Sample Midterm for CS 240A

Carlo Zaniolo

Computer Science Department
University of California at Los Angeles
zaniolo@cs.ucla.edu
Problem 1: 32 Points

The following database relation is given, where for each supplier, there is a list containing all parts sold by the supplier:

```
supplies(Sname, List_of_parts)
```

1. Write a Datalog program (with recursive rules) to count the number of parts each supplier supplies.

```
countparts(S, L, 1) ← supplies(S, [X|L]).
countparts(S, L, C1) ← countparts(S, [X|L]), C1 = C + 1.
finalcount(S, C) ← countparts(S, [], C).
?finalcount(S, C)
```
Problem 1: Cont.

1. Explain how a deductive database system will compile the recursive rules of your program: i.e., show the recursive method(s) chosen and rules rewritten according such method(s). This is straigth bottom up. Compilation will use differential fixpoint:

\[
\begin{align*}
\delta\text{countparts}(S, L, 1) & \leftarrow \text{supplies}(S, [X|L]). \\
\delta\text{countparts}(S, L, C1) & \leftarrow \delta\text{countparts}(S, [X|L]), C1 = C + 1. \\
\text{finalcount}(S, C) & \leftarrow \delta\text{countparts}(S, [], C).
\end{align*}
\]
Problem 1 in a top-down fashion

?finalcount(S, FC)
finalcount(S, FC) ← supplies(S, L), cnt(S, L, 0, FC).
cnt(S, [X|L], C, FC) ← C1 = C + 1, cnt(S, L, C1, FC).
cnt(S, [], C, C).

We have the following bindings:

?finalcount^ff
finalcount^ff ← supplies^ff, cnt^bbbf
cnt^bbbf ← C1 = C + 1, cnt^bbbf.
cnt(S, [], C, C).

Thus we have right-recursive rules.
Problem 2: \texttt{emp(Eno, Sal, Dno); dept(Dno, Dname)}

Where \texttt{emp.Eno} and \texttt{dept.Dno} are keys for their relations, and \texttt{emp.Dno} is a foreign key referencing \texttt{dept.Dno}

1. Write a DB2 rule that, whenever a department is deleted from the database, sets to 0000 the value of the \texttt{Dno} attribute for those tuples in relation \texttt{emp} that would violate the foreign-key constraint because of this deletion.

\texttt{CREATE TRIGGER dept-del}
\texttt{AFTER DELETE ON dept}
\texttt{FOR EACH ROW}
\texttt{UPDATE emp}
\texttt{SET Dno=0000}
\texttt{WHERE emp.Dno = old.Dno}
Problem 2: \texttt{emp(Eno, Sal, Dno); dept(Dno, Dname)}

2. Say that the database administrator decides that 0000 is no longer acceptable for undefined or non-existing departments, and 9999 should be used instead of 0000, from now on. Thus, the administrator performs a global update that changes all \texttt{emp} records where \texttt{Dno} = 0000 to 9999. But there remains the problem that (i) existing applications still perform updates in which the \texttt{Dno} of an employee is assigned the value of 0000, and (ii) your active rule of point 1, does the same.

Would you please help this administrator, an write a DB2 rule that solves the problem without requiring changes in existing applications or in the active rule above? Please, explain how your rule works.

Answer: Say that the old dummy department \texttt{Dno=0000} has been updated to \texttt{Dno=9999}. To be sure that we do not incur in errors due to
the foreign key constraint, we need to use the BEFORE semantics.

```
CREATE TRIGGER new-dept-upd
    BEFORE UPDATE of Dno ON emp
FOR EACH ROW
UPDATE emp
    WHEN old.Dno= 0000
SET Dno=9999
```
Problem 2. Cont.

3. Could the problem of point 2 be solved in a system that only supports deferred semantics for triggers? Explain the integrity-constraint-enforcement policy such a system will have to implement to make this possible.

Answer: The database system must delay checking and enforcing constraints such as foreign key constraints until the commit point.
Problem 3. 32 Points

The fact table in our data warehouse is as follows:

\[
\text{facts}(\text{Date}, \text{Place}, \text{Product}, \text{Retailer}, \text{Sales})
\]

This shows Sales in the four dimensions of Product, Date, Place and Retailer chain carrying the product.

1. Draw the complete lattice of cuboids for these four dimensions.

2. Write a DB2 SQL statement to compute the complete cube for sales in all four dimensions.
3. Assuming that sort-based aggregates are used, how many times will the data have to be sorted during the previous cube query?

4. There are $1.6 \times 10^9$ records in our relation. We only keep data on 12 retailers; since this number is rather low, we use a bitmap index without any compression. How many bits does this index require?