Assignment 04
EPPS 6354 Information Management

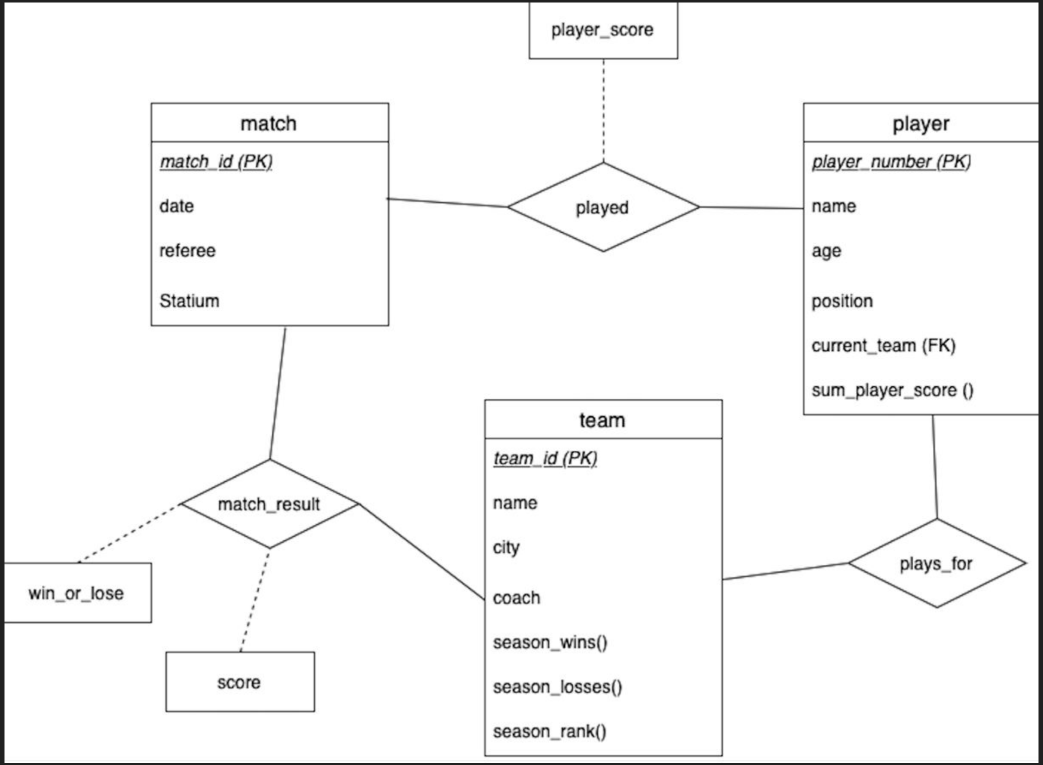
Oliver Jack Myers
UTD Spring 2025

1. Difference between weak and a strong entity set:

- A **weak entity** set does not have a **primary key** of its own. Instead, it relies on an **identifying entity set** and **its primary key**. The relationship linking the weak entity set to its owner is called the **identifying relationship.** Where as a **strong entity** set has its own **primary key**. It does not depend on any other entity set for its key.

- **Ex:** a hotel is a strong entity set because we can uniquely identify it by its primary key (hotel_Id, for example), whereas a room of that hotel would be a weak entity set because we can’t uniquely identify a room without considering the hotel

2. E-R diagram for scoring statistics of sports teams.



player_score

match
match_id (PK)
date
referee
Statium

played

player
player_number (PK)
name
age
position
current_team (FK)
sum_player_score ()

team
team_id (PK)
name
city
coach
season_wins()
season_losses()
season_rank()

match_result

win_or_lose

score

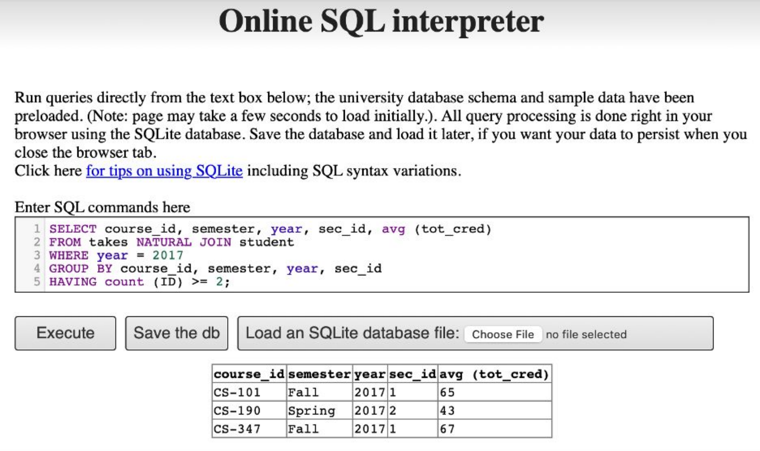
plays_for

Summary Statistics explained:

- **Season_rank()** = computed by taking all teams rank and organizing them by score.
- **Season_wins()** = computed by adding up total wins for the season.
- **Season_losses()** = computed by adding up total losses for the season.
- **Sum_player_score()** = computed by adding up all of the players scores together for the season.

3.A.i Why doesn’t NATURAL JOIN section change results?

Original query



Online SQL interpreter

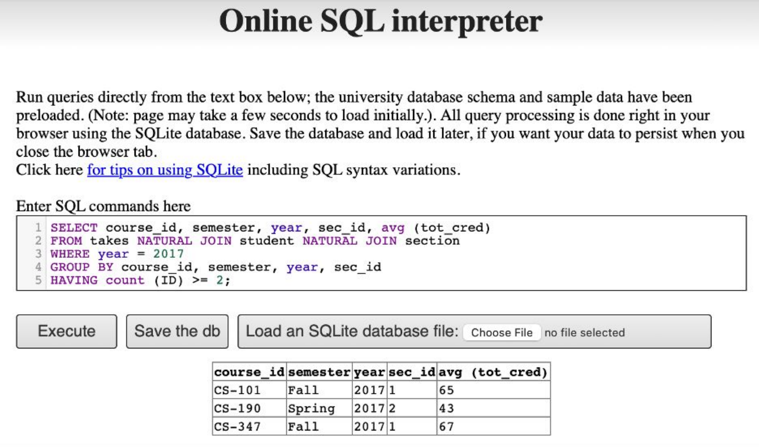
Run queries directly from the text box below; the university database schema and sample data have been preloaded. (Note: page may take a few seconds to load initially.). All query processing is done right in your browser using the SQLite database. Save the database and load it later, if you want your 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
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

Modified query



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.

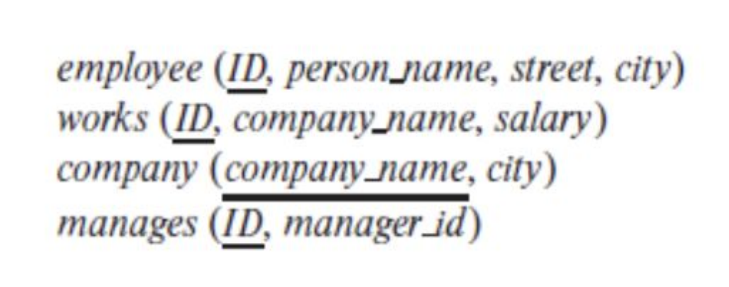
Enter SQL commands here

```
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;
```

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

Appending **natural join** section doesn't change the result because every takes row already matches a section row. More importantly, the join uses the same columns (course_id, semester, year, sec_id) that the query GROUP BYs, so the groups and the aggregate calculations remain identical."

3.C Write a Query:



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

```
SELECT ID

FROM employee

NATURAL LEFT OUTER JOIN manages

WHERE manager_id IS NULL;
```