

Программа Перезапуск

Модуль SQL.

Занятие 4

Преподаватель: Марат Гарафутдинов

Что смотрели на предыдущих занятиях

- Почему SQL нужен
- Реляционные таблицы
- Базовые операции
 - SELECT, INSERT, UPDATE, DELETE
- Операторы
 - WHERE, GROUP BY, AND, OR, NOT
- Операции JOIN
- Подзапросы
- Конструкции CASE WHEN
- Операторы ANY, ALL, IS NOT NULL

Работаем с DBeaver Community Edition

Реквизиты:

52.157.159.24

login: student0

password: student0_password

database: sql_ex_for_student0

Но сейчас работаем только со своей базой!

Оператор: LIKE

Оператор SQL LIKE устанавливает соответствие символьной строки с шаблоном.

Символ % соответствует любому количеству любых символов, а также их отсутствию.

Символ _ соответствует ровно одному любому символу.

Найти все корабли, имена классов которых заканчиваются на букву 'o'

```
SELECT *  
FROM Ships  
WHERE class LIKE '%o' ;
```

Найти все корабли, имена классов которых заканчиваются на букву 'o', но не на 'go'

```
SELECT *  
FROM Ships  
WHERE class NOT LIKE '%go' AND  
class LIKE '%o' ;
```

Типы данных

тип	описание	пример
INT, NUMBER	Целое число	11, 53, 0, -4
FLOAT, DECIMAL	Дробное число	11.12, 42.0, -14.2
VARCHAR, TEXT	Текст	"Mexico" "" "12.4"
TIME, DATETIME, TIMESTAMP	Дата и время	2008-10-29 14:56:59 1225292219

Базовые типы данных. Информацию по остальным можно найти в документации:
<https://metanit.com/sql/postgresql/2.3.php>
<https://www.postgresql.org/docs/9.5/datatype.html>

Команда: CREATE TABLE

```
create table [IF NOT EXISTS] employee (
```

```
    employee_id smallint AUTO_INCREMENT PRIMARY KEY,
```

```
    last_name varchar(15),
```

```
    first_name varchar(15),
```

```
    middle_initial varchar(1),
```

```
    manager_id smallint,
```

```
    job_id smallint,
```

```
    hire_date datetime,
```

```
    salary numeric(7,2),
```

```
    commission numeric(7,2),
```

```
    department_id smallint
```

```
);
```

```
CREATE TABLE films2 AS  
TABLE films;
```

```
CREATE TABLE films_recent AS  
SELECT * FROM films WHERE  
date_prod >= '2002-01-01';
```

Команда: CREATE VIEW

CREATE VIEW описывает вид данных основанный на одной или более основной таблицы в базе данных.

Возвращаемые строки определены инструкцией SELECT, который перечисляет столбцы исходных таблиц.

Только определение вида существует в базе данных; вид непосредственно не представляет, из себя, сохраненные данные.

Это позволяет выполнять операции выбора, project, join и union над видами, как если бы они были таблицами.

```
CREATE VIEW name [(view_col [,  
view_col ...]]  
AS <select> [WITH CHECK OPTION];
```

```
CREATE VIEW films recent AS  
SELECT * FROM films WHERE  
date_prod >= '2002-01-01';
```

TABLE vs VIEW

Таблица: таблица-это предварительное хранилище для хранения данных и информации.

Таблица - это набор связанных записей данных, состоящий из столбцов и строк.

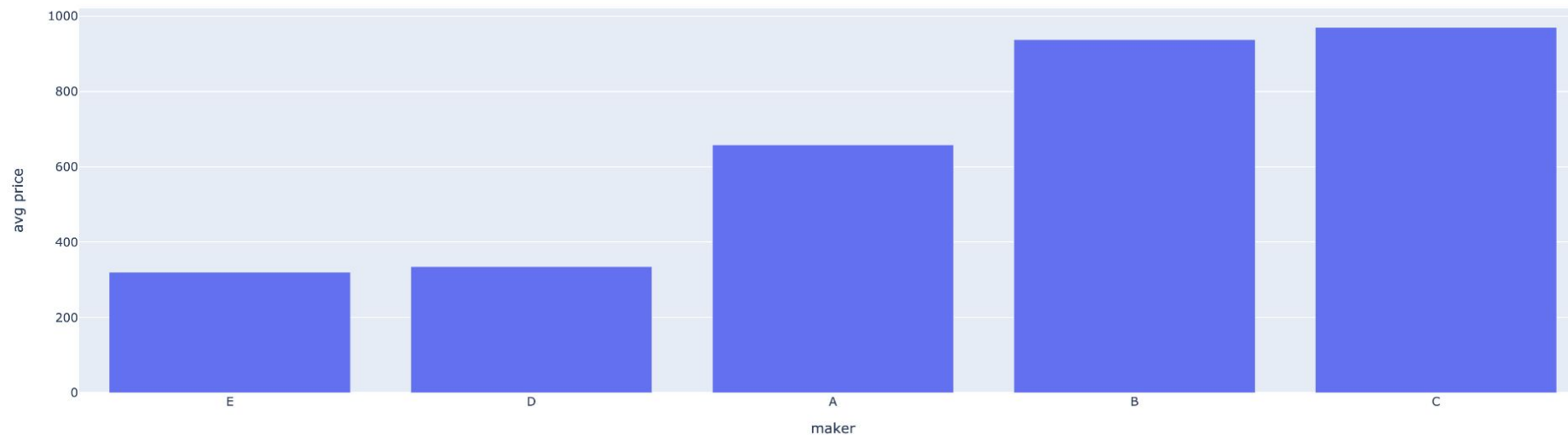
View: a view-это виртуальная таблица, содержимое которой определяется запросом.

Преимущества перед таблицей заключаются в следующем:

- Представления могут использоваться в качестве механизмов безопасности, позволяя пользователям получать доступ к данным через view, не предоставляя пользователям разрешений на прямой доступ к базовым таблицам view.
- Он действует как абстрактный слой для нижестоящих систем, поэтому любое изменение схемы не подвергается воздействию и, следовательно, нижестоящие системы не затрагиваются.
- Мы можем объединить столбцы/строки из нескольких таблиц или другой view и иметь консолидированный view.

Визуализация данных в Colab

```
fig = px.bar(x=df.make.to_list(), y=df.avg.to_list(), labels={'x':'maker', 'y':'avg price'})  
fig.show()
```



https://colab.research.google.com/drive/1j4XdGIU_NYPVpv74vQa9HUOAkxsgUez?usp=sharing

Демонстрация и решение упражнений