Gunnar Martin
CS340
Spring 2015

Final Project: Flight Booking Database

Outline

This database is a simple flight scheduling and booking database. The airline schedules flights that have a date, an origin, a destination and an aircraft, using the aircraft available in the fleet. Passengers search for flights to see if they would like to purchase a ticket on a flight. Each ticket has a passenger, a flight ID and can also have upgrades such as free movies, beverages, or food. Each passenger has a first and last name, along with a date of birth.

Tracking this type of data would be interesting to see where an airline would schedule flight and where passengers would choose to fly. It would also be interesting to watch over time as an airline would adjust its flight schedule to maximize revenue by aligning its supply of flights with the demands of the passengers.

Database Outline in Words

I found it easier to describe my database outline by walking through my entities and relationships in an outline format, instead of writing it in paragraph form.

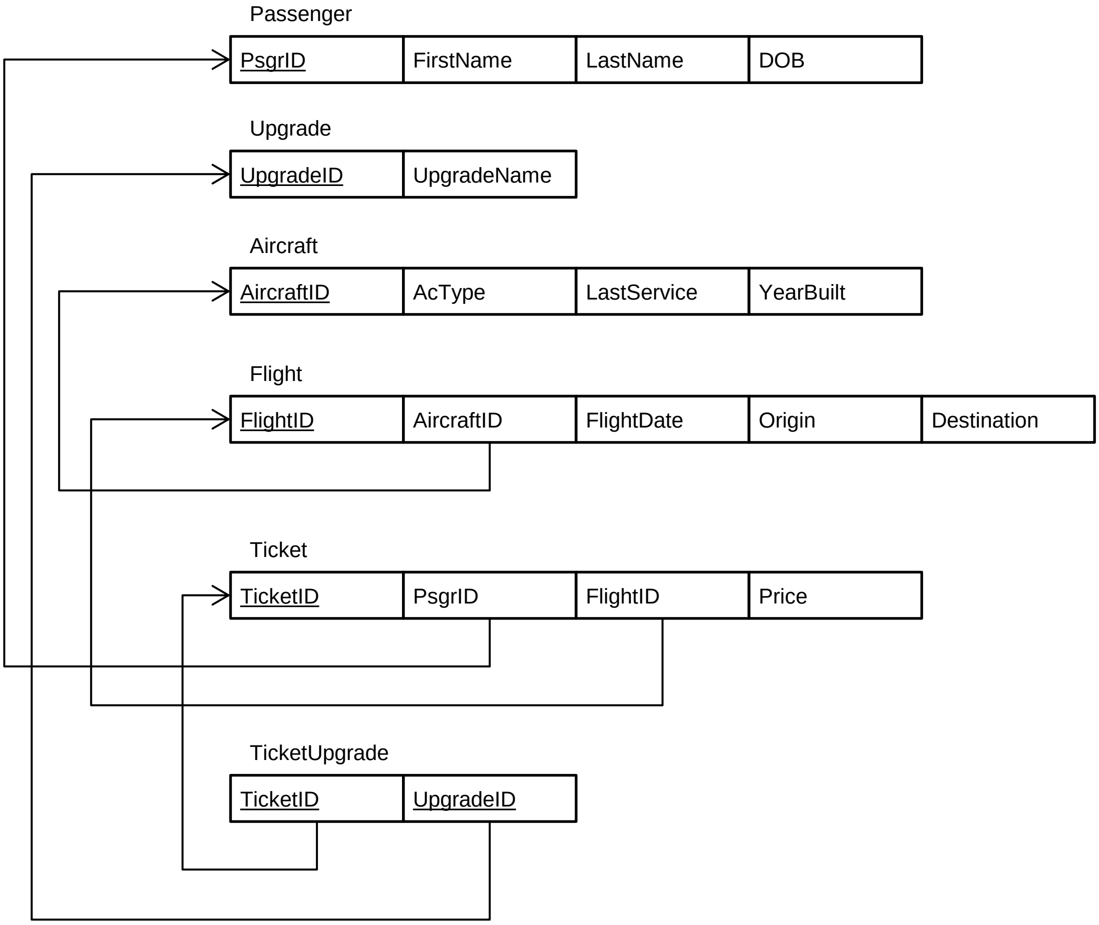
Entities

1. Passenger: The individuals booking flights.
   - Attributes: Passenger ID (primary), First Name, Last Name, Date of Birth
   - Constraints:
     - Each Passenger ID must be unique
2. Upgrade: Extra amenities that can be applicable to a passenger's ticket.
   - Attributes: Upgrade ID (primary), Upgrade Name
   - Constraints:
     - Each Upgrade ID must be unique
3. Aircraft: This is the inventory of the physical aircraft available to service the flights.
   - Attributes: Aircraft ID (primary), Aircraft Type, Last Date of Service, Year Built
   - Constraints:
     - Each Aircraft ID must be unique
4. Flight: These are the flights the airline has scheduled that passengers can buy tickets for.
   - Attributes: Flight ID (primary), Aircraft ID (foreign), Flight Date, Origin, Destination
   - Constraints:
     - Each Flight ID must be unique
     - Each flight can only have one Aircraft ID
5. Ticket: Passengers buy tickets for a flight for a price
   - Attributes: Ticket ID (primary), Passenger ID (foreign), Flight ID (foreign), Price
   - Constraints:
     - Each ticket ID must be unique
     - Each ticket can only have one Passenger ID
     - Each ticket can only have one Flight ID

Relationships

1. Passenger to ticket (one to many)
   - A passenger can have zero or more tickets
   - A ticket must belong to one and only one passenger
2. Aircraft to flight (one to many)
   - An aircraft can service zero or more flights
   - A flight must use one and only one aircraft
3. Flights to ticket (one to many)
   - A flight can have zero or more tickets for that flight
   - A ticket must apply to one and only one flight
4. Upgrade to tickets (many to many)
   - An upgrade can be attributable to zero or more tickets
   - A ticket can include zero or more upgrades

Database Schema



Passenger

PsgrID	FirstName	LastName	DOB

Upgrade

UpgradeID	UpgradeName

Aircraft

AircraftID	AcType	LastService	YearBuilt

Flight

FlightID	AircraftID	FlightDate	Origin	Destination

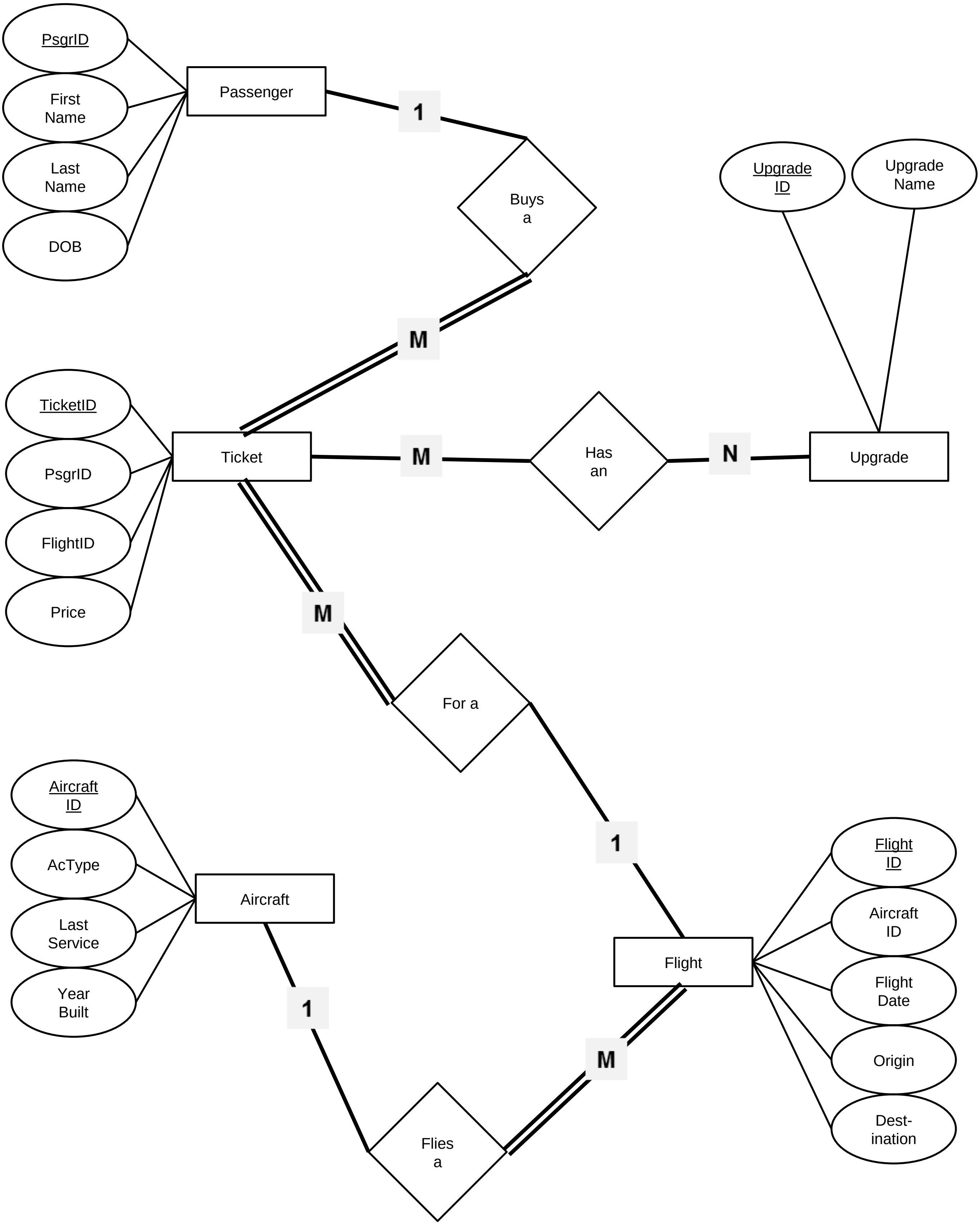
Ticket

TicketID	PsgrID	FlightID	Price

TicketUpgrade

TicketID	UpgradeID

ER Diagram


PsgrID
First Name
Last Name
DOB
Passenger
1
Buys a
M
Upgrade ID
Upgrade Name
TicketID
PsgrID
FlightID
Price
Ticket
M
Has an
N
Upgrade
M
For a
1
Aircraft ID
AcType
Last Service
Year Built
Aircraft
1
Flight ID
Aircraft ID
Flight Date
Origin
Dest-ination
Flight
M
Flies a

```sql
-- Gunnar Martin
-- CS340
-- Spring 2015
-- Final Project: Flight Booking DATABASE

-- Table Creation Queries

-- Drop these tables if they exist

DROP TABLE IF EXISTS TicketUpgrade;
DROP TABLE IF EXISTS Ticket;
DROP TABLE IF EXISTS Flight;
DROP TABLE IF EXISTS Passenger;
DROP TABLE IF EXISTS Upgrade;
DROP TABLE IF EXISTS Aircraft;

-- create passenger table
CREATE TABLE Passenger (
  PsgrID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  FirstName VARCHAR(255) NOT NULL,
  LastName VARCHAR(255) NOT NULL,
  DOB DATE NOT NULL,
  PRIMARY KEY  (PsgrID)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- create Upgrade table
CREATE TABLE Upgrade (
			UpgradeID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
			UpgradeName VARCHAR(255) NOT NULL,
			PRIMARY KEY (UpgradeID)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- create Aircraft table
CREATE TABLE Aircraft (
			AircraftID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
			AcType VARCHAR(255) NOT NULL,
			LastService DATE NOT NULL,
			YearBuilt SMALLINT UNSIGNED NOT NULL,
			PRIMARY KEY (AircraftID)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- create Flight table
CREATE TABLE Flight (
			FlightID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
			AircraftID SMALLINT UNSIGNED NOT NULL,
			FlightDate DATE NOT NULL,
			Origin VARCHAR(3) NOT NULL,
			Destination VARCHAR(3) NOT NULL,
			PRIMARY KEY (FlightID),
			CONSTRAINT fk_flight_aircraft FOREIGN KEY (AircraftID) REFERENCES Aircraft (AircraftID)
ON DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Gunnar Martin
-- CS340
-- Spring 2015
-- Final Project: Flight Booking DATABASE

-- Table Creation Queries cont.

-- create Ticket table
CREATE TABLE Ticket (
			TicketID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
			PsgrID SMALLINT UNSIGNED NOT NULL,
			FlightID SMALLINT UNSIGNED NOT NULL,
			Price SMALLINT UNSIGNED NOT NULL,
			PRIMARY KEY (TicketID),
			CONSTRAINT fk_ticket_psgr FOREIGN KEY (PsgrID) REFERENCES Passenger (PsgrID)
ON DELETE RESTRICT ON UPDATE CASCADE,
			CONSTRAINT fk_ticket_flight FOREIGN KEY (FlightID) REFERENCES Flight (FlightID) ON
DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- create TicketUpdgrade table
CREATE TABLE TicketUpgrade (
			TicketID SMALLINT UNSIGNED NOT NULL,
			UpgradeID SMALLINT UNSIGNED NOT NULL,
			PRIMARY KEY (TicketID, UpgradeID),
			CONSTRAINT fk_tu_ticket FOREIGN KEY (TicketID) REFERENCES Ticket (TicketID) ON
DELETE RESTRICT ON UPDATE CASCADE,
			CONSTRAINT fk_tu_upgrade FOREIGN KEY (UpgradeID) REFERENCES
Upgrade(UpgradeID) ON DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Gunnar Martin
-- CS340
-- Spring 2015
-- Final Project: Flight Booking DATABASE

-- General Use Queries

-- Passenger SELECT
SELECT
Passenger.PsgrID,
Passenger.FirstName,
Passenger.LastName,
Passenger.DOB
FROM Passenger;

-- Passenger INSERT
INSERT INTO Passenger(FirstName, LastName, DOB)
VALUES ([FirstName],[LastName],[DOB]);

-- Passenger UPDATE
UPDATE Passenger SET FirstName = [FirstName], LastName = [LastName]
WHERE PsgrID = [PsgrID];

-- Upgrade SELELCT
SELECT
Upgrade.UpgradeID,
Upgrade.UpgradeName
FROM Upgrade;

-- Upgrade INSERT
INSERT INTO Upgrade(UpgradeName) VALUES ([UpgradeName]);

-- Aircraft SELECT
SELECT
Aircraft.AircraftID,
Aircraft.AcType,
Aircraft.LastService,
Aircraft.YearBuilt
FROM Aircraft;

-- Aircraft UPDATE
UPDATE Aircraft SET LastService =  [LastService]
WHERE AircraftID = [AircraftID];

-- Flight SELECT
SELECT
Flight.FlightID,
Flight.AircraftID,
Aircraft.AcType,
Flight.FlightDate,
Flight.Origin,
Flight.Destination
FROM Flight
LEFT JOIN Aircraft
ON Flight.AircraftID = Aircraft.AircraftID
ORDER BY Flight.FlightDate ASC;
```

```
-- Flight INSERT
INSERT INTO Flight(AircraftID, FlightDate, Origin, Destination)
VALUES ([AircraftID], [FlightDate],[Origin], [Destination]);

-- Ticket SELECT
SELECT
Ticket.TicketID,
Ticket.PsgrID,
Passenger.LastName,
Passenger.FirstName,
Ticket.FlightID,
Flight.FlightDate,
Flight.Origin,
Flight.Destination,
Ticket.Price
FROM Ticket
LEFT JOIN Passenger
ON Ticket.PsgrID = Passenger.PsgrID
LEFT JOIN Flight
ON Ticket.FlightID = Flight.FlightID
ORDER BY Flight.FlightID ASC;

-- Ticket INSERT
INSERT INTO Ticket(PsgrID, FlightID, Price)
VALUES ([PsgrID],[FlightID],[Price]);

-- Ticket Upgrade SELECT
SELECT
TicketUpgrade.TicketID,
TicketUpgrade.UpgradeID,
Upgrade.UpgradeName,
Passenger.LastName,
Passenger.FirstName,
Ticket.FlightID,
Flight.FlightDate,
Flight.Origin,
Flight.Destination
FROM TicketUpgrade
LEFT JOIN Ticket
ON TicketUpgrade.TicketID = Ticket.TicketID
LEFT JOIN Upgrade
ON TicketUpgrade.UpgradeID = Upgrade.UpgradeID
LEFT JOIN Passenger
ON Ticket.PsgrID = Passenger.PsgrID
LEFT JOIN Flight
ON Ticket.FlightID = Flight.FlightID
ORDER BY TicketUpgrade.TicketID ASC;
```

```
-- Ticket Upgrade Filter
SELECT
TicketUpgrade.TicketID,
TicketUpgrade.UpgradeID,
Upgrade.UpgradeName,
Passenger.LastName,
Passenger.FirstName,
Ticket.FlightID,
Flight.FlightDate,
Flight.Origin,
Flight.Destination
FROM TicketUpgrade
LEFT JOIN Ticket
ON TicketUpgrade.TicketID = Ticket.TicketID
LEFT JOIN Upgrade
ON TicketUpgrade.UpgradeID = Upgrade.UpgradeID
LEFT JOIN Passenger
ON Ticket.PsgrID = Passenger.PsgrID
LEFT JOIN Flight
ON Ticket.FlightID = Flight.FlightID
WHERE Ticket.PsgrID = [PsgrID]
ORDER BY TicketUpgrade.TicketID ASC;

-- Ticket Upgrade INSERT
INSERT INTO TicketUpgrade(TicketID, UpgradeID) VALUES ([TicketID],[UpgradeID]);

-- Ticket Upgrade DELETE
DELETE FROM TicketUpgrade WHERE TicketID = [TicketID] AND UpgradeID = [UpgradeID];
```