

При выполнении дз я работал с PostgreSQL.

**Часть 1. Задание 1.** Я четко следовал условию: выполняю группировку по городу и по возрастной категории (ведь условию сказано “для каждого города” и “сгруппированных по возрастным категориям”). Также в условии сказано: “Для каждого города выведите число покупателей... отсортированных по убыванию количества покупателей в каждой категории.” - я это интерпретирую, как сортировку и по городу, и по убыванию кол-ва покупателей (иначе один и тот же город может неупорядоченно появляться в результирующей таблице - не будет соответствия количества покупателей каждому городу).

P.S. Я привел в решении два запроса (на всякий случай, чтобы не было неоднозначностей): в одном группировка по возрасту человека, ведь “под возрастной категорией подразумевается возраст человека”; другой – “именно для категорий”, которые определяются оператором case.

**Часть 1. Задание 2.** Тут проще: просто следую условию, постепенно всё добавляю: сначала выбираю товары с названиями, соответствующими нашим шаблонам (ilike - регистронезависимо), далее выполняю группировку по категории, после выбираю нужные столбцы.

**Часть 2. Задание 1.** В задании просят средний рейтинг категорий продавца, я его округлил до двух знаков, чтобы мы в результате не смотрели на много знаков после запятой.

В условии немного неоднозначно прописаны “остальные”, но в процессе обсуждения в беседе мы пришли к выводу, что роог – отличные от rich по суммарной выручке, т.е. те, кто продает более одной категории, но чья суммарная выручка  $\leq 50\,000$ . Но возникает вопрос, что делать с не роог и не rich – раз просят выводить метку роог или rich, то тех, кто не попадает, под данные категории просто не будем выводить (также обсудили это в беседе). Итак, rich:  $> 1$  категории и суммарная выручка  $> 50\,000$ ; роог:  $> 1$  категории и суммарная выручка  $\leq 50\,000$ ; ни роог, ни rich:  $\leq 1$  категории.

По коду: в табличном выражении with (я его использую, чтобы код остался читаемым) я выполняю группировку по seller\_id и получаю seller\_id, total\_categ, avg\_rating, total\_revenue, также я сразу отмечаю категорию ‘Bedding’, которая согласно условию не должна учитываться в расчетах. Раз мы решили тех, кто ни rich, ни роог, не выводить, то я добавляю в where условие total\_categ  $> 1$ . При использовании полей из табличного выражения я определяю согласно вышеуказанным замечаниям категорию rich или роог. И в конце делаю сортировку по seller\_id.

Замечу также, что под фразами “продают более одной категории” или “количество категорий” я понимаю именно уникальные категории (с этой целью делаю `distinct`), потому что, как мне кажется, репрезентативно именно разнообразие категорий: например, если кто-то продает в категории шины 2 вида покрышек, то это всё еще одна категория продажи.

**Часть 2. Задание 2.** Здесь не буду использовать табличное выражение для получения неуспешных продавцов (ведь нам не требуется делать `case` и вычислять кучу столбцов). Таких продавцов определяю через условие `having` в группировке.

Также заметим, что у одного продавца может быть несколько категорий, значит, и несколько дат регистрации. Будем считать датой регистрации каждого продавца самую раннюю дату регистрации из имеющихся для каждого продавца (обсуждалось в беседе). Т.е. под продавцом я понимаю уникальный `seller_id` (соответственно группировка именно по нему).

Вместо времени, когда выполняется задание так и хочется написать `current_date`, но все же раз нужно время выполнения, то будем использовать `'2024-11-08'`.

Под “разницей между максимальным и минимальным сроком доставки среди неуспешных продавцов” я понимаю максимум срока доставки среди всех неуспешных продавцов вычесть аналогичный минимум.

По коду: в табличном выражении `t` я получаю `seller_id` и `month_from_registration`, но сразу здесь `max_delivery_difference` посчитать не можем (ведь мы работаем с группами), но можем посчитать максимум и минимум в каждой группе. Далее возьмем с помощью оконной функции максимум из этих максимумов по группам, и аналогично вычисляется минимум, после считается искомое поле `max_delivery_difference`. Альтернативой данному решению может быть решение, где в еще одном табличном выражении вычисляется данная разница, но тогда потребуется подзапрос для проверки того, что `seller_id` относится к неуспешным продавцам. Остановимся на первом решении, дабы избежать еще одного табличного выражения и подзапроса; более того, полученное решение получилось очень даже читаемым.

**Часть 2. Задание 3.** В данном задании “Отберите продавцов, зарегистрированных в 2022 году” я интерпретирую, как только те записи продавцов, где год даты регистрации 2022 (учтем это в условии `where`). Остальные условия с агрегатными функциями прописаны в `having`. Из интересного - использование агрегатной функции `string_agg`, которая помогает нам собрать пару категорий.