Documentation of Schema

Note that the designers of this DB schema didn’t produce any documentation, so there are multiple ways of interpreting the meanings of some attributes. This is one interpretation:

1. AIRPORT (airport\_code, name, city, state) Each row of the AIRPORT table will hold :

• airport\_code : The code of an airport (Example: EWR, JFK, SFO, LAX)

• name : The name of the airport

• city : The city in which the airport is located.

• state : The state in which the airport is located.

2. FLIGHT (number, airline, weekdays): Note that FLIGHT is an abstraction, an “entity set” if you like, rather than a particular airplane taking off, flying, and landing.)

Each row of the FLIGHT table will hold :

• Number : The Number of a flight

• Airline : The name of the airline (company) that operates the flight

• Weekdays : The days of the week on which the flight operates

3. FLIGHT\_LEG (flight\_number, leg\_number, departure\_airport\_code, scheduled\_departure\_time, arrival\_airport\_code, scheduled\_arrival\_time): Note that a FLIGHT\_LEG is a “leg” or “segment” of a FLIGHT. Note further that FLIGHT\_LEG is an abstraction, rather than a particular airplane taking off, flying, and landing.)

Each row of the FLIGHT\_LEG table will hold :

• flight\_number: The Number of a flight

• leg\_number: The number of a leg (segment) of the flight

• departure\_airport\_code: The airport\_code of the airport from which the leg of the flight departs

• scheduled\_departure\_time : The scheduled departure time of the leg of the flight

• arrival \_airport\_code: The airport\_code of the airport at which the leg of the flight arrives

• scheduled\_ arrival \_time : The scheduled arrival time of the leg of the flight

4. LEG\_INSTANCE (flight\_number, leg\_number, date, number\_of\_available\_seats, airplane\_id, Departure\_airport\_code, departure\_time, arrival\_airport\_code, arrival\_time) Note that a LEG\_INSTANCE is/represents an actual airplane

taking off, flying, and landing.

Each row of the LEG\_INSTANCE table will hold :

• flight\_number: The number of a flight

• leg\_number: The leg\_number of a leg of the flight

• date : A date on which the instance of the leg of the flight will operate

• number\_of\_available\_seats : The number of seats available on the leg of the flight on the given date

• airplane\_id : The identifier of the particular airplane (aircraft) that will will be used on the leg of the flight on the given date

• departure\_airport\_code: The airport\_code of the airport from which the leg of the flight will depart on the given date. (It seems unnecessary to repeat this airport\_code here when it’s already in FLIGHT\_LEG!!!)

• departure\_time : The time at which the leg of the flight will depart on the given date. (possibly changed to actual departure time after the LEG\_INSTANCE has flown!!!)

• arrival \_airport\_code: The airport\_code of the airport at which the leg of the flight will arrive on the given date. (It seems peculiar to repeat this airport\_code here when it’s already in FLIGHT\_LEG!!!)

• arrival \_time : The time at which the leg of the flight will arrive on the given date. (possibly changed to actual arrival time after the LEG\_INSTANCE has flown!!!)

5. FARES (flight\_number, fare\_code, amount, restrictions) Note that it is peculiar that there is no fare information for FLIGHT\_LEG since passengers can reserve seats on, make seat reservations for individual leg instances

Each row of the FARES table will hold :

• flight\_number: The number of a flight

• fare\_code: The code for a type of fare that can be booked on the flight

• amount : The cost of that type of fare on the flight

• restrictions : The restrictions imposed on the type of fare on the flight

6. AIRPLANE\_TYPE (type\_name, max\_seats, company) Each row of the AIRPLANE\_TYPE table will hold :

• Type\_name: The name of a type of airplane

• Max\_seats: The maximum number of seats the type of airplane can be configured to hold

• company : The company that manufactures airplanes of the type.

7. CAN\_LAND (airplane\_type\_name, airport\_code) Each row of the CAN\_LAND table will hold :

• Airplane\_Type: The type\_name of a type of airplane

• Airport\_code: The airport\_code of an airport at which the type of airplane can land.

8. AIRPLANE (airplane\_id, total\_number\_of\_seats, airplane\_type) Each row of the AIRPLANE table will hold :

• Airplane\_id: The identifier of an airplane

• Total\_number\_of\_seats : The total number of seats on that airplane

• Airplane\_type : The type\_name of the type of the airplane

9. SEAT\_RESERVATION (flight\_number, leg\_number, date, seat\_number, customer\_name, customer\_phone)

Each row of the SEAT\_RESERVATION table will hold :

• flight\_number: The number of a flight

• leg\_number: The number of a leg of the flight

• date : A date on which an instance of the flight will be operating

• seat\_number : The number of a seat on the airplane that will be used on the instance of the flight on that date

• customer\_name : The name of the customer who has reserved the seat on the airplane that will be used on the instance of the flight on that date

• Customer\_phone : The phone number of the customer who has reserved the seat on the airplane that will be used on the instance of the flight on that date

Database Instance

AIRPORT

|  |  |  |  |
| --- | --- | --- | --- |
| AIRPORT\_CODE | NAME | CITY | STATE |
| LAX | Los Angele  S | Los Angeles | California |
| JFK | Kennedy | New York | New York |
| EWR | Newark | Newark | New Jersey |
| LGA | La Guardia | New York | New York |
| PHX | Phoenix | Phoenix | Arizona |
| HNL | Honolulu | Honolulu | Hawaii |

FLIGHT

|  |  |  |
| --- | --- | --- |
| NUMBER | AIRLINE | WEEKDAYS |
| 132 | American Airlines | M,T,W,Th,F |
| 11 | United Airlines | M,W,F |
| 298 | Continental Airlines | M,W,F,S |
| 67 | Aloha Airlines | M,F,S |
| 897 | Northwest Airlines | M,T,W,Th,F,S,Su |
| 42 | Delta Airlines | M,T,W,Th,F,S |

FLIGHT\_LEG

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FLIGHT\_  NUMBER | LEG\_  NUMBER | DEPARTURE\_  AIRPORT\_ CODE | SCHEDULED\_  DEPARTURE\_ TIME | ARRIVAL\_  AIRPORT  \_CODE | SCHEDULED\_  ARRIVAL  \_TIME |
| 132 | 1 | JFK | 08:00 | PHX | 11:00 |
| 132 | 2 | PHX | 12:00 | LAX | 13:30 |
| 132 | 3 | LAX | 14:30 | HNL | 15:30 |
| 67 | 1 | HNL | 11:00 | LAX | 17:00 |
| 67 | 2 | LAX | 17:30 | PHX | 19:50 |

LEG\_INSTANCE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FLIGHT\_ NUMBER | LEG\_ NUMBER | DATE | NUMBER\_ OF\_  AVAIL- ABLE\_  SEATS | AIR- PLANE  \_ ID | DEPARTURE\_ AIRPORT\_  CODE | DEPART- URE\_  TIME | ARRIVAL  \_ AIRPORT  \_  CODE | ARRIVAL  \_TIME |
| 132 | 1 | 9/15/03 | 112 | 32453 | JFK | 08:10 | PHX | 11:04 |
| 132 | 2 | 9/15/03 | 92 | 32453 | PHX | 12:05 | LAX | 13:25 |
| 132 | 3 | 9/15/03 | 86 | 32453 | LAX | 14:30 | HNL | 16:00 |

FARES

|  |  |  |  |
| --- | --- | --- | --- |
| FLIGHT\_NUMBER | FARE\_CODE | AMOUNT | RESTRICTIONS |
| 132 | ECON | 450.00 | Non-refundable |
| 132 | BUSINESS | 900.00 | none |
| 132 | FIRST | 2100.00 | none |

AIRPLANE\_TYPE

|  |  |  |
| --- | --- | --- |
| TYPE\_NAME | MAX\_SEATS | COMPANY |
| 737 | 200 | Boeing |
| 747 | 245 | Boeing |
| 757 | 315 | Boeing |
| 767 | 350 | Boeing |

CAN\_LAND

|  |  |
| --- | --- |
| AIRPLANE\_TYPE\_NAME | AIRPORT\_CODE |
| 737 | LAX |
| 737 | JFK |
| 737 | EWR |
| 737 | LGA |
| 737 | PHX |
| 737 | HNL |
| 747 | LAX |

AIRPLANE

|  |  |  |
| --- | --- | --- |
| AIRPLANE\_ID | TOTAL\_NUMBER\_OF\_SEATS | AIRPLANE\_TYPE |
| 32453 | 180 | 737 |
| 76543 | 205 | 747 |

SEAT\_RESERVATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FLIGHT\_  NUMBER | LEG\_  NUMBER | DATE | SEAT\_  NUMBER | CUSTOMER\_  NAME | CUSTOMER\_  PHONE |
| 132 | 1 | 10/20/03 | 23A | John Smith | 555-555-5555 |
| 132 | 2 | 10/20/03 | 23A | John Smith | 555-555-5555 |
| 132 | 3 | 10/20/03 | 23A | John Smith | 555-555-5555 |
| 132 | 1 | 10/20/03 | 16A | Jane Doe | 666-666-6666 |
| 132 | 1 | 10/20/03 | 11A | Jane Roe | 777-777-7777 |
| 132 | 1 | 10/20/03 | 19E | Jane Low | 888-888-8888 |

Foreign Key References

The following is a copy of the original schema with sources of necessary foreign key references marked in red. Please check this out and post questions if you have any.

The table after this one indicates the targets for each fkr and introduces a notation for describing fkrs.

**Airline Relational Database Schema**

airport(airport\_code, name, city, state)

flight(number, airline, weekdays)

flight\_leg(flight\_number, leg\_number, departure\_airport\_code, scheduled\_departure\_time, arrival\_airport\_code, scheduled\_arrival\_time)

leg\_instance(flight\_number, leg\_number, date, number\_of\_available\_seats, airplane\_id, departure\_airport\_code, departure\_time, arrival\_airport\_code, arrival\_time)

fares(flight\_number, fare\_code, amount, restrictions)

airplane\_type(type\_name, max\_seats, company)

can\_land(airplane\_type\_name, airport\_code)

airplane(airplane\_id, total\_number\_of\_seats, airplane\_type)

seat\_reservation(flight\_number, leg\_number, date, seat\_number, customer\_name, customer\_phone)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Airline Relational Database Schema**

airport(airport\_code, name, city, state)

doesn’t need any outgoing fkrs

flight(number, airline, weekdays)

doesn’t need any outgoing fkrs

flight\_leg(flight\_number, leg\_number, departure\_airport\_code, scheduled\_departure\_time, arrival\_airport\_code, scheduled\_arrival\_time)

flight\_leg.flight number needs an fkr to flight.number,

which we can represent as

flight\_leg.flight 🡪 flight.number

AND

flight\_leg.departure\_airport\_code needs an fkr to airport.airport\_code

which we can represent as

flight\_leg.departure\_airport\_code 🡪 airport.airport\_code

AND

flight\_leg.arrival\_airport\_code needs an fkr to airport.airport\_code

which we can represent as

flight\_leg.arrival\_airport\_code 🡪 airport.airport\_code

leg\_instance(flight\_number, leg\_number, date, number\_of\_available\_seats, airplane\_id, departure\_airport\_code, departure\_time, arrival\_airport\_code, arrival\_time)

(leg\_instance,flight\_number, leg\_instance.leg\_number) needs an fkr to

(flight\_leg.flight\_number, flight\_leg.leg\_number)

which we can represent as

(leg\_instance,flight\_number, leg\_instance.leg\_number) 🡪

(flight\_leg.flight\_number, flight\_leg.leg\_number)

BUT

there’s no need for an fkr from leg\_instance.departure\_airport\_code

OR

from leg\_instance.arrival\_airport\_code

BECAUSE

the constraints that they impose are already imposed, transitively, by the two-attribute fkr from

leg\_instance to flight\_leg

AND

the two fkrs to airport\_code from leg\_instance to flight\_leg

fares(flight\_number, fare\_code, amount, restrictions)

fares.flight\_number needs an fkr to flight.number

which we can represent as

fares.flight\_number 🡪 flight.number

airplane\_type(type\_name, max\_seats, company)

doesn’t need any outgoing fkrs

can\_land(airplane\_type\_name, airport\_code)

can\_land.airplane\_type\_name needs an fkr to airplane\_type.type\_name

AND

can\_land.airport\_code needs an fkr to airport.airport\_code

These can be represented by

can\_land.airplane\_type\_name 🡪 airplane\_type.type\_name

AND

can\_land.airport\_code 🡪 airport.airport\_code

airplane(airplane\_id, total\_number\_of\_seats, airplane\_type)

airplane.airplane\_type needs an fkr to airplane\_type.type\_name

which we can represent as

airplane.airplane\_type 🡪 airplane\_type.type\_name

seat\_reservation(flight\_number, leg\_number, date, seat\_number, customer\_name, customer\_phone)

(seat\_reservation.flight\_number, seat\_reservation.leg\_number, seat\_reservation.date)

needs an fkr to

(leg\_instance.flight\_number, leg\_instance.leg\_number, leg\_instance.date)

which can be represented as

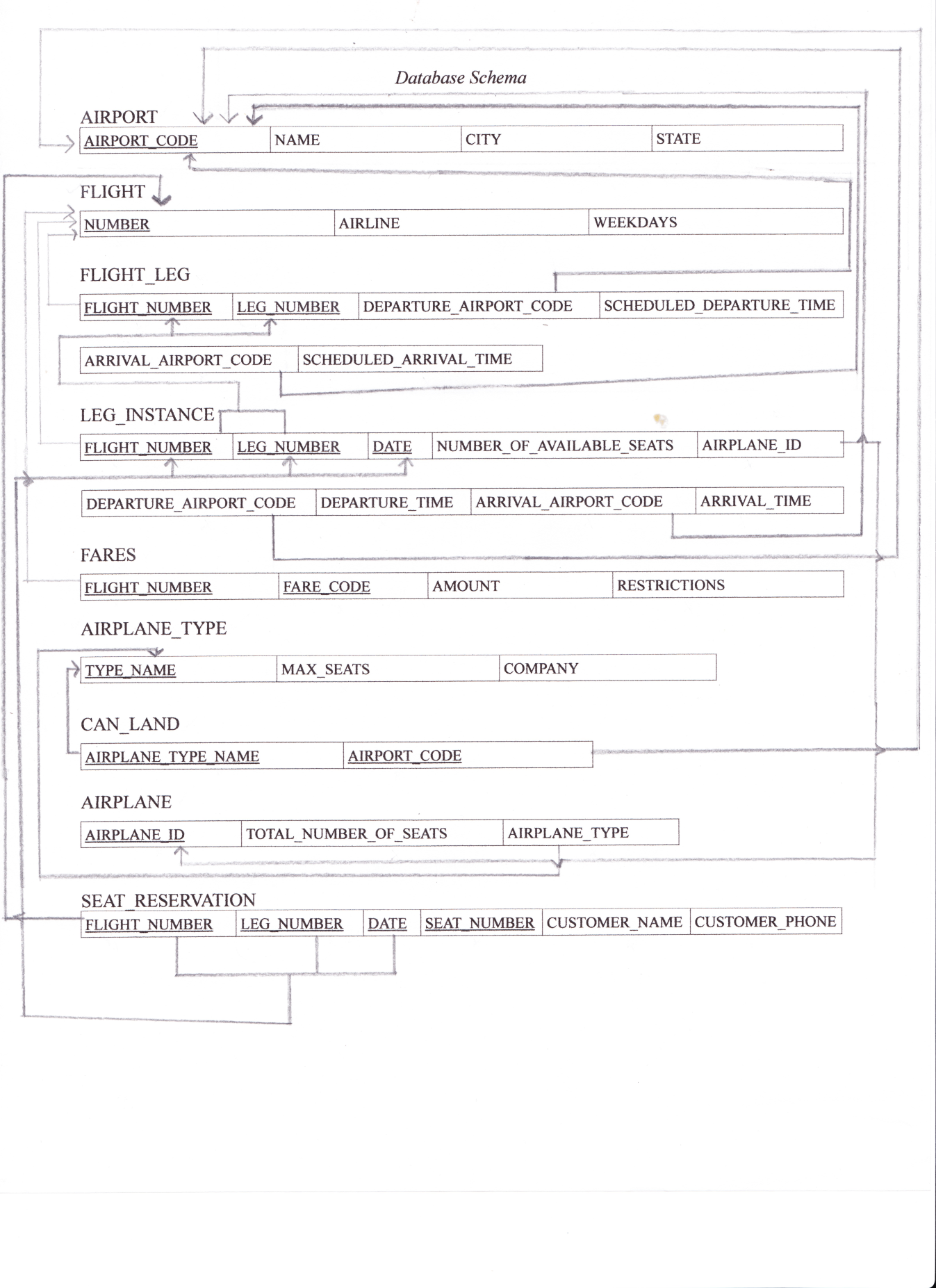
(seat\_reservation.flight\_number,seat\_reservation.leg\_number,seat\_reservation.date)🡪 \(leg\_instance.flight\_number, leg\_instance.leg\_number, leg\_instance.date)

BUT there’s no need for an fkr from seat\_reservation.flight\_number or from seat\_reservation.leg\_instance, or

from seat\_reservation.date because the fkr listed above already imposes these constraints, transitively, through

the fkrs from the two fkrs from attributes of leg\_instance.

Here’s an fkr diagram submitted by a class member

**