



ArrayQL Working Group Announcement

Kian-Tat Lim
SLAC National Accelerator Laboratory

Peter Baumann

Jacek Becla

Kian-Tat Lim

Martin Kersten

Dave Maier

Mike Stonebraker

Dimension (indexing)

Order

Adjacency

History of Arrays in Relational Databases

Long history

SQL:2003

In stored procedures and as column types

Extended query language

Multiple attributes per array cell

Extensive operations on single arrays

- For each array in a set

First-Class Arrays

Named arrays

Operations combining two (or more) arrays

First-Class Array Implementations

SciDB/Paradigm4

- Only arrays

MonetDB SciQL

- Arrays within existing SQL databases
- Arrays and tables interoperate well

Working Group Goals

Two tracks

- Common algebra to describe array operations
- Common syntax for first-class arrays for common uses

Consensus-building

Neutral broker

Research and ideas

Project management

Box

- Dimensions and bounds

Valid

- Cells with content

Content

- Tuples of attributes

Algebra: Operators

Rename

Shift

Rebox

Filter

Fill

Apply

Combine

InnerDJoin

InnerEJoin

Reduce

Draft Syntax

```
CREATE ARRAY array-name ...
```

```
SELECT ... FROM subarray-expr ...
```

```
CREATE ARRAY array-name FROM  
    SELECT ... FROM subarray-expr
```

Purposely looks like SQL

No inserts or updates yet

CREATE ARRAY

```
CREATE ARRAY matrix (  
  x INTEGER DIMENSION [-2:2],  
  y INTEGER DIMENSION [0:*],  
  z INTEGER DIMENSION [1:4],  
  v1 FLOAT DEFAULT 0.0,  
  v2 INTEGER  
);
```

DIMENSION clause with bounds

Queries

```
SELECT [1:3] AS i, [1:3] AS j, SUM(product)
FROM (
  SELECT [1:3] AS i, [1:3] AS j, [1:3] AS k,
         a.iv * b.rv AS product
  FROM a[i, k], b[k, j]
) AS tmpArray
GROUP BY i, j;
```

SELECT clause

Output dimension expressions

```
SELECT [1:3] AS i, [1:3] AS j, [1:3] AS k,  
       a.iv * b.rv AS product  
FROM a[i, k], b[k, j]
```

FROM clause

Subarray expressions

```
SELECT [1:3] AS i, [1:3] AS j, [1:3] AS k,  
       a.iv * b.rv AS product  
FROM a[i, k], b[k, j]
```


WHERE clause, GROUP BY

Both can include dimensions

GROUP BY attribute not yet available

Examples

```
SELECT *, [x] AS i, [y] AS j FROM matrix;
```

```
SELECT *, [2:3] AS i, [1:2] AS j FROM matrix;
```

```
SELECT *, [x] AS i, [y] AS j FROM matrix[j, i];
```

```
SELECT *, [x] AS i, [y] AS j FROM matrix[i, j]  
WHERE i MOD 2 = 1;
```

```
SELECT [1:3] AS i, v1, v2 FROM matrix[i, 1];
```

```
SELECT [1:3] AS i, [1:3] AS j, a.iv + c.fv FROM a, c[i+1, j];
```

```
SELECT [1:3] AS i, SUM(rv) FROM b GROUP BY m;
```

```
SELECT [1:3] AS i, [1:3] AS j, d.iv, b.rv  
FROM d[i, j], b[d.iv, j];
```

How To Contribute



Read draft documents

Send comments to arraydb-1@slac.stanford.edu