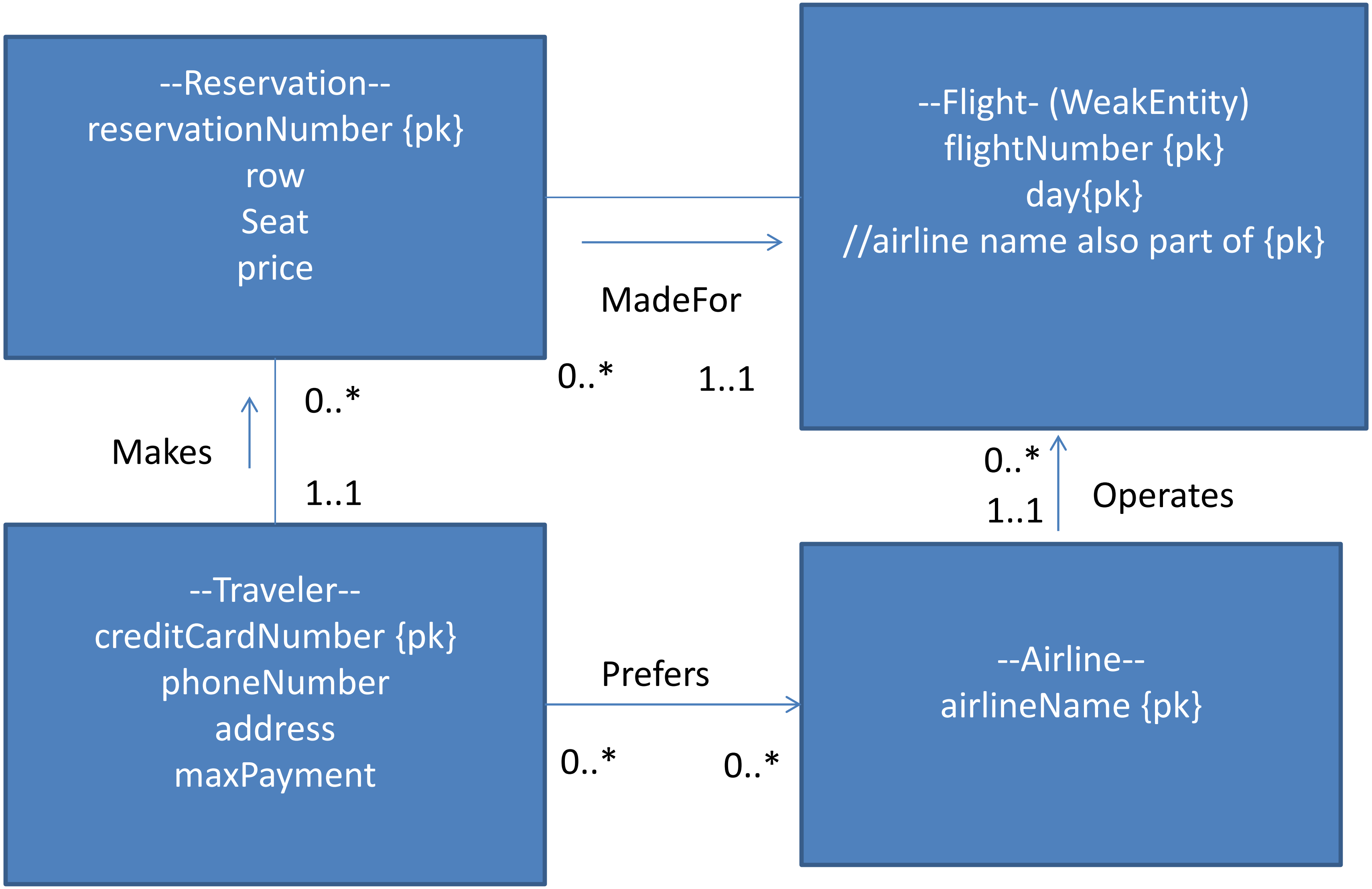
Flight Reservation System: E/R



--Reservation--
reservationNumber {pk}
row
Seat
price

--Flight- (WeakEntity)
flightNumber {pk}
day{pk}
//airline name also part of {pk}

--Traveler--
creditCardNumber {pk}
phoneNumber
address
maxPayment

--Airline--
airlineName {pk}

MadeFor
0..*
1..1

Makes
0..*
1..1

Operates
0..*
1..1

Prefers
0..*
0..*

Flight Reservation System: Relational

Reservation

reservation Number	row	seat	*traveler*	price	*flightNumber*	*day*	*airlineName*

Traveler

creditCardNumber	phoneNumber	address	maxPayment

Prefers

creditCardNumber	*airlineName*

Airline

airlineName

Flight

flightNumber	day	*airlineName*

Flight Reservation System: Query

- Assume “name” is an attribute of Traveler

(it was not included in the original E/R or description – a typo on my part)

- RA: $\Pi_{name,airline,flightNumber, row, seat}$ $(\sigma_{price>=1000}$(Reservation) NATURAL JOIN ($\Pi_{name,creditCardNumber}$(Traveler)))

Note this projects traveler down to just name and credit card number (the only fields we need – the name to print out and the creditCardNumber to use to join against reservation. This traveler information is natural joined with all reservations that cost >= $1000.00. From this resulting table, we can extract the name, airline, flightNumber, row, and seat.

- SQL: SELECT name, airline, flightNumber, row, seat FROM (SELECT * FROM Reservation WHERE price >= 1000.00) AS HP NATURAL JOIN (SELECT name,creditCardNumber FROM Traveler) AS TR;

This uses two subqueres to do the selection of reservations costing over 1000 and to get the name,creditCardNumber fields from all travelers. The natural join then does the joining required of the two tables (expensive reservations, travelers) to generate a third table. All names, airliens, flightNumbers, rows and seats are then pulled out of this table.