



# **Создание базы данных, основы работы с таблицами**

# Команда CREATE DATABASE

- Синтаксис команды CREATE DATABASE имеет вид:
- CREATE DATABASE [IF NOT EXISTS] имя\_базы\_данных
- [спецификация\_create[,спецификация\_create]...]
- Команда CREATE DATABASE создает базу данных с указанным именем. Для использования команды необходимо иметь привилегию CREATE для базы данных. Если база данных с таким именем существует, генерируется ошибка.
- спецификация\_create:
  - [DEFAULT] CHARACTER SET имя\_набора\_символов
  - [DEFAULT] COLLATE имя\_порядка\_сопоставления

# Работа с таблицами

Таблица 3.1. Персональные данные

Имя	Возраст	Страна	e-mail
Михаил Петров	28	Россия	misha@yandex.ru
Джон Доусон	32	Австралия	j.dow@australia.com
Морис Дрюон	48	Франция	md@france.fr
Снежана	19	Болгария	sneg@bulgaria.com

# Команда SQL для создания такой таблицы выглядит следующим образом:

- CREATE TABLE employee\_data
- (- emp\_id int unsigned not null auto\_increment primary key,
- f\_name varchar(20),
- l\_name varchar(20),
- title varchar(30),
- age int,
- yos int,
- salary int,
- perks int,
- email varchar(60)
- );

- За ключевыми словами CREATE TABLE следует имя создаваемой таблицы employee\_data. Каждая строка внутри скобок представляет один столбец. Эти столбцы хранят для каждого сотрудника идентификационный номер ( emp\_id ), фамилию ( l\_name ), имя ( f\_name ), должность ( title ), возраст ( age ), стаж работы в компании ( yrs ), зарплату ( salary ), надбавки ( perks ), и адрес e-mail ( email ).

- За именем каждого столбца следует тип столбца. Типы столбцов определяют тип данных, которые будет содержать столбец. В данном примере столбцы `f_name`, `l_name`, `title` и `email` будут содержать текстовые строки, поэтому тип столбца задан как `varchar`, что означает переменное количество символов. Максимальное число символов для столбцов `varchar` определяется числом, заключенным в скобки, которое следует сразу за именем столбца. Столбцы `age`, `uos`, `salary` и `perks` будут содержать числа (целые), поэтому тип столбца задается как `int`. Первый столбец ( `emp_id` ) содержит идентификационный номер ( `id` ) сотрудника.

- `int`: определяет тип столбца как целое число.
- `unsigned`: определяет, что число будет без знака (положительное целое).
- `not null`: определяет, что значение не может быть `null` (пустым); то есть каждая строка в этом столбце должна иметь значение.
- `auto_increment`: когда MySQL встречается со столбцом с атрибутом `auto_increment`, то генерируется новое значение, которое на единицу больше, чем наибольшее значение в столбце. Поэтому мы не должны задавать для этого столбца значения, MySQL генерирует их самостоятельно. Из этого также следует, что каждое значение в этом столбце будет уникальным.
- `primary key`: помогает при индексировании столбца, что ускоряет поиск значений. Каждое значение должно быть уникально. Ключевой столбец необходим для того, чтобы исключить возможность совпадения данных. Например, два сотрудника могут иметь одно и то же имя, и тогда встанет проблема – как различать этих сотрудников, если не задать им уникальные идентификационные номера. Если имеется столбец с уникальными значениями, то можно легко различить две записи. Лучше всего поручить присваивание уникальных значений самой системе MySQL.