

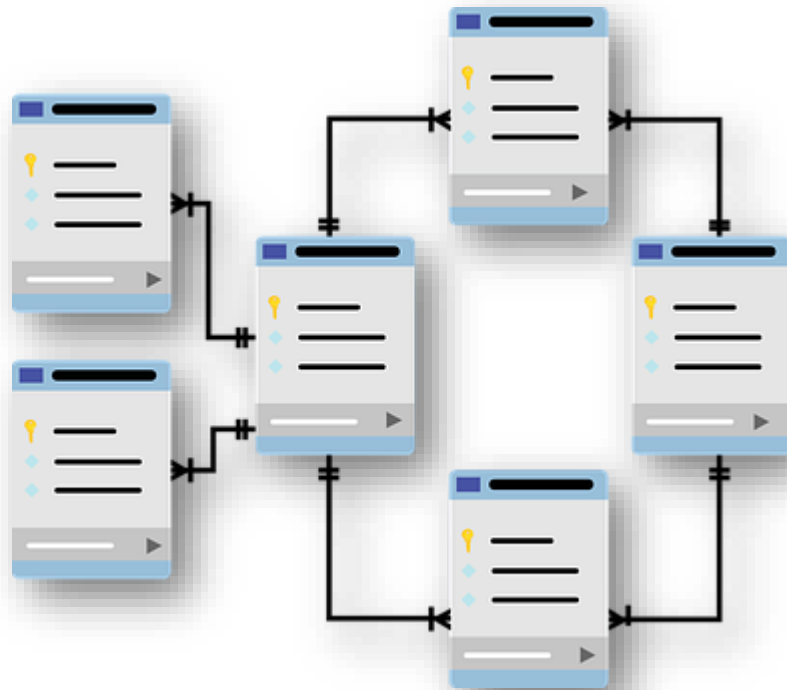


---

# DATABASE DESIGN

---

Vehicle Insurance Company



PRANJAL | SHUBHAM | DEEPANSHU | VINITA

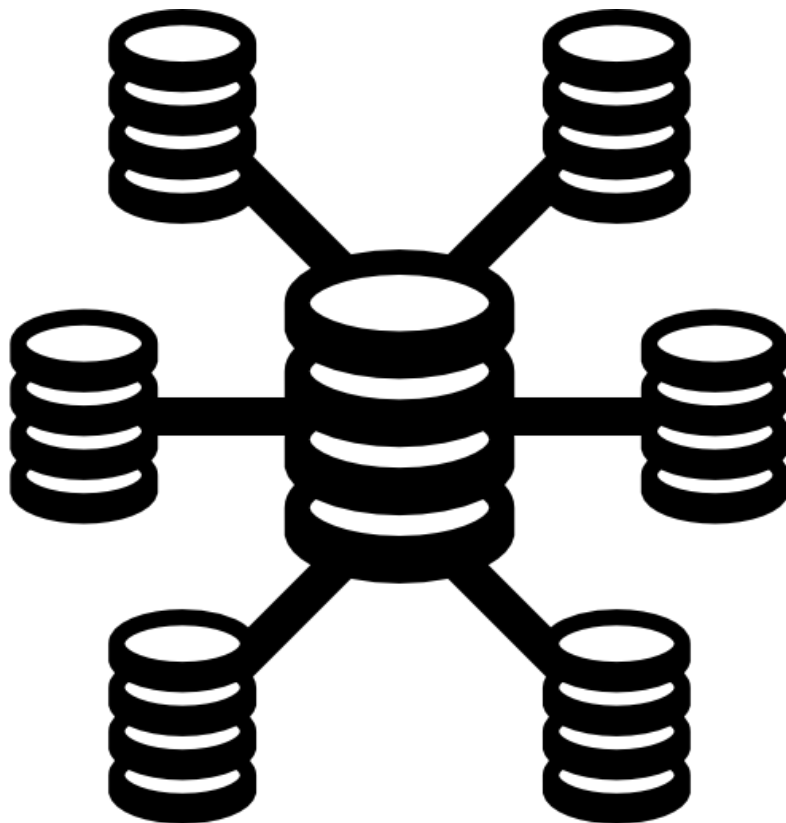
18BCS115 | 18BCS094 | 18BCS114 | 18BCS109

Indian Institute of Information Technology, Dharwad





# INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD



SUBMITTED BY : SHUBHAM SHARMA

VINITA YADAV

DEEPANSHU SACHDEVA

PRANJAL AGGARWAL

SUPERVISED BY : DR UMA S

MS. SUPRIYA

## **Title of the Project**

Design and implement a database for a Vehicle Insurance Company

## **Intent of the Document**

This documentation includes a proper discussion of how the given project was designed and implemented which included modelling of the entities, designing the database system, insertion of values in each table, normalisation, coding of SQL queries, etc.

## **Objective**

To gain practical knowledge on the working of a DBMS, encountering and overcoming the problems and challenges faced during its implementation, become comfortable in using the database modelling and designing tools and being able to apply all these principles for creating solutions for realistic business scenarios. Lastly, an important objective of this project is to be able to collaborate and work as team and bring out value for everybody.

## **Abstract**

The “Vehicle Insurance Management System” has been developed to override the problems prevailing in the practicing manual system. This database system is supported to eliminate and, in some cases, reduce the hardships faced by existing manual systems.

Main objective of this project is to design a simple software application for insurance companies for managing customers who buy new vehicles and take insurance for those vehicles. Details of payment, time period, vehicle details, customer personal details, insurance specifications are updated to the database.

We can add, delete, modify, existing records and search for old records within short time.

## **Project Implementation**

The project was implemented inline with the following steps:

- Database Development
  - Conceptual Data Model (CDM)
  - Logical Data Model (LDM)
  - Physical Data Model (PDM)
- Executing the given queries
- Preparation of report based on the work done
- Exploring and understanding new ways and different topics in DBMS
- Working on the suggestions given by the supervisors and improving the database

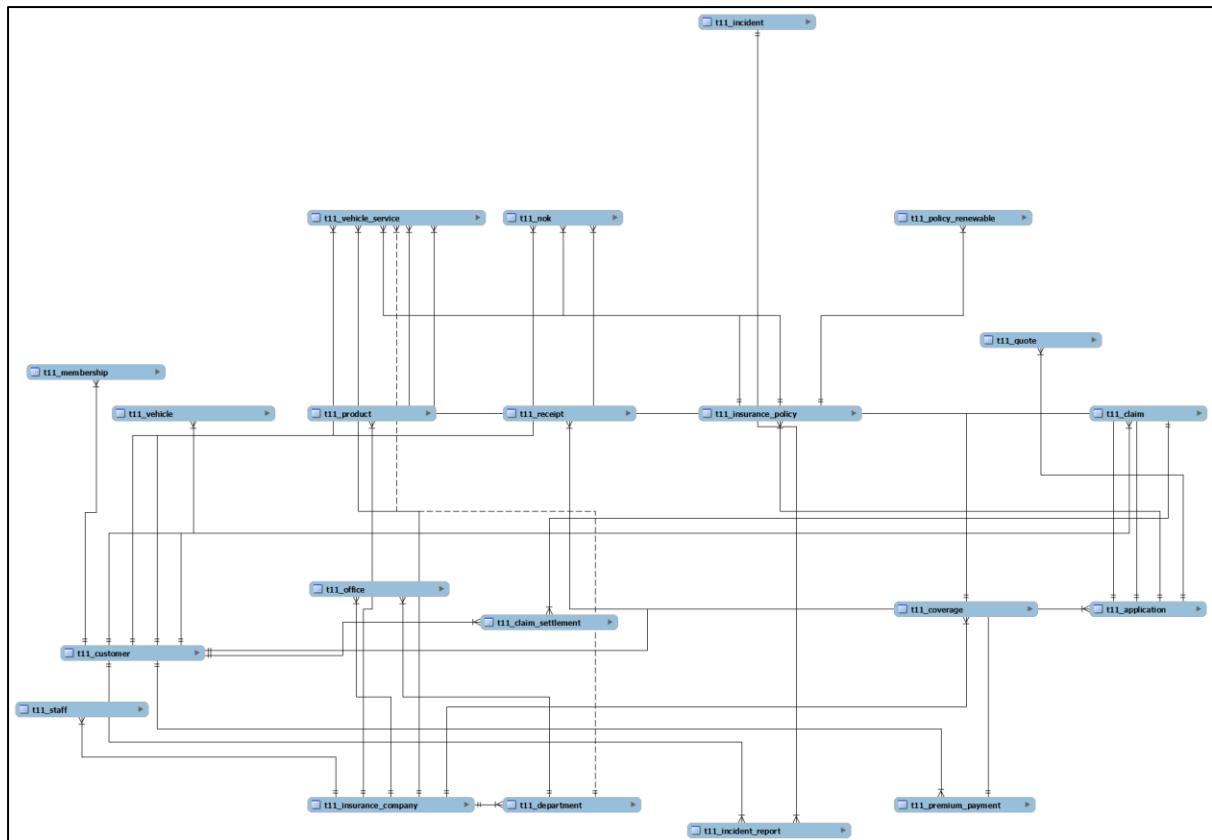
## **Technology Stack**

The following technologies were used while implementing the project:

- MySQL Workbench (for designing the database)
- GitHub (for collaborating on the SQL queries)
- Google Meet (for regular team discussions and suggestions)

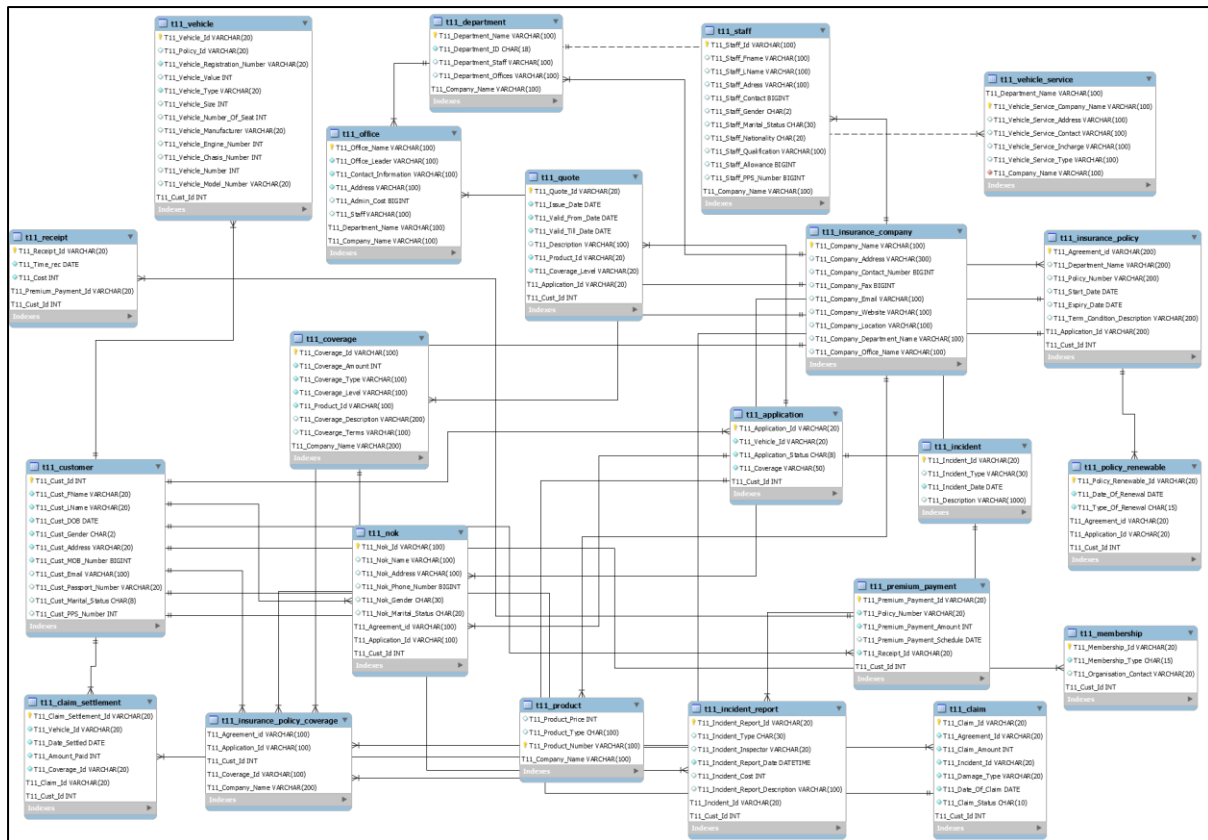
## **Conceptual Data Model (CDM)**

The Conceptual Data Model is a structured business view of the data required to support business processes, record business events, and track related performance measures. This model focuses on identifying the data used in the business but not its processing flow or physical characteristics. This model's perspective is independent of any underlying business applications. The conceptual data model represents the overall structure of data required to support the business requirements independent of any software or data storage structure.



## Logical Data Model (LDM)

The Logical Data Model is used to define the structure of data elements and to set relationships between them. The logical data model adds further information to the conceptual data model elements. The advantage of using a Logical data model is to provide a foundation to form the base for the Physical model. However, the modelling structure remains generic. At this Data Modelling level, no primary or secondary key is defined. At this Data modelling level, you need to verify and adjust the connector details that were set earlier for relationships.



## Physical Data Model (PDM)

A Physical Data Model describes a database-specific implementation of the data model. It offers database abstraction and helps generate the schema. This is because of the richness of meta-data offered by a Physical Data Model. The physical data model also helps in visualizing database structure by replicating database column keys, constraints, indexes, triggers, and other RDBMS features.

## Queries

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

```
SELECT cust.* FROM T11_Customer AS Cust,
T11_Vehicle AS v, T11_Incident_Report AS ir, T11_Claim AS c
WHERE ir.T11_Cust_Id = cust.T11_Cust_Id
AND c.T11_Cust_Id = cust.T11_Id AND c.T11_Claim_Status = 'Pending';
```

MySQL Workbench

Test x

File Edit View Query Database Server Tools Scripting Help

Final\_Insertions queries Views NORMALIZATION INSERTION

Limit to 1000 rows

```

13 -- QUERY - 1
14 SELECT cust.* FROM T11_Customer AS cust,
15 T11_Vehicle AS v,
16 T11_Incident_Report AS ir,
17 T11_Claim AS c
18 WHERE ir.T11_Cust_Id = cust.T11_Cust_Id
19 AND c.T11_Cust_Id = cust.T11_Cust_Id AND c.T11_Claim_Status = 'Pending';

```

Result Grid

T11_Cust_Id	T11_Cust_FName	T11_Cust_LName	T11_Cust_DOB	T11_Cust_Gender	T11_Cust_Address	T11_Cust_MOB_Number	T11_Cust_Email	T11_Cust_Passport_Number	T11_Cust_Marital_Status	T11_Cust_Premium_Payment_Amount
217	Sana	Iqbal	1987-01-16	F	Hyderabad	7908887689	sana@biker.com	7896736	Single	30
219	Rajesh	Pilot	1945-02-10	M	Ghaziabad	789875783	rajesh@iaf.com	672867	Married	55
216	Grace	Kelly	1929-11-12	F	Pennsylvania	79217654897	kelly@actress.com	873078	Married	52
219	Rajesh	Pilot	1945-02-10	M	Ghaziabad	789875783	rajesh@iaf.com	672867	Married	55
217	Sana	Iqbal	1987-01-16	F	Hyderabad	7908887689	sana@biker.com	7896736	Single	30

Result 6 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
3	19.43.52	SELECT cust.*, v.* FROM T11_Customer AS cust, T11_Vehicle AS v, T11_Incident_Report AS ir, T11_Claim AS c WHERE ir.T11_Cust_Id = cust.T11_Cust_Id AND c.T11_Cust_Id = cust.T11_Cust_Id AND c.T11_Claim_Status = 'Pending';	60 row(s) returned	0.093 sec / 0.000 sec
4	19.44.13	SELECT v.* FROM T11_VEHICLE AS v, T11_Customer AS c, T11_PREMIUM_PAYMENT AS pp WHERE c.T11_Cust_Id = v.T11_Cust_Id AND pp.T11_Cust_Id = c.T11_Cust_Id AND pp.T11_Premium_Payment_Amount > (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust);	13 row(s) returned	0.078 sec / 0.000 sec
5	19.44.14	SELECT v.* FROM T11_VEHICLE AS v, T11_Customer AS c, T11_PREMIUM_PAYMENT AS pp WHERE c.T11_Cust_Id = v.T11_Cust_Id AND pp.T11_Cust_Id = c.T11_Cust_Id AND pp.T11_Premium_Payment_Amount > (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust);	13 row(s) returned	0.000 sec / 0.000 sec
6	19.44.16	SELECT v.* FROM T11_VEHICLE AS v, T11_Customer AS c, T11_PREMIUM_PAYMENT AS pp WHERE c.T11_Cust_Id = v.T11_Cust_Id AND pp.T11_Cust_Id = c.T11_Cust_Id AND pp.T11_Premium_Payment_Amount > (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust);	13 row(s) returned	0.000 sec / 0.000 sec
7	19.44.17	SELECT v.* FROM T11_VEHICLE AS v, T11_Customer AS c, T11_PREMIUM_PAYMENT AS pp WHERE c.T11_Cust_Id = v.T11_Cust_Id AND pp.T11_Cust_Id = c.T11_Cust_Id AND pp.T11_Premium_Payment_Amount > (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust);	13 row(s) returned	0.000 sec / 0.000 sec
8	19.44.52	SELECT cust.* FROM T11_Customer AS cust, T11_Vehicle AS v, T11_Incident_Report AS ir, T11_Claim AS c WHERE ir.T11_Cust_Id = cust.T11_Cust_Id AND c.T11_Cust_Id = cust.T11_Cust_Id AND c.T11_Claim_Status = 'Pending';	60 row(s) returned	0.000 sec / 0.000 sec

Query - 2: Retrieve customer details who has premium payment amount greater than the sum of all the customerIds in the database.

```

SELECT cust.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp
WHERE pp.T11_Cust_Id = cust.T11_Cust_Id
HAVING (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) <
T11_Premium_Payment_Amount;

```

MySQL Workbench

Test x

File Edit View Query Database Server Tools Scripting Help

Final\_Insertions queries Views NORMALIZATION INSERTION

Limit to 1000 rows

```

38
39
40 -- QUERY 2
41
42 SELECT cust.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp
43 WHERE pp.T11_Cust_Id = cust.T11_Cust_Id
44 HAVING (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < T11_Premium_Payment_Amount;
45
46

```

Result Grid

T11_Cust_Id	T11_Cust_FName	T11_Cust_LName	T11_Cust_DOB	T11_Cust_Gender	T11_Cust_Address	T11_Cust_MOB_Number	T11_Cust_Email	T11_Cust_Passport_Number	T11_Cust_Marital_Status	T11_Cust_Premium_Payment_Amount
201	Paul	Walker	1973-09-12	M	California	8937629017	paulwalker@actor.com	33694855	Married	15
202	Jaspal	Bhatti	1955-03-03	M	Amritsar	6710947824	jaspalbhatti@comedian.com	7922033	Married	15
205	James	Dean	1931-02-08	M	Indiana	7826501975	jamesdean@actor.com	5468084	Single	15
206	John Fitzgerald	Kennedy	1917-05-29	M	Brookline	9825908717	jfk@uspresident.com	95084	Married	15
207	Sam	Kinson	1953-12-08	M	Washington	9027561813	sam@comedian.com	7721945	Divorced	15

Result 14 x

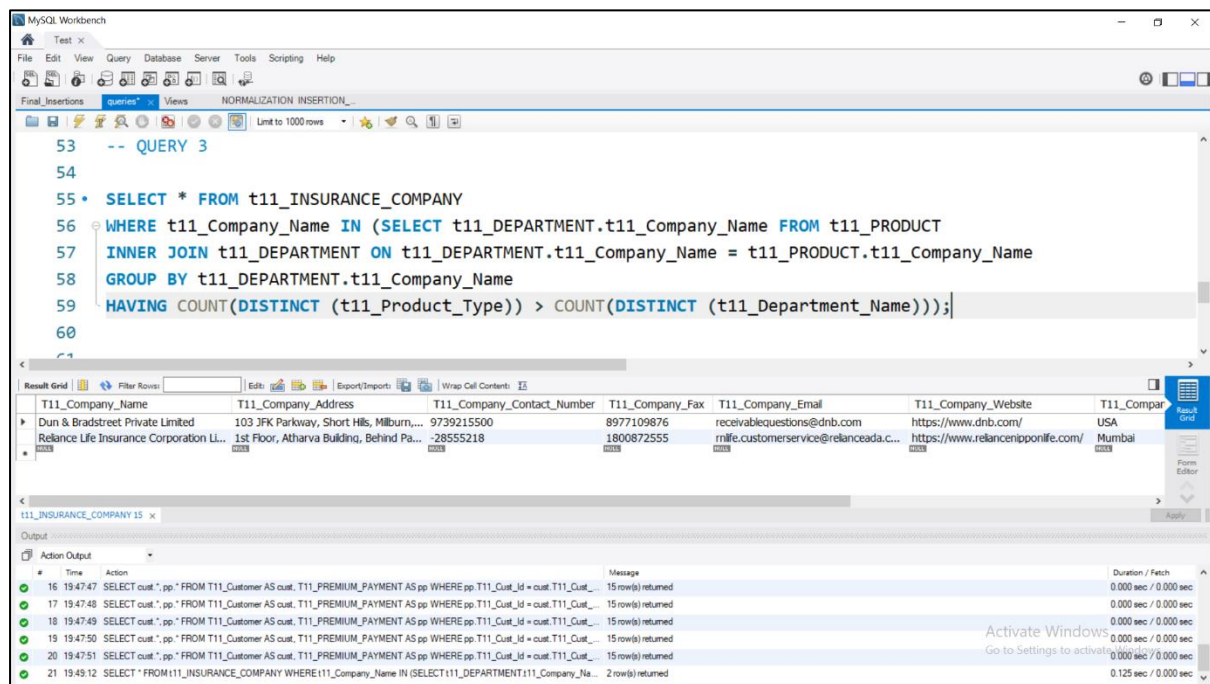
Output

Action Output

#	Time	Action	Message	Duration / Fetch
15	19.47.46	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.016 sec / 0.000 sec
16	19.47.47	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.000 sec / 0.000 sec
17	19.47.48	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.000 sec / 0.000 sec
18	19.47.49	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.000 sec / 0.000 sec
19	19.47.50	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.000 sec / 0.000 sec
20	19.47.51	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_Id AND (SELECT SUM(cust.T11_Cust_Id) FROM T11_Customer AS cust) < pp.T11_Premium_Payment_Amount;	15 row(s) returned	0.000 sec / 0.000 sec

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

```
SELECT * FROM t11_INSURANCE_COMPANY
WHERE t11_Company_Name IN (SELECT t11_DEPARTMENT.t11_Company_Name
FROM t11_PRODUCT
INNER JOIN t11_DEPARTMENT ON t11_DEPARTMENT.t11_Company_Name =
t11_PRODUCT.t11_Company_Name
GROUP BY t11_DEPARTMENT.t11_Company_Name
HAVING COUNT(DISTINCT (t11_Product_Type)) > COUNT(DISTINCT
(t11_Department_Name)));
```

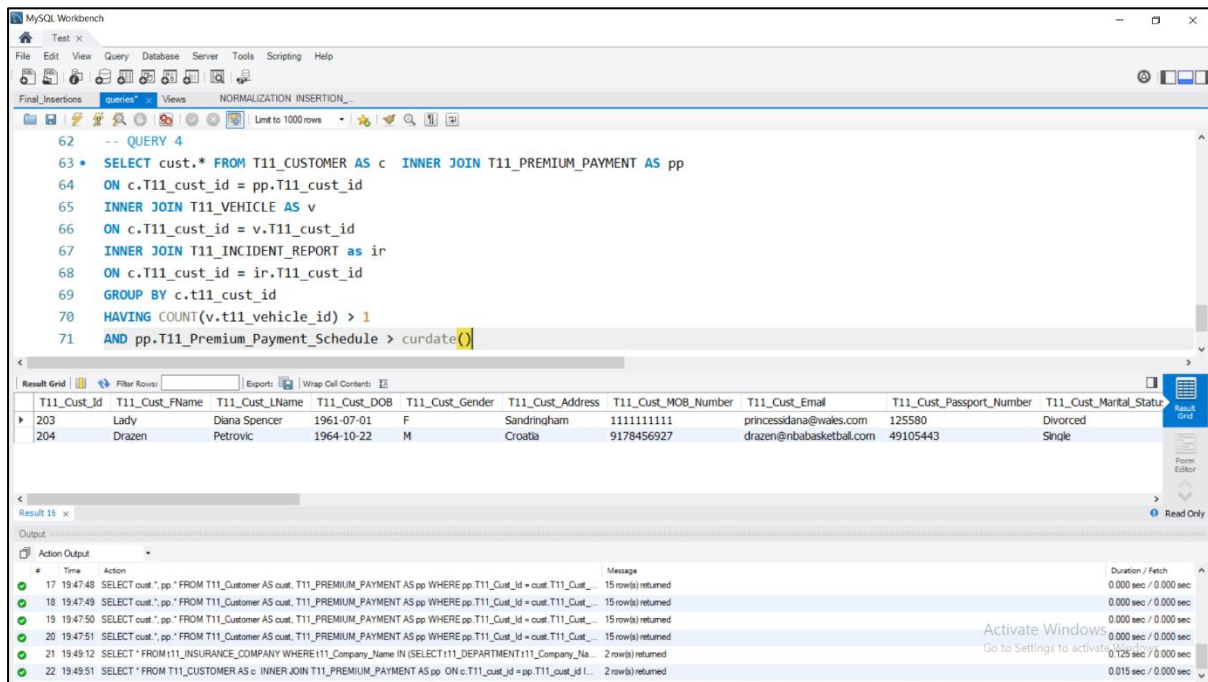


Query - 4: Select Customers who have more than one Vehicle, where the premium for one of the Vehicles is not paid and it is involved in accident.

```
SELECT cust.* FROM T11_CUSTOMER AS c INNER JOIN T11_PREMIUM_PAYMENT
AS pp ON c.T11_cust_id = pp.T11_cust_id
INNER JOIN T11_VEHICLE AS v
ON c.T11_cust_id = v.T11_cust_id
INNER JOIN T11_INCIDENT_REPORT as ir
ON c.T11_cust_id = ir.T11_cust_id
```

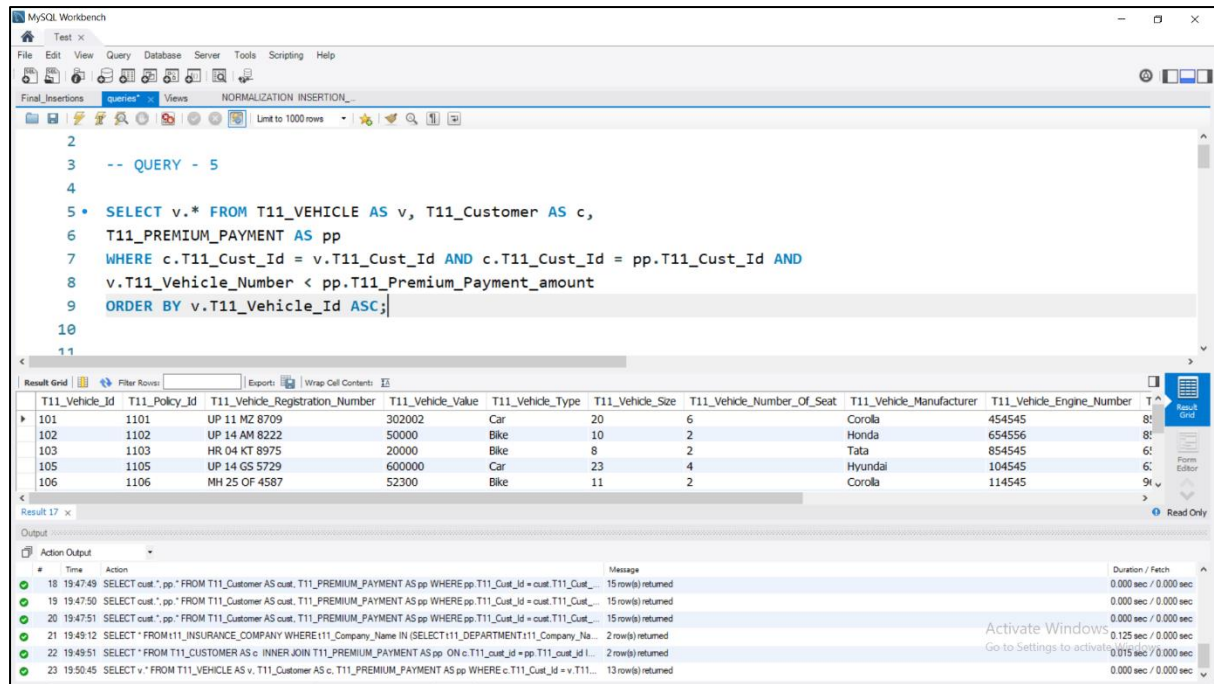


GROUP BY c.t11\_cust\_id  
 HAVING COUNT(v.t11\_vehicle\_id) > 1  
 AND pp.T11\_Premium\_Payment\_Schedule > curdate()



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

SELECT v.\* FROM T11\_VEHICLE AS v, T11\_Customer AS c,  
 T11\_PREMIUM\_PAYMENT AS pp  
 WHERE c.T11\_Cust\_Id = v.T11\_Cust\_Id AND c.T11\_Cust\_Id = pp.T11\_Cust\_Id AND  
 v.T11\_Vehicle\_Number < pp.T11\_Premium\_Payment\_amount  
 ORDER BY v.T11\_Vehicle\_Id ASC;



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 cust.*, c.T11_Claim_Amount, cov.T11_Coverage_Amount, cs.T11_Claim_Id
cs.T11_Coverage_Id,cs.T11_Claim_Settlement_Id,cs.T11_Vehicle_Id
FROM T11_Customer AS cust,
T11_Claim_Settlement AS cs,
T11_Claim AS c, T11_Coverage AS cov
WHERE cust.T11_Cust_Id = cs.T11_Cust_Id
AND cs.T11_Claim_Id = c.T11_Claim_Id
AND cs.T11_Coverage_Id = cov.T11_Coverage_Id AND
c.T11_Claim_Amount < cov.T11_Coverage_Amount AND
c.T11_Claim_Amount > (SELECT SUM(cs.T11_Claim_Id + cs.T11_Coverage_Id +
cs.T11_Claim_Settlement_Id + cs.T11_Vehicle_Id) FROM T11_Claim_Settlement AS
cs);

```

MySQL Workbench

Test x

File Edit View Query Database Server Tools Scripting Help

Final\_Insertions queries Views NORMALIZATION INSERTION

Limit to 1000 rows

```

24 * SELECT cust.*, c.t11_claim_amount, cov.t11_coverage_amount, cs.T11_Claim_Id, cs.T11_Coverage_Id, cs.T11_Claim_Settlement_Id, cs.T11_Vehicle_Id
25 FROM T11_Customer AS cust,
26 T11_Claim_Settlement AS cs,
27 T11_Claim AS c,
28 T11_Coverage AS cov
29 WHERE cust.T11_Cust_Id = cs.T11_Cust_Id
30 AND cs.T11_Claim_Id = c.T11_Claim_Id
31 AND cs.T11_Coverage_Id = cov.T11_Coverage_Id
32 AND c.T11_Claim_Amount < cov.T11_Coverage_Amount
33 AND c.T11_Claim_Amount > (SELECT SUM(cs.T11_Claim_Id + cs.T11_Coverage_Id + cs.T11_Claim_Settlement_Id + cs.T11_Vehicle_Id)
34 FROM T11_Claim_Settlement AS cs);

```

Result Grid

T11_Cust_Id	T11_Cust_FName	T11_Cust_LName	T11_Cust_DOB	T11_Cust_Gender	T11_Cust_Address	T11_Cust_MOB_Number	T11_Cust_Email	T11_Cust_Passport_Number	T11_Cust_Marital_Status
204	Drazen	Petrovic	1964-10-22	M	Croatia	9178456927	drazen@nbabasketball.com	49105443	Single
214	Pierre	Curie	1859-05-15	M	Paris	9830553876	curie@scientist.com	765432	Married
219	Rajesh	Pilot	1945-02-10	M	Ghaziabad	789875783	rajesh@iaf.com	672867	Married
202	Jaspal	Bhatti	1955-03-03	M	Amritsar	6710947824	jaspalbhatti@comedian.com	7922033	Married
214	Pierre	Curie	1859-05-15	M	Paris	9830553876	curie@scientist.com	765432	Married

Result 18 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
19	19:47:50	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_...	15 row(s) returned	0.000 sec / 0.000 sec
20	19:47:51	SELECT cust.*, pp.* FROM T11_Customer AS cust, T11_PREMIUM_PAYMENT AS pp WHERE pp.T11_Cust_Id = cust.T11_Cust_...	15 row(s) returned	0.000 sec / 0.000 sec
21	19:49:12	SELECT * FROM T11_INSURANCE_COMPANY WHERE T11_Company_Name IN (SELECT T11_DEPARTMENT T11_Company_Na...	2 row(s) returned	0.125 sec / 0.000 sec
22	19:49:51	SELECT * FROM T11_CUSTOMER AS c INNER JOIN T11_PREMIUM_PAYMENT AS pp ON c.T11_cust_id = pp.T11_cust_id l...	2 row(s) returned	0.015 sec / 0.000 sec
23	19:50:45	SELECT v.* FROM T11_VEHICLE AS v, T11_Customer AS c, T11_PREMIUM_PAYMENT AS pp WHERE c.T11_Cust_Id = v.T11...	13 row(s) returned	0.000 sec / 0.000 sec
24	19:51:50	SELECT cust.*, c.t11_claim_amount, cov.t11_coverage_amount, cs.T11_Claim_Id, cs.T11_Coverage_Id, cs.T11_Claim_Settlement...	5 row(s) returned	0.062 sec / 0.000 sec

Activate Windows  
Go to Settings to activate Windows.

## Views

### View-1 for Company Insurance Employee

```

CREATE VIEW T11_Company_Insurance_Employee_View AS
SELECT c.T11_Cust_FNAME, c.T11_Cust_LNAME, ip.* FROM T11_Customer AS c
INNER JOIN T11_Insurance_Policy AS ip
ON ip.T11_Cust_Id = ip.T11_Cust_Id;

```

MySQL Workbench

Test x

File Edit View Query Database Server Tools Scripting Help

Final\_Insertions queries Views NORMALIZATION INSERTION\_

Limit to 1000 rows

```

3 -- View for Company Insurance Employee
4 CREATE VIEW T11_Company_Insurance_Employee_View AS
5 SELECT c.T11_Cust_FNAME, c.T11_Cust_LNAME, ip.* FROM T11_Customer AS c
6 INNER JOIN T11_Insurance_Policy AS ip
7 ON ip.T11_Cust_Id = ip.T11_Cust_Id;
8
9 SELECT * FROM T11_Company_Insurance_Employee_View;

```

Result Grid

T11_Cust_FNAME	T11_Cust_LNAME	T11_Agreement_Id	T11_Department_Name	T11_Policy_Number	T11_Start_Date	T11_Expiry_Date	T11_Term_Condition_Description	T11_Application_Id
Paul	Walker	601	Dun & Bradstreet Vehicle Insurance DL...	78901	2016-11-22	2026-07-30	Covers the damage or loss to the ins...	701
Jaspal	Bhatti	601	Dun & Bradstreet Vehicle Insurance DL...	78901	2016-11-22	2026-07-30	Covers the damage or loss to the ins...	701
Lady	Diana Spencer	601	Dun & Bradstreet Vehicle Insurance DL...	78901	2016-11-22	2026-07-30	Covers the damage or loss to the ins...	701
Drazen	Petrovic	601	Dun & Bradstreet Vehicle Insurance DL...	78901	2016-11-22	2026-07-30	Covers the damage or loss to the ins...	701
James	Dean	601	Dun & Bradstreet Vehicle Insurance DL...	78901	2016-11-22	2026-07-30	Covers the damage or loss to the ins...	701

T11\_Company\_Insurance\_Emp... x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
27	19:55:52	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
28	19:55:50	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
29	19:55:51	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
30	19:55:52	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
31	19:55:53	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
32	19:55:53	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.016 sec

Activate Windows  
Go to Settings to activate Windows.

## View-2 for Head of Department for the Insurance Company

```

CREATE VIEW T11_HOD_View AS
SELECT s.T11_Staff_Id, s.T11_Staff_FName, s.T11_Staff_LName,
s.T11_Staff_Adress, s.T11_Staff_Allowance, s.T11_Company_Name
FROM T11_Staff AS s;

```

MySQL Workbench

Test x

File Edit View Query Database Server Tools Scripting Help

Final\_Insertions queries Views NORMALIZATION INSERTION\_

Limit to 1000 rows

```

11 -- View For Head of Department
12 CREATE VIEW T11_HOD_View AS
13 SELECT s.T11_Staff_Id, s.T11_Staff_FName, s.T11_Staff_LName,
14 s.T11_Staff_Adress, s.T11_Staff_Allowance, s.T11_Company_Name
15 FROM T11_Staff AS s;
16
17 SELECT * FROM T11_HOD_View;

```

Result Grid

T11_Staff_Id	T11_Staff_FName	T11_Staff_LName	T11_Staff_Adress	T11_Staff_Allowance	T11_Company_Name
501	Deepanshu	Sachdeva	Ghaziabad	125000	Dun & Bradstreet Private Limited
502	Harley	McQuene	Hyderabad	170000	Reliance Life Insurance Corporation LL...
503	Pranjal	Aqarwal	Varanasi	200000	Bupa Insurance Group
504	Shubham	Sharma	Gwalior	300000	Aviva Insurance Company Limited
505	Nikita	Reddy	Lucknow	50000	Birkshire Hathaway Inc.
506	Vinita	Yadav	Hyderabad	90000	The Hartford Financial Services Group...

T11\_HOD\_View 7 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
28	19:55:50	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
29	19:55:51	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
30	19:55:52	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
31	19:55:53	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.000 sec
32	19:55:53	SELECT * FROM T11_Company_Insurance_Employee_View LIMIT 0, 1000	400 row(s) returned	0.000 sec / 0.016 sec
33	19:56:11	SELECT * FROM T11_HOD_View LIMIT 0, 1000	20 row(s) returned	0.031 sec / 0.000 sec

Activate Windows  
Go to Settings to activate Windows.

## **Conclusion**

The project was successfully executed and completed timely which was possible because of the great team spirit shown by each member and their contributions. The project was completed in the following major steps:

- Table Creations
- Data Insertions
- Testing Data for Anomalies
- Query Creation and Execution

Work done in all the above steps were divided equally among all the four members and all the tasks which required collaboration were done on video conferencing with inputs from each and every member.