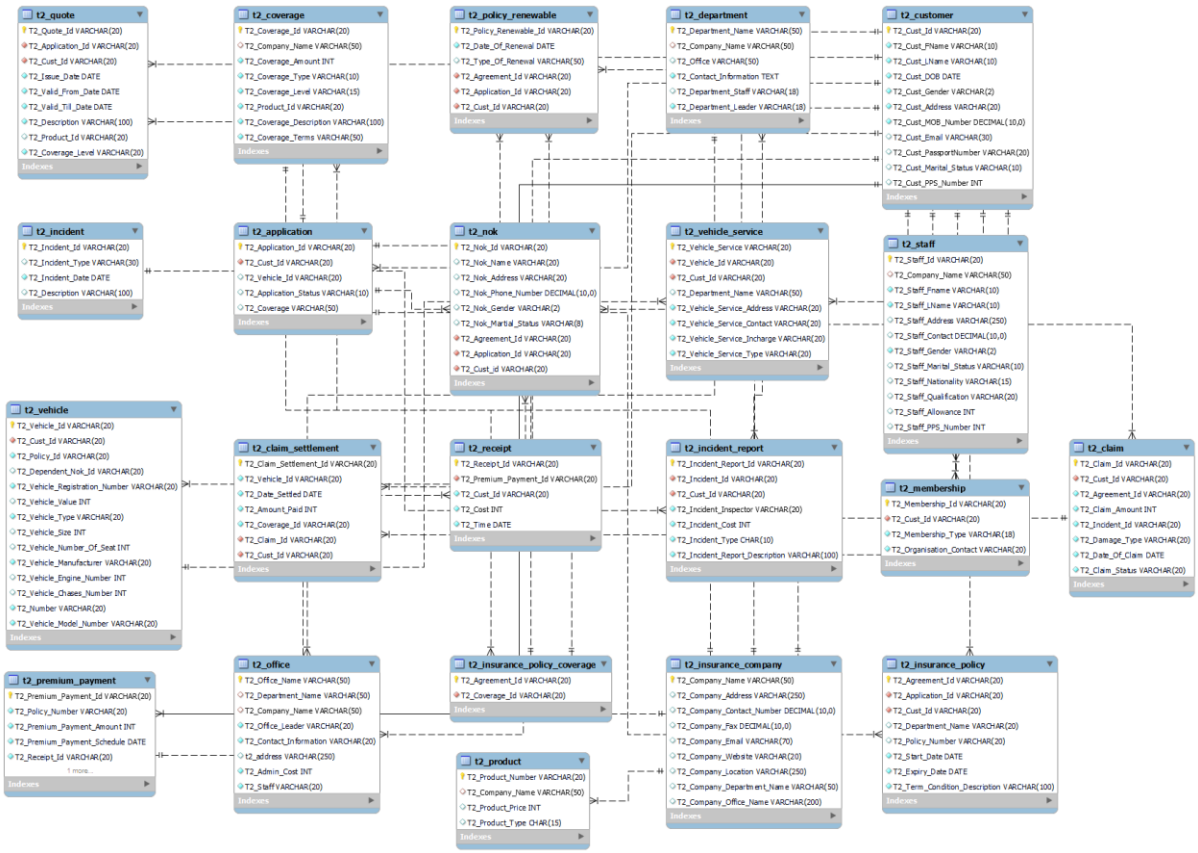
We created tables using creating a statement and then inserted data into it using insert into statements and then in the middle if we had to change the tables. We used alter statements, and then to update statements we used update statements to update the values, and then after creating the tables and inserting them, we had to alter most of the table's data according to the queries, and then we wrote queries mainly using joins, subqueries, group by, order and many other statements.

GANTT CHART:



PHYSICAL DATA MODEL (PDM):

This physical data model contains all the twenty-two tables of the given database for a vehicle insurance company with valid primary key and foreign key constraints also the relationships between the tables.



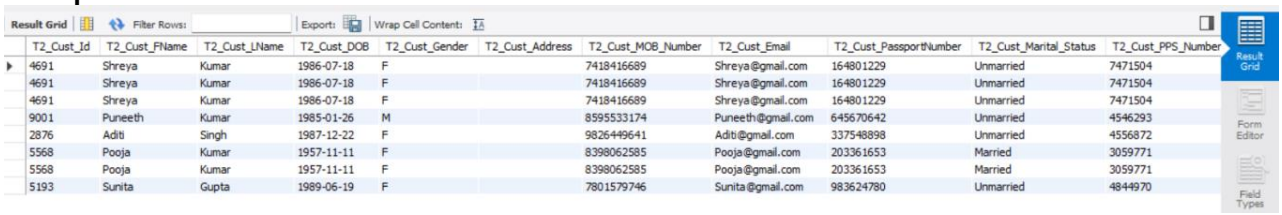
QUERIES:

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

Code:

```
select
  distinct t2_customer.*,
  t2_vehicle.*
from
  t2_customer
  inner join t2_vehicle t2_vehicle on t2_customer.t2_cust_id = t2_vehicle.t2_cust_id
  inner join t2_claim on t2_claim.t2_cust_id = t2_customer.t2_cust_id
WHERE
  t2_claim.T2_Incident_Id IS NOT NULL
  AND t2_claim_status like 'pending';
```

Output:



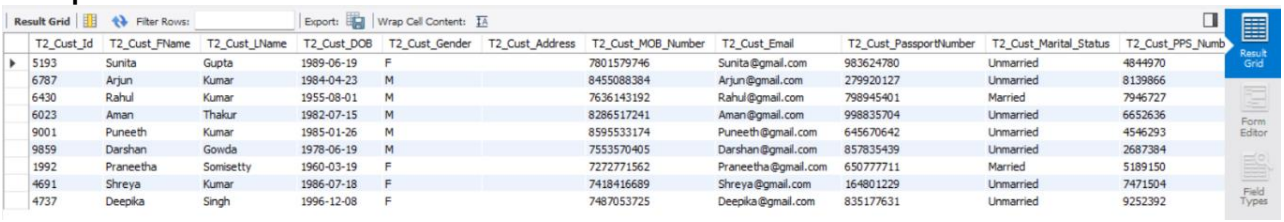
T2_Cust_Id	T2_Cust_FName	T2_Cust_LName	T2_Cust_DOB	T2_Cust_Gender	T2_Cust_Address	T2_Cust_MOB_Number	T2_Cust_Email	T2_Cust_PassportNumber	T2_Cust_Marital_Status	T2_Cust_PPS_Number
4691	Shreya	Kumar	1986-07-18	F		7418416689	Shreya@gmail.com	164801229	Unmarried	7471504
4691	Shreya	Kumar	1986-07-18	F		7418416689	Shreya@gmail.com	164801229	Unmarried	7471504
4691	Shreya	Kumar	1986-07-18	F		7418416689	Shreya@gmail.com	164801229	Unmarried	7471504
9001	Puneeth	Kumar	1985-01-26	M		8595533174	Puneeth@gmail.com	645670642	Unmarried	4546293
2876	Aditi	Singh	1987-12-22	F		9826449641	Aditi@gmail.com	337548898	Unmarried	4556872
5568	Pooja	Kumar	1957-11-11	F		8398062585	Pooja@gmail.com	203361653	Married	3059771
5568	Pooja	Kumar	1957-11-11	F		8398062585	Pooja@gmail.com	203361653	Married	3059771
5193	Sunita	Gupta	1989-06-19	F		7801579746	Sunita@gmail.com	983624780	Unmarried	4844970

2. Retrieve customer details who have a premium payment amount greater than the sum of all the customer IDs in the database – premium payment, customer

Code:

```
select
  t2_customer.*,
  t2_premium_payment.*
from
  t2_customer
  RIGHT JOIN t2_premium_payment on t2_customer.t2_cust_id = t2_premium_payment.t2_cust_id
WHERE
  t2_premium_payment.T2_Premium_Payment_Amount > (
    SELECT
      SUM(CAST(T2_Cust_Id AS UNSIGNED))
    FROM
      T2_CUSTOMER
  );
```

Output:



T2_Cust_Id	T2_Cust_FName	T2_Cust_LName	T2_Cust_DOB	T2_Cust_Gender	T2_Cust_Address	T2_Cust_MOB_Number	T2_Cust_Email	T2_Cust_PassportNumber	T2_Cust_Marital_Status	T2_Cust_PPS_Numb
5193	Sunita	Gupta	1989-06-19	F		7801579746	Sunita@gmail.com	983624780	Unmarried	4844970
6787	Arjun	Kumar	1984-04-23	M		8455088384	Arjun@gmail.com	279920127	Unmarried	8139866
6430	Rahul	Kumar	1955-08-01	M		7636143192	Rahul@gmail.com	798945401	Married	7946727
6023	Aman	Thakur	1982-07-15	M		8286517241	Aman@gmail.com	998835704	Unmarried	6652636
9001	Puneeth	Kumar	1985-01-26	M		8595533174	Puneeth@gmail.com	645670642	Unmarried	4546293
9859	Darshan	Gowda	1978-06-19	M		7553570405	Darshan@gmail.com	857835439	Unmarried	2687384
1992	Praneetha	Somisetty	1960-03-19	F		7272771562	Praneetha@gmail.com	650777711	Married	5189150
4691	Shreya	Kumar	1986-07-18	F		7418416689	Shreya@gmail.com	164801229	Unmarried	7471504
4737	Deepika	Singh	1996-12-08	F		7487053725	Deepika@gmail.com	835177631	Unmarried	9252392

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

Code:

```
SELECT
  DISTINCT (t2_COMPANY_NAME)
FROM
  t2_insurance_company
WHERE
  t2_COMPANY_NAME IN (
    SELECT
      t2_insurance_company.t2_COMPANY_NAME
    FROM
      t2_insurance_company
    WHERE
      t2_insurance_company.t2_Company_Name IN (
        SELECT
          t2_Company_Name
        FROM
          t2_office
        GROUP BY
          t2_Company_Name
        HAVING
          t2_Company_Name IN (
            SELECT
              t2_office.t2_Company_Name
            FROM
              t2_product
              INNER JOIN t2_office ON t2_office.t2_company_name = t2_product.t2_Company_Name
            GROUP BY
              t2_office.t2_Company_Name
            HAVING
              COUNT(DISTINCT (t2_product_number)) > COUNT(DISTINCT (t2_department_name))
          )
      )
```

```
)
AND t2_COMPANY_NAME IN (
  SELECT
    t2_customer.t2_COMPANY_NAME
  FROM
    t2_insurance_company t2_customer
    INNER JOIN t2_product ON t2_product.t2_COMPANY_NAME = t2_customer.t2_COMPANY_NAME
  GROUP BY
    t2_product.t2_COMPANY_NAME
  HAVING
    COUNT(*) > ALL (
      SELECT
        COUNT(*)
      FROM
        t2_insurance_company
      GROUP BY
        t2_COMPANY_NAME
      HAVING
        COUNT(t2_COMPANY_LOCATION) > 1
    )
);
```

Output:

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 an accident

Code:

```
select   *
from
  T2_CUSTOMER
where
  T2_CUSTOMER.T2_Cust_id in (
    select
      T2_Cust_Id
    from
      T2_VEHICLE
    where
      t2_VEHICLE.T2_Policy_Id not in (
        select
          T2_Policy_Number
        from
          t2_PREMIUM_PAYMENT      )
      and t2_VEHICLE.t2_Cust_Id in (
        select
          t2_Cust_Id
        from
          t2_VEHICLE
        GROUP BY
          t2_VEHICLE.t2_Cust_Id
        having
          count(t2_VEHICLE.t2_Cust_Id) > 1       )
      and t2_VEHICLE.t2_Cust_Id in (
        select
          t2_Cust_Id
        from
          t2_INCIDENT_REPORT
        where
          t2_Incident_Type = 'accident'
```

Output:

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

T2_Cust_Id	T2_Cust_FName	T2_Cust_LName	T2_Cust_DOB	T2_Cust_Gender	T2_Cust_Address	T2_Cust_MOB_Number	T2_Cust_Email	T2_Cust_PassportNumber	T2_Cust_Marita
4691	Shreya	Kumar	1986-07-18	F		7418416689	Shreya@gmail.com	164801229	Unmarried
5568	Pooja	Kumar	1957-11-11	F		8398062585	Pooja@gmail.com	203361653	Married
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

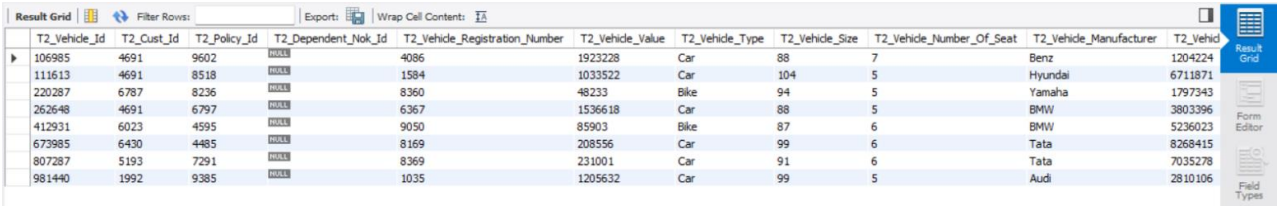
Result Grid | Form Editor

5. Select all vehicles which have a premium more than their vehicle number.

Code:

```sql
SELECT
  t2_vehicle.*
FROM
  t2_vehicle
  INNER JOIN t2_customer ON t2_customer.T2_Cust_Id = t2_vehicle.T2_Cust_Id
  INNER JOIN t2_premium_payment ON t2_premium_payment.T2_Cust_Id = t2_customer.T2_Cust_Id
WHERE
  CAST(t2_vehicle.T2_Vehicle_Id AS UNSIGNED) < t2_premium_payment.T2_Premium_Payment_Amount;
```

Output:



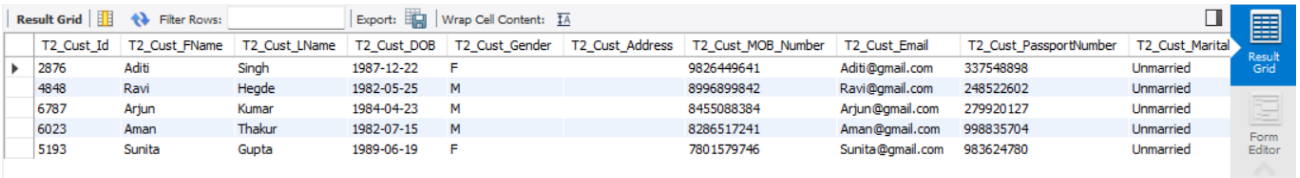
T2_Vehicle_Id	T2_Cust_Id	T2_Policy_Id	T2_Dependent_Nok_Id	T2_Vehicle_Registration_Number	T2_Vehicle_Value	T2_Vehicle_Type	T2_Vehicle_Size	T2_Vehicle_Number_Of_Seat	T2_Vehicle_Manufacturer	T2_Vehicl
106985	4691	9602	NULL	4086	1923228	Car	88	7	Benz	1204224
111613	4691	8518	NULL	1584	1033522	Car	104	5	Hyundai	6711871
220287	6787	8236	NULL	8360	48233	Bike	94	5	Yamaha	1797343
262648	4691	6797	NULL	6367	1536618	Car	88	5	BMW	3803396
412931	6023	4595	NULL	9050	85903	Bike	87	6	BMW	5236023
673985	6430	4485	NULL	8169	208556	Car	99	6	Tata	8268415
807287	5193	7291	NULL	8369	231001	Car	91	6	Tata	7035278
981440	1992	9385	NULL	1035	1205632	Car	99	5	Audi	2810106

6. Retrieve Customer details whose Claim Amount is less than the Coverage Amount and Claim Amount is greater than the Sum of (CLAIM_SETTLEMENT_ID, VEHICLE_ID, CLAIM_ID, CUST_ID)

Code:

```sql
SELECT
  T2_CUSTOMER.*
FROM
  T2_CUSTOMER
  INNER JOIN T2_VEHICLE ON T2_VEHICLE.T2_Cust_Id = T2_CUSTOMER.T2_Cust_Id
  INNER JOIN T2_CLAIM ON T2_CLAIM.T2_Cust_Id = T2_CUSTOMER.T2_Cust_Id
  INNER JOIN t2_insurance_policy_coverage on t2_insurance_policy_coverage.T2_Agreement_Id = t2_claim.T2_Agreement_Id
  INNER JOIN T2_COVERAGE ON T2_COVERAGE.T2_Coverage_Id = T2_INSURANCE_POLICY_COVERAGE.T2_Coverage_Id
  INNER JOIN T2_CLAIM_SETTLEMENT ON T2_CLAIM_SETTLEMENT.T2_Claim_Id = T2_CLAIM.T2_Claim_Id
WHERE
  T2_CLAIM.T2_Claim_Amount < T2_COVERAGE.T2_Coverage_Amount
  AND T2_COVERAGE.T2_Coverage_Amount > (
    CAST(
      T2_CLAIM_SETTLEMENT.T2_Claim_Settlement_Id AS UNSIGNED
    ) + CAST(T2_VEHICLE.T2_Vehicle_Id AS UNSIGNED) + CAST(T2_CLAIM.T2_Claim_Id AS UNSIGNED) + CAST(T2_CUSTOMER.T2_Cust_Id AS UNSIGNED)
  );
```

Output:



T2_Cust_Id	T2_Cust_FName	T2_Cust_LName	T2_Cust_DOB	T2_Cust_Gender	T2_Cust_Address	T2_Cust_MOB_Number	T2_Cust_Email	T2_Cust_PassportNumber	T2_Cust_Marital
2876	Aditi	Singh	1987-12-22	F		9826449641	Aditi@gmail.com	337548898	Unmarried
4848	Ravi	Hegde	1982-05-25	M		8996899842	Ravi@gmail.com	248522602	Unmarried
6787	Arjun	Kumar	1984-04-23	M		8455088384	Arjun@gmail.com	279920127	Unmarried
6023	Aman	Thakur	1982-07-15	M		8286517241	Aman@gmail.com	998835704	Unmarried
5193	Sunita	Gupta	1989-06-19	F		7801579746	Sunita@gmail.com	983624780	Unmarried

CHALLENGES:

1. Data too long error
2. We had to change the datatype
3. We had to change the datatype of a primary key by cast function while writing query
4. To retrieve data we need to make some changes to data in tables according to the query

CONCLUSION:

There were some big and small challenges but we succeeded in making a functional DB. We made a physical data model (PDM) containing all twenty-two tables and we created a code to be run in MySQL Workbench. After going through the challenges mentioned above finally, we have successfully created the database for a vehicle insurance company and executed all the given queries.