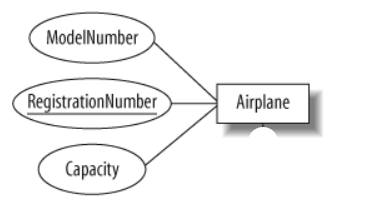
* We’ve treated the capacity of the airplane as an attribute of an individual airplane. If, instead, we assumed that the capacity is determined by the model number, we would have created a new AirplaneModel entity with the attributes ModelNumber and Capacity. The Airplane entity would then not have a Capacity attribute.

Yêu cầu chưa được thực hiện này yêu cầu chúng ta mở thêm một thực thể mới mang tên **AirplaneModel** bao gồm các thuộc tính như **ModelNumber** và **Capacity** và như chúng ta đã biết 2 thuộc tính trên ban đầu là thuộc về thực thể **Airplane**.

=> Vậy ta có thể suy ra yêu cầu của đề bài đang muốn chúng ta mở rộng cho lớp thực thể **Airplane** và 2 thuộc tính đã được tách ra đó sẽ được chuyển sang cho thực thể mới là **AirplaneModel** trong đó thuộc tính **ModelNumber** sẽ đóng vai trò là khoá chính giúp phân biệt các đối tượng dữ liệu với nhau. Thực thể **AirplaneModel** đóng vai trò là thực thể mạnh và thực thể **Airplane** sẽ đóng vai trò là thực thể yếu phụ thuộc vào thực thể mạnh **AirplaneModel** thông qua mối quan hệ **1 – nhiều** giữa 2 thực thể.



Thực thể Airplane ban đầu

Diagram

Description automatically generated

Thực thể Airplane sau khi được mở rộng

* We’ve mapped a different flight number to each flight between two destinations. Airlines typically use a flight number to identify a given flight path and schedule, and they specify the date of the flight independently of the flight number. For example, there is one IR655 flight on April 1, another on April 2, and so on. Different airplanes can operate on the same flight number over time; our model would need to be extended to support this.

Yêu cầu chưa được thực hiện này yêu cầu chúng ta mở rộng và nâng cấp chức năng cho mô hình thực thể ban đầu, mô hình thực thể ban đầu bao gồm:

Diagram

Description automatically generated

Thực thể Flight ban đầu

Ta nhận thấy là thực thể này mang chức năng lưu trữ như một số hiệu bay của từng chuyến bay, tức là nó bao gồm các thông tin như: Thời gian đi, thời gian đến, ngày đi, ngày đến, đi từ đâu, đến đâu và mã số hiệu chuyến bay.

Nhưng mô hình trên cũng có vấn đề như việc lưu trữ các thông tin dữ liệu chưa được tối ưu hoá và hiệu suất cũng không được ổn định, khi ta biểu diễn thực thế đó lên CSDL nó sẽ là 1 bảng có rất nhiều cột và mỗi cột đều có thể lặp đi lặp lại dữ liệu nhiều lần, điều đó sẽ khiến cho dung lượng lưu trữ trong CSDL trở nên rất nặng và việc truy vấn dữ liệu sẽ rất khó khăn khi có nhiều bản ghi (record) trong bảng. Do đó ta cần chuẩn hoá dữ liệu cho thực thể này.

Ta phân tách các thuộc tính cùa thực thể **Flight** ban đầu như sau:

Diagram

Description automatically generated

Thực thể Flight được mở rộng

Đối với 2 thuộc tính **From** và **To** ở thực thể **Flight** ban đầu, ta có thể tách chúng thành một thực thể mới mang tên **Location** và thêm một thuộc tính khoá chính **LocationID** để giúp phân biệt giữa các đối tượng dữ liệu. Và lý do ta chọn tách 2 thuộc tính trên thành một thực thể mới là vì trong thực tế địa điểm bay từ chỗ này đến chố khác tại một sân bay là cố định (Ví dụ sân bay Tân Sân Nhất có chuyến bay từ TPHCM – Hà Nội hoặc TPHCM – Đà Nẵng).

Và khi này ta sẽ có bao gồm 2 thực thể chính: **Flight** (Chuyến bay), **Location** (Địa điểm) và giữa 2 thực thể đó ta tạo mối quan hệ **nhiều – nhiều** mang tên **Schedule** (Lịch trình). Lý do chúng có quan hệ n – n là bởi vì trong thực tế số hiệu chuyến bay có thể tái sử dụng lại nhiều lần và vì thế một số hiệu chuyến bay có thể có nhiều địa điểm sân bay và một địa điểm sân bay có thể có nhiều số hiệu chuyến bay đi đến.

Thực thể mới **Schedule** này biễu diễn thông tin của một lịch trình bay bao gồm như: Số hiệu chuyến bay (FlightNumber), Địa điểm bắt đầu và kết thúc (From, To) và Thời gian đi, Thời gian đến (DepartureTime, ArrivalTime) và Ngày đi, Ngày đến (DepartureDate, ArrivalDate)

* The system also assumes that each leg of a multihop flight has a different FlightNumber. This means that a flight from Dubai to Christchurch via Singapore and Melbourne would need a different FlightNumber for the Dubai-Singapore, Singapore-Melbourne, and Melbourne-Christchurch legs.
* Our database also has limited ability to describe airports. In practice, each airport has a name, such as “Melbourne Regional Airport,” Mehrabad,” or “Tullamarine.”. The name can be used to differentiate between airports, but most passengers will just use the name of the town or city. This can lead to confusion, when, for example, a passenger could book a flight to Melbourne, Florida, USA, instead of Melbourne, Victoria, Australia. To avoid such problems, the International Air Transport Association (IATA) assigns a unique airport code to each airport; the airport code for Melbourne, Florida, USA is MLB, while the code for Melbourne, Victoria, Australia is MEL. If we were to model the airport as a separate entity, we could use the IATA-assigned airport code as the primary key. Incidentally, there’s an alternative set of airport codes assigned by the International Civil Aviation Organization (ICAO); under this code, Melbourne, Florida is KMLB, and Melbourne, Australia is YMML.