1. *(8 points)* Consider the ER diagram in Figure 7.22. Assume that an employee may work in up to two departments or may not be assigned to any department. Assume that each department must have one and may have up to three phone numbers. **Supply \((\text{min, max)}\) constraints on this diagram.** State clearly any additional assumptions you make. **Under what conditions would the relationship \text{HAS\_PHONE} be redundant in this example?**

Assume the following additional system requirements:

- Each department can have anywhere between 1 and 30 employees.
- Each phone is used by one, and only one, department.
- Each phone is assigned to at least one, and may be assigned to up to 30 employees.
- Each employee is assigned at least one, but no more than 5 phones.

![Figure 7.22](image)

*Figure 7.22* Part of an ER diagram for a COMPANY database.

The relationship \text{HAS\_PHONE} would be redundant when employees have all the phones of their department(s) and none of any other department.

2. *(8 points)* Consider the ER schema for the MOVIES database in Figure 7.24. Assume that MOVIES is a populated database. \text{ACTOR} is used as a generic term and includes actresses. Given the constraints shown in the ER schema, respond to the following statements with True, False, or Maybe. Assign a response of Maybe to statements that, while not explicitly shown to be True, cannot be proven False based on the schema as shown. Briefly justify each answer.
(a) There are no actors in this database that have been in no movies.

T—**Actor** has full participation in **Performs_In**.

(b) There are some actors who have acted in more than ten movies.

M—The max cardinality on **Actor-Performs_In-Movie** is \( N \), so this is neither required nor ruled out.

(c) A movie can have only a maximum of two lead actors.

T—The max cardinality on **Movie-Has_Lead_Role-Actor** (reading the relationship backwards) is 2.

(d) Every director has been an actor in some movie.

M—This can’t be false because the relationship **Also_A_Director** exists, but it can’t be true because **Director** doesn’t have total participation in it.

(e) No producer has ever been an actor.

M—The same reasoning as the previous question applies here, looking at the **Actor.Producer** relationship.

(f) There are movies with more than a dozen actors.

M—The max cardinality on **Movie-Is_Performed_In_By-Actor** (reading backwards) is \( M \), so this is possible but not required.

(g) Most movies have one director and one producer.

M—The min and max cardinality bounds are just that, min/max bounds. They can say nothing about average, typical, or “most” cases. *Not that they aren’t important for examining the performance of a design!*

(h) No movie has a director who also acted in that movie.

M—Like before, nothing about these constraints either mandates or prohibits this.
3. (16 points) A database is being constructed to keep track of the teams and games of a sports league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of the game. Design an ER schema diagram for this application, stating any assumptions you make. Choose your favorite team sport (e.g., baseball, curling, kabbadi, ...). Be sure your design is described in a way understandable by someone not familiar with that sport.

Here’s one example solution as provided by the textbook authors...
The following design may be used for a baseball league. Here, we assumed that each game in the schedule is identified by a unique Game#, and a game is also identified uniquely by the combination of Date, starting Time, and Field where it is played. The Performance attribute of PARTICIPATE is used to store information on the individual performance of each player in a game. This attribute can be designed to keep the information needed for statistics, and may be quite complex. One possible design for the Performance attribute may be the following (using the notation of Figure 7.8):

\[
\text{Performance}(\text{Hitting}(\text{AtBat#}, \text{Inning#}, \text{HitType}, \text{Runs}, \text{RunsBattedIn}, \text{StolenBases}), \text{Pitching}(\text{Inning#}, \text{Hits}, \text{Runs}, \text{EarnedRuns}, \text{StrikeOuts}, \text{Walks}, \text{Outs}, \text{Balks}, \text{WildPitches}), \text{Defense}(\text{Inning#}, \text{FieldingRecord}(\text{Position}, \text{PutOuts}, \text{Assists}, \text{Errors})))
\]

Here, performance is a composite attribute made up of three multivalued components: Hitting, Pitching, and Defense. Hitting has a value for each AtBat of a player, and records the HitType (suitable coded; for example, 1 for single, 2 for double, 3 for triple, 4 for home run, 0 for walk, −1 for strikeout, −2 for fly out, ...) and other information concerning the AtBat. Pitching has a value for each inning during which the player pitched. Defense has a value for each inning a player played a fielding position. We can have a less detailed or a more detailed design for the performance of a player in each game, depending on how much information we need to keep in the database. Suitable variations of the ER diagram shown below can be used for other sports.