

11

12

8. A method for structuring a time index in a computer-based temporal database management system which comprises:

- a) assembling in computer memory an ordered series of indexing time points defining time intervals during which at least one of a plurality of concurrent object version in said database is valid; and
- b) associating with each indexing point pointer means identifying all database object versions that are valid at the time represented by said each indexing point.

9. A method according to claim 8 which further comprises selecting as said indexing time points those points on a scale of regularly occurring time points that define a change of state of said database with respect to an object version.

10. A method according to claim 9 which comprises selecting time points:

- a) at which a database object version begins; and
- b) next following the scalar time point at which a database object version terminates.

11. A method according to claim 8 wherein said assembling comprises inserting said indexing time points as the population of the leaf nodes of a B+-tree index structure.

12. A method according to claim 11 which further comprises associating with each said indexing time

point a pointer to a bucket of pointers identifying said valid database object versions.

13. A method according to claim 12 which further comprises associating with said time index B+-tree, by means of a leaf node pointer therefrom to said time index B+-tree, a higher level B+-tree index of database attributes related to the time interval object versions of said database.

14. A method according to claim 12 which further comprises, upon inserting in said database a new object version whose time interval extends from a first time point to a second time point:

- a) if there exists no indexing point at said first time point, creating an indexing point at said first time point and creating a pointer bucket for said indexing point;
- b) if there exists no indexing point at the next scalar time point following said second time point, creating an indexing point at said next time point and creating a pointer bucket for said indexing point; and
- c) inserting a pointer to said new object version in the bucket for the indexing point at said first time point and in the buckets for all subsequent indexing points prior to said second time point.

* * * * *

30

35

40

45

50

55

60

65