13

Schema Specification

Richard T. Snodgrass, Christian S. Jensen, and Fabio Grandi

1 Introduction

In this chapter, the Create statement is enhanced to support the definition of valid-time relations. Additionally, the Alter statement should be extended to allow these definitions to be later changed.

In the language extension to be discussed shortly, we adopted the following goals.

1. Extensions should be upward compatible with current SQL2.
2. Extensions should be as minimal as possible.
3. As few reserved words as possible should be introduced.
4. Punctuation should be consistent with the rest of the language.
5. Extensions should be consistent and compatible with user-defined time syntax.
6. The clean lines of the BCDM should be retained.
7. Defaults should be carefully chosen to reflect common usage and to enable a suitable reduction proof (see (1)).

2 Informal Definition

We examine several examples of the new Create and Alter statements, to provide a very informal description. As will be seen, this is an extension of the previous syntax.

The `Employee` relation, with `Name`, `Dept`, and `Salary` attributes, would be defined as follows in SQL2.

```
CREATE TABLE Employee (Name CHAR, Dept CHAR, Salary INT)
```

An `ON` clause can be added at the end, e.g.,

```
CREATE TABLE Employee (Name CHAR, Dept CHAR, Salary INT)
ON COMMIT PRESERVE ROWS
```

The specification of a valid-time relation is accomplished through an additional clause, placed before the `ON` clause. To specify a valid-time relation,

```
CREATE TABLE Employee (Name CHAR, Dept CHAR, Salary INT)
AS VALID
```

The precision and scale of the last component of the `AS VALID` clause can also be specified, e.g.,

```
CREATE TABLE Employee (Name CHAR, Dept CHAR, Salary INT)
AS VALID(11,6)
```

The Alter statement can be used to later modify the schema. The precision can be changed as follows.

```
ALTER TABLE Employee ADD VALID (11,0)
```

Finally, the table can be changed back into a snapshot relation.

```
ALTER TABLE Employee DROP VALID
```

3 Syntactic Considerations

We considered and rejected the following alternatives, for the reasons given.

- `CREATE INTERVAL TABLE` is awkward when the precision must be specified, e.g., `CREATE INTERVAL(11,6) TABLE Employee`. Also, this conflicts with the sense in SQL-92 of intervals as *unanchored* durations of time.
- A distinguished attribute, say with name `ValidTime`, is incompatible with TSQL2's temporal projection syntax.
- As another <table element> is inappropriate, because the <column definition> and <table constraint definition> clauses that make up a <table element> both heavily refer to table columns, whereas the valid as clause applies to the table as a whole.