

Создание простейших таблиц

1) Создайте таблицу **orders** с полем **id** типа **INT**, полем **state** для хранения статуса заказа и полем **amount** для хранения суммы заказа. Статус заказа умещается в строку в 10 символов, а сумма заказа имеет целочисленный тип данных.

```
CREATE TABLE orders (id int, state varchar(10), amount int);
```

2) Создайте таблицу **users** с полем **id** типа **INT** и двумя строковыми полями, которые будут хранить имя (**first_name**) и фамилию (**last_name**). Длина имени не превышает 20 символов, а фамилии 50 символов. Также добавьте в таблицу поле **birthday** типа **DATE**.

Заполните таблицу тремя пользователями:

1. Дмитрий Иванов, 12 августа 1995
2. Светлана Демчук, 8 июля 1993
3. Денис Антонов, 23 декабря 1996

```
CREATE TABLE users (  
    id INT,  
    first_name VARCHAR(20),  
    last_name VARCHAR(50),  
    birthday DATE  
);  
INSERT INTO users (id, first_name, last_name, birthday)  
VALUES  
    (1, 'Дмитрий', 'Иванов', '1995-08-12'),  
    (2, 'Светлана', 'Демчук', '1993-07-08'),  
    (3, 'Денис', 'Антонов', '1996-12-23');
```

3) Создайте таблицу **messages** с со следующими полями:

1. **id** типа **INT**;
2. **subject** типа **VARCHAR** длиной 100 символов
3. **message** тип **TEXT**;
4. **add_date** типа **DATETIME**;
5. **is_public** логического типа.

Добавьте в таблицу сообщение с темой «Первое сообщение» и текстом «Это мое первое сообщение!». Дату установите 12 декабря 2016 года 14 часов, 16 минут. Сообщение должно быть публичным.

```
CREATE TABLE messages (  
    id INT,  
    subject VARCHAR(100),  
    message TEXT,  
    add_date DATETIME,  
    is_public BOOLEAN  
);  
INSERT INTO messages (id, subject, message, add_date, is_public)  
VALUES  
    (1, 'Первое сообщение', 'Это мое первое сообщение!', '2016-12-12  
14:16:00', 1);
```

4)Создайте таблицу **rating**

1. **id** типа **INT**;
2. **car_id** типа **INT**;
3. **user_id** типа **INT**;
4. **rating** типа **FLOAT**.

Добавьте в неё 4 записи так, чтобы получалась таблица ниже:

rating			
id	car_id	user_id	rating
1	1	1	4.54
2	1	2	3.34
3	2	3	4.19
4	2	11	1.12

```
CREATE TABLE rating (  
    id INT,  
    car_id INT,  
    user_id INT,  
    rating FLOAT  
);  
INSERT INTO rating(id,car_id, user_id, rating)  
VALUES (1,1,1,4.54),  
(2,1,2,3.34),  
(3,2,3,4.19),  
(4,2,11,1.12);
```

Числовые поля

1) Создайте таблицу **products** для хранения информации о товарах в магазине. Выберите оптимальные поля для хранения данных в соответствии с условиями:

1. **id** типа INT – только положительные числа;
2. **name** – символьный тип до 100 символов (включительно);
3. **count** – количество товара на складе (до 200 штук), целое положительное;
4. **price** – цена в рублях без копеек (не более 1 млн рублей).

Заполните таблицу тремя товарами:

- Холодильник, 10 штук, 50 000 рублей.
- Стиральная машина, 0 штук, 23 570 рублей.
- Утюг, 3 штуки, 7300 рублей

```
CREATE TABLE products
(id INT UNSIGNED,
name VARCHAR (100),
count TINYINT UNSIGNED,
price MEDIUMINT UNSIGNED);
INSERT INTO products (name, count, price)
VALUES
('Холодильник', 10, 50000),
('Стиральная машина', 0, 23570),
('Утюг', 3, 7300);
```

2) Создайте таблицу **orders** для хранения заказов в магазине. Выберите оптимальные поля для хранения данных в соответствии с условиями:

1. **id** – тип INT. Только положительные числа.
2. **product_id** – для хранения номера товара. Только положительные числа от 0 до 4294967295.
3. **sale** – скидка. Целое положительное число от 0 до 100.
4. **amount** – сумма заказа. Денежный тип. Максимальная сумма заказа 999999.99 рублей.

Добавьте три записи так, чтобы получалась таблица ниже:

orders			
id	product_id	sale	amount
1	245	0	230.50
2	17	15	999999.99
3	145677	21	1240.00

```
CREATE TABLE orders
(id INT UNSIGNED,
product_id INT UNSIGNED,
sale TINYINT UNSIGNED,
amount DECIMAL (8,2));
INSERT INTO orders VALUES
(1, 245, 0, 230.50),
(2, 17, 15, 999999.99),
(3, 145677, 21, 1240.00);
```

3) Создайте таблицу **films** с информацией о фильмах. Выберите оптимальные поля для хранения данных в соответствии с условиями:

1. **id** типа INT, только положительные числа.
2. **name** – символьное поле длиной 100.
3. **rating** – рейтинг, вещественное число. Принимает положительные значения от 0 до 10.
4. **country** – страна фильма. Символьное поле, содержащее ровно 2 символа.

Добавьте в неё 3 записи так, чтобы получалась таблица ниже:

id	name	rating	country
1	Большая буря	3.45	RU
2	Игра	7.5714	US
3	Война	10.0	RU

```
CREATE TABLE films
(id INT UNSIGNED,
name VARCHAR (100),
rating FLOAT UNSIGNED,
country CHAR(2));
INSERT INTO films
VALUES
(1, 'Большая буря', 3.45, 'RU'),
(2, '', 7.5714, 'US'),
(3, '', 10.0, 'RU');
```

4) Создайте таблицу **files** для хранения информации о файлах. Выберите оптимальные поля исходя из условий ниже:

1. **id** — типа INT, только положительные числа.
2. **filename** — текстовое поле длиной 255 символов для хранения имени файла.
3. **size** — целочисленное поле для хранения размера файла в байтах. Только положительные числа. Могут храниться данные до 100 Гб.
4. **filetype** — поле для хранения типа файла, строка до 3 символов.

Добавьте 3 записи так, чтобы получалась таблица ниже:

files			
id	filename	size	filetype
1	big_archive.zip	81604378624	zip
2	movie_37.mp4	7838315315	mp4
3	music007.mp3	5242880	mp3

```
CREATE TABLE files
(id INT UNSIGNED,
filename VARCHAR (255),
size BIGINT,
filetype VARCHAR (3));
INSERT INTO files
VALUES
(1, 'big_archive.zip', 81604378624, 'zip'),
(2, 'movie_37.mp4', 7838315315, 'mp4'),
(3, 'music007.mp3', 5242880, 'mp3');
```

Строковые поля

1) Создайте таблицу **users** , в которой будут следующие поля:

1. **id** — идентификатор, целые положительные числа.
2. **first_name**— имя, строки до 50 символов.
3. **last_name** — фамилия, строки до 60 символов.
4. **bio** — биография, текст до 65000 символов.

Добавьте 3 записи так, чтобы получалась таблица ниже:

users			
id	first_name	last_name	bio
1	Антон	Кулик	С отличием окончил 39 лицей.
2	Сергей	Давыдов	
3	Дмитрий	Соколов	Профессиональный программист.

```
CREATE TABLE users (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  first_name VARCHAR(50),  
  last_name VARCHAR(60),  
  bio TEXT(65000)  
);  
INSERT INTO users(id,first_name, last_name, bio)  
VALUES (1,"Антон", "Кулик", "С отличием закончил 39 лицей."),  
(2,"Сергей", "Давыдов", " "),  
(3,"Дмитрий", "Соколов", "Профессиональный программист.");
```


- 2) Создайте таблицу **books** для хранения данных о книгах. В таблице должны быть следующие поля:
1. **id** – идентификатор, целое положительное.
 2. **name** – название, строка длиной не более 100 символов.
 3. **description** – описание книги длиной не более 1000 символов.
 4. **isbn** – международный стандартный книжный номер. 10 или 13 знаков.

Добавьте 3 записи так, чтобы получалась таблица ниже:

books			
id	name	description	isbn
1	MySQL 5	Хорошая книга.	5941579284
2	Изучаем SQL	Полезная книга.	5932860510
3	Изучаем Python. 4-е издание	Подробная книга о Python.	9785932861592

```
CREATE TABLE books (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100),  
    description VARCHAR(1000),  
    isbn VARCHAR(13));  
INSERT INTO books (id, name, description, isbn )  
VALUES ( 1, 'MySQL 5', 'Хорошая книга.', '5941579284'),  
(2, 'Изучаем SQL', 'Полезная книга', '5932860510'),  
(3, 'Изучаем Python.4-е издание', 'Подробная книга о  
Python', '9785932861592');
```

3) Создайте таблицу **apartments** для хранения информации о квартирах на сайте жилищного комплекса. В таблице должны быть следующие поля:

1. **id** — идентификатор, целое положительное.
2. **image** — поле для хранения пути до изображения, строка не более 100 символов.
3. **price** — цена не более 99 999 999. Целое положительное число.
4. **square** — площадь квартиры. Не более 150 квадратных метров. Целое положительное число.

Добавьте 3 записи так, чтобы получалась таблица ниже:

id	image	price	square
1	/apartments/1/cover.jpg	5250000	90
2	/apartments/2/cover-3.jpg	7500000	103
3		2300000	56

```
CREATE TABLE apartments (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  image VARCHAR(100),  
  price INT UNSIGNED CHECK(price <= 99999999),  
  square INT UNSIGNED CHECK(square <= 150));  
INSERT INTO apartments (id, image, price, square)  
VALUES ( 1, '/apartments/1/cover.jpg', 5250000, 90),  
(2, '/apartments/2/cover-3.jpg', 7500000, 103),  
(3, '', 2300000, 56);
```

4) На сайте действует следующая система: сперва сотрудник заносит в таблицу URL-адрес фотографии в интернете, а после скрипт на языке Python скачивает файл из сети на жесткий диск и добавляет в эту же таблицу путь до него.

Создайте таблицу **files**, которая поможет сотруднику и скрипту выполнить свои задачи. Таблица должна содержать следующие поля:

1. **id** — целое положительное
2. **file_url** — url файла, строка длиной до 500 символов.
3. **file_path** — путь до файла на жестком диске, строка до 200 символов.
4. **size** — размер скачанного файла в байтах, целое положительное. Размер файла может достигать 100 Гб.

Добавьте 3 записи так, чтобы получалась таблица ниже:

files			
id	file_url	file_path	size
1	http://archives.com/archives/big_archive.zip	files/2018/02/archive.zip	81604378624
2	http://movies.com/movies/movie.mp4		0
3	http://movies.com/best-songs/song.mp3	files/2018/03/song.mp3	5242880

```
CREATE TABLE files (  
    id INT PRIMARY KEY,  
    file_url VARCHAR(500),  
    file_path VARCHAR(200),  
    size BIGINT  
);
```

```
INSERT INTO files (id, file_url, file_path, size) VALUES  
(1, "http://archives.com/archives/big_archive.zip",  
"files/2018/02/archive.zip", 81604378624),  
(2, 'http://movies.com/movies/movie.mp4', '', 0),  
(3, 'http://movies.com/best-songs/song.mp3',  
'files/2018/03/song.mp3', 5242880);
```

Дата и время

- 1) Создайте таблицу **users** для хранения информации о пользователях сайта. В таблице должны быть следующие поля:
 1. **id** – идентификатор, целое положительное;
 2. **email** – адрес электронной почты, строка не более 100 символов;
 3. **date_joined** – дата регистрации (достаточно хранить дату, без времени)
 4. **last_activity** – дата и время последней активности (с точностью до секунд).

Добавьте 3 записи, чтобы получалась таблица ниже:

users			
id	email	date_joined	last_activity
1	user1@domain.com	2014-12-12	2016-04-08 12:34:54
2	user2@domain.com	2014-12-12	2017-02-13 11:46:53
3	user3@domain.com	2014-12-13	2017-04-04 05:12:07

```
CREATE TABLE users
(id INT UNSIGNED,
email VARCHAR (100),
date_joined DATE,
last_activity DATETIME);
INSERT INTO users VALUES
(1, 'user1@domain.com', '2014-12-12', '2016-04-08 12:34:54'),
(2, 'user2@domain.com', '2014-12-12', '2017-02-13 11:46:53'),
(3, 'user3@domain.com', '2014-12-13', '2017-04-04 05:12:07');
```

2) Создайте таблицу **calendar** для хранения календаря посетителей.
В таблице должны быть следующие поля:

1. **id** – идентификатор записи в календаре, целое положительное;
2. **user_id** – идентификатор пользователя, целое (INT) положительное;
3. **doctor_id** – идентификатор доктора, целое (INT) положительное;
4. **visit_date** – дата и время визита (точность до секунд).

Добавьте 5 записей, чтобы получалась таблица ниже:

id	user_id	doctor_id	visit_date
1	1914	1	2017-04-08 12:00:00
2	12	1	2017-04-08 12:30:00
3	4641	2	2017-04-09 09:00:00
4	784	1	2017-04-08 13:00:00
5	15	2	2017-04-09 10:00:00

```
CREATE TABLE calendar
(id INT UNSIGNED,
user_id INT UNSIGNED,
doctor_id INT UNSIGNED,
visit_date DATETIME);
INSERT INTO calendar
VALUES
(1, 1914, 1, '2017-04-08 12:00:00'),
(2, 12, 1, '2017-04-08 12:30:00'),
(3, 4641, 2, '2017-04-09 09:00:00'),
(4, 784, 1, '2017-04-08 13:00:00'),
(5, 15, 2, '2017-04-09 10:00:00');
```

3) Создайте таблицу **temperature** для хранения записей о температуре. Добавьте в неё следующие поля:

1. **id** – идентификатор, целые положительные числа;
2. **city_id** – идентификатор города, целые положительные числа;
3. **temperature** – температура, целое число в диапазоне от -100 до 100 градусов.
4. **wind_speed** – целое положительное число для хранения скорости ветра.
Историческая максимальная скорость ветра была 113 м/с.
5. **mdate** – дата и время измерения (с точностью до секунд).

Добавьте 5 записей, чтобы получалась таблица ниже:

temperature				
id	city_id	temperature	wind_speed	mdate
1	456	17	7	2017-02-08 12:00:00
2	456	19	6	2017-02-08 12:10:00
3	456	20	6	2017-02-08 12:20:00
4	471	-7	12	2017-02-08 12:20:01
5	44	-43	17	2017-02-08 12:23:12

```
CREATE TABLE temperature
(id INT UNSIGNED,
city_id INT UNSIGNED,
temperature TINYINT,
wind_speed TINYINT UNSIGNED,
mdate DATETIME);
INSERT INTO temperature
VALUES
(1, 456, 17, 7, '2017-02-08 12:00:00'),
(2, 456, 19, 6, '2017-02-08 12:10:00'),
(3, 456, 20, 6, '2017-02-08 12:20:00'),
(4, 471, -7, 12, '2017-02-08 12:20:01'),
(5, 44, -43, 17, '2017-02-08 12:23:12');
```

- 4) Создайте таблицу **arrival** для хранения времени прихода сотрудников на работу. Добавьте в неё следующие поля:
1. **id** – идентификатор, целое положительное;
 2. **user_id** – идентификатор пользователя, целое положительное;
 3. **a_date** – дата прихода;
 4. **a_time** – время прихода.

Добавьте 4 записи, чтобы получалась таблица ниже:

id	user_id	a_date	a_time
1	10	2017-03-09	08:45:41
2	12	2017-03-09	08:46:12
3	7	2017-03-09	08:53:01
4	31	2017-03-09	09:01:17

```
CREATE TABLE arrival
(id INT UNSIGNED,
user_id INT UNSIGNED,
a_date DATE,
a_time TIME);
INSERT INTO arrival
VALUES
(1, 10, '2017-03-09', '08:45:41'),
(2, 12, '2017-03-09', '08:46:12'),
(3, 7, '2017-03-09', '08:53:01'),
(4, 31, '2017-03-09', '09:01:17');
```

5) Создайте таблицу **cars** для хранения списка автомобилей дилерского центра. Таблица должна содержать следующие поля:

1. **id** – идентификатор, целое положительное;
2. **mark** – марка автомобиля, строка до 20 символов;
3. **model** – модель автомобиля, строка до 20 символов;
4. **year** – год выпуска, используйте тип YEAR;
5. **mileage** – пробег, целое число от 0 до 100 000.

Добавьте 3 автомобиля, чтобы получалась таблица ниже:

id	mark	model	year	mileage
1	Toyota	Camry	2015	32000
2	Mazda	CX-5	2016	17000
3	Nissan	Patrol	2016	60000

```
create table cars
(id int unsigned,
 mark varchar(20),
 model varchar(20),
 year year,
 mileage mediumint unsigned);
insert into cars
VALUES
(1, 'Toyota', 'Camry', '2015', 32000),
(2, 'Mazda', 'CX-5', '2016', 17000),
(3, 'Nissan', 'Patrol', '2016', 60000);
```


NULL

1) Создайте таблицу **products** со следующими полями:

1. **id** — номер товара, целое положительное, не может принимать NULL.
2. **name** — название товара до 120 символов, не может принимать NULL.
3. **category_id** — категория товара, целое положительное, может принимать NULL.
4. **price** — цена, денежный тип из 10 знаков, из них два после запятой. Не может принимать NULL.

Добавьте 4 записи так, чтобы получалась таблица ниже:

products			
id	name	category_id	price
1	Подгузники (12 шт)	3	700.00
2	Подгузники (24 шт)	3	1250.00
3	Спиннер	NULL	250.40
4	Пюре слива	4	47.50

```
CREATE TABLE products (  
    id INT PRIMARY KEY,  
    name VARCHAR(120) NOT NULL,  
    category_id INT,  
    price DECIMAL(10, 2) NOT NULL  
);  
INSERT INTO products (id, name, category_id, price) VALUES  
(1, 'Подгузники(12 шт)', 3, 700.00),  
(2, 'Подгузники(24 шт)', 3, 1250.00),  
(3, 'Спиннер', NULL, 250.40),  
(4, 'Пюре слива', 4, 47.50);
```

2) Выберите из таблицы **products** название, количество и цены всех товаров, у которых нет категории.

Данные отсортируйте по цене.

```
SELECT name, COUNT(*) AS quantity, price
FROM products
WHERE category_id IS NULL
GROUP BY name, price
ORDER BY price;
```

Output:

name	quantity	price
Спиннер	1	250.40

BOOL, ENUM, SET

1) Создайте таблицу **articles** для хранения данных о статьях. В таблице должны быть следующие поля:

- **id** — идентификатор, целое положительное, NULL запрещен
- **name** — название статьи, строка до 80 символов.
- **text** — текст статьи.
- **state** — статус статьи. Поле из 3 вариантов: draft (черновик), correction (корректурa), public (опубликована).

Добавьте 3 записи так, чтобы получалась таблица ниже:

articles			
id	name	text	state
1	Новое в Python 3.6		draft
2	Оптимизация SQL запросов	При больших объемах данных ...	correction
3	Транзакции в MySQL	По долгу службы мне приходится ...	public

```
CREATE TABLE articles
(id int unsigned NOT NULL,
 name varchar (80),
 text text,
 state enum ('draft', 'correction', 'public'));
insert into articles
VALUES
(1, 'Новое в Python 3.6', ' ', 'draft'),
(2, 'Оптимизация SQL запросов', 'При больших объемах данных...',
'correction'),
(3, 'Транзакции в MySQL', 'По долгу службы мне приходится...', 'public');
```

2) Создайте таблицу **rooms** для хранения номеров в отеле:

1. **id** — идентификатор, целое положительное. NULL запрещен.
2. **number** — номер комнаты, целое положительное. Всего в отеле 107 комнат. NULL запрещен.
3. **beds** — количество спальных мест. Выбор из **1+1**, **2+1**, **2+2**. Можно выбрать только один вариант. NULL запрещен.
4. **additional** — дополнительные удобства в номере. Можно выбрать несколько вариантов из списка: **conditioner**, **bar**, **fridge** и **wifi**.

Добавьте 3 записи так, чтобы получалась таблица ниже:

rooms			
id	number	beds	additional
1	10	1+1	conditioner,bar,wifi
2	12	2+1	
3	24	2+2	fridge,bar,wifi

```
CREATE TABLE rooms
(id INT UNSIGNED NOT NULL,
number TINYINT UNSIGNED NOT NULL,
beds ENUM ('1+1', '2+1', '2+2') NOT NULL,
additional SET('conditioner', 'bar', 'fridge', 'wifi') NULL);
INSERT INTO rooms
VALUES
(1, 10, '1+1', 'conditioner,bar,wifi'),
(2, 12, '2+1', NULL),
(3, 24, '2+2', 'fridge,bar,wifi');
```

3) Создайте таблицу **orders** для хранения заказов в интернет-магазине:

1. **id** — идентификатор, целое положительное.
2. **user_id** — идентификатор пользователя, который оформил заказ. Целое положительное, NULL запрещен.
3. **amount** — стоимость заказа. DECIMAL из 10 цифр, 2 цифры после десятичной точки.
4. **created** — дата и время создания заказа. NULL запрещен.
5. **state** — статус заказа. Выбор из **new**, **cancelled**, **in_progress**, **delivered**, **completed**. Можно выбрать только один вариант.
6. **options** — дополнительные опции заказа. Можно выбрать несколько вариантов из списка: **pack**, **fitting**, **call** и **intercom**.

Добавьте 3 записи так, чтобы получалась таблица ниже:

orders					
id	user_id	amount	created	state	options
456	763	14299.00	2018-02-01 17:45:59	completed	pack,call
457	1987	249.50	2018-02-01 18:23:04	delivered	pack,intercom
459	78	2300.12	2018-02-01 22:12:09	in_progress	

```
CREATE TABLE orders
(id INT UNSIGNED NOT NULL,
user_id INT UNSIGNED NOT NULL,
amount DEC (10, 2),
created DATETIME NOT NULL,
state ENUM('new', 'cancelled', 'in_progress', 'delivered', 'completed'),
options SET('pack', 'fitting', 'call', 'intercom') NULL);
INSERT INTO orders
VALUES
(456, 763, 14299, '2018-02-01 17:45:59', 'completed', 'pack,call'),
(457, 1987, 249.5, '2018-02-01 18:23:04', 'delivered', 'pack,intercom'),
(459, 78, 2300.12, '2018-02-01 22:12:09', 'in_progress', NULL);
```

ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ

1) Создайте таблицу **orders** для хранения списка заказов:

1. **id** — идентификатор, целое положительное.
2. **user_id** — идентификатор пользователя, который оформил заказ. Целое положительное, NULL запрещен.
3. **amount** — стоимость заказа. Целое положительное число не более 1 млн. NULL запрещен, по умолчанию 0.
4. **created** — дата и время создания заказа. NULL запрещен.
5. **state** — статус заказа. Выбор из **new**, **cancelled**, **in_progress**, **delivered**, **completed**. Можно выбрать только один вариант. NULL запрещен. По умолчанию должен стоять **new**.

Добавьте 3 записи так, чтобы получалась таблица ниже:

orders				
id	user_id	amount	created	state
1	56	5400	2018-02-01 17:46:59	new
2	90	249	2018-02-01 19:13:04	new
3	78	2200	2018-02-01 22:43:09	new

```
CREATE TABLE orders
(id INT UNSIGNED NOT NULL,
user_id INT UNSIGNED NOT NULL,
amount MEDIUMINT(7) DEFAULT 0,
created DATETIME NOT NULL,
state ENUM('new', 'cancelled', 'in_progress', 'delivered', 'completed')
DEFAULT 'new');
```

```
INSERT INTO orders
VALUES
(1, 56, 5400, '2018-02-01 17:46:59', DEFAULT),
(2, 90, 249, '2018-02-01 19:13:04', DEFAULT),
(3, 78, 2200, '2018-02-01 22:43:09', DEFAULT);
```

2) Создайте таблицу **users** для хранения списка пользователей сайта:

1. **id** — идентификатор, целое положительное.
2. **first_name** — имя, строка до 20 символов. NULL запрещен.
3. **last_name** — фамилия, строка до 20 символов. NULL запрещен.
4. **patronymic** — отчество, строка до 20 символов. NULL запрещен, по умолчанию пустая строка.
5. **is_active** — отметка об активности пользователя. Логическое поле, по умолчанию TRUE.
6. **is_superuser** — отметка администратора. Логическое поле, по умолчанию FALSE.

Добавьте 3 записи так, чтобы получалась таблица ниже:

users					
id	first_name	last_name	patronymic	is_active	is_superuser
1	Дмитрий	Иванов		TRUE	FALSE
2	Анатолий	Белый	Сергеевич	TRUE	TRUE
3	Андрей	Крючков		FALSE	FALSE

```
CREATE TABLE users
(id INT UNSIGNED NOT NULL,
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR (20) NOT NULL,
patronymic VARCHAR (20) NOT NULL DEFAULT '',
is_active BOOL DEFAULT TRUE,
is_superuser BOOL DEFAULT FALSE);
INSERT INTO users
VALUES
(1, 'Дмитрий', 'Иванов', DEFAULT, DEFAULT, DEFAULT),
(2, 'Анатолий', 'Белый', 'Сергеевич', DEFAULT, TRUE),
(3, 'Андрей', 'Крючков', DEFAULT, FALSE, DEFAULT);
```

3) Создайте таблицу **logs** для хранения логов посещения сайта:

1. **date** — время посещения в формате DATETIME, NULL запрещен, по умолчанию должно проставляться время создания записи.
2. **browser** — название браузера, строка до 500 символов. NULL запрещен, по умолчанию пустая строка.
3. **is_bot** — отметка о том является ли посетитель ботом или нет. Логический тип, по умолчанию FALSE. NULL запрещен.

Добавьте 3 записи так, чтобы получалась таблица ниже:

logs		
date	browser	is_bot
2018-03-19 19:50:01	Chrome 64.0.1.417	FALSE
2018-03-19 19:55:11	Firefox 55 (yandex bot)	True
2018-03-19 19:56:12	Chrome 63.0.0.471	FALSE

```
CREATE TABLE logs (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    browser VARCHAR(500) NOT NULL DEFAULT '',  
    is_bot BOOLEAN NOT NULL DEFAULT FALSE  
);  
INSERT INTO logs (date, browser, is_bot) VALUES  
(NOW(), "Chrome 64.0.1.417", FALSE),  
(NOW(), "Firefox 55(yandex bot", True),  
(NOW(), "Chrome 63.0.0.417", FALSE);
```


4) Создайте таблицу **reviews** с отзывами о компании Shultais Education:

1. **id** — идентификатор, целое положительное.
2. **user_id** — идентификатор пользователя, который оставил отзыв. Целое положительное, NULL запрещен.
3. **date** — дата и время оставления отзыва, NULL запрещен, по умолчанию должно проставляться время создания записи.
4. **course** — курс, для которого оставлен отзыв. Выбор из **python**, **sql**, **all**. Можно выбрать только один вариант. NULL запрещен. Значение по умолчанию — **all**.
5. **text** — текст отзыва, поле типа TEXT. NULL запрещен.
6. **public** — отметка о том является ли отзыв публичным. Логический тип, по умолчанию FALSE. NULL запрещен.

Добавьте 3 записи так, чтобы получалась таблица ниже:

reviews						
id	user_id	date	course	text	public	
1	817	2018-01-11 19:50:01	python	Это прекрасная возможность получить новые очки в программировании. Доступное объяснение позволяет без проблем изучить желаемый материал	TRUE	
2	1289	2018-02-16 08:55:11	python	Проект на мой взгляд отличный: 1. Короткие видеоролики без воды 2. Интересные задачи и практика 3. Очень быстрая обратная связь	TRUE	
3	2914	2018-03-19 12:56:12	all	Хорошая затея. Но проект ещё нуждается в развитии.	TRUE	

```
CREATE TABLE reviews (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    course ENUM('python', 'sql', 'all') NOT NULL DEFAULT 'all',  
    text TEXT NOT NULL,  
    public BOOLEAN NOT NULL DEFAULT FALSE  
);
```

```
INSERT INTO reviews (id, user_id, date, course, text, public) VALUES  
(1, 817, NOW(), 'python', 'Это прекрасная возможность получить новые  
очки в программировании. Доступное объяснение позволяет без проблем  
изучить желаемый материал.', TRUE),  
(2, 1289, NOW(), 'python', 'Проект на мой взгляд отличный: 1. Короткие  
видеоролики без воды 2. Интересные задачи и практики. 3. Очень  
быстрая обратная связь', FALSE),  
(3, 2914, NOW(), 'all', 'Хорошая затея, но проект еще нуждается в  
развитии ', TRUE);
```

