Web Based Application for Insurance Services

A PROJECT REVIEW FINAL REPORT

Submitted by

19BCE2599 Tenzin Khorlo

19BCE2600 Lhendup Dorji

19BCE2607 Pritam Chaurasiya

19BCE2609 Rishi Srikaanth

CSE 2004 Database Management Systems

Guided by

Professor Pradeepa M

Assistant Professor Sr. Grade 1

School of Information Technology and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester 2020-21

Abstract

The present system of the insurance companies is characterized by the manual method as a result serious threat has been posed to the operation of the service and too much workload on the staffs. The manual method involves the marketing staffs moving from one location to the other to meet up with the requirement of their broker and also the files and data of their broker are stored in cabinet which are easily destroyed by rodents.

With regards to this method the insurance computer application for insurance company would be developed this would have the ability to remotely connect insurance brokers in any location for them to carry out their insurance services and also their data would be stored in a secured database. In the software design, to achieve this task we intend to use the Oracle SQL as database and use java IDE Eclipse to make the app.

LIST OF TABLES: **Page no:**

LIST OF FIGURES **Page no:**

TABLE OF CONTENTS

CHAPTER 1

1.1 Introduction

Insurance is a means of protection from financial loss. It is a form of risk management, primarily used to hedge against the risk of a contingent or uncertain loss. The insured receives a contract, called the insurance policy, which details the conditions and circumstances under which the insurer will compensate the insured. The amount of money charged by the insurer to the policyholder for the coverage set forth in the insurance policy is called the premium. If the insured experiences a loss which is potentially covered by the insurance policy, the insured submits a claim to the insurer for processing by a claims adjuster. The insurer may hedge its own risk by taking out reinsurance, whereby another insurance company agrees to carry some of the risks, especially if the primary insurer deems the risk too large for it to carry.

Insurance is an important area of the business service industry. The project is based on implementing a web-based application for insurance services that shows the rates offered by different insurance agencies. The main types of insurance dealt in this project are home insurance, auto insurance, farm insurance, and health insurance. Depending upon the user information, real time quotes are generated from different companies. This project is intended to provide and manage a good customer relationship.

Chapter 2

2.1 Review of literature

In this section, literature survey is given. Accordingly, research papers are reviewed and analyzed based on the prediction methods used:

Table 2.1 Review of Literature

Title, Name of Author And Date of Publication, Publisher info	Problem statement and objectives	Their Proposed Methodology	Outcome and Limitations
1) "Implementation of Futuristic Web Based Application for insurance Services" by Ele, Sylvester, *Department of Computer Science, University of Calabar, Nigeria.* Published in 2018 in American Journal of Engineering Research.	Their project was designed as a means to avert the low insurance penetration in Nigeria by promoting and managing a good customer relationship/experience using our modern day web technology.	The development methodology adopted for the proposed system is a hybrid of Agile development methodology (for development process management) and object oriented analysis and design-OOAD (for analysis and design).	1) He successfully concluded that E-insurance system was efficient, reliable and usable. 2) It was well explained with practical examples. However, explanation for practical output was insufficient and the images of their experiment result was not clear.
2) Web Based Application for Insurance Services Case Study of the Insurance Company by Igwe E.kelvin, who is a software developer, blogger and the CEO of K-Tech System. He is graduate of Computer Science and Engineering. Published date is not mentioned. Both journals can be found this journal in this link: www.academia.edu or from Academia App.	1) To design web based application for insurance service. 2) To provide logical programming which is capable of providing insurance policy holders easy access to any kind of service provided by insurance service.	He intends to achieve the task using Microsoft Visual Studio platform and Microsoft Access 2007 as database to store information.	1) He only explained why E-insurance system is needed and scope of it. 2) It provides essential knowledge to beginner. 3) However, it does not provide practical examples and project was incomplete.

3) "Web Based Application For Insurance Services Case Study Of The Insurance Company", by Esedebe Fidelia Ogechukwu, in July, 2013, a graduate of Computer Science of Information and Technology, Caritas University Amorji – Nike Enugu, Nigeria.	1) To ensure effective insurance service communication around the globe from a remote location using the web application. 2) To promote growth and financial stability of insurance companies and effectively enable policy holders monitor their service around the globe. 3) To professionalize insurance services and develop insurance consciousness among the general populace.	They proposed to use Visual studio 2008 platform and Microsoft access as database.	1) His project detailed and was well explained with practical example. 2) His project was successfully done and output was also displayed. 3) But, there was no ER diagram.
4) "Teach yourself SQL in 21 days, second edition ," ,by Auwal Gene , available at link : http://www.auwalgene.com/@mystudents/lecturenotes/teach_urself_sql.pdf	To Help reader to learn: 1. The process to create table . 2. The various constraints and its implementation. 3.	1. He explains the concepts of classes, objects, properties, methods, and events. 2. He teaches the process of visual program design and development.	1) He explains the process in detail with example making it easier for learners. 2) Recommended for Beginners. 3) However, he uses programming as an example in between which only the ones who know programming can understand.
5) "Web Based Project Ideas & Topics" by nevon Projects, electronics |Software and mechanical kits. Available in link below: https://nevonprojects.com/web-based-project-ideas-topics/	1) To help user learn more about Java Projects using bootstrap,html and c 2) To help learner gain basic knowledge to create web pages and forms.	1) They teach the basics of html, bootstrap. 2) They also teach how to connect database to server.	1) They have given various source codes of different projects with comment line explaining some lines of codes. 2) Not recommended for beginners.

Chapter 3

3.1 Problem Statement

Insurance Companies have been experiencing difficulties trying to attend to their customers quickly and efficiently when required due to lack of adequate documentation of data. Web based application for insurance services has been acknowledged as the fourth site technology that could foster communication of insurance service very quickly and efficiently reaching people around the world irrespective of the location. So we look forward to developing a simple app that can store information of insurance, clients, admin and agent which can benefit the insurance company for keeping records in the long run.

3.2 Objectives

- The aim of this project is based on implementing a web based application for insurance services in other to help insurance companies keep records of every details effectively and thus, have better interaction with clients.
- It is also aimed to promote growth and financial stability of insurance companies and effectively enable policy holders monitor their service with transparency.
- It is aimed to ensure effective insurance service communication around the globe.
- To establish a sound national insurance market; and also add speed to their data processing and retrieving.
- To reduce Data Redundancy and facilitate faster Searching of information.

Chapter 4

HARDWARE AND SOFTWARE REQUIREMENTS:

4.1. Hardware Requirements

- Computer system
- Hard disk size 40 GB and above
- Processor speed of 1.6 GHz
- Ram size of 512 MB and above
- Network connection is required to connect Database with IDE

4.2. Software Requirements

- Windows operating system
- Sql plus oracle(as Database)
- Eclipse as java IDE(Contains JavaScript Source files and classes)
- Ojdbc.jar (Jar File required to connect Database with java IDE)
- Oracle Data Source (Used to connect to Database) found in ojdbc file.

CHAPTER 5

IMPLEMENTATION:

5.0 Proposed Methodology:

The proposed system is for making easier to manage policy holder details, agent details, policy details, claimant details and payment details. So this will be developed for managing the insurance management system. The overall system is control through the main menu. The main menu contains 4 parts: "User Login", "Administrator Login", "About us", "Exit"

5.1 Requirement Analysis

5.1.1 FOR ACCESSING:

5.1.1.A USER:

User needs to Log in using their Claimant_id as their User_id and enter password which will be stored in database. And they can view their own info, their Insurance details and their agent details

5.1.1.B ADMIN:

For Admin they login using Pholder_id as their user_id and enter password. They can view their own information as well as overall data of users and agents and their contracts of Sales.

5.1.2 TO GET POLICY KEYS AND INSURE

5.1.2.A CAR INSURANCE:-

Customer Visits an Agent and asks him to insure his vehicle. Agent takes the customer details and the vehicle details (i.e. RC Book, and vehicle registration papers). Agent forwards the Details to the Company. Company gives the Quotation to the agent for different coverage's. Agent gives the quotation amount to the customer. Customer as per his convenience asks to issue the policy under the coverage of his choice. Agent takes the cheque or cash for the policy from the customer. Agent gives the cheque to the company and request them to issue the policy for the vehicle. Company arranges a check for the vehicle to be insured. If the vehicle is in Condition and passes the test set by the insurance company the vehicle is then insured by the company. The policy copy is then issued to the customer and sent to the mail address. The agent receives the commission as set by the company after the issuance of the policy.

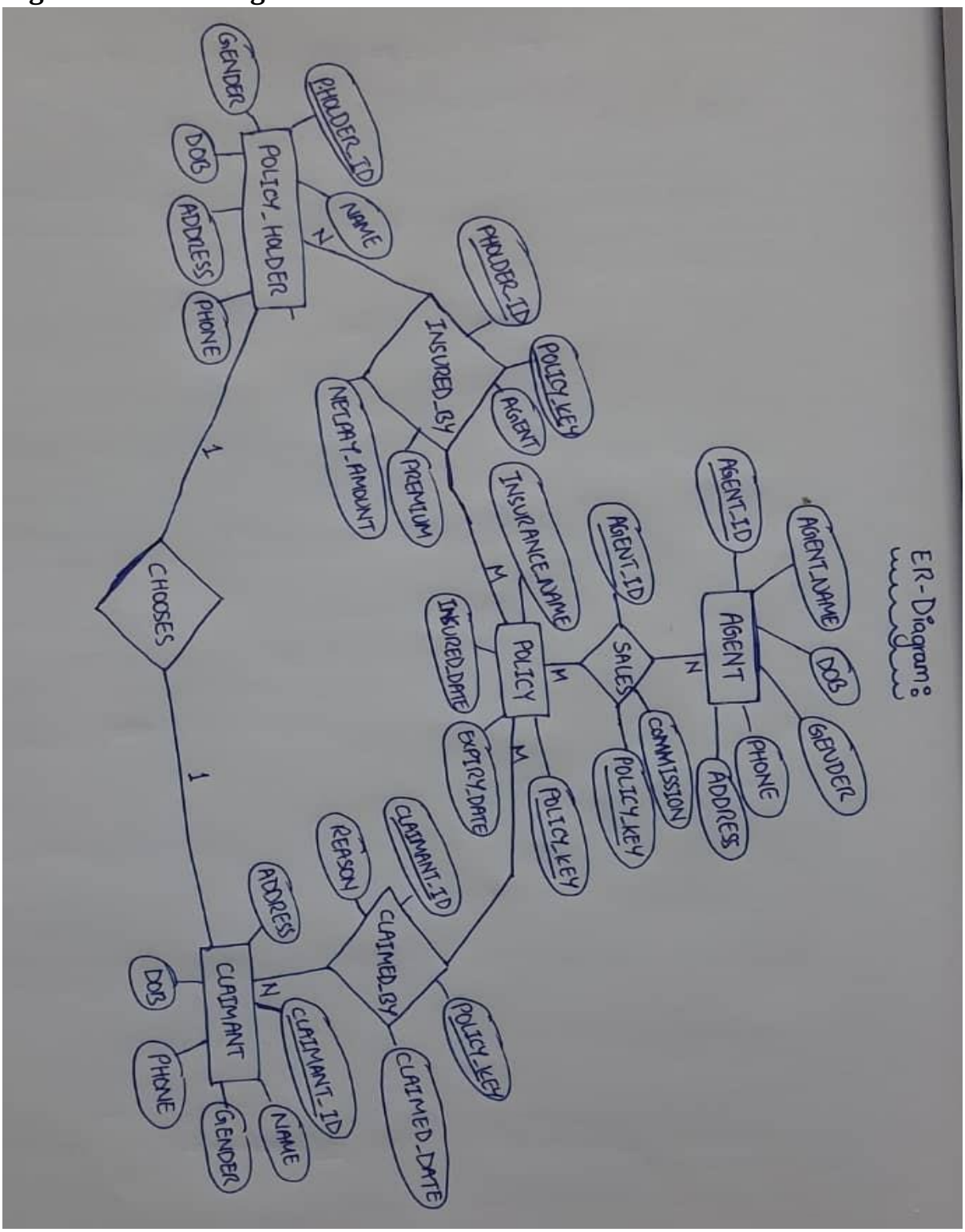
5.1.2.B HEALTH INSURANCE:-

Customer Visits an Agent and asks for a Health Insurance. Agent Proposes the best suited health insurance to the customer as per his/her need. Customer decides whether he is interested in that policy if yes he asks for the quotation. Agent takes the customer details and gives it to the company. Company then according to the customer details specified by the agent depending on age of the customer gives different quotations on the basis of sum insured by the customer. Agent asks the customer the amount he/she wants to get insured for. Agent then gives the quotation to the customer. Customer then informs the agent whether he\she wants to get insured or not. If yes agent informs the company. If the customer is of age less than 45 he is issued the policy without any medical test. If the age of the customer is Greater than 45 then the company arranges the medical tests for the customer. Customer has to pay for the medical tests from his own pocket. If the customer passes the tests and the policy is issued then the amount of tests paid by the customer is then refunded. Once the customer is eligible for the policy after the test he/she has to give the premium amount to the agent. Agent gives the amount to the company. Company then issues a policy and the person is insured. Company sends the customer copy to the customer on the postal address specified by the customer.

5.2 ER Diagram:

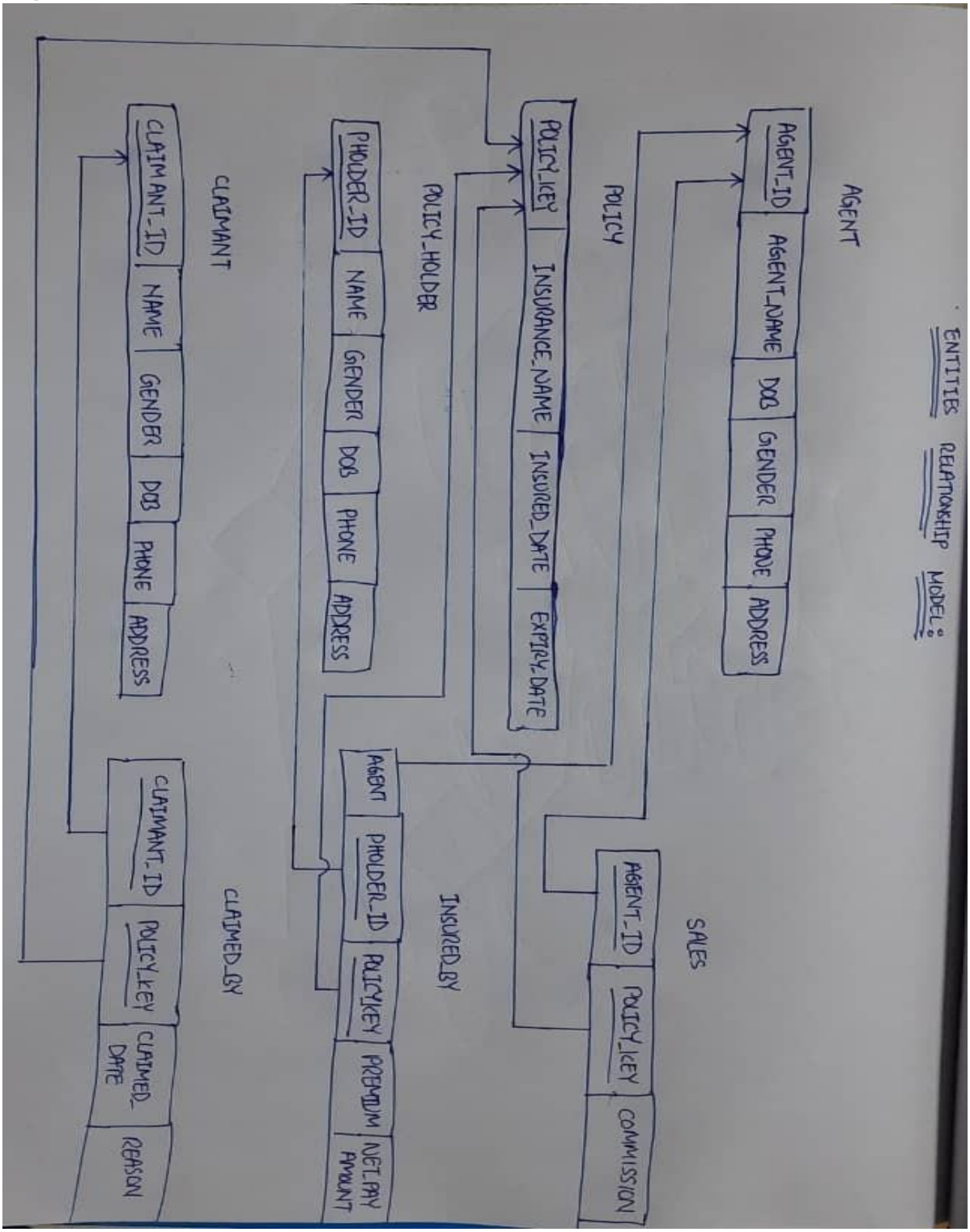
5.2.1 ER Diagram

Figure 5.1 : ER Diagram



5.2.2 Entities Relations Model:

Figure 5.2 : Entities Relations Model



5.3 Normalization:

5.3.1 NORMALIZATION OF ALL TABLES

Figure 5.3a : Normalization part 1

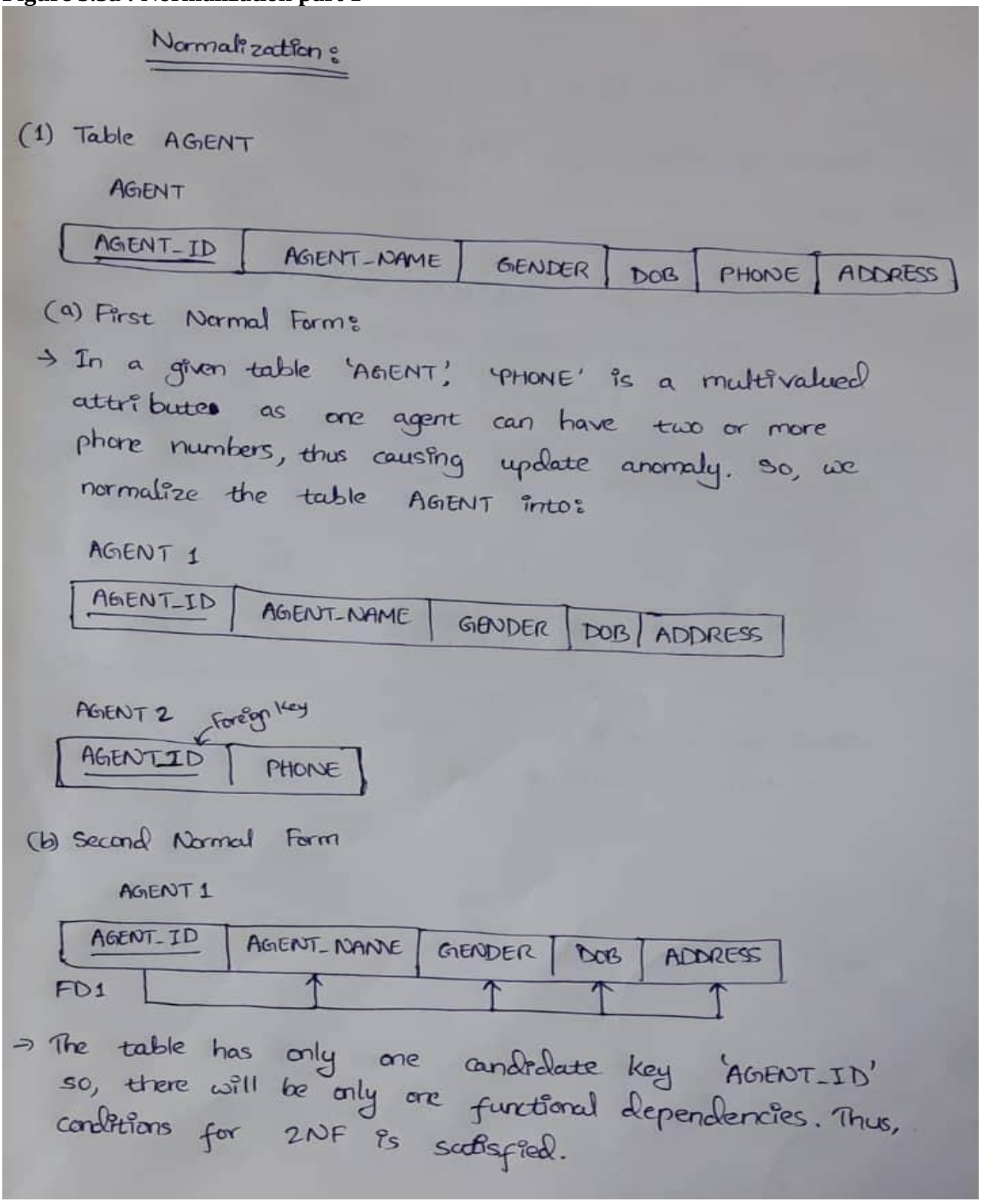


Figure 5.3b : Normalization part 2

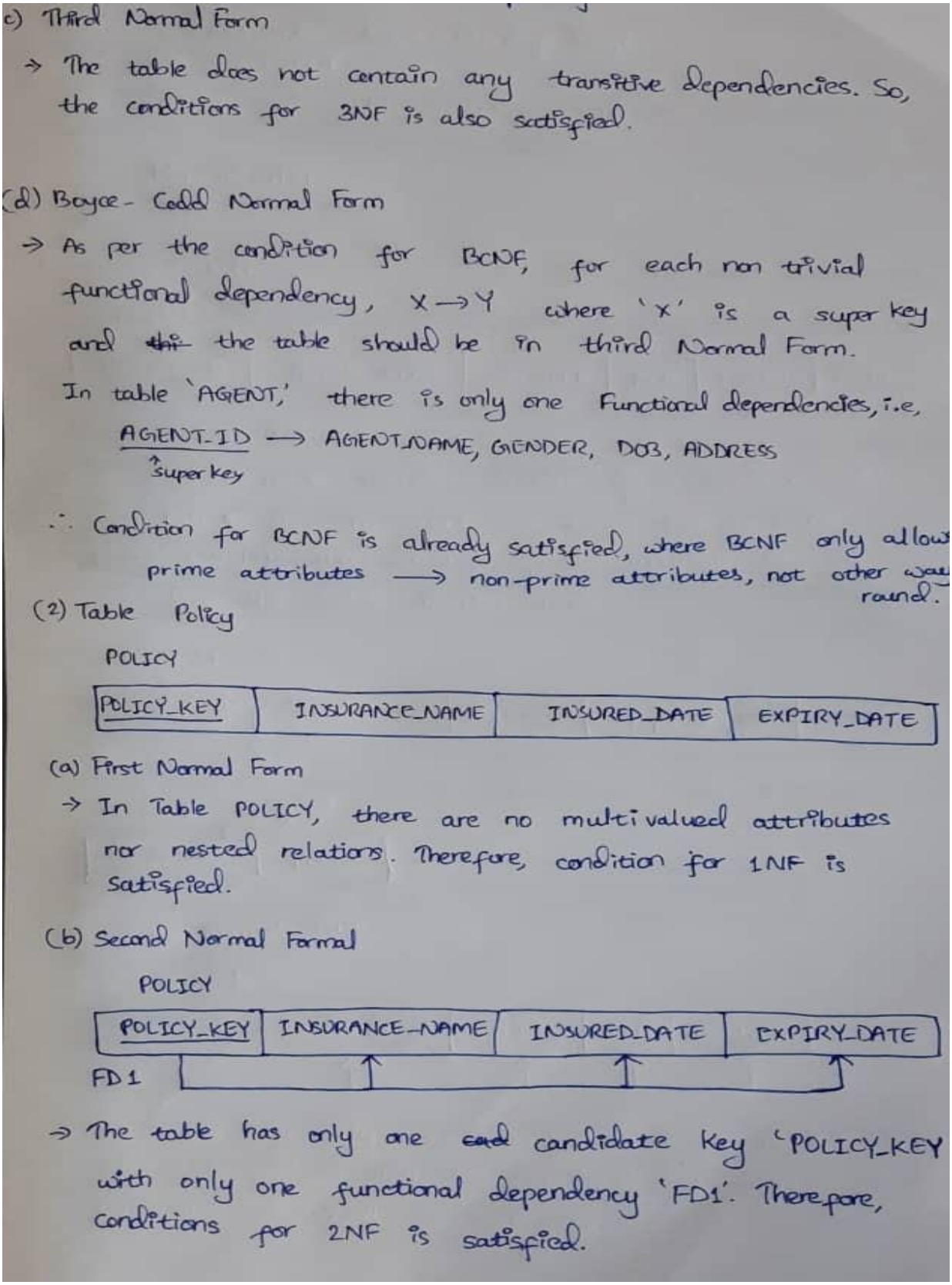


Figure 5.3c : Normalization part 3

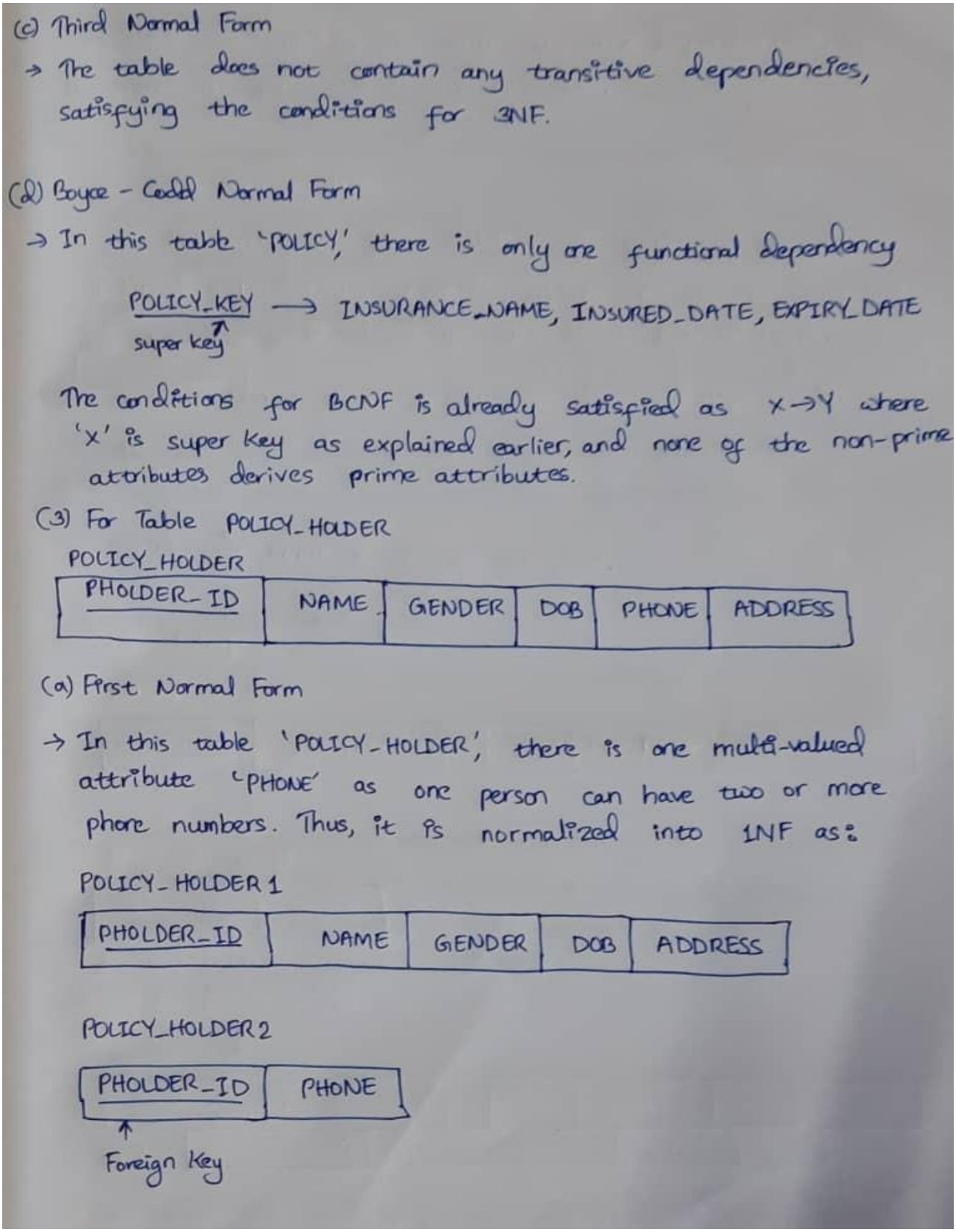
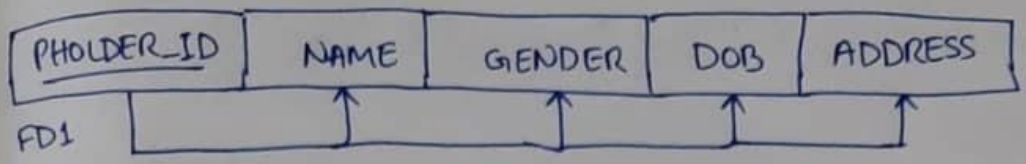


Figure 5.3d : Normalization part 4

b) Second Normal Form

POLICY_HOLDER 1



→ There is only one candidate key 'PHOLDER_ID' with one functional dependency 'FD1'. So, the conditions for 2NF is already satisfied.

(c) Third Normal form

→ There are no ~~functional de~~ transitive dependencies among non-prime attributes. Thus, conditions for 3NF is satisfied.

(d) Boyce Codd Normal form

→ In this table, the functional dependency is

PHOLDER_ID ⟶ NAME, GENDER, DOB, ADDRESS

(PHOLDER_ID: super key)

Since, for functional ~~deperey~~ dependency X→Y where 'X' is super key, the conditions for BCNF is already satisfied, where none of the non-prime attributes is deriving prime attributes.

Figure 5.3e : Normalization part 5

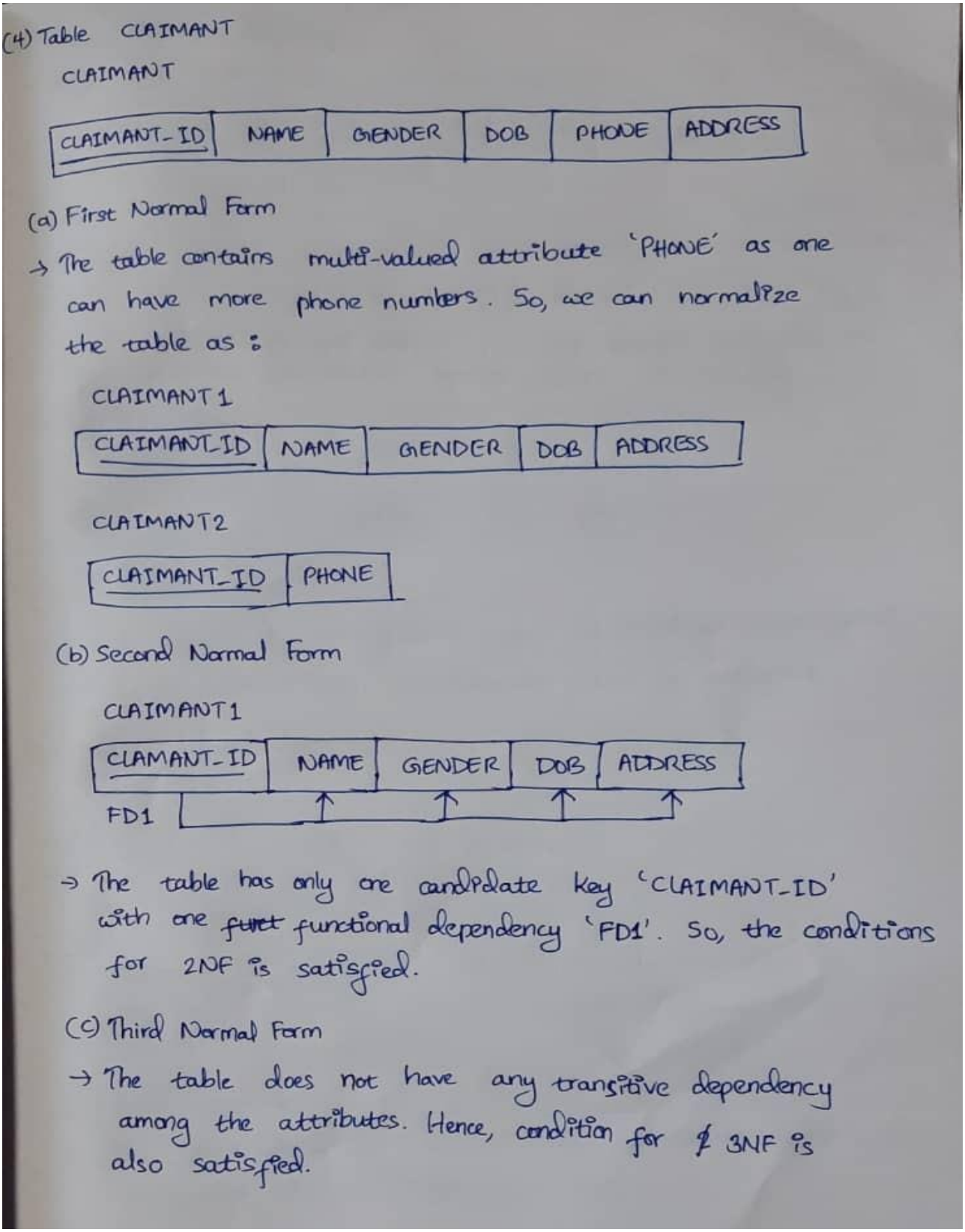


Figure 5.3f : Normalization part 6

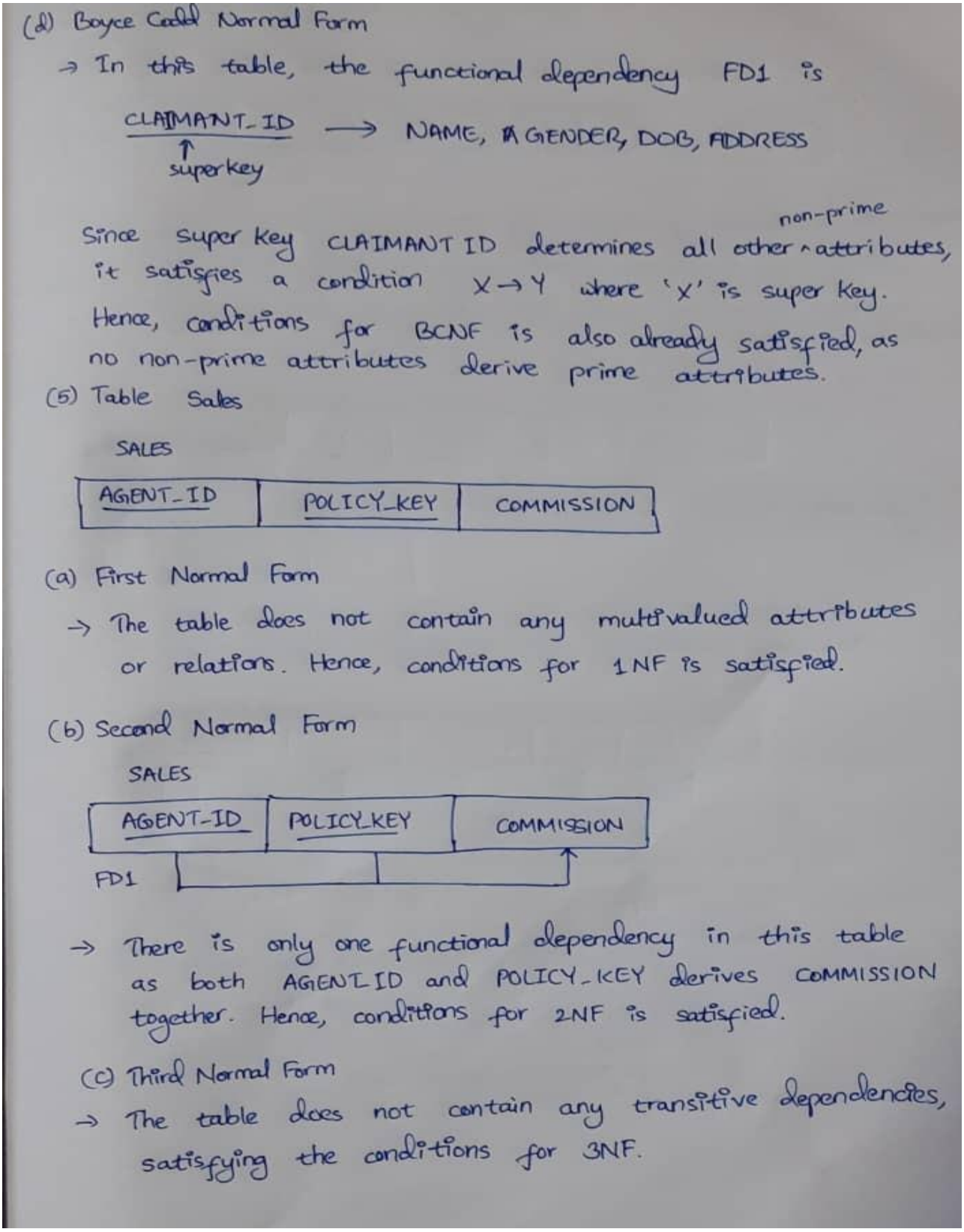


Figure 5.3g : Normalization part 7

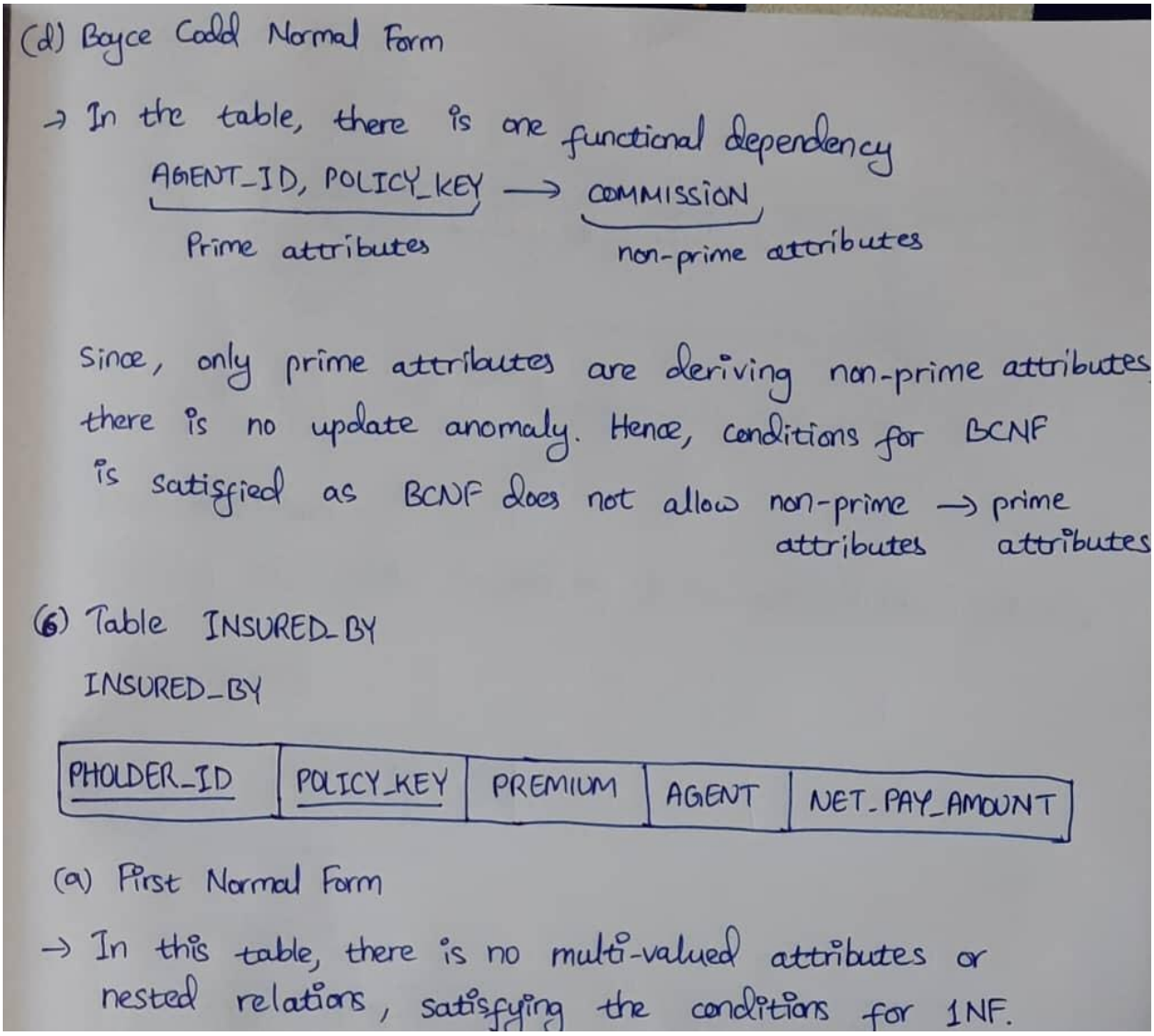


Figure 5.3h : Normalization part 8

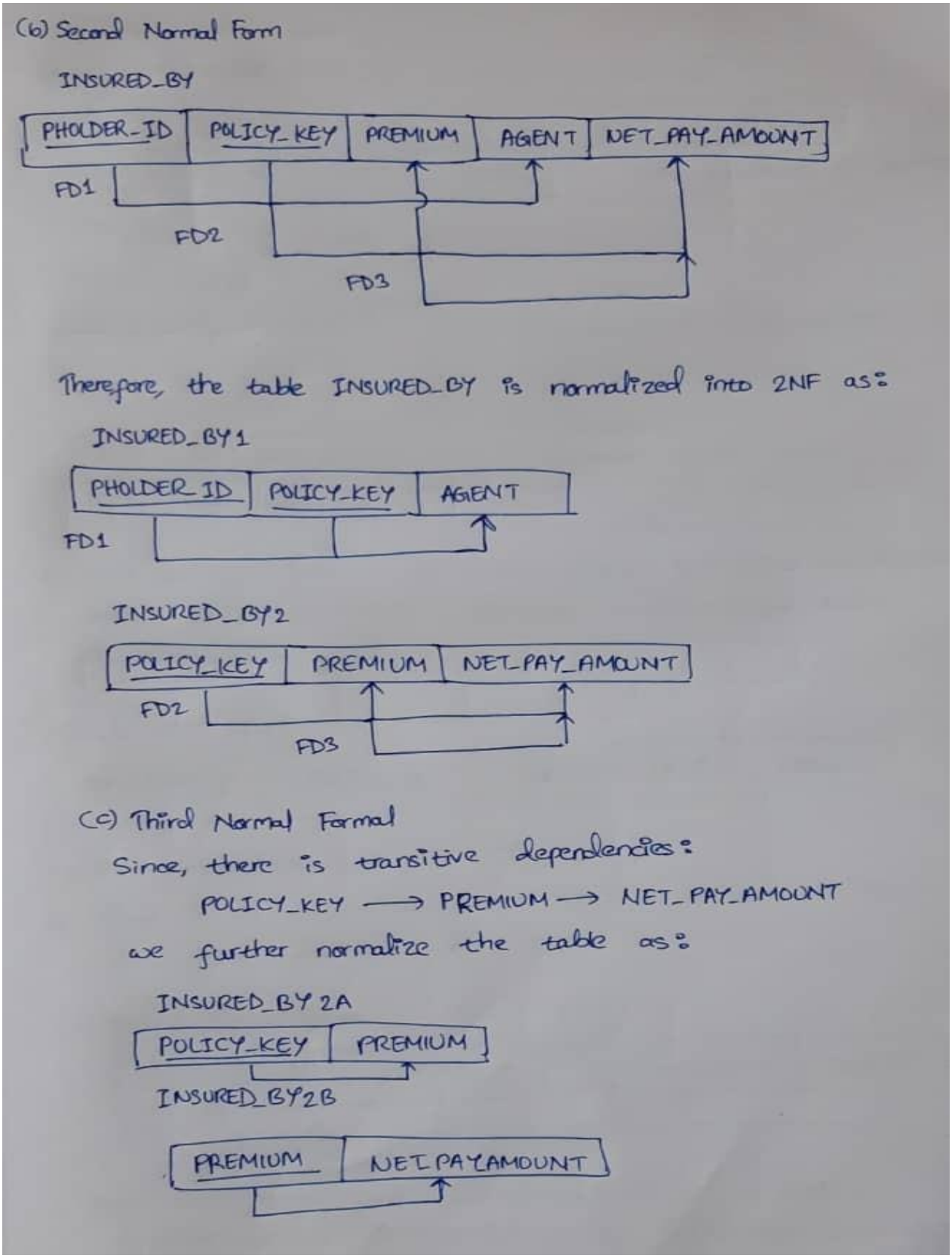


Figure 5.3i : Normalization part 9

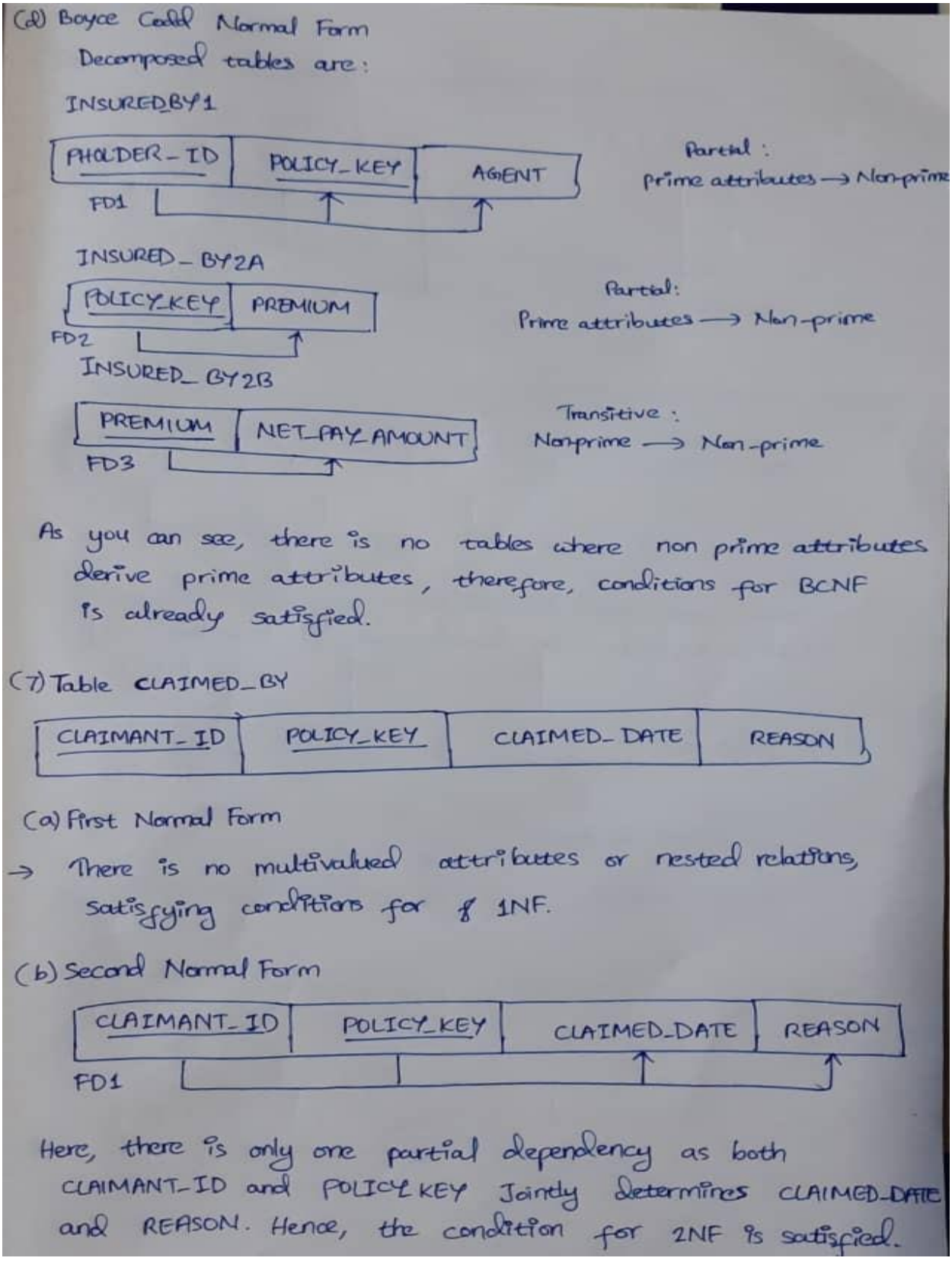
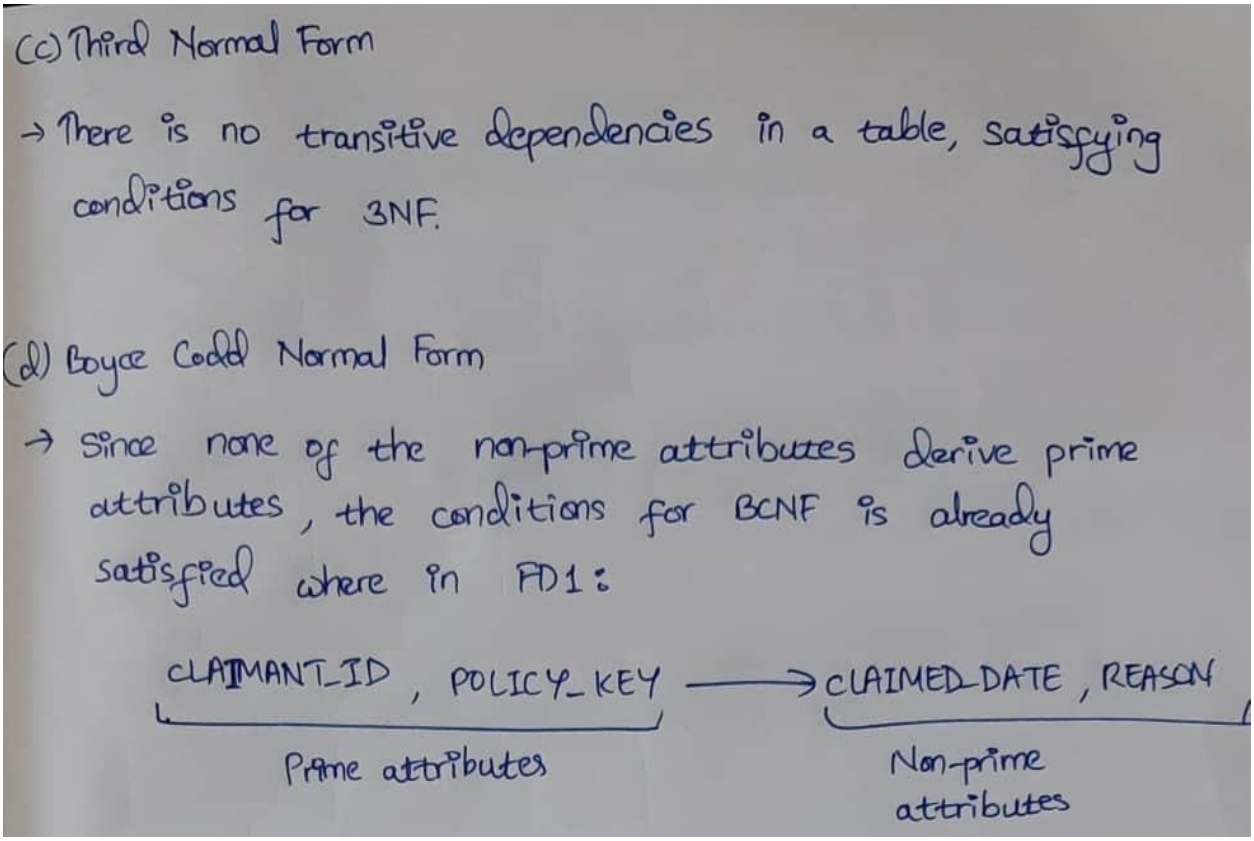


Figure 5.3j : Normalization part 10



Chapter 6

RESULTS AND DISCUSSION

6.1 Creation of tables using DML commands:

1) Table Agent:

After Normalizing, we get table Agent1 and Agent2. Creating table Agent1 and Agent2:

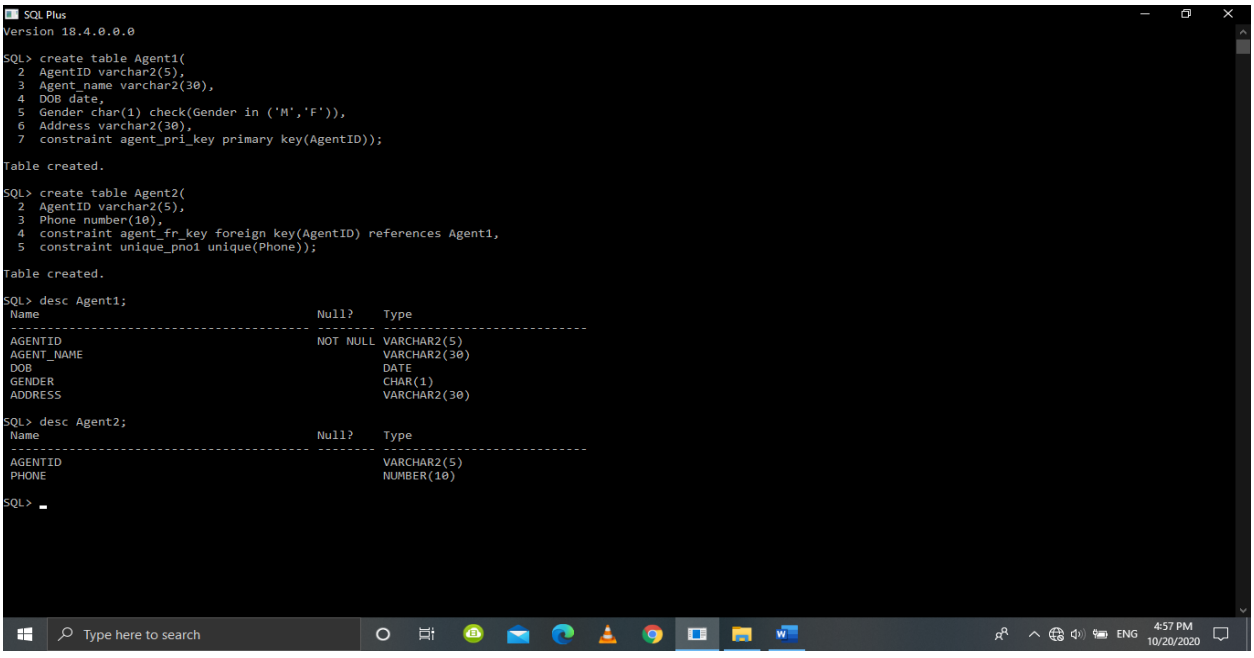
Table 6.1: Agent1

Attributes	Data types	Constraints
Agent_id	Varchar2(5)	Primary key
Agent_name	Varchar2(30)	
DOB	Date	
Gender	Char(1);	Check (Gender in ('M','F'))
Address	Varchar2(30)	

Table 6.2 : Agent2

Attributes	Data types	Constraints
Agent_id	Varchar2(5)	Foreign key
Phone	Number(10)	Unique

Figure 6.1 : Creation of table Agent1 and Agent2

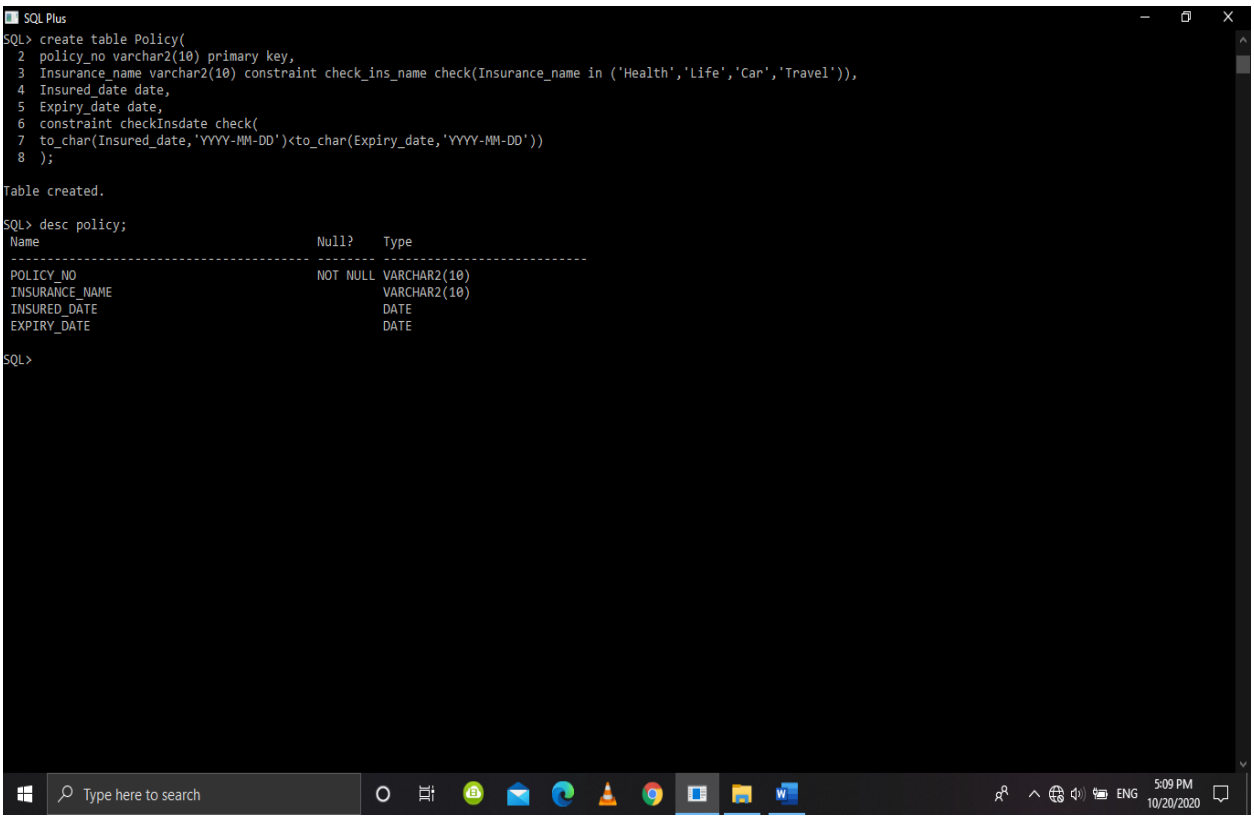


2) Table policy

Table 6.3 : Policy

Attributes	Data types	Constraints
Policy_no	Varchar2(10)	Primary key
Insurance_Name	Number(10)	Check(in (‘Health’, ‘Car’, ‘Life’, ’Travel’))
Insured_date	Date	Check if it is before Expiry date
Expiry_date	Date	

Figure 6.2 : Creation of table Policy



3) Table Policy_Holder:

After Normalizing, we get table Policy_Holder1 and Policy_Holder2. Creating table Policy_Holder1 and Policy_Holder2:

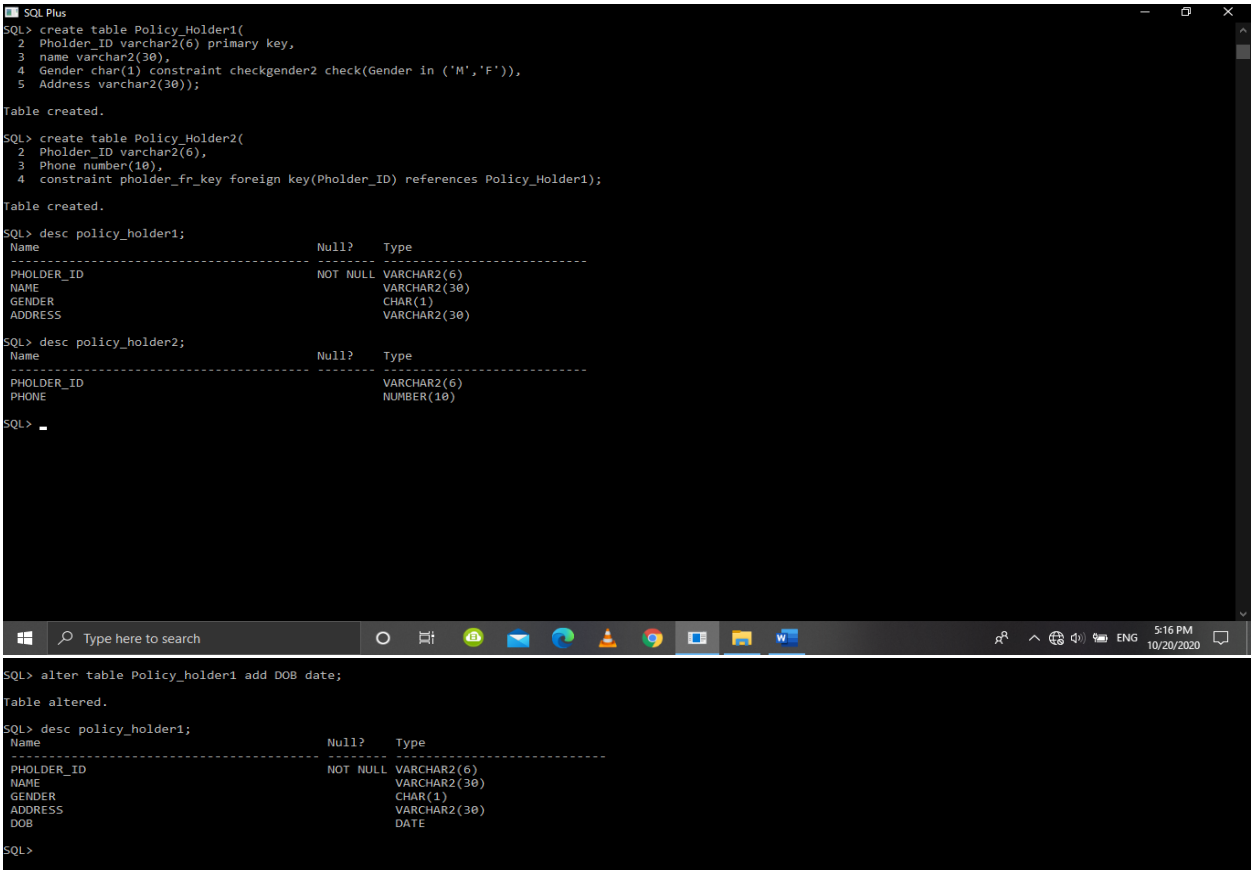
Table 6.4 : Policy_Holder1

Attributes	Data types	Constraints
PHolder_id	Varchar2(6)	Primary key
Name	Varchar2(30)	
DOB	Date	
Gender	Char(1);	Check (Gender in ('M','F'))
Address	Varchar2(30)	

Table 6.5 : Policy_Holder2

Attributes	Data types	Constraints
PHolder_id	Varchar2(6)	Foreign key
Phone	Number(10)	Unique

Figure 6.3 : Creation of table Policy_Holder1 and Policy_Holder2



4) Claimant

After Normalizing, we get table Claimant1 and Claimant2. Creating table Claimant1 and Claimant2:

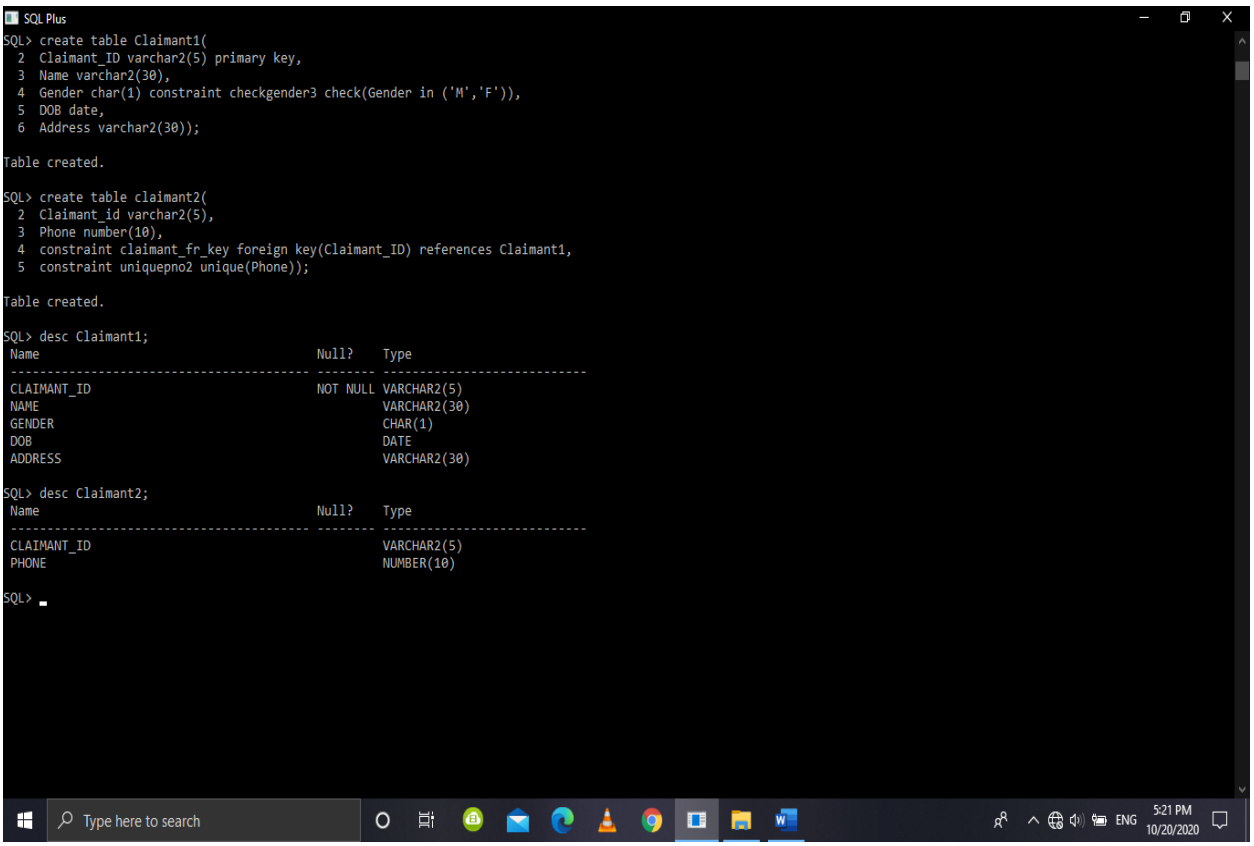
Table 6.6 : Claimant1

Attributes	Data types	Constraints
Claimant_id	Varchar2(5)	Primary key
Name	Varchar2(30)	
DOB	Date	
Gender	Char(1);	Check (Gender in ('M','F'))
Address	Varchar2(30)	

Table 6.7 : Claimant2

Attributes	Data types	Constraints
Claimant_id	Varchar2(5)	Foreign key
Phone	Number(10)	Unique

Figure 6.4 : Creation of table Claimant1 and Claimant2

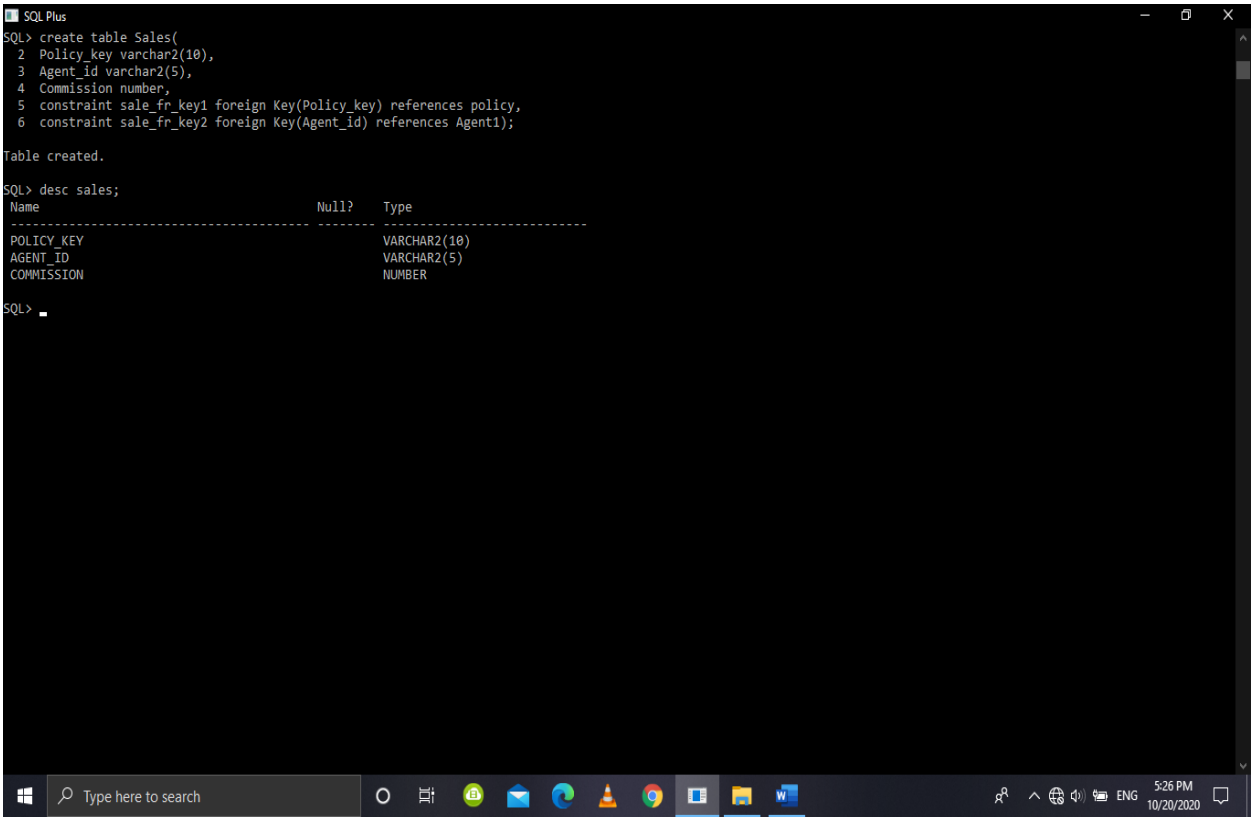


5) Table Sales

Table 6.8 : Sales

Attributes	Data types	Constraints
Policy_key	Varchar2(10)	Primary key, Foreign key
Agent_id	Varchar2(5)	Foreign key
Commission	number	
Claimant_id	Varchar2(5)	Foreign key

Figure 6.5 : Creation of table Sales




```
SQL> alter table sales add Claimant_id varchar2(5) add constraint claimfrkey foreign key(claimant_id) references claimant1;

Table altered.
```

6) Table Insured_by

After Normalizing, we get table Insured_by1A and Insured_by1B and Insured_by2. Creating table Insured_by1A and Insured_by1B and Insured_by2:

Table 6.9 : Insured_by1

Attributes	Data types	Constraints
Policy_key	Varchar2(10)	Foreign key, Primary key
Pholder_id	Varchar2(6)	Foreign key
Agent	Varchar2(5)	Foreign key

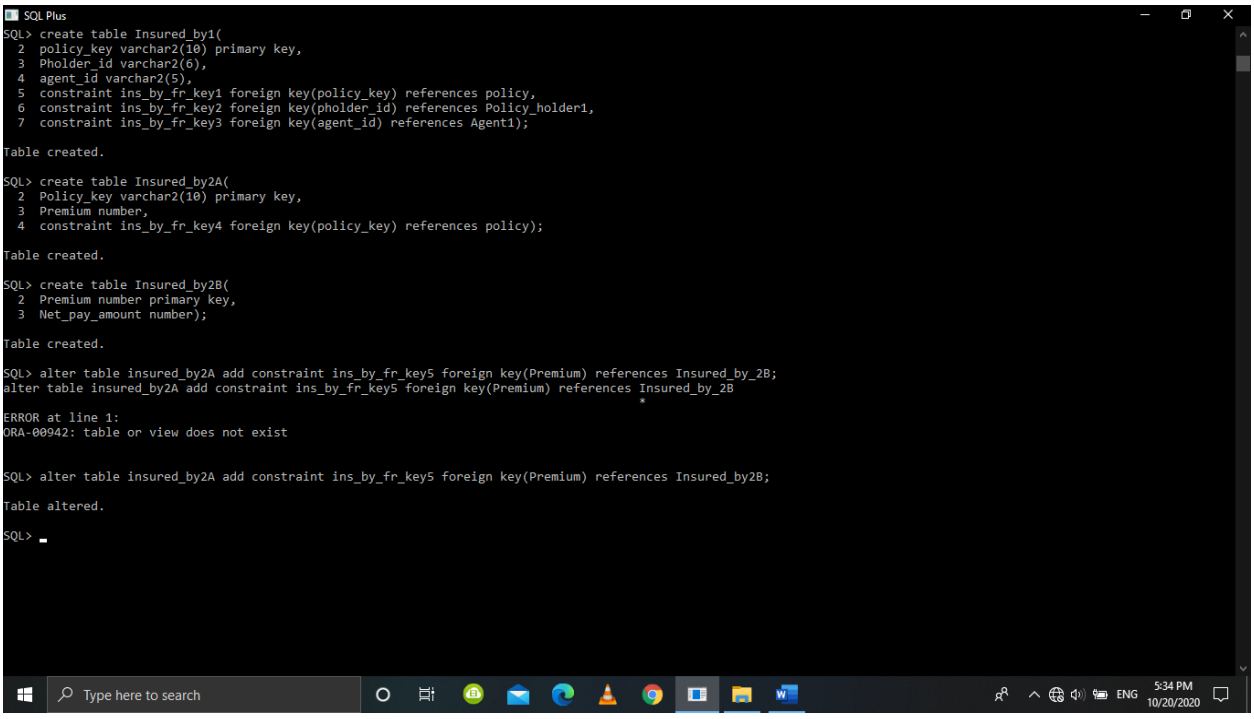
Table 6.10 : Insured_by2A

Attributes	Data types	Constraints
Policy_key	Varchar2(10)	Primary key
Premium	Number	Foreign key

Table 6.11 : Insured_by2B

Attributes	Data types	Constraints
Premium	Number	Primary key
Net_pay_Amount	Number	

Figure 6.6 : Creation of table Insured_by1, Insured_by2A and Insured_by2B

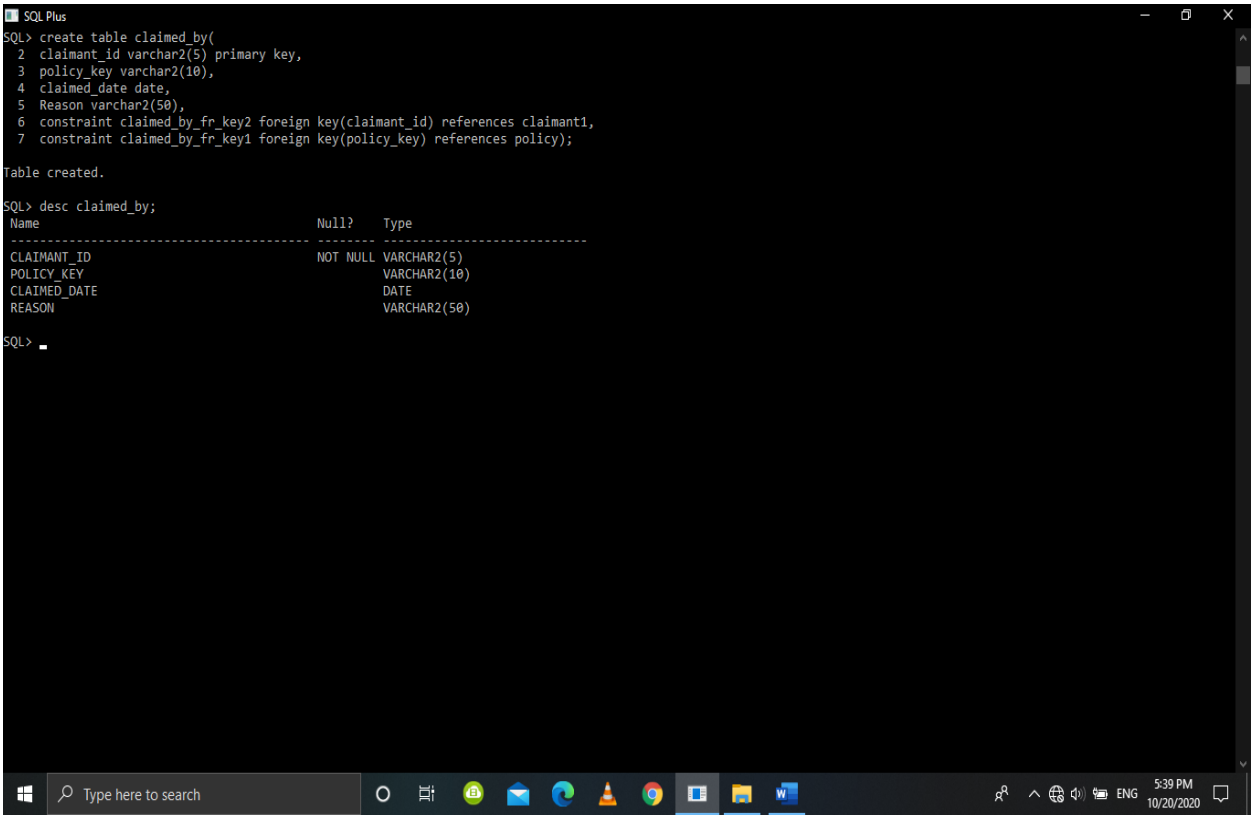


7) Table Claimed_by

Table 6.12 : Claimed_by

Attributes	Data types	Constraints
Claimant_id	Varchar2(5)	Foreign key
Policy_key	Number(11)	Foreign Key
Claimed_Date	Date	
Reason	Varchar2(50)	

Figure 6.7 : Creation of table Claimed_by



8) Log in table: These extra tables are created to store password of User and admin for login purpose:

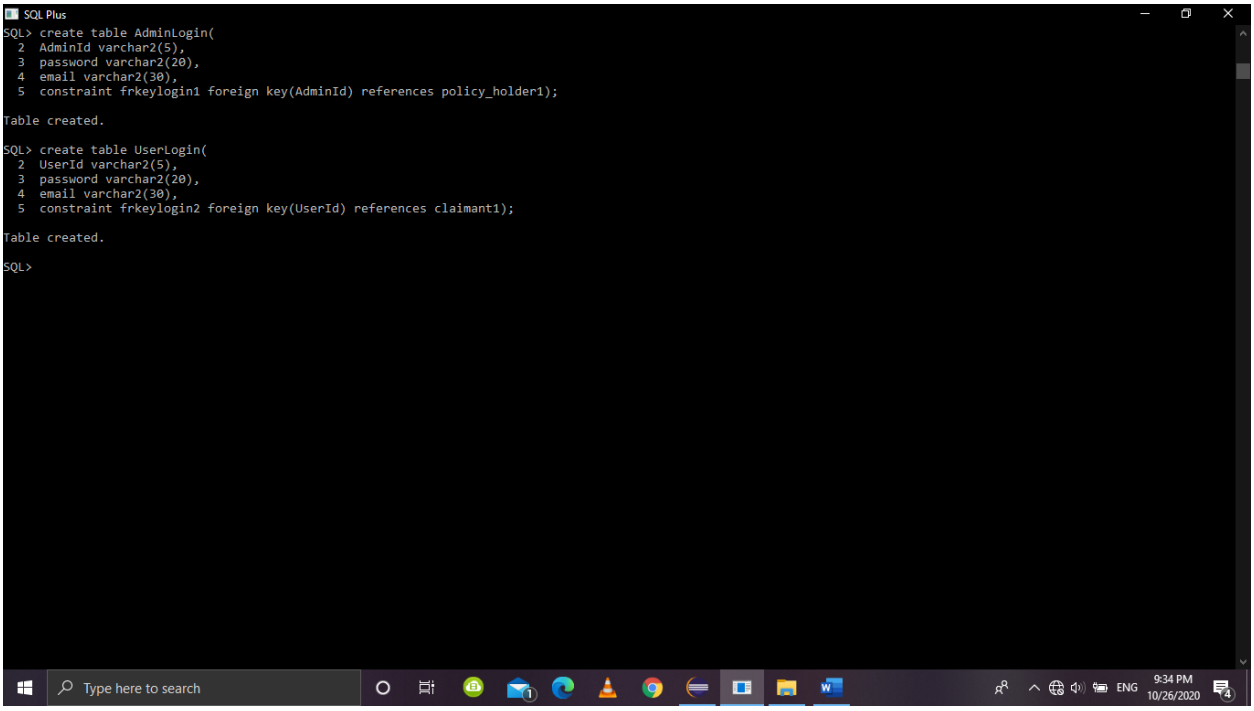
a) Table 6.13 : UserLogin:

Attributes	Data types	Constraints
User_id	Varchar2(5)	Primary, Foreign key
Password	Varchar2(20)	
Email	Varchar2(30)	

b) Table 6.14 : AdminLogin:

Attributes	Data types	Constraints
Admin_id	Varchar2(5)	Primary, Foreign key
Password	Varchar2(20)	
Email	Varchar2(30)	

Figure 6.8 : Creation of table UserLogin and AdminLogin



6.2 Updated values in database:

Figure 6.9 : Updated values in table Claimant1 and Claimant2

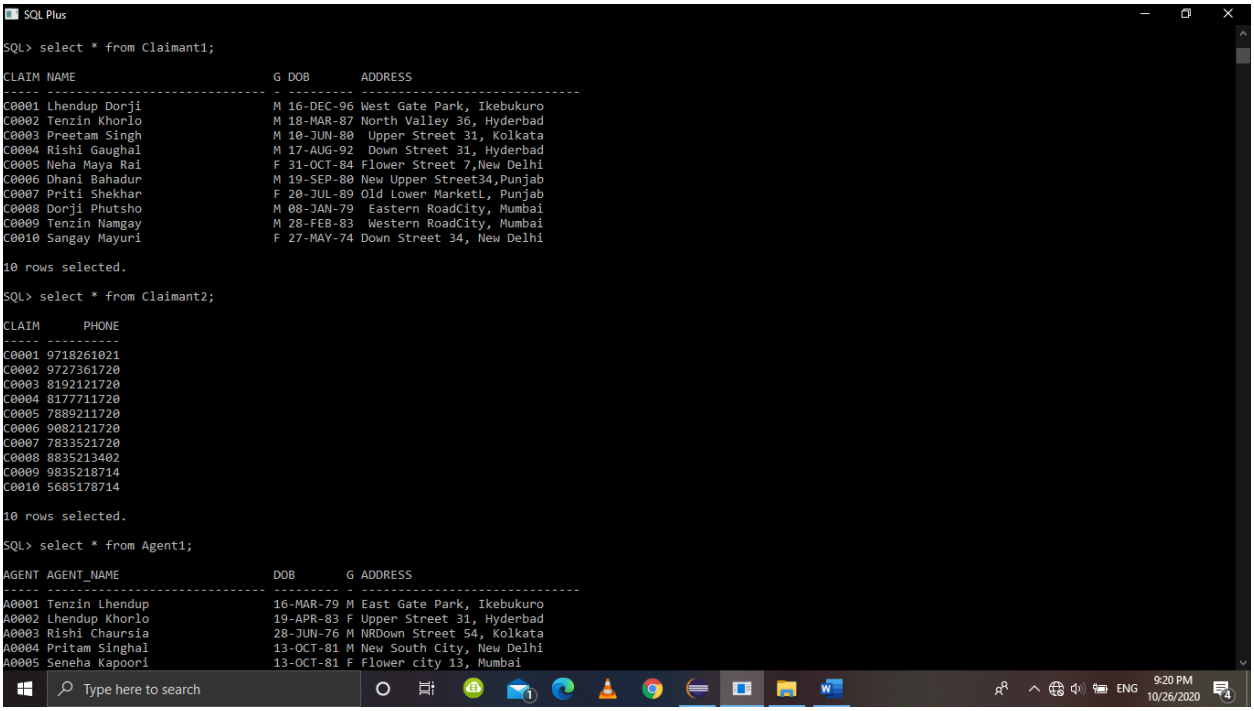


Figure 6.10 : Updated values in table Agent1, Agent2 and Policy

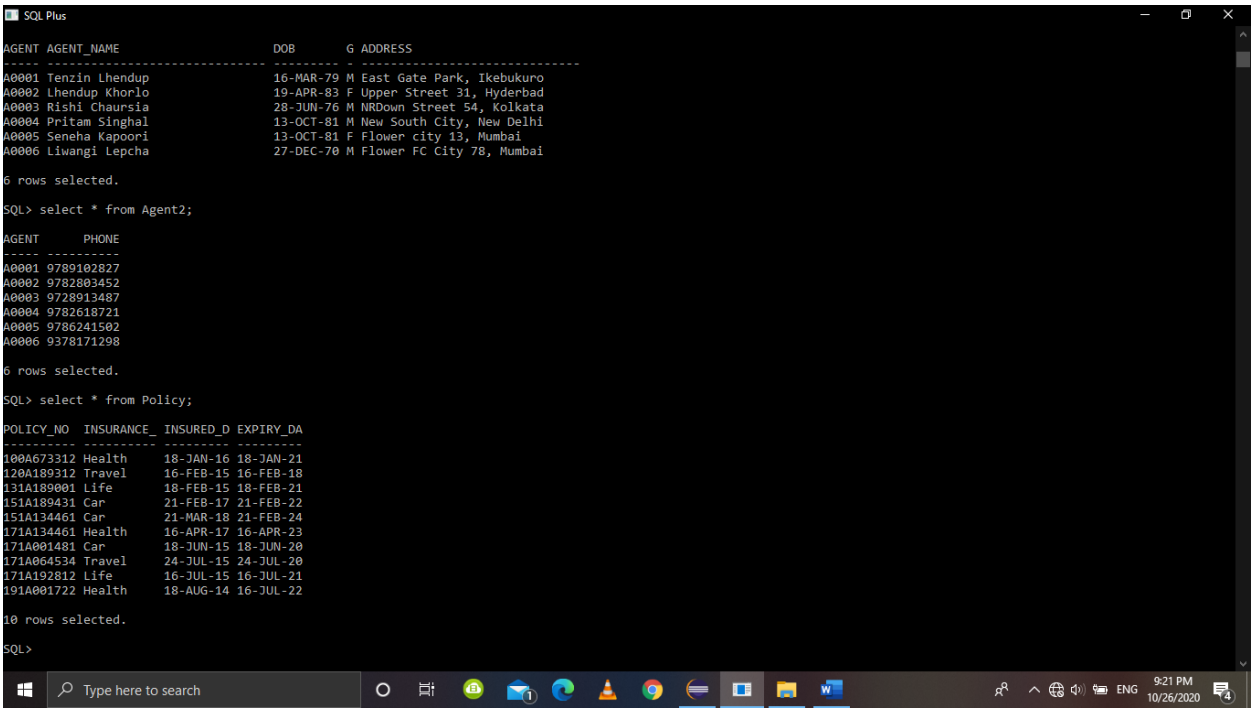


Figure 6.11 : Updated values in tables Policy_holder1, Policy_Holder2 and Sales

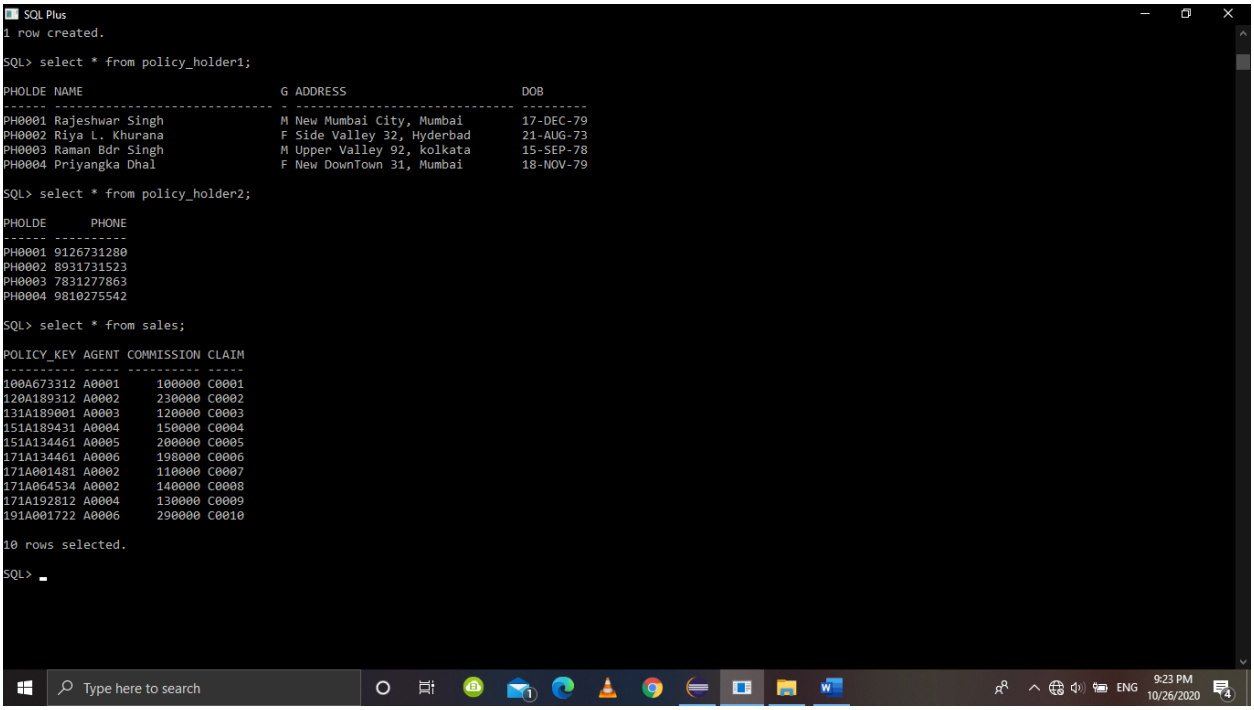


Figure 6.12 : Updated values in table Insured_by1, Insured_by2A and Insured_by2B

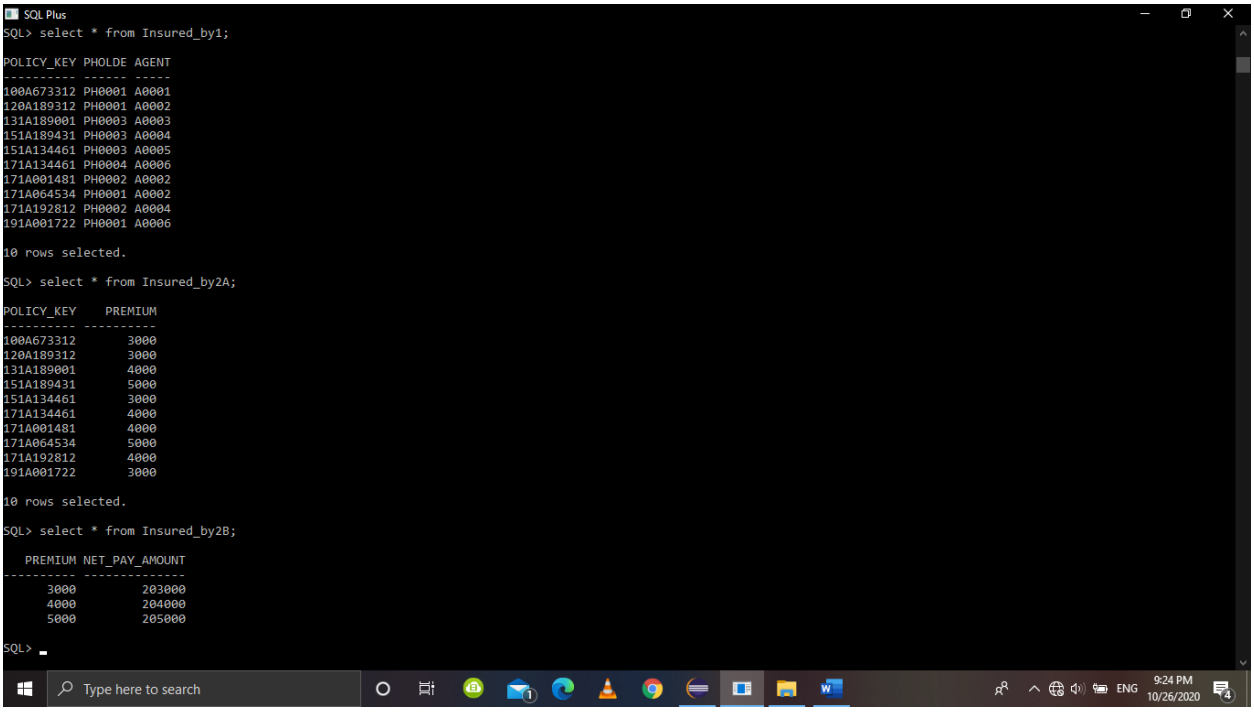


Figure 6.13 : Updated values in tables insured_by2B and Claimed_by

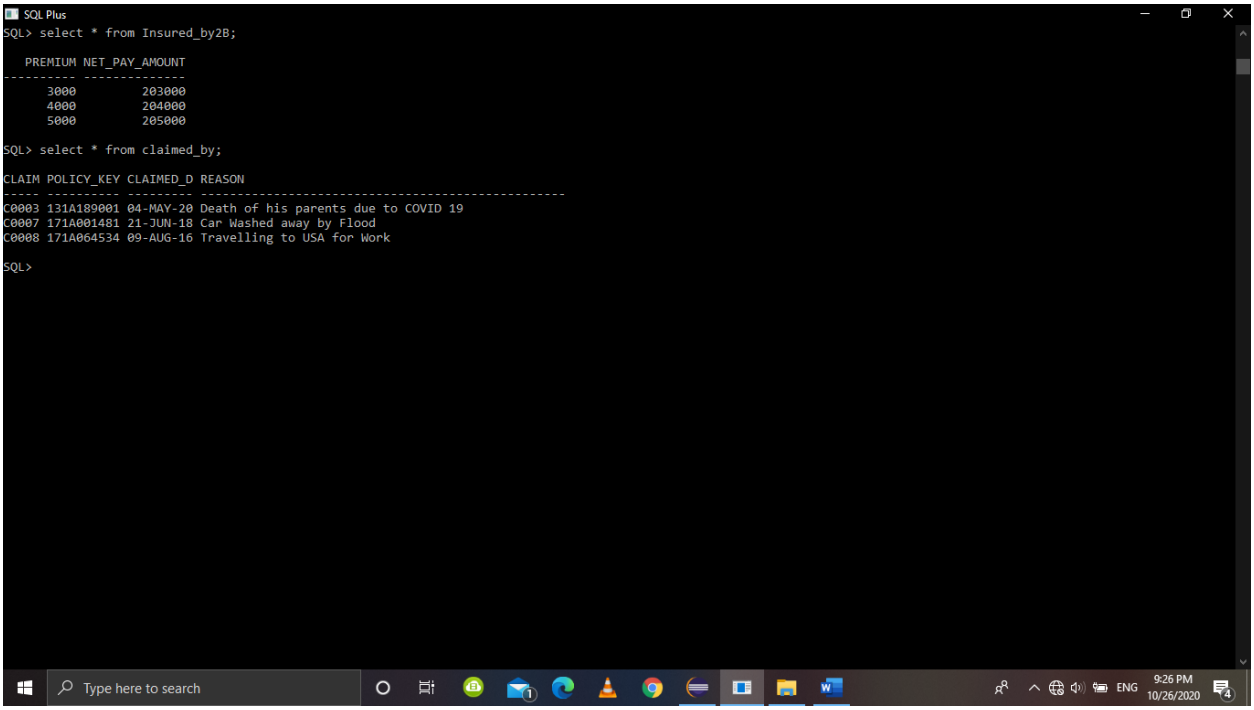
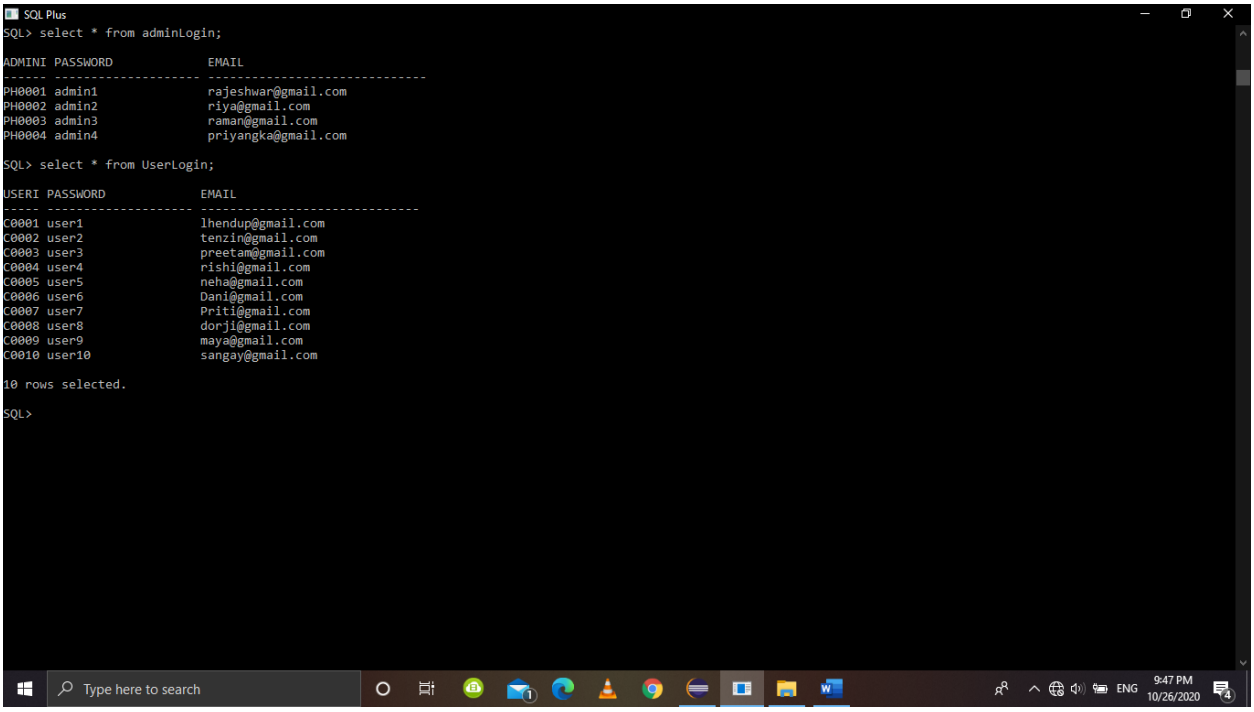


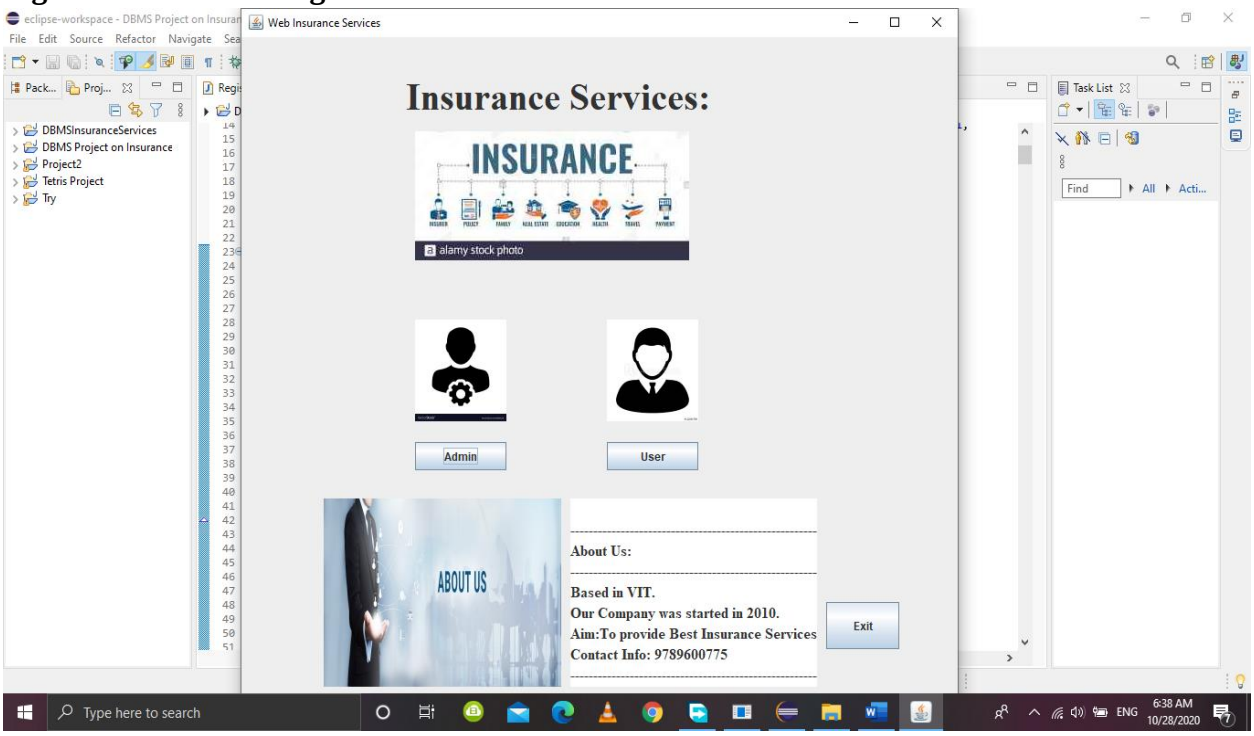
Figure 6.14 : Updated values in tables UserLogin and AdminLogin



6.3 Implementation in Eclipse(App Creation):

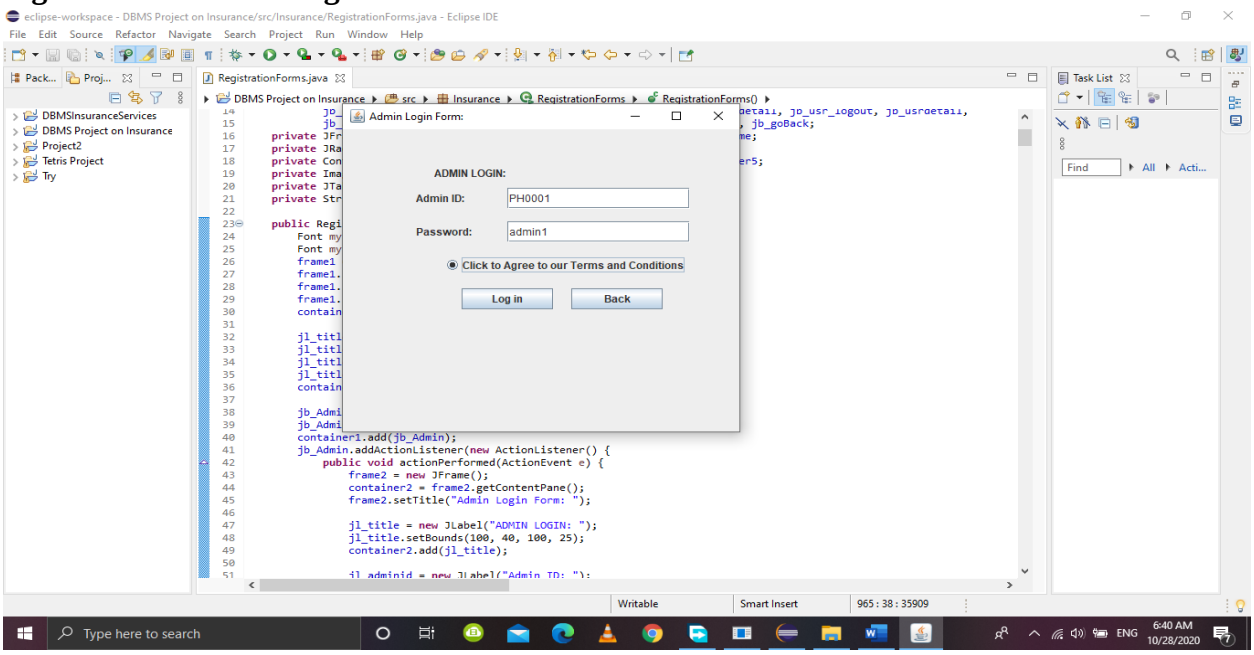
1) Home Page:

Figure 6.15 : Home Page



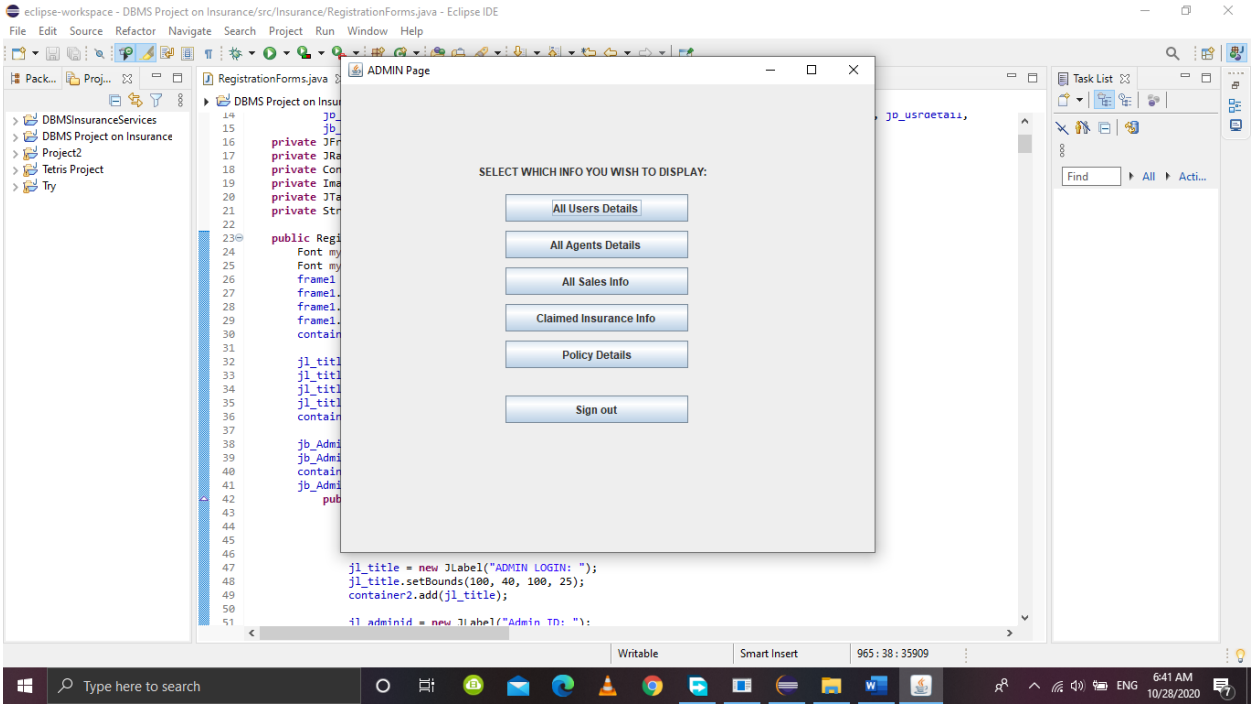
2) After Clicking on Admin:

Figure 6.16 : Admin Login Form



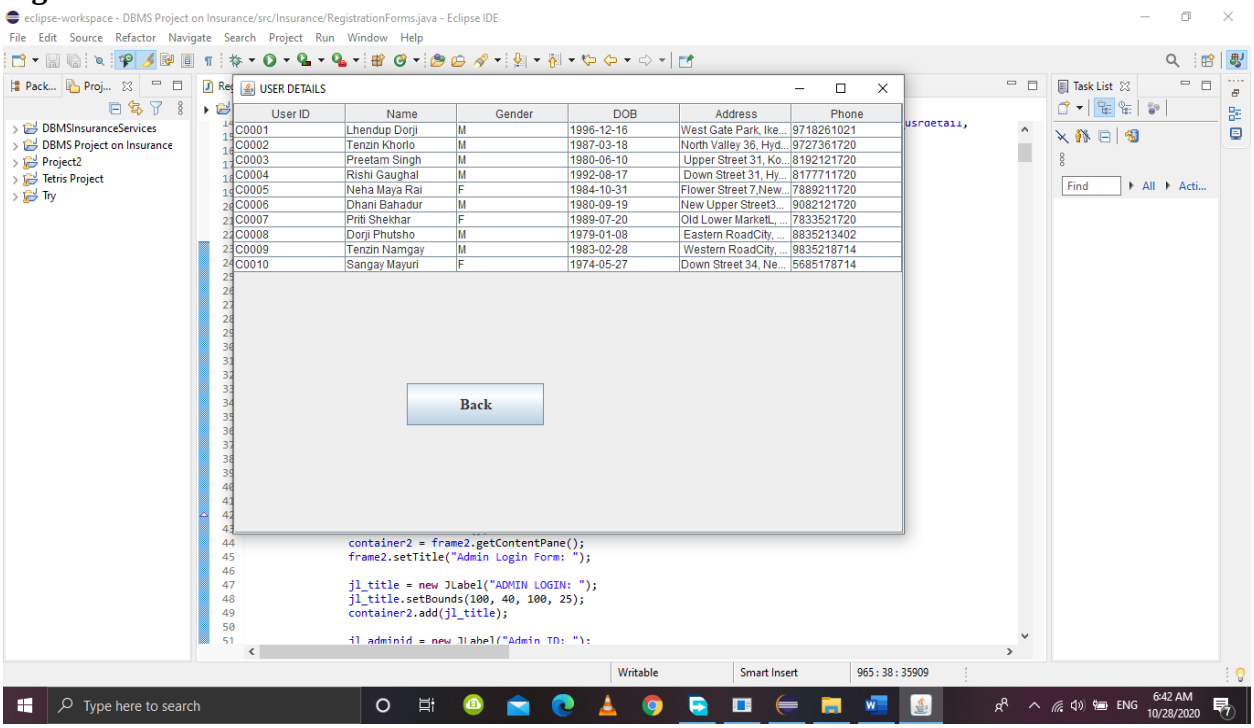
3) We get admin page

Figure 6.17 : Admin Page



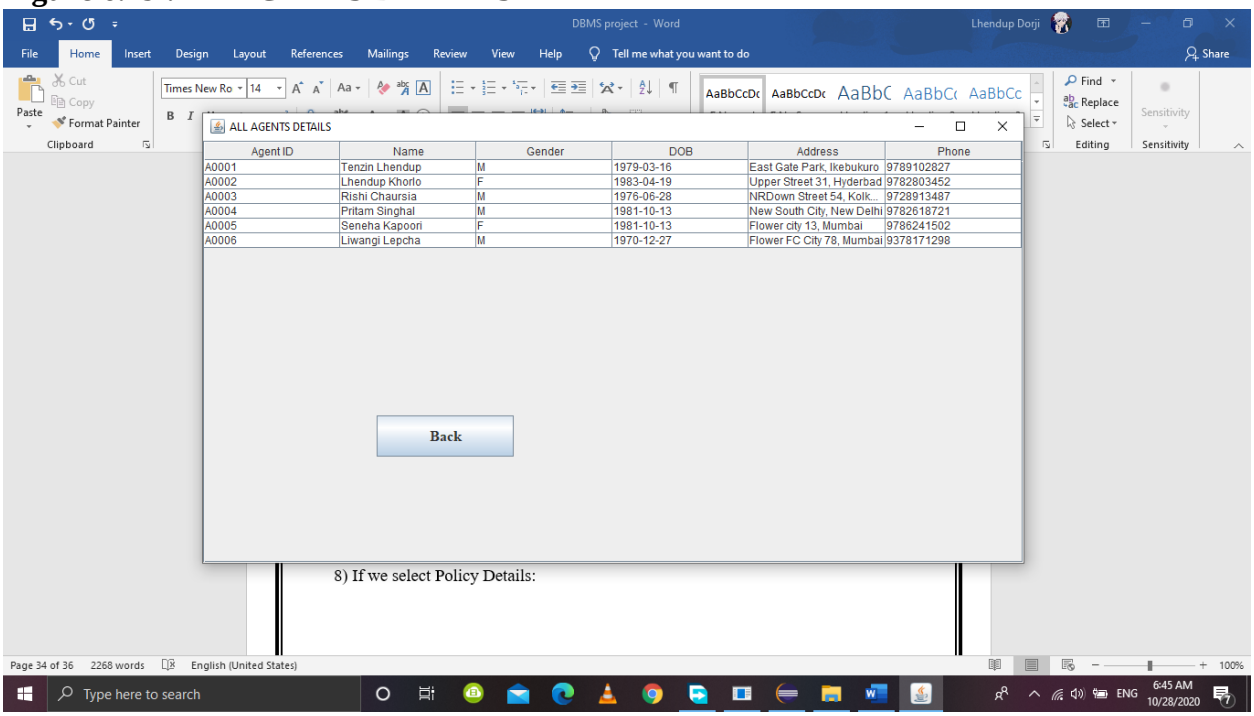
4) If we select All User Details and Back button can be used to go back to previous page:

Figure 6.18 : User Details



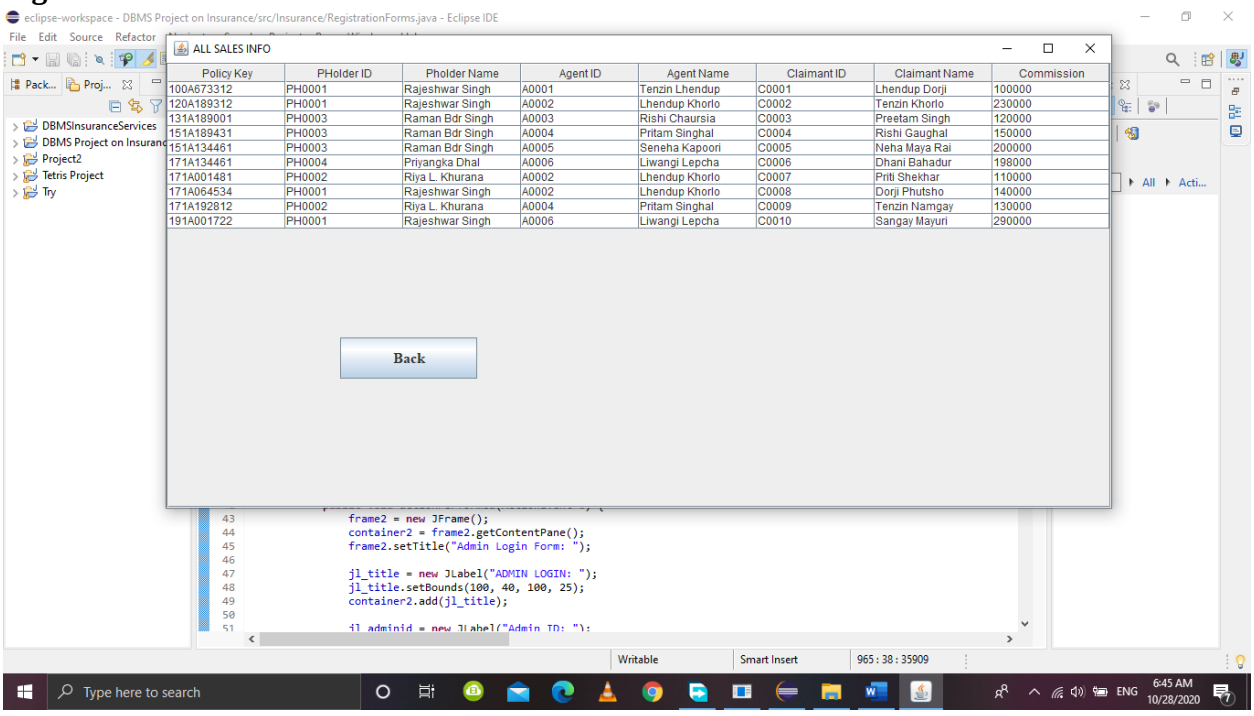
5) If we select All Agents Details:

Figure 6.19 : All AGENTS DETAILS



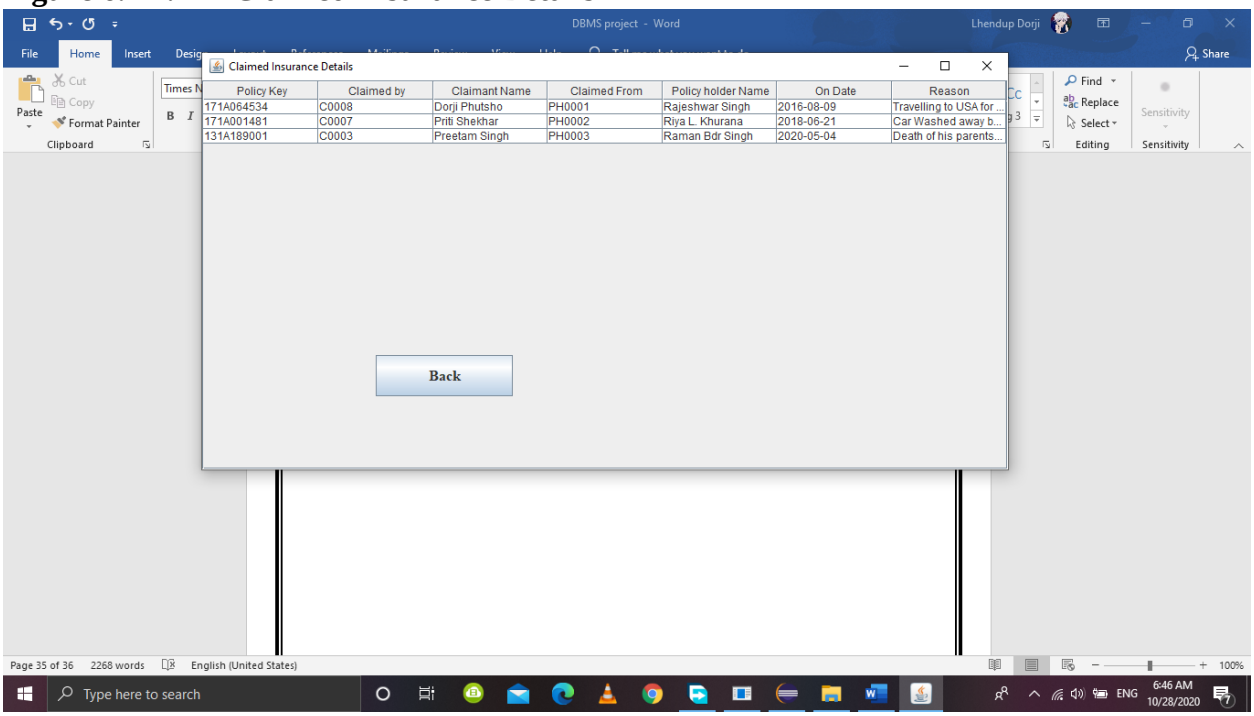
6) If we select All Sales Info:

Figure 6.20 : All Sales Info



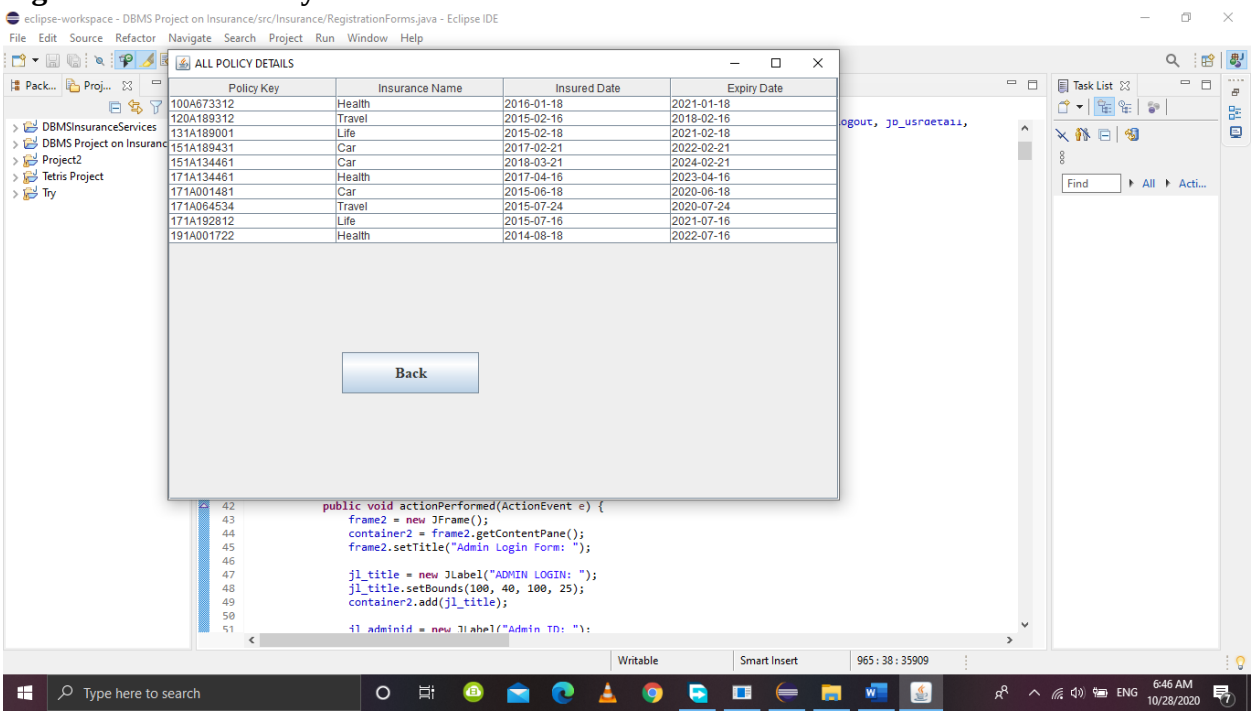
7) If we select All Claimed Insurance Info:

Figure 6.21 : All Claimed Insurance Details



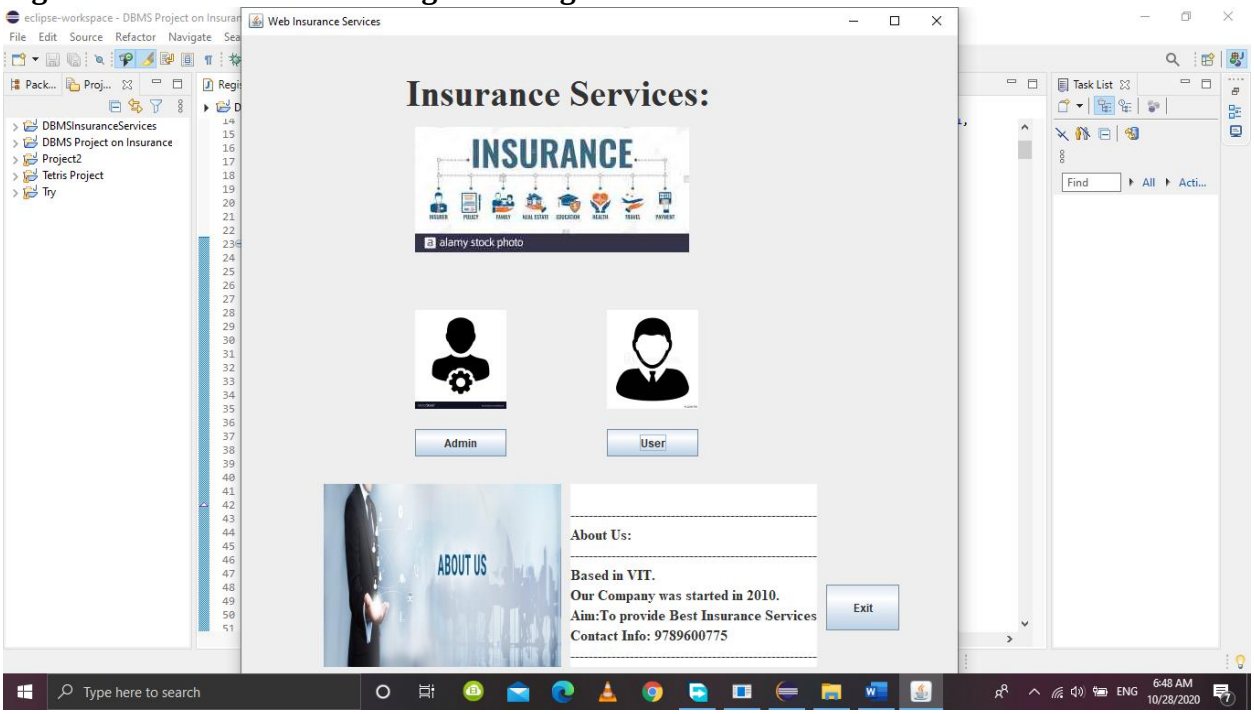
8) If we select Policy Details:

Figure 6.22 : All Policy Details



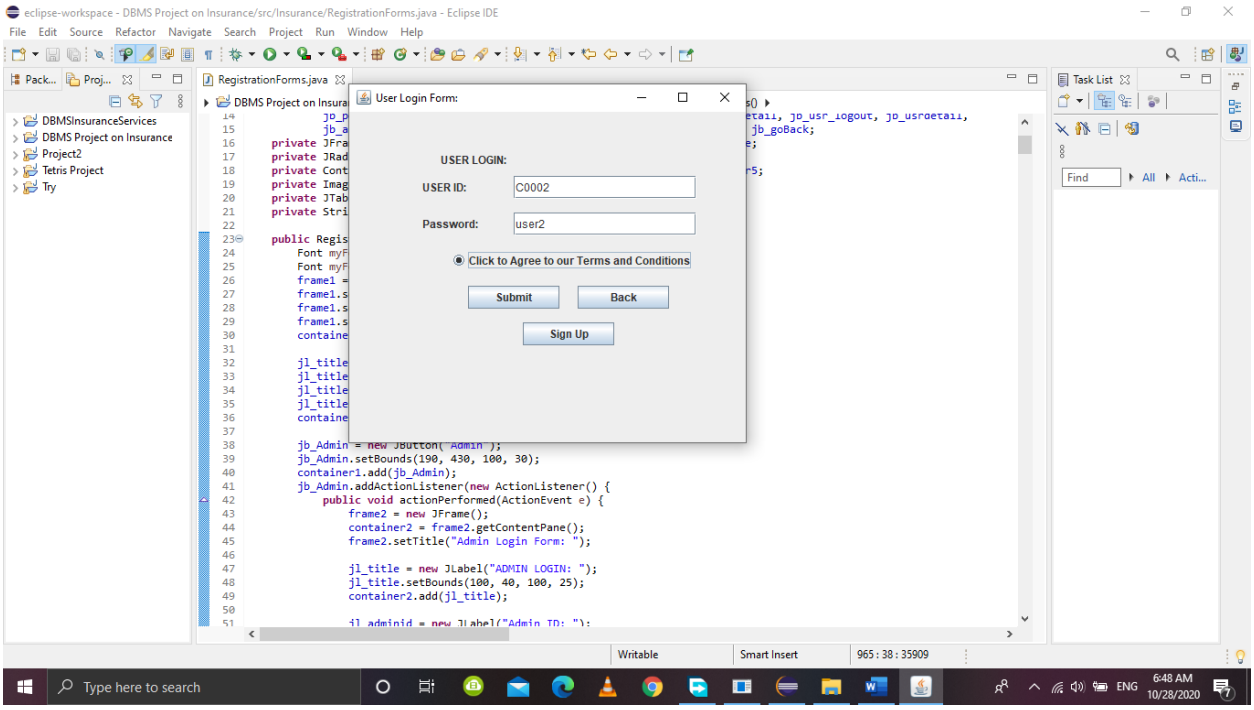
9) If we select Log Out we reach Home Page:

Figure 6.23 : Back to Home Page after Sign out from Admin



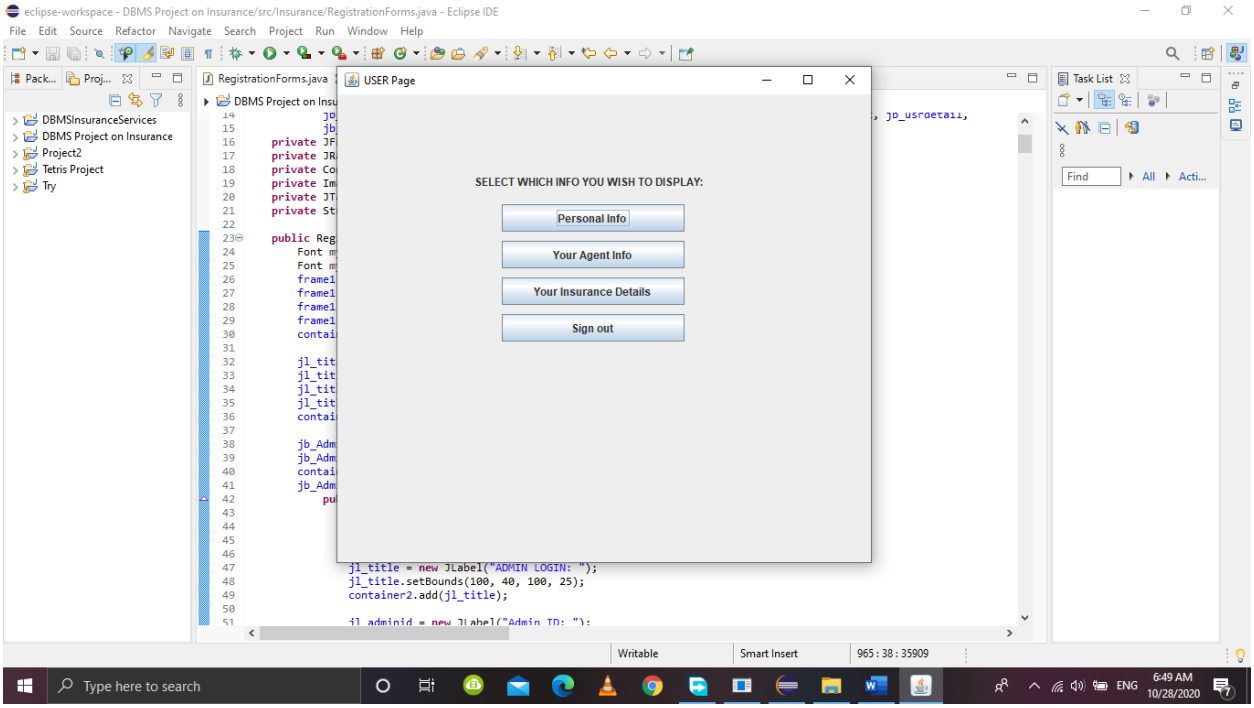
10) Selecting User :

Figure 6.24 : User Login Form



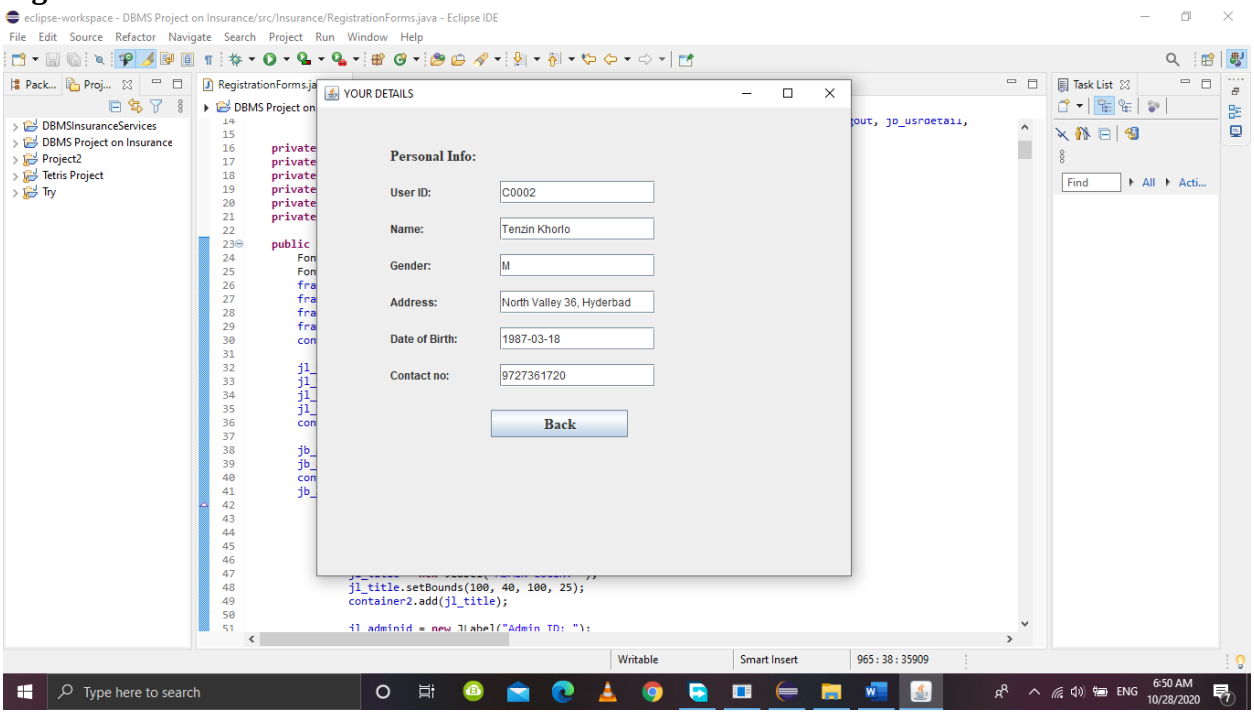
11) We get User Page after logging in:

Figure 6.25 : User Page



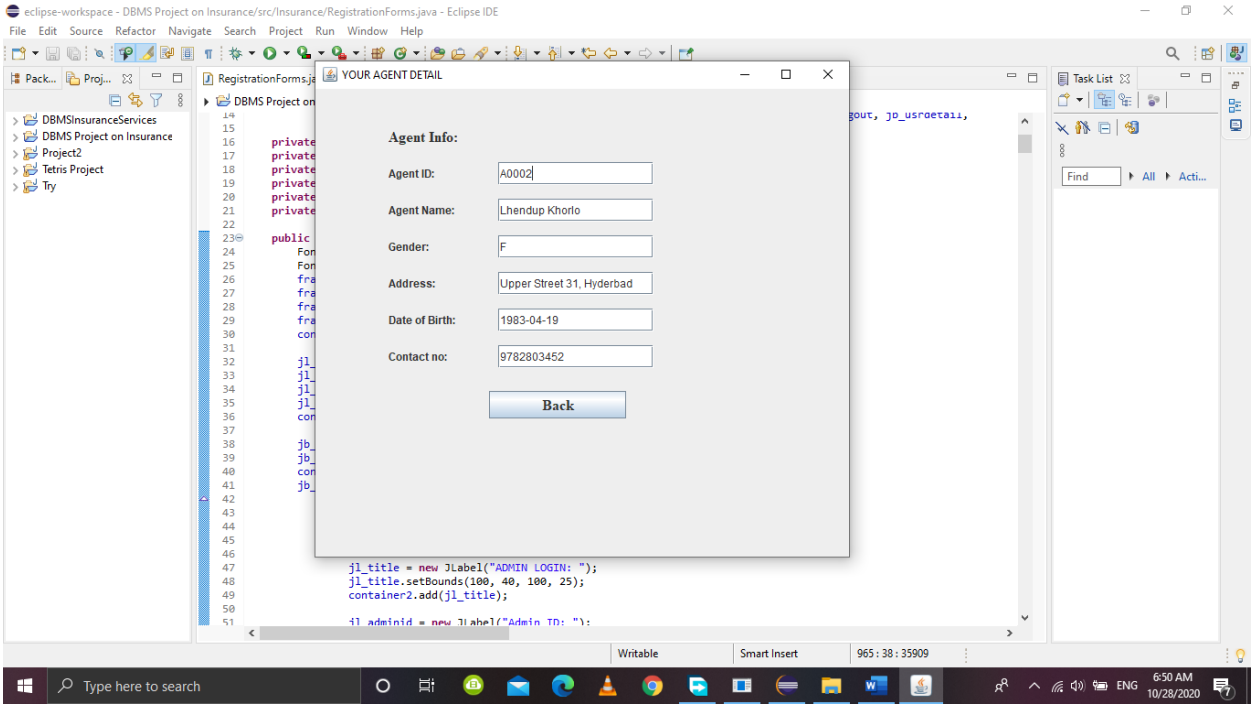
12) If we select Personal Info:

Figure 6.26 : User Personal Info



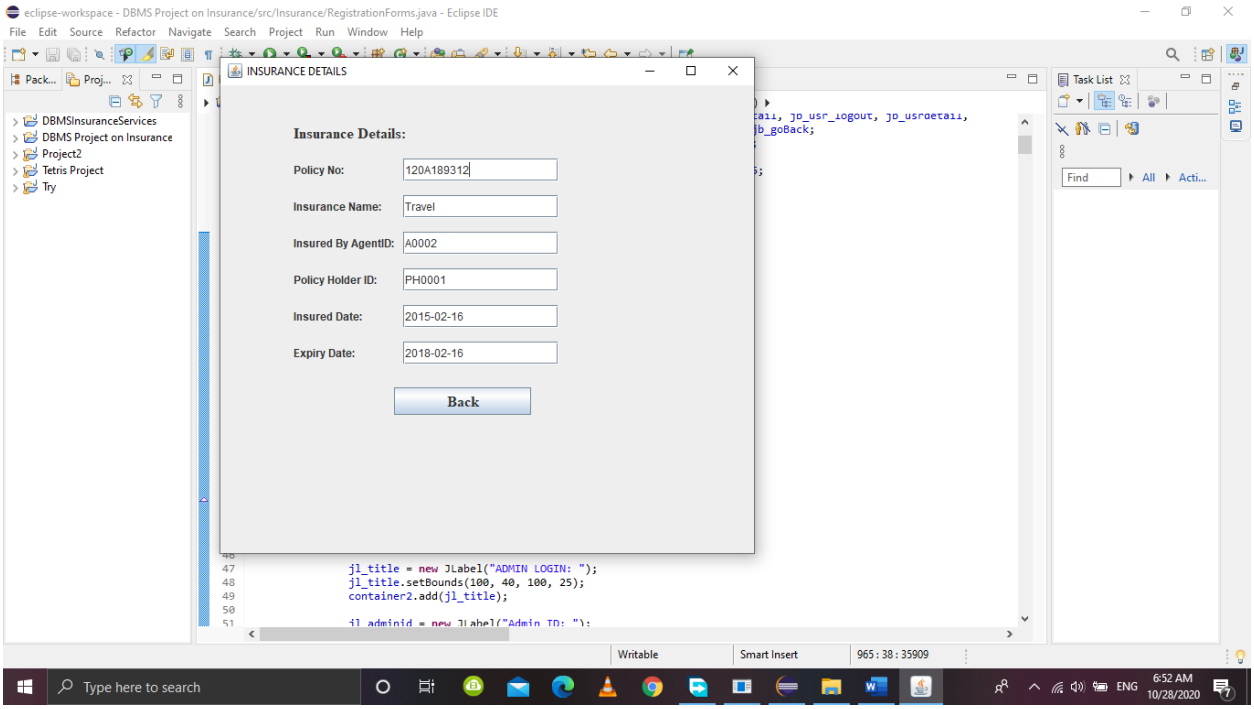
13) If we select Your Agent info:

Figure 6.27 : User's Agent Info



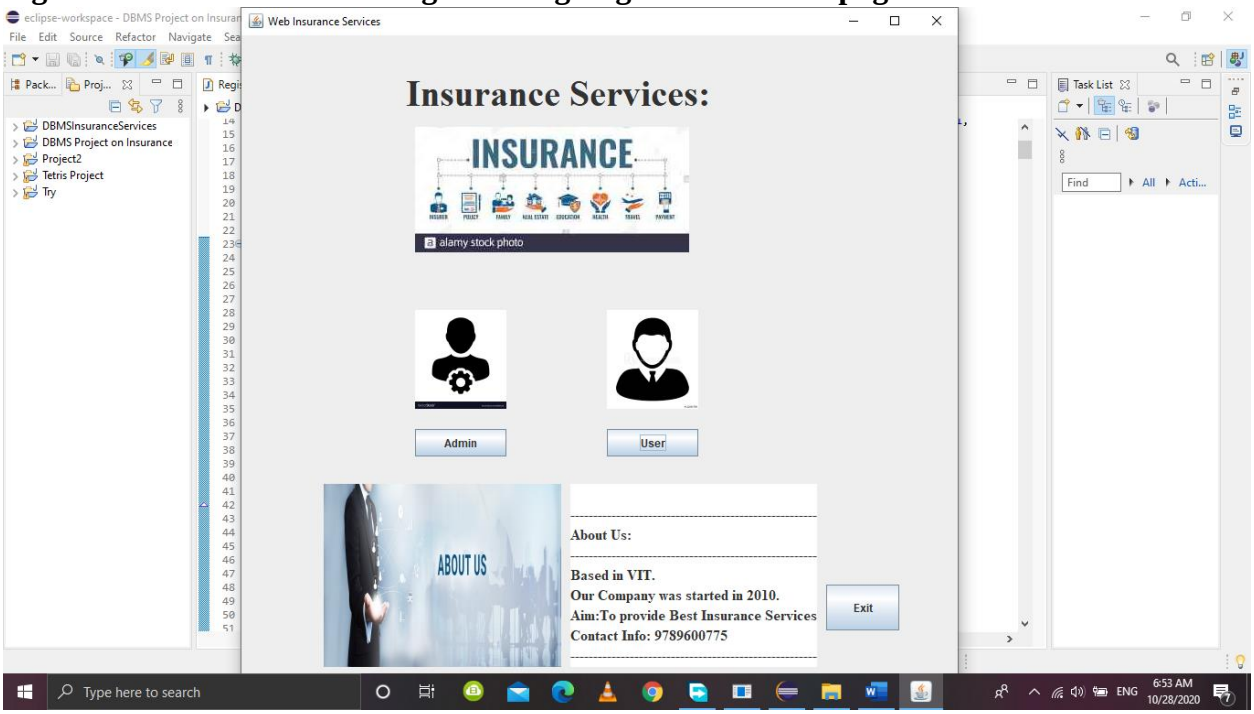
14) If we select Your Insurance Details:

Figure 6.28 : User's Insurance Details



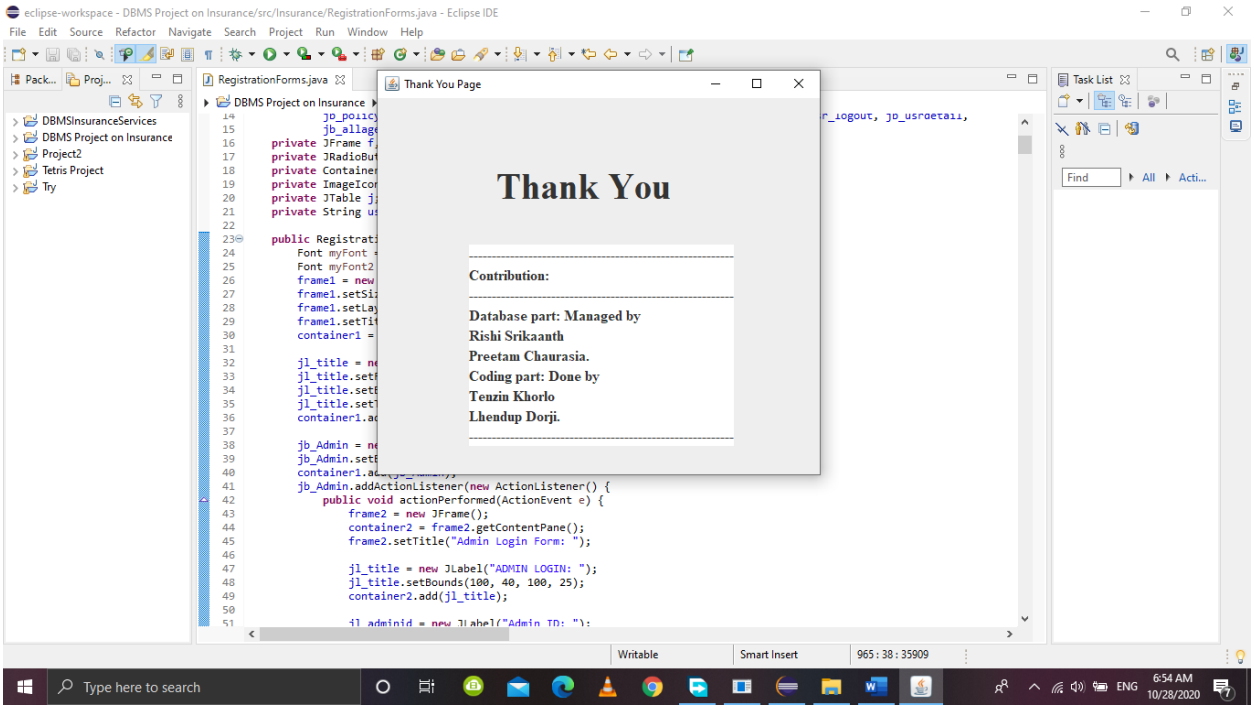
15) If we select Sign in button, we reach back to Home Page:

Figure 6.29 : Back to Home Page after signing out from User page



16) If we select Exit button, we get Thank you Page:

Figure 6.30 : Thank You Page and Our Contributions



CHAPTER 7

7.1 CONCLUSION

We have successfully done our project and our main objectives of keeping records of all admin, clients , agents and insurance details have been accomplished. The problems of insurance companies while trying to attend to their customers quickly and efficiently due to lack of adequate documentation of data can be easily solved by the web-based application of insurance management as we can see how efficiently we can store all the data. This application definitely ensures the insurance service communication around the globe as the customers and companies can interact virtually. On the top of that it also reduces data redundancy and facilitates faster searching of information without waste of people's time.

7.2 REFERENCES:

- **From App Academia:**

1) **Ele, Sylvester I., O. A. Ofem And Obono I. Ofem.*Department of Computer Science, University of Calabar, Nigeria.*"Implemetation of Futuristic Web-Based Application for Insurance Services", 2018.**

2) **Igwe E. Kelvin. Department of Computer Science, CEO of K-tech system, Ishiagu, Nigeria. "WEB BASED APPLICATION FOR INSURANCE SERVICES CASE STUDY OF THE INSURANCE COMPANY".**

- **From Google Chrome:**

3) Uploaded by Nnamdi Chimaobi. Department of science."Project on Web Based Application for Insurance Services", on Nov 22, 2012.

Available in: https://www.scribd.com/doc/114107737/Project-on-Web-Based-Application-for-Insurance-Services

4) "Web Based Application For Insurance Services Case Study Of The Insurance Company", by Esedebe Fidelia Ogechukwu, in July, 2013, a graduate of Computer Science of Information and Technology, Caritas University Amorji – Nike Enugu, Nigeria.
Availablein:https://www.sau.edu.ng/materials/WEB_BASED_APPLICATION_FOR_INSURANCE_SERVICES%20(1).pdf

5) "Teach yourself SQL in 21 days, second edition ," ,by Auwal Gene
Available in:
http://www.auwalgene.com/@mystudents/lecturenotes/teach_urself_sql.pdf

6) "Web Based Project Ideas & Topics" by nevon Projects, electronics |Software and mechanical kits.
Available in: https://nevonprojects.com/web-based-project-ideas-topics/

7) To create JTable in java:
https://docs.oracle.com/javase/tutorial/uiswing/components/table.html

8) to connect database with Java IDEs:
https://www.infoworld.com/article/3388036/what-is-jdbc-introduction-to-java-database-connectivity.html

9) To create JFrame, JTextArea, JTextField and all basics of java:
https://www.javatpoint.com/java-jframe

10) Java Tutorial:
https://www.youtube.com/watch?v=r59xYe3Vyks&t=2s

7.3 Appendices:

Appendix I: About Eclipse IDE

Eclipse is an integrated development environment (IDE) used in computer programming.[6] It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since the introduction of the OSGi implementation (Equinox) in version 3 of Eclipse, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse software development kit (SDK) is free and open-source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

Appendix II: About SQL

SQL -Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.

SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL),[b] a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages to utilize Edgar F. Codd's relational model. The model was described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

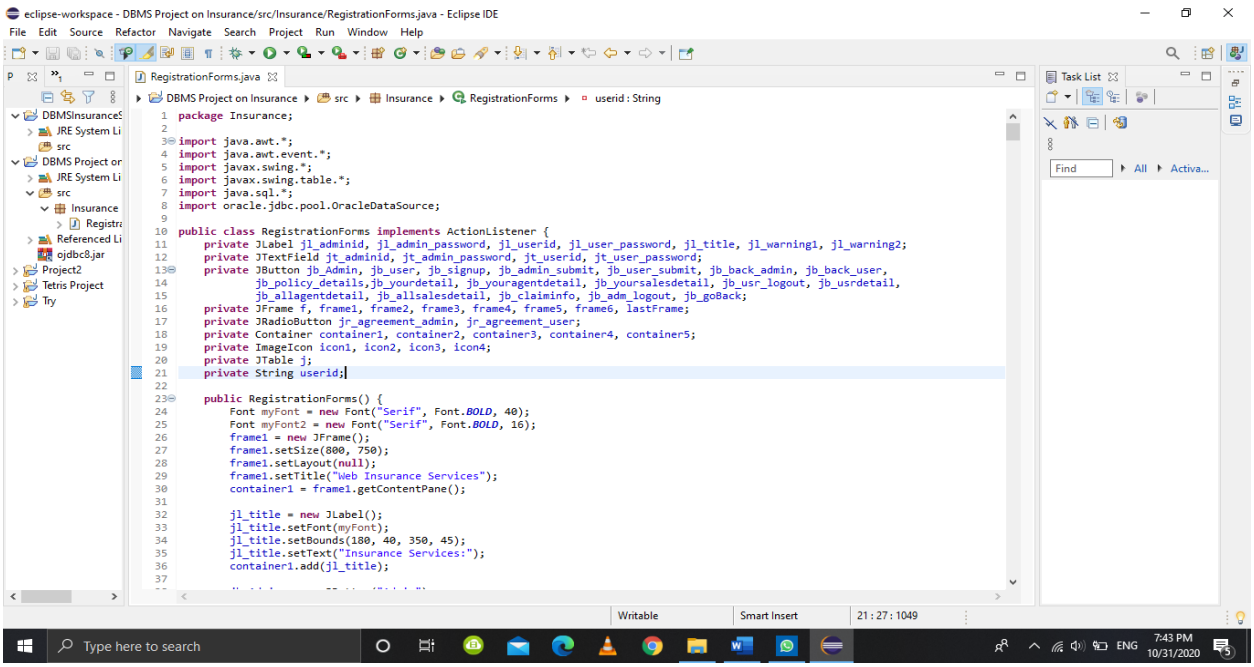
SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then the standard has been revised to include a larger set of features. Despite the existence of standards, most SQL code requires at least some changes before being ported to different database systems.

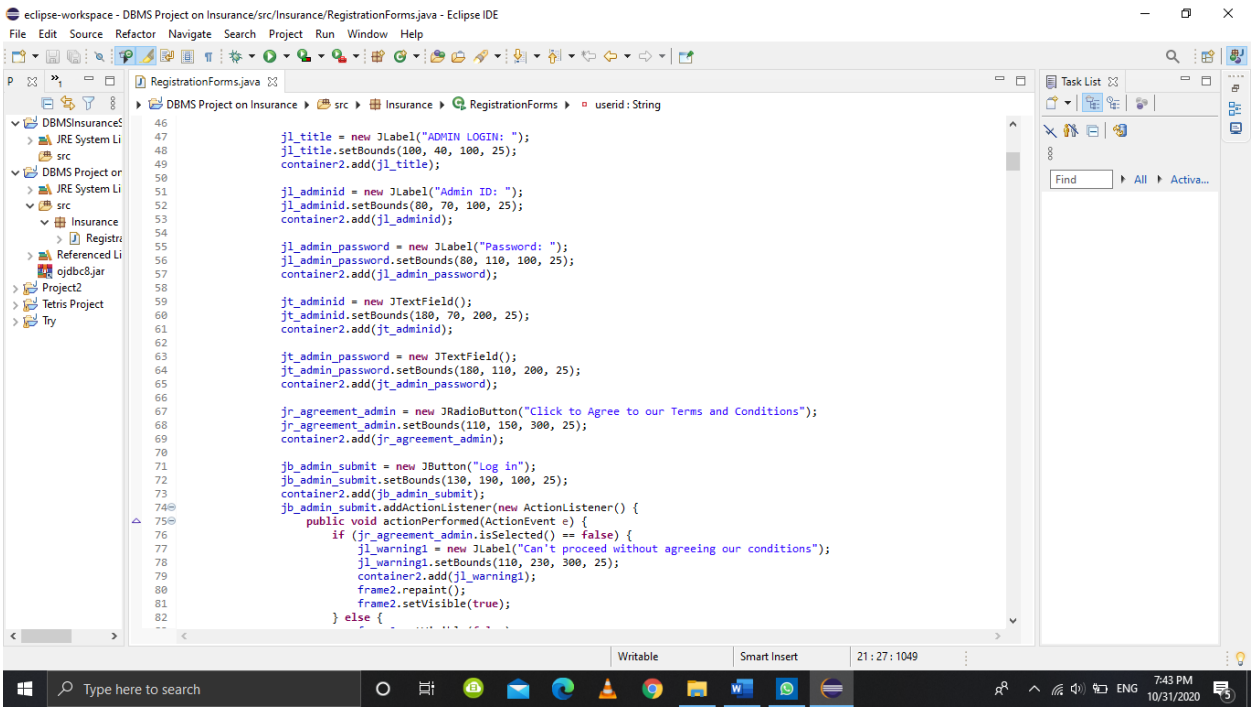
Appendix III: Codes

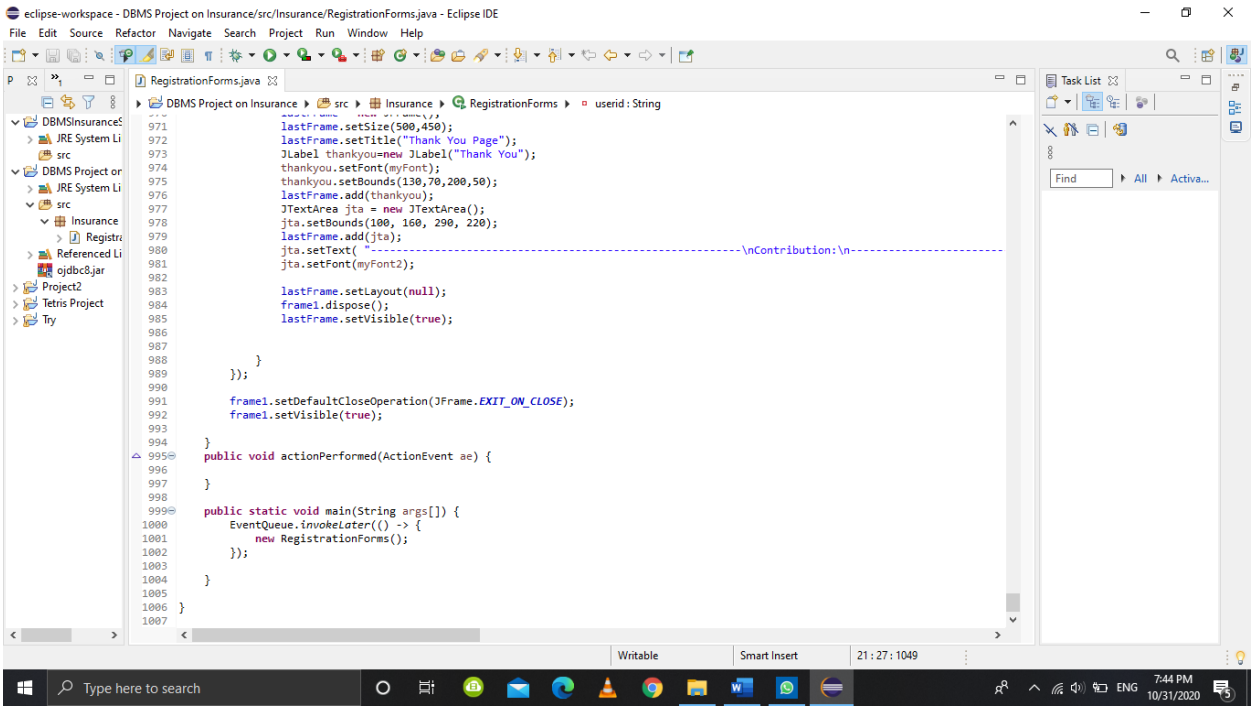
One can find the full codes for this application in Git HUB in the link given Below:

https://github.com/preetam077/DBMS-J-component.git

Here are some figures of implemented codes:




eclipse-workspace - DBMS Project on Insurance/src/Insurance/RegistrationForms.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RegistrationForms.java
DBMS Project on Insurance ▸ src ▸ Insurance ▸ RegistrationForms ▸ userid : String
Task List
Find All Activa...
DBMSInsurance
JRE System Li
src
DBMS Project or
JRE System Li
src
Insurance
Registra
Referenced Li
ojdbc8.jar
Project2
Tetris Project
Try
46
47 jl_title = new JLabel("ADMIN LOGIN: ");
48 jl_title.setBounds(100, 40, 100, 25);
49 container2.add(jl_title);
50
51 jl_adminid = new JLabel("Admin ID: ");
52 jl_adminid.setBounds(80, 70, 100, 25);
53 container2.add(jl_adminid);
54
55 jl_admin_password = new JLabel("Password: ");
56 jl_admin_password.setBounds(80, 110, 100, 25);
57 container2.add(jl_admin_password);
58
59 jt_adminid = new JTextField();
60 jt_adminid.setBounds(180, 70, 200, 25);
61 container2.add(jt_adminid);
62
63 jt_admin_password = new JTextField();
64 jt_admin_password.setBounds(180, 110, 200, 25);
65 container2.add(jt_admin_password);
66
67 jr_agreement_admin = new JRadioButton("Click to Agree to our Terms and Conditions");
68 jr_agreement_admin.setBounds(110, 150, 300, 25);
69 container2.add(jr_agreement_admin);
70
71 jb_admin_submit = new JButton("Log in");
72 jb_admin_submit.setBounds(130, 190, 100, 25);
73 container2.add(jb_admin_submit);
74 jb_admin_submit.addActionListener(new ActionListener() {
75 public void actionPerformed(ActionEvent e) {
76 if (jr_agreement_admin.isSelected() == false) {
77 jl_warning1 = new JLabel("Can't proceed without agreeing our conditions");
78 jl_warning1.setBounds(110, 230, 300, 25);
79 container2.add(jl_warning1);
80 frame2.repaint();
81 frame2.setVisible(true);
82 } else {
Writable
Smart Insert
21 : 27 : 1049
Type here to search
ENG
7:43 PM
10/31/2020


eclipse-workspace - DBMS Project on Insurance/src/Insurance/RegistrationForms.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
RegistrationForms.java
DBMS Project on Insurance ▸ src ▸ Insurance ▸ RegistrationForms ▸ userid : String
Task List
Find All Activa...
971 lastFrame.setSize(500,450);
972 lastFrame.setTitle("Thank You Page");
973 JLabel thankyou=new JLabel("Thank You");
974 thankyou.setFont(myFont);
975 thankyou.setBounds(130,70,200,50);
976 lastFrame.add(thankyou);
977 JTextArea jta = new JTextArea();
978 jta.setBounds(100, 160, 290, 220);
979 lastFrame.add(jta);
980 jta.setText("----------\nContribution:\n----------
981 jta.setFont(myFont2);
982
983 lastFrame.setLayout(null);
984 frame1.dispose();
985 lastFrame.setVisible(true);
986
987
988 }
989 });
990
991 frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
992 frame1.setVisible(true);
993
994 }
995 public void actionPerformed(ActionEvent ae) {
996
997 }
998
999 public static void main(String args[]) {
1000 EventQueue.invokeLater(() -> {
1001 new RegistrationForms();
1002 });
1003
1004 }
1005
1006 }
1007
Writable
Smart Insert
21 : 27 : 1049
Type here to search
ENG
7:44 PM
10/31/2020

THANK YOU