Создать docker compose с двумя сервисами, чтобы по одной команде они поднимались и останавливались:
Сервис 1: БД Postgres с одной таблицей , куда вставлена одна строчка данных.
Сервис 2: Программу на любом языке программирования, которая читает данные из этой БД и печатает эту строчку.

Вот пример docker-compose.yml файла и кода для выполнения вашего запроса. Я использую два сервиса: Postgres и Python (с использованием библиотеки psycopg2 для подключения к базе данных).

Структура проекта:

```
project/
├── docker-compose.yml
├── app/
│   ├── Dockerfile
│   ├── requirements.txt
│   └── app.py
└── init.sql
```

1. docker-compose.yml

```
version: '3.8'

services:
  db:
    image: postgres:15
    container_name: postgres_container
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: password
      POSTGRES_DB: testdb
    volumes:
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    ports:
      - "5432:5432"

  app:
    build:
      context: ./app
    container_name: python_app
    depends_on:
      - db
    environment:
```

```
    DB_HOST: db
    DB_PORT: 5432
    DB_NAME: testdb
    DB_USER: user
    DB_PASSWORD: password
```

2. init.sql (инициализация базы данных)

Этот файл создаёт таблицу и вставляет одну строку данных в базу данных Postgres.

```
CREATE TABLE example_table (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50)
);

INSERT INTO example_table (name) VALUES ('Hello, Docker!');
```

3. app/Dockerfile

Создаём Dockerfile для Python-приложения.

```
# Используем официальный образ Python
FROM python:3.9-slim

# Устанавливаем рабочую директорию
WORKDIR /app

# Копируем файлы приложения в контейнер
COPY requirements.txt requirements.txt

# Устанавливаем зависимости
RUN pip install --no-cache-dir -r requirements.txt

# Копируем приложение
COPY . .

# Команда для запуска приложения
CMD ["python", "app.py"]
```

4. app/requirements.txt

Указываем зависимости для Python-приложения.

```
psycopg2-binary==2.9.6
```

5. app/app.py

Программа на Python, которая подключается к базе данных и выводит строку из таблицы.

```
import os
import psycopg2

# Получаем параметры подключения из переменных окружения
DB_HOST = os.getenv("DB_HOST", "localhost")
DB_PORT = os.getenv("DB_PORT", "5432")
DB_NAME = os.getenv("DB_NAME", "testdb")
DB_USER = os.getenv("DB_USER", "user")
DB_PASSWORD = os.getenv("DB_PASSWORD", "password")

def main():
    try:
        # Подключаемся к базе данных
        conn = psycopg2.connect(
            host=DB_HOST,
            port=DB_PORT,
            database=DB_NAME,
            user=DB_USER,
            password=DB_PASSWORD,
        )
        cursor = conn.cursor()

        # Выполняем запрос к таблице
        cursor.execute("SELECT * FROM example_table;")
        rows = cursor.fetchall()

        # Выводим данные из таблицы
        for row in rows:
            print(f"ID: {row[0]}, Name: {row[1]}")

        # Закрываем соединение с базой данных
        cursor.close()
        conn.close()
    except Exception as e:
        print(f"Error: {e}")

if __name__ == "__main__":
    main()
```

Как запустить проект:

1. Убедитесь, что у вас установлен Docker и Docker Compose.

2. Перейдите в папку с проектом.

3. Выполните команду для запуска всех сервисов:

```
docker-compose up --build
```

4. Программа автоматически подключится к базе данных, прочитает строку из таблицы и выведет её в консоль.

Как остановить проект:

Чтобы остановить и удалить контейнеры, выполните:
docker-compose down