ECS 165A Database Systems — Fall 2013
Dr. Bertram Ludäscher — November 7, 2012

Assignment 4 – Referential Integrity, Triggers

Due: Friday, November 15th, 11:59pm (submission details via class list)

Problem 1: Referential Actions.

a) Assume you have two tables `A` and `B` where `B` has a foreign key `FK` which references the primary key `K` of `A`. Give an example for `A` and `B` where the use of `on delete cascade` would be appropriate, and a second example where it wouldn't be.

b) Create the schema from (a) in PostgreSQL, insert at least 3 values into each of the tables, then issue a delete statement on the parent table which results in cascaded deletions of tuples in the child table. Then do the same without a `delete cascade`. Document this by giving the corresponding SQL statements and the trace of a sample run.

c) Create two tables `C` and `D` which reference each other. What message do you get when you try to insert tuples into either table?

d) Solve the problem in (c) and document your solution with the output you get from PostgreSQL.[1]

e) Give an example instance of `C` and `D`, having at least 4 tuples in each of `C` and `D`, such that a deletion of a single tuple from C would result in cascaded deletions of all tuples in `D` and `C`. Try this in PostgreSQL and document and explain the output you obtain!

Problem 2: Datalog. Formulate the following queries in Datalog (cf. Assignment 2, Problem 1):

1. List all city information for cities with a population greater than 1 million.
2. What are `countryCode`s of countries that have cities with a population > 1 million?
3. List all African capital cities together with their city population and the country they are located in.
4. What countries have a border with another country?
5. What countries have at least two borders with another country?
6. What countries have no border with another country?
7. List all countries in Europe and Asia.
8. List all countries that are located in both Europe and Asia.
9. What countries are located on multiple continents?
10. Find the city (or cities) with the largest population.
11. What rivers flow in and out of the same lake?
12. What countries have all the religions that are in the `UDEF_RELIGION` relation?[2]

[1] This is a useful resource, even if written for Oracle: `http://infolab.stanford.edu/~ullman/fcdb/oracle/or-triggers.html`. In PostgreSQL you need an explicit `BEGIN TRANSACTION;` before the `INSERT` statements.

[2] This is a new table that holds the names of religions that a user is interested in.

Problem 3: Bill of Materials, Triggers (EXTRA CREDIT)

A bill of materials (BOM) describes a composite object in terms of its subparts, which themselves can be composed of further subparts etc. until the most basic parts are reached. Thus, a BOM can be seen as a tree in which each node represents a part, and in which children nodes represent subparts.

a) Create an SQL schema for a bill of materials:

`BOM(PartID INTEGER, SuperPartID INTEGER, Price INTEGER).`

Each part is identified by a part-id (an integer), has any number of subparts, and a price. The schema is: Populate your schema in PostgreSQL using the following information:

$$\texttt{subparts}(1) = \{2, 3\}, \texttt{subparts}(2) = \{4, 5\}, \texttt{subparts}(3) = \{6\}$$

The price `pr` of each part is as follows:

$$\texttt{pr}(1) = 100, \texttt{pr}(2) = 50, \texttt{pr}(3) = 30, \texttt{pr}(4) = 20, \texttt{pr}(5) = 10, \texttt{pr}(6) = 20$$

Submit SQL statements that create and populate the `BOM` table as shown above. Use `NULL` as the SuperPartID for part 1.

b) Assume you want to enforce the constraint that *any part can have at most 5* ***direct*** *subparts* (i.e., immediate child nodes). Specify the critical database operation(s) that can violate this constraint in a comment somewhere in your file like this:

```
-- Critical operation(s): OP1, OP2 (if another operation)
```

Next, implement a trigger that maintains this constraint: raise an exception using `RAISE EXCEPTION` when it is violated, with the message:

```
'This part already has 5 subparts.'
```

Name your trigger function `check_children_function()` and your trigger `check_children`.

c) Consider a second integrity constraint: *no part can be cheaper than the sum of its* ***direct*** *subparts.* Specify the critical database operation(s) that can violate this constraint in a comment somewhere in your file like this:

```
-- Critical operation(s): OP1, OP2 (if another operation)
```

Next, implement a trigger that maintains it. As in (b), the trigger should raise an exception, with the message:

```
'The super part cannot be cheaper than the sum of its direct subparts.'
```

Name your trigger function `check_price_function()` and your trigger `check_price`.

Problem 4: Datalog (EXTRA CREDIT)

Consider a binary relation `parent(`C`,`P`)` which means that P is a parent of C. Write Datalog rules for the following relations:

a) `grandparent(`C`,`G`)`: G is a grandparent of C.

b) `ancestor(`C`,`A`)`: A is an ancestor of of C.

c) `samegen(`X`,`Y`)`: X and Y are in the same generation, i.e., (i) they share a parent P, or (ii) their parents P_1 and P_2 are themselves in the same generation.

d) `lca(`C`,`D`,`A`)`: A is the lowest common ancestor of C and D, i.e., A is an ancestor of both C and D, but there is no other shared ancestor A' of C and D which is "lower" than A.