Quiz 2: Subqueries and Design

Q1 Functional Depenencies

8 Points

Consider the relation R(A, B, C, D, E). Say that we know the following functional dependencies hold:

A ➜ B

AC ➜ E

B ➜ D

Q1.1

2 Points

True/False: We can also claim that A ➜ C.

○ True

○ False

Save Answer   Last saved on **Feb 22 at 8:10 PM**

Q1.2

2 Points

True/False: We can also claim that A ➜ D.

○ True

○ False

Q1.3
2 Points

True/False: We can also claim that AC ➔ B.

○ True

○ False

Save Answer   Last saved on **Feb 22 at 8:10 PM**

Q1.4
2 Points

True/False: the set of attributes {A, B, C} is a super-key of relation R.

○ True

○ False

Save Answer   Last saved on **Feb 22 at 8:10 PM**

Q2 BCNF Decomposition
7 Points

Consider a relation R(A, B, C, D, E) satisfying the following functional dependencies:

CD ➔ AB

DE ➔ AC

Decompose R into BCNF. Select the answer that represents your decomposed relations:

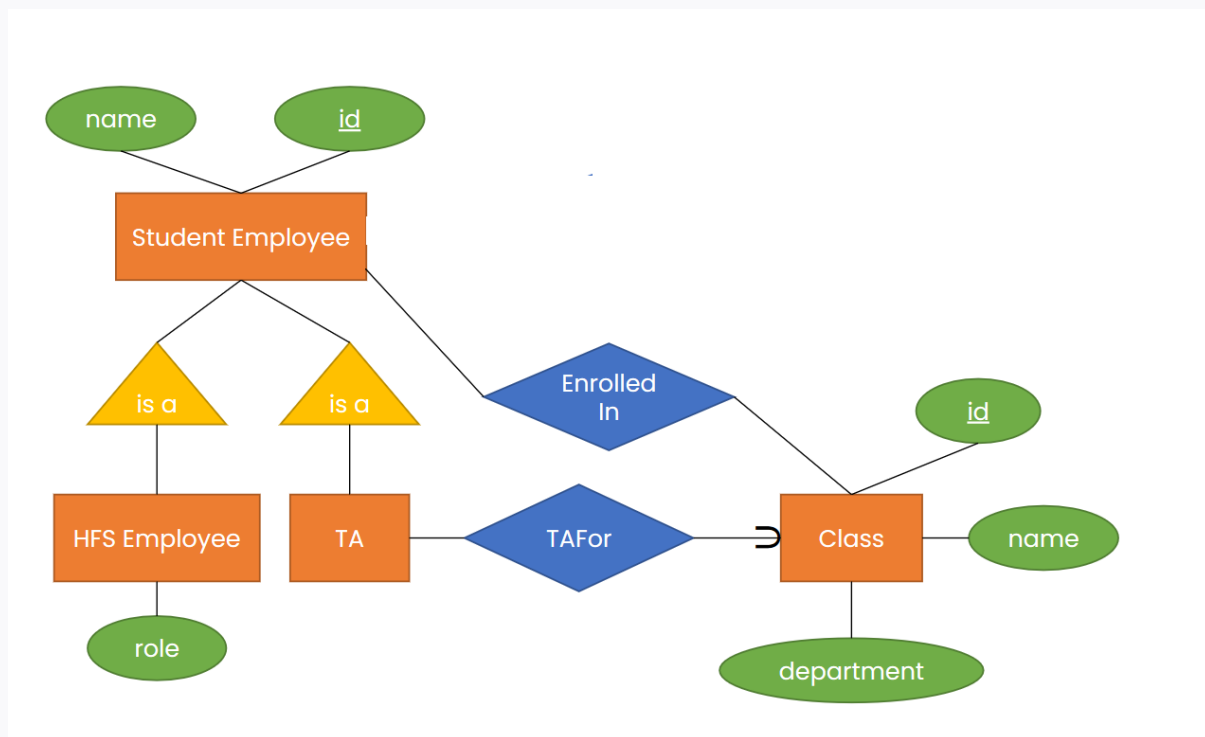○ R1(A, C, D, E), R2(B, C, D)

○ R1(A, B, C, D), R2(C, D, E)

R1(A, B, C, D), R2(C, D, E)

○ R1(A, B, C, D), R2(A, B, E)

○ R1(A, C, D, E), R2(B, D, E)

○ R1(A, C, D, E), R2(A, C, B)

Q3 Schema Design

13 Points

Q3.1

10 Points

Write an example schema for the ER diagram shown above. You do not have to write full CREATE TABLE statements, just write each table's name and attributes like this:

TableName(**attr1**, attr2, attr3)
TableTwo(**attr1,attr2**,attr3)

Please also distinguish the attributes that make up the primary key for each table in some way. The easiest way to do this might be to bold the attribute(s) as we have done, by wrapping the attribute in two asterixes, like so: **attr** -> **attr**.

Foreign key references can be added as follows: TableTwo(**attr1** REFERENCES TableName(attr1),**attr2**,attr3). If you need to indicate UNIQUE or NOT NULL, you can do so with those keywords in your summarized schemas. If you aren't sure how this should work, write full CREATE TABLE statements with whatever datatypes you choose.

```
Class (**id**, name, department);
Student_Employee(**id**, name);
Enrolled_In(
**ClassID** REFERENCES Class(id),
**StudentID** REFERENCES Student_Employee(id)
);
HFS_Employee(**StudentID** REFERENCES
Student_Employee(id), role);
TA(**StudentID** REFERENCES Student_Employee(id),
ClassID REFERENCES Class(id));
```

Save Answer   Last saved on **Feb 22 at 8:46 PM**

Q3.2
3 Points

What is the advantage of using subclasses for different types of student employees?

Because they will reference the value of the primary key in their superclasses, student employees, as the value of their own primary key, which insures the uniqueness of the id between different types of student employees.

Save Answer   ***Unsaved Changes**

Q4 Subqueries
4 Points

What are "correlated" subqueries, and why should we try to avoid them when possible?

Correlated subquery is the subquery that uses values from the outer query. Thus for every value in the outer query, we will execute the correlated subquery for once. As a result, we will have to execute the correlated query for lots of times if the number of values of the outer query is really large. And multiple execution will substantially slow the running time of the written code.

Save Answer   *Unsaved Changes

Save All Answers   Submit & View Submission ❯