









## **QUESTION 01**

**Foreign Key:** I don't have any tool to represent foreign key in GitMind tool, it is represented with dotted underline.

In Participant entity Fks are Team Id , Event id,. In Ticket Entity FK are Event\_Id. Same as in Team Entity Department Id is Fk , in Event entity Exe\_Head\_ID is fk which represent the head (President) of event as mentioned in the case study.

### **Some Assumptions:**

- Participant can buy many ticket but each ticket is sold to only one participant
- As Participant can buy many tickets so it can also participate in more than 1one event
- Ticket is consider as weak entity why? Because it can not exist without event and participant.
- Each event have many only one executive but each executive maybe reponsible of more then one event. Same as executive may belong to any team , but it belongs to only one team and team may have many executives, but both are partially participated in a relation.
- But I assumed that team must have participants ,also every participant must have team so it is strongly participating.

## **QUESTION 02**

- It's important to talk about Cardinality, Relationship, min, max constraints, as it is based on some assumptions

### **The relationship between the Stadium and Match entities has the following constraints:**

- Each match is required to be played in one and only one stadium (minimum cardinality of Match to Stadium relationship is 1).
- A stadium may host multiple matches at different times, but it is not required to host any matches (minimum cardinality of Stadium to Match relationship is 0).
- Each match can only be played in one stadium (maximum cardinality of Match to Stadium relationship is 1).
- A stadium can host many matches, but each match can only be associated with one stadium (maximum cardinality of Stadium to Match relationship is many).
- **In summary**, a stadium must be linked to at least one match, but it can be linked to many matches. On the other hand, each match must be linked to exactly one stadium.

### **The relationship between the Match and Team entities has the following minimum and maximum cardinalities:**

- At least two teams must participate in each match (minimum cardinality of Match to Team relationship is 2).
- A team may not necessarily participate in any match (minimum cardinality of Team to Match relationship is 0).
- A match can involve many teams (maximum cardinality of Match to Team relationship is many).
- A team can participate in many matches at different times (maximum cardinality of Team to Match relationship is many).
- **In summary**, the constraints between the Match and Team entities can be stated as follows:
  - Each match must have at least two teams participating, and may have many teams.
  - A team can participate in many matches, but it is not required to participate in any match.

### **The "belongs to" relationship between the Match and Inning entities has minimum and maximum cardinalities as follows:**

-At least one inning must be associated with each match (minimum cardinality of Match to Inning relationship is 1).

-An inning can only be associated with a single match (maximum cardinality of Inning to Match relationship is 1).

-A match can have up to two innings associated with it (maximum cardinality of Match to Inning relationship is 2).

-An inning cannot exist independently and must always belong to a match (minimum cardinality of Inning to Match relationship is 1).

-In summary, a match must have at least one inning, and can have up to two innings associated with it. An inning cannot exist without being associated with a match and can only belong to one match.

**The "maximum innings" relationship between the Match and Inning entities has the following minimum and maximum cardinalities:**

-A match can have a maximum of two innings, and each inning can only belong to one match.

-An inning must belong to a match and cannot exist on its own (minimum cardinality of Inning to Match relationship is 1).

-A match must have at least one inning, but can have up to two innings (minimum cardinality of Match to Inning relationship is 1).

-Thus, the constraints between the Match and Inning entities in the "maximum innings" relationship are defined by these cardinalities.

**The relationship between the Team and Player entities in the "have" relationship is defined by minimum and maximum cardinalities as follows:**

-Each team must have a minimum of 11 players (minimum cardinality of Team to Player relationship is 11).

-A player can only be part of one team (maximum cardinality of Player to Team relationship is 1).

-A team can have 15 max players (maximum cardinality of Team to Player relationship is 15).

-Every player must be associated with a team (minimum cardinality of Player to Team relationship 1).

-Therefore, we can summarize the minimum and maximum constraints between the Team and Player entities in the "have" relationship as follows:

-Each team must have at least 11 players, and maximum of 15 players.

-A player can only be associated with one team .

-Every player must be part of a team.

**In the "has" relationship between Player and Statistics entities, the minimum and maximum cardinalities are as follows:**

-A player can be associated with zero or many statistics (minimum cardinality of Player to Statistics relationship is 0, and maximum cardinality is many).

-Each statistic is associated with only one player (maximum cardinality of Statistics to Player relationship is 1).

-For a statistic to exist, it must be associated with a player (minimum cardinality of Statistics to Player relationship is 1).

-The maximum cardinality of the Player to Statistics relationship is "many". This means that a player can have multiple statistics associated with them

-In summary, a player can have multiple statistics, and each statistic can only be associated with one player. Furthermore, every statistic must be associated with a player.

**ATTRIBUTES:**

-We don't have any tool to represent foreign key, as it has dotted line, I'm giving here the list of foreign keys such as in match entity team player id, in inning entity match id, teambat\_id is considered as foreign key.

-No of players is not considered as derived attribute because it is fixed in quantity.



### **QUESTION:03**

#### **MATCH PLAYER ENTITY:**

This Entity is used to store the record of player participation and match record of the player, we also make the participation totally, as we assumed that if there is any player then it must play the match so also it is strongly participated in relation or participating and plays, also if there is any record of match\_player then it must be of match so it is also totally participating and every map is totally participate in record relation because we're assuming that if there is any match done then its details will be saved in match\_player entity.

#### **TERNARY RELATION BETWEEN MAP, VEHICLE AND WEAPON:**

As, map can use specific weapon and specific so we make the only one relation "Special Use" between all these three entities as it increase the complexity while implementing it in DBMS, in that case we can also make binary relation with each but that thing just increasing the relations between entities.

### **QUESTION:04**

- Each Post is exactly made by only 1 user.
- Here job is job posted which is done by companies and it is dependent on company without company job entity can't exist so it is weak entity.
- Ternary relation I made between between job, post and company; as company can post the jobs on linked-in.
- One company can post many jobs on linkedin, but each job post is done by only 1 company and only 1 post will be done on linkedin of 1 job.
- There will be many posted done by the company related to job.
- As job is weak entity so relation-ship will become identifying.

#### **Problem**

A user can like the post once not twice, I think it is given for implementing it in DBMS as a constraint because I don't really think so how can we show user can like the post once as in constraint we just tell that minimum user is 0 means no one like the post and maximum users will be many, same as user can like no post or many posts, so how can we implement that thing?