A global IT company that helps businesses leverage disruptive technologies to better compete.

Our services and software solutions unleash the power of cloud, data and analytics to drive better business outcomes for our clients.

Our 20 years in data, commitment to hiring the best talent, and our deep technical and business expertise allow us to meet our promise of using technology to deliver the best outcomes faster.
DIVERSIFIED, BLUE CHIP CLIENT BASE

<table>
<thead>
<tr>
<th>Media/Information Services</th>
<th>Retail</th>
<th>E-commerce</th>
<th>SaaS</th>
<th>Financial Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOX Sports</td>
<td>SONOS</td>
<td>Tech Data</td>
<td>Grasshopper</td>
<td>PIRAEUS BANK</td>
</tr>
<tr>
<td>NATIONAL GEOGRAPHIC</td>
<td>URBAN OUTFITTERS</td>
<td>Arc'teryx</td>
<td>mapp</td>
<td>COMMERZBANK</td>
</tr>
<tr>
<td>AEG</td>
<td>ALLSAINTS</td>
<td>freshdirect</td>
<td>etouches</td>
<td>Banque Saudi Fransi</td>
</tr>
<tr>
<td>BBM</td>
<td></td>
<td>Camelot GLOBAL</td>
<td>Entrust Datcard</td>
<td>Paymentus</td>
</tr>
<tr>
<td>medianet</td>
<td></td>
<td>rockyou</td>
<td></td>
<td>TaxAudit</td>
</tr>
<tr>
<td>FAST RETAILING</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© The Pythian Group Inc., 2019
About me

- > 20 years in IT
- 2 kids (5 and 8) and 2 dogs
- 3 Years with Pythian
- 6 Years working with MySQL
- AWS Certified Solutions Architect - Associate
- GCP Certified Professional Cloud Architect
Getting Started with PostgreSQL

by Kevin Markwardt
Introduction

- My Journey on Learning PostgreSQL
- Looking at it from a MySQL perspective
- Review Major Differences that I have found
- Look at what makes PostgreSQL special
Major Differences

- **Command Line and Clients**: mysql and mysqladmin (MySQL) - psql and pg_ctl (PostgreSQL)
- **Client Commands**: SHOW DATABASES; (MySQL) - \l (PostgreSQL)
- **Database and Schemas**: Databases (MySQL) - Schemas (PostgreSQL)
- **Users**: Users (MySQL) - Roles (PostgreSQL)
- **Connections**: Threads (MySQL) - Processes (PostgreSQL)
- **MVCC**: Purge (MySQL) - Vacuuming (PostgreSQL)
- **Memory Management**: Buffer Pool (MySQL) - Shared Memory (PostgreSQL)
- **Replication**: Binary Logs (MySQL) - WAL Logs (PostgreSQL)
- **File System**: File Per Table (MySQL) - PGDATA (PostgreSQL)
Major Differences - Command Line and Clients (MySQL)

- **mysql** - Command line client. Default login is with root user with no password from localhost.

```
root@mysql:~# mysql
Welcome to the MySQL monitor. Commands end with ; or \
Your MySQL connection id is 82
Server version: 5.7.25-28-log Percona Server (GPL), Release '28', Revision 'c335905'

Copyright (c) 2009-2019 Percona LLC and/or its affiliates
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
root@mysql:~#`
```

- **mysqladmin** - Administrative client
- **mysqldump** - Logical backup
- **MySQL Workbench** - Graphical client
Major Differences - Command Line and Clients (PostgreSQL)

- **psql** - Command line client. Default login is done with the postgres user account

```
[root@pg-base1 vagrant]# su - postgres -c 'psql'
psql (10.7)
Type "help" for help.

postgres=# \\q
[root@pg-base1 vagrant]#
```

- **pg_ctl** - Administrative client
- **pg_dump** - Logical backup for one database
- **pg_dumpall** - Logical backup of all the databases
- **pg_basebackup** - Binary backup
- **pgAdmin** - Graphical client
# Major Differences - Client Commands

## MySQL

- **Client Commands**
  - `SHOW DATABASES;` - List databases
  - `use <db_name>` - Connect to database
  - `SELECT user, host FROM mysql.user` - List users
  - `SHOW TABLES;` - List tables
  - `DESCRIBE TABLE <table_name>` - Describe table
  - `SHOW PROCESS LIST;` - Process list
  - `SHOW VARIABLES;` - Show settings

## PostgreSQL

- **Client commands**
  - `\l` - List databases
  - `\c <db_name>` - Connect to database
  - `\du` - List users
  - `\dt` - List tables
  - `\d <table>` - Describe table
  - `select * from pg_stat_activity;` - Process list
  - `show all` - Show settings
  - `\?` - Show commands available
Major Differences - Database and Schemas

- MySQL has Databases Only
- PostgreSQL has both Databases and Schemas
  - Schemas add a layer of isolation within a database between tables
  - Tables in different schemas but in the same database can have the same name
  - Used to organize database objects into logical groups
  - “public” is the default schema in a database
Major Differences - Users (MySQL)

- Users are created with the “CREATE USER” command
- Privileges are granted with the “GRANT” commands
- Restrictions on where users can login from are based upon IP address or hostname
- MySQL 8.0 and MariaDB 10.0 has ROLES that allows permissions to be granted to a role and then users added or removed from the role

```
mysql> CREATE USER 'app_user'@'%' IDENTIFIED BY 'SUPERSECRETpassword';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'app_user'@'';
Query OK, 0 rows affected (0.00 sec)
```
Major Differences - Users (PostgreSQL)

- Users has been depreciated and there are only ROLES
- A ROLE can be a user or a group. A ROLE becomes a user when it is given the LOGIN permission. By default a ROLE is created with NOLOGIN
- Permissions to databases and schemas are granted to roles
- Restricting where a ROLE can login from is done in the `pg_hba.conf` configuration file.

```
# IPv4 local connections:
host  postgres  postgres  192.168.10.102/32  md5
```

- File is scanned on each connection request
- Grants to database objects
  - Table, column, view, sequence, database, foreign-data wrapper, foreign server, function, procedural language, schema, or tablespace
Major Differences - Connections (MySQL)

Threads

- MySQL runs as a single process
- New connections coming into MySQL spawn a new thread within the MySQL process
- Threads are lightweight
- Each Thread connection is **256k** on a 64-bit system by default
- Less CPU in opening and closing threaded connections
- If a thread crashes, the entire MySQL process crashes
- The “SHOW PROCESSLIST” command should actually be called “SHOW THREADLIST”
Processes

- **Postmaster (Daemon)** - Is the main process in PostgreSQL. As a new connection comes in it spawns a backend process.
- **Backend processes** - A new process that is spawned from the postmaster process when a new connection is established to handle the requests that.
- Memory consumption can be up to **10MB** per connection.
- When scaling PostgreSQL, connection planning is a very critical component to consider.
Major Differences - MVCC (MySQL)

- Undo logs stores the data changed in a transaction for rollback
- A row is not physically deleted until MySQL discards the update undo log record
- Commit transactions regularly, or InnoDB cannot discard data from the update undo logs
- Old data in tables is automatically cleaned up with the purge process that is run after commits
Major Differences - MVCC (PostgreSQL)

- Deleted records are marked with a `transaction_id` and removed from view.
- Updates insert a new row and the old row is marked with a `transaction_id`.
- Unused data is still in the table, known as tuples.
- A VACUUM process goes through the tables and removes the dead tuples.
- VACUUM FULL command can reclaim disk space, but locks the table.
- `pg_repack` (previously `pg_reorg`) does an online VACUUM FULL without holding an exclusive lock.
- Recommended blog: [https://www.percona.com/blog/2018/08/06/basic-understanding-bloat-vacuum-postgresql-mvcc/](https://www.percona.com/blog/2018/08/06/basic-understanding-bloat-vacuum-postgresql-mvcc/)
InnoDB buffer pool - Stores many in-memory data structures of InnoDB, buffers, caches, indexes and even row-data (Usually 80% of the system memory)

InnoDB log buffer - Allows transactions to run without having to write the log to disk before the transactions commit
Major Differences - Memory Management (PostgreSQL)

- **Backend Process** - Created by the primary PostgreSQL server process to handle client queries
  - Each client connection creates a backend process
  - Repeated connections are handled by outside tool like pg_bouncer
- **Shared Memory** - Used by all processes of a PostgreSQL server
  - Determines how much memory is dedicated to PostgreSQL to use for caching data and other processes
  - Usually 25% to 40% of system memory
Major Differences - Replication (MySQL)

- Uses binary logs which are logs of changes that have been made to the database
- Binary log data is replicated using Row, Statement, or Mixed replication formats
- Many types of replication configurations such as Master/Slave, Master/Master, Percona XtraDB or Galera Clustering, Multi Master, and more
- Replication assumes all changes are to be replicated to slaves unless filters are put into place

Diagram:

- Master
  - Slave connects to Master
  - I/O thread asks for data
  - Binlog dump thread sends contents to I/O Thread
  - SQL thread applies data

- Slave
  - I/O
  - SQL

- Database
- Binary Log
- Relay Log

MySQL DML commands

Binlog dump

© The Pythian Group Inc., 2019
Major Differences - Streaming Replication (Postgres)

- Before PostgreSQL 10 there was only streaming replication
- Streaming replication implements file-based log shipping by transferring WAL records one file (WAL segment) at a time
- If replication breaks or a failover occurs, the slave needs to be rebuilt
- Primary and standby nodes must be the same version
- Everything must be replicated, no filtering
Major Differences - Logical Replication (PostgreSQL)

- PostgreSQL 10 and greater introduces logical replication
- Data is replicated from a publisher that has a publication to subscribers that have a subscription
- Publications can be configured on a subset of data
- When subscription is created, the data is then backed up from publisher and restored to the subscriber
- Changes are then replicated using WAL logs
- A publication can have multiple subscriptions from different servers
- Cross version support and can be used for migrations and upgrades
- Multi source replication for consolidation
- Truncate and DDL such as ALTER TABLE are not supported in replication
- UPDATE and DELETE operations on missing data will be automatically skipped
Major Differences - File System (MySQL)

- Data Files are located in the data directory
- Logs are also stored in the data directory unless configured differently
- File per table config allows for tables to be stored individually on disk instead of shared table space
- Folder structure is similar to the layout within the database. Each database is a folder, and the tables are files within that folder
- Partitioning a table splits the table IBD file across multiple files
- /var/lib/mysql (Default path)
  - Database
    - Table IBD (table data file)
    - Table FRM (table definition file)
The data directory is known as a database cluster. Which is a single directory under which all data will be stored.

SHOW DATA_DIRECTORY; can be used to display the data directory
- /var/lib/pgsql/10/data

SELECT pg_relation_filepath('people'); can be used to find the path of a table
- base/16385/16388

Using this information you can then see things like file size on disk by looking at the file path /var/lib/pgsql/10/data/base/16385/16388

After 1 GB or configured size, tables are split across multiple files, adjusted using the configuration option --with-segsize when building PostgreSQL.
Minor Differences to Note

- PostgreSQL has a true Boolean data type where MySQL you have to use TinyINT with a 0 and 1
- PostgreSQL has SERIAL where MySQL has AUTO_INCREMENT
- MySQL is not case specific for string comparison. PostgreSQL is
- PostgreSQL is truly open source where MySQL is owned by Oracle and has components that need to be purchased
- MySQL is a relational database (RDB) where PostgreSQL is an object relational database (ORD), allowing for PostgreSQL to be customizable and extensible
- And many more ….
PostgreSQL specific features

- PostgreSQL is a developers dream database
- PostgreSQL has Procedural Languages PL/pgSQL, PL/Tcl, PL/Perl, and PL/Python
- Has Data domain objects for constraints across tables
- PostgreSQL has table inheritance
- Has materialized views, used in data warehousing
- Is Highly extensible, so you can add your own data types, operators, index types, and functional languages. You can even override different parts of the system, using plugins to alter the execution of commands, or add a new optimizer.
References

- http://blog.dumper.io/showdown-mysql-8-vs-postgresql-10/
- https://eng.uber.com/mysql-migration/
- https://www.datadoghq.com/blog/postgresql-monitoring/
- http://www.interdb.jp/pg/pgsql02.html
- https://severalnines.com/blog/architecture-and-tuning-memory-postgresql-databases
- https://severalnines.com/blog/postgresql-streaming-replication-deep-dive
- https://www.postgresql.org/docs/10/creating-cluster.html
Thank You