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”.



Before



After



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;



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;



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.



Salaries incremented.

