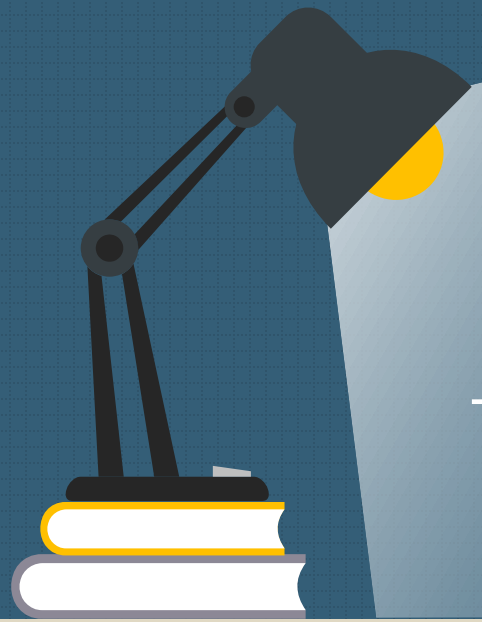


EPPS6354 Information Management

Assignment 4

Dohyo Jeong

University of Texas at Dallas

Assignment 4

Q1

▶ Explain the difference between a weak and a strong entity set.

Strong Entity Set:

Independent: Exists independently of other entities and can be identified uniquely without relying on them.

Primary key: Has its own primary key that uniquely identifies each instance within the set.

Participation: May or may not participate in relationships with other entities.

Representation: Shown as a single rectangle in an E-R diagram.

Weak Entity Set:

Dependent: Relies on a strong entity set for its existence and cannot be uniquely identified without it.

No primary key: Does not have its own primary key. And identified through a combination of its own attributes and a foreign key referencing the strong entity set it depends on.

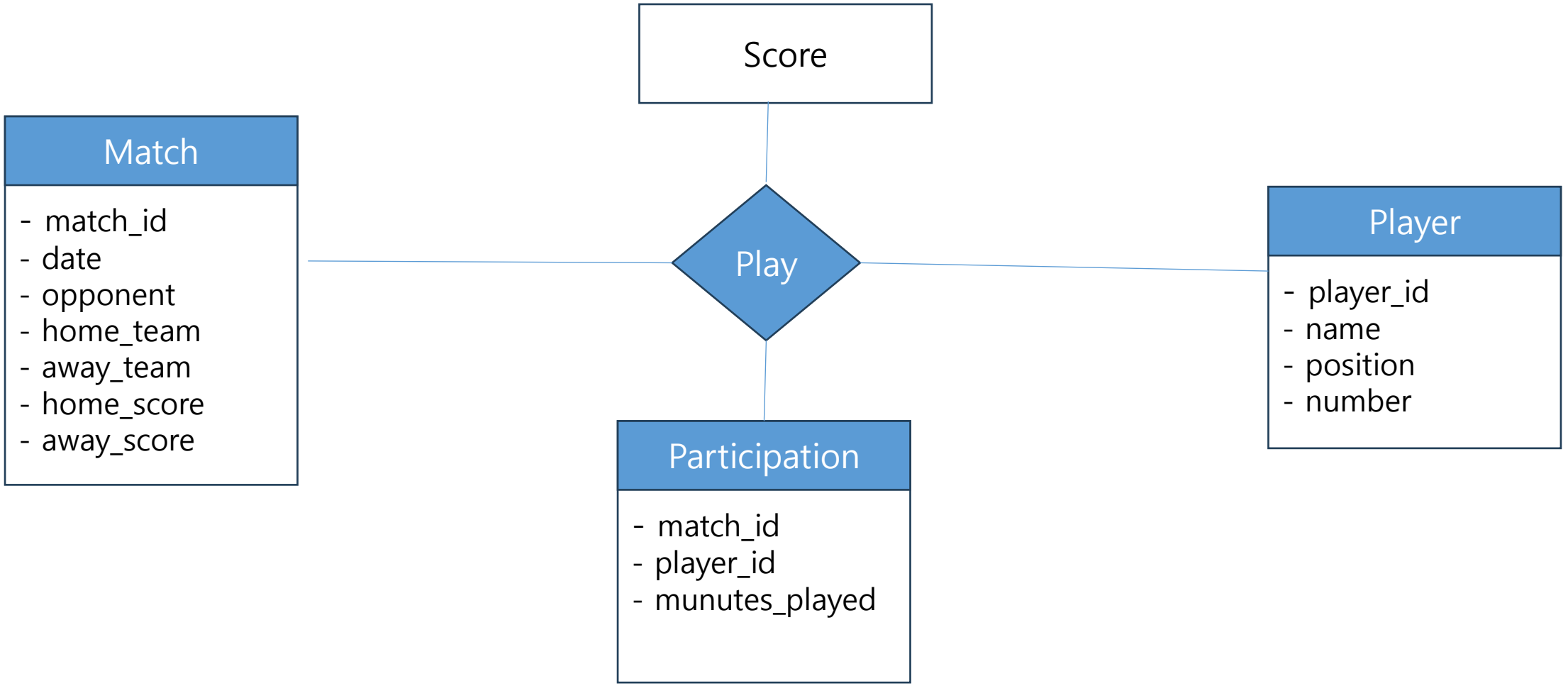
Total participation: Always participates in a relationship with the strong entity set it depends on.

Representation: Shown as a double rectangle connected to the strong entity set by a line with a crow's foot at the weak entity.

Assignment 4

Q2

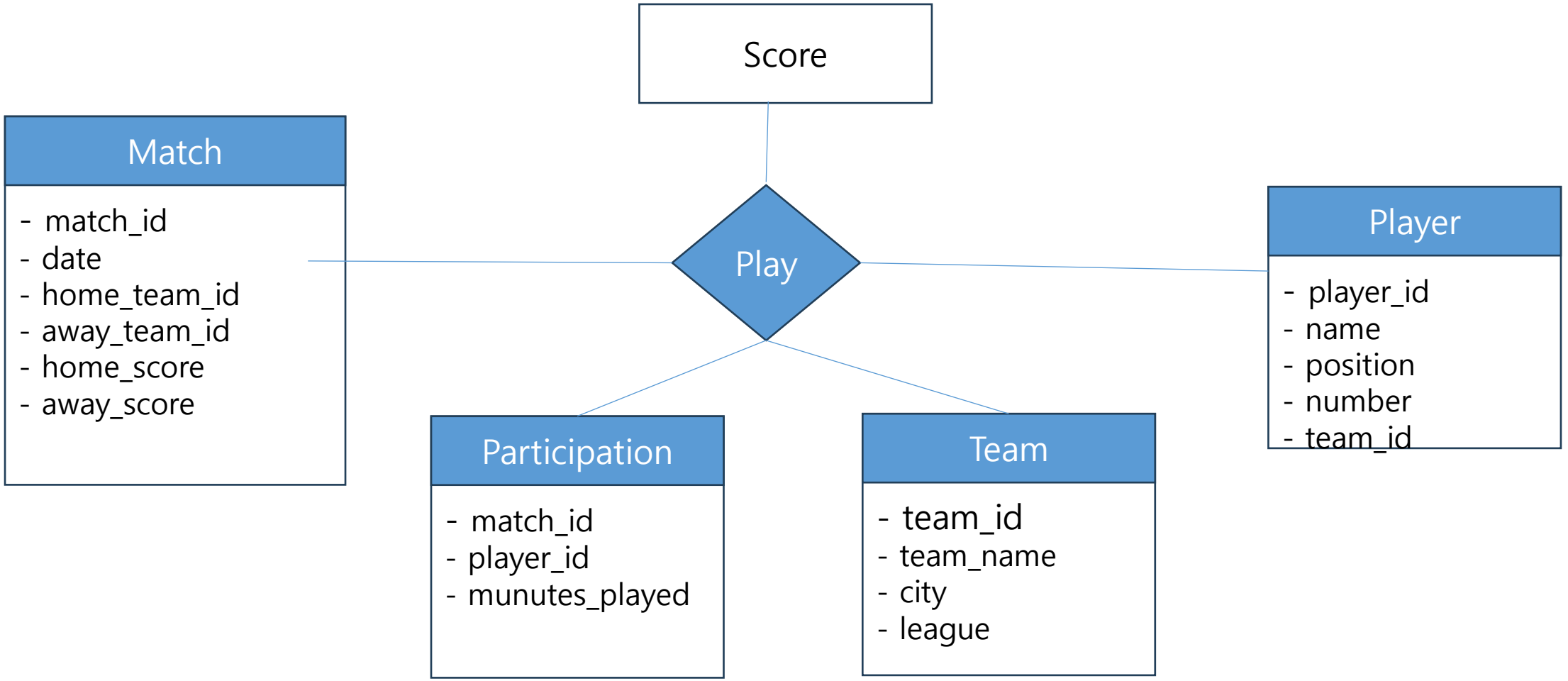
- Explain the difference between a weak and a strong entity set.



Assignment 4

Q2

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



Assignment 4

Q3

▶ Explain why appending natural join section in the from clause would not change the result

Natural join relies on matching attribute names: The natural join automatically joins tables based on columns with the same name and data type. However, the section table likely doesn't have attributes named course_id, semester, year, or sec_id. These attributes are specific to the takes table.

section table isn't relevant for aggregation: The group by clause already considers course_id, semester, year, and sec_id for grouping and calculating the average tot_cred. Adding section wouldn't contribute any new information to these groups.

having clause filters based on student count: The having count(ID) >= 2 clause ensures that only groups with at least two students enrolled are considered. Since sections don't have students enrolled, this filter wouldn't be affected by joining with the section table.

In essence, joining with the section table in this scenario wouldn't introduce any relevant information for the final result. It wouldn't affect the grouping or filtering criteria, and the natural join wouldn't find matching attributes to connect the tables meaningfully.

Therefore, ***appending natural join section wouldn't change the query's outcome.***

Assignment 4

Q3

- Test the results using the Online SQL interpreter

Enter SQL commands here

```
1 select course_id, semester, year, sec_id, avg (tot_cred)
2 from takes natural join student
3 where year = 2017
4 group by course_id, semester, year, sec_id
5 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

Assignment 4

Q3

- 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).

Enter SQL commands here

```
1 SELECT S.ID
2 FROM STUDENT S
3 LEFT JOIN TAKES T ON S.ID = T.ID
4 WHERE T.ID IS NULL;
```

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

ID
70557

Assignment 4

Q3

- 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).

```
SELECT e.ID
FROM employee e
LEFT OUTER JOIN manages m ON e.ID = m.ID
WHERE m.manager_id IS NULL;
```

Explanation:

- SELECT e.ID: Selects only the employee ID from the employee table.
- FROM employee e: Starts the query from the employee table and aliases it as e.
- LEFT OUTER JOIN manages m ON e.ID = m.ID: Performs a left outer join with the manages table on the matching ID attribute. This includes all employees, even those without a manager entry.
- WHERE m.manager_id IS NULL: Filters for employees where the manager_id in the manages table is NULL. This indicates they have no manager assigned.
- This query effectively leverages the left outer join to include all employees and then filters based on the null manager_id to identify those without a manager.


Thank
you!