DATABASE DESIGN FOR HEALTH INSURANCE COMPANY

Pawan Dasharath Patil
pxp180029

Praveen Ramani
pxr170005

Rushikesh Kulkarni
rsk180001

Table of contents

Requirements

Health insurance:

Health insurance is a type of insurance coverage that covers the cost of an insured individual's medical and surgical expenses. Medical expenses can be very expensive and so health insurance covers the risk of incurring a medical expense and spreads the risk over a large number of people. The purpose of insurance, in general, is to protect people from financial losses and health insurance is a type of insurance which pertains to health related risks.

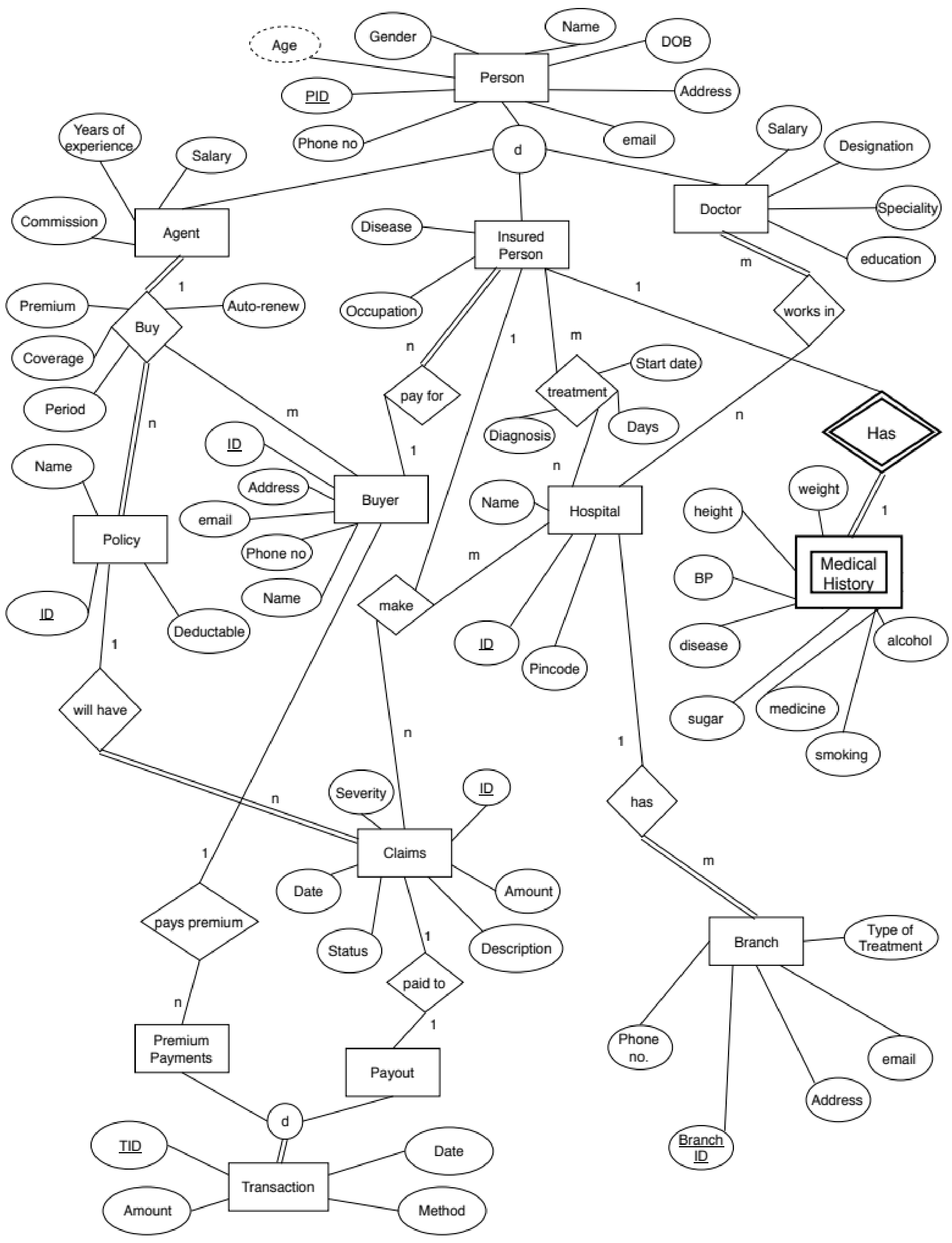
Important key words explained:

- **Deductible:** The amount you owe for covered health care services before your health insurance or plan begins to pay.
- **Co-payment:** An amount you pay as your share of the cost for a medical service or item, like a doctor's visit.
- **Co-insurance:** Your share of the cost for a covered health care service, usually calculated as a percentage (like 20%) of the allowed amount for the service.
- **Premium:** The amount you pay for your health insurance or plan each month.
- **Out-of-Pocket Maximum:** This is the most you’ll pay toward your healthcare in a given year. Let’s say you have an insurance plan with an out-of-pocket maximum of $5,000, once you’ve reached that amount, the insurance company picks up 100% of the costs for the rest of the plan year (excluding co-pays).
- **Network:** The doctors, hospitals, and suppliers your health insurer has contracted with to deliver health care services to their members.
- **Annual Coverage Limit:** This is the maximum amount health insurance can pay for you in a given year. Any expenses above this limit should be paid by the individual person.

Working of Health Insurance Company:

Health Insurance Company has different types of policies for buyers. Each policy has specific benefits, conditions and coverage limit. Buyers buy these policies and for each policy pay certain amount per month as a premium. Company uses this money to pay medical expenses of insured persons. Each company has many policy agents to handle policies of buyers. These agents ask for medical history of insured persons, take their information and suggest suitable policies. If an insured person undergoes medical illness or accident, insurance company pays the amount after out of pocket maximum limit is reached. For amount to be paid by the company patient must get treatment only in a hospital which is in insurance company’s network. These hospitals submit medical claims of patients to the insurance company. Medical claims have information of patient, detail description of illness, total expenses incurred and medical reports. Insurance company checks and approves medical claims. After approval of claims company pays medical expenses to the hospital.

EER Diagram:



Person
Age
Gender
Name
DOB
PID
Address
Phone no
email
d
Agent
Years of experience
Salary
Commission
Doctor
Salary
Designation
Speciality
education
Insured Person
Disease
Occupation
Buy
Premium
Coverage
Period
Auto-renew
works in
pay for
treatment
Start date
Days
Diagnosis
Has
Buyer
ID
Address
email
Phone no
Name
Hospital
Name
ID
Pincode
Medical History
height
weight
BP
disease
sugar
medicine
smoking
alcohol
Policy
Name
ID
Deductable
make
will have
Claims
Severity
ID
Date
Amount
Status
Description
has
Branch
Type of Treatment
Phone no.
email
Address
Branch ID
pays premium
paid to
Premium Payments
Payout
d
Transaction
TID
Amount
Date
Method
1
n
m

The EER Diagram can be Summarized as –

1) There is a ternary relationship between Agent , Buyer and Policy where Agent(One) and Buyer(s) (1…N) are required to buy the Policy(1…N).
2) One Policy will have many claims and one claim will have only one buyer (1:N).
3) A buyer can pay many premiums and one premium can only be paid by one buyer (1:N).
4) One claim is paid to one Payout and one Payout refers to only one claim (1:1).
5) One buyer has many insured persons and one insured person pays for one buyer.(1:N).
6) There is a ternary relationship between Insured Person , Hospital and Claim where Insured Person (One) and Hospital(s) (1…N) are required to make the Claim(s) (1…N).
7) One Hospital has many Doctors and One Doctor works in many hospitals.(M:N).
8) One Hospital has many branches and One branch represents one hospital (1:1).
9) One Insured Person can get Treatment from many hospitals. One hospital can do treatment for many Insured Persons.
10) One Insured Person has a Medical History. One Medical history is of one person.(1:1).

MAPPING OF ERD IN RELATIONAL SCHEMA:

1. POLICY

<u>PolicyID</u>	Name	Co-insurance	Description	Deductible	Out of Pocket	Annual Coverage

- **Primary Key** : PolicyID
- **Foreign Keys** : None

2. BUYER

<u>BuyerID</u>	Name	Address

- **Primary Key** : BuyerID
- **Foreign Keys** : None

3. PERSON

<u>PersonID</u>	Name	Gender	Age	DOB	Address	IPflag	Agflag	Dflag	Salary

Eductaion	Experience	Occupation	Designation	Commission	Married	**BuyerID**

- **Primary Key** : PersonID
- **Foreign Keys** : FOREIGN KEY (BuyerID) REFERENCES BUYER(BuyerID)

4. BUYER POLICIES

Auto-renew	Premium	Period	**<u>Agent ID</u>**	**<u>BuyerID</u>**	**<u>PolicyID</u>**

- **Primary Key** : AgentID, BuyerID, PolicyID
- **Foreign Keys** : FOREIGN KEY (AgentID) REFERENCES PERSON(PersonID), FOREIGN KEY (BuyerID) REFERENCES BUYER(BuyerID), FOREIGN KEY (PolicyID) REFERENCES POLICY(PolicyID)

5. HOSPITAL

<u>HosptitalID</u>	Name

- **Primary Key** : HospitalID
- **Foreign Keys** : None

6. BRANCH

<u>BranchID</u>	Address	**HospitalID**

- **Primary Key** : BranchID
- **Foreign Keys** : FOREIGN KEY (HospitalID) REFERENCES HOSPITAL(HospitalID)

7. WORKS-IN

HospitalID	**DoctorID**

- **Primary Key** : HospitalID, DoctorID
- **Foreign Keys** : FOREIGN KEY (HospitalID) REFERENCES HOSPITAL(HospitalID), FOREIGN KEY (DoctorID) REFERENCES PERSON(PersonID)

8. MEDICAL HISTORY

InsuredPersonID	Height	Weight	Smoking	Alcohol consumption	Injuries	Disease

Sugery Birth place

- **Primary Key** : InsuredPersonID
- **Foreign Keys** : FOREIGN KEY (InsuredPersonID) REFERENCES PERSON(PersonID)

9. TREATMENT

PersonID	**BranchID**	Days	Start date	Diagnosis

- **Primary Key** : PersonID, BranchID
- **Foreign Keys** : FOREIGN KEY (PersonID) REFERENCES PERSON(PersonID), FOREIGN KEY (BranchID) REFERENCES BRANCH(BranchID)

10. CLAIM

ClaimID	Date	Description	Amount	Status	Severity	**PolicyID**

- **Primary Key** : ClaimID
- **Foreign Keys** : FOREIGN KEY (PolicyID) REFERENCES POLICY(PolicyID)

11. CLAIM-SUBMISSION

ClaimID	**HospitalID**	**InsuredPersonID**

- **Primary Key** : ClaimID, HospitalID, InsuredPersonID

- **Foreign Keys** : FOREIGN KEY (ClaimID) REFERENCES CLAIM(ClaimID), FOREIGN KEY (HospitalID) REFERENCES HOSPITAL(HospitalID), FOREIGN KEY (InsuredPersonID) REFERENCES PERSON(InsuredPersonID).

12. TRANSACTION

TransactionID	Amount	Date	Method

- **Primary Key** : TransactionID
- **Foreign Keys** : None

13. PREMIUIM-PAYMENTS

TransactionID	**BuyerID**

- **Primary Key** : TransactionID, BuyerID
- **Foreign Keys** : FOREIGN KEY (TransactionID) REFERENCES TRANSACTION(TransactionID), FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID)

14. CLAIMS-PAYMENT

TransactionID	**ClaimID**

- **Primary Key** : TransactionID, ClaimID
- **Foreign Keys** : FOREIGN KEY (TransactionID) REFERENCES TRANSACTION(TransactionID), FOREIGN KEY (ClaimID) REFERENCES Claim(ClaimID)

15. BUYER CONTACT

BuyerID	Phone	Email

- **Primary Key** : BuyerID, Phone, Email
- **Foreign Keys** : FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID)

16. PERSON CONTACT

PersonID	Phone	Email

- **Primary Key** : PersonID, Phone, Email
- **Foreign Keys** : FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

17. **HOSPITAL CONTACT**

BranchID	Phone	Email

- **Primary Key** : BranchID, Phone, Email
- **Foreign Keys** : FOREIGN KEY (BranchID) REFERENCES BRANCH(BranchID)

NORMALIZATION:

The following Functional Dependencies exists in the relational schema –

- **POLICY** {PolicyID -> Name, Co-insurance, Description, Deductible, Out of Pocket, Annual Coverage}
- **BUYER** {BuyerID -> Name, Address}
- **PERSON** {PersonID -> Name, Gender, Age, D.O.B, Address, IPflag, Agflag, Dflag, Salary, Education, Experience, Occupation, Designation, Commission, Married, BuyerID}
- **BUYER POLICIES** {AgentID, PolicyID, BuyerID -> Auto-renew, Premium,Period}
- **HOSPITAL** {HosptalID -> Name }
- **BRANCH** {BranchID -> HospitalID, Address, Street, City, PinCode, State, Country}
- **MEDICAL HISTORY** {BranchID -> Height, Weight, Smoking, Alcohol consumption, Injuries, Disease, Medicines, Sugery, Birth Place}
- **TREATMENT** {PersonID, BranchID -> Days, Start Date, Diagnosis}
- **CLAIM** {ClaimID -> Date, Description, Amount, Status, Severity}
- **TRANSACTION** {TransactionID -> Amount, Date, Method}

The above functional dependencies are in third normal form since there is no partial dependency, transitive dependency.

SQL statements to create tables in database and add constraints:

```
CREATE TABLE "RSK180001"."BOUGHT_POLICIES"
 (      "POLICY_ID" NUMBER(20,0) NOT NULL ENABLE,
        "BUYER_ID" NUMBER(20,0) NOT NULL ENABLE,
        "AGENT_ID" NUMBER(20,0) NOT NULL ENABLE,
        "AUTO_RENEW" VARCHAR2(3 BYTE) DEFAULT 'NO',
        "PREMIUM" NUMBER(5,0),
        "PERIOD" NUMBER(3,0),
         CONSTRAINT "BOUGHT_POLICIES_PK" PRIMARY KEY
         ("POLICY_ID", "BUYER_ID", "AGENT_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE,
         CONSTRAINT "FK1_POLICY_ID" FOREIGN KEY ("POLICY_ID")
         REFERENCES "RSK180001"."POLICY" ("POLICY_ID") ENABLE,
         CONSTRAINT "FK2_BUYER_ID" FOREIGN KEY ("BUYER_ID")
         REFERENCES "RSK180001"."BUYER" ("BUYER_ID") ENABLE,
         CONSTRAINT "FK3_AGENT_SSN" FOREIGN KEY ("AGENT_ID")
         REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE
 );



CREATE TABLE "RSK180001"."BUYER"
 (      "BUYER_ID" NUMBER(10,0) NOT NULL ENABLE,
        "NAME" VARCHAR2(200 BYTE),
         CONSTRAINT "BUYER_PK" PRIMARY KEY ("BUYER_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE
 );


CREATE TABLE "RSK180001"."BUYER_ADDRESS"
 (      "ADDRESS_ID" NUMBER(10,0) NOT NULL ENABLE,
        "ADDRESS" VARCHAR2(200 BYTE),
        "STREET" VARCHAR2(200 BYTE),
        "CITY" VARCHAR2(200 BYTE),
        "STATE" VARCHAR2(200 BYTE) NOT NULL ENABLE,
        "PINCODE" NUMBER(10,0) NOT NULL ENABLE,
        "COUNTRY" VARCHAR2(200 BYTE) NOT NULL ENABLE,
        "BUYER_ID" NUMBER(10,0) NOT NULL ENABLE,
         CONSTRAINT "BUYER_ADDRESS_PK" PRIMARY KEY
         ("ADDRESS_ID", "BUYER_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
         STATISTICS
         TABLESPACE "USERS"  ENABLE
 );
```

```
CREATE TABLE "RSK180001"."CLAIM"
 (      "CLAIM_ID" NUMBER(20,0) NOT NULL ENABLE,
        "SUBMISSION_DATE" DATE,
        "AMOUNT" NUMBER(10,0),
        "STATUS" VARCHAR2(20 BYTE),
        "SEVERITY" VARCHAR2(20 BYTE),
        "DESCRIPTION" VARCHAR2(100 BYTE),
         CONSTRAINT "CLAIM_PK" PRIMARY KEY ("CLAIM_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE
 );
```

```
CREATE TABLE "RSK180001"."CLAIM_SUBMISSION"
 (      "CLAIM_ID" NUMBER(20,0) NOT NULL ENABLE,
        "HOSPITAL_BRANCH_ID" NUMBER(10,0) NOT NULL ENABLE,
        "INSURED_PERSON_ID" NUMBER(20,0) NOT NULL ENABLE,
        CONSTRAINT "CLAIM_SUBMISSION_PK" PRIMARY KEY
        ("CLAIM_ID", "HOSPITAL_BRANCH_ID", "INSURED_PERSON_ID")
        USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
        TABLESPACE "USERS"  ENABLE,
        CONSTRAINT "FK1_CLAIM_ID" FOREIGN KEY ("CLAIM_ID")
        REFERENCES "RSK180001"."CLAIM" ("CLAIM_ID") ENABLE,
        CONSTRAINT "FK2_HOSPITAL_BRANCH_ID" FOREIGN KEY
        ("HOSPITAL_BRANCH_ID")
        REFERENCES "RSK180001"."HOSPITAL_BRANCH" ("BRANCH_ID")
        ENABLE,
        CONSTRAINT "FK3_PERSON_ID" FOREIGN KEY
        ("INSURED_PERSON_ID")
        REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE
 );
```

```
CREATE TABLE "RSK180001"."HOSPITAL"
 (      "ID" NUMBER(20,0) NOT NULL ENABLE,
        "NAME" VARCHAR2(100 BYTE),
         CONSTRAINT "HOSPITAL_PK" PRIMARY KEY ("ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
         STATISTICS
         TABLESPACE "USERS"  ENABLE
 );
```

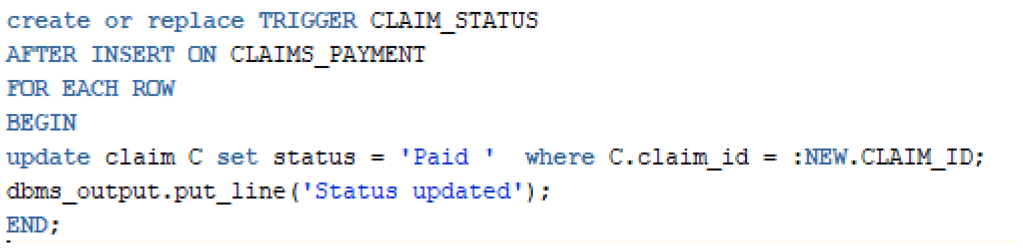
```
CREATE TABLE "RSK180001"."HOSPITAL_BRANCH"
 (      "BRANCH_ID" NUMBER(10,0) NOT NULL ENABLE,
        "HOSPITAL_ID" NUMBER(20,0),
          "ADDRESS" VARCHAR2(200 BYTE),
          "STREET" VARCHAR2(200 BYTE),
          "CITY" VARCHAR2(100 BYTE),
          "PINCODE" NUMBER(10,0) NOT NULL ENABLE,
          "STATE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
          "COUNTRY" VARCHAR2(100 BYTE) NOT NULL ENABLE,
           CONSTRAINT "HOSPITAL_BRANCH_PK" PRIMARY KEY
           ("BRANCH_ID")
            USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
            TABLESPACE "USERS"  ENABLE
 ) ;




CREATE TABLE "RSK180001"."MEDICAL_HISTORY"
 (        "PSSN" NUMBER(20,0) NOT NULL ENABLE,
          "NAME" VARCHAR2(50 BYTE),
          "WEIGHT" NUMBER(3,0),
          "SMOKE" VARCHAR2(3 BYTE),
          "ALCOHOLCONSUMPTION" VARCHAR2(3 BYTE),
          "MEDICINES" VARCHAR2(100 BYTE),
          "BIRTHPLACE" VARCHAR2(100 BYTE),
          "INJURIES" VARCHAR2(100 BYTE),
          "DISEASE" VARCHAR2(100 BYTE),
           CONSTRAINT "MEDICAL_HISTORY_PK" PRIMARY KEY ("PSSN")
           USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
           TABLESPACE "USERS"  ENABLE,
           CONSTRAINT "FK1_PERSON_SSN" FOREIGN KEY ("PSSN")
           REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE
 );


CREATE TABLE "RSK180001"."PERSON"
 (        "SSN" NUMBER(20,0) NOT NULL ENABLE,
          "NAME" VARCHAR2(200 BYTE) NOT NULL ENABLE,
          "GENDER" VARCHAR2(10 BYTE),
          "AGE" NUMBER(3,0),
          "DOB" DATE NOT NULL ENABLE,
          "SALARY" NUMBER(10,0),
          "EDUCATION" VARCHAR2(20 BYTE),
          "OCCUPATION" VARCHAR2(20 BYTE),
          "COMISSION" NUMBER(10,0),
          "MARRIED" VARCHAR2(3 BYTE),
          "YEARS_OF_EXPERIENCE" NUMBER(3,0),
          "IPFLAG" NUMBER(1,0),
          "DFLAG" NUMBER(1,0),
          "AFLAG" VARCHAR2(1 BYTE),
        "BUYER_ID" NUMBER(10,0),
         CONSTRAINT "PERSON_PK" PRIMARY KEY ("SSN")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE,
         CONSTRAINT "FK1_BUYER_ID" FOREIGN KEY ("BUYER_ID")
         REFERENCES "RSK180001"."BUYER" ("BUYER_ID") ENABLE
);


CREATE TABLE "RSK180001"."PERSON_ADDRESS"
(       "ADDRESS_ID" NUMBER(10,0) NOT NULL ENABLE,
        "ADDRESS" VARCHAR2(100 BYTE),
        "STREET" VARCHAR2(100 BYTE),
        "CITY" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "PINCODE" NUMBER(10,0) NOT NULL ENABLE,
        "STATE" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "COUNTRY" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "PERSON_ID" NUMBER(20,0) NOT NULL ENABLE,
         CONSTRAINT "PERSON_ADDRESS_PK" PRIMARY KEY
         ("ADDRESS_ID", "PERSON_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
         STATISTICS
         TABLESPACE "USERS"  ENABLE
);



CREATE TABLE "RSK180001"."POLICY"
(       "POLICY_ID" NUMBER(20,0) NOT NULL ENABLE,
        "NAME" VARCHAR2(50 BYTE),
        "DESCRIPTION" VARCHAR2(100 BYTE),
        "DEDUCTIBLE" NUMBER(10,0),
        "OUTOFPOCKET" NUMBER(10,0),
        "CO_INSURANCE" NUMBER(10,0),
        "ANNUAL_COVERAGE" NUMBER(10,0),
         CONSTRAINT "POLICY_PK" PRIMARY KEY ("POLICY_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE
);



CREATE TABLE "RSK180001"."WORKS_IN"
(       "HOSPITAL_ID" NUMBER(20,0) NOT NULL ENABLE,
        "DOCTOR_ID" NUMBER(20,0) NOT NULL ENABLE,
         CONSTRAINT "WORKS_IN_PK" PRIMARY KEY ("HOSPITAL_ID",
        "DOCTOR_ID")
         USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
         TABLESPACE "USERS"  ENABLE,
         CONSTRAINT "FK1_HOSPITAL_ID" FOREIGN KEY ("HOSPITAL_ID")
```

```
          REFERENCES "RSK180001"."HOSPITAL" ("ID") ENABLE,
          CONSTRAINT "FK2" FOREIGN KEY ("DOCTOR_ID")
          REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE
 );


CREATE TABLE "RSK180001"."TREAMENT"
 (       "PERSON_ID" NUMBER(20,0) NOT NULL ENABLE,
         "HSP_BRANCH_ID" NUMBER(10,0) NOT NULL ENABLE,
         "DAYS" NUMBER(3,0),
         "START_DATE" DATE,
         "DIAGNOSIS" VARCHAR2(100 BYTE),
          CONSTRAINT "TREAMENT_PK" PRIMARY KEY ("PERSON_ID",
         "HSP_BRANCH_ID")
          USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
          STATISTICS
          TABLESPACE "USERS"  ENABLE,
          CONSTRAINT "FK1_PERSON_ID" FOREIGN KEY ("PERSON_ID")
          REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE,
          CONSTRAINT "FK2_BRANCH_ID" FOREIGN KEY
          ("HSP_BRANCH_ID")
           REFERENCES "RSK180001"."HOSPITAL_BRANCH" ("BRANCH_ID")
           ENABLE
 ) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS" ;



CREATE TABLE "RSK180001"."TRANSACTION"
 (       "TRANSACTION_ID" NUMBER(20,0) NOT NULL ENABLE,
         "AMOUNT" NUMBER(20,0),
         "TRANSACTION_DATE" DATE,
         "PAYMENT_METHOD" VARCHAR2(20 BYTE),
          CONSTRAINT "TRANSACTION_PK" PRIMARY KEY
          ("TRANSACTION_ID")
          USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
          STATISTICS
          TABLESPACE "USERS"  ENABLE
 ) SEGMENT CREATION DEFERRED
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS" ;



CREATE TABLE "RSK180001"."PREMIUM_PAYMENTS"
 (       "TRANSACTION_ID" NUMBER(20,0) NOT NULL ENABLE,
         "BUYER_ID" NUMBER(20,0) NOT NULL ENABLE,
          CONSTRAINT "FK1_TRANSACTION_ID" FOREIGN KEY
          ("TRANSACTION_ID")
          REFERENCES "RSK180001"."TRANSACTION" ("TRANSACTION_ID")
          ENABLE,
          CONSTRAINT "FK2_PAID_BY" FOREIGN KEY ("BUYER_ID")
          REFERENCES "RSK180001"."BUYER" ("BUYER_ID") ENABLE
          ) SEGMENT CREATION DEFERRED
         PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
         NOCOMPRESS LOGGING
         TABLESPACE "USERS" ;


 CREATE TABLE "RSK180001"."CLAIMS_PAYMENT"
  (      "CLAIM_ID" NUMBER(20,0) NOT NULL ENABLE,
         "TRANSACTION_ID" NUMBER(20,0) NOT NULL ENABLE,
          CONSTRAINT "CLAIMS_PAYMENT_PK" PRIMARY KEY
          ("TRANSACTION_ID", "CLAIM_ID")
          USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
          STATISTICS
          TABLESPACE "USERS"  ENABLE,
          CONSTRAINT "FK1_CLAIM_PAYMENT" FOREIGN KEY
          CLAIM_ID")
          REFERENCES "RSK180001"."CLAIM" ("CLAIM_ID") ENABLE,
          CONSTRAINT "FK2_TRANS_ID" FOREIGN KEY
          ("TRANSACTION_ID")
          REFERENCES "RSK180001"."TRANSACTION" ("TRANSACTION_ID")
          ENABLE
  ) SEGMENT CREATION DEFERRED
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
 TABLESPACE "USERS" ;



 CREATE TABLE "RSK180001"."HOSPITAL_CONTACT"
  (      "HOSPITAL_BRANCH_ID" NUMBER(20,0) NOT NULL ENABLE,
         "PHONE" NUMBER(10,0),
         "EMAIL" VARCHAR2(20 BYTE),
          CONSTRAINT "HOSPITAL_CONTACT_PK" PRIMARY KEY
          ("HOSPITAL_BRANCH_ID")
          USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
          STATISTICS
          TABLESPACE "USERS"  ENABLE,
          CONSTRAINT "FK_BRANCH_CONTACT" FOREIGN KEY
          ("HOSPITAL_BRANCH_ID")
           REFERENCES "RSK180001"."HOSPITAL_BRANCH" ("BRANCH_ID")
          ENABLE
  ) SEGMENT CREATION DEFERRED
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
```

```
TABLESPACE "USERS" ;




 CREATE TABLE "RSK180001"."PERSON_CONTACT"
  (	"SSN" NUMBER(20,0) NOT NULL ENABLE,
	"PHONE" NUMBER(10,0),
	"EMAIL" VARCHAR2(20 BYTE),
	 CONSTRAINT "PERSON_CONTACT_PK" PRIMARY KEY ("SSN",
	"PHONE", "EMAIL")
	 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
	 STATISTICS
	 TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "FK_PERSON_CONTACT" FOREIGN KEY ("SSN")
	 REFERENCES "RSK180001"."PERSON" ("SSN") ENABLE
  ) SEGMENT CREATION DEFERRED
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS" ;


 CREATE TABLE "RSK180001"."BUYER_CONTACTS"
  (	"BUYER_ID" NUMBER(20,0) NOT NULL ENABLE,
	"PHONE" NUMBER(10,0),
	"EMAIL" NUMBER(10,0),
	 CONSTRAINT "BUYER_CONTACTS_PK" PRIMARY KEY
	("BUYER_ID", "PHONE", "EMAIL")
	 USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE
	 STATISTICS
	 TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "FK1_BUYER_CONTACT" FOREIGN KEY
	 ("BUYER_ID")
	  REFERENCES "RSK180001"."BUYER" ("BUYER_ID") ENABLE
  ) SEGMENT CREATION DEFERRED
 PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
TABLESPACE "USERS" ;
```

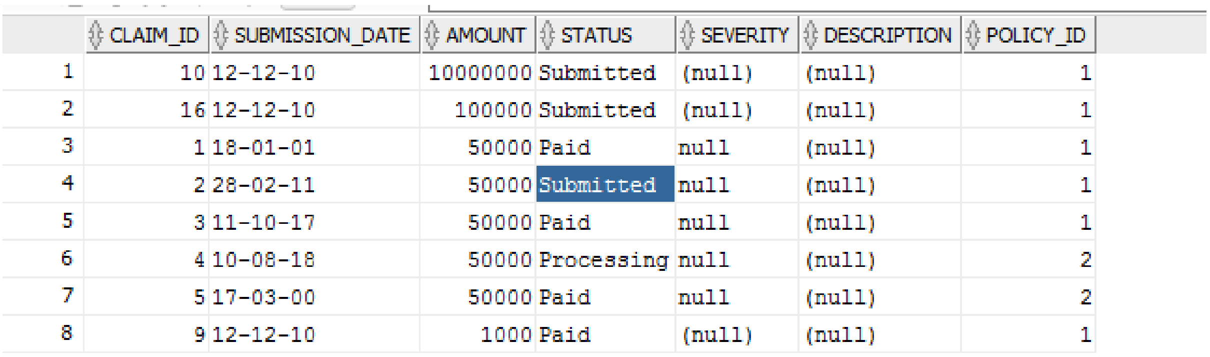
PL/SQL - Triggers:

1) Once claim is approved and payment is made to the buyer, status of the claim is updated to "PAID".



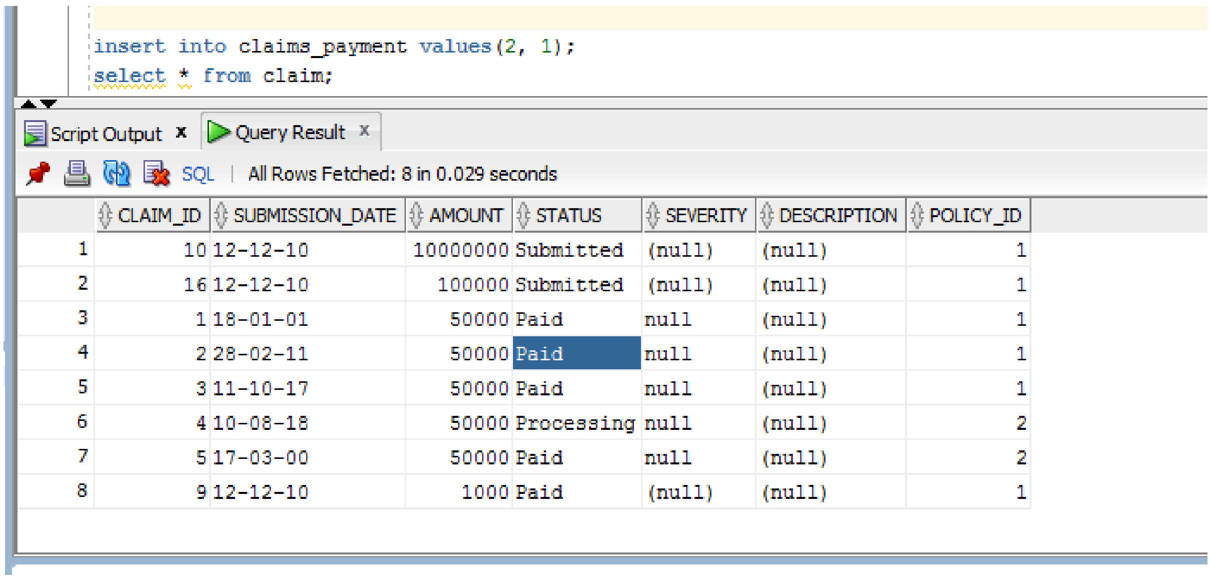
```
create or replace TRIGGER CLAIM_STATUS
AFTER INSERT ON CLAIMS_PAYMENT
FOR EACH ROW
BEGIN
update claim C set status = 'Paid '  where C.claim_id = :NEW.CLAIM_ID;
dbms_output.put_line('Status updated');
END;
```

Before



	CLAIM_ID	SUBMISSION_DATE	AMOUNT	STATUS	SEVERITY	DESCRIPTION	POLICY_ID
1	10	12-12-10	10000000	Submitted	(null)	(null)	1
2	16	12-12-10	100000	Submitted	(null)	(null)	1
3	1	18-01-01	50000	Paid	null	(null)	1
4	2	28-02-11	50000	Submitted	null	(null)	1
5	3	11-10-17	50000	Paid	null	(null)	1
6	4	10-08-18	50000	Processing	null	(null)	2
7	5	17-03-00	50000	Paid	null	(null)	2
8	9	12-12-10	1000	Paid	(null)	(null)	1

After



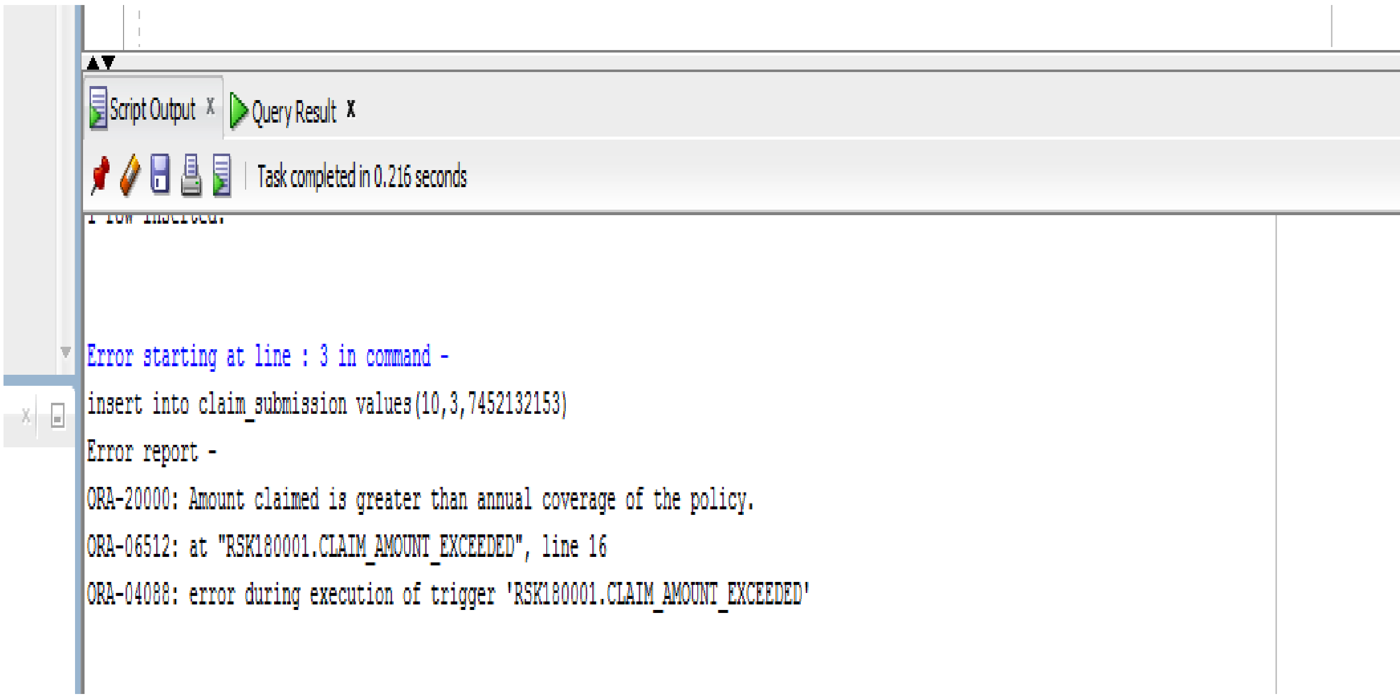
```
insert into claims_payment values(2, 1);
select * from claim;
```

All Rows Fetched: 8 in 0.029 seconds

	CLAIM_ID	SUBMISSION_DATE	AMOUNT	STATUS	SEVERITY	DESCRIPTION	POLICY_ID
1	10	12-12-10	10000000	Submitted	(null)	(null)	1
2	16	12-12-10	100000	Submitted	(null)	(null)	1
3	1	18-01-01	50000	Paid	null	(null)	1
4	2	28-02-11	50000	Paid	null	(null)	1
5	3	11-10-17	50000	Paid	null	(null)	1
6	4	10-08-18	50000	Processing	null	(null)	2
7	5	17-03-00	50000	Paid	null	(null)	2
8	9	12-12-10	1000	Paid	(null)	(null)	1

2) If amount of the claim submitted by the hospital is greater than the annual coverage of the policy, database trigger gives warning message.

```
CREATE OR REPLACE  TRIGGER CLAIM_AMOUNT_EXCEEDED
AFTER INSERT ON CLAIM_SUBMISSION
FOR EACH ROW
DECLARE
claimAmount CLAIM.AMOUNT%TYPE;
COVERAGE POLICY.ANNUAL_COVERAGE%TYPE;

BEGIN
SELECT AMOUNT INTO claimAmount
FROM CLAIM WHERE CLAIM_ID = :NEW.CLAIM_ID;

SELECT P.ANNUAL_COVERAGE INTO COVERAGE FROM POLICY P JOIN
CLAIM C ON C.POLICY_ID = P.POLICY_ID
where  C.CLAIM_ID = :NEW.CLAIM_ID ;

IF (claimAmount > COVERAGE) THEN
RAISE_APPLICATION_ERROR(-20000, 'Amount claimed is greater than annual
coverage of the policy.');
END IF;

END;
```



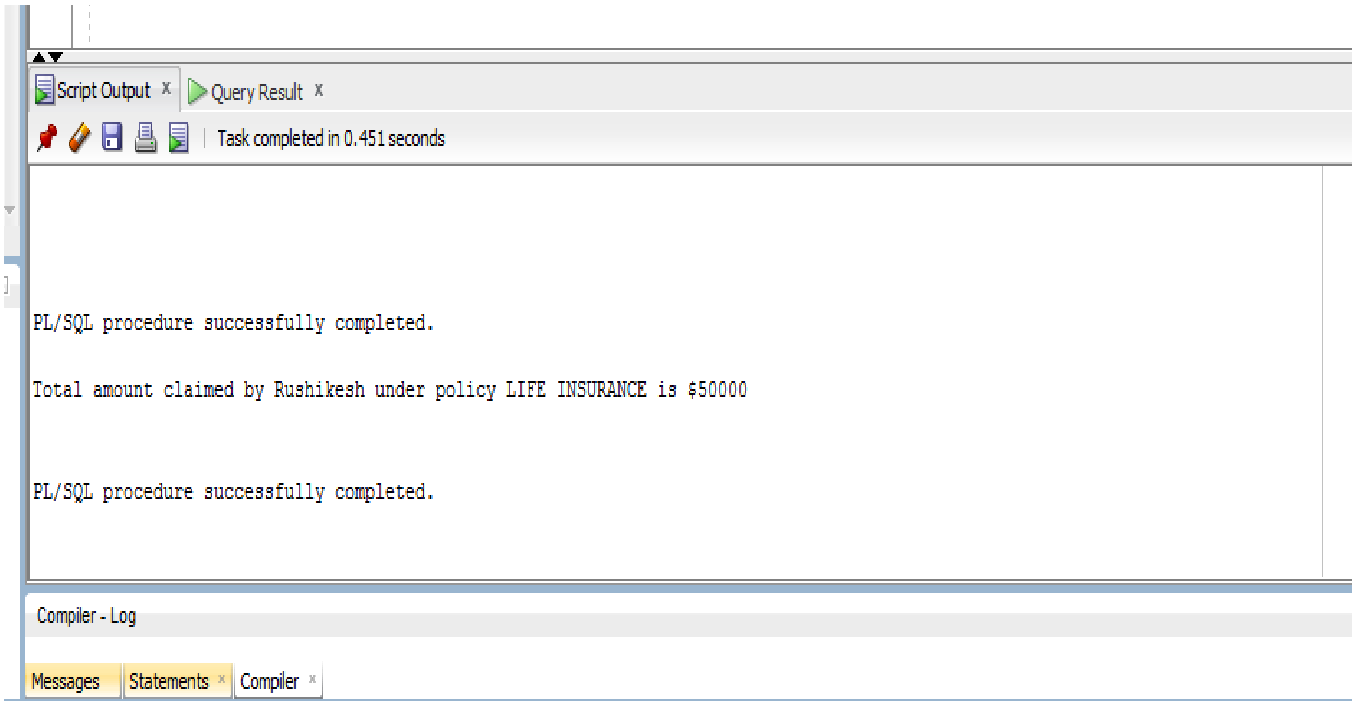
```
Error starting at line : 3 in command -
insert into claim_submission values(10,3,7452132153)
Error report -
ORA-20000: Amount claimed is greater than annual coverage of the policy.
ORA-06512: at "RSK180001.CLAIM_AMOUNT_EXCEEDED", line 16
ORA-04088: error during execution of trigger 'RSK180001.CLAIM_AMOUNT_EXCEEDED'
```

PL/SQL - Procedures:

1) Finding the total Claim Amount Paid to each person.

```
CREATE OR REPLACE PROCEDURE TOTALAMOUNTCLAIMED (PERSONID IN
PERSON.SSN%TYPE , POLICYID IN POLICY.POLICY_ID%TYPE) AS
TOTALAMOUNTCLAIMED  CLAIM.AMOUNT % TYPE;
PERSONMNAME  PERSON.NAME % TYPE;
POLICYNAME POLICY.NAME % TYPE;
BEGIN
   SELECT SUM(C.AMOUNT) INTO TOTALAMOUNTCLAIMED FROM CLAIM C JOIN
CLAIM_SUBMISSION CS ON CS.CLAIM_ID = C.CLAIM_ID
   WHERE C.POLICY_ID = POLICYID AND CS.INSURED_PERSON_ID = PERSONID AND
    C.STATUS = 'PAID';
   SELECT P.NAME , PO.NAME INTO PERSONMNAME , POLICYNAME FROM PERSON P ,
POLICY PO WHERE SSN = PERSONID AND PO.POLICY_ID = POLICYID;
   IF TOTALAMOUNTCLAIMED IS NULL THEN
      DBMS_OUTPUT.PUT_LINE('TOTAL AMOUNT CLAIMED BY PERSON = '|| 0);
   ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('TOTAL AMOUNT CLAIMED BY ' || PERSONMNAME  || ' 
UNDER POLICY ' ||POLICYNAME||' IS $' || TOTALAMOUNTCLAIMED);
    END IF;
END;
```

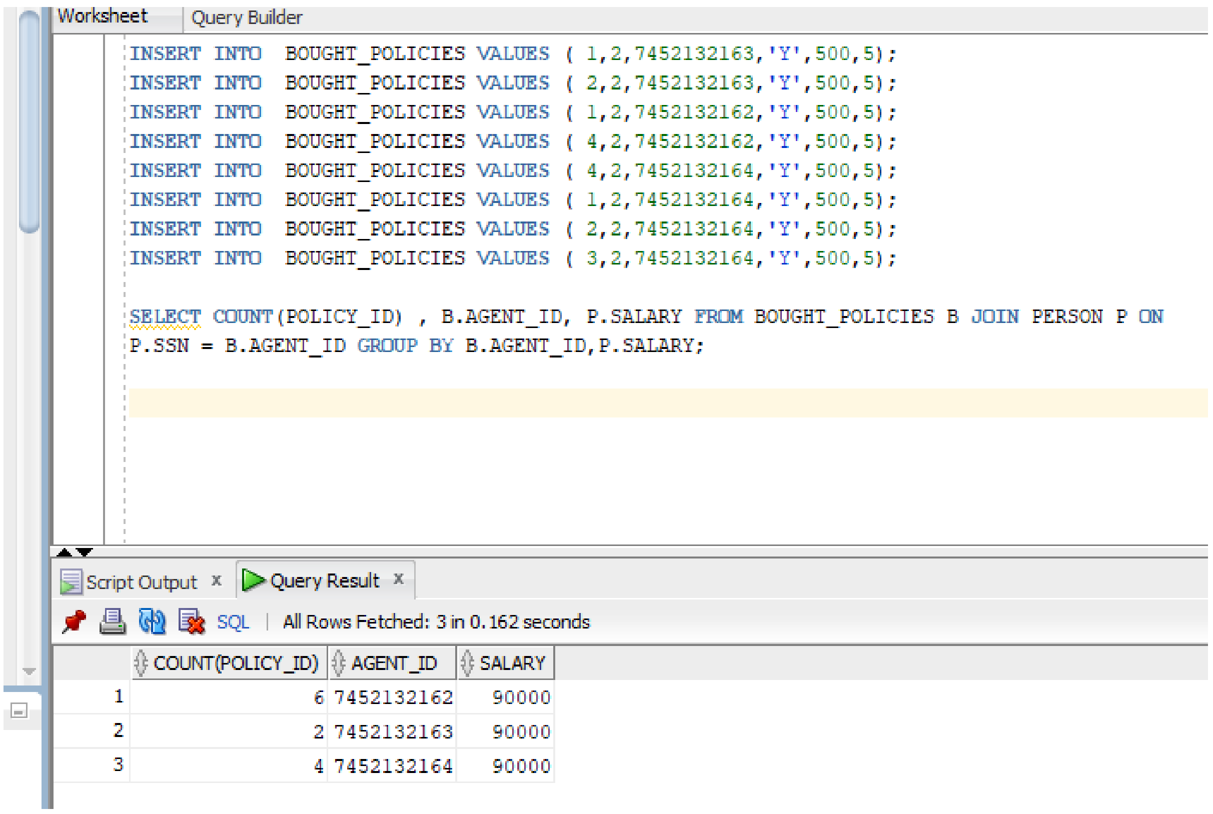


```
PL/SQL procedure successfully completed.

Total amount claimed by Rushikesh under policy LIFE INSURANCE is $50000


PL/SQL procedure successfully completed.
```

2) Based on the number of policies sold, the agent’s salary is increased.

```
CREATE OR REPLACE PROCEDURE SALARY_INCREASE
AS
POLICYCOUNT INT;
AGENT PERSON.SSN%TYPE;
SALARYINCR PERSON.SALARY%TYPE;
CURSOR AGENTCUR IS
SELECT COUNT(POLICY_ID) , B.AGENT_ID, P.SALARY FROM BOUGHT_POLICIES B JOIN
PERSON P ON
P.SSN = B.AGENT_ID GROUP BY B.AGENT_ID,P.SALARY;

BEGIN
   OPEN AGENTCUR;
    LOOP
        FETCH AGENTCUR INTO POLICYCOUNT, AGENT, SALARYINCR;
        EXIT WHEN (AGENTCUR%NOTFOUND);
        DBMS_OUTPUT.put_line(SALARYINCR);
           IF POLICYCOUNT < 3 THEN
             SALARYINCR:= SALARYINCR * 1.1;
           ELSIF POLICYCOUNT < 5 THEN
             SALARYINCR:= SALARYINCR * 1.2;
           ELSIF POLICYCOUNT < 10 THEN
             SALARYINCR:= SALARYINCR * 1.3;
           ELSE
             SALARYINCR:= SALARYINCR * 1.5;
           END IF;
           DBMS_OUTPUT.put_line(SALARYINCR);
           UPDATE PERSON SET SALARY = SALARYINCR WHERE SSN = AGENT;
    END LOOP;
CLOSE AGENTCUR;
END;
```

Salaries of Agents before increment.

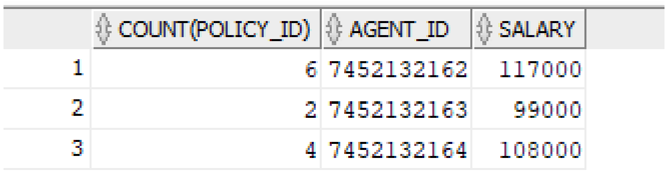


```
INSERT INTO  BOUGHT_POLICIES VALUES ( 1,2,7452132163,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 2,2,7452132163,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 1,2,7452132162,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 4,2,7452132162,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 4,2,7452132164,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 1,2,7452132164,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 2,2,7452132164,'Y',500,5);
INSERT INTO  BOUGHT_POLICIES VALUES ( 3,2,7452132164,'Y',500,5);

SELECT COUNT(POLICY_ID) , B.AGENT_ID, P.SALARY FROM BOUGHT_POLICIES B JOIN PERSON P ON
P.SSN = B.AGENT_ID GROUP BY B.AGENT_ID,P.SALARY;
```

All Rows Fetched: 3 in 0.162 seconds

	COUNT(POLICY_ID)	AGENT_ID	SALARY
1	6	7452132162	90000
2	2	7452132163	90000
3	4	7452132164	90000

Salaries incremented.



	COUNT(POLICY_ID)	AGENT_ID	SALARY
1	6	7452132162	117000
2	2	7452132163	99000
3	4	7452132164	108000