

SQL/MED

Doping for PostgreSQL

Peter Eisentraut

Senior Software Engineer
Lab Development
F-Secure Corporation

PGCon 2009



SQL/MED: Management of External Data

- ▶ MED = **M**anagement of **E**xternal **D**ata
- ▶ Methods to access data stored outside the database system through normal SQL
- ▶ SQL/MED is ISO/IEC 9075-9



Applications and Use Cases

- ▶ Connect to other DBMS (like DBI-Link)
 - ▶ Other primary data storage: Oracle, MySQL, ...
 - ▶ Data warehouses etc.: Greenplum, Truviso, ...
- ▶ Connect to other PostgreSQL instances (like dblink)
- ▶ Read non-SQL data
 - ▶ Files: CSV, XML, JSON, ...
 - ▶ File systems: Google FS, Hadoop FS, Lustre, ...
 - ▶ Databases: CouchDB, BigTable, NDB, S3, ... (“Cloud” stuff)
 - ▶ Memcache
- ▶ Clustering, partitioning (think PL/Proxy)
- ▶ Manage data stored in file system
 - ▶ Images
 - ▶ Video
 - ▶ Engineering data

Applications and Use Cases

- ▶ Connect to other DBMS (like **DBI-Link**)
 - ▶ Other primary data storage: Oracle, MySQL, ...
 - ▶ Data warehouses etc.: Greenplum, Truviso, ...
- ▶ Connect to other PostgreSQL instances (like **dblink**)
- ▶ Read non-SQL data
 - ▶ Files: **CSV**, **XML**, JSON, ...
 - ▶ File systems: Google FS, Hadoop FS, Lustre, ...
 - ▶ Databases: CouchDB, BigTable, NDB, S3, ... (“Cloud” stuff)
 - ▶ Memcache
- ▶ Clustering, partitioning (think **PL/Proxy**)
- ▶ Manage data stored in file system
 - ▶ Images
 - ▶ Video
 - ▶ Engineering data

Why do we care?

- ▶ Unifies existing ad-hoc solutions.
- ▶ Powerful new functionality
- ▶ Makes PostgreSQL the center of data management.
- ▶ 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 All data appears as tables.

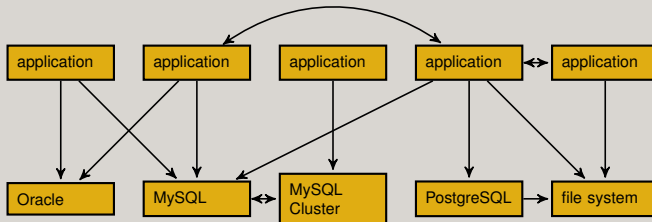
Access control Use GRANT/REVOKE for everything.

Standard APIs Mix and share.

Centralized control Manage all data through the DBMS.

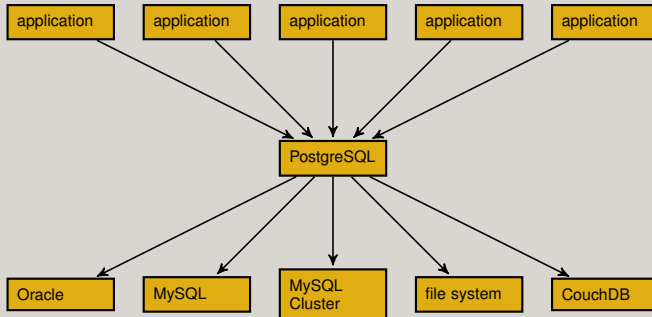
Implications for Application Design and Deployment

Before



Implications for Application Design and Deployment

After



The Two Parts of SQL/MED

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**

Wrapper Interface Concepts

- ▶ Define a foreign table ...
 - ▶ think: a dblink view
- ▶ On a foreign server ...
 - ▶ think: dblink_connect
- ▶ Accessed through a foreign-data wrapper
 - ▶ think: dblink.so library

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.

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.

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.

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"]');
```


Routine Mappings

Routine mappings: passing a function/procedure through to a foreign server

```
CREATE ROUTINE MAPPING <routine mapping name>  
  FOR <specific routine designator>  
  SERVER <foreign server name>  
  [ <generic options> ];
```

Routine Mappings Examples

Example like PL/Proxy:

```
CREATE ROUTINE MAPPING myfunc(a int, b text)
  SERVER plproxydb
  OPTIONS (cluster 'somecluster',
          runon 'hashtext(a)');
```

Example XML-RPC:

```
CREATE ROUTINE MAPPING process(data xml)
  SERVER xmlrpc
  OPTIONS (request '<methodCall>...</methodCall>');
```

Wrapper Interface Access Control

- ▶ GRANT USAGE ON FOREIGN DATA WRAPPER
- ▶ GRANT USAGE FOREIGN SERVER
- ▶ Foreign tables and routines have regular privileges.
- ▶ Passwords for remote access can be managed via user mappings.
- ▶ Front-to-end Kerberos or SSL support could be cool.

Importing Foreign Schemas

Automatically create foreign tables based on tables available remotely.

```
IMPORT FOREIGN SCHEMA someschema  
  LIMIT TO (tab1, tab2, tab2)  
  FROM SERVER extradb  
  INTO myschema;
```

(SQL standard doesn't support foreign routine import.)

Status of SQL/MED in PostgreSQL 8.4

PostgreSQL 8.4 has:

- ▶ CREATE FOREIGN DATA WRAPPER, but no library support
- ▶ CREATE SERVER
- ▶ CREATE USER MAPPING
- ▶ ACL support
- ▶ Doesn't really do anything :-)
- ▶ Plans for PL/Proxy to store connection information

Status of SQL/MED Elsewhere

- ▶ IBM DB2 provides a full implementation.
- ▶ MySQL and Farrago use some syntax elements.
- ▶ No other known implementations.
- ▶ Some vendors have had their custom remote access functionality.

Plan & Issues

PostgreSQL 8.5 and beyond ...

- ▶ Write wrapper library and foreign table support
- ▶ Supply a few foreign-data wrapper libraries
- ▶ Use standard wrapper interface API or design our own API?
- ▶ Optimizations, e. g., passing query qualifications to foreign servers
- ▶ Distributed transactions
- ▶ Needs careful security evaluation (remember dblink issues)

Plan & Issues

PostgreSQL 8.5 and beyond ...

- ▶ Write wrapper library and foreign table support
- ▶ Supply a few foreign-data wrapper libraries
- ▶ Use standard wrapper interface API or design **our own API?**
- ▶ Optimizations, e. g., passing query qualifications to foreign servers
- ▶ Distributed transactions
- ▶ Needs careful security evaluation (remember dblink issues)

Datalink Concepts

- ▶ 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

Datalink Use Cases

- ▶ Certain types of data are primarily uses as files with external applications.
- ▶ Handling very large files (e. g., video) by DBMS is inefficient
- ▶ Use of distributed files systems
- ▶ Handle files stored on web server, FTP server, etc.

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')  
);
```

- ▶ SQL specifies support for `file:` and `http:`.
- ▶ This variant doesn't do anything except store URLs.

DATALINK Attributes: Link and Integrity Control

NO LINK CONTROL Datalink value need not reference an existing file/URL.

FILE LINK CONTROL Datalink value must reference an existing file/URL.

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.

INTEGRITY NONE (implied for NO LINK CONTROL)

DATALINK Attributes: Unlinking and Recovery Behavior

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

ON UNLINK RESTORE File's original permissions are restored when deleted from database.

ON UNLINK NONE No change in file permissions when file reference is deleted from database.

RECOVERY YES PITR applies to referenced files.

RECOVERY NO PITR does not apply to referenced files.

DATALINK Attributes: Access Permissions

READ PERMISSION FS File system controls file read permission.

READ PERMISSION DB Database system controls file read permission.

WRITE PERMISSION FS File system controls file write permission.

WRITE PERMISSION ADMIN Writes to the file are managed by the database system.

WRITE PERMISSION BLOCKED Writing to file is blocked.

How to Implement Datalinks

Implementation challenges:

- ▶ OS-dependent
- ▶ File-system dependent
- ▶ Application-dependent

Possibilities:

- ▶ Kernel modules
- ▶ LD_PRELOAD
- ▶ Extended FS attributes
- ▶ Lots of hocus pocus

Don't hold your breath.



Summary

SQL/MED

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

Further reading:

- ▶ <http://wiki.postgresql.org/wiki/SqlMedConnectionManager>
(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”)

Rights and Attributions

This presentation "SQL/MED: Doping for PostgreSQL" was authored by Peter Eisentraut and is licensed under the [Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported license](#).

- ▶ The image on page 2 is [from the Open Clip Art Library](#) and is in the public domain.
- ▶ The image on page 5 is [from the Open Clip Art Library](#) and is in the public domain.
- ▶ The image on page 9 is "The fork in the road" by Flickr user [i_yudai](#), available under the [Creative Commons Attribution 2.0 Generic license](#).
- ▶ The image on page 31 is "Fork in a Steve" by Flickr user [morgantepsic](#), available under the [Creative Commons Attribution-Share Alike 2.0 Generic license](#).