

Can't We All Just Get Along??—Interacting with Other Popular Databases

Presented by

Greg Mays

Principal Consultant, B2B Solutions



Introduction

ORACLE®



PostgreSQL



Agenda

- History/Origins of MySQL and PostgreSQL
- General Feature Comparison
- Advanced Feature Comparison
- Interesting Insights
- Connecting Oracle to MySQL and PostgreSQL via Heterogeneous Services



MySQL

Company: MySQL AB (now a subsidiary of Oracle)

Initial Release: May 1995

Current Release: 5.1.51 (Sept 2010)

Written In: C, C++

Trivia: MySQL was named after original developer Michael Widenius' daughter My.



MySQL

Platforms: AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, eComStation, OS/2 Warp, QNX, IRIX, Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos, Tru64 and Microsoft Windows. A port of MySQL to OpenVMS also exists.

Well known users:

- Facebook
- Zappos.com
- Wikipedia



MySQL

License: Open Source GNU license or via EULA
with MySQL AB (now Oracle)



PostgreSQL

Company: Open Source community maintained

Initial Release: June 1990

Current Release: 9 (Sept 2010)

Written in: C

Trivia: “PostgreSQL evolved from the [Ingres](#) project at the [University of California, Berkeley](#). In 1982, the project leader, [Michael Stonebraker](#), left Berkeley to make a proprietary version of Ingres. He returned to Berkeley in 1985 and started a post-Ingres project to address the problems with contemporary database systems that had become increasingly clear during the early 1980s. The new project, Postgres, aimed to add the fewest features needed to completely support types. These features included the ability to define types and to fully describe relationships – something used widely before but maintained entirely by the user. In Postgres, the database “understood” relationships, and could retrieve information in related tables in a natural way using *rules*. Postgres used many of the ideas of Ingres, but not its code.” (from Wikipedia)



PostgreSQL

Platforms: FreeBSD, Unix/Linux, Mac OS X, Solaris, Windows

Well known users:

- Yahoo (highly modified implementation for web user behavioral analysis, 2 petabytes)
- MySpace
- Skype
- Reddit.com



PostgreSQL

License: Open source license similar to BSD and MIT. Allows users to do anything they want with the code but cannot hold group legally liable for any problems.

Third Party Distributions: EnterpriseDB widely known but a number of other “forks”:

http://wiki.postgresql.org/wiki/PostgreSQL_derived_databases



General Feature Comparison

Both MySQL and PostgreSQL support similar basic features to Oracle:

- Stored Procedures
- Triggers
- Views
- Multi-language support



General Feature Comparison

- Datatypes – All support comparable datatypes along with binary object and text data capabilities though MySQL and Postgresql have more math optimal numeric types (smallint, decimal, real, double, etc.)
- Stored Procedures
- Triggers
- Views
- Multi-language support



Advanced Feature Comparison

Query Optimizer:

Oracle's Cost Based Optimizer and features which support tremendously complex queries is almost without peer.

MySQL has a stated limit of max of 61 tables in a join.

(<http://dev.mysql.com/doc/refman/5.5/en/restrictions.html>)



Advanced Feature Comparison

Index Types:

Oracle has a huge range of index types and index related functionality.

MySQL supports b-tree, hash, fulltext, cluster, and GIS (spatial) indexes but doesn't have the equivalent of a function based index.

PostgreSQL supports b-tree, hash, r-tree, and custom GIST type (allows user-defined types and function based indexes to be created)



Advanced Feature Comparison

Auditing

Oracle provides extensive auditing capability.

PostgreSQL has fairly comparable auditing capabilities.

MySQL doesn't have specific auditing capabilities though they could be built using triggers, etc.



Advanced Feature Comparison

Transaction Support

ACID (Atomicity, Consistency, Isolation, Durability) is debated heavily in comparisons of Oracle to its open source counterparts and in general most agree that MySQL has come a long way in this regard (depending on the storage engine chosen, old default MyISAM was NOT ACID compliant but InnoDB is) but PostgreSQL is better in this arena and Oracle is the leader.



Advanced Feature Comparison

Transaction Support

MySQL has different storage engines and the MyISAM engine is not fully ACID compliant while the InnoDB, BDB, and Cluster storage engines are.



Advanced Feature Comparison

Backups

Both databases provide “online” or “hot” backup options through integration with media managers, etc. as Oracle’s RMAN does is not in place by default.



Advanced Feature Comparison

Partitioning

PostgreSQL supports horizontal partitioning by range or list. Also supports vertical partitioning.

MySQL supports horizontal partitioning by range, list, internal or linear hashing functions...but does not support vertical partitioning.



Advanced Feature Comparison

Replication

Both PostgreSQL and MySQL have several replication options and support this capability.



Interesting Insights

PostgreSQL – Routine VACUUMing! These databases require routine use of the VACUUM command to:

- Recover disk space occupied by updated or deleted rows
- To update data statistics used by the query planner
- Protect against loss of very old data due to transaction id “wraparound”

Note: VACUUM can be run in parallel with other operations EXCEPT for table definition changes.



Interesting Insights

The VACUUM requirement is because PostgreSQL does not immediately remove the old version of a row (as a means to achieve MVCC) so a table that is frequently updated or deleted will require more VACUUM attention than other tables.



Interesting Insights

PostgreSQL Transaction Wraparound

- Reliance on transaction id (XID) numbers for MVCC causes a limitation that every table must be “vacuumed” at least ONCE every BILLION transactions.
- Not an onerous requirement BUT failure to do could lead to data loss!
- Recent version changes will now warn when the threshold is being approached and include an optional “auto-vacuum daemon”



Interesting Insights

Archivelog Mode equivalents...

PostgreSQL – WAL (write ahead log)
needs to be considered in backup scheme

MySQL – Setup depends on storage engine.
InnoDB most obvious choice for hot
backups with ibbackup utility which relies
on transaction logs.



Why would I need to interact with these databases from Oracle?

- Embedded with software packages being used
- One-time/one-off routine data extractions or pushes
- Acquisitions/integrations, etc.
- Hybrid systems (Web server farms for serving content alongside transactional systems)



Connecting Oracle to Other Databases

Oracle's Heterogeneous Services (ODBC)

Oracle "Transparent" Gateway products
(additional \$'s!, not all databases supported)



Connecting Oracle to Other Databases

Heterogeneous Services allows:

- Use of Oracle SQL to transparently access data in non-Oracle systems as if the data resides within an Oracle Server
- Use of Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment



Setting up Heterogeneous Services

- 1) Acquire and install appropriate ODBC driver for the desired database and install in your environment.

NOTE: This is most straightforward on Windows based systems but there are ODBC drivers available for Unix/Linux systems and other platforms as well.



Setting up Heterogeneous Services

- 2) Configure the `init<ODBC DSN>.ora`
Under the `<Oracle Home>\hs` subdirectory
copy `inithsodbc.ora` to `init<ODBC DSN>.ora`
- 3) Edit the `init<ODBC DSN>.ora` and set the
`HS_FDS_CONNECT_INFO=<datasource >`, and `HS_FDS_TRACE_LEVEL=OFF`



Setting up Heterogeneous Services

4) Edit the <Oracle Home>\network\admin\listener.ora and add entry for the DSN

```
(SID_DESC =  
  (SID_NAME = PostgreSQL30)  
  (ORACLE_HOME = C:\oracle\product\10.2.0\db_1)  
  (PROGRAM = hsodbc)  
)
```



Setting up Heterogeneous Services

5) Reload listener (lsnrctl reload)

6) Add entry to <Oracle

Home>\network\admin\tnsnames.ora

```
POSTGRESQL30 =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SID = PostgreSQL30)  
    )  
    (HS = OK)  
  )
```



Setting up Heterogeneous Services

- 7) Create database link in the Oracle database schema where you need to access the DSN

```
CREATE DATABASE LINK <ODBC DSN> CONNECT TO "<DSN Username>"  
IDENTIFIED BY "<Password>" using '<ODBC DSN/TNS service>';
```



Setting up Heterogeneous Services

8) Use it!

```
Select * from EMP@POSTGRE30;
```

```
Select * from EMP@MYSQLTEST;
```



What if???

Q: What if I'm not on Windows?

A: There are Linux/Unix ODBC drivers available.

<http://www.unixodbc.org/>

<http://www.easysoft.com/developer/interfaces/odbc/linux.html>

Q: What if that's not an option?

A: Perhaps use a Windows machine as a gateway to the data you're trying to access via DB links.



What if???

Q: Heterogeneous services isn't fully included in my 64-bit software?

A: Depending on the Oracle version the HS directory may exist BUT the hsodbc executable isn't there and can't be built 64-bit because typically drivers were 32-bit. Research carefully dependent on platform, Windows especially.



Conclusion

PostgreSQL seems technically more powerful than MySQL and closer to Oracle capability but both databases when used in realization of their strengths and weaknesses can provide high levels of capability rivaling Oracle...for significantly less \$'s.

A “hybrid” solution mixing MySQL/PostgreSQL with Oracle might be the right choice for some applications.



References/Resources

MySQL

<http://www.mysql.com>

PostgreSQL

<http://www.postgresql.org>

Oracle – Heterogeneous Services – 11g

http://download.oracle.com/docs/cd/E11882_01/server.112/e11050/toc.htm

Oracle – Heterogeneous Services – 10g

http://download.oracle.com/docs/cd/B19306_01/server.102/b14232/toc.htm

PostgreSQL for Oracle DBAs

http://wiki.postgresql.org/wiki/PostgreSQL_for_Oracle_DBAs



Resources

This presentation will be available at:

VOUG Website: www.voug.org

B2B Website: www.b2bsol.com (under
Resources)

Email: gmays@b2bsol.com

