**GenTech / Jan 31, 2023**

**CRUD**

**Выполнение запроса**

Backend —> | \*\*DBMS\*\* | —> Backend —> Frontend

**Проекция**

* Это ограничение полей (столбцов) на выходе
* Экономит выч/ресурсы и время выполнения запроса

**Получение данных на Backend**

* базы данных (массив ассоциативных массивов)
* из кеша (тоже БД)
* из файла
* парсинг данных
* API

**Фоновые задачи**

Примеры

* отправка email
* обработка видео
* прочие длительные процессы

**Валидация данных / SQL Constraints (ограничения)**

* При добавлении в таблицу

NOT NULL

UNIQUE

PRIMARY KEY (NOT NULL + UNIQUE)

CHECK

DEFAULT

FOREIGN KEY - позволяет проверить что значение есть в другой таблице

**Пример хранения BOOL**

is\_blocked: TINYINT (bool: 0 и 1)

**SQL: типы данных (отличается для каждой СУБД)**

* Инструмент описания данных

**Основные**

* числа
  + INT
  + TINYINT
  + DECIMAL / NUMERIC
* строки
  + varchar
  + text
  + char
* дата/время
  + date
  + datetime
  + time
  + timestamp

**SELECT DISTINCT**

* выводит только уникальные (отличающиеся) значения

SELECT DISTINCT Country

FROM Customers

**LIMIT**

* ограничивает ко-во записей на выходе
* также часто применяется для “пагинации” (построничного вывода)

LIMIT 5

LIMIT 5 OFFSET 5

IS NULL

IS NOT NULL

**ORDER BY**

Оператор сортировки

ORDER BY Price DESC

ORDER BY Price ASC

ORDER BY Price DESC, Country ASC

**LIKE**

Проверка по шаблону

* простой аналог regex (регулярных выражений)

WHERE CustomerName LIKE ‘%us%’

\_ - обозначает один символ

% - либо ноль либо множество

**UNION / UNION ALL**

UNION - только уникальные значения

UNION ALL - все значения

**JOIN**

Выполняет горизонт/объединение таблиц

(INNER) JOIN - “строгое” пересечение

LEFT JOIN

RIGHT JOIN

FULL JOIN

**SUBQUERIES / NESTED QUERIES**

WHERE customer\_id IN (<вложенный запрос>)

**CASE**

Условная конструкция

**Порядок подготовки БД**

* создание отдельного пользователя и наделение полномочиями
* создание БД CREATE DATABASE <название>
  + CREATE DATABASE \*\*IF NOT EXISTS\*\* <название>
* Создание таблиц CREATE TABLE <название> (<описание>)
  + CREATE TABLE IF NOT EXISTS <название> (<описание>)

id INT PRIMARY KEY

DROP DATABASE <название> - удалить БД

DROP TABLE <название> - удалит таблицу

DROP TABLE IF EXISTS <название>

**Очистика таблицы**

DELETE FROM <таблица>

TRUNCATE TABLE <таблица>

**Мат. операторы**

+

-

\*

/

%

**Операторы сравнения**

=

<>, !=, NOT =

<

>

>=

<=

**Логические операторы**

NOT

AND

OR

**Комментирование кода**

-- одностроный

/\* многострочный

комментарий \*/

**Агрегация в SQL**

Получение вычисленных данных (сборка, конструктор)

COUNT()

SUM()

AVG()

MIN()

MAX()

GROUP BY - группировка

HAVING - фильтрация агрегированных данных

**Справочная информация**

[Типы данных](https://dev.mysql.com/doc/refman/8.0/en/data-types.html)

[UUID](https://ru.wikipedia.org/wiki/UUID)

**Виды FE**

* приложение (допустим. моб.) - человеко-чит.
* веб-сайты - человеко-чит.
* API - машин.-чит.
  + боты - человеко-чит.