**Липецкий государственный технический университет**

Факультет автоматизации и информатики Кафедра автоматизированных систем управления

Задание №2

по дисциплине «Расширенный курс по базам данных»

Студент Титов В.А.

Группа АС-20-1

Руководитель Мирошников А.И.

Липецк 2023 г.

Создать таблицы, содержание поля типов:

1. **Стандартные типы данных (Числовой, символьный, логический, дата/время)**

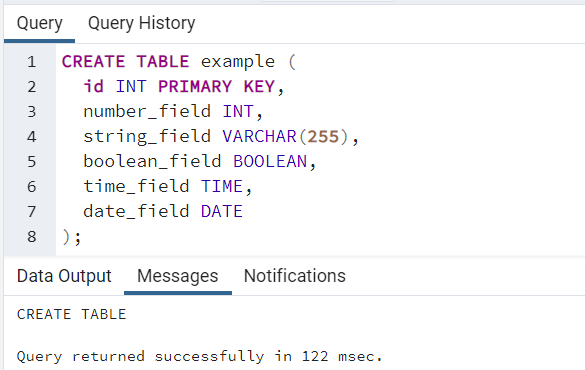


Рис.1. Стандартные типы данных

**INSERT и SELECT:**

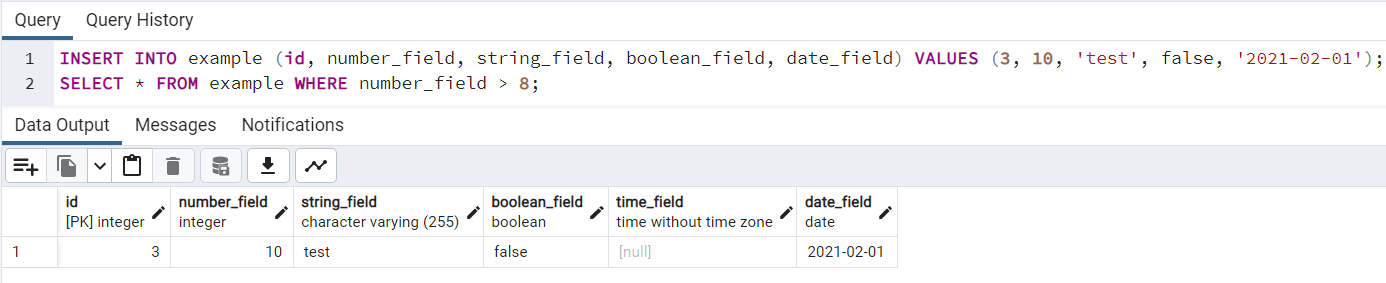


Рис.2. INSERT и SELECT

1. **Перечисления**

В SQL для создания перечислений можно использовать тип данных ENUM. Пример создания таблицы с полем типа ENUM:

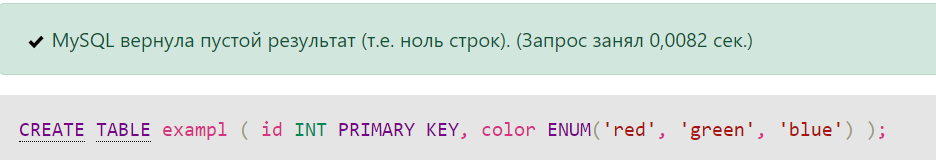


Рис.3. Перечисления

В данном примере создается таблица "exampl" с двумя полями: "id" типа INT и "color" типа ENUM, который может принимать только одно из трех значений: "red", "green" или "blue".

**INSERT и SELECT:**



Рис.4. INSERT и SELECT

1. **Массивы**

SQL не поддерживает массивы в структуре таблиц. Вместо этого, можно создать отдельную таблицу для хранения значений массива и связать ее с основной таблицей через внешний ключ. Например:

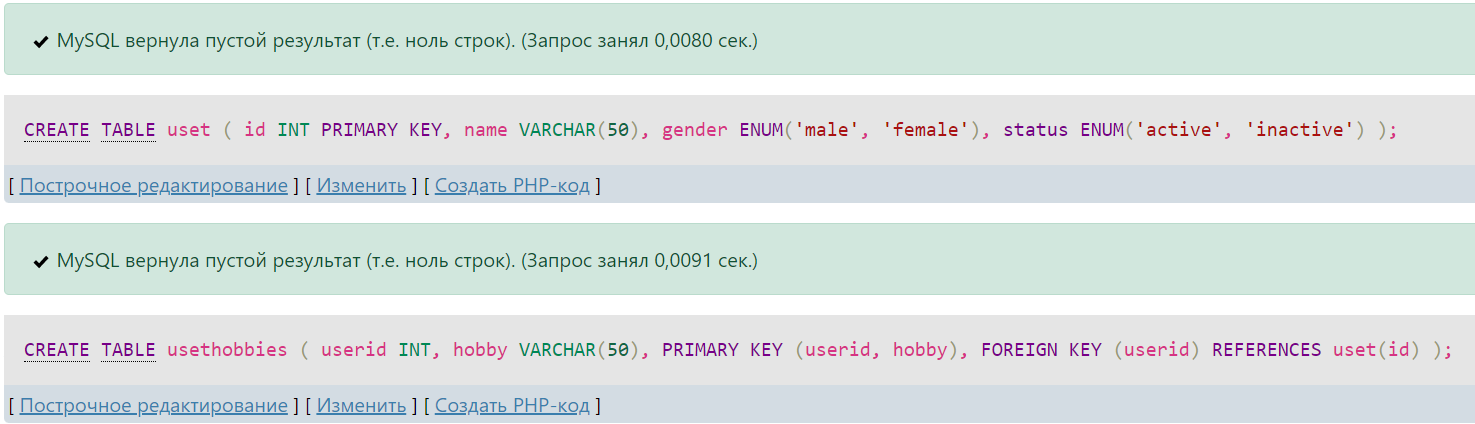
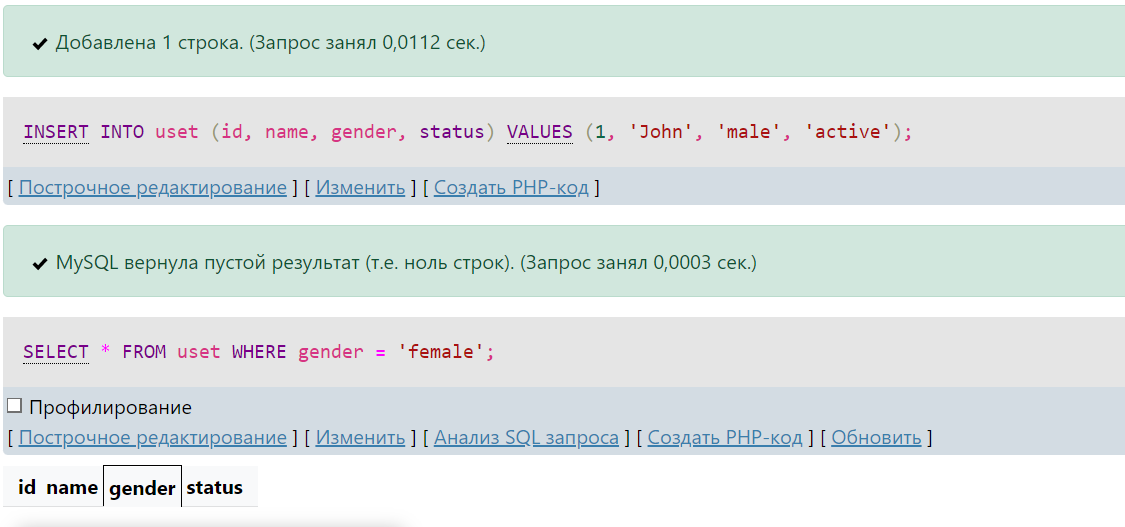


Рис.5. Массивы

В этом примере таблица usethobbies содержит хобби для каждого пользователя. Столбец userid является внешним ключом, который ссылается на столбец id таблицы uset. Ключевое слово PRIMARY KEY определяет состав первичного ключа таблицы usethobbies, который состоит из столбцов userid и hobby.

**INSERT и SELECT:**



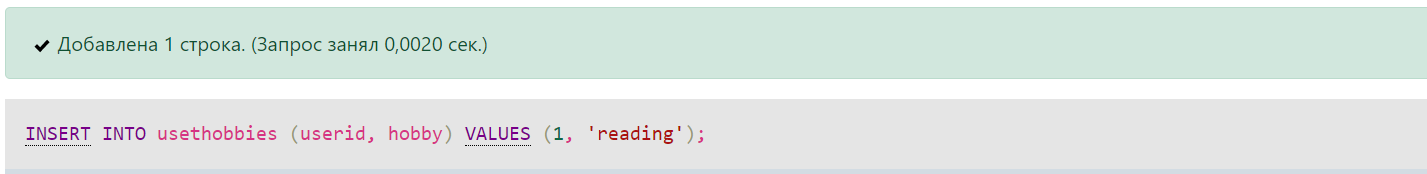


Рис.6-7. INSERT и SELECT

1. **XML и JSON**

SQL не поддерживает нативную работу с XML и JSON, но существуют специальные типы данных и функции для работы с ними в различных СУБД. Например, в PostgreSQL можно использовать тип данных JSON:

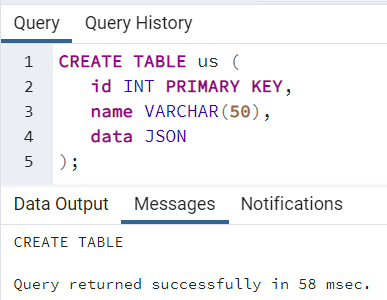


Рис.8. XML и JSON

В этом примере таблица us содержит столбец data типа JSON, который может хранить данные в формате JSON. Для работы с данным столбцом можно использовать функции JSONEXTRACT и JSONSET.

**INSERT и SELECT:**

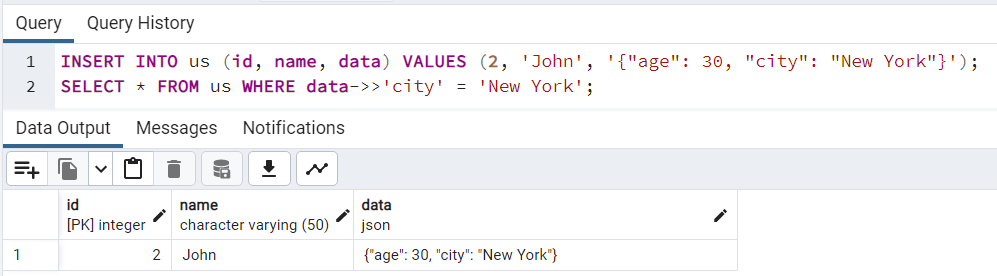


Рис.9. INSERT и SELECT

В PostgreSQL можно использовать тип данных XML:

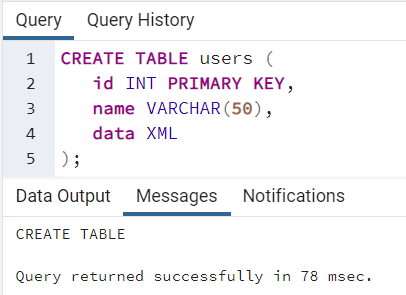


Рис.10. Создание таблицы с XML

В этом примере таблица users содержит столбец data типа XML, который может хранить данные в формате XML. Для работы с данным столбцом можно использовать функции XPath и XQuery.

**INSERT и SELECT:**

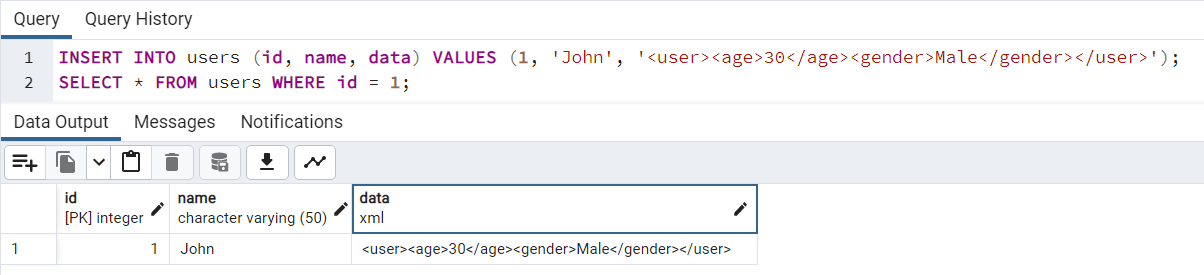


Рис.11. INSERT и SELECT

1. **Составные типы**

В SQL можно создавать составные типы данных, которые представляют собой набор полей различных типов. Обычно такие типы используются для описания сложных объектов, например, заказов или платежей.

Для создания составного типа в SQL используется оператор CREATE TYPE, за которым следует имя типа и список полей соответствующих типов. Например, в PostgreSQL можно создать тип данных ord:

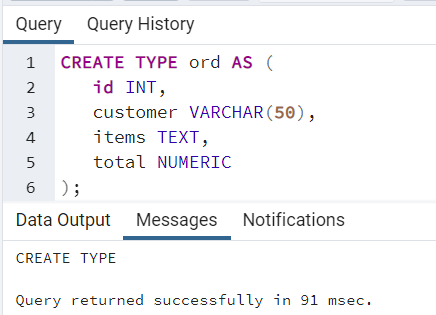


Рис.12. Создание типа ord

В этом примере тип ord содержит четыре поля: id типа INT, customer типа VARCHAR(50), items типа TEXT (массив строк) и total типа NUMERIC.

После создания типа его можно использовать для создания таблиц, например:

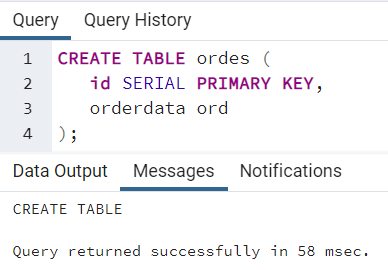


Рис.13. Пример создания таблицы с типом order

В этом примере таблица ordes содержит столбец orderdata типа ord, который может хранить данные о заказе.

**INSERT и SELECT:**

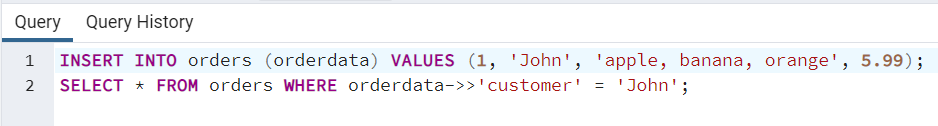


Рис.14. INSERT и SELECT

1. **Прочие типы: денежный, двоичный, геометрический, битовые строки, UUID**

**Денежный тип:**

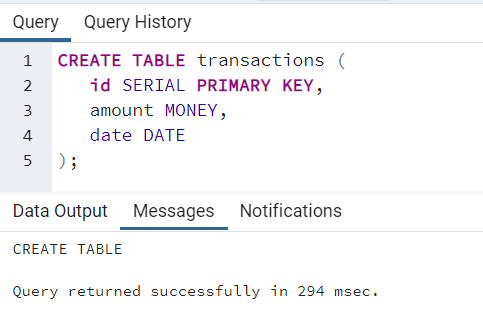


Рис.15. Денежный тип

В этом примере таблица transactions содержит столбец amount типа MONEY, который может хранить данные о сумме транзакции.

**INSERT и SELECT:**

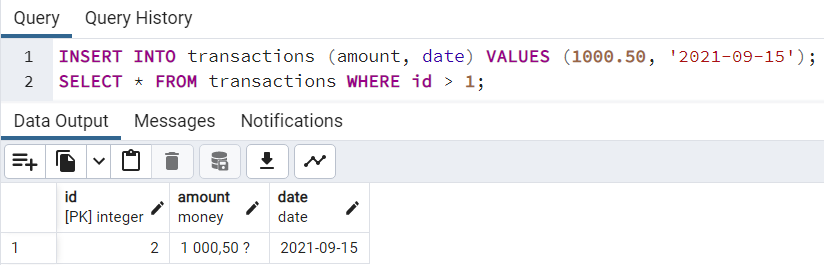


Рис.16. Денежный тип

**Двоичный тип:**

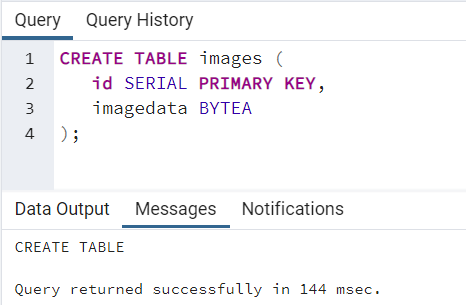


Рис.17. Двоичный тип

**INSERT и SELECT:**

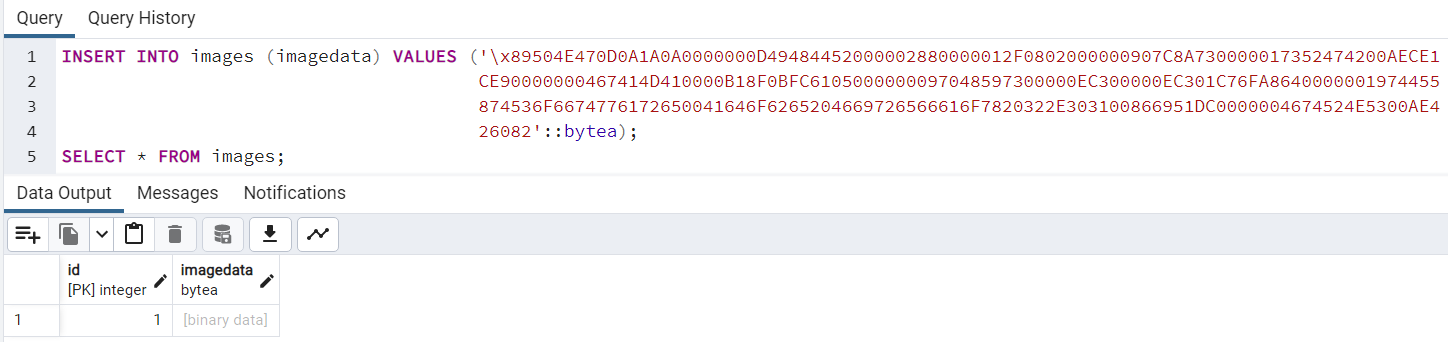


Рис.18. INSERT и SELECT

В этом примере таблица images содержит столбец imagedata типа BYTEA, который может хранить бинарные данные, например, изображения.

**Геометрический тип:**

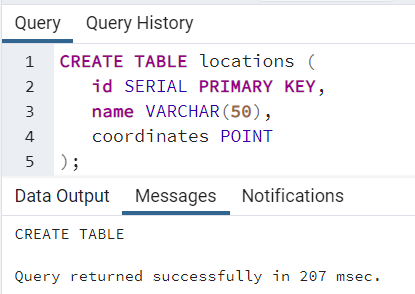


Рис.19. Геометрический тип

В этом примере таблица locations содержит столбец coordinates типа POINT, который может хранить координаты местоположения.

**INSERT и SELECT:**

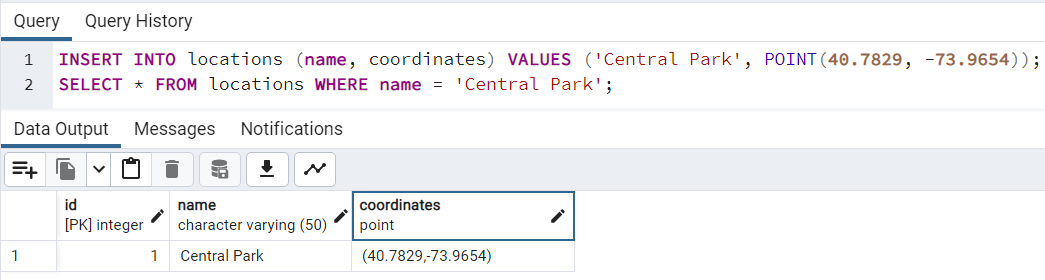


Рис.20.INSERT и SELECT

**Битовые строки:**

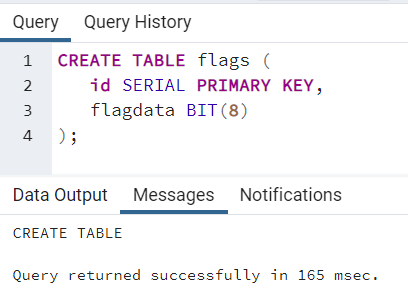


Рис.21. Битовые строки

В этом примере таблица flags содержит столбец flagdata типа BIT(8), который может хранить битовые флаги.

**INSERT и SELECT:**

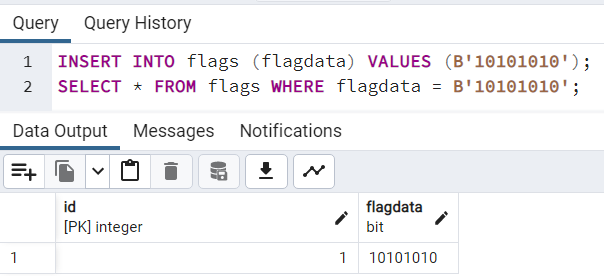


Рис.22. INSERT и SELECT

**UUID:**

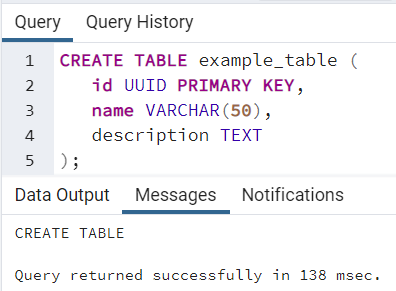


Рис.23. UUID

В этом примере мы создаем таблицу example\_table с тремя столбцами: id, name и description. Столбец id имеет тип данных UUID и является первичным ключом таблицы. Остальные столбцы имеют обычные типы данных VARCHAR и TEXT.

**Пример запроса INSERT:**

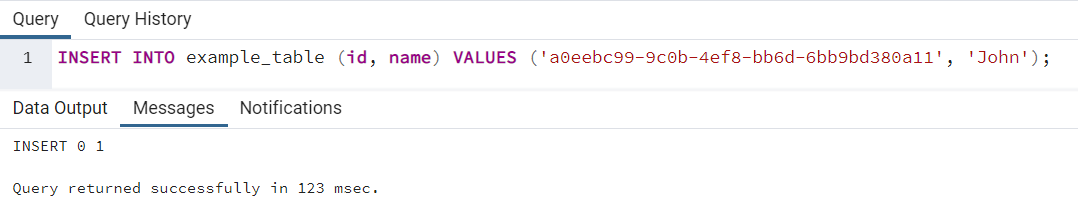


Рис.24. Пример запроса INSERT

Здесь мы добавляем новую запись в таблицу example\_table с указанием значения UUID в столбце id и имени "John" в столбце name.

**Пример запроса SELECT с условием:**



Рис.25. Пример запроса SELECT с условием

Здесь мы выбираем все записи из таблицы example\_table, где значение столбца id равно указанному UUID.