**TEAM: TECHIES CLUB (NUMBER 8)**

CS310- DBMS – GROUP PROJECT:

CAR VEHICLE INSURANCE

**Abstract**

 As a result of an accident, the insurer assumes the risk of any losses the owner or operator of a car may suffer as a result of damage to property or persons. We created tables in the given project and inserted data based on the queries required. The data was obtained from some insurance companies. The queries are written in an approachable manner. Using this project, we gain an understanding of the importance and depth of database management systems. As a result, we learn that DBMS is an easier way to store data.

**Introduction**

Database management systems are software applications used to manage and manipulate data in a database. Since most application performance issues originate in the database, monitoring and optimizing your database is essential to your business.

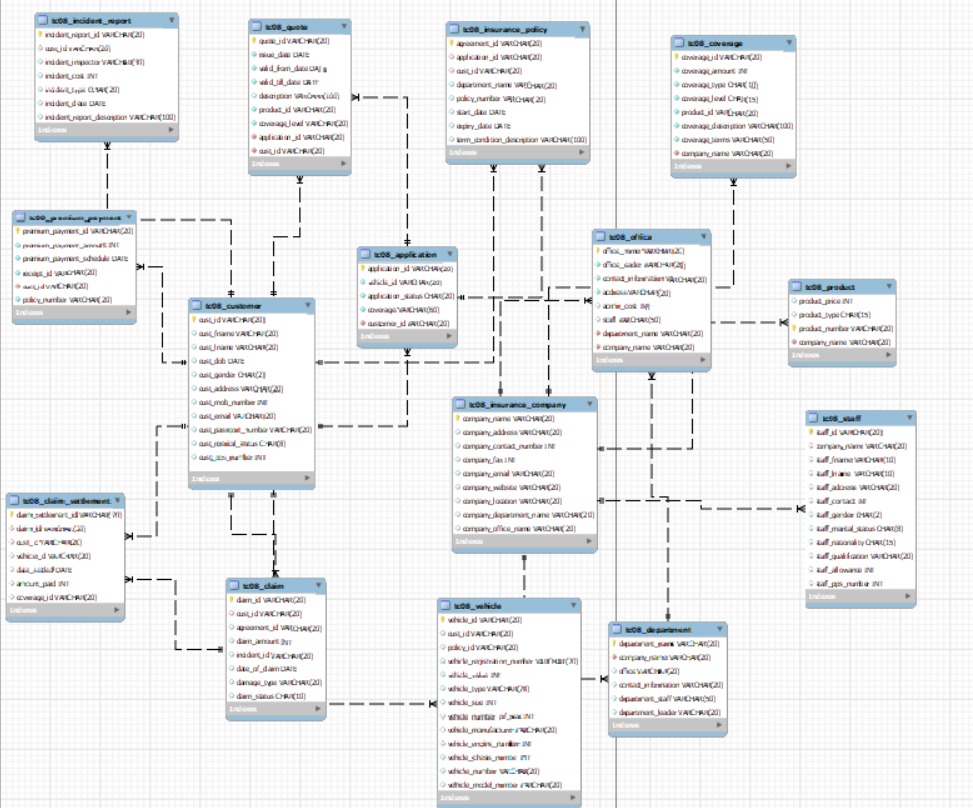
Aim:   Provide a comprehensive understanding and practical experience in data modelling, normalization techniques, transactional relational database design, and SQL query implementation.

Approach:   Our goal was to understand all of the requirements required to create an error-free database after reading the pdf of the auto insurance company. To understand how different things relate to each other, we constructed a conceptual data model. As well as the tables and data necessary to run all of the queries, we generated the entity relationship diagram (ER-Diagram).

**MODELS**

**Physical Data Model:**

A physical data model is a database-specific model that represents relational data objects (for example, tables, columns, primary and foreign keys) and their relationships.It represents how the actual database is built.It introduces the database-specific context missing in conceptual and logical data models.



**Conceptual Data Model:**

It describes the database at a very high level and is useful to understand the needs or requirements of the database.It provides a data-centric perspective of the organization by documenting how different business entities relate to one another.

**Logical Data Model:**

A type of data model that describes data elements in detail and is used to develop visual understandings of data entities, attributes, keys, and relationships.It describes data as much as

possible, without regard to how they will be physically implemented in the database.

**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

Solution:

First and foremost thing is that we have 2 conditions:

Condition-1: which customer has been involved in an incident.

Condition-2: which customer or vehicle’s claim status is pending.

So, we need to get the details of customer and vehicle from Customer table, Vehicle table which satisfies the above two conditions.

So, my sql code is:

**SELECT TC08\_customer.\*,TC08\_vehicle.\***

**FROM TC08\_customer**

Next thing is that customer should apply for vehicle.

Here we are using two tables to return the data so we need to apply joins.

So, the MySQL code is :

**INNER JOIN TC08\_vehicle**

**ON TC08\_customer.cust\_id = TC08\_vehicle.cust\_id**

In the above code we are using Inner Join to join the customer table and vehicle table. This is because we can get the customers who are common in both the tables. Which means the customer having a vehicle.

customer should also apply for claim .

So, the MySQL code is :

**INNER JOIN TC08\_claim**

**ON TC08\_customer.cust\_id = TC08\_claim.cust\_id**

In the above code we are using Inner Join to join the customer table and claim table. This is because we can get the customers who are common in both the tables. Which means the customer involved in the incident.

Till now we wrote the basic conditions of the query now the main conditions should be written one by one.

#Condition: Vehicle is involved in an incident.

Mysql code is:

**WHERE TC08\_claim.incident\_id IS NOT NULL**

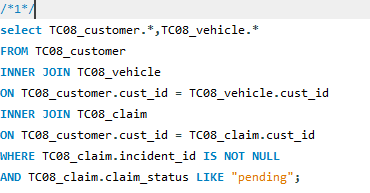
Here we used ‘where’ to specify conditions on columns.

#Condition: Vehicle’s claim status is pending.

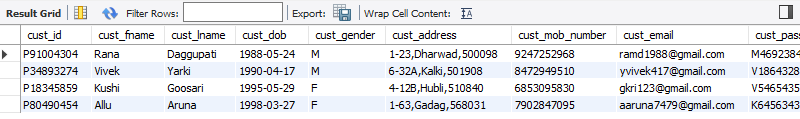
Mysql code is:

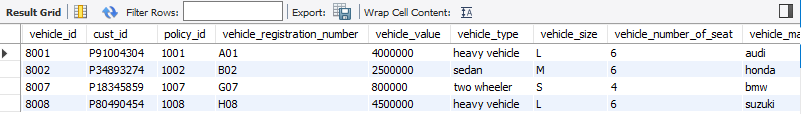
**AND TC08\_claim.claim\_status LIKE "pending"**

Our Final Query:



Retrieved Data:





So here we are getting the customer details and vehicle details who were involved in an incident and claim status is pending.

Query2:

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

Solution:

The main condition mentioned in the query is:

Condition-1: premium payment should be greater than the sum of all the customer ids

So, we need to get the customer details form the customer table who has the premium payment amount greater than the sum of all customer ids

So, my sql code is:

**SELECT TC08\_customer.\***

Next thing is that customer should apply for premium payment

Here we are using two tables to return the data so, we need to apply joins

So, the MySQL code is :

**inner JOIN TC08\_premium\_payment**

**ON TC08\_customer.cust\_id = TC08\_premium\_payment.cust\_id**

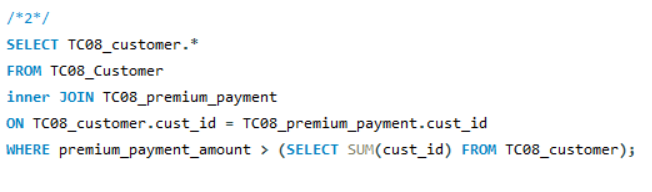
In the above code we using Inner Join to join Customers table and Premium Payment table. This is because, we can get the customers who are common in both the tables. Which means the customer applied for the premium payment and it can pay or not paid yet.

Now the condition is premium payment should be greater than the sum of all the customer ids

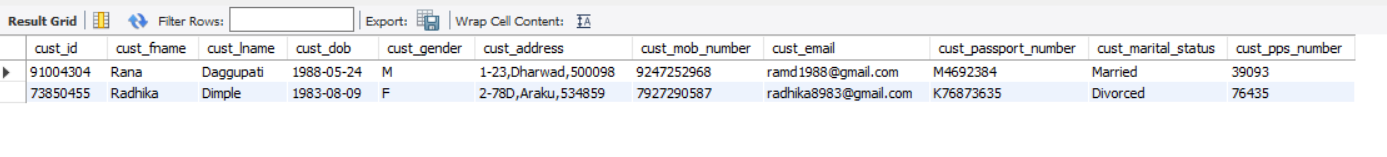
So to check this condition we have wrote this MySQL code :

**WHERE premium\_payment\_amount > (SELECT SUM(cust\_id) FROM TC08\_customer);**

🡪Our Final Query:



🡪Retrieved Data:

****

so here we are getting the customer details who has premium payment greater than the sum of all customer ids.

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

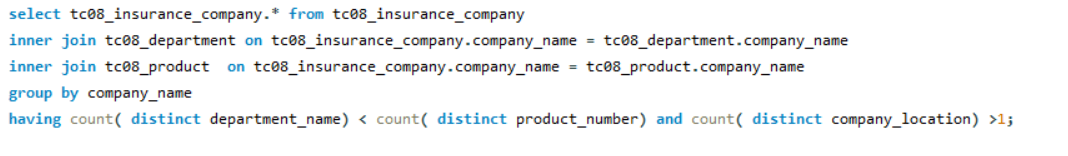
We have to satisfy two conditions in this problem to get desired output

* Number of products is greater than departments
* Departments are located in more than one location

To satisfy above conditions, I am using inner join concept, first I applied inner join for department table where company name from insurance company table and company name from department table should be same. Similarly I applied inner join for product table where company name from insurance company table is equal to company name from product table.

Now, I used group by statement to get the all the rows of company name where number of products is greater than departments ( I used distinct keyword to select different departments, to avoid duplicate values of department).

SQL query:



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.

Solution:

First and fore most thing is that we have 3 conditions

Condition 1 – Customer having more than one vehicle

Condition 2 – Premium for the vehicle is not paid

Condition 3 – Vehicle is involved in the Accident

So, we need to get the details from Customer table, Vehicle table, Premium Payment table, Incident table.

We have to return those customer details who are passing in all the above 3 conditions.

So, mysql code is

🡪



Next thing is that customer should apply for premium payment

Here we are using two tables to return the data so, we need to apply joins

So, the mysql code is

🡪

In the above code we using Inner Join to join Customers table and Premium Payment table. This is because, we can get the customers who are common in both the tables. Which means the customer applied for the premium payment and it can pay or not paid yet.

Next thing is that customer having any incident report. So, for getting this data we have to join Customer table with Incident table.

So, the mysql code is:

🡪



In the above code we using Inner Join to join Customers table and Incident table. This is because, we can get the customers who are common in both the tables. Which means the customer met with an incident.

Till now we wrote the basic conditions of the query now the main conditions should be written one by one.

#Condition: Vehicle is involved in the Accident

Mysql code is:

🡪



In the Incident table we have an attribute or column named Incident\_type which have details of reason of why the customer is asking for claim. For Example, Theft, Damage, Accident, etc…

In the above code we are trying to fetch the details of customer whose incident type is an Accident. Here we used a keyword ‘like’ it is used to find the certain word in the sentence of the column.

If it encounters with ‘Accident’ word then it will return those data.

#Condition: Premium for the vehicle is not paid.

Mysql code is:

🡪



In Premium Payment table we have an attribute named receipt\_id. The main purpose of the receipt\_id is, it will be generated if the customer pays the premium amount otherwise it will not be generated.

So, the primary logic for this condition, we have to get the data of customer who has not paid the premium amount so we can easily get the details if the respective row at the receipt\_id is null.

#Condition: Customer having more than one vehicle

Mysql code is:

🡪



For the third condition we wrote a sub query,

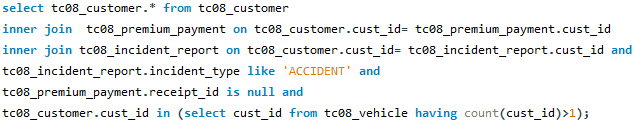
The main logic in the sub query is, if one customer is having 1 or more vehicles then in vehicles table and in the customer\_id column the respective customer\_id will be repeated.

So, we selected the vehicle table and we used an aggregate function ‘count’

If the count is greater than 1 means one of the customers is having multiple vehicles.

If all these 3 conditions satisfies then we will get the customer details.

🡪Our Final Query:



🡪Retrieved Data:



We are getting a customer who has more than one vehicle and one of the vehicle’s Premium Payment is not done and that vehicle is met with an Accident.

Query-5:

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

Solution:

The condition mentioned in the query is:

Premium of the vehicle should be more than its vehicle number.

The final output should be the details of the vehicles which satisfies the above condition.

Mysql Code:

**Select tc08\_premium\_payment.cust\_id, tc08\_premium\_payment.premium\_payment\_amount,**

**tc08\_vehicle.vehicle\_id, tc08\_vehicle.vehicle\_number**

Here I have selected customer id(cust\_id) and premium payment amount(premium\_payment\_amount) from premium payment table(tc08\_premium\_payment), vehicle id(vehicle\_id) and vehicle number(vehicle\_number) from vehicle table(tc08\_vehicle).

**From tc08\_premium\_payment**

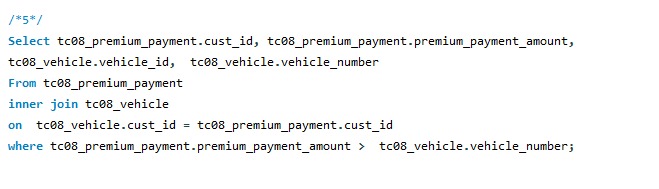
**inner join tc08\_vehicle**

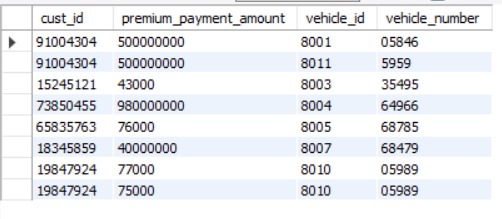
**on tc08\_vehicle.cust\_id = tc08\_premium\_payment.cust\_id**Here I used the inner join for joining the customer id from vehicle table and customer id from premium payment table. From this we can get the details of the customers who are common in both the tables.

**where tc08\_premium\_payment.premium\_payment\_amount > tc08\_vehicle.vehicle\_number;**Here the main condition comes where the premium payment amount of the vehicle in the premium payment table is greater than the vehicle number from the vehicle table.

From the previous join we get the common customer details which are common from both the tables and now with the above condition we are further eliminating the details of those customers whose vehicle number greater than premium payment amount and finally we are getting the only customer details of those whose premium payment amount is greater than the vehicle number.

**🡪Our Final Query:**

**  
  
🡪Retrieved Data/Output:**



These are the result/output data we got from the data we have.

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)

SOLUTION

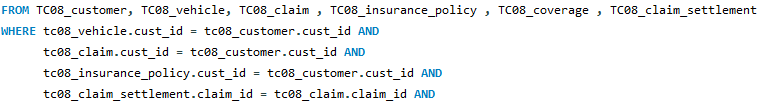
The Given question is divided into two parts:

1) The Claim amount of the vehicle involved in an incident is less than the Coverage Amount agreed by the insurance company

2) The Claim Amount is greater than the sum of the following id’s CLAIM\_ID,CLAIM\_SETTLEMENT\_ID,CUST\_ID and VEHICLE\_ID

The details from these respective tables are taken and the output for the above 2 conditions are returned

The mysql code for it is:

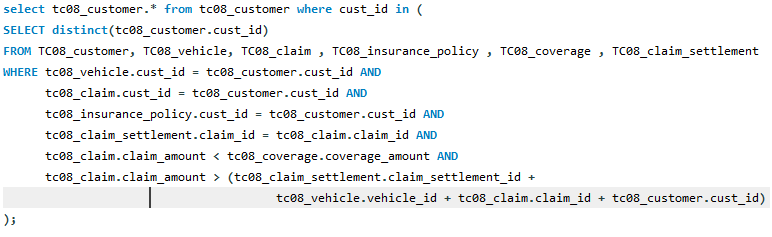


The next condition is that the claim amount is to be greater than sum of CLAIM\_ID,CLAIM\_SETTLEMENT\_ID,CUST\_ID and VEHICLE\_ID

The code for it is



The query is:



**TABLES THAT WE SELECTED TO SOLVE THE QUERIES:**

**CUSTOMER TABLE**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| cust\_id | varchar(20) | PK |
| cust\_fname | varchar(10) |  |
| cust\_lname | varchar(10) |  |
| cust\_dob | date |  |
| cust\_gender | char(2) |  |
| cust\_address | varchar(20) |  |
| cust\_mob\_number | int |  |
| cust\_email | varchar(20) |  |
| cust\_passport\_number | varchar(20) |  |
| cust\_martial\_status | char(8) |  |
| cust\_pps\_number | int |  |

**VEHICLE**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| vehicle\_id | varchar(20) | PK |
| cust\_id | varchar(20) | FK |
| policy\_id | varchar(20) |  |
| vehicle\_registration\_number | varchar(20) |  |
| vehicle\_value | int |  |
| vehicle\_type | varchar(20) |  |
| vehicle\_size | int |  |
| vehicle\_number\_of\_seats | int |  |
| vehicle\_manufacturer | varchar(20) |  |
| vehicle\_engine\_number | int |  |
| vehicle\_chassis\_number | int |  |
| vehicle\_number | varchar(20) |  |
| vehicle\_model\_number | varchar(20) |  |

**CLAIM**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| claim\_id | varchar(20) | PK |
| cust\_id | varchar(20) | FK |
| agreement\_id | varchar(20) |  |
| claim\_amount | int |  |
| incident\_id | varchar(20) |  |
| damage\_type | varchar(20) |  |
| date\_of\_claim | date |  |
| claim\_status | char(10) |  |

**CLAIM SETTLEMENT**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| claim\_settlement\_id | varchar(20) | PK |
| claim\_id | varchar(20) | FK |
| cust\_id | varchar(20) | FK |
| vehicle\_id | varchar(20) |  |
| date\_settled | date |  |
| amount\_paid | int |  |
| coverage\_id | varchar(20) |  |

**INCIDENT**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| incident\_id | varchar(20) | PK |
| incident\_type | varchar(20) |  |
| incident\_date | date |  |
| description | varchar(20) |  |

**PREMIUM PAYMENT**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| premium\_payment\_id | varchar(20) | PK |
| cust\_id | varchar(20) | FK |
| policy\_number | varchar(20) |  |
| premium\_payment\_schedule | date |  |
| premium\_paymnet\_amount | int |  |
| receipt\_id | varchar(20) |  |

**INSURANCE COMPANY**

|  |  |  |
| --- | --- | --- |
| **ATTRIBUTES** | **DATA TYPE** | **KEYS** |
| company\_name | varchar(20) | PK |
| company\_address | varchar(20) |  |
| company\_contact\_number | int |  |
| company\_fax | int |  |
| company\_email | varchar(20) |  |
| company\_website | varchar(20) |  |
| company\_location | varchar(20) |  |
| company\_department\_name | varchar(20) |  |
| company\_office\_name | varchar(20) |  |

**Staff**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Staff\_id | Varchar(20) | PK |
| Company\_Name | Varhcar(20) | FK |
| Staff\_FName | Varchar(10) |  |
| Staff\_LName | Varhcar(10) |  |
| Staff\_Address | Varchar(20) |  |
| Staff\_Contact | Integer |  |
| Staff\_Gender | Char(2) |  |
| Staff\_Marital\_Status | Char(8) |  |
| Staff\_Nationality | Char(15) |  |
| Staff\_Qualification | Varchar(20) |  |
| Staff\_Allowance | Integer |  |
| Staff\_PPS\_Number | Interger |  |

**Receipt**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Receipt\_id | Varchar(20) | PK |
| Premium\_Payment\_id | Varchar(20) | FK |
| Cust\_id | Varchar(20) | FK |
| Cost | Integer |  |
| Time | Date |  |

**Insurance\_Policy**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Agreement\_id | Varchar(20) | PK |
| Application\_id | Varchar(20) | FK |
| Cust\_id | Varchar(20) | FK |
| Department\_Name | Varchar(20) |  |
| Policy\_Number | Varchar(20) |  |
| Start\_Date | Date |  |
| Expiry\_Date | Date |  |
| Term\_Condition\_Description | Varchar(100) |  |

**Application**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Application\_id | Varchar(20) | PK |
| Cust\_id | Varchar(20) | FK |
| Vehicle\_id | Varchar(20) |  |
| Application\_Status | Char(8) |  |
| Coverage | Varchar(50) |  |

**Quote**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Quote\_id | Varchar(20) |  |
| Application\_id | Varchar(20) |  |
| Cust\_id | Varchar(20) |  |
| Issue\_id | Date |  |
| Valid\_From\_Date | Date |  |
| Valid\_Till\_Date | Date |  |
| Description | Varchar(100) |  |
| Product\_id | Varchar(20) |  |
| Coverage\_Level | Varchar(20) |  |

**Coverage**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Key** |
| Coverage\_id | Varchar(20) | PK |
| Company\_Name | Varchar(20) | FK |
| Coverage\_Amount | Integer |  |
| Coverage\_Type | Char(10) |  |
| Coverage\_Level | Char(15) |  |
| Product\_id | Varchar(20) |  |
| Coverage\_Discussion | Varchar(100) |  |
| Coverage\_Terms | Varchar(50) |  |

## Product table

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Key** |
| product\_number | varchar(20) | PK |
| company \_name | varchar(40) | FK(Insurace\_company table) |
| product\_price | int |  |
| product\_type | char(30) |  |

## Office table

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Key** |
| office\_name | varchar(20) | PK |
| department\_name | varchar(20) | FK(Department table) |
| company\_name | varchar(40) | FK(Insurance\_company table) |
| office\_leader | varchar(20) |  |
| contact\_information | varchar(20) |  |

|  |  |  |
| --- | --- | --- |
| address | Varchar(20) |  |
| admin\_cost | int |  |

**Department table**

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Key** |
| department\_name | varchar(20) | PK |
| department\_staff | varchar(20) |  |
| company\_name | varchar(20) | FK(Insurance\_company table) |
| department\_leader | Varchar(20) |  |
| office | Varchar(20) |  |
| contact\_information | varchar(20) |  |

**CONCLUSION:**

From the whole project work we learned about the rea- life problem which is an Insurance Company. We divided the project work among our team members we have created the tables and then Inserted some real-life data and then solved the queries. We learned how to work with a team, how to manage, how to divide the work ….. faced some challenges and solved it by group discussion. We learned the core concept of Data Base and Management System and Mysql codes.

**OUR TEAM:**

**ALLU HANUMA REDDY**

**BARLA SANJU JOHN**

**LALAM DIVYA SRI**

**MADDILA RAJESH NAIDU**

**NAGELLA PRANAV REDDY**

**BADIGINCHALA CHANDANA PRIYA**

**GOLLA ANJAIAH**

**MULAKALA KESAVA SRI SATYA**

**KURAPATI ABHINAV**

**ERIGI VAISHNAVI**