CHAPTER 25

Advanced Data Types and New Applications

This chapter covers advanced data types and new applications, including temporal databababases, spatial and geographic databases, multimedia databases, and mobility and personal databases. In particular, the data types mentioned above have grown in importance in recent years, and commercial database systems are increasingly providing support for such data types through extensions to the database system variously called cartridges or extenders.

This chapter is suited as a means to lay the groundwork for an advanced course. Some of the material, such as temporal and spatial data types, may be suitable for self-study in a first course.

Exercises

25.9 Will functional dependencies be preserved if a relation is converted to a temporal relation by adding a time attribute? How is the problem handled in a temporal database?

Answer: Functional dependencies may be violated when a relation is augmented to include a time attribute. For example, suppose we add a time attribute to the relation *account* in our sample bank database. The dependency *account-number*→*balance* may be violated since a customer's balance would keep changing with time.

To remedy this problem temporal database systems have a slightly different notion of functional dependency, called *temporal functional dependency*. For example, the temporal functional dependency:

$$\textit{account-number} \xrightarrow{\tau} \textit{balance}$$

over *Account-schema* means that for each instance *account* of *Account-schema*, all snapshots of *account* satisfy the functional dependency

$$account\text{-}number \rightarrow balance;$$

i.e at any time instance, each account will have a unique bank balance corresponding to it.

25.10 Consider two-dimensional vector data where the data items do not overlap. Is it possible to convert such vector data to raster data? If so, what are the drawbacks of storing raster data obtained by such conversion, instead of the original vector data?

Answer: To convert non-overlapping vector data to raster data, we set the values for exactly those pixels that lie on any one of the data items (regions); the other pixels have a default value.
The disadvantages to this approach are: loss of precision in location information (since raster data loses resolution), a much higher storage requirement, and loss of abstract information (like the shape of a region).

25.11 Study the support for spatial data offered by the database system that you use, and implement the following:

a. A schema to represent the geographic location of restaurants along with features such as the cuisine served at the restaurant and the level of expensiveness.

b. A query to find moderately priced restaurants that serve Indian food and are within 5 miles of your house (assume any location for your house).

c. A query to find for each restaurant the distance from the nearest restaurant serving the same cuisine and with the same level of expensiveness.

Answer: PostgreSQL includes support for R-tree indices over spatial data, as well as a number of built-in geometric data types (points, boxes, circles, lines, and paths) to represent spatial data, and functions to manipulate this data.

a.

```
create table restaurants (
    name varchar(30),
    location point,
    cusine varchar(30),
    price int)
```

b. Assume your house is at coordinates (21.5, 14.2), and that a price value of 2 means "moderately priced".
`<->` is the PostgreSQL operator representing "distance between".

```
select name
```

```
from restaurants
where ((point '(21.5, 14.2)') <-> location) < 5.0
    and cusine = 'Indian'
    and price <= 2
```

c.

```
select r1.name, min(r1.location <-> r2.location)
from restaurants as r1, restaurants as r2
where r1.cusine = r2.cusine
    and r1.price = r2.price
group by r1.name
```

25.12 What problems can occur in a continuous-media system if data are delivered either too slowly or too fast?

Answer: Continuous media systems typically handle a large amount of data, which have to be delivered at a steady rate. Suppose the system provides the picture frames for a television set. The delivery rate of data from the system should be matched with the frame display rate of the TV set. If the delivery rate is too low, the display would periodically freeze or blank out, since there will be no new data to be displayed for some time. On the other hand, if the delivery rate is too high, the data buffer at the destination TV set will overflow causing loss of data; the lost data will never get displayed.

25.13 List three main features of mobile computing over wireless networks that are distinct from traditional distributed systems.

Answer: Some of the main distinguishing features are as follows.

- In distributed systems, disconnection of a host from the network is considered to be a *failure*, whereas allowing such disconnection is a *feature* of mobile systems.
- Distributed systems are usually centrally administered, whereas in mobile computing, each personal computer that participates in the system is administered by the user (owner) of the machine and there is little central administration, if any.
- In conventional distributed systems, each machine has a fixed location and network address(es). This is not true for mobile computers, and in fact, is antithetical to the very purpose of mobile computing.
- Queries made on a mobile computing system may involve the location and velocity of a host computer.
- Each computer in a distributed system is allowed to be arbitrarily large and may consume a lot of (almost) uninterrupted electrical power. Mobile systems typically have small computers that run on low wattage, short-lived batteries.

25.14 List three factors that need to be considered in query optimization for mobile computing that are not considered in traditional query optimizers.

Answer: The most important factor influencing the cost of query processing in traditional database systems is that of disk I/O. However, in mobile computing, minimizing the amount of energy required to execute a query is an important task of a query optimizer. To reduce the consumption of energy (battery power), the query optimizer on a mobile computer must minimize the size and number of queries to be transmitted to remote computers as well as the time for which the disk is spinning.

In traditional database systems, the cost model typically does not include connection time and the amount of data transferred. However, mobile computer users are usually charged according to these parameters. Thus, these parameters should also be minimized by a mobile computer's query optimizer.

25.15 Give an example to show that the version-vector scheme does not ensure serializability. (Hint: Use the example from Practice Exercise 25.8, with the assumption that documents 1 and 2 are available on both mobile computers A and B, and take into account the possibility that a document may be read without being updated.)

Answer: Consider the example given in the previous exercise. Suppose that both host A and host B are not connected to each other. Further, assume that identical copies of document 1 and document 2 are stored at host A and host B.

Let $\{X = 5\}$ be the initial contents of document 1, and $\{X = 10\}$ be the initial contents of document 2. Without loss of generality, let us assume that all version-vectors are initially zero.

Suppose host A updates the number its copy of document 1 with that in its copy of document 2. Thus, the contents of both the documents (at host A) are now $\{X = 10\}$. The version number $V_{1,A,A}$ is incremented to 1.

While host B is disconnected from host A, it updates the number in its copy of document 2 with that in its copy of document 1. Thus, the contents of both the documents (at host B) are now $\{X = 5\}$. The version number $V_{2,B,B}$ is incremented to 1.

Later, when host A and host B connect, they exchange version-vectors. The version-vector scheme updates the copy of document 1 at host B to $\{X = 10\}$, and the copy of document 2 at host A to $\{X = 5\}$. Thus, both copies of each document are identical, viz. document 1 contains $\{X = 10\}$ and document 2 contains $\{X = 5\}$.

However, note that a serial schedule for the two updates (one at host A and another at host B) would result in both documents having the *same* contents. Hence this example shows that the version-vector scheme does not ensure serializability.