

Universitatea „Gheorghe Asachi” din Iași  
Facultatea de Automatică și Calculatoare

## Proiect - Baze de Date

Profesor coordonator  
Cristian-Nicolae Buțincu

Studenti  
Ursachi Octavian  
Stoica Viorel

Grupa 1308A

# Descrierea Proiectului

Scopul acestei aplicații este de a oferi o interfață grafică intuitivă pentru administrarea unei baze de date. Prin intermediul acestei interfețe, utilizatorul poate efectua operațiuni asupra datelor din baza de date, contribuind astfel la o administrare eficientă a informațiilor stocate.

## Tehnologii Folosite

Front-end:

PyQT5 si QtDesigner pentru dezvoltarea interfeței grafice.

Back-end:

Python

SQLite pentru gestionarea bazei de date.



Constrângerile au fost implementate pentru asigurarea integrității datelor și pentru a preveni erorile în cadrul bazei de date. Acestea includ constrângeri de integritate referențială(PK, FK), cât și constrângeri de tip check, unique, not null.

BD042.VEHICLES	
P * VEHICLE_ID	NUMBER (4)
VEHICLE_TYPE	VARCHAR2 (10 BYTE)
SPEED	NUMBER (3)
DIRECTION	VARCHAR2 (15 BYTE)
F INTERSECTION_ID	NUMBER (4)
VEHICLES_VEHICLE_ID_PK (VEHICLE_ID)	
VEHICLES_INTERSECTION_ID_FK (INTERSECTION_ID)	
VEHICLES_VEHICLE_ID_PK (VEHICLE_ID)	

BD042.INTERSECTIONS	
P * INTERSECTION_ID	NUMBER (4)
INTERSECTION_NAME	VARCHAR2 (30 BYTE)
INTERSECTION_LOCATION	VARCHAR2 (30 BYTE)
TRAFFIC_CONTROL_TYPE	VARCHAR2 (30 BYTE)
MAXIMUM_SPEED_LIMIT	NUMBER (4)
INTERSECTIONS_CCJ_PK (INTERSECTION_ID)	
INTERSECTIONS_CCJ_PK (INTERSECTION_ID)	

BD042.TRAFFIC_LANES	
LANE_ID	NUMBER (4)
LANE_TYPE	VARCHAR2 (50 BYTE)
TRAFFIC_DIRECTION	VARCHAR2 (50 BYTE)
F INTERSECTION_ID	NUMBER (4)
TRAFFIC_LANES_INTERSECTION_ID_FK (INTERSECTION_ID)	

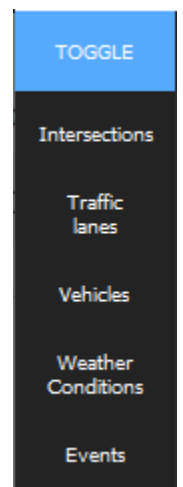
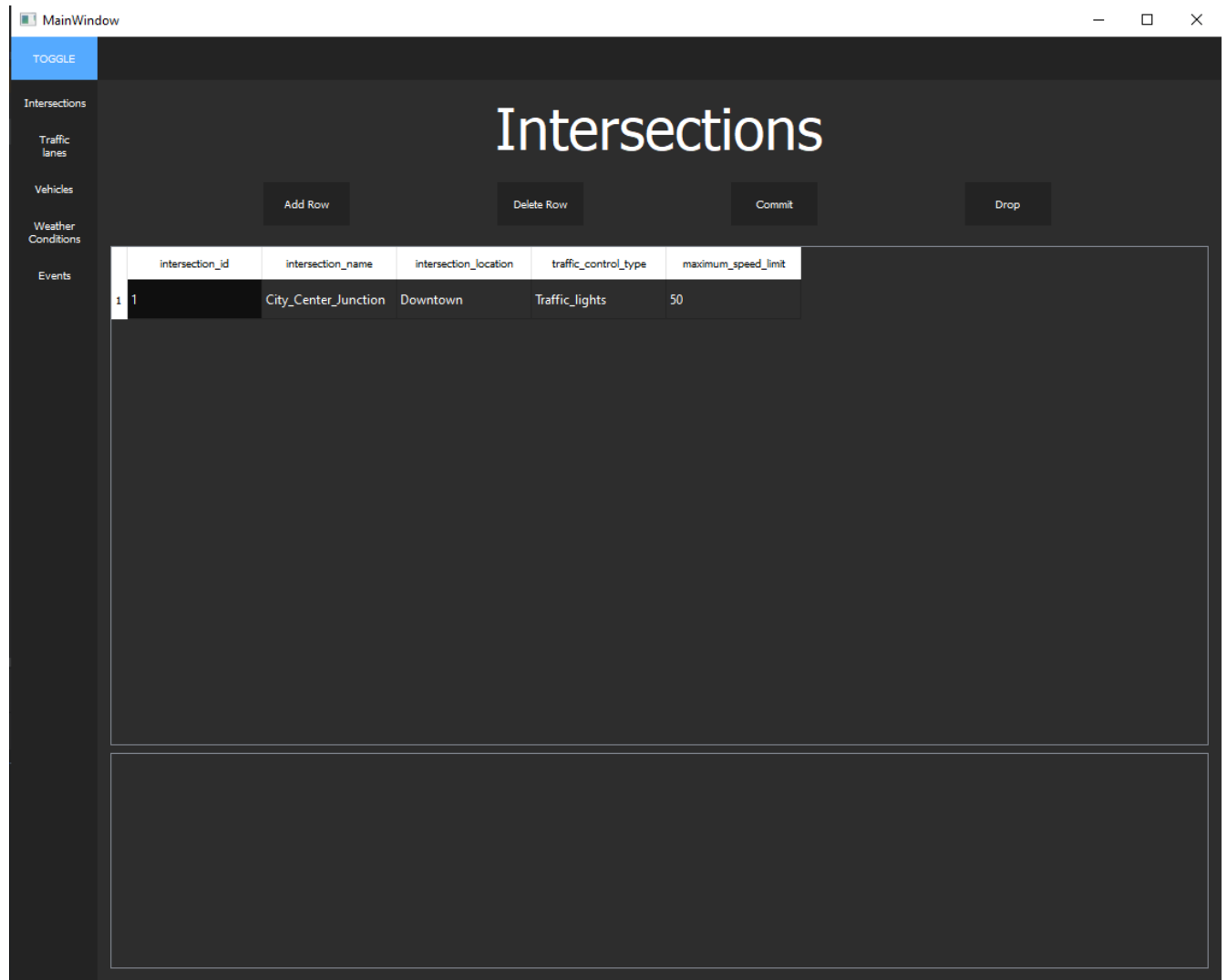
BD042.WEATHER_CONDITIONS	
UF INTERSECTION_ID	NUMBER (4)
TEMPERATURE	NUMBER (2)
PRECIPITATION	VARCHAR2 (50 BYTE)
WIND_SPEED	NUMBER (3)
UQ_WEATHER_INTERSECTION_ID (INTERSECTION_ID)	
WEATHER_INTERSECTION_ID_FK (INTERSECTION_ID)	
UQ_WEATHER_INTERSECTION_ID (INTERSECTION_ID)	

BD042.I_EVENTS	
EVENT_ID	NUMBER (4)
EVENT_TYPE	VARCHAR2 (50 BYTE)
EVENT_DESCRIPTION	VARCHAR2 (255 BYTE)
EVENT_TIME	TIMESTAMP
F INTERSECTION_ID	NUMBER (4)
EVENT_INTERSECTION_ID_FK (INTERSECTION_ID)	

# Modalitatea de Conectare la Baza de Date

Aplicația utilizează modulul cx\_Oracle pentru a stabili și menține conexiunea cu baza de date Oracle. Procesul de conectare implică configurarea parametrilor de conectare, precum utilizatorul, parola și adresa serverului Oracle

# Interfața Aplicației:



Pentru a naviga între tabele folosim bara din stânga.  
În fiecare pagină se află conținutul tabelii respective cat  
și butoane pentru a adăuga/șterge linii și butoane pentru  
a accepta sau renunța la modificările aduse.

Interfața moștenește DBManager ce se ocupă cu conectarea, extragerea și modificarea bazei de date.

```
class Ui(QMainWindow, DBManager):
    # viorelstoica1 +1 *
    def __init__(self):
        super(DBManager, self).__init__()
        super(Ui, self).__init__()

    uic.loadUi(uifile: 'UI/BD.ui', self)
```

Cu modulul uic încarcăm interfața creată în QtDesigner obținând o referință la aceasta.

O dată ce avem referința, prin metoda .findChild() găsim elementele interfeței (eg. butoane).

```
self.buttons = {
    'intersections': self.findChild(QPushButton, 'intersection_b'),
    'traffic_lanes': self.findChild(QPushButton, 'lanes_b'),
    'vehicles': self.findChild(QPushButton, 'vehicles_b'),
    'weather_conditions': self.findChild(QPushButton, 'weather_b'),
    'events': self.findChild(QPushButton, 'events_b'),
    'add_row': self.findChild(QPushButton, 'add_row_b'),
    'commit': self.findChild(QPushButton, 'commit_b'),
    'delete_row': self.findChild(QPushButton, 'delete_row_b'),
    'drop': self.findChild(QPushButton, 'drop_b')
}
```

Conectăm fiecare referință cu un apel de funcție și implementăm logica de back-end.

```
self.buttons['intersections'].clicked.connect(self.intersection_b_clicked)
```

Modificăm baza de date cu ajutorul cursorului moștenit din DBManager.

```
for i in range(len(new_rows)):
    commands.append("INSERT INTO {}({}) VALUES {}".format(*args: table_name, fields, new_rows[i]))
for command in commands:
    try:
        print(command)
        self.cur.execute(command)
        if 1: # poate un pop-up (Atentie, sigur doriti sa modificati?)
            self.cur.execute('commit')
        print('Insert Complete!')
    except Exception as e:
        print(e)
```