

۴۰۰۳۱۰۸۸

محمد حسین شهبازی

گزارش پروژه دوم دیتابیس

نحوه راه اندازی PostgreSQL :

با تکه کد زیر در کلاس database در برنامه خود به دیتابیس متصل میشویم:

```
def create_connection(self):  
    try:  
        self.connection = psycopg2.connect(  
            database="postgres",  
            user="postgres",  
            password="admin",  
            host="localhost",  
            port="5432")
```

و نمونه جدول پایگاه داده به صورت زیر هستند:

Students:

	id [PK] integer	studentid integer	major character varying (50)	birth_date date	first_name character varying (50)	last_name character varying (50)	balance integer
1	1	123	cs	2010-01-01	ali	rahimi	450
2	3	100	cs	2010-01-01	mohammad	rahmani	450

Foods:

	id [PK] integer	name character varying (50)	date timestamp without time zone	price integer	inventory integer
1	2	salad1	2024-10-01 01:00:00	50	10
2	1	soup1	2024-10-01 01:00:00	50	8

Reservations:

	id [PK] integer	studentid integer	foodid integer
1	1	123	1
2	3	100	1

transactions:

	srcreservationid integer	dstreservationid integer	date timestamp without time zone
1	[null]	1	2024-04-10 09:37:55.434668
2	[null]	2	2024-04-11 03:10:17.956836
3	2	1	2024-04-11 03:11:05.450883

توضیح نحوه پیاده سازی هر یک از عملیات ها:

به طور خلاصه به شرح زیر است:

عملیات درج و حذف دانشجویان:

عملیت درج در تابع `main()`:

```
if action == 'add_student':  
    # example:  
    # add_student 12345 cs 2010-1-1 ali rahimi 500  
    student = students.insert(args[0], args[1], args[2], args[3], args[4], args[5])  
    print('This student is added:')  
    print(student)  
    return
```

این عملیات با گرفتن اطلاعات مانند شماره دانشجویی ، نام و نام خانوادگی و....
از کاربر ستون جدیدی در دیتابیس ایجاد میکند.

عملیات حذف:

```
if action == 'del_student':  
    # example:  
    # del_student 12345  
    student_id = args[0]  
    student = students.get_by_student_id(student_id)  
    students.delete_by_student_id(student_id)  
    print('This student is deleted:')  
    print(student)  
    return
```

مشابه عملیات قبلی اما با دریافت شماره دانشجویی که یکتاست سطر مربوطه را حذف میکند. نحوه عملکرد تابع `student.delete_by_student_id()` در کد مشخص است و اینجا به توضیح مختصر بسنده میکنیم.

عملکرد افزایش موجودی حساب:

```

if action == 'add_balance':
    # example:
    # add_balance 12345 1500
    student_id = args[0]
    balance = int(args[1])
    student = students.add_balance(student_id, balance)
    print('This student balance is updates:')
    print(student)
    return

```

اینجا با دریافت شماره دانشجویی و مقداری که کاربر قصد اضافه کردن بر موجودی را دارد یک query ارسال شده و به موجودی فعلی اضافه میشود.

عملکرد: افزودن و حذف کردن غذا

افزودن:

```

if action == 'add_food':
    # example:
    # add_food soup 2024-10-1:1:00 100 10
    food = foods.insert(args[0], args[1], args[2], args[3])
    print('This food is added:')
    print(food)
    return

```

در اینجا با دریافت اطلاعات از کاربر میرویم و یک سطر جدید به دیتابیس

اضافه میکنیم. اطلاعاتی نظیر: نام غذا ، شناسه غذا و ...

حذف کردن:

```
if action == 'del_food':  
    # example:  
    # del_food 1  
    food_id = args[0]  
    food = foods.get(food_id)  
    foods.delete(food_id)  
    print('This food is deleted:')  
    print(food)  
    return
```

مانند حالت قبلیست اما با دریافت شناسه غذا که یکتاست اقدام به حذف غذا میکند.

عملکرد رزرو غذا:

```
if action == 'add_res':  
    # example:  
    # add_res 12345 1  
    student_id = args[0]  
    food_id = args[1]  
    reservations.insert_reservation(student_id, food_id)  
    student = students.get_by_student_id(student_id)  
    food = foods.get(food_id)  
    print('The food is reserved for the student.')  
    print(student)  
    print(food)  
    return
```

با دریافت شناسه غذا و شناسه دانشجو یک ستون به جدول reservations اضافه میکند و از موجودی دانشجو و تعداد غذای رزرو شده کم میکند.

عملکرد تغییر رزرو:

```
if action == 'update_res':  
    # example:  
    # update_res 12345 1 2  
    student_id = args[0]  
    food_id = args[1]  
    new_food_id = args[2]  
    reservations.update_reservation(student_id, food_id, new_food_id)  
    student = students.get_by_student_id(student_id)  
    food = foods.get(food_id)  
    new_food = foods.get(new_food_id)  
    print('The reservation is updated for the student.')  
    print(student)  
    print(food)  
    print(new_food)  
    return
```

این عملکرد با دریافت یه سری آرگومان ها از کامند لاین argparse و مشخصا با دریافت شناسه جدید غذای سفارش داده شده می آید و شناسه مبدار را در جدول

رزرو ها را برابر شناسه غذای جدید سفارش داده شده قرار داده و شناسه غذای قبلی را برابر شناسه مقصد قرار میدهد.

اسکرین شات از برنامه و دستور های پیاده سازی شده:

اضافه کردن دانشجو:

```
PS C:\Users\Shahin\Desktop\db_project2> python main.py add_student 100 cs 2010-1-1 mohammad rahmani 300
```

```
This student is added:
```

```
(3, 100, 'cs', '2010-01-01', 'mohammad', 'rahmani', 300)
```

اضافه کردن موجودی:

```
connection to PostgreSQL is closed.
```

```
PS C:\Users\Shahin\Desktop\db_project2> python main.py add_balance 100 200
```

```
This student balance is updates:
```

```
(3, 100, 'cs', '2010-01-01', 'mohammad', 'rahmani', 500)
```

اضافه کردن غذا:

```
Connection to PostgreSQL is closed!
PS C:\Users\Shahin\Desktop\db_project2> python main.py add_food salad1 2024-10-1:1:0 50 10
Connected to PostgreSQL database!
This food is added:
(2, 'salad1', '2024-10-01 01:00:00', 50, 10)
```

رزرو کردن:

```
Connection to PostgreSQL is closed!
PS C:\Users\Shahin\Desktop\db_project2> python main.py add_res 100 2

The food is reserved for the student.
(3, 100, 'cs', '2010-01-01', 'mohammad', 'rahmani', 450)
(2, 'salad1', '2024-10-01 01:00:00', 50, 9)
```

تغییر رزرو:

```
PS C:\Users\Shahin\Desktop\db_project2> python main.py update_res 100 2 1
Connected to PostgreSQL database!

The reservation is updated for the student.
(3, 100, 'cs', '2010-01-01', 'mohammad', 'rahmani', 450)
(2, 'salad1', '2024-10-01 01:00:00', 50, 10)
(1, 'soup1', '2024-10-01 01:00:00', 50, 8)
```

حذف کردن دانشجو:

```
PS C:\Users\Shahin\Desktop\db_project2> python main.py del_student 123  
Connected to PostgreSQL database!
```

This student is deleted:

```
(1, 123, 'cs', '2010-01-01', 'ali', 'rahimi', 450)
```

حذف کردن غذا:

```
PS C:\Users\Shahin\Desktop\db_project2> python main.py del_food 1  
Connected to PostgreSQL database!
```

This food is deleted:

```
(1, 'soup1', '2024-10-01 01:00:00', 50, 8)
```

پایان