1. Запросы бывают логически сложными
2. Если реализовать вопрос не получается сразу, то можно попробовать применить подзапрос, решающий часть задачи
3. Бывают такие запросы, которые сложно или невозможно реализовать без подзапросов
4. Но в то же время можно избежать подзапросов, использую соединения
5. Если можно избиваться от подзапроса, надо ли это делать? Тут 50\50, надо сравнивать читабельность запросов. Если с подзапросом выглядит лучше, то и оставить следует его при условии, что запрос с подзапросом не будет менее производительным, чем обычный запрос с соединением.

**Примеры задач**

1. Выбрать страны, в которых находятся и поставщики, и заказчики

**SELECT company\_name**

**from suppliers**

**where country in (**

**SELECT DISTINCT country**

**FROM customers**

**)**

**Но можно было бы обойтись и без запроса в запросе, используя следующий запрос:**

**select distinct suppliers.company\_name as company\_names**

**from suppliers**

**join customers using(country)**

Но тут стоит иметь ввиду, что не всегда можно переделать запрос так, чтобы был использован джойн

1. Следующая задача покажет, что подзапрос можно находиться в разных частях запрос. В ней мы группируем товары по категориям, выводим названия категорий и сумму товаров в ней, еще мы хотим вывести только определенное количество записей, найдя наименьший продакт\_ид и прибавя к нему некоторое число

**select category\_name, sum(units\_in\_stock)**

**from products**

**join categories using(category\_id)**

**group by category\_name**

**order by sum(units\_in\_stock)**

**limit (select min(product\_id) + 4 from products)**

1. Задача, когда надо вывети количество товаров, которых больше, чем среднее количество товаров.

**select product\_name, units\_in\_stock**

**from products**

**where units\_in\_stock > (select avg(units\_in\_stock)**

**from products)**

**Конструкция WHERE EXISTS**

Возвращает тру, если в подзапросе была возвращения 1+ строк.

**Примеры задач**

1. Как выбрать компании, которые делали заказы весом от 50 до 100?

**select company\_name, contact\_name**

**from customers**

**where exists (select customer\_id from orders**

**where customer\_id = customers.customer\_id**

**and freight between 50 and 100)**

Тут на что надо обратить внимание. В первую очередь, это сам подзапрос, тут соединение таблиц идет не через джойн, а через where, при этом остальной синтаксис привычен, разве что юзинг уже не получится использовать

1. Выбрать продукты, которые не покупались за определенный период

**select product\_name**

**from products**

**where not exists (select orders.order\_id from orders**

**join order\_details using(order\_id)**

**where order\_details.product\_id = product\_id**

**and order\_date between '1995-02-01' and '1995-02-15'**

**)**

Вот тут ниже два примера, как вывести имя и фамилию работника, который отвечает за поставки туда и сюда. Первый пример – там сразу идет дистинкт, а во втором его надо прописать. Тут в чем причина, если дистинкт не прописывать во втором, то он выдает инфу, дублируя имена, т.к. там у него куча order\_id, и он их тоже выгружает, а первый такой темой не обременен.

**select last\_name, first\_name**

**from employees**

**where exists(select employee\_id from orders**

**where employee\_id = employees.employee\_id**

**and ship\_country = 'UK')**

**order by first\_name**

**select distinct last\_name, first\_name, ship\_country**

**from employees**

**join orders using(employee\_id)**

**where ship\_country = 'UK'**

**order by first\_name**

-- сделать что-нибудь с продукцией по категориям, например, вывести поставщиков по категориям, указав название продуктов

**select company\_name, product\_name, category\_name**

**from suppliers**

**join products using(supplier\_id)**

**join categories using(category\_id)**

**where category\_id = '2'**

второй запрос я пытался сделать в этом же ключе, но данные он выводил уже немного другие, а учитывая, что вроде как подзапросы чуть менее очевидные, хз, что правда, а что нет. В итоге я сделал другой запрос и увидел, что первое все-таки правильно

select product\_name, category\_id

from products

where category\_id = '2'

**Подзапросы с квантификаторами ANY, ALL**

1. Задача, где надо выбрать все уникальные компании, которые сделали заказ товаров более чем на 40 единиц товара. Решение есть такое, но оно идет без подзапроса.

**select distinct company\_name**

**from customers**

**join orders using(customer\_id)**

**join order\_details using(order\_id)**

**where quantity > 40**

а вот это уже с подзапросом

**select distinct company\_name**

**from customers**

**where customer\_id = ANY(**

**select customer\_id -** тут идет выборка по кастомер\_ид, чтобы в WHERE взять по кастомер\_ид

**from orders**

**join order\_details using(order\_id)**

**where quantity > 40**

**)**

В скобках – это обычный джойн, который находит кастомер\_ид из заказов, ища в ордер\_деталях значения более 40, в основном запросе идет выбор уникальных имен компаний из списка покупателей, где кастомер\_ид – и тут уже идет подзапрос, который выбирает строки, подходящие условиям.

Сам подзапрос отдельно выглядит так:

**select customer\_id**

**from orders**

**join order\_details using(order\_id)**

**where quantity > 40**

но! Что тут нужно отметить. Запрос с подзапросом мы сделали, это хорошо для примера, но по факту изначальный вариант проще, короче, более понятный.

2 задача. Выбрать продукты, количество которых больше среднего по заказам. Сначала нужно найти среднее количество единиц товара по всем заказам

**select distinct product\_name, quantity**

**from products**

**join order\_details using(product\_id)**

**where quantity > (**

**select avg(quantity)**

**from order\_details**

**)**

3 задача. Найти все продукты, количество которых больше среднего значения количества заказанных товаров из групп, полученных группированием по продакт\_ид

Надо разбить на детали. Первая деталь – тут просто нахождение среднего значения по продакт\_ид, это не категории товаров типа молочная продукция, хлебобулки и т.д., а просто ид продукта и здесь идет среднее значение заказанных товаров:

**select avg(quantity)**

**from order\_details**

**group by product\_id**

тут идет таблица, где идет большое количество продакт\_ид, и у каждого из них свое среднее значение заказанных товаров. Нас интересуют в итоге те продукты, число которых больше, чем любое значение из этой финальной таблички

далее основной запрос:

select distinct product\_name, quantity

from products

join order\_details using(product\_id)

where quantity > all(

select avg(quantity)

from order\_details

group by product\_id

)

Но в принципе, т.к. в этой таблице есть максимальное значение так или иначе, то можно было выбрать не все значения, вообще не выставлять квантификатор, и суть не поменялась бы.

**Домашка**

1 задача.

-- Вывести продукты количество которых в продаже меньше самого малого среднего количества продуктов в деталях заказов (группировка по product\_id).

-- Результирующая таблица должна иметь колонки product\_name и units\_in\_stock.

**select distinct product\_name, units\_in\_stock**

**from products**

**join order\_details using(product\_id)**

**where units\_in\_stock < all(**

**select avg(quantity)**

**from order\_details**

**group by product\_id**

**)**

**order by units\_in\_stock**

выводим имя продукта и количество юнитов в продаже из продуктов, соединяемся с «деталями заказов» по продакт\_ид, а по количество юнитов в продаже мы выбираем только те, которые меньше, чем данные из подзапроса. В подзапросе опять будет таблица с данными по среднему значению количества в деталях заказа.

4 задача.

Вывести все товары (уникальные названия продуктов), которых заказано ровно 10 единиц (конечно же, это можно решить и без подзапроса). Или я по-тупому сделал, и есть другое решение, или оно правильное, просто задача сама по себе не очень умная

select product\_name, units\_on\_order

from products

where units\_on\_order = all(

select units\_on\_order

from products

where units\_on\_order = '10'

)

3 задача в процессе решения

--Напишите запрос, который выводит 3 заказа с наибольшей стоимостью, которые были созданы после 1 сентября 1997 года включительно

--и были доставлены в страны Южной Америки.

--Общая стоимость рассчитывается как сумма стоимости деталей заказа с учетом дисконта.

--Результирующая таблица должна иметь колонки customer\_id, ship\_country и order\_price,

--строки которой должны быть отсортированы по стоимости заказа в обратном порядке.

-- бразилия, аргентина, венесуэла

select units\_in\_stock \* unit\_price as order\_price

from products

order by order\_price desc

select freight, order\_date

from orders

where ship\_country in('Argentina', 'Brazil', 'Venezuela') and order\_date > '1997-09-01'

group by freight, order\_date

order by order\_date