```
CREATE TABLE FLIGHT_DETAILS(
    FLIGHT_NO NUMBER(10) NOT NULL,
    TOTAL_NUMBER_OF_SEATS NUMBER(5) NOT NULL,
    MODEL VARCHAR2(10),
    PRIMARY KEY(FLIGHT_NO)
    );
```

```
CREATE TABLE ROUTE_DETAILS(
    ROUTE_ID NUMBER(10) UNIQUE NOT NULL,
    SOURCE VARCHAR2(20) not null,
    DESTINATION VARCHAR2(20) NOT NULL,
    FARE NUMBER(10) not null,
    FLY_DATE DATE NOT NULL,
    PRIMARY KEY(FLY_DATE,SOURCE,DESTINATION)
);
```

```
CREATE TABLE BOOKING_DETAILS(
    BOOKING_ID NUMBER PRIMARY KEY,
    BOOKING_DATE DATE NOT NULL,
    AGENT_NO NUMBER,
    AGENT_COMM NUMBER,
    CANCELATION VARCHAR2(15) DEFAULT 'NO',
    NO_OF_CANCELATION NUMBER(1) DEFAULT 0,
    REFUND_AMOUNT NUMBER  DEFAULT 0,
    MEAL VARCHAR(10) not null,
    SEAT_PREFERENCE VARCHAR(20),
    no_of_passengers number DEFAULT 1 CHECK(no_of_passengers<=6),
    ROUTE_ID NUMBER(10) REFERENCES ROUTE_DETAILS(ROUTE_ID) NOT NULL,
    TOTAL_FARE NUMBER NOT NULL,
    TOTAL_PAID_AMOUNT NUMBER(10) NOT NULL
);
```

```
Create table airline_employee(
    emp_id number(10) primary key,
    emp_name varchar2(20) not null,
    d_o_b date not null,
    profession varchar2(20) not null,
    salary number(10) not null,
    lisence_no varchar2(15) unique,
    mobile_no number(12) unique,
    experience number(2) not null
);
```

```
CREATE TABLE PASSENGER_DETAILS(
    PESSANGER_ID NUMBER(10) PRIMARY KEY,
    NAME VARCHAR(20) not null,
    D_O_B DATE NOT NULL,
    GENDER VARCHAR(10) not null,
    PHONE_NO NUMBER not null,
    EMAIL_ID VARCHAR2(15)  not null,
    PASSPORT_NO CHAR(12)
);
```

```
Create table FLIGHT_STATUS(

   Flight_no number (10) not null,

   Route_id number(10) not null,

   Seats_remaining number(5) not null,

   Constraint fk1_status foreign key (flight_no) references FLIGHT_DETAILS(flight_no),

   Constraint fk2_status foreign key (route_id) references ROUTE_DETAILS(route_id)

);
```

```
insert into FLIGHT_DETAILS values(536249,120,'Luxuy')

insert into FLIGHT_DETAILS values(536250,100,'Budget')

insert into FLIGHT_DETAILS values(536251,100,'Slime')
```

```
insert into Route_DETAILS values(1,'Delhi','Chennai',5000,TO_DATE('14-02-2021', 'DD-MM-YYYY'))

insert into Route_DETAILS values(2,'Bengaluru','Chennai',4000 ,TO_DATE('17-12-2021', 'DD-MM-
YYYY'))

insert into Route_DETAILS values(3,'Kochi','Bengaluru',4500 , TO_DATE('04-05-2021', 'DD-MM-
YYYY'))
```

```
insert into FLIGHT_STATUS values(536249,3,100)

insert into FLIGHT_STATUS values(536249,2,120)

insert into FLIGHT_STATUS values(536249,1,120)
```

```
INSERT INTO PASSENGER_DETAILS VALUES (302,'PANDA','02-08-
1999','MALE',8978795768,'FWUUF@GMAIL.COM',786756453423);
```

PROCEDURES

```
CREATE OR REPLACE PROCEDURE V_AVAILABLE_SEATS(V_SOURCE IN VARCHAR2, V_DESTINATION IN VARCHAR2, V_FLY_DATE IN DATE,V_AVAILABLE_SEATS_NO OUT NUMBER)
AS
BEGIN
SELECT Seats_remaining INTO V_AVAILABLE_SEATS_NO
FROM FLIGHT_STATUS WHERE ROUTE_ID = (SELECT ROUTE_ID FROM ROUTE_DETAILS WHERE SOURCE = V_SOURCE AND DESTINATION =V_DESTINATION and FLY_DATE= V_FLY_DATE);
END;


DECLARE
V_SEATS NUMBER;
BEGIN
V_AVAILABLE_SEATS('Delhi','Chennai','02-14-2021',V_SEATS);
DBMS_OUTPUT.PUT_LINE(V_SEATS);
END;
```

OUTPUT=120

```
CREATE OR REPLACE PROCEDURE AGENT_COMM (V_BOOKING_ID IN NUMBER,V_BOOKING_DATE IN DATE,V_AGENT_NO IN NUMBER,V_ROUTE_ID IN NUMBER,V_no_of_passengers IN NUMBER,V_MEAL IN VARCHAR2,V_TOTAL_FARE IN NUMBER)
AS
V_AGENT_COMM NUMBER(10);
V_TOTAL_PAID_AMOUNT NUMBER(10);
BEGIN
V_AGENT_COMM:=0;
IF V_AGENT_NO IS NOT NULL THEN
V_AGENT_COMM:=V_TOTAL_FARE*0.1;
```

```
V_TOTAL_PAID_AMOUNT:=V_TOTAL_FARE+V_AGENT_COMM;

ELSE

V_TOTAL_PAID_AMOUNT:=V_TOTAL_PAID_AMOUNT;

END IF;

INSERT INTO
BOOKING_DETAILS(BOOKING_ID,BOOKING_DATE,AGENT_NO,ROUTE_ID,no_of_passengers,MEAL,TOTAL_FARE,TOTAL_PAID_AMOUNT,AGENT_COMM)

VALUES(V_BOOKING_ID
,V_BOOKING_DATE,V_AGENT_NO,V_ROUTE_ID,V_no_of_passengers,V_MEAL,V_TOTAL_FARE,V_TOTAL_PAID_AMOUNT,V_AGENT_COMM);

END;
```

EXECUTION STATEMENT-

```
BEGIN

AGENT_COMM(1011,SYSDATE,7878,1,2,'VEG',10000);

END;


SELECT AGENT_COMM FROM BOOKING_DETAILS

WHERE BOOKING_ID = 1011


 ##OUTPUT=1000
```

```
ALTER TABLE BOOKING_DETAILS

ADD CANCELATION_DATE DATE;


ALTER TABLE BOOKING_DETAILS

ADD CANCELATION_CHARGES NUMBER;
```

```
CREATE OR REPLACE PROCEDURE CANCELATION(V_BOOKING_ID IN
NUMBER,V_CANCELATION_DATE IN DATE,V_NO_OF_CANCELATION IN NUMBER)

AS

N_FLY_DATE DATE;

N_no_of_passengers NUMBER;

N_REFUND_AMOUNT NUMBER;

N_TOTAL_FARE NUMBER;

N_FARE NUMBER;

N_CANCELATION_CHARGES NUMBER;

V_DIFF NUMBER;

BEGIN

UPDATE BOOKING_DETAILS SET CANCELATION='YES',
NO_OF_CANCELATION=V_NO_OF_CANCELATION, CANCELATION_DATE=V_CANCELATION_DATE
WHERE BOOKING_ID=V_BOOKING_ID;

SELECT R.FLY_DATE,R.FARE,B.no_of_passengers,B.REFUND_AMOUNT,B.TOTAL_FARE INTO
N_FLY_DATE,N_FARE, N_no_of_passengers, N_REFUND_AMOUNT, N_TOTAL_FARE

FROM BOOKING_DETAILS B, ROUTE_DETAILS R WHERE R.ROUTE_ID=B.ROUTE_ID AND
BOOKING_ID=V_BOOKING_ID;

SELECT TO_NUMBER(TO_DATE(N_FLY_DATE)-TO_DATE(V_CANCELATION_DATE)) INTO V_DIFF from
dual;

IF TO_NUMBER(V_DIFF)> 15 THEN

N_REFUND_AMOUNT := 0.9*(V_NO_OF_CANCELATION*N_FARE);

N_CANCELATION_CHARGES := 0.1*(V_NO_OF_CANCELATION*N_FARE);

ELSIF TO_NUMBER(V_DIFF) >10 THEN

N_REFUND_AMOUNT := 0.7*(V_NO_OF_CANCELATION*N_FARE);

N_CANCELATION_CHARGES := 0.3*(V_NO_OF_CANCELATION*N_FARE);

ELSIF TO_NUMBER(V_DIFF) >5 THEN

N_REFUND_AMOUNT := 0.5*(V_NO_OF_CANCELATION*N_FARE);

N_CANCELATION_CHARGES := 0.5*(V_NO_OF_CANCELATION*N_FARE);
```

```
ELSE

N_REFUND_AMOUNT := 0*(V_NO_OF_CANCELATION*N_FARE);

N_CANCELATION_CHARGES := 1*(V_NO_OF_CANCELATION*N_FARE);

END IF;

UPDATE BOOKING_DETAILS SET
REFUND_AMOUNT=N_REFUND_AMOUNT,CANCELATION_CHARGES=N_CANCELATION_CHARGES
WHERE BOOKING_ID=V_BOOKING_ID;

END;
```

CHECK=

```
BEGIN

CANCELATION(1011,'02-01-2021',2);

END;
```

###OUTPUT

```
SELECT * FROM BOOKING_DETAILS
```

TRIGGERS

```
CREATE TABLE PASSENGER_HIST AS SELECT * FROM PASSENGER_DETAILS


CREATE OR REPLACE TRIGGER PASSENGER_DEL_TRG

AFTER DELETE ON PASSENGER_DETAILS

FOR EACH ROW

BEGIN

INSERT INTO PASSENGER_HIST VALUES (:OLD.PESSANGER_ID, :OLD.NAME,:OLD.D_O_B,
:OLD.GENDER,:OLD.PHONE_NO, :OLD.EMAIL_ID, :OLD.PASSPORT_NO);
```

```
END;



DELETE FROM PASSENGER_DETAILS WHERE PESSANGER_ID=302


##OUTPUT

SELECT * FROM PASSENGER_HIST
```

```
ALTER TABLE BOOKING_DETAILS

ADD UPDATED_DATE DATE;

ALTER TABLE BOOKING_DETAILS

ADD UPDATED_USER VARCHAR2(20)


CREATE OR REPLACE TRIGGER UPDATE_DETAILS_TRG

BEFORE INSERT OR UPDATE ON BOOKING_DETAILS

FOR EACH ROW

BEGIN

:NEW.UPDATED_DATE := SYSDATE;

:NEW.UPDATED_USER:=USER;

END;



UPDATE BOOKING_DETAILS SET MEAL='NON-VEG' WHERE AGENT_NO=2077


##OUTPUT

SELECT * FROM BOOKING_DETAILS
```

```
CREATE TABLE PASSENGER_BOOKING(PESSANGER_ID NUMBER(10) REFERENCES PASSENGER_DETAILS(PESSANGER_ID), BOOKING_ID NUMBER(10) REFERENCES BOOKING_DETAILS(BOOKING_ID))

INSERT INTO PASSENGER_BOOKING VALUES(302,1011)


CREATE OR REPLACE TRIGGER ONE_BOOKING_TRIGG

BEFORE INSERT ON PASSENGER_BOOKING

FOR EACH ROW

DECLARE

V_ROUTE NUMBER;

V_PASS_ID NUMBER(10);

V_ROUTE_ID NUMBER(10);

S_PASS_ID NUMBER(10);

CURSOR C_PASSENGER IS

SELECT P.PESSANGER_ID, B.ROUTE_ID FROM PASSENGER_BOOKING P,BOOKING_DETAILS B WHERE P.BOOKING_ID=B.BOOKING_ID;

BEGIN

OPEN C_PASSENGER;

SELECT ROUTE_ID INTO V_ROUTE FROM BOOKING_DETAILS WHERE BOOKING_ID=:NEW.BOOKING_ID;

S_PASS_ID:=:NEW.PESSANGER_ID;

LOOP

FETCH C_PASSENGER INTO V_PASS_ID,V_ROUTE_ID;

IF S_PASS_ID=V_PASS_ID AND V_ROUTE= V_ROUTE_ID THEN

raise_application_error(-20225,'CANT BOOK SAME FLIGHT TWICE!');

END IF;

EXIT WHEN C_PASSENGER %NOTFOUND;

END LOOP;
```

```
CLOSE C_PASSENGER;

END;
```

```
INSERT INTO PASSENGER_BOOKING VALUES(302,1011)
```

OUTPUT:

```
YOU CANNOT BOOK THE SAME FLIGHT TWICE
```