ДОМАШНЕЕ ЗАДАНИЕ 3 ПО SQL

Задание 1

1. 1. Создайте 3 таблицы: users и items, ratings.

Таблица users содержит информацию о пользователях, совершающих заказы на маркетплейсе. Таблица items содержит информацию о товарах маркетплейса. Таблица ratings содержит информацию об отзывах на товары маркетплейса.

В таблице users создайте колонки `user_id`, `birth_date`, `sex`, `age`. В таблице orders создайте колонки `item_id`, `description`, `price`, `category`. В таблице ratings создайте колонки `item_id`, `user_id`, `review` (отзыв пользователя на товар), `rating` (рейтинг товара, поставленный пользователем).

Реализовано в файле .sql

1. 2. Обоснование выбора типов данных:

В таблице users выбраны следующие типы данных:

- user_id INT: целое число для идентификации пользователей,
- birth_date DATE: дата рождения пользователя,
- sex CHAR(1): символ для обозначения пола (например, 'M' для мужского пола, 'F' для женского),
- age INT: целое число для хранения возраста пользователя.

В таблице items выбраны следующие типы данных:

- item_id INT: целое число для идентификации товаров,
- description VARCHAR(255): строка переменной длины для описания товара,

- price DECIMAL(10, 2): десятичное число с общим количеством цифр 10 и 2 знаками после запятой для хранения цены товара,
- category VARCHAR(50): строка переменной длины для указания категории товара.

В таблице ratings выбраны следующие типы данных:
- item_id INT: целое число для идентификации товара,
- user_id INT: целое число для идентификации пользователя,
- review VARCHAR(500): строка переменной длины для хранения отзыва пользователя,
- rating INT: целое число для хранения рейтинга товара.

1. 3. Как связаны таблицы.

Таблицы связаны с помощью внешних ключей item_id и user_id в таблице ratings, которые ссылается на соответствующие первичные ключи в таблицах items и users соответственно.

Дополнительные столбцы в таблицах можно добавить в соответствии с дополнительными требованиями или необходимостью хранить дополнительную информацию.

1. 4. Добавление записей в таблицу.

Добавьте несколько значений (~20) в каждую таблицу. Попробуйте сгенерировать значения для наполнения таблиц (а не прописывать их каждый раз вручную)

Реализовано в файле .sql

Задание 2

Тип связи/отношения между таблицами users и ratings и между таблицами items, ratings.

2. 1. users и ratings

Между таблицами users и ratings существует отношение "один ко многим". Один пользователь может оставить несколько отзывов на разные товары, поэтому user_id в таблице ratings является внешним ключом, связывающим таблицы users и ratings.

2. 2. items и ratings

Между таблицами items и ratings также существует отношение "один ко многим". Один товар может иметь несколько отзывов от разных пользователей, поэтому item_id в таблице ratings является внешним ключом, связывающим таблицы items и ratings.

Задание 3

К какой/каким из созданных таблиц вы бы предложили создать индекс.

Рекомендуется создать индексы на поля, которые участвуют в частых операциях поиска, объединения и фильтрации данных. В данном случае можно создать следующие индексы:

3. 1. В таблице users:

- Индекс на поле user_id, так как это является первичным ключом и будет использоваться для быстрого поиска конкретных пользователей.

3. 2. В таблице items:

- Индекс на поле item_id, так как это является первичным ключом и будет использоваться для быстрого поиска конкретных товаров.

3. 3. В таблице ratings:

- Индекс на поле item_id, так как это поле будет использоваться для поиска отзывов для конкретного товара.
- Индекс на поле user_id, так как это поле будет использоваться для поиска отзывов, оставленных конкретным пользователем.

Эти индексы помогут ускорить выполнение поисковых запросов, объединить данные из различных таблиц и фильтровать результаты запросов.

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.

Команды, которые смогут успешно добавить записи в таблицы, это:
-- 2
INSERT INTO Car_owner VALUES ('7984672834', 'Иван Петров')
-- 6
INSERT INTO Car VALUES ('4752909757', 'A822EY', 99, 'Skoda Rapid',
'черный', 125, 2021, 35)
-- 7
INSERT INTO Car VALUES ('7984672834', 'A822EY', 99, 'Hyundai Solaris',
'черный', 123, 2019, 20)
-- 9
INSERT INTO Car VALUES ('7984672834', 'E340BT, 77, 'Toyota RAV4',
'Серебристо-серый', 146, 2019)

В командах 1, 3, 4, 5, 8 и 10 имеются ошибки:
- В команде 1 пропущено значение для поля Mileage.
- В командах 3 и 4 неверное название таблицы, должно быть "Car_owner" вместо "Car Owner".
- В команде 5 отсутствует закрывающая кавычка для значения поля CurN.
- В команде 8 неверный формат значения для поля Owner.
- В команде 10 в значениях полей Brand и Power имеется опечатка, отсутствует закрывающая кавычка для значения поля Brand и неверное значение для поля Power, оно должно быть больше 50.

Записи с ошибками не смогут быть успешно добавлены в таблицы, так как нарушают условия синтаксиса и ограничения на структуру данных.

Задание 5

Использование заданного скрипта для шардинга таблицы на 32 документа? И генерацией из этой таблицы 16 тестовых таблиц примерно по 2 документа test.docs00, 01, 02, ..., 15

Скрипт:

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

Данный скрипт можно использовать для шардинга таблицы на 32 документа и генерации 16 тестовых таблиц.
В скрипте, используется оператор INSERT INTO с командой SELECT для каждой таблицы docs00, docs01, ..., docs15, которая выбирает все строки из исходной таблицы documents, где значение поля id при делении на 16 дает остаток равный индексу таблицы (0-15).
Таким образом, каждая таблица будет содержать по 2 документа, т.к. всего 32 документа разделены на 16 таблиц.

Чтобы выполнить шардинг таблицы на 32 документа и создать 16 тестовых таблиц, нужно выполнить данный скрипт последовательно для каждой таблицы docs00, docs01, ..., docs15. Каждая команда будет выбирать соответствующие строки из исходной таблицы documents и вставлять их в соответствующую тестовую таблицу.

Таким образом, каждая из 16 тестовых таблиц будет содержать по 2 документа, и данные будут разделены между ними на основе значения поля id.