Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет среднего профессионального образования

**ОТЧЕТ**

**о Лабораторной работе № 10**

по теме: РАЗРАБОТКА ФОРМ ДЛЯ ВСТАВКИ, ВЫБОРКИ И РЕДАКТИРОВАНИЯ ДАННЫХ

по дисциплине: Основы проектирования баз данных

Специальность:

09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Проверил:  Говорова М.М. \_\_\_\_\_\_\_\_\_\_  Дата: «04» июня 2019г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2336  \_\_\_\_\_\_ Заерко М.В. |

Санкт-Петербург 2019

1. Цель работы

Овладеть практическими навыками разработки форм для вставки, выборки и редактирования данных.

1. Практическое задание

Практическое задание состоит из следующих пунктов:

1. Изучить функции открытия соединения к базе данных средствами python.
2. Изучить основные функции для работы с SQL-запросами.
3. Разработать скрипты для модификации и выборки данных средствами python.
4. Выполнение лабораторной работы
5. Python cкрипт и результаты его выполнения

Python скрипт представлен на листинге 1, результат выполнения скрипта представлен на рисунке 1.

Листинг 1 – Скрипт работы с таблицей из базы данных

import pg8000 as connector

conn = connector.connect(user="postgres", password="1234", database="auto\_base")

cursor = conn.cursor()

acursor = conn.cursor()

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(QtWidgets.QWidget):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(800, 600)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.gridLayout = QtWidgets.QGridLayout(self.centralwidget)

self.gridLayout.setObjectName("gridLayout")

self.scrollArea = QtWidgets.QScrollArea(self.centralwidget)

self.scrollArea.setWidgetResizable(True)

self.scrollArea.setObjectName("scrollArea")

self.scrollAreaWidgetContents = QtWidgets.QWidget()

self.scrollAreaWidgetContents.setGeometry(QtCore.QRect(0, 0, 780, 539))

self.scrollAreaWidgetContents.setObjectName("scrollAreaWidgetContents")

self.gridLayout\_2 = QtWidgets.QGridLayout(self.scrollAreaWidgetContents)

self.gridLayout\_2.setObjectName("gridLayout\_2")

self.scrollArea.setWidget(self.scrollAreaWidgetContents)

self.gridLayout.addWidget(self.scrollArea, 0, 0, 1, 1)

MainWindow.setCentralWidget(self.centralwidget)

self.menubar = QtWidgets.QMenuBar(MainWindow)

self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))

self.menubar.setObjectName("menubar")

MainWindow.setMenuBar(self.menubar)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi()

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MainWindow"))

self.display\_data()

def display\_data(self):

for i in reversed(range(self.gridLayout\_2.count())):

self.gridLayout\_2.itemAt(i).widget().deleteLater()

i = 0

j = 0

labels = ["id машины","Регистрационный номер","Марка","Номер двигателя", "Тип","Пробег"]

for label in labels:

self.gridLayout\_2.addWidget(QtWidgets.QLabel(label), i, j, 1, 1)

j += 1

j = 0

query = "select \* from public.auto1"

cursor.execute(query)

i += 1

for item in cursor:

for value in item:

if j == 0:

id = value

j += 1

continue

line\_edit = QtWidgets.QLineEdit(str(value))

self.gridLayout\_2.addWidget(line\_edit, i, j, 1, 1)

buf = line\_edit

line\_edit.textChanged.connect(lambda state, line=[id, buf, j]: modify(line))

j += 1

button = QtWidgets.QPushButton("Удалить")

button.clicked.connect(lambda state, line=id: delete(line))

self.gridLayout\_2.addWidget(button, i, j+1, 1, 1)

i += 1

j = 0

def add():

pass

query = "select id\_auto from public.auto1 order by id\_auto desc limit 1;"

cursor.execute(query)

for item in cursor:

for value in item:

id = int(value) + 1

user = []

text, okPressed = QtWidgets.QInputDialog.getText(self, "Get text", "Регистрационный номер:", QtWidgets.QLineEdit.Normal, "")

if okPressed and text != '':

user.append(text)

text, okPressed = QtWidgets.QInputDialog.getText(self, "Get text", "Марка:", QtWidgets.QLineEdit.Normal, "")

if okPressed and text != '':

user.append(text)

text, okPressed = QtWidgets.QInputDialog.getText(self, "Get text", "Двигатель:", QtWidgets.QLineEdit.Normal, "")

if okPressed and text != '':

user.append(text)

text, okPressed = QtWidgets.QInputDialog.getText(self, "Get text", "Тип:", QtWidgets.QLineEdit.Normal, "")

if okPressed and text != '':

user.append(text)

text, okPressed = QtWidgets.QInputDialog.getText(self, "Get text", "Дистанция:", QtWidgets.QLineEdit.Normal, "")

if okPressed and text != '':

user.append(text)

data = (str(id), user[0], user[1], user[2], user[3], user[4])

query = "insert into public.auto1 values (%s, %s, %s, %s, %s, %s);"

cursor.execute(query, data)

conn.commit

self.display\_data()

add\_button = QtWidgets.QPushButton("Добавить")

add\_button.clicked.connect(add)

self.gridLayout\_2.addWidget(add\_button, i, j, 1, 1)

def modify(line):

j = line[2]

data = (line[1].text(), line[0])

if j == 1:

query = "update public.auto1 set n\_reg\_auto = %s where id\_auto = %s;"

if j == 2:

query = "update public.auto1 set mark = %s where id\_auto = %s;"

if j == 3:

query = "update public.auto1 set n\_engine = %s where id\_auto = %s;"

if j == 4:

query = "update public.auto1 set type = %s where id\_auto = %s;"

if j == 5:

query = "update public.auto1 set distance = %s where id\_auto = %s;"

try:

cursor.execute(query, data)

conn.commit()

except BaseException as e:

print(str(e))

pass

self.display\_data()

def delete(id):

query = "delete from public.auto1 where id\_auto = %s;"

data = (id, )

cursor.execute(query, data)

conn.commit()

self.display\_data()

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

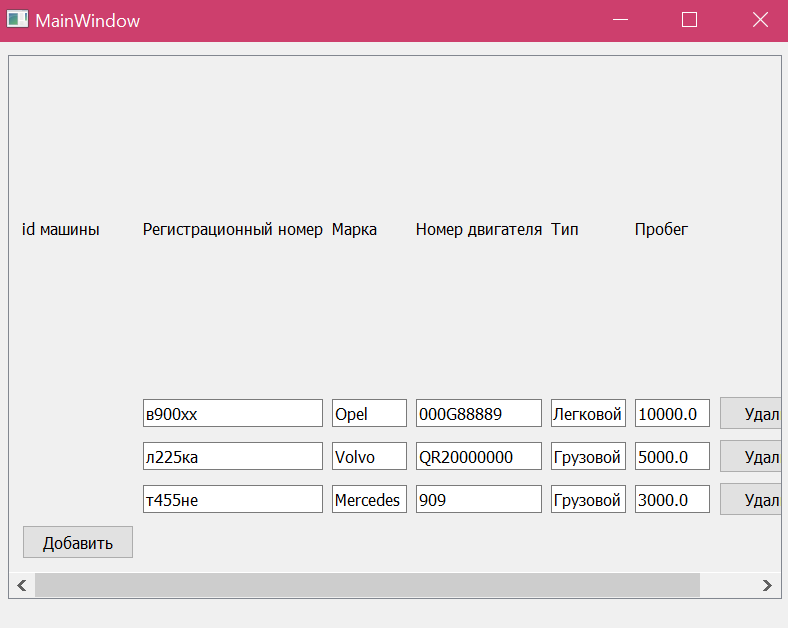


Рисунок 1 – Результаты работы python-скрипта

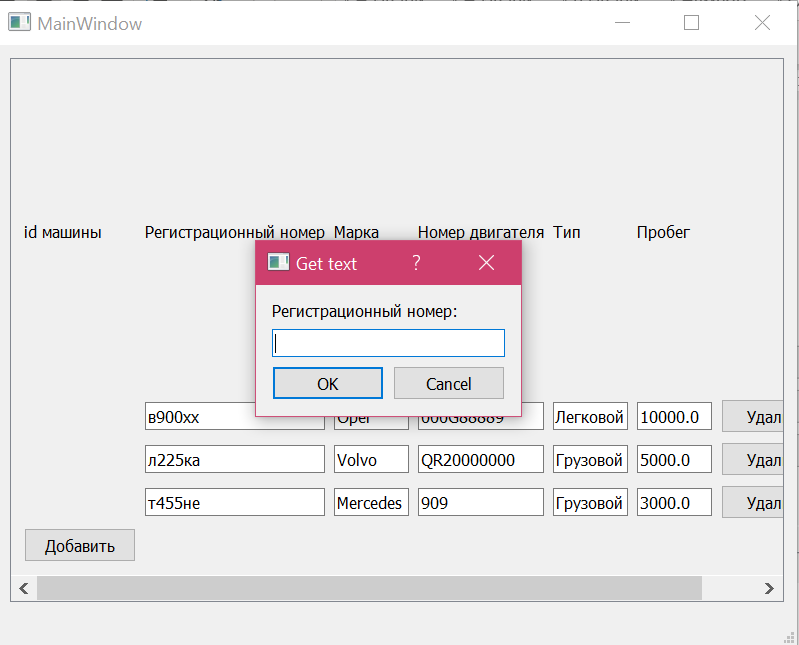


Рисунок 2 – Результаты работы python-скрипта

1. Вывод

В процессе выполнения лабораторной работы были приобретены практические навыки создания скриптов на языке python для вставки, редактирования и изменения данных.