



БАЗЫ ДАННЫХ

Управление таблицами. Типы данных

Создание новой базы данных

Для того, чтобы создавать таблицы, необходимо соединиться с СУБД и создать базу данных (если она еще не существует).

```
CREATE DATABASE `dbname` ;
```

```
CREATE DATABASE IF NOT EXISTS `dbname` ;
```

Для создания новой БД необходимо иметь соответствующие привилегии!

Создание новой таблицы

Синтаксис:

```
CREATE TABLE [ IF NOT EXISTS ] `table_name` (  
    -- описание полей (столбцов) таблицы в формате:  
    -- `имя_поля` ТИП(длина) атрибуты  
);
```

Создание новой таблицы

```
CREATE TABLE IF NOT EXISTS `users` (  
  `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `login` VARCHAR(255) NOT NULL,  
  `balance` DECIMAL(10,2) NOT NULL DEFAULT 0.00,  
  `type` ENUM('ADMIN', 'USER') NOT NULL DEFAULT 'USER',  
  `created_at` TIMESTAMP,  
  PRIMARY KEY (`id`)  
);
```

Создание новой таблицы

При создании таблицы нужно следовать некоторым общепринятым правилам:

1. Имя таблицы должно быть на латинице, не содержать пробелов и спецсимволов.
2. Имя таблицы должно быть во множественном числе, что отражает суть помещаемых в нее данных: `user` – один пользователь, `users` – список пользователей.
3. Первым столбцом должен быть первичный ключ (Primary key), обычно всегда с именем `id`.

Атрибуты полей

- **UNSIGNED** – беззнаковое целое (в такой столбец невозможно поместить отрицательные значения)
- **NOT NULL** – значение не может быть NULL
- **DEFAULT** – значение по умолчанию
- **AUTO_INCREMENT** – автоинкремент, поле увеличивается на единицу автоматически (такое поле должно быть первичным ключом)

Индексы

- `PRIMARY KEY (`col`)` – первичный ключ, может быть автоинкрементом, уникальное поле
- `UNIQUE `name` (`col`)` – уникальное поле
- `INDEX `name` (`col`)` – простой поисковый индекс

Для первичного ключа его имя всегда `PRIMARY KEY`, для остальных индексов необходимо задать имя (как правило оно дублирует имя поля).

Типы данных

- Числовые
- Строковые
- Специальные типы
- Дата и время
- Пространственные типы

Числовые типы данных

Тип данных	Диапазон значений				Размер
	SIGNED		UNSIGNED		
TINYINT	-128	127	0	255	1 байт
SMALLINT	-32 768	32 767	0	65 535	2 байта
MEDIUMINT	-8 388 608	8 388 607	0	16 777 215	3 байта
INT	-2 147 483 648	2 147 483 647	0	4 294 967 295	4 байта
BIGINT	-9 квинт	9 квинт	0	18 квинт	8 байт
FLOAT	До 7 знаков после запятой, приблизительное значение				4 байта
DOUBLE	До 15 знаков после запятой, приблизительное значение				8 байт
DECIMAL	До 65 знакомест, точное значение				2–67 байт
BIT	До 64 знакомест				1–4 байт

Числовые типы данных

***INT** -- без указания длины

***INT**(N) -- с указанием длины

TINYINT -- от -128 до 127

TINYINT(1) UNSIGNED -- от 0 до 9

SMALLINT(3) NOT NULL -- от -999 до 999, не NULL

MEDIUMINT(5) UNSIGNED -- от 0 до 99999

INT(8) UNSIGNED NOT NULL -- от 0 до 99999999, не NULL

BIGINT

Числовые типы данных

FLOAT, **DOUBLE** -- без указания точности

FLOAT(P), **DOUBLE(P)** -- с указанием точности

DECIMAL(N) -- с указанием длины, 0 знаков после запятой

DECIMAL(N, P) -- с указанием общей длины и кол-ва знаков

DECIMAL(5, 2) -- от -999.99 до 999.99

DECIMAL(3, 3) -- от -0.999 до 0.999

Строковые типы данных

Тип данных	Максимальная длина	
CHAR	255 символов	
VARCHAR	255 символов	(до 65 536 символов)
TINYTEXT, TINYBLOB	255 символов	
TEXT, BLOB	65 536 (64Kб)	
MEDIUMTEXT, MEDIUMBLOB	16 Мб	
LONGTEXT, LONGBLOB	4 Гб	

*TEXT – текстовые данные

*BLOB – бинарные данные

Строковые типы данных

CHAR(10) — всегда 10 байт, даже если строка меньше

'HELLO' — хранится как 'HELLO•••••' (10 байт)

VARCHAR(10) — максимально — 10 байт

'HELLO' — хранится как 'HELLO' (5 байт)

Специальные типы данных

Тип данных	Значение	
ENUM	Список значений с одним выбором	65 535 значений
SET	Список значений с множественным выбором	65 535 значений
JSON	Документ в JSON-формате	до 4 Гб

Типы данных для даты и времени

Тип данных	Формат	Диапазон значений	Размер
DATE	YYYY-MM-DD	1000 – 9999 год	3 байта
TIME	HH:MM:SS	-838:59:59 – 839:59:59	3 байта
DATETIME	YYYY-MM-DD HH:MM:SS	1000 – 9999 год	8 байт
TIMESTAMP	YYYY-MM-DD HH:MM:SS (число секунд с 1970 года)	1970 – 2038 год	4 байта
YEAR	YYYY	1901 – 2155 год	1 байт

Пространственные типы данных

Тип данных	Значение
GEOMETRY	Произвольная геометрия
POINT	Точка
POLYGON	Полигон
LINESTRING	Линия
GEOMETRYCOLLECTION	Коллекция геометрий
MULTIPOINT	Коллекция точек
MULTIPOLYGON	Коллекция полигонов
MULTILINESTRING	Коллекция линий

Вставка данных в таблицу

Синтаксис:

```
INSERT INTO `table_name` (список_полей)  
VALUES (список_значений);
```

```
INSERT INTO `table_name` VALUES (список_значений);
```

```
INSERT INTO `table_name`  
SET поле = значение, поле = значение, ...;
```

Вставка данных в таблицу

Вставка нескольких строк за одну команду:

```
INSERT INTO `table_name` (список_полей) VALUES  
(список_значений_для_строки_1),  
(список_значений_для_строки_2),  
...;
```

```
INSERT INTO `table_name` VALUES  
(список_значений_для_строки_1),  
(список_значений_для_строки_2),  
...;
```

Вставка данных в таблицу

Вставка данных в таблицу **без указания** списка полей означает, что список значений полей для каждой строки **должен** быть именно в том **количестве** и **порядке**, в котором указаны поля заданы **при создании** таблицы.

На практике, лучше **не** полагаться на количество и порядок, а указывать их в явном виде. Кроме того, в таком случае мы можем опустить некоторые поля.

Вставка данных в таблицу

```
INSERT INTO `users` (`login`, `created_at`)  
VALUES ('vasya', NOW());
```

```
INSERT INTO `users` (`login`, `type`, `created_at`)  
VALUES ('admin', 'ADMIN', NOW());
```

Вставка данных в таблицу

```
INSERT INTO `users` (`login`, `balance`, `created_at`)
VALUES
('petya', 100.00, NOW()),
('kolya', 50.00, NOW()),
('masha', 150.00, NOW());
```

```
INSERT INTO `users` SET
    `login` = 'petya',
    `type` = 'USER',
    `balance` = 100.00,
    `created_at` = NOW();
```