**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базы данных»

Отчет по лабораторной работе №4

«Создание оконного приложения для работы с базой данных»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-45Б |  | преподаватель каф. ИУ5 |
| Коновалов И.Н. |  | Силантьева Е.Ю. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

Цель лабораторной работы

Создать в среде Qt Creator на языке Qt Python оконное приложение для работы с базой данных.

1. Создать любое простое представление и запросить с помощью него данные.

QT5 ставится командами:

sudo apt install qt5-default  
sudo apt install qtbase5-examples qtdeclarative5-examples

Драйвер ставится командой:

sudo apt install libqt5sql5-psql (Для QT5)

QTCreator ставится командой:

sudo apt-get install qtcreator

Импортируем библиотеки:



Программный код файла AccidentsCRUD, который осуществляет все запросы к базе данных:

*import* psycopg2  
  
*def* insert(*accident\_id*: *int*, *conditions*: *str*, *accident\_date*: *str*, *place*: *str*, *officer\_id*: *int*, *city*: *str*, *conn*):  
 cur = *conn*.cursor()  
 cur.execute("INSERT INTO accident (accident\_id, conditions, accident\_date, place, officer\_id, city) VALUES (*%s*, *%s*, *%s*, *%s*, *%s*, *%s*)", (*accident\_id*, *conditions*, *accident\_date*, *place*, *officer\_id*, *city*))  
 *conn*.commit()  
 cur.close()  
  
*def* select(*accident\_date*: *str*, *place*: *str*, *conn*):  
 cur = *conn*.cursor()  
 cur.execute("SELECT \* FROM accident WHERE accident\_date = (*%s*) AND place = (*%s*)", (*accident\_date*,*place*))  
 answer = cur.fetchall()  
 *return* answer  
 cur.close()  
  
*def* select\_all(*conn*):  
 cur = *conn*.cursor()  
 cur.execute("SELECT \* FROM accident ORDER BY accident\_id")  
 answer = cur.fetchall()  
 *return* answer  
 cur.close()  
  
*def* update(*accident\_id*: *int*, *conditions*: *str*, *accident\_date*: *str*, *place*: *str*, *officer\_id*: *int*, *conn*):  
 cur = *conn*.cursor()  
 cur.execute("UPDATE accident SET conditions = (*%s*), accident\_date = (*%s*), place = (*%s*), officer\_id = (*%s*) WHERE accident\_id = (*%s*)", (*conditions*, *accident\_date*, *place*, *officer\_id*, *accident\_id*))  
 *conn*.commit()  
 cur.close()  
  
*def* delete(*accident\_id*: *int*, *conn*):  
 cur = *conn*.cursor()  
 cur.execute("DELETE FROM accident WHERE accident\_id = (*%s*)", (*accident\_id*,))  
 *conn*.commit()  
 cur.close()

Программный код файла dialogAccident.py, который выводит изображение на экран и производит базовую связь содержимого:

*# -\*- coding: utf-8 -\*-  
  
# Form implementation generated from reading ui file 'dialogAccident.ui'  
#  
# Created by: PyQt5 UI code generator 5.15.6  
#  
# WARNING: Any manual changes made to this file will be lost when pyuic5 is  
# run again. Do not edit this file unless you know what you are doing.  
  
  
from* PyQt6 *import* QtCore, QtGui, QtWidgets  
  
  
*class* Ui\_AccidentDialog(*object*):  
 *def* setupUi(*self*, *Dialog*):  
 *Dialog*.setObjectName("Dialog")  
 *Dialog*.resize(815, 543)  
 *self*.label = QtWidgets.QLabel(*Dialog*)  
 *self*.label.setGeometry(QtCore.QRect(-3, -5, 821, 551))  
 *self*.label.setText("")  
 *self*.label.setPixmap(QtGui.QPixmap("../photos/imgonline-com-ua-Blur-2l4fVmT8e17N.jpg"))  
 *self*.label.setObjectName("label")  
 *self*.tabWidget = QtWidgets.QTabWidget(*Dialog*)  
 *self*.tabWidget.setGeometry(QtCore.QRect(0, 0, 811, 541))  
 *self*.tabWidget.setMinimumSize(QtCore.QSize(0, 541))  
 *self*.tabWidget.setMaximumSize(QtCore.QSize(16777215, 541))  
 *self*.tabWidget.setTabBarAutoHide(*False*)  
 *self*.tabWidget.setObjectName("tabWidget")  
 *self*.selRow = QtWidgets.QWidget()  
 *self*.selRow.setObjectName("selRow")  
 *self*.selBack = QtWidgets.QLabel(*self*.selRow)  
 *self*.selBack.setGeometry(QtCore.QRect(-3, -5, 811, 521))  
 *self*.selBack.setText("")  
 *self*.selBack.setPixmap(QtGui.QPixmap("../photos/imgonline-com-ua-Blur-2l4fVmT8e17N.jpg"))  
 *self*.selBack.setObjectName("selBack")  
 *self*.selTableAccident = QtWidgets.QTableView(*self*.selRow)  
 *self*.selTableAccident.setGeometry(QtCore.QRect(40, 90, 721, 381))  
 *self*.selTableAccident.setStyleSheet("color: grey; background-color: rgba(255,255,255,175)")  
 *self*.selTableAccident.setObjectName("selTableAccident")  
 *self*.selUpperBack = QtWidgets.QLabel(*self*.selRow)  
 *self*.selUpperBack.setGeometry(QtCore.QRect(0, 0, 811, 61))  
 *self*.selUpperBack.setStyleSheet("background-color: rgba(255,255,255,75)")  
 *self*.selUpperBack.setText("")  
 *self*.selUpperBack.setObjectName("selUpperBack")  
 *self*.searchDate = QtWidgets.QLineEdit(*self*.selRow)  
 *self*.searchDate.setGeometry(QtCore.QRect(10, 10, 211, 41))  
 *self*.searchDate.setStyleSheet("background-color: rgba(255,255,255,50)")  
 *self*.searchDate.setObjectName("searchDate")  
 *self*.searchButton = QtWidgets.QPushButton(*self*.selRow)  
 *self*.searchButton.setGeometry(QtCore.QRect(490, 10, 131, 41))  
 *self*.searchButton.setObjectName("searchButton")  
 *self*.selReturn = QtWidgets.QPushButton(*self*.selRow)  
 *self*.selReturn.setGeometry(QtCore.QRect(660, 10, 131, 41))  
 *self*.selReturn.setObjectName("selReturn")  
 *self*.searchPlace = QtWidgets.QLineEdit(*self*.selRow)  
 *self*.searchPlace.setGeometry(QtCore.QRect(250, 10, 211, 41))  
 *self*.searchPlace.setStyleSheet("background-color: rgba(255,255,255,50)")  
 *self*.searchPlace.setObjectName("searchPlace")  
 *self*.tabWidget.addTab(*self*.selRow, "")  
 *self*.addRow = QtWidgets.QWidget()  
 *self*.addRow.setObjectName("addRow")  
 *self*.addBack = QtWidgets.QLabel(*self*.addRow)  
 *self*.addBack.setGeometry(QtCore.QRect(-3, -5, 811, 521))  
 *self*.addBack.setText("")  
 *self*.addBack.setPixmap(QtGui.QPixmap("../photos/imgonline-com-ua-Blur-2l4fVmT8e17N.jpg"))  
 *self*.addBack.setObjectName("addBack")  
 *self*.addUpperBack = QtWidgets.QLabel(*self*.addRow)  
 *self*.addUpperBack.setGeometry(QtCore.QRect(0, 0, 811, 61))  
 *self*.addUpperBack.setStyleSheet("background-color: rgba(255,255,255,75)")  
 *self*.addUpperBack.setText("")  
 *self*.addUpperBack.setObjectName("addUpperBack")  
 *self*.addReturn = QtWidgets.QPushButton(*self*.addRow)  
 *self*.addReturn.setGeometry(QtCore.QRect(660, 10, 131, 41))  
 *self*.addReturn.setObjectName("addReturn")  
 *self*.middleBack = QtWidgets.QLabel(*self*.addRow)  
 *self*.middleBack.setGeometry(QtCore.QRect(10, 80, 781, 421))  
 *self*.middleBack.setStyleSheet("background-color: rgba(255,255,255,150)")  
 *self*.middleBack.setText("")  
 *self*.middleBack.setObjectName("middleBack")  
 *self*.addAccidentIdInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addAccidentIdInput.setGeometry(QtCore.QRect(60, 90, 251, 41))  
 *self*.addAccidentIdInput.setAutoFillBackground(*False*)  
 *self*.addAccidentIdInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addAccidentIdInput.setInputMask("")  
 *self*.addAccidentIdInput.setText("")  
 *self*.addAccidentIdInput.setObjectName("addAccidentIdInput")  
 *self*.addConditionsInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addConditionsInput.setGeometry(QtCore.QRect(480, 90, 251, 41))  
 *self*.addConditionsInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addConditionsInput.setObjectName("addConditionsInput")  
 *self*.addAccidentDateInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addAccidentDateInput.setGeometry(QtCore.QRect(60, 140, 251, 41))  
 *self*.addAccidentDateInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addAccidentDateInput.setObjectName("addAccidentDateInput")  
 *self*.addPlaceInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addPlaceInput.setGeometry(QtCore.QRect(480, 140, 251, 41))  
 *self*.addPlaceInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addPlaceInput.setObjectName("addPlaceInput")  
 *self*.addTableAccident = QtWidgets.QTableView(*self*.addRow)  
 *self*.addTableAccident.setGeometry(QtCore.QRect(30, 250, 741, 241))  
 *self*.addTableAccident.setStyleSheet("color: grey; background-color: rgba(255,255,255,175)\n"  
"")  
 *self*.addTableAccident.setObjectName("addTableAccident")  
 *self*.addInsert = QtWidgets.QPushButton(*self*.addRow)  
 *self*.addInsert.setGeometry(QtCore.QRect(330, 160, 131, 41))  
 *self*.addInsert.setObjectName("addInsert")  
 *self*.addOfficerIdInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addOfficerIdInput.setGeometry(QtCore.QRect(60, 190, 251, 41))  
 *self*.addOfficerIdInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addOfficerIdInput.setObjectName("addOfficerIdInput")  
 *self*.addCityInput = QtWidgets.QLineEdit(*self*.addRow)  
 *self*.addCityInput.setGeometry(QtCore.QRect(480, 190, 251, 41))  
 *self*.addCityInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.addCityInput.setObjectName("addCityInput")  
 *self*.tabWidget.addTab(*self*.addRow, "")  
 *self*.updRow = QtWidgets.QWidget()  
 *self*.updRow.setObjectName("updRow")  
 *self*.updBack = QtWidgets.QLabel(*self*.updRow)  
 *self*.updBack.setGeometry(QtCore.QRect(-3, -5, 811, 521))  
 *self*.updBack.setText("")  
 *self*.updBack.setPixmap(QtGui.QPixmap("../photos/imgonline-com-ua-Blur-2l4fVmT8e17N.jpg"))  
 *self*.updBack.setObjectName("updBack")  
 *self*.updUpperBack = QtWidgets.QLabel(*self*.updRow)  
 *self*.updUpperBack.setGeometry(QtCore.QRect(0, 0, 811, 61))  
 *self*.updUpperBack.setStyleSheet("background-color: rgba(255,255,255,75)")  
 *self*.updUpperBack.setText("")  
 *self*.updUpperBack.setObjectName("updUpperBack")  
 *self*.updReturn = QtWidgets.QPushButton(*self*.updRow)  
 *self*.updReturn.setGeometry(QtCore.QRect(660, 10, 131, 41))  
 *self*.updReturn.setObjectName("updReturn")  
 *self*.updMiddleBack = QtWidgets.QLabel(*self*.updRow)  
 *self*.updMiddleBack.setGeometry(QtCore.QRect(10, 80, 781, 421))  
 *self*.updMiddleBack.setStyleSheet("background-color: rgba(255,255,255,150)")  
 *self*.updMiddleBack.setText("")  
 *self*.updMiddleBack.setObjectName("updMiddleBack")  
 *self*.updChange = QtWidgets.QPushButton(*self*.updRow)  
 *self*.updChange.setGeometry(QtCore.QRect(330, 160, 131, 41))  
 *self*.updChange.setObjectName("updChange")  
 *self*.updTableAccident = QtWidgets.QTableView(*self*.updRow)  
 *self*.updTableAccident.setGeometry(QtCore.QRect(30, 250, 741, 241))  
 *self*.updTableAccident.setStyleSheet("color: grey; background-color: rgba(255,255,255,175)")  
 *self*.updTableAccident.setObjectName("updTableAccident")  
 *self*.updAccidentIdInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updAccidentIdInput.setGeometry(QtCore.QRect(60, 90, 251, 41))  
 *self*.updAccidentIdInput.setAutoFillBackground(*False*)  
 *self*.updAccidentIdInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updAccidentIdInput.setInputMask("")  
 *self*.updAccidentIdInput.setText("")  
 *self*.updAccidentIdInput.setObjectName("updAccidentIdInput")  
 *self*.updAccidentDateInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updAccidentDateInput.setGeometry(QtCore.QRect(60, 140, 251, 41))  
 *self*.updAccidentDateInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updAccidentDateInput.setObjectName("updAccidentDateInput")  
 *self*.updOfficerIdInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updOfficerIdInput.setGeometry(QtCore.QRect(60, 190, 251, 41))  
 *self*.updOfficerIdInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updOfficerIdInput.setObjectName("updOfficerIdInput")  
 *self*.updConditionsInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updConditionsInput.setGeometry(QtCore.QRect(480, 90, 251, 41))  
 *self*.updConditionsInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updConditionsInput.setObjectName("updConditionsInput")  
 *self*.updPlaceInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updPlaceInput.setGeometry(QtCore.QRect(480, 140, 251, 41))  
 *self*.updPlaceInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updPlaceInput.setObjectName("updPlaceInput")  
 *self*.updCityInput = QtWidgets.QLineEdit(*self*.updRow)  
 *self*.updCityInput.setGeometry(QtCore.QRect(480, 190, 251, 41))  
 *self*.updCityInput.setStyleSheet("background-color: rgba(255,255,255,100)")  
 *self*.updCityInput.setObjectName("updCityInput")  
 *self*.tabWidget.addTab(*self*.updRow, "")  
 *self*.delRow = QtWidgets.QWidget()  
 *self*.delRow.setObjectName("delRow")  
 *self*.delBack = QtWidgets.QLabel(*self*.delRow)  
 *self*.delBack.setGeometry(QtCore.QRect(-3, -5, 811, 521))  
 *self*.delBack.setText("")  
 *self*.delBack.setPixmap(QtGui.QPixmap("../photos/imgonline-com-ua-Blur-2l4fVmT8e17N.jpg"))  
 *self*.delBack.setObjectName("delBack")  
 *self*.delUpperBack = QtWidgets.QLabel(*self*.delRow)  
 *self*.delUpperBack.setGeometry(QtCore.QRect(0, 0, 811, 61))  
 *self*.delUpperBack.setStyleSheet("background-color: rgba(255,255,255,75)\n"  
"")  
 *self*.delUpperBack.setText("")  
 *self*.delUpperBack.setObjectName("delUpperBack")  
 *self*.delReturn = QtWidgets.QPushButton(*self*.delRow)  
 *self*.delReturn.setGeometry(QtCore.QRect(660, 10, 131, 41))  
 *self*.delReturn.setObjectName("delReturn")  
 *self*.delNameText = QtWidgets.QLineEdit(*self*.delRow)  
 *self*.delNameText.setGeometry(QtCore.QRect(10, 10, 301, 41))  
 *self*.delNameText.setStyleSheet("background-color: rgba(255,255,255,50)")  
 *self*.delNameText.setObjectName("delNameText")  
 *self*.deleteButton = QtWidgets.QPushButton(*self*.delRow)  
 *self*.deleteButton.setGeometry(QtCore.QRect(320, 10, 131, 41))  
 *self*.deleteButton.setObjectName("deleteButton")  
 *self*.delTableAccident = QtWidgets.QTableView(*self*.delRow)  
 *self*.delTableAccident.setGeometry(QtCore.QRect(40, 90, 721, 381))  
 *self*.delTableAccident.setStyleSheet("color: grey; background-color: rgba(255,255,255,175)")  
 *self*.delTableAccident.setObjectName("delTableAccident")  
 *self*.tabWidget.addTab(*self*.delRow, "")  
  
 *self*.retranslateUi(*Dialog*)  
 *self*.tabWidget.setCurrentIndex(0)  
 QtCore.QMetaObject.connectSlotsByName(*Dialog*)  
  
 *def* retranslateUi(*self*, *Dialog*):  
 \_translate = QtCore.QCoreApplication.translate  
 *Dialog*.setWindowTitle(\_translate("Dialog", "Dialog"))  
 *self*.searchDate.setPlaceholderText(\_translate("Dialog", " Дата"))  
 *self*.searchButton.setText(\_translate("Dialog", "Поиск"))  
 *self*.selReturn.setText(\_translate("Dialog", "Назад"))  
 *self*.searchPlace.setPlaceholderText(\_translate("Dialog", " Место"))  
 *self*.tabWidget.setTabText(*self*.tabWidget.indexOf(*self*.selRow), \_translate("Dialog", "Просмотр"))  
 *self*.addReturn.setText(\_translate("Dialog", "Назад"))  
 *self*.addAccidentIdInput.setPlaceholderText(\_translate("Dialog", " ID"))  
 *self*.addConditionsInput.setPlaceholderText(\_translate("Dialog", " Погодные условия"))  
 *self*.addAccidentDateInput.setPlaceholderText(\_translate("Dialog", " Дата происшествия"))  
 *self*.addPlaceInput.setPlaceholderText(\_translate("Dialog", " Место ДТП"))  
 *self*.addInsert.setText(\_translate("Dialog", "Добавить"))  
 *self*.addOfficerIdInput.setPlaceholderText(\_translate("Dialog", " ID сотрудника"))  
 *self*.addCityInput.setPlaceholderText(\_translate("Dialog", " Город"))  
 *self*.tabWidget.setTabText(*self*.tabWidget.indexOf(*self*.addRow), \_translate("Dialog", "Добавление"))  
 *self*.updReturn.setText(\_translate("Dialog", "Назад"))  
 *self*.updChange.setText(\_translate("Dialog", "Изменить"))  
 *self*.updAccidentIdInput.setPlaceholderText(\_translate("Dialog", " ID"))  
 *self*.updAccidentDateInput.setPlaceholderText(\_translate("Dialog", " Дата происшествия"))  
 *self*.updOfficerIdInput.setPlaceholderText(\_translate("Dialog", " ID сотрудника"))  
 *self*.updConditionsInput.setPlaceholderText(\_translate("Dialog", " Погодные условия"))  
 *self*.updPlaceInput.setPlaceholderText(\_translate("Dialog", " Место ДТП"))  
 *self*.updCityInput.setPlaceholderText(\_translate("Dialog", " Город"))  
 *self*.tabWidget.setTabText(*self*.tabWidget.indexOf(*self*.updRow), \_translate("Dialog", "Обновление"))  
 *self*.delReturn.setText(\_translate("Dialog", "Назад"))  
 *self*.deleteButton.setText(\_translate("Dialog", "Удалить"))  
 *self*.tabWidget.setTabText(*self*.tabWidget.indexOf(*self*.delRow), \_translate("Dialog", "Удаление"))  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 *import* sys  
 app = QtWidgets.QApplication(sys.argv)  
 Dialog = QtWidgets.QDialog()  
 ui = Ui\_AccidentDialog()  
 ui.setupUi(Dialog)  
 Dialog.show()  
 sys.exit(app.exec())

Программный код файла main.py, который обрабатывает нажатие всех кнопок и запуск программы:

*class* Accident(QWidget):  
 *def \_\_init\_\_*(*self*):  
 *super*(Accident, *self*).\_\_init\_\_()  
 *self*.ui = Ui\_AccidentDialog()  
 *self*.ui.setupUi(*self*)  
 *self*.returnButtonClicked()  
 *self*.inputData()  
  
 *def* returnButtonClicked(*self*):  
 *self*.ui.selReturn.clicked.connect(*self*.openAdminWindow)  
 *self*.ui.updReturn.clicked.connect(*self*.openAdminWindow)  
 *self*.ui.delReturn.clicked.connect(*self*.openAdminWindow)  
 *self*.ui.addReturn.clicked.connect(*self*.openAdminWindow)  
 *self*.ui.searchButton.clicked.connect(*self*.showDistinct)  
 *self*.ui.deleteButton.clicked.connect(*self*.deleteDistinct)  
 *self*.ui.updChange.clicked.connect(*self*.updateDistinct)  
 *self*.ui.addInsert.clicked.connect(*self*.createDistinct)  
 *self*.ui.addInsert\_2.clicked.connect(*self*.goToParticipant)  
  
 *def* openAdminWindow(*self*):  
 *self*.WindowAdmin = WindowAdmin()  
 *self*.WindowAdmin.show()  
 *self*.close()  
 *def* goToParticipant(*self*):  
 *self*.ParticipantDialog = Participant()  
 *self*.ParticipantDialog.show()  
 *self*.close()  
  
 *def* testFunc(*self*, *full\_name*):  
 cur = conn.cursor()  
 cur.execute("SELECT \* FROM accident WHERE officer\_id = (SELECT officer\_id FROM officer WHERE full\_name = (*%s*))", (*full\_name*,))  
 ans = cur.fetchall()  
 selTableView = *self*.ui.selTableAccident  
 addTableView = *self*.ui.addTableAccident  
 updTableView = *self*.ui.updTableAccident  
 delTableView = *self*.ui.delTableAccident  
  
 model = QStandardItemModel()  
 model.setColumnCount(5)  
  
 model.setHeaderData(0, Qt.Orientation.Horizontal, "Id")  
 model.setHeaderData(1, Qt.Orientation.Horizontal, "Условия")  
 model.setHeaderData(2, Qt.Orientation.Horizontal, "Дата")  
 model.setHeaderData(3, Qt.Orientation.Horizontal, "Место")  
 model.setHeaderData(4, Qt.Orientation.Horizontal, "Город")  
  
 *for* row *in* ans:  
 lst = []  
 *for* elem *in* row:  
 lst.append(elem)  
 date = lst[2]  
 date = *str*(date)  
 lst[0] = *str*(lst[0])  
 lst[2] = date  
 lst[4] = lst[5]  
 lst.pop()  
 model.appendRow([QStandardItem(data) *for* data *in* lst])  
 selTableView.setModel(model)  
 addTableView.setModel(model)  
 updTableView.setModel(model)  
 delTableView.setModel(model)  
  
 *def* inputData(*self*):  
 rows = accidentsCRUD.select\_all(conn)  
 selTableView = *self*.ui.selTableAccident  
 addTableView = *self*.ui.addTableAccident  
 updTableView = *self*.ui.updTableAccident  
 delTableView = *self*.ui.delTableAccident  
  
 model = QStandardItemModel()  
 model.setColumnCount(5)  
  
 model.setHeaderData(0, Qt.Orientation.Horizontal, "Id")  
 model.setHeaderData(1, Qt.Orientation.Horizontal, "Условия")  
 model.setHeaderData(2, Qt.Orientation.Horizontal, "Дата")  
 model.setHeaderData(3, Qt.Orientation.Horizontal, "Место")  
 model.setHeaderData(4, Qt.Orientation.Horizontal, "Город")  
  
 *for* row *in* rows:  
 lst = []  
 *for* elem *in* row:  
 lst.append(elem)  
 date = lst[2]  
 date = *str*(date)  
 lst[0] = *str*(lst[0])  
 lst[2] = date  
 lst[4] = lst[5]  
 lst.pop()  
 model.appendRow([QStandardItem(data) *for* data *in* lst])  
 selTableView.setModel(model)  
 addTableView.setModel(model)  
 updTableView.setModel(model)  
 delTableView.setModel(model)  
  
 *def* showDistinct(*self*):  
 accDate = *self*.ui.searchDate.text()  
 place = date = *self*.ui.searchPlace.text()  
 *if* place == '':  
 cur = conn.cursor()  
 cur.execute("SELECT \* FROM accident WHERE accident\_date = (*%s*)", (accDate,))  
 ans = cur.fetchall()  
 *print*(ans)  
 *elif* accDate == '':  
 cur = conn.cursor()  
 cur.execute("SELECT \* FROM accident WHERE place = (*%s*)", (place,))  
 ans = cur.fetchall()  
 *print*(ans)  
 *else*:  
 ans = accidentsCRUD.select(accDate, place, conn)  
 selTableView = *self*.ui.selTableAccident  
 model = QStandardItemModel()  
 model.setColumnCount(6)  
  
 model.setHeaderData(0, Qt.Orientation.Horizontal, "Id")  
 model.setHeaderData(1, Qt.Orientation.Horizontal, "Условия")  
 model.setHeaderData(2, Qt.Orientation.Horizontal, "Дата")  
 model.setHeaderData(3, Qt.Orientation.Horizontal, "Место")  
 model.setHeaderData(4, Qt.Orientation.Horizontal, "ID сотрудника")  
 model.setHeaderData(5, Qt.Orientation.Horizontal, "Город")  
  
 *for* row *in* ans:  
 lst = []  
 *for* elem *in* row:  
 lst.append(elem)  
 date = lst[2]  
 date = *str*(date)  
 lst[0] = *str*(lst[0])  
 lst[2] = date  
 lst[4] = *str*(lst[4])  
 model.appendRow([QStandardItem(data) *for* data *in* lst])  
 selTableView.setModel(model)  
  
 *def* deleteDistinct(*self*):  
 toDelete = *self*.ui.delNameText.text()  
 accidentsCRUD.delete(toDelete, conn)  
 *self*.inputData()  
  
 *def* updateDistinct(*self*):  
 changeId = *self*.ui.updAccidentIdInput.text()  
 changeDate = *self*.ui.updAccidentDateInput.text()  
 changeOfficerId = *self*.ui.updOfficerIdInput.text()  
 changeConditions = *self*.ui.updConditionsInput.text()  
 changePlace = *self*.ui.updPlaceInput.text()  
 changeCity = *self*.ui.updCityInput.text()  
 accidentsCRUD.update(changeId,changeConditions, changeDate, changePlace, changeOfficerId, changeCity, conn)  
 *self*.inputData()  
  
 *def* createDistinct(*self*):  
 createId = *self*.ui.addAccidentIdInput.text()  
 createDate = *self*.ui.addAccidentDateInput.text()  
 createOfficerId = *self*.ui.addOfficerIdInput.text()  
 createConditions = *self*.ui.addConditionsInput.text()  
 createPlace = *self*.ui.addPlaceInput.text()  
 createCity = *self*.ui.addCityInput.text()  
 accidentsCRUD.insert(createId, createConditions, createDate, createPlace, createOfficerId, createCity, conn)  
 *self*.inputData()

*if* \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 w = Login()  
 w.show()  
 sys.exit(app.exec())

Выводы:

Научились создавать в среде Qt Creator на языке Qt Python оконное приложение для работы с базой данных.