CS145 Final Exam 8:30-11:30AM Friday Dec. 14, 2007

This exam is open book and notes. You may use a laptop, but please keep your sound on mute so you do not disturb others. You may access class notes or other written materials on your computer, even using the Internet if you have wireless access. However, you must not use your computer to run a DBMS, either over the Internet, or locally, should you have a DBMS that runs on your own computer.

This exam consists of 35 multiple-choice questions, so you have 5 minutes per question. Questions count 3 points each, with 1 point deducted for wrong answers (nothing deducted if you choose not to answer a question). The maximum score is 105.

Please circle your choice on each question: (a), (b), (c), or (d).

Print your name: _____

The Honor Code is an undertaking of the students, individually and collectively:

1. That they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

2. That they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code (signed): _____

注意事项：1. 审题一定要仔细看

2. 三思，避坑

In each of the first six questions, you are asked to compare two queries Q1 and Q2. You must tell whether the queries are:

1. The same [choice (a)], meaning that for every database the answers to the two queries are the same. That is, the same tuples are produced by each query, and a tuple is produced the same number of times by each query. The order in which tuples are produced is not to be considered.

2. Completely different [choice (d)], meaning that there are databases where Q1 produces more of some particular tuple, and other databases where Q2 produces more of some particular tuple. Note that the query producing the smaller number of copies of a tuple may produce zero copies of that tuple.

3. One is contained in the other but they are not the same [choice (b) or (c)]. For instance, Q1 is contained in Q2 if on every database, Q2 produces at least as many copies of each tuple as Q1 does. Note that it is possible Q2 produces one or more copies of a tuple, while Q1 produces none of that tuple.

General advice:

Do not assume a query has a trivial syntactic error and therefore produces nothing.

SQL relations may have NULL's, although in relational algebra, you should assume no NULL's unless stated otherwise.

In SQL it is possible that there may be duplicate tuples, but in relational algebra and Datalog assume the relations are sets unless stated otherwise.

For XPath and XQuery, think of the result as a bag of items, rather than a sequence (list) of items, so order does not matter.

Question 1. In the following, the schema of *R* is R(a,b).

Q1:

```
SELECT * FROM R;
```

(C)

Q2:

```
(SELECT * FROM R)
  INTERSECT
(SELECT * FROM R);
```

∩ 为集合操作，会去重

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 2. Consider XML documents containing information about students in classes and conforming

to the following DTD:

```
<!DOCTYPE Classes [
      <!ELEMENT Classes (Class*)>
      <!ELEMENT Class (Topic, Students)>
      <!ELEMENT Topic (#PCDATA)>
      <!ELEMENT Students (Student+)>
      <!ELEMENT Student EMPTY>
        <!ATTLIST Student Name #REQUIRED> ]>
```

Consider this query pair:

Q1: XPath:

```
/Classes/Class
  [Students/Student/@Name != Students/Student/@Name]/Topic
```

Q2: XQuery:

```
for $c in /Classes/Class
for $s1 in $c/Students/Student
for $s2 in $c/Students/Student
where $s1/@Name != $s2/@Name
return $c/Topic
```

Do not worry about `doc(…)` specifications or order of items in the returned result.

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 3. Assume the XML document in file xxxxxx conforms to the following DTD:

```
<!DOCTYPE X [
      <!ELEMENT A (B* C*)>
      <!ELEMENT B (#PCDATA)>
      <!ELEMENT C (#PCDATA)> ]>
```

The following are in XQuery:

Q1:

```
for $a in doc("xxxxxx")/X/A
return if (some $b in $a/B satisfies 1) then $a/C else ()
```

Q2:

```
for $a in doc("xxxxxx")/X/A
return if ($a/B) then $a/C else ()
```

Do not worry about order of items in the returned result.

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 4. Consider relations Items(itemID, sellerID) and Bids(itemID, bidderID) and the following

query pair. The answer is Bidding_closure in each query.

Q1: Datalog:

```
Bid_on(s, b) <- items(i, s) AND Bids(i, b) AND s <> b
Bidding_closure(x, y) <- Bid_on(x, y)
Bidding_closure(x, y) <- Bid_on(x, z) AND Bidding_closure(z, y)
```

Q2: SQL:

```
WITH RECURSIVE Bidding_closure(x, y) AS
        (SELECT sellerID, bidderID FROM Items, Bids WHERE items.itemID = bids.itemID AND
        sellerID <> bidderID)
        UNION
        (SELECT bc1.sellerID, bc2.bidderID FROM Bidding_closure bc1, Bidding_closure bc2 WHERE
        bc1.y = bc2.x)
SELECT * FROM Bidding_closure;
```

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 5. Suppose T is a UDT, R is a relation with a reference to T as one of its attributes, and S is a relation using T as a rowtype, all defined by:

```
CREATE TYPE T AS (a INT, B INT);
CREATE TABLE R(c INT, d REF T);
CREATE TABLE S OF T;

Q1:
SELECT rr.d->a
FROM R rr;

Q2:
SELECT ss.a()
FROM S ss;
```

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 6. The following are Datalog queries about two-attribute relations R and S.

Q1:

```
Answer(a,b) <- R(a,c) AND S(d,b)
```

Q2:

```
Answer(a,b) <- R(a,c) AND S(c,b)
```

a) Q1 and Q2 produce the same answer.
b) The answer to Q1 is always contained in the answer to Q2.
c) The answer to Q2 is always contained in the answer to Q1.
d) Q1 and Q2 produce different answers.

Question 7. Suppose relation R(A,B,C) has the tuples

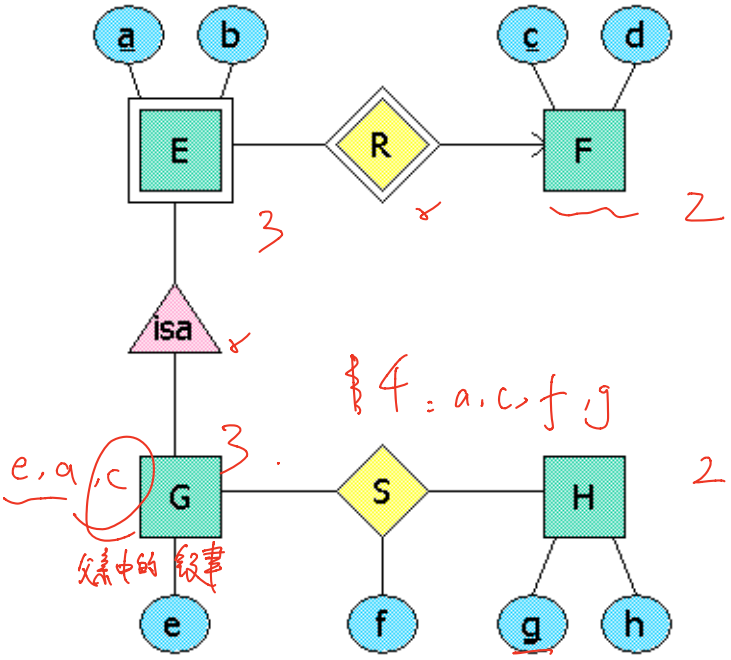
A	B	C
1	2	3
1	2	3
3	2	1

Using **bag** projection and theta-join, how many tuples appear in the result of

$\Pi_{A,B}(R) \bowtie_{R.B<S.B} \rho_{S(A,B)}(\Pi_{B,C}(R))$?

a) 2
b) 4
c) 6
d) 9

Question 8. Convert the E/R diagram below to a database schema, using the "E/R" method.



What is the sum of the number of attributes in each of the relation schemas?

a) 14
b) 15
c) 16
d) 17

The following three questions are based on the XML "Vehicles" document below.

```
<Vehicles>
   <Car manf="Hyundai">
      <Model>Azera</Model>
      <HorsePower>240</HorsePower>
   </Car>
   <Car manf="Toyota">
      <Model>Camry</Model>
      <HorsePower>240</HorsePower>
   </Car>
   <Truck manf="Toyota">
      <Model>Tundra</Model>
      <HorsePower>240</HorsePower>
   </Truck>
   <Car manf="Hyundai">
      <Model>Elantra</Model>
      <HorsePower>120</HorsePower>
   </Car>
   <Car manf="Toyota">
      <Model>Prius</Model>
      <HorsePower>120</HorsePower>
   </Car>
</Vehicles>
```

Question 9. Which of the following XPath expressions, when applied to the "Vehicles" document, does **NOT** produce a sequence of exactly three items?

a) /Vehicles/Car[@manf="Toyota"]/HorsePower
b) /Vehicles/*[HorsePower>200]/Model
c) //*[@manf="Toyota"]/@manf
d) None of the above (i.e., they all produce exactly 3 items)

Question 10. In a DTD that the "Vehicles" document satisfies, which of the following element declarations would you definitely **NOT** find?

a) <!ELEMENT Vehicles (Car*, Truck+, Car*)>
b) <!ELEMENT Vehicles (Car+, Truck*, Car)>
c) <!ELEMENT Vehicles ((Car|Truck)*)>
d) <!ELEMENT Vehicles (Car*, Truck*, Car*, Truck*, Car*)>

Question 11. Assuming the "Vehicles" document is in a file vehicles.xml, and we execute the XQuery query

```
for $c in doc("vehicles.xml")/Vehicles/Car
return <Auto><Make>$c/@manf</Make><Model>$c/Model</Model></Auto>
```

how many times will the string "Toyota" appear in the returned result?

a) 0
b) 1
c) 2
d) 3

△ 在 XQuery 中，= 比较两边若是多个项的序列，则任意一个项为 true 就为 true.

可用：eq. ne. lt. gt. le. ge 代替，这些只能用于单项组成序列，若为多项，比较失败。

△ XQuery 中可用：distinct-values 消除重复，但是这只用于：文本-字符串元素。

并将去除、标签，且不会将其添加回去!!

△ XQuery 中量词：

{ every variable in exp1 satisfies exp2
  some variable in exp1 satisfies exp2

Question 12. Consider relation 'Grades' and the query given below:

student	cs145_grade	seminar_grade
A	45	NULL
B	NULL	90
C	100	80

```
SELECT student
FROM Grades
WHERE (cs145_grade>seminar_grade AND seminar_grade>75 AND cs145_grade>90) OR
(cs145_grade<50)
```

Which students' tuples are returned?

a) B and C only.
b) A and C only.
c) A only.
d) A, B, and C.

Question 13. Which of the following relations is a counterexample to show why the rule "if A ->-> BC, then A ->-> B" does NOT hold?

a)

A	B	C
1	11	21
1	12	22

b)

A	B	C
1	11	21
1	12	22
1	11	22
1	12	21

c)

A	B	C	D
1	11	21	31
1	12	22	32

d)

A	B	C	D
1	11	21	31
1	12	22	32
1	11	22	32
1	12	21	31

Question 14. Let R(A, B, C) satisfy the following functional dependencies (FDs): AB -> C, BC -> A, and

AC -> B. The closure of A (i.e., A^+) is (a)

a) A
b) AB
c) AC
d) ABC

$A^+ =$ AB → ABC
BC → ABC
AC → ABC

Question 15. Which of the following statements are correct? (c)

I. All relations in 3NF are also in BCNF. ✗
II. All relations with only two attributes are in BCNF. ✓
III. For any relation schema, there is a dependency-preserving decomposition into 3NF.

a) I only
b) III only
c) II and III ✓
d) I and III

Question 16. For which of the following normal forms is there always a lossless-join decomposition for any relation schema? (d)

a) BCNF
b) 3NF
c) 4NF
d) All of the above ✓

Question 17. Consider a database containing two relations

Borrower(customer-name, loan-number)
Loan(loan-number, amount)

Borr:

cus	loan
Joh	null
Jo	1209

Loan:

loan	amount
null	1990
1209	1900

We define a view loan-info as

```
CREATE VIEW loan-info as
    SELECT customer-name,amount
    FROM  Borrower, Loan
     WHERE Borrower.loan-number=Loan.loan-number
```

Consider the following insertions

I. INSERT INTO *Borrower* VALUES ('Johnson', null)
INSERT INTO *Loan* VALUES (null,1900)

II. INSERT INTO *Borrower* VALUES ('Johnson', 1209)
INSERT INTO *Loan* VALUES (1209,1900)

Which of the above operations will have the effect of inserting tuple ("Johnson",1900) into *loan-info* (assuming it is not there previously)? (b)

a) I only
b) II only ✓
c) I and II
d) None of the above

NULL can't join with others.
两个 NULL 不能比较

Question 18. Which of the following statements about E/R models is/are correct?

I. Many-to-many relationships cannot be represented in E/R-diagrams
II. Relationship sets can have attributes of their own.
III. All many-to-one relationships are represented by a relationship between a weak and a non-weak entity set.

a) II only.
b) III only.
c) II and III only.
d) I and II only.

Question 19. Consider the following XML DTD:

```
<!DOCTYPE A [
  <!ELEMENT A (C, B+) | (B*, C)+>
  <!ELEMENT B (C*)>
  <!ELEMENT C (#PCDATA)>
]>
```

Which one of the following is the ***smallest*** (i.e. fewest number of elements) XML that conforms to (is valid for) this DTD?

(a) <A><B><C></C></B></A>
(b) <A><C></C><B></B></A>
(c) <A><B></B></A>
(d) <A><C></C></A>

Question 20. We have the following dealership data in our Dealership_table:

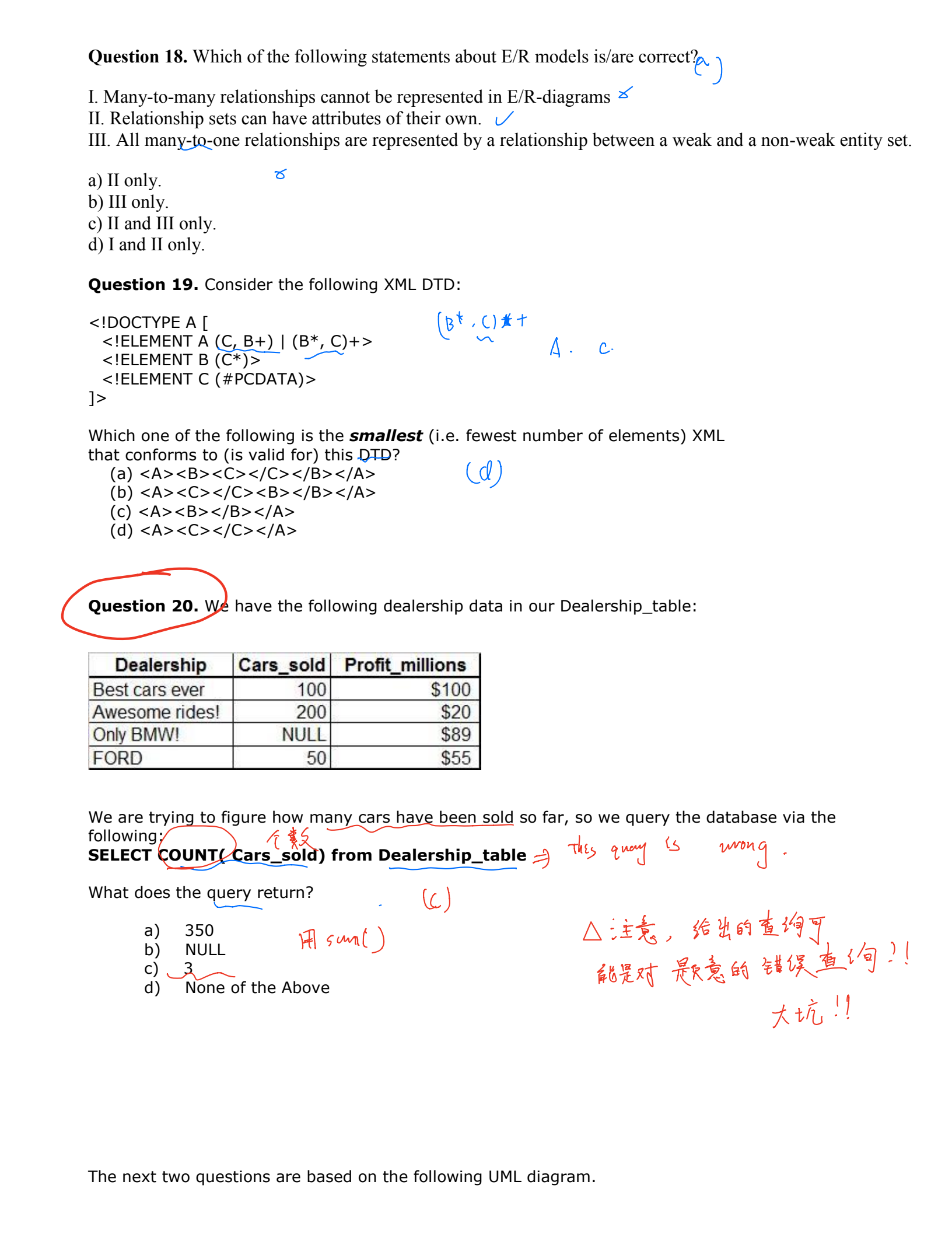
Dealership	Cars_sold	Profit_millions
Best cars ever	100	$100
Awesome rides!	200	$20
Only BMW!	NULL	$89
FORD	50	$55

We are trying to figure how many cars have been sold so far, so we query the database via the following:

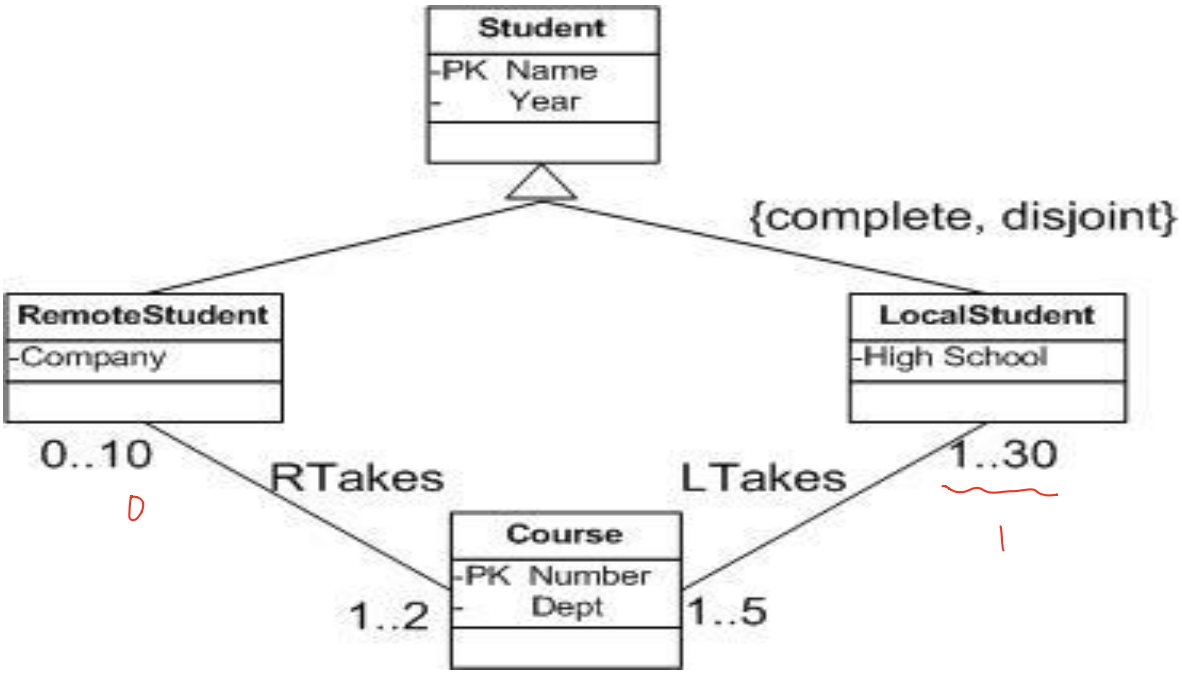
SELECT COUNT(Cars_sold) from Dealership_table

What does the query return?

a) 350
b) NULL
c) 3
d) None of the Above



The next two questions are based on the following UML diagram.



Question 21. What are the minimum and maximum number of Students in a Course?

(a) min = 1, max = 30
(b) min = 1, max = 40
(c) min = 0, max = 30
(d) min = 0, max = 40

Question 22. We have discussed three schemes in class for translating the subclass relation between RemoteStudent, LocalStudent, and Student into a relational schema. Which of the following is **not** the result of applying one of the three translating schemes to the student hierarchy of the above UML diagram?

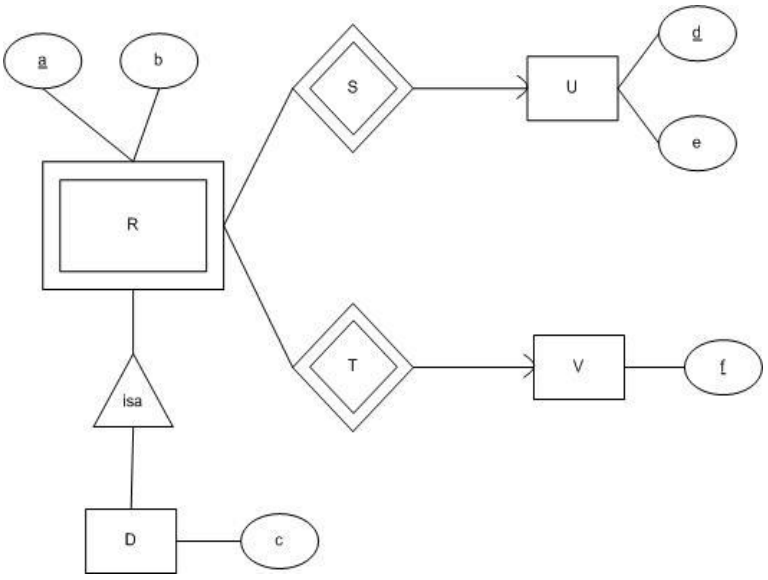
(a) Student(name, year, company, highschool)

(b) Student(name, year)
RemoteStudent(name, company)
LocalStudent(name, highschool)

(c) Student(name)
RemoteStudent(year, company)
LocalStudent(year, highschool)

(d) RemoteStudent(name, year, company)
LocalStudent(name, year, highschool)

Question 23. Consider the following E/R model:

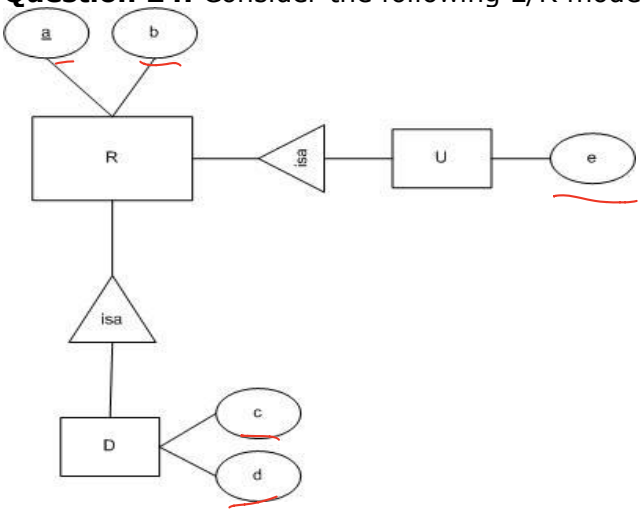


Which of the following can be found in both the E/R and OO translations of the above model to relational schema?

I. R(a, d, f, b)
II. S(a, d)
III. D(a, c)

a) I only
b) I, II
c) II, III
d) I, II, III

Question 24. Consider the following E/R model



How many attributes are there in the relation that represents the entities of D, for the E/R, OO, and Nulls approaches?

a) 2, 3, 4
b) 2, 4, 5
c) 3, 4, 4
d) 3, 4, 5

Question 25. Consider the following table Xbox_Games(name, price) and assume that these values already exist in the database ('ok_game', 40), ('good_game', 50), ('AWESOME_game', 60). We have the following two transactions:

```
T1: BEGIN TRANSACTION
        S1: UPDATE  Xbox_Games SET price=22 WHERE name='ok_game'
        S2: INSERT INTO Xbox_Games VALUES ('BAD_Game', 0)
        S3: UPDATE Xbox_Games SET price=38 WHERE name='ok_game'
    COMMIT;

T2: BEGIN TRANSACTION
        SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
        S4: SELECT AVG(price) AS average_price FROM Xbox_Games
    COMMIT;
```

Above two transactions are hitting the DBMS roughly at the same time. What are the possible values for average_price?

I. 50
II. 44
III. 37

a) I only.
b) II only.
c) I & II only.
d) I & III only.

Question 26. When the following code is executed, what shows up on the screen (i.e., what is printed by the Echo statement)?

```
<?php
    $message="your tuition has gone up by $300";
    $100=200;
    $message='your tuition has gone up by $100';
   Echo $message;
?>
```

a) "your tuition has gone up by $100"
b) "your tuition has gone up by 200"
c) "your tuition has gone up by $300"
d) "your tuition has gone up by "

Question 27. Suppose we have the following:
CREATE TABLE Beers (

```
        name      CHAR(20) PRIMARY KEY,
        origin    CHAR(120)
);
CREATE TABLE Sells (
        bar       CHAR(20),
        beer      CHAR(20),
        price     REAL,
        FOREIGN KEY(beer)
                  REFERENCES Beers(name)
                  ON DELETE SET NULL,
                  ON UPDATE CASCADE
);
```

Assume the following data already exists in our database:

Beers Table:

Name	Origin
B1	O1
B2	O2
B3	O3

Sells Table:

Bar	Beer	Price
Bar1	B1	2
Bar1	B1	1
Bar1	B2	3
Bar2	B3	4
Bar2	B3	6
Bar3	B1	4

We execute the following two queries:

```
DELETE FROM Beers where name='B1';
UPDATE Beers set Name='B4' Where Name='B3';
```

Now, what does this query return?

```
SELECT SUM(Price) FROM Sells WHERE Sells.Beer IS NOT NULL;
```

a) 3
b) 10
c) 13
d) 20

Question 28. We have the following Table: *Cars(name, price)* and the following assertion:

```
CREATE ASSERTION Check_My_Cars CHECK
    (NOT EXISTS ((SELECT * FROM Cars) EXCEPT
        (SELECT * FROM Cars WHERE name = name OR price = price)))
```

What does above assertion check?

a) Table "Cars" is not Empty.
b) Make sure that no tuple of Cars has both name and price NULL.
c) Make sure that there are no NULL's in any tuple of Cars.
d) Check that all tuples in Cars agree in the value of either the name or the price.

Question 29. We have the relations Time_Table(a, b) and also the following trigger exists (Why? Maybe because it's a good exam problem!) :

```
CREATE TRIGGER T
AFTER INSERT ON Time_Table
REFERENCING NEW ROW AS NNN
FOR EACH ROW
WHEN(NNN.a * NNN.b > 10)
INSERT INTO Time_Table VALUES(NNN.a - 1, NNN.b + 1);
```

Which one of the tuples below, inserted into an empty Times_table, would **NOT** result in Time_Table containing exactly 3 tuples?

a) (3,5)
b) (4,3)
c) (3,4)
d) (3,8)

Question 30. We have the following XML data:

```
<list>
  <cart_id>1000</cart_id>
  <cart_id>1560</cart_id>
  <cart_id>99999</cart_id>
  <cart_id>88888</cart_id>
</list>
```

Which one of the SimpleType definitions below, **DOES NOT** fully contain the Cart_Id's above?

a)

```
<xs:simpleType name="simple">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1000"/>
    <xs:maxInclusive value="99999"/>
  </xs:restriction>
</xs:simpleType>
```

b)

```
<xs:simpleType name="simple">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="999"/>
    <xs:maxInclusive value="100000"/>
  </xs:restriction>
</xs:simpleType>
```

c)

```
<xs:simpleType name="simple">
  <xs:restriction base="xs:int">
    <xs:minExclusive value="999"/>
    <xs:maxExclusive value="100000"/>
  </xs:restriction>
</xs:simpleType>
```

d)

```
<xs:simpleType name="simple">
  <xs:restriction base="xs:int">
    <xs:minExclusive value="1000"/>
    <xs:maxExclusive value="99999"/>
  </xs:restriction>
</xs:simpleType>
```

Question 31. Consider the following query on the relation R (A, B, C, D)

SELECT [ ]

FROM R
GROUP BY A, B

Assume A,B,C, D take integer values

Which of the following can appear in the position marked as [...]

I. MIN(C+D)
II. A,B
III. C,D

a) II only.
b) I and II only.
c) I, II, and III.
d) None.

Question 32. Consider relation R(A,B,C,D,E) with FDs

A-> B, AB->CD, D->ABCE

Which of the following are keys of the relation R

I A
II AB
III CD

a) I only.
b) I and II only.
c) II, and III only.
d) I, II, and III.

Question 33. Consider the ODL specification given below:

```
class X
{
 relationship Set<Y> R inverse Y::R;
 relationship Z T inverse Z::T;
};
class Y
{
 relationship Set<X> R inverse X::R;
 relationship Z S inverse Z::S;
};
class Z
{
 relationship Set<Y> S inverse Y::S;
 relationship X T inverse X::T;
 relationship Z U inverse Z::U;
};
```

Which of the following is true?

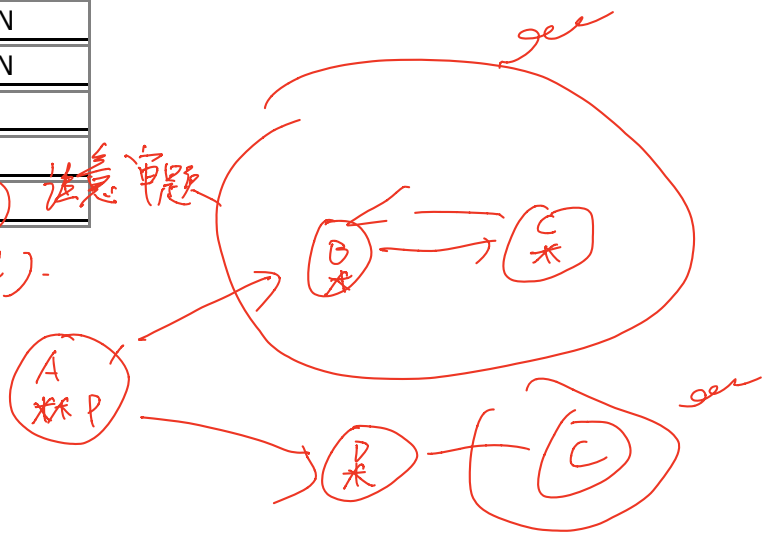
a) R and S are many-1 relationships, but not 1-1.
b) T is a many-1 relationship, but not 1-1.
c) U is a 1-1 relationship and S is a many-1 relationship, but not 1-1.
d) R and T are many-many relationships, but not many-1.

Question 34. Suppose we are told that R(A, B, C, D) is in BCNF, and that three out of the four FDs (a)-(d) listed below hold for R. Choose the FD that R doesn't satisfy.

a) A -> BCD
b) BC -> A
c) CD -> B
d) D -> C

Question 35. Assume A is the owner of the relation to which privilege P refers. After the following steps…

Step	By	Action
1	A	GRANT P TO B WITH GRANT OPTION
2	A	GRANT P TO D WITH GRANT OPTION
3	B	GRANT P TO C WITH GRANT OPTION
4	C	GRANT P TO B WITH GRANT OPTION
5	D	GRANT P TO C
6	A	REVOKE P FROM B CASCADE
7	A	REVOKE P FROM D RESTRICT



…which user(s), besides A, have privilege P?

a) C only.
b) D only.
c) C and D only.
d) B, C, and D.