Part II: Database Design and Access

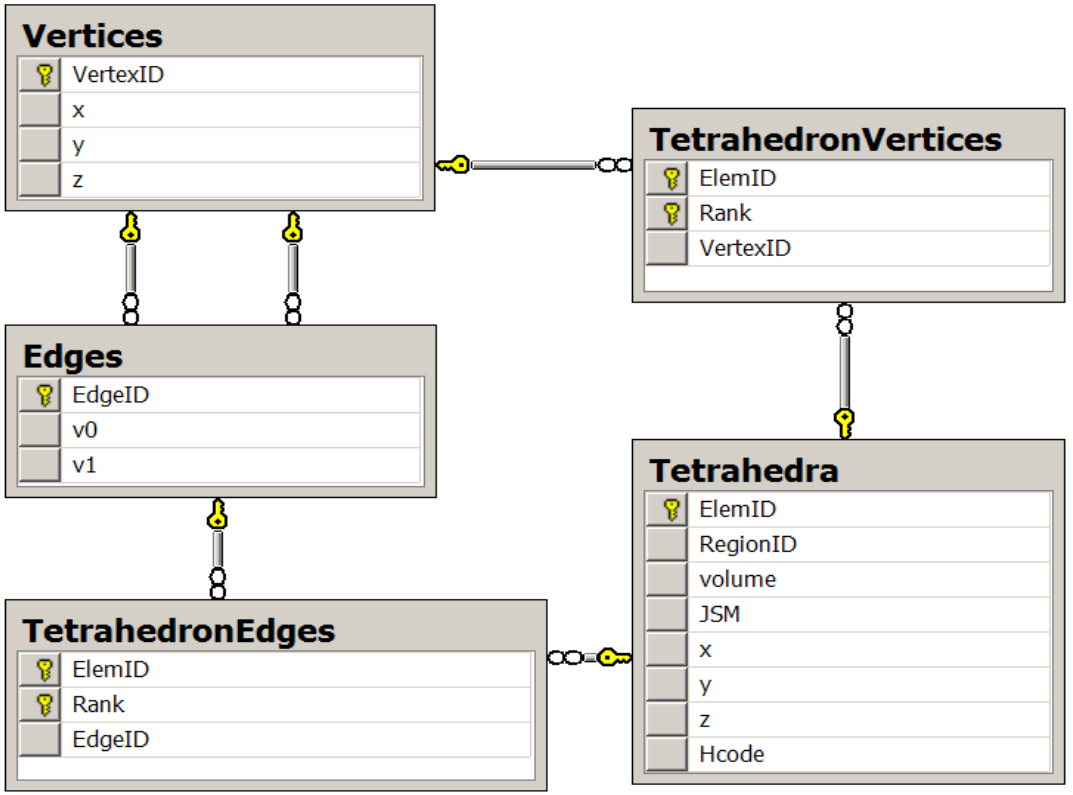
Bilgen

Brief Recap - Intro

- Goal: Simplify FEA by utilizing DB
- File-based vs. DB-based
  - byte stream / data subset
- DB gives what FEA wants: strongly typed, self-defining semantic info + data independence + support for arbitrary read/writes

Brief Recap - FEA with DB

- Represent meshes with element-vertex relations and element attributes.



Brief Recap - FEA with DB

- Read and Write (DB)
  - Bulk copy commands

    ```
    bcp DB.dbo.T in C:\F.dat –N -T
    ```

  - ASCII files, Binary files, SQL tables
- ETL problem
  - Data loading requirements (monitoring, debugging, execution..)
  - SSIS
- Simulation Data
  - Inputs (partitions of meshes)
  - Outputs (support for fast analysis)

Comutational Geometry Problems

- Locating points and interpolating field values at those points
- Issues
  - Search structures (code complexity)
  - Parallelism (code complexity)
  - Memory limitations

The Point-in-Cell Query for Unstructured Meshes

- Finding the mesh tetrahedron containing a given point
- Directed Local Search method
  - Select a candidate tetrahedron by rank comparison on a Hilbert space-filling curve
  - Containment Test and Tetrahedron-connectivity-graph traversal

DLS in SQL Server 2005

- Calculate H code for each tetrahedron and create a clustered index on it

```
public class Hilbert3D {
  // we use only the lower 20 bits of x, y, z
  public static UInt64 H_encode(UInt32 x, UInt32 y, UInt32 z);
  // this is what we will call from T-SQL
  [SqlFunction(IsDeterministic = true, IsPrecise = true, DataAccess = DataAccessKind.None)]
  public static SqlInt64 H_encode_SQL(SqlInt32 x, SqlInt32 y, SqlInt32 z) {
    return H_encode((UInt32)x, (UInt32)y, (UInt32)z)); }
}
```

```
CREATE FUNCTION fnH_encode(@x int, @y int, @z int)
RETURNS bigint AS EXTERNAL NAME [SFC].[SFC.Hilbert3D].[H_encode_SQL]
```

```
CREATE TABLE Tetrahedra (
  ElemID int    PRIMARY NONCLUSTERED KEY,
  X      float  NOT NULL,
  y      float  NOT NULL,
  z      float  NOT NULL,
  Hcode  bigint NOT NULL,
...)
UPDATE Tetrahedra SET Hcode = dbo.fnH_encode(x, y, z)
CREATE CLUSTERED INDEX Idx_Hcode ON Tetrahedra(Hcode)
```

DLS in SQL Server 2005

- Point-in-Tet Test

```
CREATE FUNCTION fnPointInTet(@ElemID int, @x float, @y float, @z float) RETURNS bit
AS EXTERNAL NAME FEMUtils.FEMUtils.Tetrahedron.PointInTet
```

- Intersection facets

```
CREATE FUNCTION fnRayIntersectsTetFace(@ElemID int, @x float, @y float, @z float) RETURNS tinyint
AS EXTERNAL NAME FEMUtils.FEMUtils.Tetrahedron.RayIntersectsTetFace
```

- GetTetFaceNeighbor

```
CREATE FUNCTION fnGetTetFaceNeighbor(@ElemID int, @oppositeVertex tinyint) RETURNS int AS
BEGIN
  DECLARE @result int            -- result is ID of desired tet
  SELECT @result = ElemID        -- find the first face
  FROM TetrahedronVertices       -- in the Tet-Vertex table
  WHERE VertexID IN (            -- where all the vertices in common with
    SELECT VertexID              -- one of the vertices of @ElemID
    FROM TetrahedronVertices     -- (this select statment returns
    WHERE ElemID  = @ElemID      -- all the vertices of the tet)
      AND Rank    != @ oppositeVertex) -- excluding the one opposite the face
  AND ElemID != @ElemID          -- it is a different face
  GROUP BY ElemID                -- group matching vertices
  HAVING COUNT(VertexID) = 3     -- insist all 3 verticies match
  RETURN COALESCE(@result, -1)   -- return -1 if no such element
END
```

Performance

- Points per second
- 20,000 points in each table T (created from a center-radius pair)
-
```
SELECT COUNT(DISTINCT dbo.fnGetTet4Point(x,
y, z)) FROM T
```

Performance

Table 1: Points per second and distinct tet count for sample clouds.

c	r	Points / s	#Distinct Tetrahedra
(2.1, 1.35, 0.65)	0.00001	2857	1
(2.1, 1.35, 0.65)	0.0001	2222	2
(2.1, 1.35, 0.65)	0.001	1818	2
(2.1, 1.35, 0.65)	0.01	741	8
(2.1, 1.35, 0.65)	0.1	689	294
(2.1, 1.35, 0.65)	0.25	689	1831
(2.1, 1.35, 0.65)	0.5	476	2443
(2.1, 1.35, 0.65)	0.6	426	2889
(0, 0, 0)	0.5	416	3249

Table 2: Points per second and distinct tet count for random sample clouds

N	$N_{c,r}$	$\mu(r)$	$\sigma(r)$	Points / s	#Distinct Tetrahedra
10	2000	0.0312	0.0130	833	54
20	1000	0.0299	0.0142	588	159
200	100	0.0267	0.0148	250	1398
2000	10	0.0246	0.0145	208	4389
10	2000	0.0623	0.0261	556	138
20	1000	0.0598	0.0284	571	412
200	100	0.0534	0.0296	312	2412
2000	10	0.0491	0.0289	167	5405
20000	1	N/A	N/A	75	7392

Thanks!