

SuperSQL 導入マニュアル

～Linux 編～

2011.06.16

五嶋 研人

目次

1. SuperSQL スタンダード版 導入手順.....	3
1.1 PostgreSQL.....	3
1.2 Java	5
1.3 JDBC	6
1.4 SuperSQL(スタンダード版).....	8
2. SuperSQL Ajax 版 導入手順.....	10
2.1 PostgreSQL.....	10
2.2 Java	10
2.3 JDBC	10
2.4 Java 実行環境.....	10
2.5 Tomcat6.....	11
2.6 SuperSQL(Ajax 版).....	12
3. SuperSQL Web アプリケーション版 導入手順	16
3.1 PostgreSQL.....	16
3.2 Java	16
3.3 JDBC	16
3.4 Java 実行環境.....	16
3.5 Tomcat6.....	16
3.6 Invoker サブレット	17
3.7 SuperSQL(Web アプリケーション版)	18

本書に示されているコマンドは、「Ubuntu (11.04)」におけるコマンドである。
他の OS (Debian 系) の場合は、「sudo」を付けずに実行すると同じ処理を行うことができる。

※Debian 系以外の Linux ディストリビューションでは、コマンドが異なる場合がある。

例) パッケージ管理コマンド

Debian 系 : apt-get

Redhat 系 : yum

また、OS やバージョンの違いにより、ファイルの場所やディレクトリの構成に違いがあるため、注意が必要である。

1. SuperSQL スタンダード版 導入手順

1.1 PostgreSQL

(1) PostgreSQL のインストール (※既にインストール済みの場合は不要)

Synaptic パッケージマネージャで「postgresql-8.4」をインストール
または、

```
sudo apt-get install postgresql-8.4
```

(2) ユーザーの追加・パスワードの設定

```
$ sudo useradd p
```

```
$ sudo passwd p
```

```
p
```

```
p
```

(3) PostgreSQL 起動

```
sudo /etc/init.d/postgresql start
```

(4) データベース接続ユーザーの作成

```
# sudo su - postgres
```

```
$ psql -l
```

```
$ createuser p
```

```
n
```

```
y
```

```
n
```

```
$ exit
```

```
# sudo usermod -d /home -U p
```

※以下は Ubuntu のみの設定

```
sudo 権限を p にも与える (p も sudo を使えるようにするため)
```

```
sudo su -
```

```
sudo chmod 755 /etc/sudoers
```

```
vi /etc/sudoers
```

```
-----  
root    ALL=(ALL) ALL
```

```
p      ALL=(ALL) ALL ←許可したいユーザー名を追加
```

```
-----  
chmod 440 /etc/sudoers
```

```
su - 一般ユーザ
```

(5) データベース作成

```
# su - p
```

```
p
```

```
$ createdb db
```

```
$ psql -l
```

(6) サンプルデータの入力

```
$ psql db
```

```
CREATE TABLE company (
```

```
  co_id integer,
```

```
  会社名 varchar (20),
```

```
  業種 id integer
```

```
);
```

```
INSERT INTO company VALUES (1,'Company1',5);
```

```
CREATE TABLE T社員 (
```

```
  社員番号 integer,
```

```
  氏名 varchar (20),
```

```
  所属 varchar (20),
```

```
  loginname varchar (8),
```

```
  psw varchar (8),
```

```
  co_id integer
```

```
);
```

```
INSERT INTO T社員 VALUES (881001,'宮田','生産技術課','miyata','miyata',1);
```

```
INSERT INTO T社員 VALUES (901111,'中村','設計課','nakamura','nakamura',1);
```

```
INSERT INTO T社員 VALUES (932020,'奥秋','資材課','okuaki','okuaki',1);
```

```
INSERT INTO T社員 VALUES (981234,'小林','生産技術課','kobayasi','kobayasi',1);
```

```
INSERT INTO T社員 VALUES (985555,'樋谷','人事課','tuchiya','tuchiya',1);
```

```
INSERT INTO T社員 VALUES (990001,'相澤','経営管理課','aizawa','aizawa',1);
INSERT INTO T社員 VALUES (991111,'小林','経営管理課','tp001','tp001',1);
```

```
¥q
exit
```

(7) 「ident」を「trust」へ変更

```
sudo vi /etc/postgresql/8.4/main/pg_hba.conf
```

[local]と[host]の「ident」を「trust」へ変更（以下のように 3箇所）

```
-----
# Database administrative login by UNIX sockets
local  all          postgres          trust

# TYPE  DATABASE  USER  CIDR-ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local  all          all          trust
# IPv4 local connections:
host   all          all          127.0.0.1/32  trust
-----
```

(8) 設定の再読み込み

```
sudo /etc/init.d/postgresql restart
sudo /etc/init.d/postgresql reload
```

1.2 Java

(1) Java (openjdk-6-jdk) のインストール（※既にインストール済みの場合は不要）

Synaptic パッケージマネージャで「Java (openjdk-6-jdk)」をインストール
または、

```
sudo apt-get install openjdk-6-jdk
```

1.3 JDBC

(1) JDBC のインストール (※既にインストール済みの場合は不要)

Synaptic パッケージマネージャで「libpg-jdbc (postgresql-jdbc)」をインストール
または、

```
sudo apt-get install libpg-jdbc
```

(2) クラスパスの設定

```
sudo vi /etc/profile
```

```
-- ↓ 一番下に追加-----
```

```
CLASSPATH=./usr/share/java/postgresql-jdbc3.jar
```

```
export CLASSPATH
```

(3) 変更を反映させる

```
source /etc/profile
```

(4) 簡単な JDBC を利用する java プログラムの実行

以下、p にて行う

```
su - p
```

```
sudo gedit a.java
```

```
import java.sql.*;

public class a {
    public static void main (String args[]) {
        Connection db = null; // DB 接続オブジェクト
        Statement st = null; // SQL 文オブジェクト
        ResultSet rs = null; // 問合せ結果オブジェクト

        String url = "jdbc:postgresql:db"; // URL
        String usr = "p"; // ユーザ名
        String pwd = ""; // パスワード

        try {
```

```

// JDBC ドライバのロード
Class.forName("org.postgresql.Driver");

// データベース接続
System.out.println("Connecting to " + url);
db = DriverManager.getConnection(url, usr, pwd);
st = db.createStatement();

// 表の作成
st.executeUpdate("create table helloTbl (item varchar(16))");

// データの登録
st.executeUpdate("insert into helloTbl values ('Hello world!')");
st.executeUpdate("insert into helloTbl values ('ようこそ! (^)/')");

// データの検索
rs = st.executeQuery("select * from helloTbl");
if (rs != null) {
    while (rs.next()) {
        String item = rs.getString("item");
        System.out.println(item);
    }
}
rs.close();

// 表の削除
st.executeUpdate("drop table helloTbl");

// データベース切断
st.close();
db.close();

// エラー処理
} catch (Exception ex) {

```



```
        ex.printStackTrace();
    }
}
}
```

コンパイル

```
sudo javac a.java
```

実行

```
java a
```

以下のメッセージが表示されれば、OK

```
-----
Connecting to jdbc:postgresql://localhost/db
Hello world!
ようこそ! (^)/
-----
```

1.4 SuperSQL(スタンダード版)

```
cd /home
```

(1) 解凍

```
tar zxvf standalone.tar.gz
```

```
jar xf /home/standalone/supersql.jar
```

//META-INF、supersql の2フォルダが作成される

(2) supersql.jar, pdflib.jar, postgresql.jar を /usr/local/bin/ へ配置

```
sudo cp /home/standalone/supersql.jar /usr/local/bin/.
```

```
sudo cp /home/standalone/lib/jdbc/postgresql.jar /usr/local/bin/.
```

```
sudo cp /home/standalone/lib/pdflib/pdflib.jar /usr/local/bin/.
```

```
chmod 777 /usr/local/bin/*
```

(3) クラスパスの設定

```
sudo vi /etc/profile
```

-- ↓ 一番下に追加 --

```
CLASSPATH=./usr/share/java/postgresql-jdbc3.jar:/usr/local/bin/supersql.jar:/usr/local/bin/postgresql.jar:/usr/local/bin/pdflib.jar
export CLASSPATH
```

変更を反映させる

```
source /etc/profile
```

(4) ホームディレクトリにコンフィグファイル (.newssql) を作成

```
sudo vi /home/test.newssql
```

```
user=p
host=localhost
db=db
outdir=/usr/local/bin/
```

```
sudo chmod 777 /home/test.newssql
```

(5) テストファイル (.sql) の作成

```
sudo vi /home/t.sql
```

```
GENERATE HTML [
  s.氏名 ,
  { { "ログイン名", s.LoginName }! { "パスワード", s.PSW } }!
  s.所属
]!
FROM   t 社員 s
```

```
sudo chmod 777 /home/t.sql
```

(6) 実行 (HTML ファイルの作成)

```
sudo java supersql.FrontEnd -c /home/test.newssql -f /home/t.sql -d /home -o t.html
```

確認 ブラウザ：/home/t.html

2. SuperSQL Ajax 版 導入手順

2.1 PostgreSQL

1.1 と同様

2.2 Java

1.2 と同様

2.3 JDBC

1.3 と同様

2.4 Java 実行環境

(1) Java 実行環境 (sun-java-jdk6) をインストール(※既にインストール済みの場合は不要)

```
sudo add-apt-repository "deb http://archive.canonical.com/ lucid partner"
```

```
sudo apt-get update
```

```
sudo apt-get install sun-java6-jdk
```

```
dpkg -l | grep sun-java6 | less
```

```
sudo update-alternatives --config java
```

```
java -version
```

(2) クラスパスの設定

```
sudo vi /etc/profile
```

```
-- ↓ 追加-----
```

```
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

```
CATALINA_HOME=/usr/share/tomcat6
```

```
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
```

```
CLASSPATH=$CLASSPATH:$JAVA_HOME/lib/tools.jar
```

```
export JAVA_HOME CATALINA_HOME PATH CLASSPATH
```

```
source /etc/profile
```

2.5 Tomcat6

(1) Tomcat6 をインストール (※既にインストール済みの場合は不要)

```
sudo apt-get install -y tomcat6 tomcat6-common tomcat6-docs tomcat6-admin
tomcat6-examples tomcat6-user
```

(2) "Tomcat6"の編集

```
sudo vi /etc/default/tomcat6
```

```
TOMCAT6_USER=tomcat6           //変更
JAVA_HOME=/usr/lib/jvm/java-6-sun //変更・追加
CATALINA_BASE=/var/lib/tomcat6  //追加
```

(3) "tomcat-users.xml"の編集

```
sudo vi /etc/tomcat6/tomcat-users.xml
```

```
<tomcat-users>
  <role rolename="manager"/>
  <user username="ユーザ名" password="パスワード" roles="manager"/>
    //ユーザ名・パスワードは任意
</tomcat-users>
```

(4) Tomcat 起動

```
sudo /etc/init.d/tomcat6 start
```

確認 : <http://localhost:8080/>

確認 : <http://localhost:8080/manager/html>

2.6 SuperSQL(Ajax 版)

(1) /etc/tomcat6/htdocs へ解凍

```
sudo mkdir /etc/tomcat6/htdocs
```

(2) "ssqlajax.js"の編集

【重要】 修正

```
sudo gedit /etc/tomcat6/htdocs/for_ajax/ssqlajax.js
```

1. バックスラッシュ(¥)が全角(\\)になっているので、半角へ変更
2. UTF-8以外の文字コードで作成されているため、UTF-8へ変更

```
sudo gedit /etc/tomcat6/htdocs/for_ajax/ssqlajax.js
```

```
[Ctrl]+[Shift]+[S]
```

```
エンコーディング:UTF-8 行末:Unix/Linux
```

```
[保存]
```

```
sudo gedit /etc/tomcat6/htdocs/for_ajax/ssqlajax.js
```

```
////////////////////////////////////  
//  
// 各種設定  
//  
////////////////////////////////////  
  
// embed servlet URL  
var servletURL =  
    "http://localhost:8080/for_ajax/servlet1/supersql.embed.EmbedServlet";  
  
// form servlet URL  
var formURL =  
    "http://localhost:8080/for_ajax/servlet2/supersql.form.FormServlet4";  
  
// config file URL  
var configfileURL = "http://localhost:8080/for_ajax/config.ssql";  
  
:
```

(3) コンフィグファイル (config.ssql) の作成

```
sudo vi /etc/tomcat6/htdocs/for_ajax/config.ssql
```

```
user=p  
host=localhost  
db=db  
outdir=/etc/tomcat6/htdocs/for_ajax
```

(4) "web.xml" の作成

```
sudo vi /etc/tomcat6/htdocs/for_ajax/WEB-INF/web.xml
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
  
<!DOCTYPE web-app  
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
  "http://java.sun.com/dtd/web-app_2_3.dtd">  
  
<web-app>  
  <servlet>  
    <servlet-name>EmbedServlet</servlet-name>  
    <servlet-class>supersql.embed.EmbedServlet</servlet-class>  
  </servlet>  
  <servlet>  
    <servlet-name>FormServlet</servlet-name>  
    <servlet-class>supersql.form.FormServlet4</servlet-class>  
  </servlet>  
  
  <servlet-mapping>  
    <servlet-name>EmbedServlet</servlet-name>  
    <url-pattern>/servlet1/*</url-pattern>  
  </servlet-mapping>  
  <servlet-mapping>  
    <servlet-name>FormServlet</servlet-name>  
    <url-pattern>/servlet2/*</url-pattern>  
  </servlet-mapping>  
</web-app>
```

Tomcat 再起動

```
sudo /etc/init.d/tomcat6 restart
```

(5) テストファイル(.sql)の作成

<blind エフェクト>

```
sudo vi /etc/tomcat6/htdocs/for_ajax/searchindex.sql
```

```
GENERATE HTML {
  { sinvoke ("▼社員から探す"@{class=searchindex},
            file="http://localhost:8080/for_ajax/typelist.sql",
            att=co.co_id)
    @ {dispdiv=type_list,in=blind,out=blind}
  }!
  { embed (file="http://localhost:8080/for_ajax/typelist.sql")
    @ {class=bytype,divid=type_list,status=hidden}
  }
} @ {cssfile=demo.css}
FROM company co;
```

```
sudo vi /etc/tomcat6/htdocs/for_ajax/typelist.sql
```

```
GENERATE HTML {
  { [s.氏名@{class=typename}]! } @ {tableborder=0}
} @ {cssfile=demo.css}
FROM t 社員 s;
```

HTML の生成

```
java supersql.FrontEnd -ajax -c /etc/tomcat6/htdocs/for_ajax/config.ssql -f
/etc/tomcat6/htdocs/for_ajax/searchindex.sql -d /etc/tomcat6/htdocs/for_ajax -o
searchindex
```

(6) WAR ファイルの作成・配備

WAR ファイルの作成

```
cd /etc/tomcat6/htdocs/for_ajax  
sudo chmod 777 *  
sudo chgrp -R tomcat6 .  
sudo jar -cvf for_ajax.war *
```

WAR ファイルの配備

ブラウザ：<http://localhost:8080/manager/html>

[WAR ファイルの配備]で、`/etc/tomcat6/htdocs/for_ajax/for_ajax.war` を配備

(7) 実行

ブラウザ：http://localhost:8080/for_ajax/servlet1/searchindex.html

3. SuperSQL Web アプリケーション版 導入手順

3.1 PostgreSQL

1.1 と同様

3.2 Java

1.2 と同様

3.3 JDBC

1.3 と同様

3.4 Java 実行環境

2.4 と同様

3.5 Tomcat6

2.5 と同様

3.6 Invoker サブレット

Tomcat6 Invoker サブレットの設定

(1) "web.xml"の編集

```
sudo gedit /etc/tomcat6/web.xml
```

以下の4箇所のコメントを外す

```
      :
<!-- ←
  <servlet>
    <servlet-name>invoker</servlet-name>
    <servlet-class>
      org.apache.catalina.servlets.InvokerServlet
    </servlet-class>
    <init-param>
      <param-name>debug</param-name>
      <param-value>0</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
  </servlet>
--> ←
      :
<!-- ←
  <servlet-mapping>
    <servlet-name>invoker</servlet-name>
    <url-pattern>/servlet/*</url-pattern>
  </servlet-mapping>
--> ←
      :
```

(2) "context.xml"の編集

```
<Context>
  ↓ 変更
<Context privileged="true">
```

(3) Tomcat 再起動

```
sudo /etc/init.d/tomcat6 restart
```

3.7 SuperSQL(Web アプリケーション版)

(1) /etc/tomcat6/htdocs へ解凍

```
sudo mkdir /etc/tomcat6/htdocs
```

(2) META-INF・WEB-INF の作成

[META-INF]の作成

```
sudo mkdir /etc/tomcat6/htdocs/web_ssql/META-INF/  
sudo cp /etc/tomcat6/htdocs/web_ssql/MANIFEST.MF  
    /etc/tomcat6/htdocs/web_ssql/META-INF/.
```

[WEB-INF]の作成

```
sudo mkdir /etc/tomcat6/htdocs/web_ssql/WEB-INF/  
sudo mkdir /etc/tomcat6/htdocs/web_ssql/WEB-INF/classes  
sudo mkdir /etc/tomcat6/htdocs/web_ssql/WEB-INF/lib  
sudo cp -r /etc/tomcat6/htdocs/web_ssql/supersql  
    /etc/tomcat6/htdocs/web_ssql/WEB-INF/classes/.  
sudo cp -r /etc/tomcat6/htdocs/for_ajax/WEB-INF/lib/*  
    /etc/tomcat6/htdocs/web_ssql/WEB-INF/lib/.
```

(3) コンフィグファイル (config.ssql) の作成

```
sudo vi /etc/tomcat6/htdocs/web_ssql/config.ssql
```

```
db=db2  
host=localhost  
user=p  
driver=postgresq  
optimizer=on  
invokeServletPath=http://localhost:8080/web_ssql/servlet/supersql.invoke.InvokeS  
ervlet2  
fileDirectory=http://localhost:8080/web_ssql
```

(4) テストファイル(.sql)の作成

<T 社員テーブルの Update>

```
sudo vi /etc/tomcat6/htdocs/web_ssql/update.ssql
```

```
GENERATE HTML
{ {"社員番号"}, {"氏名"}, {"所属"}, {"ログイン名"}, {"パスワード"}, {"co_id"} }!
[ { (asc1)s.社員番号@{pkey}, s.氏名, s.所属, s.loginname, s.psw,
    s.co_id }@{update}]!
FROM t 社員 s;
```

(5) 入り口のHTML (index.html) の作成

```
sudo gedit /etc/tomcat6/htdocs/web_ssql/index.html
```

```
<h1>Test Page ~index.html~</h1>

<a href=
"http://localhost:8080/web_ssql/servlet/supersql.invoke.InvokeServlet2?config=ht
tp://localhost:8080/web_ssql/config.ssql&query=http://localhost:8080/web_ssql/up
date.ssql">Update</a><br>
```

(6) WAR ファイルの作成・配備

WAR ファイルの作成

```
cd /etc/tomcat6/htdocs/web_ssql
```

```
sudo chmod 777 *
```

```
sudo chgrp -R tomcat6 .
```

```
sudo jar -cvf web_ssql.war *
```

WAR ファイルの配備

ブラウザ：<http://localhost:8080/manager/html>

[WAR ファイルの配備]で、/etc/tomcat6/htdocs/web_ssql/web_ssql.war を配備

(7) 実行

ブラウザ：http://localhost:8080/web_ssql/index.html