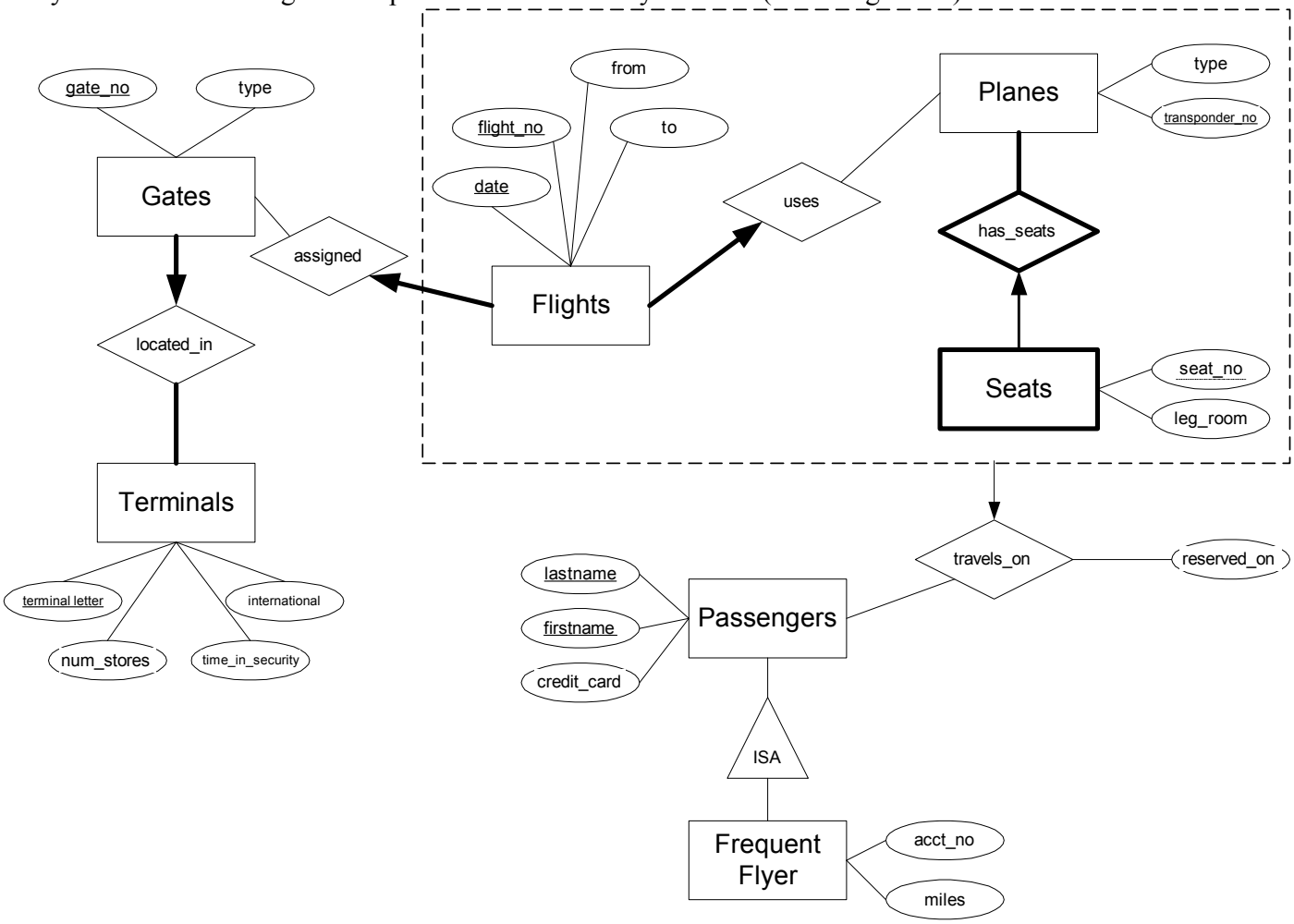
Sample Solution to Exercise

DISCLAIMER 1: This is only ONE solution, there are many. The solution provided was designed to show as many features of ER diagrams as possible and there may be other (including better) models.



Notes about sample solution:

- This model stores redundant information in the flights entity. A new instance is created for each flight on each day, however generally flight information generally does not change. Some solutions include:
  - using a separate entity for flights and flight plans
  - expand the uses relationship to include date (this introduces some other issues)
  - create a date entity and a relationship between flights and dates. These two entities and their relationship can then be aggregated
- Passengers could reserve more than 1 seat per flight, and there can be passengers (including frequent flyers) who have never flown before! It is also assumed no passenger has the same first and last name.
- Terminals have to have at least one gates, and each must gate belong to exactly one terminal.
- Unlike real airlines, there is no way to overbook planes since each combination of flight/plane/seat can only have up to one passenger (due to the aggregation and arrow to travels_on).
- Notice that flights is involved in both an aggregation (with flights, uses, planes, has_seates, and seats) and has a direct relationship with gates (which does not involve the aggregation).

DISCLAIMER 2: There are other restrictions and anomalies because of the modeling decisions in the solution… feel free to find them and point them out.

We create 3 tables:

Flights(flight_no, date, from, to, plane_id) where plane_id is a foreign key to planes and is not null
Planes(plane_id, type)
Seat(seat_no, plane_id, legroom) where plane_id is FK to planes, not null and when delete cascade.

```
CREATE TABLE Flights (
     date            DATE,
     flight_no       CHAR(5),
     from            CHAR(20),
     to              CHAR(20),
     plane_id        INTEGER NOT NULL,
     PRIMARY KEY (date, flight_no),
     FOREIGN KEY (plane_id) REFERENCES
          Planes (transponder_no) ON DELETE NO ACTION
);
```

plane_id may not be NULL because each flight must be assigned one plane (participation constraint). A Flight instance may be assigned to only one Plane instance (one to many relationship), we can incorporate the Uses relation in the Flights table itself. The "`ON DELETE NO ACTION`" clause prevents a Plane instance from being deleted before the Flights that use the Plane get assigned a different one.

```
CREATE TABLE Planes (
     transponder_no  INTEGER,
     type            CHAR(20),
     PRIMARY KEY (transponder_no)
);
```

transponder_no uniquely identifies a plane.

```
CREATE TABLE Seat_assignment (
     seat_no         CHAR(3),
     leg_room        CHAR(20),
     plane_id        INTEGER,
     PRIMARY KEY (plane_id, seat_no),
     FOREIGN KEY  (plane_id) REFERENCES
          Planes (transponder_no) ON DELETE CASCADE
);
```

Since Seats is a weak entity associated with a plane, we can merge the entity Seats and the relationship has_seats into a single table Seat_assignment. We allow cascaded deletes because Seats is a weak entity of the plane, and if the plane row is removed from the database the seats of that plane should be removed as well.

Problem 1: What if each flight has a fixed source/destination? (F->FrT)

Flight_Route(route_id, From, To)
Flights(flight_no, date, plane_id)
Uses_Route(flight_no, route_id) where route_id is a FK to Flight_Route table and not null. On delete no action since the route may be used by other flights

Problem 2: What if each flight has a fixed plane even on different days? (F->P)
Flight(flight_no, plane_id)
Flight_Schedule(flight_no, date), where **date** cannot be null by virtue of being part of a key (right?) In the ER, we have a participation constraint from Flights to its diamond leading to Date.
Flight_Route(route_id, From, To)
Uses_Route(flight_no, route_id) where route_id is a FK to Flight_Route table and not null. On delete no action since the route may be used by other flights