Inserting a row into a database in Qt

Overview

This example shows you how to insert data into an SQLite database in Qt. We show how to use an autoincrement field that generates a new ID for the new row automatically.

The table 'person' has the following columns:

- `id (integer primary key)`, **this is an autoincrement field**
- `firstname (varchar(20))`
- `lastname (varchar(30))`
- `age (integer)`

Preconditions

- Qt is installed on your platform.
  - S60:
    - Download Qt release from here: Qt pre-release
    - Install Qt: How to Install Qt
    - Check this link for installation guide: How to install the package
    - Qt Tower release has SQLite support. The required libraries are built into the Qt release.
  - Maemo:
    - More information about Qt on Maemo can be found here: Qt4 Maemo port

For Maemo SQLite development, the following packages must be installed:

- libqt4-sql
- libqt4-sql-sqlite
- libsqlite3-0
- libsqlite3-dev

Header

```
#include <QObject>
#include <QSqlDatabase>
#include <QSqlError>
#include <QSqlQuery>

class DatabaseManager : public QObject
    {
    public:
        DatabaseManager(QObject *parent = 0);
        ~DatabaseManager();

    public:
        bool openDB();
        bool createPersonTable();
        int insertPerson(QString firstname, QString lastname, int age);

    private:
        QSqlDatabase db;
    };
```

Source

The following code inserts a new row (person) and the autoincrement field generates a new ID for the row. Next, the newly generated ID is queried.

```
int DatabaseManager::insertPerson(QString firstname, QString lastname, int age)
    {
    int newId = -1;
    bool ret = false;

    if (db.isOpen())
        {
        //http://www.sqlite.org/autoinc.html
        // NULL = is the keyword for the autoincrement to generate next value

        QSqlQuery query;
        ret = query.exec(QString("insert into person values(NULL,'%1','%2',%3)")
        .arg(firstname).arg(lastname).arg(age));

        // Get database given autoincrement value
        if (ret)
```


(C) Copyright Nokia 2013. All rights reserved.

```
            {
            // http://www.sqlite.org/c3ref/last_insert_rowid.html
            newId = query.lastInsertId().toInt();
            }

        }
    return newId;
    }
```

The rest of the code:

```
bool DatabaseManager::createPersonTable()
    {
    // Create table "person"
    bool ret = false;
    if (db.isOpen())
        {
        QSqlQuery query;
        ret = query.exec("create table person "
                  "(id integer primary key, "
                  "firstname varchar(20), "
                  "lastname varchar(30), "
                  "age integer)");

        }
    return ret;
    }

bool DatabaseManager::openDB()
    {
    // Find QSLite driver
    db = QSqlDatabase::addDatabase("QSQLITE");

    #ifdef Q_OS_LINUX
    // NOTE: We have to store database file into user home folder in Linux
    QString path(QDir::home().path());
    path.append(QDir::separator()).append("my.db.sqlite");
    path = QDir::toNativeSeparators(path);
    db.setDatabaseName(path);
    #else
    // NOTE: File exists in the application private folder, in Symbian Qt implementation
    db.setDatabaseName("my.db.sqlite");
    #endif

    // Open databasee
    return db.open();
    }
```

Postconditions

A new person is added into the database.

See also

- Creating an SQLite database in Qt
- Creating a database table in Qt
- Searching for data in a database in Qt
- Deleting data from a database in Qt
- Selecting data from a database without using SQL statements in Qt
- Using QDataWidgetMapper to show data from a database in Qt
- QtSql module
- SQLite autoincrement

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.