## CS145 Final Examination

Thursday, December 12, 2002, 12:15 – 3:15PM

### Directions

The exam is *open book/notes*; any written materials may be used.

For each of the 35 questions, circle the letter (a), (b), (c), or (d) of your chosen answer. Do not circle more than one answer. If you wish to change your answer, please indicate clearly what your "final answer" is.

Score = 3 times number-right minus number-wrong, so random guessing nets you nothing on the average, and 105 is a perfect score.

If you wish to explain or demonstrate your solution to a problem for partial credit, you may use page bottoms or the backs of the pages (but warn us on the front). Please use this option sparingly, e.g., if you think the question is flawed or open to multiple interpretations, because we shall only be awarding partial credit in rare situations.

You have about 5 minutes per question. Use your time wisely, and do not spend too much time on any one question.

Do not forget to **sign the pledge** below.

I acknowledge and accept the honor code.

______________________________

Print your name here: ________________________________________

In each of the following 15 questions, you are asked to compare two queries $Q_1$ and $Q_2$. 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 $Q_1$ produces more of some particular tuple, and other databases where $Q_2$ 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, $Q_1$ is contained in $Q_2$ if on every database, $Q_2$ produces at least as many copies of each tuple as $Q_1$ does. Note that it is possible $Q_2$ produces one or more copies of a tuple, while $Q_1$ produces none of that tuple.

General advice:

- Do not assume a query has a trivial syntactic error and therefore produces nothing.
- Relations mentioned in the queries may have attributes not mentioned, but their existence should not affect the answer.
- SQL relations may have `NULL`'s, although in other languages, you should assume no `NULL`'s unless stated otherwise.
- SQL queries should be assumed to be in standard SQL unless stated otherwise.
- In SQL and OQL, it is possible that there may be duplicate tuples, but in relational algebra or Datalog assume the relations are sets unless stated otherwise.

---

**Question 1:** In the following expressions of relational algebra, the relations are $R(a, b)$ and $S(b, c)$.

$Q_1$: $\pi_{a,b}(R \bowtie S)$

$Q_2$: $R \cap \Big(\rho_{T(a)}\big(\pi_c(S)\big) \times \pi_b(S)\Big)$

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 2:**

$Q_1$:
```
SELECT a
FROM R
WHERE b >= ALL(SELECT d FROM S WHERE c=5);
```

$Q_2$:
```
SELECT a
FROM R
WHERE b >= ANY(SELECT d FROM S WHERE c=5);
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 3:**

$Q_1$:
```
SELECT DISTINCT a
FROM R
WHERE b > 0;
```

$Q_2$:
```
SELECT a
FROM R
WHERE b > 0
GROUP BY a;
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

The following two questions are based on the ODL description below, which represents sports teams and the players who play for them. *Note*: Question 24 also deals with this schema. You may wish to work it between questions 5 and 6.

```
class Team (extent Teams key name) {
    attribute string name;
    relationship Set<Player> players inverse Player::playsFor;
};

class Player (extent Players key name) {
    attribute string name;
    relationship Team playsFor inverse Team::players;
};
```

**Question 4:**

$Q_1$:
```
SELECT p.name
FROM Teams t, t.players p
WHERE t.name = "49ers";
```

$Q_2$:
```
SELECT p.name
FROM Players p
WHERE p.playsFor.name = "49ers";
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 5:**

$Q_1$:
```
SELECT q.name
FROM Players p, p.playsFor t, t.players q
WHERE p.name = "Garcia";
```

$Q_2$:
```
SELECT t.players
FROM Teams t
WHERE EXISTS p IN t.players : p.name = "Garcia";
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 6:** In the following Datalog programs, the output is the relation `Cousin` in each case.

$Q_1$:
```
Sibling(x,y) <- Par(x,p) AND Par(y,p) AND x ≠ y
Cousin(x,y) <- Par(x,xp) AND Par(y,yp) AND Sibling(xp,yp)
```

$Q_2$:
```
GrandParent(x,y) <- Par(x,z) AND Par(z,y)
Cousin(x,y) <- GrandParent(x,g) AND GrandParent(y,g) AND x ≠ y
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 7:** The following SQL queries concern a relation $Arc(x, y)$, which you may think of as representing the fact that there is an arc from node $x$ to node $y$ in a certain directed graph. You should not assume that $x$, $y$, or even $x$ and $y$ together, is a key for $Arc$ (i.e., "parallel" arcs are possible). You may, however, assume that there are no `NULL`'s in the $Arc$ relation.

$Q_1$:
```
CREATE VIEW V AS
    SELECT x, COUNT(y) AS ct
    FROM Arc
    GROUP BY x;

SELECT SUM(ct)
FROM Arc, V
WHERE Arc.x = 0 AND Arc.y = V.x;
```

$Q_2$:
```
SELECT COUNT(a2.y)
FROM Arc a1, Arc a2
WHERE a1.x = 0 AND a1.y = a2.x;
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 8:**

$Q_1$:
```
SELECT COUNT(DISTINCT b) FROM R;
```

$Q_2$:
```
SELECT COUNT(b) FROM R;
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 9:** In the following, the results of $Q_1$ and $Q_2$ should be taken to be the result of the final `SELECT * FROM R`. Assume that the schema of relation $R$ is $R(a, b)$.

$Q_1$:
```
UPDATE R SET b = 10 WHERE a = 20;
SELECT * FROM R;
```

$Q_2$:
```
DELETE FROM R WHERE a = 20;
INSERT INTO R VALUES(20,10);
SELECT * FROM R;
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 10:** In the following, assume $a$ is an attribute of some character-string type, e.g. `CHAR(10)`, and that it may be `NULL`.

$Q_1$:
```
SELECT * FROM R WHERE a IS NULL;
```

$Q_2$:
```
SELECT * FROM R WHERE a NOT LIKE '%';
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 11:** In the following, you may assume relations $R(a, b)$ and $S(b, c)$ have no `NULL`'s, but may have duplicates.

$Q_1$:
```
SELECT R.a FROM R, S
WHERE R.b = S.b;
```

$Q_2$:
```
SELECT R.a FROM R
WHERE R.b IN (
    SELECT S.b FROM S;
);
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 12:** The following are two extended-relational-algebra compuations about a relation $R(x, y, value)$, using the notation of sequences of assignments. In each case, the result is the set of tuples in the relation $Answer$. *Hint*: It may help to think of $R$ as a map in the $x$-$y$ plane, with $value$ the height of an $x$-$y$ point.

$Q_1$: $R1(x1, y1, v1, x2, y2, v2) := R \times R;$
$R2(x1, y1, v1) := \pi_{x1,y1,v1}\big(\sigma_{(x1=x2\ OR\ y1=y2)\ AND\ v1<v2}(R1)\big);$
$R3(x1, y1, v1) := \rho_{R(x1,y1,v1)}(R);$
$Answer(v1) := \pi_{v1}(R3 - R2);$

$Q_2$: $R1(v) := \pi_v\big(\gamma_{x,MAX(value)\rightarrow v}(R)\big);$
$R2(v) := \pi_v\big(\gamma_{y,MAX(value)\rightarrow v}(R)\big);$
$Answer(v) := R1 \cup R2;$

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

**Question 13:** In the following expressions of relational algebra, the relation $R$ has schema $R(a,b)$.

$Q_1$: $\pi_a(R) \times \pi_b(R)$

$Q_2$: $\pi_{a,d}\big(R \underset{R.b \neq S.c}{\bowtie} \rho_{S(c,d)}(R)\big)$

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 14:** In the following, the queries $Q_1$ and $Q_2$ are actually DTD's, and their "answers" are the sets of documents that match them.

$Q_1$:
```
<!DOCTYPE a [
    <!ELEMENT A(B*, C+)>
    <!ELEMENT B(#PCDATA)>
    <!ELEMENT C(#PCDATA)>
]>
```

$Q_2$:
```
<!DOCTYPE a [
    <!ELEMENT A((B, A)|C+)>
    <!ELEMENT B(#PCDATA)>
    <!ELEMENT C(#PCDATA)>
]>
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

---

**Question 15:** In the following SQL queries, the key of $R(a,b,c)$ is $c$.

$Q_1$:
```
SELECT a, MAX(b) FROM R
GROUP BY a;
```

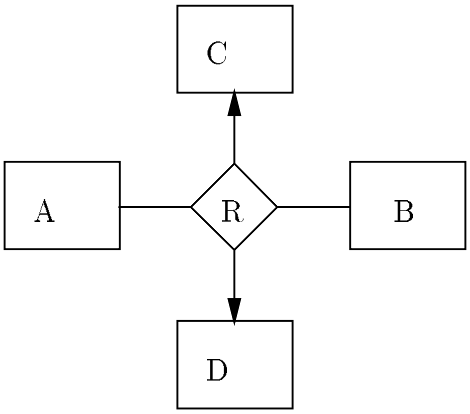
$Q_2$:
```
SELECT a, b FROM R r1
WHERE b >= ALL (SELECT b FROM R r2 WHERE r1.a = r2.a);
```

(a) $Q_1$ and $Q_2$ produce the same answer.
(b) The answer to $Q_1$ is always contained in the answer to $Q_2$.
(c) The answer to $Q_2$ is always contained in the answer to $Q_1$.
(d) $Q_1$ and $Q_2$ produce different answers.

*** In the remainder of the exam, answers have no particular form. ***

---

**Question 16:** Consider the following E/R diagram:




Below are three possible relationship sets for this E/R diagram:

| | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| I. | $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| | $a_1$ | $b_1$ | $c_1$ | $d_2$ |

| | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| II. | $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| | $a_1$ | $b_1$ | $c_2$ | $d_2$ |

| | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| III. | $a_1$ | $b_1$ | $c_1$ | $d_1$ |
| | $a_1$ | $b_2$ | $c_1$ | $d_1$ |

You may assume that different symbols stand for different values, e.g., $d_1$ is definitely not equal to $d_2$. Which of the above could *not* be the relationship set for the E/R diagram?

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

---

**Question 17:** Suppose we have a relation $R(A, B, C, D, E)$ and the FD's $A \to D$, $B \to C$, $D \to E$, and $CE \to B$. If we project $R$ (and therefore its FD's) onto schema $ABC$, what is true about the key(s) for $ABC$?

(a) Only $A$ is a key (b) Only $AB$ is a key (c) Only $AB$ and $AC$ are keys. (d) Only $AB$, $AC$, and $BC$ are keys

**Question 18:** Consider relation $R(A, B, C, D, E)$ with dependencies $A \twoheadrightarrow BC$ and $D \rightarrow B$. Of the following dependencies:

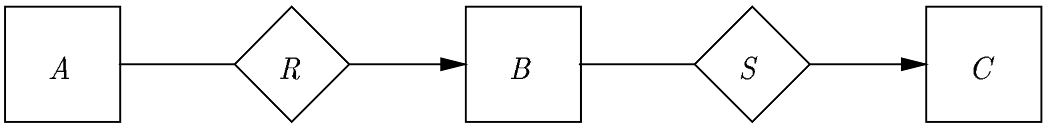
I. $A \rightarrow B$
II. $A \twoheadrightarrow C$
III. $A \twoheadrightarrow D$

which must *necessarily* (i.e., for all instances of $R$ that satisfy $A \twoheadrightarrow BC$ and $D \rightarrow B$) hold in $R$?

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

---

**Question 19:** Below is an E/R diagram:




Indicate which of the ODL specifications best mimics the intent of this E/R diagram. In both E/R and ODL, we have omitted mention of all attributes, which you may thus ignore.

(a)
```
class A {relationship Set<B> R inverse B::R;};
class B {relationship Set<A> R inverse A::R;
         relationship Set<C> S inverse C::S;};
class C {relationship Set<B> S inverse B::S;};
```

(b)
```
class A {relationship B R inverse B::R;};
class B {relationship Set<A> R inverse A::R;
         relationship C S inverse C::S;};
class C {relationship Set<B> S inverse B::S;};
```

(c)
```
class A {relationship Set<B> R inverse B::R;};
class B {relationship A R inverse A::R;
         relationship Set<C> S inverse C::S;};
class C {relationship B S inverse B::S;};
```

(d)
```
class A {relationship Set<B> R inverse B::R;};
class B {relationship Set<A> R inverse A::R;
         relationship Set<C> S inverse C::S;};
class C {relationship B S inverse B::S;};
```

**Question 20:** If $\cup$, $\cap$, and $-$ are given their *bag* interpretations, which of the following laws holds?

(a) $R \cup (S - T) = (R \cup S) - T$
(b) $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$
(c) $R \cap (S \cup T) = (R \cap S) \cup (R \cap T)$
(d) None of the above

---

**Question 21:** Suppose we have a relation $R(A, B, C, D, E)$ with functional dependencies $A \to B$, $B \to C$, and $BD \to E$. Which of the following could be the result of the decomposition of $R$ into BNCF?

(a) $AB$, $BC$, $BDE$
(b) $AD$, $BDE$, $ABC$
(c) $ABD$, $BE$, $CE$
(d) $AB$, $ACD$, $BCE$

---

**Question 22:** Suppose we make the following declarations in the Oracle system we have been using in class:

```
CREATE TYPE T AS OBJECT (a INT, b INT);
/
CREATE TYPE T2 AS OBJECT (c INT, d T);
/
CREATE TABLE R AS T2;
```

Here are three possible queries we could ask about the relation $R$:

```
I.   SELECT rr.d.a FROM R rr;
II.  SELECT d.a FROM R;
III. SELECT a FROM THE(SELECT d FROM R);
```

Which of the above queries will **not** cause an error?
(a) Only I (b) Only III (c) Only I and II (d) None; i.e., they are all erroneous

---

**Question 23:** There are ten entity sets $E_1, E_2, \ldots, E_{10}$ arranged in an ISA hierarchy, with $E_{10}$ $ISA$ $E_9$ $ISA$ $E_8$ $ISA \cdots ISA$ $E_1$. That is, each $E_i$ is a subclass of $E_{i-1}$ for $i = 2, 3, \ldots, 10$. Given that entity $e$ is in $E_1$, how many different subsets of $\{E_1, E_2, \ldots, E_{10}\}$ could be **exactly** the set of entity sets in which $e$ appears?
(a) 9 (b) 10 (c) 511 (d) 1023

**Question 24:** The following is the same player-team ODL declaration we saw in Questions 4 and 5.

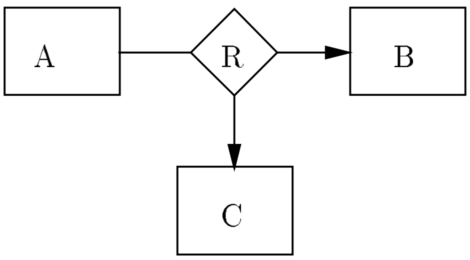
```
class Team (extent Teams key name) {
    attribute string name;
    relationship Set<Player> players inverse Player::playsFor;
};

class Player (extent Players key name) {
    attribute string name;
    relationship Team playsFor inverse Team::players;
};
```

Which of the following is a legal OQL query?

```
(a) SELECT p.playsFor.name
    FROM Players p
    WHERE p.name = "Bonds";

(b) SELECT p.name
    FROM Players p, p.playsFor t
    WHERE t.name = "Sharks";

(c) SELECT p.name
    FROM Teams t, t.players p
    WHERE t.players.playsFor = t;

(d) SELECT t.name
    FROM Teams t, t.players p
    WHERE t.players.name = p.name;
```

---

**Question 25:** Consider the following E/R diagram:




If $A$ has 50 entities, $B$ has 20 entities, and $C$ has 200 entities, what is the maximum number of triples of entities that could be in the relationship set for $R$?

(a) 50 (b) 100 (c) 1000 (d) 10,000

The next two questions refer to the relation `Emps(eName, salary, manager)`. You may assume that `eName` is a key, and that the tuple $(e, s, m)$ means that the employee with name $e$ earns salary $s$, and has manager $m$. The manager component $m$ is the name of an employee (assume the CEO manages himself/herself). Management is a hierarchy, so (other than for the CEO) there are no cycles in the graph whose nodes are the employees and whose arcs go from employee to manager. Here are three queries one could ask about this data:

I. Find, for each manager $m$, the employee(s) of $m$ that earn the highest salary. That is, we want all pairs of the form $(m, e)$ such that no employee managed by $m$ has a higher salary than $e$ does.

II. Find for each manager $m$ the employees that report indirectly to $m$. That is, we want all pairs of the form $(m, e)$ such that either there is an employee $f$ such that $m$ manages $f$ and $f$ manages $e$, or there are two employees $f$ and $g$ such that $m$ manages $f$, who manages $g$, who manages $e$, and so on.

III. Find the manager of the manager of the manager of each employee $e$. That is, we want all pairs of the form $(m, e)$ such that there are employees $f$ and $g$ such that $m$ manages $f$ who manages $g$, who manages $e$.

**Question 26:** Which of the above queries can be written in Datalog?
(a) III only (b) I and II only (c) I and III only (d) I, II, and III

**Question 27:** Which of the above queries can be written in the core relational algebra (union, intersection, difference, select, project, and product)?
(a) none (b) III only (c) I and III only (d) I, II, and III

---

**Question 28:** Suppose relation `Beers` is declared by:

```
CREATE TABLE Beers (
    name CHAR(30),
    manf CHAR(30) REFERENCES Manfs(name)
);
```

In order to execute the statement:

```
INSERT INTO BEERS VALUES('Bud', 'Anheuser Busch');
```

which of the following privileges is (among those) needed?
(a) `SELECT ON Beers`
(b) `REFERENCES ON Manfs`
(c) `SELECT ON Manfs`
(d) `UPDATE ON Beers`

**Question 29:** Initially, user $A$ is the owner of relation $R$, and no other user holds privileges on $R$. The following are executed:

by $A$: `GRANT INSERT ON R TO B WITH GRANT OPTION;`
by $B$: `GRANT INSERT ON R TO C WITH GRANT OPTION;`
by $C$: `GRANT INSERT ON R TO D WITH GRANT OPTION;`
by $D$: `GRANT INSERT ON R TO B WITH GRANT OPTION;`
by $B$: `REVOKE INSERT ON R FROM C CASCADE;`

As a result, which is the exact set of users that have the privilege `INSERT ON R`?
(a) $A$ only (b) $A$ and $B$ only (c) $A$, $B$, and $D$ only (d) $A$, $B$, $C$, and $D$

---

**Question 30:** Suppose $R$ is declared

```
CREATE TABLE R(
    a INT,
    b INT
)
```

and currently contains only the tuple $(1, 1)$. We have just executed the JDBC statement

```
PreparedStatement myStat = myCon.createStatement(
    "UPDATE R SET a = ? WHERE b = 1");
```

A suitable next step, if we wish eventually to replace the original tuple $(1, 1)$ in $R$ by $(2, 1)$ would be:
(a) `myStat.setInt(1,2);`
(b) `myStat.setInt(2,1);`
(c) `myStat.executeUpdate(2,1);`
(d) `myStat.executeUpdate(2);`

---

**Question 31:** Which of the following queries *might* produce different results when evaluated using set semantics and bag semantics, even if the relations $R$ and $S$ are themeselves sets (i.e., they contain no duplicate tuples)?
(a) $R - S$ (b) $(R \bowtie S) - (S \bowtie R)$ (c) $R \cup S$ (d) $\sigma_{a=5}(R)$

---

**Question 32:** In SQL 3-valued logic, what is the value of the following expression: `(X < Y OR X >= Y OR Z = 3)`?
(a) `TRUE` or `FALSE`, but never `UNKNOWN`
(b) `FALSE` or `UNKNOWN`, but never `TRUE`
(c) `TRUE` or `UNKNOWN`, but never `FALSE`
(d) `TRUE`, `FALSE`, or `UNKNOWN`

**Question 33:** Consider the following Datalog program with two EDB predicates, `Red` and `Blue`:

```
Answer(x,y) <- P(x,y) AND NOT Q(y,x)
P(x,y) <- Red(x,y)
P(x,y) <- P(x,z) AND Red(z,y)
Q(x,y) <- Blue(x,y)
Q(x,y) <- Blue(x,z) AND Q(z,y)
```

Suppose the EDB contains the tuples $Red(1,2)$, $Red(2,3)$, $Red(3,4)$, $Blue(2,4)$, and $Blue(3,1)$. Then which of the following tuples is **not** in the relation corresponding to the predicate `Answer`?

(a) (1,2) (b) (2,3) (c) (1,3) (d) (2,4)

---

**Question 34:** Which of the following Datalog rules is safe?

(a) `P(x,y) <- Q(y) AND x < y`
(b) `P(x,y) <- Q(y) AND R(z) AND x < y`
(c) `P(x,y) <- Q(x) AND R(y) AND NOT S(z)`
(d) `P(x,y) <- Q(x) AND R(y) AND x < y AND NOT S(y)`

---

**Question 35:** In SQL, the meaning of the symbol `||` is:

(a) Concatenation
(b) Execute in parallel
(c) Eager OR, that is, evaluate both arguments and produce the OR of the truth values of the two arguments
(d) Lazy OR, that is, evaluate the left argument; produce TRUE if the left argument is TRUE, and otherwise evaluate the right argument and produce its value as the result.