Screencast 6-2

Using server-side cursors in python

Database – cars

A database cursor is used to iterate over result of a query. Depending on cursor settings, you can move through the records in different directions.

In general, cursor is used in a scenario that is presented below:

- A cursor is declared
- The cursor is opened with a SELECT query
- Reading is done row by row with FETCH command
- Iterate until there is no more row to fetch
- The cursor is closed.

Psycopg2 library can open a server-side cursor if you pass a name to the cursor function. This library uses *cursor.itersize* attribute to change buffer size (number of records transferred to client from the server).

Task 1.

Write a function to find cars which were booked at least N times. Use server-side cursor.

```
def print_all_cars_n2(nbooked:int):
    con = psycopg2.connect(database="cars", user="postgres", password="sql", h
ost="localhost")
    cur_default = con.cursor('c')
    cur_default.execute("""select car.cid, year, make
        from car join res using(cid)
        group by car.cid, year, make
        having count(*)>=%s
        """, [nbooked])

    for record in cur_default:
        print(record[0], record[1], record[2])
    con.close()
```

Task 2.

Rewrite the function with while loop and *cursor.fetchone* function.

```
def print_all_cars_n3(nbooked:int):
    con = psycopg2.connect(database="cars", user="postgres", password="sql", h
ost="localhost")
    cur_default = con.cursor('c')
```

```
    cur_default.execute("""select car.cid, year, make
        from car join res using(cid)
        group by car.cid, year, make
        having count(*)>=%s
        """, [nbooked])

    record = cur_default.fetchone()
    while record:
        print(record[0], record[1], record[2])
        record = cur_default.fetchone()
    con.close()
```