IS 410 ER Diagram (Project)
Fall 2017
Team: Aleksander Babecki, Ray Rasolofonera, Fawaz Saheed-Baba, Diwakar Sharma, Raymond Tsang

Goal: Develop a Database system for an airline

Assumptions:

- A single passenger can purchase tickets for multiple passengers such as buying tickets for their spouse and children.
- We assume that all flights will be domestic and not far enough for multiple legs to each flight.
- Our database system is replacing a current system; due to this, there will be several customers whose records are already in the database during development.

Format:
__Entity__
- Attributes

__Employee (Supertype, disjoint, total specialization (lecture week 5, slide 18))__
- **Employee_ID**
- **Name**
- **Employee_type**

Employee
Employee_ID
Name
Employee_type

Description: Identifies employees of the airline.

__Flight Crew (subtype of Employee)__
- **Flight_num (Foreign Key)**

Flight Crew
Flight_num (Foreign Key)

Description: Identifies the current flight that pilots and attendants are tasked with.

__Flight Dispatcher (Subtype of Employee)__

- **Flight_num (Foreign Key)**
- **Alternate_destination**
- **Weather (Departure_Weather (Foreign Key), Arrival_Weather(Foreign Key))**
- **Arrival_airport (Foreign Key)**
- **Arrival_time (Foreign Key)**
- **Departure_airport (Foreign Key)**
- **Departure_time (Foreign Key)**

Flight Dispatcher
Flight_num (Foreign Key)
Alternate_destination
Weather (Departure_weather (Foreign Key), Arrival_weather (Foreign Key))
Arrival_airport (Foreign Key)
Arrival_time (Foreign Key)
Departure_airport (Foreign Key)
Departure_time (Foreign Key)

Description: Keeps track of Weather during the flight and will either delay departure if Departure_weather is dangerous or redirect Flight by changing Arrival_airport if Arrival_weather becomes dangerous.

__Passenger__

- **Passenger_ID**
- **Name**
- **Flight_num (Foreign Key)**
- **Ticket_num**
- **Seat_num**
- Address (Zipcode, Street, [City], [State])

Passenger
Passenger_ID
Name
Flight_num (Foreign Key)
Ticket_num
Seat_num
Address (Zipcode, Street, [City], [State])

Description: A single passenger can purchase tickets for multiple passengers such as buying tickets for their spouse and children. Each must have their own attributes. Unary relation. Passengers may include a Zipcode and street. If they are included, passenger city and state are derived.

__Flight__

- **Airplane_type**
- **Plane_ID**
- **Flight_num (PK)**

Flight
Flight_num
Plane_ID
Airplane_type

Description: Each flight has a listed airplane type, flight ID number, and flight number.

__Departure__

- **Flight_num (Foreign Key)**
- **Departure_airport**
- **Departure_time**
- **Departure_weather**

Departure
Flight_num (Foreign Key)
Departure_airport
Departure_time
Departure_weather

Description: Departure weather, date, and location.

__Arrival__

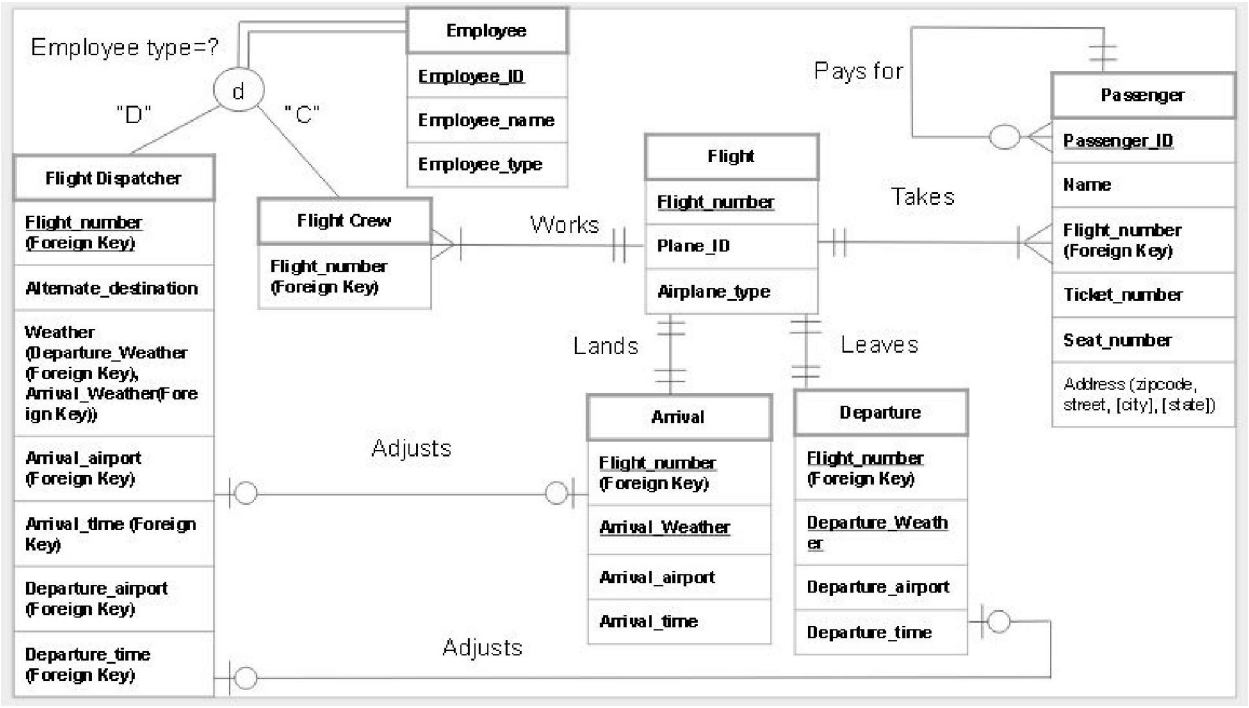
- **Flight_num (Foreign Key)**
- **Arrival_airport**
- **Arrival_time**
- **Arrival_weather**

Arrival
Flight_num (Foreign Key)
Arrival_airport
Arrival_time
Arrival_weather

Description: Arrival weather, date, and location.

Relations:

- Employee is a supertype of Flight Crew and of Flight Dispatcher. The relation is disjointed and total specialization. The subtype is indicated by assigning “D” or “C” to Employee_type.
- Flight Dispatcher uses the following foreign keys: Flight_number as a primary key to identify the particular Flight they are assigned to; Arrival_airport, Arrival_time, Departure_airport, and Departure_time are monitored by the Flight Dispatcher and can be modified based off of changes to Weather. The Weather attribute is a composite of Foreign Keys Departure_weather and Arrival_weather.
- Flight is an entity that has an attribute, Flight_number, which is used by 5 entities as a Foreign Key. Those entities are Flight dispatcher, Flight crew, Passenger, arrival and departure.
- An instance of Flight takes 1 or multiple instances of Flight Crew. An instance of Flight Crew takes 1 and only 1 Flight.
- An instance of Flight takes 1 or multiple instance of passengers. An instance of Passengers takes 1 and only 1 Flight.
- An instance of Flight has 1 and only 1 Arrival.
- An instance of Flight has 1 and only 1 Departure.
- An instance of passenger has an unary relation where passenger pays for 1 or multiple passengers, and passenger is paid by 1 and only 1 passenger.
- An instance of employee has a completeness constraint and disjoint constraint towards the instance of Flight dispatcher and the instance of Flight Crew.
- An instance of Flight Dispatcher adjusts to 0 or 1 Arrival, and an instance of arrival adjusts to 0 or 1 instance of Flight Dispatcher.
- An instance of Flight Dispatcher adjusts to 0 or 1 Departure, and an instance of Departure adjusts to 0 or 1 Flight Dispatcher.


Employee type=?
d
"D"
"C"
Employee
Employee_ID
Employee_name
Employee_type
Flight Dispatcher
Flight_number (Foreign Key)
Alternate_destination
Weather (Departure_Weather (Foreign Key), Arrival_Weather(Foreign Key))
Arrival_airport (Foreign Key)
Arrival_time (Foreign Key)
Departure_airport (Foreign Key)
Departure_time (Foreign Key)
Flight Crew
Flight_number (Foreign Key)
Works
Flight
Flight_number
Plane_ID
Airplane_type
Pays for
Takes
Passenger
Passenger_ID
Name
Flight_number (Foreign Key)
Ticket_number
Seat_number
Address (zipcode, street, [city], [state])
Lands
Leaves
Arrival
Flight_number (Foreign Key)
Arrival_Weather
Arrival_airport
Arrival_time
Departure
Flight_number (Foreign Key)
Departure_Weather
Departure_airport
Departure_time
Adjusts
Adjusts