Chapter 6

Exercises

1. What are two types of time, and how are they different? Why does it make sense to have both types of time associated with a tuple?

   ANS:

   Answer: A temporal database models the changing states of some aspects of the real world. The time intervals related to the data stored in a temporal database may be of two types - valid time and transaction time. The valid time for a fact is the set of intervals during which the fact is true in the real world. The transaction time for a data object is the set of time intervals during which this object is part of the physical database. Only the transaction time is system dependent and is generated by the database system.

   Suppose we consider our sample bank database to be bitemporal. Only the concept of valid time allows the system to answer queries such as - "What was Smith's balance two days ago?". On the other hand, queries such as - "What did we record as Smith's balance two days ago?" can be answered based on the transaction time. The difference between the two times is important. For example, suppose, three days ago the teller made a mistake in entering Smith's balance and corrected the error only yesterday. This error means that there is a difference between the results of the two queries (if both of them are executed today).

2. Suppose you have a relation containing the x, y coordinates and names of restaurants. Suppose also that the only queries that will be asked are of the following form: The query specified a point, and asks if there is a restaurant exactly at that point. Which type of index would be preferable. R-tree or B-tree?

   ANS: Answer: The given query is not a range query, since it requires only searching for a point. This query can be efficiently answered by a B-tree

   index on the pair of attributes (x, y).

3. Suppose you have a spatial database that supports region queries (with circular regions) but not nearest-neighbor queries. Describe an algorithm to find the nearest neighbor by making use of multiple region queries.

   Answer: Suppose that we want to search for the nearest neighbor of a point P in a database of points in the plane. The idea is to issue multiple

   region queries centered at P . Each region query covers a larger area of points than the previous query. The procedure stops when the result of a region query is non-empty. The distance from P to each point within this region is calculated and the set of points at the smallest distance is reported.