

Московский институт электроники и математики
Национального исследовательского университета
«Высшая школа экономики»
Факультет Информационных технологий и вычислительной техники

Курсовая работа по дисциплине
«Компьютерная графика»
по теме «Автоматическое создание и редактирование PDF»

Выполнил студент группы СБ-55

М.Г. Флейтман

Проверил

доцент

Д.А. Королев

Оглавление

Задание:.....	3
Условия:	3
Существующие аналоги.....	3
Скриншоты	4
Возможности реализации	6
Ход работы.....	7
Результаты.....	8
Скриншоты	9
Листинг	10
Список литературы	15

Задание:

Импорт данных из SQL-базы в сверстаный PDF документ [1].

Условия:

Исследование формата PDF и его возможностей. Анализ дополнительных возможностей по импорту и экспорту данных в PDF. Исследование методов импорта данных в PDF из SQL-базы

Существующие аналоги

Был найден неплохой платный аналог на windows под названием Advanced SQL To PDF Table Converter [2]. В нём есть выбор типа базы данных, данные для подключения, параметры вёрстки выходного файла. Программа не запоминает последних введённых значений, в ней нельзя сохранять настройки. Проблемой программы является отображение кириллических символов символами вопроса. После генерации PDF-файла открывается сгенерированный файл.

Скриншоты

```
mysql> select * from city
-> ;
```

id_city	name
1	Санкт-Петербург
2	Москва
3	Великий Новгород
4	Самара
5	Казань

Рисунок 1: Таблица в mysql

The screenshot shows the 'Advanced SQL To PDF Table Converter' application window. It features a multi-step configuration process:

- Step 1 - Database Type:** Radio buttons for Microsoft SQL Server, MySQL (selected), and Oracle.
- Step 2 - Database Connection Settings:** Radio buttons for Microsoft SQL Server Authentication (selected) and Windows Authentication. Fields for Database Server (127.0.0.1), User ID (root), Database Name (test), and Password (****).
- Step 3 - SQL Select Query:** A text area containing 'select * from city'.
- Step 4 (Optional) - Column Headers:** A checkbox for 'Insert Column Headers' (unchecked). A table with one header 'Column Headers' and two empty rows. Buttons for 'Enter column header...', 'Add Column Header', and 'Remove Column Header'.
- Step 5 - Output PDF File Properties:** A dropdown for 'PDF Page Size' (Letter), radio buttons for 'PDF Page Orientation' (Portrait selected, Landscape), a dropdown for 'PDF Font' (Arial), and a dropdown for 'PDF Font Size' (12).
- Step 6 - Output Directory:** A text field showing 'C:\Users\mgfile_000\Documents'.
- Step 7 (Optional) - Output PDF File Name:** A text field for 'Output PDF File Name'.
- Step 8 - Execute SQL Query / Generate PDF Output:** A large button labeled 'Execute SQL Query / Generate PDF Output'.

Рисунок 2: Интерфейс Advanced SQL To PDF

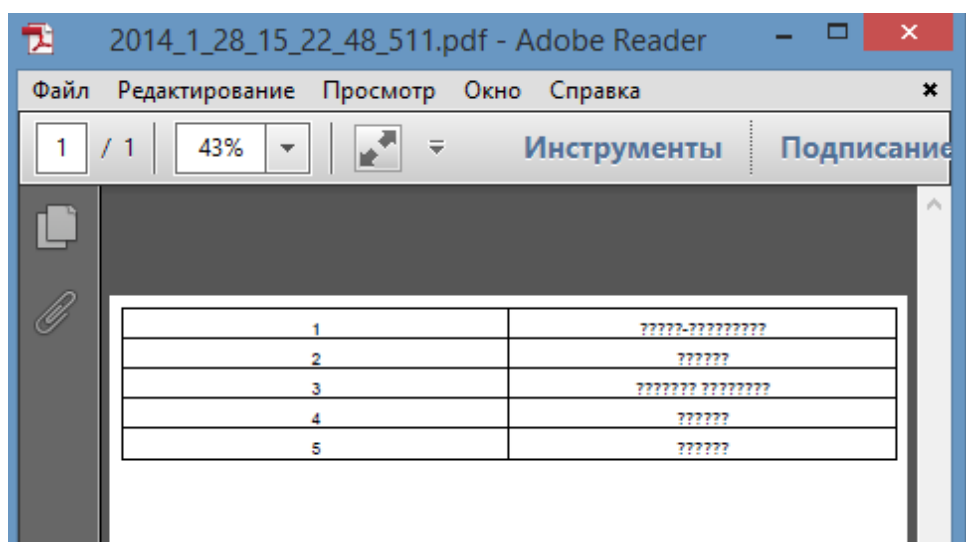


Рисунок 3: Сгенерированный pdf-файл

Возможности реализации

На python существуют собственные средства для работы с базами данных [3], существует библиотека MySQLdb, но она содержит проблемы с кодировками [4]. Достаточно удобной для работы с mysql является библиотека Connector/Python [5] из-за достаточно наглядных примеров для работы с ней и подробной инструкцией установки самой библиотеки. Для работы с PDF существует тоже достаточно большое количество библиотек [6,7], но судя по комментариям в интернете одной из самых удобных является reportlab. Графический интерфейс на python можно реализовать с помощью qt, wx, gtk и множества других библиотек.

Ход работы

- 1) Создание интерфейса:
 - a) Окно программы
 - b) Поля для ввода данных для соединения с mysql
 - c) Кнопка, по нажатию на которую берутся данные из sql и преобразует их в PDF
- 2) Реализация функции по извлечению данных из mysql
- 3) Реализация функции по конвертированию полученной таблицы в PDF-файл

Результаты

Программу можно скачать из репозитория
https://github.com/fleytman/sql_to_PDF.

Все требования указаны в файле README.md

Написанная программа имеет 5 полей для ввода:

- 1) adress – для ввода адреса сервера
- 2) login – для ввода логина на сервере
- 3) password – для ввода пароля (введённые данные показываются точками чтобы пароль нельзя было подсмотреть)
- 4) database — для выбора базы на сервере
- 5) query sql – для указания запроса к серверу

В программе имеется 1 кнопка convert, которая преобразует таблицу из базы данных в файл converted.PDF. В программе имеется проблема с отображение кириллицы [8].

Скриншоты

sql_to_dpf

adress 127.0.0.1

login root

password ●●●●

database test

query sql

SELECT * FROM `city` ORDER BY `name` DESC

convert

Рисунок 4: Окно программы

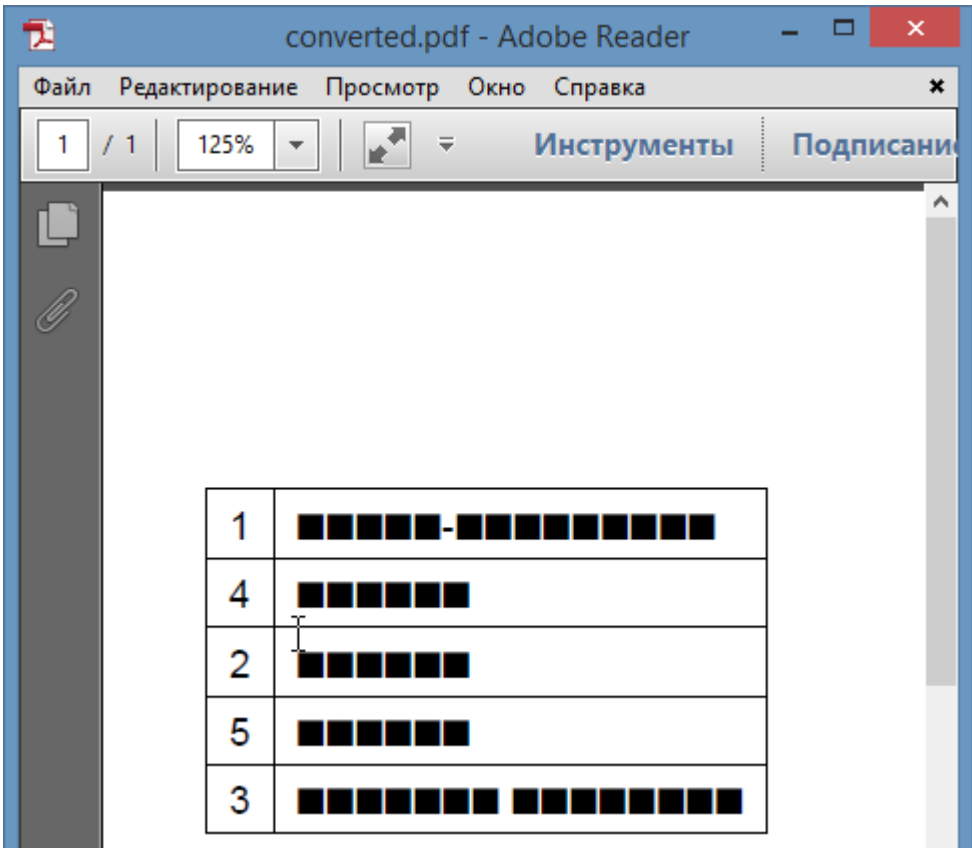


Рисунок 5: Сгенерированный PDF-файл

Листинг

```
# -*- coding: utf-8 -*-

from reportlab.lib import colors
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Table, TableStyle

import gtk
import mysql.connector
#~ import os
import sys
import locale

def on_convert_pressed(calcbutton, adress_entry, login_entry, password_entry,db_entry,query_entry):

    adress = adress_entry.get_text()

    login = login_entry.get_text()

    password = password_entry.get_text()

    db = db_entry.get_text()

    query = query_entry.get_text()

    data = connect__to_server(adress, login, password,db,query)

    convert_to_PDF(data)

    #~ os.system('converted.PDF')

    print "do"

def main():

    window = gtk.Window()

    window.set_default_size(300,200)

    window.set_title(u"sql_to_dpf")

    mainbox = gtk.VBox()

    window.add(mainbox)

    adress_box = gtk.HBox()
```

```
mainbox.pack_start(adress_box, expand=False)

login_box = gtk.HBox()

mainbox.pack_start(login_box, expand=False)

password_box = gtk.HBox()

mainbox.pack_start(password_box, expand=False)

db_box = gtk.HBox()

mainbox.pack_start(db_box, expand=False)

query_box = gtk.VBox()

mainbox.pack_start(query_box, expand=False)


adress_label = gtk.Label(u"adress")

adress_box.pack_start(adress_label)

adress_entry = gtk.Entry()

adress_entry.set_text(u"127.0.0.1")

adress_box.pack_start(adress_entry)


login_label = gtk.Label(u"login")

login_box.pack_start(login_label)

login_entry = gtk.Entry()

login_entry.set_text(u"root")

login_box.pack_start(login_entry)


password_label = gtk.Label(u"password")

password_box.pack_start(password_label)

password_entry = gtk.Entry()

password_entry.set_text(u"root")

password_entry.set_visibility(0) # поле для пароля показывает звёздочки

password_box.pack_start(password_entry)


db_label = gtk.Label(u"database")

db_box.pack_start(db_label)

db_entry = gtk.Entry()

db_entry.set_text(u"test")

db_box.pack_start(db_entry)


query_label = gtk.Label(u"query sql")
```

```

query_box.pack_start(query_label)

query_entry = gtk.Entry()

query_entry.set_text(u"SELECT * FROM `city` ORDER BY `name` DESC")

query_box.pack_start(query_entry)


convert_button = gtk.Button(u"convert")

mainbox.pack_start(convert_button, expand=False)


d_flabel = gtk.Label(u"")

mainbox.pack_start(d_flabel, expand=False)


#sw = gtk.ScrolledWindow()

#sw.set_policy(gtk.POLICY_AUTOMATIC, gtk.POLICY_AUTOMATIC)


window.connect("destroy", lambda _: gtk.main_quit())

convert_button.connect("clicked", on_convert_pressed, adress_entry, login_entry,
password_entry, db_entry, query_entry)


window.show_all()

gtk.main()


def connect__to_server(adress, login, password_db, db, query): #Данная функция соединяется с сервером, уведомляет
об успешном соединении

    from mysql.connector import errorcode

    if sys.stdin.encoding: encoding = sys.stdin.encoding

    else: encoding = locale.getdefaultlocale()[1]

    try:

        cnx = mysql.connector.connect(user=login, password=password_db, host=adress, database=db)

        cursor = cnx.cursor()

        #~ query = ('SELECT * FROM `city` ORDER BY `name` DESC')

        cursor.execute(query)

        data = []

        for name in cursor:

            i = 0

```

```

        k = len(name)

        #~ string=""

        spisok=[]

        while i < k:

            #~ string+= unicode(name[i])

            spisok.append(name[i])

            i+=1

        #~ print string

        data.append(spisok)

        print data

        print "\n"

except mysql.connector.Error as err:

    if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:

        print("Something is wrong with your user name or password")

    elif err.errno == errorcode.ER_BAD_DB_ERROR:

        print("Database does not exists")

    else:

        print(err)

else:

    cursor.close()

    cnx.close()

return data

```

```

def convert_to_PDF(data): #данная функция превращает данные в PDF

    doc = SimpleDocTemplate("converted.PDF", pagesize=letter)

    # container for the 'Flowable' objects

    elements = []

    t=Table(data)

    t.setStyle(TableStyle([('INNERGRID', (0,0), (-1,-1), 0.25, colors.black),

        ('BOX', (0,0), (-1,-1), 0.25, colors.black),

        ]))

    elements.append(t)

    # write the document to disk

    doc.build(elements)

```

```
if __name__ == "__main__":  
    main()
```

Список литературы

1. Королёв Д.А. // Компьютерная графика: Курсовая работа: [сайт]. [2013]. URL: http://cg-2013.blogspot.ru/p/blog-page_3.html (дата обращения: 6.10.2013).
2. Advanced Reliable Software. // Advanced SQL To PDF Table Converter: [сайт]. [2014]. URL: http://www.advancedreliablesoftware.com/advanced_sql_to_pdf_table_converter.html
3. Паутов А.В. MySQL: Руководство разработчика // Python Database API v2.0. 2004. URL: <http://www.rldp.ru/mysql/mysqldev/dbapi20.htm>
4. Morbo. // Проблема с кодировками в пакете MySQLdb: [сайт]. [2013]. URL: <http://vladimir-stupin.blogspot.ru/2013/02/mysqlldb.html>
5. Oracle and/or its affiliates. // MySQL Connector/Python Developer Guide: [сайт]. [2014]. URL: <https://dev.mysql.com/doc/connector-python/en/index.html>
6. Fenniak M. // pyPdf: [сайт]. [2008]. URL: <http://pybrary.net/pyPdf/>
7. Anari A. // metaPdf: [сайт]. [2012]. URL: <https://github.com/aanari/metaPdf>
8. Camel. // Создание PDF'а с кириллицей python'ом: [сайт]. [2010]. URL: <https://www.linux.org.ru/forum/development/4875828>
9. Thornton House T.R.W. ReportLab PDF Library 2013. URL: <http://www.reportlab.com/docs/reportlab-userguide.pdf>
10. PyGTK Team. // PyGTK FAQ: [сайт]. [2013]. URL: <http://faq.pygtk.org/index.py?req=index>