بسم الله الرحمن الرحيم

برمجة قواعد البيانات بالجافا

"Database in java"

برامج الـ Database تعتبر من أهم مصادر الدخل للمبرمجين والشركات البرمجيه (برامج محاسبيه ، برامج اداره وو..) ولها اقبال عالي جدا ، ولذلك على المبرمجين أن يدركوا كيفيه التعامل معها وعمل برامج يستطيعوا من خلالها أكل العيش ...

سنتحدث اليوم عن كل ما يتعلق ببرمجه تطبيقات قواعد البيانات بجافا ، لكن سنتبع طريقه جديده في طرح الدروس Intractive بدلا من طرح الدرس مره واحده وبشكل كبير وربما لن يستطيع المرء قرائته أو فهمه بشكل جيد ..

الموضوع سيكتب في عده جلسات ، وستحتوي كل منها على أمثله وشرح في موضوع ما في هذا النوع من البرمجه ، ولن نتحرك من نقطه الى أخرى الا في حال كان المتابع للموضوع استطاع تطبيقه بلا مشاكل ، وأظن تلك الطريقه أفضل خصوصا للمتابعين ، لأننا سنتحرك خطوه بخطوه ان شاء الله من الصفر ..

أيضا في نهايه الموضوع سيكون هناك مشروع لأي متابع في الموضوع وهو عمل قاعده كامله وربطها ببرنامج جافا (مثلا برنامج دليل هواتف أو برنامج لقاعده بيانات مركز تجاري أو أي مثال صغير) وستكون تحت اشرافي ان شاء الله ..

سنبدأ الليله بالحديث في بعض الأساسيات وسنستخدم قاعده بيانات أكسس MS-Access كمثال حتى اذا فهمنا الأساسيات نستطيع حينها تغيير القاعده ونستخدم أي قاعده أخرى وهنا التغيير في برنامج جافا سوف يكون في سطرين فقط !

اليوم ان شاء الله سيكون الموضوع الأول ..

حَسْبِيَ اللّهُ لا إِلَهَ إِلاَّ هُوَ عَلَيْهِ تَوَكَّلْتُ وَهُوَ رَبُّ الْعَرْشِ الْعَظِيمِ

حسبنا الله سيؤتينا الله من فضله إنا إلى الله راغبون

مقدمه :

=-=-=-=

منذ صدور أول نسخه جافا JDK 1.1 وهي تحتوي على API للوصول والتعامل مع قواعد البيانات العلائقيه ، هذه الAPI تسمى Java Database Connectivity وأختصارا JDBC .

هذه الAPI تستخدم للتعامل مع قواعد البيانات المختلفه ، لكن المشكله الرئيسيه تكمن في أن كل DBMS يختلف داخليا من الأخر ، بمعنى أن لكل منهم Format خاصه به وله API خاص فيه تختلف عن بقيه قواعد البيانات الأخرى DBMS . أكسس يختلف عن MySql وهي تختلف بدورها عن Oracle ..

لذلك واجهت SUN هذه المشكله وذلك بجعل JDBC يتحدث بلغه موحده ، ولكن سوف يتحدث الى "وسيط" Bridge ، هذا الوسيط خاص لكل DB وهو يفهم ويعرف كيف يتعامل مع الDB ، أي أن من يقوم بصناعه قاعده البيانات DBMS Vendor يقوم بعمل وسيط من خلاله تستطيع باستخدام JDBC التعامل معه .. بالطبع فإن كل من هذه الBridga خاصه بالقاعده نفسها ولكل قاعده وسيط خاص فيه يجب أن تقوم بتحميله من موقع صاحب الData Base .

بمعني أنك من خلال برنامج الجافا عندما تود الإتصال مع قاعده بيانات لأي كان ، يجب أن يتوفر هناك Bridge توفره هذه الجهه ، وستقوم أنت باعطائه تعليمات JDBC API ويقوم هذا الـ Bridge بفهمها وتحويلها الى لغه تفهمها قاعده البيانات المعينه ..

جميل ،، هذه الBridge تسمى Driver ، أي بمعني وسيط فقط يفهم أوامر منك ويقوم بتحويلها الى شكل تفهمه قاعده البيانات ..

شركه مايكروسوفت قد حلت قديما (من قبل جافا) هذه المشكله The Vendor Variation Problem من قبل وذلك باستخدم Open Database Connectivity وأختصارا ODBC . وكانت أي DBMS Vendor يوفر ODBC Driver لكي يستطيع مستخدمين ويندوز الوصول والتعامل مع قاعده البيانات ..

من هنا قامت sun بعمل Driver لهذا ODBC من خلاله تستطيع التعامل مع أي قاعده بيانات من انتاج ميكو ، وحتى أي قاعده بيانات لا يوجد لها Driver كما قرأت .. ولكن عليك ملاحظه أن استخدام JDBC-ODBC Driver هو فقط للغرض التجربه أو عندما لا يتوفر Driver للقاعده ، فقط غير ذلك وخاصه في التطبيقات التجاريه يجب أن تستخدم Driver للقاعده أفضل من JDBC-ODBC Driver ، والسبب أنك ستكون محصور في بيئه ويندوز فقط ..

ربما الكلام أعلاه غير مفهوم ولكن لا تقلق ، كلما نتقدم قليلا كلما ستفهم كثيرا ، وهنا تكمن متعه البرمجه ونظرياتها ، فقط أستمر في القرائه .

سنستخدم في هذه الجلسه قاعده بيانات أكسس وسوف نستخدم JDBC-ODBC Driver لكي نتعامل مع القاعده ، ولكن دائما في حال التعامل مع ODBC يجب أن نقوم بعمل ODBC Data Source (وهو من خلال لوحه التحكم لديك في الجهاز ، كما سنرى بعد قليل) ،، أيضا سنأخذ مثال بسيط لتشغيل أول قاعده بيانات وسوف يعمل في الشاشه السوداء console وسنتقدم باذنه تعالى رويداً رويداً ،،

ملاحظه أخيره وهي لغه Structure Query Langauge وأختصارا SQL ، وهي لغه تسمح لك بالتعامل مع قاعده البيانات (ادخال بيانات ، انشاء جداول ،استعلام عن قيمه ووو) . وفي دروسنا البسيطه هذه سوف نشرح بعض الاوامر الأساسيه في الموضوع لذلك لا تتطلب لديك معرفه بها ، لكن طبعا إن أردت الأحتراف وخاصه في تطبيقات قواعد البيانات يجب أن تلم بكل صغيره وكبيره في هذه اللغه ..

نقاط هذه الجلسه :

1) انشاء قاعده بيانات أكسس

2) عمل ODBC Data Source

3)خطوات الأتصال مع قاعده البيانات

4) تشغيل أول مثال للأتصال بالقاعده وادخال بيانات

نبدأ على بركه الله ،،

انشاء قاعده بيانات أكسس

=-=-=-=-=-=-=-=-=-=-=

هناك الكثير من اصدارات أكسس ،لكن لا تخف فجميعهم تستطيع التعامل معهم ،، نقوم الأن بعمل قاعده بيانات للمدرسه School تحتوي هذه القاعده على عده جدوال الأساتذه والطلاب والحسابات والكثير ، حاليا سوف نركز على جدول واحد في القاعده وليس على الجميع .. (تذكر هذه الدروس في ربط قاعده البيانات بجافا وليس مقدمه في تصميم قواعد البيانات!) ..

قم بفتح قاعده بيانات أكسس من قائمه file أختر new ، ثم من على الtap في اليمين أختر قاعده بيانات فارغه Blank Database .

ثم تخرج نافذه قم بكتابه اسم لقاعده البيانات وليكن School ثم أختر create .

الأن أختر create table in Desing View ، لكى تنشئ الجدول

وقم بعمل الحقول التاليه واجعلها جميعها text

ثم قم بعمل save وسيسألك عن اسم لقاعده البيانات أدخل Student ، ثم سيسألك عن هل تود انشاء مفتاح أساسي أجبه بلا no ، طبعا مثل هكذا جدول في تصميم القاعده يعتبر خاطئ ، ولكننا نستطيع التحكم بهذه العمليه من البرنامج ، أي أننا يمكن أن نصمم قاعده بيانات من غير أي علاقات (لا مفتاح أساسي، ولا مفتاح غريب) ويمكننا من البرنامج عمل كل شيء نريد والتحكم بادخال قيمه وعدم ادخالها وعرض ما نريد ..

أدخل على الجدول مره أخرى بالضغط على مرتين double click ثم أدخل أي بيانات :

الأن اغلق الجدول وأغلق القاعده وأغلق Access أيضا .. وهكذا أنتهينا من المرحله الأولى وهى انشاء قاعده بيانات أكسس .. (طبعا أغلب الأعضاء على علم بها ، ولكن حبيت أعرض الخطوات لكي أتذكر شكل القاعده قليلا لأني من زمن لم أتعامل معها ، بالاضافه الى أنه ربما أحد الأعضاء جديد بالمجال ولا يعلم شيء عن قاعده البيانات).

عمل ODBC Data Source

=-=-=-=-=-=-=-=-=-=-=

الأن بعدما قمنا بعمل القاعده School يجب أن نقوم بتسجيلها كODBC Data Source وبالتالي تستطيع الوصول للقاعده عن طريق الأسم الذي قمنا بتسجيله بها ، تابع الخطوات التاليه لكي نقوم بتسجيلها ،،

قم من خلال لوحه التحكم Control Panel بفتح Administrative Tools ثم أختر Data Sources (ODBC) وستظهر لك هذه النافذه وسوف يكون بها أسماء قواعد بيانات سابقه سجلت كODBC قمت أنت بها أو أستخدمها برنامج ما ، المهم قم بالضغط على زر Add في الtap المسى بUser DSN (أول تاب) ،

من القائمه الجديده ، أختر Driver do Microsoft Access كما هو موضح في الصوره أعلاه ثم أختر Finsih ،، مباشره سوف تظهر لك الصوره التاليه لكي تحدد الDataBase التي قمت بعملها ، أختر select ثم حدد مكان قاعده البيانات التي أنشأتها وعاده بالشكل الأفتراضي تكون قاعده بيانات أكسس موجوده في مجلد Mydocuments .

بعد أن تختار القاعده وتضغط OK سوف تظهر لك النافذه السابقه وأكتب في DSN اسم القاعده School (أو أي أسم أخر للوصول لها ولكن يفضل نفس اسم القاعده) ، خانه Description هي لشرح عن القاعده ولن نحتاجها ، أيضا تستطيع عمل اسم مستخدم وكلمه مرور لو أردت وذلك من خلال Advanced Option وضع ما تريد ، ولكن دعها الأن وقم بها في تجربتك الثانيه ، ثم اضغط OK وسترجع لأول نافذه وستجد ألأسم الذي كتبته موجود في النافذه في المنتصف اختر OK للإنهاء ..

خطوات الأتصال مع قاعده البيانات

=-=--=-=-=-=-=-=-=-=-=-=-=

التعامل مع قاعده البيانات في جافا ينحصر في الكائنات التاليه : Connection و Statment و ResultSet و ResultSetMetaData فقط .. في الحقيقه هذه الكائنات عباره عن interface تقوم الDriver التي تستخدمها بتطبيقها implmenet it .

<u>للتعامل مع قاعده البيانات سوف نحتاج الى 7 خطوات :</u>

1) تحميل الDriver

2) عمل اتصال مع قاعده البيانات

3) استخدام Connection للحصول على Statment

4) من خلال Statament تستطيع الاستعلام من القاعده ، أو ادخال بيانات للقاعده

5) في حال استعلمت من القاعده الناتج سوف يكون ResultSet يحتوي على النتيجه ، أما اذا كنت أدخلت بيانات (تعديل ، ادخال ، حذف) فالناتج هو عدد السطور التي تأثرت بالعمليه

6) كرر الخطوات 4 و 5 متى أردت ذلك ..(استعلام أو تعديل) .

7) أغلق الأتصال close

نبدأ بشرح الخطوات بشكل مبسط ،، أولا تحميل الDriver .. لكي تجلب هذا الDriver يجب أن تشير الى موقعه لديك ، وبما أن لكل قاعده بيانات Driver خاص بها اذا لكل قاعده بيانات موقع Driver يختلف عن الأخر .. بما أننا نتعامل الأن مع قاعده بيانات أكسس ونستخدم JDBC-ODBC Driver فموقع هذا الDriver يكون :

sun.jdbc.odbc.JdbcOdbcDriver

كما ذكرت كل قاعده بيانات لديها موقع Driver يختلف عن الأخر ، سنشاهد هذا ان شاء الله في الجلسات القادمه عندما نستخدم MySql أو Sqlite .. حاليا لكي نجلب هذا الDriver نقوم باستخدام الداله forName الموجوده في الكلاس Class (هذه الداله static بمعني أنك تستطيع ندائها من غير عمل كائن من الكلاس) :

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver”);
```

الخطوه الثانيه وهو عمل الأتصال مع القاعده ، وذلك من خلال الداله getConnection الموجوده في الكلاس DriverManager (هذه الداله static ايضا) وتستقبل ثلاثه قيم ، القيمه الأولى هي عنوان لقاعده البيانات URL والثانيه اسم المستخدم والثالث كلمه المرور (تذكر نحن لم نضع اسم مستخدم وكلمه مرور ، الا لو وضعتها فلا تخف لا توجد مشاكل) ..

تحديد الURL بالقاعده يكون بالشكل التالي :

```
jdbc:<sub-protocol>:<data-source<
```

بدون الأقواس الصغيره ، jdbc هذه ثابته وتعني أننا نستخدم JDBC API ، وsub-protocol هي لتحديد الDriver -نوعه- ، والdataSource لتحديد اسم القاعده ومسارها ورقم البورت لو تطلب ذلك ... نحن حاليا نتعامل مع odbc وهو سيكون sub-protocol أما data-source سوف يكون School لأننا سجلنا القاعده قبل قليل بهذا الأسم في ODBC Data Source وذكرنا أنه يفضل أن نسجل الأسم بنفس اسم القاعده ، في حال قمت بتسجليه باسم مختلف ضع الأسم هنا .. بهذا يكون لدينا URL بالشكل التالي :

jdbc:odbc:School

ويكون سطر الأتصال بهذا الشكل :

connection = DriverManager.getConnection("jdbc:odbc:School؛("""،""،"

connection هو كائن من Connection (وسوف أستخدم طريقه التسميه هذه للتوضيح ، يعني اسم الكائن هو نفسه اسم الكلاس فقط الحرف الأول هو حرف صغير) ..

اذا كانت قاعده البيانات موجوده في مكان بعيد ، سيرفر أخر ، فيجب أن تحدد ذلك الموقع :

connection = DriverManager.getConnection("jdbc:odbc://AnyServer.File.com/School """،""،("")؛

الخطوه الثالثه وهي لانشاء Statement من خلال الكائن connection :

statement = connection.createStatement؛()

الخطوه الرابعه وهي للتعامل مع القاعده سواء بادخال بيانات أو الأستعلام عن البيانات وهنا سوف نستخدم لغه SQL .. التعامل يكون مع الكائن statement وفيه طريقتين (دالتين) الأول للأستعلام وهي executeQuery والثانيه للتعديل (اضافه،حذف،تعديل) وهي executeUpdate هذه الدالتين تستقبل string فيه كود SQL للعمليه التي تريد ..

مثلا نريد تحديد جميع الصفوف في الجدول Student ، يكون لدي أمر الSQL التالي :

SELECT * FROM Student

(طبعا يمكن كتابه الاوامر بالأحرف الكبيره أو الصغيره لا يفرق not case sensitive ولكن أفضل الكبيره Captial Letter ، أتحدث هنا عن أوامر SQL فقط ، لكن اسم الجدول وأسماء الحقول يجب أن تكون مطابقه تماما والا فسوف يحصل Exception) .

أيضا نفضل وضع كود SQL في متغير string ثم تمرير هذا المتغير للداله بدلا من كتابه الأمر مباشره في الداله ..

مثال على الأستعلام من القاعده :

```
String select = "SELECT * FROM Student؛ "

ResultSet resultSet = statement.executeQuery(select؛(
```

```
String select = "SELECT number,firstName FROM Student؛ "
ResultSet resultSet = statement.executeQuery(select؛(
```

```
String select = "SELECT firstName FROM+ "
"   WHERE number >= 10+ "
"   AND number <=100+ "
"   ORDER BY number DESC؛ "
ResultSet resultSet = statement.executeQuery(select؛(
```

```
String select = "SELECT * FROM Student WHERE firstName ='wajdy؛'"
ResultSet resultSet = statement.executeQuery(select؛(
```

هذه كلها أمثله على الأستعلام من القاعده باستخدام الداله executeQuery ،والناتج سوف يكون ResultSet يحتوي على نتيجه الأستعلام ..

الكائن من ResultSet يحتوي على كميه كبيره من الدوال لكي تستخرج الناتج من الأستعلام ، على العموم حاليا سوف نتعامل مع الداله next وهي تخرج لي السطر التالي من ناتج الإستعلام المخزن في ResultSet (بمعني أن الناتج الموجود في resultset هو سطور كل

سطر يحتوي على المعلومات مثل قاعده البيانات بالضبط، لكن عليك ملاحظه أن عدد السطور قد يختلف هنا مع قاعده البيانات ، مثلا استعلمت عن طالب برقم ما ، الناتج في حال وجد طالب بهذا الرقم هو سطر واحد ، مثلا لو أستعلمت عن جميع الطلاب فالناتج في ResultSet هو نفس عدد السطور الموجوده في القاعده) .

نقطه أخرى ، وهي أن الكائن ResultSet (الذي يحتوي على السطور الناتجه من الأستعلام) ، يبدأ من السطر ما قبل الأول ، أي أننا يجب أن نستدعي الداله next قبل الوصول لأي قيمه موجوده في السطر (حتى نكون في السطر الأول) .. وهذه الداله ترجع قيمه منطقيه true|false بناء على هل يوجد سطر تالي أم لأ ..

الأن لكي نصل لقيمه معينه في سطر ، نستخدم الداله getXXX وهذه الداله لها شكلين Overloading الأول يستقبل Index يمثل لي رقم الحقل في القاعده -يبدأ الترقيم من 1- ، أما الشكل الثاني فيستقبل String يمثل اسم الحقل في القاعده ..

هذه الدوال :

int getInt (String <columnName(<

int getInt (int <columnIndex(<

String getString (String <columnName(<

String getString (int <columnIndex(<

وهناك نسخ لأنواع المتغيرات الأخرى Double,Flaot,Date .. فقط غير XXX بنوع البيانات الذي تريد ..

مثال على كيفيه معالجه الناتج من الأستعلام :

```
while ( resultSet.next( ()
   }
   System.out.printf("%-8s\t",resultSet.getString(1؛((
   System.out.printf("%-8s\t",resultSet.getString(2؛((
   System.out.printf("%-8s\t",resultSet.getString(3؛((
   System.out.printf("%-8s\t",resultSet.getString(4؛((
   System.out.println؛()

{
```

هنا لاحظ أننا قمنا بعمل حلقه (سوف تمر على جميع السطور الموجوده في ResultSet الى أن نصل للنهايه وتصبح قيمه Next خاطئه) . داخل جسم الداله استخدمنا الداله getString بالنسخه التي تستقبل رقم الindex (تذكر تبدأ من 1) .. وكل مره سنطبع الحقل الأول ، والثاني ، والثالث، والرابع .. ثم نطبع سطر فارغ .. يمكن أن تلاحظ أستخدام الداله printf بدلا من println وهي فقط لتنظيم فقط لا أكثر (وهي شبيه بـprintf في لغه السي وميزتها تكون في اضافه متغيرات وقيم ثابته في مكان واحد ، ثم تقوم فيما بعد بتحديد المتغيرات) .. اذا لم تعجبك يمكنك أن تغيرها الى printf وتطبع الناتج من الداله getString (ليس موضوعنا printf الأن ،وعلى ما أذكر هي تعمل في نسخ JDK 5 وما فوق) ..

يمكن أن تستخدم الشكل الثاني من الداله وتمرر له اسم الحقل مثال على الحقل الأول :

```
System.out.println( resultSet.getString("number؛( (")
```

وهكذا لبقيه الحقول -تستخدم الأسم ولكن بنفس الحروف بالضبط-

الأن تكرر هذه الخطوات كما تريد لكي تستعلم من القاعده ..

هذا بالنسبه للإستعلام من القاعده ، نأتي الأن لكيفيه الأدخال أو التعديل أو الحذف من قاعده البيانات .. وذلك باستخدم الداله executeUpdate . وباستخدام أوامر SQL المناسبه وهل سهله وواضحه من الأسم ، الأمثله القادمه هي لعمليات الإضافه والتعديل والحذف :

```
String insert = "INSERT INTO Student VALUES('123','Wajdy','Essam','34242424؛ "('

int result = statement.executeUpdate(insert؛(
```

```
String update = "UPDATE Student SET firstName='Ahmed' lastName = 'omer+ "'

"   WHERE number='1؛ "'

statement.executeUpdate(update؛(
```

```
String delete = "DELETE FROM Student WHERE number > 100؛ "

int result = statement.executeUpdate(delete؛(
```

برجاء ملاحظه أن أي قيمه string داخل هذا الstring يجب أن تحاط ب ' ' ، يجب أن تتأكد منها حتى لا تقع في مشاكل ..

أخيرا وهي اغلاق الإتصال مع القاعده ،وهنا تقوم بغلق الconnection والstatement وتغلق الStatement في الأول وبعدها تغلق الإتصال :

```
statement.close؛()

connection.close؛()
```

أنتهت المرحله الثالثه وهي خطوات الأتصال مع القاعده ...أخر ملاحظه وهي التعامل مع الExceptionحيث أن التعامل مع هذه الكلاسات والتي تتعامل مع جمل SQL قد تولد استثناء ويجب أن نتعامل معه لأنه من نوع Checked Exception .. لذلك يجب أما عمل try و catch للإستثناء من نوع SQLException ، أو عمل throws لهذا الإستثناء .. أيضا هناك استثناء أخر وهو ClassNotFoundException وهو ينتج في حال لم يتم العثور على الDriver (أول خطوه من خطوات الإتصال) ..

أخيرا جميع هذه الكلاسات وحتى الinterface والExceptionموجودين في الحزمه java.sql . لذلك يجب عمل Import لهذه الحزمه بالكامل (باستخدام *) أو تحديد الكلاسات التي نريدها وهو ما سأقوم به ..

تشغيل أول مثال للأتصال بالقاعده وادخال بيانات

نأخذ الأن مثال بسيط وذلك للأستعلام عن قاعده البيانات التي أنشأناها سابقا ، فقط نريد استخراج تلك المعلومات وطباعتها على الشاشه ..

سنطبق كل الخطوات التي ذكرناها أعلاه ،، وهذا هو المثال :

```
//SudanCS
//Demo.java

import java.sql.Connection؛
import java.sql.Statement؛
import java.sql.ResultSet؛
import java.sql.DriverManager؛
import java.sql.SQLException؛

public class Demo
}
   private static final String userName؛ "" =
   private static final String password؛ "" =
   private static final String URL = "jdbc:odbc:School؛"
```

```
   private static final String DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver؛ "

   private static Connection connection؛
   private static Statement  statement؛
   private static ResultSet  resultSet؛

   public static void main (String arg} ([]
      try}
         Class.forName(DRIVER؛(
         connection                                                    =
DriverManager.getConnection(URL,userName,password؛(
{
      catch(ClassNotFoundException e} (
         System.err.println("unable to load database Driver؛("!
         System.exit(1؛(
{
      catch(SQLException e}(
         System.out.println("Cannot connect to database؛("!
         System.exit(1؛(
{

      try}
```

```
        statement = connection.createStatement؛()

        String select = "SELECT * FROM Student؛ "
        resultSet = statement.executeQuery(select؛(
{
    catch (SQLException e} (
        System.err.println("cannot execute query؛("!
        System.exit(1؛(
{

    try}
        System.out.println؛()
        while ( resultSet.next} ( ()
            System.out.println("Number = " + resultSet.getString(1؛((
            System.out.println("First Name = " + resultSet.getString(2؛((
            System.out.println("Last  Name = " + resultSet.getString(3؛((
            System.out.println("Tel Number = " + resultSet.getString(4؛((
            System.out.println؛()
{
{
    catch (SQLException e} (
        System.err.println("error in retrieving data؛("!
```

```
        e.printStackTrace؛()
        System.exit(1؛(
{

    try}
//          close
        statement.close؛()
        connection.close؛()
{
    catch (SQLException e} (
        System.err.println("unable to disconnect؛("!
        System.exit(1؛(
{
{
{
```

لاحظ أننا وضعنا كل أمر تقريبا في جمله try catch والسبب حتى عندما يحصل خطأ يكون أوضح مالذي حصل ، ويمكن أن تضع جميع الجمل في try واحده فقط ، لكني لا أحبذ ذلك كبدايه ..

لاحظ أيضا أننا نستخدم string في البدايه وخزنا به اسم المستخدم (string فاضي "") ونفس الأمر بالنسبه للباسورد ، أيضا هناك متغيرات String خزنا بها الDriver و URL .. وجعلنا جميع المتغيرات static لأننا سوف نستخدمها في الداله main (وهي static) ولا يمكن أن نستخدم متغير عادي داخل داله static الا بجعله كذلك ..

صدق أو لا تصدق ، هذا المثال فقط بتغيير سطرين يمكن أن يعمل في أي قاعده بيانات أخرى ! فقط نغير الDriver وموقع URL وسيعمل المثال 100% .. هذه هي ميزه JDBC ..

صدق أو لا تصدق أنت الأن أصبحت تدرك جيدا أساسيات الربط بين قاعده البيانات وبرنامج الجافا ،، وبقليل من الأمثله سوف تستطيع أحتراف التعامل وتنشيء تطبيقات أحترافيه commerical ..

سنتوقف هنا وأتمنى أي واحد يجرب وأن يصل لهذا النقطه وسوف يكون المخرج بالشكل التالي :

في حال أنجزت كل شيء تماما ، يمكنك أن تجرب تتعامل مع الدوال التي ذكرنا أنها تقوم بالتعديل والأضافه حاول أن تدخل قيم في القاعده وجرب ...

المره القادمه باذنه تعالى نغوص أكثر وأكثر في الموضوع ، هناك الكثير من المفاجئات بانتظارك

أي سؤال أو استفسار أو ملاحظه يرحب بها ...

والسلام عليكم ورحمه الله وبركاته ،

حَسْبِيَ اللهُ لا إِلَهَ إِلاَّ هُوَ عَلَيْهِ تَوَكَّلْتُ وَهُوَ رَبُّ الْعَرْشِ الْعَظِيمِ

حسبنا الله سيؤتينا الله من فضله إنا إلى الله راغبون

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

سؤال :

شركه مايكروسوفت قد حلت قديما (من قبل جافا) هذه المشكله The Vendor Variation Problem من قبل وذلك باستخدم Open Database Connectivity وأختصارا ODBC . وكانت أي DBMS Vendor يوفر ODBC Driver لكي يستطيع مستخدمين ويندوز الوصول والتعامل مع قاعده البيانات ..

من هنا قامت sun بعمل Driver لهذا ODBC من خلاله تستطيع التعامل مع أي قاعده بيانات من انتاج ميكو ، وحتى أي قاعده بيانات لا يوجد لها Driver كما قرأت .. ولكن عليك ملاحظه أن استخدام JDBC-ODBC Driver هو فقط للغرض التجربه أو عندما لا يتوفر Driver للقاعده ، فقط غير ذلك وخاصه في التطبيقات التجاريه يجب أن تستخدم Driver للقاعده أفضل من JDBC-ODBC Driver ، والسبب أنك ستكون محصور في بيئه ويندوز فقط ..

في الحقيقة لم افهم هذه النقطة .. ايضا هل كل قواعد البيانات توفر Driver من خلاله يتم الاتصال بالقاعدة

ويكون هذا Driver وحيد ام يوجد اكثر من Driver توفره قاعدة البيانات ..

الإجابة على السؤال :

بالنسبه لODBC فهي عباره عن دوال وinterface تسمح لك بالوصول الى قاعده البيانات ، وهي Specification وليست منتج نهائي ، وتهدف الى جعل الوصول الى قاعده البيانات غير متعلق لا بنظام التشغيل ولا بلغه البرمجه ..

قامت مايكروسوفت بتطبيق هذه الـSpecification وجعلها من ضمن مكونات النظام ، الأن تستطيع الوصول لأي قاعده عن طريق هذه الدوال التي طبقتها ميكورسوفت (ببساطه ODBC مجموعه من الدوال بدون تعريف ، وأي من يريد استخدام هذه الداول يقوم بتعريف الدوال بنفسه ، هذه الداول تسمى Specification ، ومن يقدم تعريف الدوال يسمى Implementor ..)

الأن نحن نتعامل من لغه الجافا ، ولن نستطيع استدعاء هذه الدوال مباشره لأنه مكتوبه بلغه سي ،، فيجب أن نستدعي داله ما (وهي عن طريق عمل معالجات معينه -كاستخدام JNI- تقوم باستدعاء تلك الدوال) ..

وهذا بالضبط وظيفه الJDBC-ODBC Driver ،، وكما ذكرت SUN لا تنصح باستخدامه في البرامج التجاريه ، فقط لأغراض التجربه .. وهذا الDriver يأتي مع أي نسخه جافا ، كما سنرى عندما نستخدم قواعد بيانات أخرى يجب أن نقوم بتحميل الDriver الخاص بها .

بالنسبه للسؤال الثاني ، نعم أغلب قواعد البيانات توفر Driver لمنتجاتها ، والا فلن يتمكن المبرمجين من الوصول والتعامل من ذلك المنتج .. لذلك من مصلحة DBMS Vendor ذلك ،، بالنسبه لتوفر أكثر من Driver لنفس القاعده فهذا قد يختلف من نسخه لأخرى .. على العموم كنت قبل فتره أستخدم Driver لMySql المهم بعد مده رأيت شخص يستخدم Driver أخر لنفس القاعده ، لكن بعدها عرفت أن نسخه MySql التي لديه تختلف عن التى استخدمها (فهو يستخدم برنامج Appserv وهو يأتي بنسخه Mysql مسبقه) .. لذلك قد تختلف الDriver باختلاف نسخ قاعده البيانات التى تعمل عليها .. ومن هنا يفضل أن تستخدم أخر اصدار من قاعده البيانات وأخر اصدار من الDriver ان امكن ..

للمزيد عن ODBC :

Open Database Connectivity

حَسْبِيَ اللّهُ لا إِلَهَ إِلاَّ هُوَ عَلَيْهِ تَوَكَّلْتُ وَهُوَ رَبُّ الْعَرْشِ الْعَظِيمِ

حسبنا الله سيؤتينا الله من فضله إنا إلى الله راغبون

نكمل ما توقفنا فيه المره السابقه ،،

بواسطة Wajdy Essam » الاثنين نوفمبر 17، 2008 8:55 pm

الجلسه السابقه ، كنا قد أخذنا مثال على الأستعلام على البيانات الموجوده في قاعده البيانات ، وقد استخدمنا الداله executeQuery والتي ترجع ResultSet يحتوي على النتيجه ، حسنا ماذا عن إدخال البيانات الى القاعده ، أو حتى حذفها أو تعديلها . هذا ما سنتطرق اليه الليله ،بالإضافه الى الحديث حول المعاملات وأهميتها Transaction ،

عند الحاجه للتعديل (ادخال،حذف) سوف نستخدم الداله executeUpdate ، ونمرر لها تعليمه SQL المناسبه ،، مثال :

```
String insert = "INSERT INTO Accounts"
 " +VALUES (123456,'Smith"،'
'" +John James',752.85؛"(
int result = statement.executeUpdate(insert؛(
```

هنا في الكود أعلاه ، قمنا بادخال بيانات حساب جديد .. وبعد أن نستخدم الداله executeUpdate سوف تكون هناك قيمه راجعه ، وهي مهمه لأنها تدل على العمليه تمت ام لأ ..

```
String change = "UPDATE Accounts"
 " +SET surname = 'Bloggs"،'
" +firstNames = 'Fred Joseph"'
 " +WHERE acctNum = 123456؛"
statement.executeUpdate(change؛(
```

نفس الأمر ، هنا قمنا بتعديل بيانات الحساب ذو الرقم 123456 بالأسم الجديد ، لكن لاحظ أننا تجاهلنا القيمه الراجعه ، وهو أمر غير مفضل طبعا ..

```
String remove = "DELETE FROM Accounts"
 " +WHERE balance < 100؛"
result = statement.executeUpdate(remove؛(
```

هنا قمنا بالحذف من قاعده البيانات أي حساب يكون الرصيد أقل من 100 .

نأخذ الأن مثال شامل ، وهو نفس المثال في الجلسه السابقه (مثال School) .. في المثال أولا سنقوم بعرض جميع البيانات الموجوده في القاعده ، ثم سنقوم بادخال حقل جديد ، وأقوم بتغيير بيانات حقل ما (تذكر قد يكون الحقل لديك باسم مختلف ، فيجب أن تغيره اذا كان كذلك ، الصوره التاليه ستوضح لك اسم الحقل في قاعدتي قبل وبعد التعديل) ، وأخيرا سنقوم بحذف سطر من البيانات وأيضا قد يختلف على حسب البيانات التي لديك ، والمثال مفهوم ان شاء الله لأننا شرحناه سابقا ..

```
//SudanCS
//Demo.java

import java.sql.Connection؛
import java.sql.Statement؛
import java.sql.ResultSet؛
import java.sql.DriverManager؛
import java.sql.SQLException؛

public class Demo
}
   private static final String userName؛ "" =
   private static final String password؛ "" =
   private static final String URL = "jdbc:odbc:School؛"
   private static final String DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver؛ "
```

```
   private static Connection connection؛
   private static Statement  statement؛
   private static ResultSet  resultSet؛

   public static void main (String arg} ([]
      try}
         Class.forName(DRIVER؛(
         connection                                                    =
DriverManager.getConnection(URL,userName,password؛(
{
      catch(ClassNotFoundException e} (
         System.err.println("Unable to Load Driver؛("
         System.exit(1؛(
{
      catch (SQLException e} (
         System.err.println("Cannot connect with database؛("!
         System.exit(1؛(
{

      try}
         statement = connection.createStatement؛()
         System.out.println("Initil database Data؛(" :
```

```
        printTable؛()

//          perform operation
        String insert = "INSERT INTO Student+ "
"                   Values('99-99','romansy','sudani','0919820         ؛ "('
        int result = statement.executeUpdate(insert؛(
        if ( result == 0 (
           System.err.println("Unable to insert data؛("!

        String update = "UPDATE Student + "
"                   SET firstName='Wajdy+ "،'
"                   lastName='Essam+ "'
"                   WHERE number = '112-1؛ "'
        result = statement.executeUpdate(update؛(
        if ( result == 0 (
           System.err.println("Unable to Update data؛("!

        String delete = "DELETE FROM Student + "
"                   WHERE tel = '54654655؛ "'
        result = statement.executeUpdate(delete؛(
        if ( result == 0 (
```

```
            System.err.println("Unable to Delete data؛("!

          System.out.println("After Operation؛(" :
          printTable؛()

//          close connection
          connection.close؛ ()
{
      catch(SQLException e} (
          e.printStackTrace؛()
{
{

    public static void printTable () throws SQLException}
        String select = "SELECT * FROM Student؛ "
        resultSet = statement.executeQuery(select؛(
        System.out.println؛()

        while ( resultSet.next} ( ()
            System.out.println("Number = " + resultSet.getString(1؛((
                System.out.println("First Name = " + resultSet.getString(2؛((
                System.out.println("Last  Name = " + resultSet.getString(3؛((
```

```
        System.out.println("Tel Number = " + resultSet.getString(4؛((
        System.out.println؛()
{
{
{
```

Transaction المعاملات :

البرامج التجاريه والتي تتطلب دخول العملاء وطلب بعضا من الأشياء ، تتطلب معالجه خاصه transaction processing ، لأنه يمكن أن يدخل عميل للموقع ، ويقوم بطلب قطعه ما ، فيقوم نظامك بادخال هذه الطلبيه في جدول الطلبيات ، ويقوم بنقصان عدد القطع في جدول القطع .. حسناً ، لكن ماذا لو حدثت مشكله بعض أن طلب العميل القطعه ، وقبل أن تنقص من جدول القطع (مشكله في الشبكه ، الأتصال ، أيا كانت !) ، هكذا يمكن لعميل أخر أن يقوم بأخذ نفس القطعه وبالتالى مشاكل لا حصر لها ..

العمل في هذه الحالات ، هو أن أي عمليه يقوم بها العميل ، تتجمع مع بعض وهو ما يسمى بالمعامله Transactions ، ولن تنفذ الا جميعا أو ولا واحده .. وبالتالي نضمن تسلسل هذه العمليات بالترتيب ..

الأوامر التي نتعامل بها في هذه الحاله هي Commit ومعناها قم بالعمليه في القاعده مباشره ، وRollback معناها تراجع من هذه العمليه (في حال حدث فشل) .. وفي JDBC الحاله الأفتراضيه لعمليه الCommit هي true ، بمعني أن طلب أي عمليه من القاعده يتم مباشره (كما ذكرنا أنه بعض الأحيان لا نريد مثل هذه الخاصيه) لذلك هنا يجب أن نغلق هذه الخاصيه .. الفقره التاليه توضح العمليه ، وكيف يمكن تفادي المشكله في حال حصلت :

```
.....
connection.setAutoCommit(false؛(
.....
try
}
//Assumes existence of 3 SQL update strings
//called update1, update2 and update3.
statement.executeUpdate(update1؛(
```

```
statement.executeUpdate(update2؛(
statement.executeUpdate(update3؛(
link.commit؛()
{
catch(SQLException sqlEx(
}
connection.rollback؛()
System.out.println)
 *"SQL error! Changes aborted؛("* ...
{
```

.....

<u>: MetaData</u>

كثيرا ما نسمع عن هذا المصطلح والذي يعني بيانات لكن عن البيانات أيضا ، وهنا في حالتنا هذه سوف يكون بيانات عن البيانات الموجوده في القاعده Data About Data .. فعندما نقوم بالأستعلام من قاعده البيانات ويأتي الناتج ResultSet ، من الممكن معرفه بيانات عن البيانات الموجوده فيه ، وذلك باستخدام الداله MetaData والتي ترجع ResultSetMetaData ، وفي هذا الكلاس سوف يكون لدينا كميه من الدوال أهمها /

```
int getColumnCount()
```

String getColumnName(<colNumber(<

int getColumnType(<colNumber(<

String getColumnTypeName(<colNumber(<

والدوال واضحه من الأسم ، الأولى تعيد عدد الأعمده في القاعده ، والثانيه تعيد اسم العمود بالindex ، والثالثه تعيد نوع البيانات الذي يوافق العمود ، والأخيره تعيد الأسم لهذا النوع ،، نأخذ مثال بسيط يوضح لنا كيفيه استخدامه :

```
//SudanCS
//Demo.java

import java.sql.Connection؛
import java.sql.Statement؛
import java.sql.ResultSet؛
import java.sql.DriverManager؛
import java.sql.SQLException؛
import java.sql.ResultSetMetaData؛

public class Demo
}
   private static final String userName؛ "" =
```

```
   private static final String password؛ "" =
   private static final String URL = "jdbc:odbc:School؛"
   private static final String DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver؛ "

   private static Connection connection؛
   private static Statement  statement؛
   private static ResultSet  resultSet؛
   private static ResultSetMetaData metaData؛

   public static void main (String arg} ([]
      try}
         Class.forName(DRIVER؛(
         connection                                                    =
DriverManager.getConnection(URL,userName,password؛(
{
      catch(ClassNotFoundException e} (
         System.err.println("Unable to Load Driver؛("
         System.exit(1؛(
{
      catch (SQLException e} (
         System.err.println("Cannot connect with database؛("!
         System.exit(1؛(
```

```
{

    try}
        statement = connection.createStatement؛()

        String select = "select * from Student؛ "
        resultSet = statement.executeQuery(select؛(
        printTable؛()

*/
 //         perform operation
        String insert = "INSERT INTO Student+ "
"                   Values('99-99','romansy','sudani','0919820        ؛ "('
        int result = statement.executeUpdate(insert؛(
        if ( result == 0 (
          System.err.println("Unable to insert data؛("!

        String update = "UPDATE Student + "
 "                  SET firstName='Wajdy+ "،'
 "                  lastName='Essam+ "'
 "                  WHERE number = '112-1؛ "'
        result = statement.executeUpdate(update؛(
```

```
        if ( result == 0 (
           System.err.println("Unable to Update data؛("!

        String delete = "DELETE FROM Student + "
"                   WHERE tel = '54654655؛ "'
        result = statement.executeUpdate(delete؛(
        if ( result == 0 (
           System.err.println("Unable to Delete data؛("!

        System.out.println("After Operation؛(" :
        printTable؛()
/*

 //         close connection
        connection.close؛ ()
{
      catch(SQLException e} (
        e.printStackTrace؛()
{
{

   public static void printTable () throws SQLException}
```

```
        metaData = resultSet.getMetaData    ;()
        int count = metaData.getColumnCount;()

        for (int i=1 ; i<=count ; i(++
           System.out.printf("%-8s\t", metaData.getColumnName(i       ;( (
        System.out.println("\n;("

        while ( resultSet.next( ()
}
           for (int i=1 ; i<=count ; i(++
              System.out.printf("%-8s\t",resultSet.getObject(i;((

           System.out.println;()
{
{

{
```

لاحظ أنه شبيه بالسابق ، لكن تم عمل Import لـResultSetMetaData ، وتم عمل تعليقات للجزء الخاص بالأدخال والحذف (لا نريده الأن) ، ركز فقط في الداله ، وشاهد كيف قمنا بطباعه الهيدر ،،

```
metaData = resultSet.getMetaData ؛()
int count = metaData.getColumnCount؛()
```

هنا حصلنا على الmetaData ، وقمنا من خلاله بمعرفه عدد الأعمده ،،

```
for (int i=1 ; i<=count ; i(++
System.out.printf("%-8s\t", metaData.getColumnName(i ؛( (
System.out.println("\n؛("
```

من خلال حلقه بسيطه ، نقوم بأخذ اسم العمود وطباعته لدينا ..

```
while ( resultSet.next( ()
{
for (int i=1 ; i<=count ; i(++
```

```
   System.out.printf("%-8s\t",resultSet.getObject(i   ؛((
   System.out.println؛()
}
```

الجزئيه هذه مهمه ، حيث الحلقه while ستذهب سطر بسطر في المخرج ، وداخلها حلقه loop ستذهب من بدايه السطر الى نهايته (عدد الأعمده) ، وتقوم بأخذ القيمه الحاليه مهما كانت نوعها عن طريق getObject ، ومن ثم طباعتها على الشاشه ..

هذه طريقه ربما أفضل من أخذ كل string لوحده وطباعته منفصلا ، فقط كل ما عليك هو عمل حلقه بعدد الأعمده ، واستخدم getObject فقط ..

DataBase && GUI:

لن نتطرق كثيرا في هذا الموضوع ، والسبب أنه يتطلب بعض الأساسيات في GUI وسوف نتكلم عنها في القريب ان شاء الله ، لكن سوف نعرض مثال على عرض المخرج فقط في جدول بسيط .. هناك أكثر من طريقه لهذا الأمر، سنستخدم واحده سهله وهي أن الكلاس JTable يحتوي على داله بناء تستقبل Vector يمثل لي البيانات ، وVector أخر يمثل لي الهيدر الخاص بالجدول .. طبعا البيانات بما أنها عده أسطر ، وكل سطر يحتوي على عده بيانات فسوف يكون هذا الVector عباره عن Vector<Vector<Object>> .. والObject في الأخير لأنه البيانات قد تختلف مثلا حقل Int وحقل string لذلك نجعلها Object ..

المثال قمت بكتابته بطريقه أفضل من الأمثله السابقه ، وقد يكون نقطه أنطلاقك في الأمثله الأكبر ، وكما ستعرف مع الممارسه أن التعامل مع قاعده البيانات هو شيء ثابت وكل مره

تقوم باعاده نفس الكود .. ربما تقوم بكتابه كود عام ، يستخدم لأي قاعده بيانات طبعا كله على حسب رغبتك ..

```
//SudanCS
//display Result of Query in JTable

import java.awt.event.WindowListener؛
import java.awt.event.WindowEvent؛
import java.awt.event.WindowAdapter؛
import java.awt.BorderLayout؛

import javax.swing.JFrame؛
import javax.swing.JPanel ؛
import javax.swing.JScrollPane؛
import javax.swing.JTable؛

import java.sql.Connection؛
import java.sql.Statement؛
import java.sql.ResultSet؛
import java.sql.DriverManager؛
import java.sql.SQLException؛
```

```
import java.sql.ResultSetMetaData;

import java.util.Vector;

public class TestDatabase
}
   public static void main (String args[]) throws SQLException
}
      QueryFrame app = new QueryFrame;()
      app.addWindowListener)
         new WindowAdapter()
}
            public void windowClosing (WindowEvent event(
}
               try
}
                  DataBastOperation.closeConnection;()
                  System.exit(0;(
{
               catch ( SQLException e(
}
                  System.out.println("Unable to Disconnect;("!
```

```
                System.exit(1؛(
{
{
{
؛(

      app.setVisible(true؛(
{
{

class QueryFrame extends JFrame
}
   private JTable tbl؛

   public QueryFrame ()  throws SQLException
}
      setTitle("Query Result؛("
      setSize(300,150؛(

      DataBastOperation db = new DataBastOperation؛()

      String select = "SELECT * FROM Student؛ "
```

```
      tbl = new JTable( db.performeQuery(select) , db.getHeading؛ ( ()
      add ( new JScrollPane(tbl) ,  BorderLayout.CENTER؛(
{
{

class DataBastOperation
}
   private static Connection connection؛
   private Statement statement؛
   private ResultSet resultSet؛

   private String DATABASE_DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver؛ "
   private String DATABASE_URL    = "jdbc:odbc:School؛ "

   private String userName؛ "" =
   private String password؛ "" =

   public DataBastOperation()
}
      try
}
```

```
        Class.forName( DATABASE_DRIVER؛ (
        connection                                                    =
DriverManager.getConnection(DATABASE_URL,userName,password؛(
        statement  = connection.createStatement؛()
{
    catch ( ClassNotFoundException e(
}
        System.out.println("Unable To Load Database؛("
        System.exit(1؛(
{
    catch ( SQLException e(
}
        System.out.println("Unable to Connect With Database؛("!
        System.exit(1؛(
{
{

  public Vector<Vector<Object>> performeQuery ( String stmt(
}
    Vector<Vector<Object>> rows = new Vector<Vector<Object؛ () <<
    try
}
```

```
        resultSet = statement.executeQuery(stmt؛(

        while ( resultSet.next ( ()
}
           Vector<Object> rslt = new Vector<Object؛() <
           rslt.add( resultSet.getString(1؛( (
           rslt.add( resultSet.getString(2؛( (
           rslt.add( resultSet.getString(3؛( (
           rslt.add( resultSet.getString(4؛( (

           rows.add( rslt؛ (
{
{
      catch ( SQLException e(
}
        System.out.println("unable to Retrive Data؛("!
        System.exit(1؛(
{

      return rows؛
{
```

```
   public Vector<String> getHeading () throws SQLException
}
      Vector<String> head = new Vector<String؛() <
      try
}
         ResultSetMetaData metaData = resultSet.getMetaData؛()
         int count = metaData.getColumnCount؛()

         for (int i=1 ; i<=count ; i(++
            head.add( metaData.getColumnName(i؛( (



{
      catch ( SQLException e(
}
         System.out.println("unable to Retrive Data؛("!
         System.exit(1؛(
{

      return head؛
{
```

```
   public static void closeConnection () throws SQLException
}
      connection.close؛()
{
{
```

وهذه صوره من المخرج :

مثال لأخر ، لعرض الموضوع من جهه أخرى ، فسوف يساعد أكثر :

```
 //program that connect with School.mdb DataBase
 //and query the dataBase

import java.sql.Connection؛
import java.sql.Statement؛
import java.sql.DriverManager؛
import java.sql.ResultSet؛
import java.sql.ResultSetMetaData؛
```

```
import java.sql.SQLException;

import java.util.Scanner;
import java.util.Vector;

import javax.swing.JFrame;
import javax.swing.JTable;
import javax.swing.JScrollPane;
import javax.swing.JPanel;

import java.awt.BorderLayout;
import java.awt.event.WindowListener;
import java.awt.event.WindowEvent;
import java.awt.event.WindowAdapter;

public class TestDatabase
}
   private static String JDBC_DRIVER  = "sun.jdbc.odbc.JdbcOdbcDriver "
;
   private static String DATABASE_URL = "jdbc:odbc:School; "
   private static String userName; "" =
   private static String password; "" =
```

```
    private static Connection connection;
    private static Statement statement;
    private static ResultSet resultSet;
    private static ResultSetMetaData metaData;
    private static  JTable t;


    public static void main (String args([]
}
        boolean state = true;
        try
}
 //          load DataBase and Make Connection
           loadDataBase ;()

           while ( state(
}
              switch ( menu ( ()
}
                  case 1:
                     insertData;()
                     break;
```

```
            case 2:
               deleteData();
               break;

            case 3:
               updateData();
               break;

            case 4:
//                printInformation();
               break;

            case 5:
               printAll();
               break;

            case 6:
               state = false;
               break;

            default:
```

```
                System.out.println("Invalid Data , Try Again؛("!
                break؛
{
{
{

    catch ( SQLException e(
}
      e.printStackTrace؛()
      System.exit(1؛(
{
    catch ( ClassNotFoundException e(
}
      System.out.println("Cannot Load DataBase؛("!
      System.exit(1؛(
{
    finally
}
      try
}
         closeDataBase؛()
{
```

```
        catch ( SQLException e(
}
            System.out.println("Cannot Close DataBase Connection؛("!
            System.exit(1؛(
{
{
{

*/
 *      loadDataBase
 *      This Method for Loading DataBase
 *      and create Statement and Connection for dealing with DataBase
/*

   public static void loadDataBase () throws SQLException , ClassNotFoundException
}
 //       Load DataBase
      Class.forName( JDBC_DRIVER؛ (

 //       make connection to database
     connection = DriverManager.getConnection( DATABASE_URL , userName , password؛ (
```

```
//      get statement to qury database
      statement = connection.createStatement؛()
{

   public static void   closeDataBase() throws SQLException
}
      statement.close؛()
      connection.close؛()
{

*/
 *     menu
 *     This Method for Displaying Menu For User
 *     and return the user choice
/*

   public static int menu()
}
      System.out.println("  \n\n*********  School  DataBase  System
********** \n؛("
      System.out.println("Enter New Student Information      ..... [1؛("[
```

```
        System.out.println("Delete Exsist Student Information  ..... [2ؼ("[
        System.out.println("Update Exsist Student Information  ..... [3ؼ("[
        System.out.println("Print Spcified Student Information ..... [4ؼ("[
        System.out.println("Print All Student Information      ..... [5ؼ("[
        System.out.println("Exit From School System           ..... [6ؼ("[

        Scanner input = new Scanner( System.inؼ(
        System.out.print("\n\tؼ( " <<
        int choice = input.nextIntؼ ()

        return choiceؼ
{

*/
 *      printAll
 *      This Method for Displaying All Student information
/*

   public static void printAll () throws SQLException
}
        String selectAll = "SELECT * FROM Student" ;   // Sql Statement
        resultSet = statement.executeQuery(selectAllؼ(
```

```
//      get infromation about databse table
      metaData = resultSet.getMetaData      ;()

      int numberOfColumns = metaData.getColumnCount;()
      Vector <String> header = new Vector<String;()<
      Vector <Vector<Object>> rows = new Vector <Vector<Object;() <<


//      print DataBase Columns Header
      for (int i=1 ; i<=numberOfColumns ; i(++
//          System.out.printf("%-8s\t",metaData.getColumnName(i;((
          header.add( metaData.getColumnName(i;  (  (

      System.out.println;()

//      print result of Database
      while ( resultSet.next( ()
}
//          System.out.printf("%-8d\t",resultSet.getInt(1;((
//          System.out.printf("%-8s\t",resultSet.getString(2;((
//          System.out.printf("%-8s\t",resultSet.getString(3;((
```

```
//          System.out.printf("%-8s\t",resultSet.getString(4؛((
//          System.out.printf("%-8d\t",resultSet.getInt(5؛((
//          System.out.println؛()

          Vector <Object> row = new Vector<Object؛() <
          row.add( resultSet.getString(1؛((
          row.add( resultSet.getString(2؛((
          row.add( resultSet.getString(3 ؛((
          row.add( resultSet.getString(4؛((
          rows.add(row؛(
{
      t = new JTable(rows,header؛(
      DisplayTable dt = new DisplayTable(t؛(
      dt.setSize(400,200؛(
      dt.setVisible(true؛(
      dt.addWindowListener)
         new WindowAdapter()
}
            public void windowClosing(WindowEvent event(
}
               return؛
{
```

```
{
;(
{

*/
 *      getName
 *      This Method return FirstName entered by user
 *      and its used by other method
/*

   public static String getName()
}
      Scanner input = new Scanner (System.in;(
      System.out.print("\nEnter Student FirstName;(" :
      String s = input.next;()

      return s ;
{

*/
 *      deleteData
 *      This Method delete student info by firstName
```

```
/*


   public static void deleteData () throws SQLException
}
 //      not complete , you must enter all information , and delete  thing you want
      String name = getName؛()


      String s = "DELETE  FROM Student WHERE firstName = '" + name + ؛ """


      int state = statement.executeUpdate(s؛(


      if ( state != 0(
         System.out.println("\nDelete Student = " + name؛(
      else
         System.out.println("Cannot Found Student = " + name؛(


{


*/
 *     insertData
 *     This Method delete student info by firstName
```

```
/*

   public static void insertData () throws SQLException
}
      Scanner input = new Scanner (System.in؛(

      System.out.print("\nEnter Student Number؛(" :
      String id = input.next؛()

      System.out.print("\nEnter Student FirstName؛(" :
      String fn = input.next؛()


      System.out.print("\nEnter Student LastName؛(" :
      String ln = input.next؛()

      System.out.print("\nEnter Student Tel؛(" :
      String d = input.next؛()


      String s = "INSERT INTO Student VALUES+ "')
               id + "','" + fn + "','" + ln + "','" + d؛ "('"+
```

```
        int state = statement.executeUpdate(s؛(

        if ( state != 0(
           System.out.println("\nINSERT Student = " + fn؛(
        else
           System.out.println("\nINSERT Student = " + fn؛(

              {

*/
 *      updateData
 *      This Method Update  student info by firstName
/*

    public static void updateData () throws SQLException
}
 //       not complete , you must enter all information , and update thing
you want
        String name = getName؛()

        String s = "UPDATE Student SET firstName = '" + name؛ """ +
```

```
      int state = statement.executeUpdate(s؛(

      if ( state != 0(
         System.out.println("\nUpdate Student = " + name؛(
      else
         System.out.println("\nCannot Found Student = " + name؛(

{
{

class DisplayTable extends JFrame
}
   private JTable tbl؛

   public DisplayTable ( JTable t(
}
      tbl = t؛
      setTitle("Display Result   ؛("
      add(new JScrollPane(tbl) , BorderLayout.CENTER   ؛(
 {
{
```

سوف يعرض قائمه،ويتيح لك ادخال والتعديل على القاعده ، والمثال غير مكتمل بشكل 100% .. اقرأه وسوف تعرف كيف تكمله بنفسك

سأتوقف هنا ، والمره القادمه باذنه تعالى نتناول موضوع في غايه الأهميه وهو التحرك في النتيجه Scrollable ResultSets ..

سؤال :

.. هل يمكن أن نعُد الـ jdk أنهآ Drive؟

بالنسبه لJDK فهي الحزمه التي تستخدميها للبرمجه فهي تحتوي على المترجم javac والمفسر java وبرامج أخري كApplet Veiwer وjava doc . وهو يحتوي على Driver للربط مع الODBC ويسمى JDBC-ODBC Driver . لذلك ان أردت أن تستخدم هذا الDriver فلن تحتاج لتحميله لأنه يأتي مسبقا مع اللغه ..

وهذا شرح بالصور من الأخ أبو سعود الفرنسي من الفريق العربي للبرمجه لربط جافا مع MS-Sql Server 2005 :

ربط جافا مع MS-SQL Server 2005

(لاحظ أنه أستخدم Driver جديد ، بالتالي يجب تحميله لكي يعمل ، مع أني أظن ان الدرايفر الذي استخدمناه مع أكسس يفي بالغرض) .

وهذا شرح بالصور لربط MySql التي تأتي مع Appserv ، للأخ شادي :

ربط جافا مع MySql الموجوده في Appserv

سأقوم فيما بعد بتوضيح ربط جافا مع Mysql فقط ، وكيف يمكن أن نعدها بشكل جيد .. بالإضافه الى قاعده Sqlite وأيضا Derby والتي تأتي مع نسخ نت بينز الجديده ، بس نتنهي من الأساسيات بالأول وهذه الأمور سهله ان شاء الله ..

حَسْبِيَ اللهُ لا إلَهَ إلاّ هُوَ عَلَيْهِ تَوَكّلْتُ وَهُوَ رَبُّ العَرْشِ العَظِيم

حسبنا الله سيؤتينا الله من فضله إنا إلى الله راغبون

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

بواسطة ToBeOrNotToBe » الأربعاء يناير 07، 2009 3:09 am

للــرفع ..!!

ToBeOrNotToBe

طالب جديد

مشاركات: 17

اشترك في: الثلاثاء سبتمبر 09، 2008 4:38 am

أعلى

---

ok (-:

بواسطة Wajdy Essam » الأربعاء يناير 07، 2009 8:56 am

ToBeOrNotToBe

welcome again brother , we sorry for late the progress of this tutorial , but this due to some problem in the past time ،

however , we will continue to complete this tutorial even if we late many days , but now we will explain the last point we stopped in last section "Scrollable ResultSets " and cover every thing about this point

for demonstrate all the available DBMS and how it's usage it will be tedious task , so we will take one DBMS like sqlie and explain how to configure DB and how to connect to this DB from java application , and the other DBMS will be the same method but only you need to change the driver and to configure the DB with yourself

sorry for written in this bad english langauge , their is no arabic support now in my pc .

happy java programming

حَسْبِيَ اللّهُ لا إِلَهَ إِلاَّ هُوَ عَلَيْهِ تَوَكَّلْتُ وَهُوَ رَبُّ الْعَرْشِ الْعَظِيمِ

حسبنا الله سيؤتينا الله من فضله إنا إلى الله راغبون

Wajdy Essam

طالب فعال

مشاركات: 1059

اشترك في: الاثنين إبريل 21، 2008 12:43 am

الجامعة: Neelain

المستوى الدراسي: Graduate

التخصص: Computer Science

WWW

أعلى

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

بواسطة ramrode » الخميس يناير 08، 2009 10:52 am

!!! nice work

ramrode

طالب نشيط

مشاركات: 22

اشترك في: السبت أغسطس 02، 2008 7:31 pm

الجامعة: ComputerMan College

المستوى الدراسي: مهندس خرييج

التخصص: telecommunications

أعلى

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

بواسطة ToBeOrNotToBe » السبت يناير 17، 2009 4:39 am

شكرا لك على تجاوبك معنا ونحن في انتظار اكمال ما بدأته ولكن اتمنى ان يكون العمل على قاعدة البيانات MYSQL نظرا لصعوبة التعامل معها ..

ونتمنى أن لايطول انتظارنا ..

شكرا جزيلا .. استاذنا العزيز ..

ToBeOrNotToBe

طالب جديد

مشاركات: 17

اشترك في: الثلاثاء سبتمبر 09، 2008 4:38 am

أعلى

حسنا أخي ،

بواسطة Wajdy Essam » الجمعة يناير 23، 2009 12:04 pm

المعذره على الأطاله ولكن يمكنك تطبيق البرامج في أي قاعده بيانات مهما كانت وسوف تعمل ان شاء الله ..

يبقى كيف يمكن أن تعد قاعده MySQL بشكل جيد ، وصراحه القاعده ضخمه للغايه وتحتاج لكتاب كامل ان أردت أن تتعامل بكل صغيره وكبيره بها ، حيث هي تعمل في طور Client-Server كما ذكرت سابقا ، المهم معذره على الاطاله في الموضوع والسبب الانشغال بالدراسه ، ولكن باذن الله سنكمل الموضوع مهما كان .. وستنحدث عن نتصيب MySql والعمل في أساسياتها فقط أما للمزيد فتحتاج للقرائه في الكتب والمواضيع المختصه بـMySQL .

على العموم هذه روابط مفيده ان شاء الله :

Using JDBC with MySQL, Getting Started

Connecting Netbeans to a MySQL Database

Connecting to MySQL Java using NetBeans

Using MySQL With Java

سؤال فقط عن ؟

"Commit ومعناها قم بالعمليه في القاعده مباشره " هل هناك عمليات لا تتم فى القاعدة مباشرة

وما معنى

link.commit

ممكن تضع مثال يوضح بصورة اكبر Transactions

اللهم استرنى واجعل كفينى زيادة -امين امين امين

http://www.wathakker.net/lib_audio/view.php?id=436

"حَسْبُنَا اللّهُ سَيُؤْتِينَا اللّهُ مِن فَضْلِهِ إِنَّا إِلَى اللّهِ رَاغِبُونَ"

{وَاسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ وَإِنَّهَا لَكَبِيرَةٌ إِلَّا عَلَى الْخَاشِعِينَ}

يرتوي بالحب قلبي .. حب خير رسل ربي .. من به ابصرت دربي .. ياشفيعي يارسول الله هل سلمت على رسولك اليوم؟؟؟اللهم صلى على سيدنا محمد وعلى اله وصحبة وسلم.

soldierofallah

فريق العمل

مشاركات: 1158

اشترك في: السبت يناير 31، 2009 4:41 pm

مكان: جنة الفردوس بأذن الله

الجامعة: umar almukhtar

المستوى الدراسي: still ask to learn

التخصص: Purpose of Life

الاهتمامات: SaViNg ThE wOrld

WWW

YIM

أعلى

حياك الله ،،

بواسطة Wajdy Essam » السبت فبراير 14، 2009 2:53 pm

في البدايه الإعداد الإفتراضي لأي جمله SQL نقوم بتنفيذها في البرنامج ستكون Commited بمعنى أن أي عمليه (وليكن ادخال بيانات طالب في جدول) فور أن تتنفذ لن يمكن التراجع منها مطلقا Rollback .

حسناً ما الفائده من التراجع أصلا ، فنحن نعلم أننا نريد أدخال بيانات هذا الطالب ، فلماذا نعيد هذه العمليه بعد تنفيذها؟

الموضوع كله يتمحور على ثبات القاعده database integrity ، تخيل لدينا نظام بنكي وتوجد لدينا داله تحويل رصيد من حساب لأخر .. تم تنفيذ هذه الداله وذلك بنقص القيمه المراد تحويلها من الحساب الأول ، ولسبب ما حصلت مشكله (سواء في الشبكه ، علق البرنامج Hanging ،أي مشكله) قبل أن يتم تنفيذ الداله الثانيه وهي ادخال هذه القيمه في الحساب الثاني ..

هنا تكمن المشكله ،، حيث أن العمليه الأولى تم تنفيذها Commited أما الثانيه فلا .. ولا يمكن التراجع عن العمليه الأولى ..

فكان الحل الأفضل هو عدم تنفيذ هذه العمليات الحساسه كل منها على حده بل يتم تنفيذها مره واحده ، ففي حال تنفذت جميعها فلا توجد مشاكل ، والا فسوف يتم التراجع عن جميع هذه العمليات Rollback . وتجميع هذه العمليات كلها عاده يطلق عليه في قواعد البيانات باسم الTransaction(عباره عن تجميع العديد من العمليات التي نريد أن نطبقها على القاعده ، والفائده منها كما ذكرت هي أنها أما أن تنفذ كلها أو لا تنفذ اطلاقا)

لذلك اذا كانت الداله حساسه كما في داله تحويل الرصيد يفضل تنفيذ العمليات مره واحده وذلك يكون أولا باغلاق التنفيذ التلقائي في القاعده ، وبعدها يتم اجراء العمليات بشكل عادي ، وأخيرا يتم استدعاء الداله commit لتنفيذ جميع هذه العمليات واحده تلو الأخرى .. وفي حال حدث خطأ (مثلا في العمليه الثالثه من تنفيذ العمليات) SQLException يتم استدعاء الداله rollBack والتي تعيد القاعده الى حالتها الأصليه قبل تنفيذ هذه العمليات جميعها .

```
conn.setAutoCommit(false؛(

 //other statements

Statement stat = conn.createStatement؛()
```

```
//operations
stat.executeUpdate(command1؛(
stat.executeUpdate(command2؛(
stat.executeUpdate(command3؛(

//now execute all operations
conn.commit؛()

//if error accure return to last commited
conn.rollback؛()
```

باختصار فقط الموضوع متعلق حول ضمان تنفيذ مجموعه من العمليات أو عدم تنفيذ أي عمليه منهم ..

أرجوا أن تكون قد أتضحت فكرته ، وللمزيد أنظر هنا : Using Transactions

بالتوفيق ،،

نعم اتضحت الفكرة جزيت خير

ربما احاول قرائة الرابط لتطبيقها ان شاء الله

اخى الفاضل لماذا لا تضع تمارين ونعمل عليها حتى يتضح لكل عضو مدى استفادتة من الدرس ويفيد الاخرين بمحاولتة ويلقى الموضوع محاولات وتطبيقات شتى نتبدلها جميعا

لكن لا تضع لنا برامج صعبة حتى لا نهرب منها

كما ننتظر شرح قاعدة mysql والربط بها

جزاك الله خير لما تبذله من نشر العلم

اللهم استرنى واجعل كفينى زيادة -امين امين امين

http://www.wathakker.net/lib_audio/view.php?id=436

{وَاسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ وَإِنَّهَا لَكَبِيرَةٌ إِلَّا عَلَى الْخَاشِعِينَ}

يرتوي بالحب قلبي .. حب خير رسل ربي .. من به ابصرت دربي .. ياشفيعي يارسول الله .هل سلمت على رسولك اليوم؟؟؟اللهم صلى على سيدنا محمد وعلى اله وصحبة وسلم

hi again

حسناً يمكن أن تطبقوا أي مثال صغير ، يكفي القيام بمهمه الأدخال والأخراج والحذف (سواء لبيانات طلاب ، موظفين ، أجهزه) ، ويكون سهل في الاستعلام عن الأشياء الموجوده فيه ويمكن أن تستخدموا الواجهات GUI اذا أمكن ذلك ..

يكفى هذا كبدايه ،، ولن نزيد متطلبات أكثر حتى لا نكبر البرنامج ويصبح صعب التعديل ، نريد تعلم كيفيه كتابه برامج سهله التعديل والصيانه وذلك باستخدام أساليب البرمجه الموجهه والdesign pattern . باذنه تعالى فقط أنتهي من الأختبارات -نهايه الشهر الحالي- سنتناول مثال كامل لبرنامج من هالنوع مثلا متجر للأفلام أو البضائع ، وسنحاول أن نقوم بفصل الواجهه عن التطبيق عن قاعده البيانات (3 طبقات) وبالتالى تكون تعديل وصيانه أي طبقه أمر مستقل بذاته ولا يؤثر على غيره .. كنا قد تحدثنا عن هذا المفهوم في القسم العام باسم MVC وان شاء الله نقوم بتطبيقه في برنامجنا الجديد (الذي سيستخدم MySQL) ..

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

بواسطة soldierofallah » السبت فبراير 14، 2009 4:07 pm

السلام عليكم ورحمة الله وبركاتة

لكن ما طلبتة حضرتك من اضافة بيانات وحذف---- حضرتك قمت بة بالفعل ام حضرتك تقصد ان نتقن التعامل مع اوامر الاستعلام

عموما بحاول اضيف على برنامج حضرتك عمل Gui له

ولى مهلة اخر الشهر ان لم اقوم بعملة حضرتك تقوم بعملة بقى ---لكن ان شاء الله بضع محاولاتى

بالتوفيق فى اختباراتك ان شاء الله

جزيت خير على تفاعلى------- كانك تشكر شخص لانة يتعلم ويستفيد هذا كرم منك

جزاك الله خير

اللهم استرنى واجعل كفينى زيادة -امين امين امين

http://www.wathakker.net/lib_audio/view.php?id=436

"حَسْبُنَا اللّهُ سَيُؤْتِينَا اللّهُ مِن فَضْلِهِ إِنَّا إِلَى اللّهِ رَاغِبُونَ"

{وَاسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ وَإِنَّهَا لَكَبِيرَةٌ إِلَّا عَلَى الْخَاشِعِينَ}

يرتوي بالحب قلبي .. حب خير رسل ربي .. من به ابصرت دربي .. ياشفيعي يارسول الله هل سلمت على رسولك اليوم؟؟؟اللهم صلى على سيدنا محمد وعلى اله وصحبة وسلم.

Re: برمجه تطبيقات قواعد البيانات بجافا JDBC

بواسطة soldierofallah » الاثنين فبراير 16، 2009 12:27 am

السلام عليكم ورحمة الله وبركاتة

الحمد الله تقريبا اوشك على الانتهاء منة

لكن لى سؤال لحضرتك او للاعضاء

كيف اذا كنت استخدم netbeans id ان اعرف حجم panelمعينة ،ومكانها على الframe حتى استبدلها ب text area بنفس الحجم

وفى نفس الposition

اذا هناك روابط مفيدة فجزيت خير اذا لا يوجد ننتظر رد حضرتك بعد الاختبارات ان شاء الله

والسلام عليكم ورحمة الله وبركاتة

اللهم استرنى واجعل كفينى زيادة -امين امين امين

http://www.wathakker.net/lib_audio/view.php?id=436

يرتوي بالحب قلبي .. حب خير رسل ربي .. من به ابصرت دربي .. ياشفيعي يارسول الله .هل سلمت على رسولك اليوم؟؟؟اللهم صلى على سيدنا محمد وعلى اله وصحبة وسلم

متواجد ان شاء الله ،،

بواسطة Wajdy Essam » الاثنين فبراير 16، 2009 1:05 am

هل تريد استبدالها برمجيا من خلال الكود ؟ أم من خلال البرنامج ؟

فاذا كان من خلال البرنامج ، يمكنك فقط حذف البانل ووضع JTextArea في نفس المكان وقم بتكبير وتصغير حجمه الى الحجم المراد ..

أو الحل الثاني وهو الأفضل (في كلا الحالات سواء برمجيا أم بالتصميم) وهو اضافه الTextArea مباشره فوق هذه الPanel ، حيث أنه من الأفضل عدم وضع المكونات لوحدها بل يجب أن توضع فوق البانل -عمليه تنظيميه ليس إلا-.

ارجوا ان أكون فهمت السؤال ، والا ارجوا توضيحه أكثر ،

بالنسبه للروابط حول الNetBeans فيوجد الموقع الرسمي :

Java and JavaFX GUI Application Learning Trail

وبالأسفل Designing Java GUIs ستجد درسين حول التصميم بالبرنامج

وهذا كتاب عن ال NetBeans .

أيضا JavaFX ذاع صيتها كثيرا الأيام الماضيه ، حيث يعمل البرنامج المكتوب بها في أي تطبيق سواء Desktop أم Mobile ام حتى web Application

Building GUI Applications With JavaFX

Getting Started With JavaFX Technology

لم أجربها صراحه ، ولكن على ما يبدوا أن لها مستقبل جيد وخصوصا في الInterface .

التالي عرض مشاركات سابقة منذ: جميع المشاركات يوم أسبوع أسبوعان شهر3 شهور6 شهورسنة مرتبة بواسطة الكاتب وقت الإرسال عنوان تصاعدياً تنازلياً