Title	Assignment 04: Information Management (GISC 6354)
Handed Out	Thursday, February 15, 2024
Name	Victoria Ebeh
Email	Vae230000@utdallas.edu

Q1: 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)

A weak entity set has its existence dependent on another entity set referred to as its identifying entity set. Instead of having its own primary key, a weak entity uses the primary key of the identifying entity set along with extra attributes, known as discriminator attributes, to uniquely identify itself. But a strong entity set has its own independent primary key.

For example, we can consider two entity sets, "Pets", which is the identifying set with primary key PetID, and "PetVaccine" which is the weak entity set. Each vaccine record doesn't have a unique identifier on its own; its existence depends on the specific pet it is associated with. Instead of having a standalone primary key, "PetVaccine" might use a combination of attributes like PetID and additional attributes like VaccineDate to uniquely identify a vaccine record linked to a particular pet.

Q2: SQL exercise:

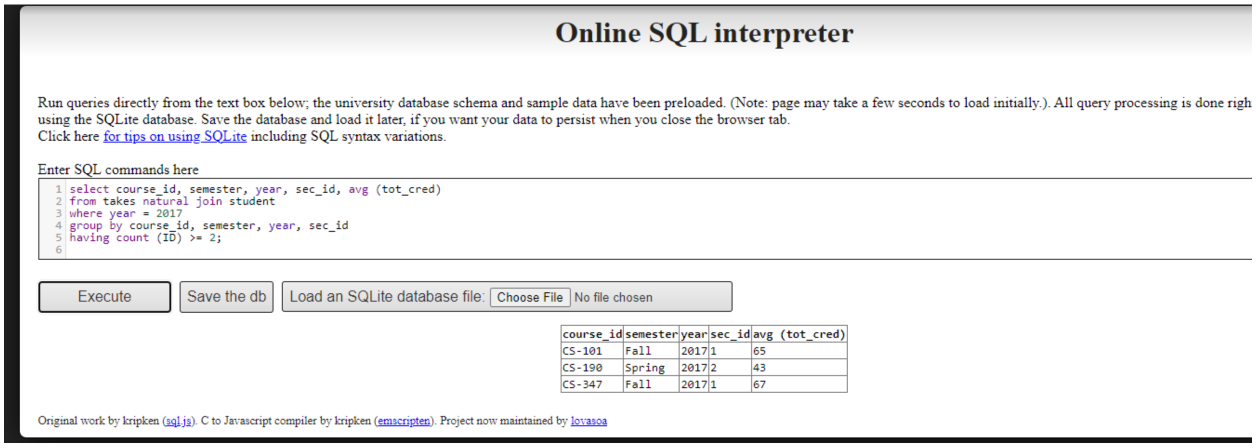
a) Consider the query

```
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;
```

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

Adding "natural join section" to the "FROM" clause wouldn't make a difference because the query already uses the "natural join" keyword to connect the "takes" and "student" tables based on shared column names. The "natural join" automatically combines columns with matching names in both tables. Since the "section" table is indirectly involved through "takes," explicitly stating "natural join section" won't change the result, as the conditions are already set by the existing join.

ii. Test the results using the Online SQL interpreter



Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done righ
using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab.
Click here for tips on using SQLite including SQL syntax variations.

Enter SQL commands here

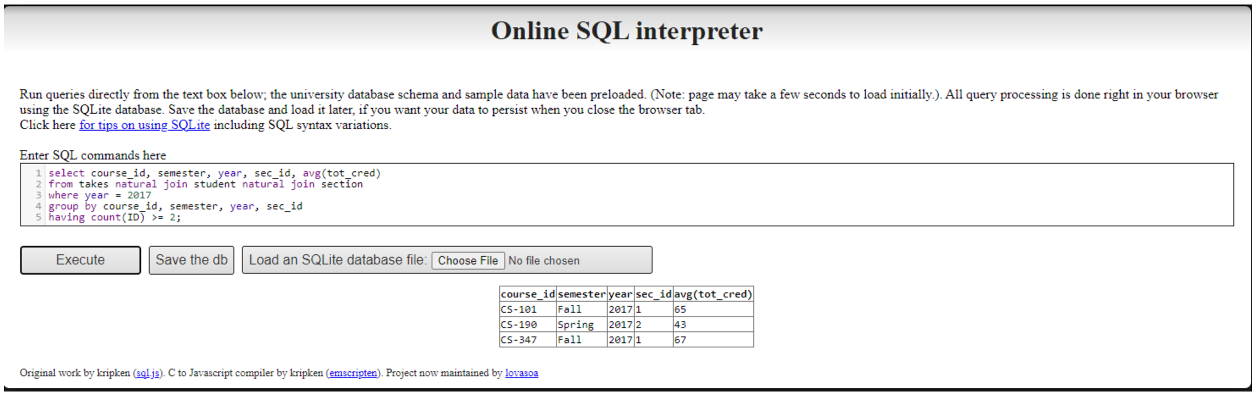
```
select course_id, semester, year, sec_id, avg (tot_cred)
from takes natural join student
where year = 2017
group by course_id, semester, year, sec_id
having count (ID) >= 2;
```

Execute | Save the db | Load an SQLite database file: Choose File No file chosen

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

Original work by kripken (sql.js). C to Javascript compiler by kripken (emscripten). Project now maintained by lovasoa

Figure 1: Image showing original query results



Online SQL interpreter

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser
using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browser tab.
Click here for tips on using SQLite including SQL syntax variations.

Enter SQL commands here

```
select course_id, semester, year, sec_id, avg(tot_cred)
from takes natural join student natural join section
where year = 2017
group by course_id, semester, year, sec_id
having count(ID) >= 2;
```

Execute | Save the db | Load an SQLite database file: Choose File No file chosen

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

Original work by kripken (sql.js). C to Javascript compiler by kripken (emscripten). Project now maintained by lovasoa

Figure 2: Image showing query results with appended 'section' table

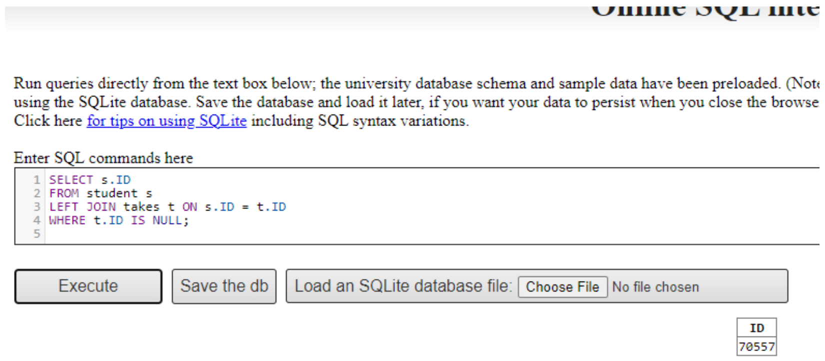
b) Write an SQL query using the university schema to find the ID of each student who has never taken a course at the university. Do this using no subqueries and no set operations (use an outer join). (Consult Ch. 4, 4.1.3)

```
SELECT s.ID

FROM student s

LEFT JOIN takes t ON s.ID = t.ID

WHERE t.ID IS NULL;
```



Online SQL inte

Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Not
using the SQLite database. Save the database and load it later, if you want your data to persist when you close the browse
Click here for tips on using SQLite including SQL syntax variations.

Enter SQL commands here

```
SELECT s.ID
FROM student s
LEFT JOIN takes t ON s.ID = t.ID
WHERE t.ID IS NULL;
```

Execute | Save the db | Load an SQLite database file: Choose File No file chosen

ID
70557

Figure 3: Results of query for student who has taken no course

Q3: 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: Practice Exercise solutions on textbook website)

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

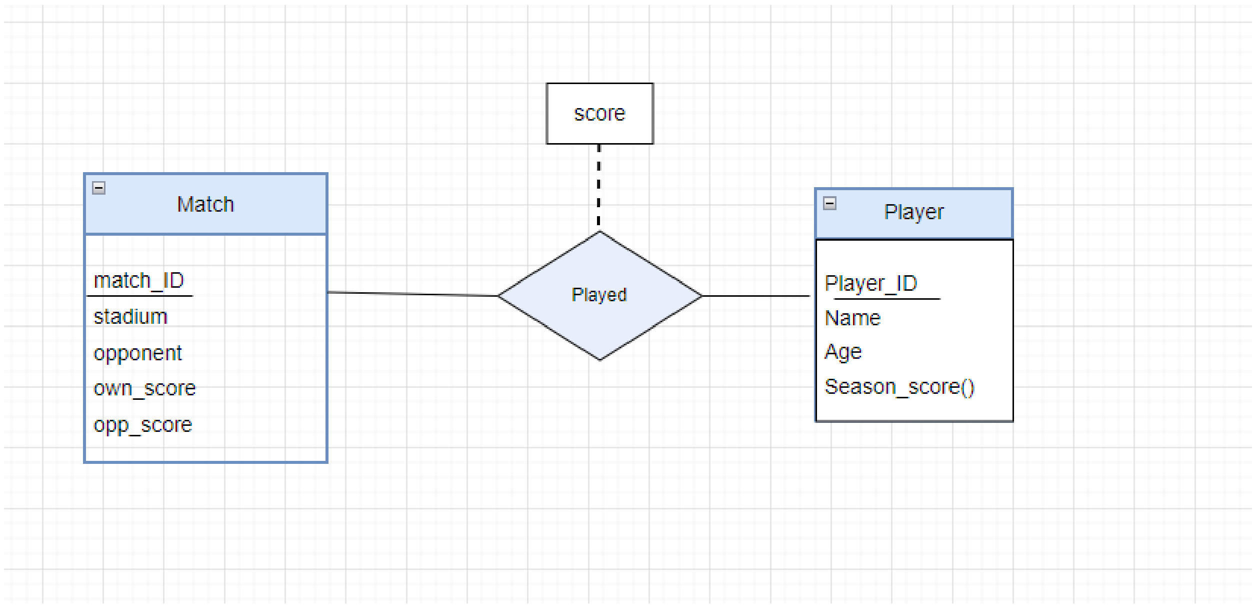


Figure 4: E-R Diagram showing favorite team statistics.

Season_score() is the derived attribute obtained by summing the score values associated with ‘player’ entity set through the ‘played’ relationship set. In type 2, points is the derived attribute obtained through the associated relationship between ‘match’ and ‘match players’ entity sets.

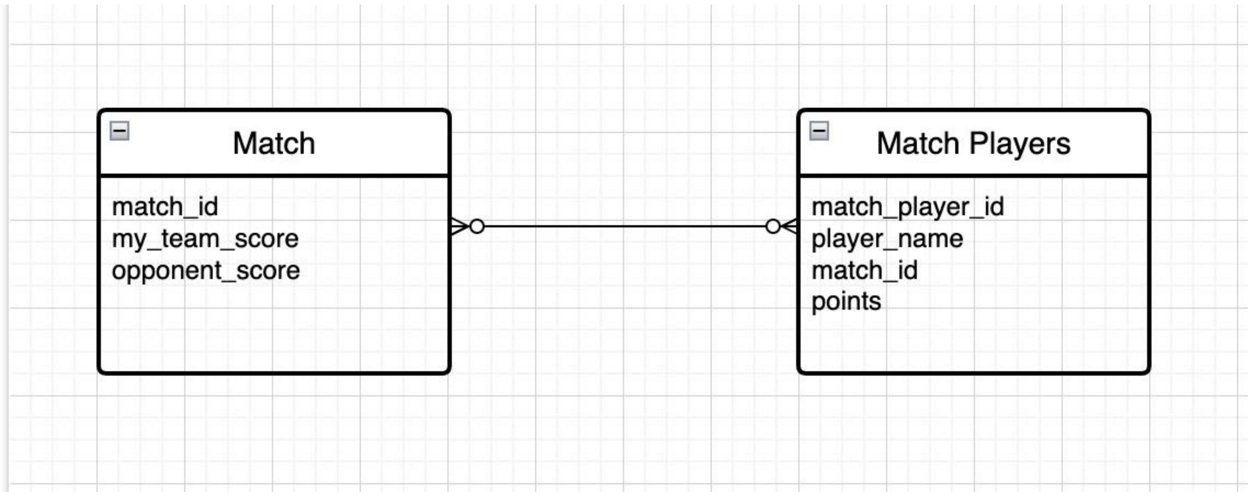


Figure 5: E-R Diagram showing favorite team statistics. Type 2

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

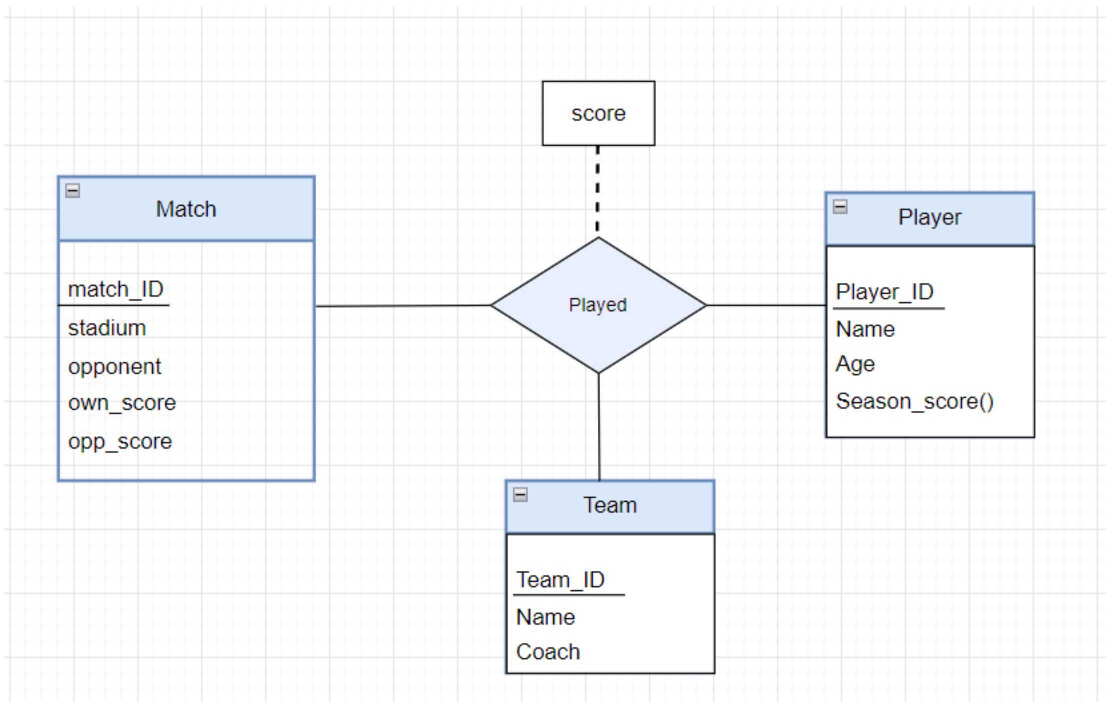


Figure 6: E-R Diagram showing favorite team statistics and their teams.

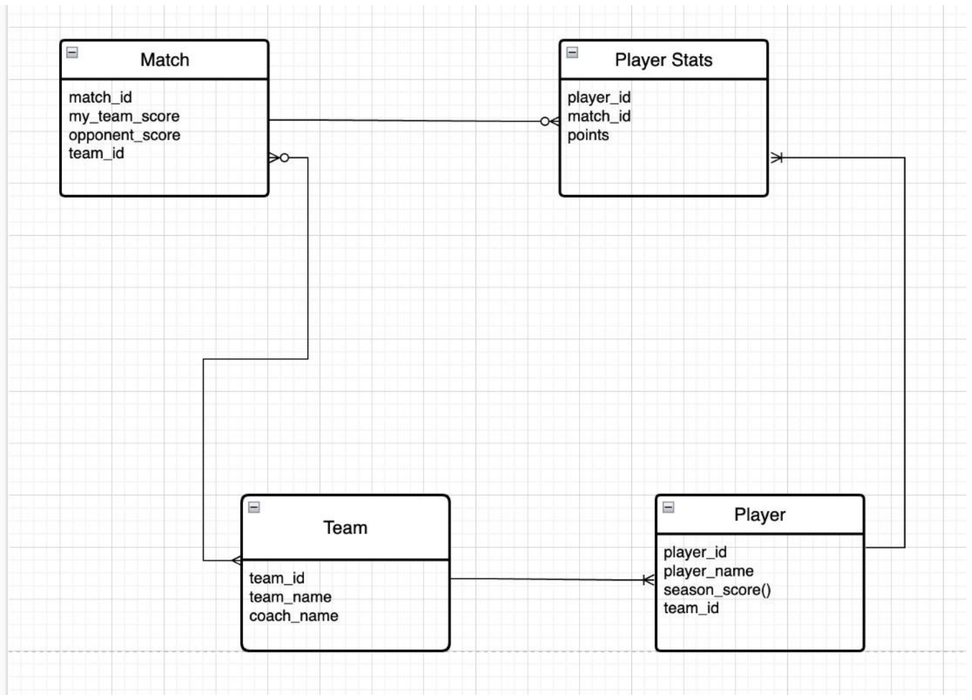


Figure 7: E-R Diagram showing favorite team statistics and their teams. Type 2