



# SQL

Structured Query Language или Язык  
Структурированных Запросов

# Как устроена база данных?

- Сервер (на одном сервере могут быть несколько баз)
- База (одна база может быть на многих серверах)
- Таблица (в одной базе может быть много таблиц)
- Ряд (или сущность) (в одной таблице может быть много сущностей)

# У базы есть несколько важных свойств

- Имя
- Права доступа
- Кодировка
- "Системные" параметры

# Простой вариант

<http://sqlfiddle.com/#!17/ec5f9/8>

# Создание пользователя для базы данных

```
CREATE ROLE tceh LOGIN PASSWORD '123'  
CREATEDB CREATEROLE  
VALID UNTIL 'infinity';
```

# Создание базы

```
CREATE DATABASE tceh_db  
  WITH ENCODING='UTF8'  
  OWNER=tceh  
  CONNECTION LIMIT=-1;
```

# Создание таблицы

Создадим последовательность, чтобы назначать id:

```
CREATE SEQUENCE public.observations_id_seq;
```

Создадим таблицу:

```
CREATE TABLE public.observations
(
  id integer NOT NULL DEFAULT nextval('observations_id_seq'::regclass),
  observation_date date DEFAULT now(),
  people_available integer,
  CONSTRAINT observations_pkey PRIMARY KEY (id)
);
ALTER TABLE public.observations
  OWNER TO tceh;
```



# Вставим первые данные

```
INSERT INTO public.observations(people_available)  
VALUES (12);
```

# Результат многократной вставки

id;observation\_date;people\_available

1;"2016-03-21";12

2;"2016-03-21";19

3;"2016-03-21";14

4;"2016-03-21";40

5;"2016-03-21";4

6;"2016-03-21";10

7;"2016-03-21";14

# Выберем нужные данные

```
SELECT /*DISTINCT*/ observation_date  
  FROM public.observations  
  WHERE people_available >= 10;
```

# Практика

<https://sqlbolt.com>

# Отношения

- Один-к-одному
- Один-ко-многим
- Многие-к-многим
- (Многие-к-одному)

# ВНЕШНИЕ КЛЮЧИ

```
CREATE TABLE public.results
(
    id integer NOT NULL DEFAULT
nextval('result_id_seq'::regclass),
    name character varying(30) NOT NULL,
    decision character varying(140) NOT NULL,
    observation_id integer NOT NULL,

    CONSTRAINT results_pkey PRIMARY KEY (id),
    CONSTRAINT results_observation_fk FOREIGN KEY (observation_id)
        REFERENCES public.observations (id) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE
INITIALLY DEFERRED
);
ALTER TABLE public.results
    OWNER TO tceh;
```

# Выборка данных с использованием внешнего ключа

```
SELECT * from observations  
      INNER /*LEFT OUTER*/ JOIN results AS r ON (  
          observations.id = r.observation_id  
      ) ;
```

# Нормальные формы

<https://habrahabr.ru/post/254773>



# Проектирование баз

<https://habrahabr.ru/post/193380>