МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Лабораторная работа №\_\_1\_\_**

по дисциплине«Постреляционные базы данных»

Тема: «Создание объектно-реляционной базы данных

на примере СУБД PostgreSQL»

ИСПОЛНИТЕЛЬ: \_\_Журавлев Н.В.\_\_\_

ФИО

группа ИУ5-24М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"12"\_февраля\_2024 г.

ПРЕПОДАВАТЕЛЬ: \_Виноградова М.В\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_\_"\_\_\_\_\_\_\_\_\_202\_ г.

Москва - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Цель работы**

* Изучить объектно-реляционную модель данных и возможности языка SQL для работы с ней;
* Освоить построение запросов на языке SQL для описания ограничений целостности, определения и использования полей сложных типов данных и наследования таблиц;
* Получить навыки создания объектно-реляционной базы данных и выполнения запросов к ней на примере СУБД PostgreSQL в среде PgAdmin.

**Задание**

1. В среде PgAdmin (PostgreSQL) создать БД. В БД создать две-три таблицы по теме, выданной преподавателем, связанные как один-ко-многим, содержащие первичные и внешние ключи и ограничения: уникальности, Not NULL, на значения, на значение по умолчанию, автоинкремент.
2. Продемонстрировать DDL сценарий создания таблицы (открывается в визуальном редакторе).
3. Открыть таблицы на редактирование и заполнить тестовыми данными. Проверить действие ограничений.
4. Продемонстрировать CRUD запросы, используя SQL (окно запросов) и визуальный редактор (мастер).
5. Добавить в таблицы поля составных типов (геометрические, ISON, диапазоны) и продемонстрировать работу с ними (в окне запросов).

**Ход работы**

Создаём таблицы с помощью команд:

CREATE TABLE account (

id SERIAL PRIMARY KEY NOT NULL,

age integer CHECK (age > 0),

height real CHECK (height > 0),

weight real CHECK (weight > 0),

login varchar,

password varchar

);

CREATE TABLE diet (

id SERIAL PRIMARY KEY NOT NULL,

name varchar NOT NULL,

description varchar,

age integer CHECK (age > 0),

height real CHECK (height > 0),

weight real CHECK (weight > 0)

);

CREATE TABLE history\_diet (

id SERIAL PRIMARY KEY NOT NULL,

id\_account integer,

id\_diet integer,

FOREIGN KEY (id\_diet) REFERENCES diet (id),

FOREIGN KEY (id\_account) REFERENCES account (id)

);

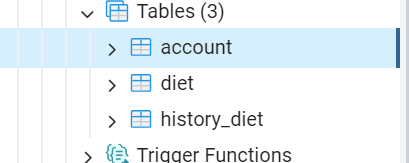


Рисунок 1

Запросы CRUD на примере таблицы history\_diet:

INSERT INTO public.history\_diet(id, id\_account, id\_diet) VALUES (?, ?, ?);

SELECT id, id\_account, id\_diet FROM public.history\_diet;

UPDATE public.history\_diet SET id=?, id\_account=?, id\_diet=? WHERE <condition>;

DELETE FROM public.history\_diet WHERE <condition>;

Добавим в таблицу account поля с типом json, line, int4range. Для этого необходимо создать этот тип с помощью:

ALTER TABLE account ADD add\_type\_json json;

ALTER TABLE account ADD add\_type\_line line;

ALTER TABLE account ADD add\_type\_range int4range;

Для добавления значений в эти типы необходимо:

UPDATE public.account SET add\_type\_json = '{"b": 0}';

UPDATE public.account SET add\_type\_line = '((0, 0), (1, 1))';

UPDATE public.account SET add\_type\_range = '(1, 10)';

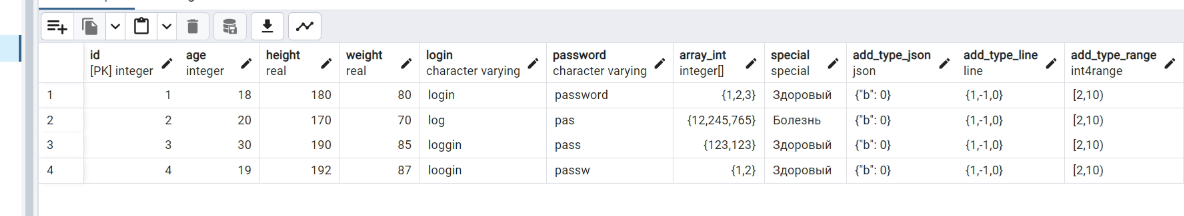


Рисунок 2

Добавим в таблицу account поле массива с помощью команды:

ALTER TABLE account ADD COLUMN array\_int integer[] [];

Результат можно вывести с помощью команды:

SELECT \* FROM public.account

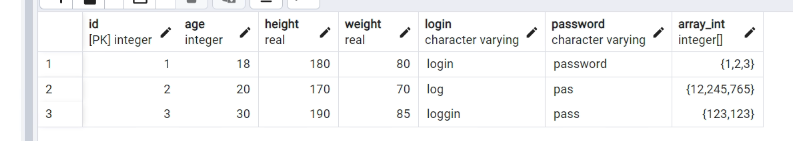


Рисунок 3

Добавим в таблицу account поле с типом ENUM. Для этого необходимо создать этот тип с помощью:

CREATE TYPE special AS ENUM ('Болезнь', 'Здоровый');

Затем добавляем это поле к таблице с помощью команды:

ALTER TABLE account ADD COLUMN special special;

Результат можно вывести с помощью команды:

SELECT \* FROM public.account

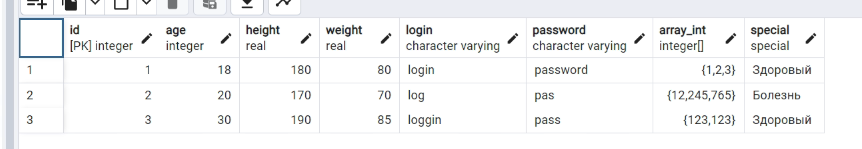


Рисунок 4

Добавим таблицу type\_account, которая наследуется от account с помощью команды:

CREATE TABLE type\_account (

type integer

) INHERITS (account);

Результат можно вывести с помощью команды:

SELECT \* FROM public.account

Для добавления используется команда:

INSERT INTO public.type\_account (age, height, weight, login, password, array\_int, special, type) VALUES (19, 192, 87, 'loogin', 'passw', ARRAY [1,2], 'Здоровый', 1)

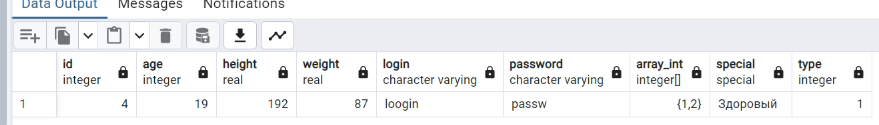


Рисунок 5

Если мы выведем строки из базовой таблицы, к ним добавятся строки из дочерней.

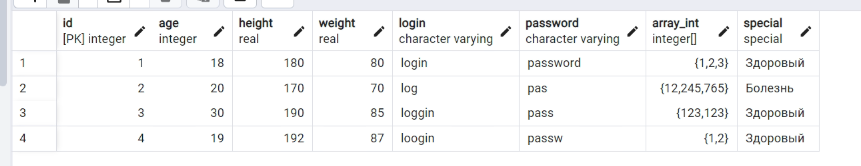


Рисунок 6

Чтобы вывести только строки базовой таблицы, необходимо использовать ONLY.

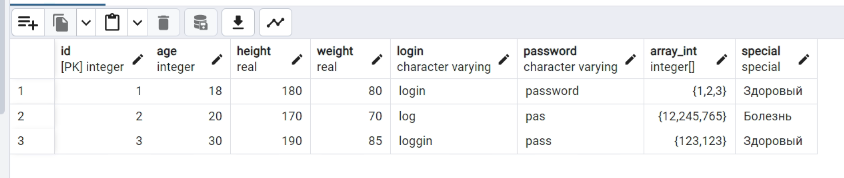


Рисунок 7

Вывод результата только дочерней таблицы происходит привычным образом через команду SELECT.

**Вывод**

В результате выполнения работы были изучены возможности для работы с объектно-ориентированной моделью данных и возможности языка SQL для работы с ней. Освоено построение запросов на языке SQL для описания ограничений целостности, определения и использования полей сложных типов данных и наследования таблиц. Получены навыки создания объектно-реляционной базы данных и выполнения запросов к ней на примере СУБД PostgreSQL в среде PgAdmin.

**Список используемой литературы**

1. Виноградов В.И., Виноградова М.В. Постреляционные модели данных и языки запросов: Учебное пособие. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. – 100с. - ISBN 978-5-7038-4283-6.
2. PostgreSQL 14.2 Documentation. – Текст. Изображение: электронные // PostgreSQL : [сайт]. – URL: https://www.postgresql.org/docs/14/index.html (дата обращения: 12.02.2024)
3. pgAdmin 4 6.5 documentation. – Текст. Изображение: электронные // pgAdmin - PostgreSQL Tools : [сайт]. – URL: https://www.pgadmin.org/docs/pgadmin4/6.5/index.html (дата обращения: 12.02.2024)
4. PostgreSQL : Документация: 14: 8.15. Массивы. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/arrays (дата обращения: 12.02.2024)
5. PostgreSQL : Документация: 14: 8.16. Составные типы. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/rowtypes (дата обращения: 12.02.2024)
6. PostgreSQL : Документация: 14: 8.7. Типы перечислений. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/datatype-enum (дата обращения: 12.02.2024)
7. PostgreSQL: Документация: 14: 5.10. Наследование. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/ddl-inherit (дата обращения: 12.02.2024)
8. PostgreSQL: Документация: 14: 38.13. Пользовательские типы. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/xtypes (дата обращения: 12.02.2024)
9. PostgreSQL: Документация: 14: CREATE TYPE. – Текст. Изображение : электронные // Компания Postgres Professional : [сайт]. – URL: https://postgrespro.ru/docs/postgresql/14/sql-createtype (дата обращения: 12.02.2024)