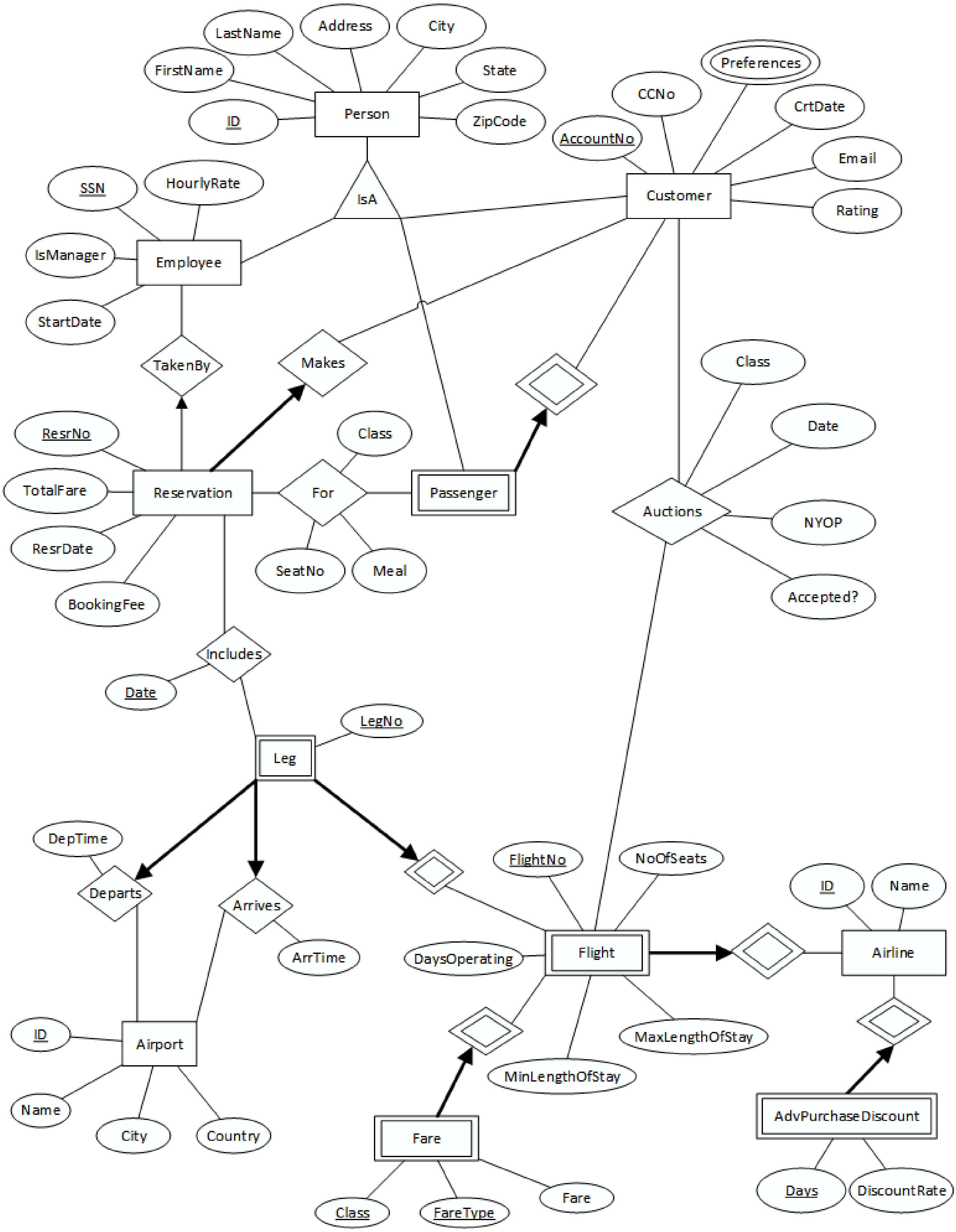
E-R Diagram


LastName
Address
City
FirstName
State
ID
Person
ZipCode
Preferences
CCNo
CrtDate
AccountNo
Email
Customer
Rating
SSN
HourlyRate
IsA
IsManager
Employee
StartDate
TakenBy
Makes
Class
ResrNo
Class
Date
TotalFare
Reservation
For
Passenger
Auctions
NYOP
ResrDate
SeatNo
Meal
BookingFee
Accepted?
Includes
Date
LegNo
Leg
DepTime
FlightNo
NoOfSeats
Departs
Arrives
ID
Name
ArrTime
DaysOperating
Flight
Airline
MaxLengthOfStay
ID
Airport
MinLengthOfStay
Name
City
Country
Fare
AdvPurchaseDiscount
Class
FareType
Fare
Days
DiscountRate

Notes:

1. Please see section 4.4.2 in the coursebook for line-based representation vs. cardinality constraints.
2. Double boxes (**Flight**, **Leg**, **AdvPurchaseDiscount**, **Fare**, **Passenger**) represent weak entities. Please refer to section 4.4.3 in the book for more information.
3. **NYOP** stands for *Name Your Own Price* that means the price a customer sets for an auction that he or she participates in.
4. For simplification, we assume that an auction is for one seat of one flight.
5. For simplification, we assume that a passenger will have the same seat number for every leg.
6. We need a separate **Passenger** entity because a customer can buy a ticket for another person who does not have an account. For example, a customer can buy a ticket for his 6 year old son.
7. A flight can have different fare types: one-way, roundtrip, minimum fare that airlines will accept in reverse auction (*hidden fare*, hidden from customers). Fares also depend on the cabin class.
8. The above diagram assumes that a seat can be of any class.
9. Auction: a customer can specify a price when bidding on an auction. If that price is lower than the hidden fares, the bid is rejected. If the price meets or exceeds the hidden fares then the bid is accepted. If a bid is accepted, a reservation will be recorded and the auction ends. If a bid is rejected, the customer can bid again with a higher price as long as the bid is before the departure date and the flights still have enough seats.
10. **'DaysOperating'** is a 7 bit string representing which days of the week the flight operates from the STARTING CITY (i.e., FIRST STOP). For example, if Continent flight #700 flies only on Sundays and Wednesdays, then the **DaysOperating** attribute value would be **1001000**.
11. Each flight has associated with it a sequence of stops. This information is stored in the entity **'Leg'**. For example, an entity with attributes "**Leg**(*Flight#=700, AirlineID=AA, LegNo=2*)" participates in relationship "**Departs**" with entity "**Airport**(*ID=DFW*, …)" and relationship "**Arrives**" with entity "**Airport**(*ID=LGA*, …)" signifies that American Airline Flight number 700's *second* leg departs from Dallas Fort Worth Airport and arrives at LaGuardia Airport.
12. The E-R Diagram assumes that each airline has an advance purchase discount program that applies to all of its flights. For example, an entity "**AdvPurchaseDiscount**(*AirlineID = AA, Days = 7, DiscountRate = 10*)" signifies that a discount rate of 10% will be applied to all 7-day advance purchase for American Airlines flights. When calculating discount, the entity instance with the max Days attribute should be used.
13. The E-R Diagram assumes that there are no circles in a flight's leg sequence, i.e., legs of a flight have a unique departure airport.
14. For simplification, the E-R Diagram assumes that a flight has fares for the whole flight, not for individual leg. For example, AA flight 700 stops at LGA, ORD, OKC, DFW. If a customer books only the OKC-DFW leg, he or she will have to pay for the whole flight from LGA to DFW.
15. The relationship between managers and employees can be added but it is not necessary for the operations in this project. So it is simplified to just an attribute '**IsManager**' in the '**Employee**' entity.
16. The E-R diagram can be extended in order to know if a flight is on-time or not.

17. The tables defined below are not defined in any particular order, so simply attempting to copy-paste the CREATE TABLE statements into the SQL command editor will likely throw exceptions. A better approach would be to define the tables in a well-defined order, or to add primary-key and other constraints separately using ALTER TABLE statements.

Relational Model

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

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


CREATE TABLE 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,
PRIMARY KEY (AirlineID, FlightNo),
FOREIGN KEY (AirlineID) REFERENCES Airline(Id),
CHECK (NoOfSeats > 0),
CHECK (MinLengthOfStay >= 0),
CHECK (MaxLengthOfStay > MinLengthOfStay)
);

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

CREATE TABLE 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),
FOREIGN KEY (DepAirportID) REFERENCES Airport(Id),
FOREIGN KEY (ArrAirportID) REFERENCES Airport(Id),
CHECK (LegNo > 0)
);

CREATE TABLE 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),
CHECK (Fare > 0)
);

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

CREATE TABLE CustomerPreferences(
AccountNo           INTEGER NOT NULL,
Preference          VARCHAR(50) NOT NULL,
PRIMARY KEY (AccountNo, Preference),
FOREIGN KEY(AccountNo) REFERENCES Customer (AccountNo)
);

CREATE TABLE 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),
UNIQUE (Id),
CHECK (SSN > 0),
CHECK (HourlyRate > 0)
);

CREATE TABLE Passenger (
Id                  INTEGER,
AccountNo           INTEGER,
PRIMARY KEY (Id, AccountNo),
FOREIGN KEY (Id) REFERENCES Person(Id),
FOREIGN KEY (AccountNo) REFERENCES Customer(AccountNo),
);
```

```
CREATE TABLE 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),
FOREIGN KEY (AccountNo) REFERENCES Customer (AccountNo),
CHECK (ResrNo > 0),
CHECK (BookingFee >= 0),
CHECK (TotalFare > BookingFee)
);

CREATE TABLE 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),
FOREIGN KEY (AirlineID, FlightNo, LegNo) REFERENCES Leg(AirlineID, FlightNo, LegNo)
);

CREATE TABLE ReservationPassenger (
ResrNo              INTEGER,
Id                  INTEGER,
AccountNo           INTEGER,
SeatNo              CHAR(5) NOT NULL,
Class               VARCHAR(20) NOT NULL,
Meal                VARCHAR(50),
PRIMARY KEY (ResrNo, Id, AccountNo),
FOREIGN KEY (ResrNo) REFERENCES Reservation (ResrNo),
FOREIGN KEY (Id, AccountNo) REFERENCES Passenger (Id, AccountNo)
);

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