Installing SQL Anywhere on Linux ARM

Created by Former Member, last modified by Jeff Albion on Sep 18, 2014

Purpose

This tutorial intends to serve as an introduction to running SQL Anywhere on Linux ARM platforms. The steps will be demonstrated on the Raspberry Pi, but should be easily modified to run on other Linux ARM based computers. It is assumed that the reader already has a Linux ARM installation, and is familiar with the Linux operating system and ARM processors. For readers new to these platforms, it is recommended to review the Raspberry Pi Quick Start Guide. The OS distro used in this tutorial is Raspbian.

Overview

This tutorial will cover installation of SQL Anywhere on the Raspberry Pi, setting up necessary environment variables, and testing the installation using the demo database.

Install SQL Anywhere

To begin, move the installer archive over to the Raspberry Pi (hereafter RPi). This can be done by placing the file onto a flash drive, an sd card, or even SFTP'ing the file over to the RPi (SSH and SFTP can be enabled through raspi-config that runs during the first boot-up of Raspbian).

Once the installer is available on the RPi, it will need to be extracted through gzip and tar. To extract the installer, execute "gzip -d *.tar.gz" and "tar xvf *.tar" with the appropriate filename. This will create a directory of either ga1600 or ebf1600. Change to the resulting directory and execute the setup script "./setup". Follow the setup instructions as would be done on a normal Linux/Unix installation.

Optional: After installing SQL Anywhere, add the necessary environment variables at boot up to use the SQL Anywhere binaries. To do this, add the following lines to the end of the /etc/bash.bashrc file:

```
# /opt/sqlanywhere16 is the default installation path
# change this below if not using the default path
export PATH="$PATH:/opt/sqlanywhere16/bin32"
export LD_LIBRARY_PATH="$LD_LIBRARY_PATH:/opt/sqlanywhere32/lib32"
```

If the device is to be used strictly ad-hoc, the environments variables (and others) can be established in a single terminal session by executing "source /opt/sqlanywhere16/bin32/sa_config.sh".

Test the Installation

Next up, it is time to test the installation to see if things are working and set up correctly. This should be as simple as executing "dbsrv16 -v"

```
root@raspberrypi:~# dbsrv16 -v
16.0.0.1972
```

However, if the environment isn't set up correctly, the following errors may be encountered:

```
root@raspberrypi:~# dbsrv16 -v
-bash: dbsrv16: command not found
root@raspberrypi:~# dbsrv16 -v
dbsrv16: error while loading shared libraries: libdbserv16_r.so: cannot open shared object file: No such file or directory
```

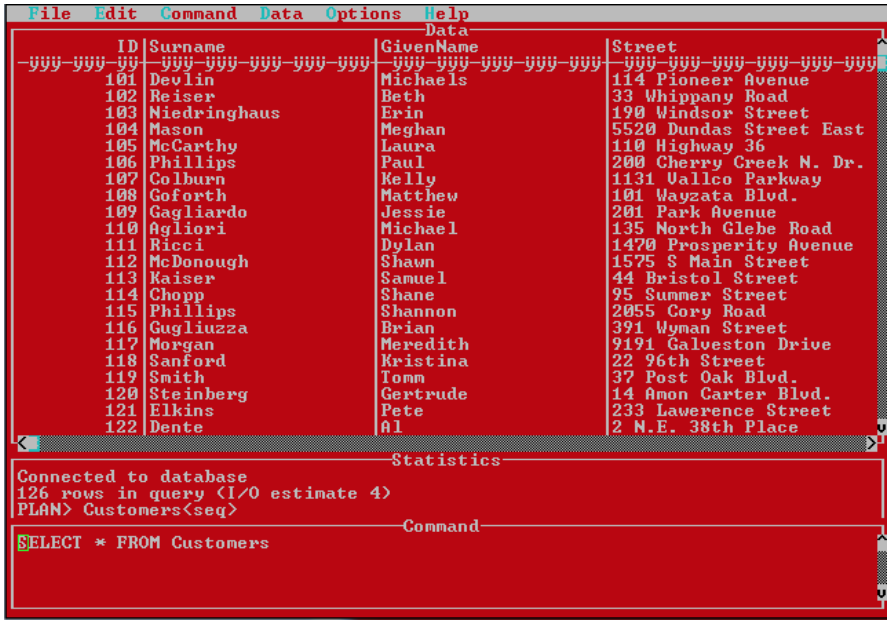
These errors indicate that either the PATH is set up incorrectly (can't find dbsrv16), or that LD_LIBRARY_PATH is set up incorrectly (error loading shared libraries). The "env" command can be used to see if either are set and if either contain typos. These should be remedied before continuing, either by ensuring they are set at startup correctly, or by executing the configuration scripts (e.g. "source /opt/sqlanywhere16/bin32/sa_config.sh") for the terminal session.

Start and Connect to the demo Database

If the server starts up, the next thing to try is starting the demo database. Make a scratch directory for the database then copy it from its installed location at /opt/sqlanywhere16/demo.db and start the database.

```
root@raspberrypi:~# mkdir dbscratch
root@raspberrypi:~# cd dbscratch
root@raspberrypi:~/dbscratch# cp /opt/sqlanywhere16/demo.db .
root@raspberrypi:~/dbscratch# dbspawn -f dbsrv16 -n mydemo -o server.out demo.db
```

dbspawn will report an error if the database is unable to be started. If there are no errors, however, it should now be possible to connect to the database. Connectivity and administration on this platform can be done with dbisqlc (the Java version of Interactive SQL and Sybase Central are not available on this platform, but can be used remotely for administration). To start dbisqlc, simply execute "dbisqlc". In the resulting interface, use the default username "dba" and password "sql" to connect to the demo database.



Once dbisqlc is up, execute a "SELECT * FROM Customers". If this command returns a result set, the demo database is running and responding successfully.

Related Content

Related Documents

Related SAP Notes/KBAs