

Обычные индексы

3) Создайте таблицу **orders** со следующими полями:

1. **id** — автоинкрементальный первичный ключ.
2. **user_id** — идентификатор пользователя, беззнаковый INT, NULL запрещен.
3. **state** — статус заказа, строка до 8 символов. По умолчанию **new**. NULL запрещен.
4. **amount** — целое положительное для хранения чисел не более 1 млн. NULL запрещен. По умолчанию **0**.

Создайте два обычных индекса для полей **user_id** и **state**. Имена индексов должны совпадать с именами столбцов.

```
CREATE TABLE orders (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT UNSIGNED NOT NULL,  
    state VARCHAR(8) NOT NULL DEFAULT 'new',  
    amount INT NOT NULL DEFAULT 0,  
    INDEX user_id (user_id),  
    INDEX state (state)  
);
```

4) Создайте таблицу **products** для хранения товаров в интернет магазине:

1. **id** — первичный ключ.
2. **category_id** — категория, целое положительное. Может принимать NULL. По умолчанию NULL.
3. **name** — название, строка до 100 символов. NULL запрещен.
4. **count** — количество, целое положительное до 255. NULL запрещен, по умолчанию 0.
5. **price** — цена типа DECIMAL с 10 знаками, 2 из которых выделены для копеек. NULL запрещен, по умолчанию 0.00.

Добавьте в таблицу два простых индекса так, чтобы ускорить поиск по категории и по цене.

Названия индексов должны быть такими же, как и названия полей, по которым эти индексы строятся.

```
CREATE TABLE products (  
    id INT PRIMARY KEY,  
    category_id INT,  
    name VARCHAR(100) NOT NULL,  
    count TINYINT UNSIGNED NOT NULL DEFAULT 0,  
    price DECIMAL(10,2) NOT NULL DEFAULT 0.00,  
    INDEX category_id (category_id),  
    INDEX price (price)  
);
```

Удаление индексов

- 3) Разработчик создал таблицу **passports** для хранения паспортов с помощью следующего запроса:

```
CREATE TABLE passports (  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  user_id INT UNSIGNED NOT NULL,  
  series VARCHAR(4) NOT NULL,  
  number VARCHAR(6) NOT NULL,  
  state ENUM('active','expired') NOT NULL DEFAULT 'active',  
  UNIQUE KEY series (series),  
  UNIQUE KEY number (number)  
)
```

В процессе эксплуатации базы данных выяснилось, что в таблицу нельзя добавить две записи с одинаковой серией или с одинаковым номером паспорта. Хотя планировали, чтобы уникальность была только для совместных данных в **series** и **number**.

Удалите неправильные индексы и создайте индекс с именем **series_number**, который удовлетворяет условию выше.

-- Удаляем неправильные индексы

```
ALTER TABLE passports  
DROP INDEX series,  
DROP INDEX number;
```

-- Создаем новый индекс series_number

```
CREATE UNIQUE INDEX series_number ON passports (series, number);
```

- 4) Удалите из таблицы **products** индексы, которые по вашему мнению установлены неправильно. Таблица создана с помощью следующего SQL запроса:

```
CREATE TABLE products (  
  id INT UNSIGNED NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  category_id INT UNSIGNED NULL DEFAULT NULL,  
  name VARCHAR(100) NOT NULL,  
  count TINYINT UNSIGNED NOT NULL DEFAULT 0,  
  price DECIMAL(10,2) NOT NULL DEFAULT 0.00,  
  UNIQUE INDEX category_id (category_id),  
  INDEX count (count),  
  INDEX main_search (category_id, price)  
);
```

```
ALTER TABLE products  
DROP INDEX category_id,  
DROP INDEX count;
```

Первичный ключ

3) Создайте таблицу **passports** со следующими полями:

1. **series** — символьного типа длиной до 4 символов. NULL запрещен.
2. **number** — символьного типа длиной до 6 символов. NULL запрещен.
3. **user_id** — беззнаковый INT. NULL запрещен.
4. **date_issue** — дата типа DATE

Создайте первичный ключ по двум полям: **series** и **number**.

Добавьте 4 записи так, чтобы получалась таблица ниже:

series	number	user_id	date_issue
3206	147345	15	2006-08-12
3216	147345	234	2016-09-23
2405	147345	1	2015-01-07
5411	147345	15	2008-03-03

```
CREATE TABLE passports (  
    series VARCHAR(4) NOT NULL,  
    number VARCHAR(6) NOT NULL,  
    user_id INT UNSIGNED NOT NULL,  
    date_issue DATE,  
    PRIMARY KEY (series, number)  
);  
  
INSERT INTO passports (series, number, user_id, date_issue)  
VALUES  
    ('3206', '147345', 15, '2006-08-12'),  
    ('3216', '147345', 234, '2016-09-23'),  
    ('2405', '147345', 1, '2015-01-07'),  
    ('5411', '147345', 15, '2008-03-03');
```

- 4) Создайте таблицу **files** для хранения данных о цифровых фильмах. В таблице должны быть следующие поля:
1. **id** - идентификатор.
 2. **film_id** — ссылка на фильм, беззнаковое целое, NULL запрещен. *Ссылки на другие поля должны быть того же типа, что и сами поля. То есть **film_id** должно быть того же типа, что и **id** (в другой таблице).*
 3. **size** — размер файла в байтах. В магазине могут быть фильмы более 100 Гб. надо запретить добавлять в таблицу файлы отрицательного размера.
 4. **filename** — название файла, символьное поле длиной 100 символов.

Добавьте 3 записи так, чтобы получалась таблица ниже:

files			
id	film_id	size	filename
1	356	28668906700	silicon_valley_s02_1080p.zip
2	4514	2684354560	dunkirk.mp4
3	87145	734003200	milk.mp4

```
CREATE TABLE files (  
    id INT UNSIGNED NOT NULL,  
    film_id INT UNSIGNED NOT NULL,  
    size BIGINT UNSIGNED NOT NULL CHECK (size >= 0),  
    filename VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id)  
);  
  
INSERT INTO files (id, film_id, size, filename)  
VALUES  
    (1, 356, 28668906700, 'silicon_valley_s02_1080p.zip'),  
    (2, 4514, 2684354560, 'dunkirk.mp4'),  
    (3, 87145, 734003200, 'milk.mp4');
```

Уникальный индекс

3) Создайте таблицу **posts** для хранения постов пользователей на сайте:

1. **id** — автоинкрементальный первичный ключ.
2. **user_id** — идентификатор пользователя, беззнаковый INT, NULL запрещен.
3. **name** — название поста, строка до 100 символов, NULL запрещен.
4. **pub_date** — дата и время публикации. По умолчанию NULL (означает, что пост еще не опубликован).
5. **slug** — идентификатор поста для генерации URL, строка до 50 символов, NULL запрещен.

В рамках всего сайта **slug** могут повторяться, но для каждого конкретного пользователя значения **slug** уникальны. Создайте подходящий под это условие ключ с именем **uslug**.

```
CREATE TABLE posts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT UNSIGNED NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    pub_date DATETIME DEFAULT NULL,  
    slug VARCHAR(50) NOT NULL,  
    UNIQUE INDEX uslug (user_id, slug)  
);
```

4) Создайте таблицу **products** для хранения данных о товарах в магазине:

1. **id** — автоинкрементальный первичный ключ.
2. **category_id** — идентификатор категории, беззнаковый INT, по умолчанию NULL.
3. **name** — название товара, строка до 100 символов, NULL запрещен.
4. **slug** — идентификатор товара для генерации URL, строка до 50 символов, NULL запрещен.
5. **ean13** — штрих-код из 13 символов. Только уникальные значения. NULL запрещен.

В рамках всего сайта **slug** могут повторяться, но для каждой конкретной категории значения **slug** уникальны. Создайте подходящий под это условие ключ с именем **category_slug**.

```
CREATE TABLE products (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    category_id INT UNSIGNED DEFAULT NULL,  
    name VARCHAR(100) NOT NULL,  
    slug VARCHAR(50) NOT NULL,  
    ean13 CHAR(13) NOT NULL UNIQUE,  
    UNIQUE INDEX category_slug (category_id, slug)  
);
```