

# Лекция 6-3. Importing and Exporting Data case

# Python PostgreSQL Using Psycopg2

## Цель:

- Использовать модуль **Psycopg2**.

# Использование Python с базой данных

## Список доступных модулей Python для работы с сервером базы данных PostgreSQL

- Psycopg2
- pg8000
- py-postgresql
- PyGreSQL
- ocpgdb
- bpgsql
- SQLAlchemy

## Psycopg2

- Он используется в большинстве фреймворков **Python** и **Postgres**.
- Он также **активно поддерживается** и поддерживает основную версию Python, то есть Python 3 и Python 2.
- Он является **потокобезопасным** и предназначен для многопоточных приложений. Обратите внимание, что потоки могут совместно использовать соединения.

# УСТАНОВКА Psycopg2

```
conda install -c anaconda psycopg2
```

```
conda update -n base -c defaults conda
```

**Перезагрузить KERNEL**

# Как подключиться к PostgreSQL в Python

## 1. Установить модуль Psycopg2

Установить и импортируйте модуль psycopg2. Импортируйте с помощью

**`import psycopg2`**

оператора, чтобы использовать методы этого модуля для связи с базой данных PostgreSQL.

# Как подключиться к PostgreSQL в Python

## 2. Использовать метод подключения ()

Используйте

**`psycopg2.connect()`**

метод с необходимыми аргументами для подключения MySQL. Он вернет

**`Connection`**

объект, если соединение установлено успешно.



# Как подключиться к PostgreSQL в Python

## 3. Используйте метод курсора().

Создайте объект курсора, используя объект подключения, возвращаемый методом

**connect()**

, для выполнения запросов PostgreSQL из Python.

# Как подключиться к PostgreSQL в Python



## 4. Используйте метод выполнения ()

Методы

**execute()**

запускают SQL-запрос и возвращают результат.

# Как подключиться к PostgreSQL в Python

## 5. Извлечь результат с помощью fetchall()

Используйте

```
cursor.fetchall()
```

или

```
fetchone()
```

или

```
fetchmany()
```

для чтения результата запроса.

# Как подключиться к PostgreSQL в Python

## 6. Закрывать объекты курсора и соединения

ИСПОЛЬЗОВАНИЕ

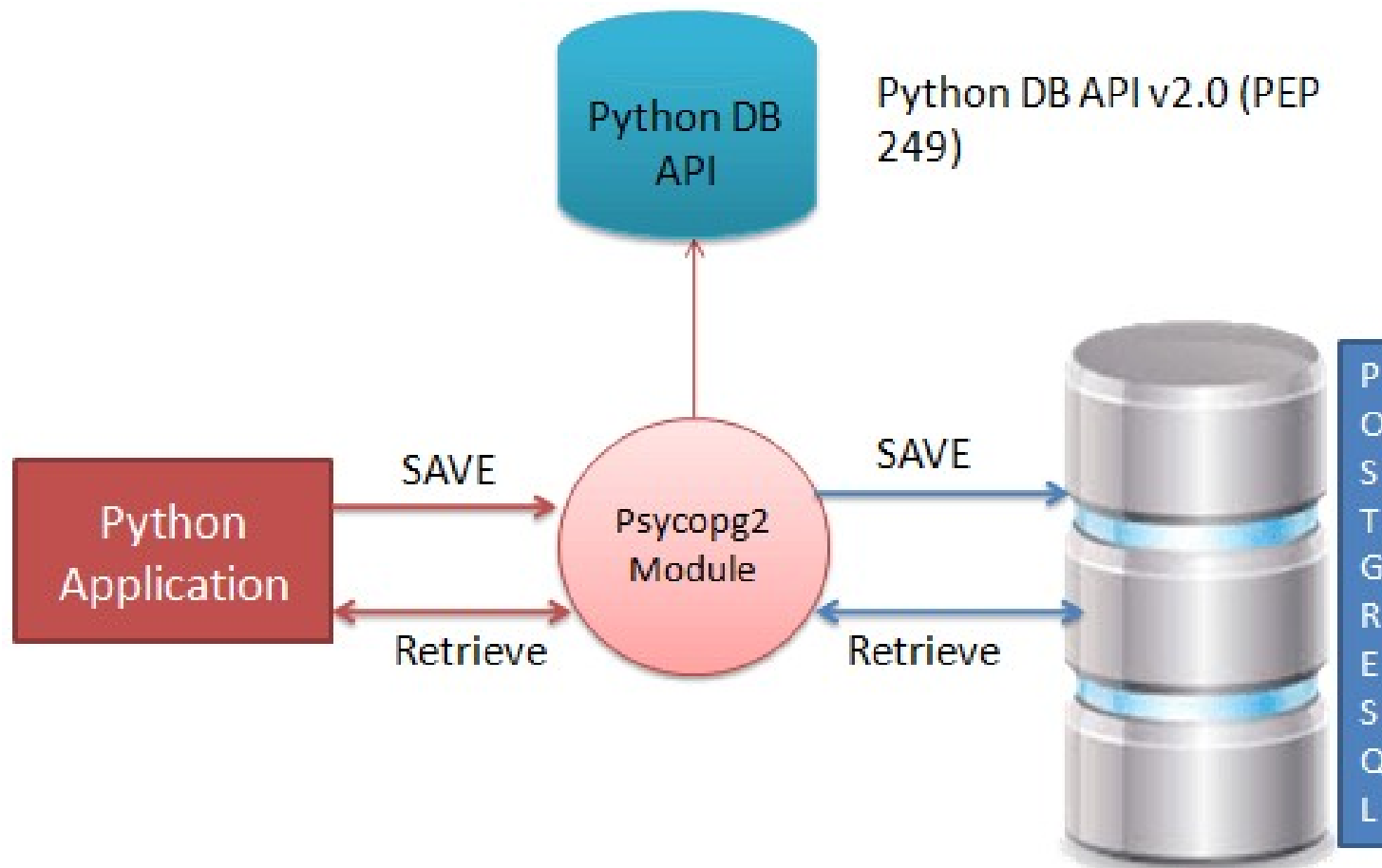
```
cursor.close()
```

и

```
connection.close()
```

метод для закрытия соединений PostgreSQL после завершения вашей работы

# Как подключиться к PostgreSQL в Python



# Как подключиться к PostgreSQL в Python

```
import psycopg2
from psycopg2 import Error

try:
    # Connect to an existing database
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    # Create a cursor to perform database operations
    cursor = connection.cursor()
    # Print PostgreSQL details
    print("PostgreSQL server information")
    print(connection.get_dsn_parameters(), "\n")
    # Executing a SQL query
    cursor.execute("SELECT version();")
    # Fetch result
    record = cursor.fetchone()
    print("You are connected to - ", record, "\n")

except (Exception, Error) as error:
    print("Error while connecting to PostgreSQL", error)
finally:
    if (connection):
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")
```

# Создание таблицы PostgreSQL из Python

```
import psycopg2
from psycopg2 import Error

try:
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    cursor = connection.cursor()
    # SQL query to create a new table
    create_table_query = '''CREATE TABLE mobile
        (ID INT PRIMARY KEY     NOT NULL,
         MODEL           TEXT     NOT NULL,
         PRICE           REAL); '''
    # Execute a command: this creates a new table
    cursor.execute(create_table_query)
    connection.commit()
    print("Table created successfully in PostgreSQL ")

except (Exception, Error) as error:
    print("Error while connecting to PostgreSQL", error)
finally:
    if connection:
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")
```

# PostgreSQL CRUD operations from Python

```
import psycopg2

try:
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    cursor = connection.cursor()
    # Executing a SQL query to insert data into table
    insert_query = """ INSERT INTO mobile (ID, MODEL, PRICE) VALUES (1, 'Iphone12', 1100)"""
    cursor.execute(insert_query)
    connection.commit()
    print("1 Record inserted successfully")
    # Fetch result
    cursor.execute("SELECT * from mobile")
    record = cursor.fetchall()
    print("Result ", record)

    # Executing a SQL query to update table
    update_query = """Update mobile set price = 1500 where id = 1"""
    cursor.execute(update_query)
    connection.commit()
    count = cursor.rowcount
    print(count, "Record updated successfully ")
    # Fetch result
    cursor.execute("SELECT * from mobile")
    print("Result ", cursor.fetchall())

    # Executing a SQL query to delete table
    delete_query = """Delete from mobile where id = 1"""
    cursor.execute(delete_query)
    connection.commit()
    count = cursor.rowcount
    print(count, "Record deleted successfully ")
    # Fetch result
    cursor.execute("SELECT * from mobile")
    print("Result ", cursor.fetchall())

except (Exception, psycopg2.Error) as error:
    print("Error while connecting to PostgreSQL", error)
finally:
    if connection:
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")
```



## Practice 07. Упражнение/мини-проект: Больничная информационная система



1. В этом упражнении используем информационную систему «Больничная информационная система».
2. Требуется создать две таблицы: **Hospital** и **Doctor** на сервере баз данных перед началом выполнения задач.

## Practice 07. Упражнение/мини-проект: Больничная информационная система

### SQL-запросы для подготовки данных

#### Create Database

```
CREATE database python_db;
```

#### Create Hospital Table

```
CREATE DATABASE python_db  
WITH  
OWNER = admin  
ENCODING = 'UTF8'  
LC_COLLATE = 'en_US.UTF-8'  
LC_CTYPE = 'en_US.UTF-8'  
TABLESPACE = pg_default  
CONNECTION LIMIT = -1  
IS_TEMPLATE = False;
```

```
CREATE TABLE Hospital (  
Hospital_Id serial NOT NULL PRIMARY KEY,  
Hospital_Name VARCHAR (100) NOT NULL,  
Bed_Count serial  
);  
INSERT INTO Hospital (Hospital_Id, Hospital_Name, Bed_Count)  
VALUES  
( '1', 'Mayo Clinic', 200),  
( '2', 'Cleveland Clinic', 400),  
( '3', 'Johns Hopkins', 1000),  
( '4', 'UCLA Medical Center', 1500);
```

# Practice 07. Упражнение/мини-проект: Больничная информационная система

## SQL-запросы для подготовки данных

### Create Doctor Table

```
CREATE TABLE Doctor (  
  Doctor_Id serial NOT NULL PRIMARY KEY,  
  Doctor_Name VARCHAR (100) NOT NULL,  
  Hospital_Id serial NOT NULL,  
  Joining_Date DATE NOT NULL,  
  Speciality VARCHAR (100) NOT NULL,  
  Salary INTEGER NOT NULL,  
  Experience SMALLINT  
);  
INSERT INTO Doctor (Doctor_Id, Doctor_Name, Hospital_Id, Joining_Date, Speciality, Salary, Experience)  
VALUES  
( '101', 'David', '1', '2005-2-10', 'Pediatric', '40000', NULL),  
( '102', 'Michael', '1', '2018-07-23', 'Oncologist', '20000', NULL),  
( '103', 'Susan', '2', '2016-05-19', 'Garnacologist', '25000', NULL),  
( '104', 'Robert', '2', '2017-12-28', 'Pediatric ', '28000', NULL),  
( '105', 'Linda', '3', '2004-06-04', 'Garnacologist', '42000', NULL),  
( '106', 'William', '3', '2012-09-11', 'Dermatologist', '30000', NULL),  
( '107', 'Richard', '4', '2014-08-21', 'Garnacologist', '32000', NULL),  
( '108', 'Karen', '4', '2011-10-17', 'Radiologist', '30000', NULL);
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система

### SQL-запросы для подготовки данных

Hospital_Id	Hospital_Name	Bed Count
1	Mayo Clinic	200
2	Cleveland Clinic	400
3	Johns Hopkins	1000
4	UCLA Medical Center	1500

hospital table

Doctor_Id	Doctor_Name	Hospital_Id	Joining_Date	Speciality	Salary	Experience
101	David	1	2005-02-10	Pediatric	40000	NULL
102	Michael	1	2018-07-23	Oncologist	20000	NULL
103	Susan	2	2016-05-19	Garnacologist	25000	NULL
104	Robert	2	2017-12-28	Pediatric	28000	NULL
105	Linda	3	2004-06-04	Garnacologist	42000	NULL
106	William	3	2012-09-11	Dermatologist	30000	NULL
107	Richard	4	2014-08-21	Garnacologist	32000	NULL
108	Karen	4	2011-10-17	Radiologist	30000	NULL

Doctor table

# Practice 07. Упражнение/мини-проект: Больничная информационная система

## Упражнение 1. Подключитесь к серверу базы данных и распечатайте его версию

```
import psycopg2

def get_connection():
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    return connection

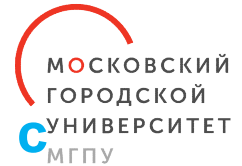
def close_connection(connection):
    if connection:
        connection.close()

def read_database_version():
    try:
        connection = get_connection()
        cursor = connection.cursor()
        cursor.execute("SELECT version();")
        db_version = cursor.fetchone()
        print("You are connected to PostgreSQL version: ", db_version)
        close_connection(connection)
    except (Exception, psycopg2.Error) as error:
        print("Error while getting data", error)

print("Question 1: Print Database version")
read_database_version()
```

Question 1: Print Database version  
You are connected to PostgreSQL version: ('PostgreSQL 14.7 (Ubuntu 14.7-0ubuntu0.22.04.1) on x86\_64-pc-linux-gnu, compiled by gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0, 64-bit',)

## Practice 07. Упражнение/мини-проект: Больничная информационная система

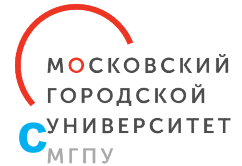


### Упражнение 2. Получить информацию о больнице и враче с использованием идентификатора больницы и идентификатора врача

Подключитесь к `python_db` и используйте `cursor.execute()` для выполнения параметризованного запроса.  
Затем используйте `cursor.fetchall()` для получения записи.  
Затем повторите `record/resultSet`, чтобы распечатать все значения столбца.

```
def get_hospital_detail(hospital_id):  
    #Read data from Hospital table  
  
def get_doctor_detail(doctor_id):  
    # Read data from Doctor table  
  
get_hospital_details(2)  
get_doctor_details(105)
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система



### Упражнение 2. Получить информацию о больнице и враче с использованием идентификатора больницы и идентификатора врача

Question 2: **Read** given hospital **and** doctor details

Printing Hospital record

Hospital Id: 2

Hospital **Name**: Cleveland Clinic

Bed **Count**: 400

Printing Doctor record

Doctor Id: 105

Doctor **Name**: Linda

Hospital Id: 3

Joining **Date**: 2004-06-04

Specialty: Gynecologist

Salary: 42000

Experience: **None**

# Practice 07. Упражнение/мини-проект: Больничная информационная система

## Упражнение 2. Получить информацию о больнице и врачах с использованием идентификатора больницы и идентификатора врача



```
import psycopg2

def get_connection():
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    return connection

def close_connection(connection):
    if connection:
        connection.close()
        print("Postgres connection is closed")

def get_hospital_detail(hospital_id):
    try:
        connection = get_connection()
        cursor = connection.cursor()
        select_query = """select * from Hospital where Hospital_Id = %s"""
        cursor.execute(select_query, (hospital_id,))
        records = cursor.fetchall()
        print("Printing Hospital record")
        for row in records:
            print("Hospital Id:", row[0], )
            print("Hospital Name:", row[1])
            print("Bed Count:", row[2])
        close_connection(connection)
    except (Exception, psycopg2.Error) as error:
        print("Error while getting data", error)

def get_doctor_detail(doctor_id):
    try:
        connection = get_connection()
        cursor = connection.cursor()
        select_query = """select * from Doctor where Doctor_Id = %s"""
        cursor.execute(select_query, (doctor_id,))
        records = cursor.fetchall()
        print("Printing Doctor record")
        for row in records:
            print("Doctor Id:", row[0])
            print("Doctor Name:", row[1])
            print("Hospital Id:", row[2])
            print("Joining Date:", row[3])
            print("Specialty:", row[4])
            print("Salary:", row[5])
            print("Experience:", row[6])
        close_connection(connection)
    except (Exception, psycopg2.Error) as error:
        print("Error while getting data", error)

print("Question 2: Read given hospital and doctor details \n")
get_hospital_detail(2)
print("\n")
get_doctor_detail(105)
```



## **Practice 07.** Упражнение/мини-проект: Больничная информационная система



**Упражнение 3. Получить список врачей по заданной специальности и зарплате**

## Practice 07. Упражнение/мини-проект: Больничная информационная система



### Упражнение 3. Получить список врачей по заданной специальности и зарплате

Определите параметризованный запрос на выборку для получения данных из таблицы в соответствии с заданной специальностью и зарплатой.

Затем используйте `cursor.execute()` для выполнения запроса.

Затем получите все записи с помощью `cursor.fetchall()`. Повторите эти записи и распечатайте каждую строку.

```
def get_specialist_doctors_list(speciality, salary):  
    #Fetch doctors details as per Speciality and Salary  
  
    get_specialist_doctors_list("Pediatric", 20000)
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система



### Упражнение 3. Получить список врачей по заданной специальности и зарплате

```
Printing doctors whose specialty is Pediatric
and salary greater than 20000
Doctor Id: 101
Doctor Name: David
Hospital Id: 1
Joining Date: 2005-02-10
Specialty: Pediatric
Salary: 40000
Experience: None
Postgres connection is closed
```

# Practice 07. Упражнение/мини-проект: Больничная информационная система

## Упражнение 3. Получить список врачей по заданной специальности и зарплате

```
import psycopg2

def get_connection():
    connection = psycopg2.connect(user="postgres",
                                   password="p",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="db")

    return connection

def close_connection(connection):
    if connection:
        connection.close()
        print("Postgres connection is closed")

def get_specialist_doctors_list(speciality, salary):
    try:
        connection = get_connection()
        cursor = connection.cursor()
        sql_select_query = """select * from Doctor where Speciality=%s and Salary > %s"""
        cursor.execute(sql_select_query, (speciality, salary))
        records = cursor.fetchall()
        print("Printing doctors whose speciality is", speciality, "and salary greater than", salary, "\n")
        for row in records:
            print("Doctor Id: ", row[0])
            print("Doctor Name:", row[1])
            print("Hospital Id:", row[2])
            print("Joining Date:", row[3])
            print("Specialty:", row[4])
            print("Salary:", row[5])
            print("Experience:", row[6], "\n")
        close_connection(connection)
    except (Exception, psycopg2.Error) as error:
        print("Error while getting data", error)

print("Question 3: Get Doctors as per given Speciality\n")
get_specialist_doctors_list("Pediatric", 20000)
```

## **Practice 07.** Упражнение/мини-проект: Больничная информационная система



**Упражнение 4. Получить список врачей из данной больницы**

## Practice 07. Упражнение/мини-проект: Больничная информационная система

### Упражнение 4. Получить список врачей из данной больницы

Реализуйте функциональность для получения всех врачей в соответствии с заданным идентификатором больницы.

**Вы должны отобразить имя врача в больнице.**

```
def get_doctors(hospital_id):  
    #Fetch All doctors within given Hospital  
    get_doctors(2)
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система

### Упражнение 4. Получить список врачей из данной больницы

Ход решения

- Определите параметризованный запрос выбора, чтобы получить название больницы в соответствии с заданным идентификатором больницы.
- Затем используйте `cursor.execute()` для выполнения этого запроса и сохраните название больницы в переменной.
- Определите параметризованный запрос выбора, чтобы получить всех врачей из таблицы докторов в соответствии с заданным идентификатором больницы.
- Затем используйте `cursor.execute()` для выполнения запроса.
- Затем получите все записи, используя `cursor.fetchall()`
- Повторите эти записи и распечатайте каждый столбец. Кроме того, отображать название больницы, которое мы получили в первом запросе в записи каждого врача.

## Practice 07. Упражнение/мини-проект: Больничная информационная система

### Упражнение 4. Получить список врачей из данной больницы

#### Ход решения

- Определите параметризованный запрос выбора, чтобы получить название больницы в соответствии с заданным идентификатором больницы.
- Затем используйте `cursor.execute()` для выполнения этого запроса и сохраните название больницы в переменной.
- Определите параметризованный запрос выбора, чтобы получить всех врачей из таблицы докторов в соответствии с заданным идентификатором больницы.
- Затем используйте `cursor.execute()` для выполнения запроса.
- Затем получите все записи, используя `cursor.fetchall()`
- Повторите эти записи и распечатайте каждый столбец. Кроме того, отображать название больницы, которое мы получили в первом запросе в записи каждого врача.



# Practice 07. Упражнение/мини-проект: Больничная информационная система

## Упражнение 4. Получить список врачей из данной больницы

```
import psycopg2

def get_connection():
    connection = psycopg2.connect(user="postgres",
                                   password="pynative@#29",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="python_db")

    return connection

def close_connection(connection):
    if connection:
        connection.close()

def get_hospital_name(hospital_id):
    # Fetch Hospital Name using Hospital id
    try:
        connection = get_connection()
        cursor = connection.cursor()
        select_query = """select * from Hospital where Hospital_Id = %s"""
        cursor.execute(select_query, (hospital_id,))
        record = cursor.fetchone()
        close_connection(connection)
        return record[1]
    except (Exception, psycopg2.Error) as error:
        print("Error while getting data from PostgreSQL", error)

def get_doctors(hospital_id):
    # Fetch Hospital Name using Hospital id
    try:
        hospital_name = get_hospital_name(hospital_id)
        connection = get_connection()
        cursor = connection.cursor()
        sql_select_query = """select * from Doctor where Hospital_Id = %s"""
        cursor.execute(sql_select_query, (hospital_id,))
        records = cursor.fetchall()

        print("Printing Doctors of ", hospital_name, "Hospital")
        for row in records:
            print("Doctor Id:", row[0])
            print("Doctor Name:", row[1])
            print("Hospital Id:", row[2])
            print("Hospital Name:", hospital_name)
            print("Joining Date:", row[3])
            print("Specialty:", row[4])
            print("Salary:", row[5])
            print("Experience:", row[6], "\n")
        close_connection(connection)
    except (Exception, psycopg2.Error) as error:
        print("Error while getting doctor's data", error)

print("Question 4: Get List of doctors of a given Hospital Id\n")
get_doctors(2)
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система



### Упражнение 5. Обновить стаж врача в годах

Значение столбца опыта для каждого врача равно `null`. Реализовать функционал для обновления стажа данного врача в годах.

```
def def update_doctor_experience(doctor_id):  
# Update Doctor Experience in Years  
update_doctor_experience(101)
```

## Practice 07. Упражнение/мини-проект: Больничная информационная система



### Упражнение 5. Обновить стаж врача в годах

Ход решения

- В таблице врачей указана дата присоединения для каждого врача.
- Получить дату присоединения данного врача.
- Чтобы получить разницу в год, мы можем вычислить разницу между сегодняшней датой и датой присоединения в годах.
- Подсчитав разницу за год, вы можете выполнить запрос на обновление таблицы, чтобы обновить опыт данного врача.

# Practice 07. Упражнение/мини-проект: Больничная информационная система



## Упражнение 5. Обновить стаж врача в годах

### Ожидаемый результат

#### До :

Печать карты врача  
Идентификатор доктора: 101  
Имя доктора: Дэвид  
Идентификатор больницы: 1  
Дата вступления: 10 февраля 2005 г. Специальность:  
Педиатр  
Зарплата: 40000  
Опыт: Нет

### Ожидаемый результат

#### После :

Печать карты врача  
Идентификатор доктора: 101  
Имя доктора: Дэвид  
Идентификатор больницы: 1  
Дата вступления: 10 февраля 2005 г.  
Специальность: Педиатр  
Зарплата: 40000  
Опыт: 15

# Practice 07. Упражнение/мини-проект: Больничная информационная система

## Упражнение 5. Обновить стаж врача в годах

```
import psycopg2
import datetime
from dateutil.relativedelta import relativedelta

def get_connection():
    connection = psycopg2.connect(user="postgres",
                                   password="pynative@#29",
                                   host="127.0.0.1",
                                   port="5432",
                                   database="python_db")

    return connection

def close_connection(connection):
    if connection:
        connection.close()

def update_doctor_experience(doctor_id):
    # Update Doctor Experience in Years
    try:
        # Get joining date
        connection = get_connection()
        cursor = connection.cursor()
        select_query = """select Joining_Date from Doctor where Doctor_Id = %s"""
        cursor.execute(select_query, (doctor_id,))
        joining_date = cursor.fetchone()

        # calculate Experience in years
        joining_date_1 = datetime.datetime.strptime(''.join(map(str, joining_date)), '%Y-%m-%d')
        today_date = datetime.datetime.now()
        experience = relativedelta(today_date, joining_date_1).years

        # Update doctor's Experience now
        connection = get_connection()
        cursor = connection.cursor()
        sql_select_query = """update Doctor set Experience = %s where Doctor_Id = %s"""
        cursor.execute(sql_select_query, (experience, doctor_id))
        connection.commit()
        print("Doctor Id:", doctor_id, " Experience updated to ", experience, " years")
        close_connection(connection)

    except (Exception, psycopg2.Error) as error:
        print("Error while getting doctor's data", error)

print("Question 5: Calculate and Update experience of all doctors  \n")
update_doctor_experience(101)
```

СПАСИБО ЗА ВНИМАНИЕ