DATABASE DESIGN FOR A VEHICLE INSURANCE COMPANY

CS301 , DBMS - V SEMESTER.

Group Members :

Sumith Sai Budde , 18bcs101

Rishabh Dwivedi , 18bcs078

Shantanu Tripathi , 18bcs116

Tushar Yadav , 18bcs106

Syed Sufyan Ahmed , 18bcs103

Ankit Rajpoot , 18bcs007

Prabhdeep Singh , 18bcs068

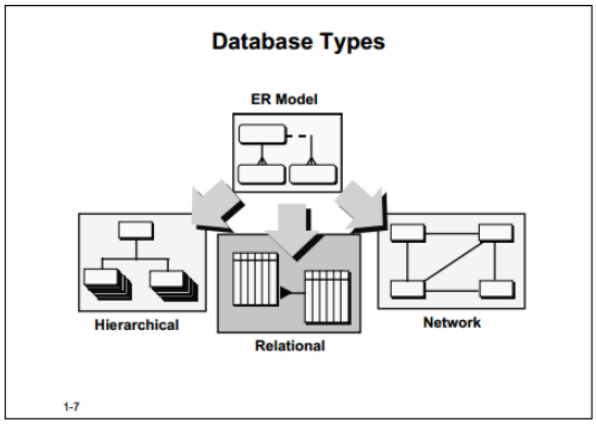
Sachin Verma , 18bcs085

Project title : A database for a Vehicle Insurance Company

Part C: Physical Data Model and Database Design

Introduction

Physical data model is used to implement into different technical software and hardware environments that is due to current state of technology and is changing as technologies change.



Normalization :

It is a relationship database concept and is done in process of building ER. If the correct entity model is being build will conform to the rules of normalization. Each rule has corresponding data model interpretation, which can be used to validate placement of attributes in ER model.

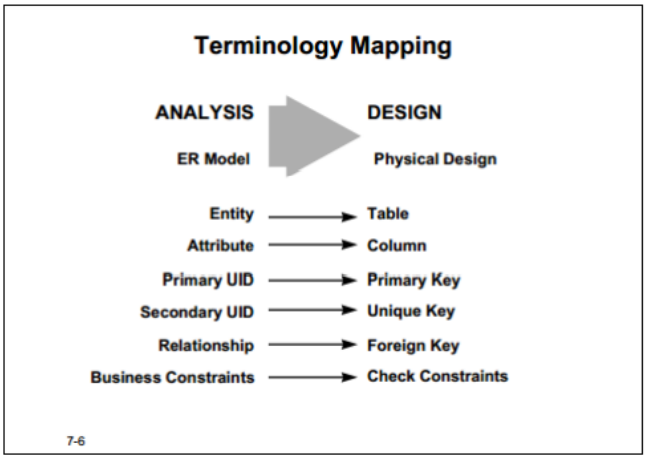
Normalization rules

First Normal Form (1NF):	All attributes are single-valued.
Second Normal Form (2NF):	An attribute must be dependent upon entity's entire unique identifier
Third Normal Form (3NF):	No non key attributes are dependent on another non key attribute.

Normalization of vehicle insurance company :

Normal form	Table
First normal form (1NF)	
Second normal form (2NF)	DEPARTMENT OFFICE VEHICLE NOK INSURANCE_POLICY CLAIM CLAIM_SETTLEMENT PREMIUM_PAYMENT QUOTE INCIDENT_REPORT POLICY_RENEWABLE
Third normal form (3NF)	CUSTOMER RECEIPT APPLICATION STAFF INSURANCE_COMPANY MEMBERSHIP PRODUCT COVERAGE INCIDENT

TERMINOLOGY



PDM for vehicle insurance service

1. Tables
2. Views
3. Stored Procedures
4. User Defined Functions

1. Tables :

Table No.	Code
1	CREATE TABLE `t7_incident` (`T7_Incident_Id` varchar(20) NOT NULL, `T7_Incident_Type` varchar(50) DEFAULT NULL, `T7_Incident_Date` date NOT NULL, `T7_Incident_Description` varchar(1000) DEFAULT NULL, PRIMARY KEY (`T7_Incident_Id`), UNIQUE KEY `XPKINCIDENT_17` (`T7_Incident_Id`));
2	CREATE TABLE `t7_customer` (`T7_CUST_Id` varchar(15) NOT NULL, `T7_CUST_FName` varchar(15) NOT NULL, `T7_CUST_LName` varchar(15) NOT NULL, `T7_CUST_DOB` date NOT NULL, `T7_CUST_Gender` char(2) NOT NULL, `T7_CUST_Address` varchar(35) NOT NULL, `T7_CUST_MOB_Number` bigint NOT NULL UNIQUE , `T7_CUST_Email` varchar(25) DEFAULT NULL UNIQUE, `T7_CUST_Passport_Number` varchar(20) DEFAULT NULL UNIQUE, `T7_CUST_Marital_Status` char(12) DEFAULT NULL, `T7_CUST_PPS_Number` int DEFAULT NULL UNIQUE, PRIMARY KEY (`T7_CUST_Id`), UNIQUE KEY `XPKCUSTOMER_1` (`T7_CUST_Id`));
3	CREATE TABLE `t7_incident_report` (`T7_Incident_Report_Id` varchar(20) NOT NULL, `T7_Incident_Type` char(50) DEFAULT NULL, `T7_Incident_Inspector` varchar(20) DEFAULT NULL, `T7_Incident_Cost` int DEFAULT NULL, `T7_Incident_Report_Description` varchar(1000) DEFAULT NULL, `T7_Incident_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Incident_Report_Id`,`T7_Incident_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKINCIDENT_REPORT_18` (`T7_Incident_Report_Id`,`T7_Incident_Id`,`T7_Cust_Id`), KEY `R_83` (`T7_Incident_Id`), KEY `R_86` (`T7_Cust_Id`), CONSTRAINT `R_83` FOREIGN KEY (`T7_Incident_Id`) REFERENCES `t7_incident` (`T7_Incident_Id`) on delete cascade on update cascade, CONSTRAINT `R_86` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on update cascade on delete cascade);

4	CREATE TABLE `t7_insurance_company` (`T7_Company_Name` varchar(70) NOT NULL, `T7_Company_Address` varchar(400) DEFAULT NULL, `T7_Company_Contact_Number` bigint DEFAULT NULL UNIQUE, `T7_Company_Fax` bigint DEFAULT NULL UNIQUE, `T7_Company_Email` varchar(50) DEFAULT NULL UNIQUE, `T7_Company_Website` varchar(50) DEFAULT NULL UNIQUE, `T7_Company_Location` varchar(50) DEFAULT NULL, PRIMARY KEY (`T7_Company_Name`), UNIQUE KEY `XPKINSURANCE_COMPANY_15` (`T7_Company_Name`));
5	CREATE TABLE `t7_department` (`T7_Department_Name` varchar(50) NOT NULL, `T7_Department_ID` varchar(50) NOT NULL, `T7_Department_Staff` varchar(50) DEFAULT NULL, `T7_Company_Name` varchar(100) NOT NULL, PRIMARY KEY (`T7_Department_Name`,`T7_Department_ID`,`T7_Company_Name`), UNIQUE KEY `XPKDEPARTMENT` (`T7_Department_Name`,`T7_Department_ID`,`T7_Company_Name`), KEY `R_56` (`T7_Company_Name`), CONSTRAINT `R_56` FOREIGN KEY (`T7_Company_Name`) REFERENCES `t7_insurance_company` (`T7_Company_Name`) on update cascade);
6	CREATE TABLE `t7_vehicle` (`T7_Vehicle_Id` varchar(20) NOT NULL, `T7_Policy_Number` varchar(20) DEFAULT NULL, `T7_Vehicle_Registration_Number` varchar(20) NOT NULL UNIQUE, `T7_Vehicle_Value` bigint DEFAULT NULL, `T7_Vehicle_Type` varchar(20) NOT NULL, `T7_Vehicle_Size` int DEFAULT NULL, `T7_Vehicle_Number_Of_Seat` int DEFAULT NULL, `T7_Vehicle_Manufacturer` varchar(20) DEFAULT NULL, `T7_Vehicle_Engine_Number` int DEFAULT NULL UNIQUE, `T7_Vehicle_Chasis_Number` int DEFAULT NULL UNIQUE, `T7_Vehicle_Number` int DEFAULT NULL UNIQUE, `T7_Vehicle_Model_Number` varchar(20) DEFAULT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Vehicle_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKVEHICLE_6` (`T7_Vehicle_Id`,`T7_Cust_Id`), KEY `R_92` (`T7_Cust_Id`), CONSTRAINT `R_92` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on delete cascade) ;
7	CREATE TABLE `t7_premium_payment` (`T7_Premium_Payment_Id` varchar(20) NOT NULL, `T7_Policy_Number` varchar(20) NOT NULL, `T7_Premium_Payment_Amount` int NOT NULL, `T7_Premium_Payment_Schedule` date NOT NULL, `T7_Receipt_Id` varchar(20) NOT NULL UNIQUE, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Premium_Payment_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKPREMIUM_PAYMENT_5` (`T7_Premium_Payment_Id`,`T7_Cust_Id`), KEY `R_85` (`T7_Cust_Id`), CONSTRAINT `R_85` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on delete cascade);

8	CREATE TABLE `t7_receipt` (`T7_Receipt_Id` varchar(20) NOT NULL, `T7_Gen_Time` date NOT NULL, `T7_Cost` int NOT NULL, `T7_Premium_Payment_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Receipt_Id`,`T7_Premium_Payment_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKRECEIPT_21` (`T7_Receipt_Id`,`T7_Premium_Payment_Id`,`T7_Cust_Id`), KEY `R_84` (`T7_Premium_Payment_Id`,`T7_Cust_Id`), CONSTRAINT `R_84` FOREIGN KEY (`T7_Premium_Payment_Id`, `T7_Cust_Id`) REFERENCES `t7_premium_payment` (`T7_Premium_Payment_Id`, `T7_Cust_Id`) on update cascade on delete restrict);
9	CREATE TABLE `t7_application` (`T7_Application_Id` varchar(20) NOT NULL, `T7_Vehicle_Id` varchar(20) NOT NULL, `T7_Application_Status` char(8) NOT NULL, `T7_Coverage` varchar(50) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Application_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKAPPLICATION_2` (`T7_Application_Id`,`T7_Cust_Id`), KEY `R_93` (`T7_Cust_Id`), CONSTRAINT `R_93` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on delete cascade on update cascade);
10	CREATE TABLE `t7_insurance_policy` (`T7_Agreement_id` varchar(20) NOT NULL, `T7_Department_Name` varchar(20) DEFAULT NULL, `T7_Policy_Number` varchar(20) DEFAULT NULL, `T7_Start_Date` date DEFAULT NULL, `T7_Expiry_Date` date DEFAULT NULL, `T7_Term_Condition_Description` varchar(400) DEFAULT NULL, `T7_Application_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKINSURANCE_POLICY_4` (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), KEY `R_95` (`T7_Application_Id`,`T7_Cust_Id`), CONSTRAINT `R_95` FOREIGN KEY (`T7_Application_Id`, `T7_Cust_Id`) REFERENCES `t7_application` (`T7_Application_Id`, `T7_Cust_Id`) on update cascade on delete restrict);
11	CREATE TABLE `t7_policy_renewable` (`T7_Policy_Renewable_Id` varchar(20) NOT NULL, `T7_Date_Of_Renewal` date NOT NULL, `T7_Type_Of_Renewal` char(15) NOT NULL, `T7_Agreement_id` varchar(20) NOT NULL, `T7_Application_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Policy_Renewable_Id`,`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKPOLICY_RENEWABLE_16` (`T7_Policy_Renewable_Id`,`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), KEY `R_101` (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), CONSTRAINT `R_101` FOREIGN KEY (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) REFERENCES `t7_insurance_policy` (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) on delete cascade on update restrict);

12	CREATE TABLE `t7_membership` (`T7_Membership_Id` varchar(20) NOT NULL, `T7_Membership_Type` char(15) NOT NULL, `T7_Organisation_Contact` varchar(20) DEFAULT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Membership_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKMEMBERSHIP_12` (`T7_Membership_Id`,`T7_Cust_Id`), KEY `R_91` (`T7_Cust_Id`), CONSTRAINT `R_91` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on delete cascade on update cascade);
13	CREATE TABLE `t7_quote` (`T7_Quote_Id` varchar(20) NOT NULL, `T7_Issue_Date` date NOT NULL, `T7_Valid_From_Date` date NOT NULL, `T7_Valid_Till_Date` date NOT NULL, `T7_Description` varchar(100) DEFAULT NULL, `T7_Product_Id` varchar(20) NOT NULL, `T7_Coverage_Level` varchar(20) NOT NULL, `T7_Application_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Quote_Id`,`T7_Application_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKQUOTE_3` (`T7_Quote_Id`,`T7_Application_Id`,`T7_Cust_Id`), KEY `R_94` (`T7_Application_Id`,`T7_Cust_Id`), CONSTRAINT `R_94` FOREIGN KEY (`T7_Application_Id`, `T7_Cust_Id`) REFERENCES `t7_application` (`T7_Application_Id`, `T7_Cust_Id`) on delete cascade on update restrict);
14	CREATE TABLE `t7_staff` (`T7_Staff_Id` varchar(200) NOT NULL, `T7_Staff_Fname` varchar(100) DEFAULT NULL, `T7_Staff_LName` varchar(100) DEFAULT NULL, `T7_Staff_Address` varchar(200) DEFAULT NULL, `T7_Staff_Contact` bigint DEFAULT NULL UNIQUE, `T7_Staff_Gender` char(2) DEFAULT NULL, `T7_Staff_Marital_Status` char(8) DEFAULT NULL, `T7_Staff_Nationality` char(15) DEFAULT NULL, `T7_Staff_Qualification` varchar(20) DEFAULT NULL, `T7_Staff_Allowance` bigint DEFAULT NULL, `T7_Staff_PPS_Number` bigint DEFAULT NULL UNIQUE, `T7_Company_Name` varchar(400) NOT NULL, PRIMARY KEY (`T7_Staff_Id`,`T7_Company_Name`), UNIQUE KEY `XPKSTAFF_9` (`T7_Staff_Id`,`T7_Company_Name`), KEY `R_105` (`T7_Company_Name`), CONSTRAINT `R_105` FOREIGN KEY (`T7_Company_Name`) REFERENCES `t7_insurance_company` (`T7_Company_Name`) on update cascade on delete cascade);
15	CREATE TABLE `t7_nok` (`T7_Nok_Id` varchar(20) NOT NULL, `T7_Nok_Name` varchar(20) DEFAULT NULL, `T7_Nok_Address` varchar(200) DEFAULT NULL, `T7_Nok_Phone_Number` bigint DEFAULT NULL, `T7_Nok_Gender` char(10) DEFAULT NULL, `T7_Nok_Marital_Status` char(8) DEFAULT NULL,

	`T7_Agreement_id` varchar(20) NOT NULL, `T7_Application_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Nok_Id`,`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKNOK_14` (`T7_Nok_Id`,`T7_Agreement_id`,`T7_Application_Id`), KEY `R_99` (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`), CONSTRAINT `R_99` FOREIGN KEY (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) REFERENCES `t7_insurance_policy` (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) on delete cascade);
16	CREATE TABLE `t7_product` (`T7_Product_Number` varchar(20) NOT NULL, `T7_Product_Price` int DEFAULT NULL, `T7_Product_Type` char(40) DEFAULT NULL, `T7_Company_Name` varchar(200) NOT NULL, PRIMARY KEY (`T7_Product_Number`,`T7_Company_Name`), UNIQUE KEY `XPKPRODUCT_20` (`T7_Product_Number`,`T7_Company_Name`), KEY `R_107` (`T7_Company_Name`), CONSTRAINT `R_107` FOREIGN KEY (`T7_Company_Name`) REFERENCES `t7_insurance_company` (`T7_Company_Name`) on delete cascade on update restrict);
17	CREATE TABLE `t7_office` (`T7_Office_Name` varchar(200) NOT NULL, `T7_Office_Leader` varchar(200) NOT NULL, `T7_Contact_Information` varchar(200) NOT NULL, `T7_Address` varchar(200) NOT NULL, `T7_Admin_Cost` int DEFAULT NULL, `T7_Staff` varchar(50) DEFAULT NULL, `T7_Department_Name` varchar(200) NOT NULL, `T7_Department_ID` varchar(200) NOT NULL, `T7_Company_Name` varchar(200) NOT NULL, PRIMARY KEY (`T7_Office_Name`,`T7_Department_Name`,`T7_Company_Name`), UNIQUE KEY `XPKOFFICE_11` (`T7_Office_Name`,`T7_Department_Name`,`T7_Company_Name`), KEY `R_104` (`T7_Department_Name`,`T7_Department_ID`,`T7_Company_Name`), CONSTRAINT `R_104` FOREIGN KEY (`T7_Department_Name`, `T7_Department_ID`, `T7_Company_Name`) REFERENCES `t7_department` (`T7_Department_Name`, `T7_Department_ID`, `T7_Company_Name`) on delete cascade on update cascade);
18	CREATE TABLE `t7_coverage` (`T7_Coverage_Id` varchar(20) NOT NULL, `T7_Coverage_Amount` int NOT NULL, `T7_Coverage_Type` char(30) NOT NULL, `T7_Coverage_Level` char(30) NOT NULL, `T7_Product_Id` varchar(20) NOT NULL, `T7_Coverage_Description` varchar(400) DEFAULT NULL, `T7_Coverage_Terms` varchar(200) DEFAULT NULL, `T7_Company_Name` varchar(400) NOT NULL, PRIMARY KEY (`T7_Coverage_Id`,`T7_Company_Name`), UNIQUE KEY `XPKCOVERAGE_19` (`T7_Coverage_Id`,`T7_Company_Name`), KEY `R_102` (`T7_Company_Name`), CONSTRAINT `R_102` FOREIGN KEY (`T7_Company_Name`) REFERENCES `t7_insurance_company` (`T7_Company_Name`) on update cascade on delete cascade);

19	CREATE TABLE `t7_insurance_policy_coverage` (`T7_Agreement_id` varchar(20) NOT NULL, `T7_Application_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, `T7_Coverage_Id` varchar(20) NOT NULL, `T7_Company_Name` varchar(200) NOT NULL, PRIMARY KEY (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`,`T7_Coverage_Id`,`T7_Company_Name`), UNIQUE KEY `XPKINSURANCE_POLICY_4_COVERAGE` (`T7_Agreement_id`,`T7_Application_Id`,`T7_Cust_Id`,`T7_Coverage_Id`,`T7_Company_Name`), KEY `R_98` (`T7_Coverage_Id`,`T7_Company_Name`), CONSTRAINT `R_97` FOREIGN KEY (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) REFERENCES `t7_insurance_policy` (`T7_Agreement_id`, `T7_Application_Id`, `T7_Cust_Id`) on update cascade, CONSTRAINT `R_98` FOREIGN KEY (`T7_Coverage_Id`, `T7_Company_Name`) REFERENCES `t7_coverage` (`T7_Coverage_Id`, `T7_Company_Name`) on delete cascade);
20	CREATE TABLE `t7_claim` (`T7_Claim_Id` varchar(20) NOT NULL, `T7_Agreement_Id` varchar(20) NOT NULL, `T7_Claim_Amount` int NOT NULL, `T7_Incident_id` varchar(20) DEFAULT NULL, `T7_Damage_Type` varchar(20) DEFAULT NULL, `T7_Date_Of_Claim` date NOT NULL, `T7_Claim_Status` char(10) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Claim_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKCLAIM_7` (`T7_Claim_Id`,`T7_Cust_Id`), KEY `R_88` (`T7_Cust_Id`), CONSTRAINT `R_88` FOREIGN KEY (`T7_Cust_Id`) REFERENCES `t7_customer` (`T7_CUST_Id`) on delete cascade on update cascade);
21	CREATE TABLE `t7_claim_settlement` (`T7_Claim_Settlement_Id` varchar(20) NOT NULL, `T7_Vehicle_Id` varchar(20) NOT NULL, `T7_Date_Settled` date NOT NULL, `T7_Amount_Paid` int NOT NULL, `T7_Coverage_Id` varchar(20) NOT NULL, `T7_Claim_Id` varchar(20) NOT NULL, `T7_Cust_Id` varchar(20) NOT NULL, PRIMARY KEY (`T7_Claim_Settlement_Id`,`T7_Claim_Id`,`T7_Cust_Id`), UNIQUE KEY `XPKCLAIM_SETTLEMENT_8` (`T7_Claim_Settlement_Id`,`T7_Claim_Id`,`T7_Cust_Id`), KEY `R_90` (`T7_Claim_Id`,`T7_Cust_Id`), CONSTRAINT `R_90` FOREIGN KEY (`T7_Claim_Id`, `T7_Cust_Id`) REFERENCES `t7_claim` (`T7_Claim_Id`, `T7_Cust_Id`) on delete restrict on update cascade);

2. Views :

What is a view :

- A view is a virtual table based on the result set of an sql statement. They minimise the storage space required.
- Views make queries faster to write.
- They provide an additional level of table security by restricting access to a predetermined set of rows and columns of a table.
- Different views can be created on the same table for different users.

view1 : a view that selects customers having coverage amount greater than sum of all the coverage ids in the database.

view2 : a view that selects customer details , membership details and vehicle details in which the customers are having either paid or Honorary membership in any organisation.

Productpriceaboveavg : a view that selects every product in the Products table with a price higher than the average product price.

Query1View : a view that retrieves customer and vehicle details who has been involved in an incident and claim status is pending

Maharashtra_Vehicles : a view that selects all the vehicles from Maharashtra state based on their Vehicle_Registration_Number using regex.

[*All the views can be found in views.sql file]

View No.	Code
1	create view view1 as select * from T7_CUSTOMER where T7_CUSTOMER.T7_Cust_Id in (select T7_Cust_Id from T7_Insurance_Policy_Coverage where T7_Insurance_Policy_Coverage.T7_Coverage_Id in (select T7_Coverage_Id from T7_COVERAGE where T7_COVERAGE.T7_Coverage_Amount > (select sum(T7_Coverage_Id) from T7_COVERAGE)));
2	create view view2 as select t7_membership.T7_Cust_Id,Concat(T7_Cust_FName,T7_Cust_LName) as T7_Cust_Name,T7_CUST_MOB_Number,T7_Organisation_Contact,T7_Membership_Id,T7_Membership_Type,T7_Vehicle_Id,T7_Vehicle_Registration_Number from T7_CUSTOMER inner join T7_MEMBERSHIP on T7_MEMBERSHIP.T7_Cust_Id = T7_CUSTOMER.T7_Cust_Id inner join T7_VEHICLE on T7_MEMBERSHIP.T7_Cust_Id = T7_VEHICLE.T7_Cust_Id where T7_MEMBERSHIP.T7_Membership_Type = 'Paid' or T7_MEMBERSHIP.T7_Membership_Type = 'Honorary' ;
3	create view Productpriceaboveavg as select T7_Product_Number,T7_Company_Name from T7_PRODUCT where T7_Product_Price > (select avg(T7_Product_Price) from T7_PRODUCT) ;

4	create view Query1View as select distinct T1.T7_Cust_Id,concat(T1.T7_Cust_FName,T1.T7_Cust_LName) as T7_Cust_Name,T1.T7_Cust_MOB_Number,T2.T7_Vehicle_Id,T2.T7_Vehicle_Registration_Number,T2.T7_Vehicle_Manufacturer,T2.T7_Vehicle_Model_Number from T7_CUSTOMER as T1 INNER JOIN T7_VEHICLE as T2 on T2.T7_Cust_id = T1.T7_Cust_Id where T2.T7_Vehicle_Id not in (select T7_Vehicle_Id from T7_CLAIM_SETTLEMENT) and T1.T7_Cust_Id = ANY (select T7_Cust_Id from T7_CLAIM where T7_Incident_Id is not null);
5	create view Maharashtra_Vehicles as select T7_Vehicle_Id,T7_Vehicle_Registration_Number,T7_Vehicle_Type,T7_Vehicle_Manufacturer,T7_Vehicle_Engine_Number,T7_Cust_Id from T7_VEHICLE where T7_Vehicle_Registration_Number like 'MH%' ;

3. Stored Procedures

- Stored Procedures are a set of declarative statements which can be stored in the database catalogue.

- They yield higher productivity .

- They reduce the traffic between the database and the application since the lengthy statements are already fed into the database and need not be sent again and again.

- They are pre compiled.

- They add code reusability feature to the database.

- We can also pass parameters to stored procedures and make them act based on the parameters passed.

[*All stored procedures can be found in stored_procedures.sql file]

How to call the stored procs : call storedpoc_name;

Example : call query1();

Stored_proc No.	Code
1	DELIMITER // create procedure query1() BEGIN select distinct T1.T7_Cust_Id,concat(T1.T7_Cust_FName,T1.T7_Cust_LName) as T7_Cust_Name,T1.T7_Cust_MOB_Number,T2.T7_Vehicle_Id,T2.T7_Vehicle_Registration_Number,T2.T7_Vehicle_Manufacturer,T2.T7_Vehicle_Model_Number from T7_CUSTOMER as T1 INNER JOIN T7_VEHICLE as T2 on T2.T7_Cust_id = T1.T7_Cust_Id where T2.T7_Vehicle_Id not in (select T7_Vehicle_Id from T7_CLAIM_SETTLEMENT) and T1.T7_Cust_Id = ANY (select T7_Cust_Id from T7_CLAIM where T7_Incident_Id is not null); END //

	DELIMITER ;

2	DELIMITER //
	create procedure query2()
	BEGIN
	select * from T7_CUSTOMER where T7_CUSTOMER.T7_Cust_Id in (select T7_Cust_Id from T7_PREMIUM_PAYMENT where T7_PREMIUM_PAYMENT.T7_Premium_Payment_Amount > (select sum(T7_Cust_Id) from T7_CUSTOMER));
	END //
	DELIMITER ;

3	DELIMITER //
	create procedure query3()
	BEGIN
	select * from T7_INSURANCE_COMPANY where T7_Company_Name in (select T7_Company_Name from T7_OFFICE group by T7_Company_Name having T7_Company_Name in (select T7_OFFICE.T7_Company_Name from T7_PRODUCT inner join T7_OFFICE on T7_OFFICE.T7_Company_Name=T7_PRODUCT.T7_Company_Name group by T7_OFFICE.T7_Company_Name having Count(distinct(T7_Product_Number))>Count(distinct(T7_Office_Name))) and count(distinct T7_Address)>1);
	END //
	DELIMITER ;

4	DELIMITER //
	create procedure query4()
	BEGIN
	select * from T7_CUSTOMER where T7_CUSTOMER.T7_Cust_id in (select T7_Cust_Id from T7_VEHICLE where T7_VEHICLE.T7_Policy_Number not in (select T7_Policy_Number from T7_PREMIUM_PAYMENT) and T7_VEHICLE.T7_Cust_Id in (select T7_Cust_Id from T7_VEHICLE GROUP BY T7_VEHICLE.T7_Cust_Id having count(T7_VEHICLE.T7_Cust_Id) > 1) and T7_VEHICLE.T7_Cust_Id in (select T7_Cust_Id from T7_INCIDENT_REPORT where T7_Incident_Type = 'Accident'));
	END //
	DELIMITER ;

5	DELIMITER //
	create procedure query5()
	BEGIN
	select * from T7_Vehicle,T7_Premium_Payment where T7_Vehicle.T7_cust_id=T7_Premium_Payment.T7_cust_id and T7_Premium_Payment.T7_Premium_Payment_Amount>T7_Vehicle.T7_Vehicle_Number;
	END //
	DELIMITER ;

6	DELIMITER // create procedure query6() BEGIN

	select * from T7_CUSTOMER where T7_CUSTOMER.T7_Cust_Id in (select T7_Cust_Id from T7_CLAIM where T7_CLAIM.T7_Claim_Amount < (select T7_COVERAGE.T7_Coverage_Amount from T7_COVERAGE where T7_CLAIM.T7_Agreement_id in (select T7_INSURANCE_POLICY_COVERAGE.T7_Agreement_id from T7_INSURANCE_POLICY_COVERAGE where T7_INSURANCE_POLICY_COVERAGE.T7_Coverage_Id = T7_COVERAGE.T7_Coverage_Id)) and T7_CLAIM.T7_Claim_Amount > (select T7_Claim_Settlement_Id+T7_Claim_Id+T7_Vehicle_Id+T7_Cust_Id from T7_CLAIM_SETTLEMENT where T7_CLAIM_SETTLEMENT.T7_Claim_id = T7_CLAIM.T7_Claim_Id)); END // DELIMITER ;

7	DELIMITER //
	create procedure selectcustomersonmaritalstatus(M_Status varchar(20))
	BEGIN
	select * from T7_CUSTOMER where T7_CUST_Marital_Status = M_Status;
	END //
	DELIMITER ;

8	DELIMITER //
	create procedure query6_udf1()
	BEGIN
	select * from T7_CUSTOMER where T7_CUSTOMER.T7_Cust_Id in (select T7_Cust_Id from T7_CLAIM where
	T7_CLAIM.T7_Claim_Amount < (select T7_COVERAGE.T7_Coverage_Amount from T7_COVERAGE where T7_CLAIM.T7_Agreement_id in
	(select T7_INSURANCE_POLICY_COVERAGE.T7_Agreement_id from T7_INSURANCE_POLICY_COVERAGE where
	T7_INSURANCE_POLICY_COVERAGE.T7_Coverage_Id = T7_COVERAGE.T7_Coverage_Id)) and
	T7_CLAIM.T7_Claim_Amount > (select
	CalcSum(T7_Claim_Settlement_Id,T7_Claim_Id,T7_Vehicle_Id,T7_Cust_Id)
	from T7_CLAIM_SETTLEMENT where T7_CLAIM_SETTLEMENT.T7_Claim_id = T7_CLAIM.T7_Claim_Id));
	END //
	DELIMITER ;

4.User Defined Functions

- Functions are same as stored procedures except that they return a value.

- Functions improves the performance against running sql multiple times for a computation.

CalcSum() : calculates the sum of 4 numbers and returns back the sum.

Example : select CalcSum(4,5,6,7);

Prints out 22 on the console.

[* Functions can be found in functions.sql file]

Function No.	Code

1	DELIMITER //
	create function CalcSum(v1 int,v2 int,v3 int,v4 int)
	RETURNS int
	BEGIN
	declare calcsum int;
	set calcsum = v1 + v2 + v3 + v4 ;
	RETURN calcsum;
	END; //
	DELIMITER ;

Database Security

Database security is described by the following aspects：

Data independence

Data independence is an important method that ensures data security; it can be divided into logical independence and physical independence. Physical independence means applications and data are store independent of each other, data is managed by DBMS and application does not need to understand it, application just need processing the data when the physical storage of data changing application without changing.

Data Security

1. Isolation protect important files in database

2. Using authorization rules, such as access control method and accounts, passwords permissions control.

3. Data Encryption

Data Integrity

Data integrity includes:

1. Data validation: ensure clean, correct and useful data.

2. Data consistency: Different users are using the same data should be identical

3. Data correctness: The input value of the data should be consistent with data in database.

Concurrency Control

When multiple user concurrent access to data, the database will have multiple transactions simultaneously accessing the same data. If not controlled , the concurrent operation may cause

inconsistency to read and store data. Locking mechanism of the database can effectively protect the database and achieve concurrency control.

Recovery

When the database fails DBMS need to find faults and fix problems, thus preventing data corruption. Moreover database should regularly back up and establish a spare machine, this makes the database to be restored as quickly as possible from the fault.

Who has access to certain parts of database

A CRUD matrix is a table showing the functions in an application containing SQL statement affecting parts of a database. It is a great way to show us what kind interaction appears between user and tables in database.

Using four SQL statements:

- Create – INSERT – to store new data
- Read – SELECT – to retrieve data
- Update – UPDATE – to change or modify data
- Delete – DELETE – to delete or remove data

[*All the grants to users can be found in crud_grants.sql file]

MODULES / ENTITIES	Customer	Manger of insurance	Insurance agent	Accountant	HR department	Damage inspector	Database administrator	Finance department	
CUSTOMER	CR	R	CRUD	R	-	-	R	-	
APPLICATION	R	R	CRD	R	R	-	CRD	R	
QUOTE	R	R	CRUD	R	R	-	R	R	
INSURANCE_POLICY	R	R	CRUD	R	-	R	CRD	R	
PREMIUM_PAYMENT	-	-	CRUD	CRUD	-	-	RU	CRUD	
VEHICLE	-	R	CRUD	-	-	-	RU	-	
CLAIM	-	R	CRUD	CRD	-	-	RU	CRUD	
CLAIM_SETTLEMENT	R	CRUD	CR	-	-	-	RU	CRUD	
STAFF	-	CR	R	R	CRUD	-	RU	-	
DEPARTMENT	-	R	R	-	CRUD	-	RU	R	
OFFICE	R	R	R	R	CRUD	-	RU	R	
MEMBERSHIP	CR	R	CRD	-	-	-	RU	R	
VEHICLE_SERVICE	R	CRD	CRU	-	-	-	RU	R	
NOK	R	R	CRUD	-	-	-	RU	R	
INSURANCE_COMPANY	R	R	R	R	R	-	RU	R	
POLICY_RENEWABLE	R	CRUD	CRU	-	-	-	RU	-	
INCIDENT	-	CRD	R	-	-	R	RU	R	
INCIDENT_REPORT	R	CRD	R	R	-	CRUD	RU	R	
COVERAGE	R	R	CRD	R	-	-	RU	R	
PRODUCT	R	CRUD	R	R	R	R	RU	R	
RECEIPT	R	CRUD	CRD	CRUD	-	-	RU	CRUD	

Conclusion :

We have created database with all documentations and reports included. Our goal was to create DB for Online vehicle insurance company along with code. There were some challenges while making the database and we have carefully figured out the solutions to those and finally created a fully functional database.

We have explored many new things like Stored Procedures, User defined Functions , Views , Grants etc . Overall the project was very interesting to work on. We have enjoyed a lot in the process apart from learning many new concepts. You can find all the CDM , LDM and PDM reports which we have documented in each phase.

PDM Model :