

CSE 305 Project  
Team:

# Where To Go!

--Programmer's Guide--

Nan Jiang

[nan.jiang.1@stonybrook.edu](mailto:nan.jiang.1@stonybrook.edu)

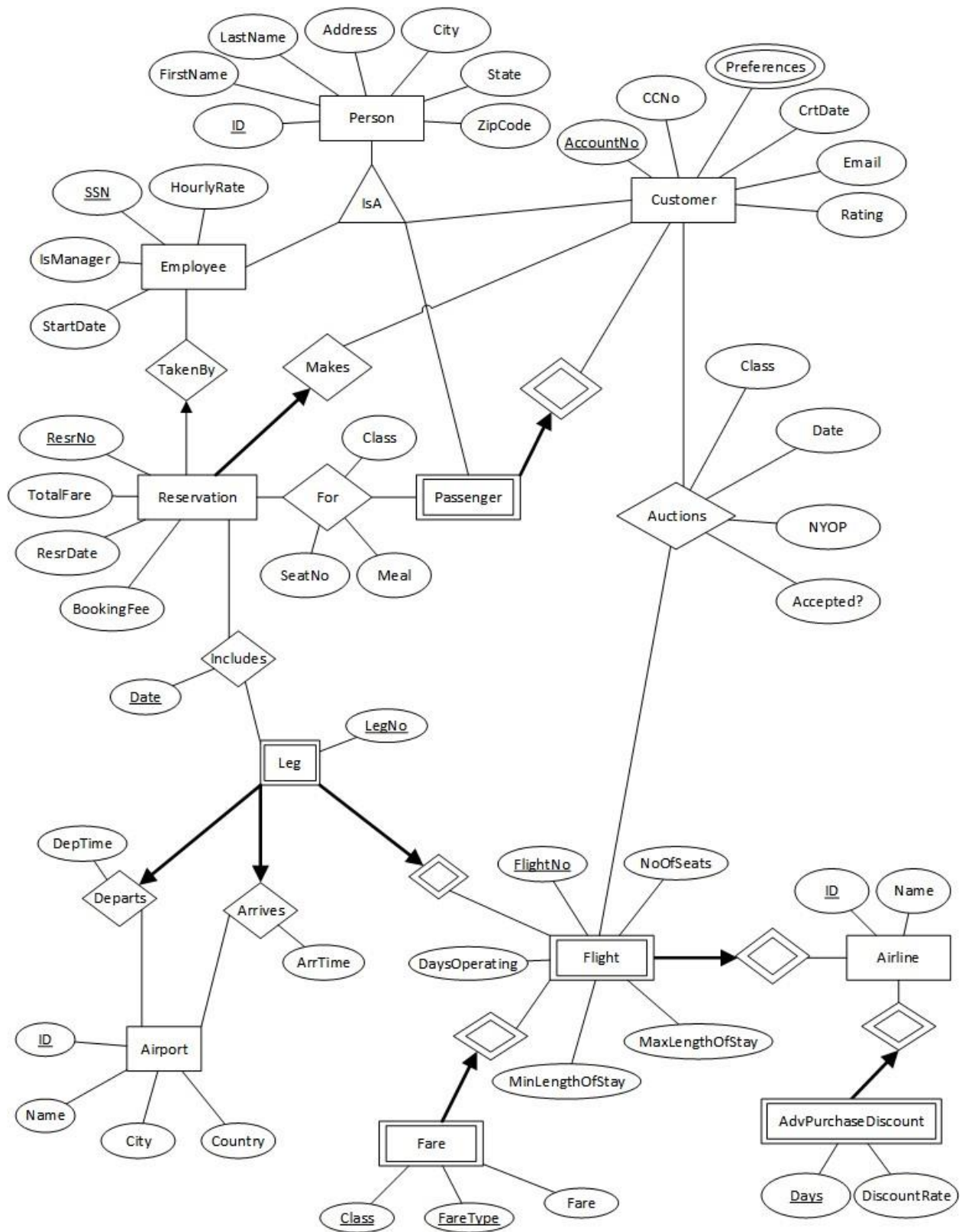
Zijun He

[zijun.he@stonybrook.edu](mailto:zijun.he@stonybrook.edu)

Yuege Chen

[yuege.chen@stonybrook.edu](mailto:yuege.chen@stonybrook.edu)

- ER Diagram



- **Relational Model**

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
CREATE TABLE IF NOT EXISTS airline (  
  Id CHAR(2),  
  Name VARCHAR(100) NOT NULL,  
  PRIMARY KEY (Id));
```

```
CREATE TABLE IF NOT EXISTS AdvPurchaseDiscount (  
  AirlineID CHAR(2),  
  Days INTEGER NOT NULL,  
  DiscountRate NUMERIC(10,2) NOT NULL,  
  PRIMARY KEY (AirlineID, Days),  
  FOREIGN KEY (AirlineID) REFERENCES Airline(Id)  
  ON DELETE CASCADE,  
  CHECK (Days > 0),  
  CHECK (DiscountRate > 0 AND DiscountRate < 100));
```

```
CREATE TABLE IF NOT EXISTS Flight (  
  AirlineID CHAR(2),  
  FlightNo INTEGER NOT NULL,  
  NoOfSeats INTEGER NOT NULL,  
  DaysOperating CHAR(7) NOT NULL,  
  MinLengthOfStay INTEGER NOT NULL,  
  MaxLengthOfStay INTEGER NOT NULL,  
  OnTime Boolean Not Null,  
  PRIMARY KEY (AirlineID, FlightNo),  
  FOREIGN KEY (AirlineID) REFERENCES Airline(Id)
```

```
ON DELETE CASCADE,  
CHECK (NoOfSeats > 0),  
CHECK (MinLengthOfStay >= 0),  
CHECK (MaxLengthOfStay > MinLengthOfStay)  
);
```

```
CREATE TABLE IF NOT EXISTS Airport (  
Id CHAR(3),  
Name VARCHAR(100) NOT NULL,  
City VARCHAR(50) NOT NULL,  
Country VARCHAR(50) NOT NULL,  
PRIMARY KEY (Id));
```

```
CREATE TABLE IF NOT EXISTS Leg (  
AirlineID CHAR(2),  
FlightNo INTEGER NOT NULL,  
LegNo INTEGER NOT NULL,  
DepAirportID CHAR(3) NOT NULL,  
ArrAirportID CHAR(3) NOT NULL,  
ArrTime DATETIME NOT NULL,  
DepTime DATETIME NOT NULL,  
PRIMARY KEY (AirlineID, FlightNo, LegNo),  
UNIQUE(AirlineID, FlightNo, DepAirportID),  
FOREIGN KEY (AirlineID, FlightNo) REFERENCES Flight(AirlineID, FlightNo)  
ON DELETE CASCADE,  
FOREIGN KEY (DepAirportID) REFERENCES Airport(Id)  
ON DELETE CASCADE,  
FOREIGN KEY (ArrAirportID) REFERENCES Airport(Id)
```

```
ON DELETE CASCADE,  
CHECK (LegNo > 0)  
);
```

```
CREATE TABLE IF NOT EXISTS Fare (  
AirlineID CHAR(2) NOT NULL,  
FlightNo INTEGER NOT NULL,  
FareType VARCHAR(20) NOT NULL,  
Class VARCHAR(20) NOT NULL,  
Fare NUMERIC(10,2) NOT NULL,  
PRIMARY KEY (AirlineID, FlightNo, FareType, Class),  
FOREIGN KEY (AirlineID, FlightNo) REFERENCES Flight(AirlineID, FlightNo)  
ON DELETE CASCADE,  
CHECK (Fare > 0)  
);
```

```
CREATE TABLE IF NOT EXISTS Person (  
Id INTEGER,  
FirstName VARCHAR(50) NOT NULL,  
LastName VARCHAR(50) NOT NULL,  
Address VARCHAR(100) NOT NULL,  
City VARCHAR(50) NOT NULL,  
State VARCHAR(50) NOT NULL,  
ZipCode INTEGER NOT NULL,  
PRIMARY KEY (Id),  
CHECK (Id > 0),  
CHECK (ZipCode > 0)  
);
```

```
CREATE TABLE IF NOT EXISTS Customer (  
  Id INTEGER NOT NULL,  
  AccountNo INTEGER,  
  CreditCardNo CHAR(16),  
  Phone VARCHAR(15),  
  Email VARCHAR(50),  
  CreationDate DateTime NOT NULL,  
  Rating INTEGER,  
  PRIMARY KEY (AccountNo),  
  FOREIGN KEY (Id) REFERENCES Person (Id)  
  ON DELETE CASCADE,  
  CHECK (Rating >= 0 AND Rating <= 10)  
);
```

```
CREATE TABLE IF NOT EXISTS CustomerPreferences(  
  AccountNo INTEGER NOT NULL,  
  Preference VARCHAR(50) NOT NULL,  
  PRIMARY KEY (AccountNo, Preference),  
  FOREIGN KEY (AccountNo) REFERENCES Customer (AccountNo)  
  ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Employee (  
  Id INTEGER NOT NULL,  
  SSN INTEGER,  
  IsManager BOOLEAN NOT NULL,  
  StartDate DATE NOT NULL,
```

HourlyRate NUMERIC(10,2) NOT NULL,  
PRIMARY KEY (SSN),  
FOREIGN KEY (Id) REFERENCES Person (Id)  
ON DELETE CASCADE,  
UNIQUE (Id),  
CHECK (SSN > 0),  
CHECK (HourlyRate > 0)  
);

CREATE TABLE IF NOT EXISTS Passenger (  
Id INTEGER,  
AccountNo INTEGER,  
Phone VARCHAR(15),  
Email VARCHAR(50),  
PRIMARY KEY (Id, AccountNo),  
FOREIGN KEY (Id) REFERENCES Person(Id)  
ON DELETE CASCADE,  
FOREIGN KEY (AccountNo) REFERENCES Customer(AccountNo)  
ON DELETE CASCADE  
);

CREATE TABLE IF NOT EXISTS Reservation (  
ResrNo INTEGER,  
ResrDate DATETIME NOT NULL,  
BookingFee NUMERIC(10,2) NOT NULL,  
TotalFare NUMERIC(10,2) NOT NULL,  
RepSSN INTEGER,  
AccountNo INTEGER NOT NULL,

PRIMARY KEY (ResrNo),  
FOREIGN KEY (RepSSN) REFERENCES Employee (SSN)  
ON DELETE CASCADE,  
FOREIGN KEY (AccountNo) REFERENCES Customer (AccountNo)  
ON DELETE CASCADE,  
CHECK (ResrNo > 0),  
CHECK (BookingFee >= 0),  
CHECK (TotalFare > BookingFee)  
);

CREATE TABLE IF NOT EXISTS Includes (  
ResrNo INTEGER,  
AirlineID CHAR(2),  
FlightNo INTEGER,  
LegNo INTEGER,  
Date DATE NOT NULL,  
PRIMARY KEY (ResrNo, AirlineID, FlightNo, LegNo),  
FOREIGN KEY (ResrNo) REFERENCES Reservation (ResrNo)  
ON DELETE CASCADE,  
FOREIGN KEY (AirlineID, FlightNo, LegNo) REFERENCES Leg(AirlineID, FlightNo,  
LegNo)  
ON DELETE CASCADE  
);

CREATE TABLE IF NOT EXISTS ReservationPassenger (  
ResrNo INTEGER,  
Id INTEGER,  
AccountNo INTEGER,  
SeatNo varchar(5) NOT NULL,



```
Class VARCHAR(20) NOT NULL,  
Meal VARCHAR(50),  
PRIMARY KEY (ResrNo, Id, AccountNo),  
FOREIGN KEY (ResrNo) REFERENCES Reservation (ResrNo)  
ON DELETE CASCADE,  
FOREIGN KEY (Id, AccountNo) REFERENCES Passenger (Id, AccountNo)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS Auctions (  
AccountNo INTEGER,  
AirlineID CHAR(2),  
FlightNo INTEGER,  
Class VARCHAR(20),  
Date DATETIME,  
NYOP NUMERIC(10,2) NOT NULL,  
Accepted BOOLEAN,  
PRIMARY KEY (AccountNo, AirlineID, FlightNo, Class, Date),  
FOREIGN KEY (AccountNo) REFERENCES Customer(AccountNo)  
ON DELETE CASCADE,  
FOREIGN KEY (AirlineID, FlightNo) REFERENCES Flight(AirlineID, FlightNo)  
ON DELETE CASCADE,  
CHECK (NYOP > 0)  
);
```

- **Insert Table**

INSERT INTO airline

(Id,Name)

values

('AB','Air Berlin'),

('AJ','Air Japan'),

('AM','Air Madagascar'),

('AA','American Airlines'),

('BA','British Airways'),

('DA','Delta Airlines'),

('JA','JetBlue Airways'),

('LU','Lufthansa'),

('SA','Southwest Airlines'),

('UA','United Airlines');

INSERT INTO Airport

(Id,Name,City,Country)

values

('TXL','Berlin Tegel','Berlin', 'Germany'),

('ORD','Chicago O'Hare International','Chicago','Illinois'),

('ATL','Hartsfield-Jackson Atlanta Int','Atlanta','United States of America'),

('TNR','Ivato International','Antananarivo','Madagascar'),

('JFK','John F. Kennedy International','New York','United States of America'),

('LGA','LaGuardia' , 'New York' , 'United States of America'),

('BOS','Logan International', 'Boston', 'United States of America'),

('LHR','London Heathrow' , 'London' , 'United Kingdom'),

('LAX','Los Angeles International' , 'Los Angeles' , 'United States of America'),

('SFO','San Francisco International' , 'San Francisco' , 'United States of America'),

```
('HND', 'Tokyo International' , 'Tokyo' , 'Japan');
```

```
INSERT INTO Person
```

```
(Id,FirstName,LastName,Address,City,State,ZipCode)
```

```
values
```

```
(1000000000,'Jane','Smith','100 Nicolls Rd','Stony Brook','New York',11790),
```

```
(1000000001,'John','Doe','123 N Fake Street','New York','New York',10001),
```

```
(1000000002,'Rick','Astley','1337 Internet Lane','Los Angeles','California',90001),
```

```
(2000000000,'Michael','Kierf','124 N Fake Street','New York','New York',10001),
```

```
(2000000001,'Annie','White','125 N Fake Street','New York','New York',10001),
```

```
(2000000002,'Christopher','Jon','126 N Fake Street','New York','New York',10001);
```

```
INSERT INTO employee
```

```
(Id,SSN,ismanager,startdate,hourlyrate)
```

```
values
```

```
(2000000000,2000000000,false,'2005-01-13',20.00),
```

```
(2000000001,2000000001,false,'2003-01-13',30.00),
```

```
(2000000002,2000000002,false,'2004-01-13',25.00);
```

```
INSERT INTO customer
```

```
(Id,AccountNo,CreditCardNo,Phone,Email,CreationDate,Rating)
```

```
values
```

```
(1000000000,1000000000,'5105105105105100','555-555-5555','awesomejane@ftw.com','2006-01-05 19:00:00',4),
```

```
(1000000001,1000000001,'4012888888881881','123-123-1234','jdoe@woot.com','2001-01-05 19:00:00',5),
```

```
(1000000002,1000000002,'4111111111111111','314-159-2653','rickroller@rolld.com','2008-01-05 19:00:00',3);
```

INSERT INTO Passenger

(Id,AccountNo,Phone,Email)

values

(100000000,100000000,'555-555-5555','awesomejane@ftw.com'),

(100000001,100000001,'123-123-1234','jdoe@woot.com'),

(100000002,100000002,'314-159-2653','rickroller@rolld.com');

INSERT INTO Flight

(AirlineID,FlightNo,NoOfSeats,DaysOperating,MinLengthOfStay,MaxLengthOfStay,Ontime)

values

('AA',111,100,'1010100',1,4,true),

('AA',117,100,'1010100',1,4,true),

('JA',111,150,'1111111',2,5,true),

('JA',110,150,'1111111',2,5,true),

('JA',112,150,'1111111',2,5,true),

('JA',212,150,'1111111',2,5,true),

('JA',211,150,'1111111',2,5,true),

('JA',210,150,'1111111',2,5,true),

('AM',1337,33,'0000011',3,6,true);

INSERT INTO Leg

(AirlineID,FlightNo,LegNo,DepAirportID,ArrAirportID,ArrTime,DepTime)

values

('AA',111,1,'LGA','LAX','2011-01-05 17:00:00','2011-01-05 11:00:00'),

('AA',111,2,'LAX','HND','2011-01-06 07:30:00','2011-01-05 19:00:00'),

('AA',117,2,'LAX','LGA','2011-01-15 17:00:00','2011-01-15 13:00:00'),

('AA',117,1,'HND','LAX','2011-01-15 10:30:00','2011-01-15 01:30:00'),

```

('JA',111,1,'SFO','BOS','2011-01-10 19:30:00','2011-01-10 14:00:00'),
('JA',110,1,'SFO','BOS','2011-01-09 19:30:00','2011-01-09 14:00:00'),
('JA',112,1,'SFO','BOS','2011-01-11 19:30:00','2011-01-11 14:00:00'),
('JA',111,2,'BOS','LHR','2011-01-11 05:00:00','2011-01-10 22:30:00'),
('JA',110,2,'BOS','LHR','2011-01-10 05:00:00','2011-01-09 22:30:00'),
('JA',112,2,'BOS','LHR','2011-01-12 05:00:00','2011-01-11 22:30:00'),
('JA',211,2,'BOS','SFO','2011-01-21 19:30:00','2011-01-21 14:00:00'),
('JA',210,2,'BOS','SFO','2011-01-20 19:30:00','2011-01-20 14:00:00'),
('JA',212,2,'BOS','SFO','2011-01-22 19:30:00','2011-01-22 14:00:00'),
('JA',211,1,'LHR','BOS','2011-01-21 05:00:00','2011-01-20 22:30:00'),
('JA',210,1,'LHR','BOS','2011-01-20 05:00:00','2011-01-19 22:30:00'),
('JA',212,1,'LHR','BOS','2011-01-22 05:00:00','2011-01-21 22:30:00'),
('AM',1337,1,'JFK','TNR','2011-01-13 23:00:00','2011-01-13 07:00:00');

```

INSERT INTO Reservation

(ResrNo,ResrDate,BookingFee,TotalFare,RepSSN,AccountNo)

values

```

(111,'2011-01-01 11:00:00',120.00,1200.00,2000000001,1000000001),
(222,'2011-01-01 11:00:00',50.00,500.00,2000000000,1000000000),
(333,'2011-01-01 11:00:00',333.33,3333.33,2000000002,1000000002);

```

INSERT INTO Includes

(ResrNo,AirlineID,FlightNo,LegNo,Date)

values

```

(111,'AA',111,1,'2011-01-05'),
(111,'AA',111,2,'2011-01-05'),
(222,'JA',111,2,'2011-01-10'),
(333,'AM',1337,1,'2011-01-13');

```

INSERT INTO Fare

(AirlineID,FlightNo,FareType,Class,Fare)

values

('AA',111,'one-way','first',1200.00),  
('AA',117,'one-way','first',1250.00),  
('JA',111,'one-way','economy',500.00),  
('JA',110,'one-way','economy',550.00),  
('JA',112,'one-way','economy',575.00),  
('JA',212,'one-way','economy',575.00),  
('JA',210,'one-way','economy',550.00),  
('JA',211,'one-way','economy',500.00),  
('AM',1337,'one-way','first',3333.33);

INSERT INTO ReservationPassenger

(ResrNo,id,AccountNo,seatNo,Class,Meal)

values

(111,100000000,100000000,'33F ','economy','Chips'),  
(222,100000001,100000001,'13A ','first','Fish and Chips'),  
(333,100000002,100000002,'1A ','first','Sushi');

INSERT INTO Auctions

(AccountNo,AirlineID,FlightNo,Class,Date,NYOP,Accepted)

values

(100000001,'AA',111,'economy','2011-01-05 11:00:00',400.00,true);

- **SQL Transactions**

—— 3.1 ——

-- add information for an employee

-- before add

select e.ssn

from employee e;

-- first add a person

START TRANSACTION;

INSERT INTO Person

(Id,FirstName,LastName,Address,City,State,ZipCode)

values

(100000003,'Zijun','He','23 willams blvd','Lake Grove','New York',11755);

-- then add this person to employee

INSERT INTO Employee

(Id,SSN,IsManager,StartDate,HourlyRate)

values

(100000003,123456789,false,'2000-09-14',100.00);

COMMIT;

-- after add

select e.ssn

from employee e;

-- edit information(hourlyrate) for an employee

START TRANSACTION;

UPDATE Employee

SET HourlyRate = 50.00

WHERE SSN = 123456789;

COMMIT;

select e.ssn,e.hourlyrate,e.IsManager

from employee e;

-- edit information(ismanager) for an employee

START TRANSACTION;

UPDATE Employee

SET IsManager = true

WHERE SSN = 123456789;

COMMIT;

-- before delete

select e.ssn

from employee e;

-- delete information for an employee

START TRANSACTION;

DELETE FROM person

WHERE id = 100000003;

COMMIT;

-- after delete

select e.ssn

from employee e;

-- obtain a sales report for 2011 January

START TRANSACTION;

SELECT R.ResrNo, R.TotalFare

FROM Reservation R

WHERE R.ResrDate >= '2011-01-01 00:00:00' AND R.ResrDate < '2011-02-01 00:00:00';

COMMIT;

-- produce a comprehensive listing of all flights

START TRANSACTION;



```
SELECT
F.AirlineID,F.FlightNo,F.NoOfSeats,F.DaysOperating,F.MinLengthOfStay,F.MaxLengthOf
Stay
FROM flight F;
COMMIT;
```

-- Produce a list of reservations by flight number AA111

```
START TRANSACTION;

SELECT distinct R.ResrNo,R.ResrDate,R.BookingFee,R.TotalFare,R.RepSSN,R.AccountNo
FROM Reservation R, Includes I
WHERE I.ResrNo = R.ResrNo AND I.FlightNo = 111 AND I.AirlineID = 'AA';
COMMIT;
```

-- Produce a list of reservations by customer name Jane Smith

```
START TRANSACTION;

SELECT R.ResrNo,R.ResrDate,R.BookingFee,R.TotalFare,R.RepSSN,R.AccountNo
FROM Reservation R, Customer C, Person P
WHERE C.AccountNo = R.AccountNo AND C.Id = P.Id AND P.FirstName = 'Jane' AND
P.LastName = 'Smith';
COMMIT;
```

-- Produce a summary listing of revenue generated by a particular flight AA111

```
START TRANSACTION;

SELECT distinct R.ResrNo,R.TotalFare
FROM Reservation R,Includes I, Leg L
WHERE R.ResrNo = I.ResrNo AND I.AirlineID = L.AirlineID AND I.FlightNo =
L.FlightNo AND I.LegNo = L.LegNo AND L.FlightNo = 111 AND L.AirlineID = 'AA';
COMMIT;
```

-- Produce a summary listing of revenue generated by a destination city Los Angeles

```
START TRANSACTION;

SELECT distinct R.ResrNo,R.TotalFare

FROM Reservation R,Includes I, Leg L, Airport A

WHERE R.ResrNo = I.ResrNo AND I.AirlineID = L.AirlineID AND I.FlightNo =
L.FlightNo AND I.LegNo = L.LegNo AND L.ArrAirportID = A.Id AND A.City = 'Los
Angeles';

COMMIT;
```

-- Produce a summary listing of revenue generated by a customer Jane Smith

```
START TRANSACTION;

SELECT distinct R.ResrNo,R.TotalFare

FROM Reservation R, Customer C, Person P

WHERE C.AccountNo = R.AccountNo AND C.Id = P.Id AND P.FirstName = 'Jane' AND
P.LastName = 'Smith';

COMMIT;
```

-- Determine which customer representative generated most total revenue

```
START TRANSACTION;

create view eRevenue(Id,FirstName,LastName,SumRevenue) as

SELECT E.Id, P.FirstName, P.LastName, SUM(R.TotalFare)

FROM Reservation R, Employee E, Person P

WHERE R.RepSSN = E.SSN AND E.Id = P.Id

GROUP BY concat(E.Id,'-', FirstName,'-', LastName);

select er.Id, er.FirstName, er.LastName, max(er.SumRevenue)

From eRevenue er;

COMMIT;
```

-- Determine which customer generated most total revenue

```
START TRANSACTION;
```

```

create view cRevenue(Id,FirstName,LastName,SumRevenue) as
SELECT C.Id, P.FirstName, P.LastName, SUM(R.TotalFare)
FROM Reservation R, Customer C, Person P
WHERE C.AccountNo = R.AccountNo AND C.Id = P.Id
GROUP BY concat(C.Id,'-', FirstName,'-', LastName);
select cr.Id, cr.FirstName, cr.LastName, max(cr.SumRevenue)
From cRevenue cr;
COMMIT;

```

-- Produce a list of most active flights

```

START TRANSACTION;
create view flightActive (airlineId,flightNo,active) as
SELECT f.AirlineID,f.FlightNo,
SUBSTR(f.DaysOperating,1,1)+SUBSTR(f.DaysOperating,2,1)+SUBSTR(f.DaysOperating,
3,1)+SUBSTR(f.DaysOperating,4,1)+
SUBSTR(f.DaysOperating,5,1)+SUBSTR(f.DaysOperating,6,1)+SUBSTR(f.DaysOperating,
7,1)
FROM Flight f;
select fa.airlineid,fa.flightNo,fa.active
from flightActive fa
order by active desc;
commit;

```

-- Produce a list of all customers who have seats reserved on a given flight AA111

```

START TRANSACTION;
SELECT distinct C.Id, P.FirstName, P.LastName
FROM ReservationPassenger RP,Reservation R, Customer C, Person P, Includes I
WHERE RP.ResrNo = R.ResrNo AND C.AccountNo = R.AccountNo AND C.Id = P.Id
AND I.ResrNo = R.ResrNo AND I.FlightNo = 111 AND I.AirlineID = 'AA';
COMMIT;

```

-- Produce a list of all flights for a given airport JFK

START TRANSACTION;

SELECT distinct concat(L.AirlineID,'-',L.FlightNo)

FROM Leg L,Includes I

WHERE I.AirlineID = L.AirlineID AND I.FlightNo = L.FlightNo AND I.LegNo = L.LegNo  
AND (L.DepAirportID = 'JFK' OR L.ArrAirportID = 'JFK') ;

commit;

-- Produce a list of all flights whose arrival and departure times are on-time/delayed

START TRANSACTION;

select f.airlineid,f.flightNo

from flight f

where f.ontime=true;

commit;

### —— 3.2 ——

-- Record A reservation, for example ResrNo = 444

-- before add

Select r.ResrNo,r.ResrDate,r.BookingFee,r.TotalFare,r.RepSSN,r.AccountNo

From Reservation r;

START TRANSACTION;

INSERT INTO Reservation

(ResrNo,ResrDate,BookingFee,TotalFare,RepSSN,AccountNo)

values

(444,'2011-01-01 11:00:00',400.00,4000.00,2000000000,1000000000);

INSERT INTO Includes

(ResrNo,AirlineID,FlightNo,LegNo,Date)

values

(444,'AA',111,1,'2011-01-05');

INSERT INTO ReservationPassenger

(ResrNo,id,AccountNo,seatNo,Class,Meal)

values

(444,100000000,100000000,'65A ','economy','Chips');

COMMIT;

-- output

Select r.ResrNo,r.ResrDate,r.BookingFee,r.TotalFare,r.RepSSN,r.AccountNo

From Reservation r;

-- Add, Edit and Delete information for a customer

-- 1.Add

-- before add

Select c.Id, c.AccountNo,c.CreditCardNo,c.Phone,c.Email,c.CreationDate,c.Rating

From Customer c;

-- add a new Customer

START TRANSACTION;

INSERT INTO Person

(Id,FirstName,LastName,Address,City,State,ZipCode)

values

(1000000004,'Nan','Jiang','400 Nicolls Rd','Stony Brook','New York',11790);

INSERT INTO Customer

(Id,AccountNo,CreditCardNo,Phone,Email,CreationDate,Rating)

values

```
(1000000004,1000000004,'5105104155105100','762-555-8767','nan.jiang@google.com','2007-01-05 19:00:00',5);
```

```
COMMIT;
```

```
-- output
```

```
Select c.Id, c.AccountNo,c.CreditCardNo,c.Phone,c.Email,c.CreationDate,c.Rating
```

```
From Customer c;
```

```
-- 2.Edit
```

```
-- before edit
```

```
Select c.id,c.Phone
```

```
From Customer c
```

```
Where c.Id = 1000000000;
```

```
-- edit customer id = 1000000000's phone number from '555-555-5555' to '555-555-6875'
```

```
START TRANSACTION;
```

```
Update Customer
```

```
Set Phone = '555-555-6875'
```

```
Where Id = 1000000000;
```

```
COMMIT;
```

```
-- output
```

```
Select c.Id,c.Phone
```

```
From Customer c
```

```
Where c.Id = 1000000000;
```

```
-- 3.Delete
```

```
-- before delete
```

```
Select c.Id
```

```
From Customer c;
```

```
-- delete a given customer, here we delete Rick Astley (Id = 1000000002)
```

```
START TRANSACTION;
```

```
DELETE FROM person
```

```
Where Id = 1000000004;
```

COMMIT;

-- OUTPUT

Select c.Id

From Customer c;

-- Produce customer mailing lists: LIST ALL CUSTOMER NAME AND ADDRESS

START TRANSACTION;

Select p.FirstName, p.LastName, p.Address, p.City, p.State, p.ZipCode

From Customer c, Person p

Where c.Id = p.Id;

COMMIT;

-- Produce a list of flight suggestions for a given customer (based on that customer's past reservations)

-- i.e for Rick Astley(100000002). we generate the list based on

-- the frequency the customer took that flight

START TRANSACTION;

select i.airlineId, i.flightno, count(distinct i.ResrNo) as frequency

from includes i, reservation r

where r.AccountNo=100000002 and r.ResrNo=i.ResrNo

group by concat(AirlineID,'-',FlightNo)

order by frequency desc;

COMMIT;

### —— 3.3 ——

-- One-Way Round-Trip Multi-City Domestic or International Flexible Date/time

-- NOTE: the prof replied on piazza that the logic is on application level, here we just provide some nessary sql code to select a specified flight.

-- Here we jsut use flight from John F. Kennedy International to Ivato International as an example

START TRANSACTION;

select l.airlineId, l.flightno

from leg l

where l.DepAirportID='JFK' and l.ArrAirportID='TNR';

COMMIT;

-- before delete

select r.ResrNo

from reservation r;

-- cancel an existing reservation, here we cancel John Doe's reservation 111.

START TRANSACTION;

DELETE FROM reservation

WHERE ResrNo = 444;

COMMIT;

-- output

select r.ResrNo

from reservation r;

-- retrieve a customer's current reservations, here we use Jane Smith(accountNum=100000000) as an example.

START TRANSACTION;

select r.ResrNo

From reservation r,includes i

WHERE r.AccountNo=100000000 and r.ResrNo=i.ResrNo and i.Date>='2011-01-04' ;--  
assume today is 2011-01-04

COMMIT;



-- Travel itinerary for a given reservation, i.e, reservation 111.

START TRANSACTION;

select l.airlineId,l.flightno,l.legno,l.DepAirportID,l.DepTime,l.ArrAirportID,l.ArrTime

From includes i,leg l

WHERE i.ResrNo=111 AND l.AirlineID=i.AirlineID and l.FlightNo=i.FlightNo and  
l.legno=i.legno;

COMMIT;

-- A customer's current bid on a given reverse auction. i.e, John Doe's bid on flight AA111

START TRANSACTION;

select a.NYOP

From auctions a

WHERE a.AccountNo=100000001 and a.AirlineID='AA' and a.FlightNo=111 and  
a.Class='economy'

and a.Date> '2011-01-04 11:00:00' ;-- assume now is 2011-01-04 11:00:00;

COMMIT;

-- The bid history for a given reverse auction. i.e, bid history on flight AA111

START TRANSACTION;

select a.AccountNo, a.NYOP

From auctions a

WHERE a.AirlineID='AA' and a.FlightNo=111 and a.Class='economy' and a.Date='2011-01-  
05 11:00:00';

COMMIT;

-- A history of all current and past reservations a customer has made, i.e, Jane  
Smith(accountNum=100000000).

START TRANSACTION;

select r.ResrNo

```
From reservation r
WHERE r.AccountNo=100000000;
COMMIT;
```

-- Best-Seller list of flights

```
START TRANSACTION;
select i.airlineId, i.flightno, count(distinct i.ResrNo) as popularity
from includes i
group by concat(AirlineID,'-',FlightNo)
order by popularity desc;
COMMIT;
```

-- Personalized flight suggestion list, i.e for Rick Astley(1000000002). we generate the list based on

-- the frequency the customer took that flight

```
START TRANSACTION;
select i.airlineId, i.flightno, count(distinct i.ResrNo) as frequency
from includes i, reservation r
where r.AccountNo=1000000002 and r.ResrNo=i.ResrNo
group by concat(AirlineID,'-',FlightNo)
order by frequency desc;
COMMIT;
```

- **Description of interaction between user interface and database server**

From the Browser, it send request and we give the home page which is **index.jsp**, and user could login from **login.jsp** and that will lead to **loginServlet**. In **loginServlet**, it will first distinguish the login is Customer or Employee, and with the value of **IsManager**, it will contact the database by different query and return different result. With the different result, it will lead to **index2.jsp** (Customer) or **Customer\_Rep.jsp** (Customer representative) or **manager.jsp** (Manager).

In **index2.jsp** (Customer) can do:

- One-Way Search
- Round-Trip Search
- Multi-City Search
- Domestic or International Search
- Flexible Date/time Search
- User Center

In **Customer\_Rep.jsp** (Customer representative) can do:

- Record a reservation
- Add, Edit and Delete information for a customer
- Check customer mailing lists
- Produce a list of flight suggestions for a given customer (based on that customer's past reservations)

In **manager.jsp** (Manager).can do:

- Add, Edit and Delete information for an employee
- Obtain a sales report for a particular month
- Produce a comprehensive listing of all flights
- Produce a list of reservations by flight number or by customer name
- Produce a summary listing of revenue generated by a particular flight, destination city, or customer
- Determine which customer representative generated most total revenue
- Determine which customer generated most total revenue
- Produce a list of most active flights
- Produce a list of all customers who have seats reserved on a given flight
- Produce a list of all flights for a given airport
- Produce a list of all flights whose arrival and departure times are on-time/delayed

