

Задание:

Домашнее задание №3

1. Выберите любое интересное вам приложение (можно продолжить работу над тем, что выбирали для предыдущих ДЗ).
2. Опишите задачу на разработку (ТЗ) нового метода API в соответствии с рассмотренным в видео шаблоном:
 - а. Описание задачи на разработку:
 - Описание бизнес-контекста – что за приложение и какую функцию разрабатываете
 - название метода
 - история версий
 - формат запроса (query, path, header) / пример запроса
 - формат тела запроса / маппинг данных в БД / пример тела запроса
 - формат ответа / пример ответа
 - коды ответов и сообщения
 - алгоритм обработки ответа
 - альтернативный сценарий
 - НФТ (требование к аудиту, логированию, размеру данных, RPS)
3. В качестве результата приложите получившееся ТЗ в текстовом формате.

Замечания к прошлому решению задачи:



Бурмистров Владимир

Добавить SQL-запросы для пагинации и описать стратегию кеширования популярных страниц.

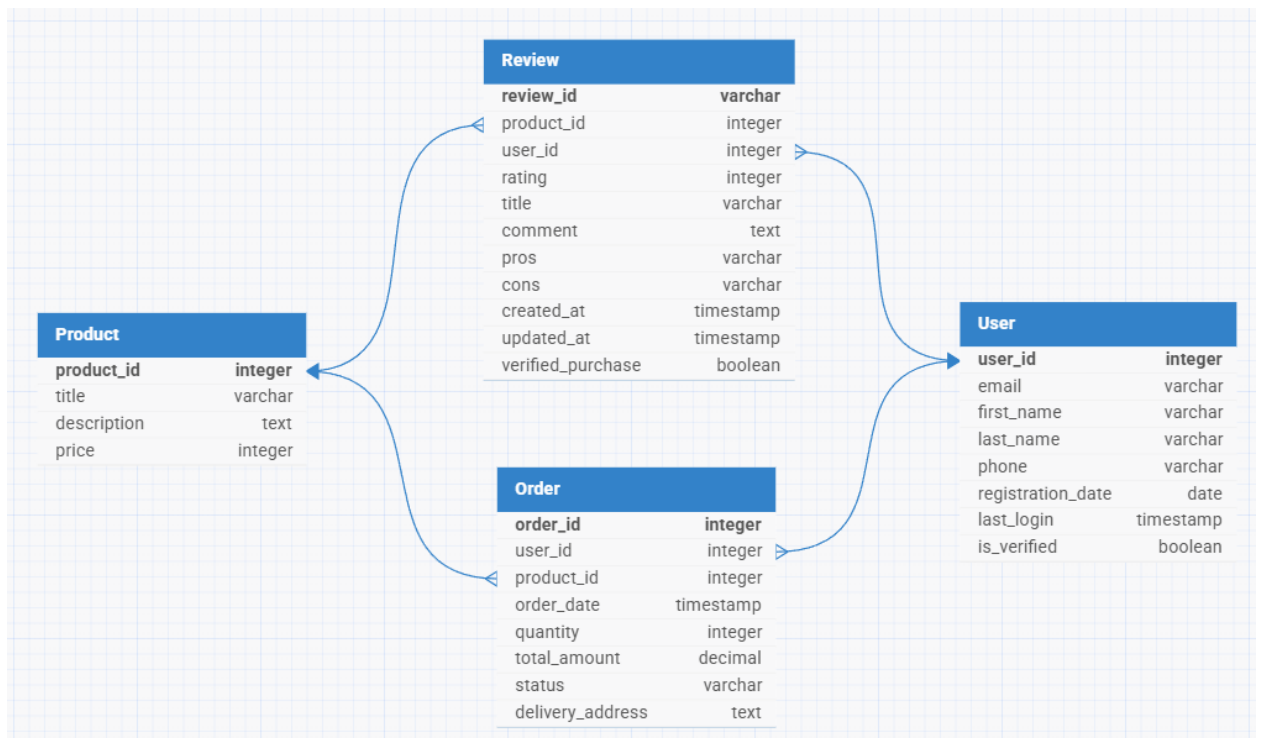
09.10.2025 05:57

Исправленное решение задачи:

Интернет-магазин с системой пользовательских отзывов и рейтингов.

Задача: я, как покупатель в интернет-магазине, хочу просматривать отзывы других покупателей на конкретный товар, чтобы принять обоснованное решение о покупке товара.

Физическая модель базы данных:



1. Название метода.

Метод для получения отзывов на конкретный товар:

GET/v1/products/{id_product}/reviews

2. История версий.

Версия	Дата	Изменения
1.0	01.09.2025	Первоначальная версия метода

3. Формат запроса.

Параметр	Тип	Расположение	Обязательность	Описание
id_product	integer	path	да	Идентификатор товара
page	integer	query	да	Номер страницы на сайте
limit	integer	query	да	Количество отзывов на странице

Пример запроса:

GET/v1/products/{product_id}/reviews?page=1&limit=10 HTTP/1.1

Host: api.marketcity.ru

Authorization: Bearer eyJhbGciOiJIUz...

4. Формат тела запроса

Тело запроса отсутствует (метод GET)

Маппинг данных в БД:

Поле в API	Поле в БД	Тип данных
product_id	product.product_id	int

5. Формат ответа

Успешный ответ (200 OK):

```
{
  "product": {
    "product_id": 12345,
    "title": "Phone 5",
    "number_of_reviews": 123,
    "rating": 4.5
  },
  "reviews": [ {
    "review_id": "phone5_1",
    "user": {
      "user_id": 88463,
      "name": "Anna",
      "lastname": "New",
      "is_verified": true
    },
    "rating": 5,
    "comment": "bbb",
    "pros": "ccc",
    "cons": "ddd",
    "created_at": "2024-05-15T10:30:00Z",
    "updated_at": "2024-05-25T10:30:00Z",
    "verified_purchase": true
  },
  {
    "review_id": "phone5_2",
    "user": {
      "user_id": 88464,
      "name": "Anna1",
      "lastname": "New1",
      "is_verified": true
    },
    "rating": 5,
```

```

        "comment" : "bbb1",
        "pros" : "ccc1",
        "cons" : "ddd1",
        "created_at" : "2024-06-15T10:30:00Z",
        "updated_at" : "2024-06-25T10:30:00Z",
        "verified_purchase" : true
    }
}

```

Структура ответа:

Поле	Тип	Описание
product_id	integer	Идентификатор товара
title	string	Название товара
number_of_reviews	integer	Количество отзывов на товаре
rating	integer	Общая оценка товара
review_id	integer	Идентификатор отзыва
user_id	integer	Идентификатор покупателя
name	string	Имя покупателя
lastname	string	Фамилия покупателя
is_verified	boolean	Наличие регистрации покупателя
rating	integer	Оценка покупателя
comment	string	Комментарий покупателя
pros	string	Достоинства товара
cons	string	Недостатки товара
created_at	datetime	Дата создания комментария
updated_at	datetime	Дата обновления комментария
verified_purchase	boolean	Факт совершения покупки товара

6. Коды ответов и сообщения

Код	Статус	Пример тела ошибки
200	OK	-
400	Bad Request	{"error": "INVALID LIMIT"}
404	Not Found	{"error": "PRODUCT WITH ID # NOT FOUND"}
500	Server Error	{"error": "INTERNAL_SERVER_ERROR"}

7. Алгоритм обработки запроса

Валидация:

- проверить наличие product_id в path-параметрах;
- проверить корректность query-параметров;
- убедиться, что товар с указанным product_id существует;
- проверить авторизацию покупателя

Логика:

- запрос к БД: получить информацию о товаре и рассчитать статистику отзывов (количество, средний рейтинг);
- запрос к БД: получить список отзывов из таблицы Review;
- сформировать ответ с товаром, покупателями и их отзывами.

Аудит:

- Записать в лог: product_id, user_id, review_id (если авторизован), timestamp.

8. Альтернативный сценарий

На товар нет отзывов (200 OK)

```
{
  "product": {
    "product_id": 12345,
    "title": "Phone 6",
    "total_reviews": 0,
    "rating": null
  },
  "reviews": [],
}
```

9. Нефункциональные требования

Требование	Значение
Логирование	Сохранять ошибки в ELK-стек
Размер данных	≤ 1,5 KB на запрос
RPS	1 000 запросов/секунду

Приложения

- Swagger-спецификация метода
- Диаграмма последовательностей (UML) для метода
- SQL-запрос для выборки данных (пример, для товара с идентификатором id=1):

Запрос 1: товар + статистика:

```
SELECT
  p.product_id,
  p.title,
  p.description,
```

```

    p.price,
    COUNT(r.review_id) as number_of_reviews,
    AVG(r.rating) as rating
FROM product AS p
LEFT JOIN review AS r ON p.product_id = r.product_id
WHERE p.product_id = 1
GROUP BY p.product_id, p.title, p.description, p.price

```

Запрос 2: отзывы

```

SELECT
    r.review_id,
    r.rating,
    r.comment,
    r.pros,
    r.cons,
    r.created_at,
    r.updated_at,
    r.verified_purchase,
    u.user_id,
    u.first_name AS name,
    u.last_name AS lastname,
    u.is_verified
FROM reviews AS r
JOIN users AS u ON r.user_id = u.user_id
WHERE r.product_id = 1
ORDER BY r.created_at DESC
LIMIT 10 OFFSET 0;

```

Требования к безопасности

- Все запросы должны проходить через API Gateway с проверкой JWT

Критерии приемки

- Метод возвращает корректные данные о товаре и отзывах
- Пагинация работает корректно при большом количестве отзывов

Стратегия кеширования популярных страниц.

Основная идея – использовать Redis – это быстрая база данных, которая хранит данные в оперативной памяти.

Когда пользователь запрашивает данные о товаре, данные сначала ищутся в кэше Redis. Если данные найдены, то они сразу возвращаются

клиенту. Если нет – запрос отправляется в основную базу данных. После получения данных из базы они возвращаются пользователю и одновременно записываются в кэш Redis.

Время хранения данных варьируется в зависимости от категории товара. Например, популярные товары – 10 минут, товары стандартной востребованности – 20 минут, редко просматриваемые товары – 30 минут.

При любом изменении (например, добавляется новый отзыв на товар), происходит инвалидация кэша. После удаления необходимо заново выполнить необходимые запросы к базе данных.