Objective:

To create a database for a Tour Booking and Travel Management system.

Description:

Below is a brief outline of what Tour Booking and Travel Management includes:
Travel agencies organize tours and customer will contact the travel agency for flight bookings and hotel bookings. Travel agency is responsible for accommodation and transport
Tour Packages include cities only.
Our Travel Agency employs both general staff, and tour guides. The booking, cancellation details and feedback systems about tour packages are also included in the database.

Entities and attributes:

Person can be specialized as a customer, tour guide, staff working for travel agency. Person shall have as attributes: sex, person id, name, age, and address with city, pincode, street, country, and street number.
Transaction shall have as attributes status of booking, transaction id and will be specialized into disjoint **Book** and C**ancel** subclasses.
Book shall have as attributes date of booking, outstanding balance, due date of payment, and type of payment.
Cancel shall have as attributes refund amount and date of cancellation.
Travel Agency shall have as attributes name, companyId, phone number, country, email address, and address with country, pincode, city, street, and street number.
Transportation Company shall have as attributes phone number, business name, transportid , type of transport and it has subclasses based on the type of transport. This entity is specialized on the basis of transportation type: land, sea, and air.
Tour package shall have as attributes number of days, number of places, cost, tour name, package id , start date, end date and a maximum number of people.
Place must be of either hotel type, branch type, or point of interest type. Place shall have as attributes: place id, names, and address which includes country, city, pincode, street, and street number. Place can be specialized into disjoint subclasses 1) **Places of interest** 2) **Travel agency branches** and 3) **hotels**.
Travel agency branches shall have as attributes attributes branch name, branch id.
Hotels shall have as attributes phone number and hotel name.
Places of interest shall have as attributes description, website address and poi name.
Guide shall have a salary attribute. Staff shall also have as attributes salary and bonus per year.

Relationship Assumptions:

Customer - Transaction(1:M): Each Customer in the database can make many transactions and each transaction is associated with only one customer.

Transaction - Tour Package(M:1): Each Transaction is for only to book/cancel one tour package. Each tour package can be associated with many transactions done by one or many customers. Each time a customer tries to book or cancel, a transaction id is generated.

Tour Package - Place(M:N): Each tour package can include many places and each place can be associated with many tour packages.

Customer - Tour package(M:N): Each Customer can give feedback to many tour packages and each package can be reviewed by many customers.

Travel agency - Tour package (M:N): A tour package can be organized by many different travel agencies and travel agencies can organize many tour packages.

Travel Guide - Tour package(1:1): Each travel guide assists exactly one tour package and vice-versa. It is assumed that the travel guides are local to give a superior travel experience.

Travel guide - Travel company(M:N): A guide can work in many travel companies and a travel company can hire many guides.

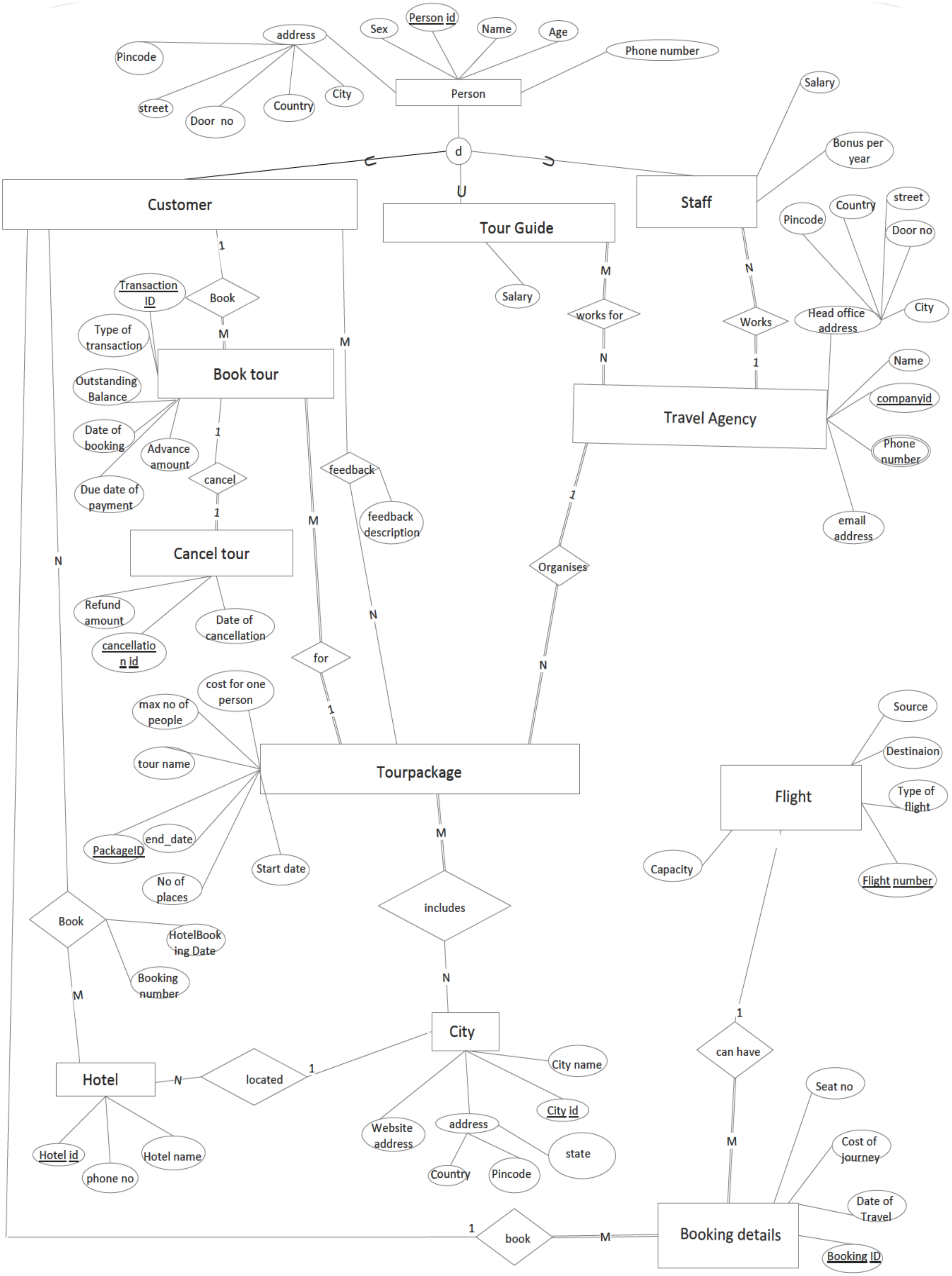
Staff - Travel agency(N:1): A person can work in only one travel company but a travel company can hire many employees.

Travel agency - Transportation company(N:M): A travel company contracts many transportation companies and a transportation company can work with many travel agencies.

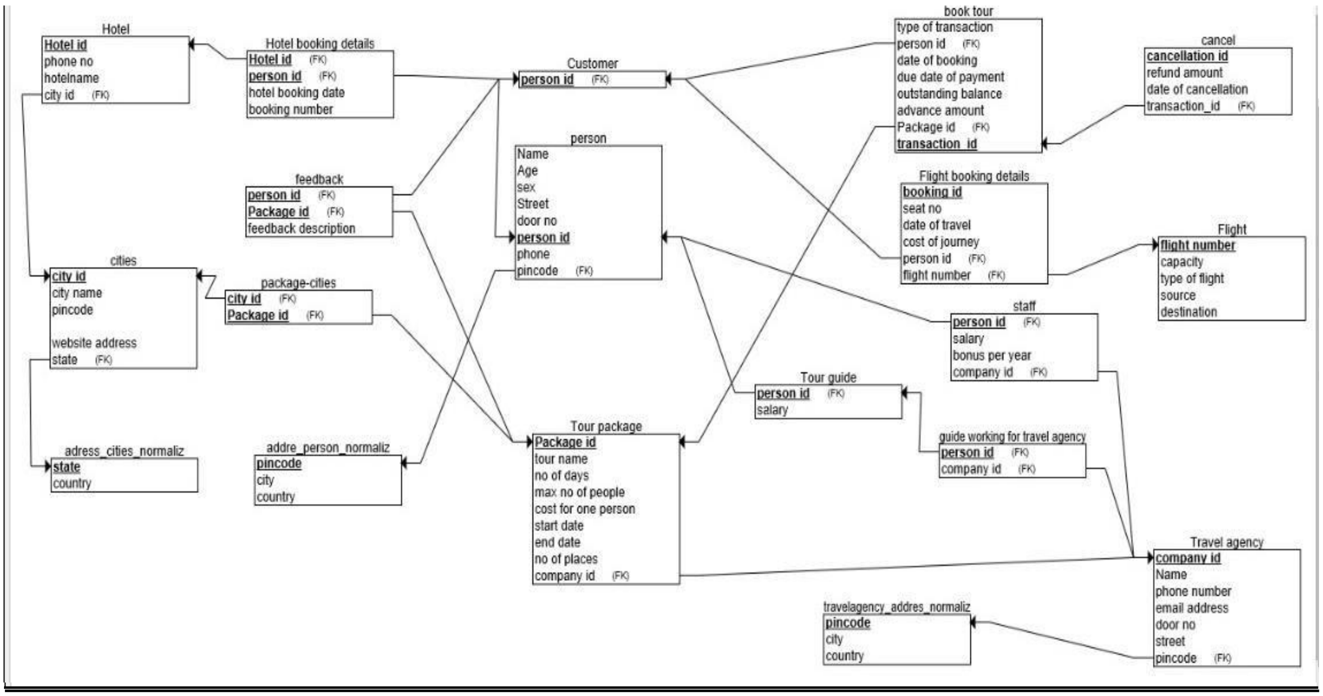
Transportation company - Tour package(N:M): A transportation company can provide transport to many tour packages and vice versa.
Travel agency - place(1:M): A Travel agency can have many branches but each branch is associated with only a single travel agency.

Assumptions:
The main assumptions are a package can be organized by maximum only one travel agency.

Entity	**Entity**	**Relationship assumptions**
customer	book tour	1:m
customer	hotel	m:n
customer	flight booking details	1:m
flight booking details	flight	m:1
booktour	canceltour	one to one
hotel	city	n:1
booktour	tourpackage	m:1
tourpackage	city	m:n
travelagency	tourpackage	1:n
tourguide	travelagency	m:n
staff	travelagency	n:1


Person id
Sex
Name
Age
address
Phone number
Pincode
Person
street
Door no
Country
City
Salary
Bonus per year
d
Customer
Tour Guide
Staff
Pincode
Country
street
Door no
1
M
N
Transaction ID
Book
works for
Works
Head office address
City
Salary
Type of transaction
M
M
N
1
Book tour
Name
Outstanding Balance
companyid
Travel Agency
Date of booking
1
Advance amount
feedback
Phone number
cancel
Due date of payment
feedback description
1
1
email address
M
N
Cancel tour
Organises
Refund amount
Date of cancellation
N
cancellation id
for
N
cost for one person
max no of people
1
Source
tour name
Tourpackage
Destinaion
Flight
Type of flight
M
end_date
PackageID
Start date
Capacity
Flight number
No of places
includes
Book
HotelBooking Date
Booking number
M
N
1
City
can have
Hotel
N
located
1
City name
Seat no
City id
M
Website address
address
state
Cost of journey
Hotel id
Hotel name
Country
Pincode
phone no
Date of Travel
1
book
M
Booking details
Booking ID

This is the relational schema before normalization:



The tables which violate 3NF normalization are person, cities,travel_agency.

Dependencies in person table are:

Candidate keys are

1)phone number

2) person id

Here door_no and street and names of persons can have same names but located in different countries and cities.

In real world we may have two cities with same name but located in different countries and we may have two different cities with the same name located in the same country.

For same pincode, we can have different streets and different door nos.

With the pincode, street, door no we cannot identify a person uniquely because in the same house we can have two different customers.

Functional dependencies are

pincode->city,country

so (pincode,city,country) form separate table with pincode as primary key.

Dependecies in cities table are :

state->country

candidate keys are:

1)pincode

2)website address

3)city_id

City name is not a candidate key we can have two same city names within same state and we can also have two same city names in two different states.

But a country can be uniquely determined by the state.

(state, country) separate table.

Dependencies in trave_agency:

Here in cities candidate keys are:

1)phone number

2)email address

3)company_id

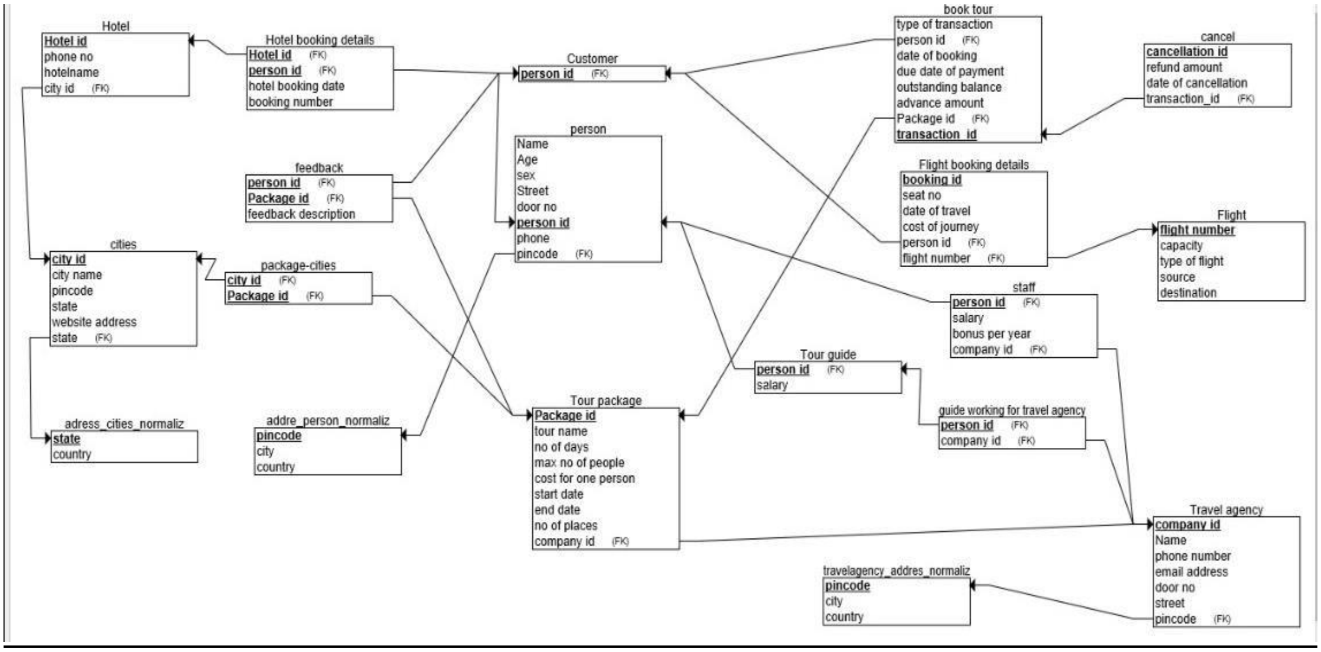
4)door_no, street, pincode

In same street we can have different door no, with the street, pincode,door_no we can uniquely identify the tuple.

Functional dependency

(pincode->city,country) . so separate table (pincode, city,country) is created

This is the relational schema after normalization:



SQL CODE for creation of tables:

```
CREATE TABLE Person
(
  Name varchar(300) NOT NULL,
   PHONE int not null,
  Age INT not null,
  sex CHAR(1) NOT NULL,
  pincode int  not null,
  Street VARCHAR (200),
  door_no int  ,

  birthdate DATE NOT NULL,
  person_id INT NOT NULL,
  PRIMARY KEY (person_id)

);
CREATE TABLE travelagency_addres_normaliz
(
  pincode INT NOT NULL, city
  CHAR(200) NOT NULL, country
  CHAR(400) NOT NULL,
  PRIMARY KEY (pincode)
);
CREATE TABLE adress_cities_normaliz
(
  state CHAR(400) NOT NULL,
  country CHAR(400) NOT NULL,
  PRIMARY KEY (state));
CREATE TABLE addre_person_normaliz
(
  pincode INT NOT NULL, city
  CHAR(400) NOT NULL, country
  CHAR(400) NOT NULL,
  PRIMARY KEY (pincode)
);


CREATE TABLE Customer
(
  person_id INT NOT NULL,
  PRIMARY KEY (person_id),

  constraint customervsperson
  foreign key (person_id)
  references Person(person_id) on delete cascade
   );
CREATE TABLE Tour_guide
(
  salary INT NOT NULL,
  person_id INT NOT NULL,
  PRIMARY KEY (person_id),
  constraint tourguidevsperson
  foreign key (person_id)
  references Person(person_id) on delete cascade
```

```
);

CREATE TABLE Book_tour
(type_of_transaction VARCHAR(100) NOT NULL,
  Transaction_id INT NOT NULL,
  package_id int not null, person_id INT
  NOT NULL, date_of_booking DATE NOT
  NULL, duedate_of_payment DATE NOT
  NULL, outstanding_balance INT NOT
  NULL, advance_amount INT NOT NULL,
  PRIMARY KEY (Transaction_id),
  constraint bookingvsperson
  FOREIGN KEY (person_id) REFERENCES Customer(person_id)on delete cascade,
  constraint bookingtourvspackage foreign key (package_id) References
tour_package(package_id)
on delete cascade

);

CREATE TABLE cancel_tour
(
cancellation_id int not null,
  refund_amount INT not null,
  date_of_cancellation DATE not null,
  Transaction_id INT NOT NULL,
  PRIMARY KEY (cancellation_id),
  constraint cancellationvsbookingtour
  FOREIGN KEY (Transaction_id) REFERENCES Book_tour(Transaction_id)on delete cascade
);

CREATE TABLE cities
(
  city_id INT NOT NULL,
  city_name VARCHAR(100) NOT NULL,
  pincode INT,

  state VARCHAR(100),
  website_address VARCHAR(600),
  PRIMARY KEY (city_id)
);

CREATE TABLE Flight
(
  capacity INT,
  flight_number VARCHAR(10) NOT NULL,
  type_of_flight CHAR(50) NOT NULL,
  source CHAR(10) NOT NULL,
  destination CHAR(10) NOT NULL,
  PRIMARY KEY (flight_number)
);

CREATE TABLE Travel_agency
```

```
(
  Name CHAR(400) NOT NULL,
  company_id INT NOT NULL,
  phone_number VARCHAR(11),

  email_address VARCHAR(400),

  pincode VARCHAR(10),
  door_no VARCHAR(10),
  street VARCHAR(300),
  PRIMARY KEY (company_id)
);

CREATE TABLE staff
(
  salary INT, bonus_per_year
  INT, person_id INT NOT
  NULL, company_id INT NOT
  NULL, PRIMARY KEY
  (person_id), constraint
  staffvsperson
  FOREIGN KEY (person_id) REFERENCES Person(person_id) on delete cascade,
constraint  staffvstravelagency
  FOREIGN KEY (company_id) REFERENCES Travel_agency(company_id)on delete set null
);

CREATE TABLE Tour_package
```

```
(
  Package_id INT NOT NULL,
  tour_name VARCHAR(200) NOT NULL,
  no_of_days INT,
  max_no_of_people INT,
  cost_for_one_person INT,
  start_date DATE,
  end_date DATE ,
  no_of_places INT,
  company_id INT NOT NULL,

  PRIMARY KEY (Package_id),
  constraint tourpackagevstravelagency
  FOREIGN KEY (company_id) REFERENCES Travel_agency(company_id)on delete cascade

);

CREATE TABLE feedback
(
  feedback_description VARCHAR(400),
  person_id INT NOT NULL,
  Package_id INT NOT NULL,
  PRIMARY KEY (person_id, Package_id),
  constraint feedbackvsperson
  FOREIGN KEY (person_id) REFERENCES Customer(person_id) on delete cascade,
  constraint feedbackvstourpackage
  FOREIGN KEY (Package_id) REFERENCES Tour_package(Package_id)on delete cascade
);

CREATE TABLE Hotel
(
  Hotel_id INT NOT NULL,
  phone_no INT,
  hotelname VARCHAR(200) NOT NULL,
  city_id INT NOT NULL,
  PRIMARY KEY (Hotel_id),
  constraint cityvshotel
  FOREIGN KEY (city_id) REFERENCES cities(city_id)on delete cascade
);

CREATE TABLE Hotel_booking_details
(
  hotel_booking_date DATE NOT NULL,
  booking_number INT NOT NULL,
  Hotel_id INT NOT NULL,
  person_id INT NOT NULL,
  PRIMARY KEY (Hotel_id, person_id),
  constraint hotelbookingvshotel
  FOREIGN KEY (Hotel_id) REFERENCES Hotel(Hotel_id)on delete cascade,
  constraint hotelbookingvsperson
  FOREIGN KEY (person_id) REFERENCES Customer(person_id) on delete cascade

);
```

```
CREATE TABLE package_cities
(
  city_id INT NOT NULL, Package_id
  INT NOT NULL, PRIMARY KEY
  (city_id, Package_id), constraint
  packagecitiesvscities
  FOREIGN KEY (city_id) REFERENCES cities(city_id) on delete set null,
  constraint packagecitiesvstoupackage
  FOREIGN KEY (Package_id) REFERENCES Tour_package(Package_id)on delete set null
);

CREATE TABLE Flight_booking_details
(
  seat_no INT,
  booking_id INT NOT NULL,
  date_of_travel DATE,
  cost_of_journey INT,
  person_id INT NOT NULL,
  flight_number VARCHAR(10) NOT NULL,
  PRIMARY KEY (booking_id),
  constraint flightbookingvsperson
 FOREIGN KEY (person_id) REFERENCES Customer(person_id) on delete cascade,
 constraint flightbookingvsflight
  FOREIGN KEY (flight_number) REFERENCES Flight(flight_number)  on delete cascade
);
```

```
CREATE TABLE guide_travel_agency
(
   person_id INT NOT NULL,
  company_id INT NOT NULL,
  PRIMARY KEY (person_id, company_id),

  FOREIGN KEY (person_id) REFERENCES Tour_guide(person_id) on delete cascade,
  FOREIGN KEY (company_id) REFERENCES Travel_agency(company_id)on delete set null
);
```

Sql code for insertion of tuples:

```
INSERT INTO addre_person_normaliz( pincode,country,city)
VALUES ('75080', 'USA','Richardson');
INSERT INTO addre_person_normaliz( pincode, country,city)
VALUES ('20147','USA','ASHBURN');
INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('75013','USA','ALLEN');
INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('98745','japan','tokyo');
INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('12345','USA','manchester');



INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('41254','USA','SACREMENTO');
INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('63214','USA','MIAMI');
INSERT INTO addre_person_normaliz( pincode,  country,city)
VALUES ('98741','USA','BOSTON');





INSERT INTO Person (name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ('EDWARD', '7895213256','42','M','75080','West Campbell Road', '800','04-MAR-1990','5214');
INSERT INTO Person (Name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
```

```
VALUES ('SAMUEL','7895146514','35','M','20147','HAY ROAD', '43383','01-JAN-1981','5215');
INSERT INTO Person (Name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ('JOE', '7036541245','21','M','75013','SAN JACINTO DRIVE', '8305','14-FEB-1995','5216');

INSERT INTO Person (name,PHONE,Age, sex, pincode, Street, door_no,birthdate,person_id)
VALUES ( 'KURIAN','7035648999',34,'M','75013','SAN JACINTO DRIVE', '1234','17-SEP-
1986','5217');
INSERT INTO Person (name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ( 'NEST','8945673245',78,'F','98745','LEXINGTON AVENUE', '8741','28-FEB-1982','5218');
INSERT INTO Person (NAME,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ( 'STEPHEN','98456732487','21','F','12345','SALADO SPRINGS', '20546','03-MAR-
1995','5219');

INSERT INTO Person (name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ('DANIEL', '2149875463','60','M','41254','RENNER ROAD', '10002','14-MAR-
1956','5220');
INSERT INTO Person (name,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ('SETH','9721457896','45','F','63214','SPRING CIRCLE', '10236','15-JAN-1971','5221');
INSERT INTO Person (NAME,PHONE,Age, sex, pincode, Street, door_no, birthdate,person_id)
VALUES ('ELI', '5124567895','40','M','98741','OUTPOST STREET', '45632','14-FEB-1976','5222');

INSERT INTO Customer (person_id)
VALUES ('5214');
INSERT INTO Customer (person_id)
VALUES ('5215');
INSERT INTO Customer (person_id)
VALUES ('5216');
```

```
INSERT INTO tour_guide (salary, person_id)

VALUES ('80000','5217');

INSERT INTO tour_guide (salary, person_id)

VALUES ('92000','5218');

INSERT INTO tour_guide (salary, person_id)

VALUES ('20000','5219');
```

```
INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)

VALUES ('01-MAR-2015','01-MAR-2015','0','685','ONLINE CREDIT CARD','1','5214','700');

INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)

VALUES ('01-FEB-2016','01-JUN-2016','128','128','IN PERSON CASH','2','5215','710');

INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)

VALUES ('08-APR-2016','10-APR-2016','0','1053','IN PERSON CREDIT CARD','3','5216','720');

INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)

VALUES ('10-APR-2015','15-APR-2015','0','2000','IN PERSON CREDIT CARD','4','5214','730');

INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)

VALUES ('01-FEB-2016','01-JUN-2016','0','256','IN PERSON CASH','5','5216','710');
```

```
INSERT INTO BOOK_TOUR
(date_of_booking,duedate_of_payment,outstanding_balance,advance_amount,type_of_transaction,Transaction_id,person_id,package_id)
VALUES ('01-FEB-2016','01-JUN-2016','128','128','IN PERSON CASH','6','5216','710');
```

```
INSERT INTO cancel_tour (refund_amount,date_of_cancellation,Transaction_id,cancellation_id)
VALUES ('2000','16-apr-2015','4','23');
INSERT INTO cancel_tour (refund_amount,date_of_cancellation,Transaction_id,cancellation_id)
VALUES ('685','04-jun-2015','1','24');
```

```
INSERT INTO cities (city_id,city_name,pincode,state,website_address)
VALUES ('100','pantego','32456','TEXAS','WWW.PANTEGOTX.COM');
INSERT INTO cities (city_id,city_name,pincode,state,website_address)
VALUES ('110','LASVEGAS','74120','NEVADa','WWW.LASVEGAS.COM');
INSERT INTO cities (city_id,city_name,pincode,state,website_address)
VALUES ('120','SANDIEGO','23104','CALIFORNIA','WWW.SANDIEGO.COM');
INSERT INTO cities (city_id,city_name,pincode,state,website_address)
VALUES ('130','LONDON','111124','bangor','WWW.LONDONEYE.COM');
```

```
INSERT INTO Adress_cities_normaliz (country,state)
VALUES ('USA','TEXAS');
```

```
INSERT INTO Adress_cities_normaliz (country,state)
VALUES ('USA','NEVADa');
INSERT INTO Adress_cities_normaliz (country,state)
VALUES ('USA','CALIFORNIA');
INSERT INTO Adress_cities_normaliz (country,state)
VALUES ('UNITED KINGDOM','bangor');

INSERT INTO flight (capacity,flight_number,type_of_flight,source,destination)
VALUES ('356','UUZ877','airbus838','IAD','DFW');
INSERT INTO flight (capacity,flight_number,type_of_flight,source,destination)
VALUES ('128','E845DE','boying740','DFW','SND');
INSERT INTO flight (capacity,flight_number,type_of_flight,source,destination)
VALUES ('356','YY21DA','dreamliner','DFW','LHR');
INSERT INTO flight (capacity,flight_number,type_of_flight,source,destination)
VALUES ('354','YY21D7','airbusa330','LHR','SND');
INSERT INTO flight (capacity,flight_number,type_of_flight,source,destination)
VALUES ('354','YY21AA','boying737','DFW','SND');

INSERT INTO
travel_agency(Name,company_id,phone_number,email_address,pincode,door_no,street)
VALUES('Sunrise Travels','1','879456213','sunrise@st.com','75013','1706','staffod avenue');
INSERT INTO
travel_agency(Name,company_id,phone_number,email_address,pincode,door_no,street)
VALUES('Moonlit
Travels','2','+4454123647','moonlittravel@mlt.com','5432','34567','musheerbad');
INSERT INTO
travel_agency(Name,company_id,phone_number,email_address,pincode,door_no,street)
VALUES('TOON-TOWN
Travels','3','5717418965','tripleT@toontown.com','20146','47785','nantucket drive');
```

```
INSERT INTO travelagency_addres_normaliz(country,city,pincode)

VALUES('USA','ALLEN','75013');

INSERT INTO travelagency_addres_normaliz(country,city,pincode)

VALUES('INDIA','delhi','5432');

INSERT INTO travelagency_addres_normaliz(country,city,pincode)

VALUES('USA','fairfax','20146');



INSERT INTO staff(salary,bonus_per_year,person_id,company_id)

VALUES ('70000','1000','5220','1');

INSERT INTO staff(salary,bonus_per_year,person_id,company_id)

VALUES ('40000','3300','5221','2');

INSERT INTO staff(salary,bonus_per_year,person_id,company_id)

VALUES ('56987','2000','5222','3');


INSERT INTO
tour_package(package_id,tour_name,no_of_days,max_no_of_people,cost_for_one_person,start_date,end_date,no_of_places,company_id)

VALUES ('700','europe','5','40','685','15-JUN-2016','20-JUN-2016','1','1');

INSERT INTO
tour_package(package_id,tour_name,no_of_days,max_no_of_people,cost_for_one_person,start_date,end_date,no_of_places,company_id)

VALUES ('710','asia','4','30','256','15-JUN-2017','19-JUN-2017','1','2');

INSERT INTO
tour_package(package_id,tour_name,no_of_days,max_no_of_people,cost_for_one_person,start_date,end_date,no_of_places,company_id)

VALUES ('720','STANDARD','6','100','1053','15-SEP-2016','21-SEP-2016','1','3');
```

```
INSERT INTO
tour_package(package_id,tour_name,no_of_days,max_no_of_people,cost_for_one_person,start_date,end_date,no_of_places,company_id)

VALUES ('730','DELUXE','3','20','2000','01-AUG-2016','04-AUG-2016','2','3');


INSERT INTO feedback(feedback_description,person_id,package_id)

VALUES('good', '5214','700');

INSERT INTO feedback(feedback_description,person_id,package_id)

VALUES('enjoyed a lot in the trip', '5215','710');

INSERT INTO feedback(feedback_description,person_id,package_id)

VALUES('BOOKING EXPERIENCE WAS GREAT. HOPE TRIP IS GOOD', '5216','720');



INSERT INTO HOTEL (Hotel_id,phone_no,hotelname,city_id)

VALUES('1','7039737336','HOLIDAY INN','100');

INSERT INTO HOTEL (Hotel_id,phone_no,hotelname,city_id)

VALUES('2','2145789856','HILTON','110');

INSERT INTO HOTEL (Hotel_id,phone_no,hotelname,city_id)

VALUES('3','123456789','QUALITY INN','120');

INSERT INTO HOTEL (Hotel_id,phone_no,hotelname,city_id)

VALUES('4','1230456954','MARRIOTT','130');


INSERT INTO hotel_booking_details(hotel_booking_date,booking_number,Hotel_id,person_id)

VALUES('01-MAR-2015','1','1','5214');

INSERT INTO hotel_booking_details(hotel_booking_date,booking_number,Hotel_id,person_id)

VALUES('01-FEB-2016','2','2','5215');

INSERT INTO hotel_booking_details(hotel_booking_date,booking_number,Hotel_id,person_id)

VALUES('08-APR-2016','3','3','5216');
```

```
INSERT INTO hotel_booking_details(hotel_booking_date,booking_number,Hotel_id,person_id)
VALUES('10-APR-2015','4','4','5214');


INSERT INTO package_cities(city_id,package_id)
VALUES('100','700');
INSERT INTO package_cities(city_id,package_id)
VALUES('110','710');
INSERT INTO package_cities(city_id,package_id)
VALUES('120','720');
INSERT INTO package_cities(city_id,package_id)
VALUES('130','730');
INSERT INTO package_cities(city_id,package_id)
VALUES('120','730');


INSERT INTO Flight_booking_details(seat_no,booking_id,date_of_travel,cost_of_journey,person_id,flight_number)
VALUES ('28','1','15-JUN-2016','200','5214','UUZ877');
INSERT INTO Flight_booking_details(seat_no,booking_id,date_of_travel,cost_of_journey,person_id,flight_number)
VALUES ('31','2','15-JUN-2017','300','5215','E845DE');
INSERT INTO Flight_booking_details(seat_no,booking_id,date_of_travel,cost_of_journey,person_id,flight_number)
VALUES ('28','3','15-SEP-2016','256','5216','YY21DA');
INSERT INTO Flight_booking_details(seat_no,booking_id,date_of_travel,cost_of_journey,person_id,flight_number)
VALUES ('1','4','01-AUG-2016','358','5214','YY21D7');
```

```
INSERT INTO
Flight_booking_details(seat_no,booking_id,date_of_travel,cost_of_journey,person_id,flight_number)

VALUES ('1','5','03-AUG-2016','358','5214','YY21AA');


INSERT INTO guide_travel_agency(person_id,company_id)

VALUES('5217','1');

INSERT INTO guide_travel_agency(person_id,company_id)

VALUES('5218','2');

INSERT INTO guide_travel_agency(person_id,company_id)

VALUES('5218','3');

INSERT INTO guide_travel_agency(person_id,company_id)

VALUES('5219','2');
```

Procedure1:

This is the procedure to get the names of hotels,persons booked in past 6 months.

```
CREATE OR REPLACE PROCEDURE SIX_MONTH_HOTEL /*HOTEL BOOKED IN PAST 6 MONTHS*/

IS

CURSOR HOTEL_DETAILS IS

   SELECT NAME, HOTELNAME

  FROM HOTEL H , HOTEL_BOOKING_DETAILS H2, PERSON P

  WHERE TRUNC(H2.HOTEL_BOOKING_DATE)>=TRUNC(SYSDATE-180)  AND H2.HOTEL_ID=H.HOTEL_ID
AND P.PERSON_ID=H2.person_ID;

THISNAME VARCHAR(30);

THISHOTELNAME VARCHAR(200);

BEGIN

OPEN HOTEL_DETAILS;

LOOP

  FETCH HOTEL_DETAILS INTO  THISNAME, THISHOTELNAME;

    EXIT WHEN (HOTEL_DETAILS%NOTFOUND);

   DBMS_OUTPUT.PUT_LINE('Name of customer is:'||THISNAME||' and Name of hotel is:
'||THISHOTELNAME);

END LOOP;

CLOSE HOTEL_DETAILS;

END;

/


SET SERVEROUTPUT ON

EXECUTE SIX_MONTH_HOTEL;
```

Procedure 2:

This is the procedure to get the names of cities which are visited by the customers for the given package organized by the particular travel agency.

```
create or replace procedure citiess(agencyname in char, packagename in varchar) as

cursor cities is select p.name person_name,c1.city_name,a.name , t.tour_name

from  customer c,person p,travel_agency a,book_tour b, tour_package t ,cities
c1,package_cities p1

where  b.package_id=t.package_id and  t.company_id=a.company_id and a.name=agencyname
and

t.tour_name=packagename and c.person_id=b.person_id and

c.person_id=p.person_id and t.package_id=p1.package_id and p1.city_id=c1.city_id;

cityy  cities%rowtype;

begin

open cities;

loop

fetch cities into cityy;

exit when cities%NOTFOUND;

dbms_output.put_line('travel agency name:  '|| cityy.name|| 'and

citynames:'||cityy.city_name||' and package name: '||cityy.tour_name||'and customer name:
'||cityy.person_name);

end loop;

close cities;

end;

/
```

```
call citiess('Moonlit Travels', 'asia');
```

Procedure 3:

To get the most popular(most booked) package name and the customer names who booked that package.

```
Create or replace procedure TOP_PACKAGE
IS
CURSOR PACKAGE IS
select t1.tour_name, p.name from (select pkg_id
from
(
select p.package_id pkg_id,
count(*) cnt
from tour_package p, book_tour b
where b.transaction_id not in (select c.transaction_id from cancel_tour c) and
p.package_id=b.package_id
group by p.package_id
)
where
cnt=(
select max(cnt) from
(
select p.package_id pkg_id,
count(*) cnt
from tour_package p, book_tour b
where b.transaction_id not in (select c.transaction_id from cancel_tour c) and
p.package_id=b.package_id
group by p.package_id
)
))t, book_tour b,person p ,tour_package t1
where t.pkg_id=b.package_id and  p.person_id=b.person_id and t1.package_id=t.pkg_id;

thistourname varchar(150);
thisname varchar(200);
BEGIN
OPEN package;

LOOP
  FETCH package INTO thistourname,thisname;
  EXIT WHEN (package%NOTFOUND);

   dbms_output.put_line('tour name:'|| THIStourNAME||' customer name: '|| THISname);
END LOOP;
CLOSE package;
END;

/

SET SERVEROUTPUT ON
EXECUTE top_package;
```

Trigger 1:

This is the trigger to delete the transactions permanently from the database if the no of cancellations in the cancelled table is more than 5000.

```
CREATE or replace  TRIGGER deletion

BEFORE insert ON cancel_tour

      for each row

      declare

    thisnumber int;

 begin

 SELECT count(*)into thisnumber

  FROM cancel_tour c;


   if(thisnumber>5000)then delete  from book_tour b  where b.transaction_id in

   (select c.transaction_id from cancel_tour c, book_tour b where
c.transaction_id=b.transaction_id);

   end if;

    if (thisnumber>5000)

   then

   delete from cancel_tour c;

   end if;

   end;
```

Trigger 2:

This is the trigger to increase the bonus of the staff if the corresponding company have 10000 bookings.

```
CREATE OR REPLACE TRIGGER bonus_updates
  before INSERT
  ON book_tour
FOR EACH ROW
BEGIN
   update staff s
  set bonus_per_year=bonus_per_year+'5000'
  where s.company_id in
   (select p.company_id
   from
   book_tour b, tour_package p
   where b.transaction_id not in (select c.transaction_id from cancel_tour c) and
b.package_id=p.package_id
   group by p.company_id
   having count(p.company_id)=1000;
```