

Задание

Требование к оформлению результата:

Ответы прописаны не порядку в котором предоставлялся файл(то есть просто набор ответов)

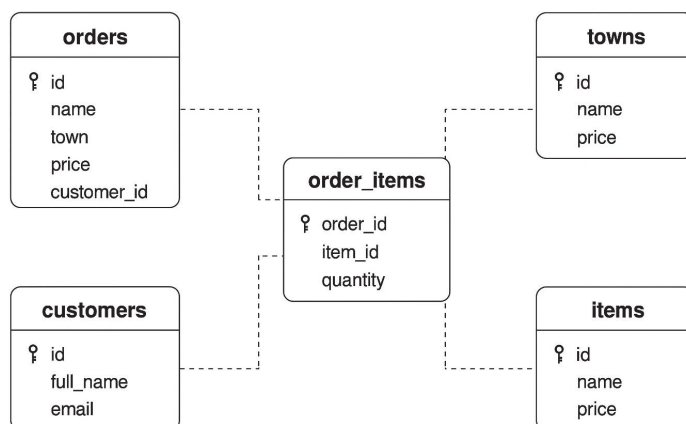
1. Базы данных - тест

1	Содержит ли какую-то информацию таблица, в которой нет полей?	1. Содержит информацию о структуре БД 2. Таблица без полей существовать не может
2	В записи файла реляционной БД может содержаться:	1. Неоднородная информация (данные разных типов)
3	Чем первичный ключ отличается от внешнего ключа?	1. Значения первичного ключа всегда должны быть уникальными и не могут быть null, значения внешнего ключа могут повторяться 2. Первичный ключ является идентификатором для строки, а внешний ключ используется для связывания таблиц

4	В какой нормальной форме говорится о том, что все атрибуты зависят от первичного ключа, а не от его части?	1. 2НФ
5	В каком порядке в СУБД выполняются операторы SELECT, FROM, GROUP BY?	1. Сначала FROM, потом GROUP BY и только потом SELECT
6	Чем отличается оператор WHERE от HAVING	<p>1. Оператор HAVING применяется для фильтрации групп, а WHERE - для фильтрации отдельных строк</p> <p>2. HAVING работает только с агрегатными функциями, а WHERE может работать с любыми типами выражений</p> <p>3. WHERE может использоваться для фильтрации по любому полю или выражению, а HAVING - только для фильтрации по выражению в списке выбора или агрегатной функции</p> <p>4. HAVING всегда используется после GROUP BY, а WHERE может использоваться до или после GROUP BY</p>
7	Какой результат покажет выполнение операторов SELECT COUNT (*)?	1. Число строк таблицы, указанной во FROM, включая значение NULL
8	В таблице «Animals» базы данных зоопарка содержится информация обо всех обитающих там животных, в том числе о лисах: red fox, grey fox, little fox.	1. SELECT age FROM Animals WHERE Animal LIKE "%fox"

	Напишите запрос, возвращающий информацию о возрасте лис	SELECT age FROM Animals WHERE Animal LIKE '%fox';					
9	Чем отличается DELETE от TRUNCATE?	1. DELETE используется для удаления одной или нескольких строк из таблицы, а TRUNCATE используется для удаления всех строк из таблицы 2. DELETE может использовать условие WHERE, а TRUNCATE всегда удаляет все записи из таблицы					
10	<div>Дана таблица:</div> <table><tr><td>COLOR</td></tr><tr><td>BLUE</td></tr><tr><td>RED</td></tr><tr><td>null</td></tr><tr><td>RED</td></tr></table> <div>Каким будет результат запроса? SELECT COUNT (DISTINCT color) FROM Table</div>	COLOR	BLUE	RED	null	RED	1. 3
COLOR							
BLUE							
RED							
null							
RED							

2. Базы данных - ER



3. Интеграции

Проектирование REST API для интернет-магазина

1. REST API

1. Получение списка товаров

GET /api/products

Ответ:

```
[  
  {  
    "id": 1,  
    "name": "Кроссовки Nike Air Zoom",  
    "price": 7999,  
    "thumbnail_url": "https://example.com/images/1.jpg",  
    "short_description": "Лёгкие беговые кроссовки"  
  }  
]
```

2. Получение информации о конкретном товаре

GET /api/products/{id}

Ответ:

```
{  
  "id": 1,  
  "name": "Кроссовки Nike Air Zoom",  
  "price": 7999,  
  "images": [  
    "https://example.com/images/1.jpg",  
    "https://example.com/images/1_side.jpg"  
  ],  
  "description": "Лёгкие беговые кроссовки...",  
  "sizes": ["38", "39", "40", "41", "42", "43"]  
}
```

3. Добавление товара в корзину

Проектирование REST API для интернет-магазина

POST /api/cart

Тело:

```
{  
  "product_id": 1,  
  "quantity": 2,  
  "size": "42"  
}
```

Ответ:

```
{  
  "message": "Товар добавлен в корзину",  
  "cart": {  
    "items": [  
      {  
        "product_id": 1,  
        "name": "Кроссовки Nike Air Zoom",  
        "quantity": 2,  
        "price": 7999,  
        "total_price": 15998,  
        "size": "42"  
      }  
    ],  
    "total_items": 2,  
    "total_price": 15998  
  }  
}
```

4. Получение содержимого корзины

GET /api/cart

Ответ:

Проектирование REST API для интернет-магазина

```
{
  "cart": {
    "items": [
      {
        "product_id": 1,
        "name": "Кроссовки Nike Air Zoom",
        "quantity": 2,
        "price": 7999,
        "total_price": 15998,
        "size": "42",
        "thumbnail_url": "https://example.com/images/1.jpg"
      }
    ],
    "total_items": 2,
    "total_price": 15998
  }
}
```

Проектирование REST API для интернет-магазина

2. Sequence UML Диаграмма

```
Client          API Server          Database
|               |                   |
|--- GET /products -----> |
|               |--- SELECT * FROM products --->|
|               |<----- JSON (list of products)--|
|<-- JSON (products list) -----|
|
|--- GET /products/1 ----->|
|               |--- SELECT * FROM products WHERE id=1 -->|
|               |<----- JSON (product details) -----|
|<-- JSON (product details) -----|
|
|--- POST /cart ----->|
|               |--- INSERT INTO cart_items(...) ----->|
|<-- JSON (confirmation) -----|
|
|--- GET /cart -----> |
|               |--- SELECT * FROM cart_items WHERE user_id=? -->|
|               |<----- JOIN with products info -----|
|<-- JSON (cart with items) -----|
```

4. Алгоритмическое мышление

```
[Старт]
| v
[Включить смартфон]
| v
[Разблокировать экран]
| v
[Открыть банковское приложение]
| v
[Авторизоваться (по пин-коду, отпечатку, Face ID)]
| v
[Выбрать раздел "Платежи" или "Мобильная связь"]
| v
[Выбрать оператора или номер телефона для пополнения]
| v
[Ввести сумму пополнения: 100 ]
| v
[Выбрать карту списания]
| v
[Подтвердить операцию (например, по СМС или в приложении)]
| v
[Операция выполнена?] -- Да --> [Показать чек / уведомление] --> [Готово]
|
-- Нет --> [Показать ошибку] --> [Повторить операцию]
```