

Information Management (EPPS 6354) Assignment 5

Submitted by
Samuel B. Adelusi
(BSA210004)

February 2023

School of Economic, Political and
Policy Sciences



**THE UNIVERSITY
OF TEXAS AT DALLAS**

Question 1

An E-R diagram can be viewed as a graph. What do the following mean in terms of the structure of an enterprise schema?

a) The graph is disconnected.

ANSWER

If a pair of entity sets are connected by a path in an E-R diagram, the entity sets are related, though perhaps indirectly.

A disconnected graph implies that there are pairs of entity sets that are **unrelated** to each other. If we split the graph into connected components, we have, in effect, a separate database corresponding to each connected component.

A disconnected graph in an E-R diagram means that there are one or more entity sets or relationship sets that are not connected to the rest of the diagram. In other words, there are parts of the schema that are not related to other parts of the schema. This could indicate a potential problem in the design of the schema, as it may lead to data redundancy, inconsistencies, or incomplete data representation.

Question 1

b) The graph has a cycle.

ANSWER

In an E-R diagram, a cycle in the graph means that there is a circular dependency between entities or relationships. This indicates that an entity depends on another entity or a relationship depends on another relationship that ultimately depends on the first entity or relationship, creating a loop.

For example, if entity A depends on entity B, and entity B depends on entity C, and entity C depends on entity A, then a cycle is created.

A cycle in an E-R diagram can indicate a potential problem in the schema design. It can lead to data inconsistencies and create issues with data manipulation and retrieval. For instance, it may become difficult to insert, update or delete data due to the presence of cyclic dependencies.

To address this issue, it is necessary to identify the entities and relationships involved in the cycle and to evaluate whether the cycle is necessary for the schema or if it can be eliminated by making changes to the design. Often, cyclic dependencies can be resolved by introducing an intermediary entity or by redesigning the relationships between entities.

Question 2

We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?

ANSWER:

A weak entity set is an entity set that does not have a primary key attribute of its own and depends on another entity set, called the owner entity set, to define its primary key. In other words, a weak entity cannot be uniquely identified without the context of the owner entity to which it is related.

While it is true that we can convert any weak entity set to a strong entity set by simply adding appropriate attributes, there are several reasons why we have weak entity sets in database design.

Firstly, weak entities are used to represent real-world situations where there are dependent objects that cannot exist without their owners. For example, consider an order in an e-commerce system. An order may have multiple line items, each representing a specific product that is ordered. The line item cannot exist without an order, so it is a weak entity that depends on the order entity.

Question 2

Secondly, by modeling weak entities explicitly, we can reduce data redundancy and improve data integrity. In the example above, if we simply added attributes to the order entity to represent the line items, we would have redundant data for each line item, which could lead to data inconsistencies if the same product is ordered multiple times. By modeling line items as a weak entity, we ensure that each line item has a unique identity and that the relationship between orders and line items is properly represented.

Finally, by explicitly modeling weak entities, we can improve the efficiency of database queries. For example, by indexing the primary key of a weak entity, we can perform faster lookups for related records.

Weak entity set is required for following reasons:

1. To avoid the inconsistencies caused by duplicating the key of the strong entity.
2. Weak entities reflect the logical structure of an entity being dependent on another entity.
3. Weak entities can be deleted automatically when their strong entity is deleted

In summary, we have weak entity sets in database design to accurately represent real-world situations, to reduce data redundancy and improve data integrity, and to improve the efficiency of database queries.

Question 4

Answer

4a (i) Find ID and name of each employee who lives in the same city as the location of the company for which the employee works.

```
SELECT e.ID, e.person_name
FROM Employee e
INNER JOIN Works w ON e.ID = w.ID
WHERE c.city = e.city
```

4a (ii) Find ID and name of each employee who lives in the same city and on the same street as does her or his manager.

```
SELECT e.ID, e.person_name
FROM Employee e
INNER JOIN Works w ON e.ID = w.ID
WHERE e.city = c.city AND e.street =
      (SELECT street FROM Employee WHERE ID = m.manager_id);
```

Question 4

4a (iii) Find ID and name of each employee who earns more than the average salary of all employees of her or his company.

```
SELECT e.ID, e.person_name
FROM Employee e, Works w
WHERE e.ID = w.ID
      AND w.salary > (SELECT AVG(salary) FROM Works);
```

Question 4

4b (i) Consider the following SQL query that seeks to find a list of titles of all courses taught in Spring 2017 along with the name of the instructor.

```
select name, title  
from instructor natural join teaches natural join section natural join course  
where semester = 'Spring' and year = 2017
```

What is wrong with this query? (Hint: check book website)

The issue with the original SQL query is that it assumes that the teaches table has only one row for each course instructor, which is not always the case if an instructor teaches multiple sections of the same course in the same semester.

A **NATURAL JOIN** compares all columns of two tables which have the same column-name and the **resulting** joined table contains those columns once.

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with the same name of associated tables will appear once only.

Natural Join: Guidelines

- The associated tables have one or more pairs of identically named columns.
- The columns must be the same data type.
- Don't use ON clause in a natural join.

Question 3a

- (i) Yes it will not change any result because natural join will not use comparison operator.

A **NATURAL JOIN** compares all columns of two tables which have the same column-name and the **resulting** joined table contains those columns once.

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with the same name of associated tables will appear once only.

Natural Join: Guidelines

- The associated tables have one or more pairs of identically named columns.
- The columns must be the same data type.
- Don't use ON clause in a natural join.

(ii)

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

