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

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Dharwad

ज्ञानेन विकासः

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

OBJECTIVE

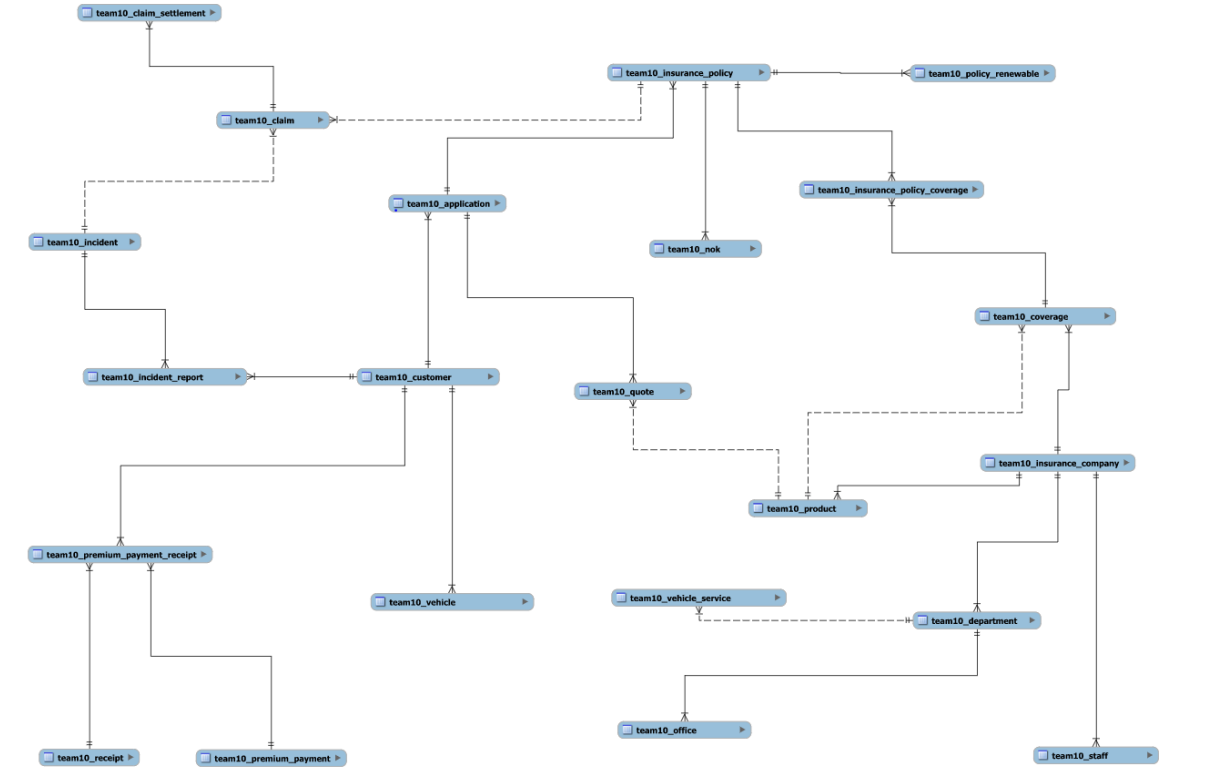
The database in question has been built for a company that provides insurance for vehicle related needs and incidents. Various concepts learnt throughout the course of Database Management Systems intend to be implied during the execution of this project.

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

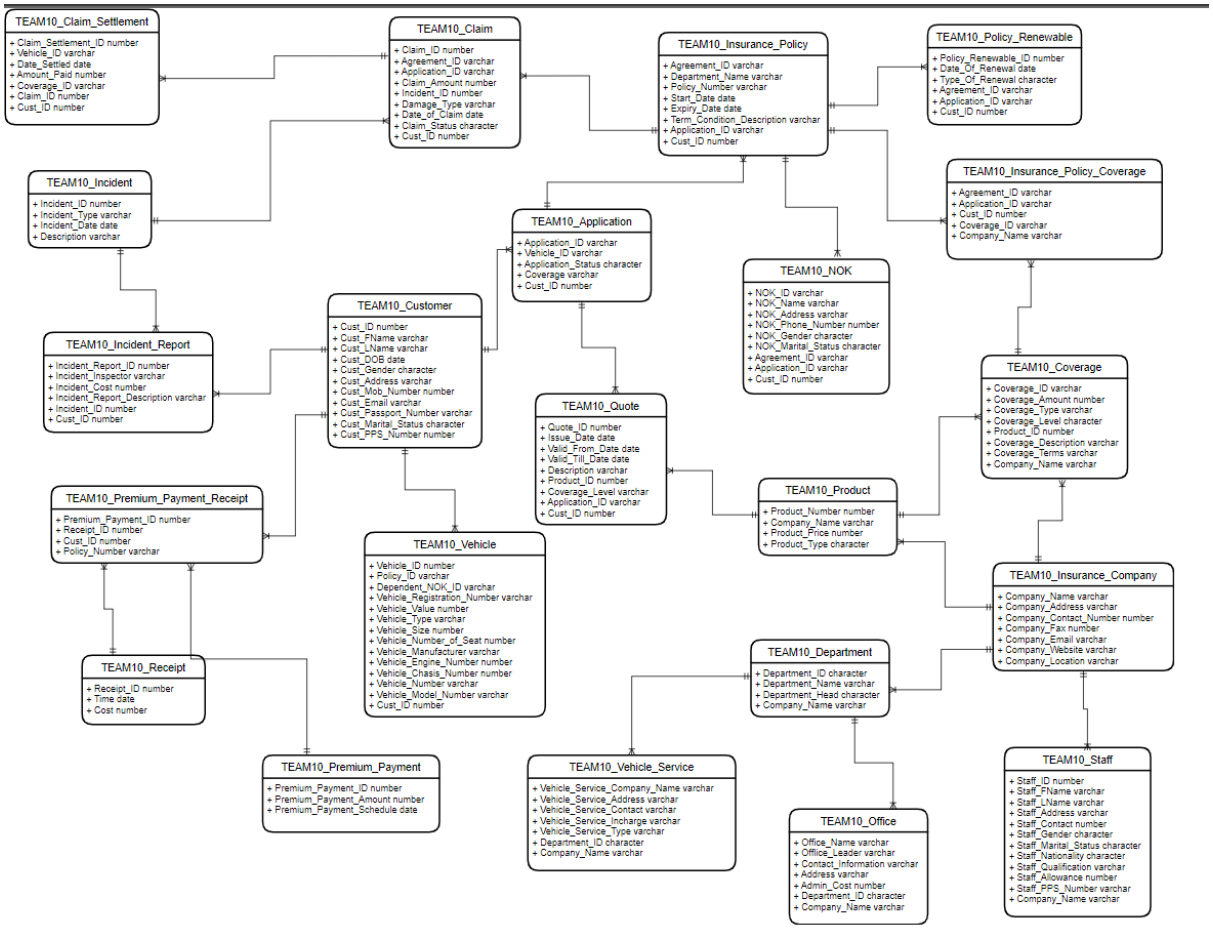
The aim of the project is to gain practical experience in data modelling and handling. It will serve to provide a better understanding of DBMS concepts such as normalisation, stored procedures, cascading and such. Implementation of different SQL queries will serve to provide more experience in the same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

ENTITY - RELATIONSHIP DIAGRAMS

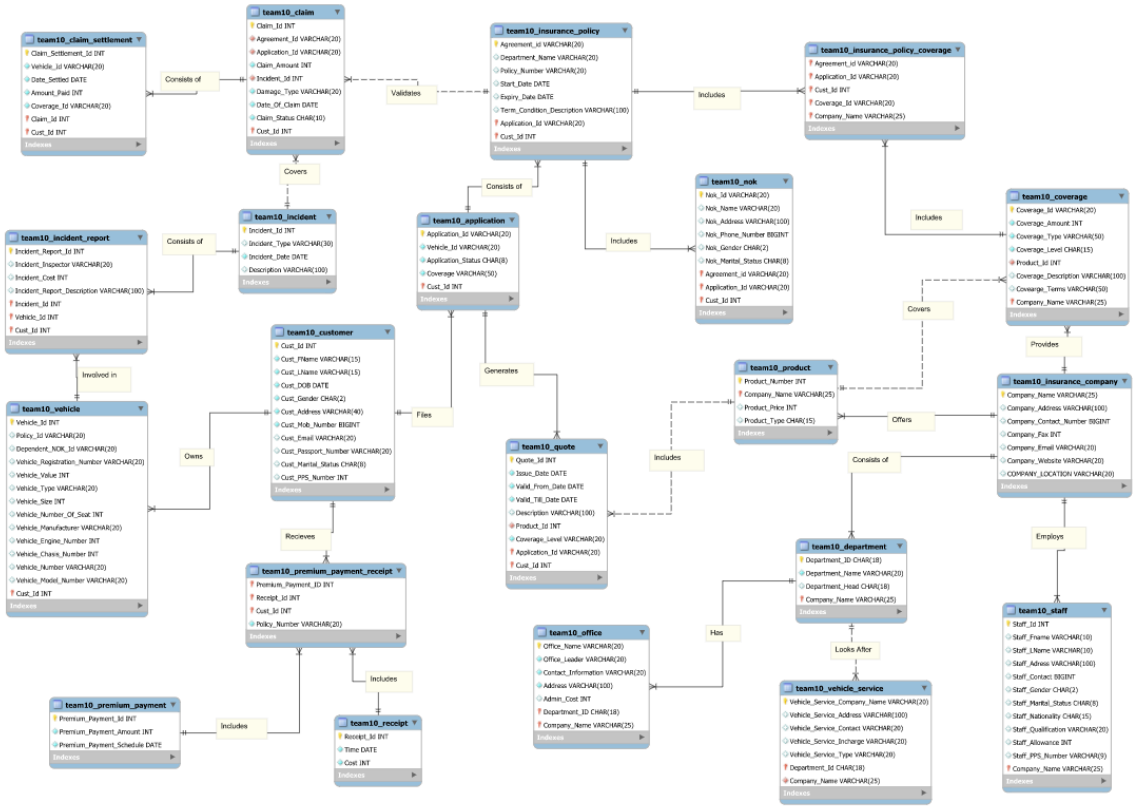
Conceptual Data Model



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.