Database System Implementation

Solutions for Chapter 11

Return to Top

Solutions for Section 11.1

Exercise 11.1.2(b)

One of the problems that needs to be handled is that there is no natural `id` for a system composed of a computer and monitor from company A. We shall thus leave this field null in `Systems`, although there might be some reasonable way to treat integers as strings and concatenate them to invent system identifiers. The insert operation is:

```
INSERT INTO Systems(processor, mem, disk, screenSize)
    SELECT speed, memory, hd, screen
    FROM Computers CROSS JOIN Monitors;
```

Remember that `Systems.processor` is analogous to `Computers.speed`, not `Computers.proc`.

Exercise 11.1.3

The simplest approach is for the global schema to mimic company A's schema. The systems sold by company B can be represented by using the same identifier for both the computer and the monitor. However, since the attributes `proc` of `Computers` and `maxResX` and `maxResY` of `Monitors` are not known for systems of company B, we shall have to use `NULL` as the value for each of these components.

Exercise 11.1.8(a)

At Dealer 1, the query would be:

```
SELECT serialNo
FROM Cars
WHERE autoTrans = 'yes';
```

while at Dealer 2 the query would be:

```
SELECT serialNo
FROM Options
WHERE option = 'autoTrans';
```

Return to Top

Solutions for Section 11.2

Exercise 11.2.1

```
SELECT *
FROM AutosMed
WHERE color = '$c';
    =>
SELECT serialNo, model, color, autoTrans, 'dealer1'
FROM Cars, GtoL
WHERE Cars.color = GtoL.globalColor AND
      GtoL.localColor = '$c';
```

Exercise 11.2.2(a)

At Company A we need:

```
SELECT *
FROM PCMed
WHERE speed = $s;
    =>
SELECT 'A', speed, memory, hd, screen
FROM Computers CROSS JOIN Monitors
WHERE speed = $s;
```

At B we use:

```
SELECT *
FROM PCMed
WHERE speed = $s;
    =>
SELECT 'B', processor, mem, disk, screenSize
FROM Systems
WHERE processor = $s;
```

Exercise 11.2.3(a)

1. Use the templates described in the solution to Exercise 11.2.2(a) to get all tuples for systems with a 400 megahertz speed.

2. Then, select only those `disk` attribute has the value 12.0.

Return to Top

Solutions for Section 11.3

Exercise 11.3.1

For part (a): in the usual meanings of the terms, `quant` and `price` would be dependent attributes, and the others would be dimension attributes. For part (b): There is no one right answer. Here is a reasonable choice. We have assumed that there is an ID attribute for each of the attributes of `Orders`; the ID's appear in the tuples of `Orders` and link those tuples to tuples of the dimension tables. We use the same names for attributes of `Orders` and their corresponding dimension tables.

```
Cust(ID, name address, phone, creditCard)
Date(day, month, year)
Proc(ID, manf, model, name, speed)
Disk(ID, manf, model, cylinders, capacity, surfaces, rotSpeed)
CD(ID, manf, type, speed)
```

Return to Top

Solutions for Section 11.4

Exercise 11.4.1(a)

The ratio is (11/10)^10, or about 2.59.

Exercise 11.4.2(a)

We would use the tuples of the form

```
Sales(*, 'blue', *, d, val, cnt)
```

where *d* ranges over all dealers.

Exercise 11.4.3

In addition to the suggested processor type, manufacturer, and speed, we might choose to split the CD dimension into type (CD or DVD, e.g.), model number, manufacturer, and speed. The hard-disk component might be split into manufacturer, model, capacity, and (rotational) speed. Finally, we may wish to analyze customers by including such dimensions as their zip-code and their method of payment. Zip codes may be part of a hierarchy of locations, including cities, states, and perhaps regions.

The dependent attributes will be quantity ordered, and total price of the order.

Exercise 11.4.8(b)

Regions are placed into the diagram ``in parallel'' with States. That is, the diagram looks like:

```
           All
        /       \
    States    Regions
        \       /
          Cities
             |
          Dealers
```

Exercise 11.4.10

Really large. For example, suppose that there are *n* independent attributes. Let *F* have only one tuple, say (1,1,...,1), with a nonzero dependent attribute. Every point in the cube that consists of *'s and 1's only will have a nonzero dependent attribute. There are *2^n* such points, so the ratio of the size of CUBE(*F*) to *F* can be as high as *2^n*.

It is easy to see the ratio can be no higher, since each point in an *n*-dimensional space contributes to only *2^n* aggregations.

Return to Top