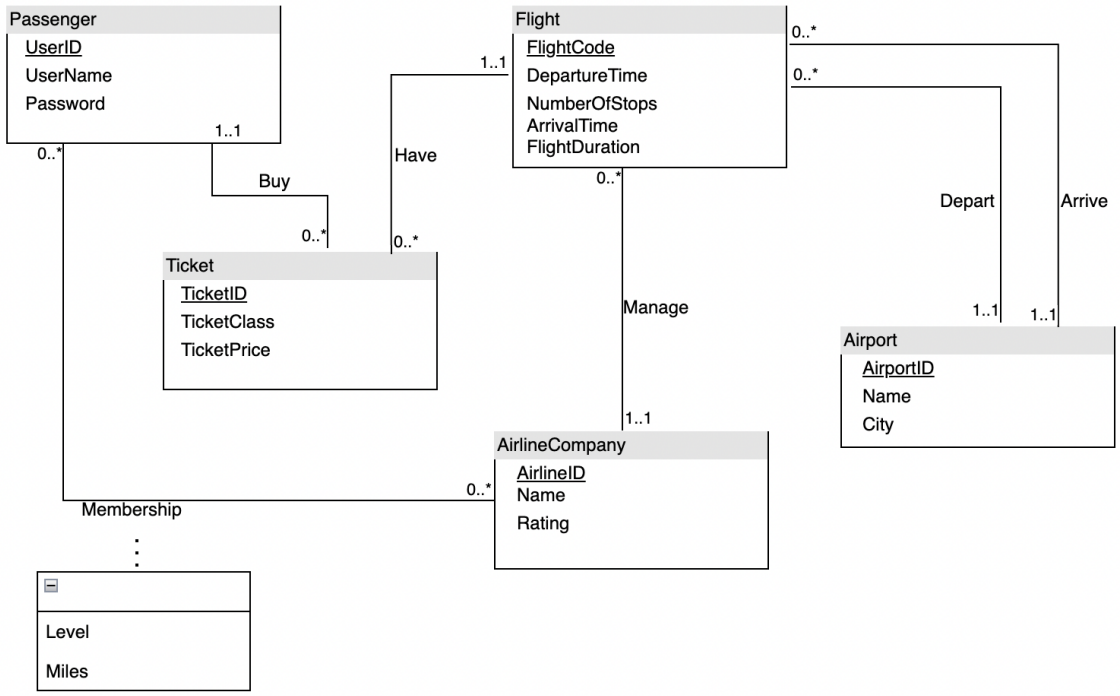
UML Diagram:



Convert your conceptual database design (ER/UML) to the logical design (relational schema):

AirlineCompany(AirlineID: INT [PK],
Name: VARCHAR(50),
Rating: DOUBLE)

Airport(AirPortID: INT [PK],
Name: VARCHAR(50),
City: VARCHAR(50))

Passenger(UserID: INT [PK],
UserName: VARCHAR(50),
Password: VARCHAR(50))

Flight(FlightCode: VARCHAR(50) [PK],
DepartureTime: DateTime,
NumberOfStops: INT,
ArrivalTime: DateTime,
AirlineID: INT [FK to AirlineCompany.AirlineID],

DepartingAirportID: INT [FK to Airport.AirPortID],
ArrivingAirportID: INT [FK to Airport.AirPortID])

Ticket(TicketID: INT [PK],
TicketClass: VARCHAR(50),
TicketPrice: REAL,
UserID: INT [FK to Passenger.UserID],
FlightCode: VARCHAR(50) [FK to Flight.FlightCode])

Membership(UserID: INT [PK, FK to Passenger.UserID],
FlightCode: VARCHAR(50) [PK, FK to Flight.FlightCode],
Level: VARCHAR(50)
Miles: REAL)

Assumptions of the ER/UML diagram:
Flight: Each flight is uniquely identified by “FlightCode”. This entity also contains 4 more attributes: “DepartureTime”, “NumberOfStops”, “ArrivalTime”, “FlightDuration”.

Ticket: Each ticket is uniquely identified by “TicketID”. This entity also contains 2 more attributes: “TicketClass”, and “TicketPrice”.

AirlineCompany: Each Airline Company is uniquely identified by “AirlineID”. This entity also contains 2 more attributes: “Name”, and “Rating”.

Airport: Each news is uniquely identified by its ID named “AirportID”. This entity also contains 2 more attributes: “Name”, and “City”.

Passenger: Each passenger is uniquely identified by “UserID”. This entity also contains 2 more attributes: “UserName”, and “Password”.

A description of each relationship and its cardinality:

Depart (Flight to Airport, many to one): A flight can only depart from one airport, an airport can host many flights

Arrive (Flight to Airport, many to one): A flight can only arrive at one airport, an airport can host many flights

Manage (Flight to AirlineCompany ,many to one): An airline company have many flights, a flight only belong to one company

Have (Flight to Ticket, one to many): There are many tickets for a flight, with a ticket, you can only take take one flight.

Buy (Ticket to Passenger, many to one): Passengers can buy many tickets, each ticket can only be bought by one passenger

Membership (Passenger to AirlineCompany, many to many): A passenger can have mambership of many airline companies, an airline company can have many members