TABLE: Agency

```
CREATE TABLE Agency(agencyid integer primary key,name varchar(40) NOT NULL,address varchar(70) NOT NULL,supervisor varchar(30),phone varchar(10) NOT NULL);

ALTER TABLE AGENCY ADD CONSTRAINT PHONE_CHCK CHECK (char_length(phone) = 10);
```

TABLE: City

```
CREATE TABLE City(cityid integer primary key,cityname varchar(30) NOT NULL,agencyid integer NOT NULL,is_tourist_city integer NOT NULL,foreign key(agencyid) references Agency on delete cascade on update cascade);
```

TABLE: Package

```
CREATE TABLE Package(packageid integer primary key,pname varchar(20) NOT NULL,type varchar(20) NOT NULL,transport_mode varchar(30) NOT NULL,source_city integer NOT NULL,destination_city

integer NOT NULL,days integer NOT NULL,nights integer NOT NULL,price double precision NOT NULL,foreign key(source_city) references City on delete cascade on update cascade,

foreign key(destination_city) references City on delete cascade on update cascade);

ALTER TABLE Package ADD CONSTRAINT DAY_CHECK CHECK ((days) <=20 );

ALTER TABLE Package ADD CONSTRAINT NIGHT_CHECK CHECK ((nights) <=19 );
```

TABLE: Insurance

```
CREATE TABLE Insurance(insuranceid integer primary key,company varchar(20) NOT NULL,due_date date NOT NULL);

ALTER TABLE Insurance ADD CONSTRAINT DATE_CHECK CHECK ((due_date) >=now() );
```

TABLE: Passenger

```
CREATE TABLE Passenger(passengerid integer primary key,groupid integer NOT NULL,fname varchar(30) NOT NULL,lname varchar(30) NOT NULL,aadharno varchar(12),birthdate date,

phone varchar(10),gender varchar(6),insuranceid integer,foreign key(insuranceid) references Insurance on delete cascade on update cascade);

ALTER TABLE Passenger ADD CONSTRAINT AADHAR_Check CHECK (char_length(aadharno) =12 );

ALTER TABLE Passenger ADD CONSTRAINT PHONE_CHECK2 CHECK (char_length(phone) = 10);

ALTER TABLE Passenger ADD CONSTRAINT BDATE_CHECK CHECK ((birthdate) <=now() );
```

TABLE: Group_Package_Hotel

CREATE TABLE Group_Package_Hotel(groupid integer,packageid integer,hotelid integer,foreign key(packageid) references Package on update cascade on delete cascade,foreign key(hotelid) references Hotel on delete cascade on update cascade,primary key(groupid,packageid,JOURNEY_DATE));

ALTER TABLE GROUP_PACKAGE_HOTEL ADD COLUMN JOURNEY_DATE DATE

TABLE: Booking

CREATE TABLE Booking(bookingid integer primary key,date_of_booking date NOT NULL,groupid integer NOT NULL,packageid integer NOT NULL,price double precision NOT NULL,foreign key(groupid,packageid) references Group_Package_Hotel on delete cascade on update cascade)

TABLE: Ticket

CREATE TABLE Ticket(ticketno bigint primary key,date_ticket date NOT NULL,bookingid integer NOT NULL,foreign key(bookingid) references Booking on delete cascade on update cascade)

TABLE: Payment

CREATE TABLE Payment(paymentid bigint primary key,ticketno bigint NOT NULL,payment_mode varchar(30),payment_date date NOT NULL,foreign key(ticketno) references Ticket on delete cascade on update cascade)

TABLE: Feedback

CREATE TABLE Feedback(feedbackid integer primary key,feedback_date date NOT NULL,msg varchar(100));

TABLE: Service

CREATE TABLE Service(serviceid integer primary key,sname varchar(40),service_type varchar(20),ratings double precision,address varchar(60),due_date date,company varchar(30),hotel_insurance_id integer);

ALTER TABLE Service ADD CONSTRAINT RATINGS_CHECK CHECK ((ratings >=1.0 and ratings<=5.0));

ALTER TABLE Service ADD CONSTRAINT Due_date_check CHECK ((due_date) >=now());

TABLE: Hotel

CREATE TABLE Hotel(hotelid integer primary key,address varchar(60) NOT NULL,ratings double precision,hotel_name varchar(30) NOT NULL,cityid integer NOT NULL,foreign key(cityid) references City on delete cascade on update cascade);

ALTER TABLE Hotel ADD CONSTRAINT HOTEL_RATINGS CHECK ((ratings >=1.0 and ratings<=10.0));

TABLE: Admin

CREATE TABLE Admin(adminid integer primary key,name varchar(30),email varchar(30) NOT NULL,password varchar(30) NOT NULL)

TABLE: Offer

CREATE TABLE Offer(offerid integer primary key,offercode varchar(11) NOT NULL,start_date date,end_date date,offer_name varchar(40) NOT NULL,description varchar(50),discount INTEGER);

ALTER TABLE Offer ADD CONSTRAINT CODE_CHECK CHECK (char_length(offercode) <= 10);

alter table Offer add constraint check_dates1 check (start_date < end_date);

TABLE: Passenger_Service

CREATE TABLE Passenger_Service(passengerid integer,serviceid integer, foreign key(serviceid) references Service on delete cascade on update cascade,foreign key(passengerid) references Passenger ON DELETE CASCADE ON UPDATE CASCADE)

Table: Passenger_Feedback

CREATE TABLE Passenger_Feedback(passengerid integer, feedbackid integer, foreign key(passengerid) references passenger on delete cascade on update cascade, foreign key(feedbackid) references feedback on delete cascade on update cascade);

Table: Passenger_Offer

```
CREATE TABLE Passenger_Offer(groupid integer,offerid integer,foreign key(offerid) references offer
on delete cascade on update cascade,primary key(offerid,groupid));


create table supervisor_phone(supervisor varchar(30),phone varchar(10))



CREATE SEQUENCE ADMINSQ START 10000

CREATE SEQUENCE AGENCYSQ START 200

CREATE SEQUENCE CITYSQ START 1000

CREATE SEQUENCE HOTELSQ START 3500

CREATE SEQUENCE INSQ START 5710

CREATE SEQUENCE PACKSQ START 700

CREATE SEQUENCE PASSSQ START 1200

CREATE SEQUENCE BOOKSQ START WITH 7000

CREATE SEQUENCE TICKETSQ START WITH 50001

CREATE SEQUENCE PAYSQ START WITH 300001

CREATE SEQUENCE OFFERSQ START WITH 11001

CREATE SEQUENCE FEEDSQ START WITH 900000

CREATE SEQUENCE SERVICESQ START WITH 14000


ALTER TABLE SERVICE ALTER COLUMN SERVICEID SET DEFAULT NEXTVAL('SERVICESQ');

ALTER TABLE ADMIN ALTER COLUMN ADMINID SET DEFAULT NEXTVAL('ADMINSQ');

ALTER TABLE AGENCY ALTER COLUMN AGENCYID SET DEFAULT NEXTVAL('AGENCYSQ');

ALTER TABLE BOOKING ALTER COLUMN BOOKINGID SET DEFAULT NEXTVAL('BOOKSQ');

ALTER TABLE CITY ALTER COLUMN CITYID SET DEFAULT NEXTVAL('CITYSQ');

ALTER TABLE HOTEL ALTER COLUMN HOTELID SET DEFAULT NEXTVAL('HOTELSQ');

ALTER TABLE INSURANCE ALTER COLUMN INSURANCEID SET DEFAULT NEXTVAL('INSQ');
```

```
ALTER TABLE PACKAGE ALTER COLUMN PACKAGEID SET DEFAULT NEXTVAL('PACKSQ');

ALTER TABLE PASSENGER ALTER COLUMN PASSENGERID SET DEFAULT NEXTVAL('PASSSQ');

ALTER TABLE TICKET ALTER COLUMN TICKETNO SET DEFAULT NEXTVAL('TICKETSQ');

ALTER TABLE PAYMENT ALTER COLUMN PAYMENTID SET DEFAULT NEXTVAL('PAYSQ');

ALTER TABLE OFFER ALTER COLUMN OFFERID SET DEFAULT NEXTVAL('OFFERSQ');

ALTER TABLE FEEDBACK ALTER COLUMN FEEDBACKID SET DEFAULT NEXTVAL('feedsq');

alter table agency drop column phone
```

TRIGGER->1

```
CREATE OR REPLACE FUNCTION COUNTPRICEFORTICKET()

RETURNS TRIGGER AS

$$

DECLARE

MYID INTEGER;

MYPRICE DOUBLE PRECISION;

BEGIN

SELECT COUNT(*) AS CNT INTO MYID FROM PASSENGER WHERE GROUPID=NEW.GROUPID GROUP BY
GROUPID;

SELECT PRICE INTO MYPRICE FROM PACKAGE WHERE PACKAGEID=NEW.PACKAGEID;

NEW.PRICE:=MYPRICE * MYID;

RETURN NEW;

END;

$$

LANGUAGE plpgsql;
```

```
CREATE TRIGGER COUNTPRICE BEFORE INSERT ON BOOKING

FOR EACH ROW EXECUTE PROCEDURE COUNTPRICEFORTICKET();
```

Trigger->2

```
CREATE OR REPLACE FUNCTION INSERTINSURANCE()

RETURNS TRIGGER AS

$$

BEGIN

INSERT INTO SERVICE(SERVICE_TYPE,COMPANY,DUE_DATE,HOTEL_INSURANCE_ID)
VALUES('INSURANCE',NEW.COMPANY,NEW.DUE_DATE,NEW.INSURANCEID);

RETURN NEW;

END;

$$

LANGUAGE plpgsql;


CREATE TRIGGER INSERTINSURANCE BEFORE INSERT ON INSURANCE

FOR EACH ROW EXECUTE PROCEDURE INSERTINSURANCE();
```

Trigger->3

```
CREATE OR REPLACE FUNCTION INSERTFORHOTEL()

RETURNS TRIGGER AS

$$

BEGIN

INSERT INTO SERVICE(SNAME,SERVICE_TYPE,RATINGS,ADDRESS,HOTEL_INSURANCE_ID) VALUES
(NEW.HOTEL_NAME,'HOTEL',NEW.RATINGS,NEW.ADDRESS,NEW.HOTELID);

RETURN NEW;END;

$$

LANGUAGE plpgsql;



CREATE TRIGGER INSERTHOTEL BEFORE INSERT ON HOTEL
```

```
FOR EACH ROW EXECUTE PROCEDURE INSERTFORHOTEL();
```

TRIGGER➔4

```
CREATE OR REPLACE FUNCTION ENTRYTABLE()
RETURNS TRIGGER AS
$$
DECLARE
HID INTEGER;
R1 PASSENGER%ROWTYPE;
BEGIN
SELECT SERVICEID INTO HID FROM SERVICE WHERE HOTEL_INSURANCE_ID=NEW.HOTELID;
FOR R1 in SELECT PASSENGERID FROM PASSENGER WHERE GROUPID=NEW.GROUPID
LOOP
        INSERT INTO PASSENGER_SERVICE VALUES(R1.PASSENGERID,HID);
END LOOP;
RETURN NEW;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER ENTRYTABLE BEFORE INSERT ON GROUP_PACKAGE_HOTEL
FOR EACH ROW EXECUTE PROCEDURE ENTRYTABLE();
```

TRIGGER➔5

```
CREATE OR REPLACE FUNCTION ENTRYFORINSURANCE()
RETURNS TRIGGER AS
```

```
$$
DECLARE
IID INTEGER;
BEGIN
	IF NEW.INSURANCEID IS NOT NULL THEN
		SELECT SERVICEID INTO IID FROM SERVICE WHERE
HOTEL_INSURANCE_ID=NEW.INSURANCEID;
		INSERT INTO PASSENGER_SERVICE VALUES(NEW.PASSENGERID,IID);
	END IF;
RETURN NEW;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER ENTRYFORINSURANCE AFTER INSERT ON PASSENGER
FOR EACH ROW EXECUTE PROCEDURE ENTRYFORINSURANCE();

SELECT * FROM PASSENGER_Service
```