Midterm Solutions

Zaki Malik

October 16, 2008

1. (3 points)

Assume the schema consists of two relations R(A,B,C) and S(D,E). Consider the following expressions:

(a) $\prod_{A,C}(\sigma_{D=2}(R \bowtie_{A=E} (S)))$

(b) $(\prod_{A,C}(R)) \bowtie_{A=E} (\sigma_{D=2}(S))$

Are algebraic expressions (a) and (b) equivalent? Use no more than two sentences to explain your answer.

SOLUTION

No: Note that the arity (number of attributes) of the relation output of expression (a) is 2 while the arity of the relation output of expression (b) is 4

2. (2 points)
Define a trigger

SOLUTION

Trigger is a procedure that starts automatically if specified changes occur to the DBMS.

3. (5 points: 1 + 1 + 1 + 1 + 1)
What are the ACID properties? Define each property.

SOLUTION
ACID Properties are:
– Atomic : Whole transaction or none is done.
– Consistent : Database constraints preserved.
– Isolated : It appears to the user as if only one process executes at a time.
– Durable : Effects of a process survive a crash.

4. (3 points)

What does the following expression state (write it in English)

$$\pi_{sname}(\pi_{sid}(\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)$$

SOLUTION

Find the names of suppliers who supply a red part that costs less than 100 dollars.

5. (22 points: 11 + 11)

This question tests how well you understand the algorithm for converting E/R diagrams to relational schemas. An E/R diagram when converted to relations (using the mechanical construction that we know and love) gives rise to the following relations:

R(<u>a</u>,b,c)

S(<u>a,d</u>)

T(<u>a,d</u>,f,g)

You may assume that the same symbols refer to the same attribute and different symbols refer to different attributes (e.g., the attributes a in the relations R, S, and T are the same). Your task is to reverse-engineer the E/R diagram from these relations; in other words, what E/R diagram could have produced these relations? For full credit, give two different E/R diagrams that could have produced these relations.

SOLUTION

The key to the solution is the statement "You may assume that the same symbols refer to the same attribute and different symbols refer to different attributes (e.g., the attributes a in the relations R, S, and T are the same)". The attribute a must come from the same entity set in the E/R diagram. The only two ways we know of "transferring" an attribute from one entity set to another is via weak entity sets and via inheritance. The solution using weak entity set must keep in mind that supporting relationships do not get their own relations.

5. (22 points: 11 + 11)

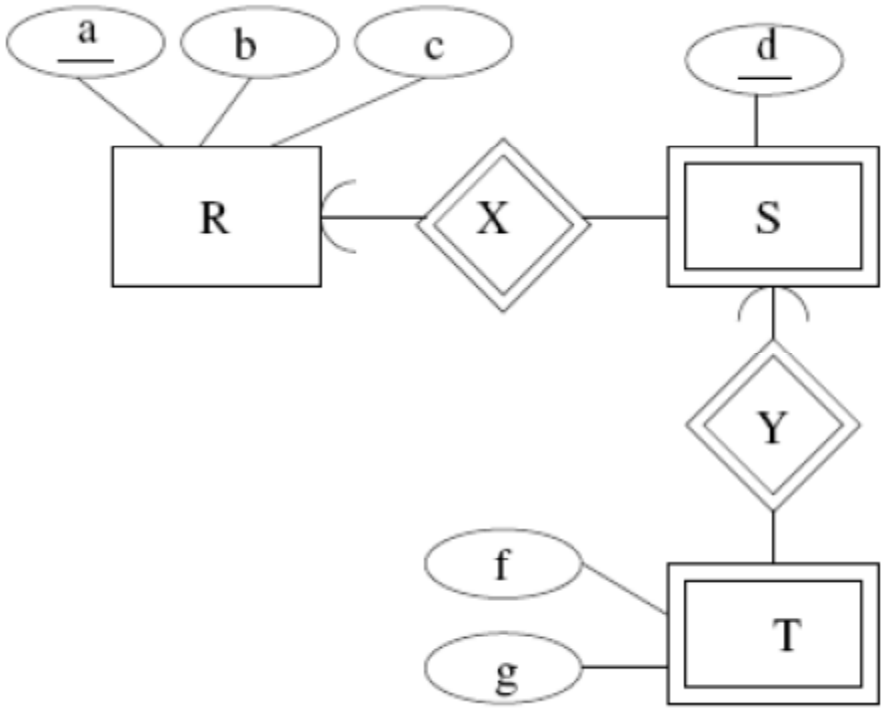
This question tests how well you understand the algorithm for converting E/R diagrams to relational schemas. An E/R diagram when converted to relations (using the mechanical construction that we know and love) gives rise to the following relations:

R(<u>a</u>,b,c)

S(<u>a,d</u>)

T(<u>a,d</u>,f,g)

You may assume that the same symbols refer to the same attribute and different symbols refer to different attributes (e.g., the attributes a in the relations R, S, and T are the same). Your task is to reverse-engineer the E/R diagram from these relations; in other words, what E/R diagram could have produced these relations? For full credit, give two different E/R diagrams that could have produced these relations.



5. (22 points: 11 + 11)

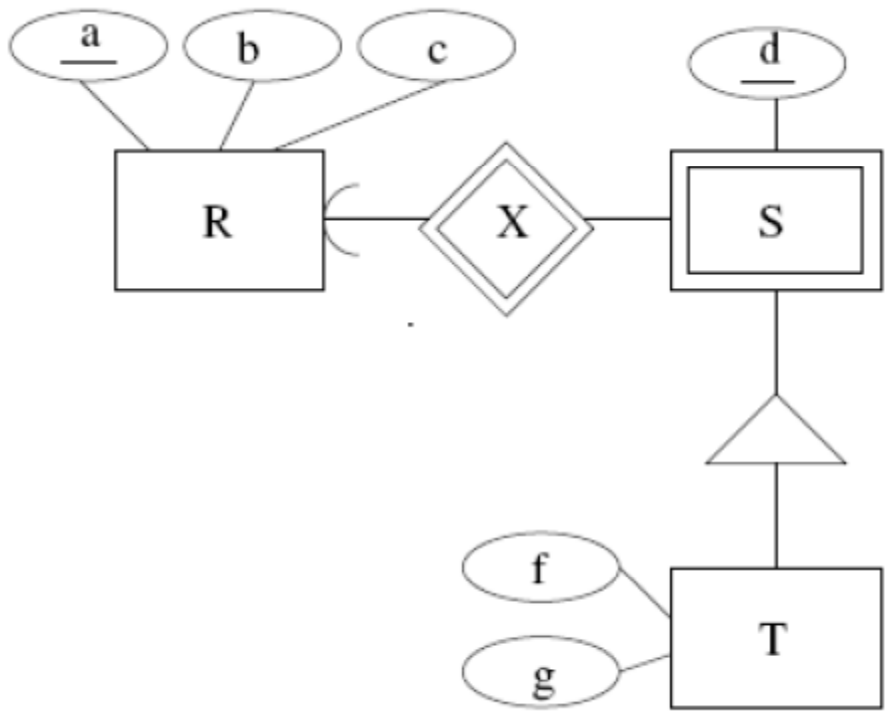
This question tests how well you understand the algorithm for converting E/R diagrams to relational schemas. An E/R diagram when converted to relations (using the mechanical construction that we know and love) gives rise to the following relations:

R(<u>a</u>,b,c)

S(<u>a,d</u>)

T(<u>a,d</u>,f,g)

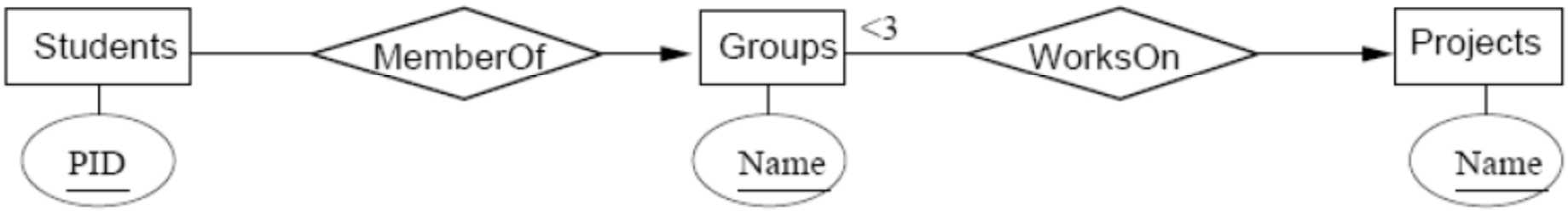
You may assume that the same symbols refer to the same attribute and different symbols refer to different attributes (e.g., the attributes a in the relations R, S, and T are the same). Your task is to reverse-engineer the E/R diagram from these relations; in other words, what E/R diagram could have produced these relations? For full credit, give two different E/R diagrams that could have produced these relations.



6. (10 points)
Draw an E/R diagram to model project groups in CS4604. Keep in mind that each enrolled student (identified by a PID) can work at most one project. Each project, identified uniquely by its name, can have at most two groups working on it. Be sure to identify all the appropriate multiplicity and referential integrity constraints in the diagram. Indicate key attributes in each entity set.

SOLUTION

The key to the answer is to introduce an entity set for project groups.
Both MemberOf and WorksOn must be many-one to ensure that each student works on at most one project.



7. (30 points)

Consider the relational database whose schema is shown below:

lives(<u>*person-name*</u>,*street*,*city*)

works(<u>*person-name*</u>, *company-name*,*salary*)

located-in(<u>*company-name*</u>,*city*)

manages(<u>*person-name*</u>,*manager-name*)

The primary key for each relation is denoted by the underlined attribute.

Write the following queries in Relational Algebra:

a) Find the name of all employess who work for the City Bank company (which is a specific company in the database).

b) Find the name and city of all employees who work for City Bank.

c) Find the name, street and city of all employees who work for City Bank and earn more than $10,000.

SOLUTIONS

a) (3 points)

$$\pi_{person-name}(\sigma_{company-name='City\ Bank'}(works))$$

b) (4 points)

$$\pi_{person-name,city}(\sigma_{(company-name='City\ Bank')\wedge(works.person-name=lives.person-name)}(works \times lives))$$

c) (5 points)

$$\pi_{person-name,street,city}(\sigma_{(company-name='City\ Bank')\wedge(salary>10000)\wedge(works.person-name=lives.person-name)}(works \times lives))$$

7. (30 points)
Consider the relational database whose schema is shown below:
lives(*person-name*,*street*,*city*)
works(*person-name*, *company-name*,*salary*)
located-in(*company-name*,*city*)
manages(*person-name*,*manager-name*)

The primary key for each relation is denoted by the underlined attribute.
Write the following queries in Relational Algebra:

d) Find all employees who live in the same city and on the same street as their manager.

d) (7 points)

$$\pi_{lives.person-name}(\sigma_{(lives.city=lives2.city)\wedge(lives.street=lives2.street)}(\sigma_{(lives.person-name=manages.person-name)\wedge(manages.manager-name=lives2.person-name)}(lives \times manages \times (\rho_{lives2}(lives)))))$$

7. (30 points)
Consider the relational database whose schema is shown below:
lives(<u>*person-name*</u>,*street*,*city*)
works(<u>*person-name*</u>, *company-name*,*salary*)
located-in(<u>*company-name*</u>,*city*)
manages(<u>*person-name*</u>,*manager-name*)

The primary key for each relation is denoted by the underlined attribute.
Write the following queries in Relational Algebra:

e) Find all persons who do not work for City Bank
f) Find all employees who live in the same city as the company they work for.

e) (3 points)

$$\pi_{person-name}(\sigma_{company-name \neq 'City\ Bank'}(works))$$

f) (8 points)

$$\pi_{lives.person-name}(\sigma_{(lives.person-name=located-in.company-name)}(\sigma_{(lives.person-name=works.person-name)\wedge(works.company-name=located-in.company-name)}(lives \times works \times located-in)))$$

8. (25 points)
Consider the schema:

Suppliers (<u>sid: integer</u>, sname: string, address: string)
Parts (<u>pid: integer</u>, pname: string, color:string)
Catalog (<u>sid: integer, pid: integer</u>, cost: real)

The key fields are underlined, and the domain of each field is listed after the field name. The *Catalog* relation lists the prices charged for parts by *Suppliers*. Write the following queries in SQL.

a) (5 points)
Find the *sids* of suppliers who supply some red part or are at 221 Packer Ave.

```
SELECT S.sid
FROM   Suppliers S
WHERE  S.address = '221 Packer street'
       OR S.sid IN ( SELECT C.sid
                     FROM   Parts P, Catalog C
                     WHERE  P.color='red' AND P.pid = C.pid )
```

b) (5 points)
Find pairs of *sids* such that the supplier with the first *sid* charges more for some part than the supplier with the second *sid*.

```
SELECT C1.sid, C2.sid
FROM   Catalog C1, Catalog C2
WHERE  C1.pid = C2.pid AND C1.sid ≠ C2.sid
       AND C1.cost > C2.cost
```

8. (25 points)
Consider the schema:

Suppliers (<u>sid: integer</u>, sname: string, address: string)
Parts (<u>pid: integer</u>, pname: string, color:string)
Catalog (<u>sid: integer, pid: integer</u>, cost: real)

The key fields are underlined, and the domain of each field is listed after the field name. The *Catalog* relation lists the prices charged for parts by *Suppliers*. Write the following queries in SQL.

c) (6 points)
Find the *pids* of parts that are supplied by at least two different suppliers.

```
SELECT  C.pid
FROM    Catalog C
WHERE   EXISTS (SELECT  C1.sid
                FROM    Catalog C1
                WHERE   C1.pid = C.pid AND  C1.sid ≠ C.sid )
```

8. (25 points)
Consider the schema:

Suppliers (sid: integer, sname: string, address: string)
Parts (pid: integer, pname: string, color:string)
Catalog (sid: integer, pid: integer, cost: real)

The key fields are underlined, and the domain of each field is listed after the field name. The *Catalog* relation lists the prices charged for parts by *Suppliers*. Write the following queries in SQL.

d) (7 points)
Find the *pids* of the most expensive parts supplied by at least two different suppliers.

```
SELECT C1.pid
FROM Catalog C1, Catalog C2
WHERE C1.pid = C2.pid AND C1.sid <> C2.sid
        AND C1.cost > C2.cost ;
```

8. (25 points)
Consider the schema:

Suppliers (<u>sid: integer</u>, sname: string, address: string)
Parts (<u>pid: integer</u>, pname: string, color:string)
Catalog (<u>sid: integer, pid: integer</u>, cost: real)

The key fields are underlined, and the domain of each field is listed after the field name. The *Catalog* relation lists the prices charged for parts by *Suppliers*. Write the following queries in SQL.

e) (2 points)
Write the equivalent SQL expression for the following:

$$\pi_{sid}(\pi_{pid}(\sigma_{color='red' \vee color='green'} Parts) \bowtie catalog)$$

SOLUTION

```
SELECT  C.sid
FROM    Catalog C, Parts P
WHERE   (P.color = 'red' OR P.color = 'green')
        AND P.pid = C.pid
```