**Задание 1**

По таблице ниже при помощи SQL-запросов:

1. Разработать структуру таблиц
2. Выбрать правильные типы данных для столбцов и добавить указанные ограничения
3. При необходимости добавить недостающие столбцы и таблицы
4. Добавить первичные ключи, индексы и связи между таблицами

|  |  |  |  |
| --- | --- | --- | --- |
| **Таблица** | **Столбец** | **Ограничения** | **Возможные значения** |
| Товары | Название | Обязательно и уникально |  |
| Жизненно-важный | Обязательно | Да / Нет |
| Тип товара | Обязательно | Множество значений |
| Рецептурность | Необязательно | Множество значений |
| Аптеки | Название | Обязательно |  |
| Населенный пункт | Обязательно | Множество значений |
| Формат аптеки | Обязательно | Множество значений |
| Продажи | Время продажи | Обязательно | В формате «дата+время» |
| Номер чека | Обязательно |  |
| Аптека | Обязательно |  |
| Товар | Обязательно |  |
| Кол-во упаковок | Обязательно |  |
| Цена | Обязательно |  |

**Задание 2**

Разработать хранимые процедуры для **редактора товаров**:

* Вывод списка товаров с фильтрами по ЖВ (одно значение) и по типу (список значений)
* Добавление/изменение товара
  + Необходимо добавить проверки входных данных согласно ограничениям из таблицы выше
  + В случае ввода некорректных данных выводить текст с читабельной ошибкой
* Удаление товара

**Задание 3**

Написать SQL-запрос:

1. Фильтры:
   1. Дата продажи – 1е кварталы 2020 и 2021
   2. Аптека - города Краснодар, Сочи, Ставрополь и Севастополь, формат - «Клуб»
2. В разрезе **регионов** и **типов товаров** вывести:
   1. **среднюю цену** по всем попавшим в фильтр чекам
   2. **кол-во уникальных аптек**, где цена продажи товара была больше 300р.
   3. **список уникальных населенных пунктов** через точку с запятой.

***---****ЧЕРЕЗ точку с запятой вопрос*

1. Пронумеровать **типы в разрезе региона** по убыванию средней цены
2. Оставить те записи, где порядковый номер делится на 3

**Задание 4**

1. Выбрать **товары**, которые за март 2021 хотя бы на одной аптеке продались в кол-ве, превышающем 15 упаковок
2. Среди них выбрать самую высокую цену в марте 2021 и вывести дату, город, аптеку, товар, цену продажи.

**Задание 5**

Рассчитать **выручку** за апрель и май 2021 в разрезе **товара** и вывести **разницу выручки** мая относительно апреля.

|  |  |  |
| --- | --- | --- |
| **Апрель 2021** | **Май 2021** | **Сравнение** |
| |  |  | | --- | --- | | Товар | Выручка | | Тов1 | 100 | | Тов2 | 200 | | Тов3 | 300 | | |  |  | | --- | --- | | Товар | Выручка | | Тов1 | 150 | | Тов3 | 200 | | Тов4 | 100 | | |  |  |  |  | | --- | --- | --- | --- | | Товар | Выручка Апрель | Выручка Май | Разница | | Тов1 | 100 | 150 | 50 | | Тов2 | 200 |  | -200 | | Тов3 | 300 | 200 | -100 | | Тов4 |  | 100 | 100 | |

**Задание 6**

При помощи SQL-запроса сформировать JSON следующего вида:

{

“Номер чека” : [ “Месяц” , “Аптека”, “Сумма уп” ],

“Номер чека” : [ “Месяц” , “Аптека”, “Сумма уп” ],

………..

}

**Задание 7**

Со временем кол-во строк в таблице продаж превысило 100 млн строк и запросы стали выполняться заметно медленнее.

Какие пути оптимизации выборки из данной таблицы Вы можете предложить?

1. Запустить EXPLAIN (ANALYZE) на используемых запросах выборки,
   * проверить работает ли поиск по индексам, либо Seq Scan. Если не работает, то перестроить индексы
   * посмотреть как долго будет выполняться запрос
2. VACUUM / AUTOVACUUM
3. Попробовать оптимизировать запросы:
   * вместо подзапросов использовать JOIN или наоборот, сравнить результаты выполнения с помощью планировщика

**Задание 8**

Напишите SQL-запрос для вывода списка запросов, выполняющихся в настоящий момент на сервере.

select \* from pg\_stat\_activity