Predloga kode za 4.domačo nalogo:

Za prvi del si zamislite neko atomarno transakcijo, recimo “sprejem novega pacienta v zdravstveni dom”:

cursor.execute("START TRANSACTION")

try:

 cursor.execute("SELECT ID\_PACIENT FROM PACIENT WHERE EMSO = 0101990500101")

 existing\_patient\_id = cursor.fetchone()

 if not existing\_patient\_id:

 cursor.execute("INSERT INTO PACIENT(EMSO, IME, PRIIMEK) \

 VALUES (0101990500101, Joze, Novak)")

 cursor.execute("SELECT ID\_PACIENT FROM PACIENT WHERE \

 EMSO = 0101990500101")

 existing\_patient\_id = cursor.fetchone()[0]

 else:

 existing\_patient\_id = existing\_patient\_id[0]

 cursor.execute(f"INSERT INTO ZDRAVLJENJE (ID\_PACIENT, DATUM\_ZAC\_ZDRAVLJENJA) \

 VALUES ({existing\_patient\_id}, 2020-01-01)")

 cursor.execute("COMMIT")

except Exception:

 cursor.execute("ROLLBACK")

conn.commit()

Drugi del pa prikaže težave sočasnega dostopa, kjer uporabljate npr.

cursor.execute("SET GLOBAL TRANSACTION ISOLATION LEVEL READ UNCOMMITTED") ipd.

In lahko funkcijo sleep(n)

Več pa ne smemo izdati, saj bo potem prelahko. Namig: poglejte si transakcije A in B v Jupiter notebook iz vaj.

Še obljuba iz vaj:
pri datoteki socasen.py lahko recimo dodamo izpis procesov, ki so trenutno aktivni

def select(tableName):

 connection = connect(connString)

 try:

 cursor = connection.cursor()

 data = cursor.execute("SELECT \* FROM " + \

 tableName).fetchall()

 for item in data:

 print(item)

 except Exception as exc:

 print("EXCEPTION", exc)

#se procesi

 try:

 cursor = connection.cursor()

 data = cursor.execute("SHOW FULL PROCESSLIST").fetchall()

 for item in data:

 print(item)

 except Exception as exc:

 print("EXCEPTION", exc)

 connection.close()