DBMS Project

Project Report on
Vehicle Insurance Company

Presented by:

Anil Kumar-20BCS015
Dhanisht Kumar-20BCS040
Janardhana Reddy- 20BCS062
Samarth Kolkar-20BCS115
Ravikant-20BCS110
Khushi G K-20BCS071
M Manjunath-20BCS078
Kavya Garg-20BCS069

Under the Guidance of

Dr. Pramod Yelmewad
Dr. Supriya Nadiger
Dr. Uma Sheshadri
IIIT Dharwad

By DBMS Team 11

Approved by:
Indian Institue of Information Technology, Dharwad

INDIAN INSTITUTE OF
INFORMATION
TECHNOLOGY

Table of Contents

Abstract

A contract by which the insurer assumes the risk of any loss the owner or operator of a car may incur through damage to property or persons as the result of an accident. In the given project we have created the tables and inserted data according to the queries required. The data was retrieved from some insurance companies. The queries are written in an easy and approachable method. The project allows us to understand the depth and importance of database management systems. It helps us understand that using DBMS is an easier way to store data.

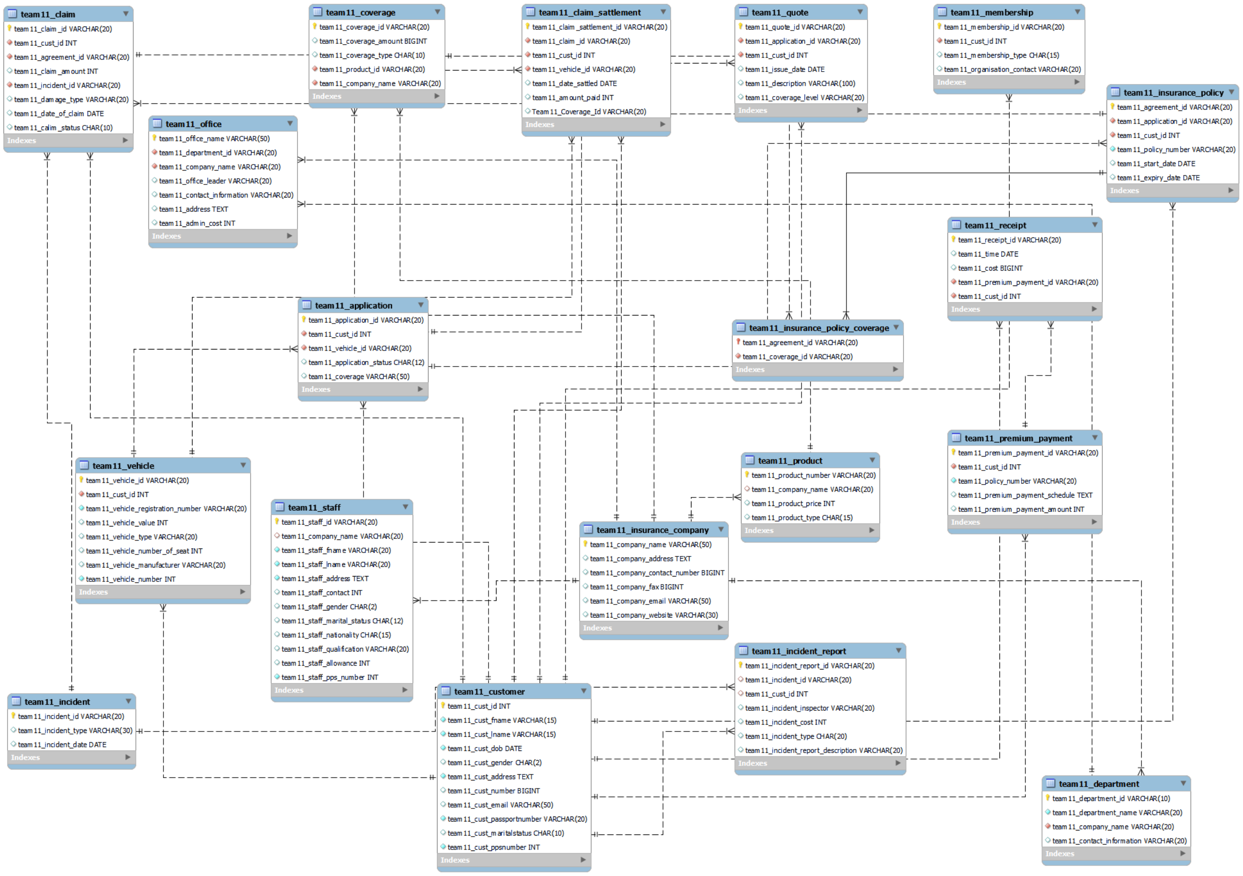
Introduction

Database management systems are software systems used to manage and manipulate data in a database. As most application performance issues originate in the database, knowing how to monitor and optimize your database is essential to your operations.

Aim: To provide a comprehensive understanding and practical experience in data modeling, use of normalization techniques, transactional relational database design and implementation of SQL queries.

Approach: We attempted to understand all of the requirements needed to develop an error-free database after reading the auto insurance company's pdf. To comprehend the relationships between different things, we constructed a conceptual data model. We also generated an entity relationship diagram (ER-Diagram), as well as all of the tables and data needed to run all of the queries.

Physical Data Model:


team11_claim
team11_claim_id VARCHAR(20)
team11_cust_id INT
team11_agreement_id VARCHAR(20)
team11_claim_amount INT
team11_incident_id VARCHAR(20)
team11_damage_type VARCHAR(20)
team11_date_of_claim DATE
team11_calim_status CHAR(10)
Indexes
team11_coverage
team11_coverage_id VARCHAR(20)
team11_coverage_amount BIGINT
team11_coverage_type CHAR(10)
team11_product_id VARCHAR(20)
team11_company_name VARCHAR(20)
Indexes
team11_claim_sattlement
team11_claim_sattlement_id VARCHAR(20)
team11_claim_id VARCHAR(20)
team11_cust_id INT
team11_vehicle_id VARCHAR(20)
team11_date_sattled DATE
team11_amount_paid INT
Team11_Coverage_Id VARCHAR(20)
Indexes
team11_quote
team11_quote_id VARCHAR(20)
team11_application_id VARCHAR(20)
team11_cust_id INT
team11_issue_date DATE
team11_description VARCHAR(100)
team11_coverage_level VARCHAR(20)
Indexes
team11_membership
team11_membership_id VARCHAR(20)
team11_cust_id INT
team11_membership_type CHAR(15)
team11_organisation_contact VARCHAR(20)
Indexes
team11_insurance_policy
team11_agreement_id VARCHAR(20)
team11_application_id VARCHAR(20)
team11_cust_id INT
team11_policy_number VARCHAR(20)
team11_start_date DATE
team11_expiry_date DATE
Indexes
team11_office
team11_office_name VARCHAR(50)
team11_department_id VARCHAR(20)
team11_company_name VARCHAR(20)
team11_office_leader VARCHAR(20)
team11_contact_information VARCHAR(20)
team11_address TEXT
team11_admin_cost INT
Indexes
team11_receipt
team11_receipt_id VARCHAR(20)
team11_time DATE
team11_cost BIGINT
team11_premium_payment_id VARCHAR(20)
team11_cust_id INT
Indexes
team11_application
team11_application_id VARCHAR(20)
team11_cust_id INT
team11_vehicle_id VARCHAR(20)
team11_application_status CHAR(12)
team11_coverage VARCHAR(50)
Indexes
team11_insurance_policy_coverage
team11_agreement_id VARCHAR(20)
team11_coverage_id VARCHAR(20)
Indexes
team11_premium_payment
team11_premium_payment_id VARCHAR(20)
team11_cust_id INT
team11_policy_number VARCHAR(20)
team11_premium_payment_schedule TEXT
team11_premium_payment_amount INT
Indexes
team11_vehicle
team11_vehicle_id VARCHAR(20)
team11_cust_id INT
team11_vehicle_registration_number VARCHAR(20)
team11_vehicle_value INT
team11_vehicle_type VARCHAR(20)
team11_vehicle_number_of_seat INT
team11_vehicle_manufacturer VARCHAR(20)
team11_vehicle_number INT
Indexes
team11_product
team11_product_number VARCHAR(20)
team11_company_name VARCHAR(20)
team11_product_price INT
team11_product_type CHAR(15)
Indexes
team11_staff
team11_staff_id VARCHAR(20)
team11_company_name VARCHAR(20)
team11_staff_fname VARCHAR(20)
team11_staff_lname VARCHAR(20)
team11_staff_address TEXT
team11_staff_contact INT
team11_staff_gender CHAR(2)
team11_staff_marital_status CHAR(12)
team11_staff_nationality CHAR(15)
team11_staff_qualification VARCHAR(20)
team11_staff_allowance INT
team11_staff_pps_number INT
Indexes
team11_insurance_company
team11_company_name VARCHAR(50)
team11_company_address TEXT
team11_company_contact_number BIGINT
team11_company_fax BIGINT
team11_company_email VARCHAR(50)
team11_company_website VARCHAR(30)
Indexes
team11_incident_report
team11_incident_report_id VARCHAR(20)
team11_incident_id VARCHAR(20)
team11_cust_id INT
team11_incident_inspector VARCHAR(20)
team11_incident_cost INT
team11_incident_type CHAR(20)
team11_incident_report_description VARCHAR(20)
Indexes
team11_incident
team11_incident_id VARCHAR(20)
team11_incident_type VARCHAR(30)
team11_incident_date DATE
Indexes
team11_customer
team11_cust_id INT
team11_cust_fname VARCHAR(15)
team11_cust_lname VARCHAR(15)
team11_cust_dob DATE
team11_cust_gender CHAR(2)
team11_cust_address TEXT
team11_cust_number BIGINT
team11_cust_email VARCHAR(50)
team11_cust_passportnumber VARCHAR(20)
team11_cust_maritalstatus CHAR(10)
team11_cust_ppsnumber INT
Indexes
team11_department
team11_department_id VARCHAR(10)
team11_department_name VARCHAR(20)
team11_company_name VARCHAR(20)
team11_contact_information VARCHAR(20)
Indexes

Tables:

Customers Table:

Attribute	**Datatype**	**Key**
Customer_id	int	PK
Customer_fname	varchar(15)	
Customer_lname	varchar(15)	
Customer_dob	date	
Customer_gender	char(2)	
Customer_address	text	
Customer_number	bigint	
Customer_email	varchar(50)	
Customer_passportnumber	varchar(20)	
Customer_maritalstatus	char(10)	
Customer_ppsnumber	int	

Membership Table:

Attributes	**Datatype**	**Key**
Memnership_id	varchar(20)	PK
Customer_id	int	FK(Customer table)
Memnership_type	char(15)	
Organisation_contact	varchar(20)	

Vehicle table:

Attribute	**Datatype**	**Key**
Vehicle_id	varchar(20)	PK
Customer_id	int	FK(Customer table)
Vehicle_registration_number	varchar(20)	
Vehicle_value	int	
Vehicle_type	varchar(20)	
Vehicle_number_of_seat	int	
Vehicle_manufacturer	varchar(20)	
Vehicle_number	int	

Application Table:

Attribute	**Datatype**	**Key**
Application_id	varchar(20)	PK
Customer_id	int	FK(Customer table)
Vehicle_id	varchar(20)	FK(Vehicle table)
Application_status	char(12)	
Coverage	varchar(50)	

Quote table:

Attributes	**Datatype**	**Key**
Quote_id	varchar(20)	PK
Application_id	varchar(20)	FK(Application table)
Customer_id	int	FK(Customer table)

Issue_date	date	
Description	varchar(100)	
Coverage_level	varchar(20)	

Insurance Policy table:

Attribute	**Datatype**	**Key**
Agreement_id	varchar(20)	PK
Application_id	varchar(20)	FK(Application table)
Customer_id	int	FK(Customer table)
Policy_number	varchar(20)	
Start_date	date	
Expiry_date	date	

Premium payment:

Attributes	**Datatype**	**Key**
premium _paymet_id	varchar(20)	PK
cutomer_id	int	FK(Cutomer table)
policy_number	varchar(20)	
premium_payment_schedule	text	
premium_payment_amount	int	

Receipt:

Attributes	**Datatype**	**Key**
receipt_id	varcahar(20)	PK
time	date	

cost	bigint	
premium_payment_id	varchar(20)	FK(Premium_payment table)
customer_id	int	FK(Customer table)

Incident table:

Attributes	Datatype	Key
incident_id	varchar(20)	PK
incident_type	varchar(30)	
incident_date	date	

Claim table:

Attributes	Datatype	Key
claim_id	varchar(20)	PK
customer_id	int	FK(Customer table)
agreement_id	varchar(20)	FK(Insurance_policy table)
claim_amount	int	
incident_id	varchar(20)	FK(Incident table)
damage_type	varchar(20)	
date_of_claim	date	
claim_status	char(10)	

Claim Settlement table:

Attributes	Datatype	Key
claim_settlement_id	varchar(20)	PK
claim_id	varchar(20)	FK(Claim table)
customer_id	int	FK(Customer table)

vehicle_id	varchar(20)	FK(Vehicle table)
date_settled	date	
amount_paid	int	

Insurance Company table:

Attributes	Datatype	Key
company_name	varchar(50)	PK
company_address	text	
company_contact_number	bigint	
company_fax	bigint	
company_email	varchar(50)	
company_website	varchar(30)	

Department table:

Attributes	Datatype	Key
department_id	varchar(10)	PK
department_name	varchar(20)	
company_name	varchar(20)	FK(Insurance_company table)
contact_information	varchar(20)	

Staff table:

Attributes	Datatype	Key
staff_id	varchar(20)	PK
company_name	varchar(20)	FK(Insurance_company table)

staff_fname	varchar(20)	
staff_lname	varchar(20)	
staff_address	text	
staff_contact	int	
staff_gender	char(2)	
staff_marital_status	char(12)	
staff_nationality	char(15)	
staff_qualification	varchar(20)	
staff_allowance	int	
staff_pps_number	int	

Product table:

Attributes	**Datatype**	**Key**
product_number	varchar(20)	PK
company _name	varchar(20)	FK(Insurace_company table)
product_price	int	
product_type	char(15)	

Office table:

Attributes	**Datatype**	**Key**
office_name	varchar(50)	PK
department_id	varchar(20)	FK(Department table)
company_name	varchar(20)	FK(Insurance_company table)
office_leader	varchar(20)	
contact_information	varchar(20)	

address	text	
admin_cost	int	

Coverage table:

Attributes	Datatype	Key
coverage_id	varchar(20)	PK
coverage_amount	bigint	
coverage_type	char(10)	
product_id	varchar(20)	FK(Product table)
company_name	varchar(20)	FK(Insurance_company table)

Insurance Policy Coverage table:

Attributes	Datatype	Key
agreement_id	varchar(20)	PK FK(Insurance_policy table)
coverage_id	varchar(20)	FK(Coverage table)

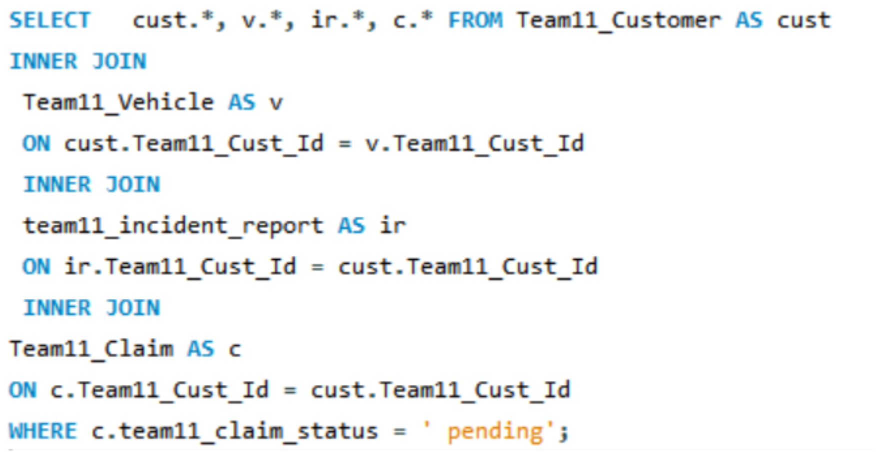
Incident Report table:

Attributes	Datatype	Key
incident_report_id	varchar(20)	PK
incident_id	varchar(20)	FK(Incident table)
customer_id	int	FK(Customer table)
incident_inspector	varchar(20)	
incident_cost	int	
incident_type	char(20)	

incident_report_description	varchar(20)	

Queries:

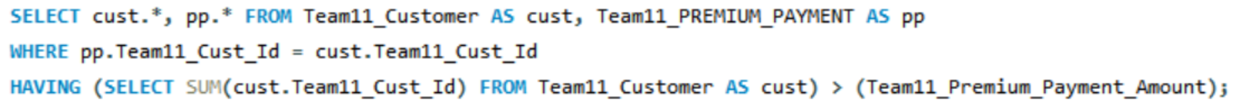
1. Retrieve Customer and Vehicle details who has been involved in an incident and claim status is pending – Customer, vehicle, claim status, incident



```
SELECT   cust.*, v.*, ir.*, c.* FROM Team11_Customer AS cust
INNER JOIN
 Team11_Vehicle AS v
 ON cust.Team11_Cust_Id = v.Team11_Cust_Id
 INNER JOIN
 team11_incident_report AS ir
 ON ir.Team11_Cust_Id = cust.Team11_Cust_Id
 INNER JOIN
Team11_Claim AS c
ON c.Team11_Cust_Id = cust.Team11_Cust_Id
WHERE c.team11_claim_status = ' pending';
```

In this query we are supposed to select the customer and vehicle details who has been involved in any kind of incident and has not claimed their claim amount. First we selected the required tables from the database. We then used inner join to attach Vehicle as v, Incident report as ir, Claim as c using foreign key constraints specified in the table. We used the condition WHERE c.team11_claim_status = ' pending' to check the claim status.

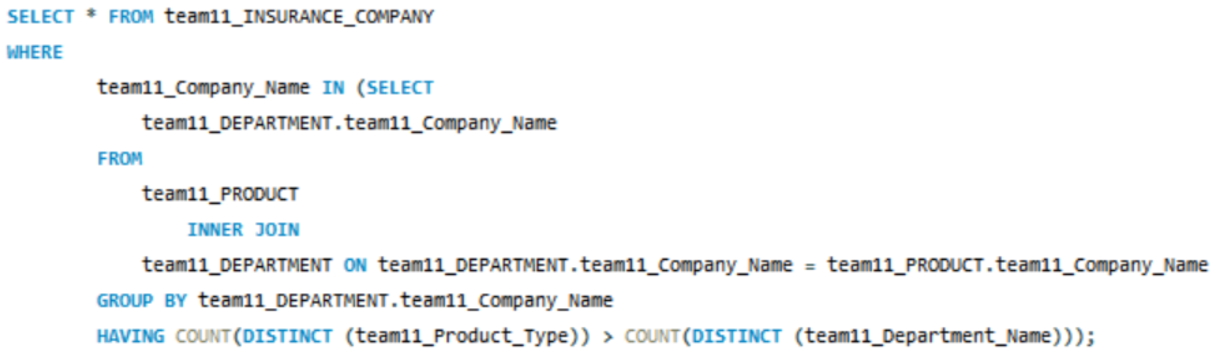
2. Retrieve customer details who has premium payment amount greater than the sum of all the customerIds in the database – premium payment, customer



```
SELECT cust.*, pp.* FROM Team11_Customer AS cust, Team11_PREMIUM_PAYMENT AS pp
WHERE pp.Team11_Cust_Id = cust.Team11_Cust_Id
HAVING (SELECT SUM(cust.Team11_Cust_Id) FROM Team11_Customer AS cust) > (Team11_Premium_Payment_Amount);
```

In this query we have to select the customer details whose premium payment amount is greater than the sum of all the customer_id in the database. We select the customer and premium payment tables using foreign key constraints. The where condition is used in which it is specified that the sum of all cutomer ids should be greater than the premium payment amount int the table.

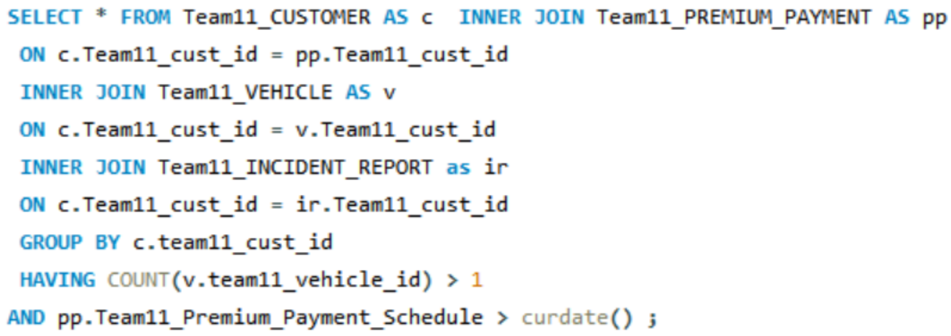
3. Retrieve Company details whose number of products is greater than departments, where the departments are located in more than one location—company, product, departments, office.



```
SELECT * FROM team11_INSURANCE_COMPANY
WHERE
        team11_Company_Name IN (SELECT
            team11_DEPARTMENT.team11_Company_Name
        FROM
            team11_PRODUCT
                INNER JOIN
            team11_DEPARTMENT ON team11_DEPARTMENT.team11_Company_Name = team11_PRODUCT.team11_Company_Name
        GROUP BY team11_DEPARTMENT.team11_Company_Name
        HAVING COUNT(DISTINCT (team11_Product_Type)) > COUNT(DISTINCT (team11_Department_Name)));
```

In this query we have retrieve customer details of which the departments is lesser than the number of products. First select columns from insurance company. We used the where clause with the condition of selecting company_name in which we select departmanent. company_name from product table. Use inner join for foreign key constraint. Use group by clause for company_name in department table having number of distinct product_type greater than the count of distinct department_name.

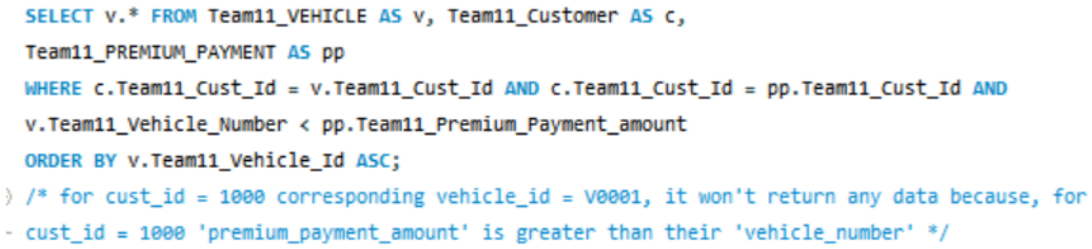
4. Select Customers who have more than one Vehicle, where the premium for one of the Vehicles is not paid and it is involved in accident.



```
SELECT * FROM Team11_CUSTOMER AS c  INNER JOIN Team11_PREMIUM_PAYMENT AS pp
 ON c.Team11_cust_id = pp.Team11_cust_id
 INNER JOIN Team11_VEHICLE AS v
 ON c.Team11_cust_id = v.Team11_cust_id
 INNER JOIN Team11_INCIDENT_REPORT as ir
 ON c.Team11_cust_id = ir.Team11_cust_id
 GROUP BY c.team11_cust_id
 HAVING COUNT(v.team11_vehicle_id) > 1
AND pp.Team11_Premium_Payment_Schedule > curdate() ;
```

In this query we have to select customers with more than one vehicle and the premium for one of it is not paid which was involved in an accident. First we select the customer and inner join the premium payment table, vehicle table and incident report table on customer id as foreign key constraint. In the body we used group clause for customer_id having more than one vehicle_id and premium_payment_schedule is greater than current date in the database.

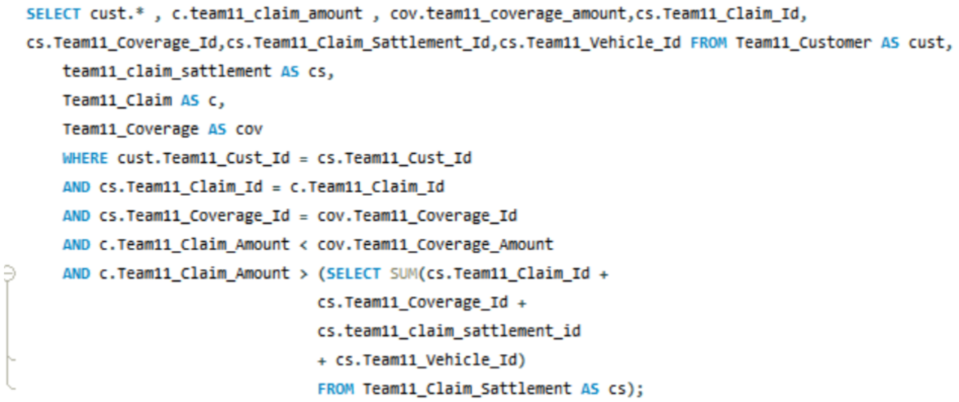
5. Select all vehicles which have premium more than its vehicle number.



```
SELECT v.* FROM Team11_VEHICLE AS v, Team11_Customer AS c,
Team11_PREMIUM_PAYMENT AS pp
WHERE c.Team11_Cust_Id = v.Team11_Cust_Id AND c.Team11_Cust_Id = pp.Team11_Cust_Id AND
v.Team11_Vehicle_Number < pp.Team11_Premium_Payment_amount
ORDER BY v.Team11_Vehicle_Id ASC;
/* for cust_id = 1000 corresponding vehicle_id = V0001, it won't return any data because, for
cust_id = 1000 'premium_payment_amount' is greater than their 'vehicle_number' */
```

In this Query, we have to select all the vehicle details whose premium is more than it's vehicle number. First, we selected vehicle table, customer table and premium payment table using customer_id foreign key constraint. In the where clause we used the condition such that vehicle_number is lesser than premium_payment_amount which is ordered by vehicle_id in ascending order.

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.team11_claim_amount , cov.team11_coverage_amount,cs.Team11_Claim_Id,
cs.Team11_Coverage_Id,cs.Team11_Claim_Sattlement_Id,cs.Team11_Vehicle_Id FROM Team11_Customer AS cust,
    team11_claim_sattlement AS cs,
    Team11_Claim AS c,
    Team11_Coverage AS cov
    WHERE cust.Team11_Cust_Id = cs.Team11_Cust_Id
    AND cs.Team11_Claim_Id = c.Team11_Claim_Id
    AND cs.Team11_Coverage_Id = cov.Team11_Coverage_Id
    AND c.Team11_Claim_Amount < cov.Team11_Coverage_Amount
    AND c.Team11_Claim_Amount > (SELECT SUM(cs.Team11_Claim_Id +
                                 cs.Team11_Coverage_Id +
                                 cs.team11_claim_sattlement_id
                                 + cs.Team11_Vehicle_Id)
                                 FROM Team11_Claim_Sattlement AS cs);
```

In this query, the customer details should be retrieved whose claim amount is lesser than the coverage amount. It must also include the query which says that claim amount is greater than sum of claim settlement id, vehicle id, customer id and claim id. First, we start by selecting the tables claim settlement table from which the columns claim_amount, vehicle_id, coverage_amount, coverage_id, claim_settlement_id, and claim_id. These tables are connected by foreign key customer_id. The condition specified is that the claim_amount is lesser than coverage_amount. In the subquery, the sum of claim_id, coverage_id, claim_settlement_id and vehicle_id is selected. The main query also contains the condition that claim_amount is greater than the subquery.

Conclusion

The tables were created successfully. Some challenges occurred in regards to foreign key constraints. The alter table was not used for foreign key constraints. The data was successfully inserted and some update was done. The queries were completed smoothly.