

Задание 2

Какой тип связи/отношения между таблицами **users** и **ratings**?
Между таблицами **items**, **ratings**?

Таблицы **users** и **ratings** связаны посредством связи "один ко многим". Т.е. одному пользователю может принадлежать несколько отзывов, но при этом у каждого отзыва лишь один автор-пользователь.

Таблицы **items** и **ratings** тоже связаны как "один ко многим", поскольку у одного товара может быть несколько отзывов, но конкретный отзыв относится к одному товару.

Задание 3

К какой/каким из созданных таблиц вы бы предложили создать индекс? Объясните, почему.

Первичные ключи позволяют оптимизировать работу с таблицей за счет более быстрого обращения к искомым данным (проходимся по отдельной базе, где хранятся индексы, и переходим по закрепленным к индексам ссылкам. Иначе пришлось бы пройти по каждой строке в исходной таблице с целью найти необходимые данные, что более затратно по времени).

В первом задании при создании таблицы **ratings** я добавила определение первичного ключа **primary key(user_id, item_id)**. Это связано с тем, что, во-первых, в реляционных БД, по-хорошему, следует иметь первичные ключи для каждой из таблиц, во-вторых, т.к. для каждого отзыва возможен лишь один товар, для которого этот отзыв оставлен, и один автор данного отзыва. Для **primary key** по умолчанию создаются индексы. Суммируя все вышесказанное, я бы создала индексы для таблицы **ratings** на основе полей **user_id** и **item_id** с целью быстрее находить в таблице **ratings** все отзывы конкретного пользователя или отзывы по конкретному товару, что является очень популярными запросами со стороны клиентов.

Задание 4

Есть две таблицы в базе, созданные следующим образом:

```
CREATE TABLE Car_owner  
(INN CHAR(10) PRIMARY KEY,  
  Name VARCHAR(250) NOT NULL  
);
```

```
CREATE TABLE Car (
    Owner CHAR(10) REFERENCES Car_owner(INN),
    CurN CHAR(6),
    Region INTEGER,
    Brand VARCHAR(50) NOT NULL,
    Color VARCHAR(50) NOT NULL,
    Power INTEGER NOT NULL CHECK (Power>50),
    CarYear INTEGER NOT NULL,
    Mileage INTEGER,
    PRIMARY KEY(CurN, Region)
);
```

В таблицу последовательно (!!!!) добавляются следующие записи. “Последовательно” означает, что сначала выполняется команда 1, затем команда 2 и т.д.

Отметьте те команды, которые смогут добавить записи в таблицу. Объясните, почему.

```
-- 1
INSERT INTO Car VALUES ('7984672834', 'E340BT, 77, Lada
Granta', 'Красный', 87, 2017, 35)
```

- не хватает апострофов слева и справа от числа 77. Поэтому будет ошибка, т.к. CurN принимает строку только из 6 символов. Да и заполнение данных окажется сдвинутым и тоже приведет к ошибке.

Помимо этого существует связь между данной таблицей и таблицей Car_owner через primary key. Номера 7984672834 пока нет в таблице Car_owner, по этой причине код также бы не сработал.

```
-- 2
INSERT INTO Car_owner VALUES ('7984672834', 'Иван Петров')
```

- Все верно, код выполнится.

```
-- 3
INSERT INTO Car_ownerCar_owner VALUES ('7984672834', 'Татьяна
Иванова')
```

- Таблицы с названием Car_ownerCar_owner не существует. Код не выполнится. Но, если опечатку убрать, то тоже не сработает, так как ключ с номером 7984672834 уже существует у Ивана Петрова.

```
-- 4
INSERT INTO Car Owner VALUES ('4752909757', 'Иван Петров')
```

- В названии таблицы пропущено нижнее подчеркивание. Код не работает. При исправлении данной ошибки будет работать верно.

```
-- 5
INSERT INTO Car VALUES ('6239572784', 'E340BT, 77, 'Volkswagen
Polo', 'Синий', 105, 2018, 40)
```

- Снова пропущен апостроф. В этот раз после E340BT. Ошибка.

```
-- 6
INSERT INTO Car VALUES ('4752909757', 'A822EY', 99, 'Skoda
Rapid', 'черный', 125, 2021, 35)
```

- Не работает, так как такого первичного ключа 4752909757 еще не существует в таблице Car_owner.

```
-- 7
INSERT INTO Car VALUES ('7984672834', 'A822EY', 99, 'Hyundai
Solaris', 'черный', 123, 2019, 20)
```

- Будет работать. Это машина Ивана Петрова с INN 7984672834

```
-- 8
INSERT INTO Car VALUES ('74478679847', '8971HP, 199, 'Kia
Sportage', 'белый', 18, 2017, 35)
```

- Снова опечатка в апострофах. Не работает

```
-- 9
INSERT INTO Car VALUES ('7984672834', 'E340BT, 77, 'Toyota
RAV4', 'Серебристо-серый, 146, 2019)
```

- Снова опечатка в апострофах. Не работает

```
-- 10
INSERT INTO Car VALUES ('7984672834', 'H454EE', 98, 'Skoda
Rapid', 'черный', 45, 2021, 0)
```

- Не работает, поскольку на параметр Power стоит CHECK (Power>50), а здесь Power = 45.

Задание 5

Можно ли использовать данный скрипт для шардинга таблицы на 32 документа? И генерацией из этой таблицы 16 тестовых таблиц примерно по 2 документа test.docs00, 01, 02, ..., 15

```
INSERT INTO docs00
SELECT * FROM documents WHERE (id%16)=0
...
INSERT INTO docs15
SELECT * FROM documents WHERE (id%16)=15
```

Да, можно. У нас условие разбиения - остаток от деления на 16, а их как раз 16 штук, что дает возможность шардирования на 16 тестовых таблиц. Однако, нельзя утверждать, что в тестовых таблицах будет ровно по 2 документа, так как кол-во документов в каждой таблице зависит от набора номеров id. Т.е. может быть ситуация, что в одной из таблиц 5 документов, а в каких-то ни одного. Но в среднем случае, все же, в таблицах будет примерно по 2 документа :)