

SQL/MED and PostgreSQL

Peter Eisentraut

FOSDEM 2009

What is SQL/MED?

- MED = Management of External Data
- Methods to access non-SQL data in SQL
- SQL/MED is ISO/IEC 9075-9

- Connect to other DBMS (like DBI-Link)
- Connect to other PostgreSQL instances (like dblink)
- Read non-SQL data (CSV, XML)
- Manage data stored in file system
 - Images
 - Video
 - Engineering data

Why do we care?

- Powerful new functionality
- Unifies existing ad-hoc solutions.
- Implementation has begun in PostgreSQL 8.4.
- Several people have plans for PostgreSQL 8.5.
- See status report later in this presentation.

Advantages

- Schema integration
- Access control
- Standard API
- Centralized control through DBMS

Wrapper interface *Access other data sources, represent them as SQL tables*

Datalinks *Manage files stored in file system, represent file references as column values*

Wrapper Interface Concepts

- Define a *foreign table* . . .
- On a *foreign server* . . .
- Accessed through a *foreign-data wrapper*

Example: Foreign-Data Wrappers

Foreign-data wrapper (FDW): a library that can communicate with external data sources

```
CREATE FOREIGN DATA WRAPPER foosql
  LIBRARY 'foosql_fdw.so'
  LANGUAGE C;
```

- PostgreSQL communicates with `foosql_fdw.so` using SQL/MED FDW API.
- `foosql_fdw.so` communicates with FooSQL server using their own protocol.
- In theory, FooSQL, Inc. would ship `foosql_fdw.so` with their product.
- In practice, this is not so wide-spread.

Example: Foreign Servers

Foreign server: an instance of an external data source accessed through a FDW

```
CREATE SERVER extradb  
  FOREIGN DATA WRAPPER foosql  
  OPTIONS (host 'foo.example.com', port '2345');
```

- Options depend on FDW.

Example: User Mappings

User mapping: additional user-specific options for a foreign server

```
CREATE USER MAPPING FOR peter SERVER extradb  
  OPTIONS (user 'peter', password 'seKret');
```

- Options depend on FDW.
- Putting connection options into server vs. user mapping is a matter of convention or convenience.

Example: Foreign Tables

Foreign table: a table stored on a foreign server

```
CREATE FOREIGN TABLE data
  SERVER extradb
  OPTIONS (tablename 'DATA123');
```

- Now you can read and write the table as if it were local (depending on FDW features/implementation).
- Options specified for FDW, server, and user mapping are used as connection parameters (depending on FDW).

Another Wrapper Interface Example

Possible setup for accessing HTML tables stored in a web site as SQL tables:

```
CREATE FOREIGN DATA WRAPPER htmlfile  
  LIBRARY 'html_fdw.so'  
  LANGUAGE C;
```

```
CREATE SERVER intranetweb  
  FOREIGN DATA WRAPPER htmlfile  
  OPTIONS (baseurl 'http://intranet/data');
```

```
CREATE FOREIGN TABLE data  
  SERVER intranetweb  
  OPTIONS (path 'foo.html#//table[@id="table1"]');
```

More Wrapper Interface Features

- GRANT **and** REVOKE
- IMPORT FOREIGN SCHEMA
- CREATE ROUTINE MAPPING

PostgreSQL 8.4 has:

- CREATE FOREIGN DATA WRAPPER, but no library support
- CREATE SERVER
- CREATE USER MAPPING
- ACL support
- Used by dblink and PL/Proxy to store connection information

- IBM DB2 provides a full implementation
- MySQL and Farrago use some syntax elements
- No other known implementations

- Write wrapper library and foreign table support for PostgreSQL 8.5
- Supply a few foreign-data wrapper libraries with PostgreSQL 8.5
- Use standard wrapper interface API or design our own API?

- Files are referenced through a new `DATALINK` type
- Database system has control over external files
- No need to store file contents in database system
- Access control and integrity mechanisms of DBMS can be extended to file system

Example: Simple DATALINK Type

```
CREATE TABLE persons (  
    id        integer,  
    name      text,  
    picture  DATALINK [NO LINK CONTROL]  
);
```

```
INSERT INTO persons VALUES (  
    1,  
    'Jon Doe',  
    DLVALUE('file://some/where/1.jpg')  
);
```

- This variant doesn't do anything except store URLs.

DATALINK Attributes

Selection of additional possible attributes for `DATALINK` fields:

FILE LINK CONTROL Datalink value must reference an existing file.

INTEGRITY ALL Referenced files can only be renamed or deleted through SQL.

INTEGRITY SELECTIVE Referenced files can be renamed or deleted through SQL or directly.

READ PERMISSION DB Database system controls file read permission.

RECOVERY YES PITR applies to referenced files.

ON UNLINK DELETE File is deleted from file system when deleted from database.

How to Implement Datalinks

How to implement this?

- OS-dependent
- File-system dependent
- Application-dependent
- Lots of hocus pocus needed
- Possibilities: kernel modules, LD_PRELOAD, extended FS attributes
- Don't hold your breath.

SQL/MED:

- Wrapper interface
- Datalinks
- Substantial support planned for PostgreSQL 8.5 and beyond

References:

- <http://wiki.postgresql.org/wiki/SqlMedConnectionFactory> (Martin Pihlak)
- <http://www.sigmod.org/record/issues/0103/JM-Sta.pdf> (Jim Melton et al.)
- <http://www.sigmod.org/record/issues/0209/jimmelton.pdf> (Jim Melton et al.)
- ISO/IEC 9075-9:2008 (“SQL/MED”)