Transitioning Your Web Site to PostgreSQL

John Gotts

Linux Savvy Solutions, LLC

http://www.linuxsavvy.com/staff/jgotts

Installation

Red Hat Linux:

6.4.2 pre-installed with 6.0

6.5.2 pre-installed with 6.1

Other Distributions:

6.5.2 available at http://www.postgresql.org/

Building and installation documented in Administrator's Guide

Setting up the Database

To initialize the database structure:

```
# su - postgres
% initdb --pglib=/usr/lib/pgsql
--pgdata=/var/lib/pgsql
```

Start postmaster (and create the appropriate symlink for your runlevel):

```
# cd /etc/rc.d/rc3.d
# ln -s ../init.d/postgresql
S85postgresql
# ../init.d/postgresql start
```

Create your First Database

```
# su - postgres
% createdb postgres
% psql
...
postgres=> create table
player_ids (id int4 not null
primary key, name text);
...
postgres=> create table
player_stats (id int4 not null
primary key, team text, hits
int4, runs int4, rbi int4);
```

Tabular (Normal) Representation

player_ids:

name	id
Ausmus	1
Palmer	2
...	...

player_stats:

id	team	hits	runs	rbi
1	Tigers	10	2	3
2	Tigers	5	1	2
...	...	...	...	...

Entering Data into the Database

```
postgres=> insert into
player_ids values (1, 'Ausmus');
...
postgres=> insert into
player_stats values (1,
'Tigers', 10, 2, 3);
...
postgres=> insert into
player_ids values (2, 'Plamer');
```

Oops, I made a typo!

```
postgres=> delete from
player_ids where id=2;
```

Querying (Selecting) Data

```
postgres=> select * from
player_stats;
id|team   |hits|runs|rbi
--+-------+----+----+---
 1|Tigers|  10|   2|  3
 2|Tigers|   5|   1|  2
(2 rows)
```

With conditions:

```
postgres=> select * from
player_stats where rbi=2;
id|team   |hits|runs|rbi
--+-------+----+----+---
 2|Tigers|   5|   1|  2
(1 row)
```

Using PostgreSQL in your Programs

Interfaces:

C/C++, via libpq/libpq++

Tcl/Tk (pgtcl) - an example program is pgaccess, an PostgreSQL GUI admin tool

Perl, using the Pg module

Java (JDBC), Python, etc.

Included are Perl and C examples which illustate how to connect with the database and perform SQL queries.

```
/* Naive, pedagogical implementation of
   database insertion code. */

#include <stdio.h>
#include "libpq-fe.h"

#define QUERY_BUFFER_SIZE 8192
#define TEAM_WIDTH 10

int readLine(FILE *fp, char **line) {
    /* */
}

int parseLine(char *line, int *id, char
*team, int *hits, int *runs, int *rbi) {
    /* */
}

int main(void) {
    const char *pghost, *pgport,
    *pgoptions, *pgtty, *dbName, *dbUser,
    *dbPassword, *filename;
    FILE *fp;
    char *line;
    char team[TEAM_WIDTH];
    char queryString[QUERY_BUFFER_SIZE];
    int id, team, hits, runs ,rbi, oid;
    pghost = "localhost";
    pgport = NULL;
    pgoptions = NULL;
    pgtty = NULL;
    dbName = "postgres";
    dbUser = "postgres";
    dbPassword = NULL;
```

```
    PGconn *conn;
    PGresult *res;

    filename = “player_stats”;
    conn = PQsetdbLogin(pghost, pgport,
pgoptions, pgtty, dbName, dbUser,
dbPassword);

    fp = fopen(filename, “r”);
    while (readLine(fp, &line)) {
        parseLine(line, &id, team, &hits,
&runs, &rbi);
        free(line);
        snprintf(queryString,
QUERY_BUFFER_SIZE, "insert into
player_stats values (%d)”, id);
        res = PQexec(conn, queryString);
        oid = atoi(PQoidStatus(res));
        PQclear(res);
        snprintf(queryString,
QUERY_BUFFER_SIZE, "update player_stats set
team='%s',hits=%d,runs=%d,rbi=%d where
oid=%d”, team, hits, runs, rbi, oid);
        PQexec(conn, queryString);
        PQclear(res);
    }
    fclose(fp);
}
```

```
#!/usr/bin/perl

    # Naive Perl implementation of database
    # insertion code.

    use Pg;

    $pghost = "localhost";
    $pgport = “”;
    $pgoptions = “”;
    $pgtty = “”;
    $dbName = "postgres";
    $dbUser = "postgres";
    $dbPassword = “”;

    $conn = Pg::setdbLogin($pghost,
$pgport, $pgoptions, $pgtty, $dbName,
$dbUser, $dbPassword);

    open (FH, “player_stats”);
    while (<FH>) {
        ($id, $team, $hits, $runs, $rbi) =
split /\s+/;
        $res = $conn->exec(“insert into
player_stats values ($id)”);
        $oid = $res->oidStatus;
        $conn->exec(“update player_stats set
team='$team',hits=$hits,runs=$runs,rbi=$rbi
where oid=$oid”);
    }
    close FH;
```

PostgreSQL vs. MySQL

Nearly identical programming languages - ANSI standard SQL, though interfaces differ somewhat

Some types differ, e.g. MySQL does not support different sized integers, PostgreSQL offers *text* type

Administrative details vary somewhat, in particular setting up the database and creating new databases

In general PostgreSQL is more robust and can handle larger databases better, but MySQL is leaner and faster for smaller applications

Both databases are in active development and both databases are free software (open source)

More Information

PostgreSQL Web Site:

http://www.postgresql.org/

pgaccess:

http://www.flex.ro/pgaccess/

Sample code, including sources for psql

PostgreSQL Administrator's, Programmer's, and User's Guides

PostgreSQL Tutorial

Books:

The Practical SQL Handbook, Bowman et al

Final Words

These slides will be available at:

http://www.linuxsavvy.com/staff/jgotts/

If you have any questions, don't hesitate to contact me at:

jgotts@linuxsavvy.com