

A database for a Vehicle Insurance Company

Indian Institute of Information Technology, Dharwad

Team 3

Derik Lytten (17BCS008) - CDM/LDM Assist

Rayhaan (18BCS055) - CDM/LDM Assist

Sahana N H (18BCS086) - CDM/LDM Assist

Amitabh Paliwal (18BCS004) - PDM Developer

Souvik Das (18BCS099) - PDM Developer

Physical Data Model(PDM)

PDM for vehicle insurance service

- Creating table and testing sql queries using MySQL Workbench

```
CREATE DATABASE IF NOT EXISTS mydb;
```

```
USE mydb;
```

```
CREATE TABLE IF NOT EXISTS t3_INCIDENT
```

```
(
```

```
t3_Incident_Id VARCHAR(20) NOT NULL ,
```

```
t3_Incident_Type VARCHAR(50) NULL ,
```

```
t3_Incident_Date DATE NOT NULL ,
```

```
t3_Description VARCHAR(1000) NULL ,
```

```
CONSTRAINT XPKINCIDENT_17 PRIMARY KEY (t3_Incident_Id)
```

```
);
```

```
CREATE UNIQUE INDEX XPKINCIDENT_17 ON t3_INCIDENT (t3_Incident_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_CUSTOMER
```

```
(
```

```
t3_Cust_Id VARCHAR(15) NOT NULL ,
```

```
t3_Cust_FName VARCHAR(15) NOT NULL ,
```

```
t3_Cust_LName VARCHAR(15) NOT NULL ,
```

```
t3_Cust_DOB DATE NOT NULL ,
```

```
t3_Cust_Gender CHAR(2) NOT NULL ,
```

```
t3_Cust_Address VARCHAR(35) NOT NULL ,
```

```
t3_Cust_MOB_Number BIGINT NOT NULL ,
```

```
t3_Cust_Email VARCHAR(25) NULL ,
```

```
t3_Cust_Passport_Number VARCHAR(20) NULL ,
```

```
t3_Cust_Marital_Status CHAR(12) NULL ,  
t3_Cust_PPS_Number INTEGER NULL ,  
CONSTRAINT XPKCUSTOMER PRIMARY KEY (t3_Cust_Id)  
);
```

```
CREATE UNIQUE INDEX XPKCUSTOMER_1 ON t3_CUSTOMER (t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_INCIDENT_REPORT  
(  
    Incident_Report_Id VARCHAR(20) NOT NULL ,  
    t3_Incident_Type CHAR(50) NULL ,  
    t3_Incident_Inspector VARCHAR(20) NULL ,  
    t3_Incident_Cost INTEGER NULL ,  
    t3_Incident_Report_Description VARCHAR(1000) NULL ,  
    t3_Incident_Id VARCHAR(20) NOT NULL ,  
    t3_Cust_Id VARCHAR(20) NOT NULL ,  
    CONSTRAINT XPKINCIDENT_REPORT_18 PRIMARY KEY  
    (Incident_Report_Id,t3_Incident_Id,t3_Cust_Id),  
    CONSTRAINT R_83 FOREIGN KEY (t3_Incident_Id) REFERENCES t3_INCIDENT (t3_Incident_Id) ON  
    DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT R_86 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER (t3_Cust_Id) ON  
    DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE UNIQUE INDEX XPKINCIDENT_REPORT_18 ON t3_INCIDENT_REPORT (Incident_Report_Id ASC,t3_Incident_Id  
ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_INSURANCE_COMPANY
(
    t3_Company_Name VARCHAR(70) NOT NULL ,
    t3_Company_Address VARCHAR(400) NULL ,
    t3_Company_Contact_Number bigint NULL ,
    t3_Company_Fax bigint NULL ,
    t3_Company_Email VARCHAR(50) NULL ,
    t3_Company_Website VARCHAR(50) NULL ,
    t3_Company_Location VARCHAR(50) NULL,
    t3_Company_Department_Name VARCHAR(50) NULL ,
    t3_Company_Office_Name VARCHAR(50) NULL ,
    CONSTRAINT XPKINSURANCE_COMPANY_15 PRIMARY KEY
    (t3_Company_Name)
);

CREATE UNIQUE INDEX XPKINSURANCE_COMPANY_15 ON t3_INSURANCE_COMPANY
(t3_Company_Name ASC);

-- Table 5

CREATE TABLE IF NOT EXISTS t3_DEPARTMENT
(
    t3_Department_Name VARCHAR(50) NOT NULL ,
    t3_Department_ID VARCHAR(50) NOT NULL ,
    t3_Department_Staff VARCHAR(50) NULL ,
    t3_Company_Name VARCHAR(100) NOT NULL ,
    CONSTRAINT XPKDEPARTMENT PRIMARY KEY
    (t3_Department_Name,t3_Department_ID,t3_Company_Name),
    CONSTRAINT R_56 FOREIGN KEY (t3_Company_Name)
```

REFERENCES t3_INSURANCE_COMPANY (t3_Company_Name) ON DELETE CASCADE ON UPDATE
CASCADE

);

CREATE UNIQUE INDEX XPKDEPARTMENT ON t3_DEPARTMENT

(t3_Department_Name ASC,t3_Department_ID ASC,t3_Company_Name ASC);

CREATE TABLE t3_VEHICLE_SERVICE

(

t3_Department_Name VARCHAR(100) NOT NULL ,

t3_Vehicle_Service_Company_Name VARCHAR(10) NOT NULL ,

t3_Vehicle_Service_Address VARCHAR(200) NULL ,

t3_Vehicle_Service_Contact VARCHAR(200) NULL ,

t3_Vehicle_Service_Incharge VARCHAR(200) NULL ,

t3_Vehicle_Service_Type VARCHAR(200) NULL ,

t3_Department_ID VARCHAR(200) NOT NULL ,

t3_Company_Name VARCHAR(200) NOT NULL ,

CONSTRAINT XPKVEHICLE_SERVICE PRIMARY KEY

(t3_Vehicle_Service_Company_Name,t3_Department_Name),

CONSTRAINT R_50 FOREIGN KEY (t3_Department_Name, t3_Department_ID,

t3_Company_Name) REFERENCES t3_DEPARTMENT (t3_Department_Name,

t3_Department_ID, t3_Company_Name) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKVEHICLE_SERVICE ON t3_VEHICLE_SERVICE

(t3_Vehicle_Service_Company_Name ASC,t3_Department_Name ASC);

```
CREATE TABLE t3_VEHICLE
(
    t3_Vehicle_Id VARCHAR(20) NOT NULL ,
    t3_Policy_Id VARCHAR(20) NULL ,
    t3_Vehicle_Registration_Number VARCHAR(20) NOT NULL ,
    t3_Vehicle_Value bigint NULL ,
    t3_Vehicle_Type VARCHAR(20) NOT NULL ,
    t3_Vehicle_Size INTEGER NULL ,
    t3_Vehicle_Number_Of_Seat INTEGER NULL ,
    t3_Vehicle_Manufacturer VARCHAR(20) NULL ,
    t3_Vehicle_Engine_Number INTEGER NULL ,
    t3_Vehicle_Chassis_Number INTEGER NULL ,
    t3_Vehicle_Number INTEGER NULL ,
    t3_Vehicle_Model_Number VARCHAR(20) NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKVEHICLE_6 PRIMARY KEY (t3_Vehicle_Id,t3_Cust_Id),
    CONSTRAINT R_92 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER
    (t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE UNIQUE INDEX XPKVEHICLE_6 ON t3_VEHICLE (t3_Vehicle_Id ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE t3_PREMIUM_PAYMENT
(
    t3_Premium_Payment_Id VARCHAR(20) NOT NULL ,
    t3_Policy_Number VARCHAR(20) NOT NULL ,
```

```
t3_Premium_Payment_Amount INTEGER NOT NULL ,  
t3_Premium_Payment_Schedule DATE NOT NULL ,  
t3_Receipt_Id VARCHAR(20) NOT NULL ,  
t3_Cust_Id VARCHAR(20) NOT NULL ,  
CONSTRAINT XPKPREMIUM_PAYMENT_5 PRIMARY KEY  
(t3_Premium_Payment_Id,t3_Cust_Id),  
CONSTRAINT R_85 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER  
(t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE  
);  
  
CREATE UNIQUE INDEX XPKPREMIUM_PAYMENT_5 ON  
t3_PREMIUM_PAYMENT  
(t3_Premium_Payment_Id ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_RECEIPT  
(  
t3_Receipt_Id VARCHAR(20) NOT NULL ,  
t3_Time DATE NOT NULL ,  
t3_Cost INTEGER NOT NULL ,  
t3_Premium_Payment_Id VARCHAR(20) NOT NULL ,  
t3_Cust_Id VARCHAR(20) NOT NULL ,  
CONSTRAINT XPKRECEIPT_21 PRIMARY KEY (t3_Receipt_Id,t3_Premium_Payment_Id,t3_Cust_Id),  
CONSTRAINT R_84 FOREIGN KEY (t3_Premium_Payment_Id, t3_Cust_Id)  
REFERENCES t3_PREMIUM_PAYMENT (t3_Premium_Payment_Id, t3_Cust_Id) ON DELETE  
CASCADE ON UPDATE CASCADE  
);  
  
CREATE UNIQUE INDEX XPKRECEIPT_21 ON t3_RECEIPT (t3_Receipt_Id ASC,t3_Premium_Payment_Id ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_APPLICATION
(
    t3_Application_Id VARCHAR(20) NOT NULL ,
    t3_Vehicle_Id VARCHAR(20) NOT NULL ,
    t3_Application_Status CHAR(8) NOT NULL ,
    t3_Coverage VARCHAR(50) NOT NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKAPPLICATION_2 PRIMARY KEY (t3_Application_Id,t3_Cust_Id),
    CONSTRAINT R_93 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER
    (t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE UNIQUE INDEX XPKAPPLICATION_2 ON t3_APPLICATION
(t3_Application_Id ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_INSURANCE_POLICY
(
    t3_Agreement_id VARCHAR(20) NOT NULL ,
    t3_Department_Name VARCHAR(20) NULL ,
    t3_Policy_Number VARCHAR(20) NULL ,
    t3_Start_Date DATE NULL ,
    t3_Expiry_Date DATE NULL ,
    t3_Term_Condition_Description VARCHAR(400) NULL ,
    t3_Application_Id VARCHAR(20) NOT NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
```


CONSTRAINT XPKINSURANCE_POLICY_4 PRIMARY KEY

(t3_Agreement_id,t3_Application_Id,t3_Cust_Id),

CONSTRAINT R_95 FOREIGN KEY (t3_Application_Id, t3_Cust_Id) REFERENCES t3_APPLICATION

(t3_Application_Id, t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKINSURANCE_POLICY_4 ON t3_INSURANCE_POLICY

(t3_Agreement_id ASC,t3_Application_Id ASC,t3_Cust_Id ASC);

CREATE TABLE t3_POLICY_RENEWABLE

(

t3_Policy_Renewable_Id VARCHAR(20) NOT NULL ,

t3_Date_Of_Renewal DATE NOT NULL ,

t3_Type_Of_Renewal CHAR(15) NOT NULL ,

t3_Agreement_id VARCHAR(20) NOT NULL ,

t3_Application_Id VARCHAR(20) NOT NULL ,

t3_Cust_Id VARCHAR(20) NOT NULL ,

CONSTRAINT XPKPOLICY_RENEWABLE_16 PRIMARY KEY

(t3_Policy_Renewable_Id,t3_Agreement_id,t3_Application_Id,t3_Cust_Id),

CONSTRAINT R_101 FOREIGN KEY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id)

REFERENCES t3_INSURANCE_POLICY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id) ON DELETE

CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKPOLICY_RENEWABLE_16 ON

t3_POLICY_RENEWABLE

(t3_Policy_Renewable_Id ASC,t3_Agreement_id ASC,t3_Application_Id

ASC,t3_Cust_Id ASC);

```
CREATE TABLE IF NOT EXISTS t3_MEMBERSHIP
(
    t3_Membership_Id VARCHAR(20) NOT NULL ,
    t3_Membership_Type CHAR(15) NOT NULL ,
    t3_Organisation_Contact VARCHAR(20) NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
    CONSTRAINT XPKMEMBERSHIP_12 PRIMARY KEY
    (t3_Membership_Id,t3_Cust_Id),
    CONSTRAINT R_91 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER
    (t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE UNIQUE INDEX XPKMEMBERSHIP_12 ON t3_MEMBERSHIP
(t3_Membership_Id ASC,t3_Cust_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_QUOTE
(
    t3_Quote_Id VARCHAR(20) NOT NULL ,
    t3_Issue_Date DATE NOT NULL ,
    t3_Valid_From_Date DATE NOT NULL ,
    t3_Valid_Till_Date DATE NOT NULL ,
    t3_Description VARCHAR(100) NULL ,
    t3_Product_Id VARCHAR(20) NOT NULL ,
    t3_Coverage_Level VARCHAR(20) NOT NULL ,
    t3_Application_Id VARCHAR(20) NOT NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
```

CONSTRAINT XPKQUOTE_3 PRIMARY KEY

(t3_Quote_Id,t3_Application_Id,t3_Cust_Id),

CONSTRAINT R_94 FOREIGN KEY (t3_Application_Id, t3_Cust_Id) REFERENCES

t3_APPLICATION (t3_Application_Id, t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKQUOTE_3 ON t3_QUOTE

(t3_Quote_Id ASC,t3_Application_Id ASC,t3_Cust_Id ASC);

CREATE TABLE IF NOT EXISTS t3_STAFF

(

t3_Staff_Id VARCHAR(200) NOT NULL ,

t3_Staff_Fname VARCHAR(100) NULL ,

t3_Staff_LName VARCHAR(100) NULL ,

t3_Staff_Address VARCHAR(200) NULL ,

t3_Staff_Contact BIGINT NULL ,

t3_Staff_Gender CHAR(2) NULL ,

t3_Staff_Marital_Status CHAR(8) NULL ,

t3_Staff_Nationality CHAR(15) NULL ,

t3_Staff_Qualification VARCHAR(20) NULL ,

t3_Staff_Allowance bigint NULL ,

t3_Staff_PPS_Number bigint NULL ,

t3_Company_Name VARCHAR(400) NOT NULL ,

CONSTRAINT XPKSTAFF_9 PRIMARY KEY (t3_Staff_Id,t3_Company_Name),

CONSTRAINT R_105 FOREIGN KEY (t3_Company_Name) REFERENCES

t3_INSURANCE_COMPANY (t3_Company_Name) ON DELETE CASCADE ON UPDATE CASCADE

);

```
CREATE UNIQUE INDEX XPKSTAFF_9 ON t3_STAFF (t3_Staff_Id ASC,t3_Company_Name ASC);
```

```
CREATE TABLE t3_NOK
```

```
(
```

```
t3_Nok_Id VARCHAR(20) NOT NULL ,
```

```
t3_Nok_Name VARCHAR(20) NULL ,
```

```
t3_Nok_Address VARCHAR(200) NULL ,
```

```
t3_Nok_Phone_Number BIGINT NULL ,
```

```
t3_Nok_Gender CHAR(10) NULL ,
```

```
t3_Nok_Marital_Status CHAR(8) NULL ,
```

```
t3_Agreement_id VARCHAR(20) NOT NULL ,
```

```
t3_Application_Id VARCHAR(20) NOT NULL ,
```

```
t3_Cust_Id VARCHAR(20) NOT NULL ,
```

```
CONSTRAINT XPKNOK_14 PRIMARY KEY
```

```
(t3_Nok_Id,t3_Agreement_id,t3_Application_Id,t3_Cust_Id),
```

```
CONSTRAINT R_99 FOREIGN KEY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id)
```

```
REFERENCES t3_INSURANCE_POLICY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id) ON DELETE
```

```
CASCADE ON UPDATE CASCADE
```

```
);
```

```
CREATE UNIQUE INDEX XPKNOK_14 ON t3_NOK (t3_Nok_Id ASC,t3_Agreement_id ASC,t3_Application_Id ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_PRODUCT
```

```
(
```

```
t3_Product_Number VARCHAR(200) NOT NULL ,
```

```
t3_Product_Price INTEGER NULL ,
```

```
t3_Product_Type CHAR(40) NULL ,  
t3_Company_Name VARCHAR(200) NOT NULL ,  
CONSTRAINT XPKPRODUCT_20 PRIMARY KEY (t3_Product_Number,t3_Company_Name),  
CONSTRAINT R_107 FOREIGN KEY (t3_Company_Name) REFERENCES t3_INSURANCE_COMPANY  
(t3_Company_Name) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE UNIQUE INDEX XPKPRODUCT_20 ON t3_PRODUCT (t3_Product_Number ASC,t3_Company_Name ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_OFFICE  
(  
t3_Office_Name VARCHAR(200) NOT NULL ,  
t3_Office_Leader VARCHAR(200) NOT NULL ,  
t3_Contact_Information VARCHAR(200) NOT NULL ,  
t3_Address VARCHAR(200) NOT NULL ,  
t3_Admin_Cost INTEGER NULL ,  
t3_Staff VARCHAR(50) NULL ,  
t3_Department_Name VARCHAR(200) NOT NULL ,  
t3_Department_ID VARCHAR(200) NOT NULL ,  
t3_Company_Name VARCHAR(200) NOT NULL ,  
CONSTRAINT XPKOFFICE_11 PRIMARY  
KEY(t3_Office_Name,t3_Department_Name,t3_Company_Name),  
CONSTRAINT R_104 FOREIGN KEY (t3_Department_Name,t3_Department_ID,t3_Company_Name)  
REFERENCES t3_DEPARTMENT (t3_Department_Name,t3_Department_ID,t3_Company_Name) ON  
DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE UNIQUE INDEX XPKOFFICE_11 ON t3_OFFICE (t3_Office_Name ASC,t3_Department_Name ASC,t3_Company_Name ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_COVERAGE
(
    t3_Coverage_Id VARCHAR(20) NOT NULL ,
    t3_Coverage_Amount INTEGER NOT NULL ,
    t3_Coverage_Type CHAR(30) NOT NULL ,
    t3_Coverage_Level CHAR(30) NOT NULL ,
    t3_Product_Id VARCHAR(30) NOT NULL ,
    t3_Coverage_Description VARCHAR(400) NULL ,
    t3_Coverage_Terms VARCHAR(50) NULL ,
    t3_Company_Name VARCHAR(400) NOT NULL ,
    CONSTRAINT XPKCOVERAGE_19 PRIMARY KEY
    (t3_Coverage_Id,t3_Company_Name),
    CONSTRAINT R_102 FOREIGN KEY (t3_Company_Name) REFERENCES
    t3_INSURANCE_COMPANY (t3_Company_Name) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE UNIQUE INDEX XPKCOVERAGE_19 ON t3_COVERAGE
(t3_Coverage_Id ASC,t3_Company_Name ASC);
```

```
CREATE TABLE IF NOT EXISTS t3_INSURANCE_POLICY_COVERAGE
(
    t3_Agreement_id VARCHAR(20) NOT NULL ,
    t3_Application_Id VARCHAR(20) NOT NULL ,
    t3_Cust_Id VARCHAR(20) NOT NULL ,
    t3_Coverage_Id VARCHAR(20) NOT NULL ,
```

```
t3_Company_Name VARCHAR(200) NOT NULL ,  
  
CONSTRAINT XPKINSURANCE_POLICY_4_COVERAGE PRIMARY KEY  
(t3_Agreement_id,t3_Application_Id,t3_Cust_Id,t3_Coverage_Id,t3_Company_Name),  
  
CONSTRAINT R_97 FOREIGN KEY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id)  
REFERENCES t3_INSURANCE_POLICY (t3_Agreement_id, t3_Application_Id, t3_Cust_Id) ON DELETE  
CASCADE ON UPDATE CASCADE,  
  
CONSTRAINT R_98 FOREIGN KEY (t3_Coverage_Id, t3_Company_Name)  
REFERENCES t3_COVERAGE (t3_Coverage_Id, t3_Company_Name) ON DELETE CASCADE ON  
UPDATE CASCADE  
  
);  
  
CREATE UNIQUE INDEX XPKINSURANCE_POLICY_4_COVERAGE ON  
t3_INSURANCE_POLICY_COVERAGE  
(t3_Agreement_id ASC,t3_Application_Id ASC,t3_Cust_Id ASC,t3_Coverage_Id  
ASC,t3_Company_Name ASC);  
  
  
CREATE TABLE IF NOT EXISTS t3_CLAIM  
(  
  
t3_Claim_Id VARCHAR(20) NOT NULL ,  
  
t3_Agreement_id VARCHAR(20) NOT NULL ,  
  
t3_Claim_Amount INTEGER NOT NULL ,  
  
t3_Incident_Id VARCHAR(20) NOT NULL ,  
  
t3_Damage_Type VARCHAR(20) NOT NULL ,  
  
t3_Date_Of_Claim DATE NOT NULL ,  
  
t3_Claim_Status CHAR(10) NOT NULL ,  
  
t3_Cust_Id VARCHAR(20) NOT NULL ,  
  
CONSTRAINT XPKCLAIM_7 PRIMARY KEY (t3_Claim_Id,t3_Cust_Id),
```

CONSTRAINT R_88 FOREIGN KEY (t3_Cust_Id) REFERENCES t3_CUSTOMER

(t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKCLAIM_7 ON t3_CLAIM

(t3_Claim_Id ASC,t3_Cust_Id ASC);

CREATE TABLE IF NOT EXISTS t3_CLAIM_SETTLEMENT

(

t3_Claim_Settlement_Id VARCHAR(20) NOT NULL ,

t3_Vehicle_Id VARCHAR(20) NOT NULL ,

t3_Date_Settled DATE NOT NULL ,

t3_Amount_Paid INTEGER NOT NULL ,

t3_Coverage_Id VARCHAR(20) NOT NULL ,

t3_Claim_Id VARCHAR(20) NOT NULL ,

t3_Cust_Id VARCHAR(20) NOT NULL ,

CONSTRAINT XPKCLAIM_SETTLEMENT_8 PRIMARY KEY

(t3_Claim_Settlement_Id,t3_Claim_Id,t3_Cust_Id),

CONSTRAINT R_90 FOREIGN KEY (t3_Claim_Id, t3_Cust_Id) REFERENCES t3_CLAIM

(t3_Claim_Id, t3_Cust_Id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE UNIQUE INDEX XPKCLAIM_SETTLEMENT_8 ON

t3_CLAIM_SETTLEMENT

(t3_Claim_Settlement_Id ASC,t3_Claim_Id ASC,t3_Cust_Id ASC);

TESTING QUERY 1

SELECT

*

FROM

t3_CUSTOMER,

t3_VEHICLE

WHERE

t3_CUSTOMER.t3_Cust_Id IN (SELECT

t3_Cust_Id

FROM

t3_INCIDENT_REPORT

WHERE

t3_Cust_Id IN (SELECT

t3_Cust_Id

FROM

t3_CLAIM

WHERE

t3_Claim_Status = 'Pending'))

AND t3_CUSTOMER.t3_Cust_Id = t3_VEHICLE.t3_Cust_Id;

OUTPUT QUERY 1

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window displays a SQL query in the editor, which is a complex join query involving several tables. The query is as follows:

```

SELECT *
FROM
  t3_CUSTOMER,
  t3_VEHICLE
WHERE
  t3_CUSTOMER.t3_Cust_Id IN (SELECT
    t3_Cust_Id
  FROM
    t3_INCIDENT_REPORT
  WHERE
    t3_Cust_Id IN (SELECT
      t3_Cust_Id
    FROM
      t3_CLAIM
    WHERE
      t3_Claim_Status = 'Pending'))
  AND t3_CUSTOMER.t3_Cust_Id = t3_VEHICLE.t3_Cust_Id;

```

The results are displayed in a table with the following columns: #, t3_Cust_Id, t3_Cust_FName, t3_Cust_LName, t3_Cust_DOB, t3_Cust_Gender, t3_Cust_Address, t3_Cust_MOB_Numbe, t3_Cust_Email, and t3_Cust_Passport. The results show 5 rows of data:

#	t3_Cust_Id	t3_Cust_FName	t3_Cust_LName	t3_Cust_DOB	t3_Cust_Gender	t3_Cust_Address	t3_Cust_MOB_Numbe	t3_Cust_Email	t3_Cust_Passport
1	30219	Diksha	Chatterjee	1997-10-23	F	Faridabad	6607568394	mail0843@company.com	6998758364
2	30202	Rohit	Roy	2000-07-28	M	Pune	1107568394	mail0342162@company.com	8126759364
3	30210	Mohammad	Rajhman	1979-04-23	M	Vizag	4007568394	mail758@company.com	7866651264
4	30216	Yashvi	Mehta	1989-10-23	F	Lucknow	2207568394	mail08s423@company.com	7016758364
5	30201	Amitabh	Paliwal	1994-09-06	M	Haldwani	6963456969	mail904@company.com	9103258395

The bottom status bar indicates that the SQL script was saved to 'home/lifenomeng/Desktop/Queries/3.sql'.

TESTING QUERY 2

SELECT

*

FROM

t3_CUSTOMER

WHERE

t3_Cust_id IN (SELECT

t3_Cust_Id

FROM

t3_PREMIUM_PAYMENT

WHERE

t3_Premium_Payment_Amount > (SELECT

```

SUM(t3_Cust_Id)

FROM

t3_CUSTOMER));

```

OUTPUT QUERY 2

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```

1 use mydb;
2 SELECT
3 *
4 FROM
5 t3_CUSTOMER
6 WHERE
7 t3_Cust_Id IN (SELECT
8 t3_Cust_Id
9 FROM
10 t3_PREMIUM_PAYMENT
11 WHERE
12 t3_Premium_Payment_Amount > (SELECT
13 SUM(t3_Cust_Id)
14 FROM
15 t3_CUSTOMER));
16

```

The result grid displays the following data:

#	t3_Cust_Id	t3_Cust_FName	t3_Cust_LName	t3_Cust_DOB	t3_Cust_Gender	t3_Cust_Address	t3_Cust_MOB_Numbe	t3_Cust_Email	t3_Cust_Passport
1	30203	Deepak	Gupta	1999-07-23	M	Bangalore	3307568394	mail0843@company.com	7396700364
2	30212	Aditi	Singhal	1967-07-23	F	Lucknow	1107568394	mail084322@company.com	7442458364
3	30207	Sunny	Gupta	1974-06-23	M	Surat	7007568394	mail0842@company.com	7496126664
4	30205	Divyani	Jha	1998-09-23	F	Bangalore	9907568394	mail0743@company.com	9496658064
5	30204	Bhavbhuti	Pande	1999-12-23	M	Bangalore	4407568394	mail041@company.com	7256765364

The status bar at the bottom indicates "Query Completed".

TESTING QUERY 3

```

SELECT

*

FROM

t3_INSURANCE_COMPANY

WHERE

t3_Company_Name IN (SELECT

t3_Company_Name

FROM

```

t3_OFFICE

GROUP BY t3_Company_Name

HAVING COUNT(DISTINCT (t3_Address)) > 1)

AND t3_Company_Name IN (SELECT

t3_DEPARTMENT.t3_Company_Name

FROM

t3_PRODUCT

INNER JOIN

t3_DEPARTMENT ON t3_DEPARTMENT.t3_Company_Name = t3_PRODUCT.t3_Company_Name

GROUP BY t3_DEPARTMENT.t3_Company_Name

HAVING COUNT(DISTINCT (t3_Product_Type)) > COUNT(DISTINCT (t3_Department_Name)));

OUTPUT QUERY 3

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 use mydb;
2
3 SELECT
4 *
5 FROM
6 t3_INSURANCE_COMPANY
7 WHERE
8 t3_Company_Name IN (SELECT
9 t3_Company_Name
10 FROM
11 t3_OFFICE
12 GROUP BY t3_Company_Name
13 HAVING COUNT(DISTINCT (t3_Address)) > 1)
14 AND t3_Company_Name IN (SELECT
15 t3_DEPARTMENT.t3_Company_Name
16 FROM
17 t3_PRODUCT
18 INNER JOIN
19 t3_DEPARTMENT ON t3_DEPARTMENT.t3_Company_Name = t3_PRODUCT.t3_Company_Name
20 GROUP BY t3_DEPARTMENT.t3_Company_Name
21 HAVING COUNT(DISTINCT (t3_Product_Type)) > COUNT(DISTINCT (t3_Department_Name)));
  
```

The Results Grid shows the following data:

#	t3_Company_Name	t3_Company_Address	t3_Company_Contact_Numbr	t3_Company_Fax	t3_Company_Email	t3_Company_Website
1	Birla Sun Vehicle Insurance Compa...	One World Center, Tower 1, 16th FL...	18002707000	18002707000	care.Vehicleinsurance@adityabirla...	https://www.birlasun.co.in
2	Kotak Vehicle Insurance Company ...	Kotak Mahindra Vehicle Insurance ...	1800287867160	556622020	KotakVehicleinsurance@help.com	KotakVehicleInsurance.com
3	SBI Vehicle Insurance Company L...	SBI Vehicle Corporate Solutions, 7L...	180571584020	5565871687	sbiVehicle@help.com	sbiVehicle.com

The status bar at the bottom indicates "Query Completed".

TESTING QUERY 4**FROM****t3_CUSTOMER****WHERE****t3_Cust_id IN (SELECT****t1.t3_Cust_Id****FROM****t3_VEHICLE AS t1****INNER JOIN****t3_INCIDENT_REPORT AS t2 ON t1.t3_Cust_Id = t2.t3_Cust_Id****LEFT JOIN****t3_PREMIUM_PAYMENT AS t3 ON t2.t3_Cust_Id = t3.t3_Cust_Id WHERE t3.t3_Cust_Id IS NULL****GROUP BY t1.t3_Cust_Id****HAVING COUNT(t1.t3_Vehicle_Id) > 1);**

OUTPUT QUERY 4

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

1 use mydb;
2 SELECT DISTINCT
3 *
4 FROM
5 t3_CUSTOMER
6 WHERE
7 t3_Cust_Id IN (SELECT
8 t1.t3_Cust_Id
9 FROM
10 t3_VEHICLE AS t1
11 INNER JOIN
12 t3_INCIDENT_REPORT AS t2 ON t1.t3_Cust_Id = t2.t3_Cust_Id
13 LEFT JOIN
14 t3_PREMIUM_PAYMENT AS t3 ON t2.t3_Cust_Id = t3.t3_Cust_Id WHERE t3.t3_Cust_Id IS NULL
15 GROUP BY t1.t3_Cust_Id
16 HAVING COUNT(t1.t3_Vehicle_Id) > 1);
17

```

The result grid shows the following data:

#	t3_Cust_Id	t3_Cust_FName	t3_Cust_LName	t3_Cust_DOB	t3_Cust_Gender	t3_Cust_Address	t3_Cust_MOB_Numbe	t3_Cust_Email	t3_Cust_Passport_Nu
1	30205	Diyanti	Jha	1998-09-23	F	Bangalore	9907568394	mail0743@company.com	9496658064
2	30208	Shivang	Bisht	1999-07-23	M	Bangalore	9907568394	mail0732@company.com	7496545564
3	30211	Aakriti	Pant	1999-07-23	F	Hyderabad	7707568394	mail0451@company.com	7696658364

The status bar at the bottom indicates "Query Completed".

TESTING QUERY 5

SELECT

t1.*

FROM

t3_VEHICLE AS t1,

t3_PREMIUM_PAYMENT AS t2

WHERE

t2.t3_Premium_Payment_Amount > t1.t3_Vehicle_Number

AND t1.t3_Cust_Id = t2.t3_Cust_Id;

OUTPUT QUERY 5

MySQL Workbench - Nov 24 00:13

actual

File Edit View Query Database Server Tools Scripting Help

Administration Schemas

Filter objects

1 use mydb;
2 SELECT
3 t1.*
4 FROM
5 t3_VEHICLE AS t1,
6 t3_PREMIUM_PAYMENT AS t2
7 WHERE
8 t2.t3_Premium_Payment_Amount > t1.t3_Vehicle_Number
9 AND t1.t3_Cust_Id = t2.t3_Cust_Id;

Limit to 1000 rows

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Object Info Session

No object selected

Result Grid

#	t3_Vehicle_Id	t3_Policy_Id	t3_Vehicle_Registration_Numb	t3_Vehicle_Value	t3_Vehicle_Type	t3_Vehicle_Size	t3_Vehicle_Number_Of_Seats	t3_Vehicle_Manufacture
1	30705	2369893	UK 43 CX 5047	75000	2 wheeler	13	2	TVS
2	30704	2487318	MP 27 LO 1161	46990	2 wheeler	19	2	Hero
3	30705	4503208	WB 12 EV 5245	1690000	4 wheeler	25	8	Mahindra
4	30706	4883265	UP 76 FU 4564	80000	2 wheeler	15	2	Hero
5	30707	2855545	UK 89 GH 2646	800000	4 wheeler	12	4	Volkswagon
6	30709	2567548	UP 45 CD 4634	1100000	4 wheeler	15	4	Ford
7	30718	4454729	HP 56 HI 3727	90000	2 wheeler	15	2	Hero

Result 9

Query Completed

TESTING QUERY 6

SELECT DISTINCT

t1.*

FROM

t3_CUSTOMER AS t1,

t3_COVERAGE AS t2,

t3_CLAIM_SETTLEMENT AS t3,

t3_CLAIM AS t4

WHERE

t4.t3_Claim_Amount < t2.t3_Coverage_Amount

AND t4.t3_Claim_Amount > (t3.t3_Claim_Settlement_Id + t3.t3_Vehicle_Id + t3.t3_Claim_Id +
t3.t3_Cust_Id)

AND t1.t3_Cust_Id = t4.t3_Cust_Id;

OUTPUT QUERY 6

The screenshot displays the MySQL Workbench interface. The main window shows a SQL query in the editor, which is a complex join query involving tables t1, t2, t3, and t4. The query filters for claims where the claim amount is less than the coverage amount and where the claim settlement ID, vehicle ID, claim ID, and customer ID are all non-zero. The results are displayed in a table with 10 rows and 10 columns.

Query:

```

1 use mydb;
2 SELECT DISTINCT
3   t1.*
4 FROM
5   t3_CUSTOMER AS t1,
6   t3_COVERAGE AS t2,
7   t3_CLAIM_SETTLEMENT AS t3,
8   t3_CLAIM AS t4
9 WHERE
10  t4.t3_Claim_Amount < t2.t3_Coverage_Amount
11  AND t4.t3_Claim_Amount > 0
12  AND t1.t3_Cust_Id = t4.t3_Cust_Id;

```

Result Grid:

#	t3_Cust_Id	t3_Cust_FName	t3_Cust_LName	t3_Cust_DOB	t3_Cust_Gender	t3_Cust_Address	t3_Cust_MOB_Numbe	t3_Cust_Email	t3_Cust_Passwor
1	30212	Aditi	Singhal	1967-07-23	F	Lucknow	1107568394	mail084322@company.com	7442458364
2	30201	Amitabh	Paliwal	1994-09-06	M	Haldwani	6963456969	mail0904@company.com	9103258395
3	30216	Yashvi	Mehta	1989-10-23	F	Lucknow	2207568394	mail08s423@company.com	7016758364
4	30210	Mohammad	Rayhaan	1979-04-23	M	Vizag	4007568394	mail758@company.com	7666651264
5	30214	Sahana	NH	1998-07-23	F	Mumbai	8307468394	mail384@company.com	7666756364
6	30202	Rohit	Roy	2000-07-23	M	Pune	1107568394	mail0342182@company.com	8126759364
7	30219	Diksha	Chatterjee	1997-10-23	F	Faridabad	6607568394	mail0843@company.com	6998758364
8	30208	Shivang	Bisht	1969-07-23	M	Bangalore	9007568394	mail0732@company.com	7496546564
9	30204	Bhavbhuti	Pande	1999-12-23	M	Bangalore	4407568394	mail041@company.com	7256765364
10	30217	Souvik	Das	1989-12-04	M	Kolkata	9697643296	mail399@company.com	6583943462

Conclusion

We have a class project created database (DB) with all documentations and reports included. Our goal was to create a DB for Online vehicle insurance companies with code generated for MySQL workbench. There were some big and small challenges but we succeeded in making a functional DB. We started to build conceptual data models (CDM), we continued with logical data models (LDM) and then we made physical data models (PDM). From the physical data model we created a code to be run in MySQL database management system (DBMS). For better understanding for a reader and for our learning we included some theory in each phase we did and documented in the project initial document (PID) with reports of progress and work being done.