Santhoshi Rishitha

EPPS 6354

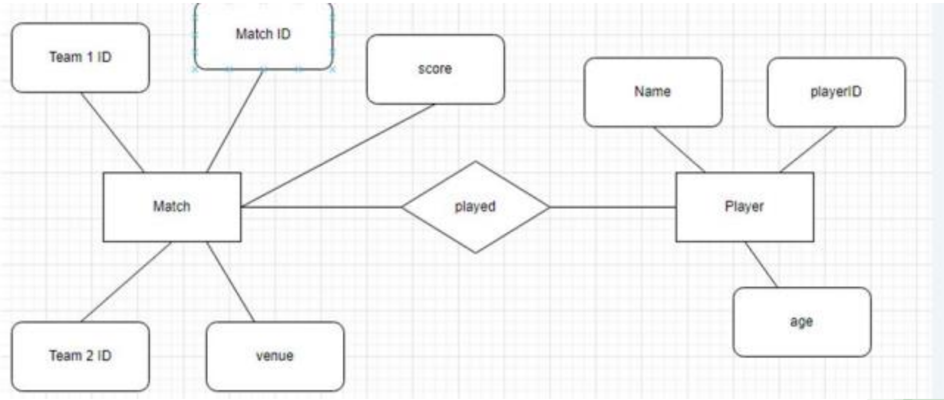
Assignment 4

- **1. Explain the difference between a weak and a strong entity set. Use an example other than the one in Chapter 6 to illustrate. (Consult Ch. 6, 6.5.3)**

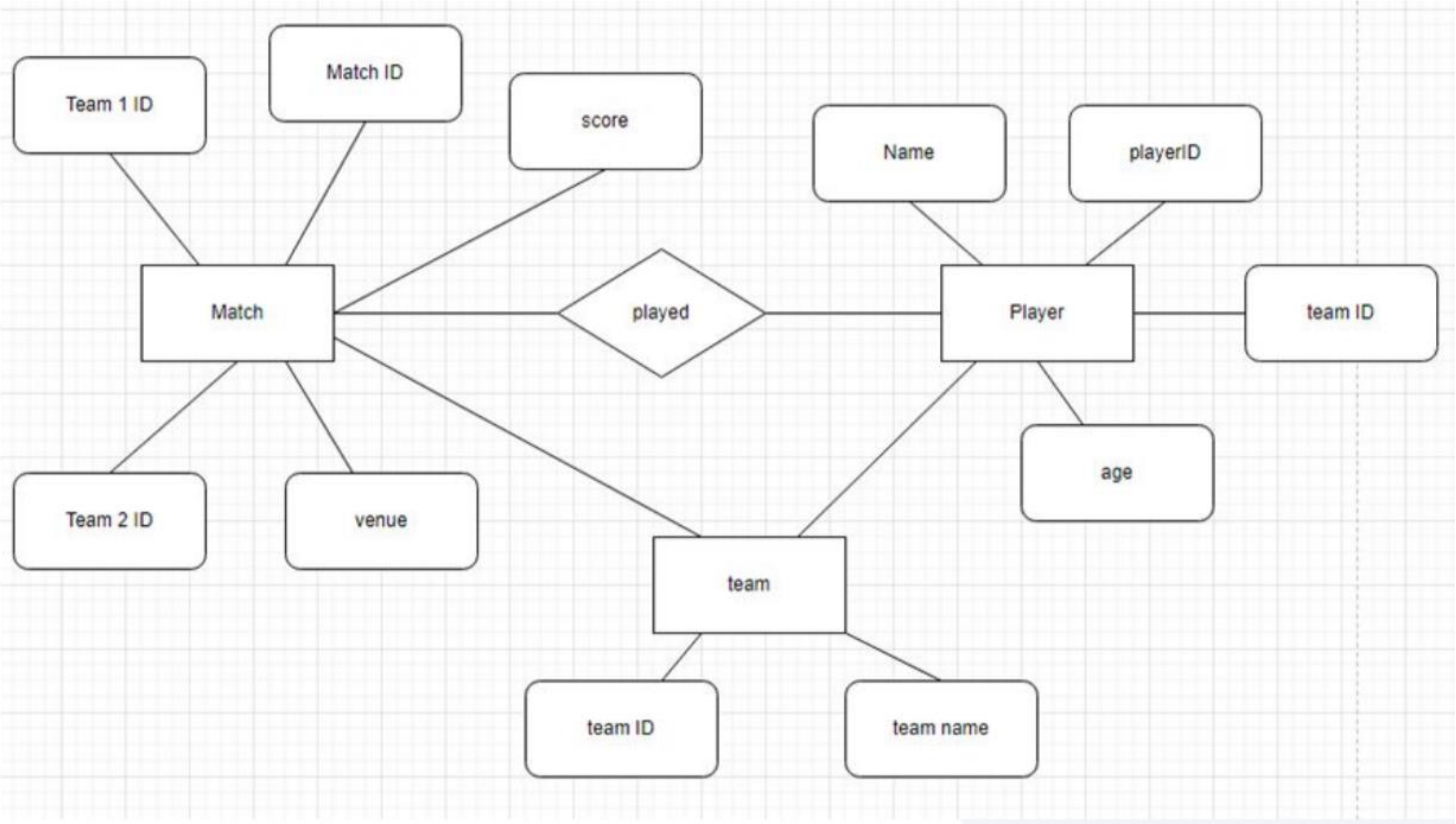
Strong entities are the ones that isn’t dependent on any other attribute. They are the ones that have a primary key as well. Weak entities need a strong entity. For example in a Healthcare records system, a patient is a strong entity and data about its “Emergency Contact” is a weak entity

Design an E-R diagram for keeping track of the scoring statistics of your favorite sports team. You should store the matches played, the scores in each match, the players in each match, and individual player scoring statistics for each match. Summary statistics should be modeled as derived attributes with an explanation as to how they are computed. (Consult: https://www.db-book.com/db7/practice-exer-dir/PDF-dir/6s.pdf)

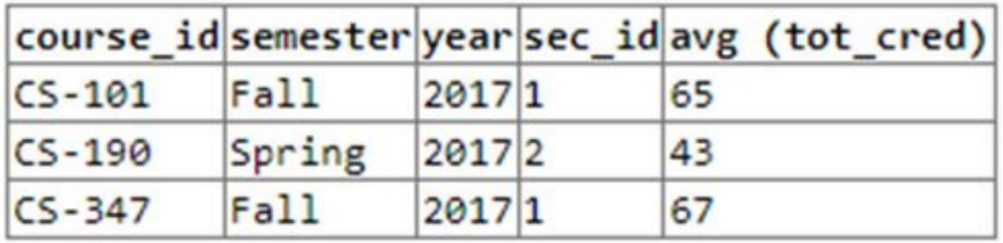
- a Draw the E-R diagram using draw.io. Read this website for instructions.



b) Expand to all teams in the league (Hint: add team entity)



3. SQL exercise: a) Consider the query select course_id, semester, year, sec_id, avg (tot_cred) from takes natural join student where year = 2017 group by course_id, semester, year, sec_id having count (ID) >= 2;



course_id	semester	year	sec_id	avg (tot_cred)
CS-101	Fall	2017	1	65
CS-190	Spring	2017	2	43
CS-347	Fall	2017	1	67

- i. Explain why appending natural join section in the from clause would not change the result. (Consult Ch. 4, 4.1.1)

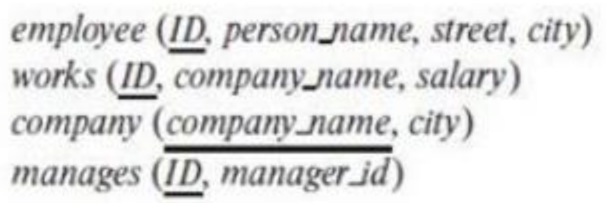
No columns for the takes table are mentioned, hence no value from takes will be shown

- ii. Test the results using the Online SQL interpreter (https://www.db-book.com/db7/university-lab-dir/sqljs.html)

b) Write an SQL query using the university schema to find the ID of each student who has never taken a course at the university. Do this using no subqueries and no set operations (use an outer join). (Consult Ch. 4, 4.1.3)

```
select s.ID from student s
left outer join takes t on s.ID=t.ID
where t.ID is null
```

Consider the following database, write a query to find the ID of each employee with no manager. Note that an employee may simply have no manager listed or may have a null manager(use natural left outer join). (Consult Ch. 4, 4.1.3)



employee (*ID*, *person_name*, *street*, *city*)
works (*ID*, *company_name*, *salary*)
company (*company_name*, *city*)
manages (*ID*, *manager_id*)

```
select e.ID from employee e left outer join manages m on e.ID = m.ID where m.manager_id is null;
```