Using SQL Arrays

HyperSQL supports the full semantics and all the related features of Standard SQL ARRAY type. The ARRAY type is a relatively recent addition to the SQL Standard and allows storing an ARRAY of attributes in a column of a table.

The ARRAY type is covered in the Guide in these sections:

Array Usage

Array Functions

Using an ARRAY, instead of a separate table, is a choice for the developer. An ARRAY is used if it is more appropriate to the data model. It is also more efficient to store an array instead of storing each of its elements as a row in a separate table. In the examples given here, the array as a whole is an attribute of the row and it is more appropriate to store the array in the row, instead of storing its elements in a separate table.

Example One : A list of passwords

This complete example was posted by Michael Dever in our Open Discussion Forum. It uses an ARRAY to implement a rotating password list for each user of an application.

Example of ARRAY usage

Example Two : Polygons

Polygon data is stored as an ordered list of coordinates. It is possible to store a Java array directly in a column of type OTHER. However, this solution makes the elements of the array inaccessible to SQL queries. In the example below, the coordinates are stored in an array.

```
CREATE TABLE T (ID INT PRIMARY KEY, X_COORD DOUBLE ARRAY, Y_COORD DOUBLE ARRAY)

-- optional constraint to check arrays are of equal size
ALTER TABLE T ADD CONSTRAINT COORD_CHECK CHECK (CARDINALITY(X_COORD) = CARDINALITY(Y_COORD))

INSERT INTO T VALUES 1, ARRAY[3.45, 23.64, 14.01], ARRAY[2.01, 6.05, 25.34]
INSERT INTO T VALUES 2, ARRAY[5.45, 20.64, 12.01], ARRAY[3.01, 7.05, 22.34]

-- this query returns the two arrays, which can be accessed with the getArray() method of the JDBC ResultSet interface
SELECT X_COORD, Y_COORD FROM T WHERE ID = 2

-- this query returns the first coordinate from the two arrays, counting the array index from 1
SELECT X_COORD[1], Y_COORD[1] FROM T WHERE ID = 2

-- this is the same as the last query but uses trigraphs instead of square brackets
SELECT X_COORD ??( 1 ??), Y_COORD ??( 1 ??) FROM T WHERE ID = 2

-- this query returns a table
-- note the UNNEST table expression can combine two or more scalar arrays into a table
SELECT ID, X, Y, POS FROM T, UNNEST(X_COORD, Y_COORD) WITH ORDINALITY COORDS(X, Y, POS) WHERE ID = 2

ID X       Y       POS
-- ------- ------- ---
2  5.45E0  3.01E0  1
2  20.64E0 7.05E0  2
2  12.01E0 22.34E0 3
```

SOURCEFORGE

This page last updated 23 Sep 2012