

Использование SQLdb в Lazarus: основы работы с базами данных

Публикации Lazarus

11.03.2009
aRix

Пакет **SQLdb** включает в себя набор модулей для доступа к различным SQL базам данных. Собственно сам пакет находится в файле sqldb.lpk. Найти визуальные компоненты можно на вкладке **SQLdb**.

Компоненты

Пакет **SQLdb** включает следующие компоненты:

TSQLQuery

TSQLQuery потомок **TDataset** и он представляет данные из SQL запроса в виде таблицы. Также этот компонент можно использовать для выполнения запроса, не возвращающих данных.

TSQLTransaction

Создает транзакцию на сервере базы данных. Я не буду подробно рассказывать об этом компоненте, поскольку практически не использую его. Важно, что он имеет методы для фиксации и отката транзакции.

TXXXConnection

Где XXX название базы данных к которой необходимо подключиться. Каждый из этих компонентов переводит стандартные запросы от **SQLQuery** и **SQLTransaction** в запросы к конкретной базе данных, сглаживая их различия. В версии 0.9.26 доступны следующие компоненты

- TIBConnection (Borland Interbase / Firebird)
- TODBConnection (ODBC соединение)
- TOracleConnection (Oracle)
- TMySQL40Connection (MySQL - различных версий)
- TMySQL41Connection
- TMySQL50Connection

- TPQConnection (PostgreSQL)
- TSQLite3Connection (fpc version 2.2.2)

Использование пакета SQLdb в Lazarus

Для использования SQLdb в Lazarus Вам понадобятся компоненты **TSQLQuery** и **TSQLTransaction**, а также один из Connection компонентов.

Вы должны проделать ряд шагов:

- Перейдите на вкладку SQLdb в панели компонентов.
- Добавьте один из T___Connection компонентов на форму и заполните необходимые свойства для соединения с базой данных. Конкретные свойства зависят от конкретной базы данных. Убедитесь, что установили свойство "connected" в true.
- Добавьте TSQLTransaction компонент. Вернитесь к Connection и установите в его свойстве Transaction ссылку на вновь добавленный TSQLTransaction компонент. Необходимо также установить свойство transaction. После этого установите свойство active в true.
- Добавьте SQLQuery компонент и установите в свойствах Database и Transaction ссылки на ранее добавленные компоненты.
- Установите свойство SQL компонента SQLQuery. Для проверки можете записать "select * from ".
- Установите свойство active в true. Данные будут доступны после того как вы выполните запрос.

После этого добавьте элементы управления

- Добавьте компонент TDatasource из панели (Data Access) и установите в свойстве dataset ссылку на SQLQuery компонент.
- Добавьте компоненты управления из вкладки Data Controls. Например, вы можете добавить TDBGrid и установить в его свойстве datasource ссылку на Datasource компонент, который вы ранее добавили на форму. После этого Вы можете увидеть результат своего запроса в DBGrid элементе.

Работа с TSQLQuery

Результат запросов, выполняемых с помощью **TSQLQuery**, можно непосредственно отображать в **TDBGrid**, однако **TDBGrid** не подходит для редактирования отдельных полей базы данных. Для этих целей можно использовать компонент **TDBEdit**, установив его свойство DataSource. В свойстве DataField необходимо указать имя редактируемого поля.

Компонент **TDBNavigator** значительно упрощает навигацию по записям базы данных. После того как запись выбрана с помощью DBNavigator или в DataGrid, соответствующие данные отображаются в **TDBEdit**. Если нажать кнопку Edit, эти данные можно будет изменять. Нажмите кнопку Post для внесения изменений в базу данных или Cancel для отмены изменений.

1. Добавьте TSQLQuery на форму/datamodule, и установите свойства Database, Transaction и SQL.

2. Добавьте TDataSource, и установите в свойстве Dataset ссылку на TSQLQuery компонент.
3. Добавьте TDBGrid на форму, и установите в свойстве DataSource ссылку на TDataSource компонент.
4. При желании можете добавить на форму панель TDBNavigator, и установить в свойстве DataSource ссылку на TDataSource.

После того, как вы проделаете эти шаги, установите свойство Active (компонент TSQLConnection и TSQLTransaction) в true. После этого результат запроса будет отображаться на форме.

Обновление данных

TSQLQuery кеширует все изменения. Для того, чтобы изменения были внесены в базу данных необходимо вызвать метод ApplyUpdates. Если не вызывать метод ApplyUpdates, то все изменения будут утеряны.

Свойство Primary key

При изменении записи, **TSQLQuery** необходимо знать какие поля составляют первичный ключ, который можно использовать для изменения записи.

Конструкция инструкции SQL управляется свойствами UsePrimaryKeyAsKey и ProviderFlags.

Свойство Providerflags включает в себя три флага:

pflnkey - поле является частью первичного ключа.

pflnWhere - поле должно использоваться в области WHERE SQL запроса.

pflnUpdate - Update или insert запросы должны включать это поле.

По умолчанию ProviderFlags содержит только pflnUpdate.

Если у таблицы задан первичный ключ, Вам просто необходимо установить UsePrimaryKeyAsKey в true. Это установит флаг pflnKey для поля первичного ключа.

Если таблица не содержит первичного ключа, но в ней есть несколько полей, которые могут однозначно идентифицировать запись, Вы можете включить pflnKey опцию в свойстве ProviderFlags всех полей, однозначно идентифицирующих запись.

Свойство UpdateMode будет определять какие поля будут использоваться в области WHERE:

upWhereKeyOnly

Когда TSQLQuery требуется конструкция WHERE, он собирает все поля у которых установлен флаг pflnKey в свойстве ProviderFlags и использует их значения для создания условия WHERE, однозначно определяющее поля для модификации (Обычно используется в UPDATE и DELETE).

upWhereChanged

В дополнении к полям, у которых установлен флаг pflnKey в свойстве ProviderFlags, все поля с флагом pflnWhere в свойстве ProviderFlags и которые изменились, будут включены в WHERE условие.

upWhereAll

В дополнении к полям, у которых установлен флаг pflnKey в свойстве ProviderFlags, все поля с флагом pflnWhere в свойстве ProviderFlags будут включены в WHERE условие.

Управление изменениями

Возможно определить, какие поля должны быть модифицированы: Как упомянуто выше: только поля, которые имеют pflnUpdate в их свойства ProviderOptions будут включены в SQL инструкции UPDATE ИЛИ INSERT. По умолчанию, pflnUpdate всегда включается в ProviderOptions.

Замечания

В базе MySQL 5.0.45 свойство active TSQLTransaction нельзя установить в true, однако все работает и компилируется.

Источник: homepg.ru

Оригинал: wiki.lazarus.freepascal.org/SQLdb_Package

Актуальные версии

FPC 3.2.2 release

Lazarus 3.2 release

MSE 5.10.0 release

fpGUI 1.4.1 release

[links](#)

Рекомендуем практичный [веб-сайт](#) полезных советов sovetnika.net, смотри в удобном формате на гаджете.

