

# Proposed ER Diagram for CS2102 Project

CS2102

October 7, 2021

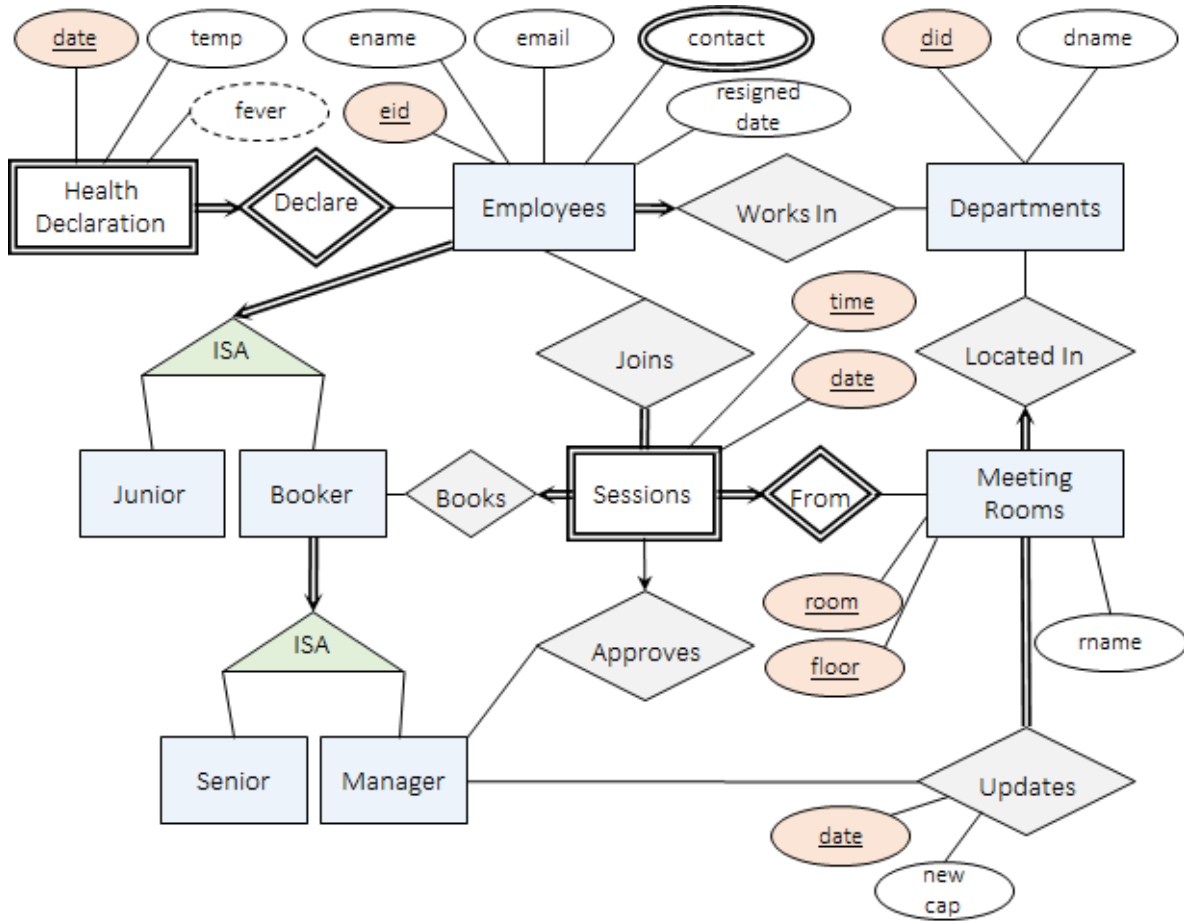


Figure 1: The proposed ER diagram.

## 1 Constraints

1. Each employee is assigned a unique employee ID (eid).
2. Each employee is assigned a unique email address (email).
3. Each employee records the following information: name (name), contact numbers (contact) and resignation date (resigned date).
4. Each department is identified by their unique department ID (did).
5. Each department records the following information: department name (dname).

6. Each meeting rooms can be uniquely identified by their floor number (**floor**) and their room number (**room**).
7. Each meeting rooms records the following information: room name (**rname**).
8. An employee must belong to exactly one department.
9. A department may have zero or more employee.
10. A meeting room must be located in exactly one department.
11. A department may have zero or more meeting room.
12. Each employee must be one and only one of the three kinds of employees: junior, senior or manager.
13. A junior employee cannot book any meeting room.
14. A senior or a manager can book meeting rooms.
15. A meeting room can only be booked by one group for the given date and time.
16. If an employee is having a fever, they cannot book a room.
17. Any employee can join a booked meeting.
18. The employee booking the room immediately joins the booked meeting.
19. If an employee is having a fever, they cannot join a booked meeting.
20. A manager approves a booked meeting
21. A manager can only approve a booked meeting in the same department as the manager (*i.e., the manager and the meeting room is in the same department*).
22. A booked meeting is approved at most once.
23. Once approved, there should be no more changes in the participants and the participants will definitely come to the meeting on the stipulated day.
24. A manager from the same department as the meeting room may change the meeting room capacity.
25. A booking can only be made for future meetings.
26. An employee can only join future meetings.
27. An approval can only be made on future meetings.
28. Every employee must do a daily health declaration.
29. A health declaration records temperature and the date.
30. A health declaration for a given employee can be uniquely identified by the date.
31. If the declared temperature is higher than 37.5 Celsius, the employee is having a fever.
32. The declared temperature can only be between 34 (*hypothermia*) to 43 Celsius (*hyperthermia*).
33. When an employee resign, all past records are kept.
34. When an employee resign, they are no longer allowed to book or approve any meetings.
35. Contact tracing constraints are omitted from this list.

## 1.1 Non-Captured Constraints

The following constraints are not captured: 2 (*can be captured in schema*), 16, 18 (*partially captured by total participation constraints from **Employees** to **Joins** which ensures that at least someone joins immediately but we cannot be sure that it is indeed the employee who books the meeting room*), 19, 23, 25, 26, 27, 32, 34 as well as any constraints related to contact tracing as summarised in 35. Constraints 25, 26, 27 and 32 are *common sense constraints*.

## 2 Design Decisions

### 2.1 Employees

In modern age, an employee may have multiple contact number. For instance, they may have landline and mobile phones. The use of multi-valued attributes is to reflect this but in practice, we will only allow up to three contact numbers:

1. Home phone number
2. Mobile phone number
3. Office phone number

### 2.2 Health Declaration

Health declaration is made into a weak entity set dependent on employees because each health declaration must be made by an employee. This allows the date to be a partial key. A more fine-grained partial key of date and time is unnecessary as we can simply replace the old value. Additionally, fever is a derived attribute because we can set it automatically from temperature recorded.

### 2.3 Meeting Rooms

The meeting room has no attribute (*derived or not*) for capacity. An update provides the date as a partial key which can be used to distinguish the actual capacity at any given time. To ensure that a meeting room have a capacity, total participation constraints between meeting rooms and updates is added.

However, this design complicates implementation due to a search required (via SQL queries) for the current applicable capacity. Note that this design allows for fine-grained management of capacity.

To see the problem with having derived attribute, consider the following case for the meeting room  $R$  with capacity  $C_0$  on the current date  $D$ :

- A meeting room  $R$  has its capacity reduced to  $C_1$  for date  $D+10$  (*i.e., 10 days from now*). Two things happens:
  1. All booked meetings after date  $D+10$  where the number of participant is greater than the capacity are removed.
  2. The derived attribute capacity is set to the new value  $C_1$ .
- A meeting is booked for date  $D+5$  with  $C_0$  number of participants. What should be the behaviour?
  - If the checking is done against the reduced capacity  $C_1$  via the derived attribute, then it is wrong as it should be allowed.
  - If the checking is done by querying the actual capacity for the given date which will reflect  $C_0$ , then the derived attribute is unnecessary.

A different more dangerous problem in light of the current pandemic can only be considered where the capacity is increased. To avoid the complications above, we decide on a more complicated implementation.

## 2.4 Booking and Session

A session is a weak entity set dependent on the meeting room to ensure that each meeting room can only have exactly one session with a session having a partial key of date and time. Here we have the assumption that time is the start time with each session lasting only one hour.

An alternative representation of using an aggregate was considered but it allows for a meeting room to be booked by different senior/manager at the same date and time due to the tuple  $\langle \text{eid (from Booker)}, \text{date, time, room (from Meeting Room)}, \text{floor (from Meeting Room)} \rangle$  being the key.

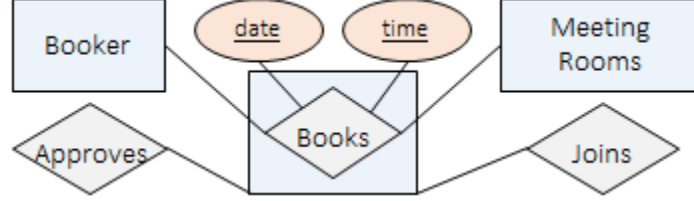


Figure 2: Aggregate alternative.

To ensure that a session must be created at booking time, a key and total participation constraints to books is added. Furthermore, to ensure that only a single approval is made, a key participation constraint to approves is added. Lastly, to partially ensure Constraint 18 is satisfied, a total participation constraints between employees and joins is added.

## 2.5 Junior, Senior and Manager

The two-level ISA hierarchy is used to first separate an employee that is allowed to book (*i.e.*, a *booker*) and an employee that is not allowed to book (*i.e.*, a *junior*). Secondly, we then separate a manager from a senior since only a manager can approve a meeting.

An alternative three-level ISA hierarchy was considered as a reflection of increasing privileges/capabilities of each employee but not chosen due to the possible increase in complexities in the implementations. Note that in the alternative, the arrow is unnecessary as there can never be an overlap due to the absence of choices of sub-entity sets.

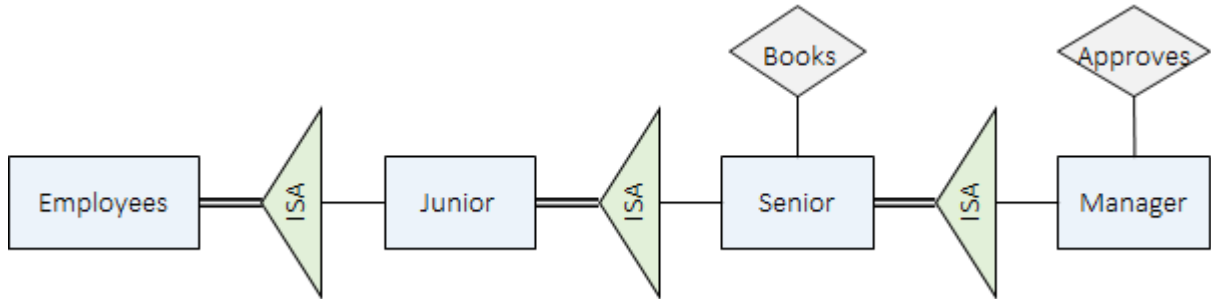


Figure 3: ISA alternative.

Additionally, the chosen design allows for a possible distinct responsibilities between senior employees and manager in the future. On the other hand, the alternative design does not allow for this possibilities since a manager is a senior employee with added responsibilities.