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

Оглавлеsние

[Задание: 3](#_Toc379189724)

[Условия: 3](#_Toc379189725)

[Существующие аналоги 3](#_Toc379189726)

[Скриншоты 4](#_Toc379189727)

[Возможности реализации 6](#_Toc379189728)

[Ход работы 7](#_Toc379189729)

[Результаты 8](#_Toc379189730)

[Скриншоты 9](#_Toc379189731)

[Листинг 10](#_Toc379189732)

[Список литературы 15](#_Toc379189733)

# Задание:

Импорт данных из SQL-базы в свёрстанный 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

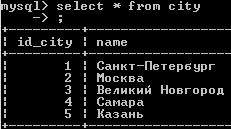


Рисунок : Таблица в mysql

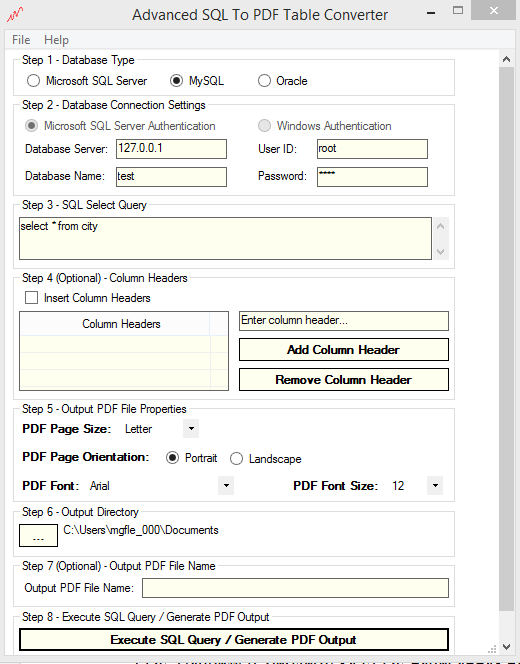


Рисунок 2: Интерфейс Advanced SQL To PDF

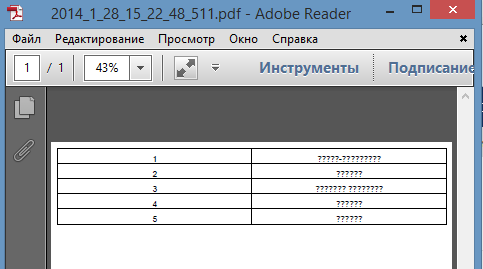


Рисунок : Сгенерированный pdf-файл

## 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 библиотки. 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:
   1. Окно программы
   2. Поля для ввода данных для соединения с mysql
   3. 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](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 показыются 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

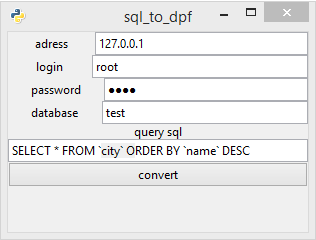


Рисунок : Окно программы

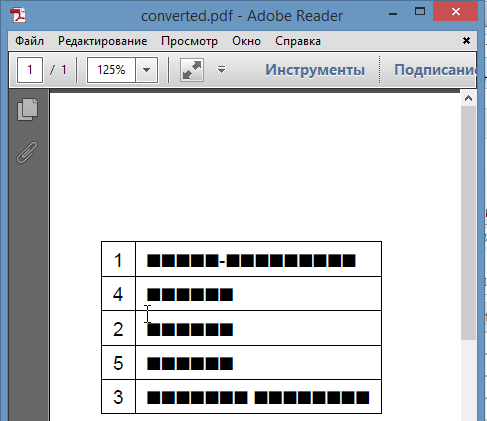


Рисунок : Сгенерированный PDF-файл

# 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 соединение

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

x

|  |  |
| --- | --- |
| 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/​mysqldb.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 |

x