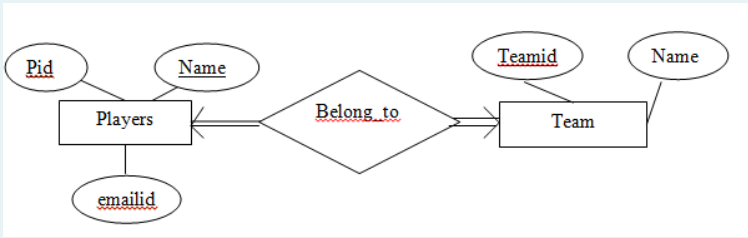


[Dashboard](#) / [My courses](#) / [1152_CS F212](#) / [Topic 4](#) / [Lab 3 Quiz Part A](#)

Started on	Friday, 11 February 2022, 3:01 PM
State	Finished
Completed on	Friday, 11 February 2022, 3:15 PM
Time taken	13 mins 3 secs
Grade	3.50 out of 10.00 (35%)

Question **1**
Complete
Mark 1.00 out of 2.00



Consider the relational schema corresponding to above ER without referential integrity and mapping constraints. Thus Players table has columns Pid as a primary key, Name of the player and emailed. The Teams table has columns Teamid as a primary key, Name of the team.

How to implement the following constraints?

All players must belong to some team and one player belongs to one team only.

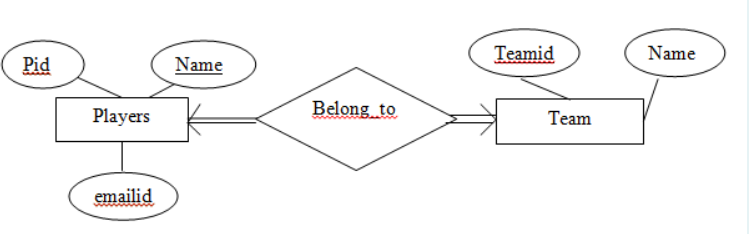
There can be one or more correct answer. Mark all correct options

- ☐ a. By creating a separate Belong_to table with columns Pid and Teamid as foreign keys.
- ☒ b. By adding a column Teamid in Players table as a foreign key and putting 'NOTNULL' constraint on Teamid.
- ☒ c. By adding a column Teamid in Players table as a foreign key and putting 'unique' constraint on Teamid.
- ☒ d. By creating a separate Belong_to table with columns Pid and Teamid as foreign keys and together as composite primary key.
- ☐ e. By adding a column Teamid in Players table as a foreign key.

The correct answer is:

By adding a column Teamid in Players table as a foreign key and putting 'NOTNULL' constraint on Teamid.

Question **2**
Complete
Mark 2.50 out of 3.00



Consider the relational schema corresponding to above ER without referential integrity and mapping constraints. Thus Players table has columns Pid as a primary key, Name of the player and emailed. The Teams table has columns Teamid as a primary key, Name of the team.

How to implement the following constraints?

A team can be without a captain but if it has, it can have only one captain which is one of the players (Pid) from Players table. Multiple teams cannot have same player as their captain i.e. a player cannot be a captain for more than one team.

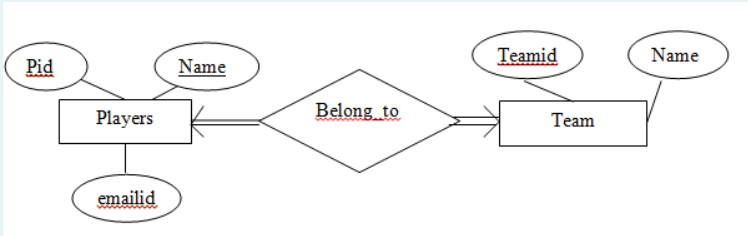
Mark all correct options

- ☐ a. Alter Teams table by adding a column Captain_id as foreign key referring to Pid of Players table.
- ☒ b. Alter Teams table by adding a column Captain_id as foreign key referring to Pid of Players table with 'Unique' constraint.
- ☐ c. Alter Teams table by adding a column Captain_id with same datatype as Pid of Players and set default value as NULL.
- ☐ d. Alter Teams table by adding a column Captain_id with same datatype as Pid of Players.
- ☒ e. Alter Teams table by adding a column Captain_id with same datatype as Pid of Players and set default value as NULL with unique constraint.
- ☐ f. Maintain a separate table with only Teamid and Captain_id as foreign keys and making together as composite primary key.
- ☐ g. Alter Teams table by adding a column Captain_id as foreign key referring to Pid of Players table and making columns Teamid and Captain_id as composite primary key.

The correct answer is:

Alter Teams table by adding a column Captain_id as foreign key referring to Pid of Players table with 'Unique' constraint.

Question **3**
Complete
Mark 0.00 out of 2.00



Consider the relational schema corresponding to above ER without referential integrity and mapping constraints. Thus Players table has columns Pid as a primary key, Name of the player and emailed. The Teams table has columns Teamid as a primary key, Name of the team. A team can be without a captain but if it has, it can have only one captain which is one of the players (Pid) from Players table.

How to implement the following constraints?

If a captain player record is removed from the Players table, corresponding information must be correctly managed from other tables.

Mark all correct options

- ☐ a. By putting 'ON DELETE SET NULL' constraint on Players table.
- ☐ b. Do nothing. As Pid is a foreign key in Team table, as soon as its record is deleted from the Players table, it becomes invalid Pid in Team table and thus gets automatically deleted.
- ☐ c. By putting 'ON DELETE CASCADE' constraint on Teams table.
- ☐ d. First update the value of Captain_id in Teams table as NULL and then delete the corresponding entry from Players table.
- ☒ e. By putting 'ON DELETE SET NULL' constraint on Teams table.
- ☐ f. By putting 'ON DELETE CASCADE' constraint on Players table.

The correct answer is:

By putting 'ON DELETE SET NULL' constraint on Players table.

Question **4**
Not answered
Marked out of 3.00

Consider the relation Manages (manager_id, employee_id). How to implement following constraints?

- one manager manages one to many employees
- an employee reports to only one manager (if exists).
- Manager is one of the employees (thus manager_id has employee_id as information. Both are invalid if it does not exist in Employee table).
- An employee cannot be the manager of self (thus both the columns cannot have same employee id as information. The topmost employees don't have manager).

There can be one or more correct answer. Mark all correct options

- ☐ a. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and by applying NOT NULL constraint on manager_id.
- ☐ b. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and by applying UNIQUE and NOT NULL constraint on employee_id.
- ☐ c. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and by applying UNIQUE constraint on employee_id.
- ☐ d. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and declaring both as composite primary key of Manager table.
- ☐ e. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and declaring employee_id as a primary key of Manager table.
- ☐ f. By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and by applying UNIQUE constraint on both the columns.

The correct answers are:

By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and declaring employee_id as a primary key of Manager table.,

By declaring both manager_id and employee_id as foreign key referring to employee_id of Employee table and by applying UNIQUE and NOT NULL constraint on employee_id.

◀ Insert statements for Lab 3 exercise.

Jump to...

Lab 3 Quiz Part B ▶