

SQL

Занятие 7

DDL операции

Бояр Владислав

DDL операции

Группы операторов SQL

- **DML (Data Manipulation Language)** – манипуляции с данными
 - SELECT – выборка;
 - INSERT – вставка;
 - UPDATE – обновление;
 - DELETE – удаление.
- **DDL (Data Definition Language)** – работа с объектами БД
 - CREATE – создание;
 - ALTER – изменение;
 - DROP – удаление.
- **DCL (Data Control Language)** – определение доступа к данным
 - GRANT – предоставление прав;
 - REVOKE – отзыв прав;
 - DENY – запрет действий.
- **TCL (Transaction Control Language)** – управление транзакциями


Группы операторов SQL

- **DML (Data Manipulation Language)** – манипуляции с данными
 - SELECT – выборка;
 - INSERT – вставка;
 - UPDATE – обновление;
 - DELETE – удаление.
- **DDL (Data Definition Language)** – работа с объектами БД
 - CREATE – создание;
 - ALTER – изменение;
 - DROP – удаление.
- **DCL (Data Control Language)** – определение доступа к данным
 - GRANT – предоставление прав;
 - REVOKE – отзыв прав;
 - DENY – запрет действий.
- **TCL (Transaction Control Language)** – управление транзакциями

CREATE

- **CREATE** позволяет создавать объекты БД (схемы, таблицы, функции и т.д.)
- **CREATE TABLE** создаёт новую, изначально пустую таблицу в БД.
- Временные таблицы автоматически удаляются в конце сеанса.
- Синтаксис создания таблицы:

```
CREATE [TEMPORARY] TABLE schema_name.table_name [IF EXISTS] (  
    field_name_1 data_type(integer, varchar, ...),  
    field_name_2 ...,  
    ...  
    field_name_n ...  
);
```



ALTER

- Позволяет изменить параметры объекта БД (изменить имя колонки, установить ограничение на поле и тд);
- **ALTER TABLE** изменяет параметры таблицы.

Общие синтаксис:

ALTER TABLE table_name действие;

Примеры действий:

ADD COLUMN column_name column_data_type -- добавляет поле

DROP COLUMN column_name -- удаляет поле

RENAME TO new_table_name -- переименовывает таблицу

RENAME COLUMN old_column_name **TO** new_column_name -- переименовывает поле

SET SCHEMA new_schema -- перемещает таблицу в другую схему

ALTER COLUMN column_name **TYPE** new_data_type -- изменение типа данных поля

DROP

- Удаляет объекты БД;
- **DROP TABLE** – удаляет таблицу из БД.
- Чтобы опустошить таблицу, не удаляя её саму, вместо этой команды следует использовать DELETE или TRUNCATE.

Синтаксис:

```
DROP TABLE [IF EXISTS] table_name [CASCADE];
```




Ограничения полей

Позволяют определить множество допустимых значений в таблице.

Основные виды ограничений:

- Первичный ключ (PRIMARY KEY)
 - уникальные, ненулевые значения;
 - может быть только один для таблицы;
 - При создании РК, по полям ключа автоматически создаётся индекс;
- Внешний ключ (FOREIGN KEY);
 - ссылается на поле из другой таблицы;
 - значение внешнего ключа может содержать только данные из поля, на которое он ссылается;

Ограничения полей

- Наличие нулевых значений (NULL / NOT NULL)
 - Проверка условия (CHECK)
 - **CHECK**(**length**(field_name) = 3)
 - **CHECK**(field_name **IN** [value_1, value_2, value_3])
 - Только уникальные значения (UNIQUE)
 - Значение по умолчанию (DEFAULT)
- 

Представление

- Представление (VIEW) – именованный запрос, к которому можно обратиться по имени.
- Само представление не хранит данные, только запрос.
- Данные, возвращаемые представлением, обновляются в момент обращения к нему.
- Используется для формирования витрин данных и ограничения доступа к данным.

-- Создание представления

```
CREATE [OR REPLACE] VIEW view_name AS  
SELECT ...  
;
```

-- Обращение к представлению

```
SELECT ... FROM view_name;
```



Материализованное представление

- Материализованное представление (MATERIALIZED VIEW) – представление, которое сохраняет результат запроса в виде таблицы.
- В отличие от таблицы, сохраняет исходный запрос.
- Можно обновить данные, которые хранит материализованное представление.

-- Создание мат.представления

```
CREATE MATERIALIZED VIEW mat_view_name AS  
SELECT ...  
;
```

-- Обновление данных мат.представления

```
REFRESH MATERIALIZED VIEW mat_view_name;
```

Практика



Загружаем данные из файла: "da_data_n1-2.csv"

Таблицу каждый создаёт в своей схеме

При загрузке задаём соответствующие типы полей

После загрузки данных переименуем таблицу в goods и зададим первичный ключ

На основе загруженных данных создадим представление:

- Наименование товара
- Текущая стоимость товара (последняя из имеющихся данных)
- Минимальная стоимость товара
- Среднегодовая стоимость товара
- Среднемесячная стоимость товара
- Максимальная стоимость товара