

Practice 05-4 Импорт и экспорт данных.
Python Select из таблицы PostgreSQL с
использованием Psycopg2

Python PostgreSQL Using Psycopg2



Цель:

- выполнять запрос **PostgreSQL SELECT** из **Python** с использованием модуля **Psycopg2**.

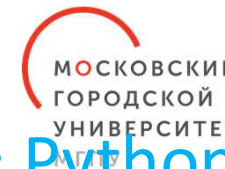
Practice 05-4. Использование «mobile» таблицы, созданной в базе данных PostgreSQL.

ЗАДАНИЕ

1. Создать таблицу в PostgreSQL, согласно рисунка

id	name	price
1	Apple iPhone XS	1000
2	Samsung Galaxy S9	900
3	Google Pixel	700
4	LG	800

Practice 05-4. Использование «mobile» таблицы, созданной в базе данных PostgreSQL.



Шаги для выполнения запроса PostgreSQL SELECT из Python

1. Подключиться к PostgreSQL из Python с помощью модуля Psycopg2.
2. Определить запрос PostgreSQL SELECT. Подготовьте запрос SQL SELECT для выборки строк из таблицы. Можно выбрать все или ограниченные строки в зависимости от ваших потребностей.
3. Получить объект курсора из подключения. Затем используйте метод `connection.cursor()` для создания объекта курсора Psycopg2. Этот метод создает новый объект `psycopg2.extensions.cursor`.
4. Выполнение запроса SELECT с помощью метода `execute()`. Выполните запрос на выборку, используя метод `cursor.execute()`.
5. Извлечь все строки из результата. После успешного выполнения операции Select используйте метод `fetchall()` объекта курсора, чтобы получить все строки из результата запроса. он возвращает список строк.
6. Итерировать каждую строку. Повторите список строк, используя цикл `for`, и получите доступ к каждой строке отдельно (доступ к данным столбца каждой строки, используя имя столбца или номер индекса).
7. Закройте объект курсора и объект подключения к базе данных используйте методы `cursor.close()` и `connection.close()`, чтобы закрыть открытые соединения после завершения работы.

Practice 05-4. Использование «mobile» таблицы, созданной в базе данных PostgreSQL.

ОТВЕТ

1. Создать таблицу в PostgreSQL, согласно рисунка

```
import psycopg2
from psycopg2 import Error

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

    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")
```

Practice 05-4. Использование «mobile» таблицы, созданной в базе данных PostgreSQL.

ОТВЕТ

1. Добавить данные таблицу в PostgreSQL, согласно рисунка

```
import psycopg2

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

    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)

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 05-4. Использование «mobile» таблицы, созданной в базе данных PostgreSQL.

ОТВЕТ

1. Извлечь строки из таблицы PostgreSQL с помощью **fetchall()**

```
import psycopg2

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

    cursor = connection.cursor()
    postgresQL_select_Query = "select * from mobile"

    cursor.execute(postgresQL_select_Query)
    print("Selecting rows from mobile table using cursor.fetchall()")
    mobile_records = cursor.fetchall()

    print("Print each row and it's columns values")
    for row in mobile_records:
        print("Id = ", row[0], )
        print("Model = ", row[1])
        print("Price = ", row[2], "\n")

except (Exception, psycopg2.Error) as error:
    print("Error while fetching data from PostgreSQL", error)

finally:
    # closing database connection.
    if connection:
        cursor.close()
        connection.close()
        print("PostgreSQL connection is closed")
```

Practice 05-5. Работа с интегрированными сервисами.



ЗАДАНИЕ

1. Создать ERD-диаграмму согласно варианта задания.
2. Сгенерировать более 10000 записей для таблицы в PostgreSQL.
3. Внести данные в таблицу с помощью Psycorg2.
4. Определить ключевые поля и агрегированные данные.
5. Выгрузить данные из СУБД Postgresql с помощью Psycorg2.
6. Провести анализ данных в Yandex.DATALENS.

Practice 05-5. Работа с интегрированными сервисами.

ОТВЕТ

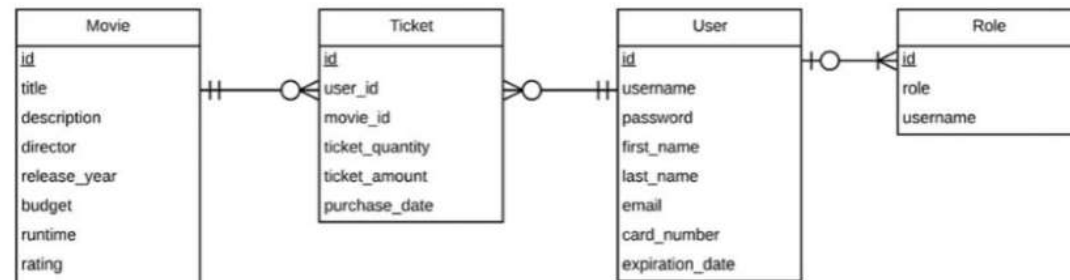
1. Для создания ERD-диаграммы использовать <https://online.visual-paradigm.com/diagrams/templates/entity-relationship-diagram/notations-for-traditional-erd/> или <https://draw.io/>
1. Для генерации воспользоваться SQL Test Data Generator Tool <https://www.coderstool.com/sql-test-data-generator>

Practice 05-5. Работа с интегрированными сервисами

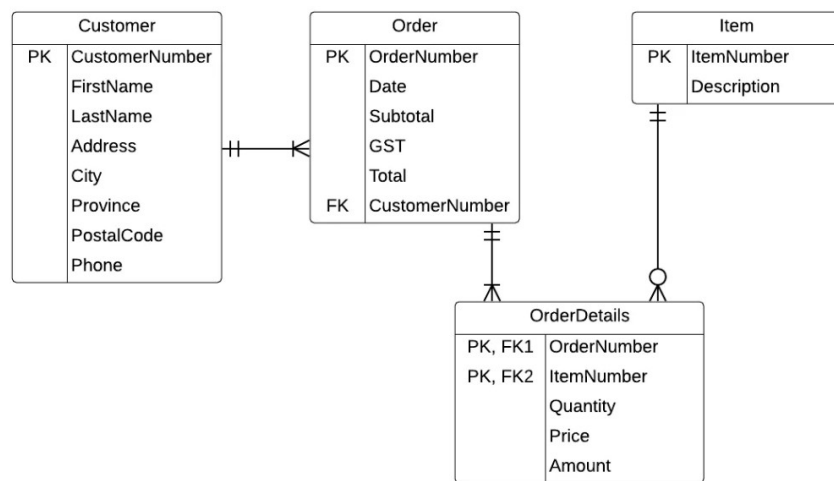
1.



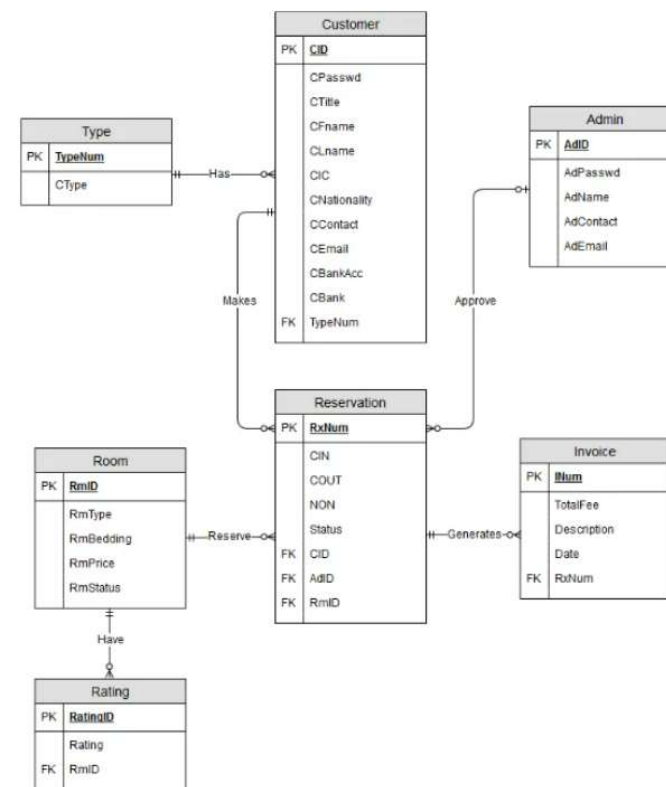
2.



3.

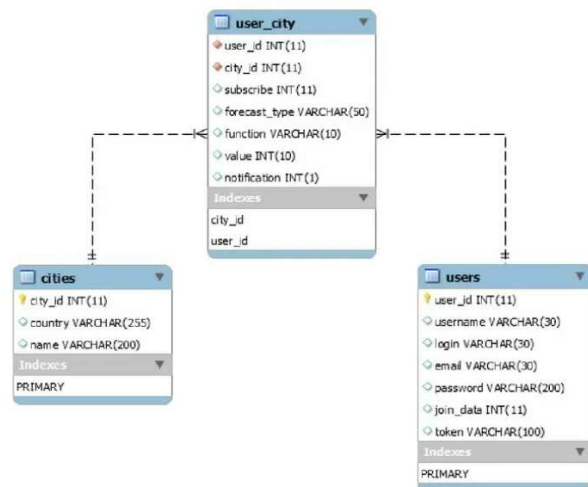


4.

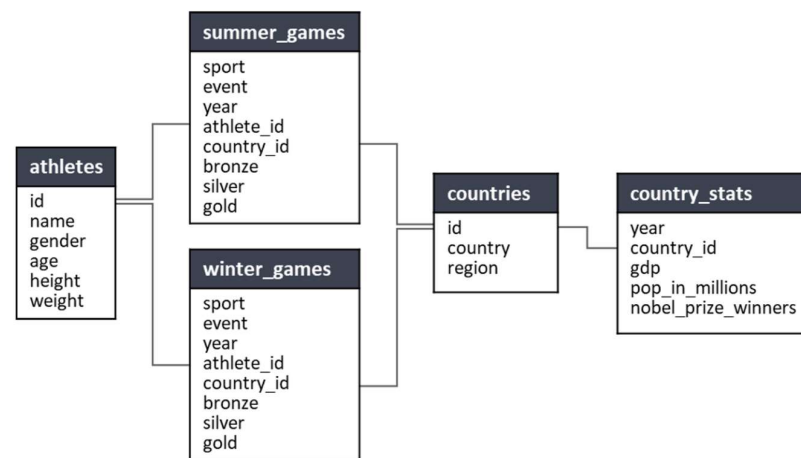


СКИЙ
КОЙ
ИТЕТ

5. MI

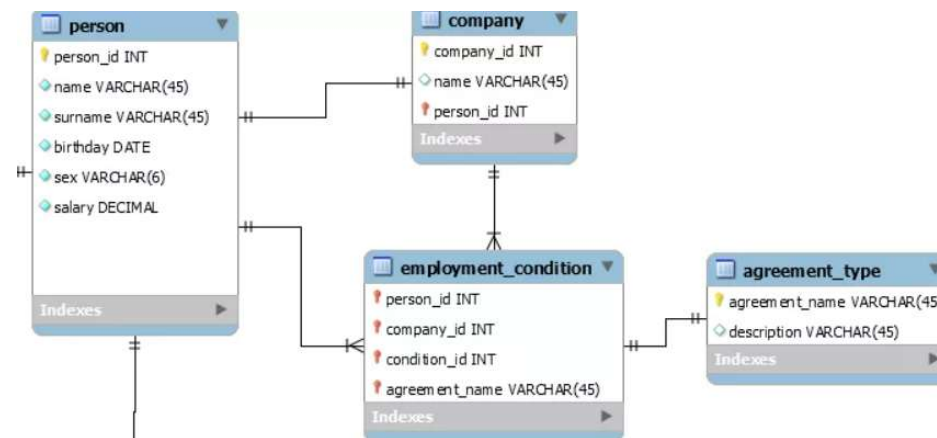
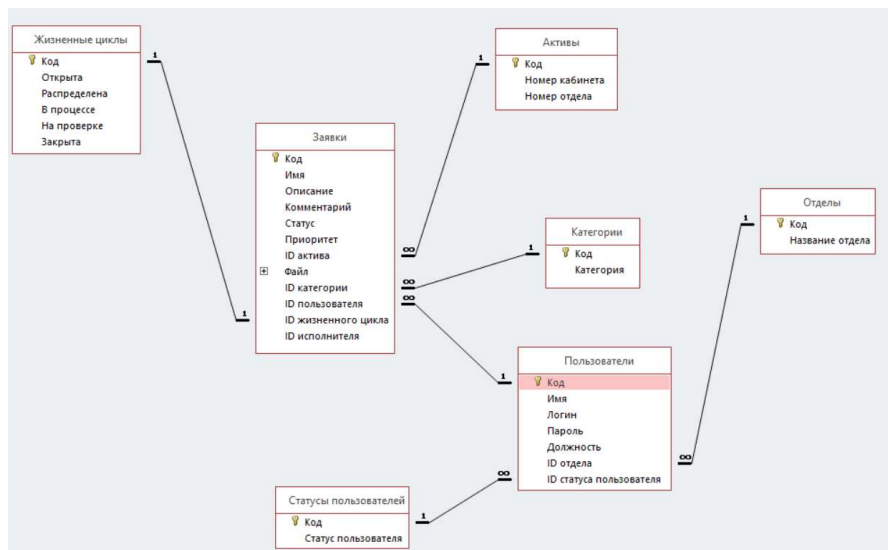


6.



8.

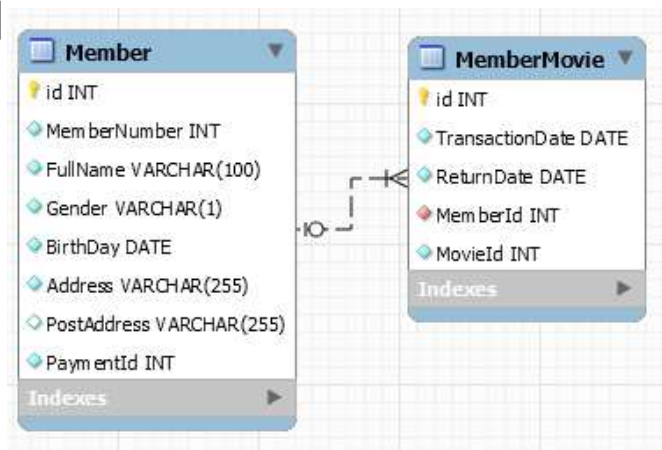
7.



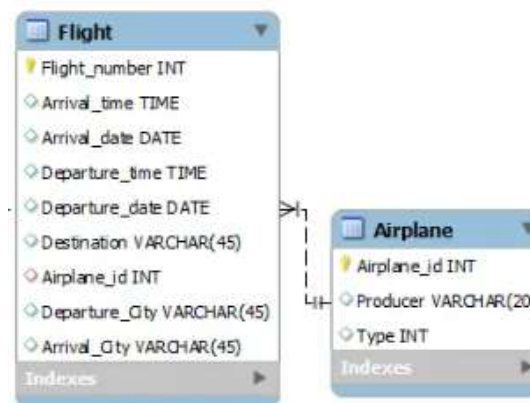
11

Practice 05-5. Работа с интегрированными сервисами

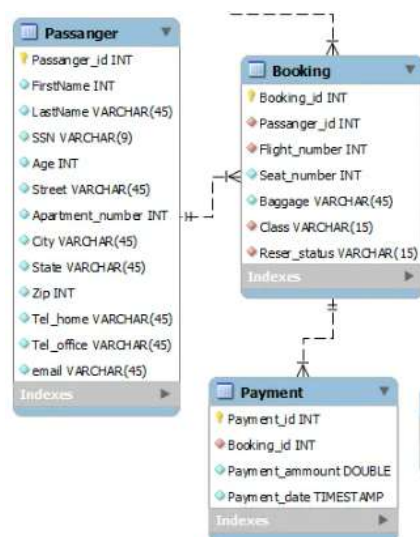
9. МИ



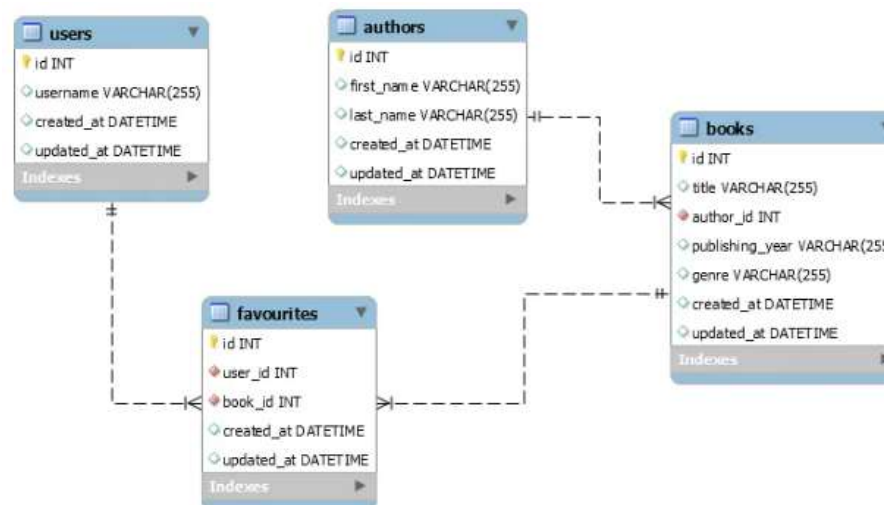
10.



11.

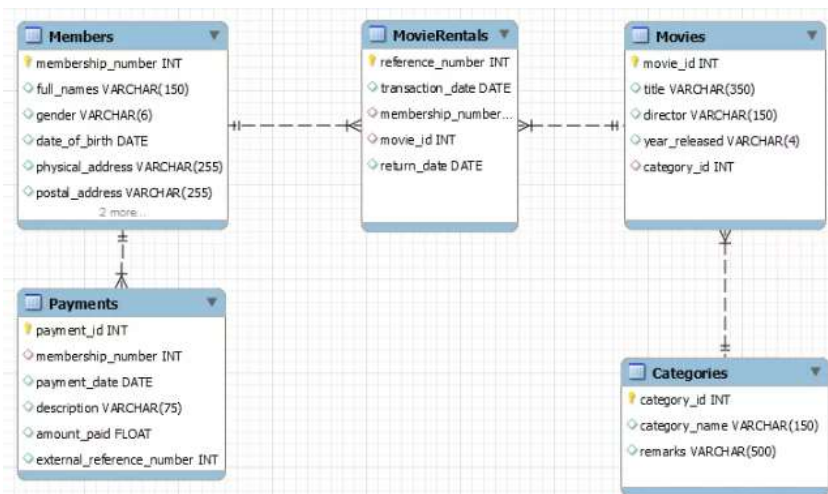


12.

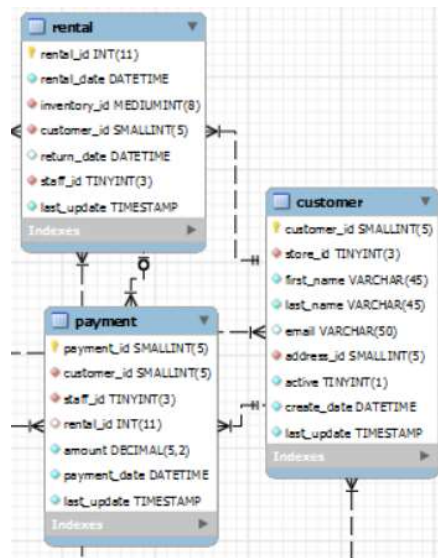


Practice 05-5. Работа с интегрированными сервисами.

13.

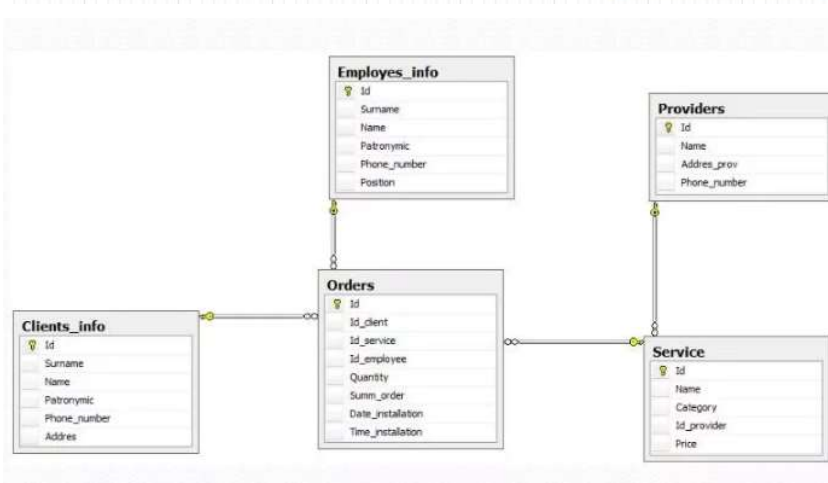


14.

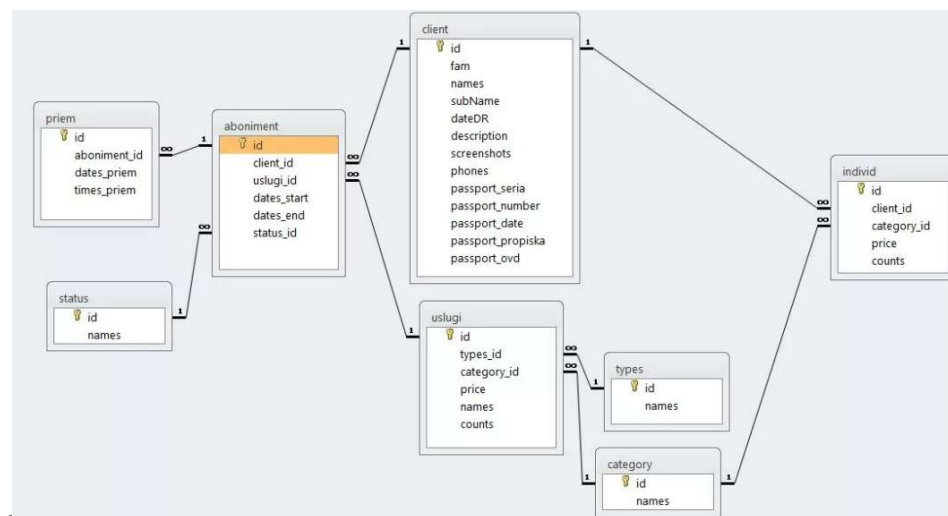


МОСКОВСКИЙ
ГОРОДСКОЙ
УНИВЕРСИТЕТ
МГПУ

15.



16.



13

СПАСИБО ЗА ВНИМАНИЕ