Solution

Problem 1: (c)

R INTERSECT R is R, with duplicates eliminated.

Problem 2: (b)

Q1 asks for the topics of classes that have any two students with different names. Q2 asks for the same, but produces the topic of the class once for every pair of students in the class that have different names.

Problem 3: (a)

In both queries, the condition in the if-clause asks whether the sequence $a/B is empty.

Problem 4: (a)

Q2 is the direct translation of Q1 from Datalog to recursive SQL. Note that Datalog produces sets, and the UNION operator in SQL also produces sets as a default.

Problem 5: (d)

There is no relationship between the tuples in S and the tuples referred to by R.d.

Problem 6: (c)

Q1 produces all pairs of a first component from some tuple r of R paired with the second component of some tuple s from from S. Q2 produces that pair if and only if r has a second component that agrees with the first component of s.

Problem 7: (c)

The projection of R onto AB is the three tuples {12, 12, 32}. The projection onto BC is {23, 23, 21}. After renaming, the join asks for pairs where the second component of tuples in the first group is less than the second component of the second group. Thus, each of the three tuples of the first group joins with the first two tuples of the second group, {23, 23}, but not with the third tuple, 21, of the second group. Thus, the result is {1223, 1223, 1223, 1223, 3223, 3223}.

Problem 8: (a)

The database schema is E(a,b,c), F(c,d), G(a,c,e), H(g,h), and S(a,c,f,g).

Problem 9: (a)

There are only two Cars (as opposed to Trucks) manufactured by Toyota. (b) produces the three Model elements for the Azera, Camry, and Tundra. (c) produces three copies of the string "Toyota," for the three Cars and Trucks with manf="Toyota".

Problem 10: (b)

The problem with (b) is that there can be only one Car element after the Truck elements, and the document has two Car elements after the Truck element.

Problem 11: (a)

Since there are no curly braces in the return-clause, the result has literal strings like `$c/@manf`. "Toyota" does not appear at all.

Problem 12: (b)

The first line of the where-clause produces student C, and the second line of the where-clause produces student A. Since both parts of the where-clause compare cs145_grade with a value, student B, with NULL in that component, cannot be returned.

Problem 13: (a)

The relation satisfies A->->BC, because it is a trivial dependency for any relation with schema ABC. It does not satisfy A->->B, because if we swap B's in the two tuples, we get tuples that are not in relation (a). (b) and (d) cannot be counterexamples, because they satisfy A->->B, and (c) cannot be a counterexample because it doesn't satisfy A->->BC.

Problem 14: (a)

Since there is no FD with only A on the left, we never get started when trying to compute A^+.

Problem 15: (c)

II was a challenge-problem exercise, and III is a consequence of Algorithm 3.26 (synthesis for 3NF). I does not hold, as discussed in Example 3.25.

Problem 16: (d)

Algorithm 3.20 is a lossless-join decomposition into BCNF (and thus into 3NF as well, although Algorithm 3.26 can also serve for that purpose. Algorithm 3.33 is a lossless-join decomposition into 4NF.

Problem 17: (b)

For II, the two tuples inserted join to give the desired tuple in the view. However, in I, the two tuples both have NULL in the join attributes, and so cannot join with any tuple. Neither affects the view at all.

Problem 18: (a)

I is false; a many-many relationship is represented by a diamond and no arrows. II is true. III is nonsense. There are many examples of many-one relationships that do not support a weak entity set, e.g., the "favorite-beer" relationship discussed in class.

Problem 19: (d)

An A can consist of one C followed by one or more B's, or zero or more B's and one C. The latter clearly leads to documents with the fewest elements, an A with one C-child and no B-children.

Problem 20: (c)

We may be "trying" to count cars, but the query only counts tuples. Since counting an attribute ignores tuples that have NULL in that component, the result of the query is 3.

Problem 21: (b)

A course has 0-10 remote students and 1-30 local students. Every student is exactly one of these. Thus, there must be at least one student, and there can be at most 40 students.

Problem 22: (c)

The problem with (c) is that the tuples for RemoteStudent and LocalStudent don't have a name, and thus names cannot be related to years or the other attributes that remote or local students have.

Problem 23: (a)

S(a,d) is never part of the database schema, because supporting relationships without attributes of their own are never part of the translation. D(a,c) is also part of neither translation, because the key for D is a, d, and f.

Problem 24: (d)

In the E/R approach, it's D(a,c,d); in the OO approach it's D(a,b,c,d). In the nulls approach, there is one relation for R and all its subclasses, which includes all attributes: R(a,b,c,d,e).

Problem 25: (d)

T2 must appear to be executed either completely before or completely after T1. If before, T2 produces the average of 40, 50, and 60 = 50. If after, T2 produces the average of 38, 50, 60, and 0 = 37.

Problem 26: (a)

Singly quoted strings in PHP are treated literally, and there is no substitution for apparent variable names.

Problem 27: (c)

First the delete statement causes all occurrences of B1 in Sells.Beer to be set to NULL. Then, the update statement causes the occurrences of B3 in Sells.Beer to be set to B4. Then, the query computes the sum of the Price values in the third, fourth, and fifth tuples of Sells --- those tuples where Beer is not NULL. This sum is 3+4+6 = 13.

Problem 28: (b)

The outer subquery must be empty, which means that every Cars tuple must satisfy name=name OR price=price. The only way that condition is *not* satisfied is if name and price are both NULL.

Problem 29: (c)

When we insert (3,4), the trigger causes the insertion of (2,5). Since 2*5 is not greater than 10, there is no third tuple inserted. On the other hand, the other choices to result in exactly three tuples. For example, inserting (3,5) triggers the insertion of (2,6), which triggers the insertion of (1,7), and nothing more.

Problem 30: (d)

Choice (d) says that values must be strictly bigger than 1000 and strictly less than 99999. Both conditions are violated, by the first and third Cart_ID elements, respectively. Even one violation would be enough to make (d) the correct answer.

Problem 31: (b)

Anything can appear in an aggregation, so I is OK. II is also OK, because both A and B are grouping attributes. But III is not OK, because C and D are neither aggregated nor grouping attributes.

Problem 32: (a)

A^+ = ABCDE, so A is a key. AB is not a key, since there is a proper subset (A) that is a key. Neither is CD a key. Even though CD^+ = ABCDE, there is a proper subset, D, that is a key.

Problem 33: (c)

R has sets at both ends, so it is many-many. T has sets at neither end, so it is 1-1. The same is true for U, which is self-inverse. Finally, S has a set at one end, but not the other, so it is many-1.

Problem 34: (d)

If the first three FD's, A->BCD, BC->A, and CD->B hold, then A, BC, and CD are all keys, so R is in BCNF. Moreover, the fourth FD, D->C must not hold, or else BCNF is violated. If A->BCD, BC->A, and D->C hold, then A and BD are keys. If so, CD->B would violate BCNF if it held. Similar reasoning rules out the other two possibilities.

Problem 35: (c)

The grants leave us with a grant diagram that looks like:

```
CP<---DP*<---AP**--->BP*<-->CP*
```

The double-arrow line between BP* and CP* represents arcs in both directions. The first Revoke severs the arc from AP** to BP*, which cuts off that node and CP* as well. The second Revoke has no effect, because D has granted P to C. Thus, AP** reaches DP* and CP, resulting in privilege P being held by A, C, and D.