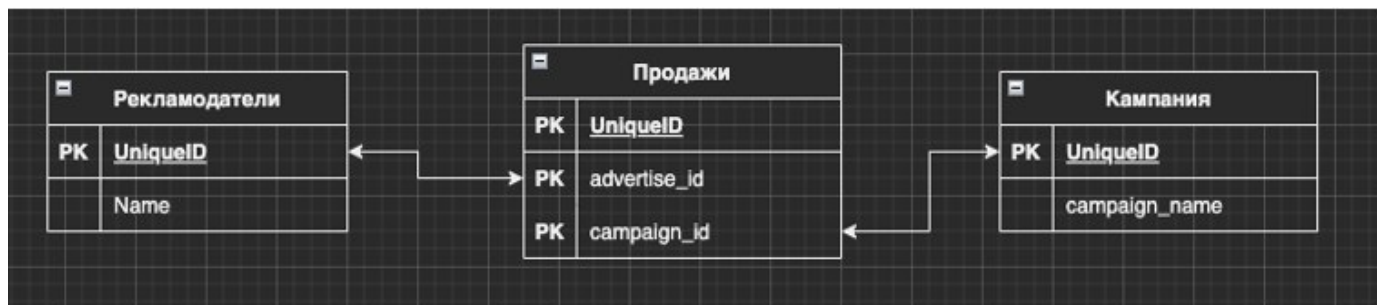


## Домашняя работа 3: Получение денормализованных таблиц из нормализованных

1. Денормализуйте таблицу так, чтобы не нужно было для каждого рекламодателя постоянно подсчитывать количество кампаний и продаж.



Создадим новую денормализованную таблицу, которая будет содержать предварительно подсчитанные агрегаты.

Предлагаемая структура денормализованной таблицы:

Таблица может выглядеть следующим образом:

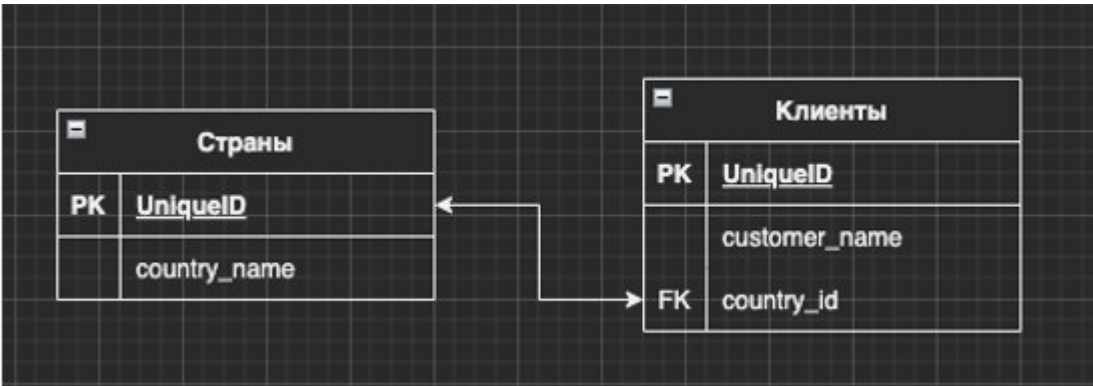
advertise_id	advertiser_name	campaign_count	sales_count
1	Advertiser 1	5	100
2	Advertiser 2	3	50

SQL для денормализации:

```
CREATE TABLE denormalized_advertisers AS
SELECT a.UniqueID           AS advertise_id,
       a.Name               AS advertiser_name,
       COUNT(DISTINCT c.UniqueID) AS campaign_count,
       COUNT(s.UniqueID)      AS sales_count
FROM   advertisers a
       LEFT JOIN
       sales s ON a.UniqueID = s.advertise_id
       LEFT JOIN
       campaigns c ON s.campaign_id = c.UniqueID
GROUP BY a.UniqueID, a.Name;
```

После выполнения этого запроса будет создана новая таблица `denormalized_advertisers`, где для каждого рекламодателя указано заранее подсчитанное количество кампаний и продаж.

2. В базе данных есть две таблицы: страны и клиенты. Одной из потребностей компании является исследование клиентов и стран с точки зрения эффективности продаж, поэтому часто выполняются объединения между таблицами: клиенты и страны. Что нужно сделать, чтобы ограничить частое объединение этих двух таблиц?



Чтобы ограничить частое объединение таблиц Страны и Клиенты , можно денормализовать данные, добавив информацию о стране прямо в таблицу Клиенты . Это уменьшит количество операций объединения, так как вся необходимая информация будет доступна в одной таблице.

Шаги оптимизации:

- 1. Добавить колонку country\_name в таблицу Клиенты
- 2. Обновить данные в таблице Клиенты :

Заполнить новое поле country\_name с помощью SQL-запроса, объединяющего обе таблицы:

```
ALTER TABLE Клиенты
  ADD COLUMN country_name STRING;

UPDATE Клиенты
SET country_name = (SELECT c.country_name
                    FROM Страны c
                    WHERE Клиенты.country_id = c.UniqueID);
```

Теперь таблица Клиенты будет выглядеть так:

UniqueID	customer_name	country_id	country_name
1	Иван Иванов	101	Россия
2	Джон Смит	102	США

3. Вернемся к первому примеру. Предположим, компания хочет регулярно извлекать данные о продажах, например, о кампаниях или рекламодателях с полными именами. Как мы можем решить проблему постоянной необходимости объединения таблиц?

Чтобы решить проблему постоянного объединения таблиц для извлечения данных о кампаниях или рекламодателях с полными именами, можно применить несколько подходов в зависимости от бизнес-потребностей:

1. Денормализация данных

Добавим информацию о рекламодателях и кампаниях в таблицу Продажи . Это упростит запросы, так как все необходимые данные будут доступны в одной таблице.

Шаги:

1.1. Добавим в таблицу Продажи следующие колонки: • advertiser\_name — имя рекламодателя. • campaign\_name — имя кампании.

1.2. Заполним новые колонки:

```
ALTER TABLE Продажи
  ADD COLUMN advertiser_name STRING;
ALTER TABLE Продажи
  ADD COLUMN campaign_name STRING;

UPDATE Продажи
SET advertiser_name = (SELECT a.Name
                      FROM Рекламодатели a
                      WHERE Продажи.advertise_id = a.UniqueID),
    campaign_name    = (SELECT c.campaign_name
                      FROM Кампания c
                      WHERE Продажи.campaign_id = c.UniqueID);
```

После обновления данные в таблице Продажи будут выглядеть так:

UniqueID	advertise_id	campaign_id	advertiser_name	campaign_name
1	101	201	Advertiser A	Campaign X
2	102	202	Advertiser B	Campaign Y

Преимущества: • Все данные доступны в одной таблице, что исключает необходимость частых объединений. • Ускорение выполнения аналитических запросов.

Недостатки: • Увеличение размера таблицы Продажи из-за добавления избыточных данных. • Необходимость синхронизации при изменении данных в таблицах Рекламодатели или Кампания .

2. Материализованное представление

Если денормализация данных нежелательна, можно создать материализованное представление, которое автоматически объединяет таблицы и сохраняет результаты для ускорения запросов.

Шаги:

2.1. Создадим материализованное представление:

```
CREATE
MATERIALIZED VIEW sales_with_details AS
SELECT s.UniqueID AS sale_id,
       s.advertise_id,
       s.campaign_id,
       a.Name      AS advertiser_name,
       c.campaign_name
FROM Продажи s
     LEFT JOIN Рекламодатели a ON s.advertise_id = a.UniqueID
     LEFT JOIN Кампания c ON s.campaign_id = c.UniqueID;
```

## 2.2. Используем представление в запросах:

Теперь можно запрашивать данные прямо из представления `sales_with_details`, без явного объединения таблиц.

Пример запроса:

```
SELECT sale_id, advertiser_name, campaign_name
FROM sales_with_details
WHERE advertiser_name = 'Advertiser A';
```

Преимущества:

- Автоматическое управление объединениями.
- Ускорение выполнения часто используемых запросов.
- Меньшая избыточность данных по сравнению с полной денормализацией.

Недостатки:

- Необходимость обновления материализованного представления при изменении данных в исходных таблицах.

Рекомендуемое решение:

Если регулярно используются данные о продажах, рекламодателях и кампаниях, материализованное представление — наиболее подходящий вариант, так как оно сочетает производительность и гибкость без избыточности данных. Однако для максимальной скорости в аналитике денормализация таблицы Продажи тоже может быть эффективным решением.