

آموزش ذخیره رکورد جدید در جاوا

قبل از اینکه بتوانید یک رکورد جدید را ذخیره کنید، باید که نشانگر را به بخشی به نام **Insert Row** حرکت دهید. این امر یک رکورد خالی در **ResultSet** ایجاد می کند. سپس داده را به **ResultSet** اضافه می کنید:

مثال :

```
rs.moveToInsertRow( );
rs.updateInt("ID", newID);
rs.updateString("First_Name", first);
rs.updateString("Last_Name", last);
rs.updateString("Job_Title", job);
rs.insertRow( );
```

پس از افزودن داده به **ResultSet**، آخرین خط یک ردیف جدید اضافه می کند.

به هر حال برای اعمال هرگونه تغییر در دیتابیس، آنچه ما انجام خواهیم داد بستن آبجکت **Statement** و آبجکت **ResultSet** می باشد. سپس می توانیم هر چیزی را مجدداً بارگذاری کنیم. اگر این کار را انجام ندهیم، خطر اضافه نشدن رکورد جدید یا به **ResultSet** یا به دیتابیس وجود دارد. (این بستگی به نوع درایوری دارد که استفاده کرده ایم).

برای بستن **Statement** یا **ResultSet**، باید فرمان **close** را انتشار دهید:

```
stmt.close( );
rs.close( );
```

کد مربوط به بارگذاری مجدد موارد، همان کدی است که در هنگام اولین بارگذاری فرم نوشتید:

```
stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
String sql = "SELECT * FROM Workers";
rs = stmt.executeQuery(sql);
rs.next( );
int id_col = rs.getInt("ID");
String id = Integer.toString(id_col);
String first2 = rs.getString("First_Name");
String last2 = rs.getString("Last_Name");
String job2 = rs.getString("Job_Title");
textID.setText(id);
textFirstName.setText(first2);
textLastName.setText(last2);
textJobTitle.setText(job2);
```

در اینجا در واقع کار متفاوتی انجام نمی دهید: فقط همه ی رکوردها را مجدداً انتخاب کرده و اولین مورد را در فیلدهای **Text** قرار می دهید.

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

در اینجا کدی را مشاهده می کنید که یک رکورد جدید در دیتابیس ذخیره می کند (واضح است که بیشترین قسمت این کد می تواند وارد متود خود شود):



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>

```
String first = textFirstName.getText();
String last = textLastName.getText();
String job = textJobTitle.getText();
String ID = textID.getText();
int newID = Integer.parseInt(ID);

try {
    rs.moveToInsertRow();

    rs.updateInt("ID", newID);
    rs.updateString("First_Name", first);
    rs.updateString("Last_Name", last);
    rs.updateString("Job_Title", job);

    rs.insertRow();

    stmt.close();
    rs.close();

    stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                               ResultSet.CONCUR_UPDATABLE);

    String sql = "SELECT * FROM Workers";
    rs = stmt.executeQuery(sql);

    rs.next();
    int id_col = rs.getInt("ID");
    String id = Integer.toString(id_col);
    String first2 = rs.getString("First_Name");
    String last2 = rs.getString("Last_Name");
    String job2 = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first2);
    textLastName.setText(last2);
    textJobTitle.setText(job2);

    btnFirst.setEnabled(true);
    btnPrevious.setEnabled(true);
    btnNext.setEnabled(true);
    btnLast.setEnabled(true);
    btnUpdateRecord.setEnabled(true);
    btnDelete.setEnabled(true);
    btnNewRecord.setEnabled(true);
}
```

(مسئله ی دیگر این است که ستون ID باید منحصر به فرد باشد. به طور ایده آل، شما مسیری برای دریافت آخرین شماره ی ID و سپس اضافه کردن یک عدد به آن نوشته اید. دیتابیس های دیگر، مانند MySQL، دارای یک مقدار AutoIncrement می باشد برای مراقبت از این موارد. فقط اطمینان حاصل کنید که مقدار ID مقداری نیست که قبلا استفاده کرده اید، در غیر اینصورت یک پیغام خطا دریافت خواهید کرد. یا مسیری برای دریافت یک ID منحصر به فرد بنویسید.)

برنامه ی خود را اجرا کرده و آن را امتحان کنید. اکنون قادر هستید که رکوردهای جدید را در دیتابیس خود ذخیره کنید. در بخش بعدی در مورد حذف رکوردها فرا خواهید گرفت .



آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>