*A project report on*

**VEHICLE INSURANCE DATABASE SYSTEM**

*Submitted by*

**TEAM 10**

Adithya Hegde : 20BCS006

Esha : 20BCS045

Madineni Rohith : 20BCS081

Maitreyi : 20BCS083

Mohammed Abdul Haseeb : 20BCS085

Rohit Khetan : 20BCS114

Samuel Mathew : 20BCS116

Shaunak Maduskar : 20BCS119

Singh Sweekruti Narendra : 20BCS124

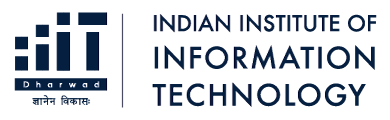
Sudeepto Chatterjee : 20BCS130

***Under the guidance of***

Prof. Uma Seshadri

Dr. Pramod Yelmewad

Dr. Supriya Nadiger



ACKNOWLEDGEMENT

This project has been done as part of the database management system course, under the guidance of Prof. Uma Seshadri, Dr. Pramod Yelmewad and Dr. Supriya Nadiger as part of the BTech degree at IIIT Dharwad. The team members are eternally grateful to the professors for their constant support and guidance through this project.

* Team 10: Impeccable Innovators

INDEX

1. Objective
2. Entity-Relationship Diagrams
   1. Conceptual Data Model
   2. Logical Data Model
   3. Physical Data Model
3. Results
4. Conclusion

OBJECTIVE

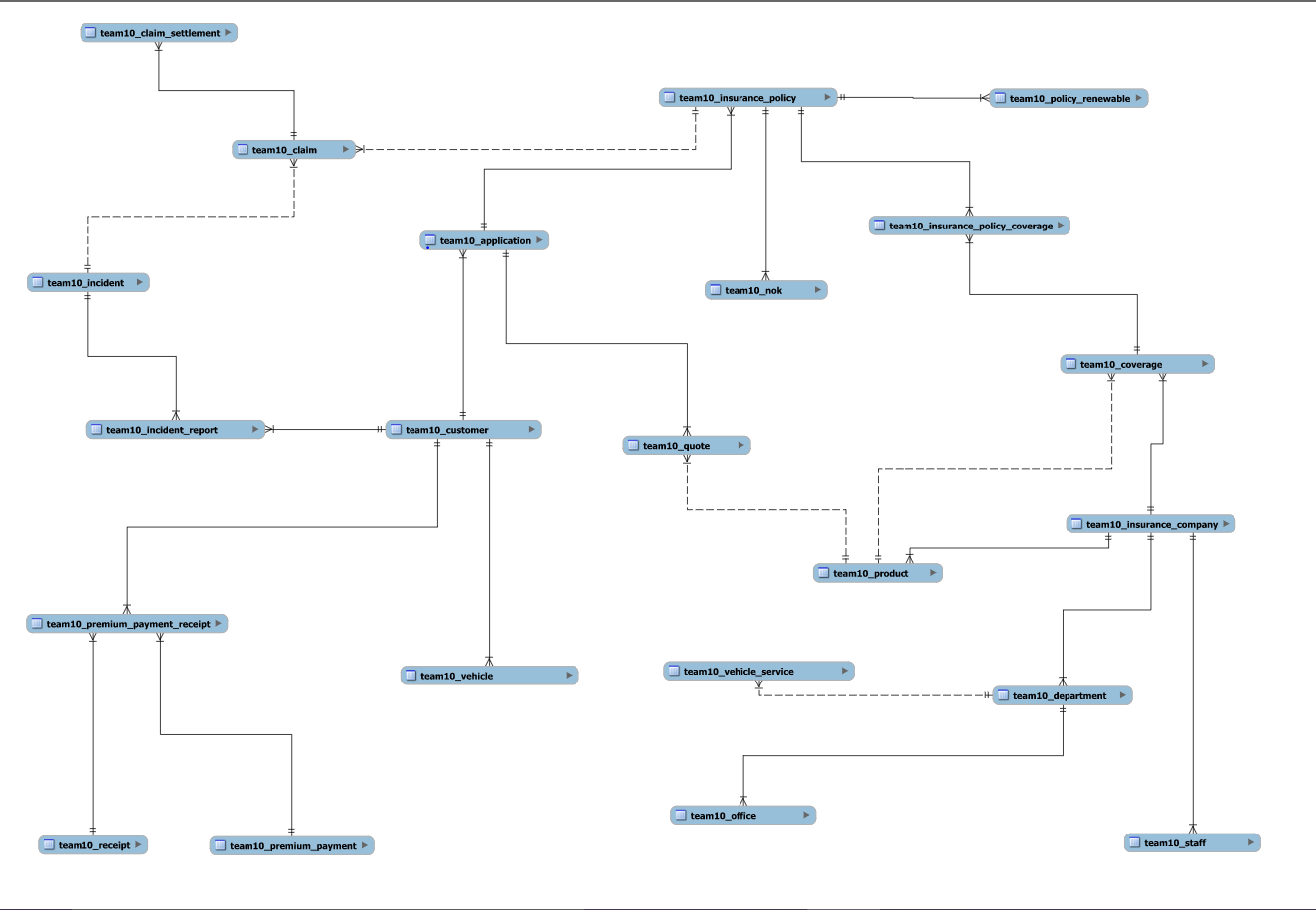
The database in question has been built for a company that provides insurance for vehicle related needs and incidents. Various concepts learnt throughout the course of Database Management Systems intend to be implied during the execution of this project.

The database consists of twenty-one tables. Each of these tables has a set of relationships with a set of other tables.

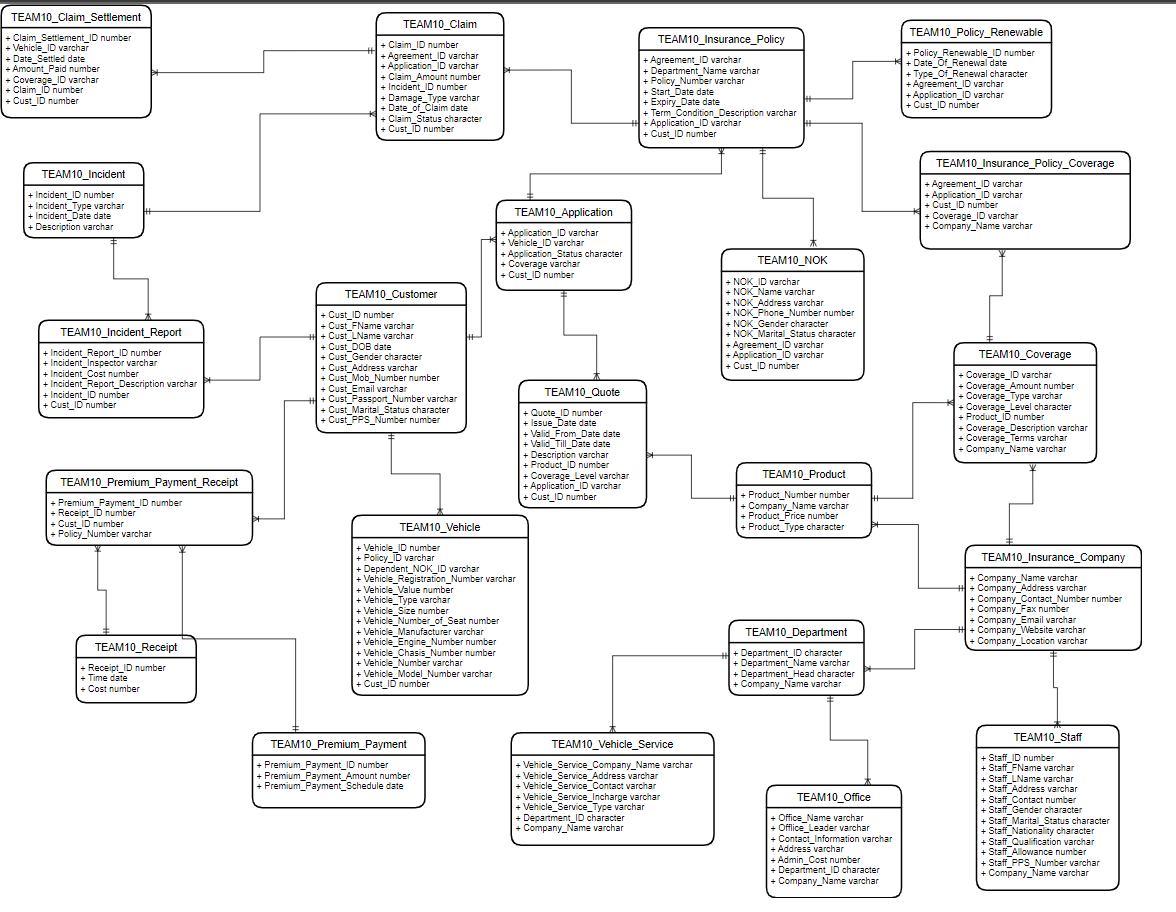
The aim of the project is to gain practical experience in data modelling and handling. It will serve to provide a better understanding of DBMS concepts such as normalisation, stored procedures, cascading and such. Implementation of different SQL queries will serve to provide more experience in the same field.

ENTITY - RELATIONSHIP DIAGRAMS

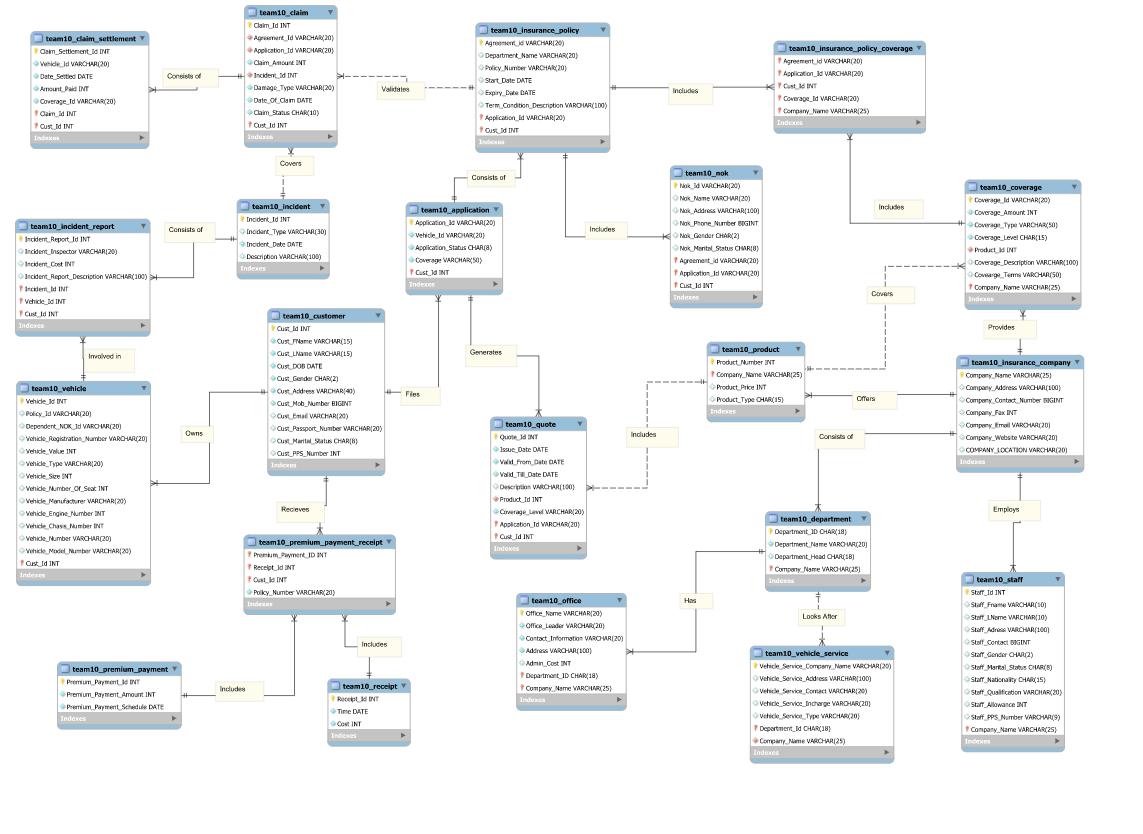
Conceptual Data Model



Logical Data Model



Physical Data Model



Changes from LDM to PDM:

* Policy Renewable table was excluded
* Added Vehicle\_Id to Incident Report table
* Changed Foreign Key for Incident Report table from Cust\_Id referencing the Customer table to (Vehicle\_Id, Cust\_Id) referencing the Vehicle table

RESULTS

Query 1  
Retrieve Customer and Vehicle details who have been involved in an incident and claim status is pending:

**CREATE** **VIEW** pendingClaims **AS**

**SELECT** Cust\_Id**,** Agreement\_ID **FROM** TEAM10\_CLAIM **WHERE** claim\_status **=** 'PENDING'**;**

**SELECT** **\*** **FROM** TEAM10\_VEHICLE **NATURAL** **JOIN** TEAM10\_CUSTOMER

**WHERE** Policy\_Id **IN** **(SELECT** Policy\_Number **FROM** TEAM10\_INSURANCE\_POLICY **WHERE** Agreement\_ID **IN** **(SELECT** Agreement\_ID **FROM** pendingClaims**));**

Query 2  
Retrieve customer details who have premium payment amount greater than the sum of all the CustomerIds in the database:

DELIMITER $$

**CREATE** **FUNCTION** sum\_customerIDs**()**

**RETURNS** INT

**DETERMINISTIC**

**BEGIN**

**DECLARE** **sum** INT**;**

**SET** **sum** **=** **(SELECT** **sum(**cust\_id**)** **FROM** TEAM10\_CUSTOMER**);**

**RETURN** **sum;**

**END;**

$$

**SELECT** **\*** **FROM** team10\_customer

**WHERE** team10\_customer**.**cust\_id **IN**

**(SELECT** r**.**cust\_id **FROM** team10\_premium\_payment\_receipt **AS** r

**INNER** **JOIN** team10\_premium\_payment **AS** p

**ON** p**.**Premium\_Payment\_Id **=** r**.**Premium\_Payment\_Id

**WHERE** p**.**Premium\_Payment\_Amount **>** sum\_customerIDs**());**

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

**CREATE** **VIEW** No\_of\_products **AS**

**SELECT** Company\_Name**,** **count(**Product\_Number**)** **AS** NP **FROM** team10\_product

**GROUP** **BY** Company\_Name**;**

DELIMITER $$

**CREATE** **FUNCTION** getNumOfOffices**(**DepartmentID char**(**18**))**

**RETURNS** INT

**DETERMINISTIC**

**BEGIN**

**DECLARE** numberOfOffices INT**;**

**SET** numberOfOffices **=** **(SELECT** **COUNT(\*)** **FROM** TEAM10\_OFFICE **WHERE** Department\_ID **=** DepartmentID**);**

**RETURN** numberOfOffices**;**

**END;**

$$

**CREATE** **VIEW** No\_of\_departments **AS**

**SELECT** Company\_Name**,** **count(**Department\_ID**)** **AS** ND **FROM** team10\_department

**WHERE** getNumOfOffices**(**Department\_ID**)** **>** 1

**GROUP** **BY** Company\_Name**;**

**SELECT** **\*** **FROM** team10\_insurance\_company **WHERE** Company\_Name **IN** **(**

**SELECT** No\_of\_products**.**Company\_Name **FROM** No\_of\_products **LEFT** **JOIN** No\_of\_departments **ON** No\_of\_products**.**Company\_Name **=** No\_of\_departments**.**Company\_Name

**WHERE** NP **>** **IFNULL(**ND**,** 0**));**

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:  
  
*Assumption: Premium Payment is made on an annual basis and an incident occurring over a year since the last payment will not be covered*

DELIMITER $$

**CREATE** **FUNCTION** getLatestPaymentDate**(**Vehicle\_Id\_Num INT**)**

**RETURNS** DATE

**DETERMINISTIC**

**BEGIN**

**DECLARE** lastDate DATE**;**

**DECLARE** associatedPolicyNum VARCHAR**(**20**);**

**SET** associatedPolicyNum **=** **(SELECT** Policy\_ID **FROM** TEAM10\_VEHICLE **WHERE** Vehicle\_Id **=** Vehicle\_Id\_Num**);**

**SET** lastDate **=** **(SELECT** Time **FROM** TEAM10\_PREMIUM\_PAYMENT\_RECEIPT **NATURAL** **JOIN** TEAM10\_RECEIPT **WHERE** Policy\_Number **=** associatedPolicyNum **ORDER** **BY** Time **DESC** **LIMIT** 1**);**

**RETURN** lastDate**;**

**END;**

$$

DELIMITER $$

**CREATE** **FUNCTION** findNumberOfCars**(**Customer INT**)**

**RETURNS** INT

**DETERMINISTIC**

**BEGIN**

**DECLARE** numberOfCars INT**;**

**SET** numberOfCars **=** **(SELECT** **COUNT(\*)** **FROM** TEAM10\_VEHICLE **WHERE** Cust\_Id **=** Customer **AND** Policy\_Id **IS** **NOT** **NULL** **GROUP** **BY** Cust\_Id**);**

**RETURN** numberOfCars**;**

**END;**

$$

**SELECT** **DISTINCT** Cust\_Id **FROM** TEAM10\_INCIDENT\_REPORT

**INNER** **JOIN** TEAM10\_INCIDENT **ON** TEAM10\_INCIDENT\_REPORT**.**Incident\_Id **=** TEAM10\_INCIDENT**.**Incident\_Id

**WHERE** findNumberOfCars**(**Cust\_Id**)** **>** 1 **AND** **DATEDIFF(**Incident\_Date**,** getLatestPaymentDate**(**Vehicle\_Id**))** **>** 365**;**

Query 5  
Select all vehicles which have premium more than its vehicle number:

**CREATE** **VIEW** VehicleListPremiumNumber **AS**

**(**

**SELECT** **DISTINCT** **(**Vehicle\_Id**)** **FROM** TEAM10\_VEHICLE **AS** T1

**INNER** **JOIN** TEAM10\_CUSTOMER **AS** T2 **ON** T1**.**Cust\_Id **=** T2**.**Cust\_Id

**INNER** **JOIN** TEAM10\_PREMIUM\_PAYMENT\_RECEIPT **AS** T3 **ON** T2**.**Cust\_Id **=** T3**.**Cust\_Id

**INNER** **JOIN** TEAM10\_PREMIUM\_PAYMENT **AS** T4 **ON** T3**.**Premium\_Payment\_ID **=** T4**.**Premium\_Payment\_Id

**WHERE** T1**.**Vehicle\_Number **<** T4**.**Premium\_Payment\_Amount **AND** Policy\_Number **=** Policy\_Id

**);**

**SELECT** **\*** **FROM** TEAM10\_VEHICLE **WHERE** Vehicle\_Id **IN** **(SELECT** **\*** **FROM** VehicleListPremiumNumber**);**

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** **DISTINCT** cus**.\***

**FROM** team10\_claim c

**INNER** **JOIN** team10\_customer cus **ON** c**.**Cust\_Id **=** cus**.**Cust\_Id

**INNER** **JOIN** team10\_claim\_settlement cs **ON** cus**.**Cust\_Id **=** cs**.**Cust\_Id

**INNER** **JOIN** team10\_coverage cov **ON** cs**.**Coverage\_Id **=** cov**.**Coverage\_Id

**WHERE** c**.**claim\_amount**>(**cs**.**claim\_settlement\_id**+**cs**.**vehicle\_id**+**cs**.**claim\_id**+**cs**.**cust\_id**)**

**AND** c**.**claim\_amount**<**cov**.**coverage\_amount**;**

CONCLUSION

The given vehicle insurance database was created, with all the given constraints, keeping in mind the entity types and relationships. The database was understood and developed, by making the CDM, LDM, PDM and then adding relevant data. Finally, the queries given to us were executed innovatively, and efficiently. The project helped us to learn how to use real world data, and gain practical experience in dealing with a database and to inculcate a habit of using sound design principles keeping realistic business scenarios in mind. We also learnt how to collaborate together as a team to build a project and learn from each other.

Therefore, the project was successfully completed.