MS Access as a Front End for PostgreSQL

Matt Gushee

21st June 2002

1 Introduction

This document briefly describes a procedure for configuring Microsoft Access and PostgreSQL to work together as client and server.

1.1 Why?

Professional-quality web applications require a robust, high-performance database engine. PostgreSQL is one of the best databases available, and its price (free) is hard to beat. From a developer's perspective, however, it is useful to have a graphical interface for table and query design, and this is a feature that is lacking in PostgreSQL. Microsoft Access provides an easy-to-use GUI and can make an excellent client for PostgreSQL.

1.2 Prerequisites

This document assumes that you have working installations of PostgreSQL and Access. The command-line and SQL examples for PostgreSQL are taken from a Linux environment; you may encounter minor differences if you are running PostgreSQL on Windows.

2 PostgreSQL Setup

Before performing these steps, you should have PostgreSQL installed and running. You also need to have administrative privileges over the database.

2.1 Create the Database

The easiest way to create a PostgreSQL database is to use the `createdb` command:

```
bash-2.03$ createdb skills_matrix
CREATE DATABASE
```

Next we log in to our new database...

```
bash-2.03$ psql skills_matrix
Welcome to psql, the PostgreSQL interactive terminal.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help on internal slash commands
      \g or terminate with semicolon to execute query
      \q to quit
```

...and create some tables:

```
skills_matrix=# CREATE TABLE categories (
skills_matrix(# Category_ID INT PRIMARY KEY,
skills_matrix(# Category_Name VARCHAR(64) NOT NULL,
skills_matrix(# HTML_ID VARCHAR(16) NOT NULL
skills_matrix(# );
NOTICE: CREATE TABLE/PRIMARY KEY will create implicit index 'cate-
gories_pkey' for table 'categories'
CREATE
skills_matrix=#
```

2.2 Create a User

There are two ways to create a PostgreSQL user: you can use the `createuser` script from the command line, or you can log in to a database and use a SQL `CREATE USER` statement. I recommend the latter approach because it is easier that way to set some important options such as the password; also, you will need to log in to the database anyway to grant permissions to the user.

```
skills_matrix=# CREATE USER joebob;
CREATE USER
skills_matrix=#
```

However, there is one problem with the above command. Our new user has no password, which means he or she can access the database without no authentication. We can create a user with a password this way:

```
skills_matrix=# CREATE USER joebob WITH PASSWORD 'griddlewhelp';
CREATE USER
skills_matrix=#
```

2.3 Grant Permissions

By default, a new user has no access to any resources in the database. You can use the SQL `GRANT` command to give the user privileges on a per-table basis:

```
skills_matrix=# GRANT SELECT ON categories TO joebob;
CHANGE
skills_matrix=#
```

If you wish to give a user full control over a given table, rather than granting each permission individually, you can simply use the `ALL` keyword, as shown here:

```
skills_matrix=# GRANT ALL ON members TO joebob;
CHANGE
skills_matrix=#
```

2.4 Configure Remote Access

In order to allow remote users to connect to your database, you must edit the file $PG_DATA/pg_hba.conf.

```
# TYPE DATABASE IP_ADDRESS MASK AUTHTYPE MAP
....
host all 10.0.0.5 255.255.255.255 password
```

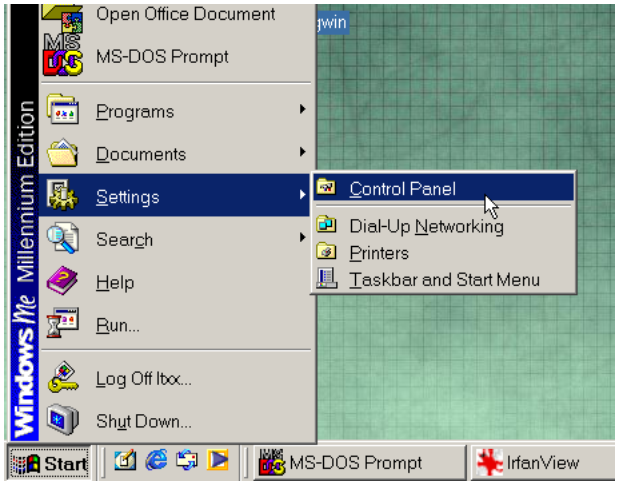
This line says that access is being given to a particular host or group of hosts; these hosts can access all databases in the system; the host being given access has an IP address of 10.0.0.5, with a netmask of 255.255.255.255 (i.e., the IP address must be *exactly* 10.0.0.5); and clear-text passwords will be used for authentication. The optional MAP parameter is omitted.

2.5 Restart the PostgreSQL Daemon

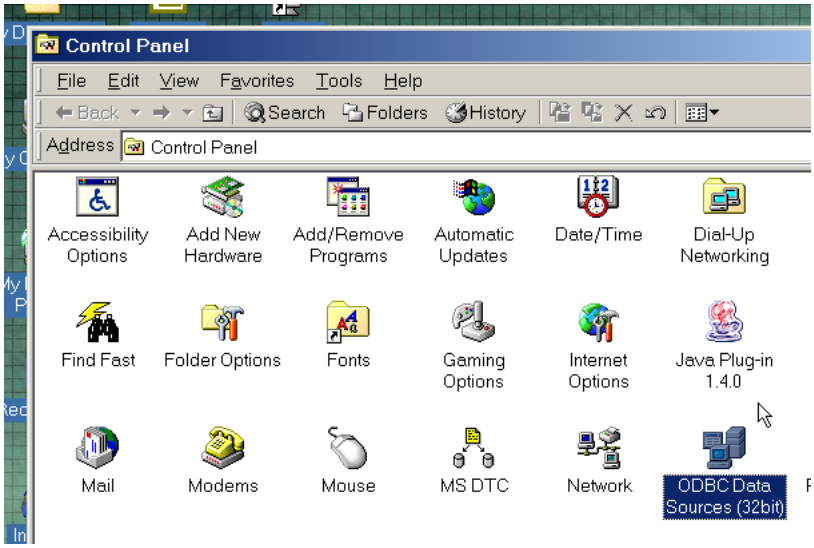
```
swordfish:/var/lib/postgres/data# su postgres
swordfish:~/data$ /opt/bin/pg_ctl restart
waiting for postmaster to shut down.....done
postmaster successfully shut down
postmaster successfully started
swordfish:~/data$ DEBUG: database system was shut down at 2002-06-
21 11:59:04 MDT
DEBUG: CheckPoint record at (0, 27162456)
DEBUG: Redo record at (0, 27162456); Undo record at (0, 0); Shut-
down TRUE DEBUG: NextTransactionId: 3239; NextOid: 48121
DEBUG: database system is in production state
```

3 Setting up an ODBC Data Source

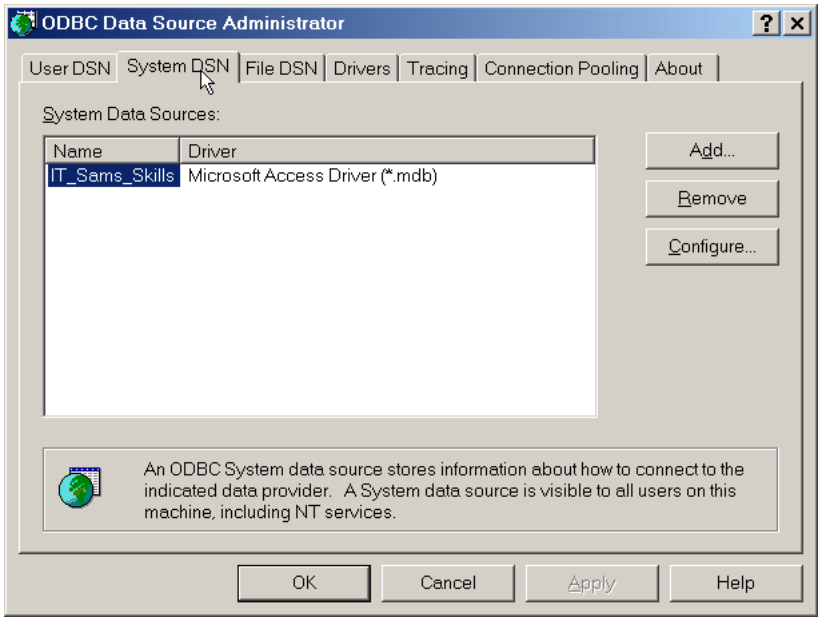
3.1 Open the Windows Control Panel



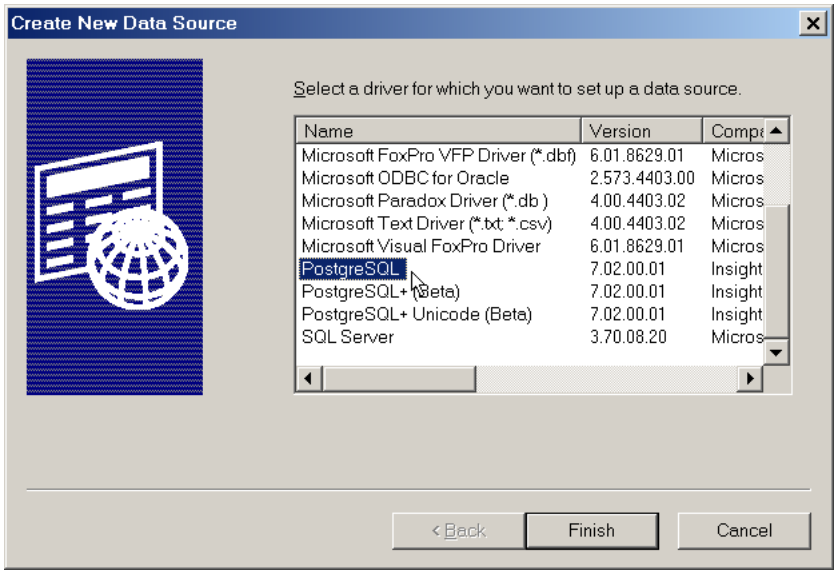
3.2 Select “ODBC Data Sources”



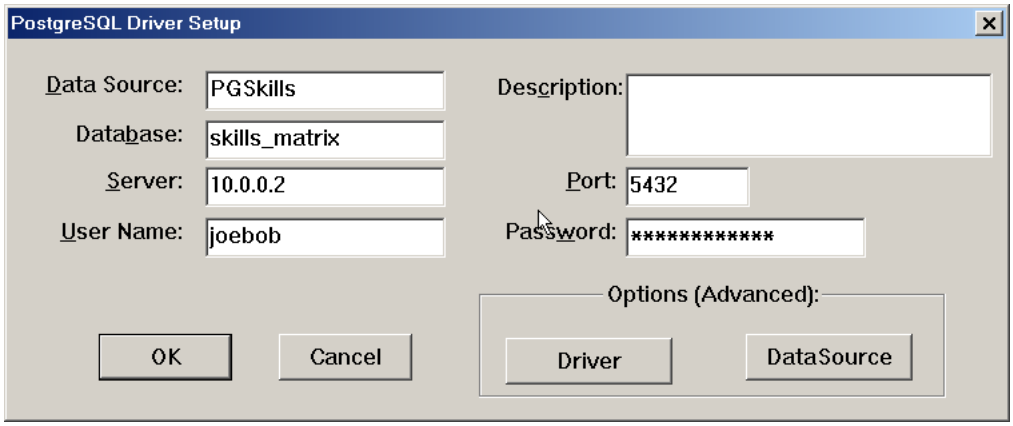
3.3 Select the “System DSN” Tab



3.4 Select the PostgreSQL Driver

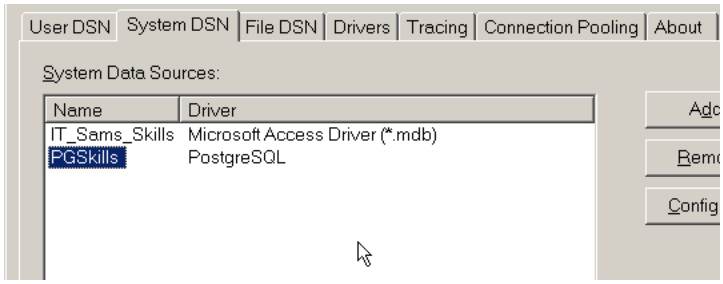


3.5 Fill in the Setup Form



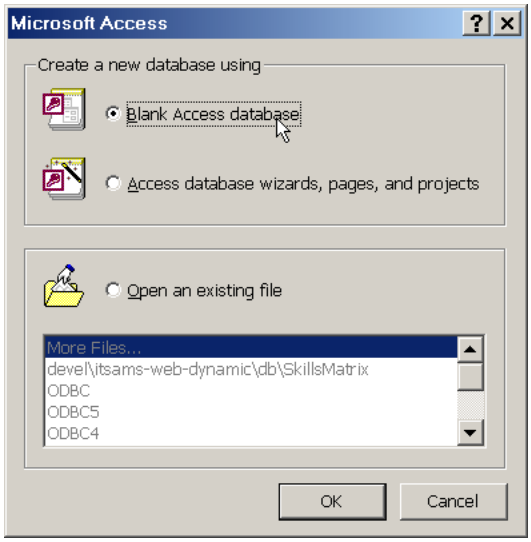
Field	*Description*
Data Source	Any name you wish to give the data source. Names with no spaces are preferable.
Database	The real name of the database you wish to access.
Server	The hostname or IP address of the database server.
Port	The port that the database daemon listens on. Usually 5432 for PostgreSQL.
User Name	The user name you created earlier.
Password	The password you created earlier.

3.6 Your Data Source is Ready

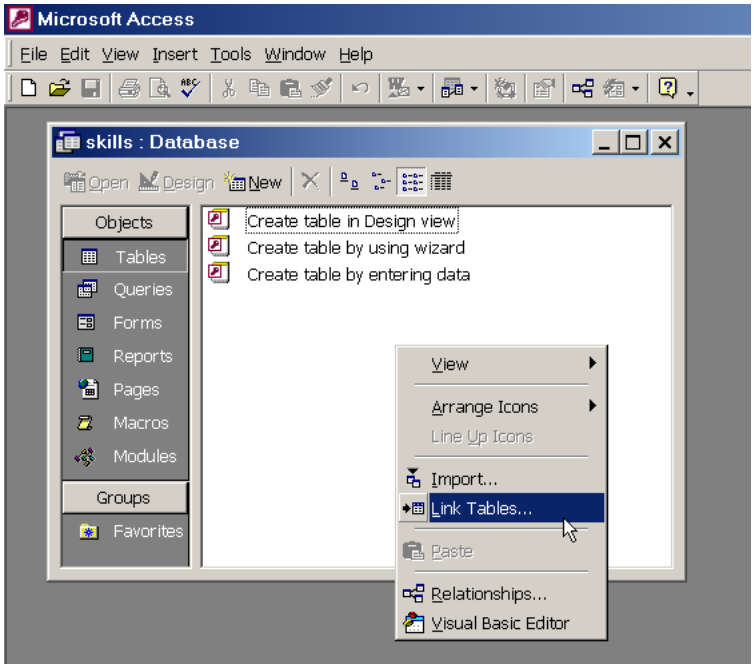


4 Creating a Linked Database in Access

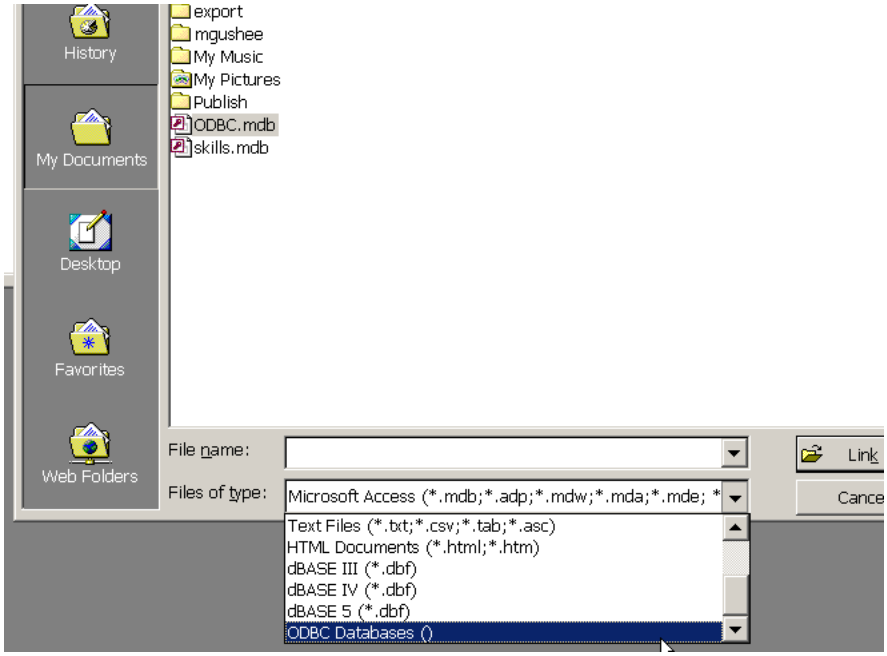
4.1 Create a Blank Database



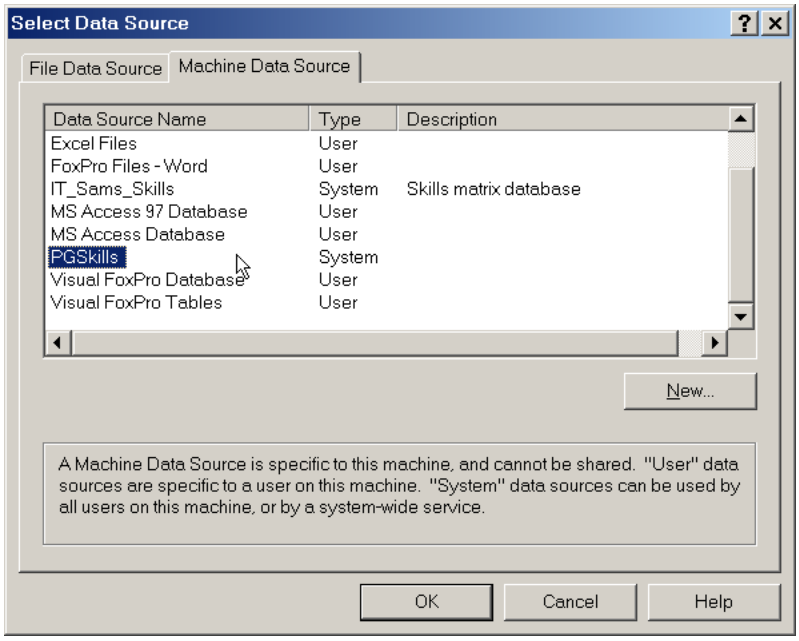
4.2 Right-click in the Tables Window and Select “Link Tables”



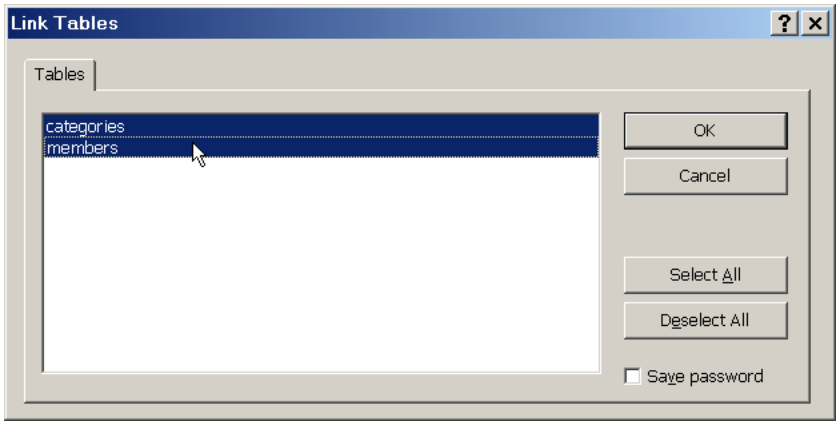
4.3 Select “ODBC Databases”



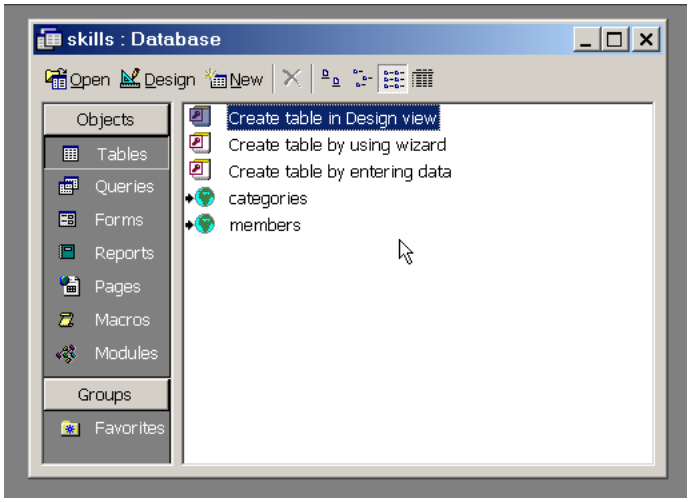
4.4 Select the PostgreSQL Data Source



4.5 Select All Tables You Wish to Include



4.6 Done!


skills : Database
Open
Design
New
Objects
Tables
Queries
Forms
Reports
Pages
Macros
Modules
Groups
Favorites
Create table in Design view
Create table by using wizard
Create table by entering data
categories
members