Вариант 7 Кривдин Михаил

Уровень 1

1. Дана схема базы данных в виде следующих отношений. С помощью операторов SQL создать логическую структуру соответствующих таблиц для хранения в СУБД, используя известные средства поддержания целостности (NOT NULL, UNIQUE, и т.д.). Обосновать выбор типов данных и используемые средства поддержания целостности. При выборе подходящих типов данных использовать информацию о конкретных значениях полей БД (см. прил.1)

СУДНО

|  |  |  |  |
| --- | --- | --- | --- |
| ИДЕНТИФИКАТОР | НАЗВАНИЕ | ПОРТ ПРИПИСКИ | ЛЬГОТА,% |

МЕСТА ПОГРУЗКИ

|  |  |  |  |
| --- | --- | --- | --- |
| ИДЕНТИФИКАТОР | ПРИЧАЛ | ПОРТ | ОТЧИСЛЕНИЯ НА ПОГРУЗКУ,% |

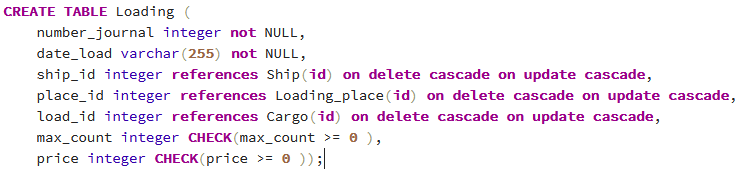
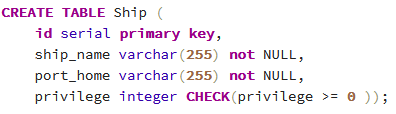
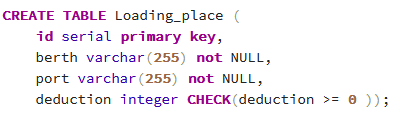
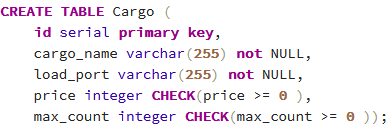
ГРУЗ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ИДЕНТИФИКАТОР | НАЗВАНИЕ | ПОРТ СКЛАДИРОВАНИЯ | СТОИМОСТЬ, РУБ | МАКС.  КОЛ-ВО |

ПОГРУЗКА

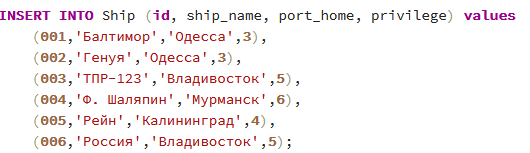
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| НОМЕР ВЕДОМОСТИ | ДАТА | СУДНО | МЕСТО ПОГРУЗКИ | ГРУЗ | КОЛ-ВО | СТОИМОСТЬ, РУБ |

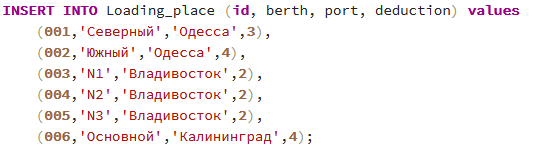
Создаю таблицы varchar(255) в основном потому что это строки и они вряд ли будут больше 255, т.к это названия портов, судов и т.д. Почти у всех есть Not NULL я предположил, что льгот и прочего в теории может просто не быть, но они обязательно с ограничениями , потому что отрицательных льгот и отчислений не бывает. On delete cascade и update нужны внешним ключам, чтобы при изменения значениях в основных таблица внешние ключи тоже менялись или удалялись.

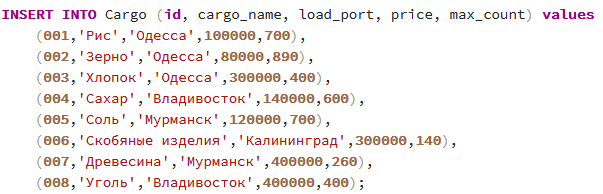


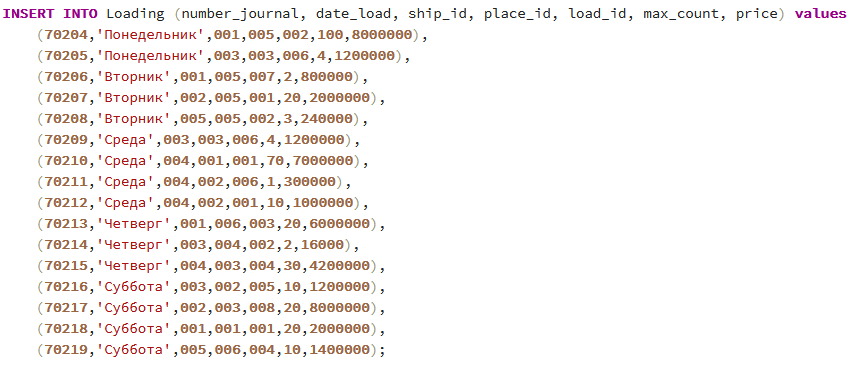
1. Ввести в ранее созданные таблицы конкретные данные (см. прил. 1). Использовать скрипт-файл из операторов INSERT или вспомогательную утилиту .

Просто делаю inert в таблицы.



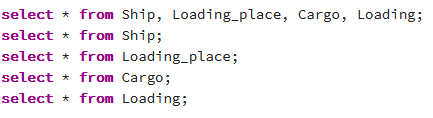






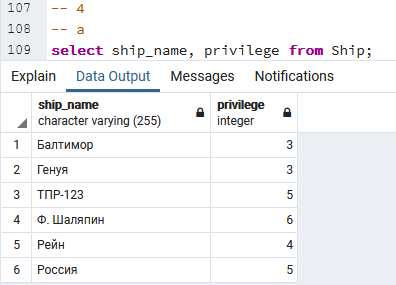
1. Используя оператор SELECT создать запрос для вывода всех строк каждой таблицы. Проверить правильность ввода. При необходимости произвести коррекцию значений операторами INSERT, UPDATE, DELETE.

Просто вывожу все столбы из всех таблиц и из каждой в отдельности



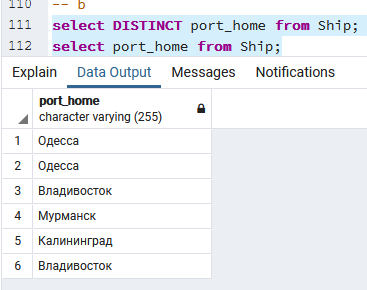
1. Создать запросы для вывода:
   1. всех различных названий судов и их льгот;

опять селект



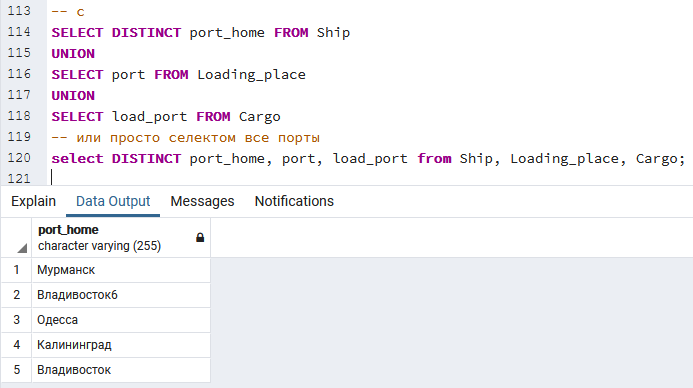
* 1. всех различных адресов приписки судов;

еще раз селект



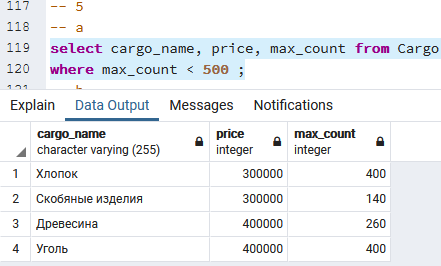
* 1. всех различных портов.

И ещё раз селект, но с union, который объединяет строки



1. Создав запрос получить следующую информацию:
   1. название и стоимость грузов, находящихся в количестве менее 500 единиц;

Селект с условием

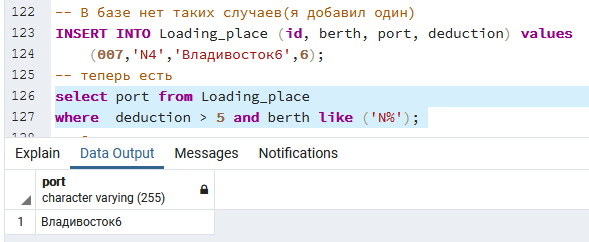


* 1. адрес порта, где есть причалы под номерами ( в названии есть “N”) или отчисления на погрузку более 5%;

Я вставил данные, которые нужны для примера вывода запроса

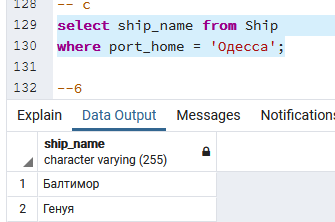
Селект с условием и логическим оператором И

Оператор like сравнивает строку из пристони с шаблоном N+любые символы



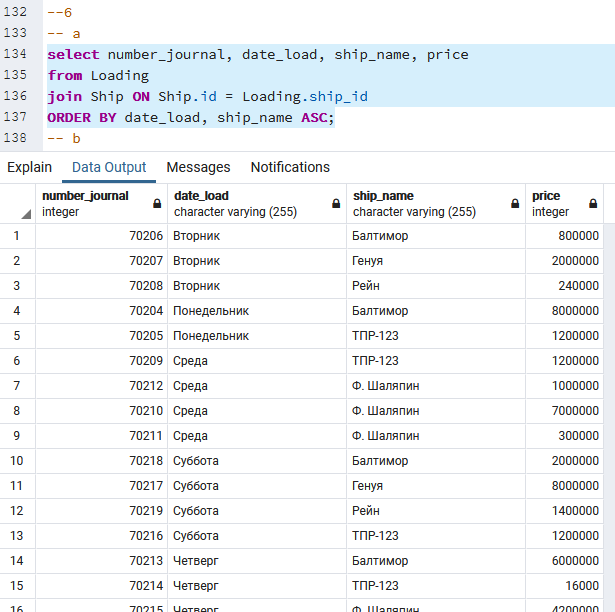
* 1. названия судов из Одессы.

Селект с условием



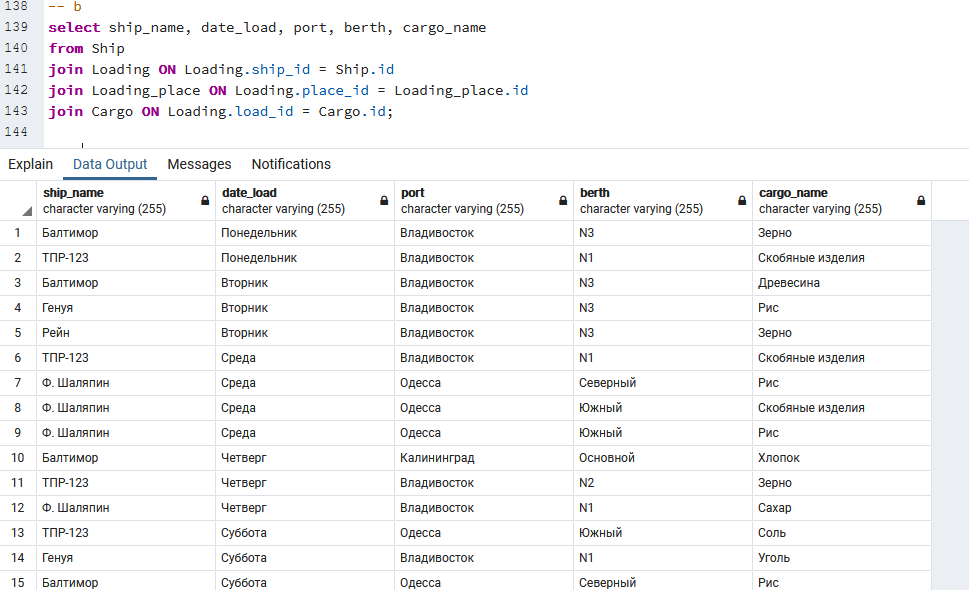
1. На основании данных о погрузке вывести все данные в таком формате:
   1. номер ведомости, дата, название судна, стоимость. Отсортировать по дате и названию судна;

join присоединяет одну таблицу к другой по какому-то столбцу(в данном случае id) по умолчанию используется inner join. Но можно ещё left, right,full, но мне иннера достаточно. Order by сортирует по какому-нибудь столбцу(или нескольким) в нужном порядке. По умолчанию стоит ASC, но можно и обратный DESC



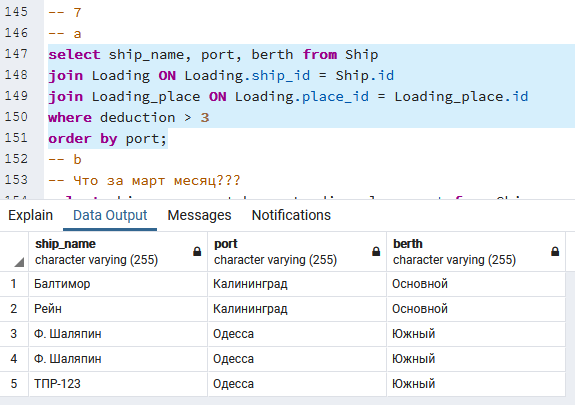
* 1. название судна, дата, название места погрузки, груз.

Селектор с джоинами



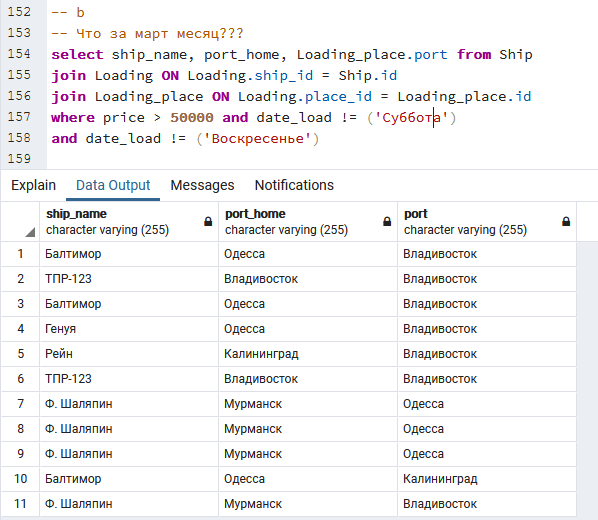
1. Вывести:
   1. названия судов, производивших погрузку в тех чужих портах, где отчисления на погрузку более 3%. Вывести с названиями этих портов и именами причалов. Отсорировать по портам;

селектор с джоинами, условием и отсортированый вывод. Вместо джоина можно использовать просто условия, что выводить только те, у которых совпадаю нужные id



* 1. названия и адреса приписки судов, перевозивших (грузивших) зрузы со стоимостью более 50000 ед. не позднее марта месяца. Вывести вместе с названиями портов;

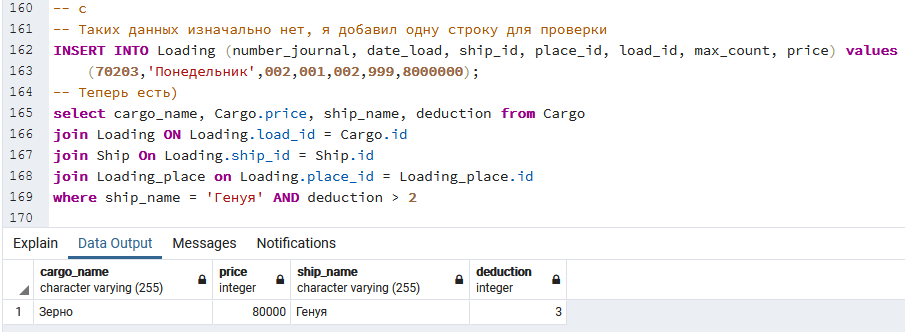
Все тоже самое



* 1. название и стоимость грузов, которые грузили на “Геную” на причалах, где отчисления на погрузку более 2%;

Все тоже самое

И добавляю необходимые данные

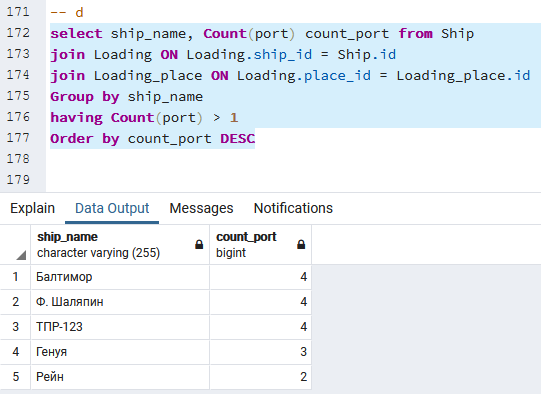


* 1. название судов, производивших погрузку более чем в одном порту. Добавить названия портов.

Количество портов обозначаю, как count\_port (без As для краткости).

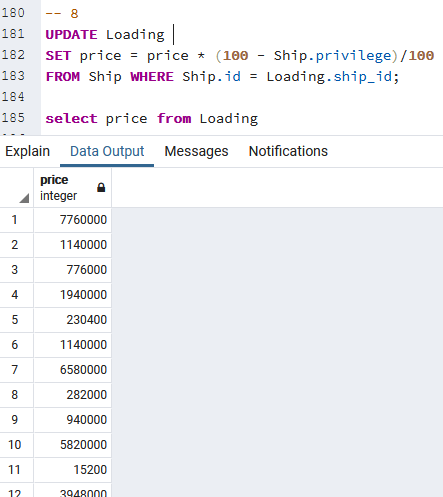
Группирую по имени судна. По сути мы получаем много строк с именем судна и портом. После разбиваем их на группы по имени судна. Применяем функцию подсчета в каждой группе по столбцу порт и оставляем те, которых больше единицы.

Отсортировываю по убыванию количества портов



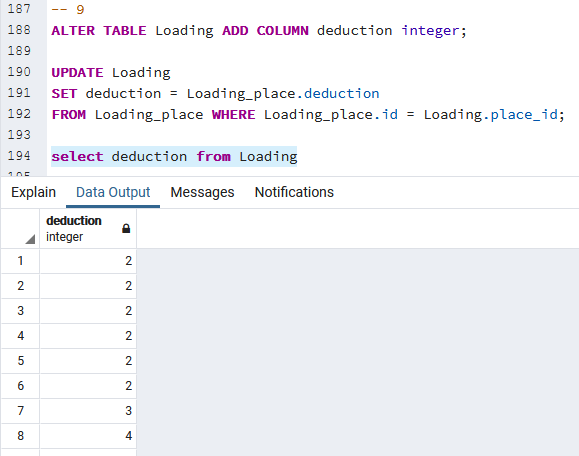
1. Создать запрос для модификации всех значений столбца с суммарной величиной стоимости таблицы погрузка, чтобы он содержал истинную оплачиваемую сумму ( с учетом льгот у судов). Вывести новые значения.

Обновляем таблицу Погруззки. Устанавливаем в столбец стоимости нужные данные. Нужные данные берем из таблицы судна сопоставив id обоих таблиц



1. Расширить таблицу с данными о погрузке столбцом, содержащим величины отчислений на погрузку. Создать запрос для ввода конкретных значений во все строки таблицы

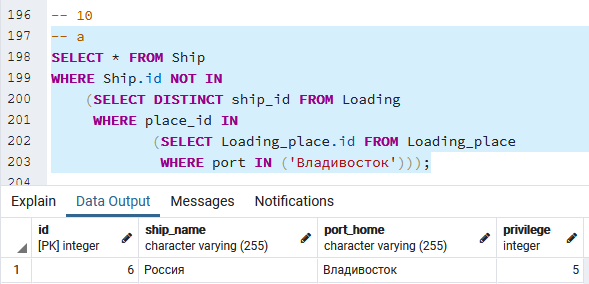
Добавляю столбец в таблицу Погрузки. Обновляю данные.



Уровень 2

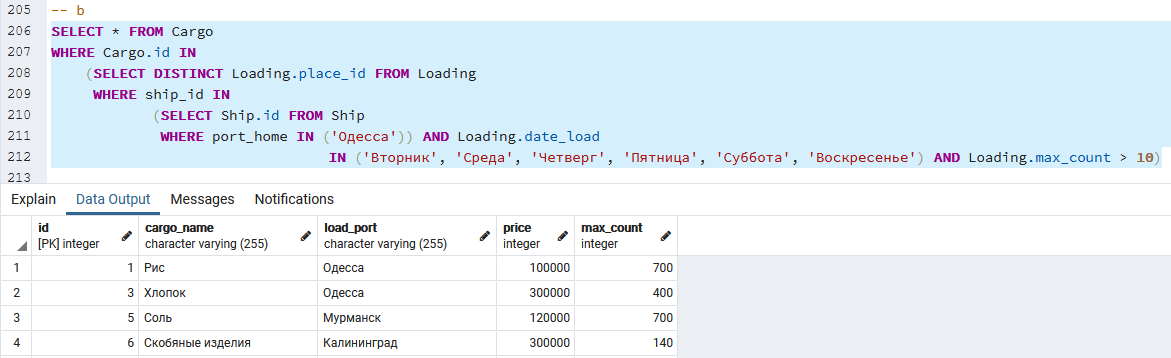
1. Используя операцию IN (NOT IN) реализовать следующие запросы:
   1. найти суда не производившие погрузки на причалах Владивостока;

Из таблицы место погрузки мы выстакиваем id всех портов с названием Владивосток. Из получившихся данных мы вытаскивем id Судов через таблицу Загрузки. Из полученных данных смотрим какого судна там нет. In проверяет совпадает ли наше значение со значениями в списке. В нашем случае нужно, чтобы не совпадало, поэтому добавляем отрицание NOT. То есть если какого-то судна нет в списке, то мы его выводим.



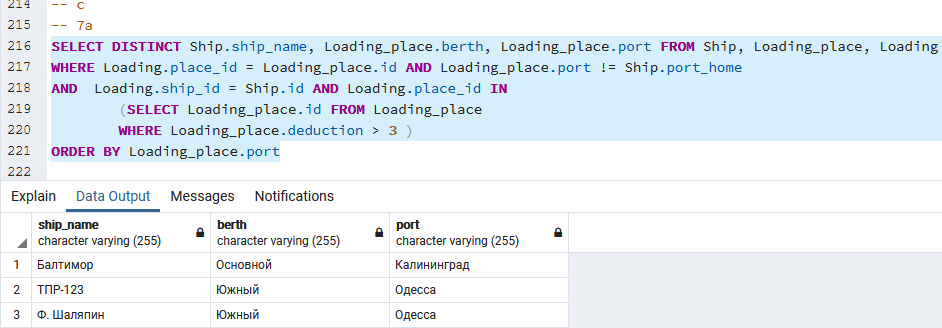
* 1. найти грузы, которые грузили на суда из Одессы после понедельника в количестве более 10 единиц;

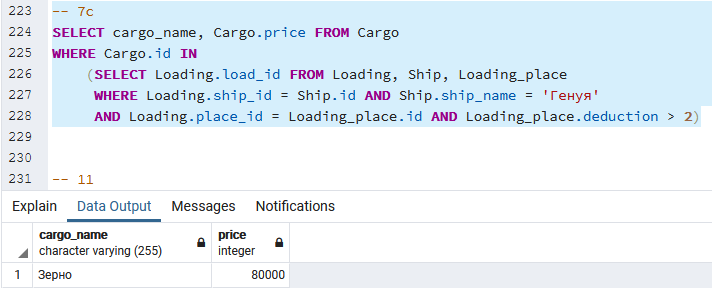
То же самое, только условия другие.



* 1. запрос задания 7.а и 7.с.

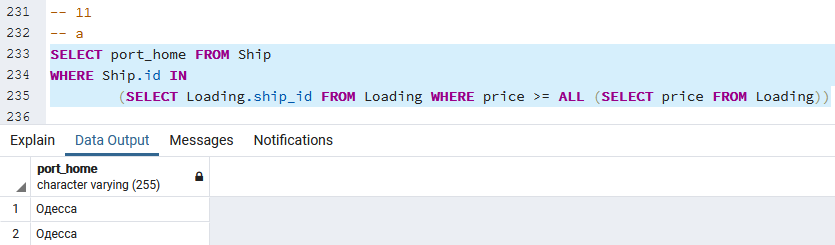
Как 7а/c, только с IN и вместо join, условие в where.





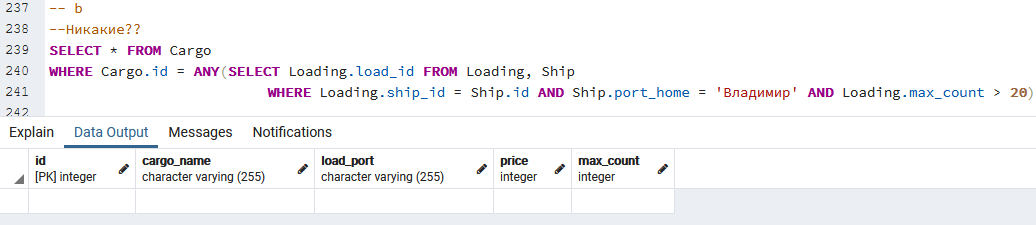
1. Используя операции ALL-ANY реализовать следующие запросы:
   1. определить порт приписки судна, производившего самую дорогую погрузку в среду;

Условие стоимость больше или равно всех

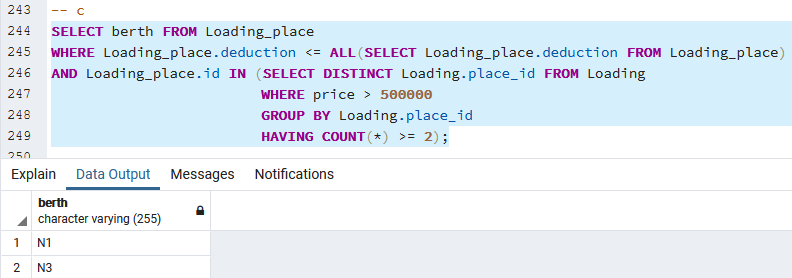


* 1. какие грузы грузились на суда из Владивостока в количестве более 20 единиц;

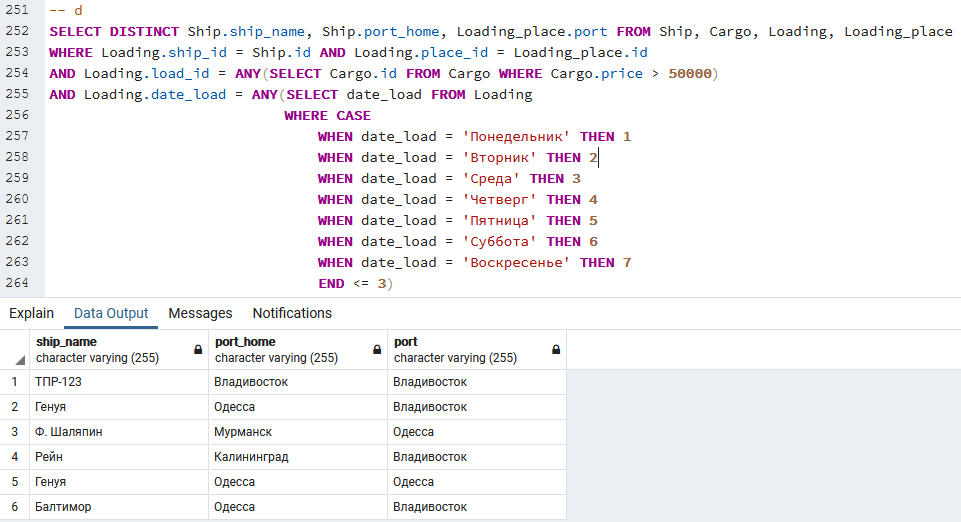
Больше 20 нет. Ну можно добавить)



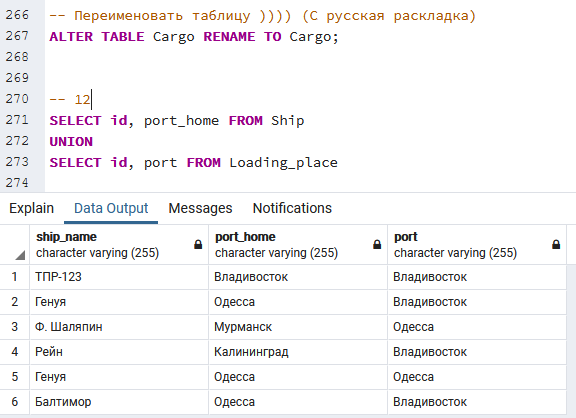
* 1. найти причал с наименьшими отчислениями среди тех, на кторых проходила погрузка со стоимостью более 500000руб. не менее двух раз;



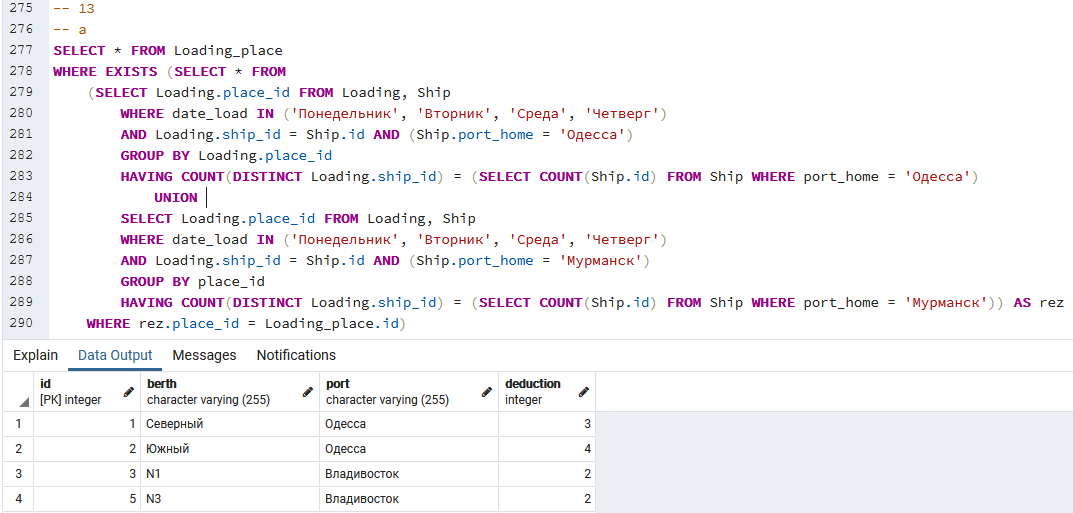
* 1. запрос задания 7.b.



1. Используя операцию UNION получить порты приписки судов и места складирования грузов.

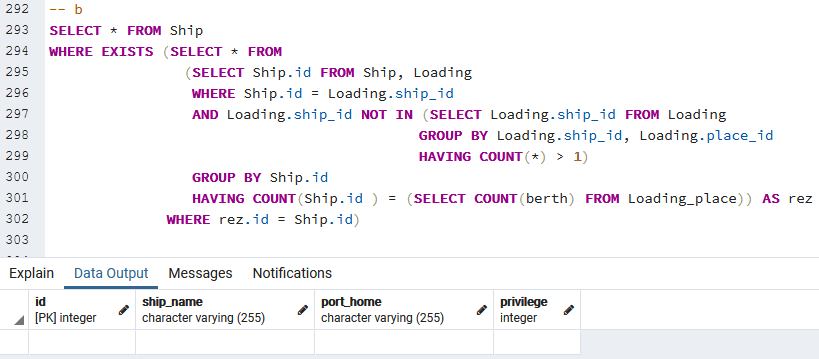


1. Используя операцию EXISTS ( NOT EXISTS ) реализовать нижеследующие запросы. В случае, если для текущего состояния БД запрос будет выдавать пустое множество строк, требуется указать какие добавления в БД необходимо провести.
   1. найти места погрузки, на которых проходила погрузка всех судов из Одессы или Мурманска до пятницы;



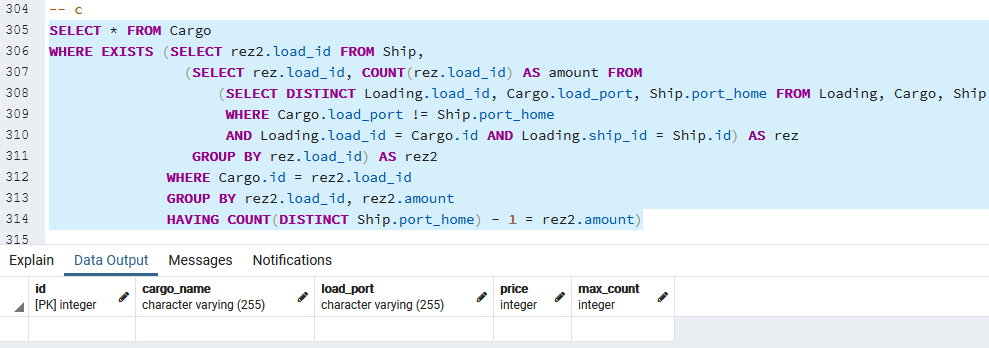
* 1. какие суда проходили погрузку на всех причалах не более раза;

Никакие



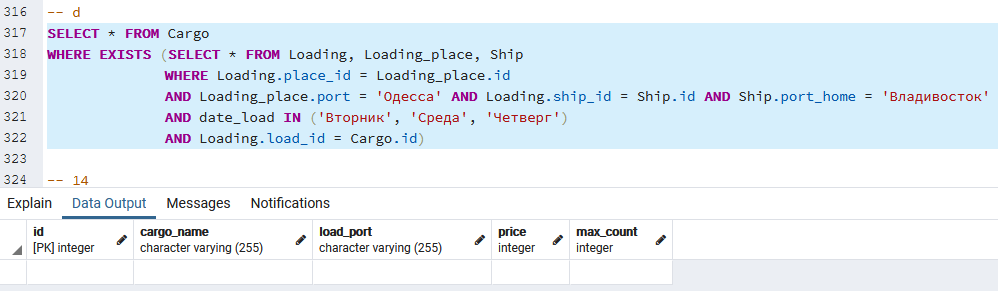
* 1. найти грузы, которые грузились на все суда с иным портом приписки чем порт складирования этого груза;

Никакие. Надо добавить в таблицу погрузка какие-нибудь груза, которые грузили в большенство судов, но не грузили в судно с таким же портом приписки. Например Скобяные изделия еще в на два каких-нибудь судна, кроме калининградского.

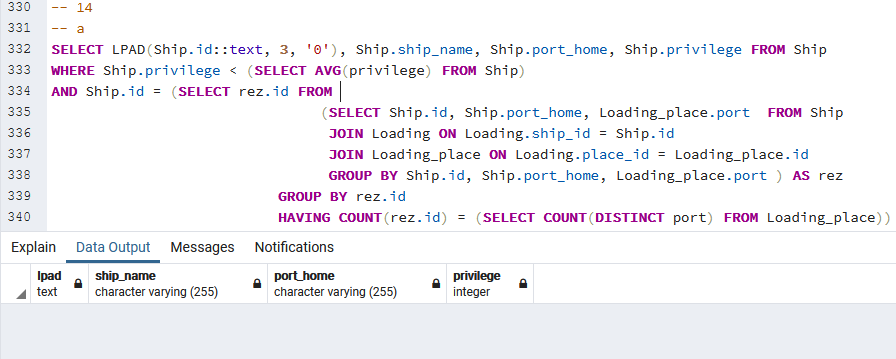


* 1. какие грузы грузлись на суда из Владивостока только причалах Одесского порта со вторника по четверг.

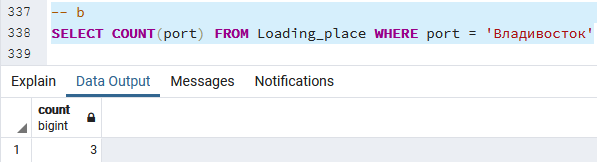
Никакие.



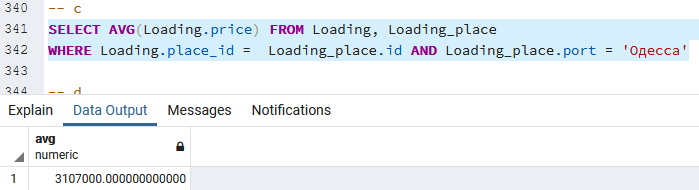
1. Реализовать запросы с использованием аггрегатных функций:
   1. найти среди тех судов, которые производили погрузку во всех портах, такие, которые имеют льготы меньше среднего;



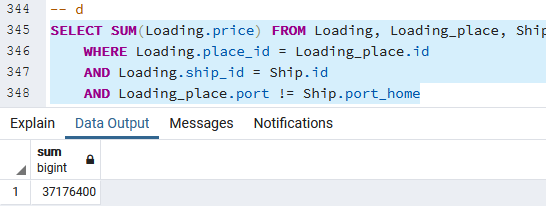
* 1. определить число причалов в порту Владивосток;



* 1. найти среднюю стоимость погрузки на причалах Одесского порта;

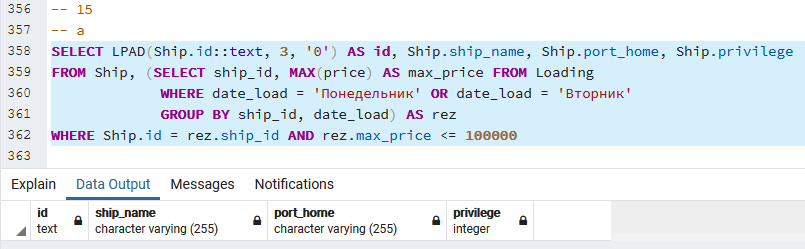


* 1. найти суммарную стоимость погрузок, произведенных судами в чужих портах.

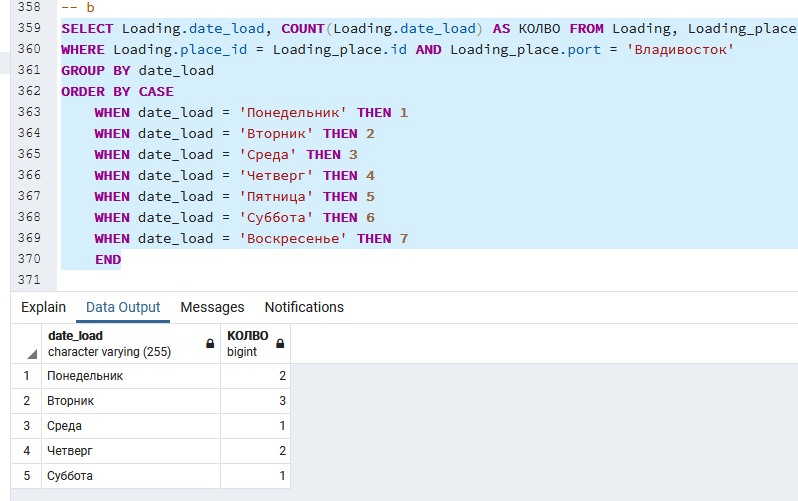


1. Используя средства группировки реализовать следующие запросы:
   1. определить суда, у которых стоимость самой дорогой погрузки в понедельник или во вторник не превышала 100000;

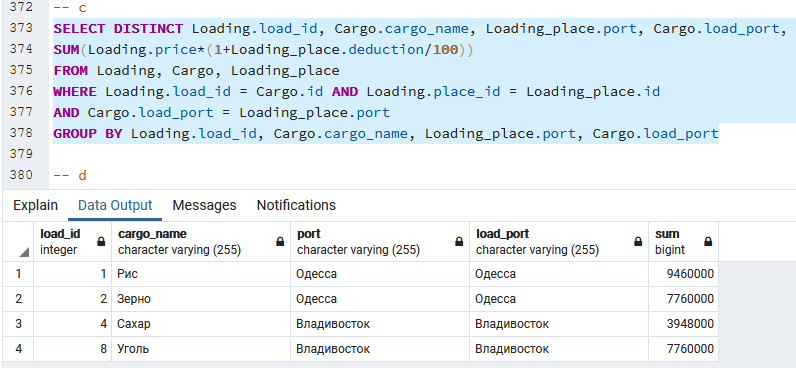
Никакие.



* 1. определить для каждого дня недели число произведенных погрузок на причалах Владивостока;



* 1. для каждой пары “груз-место погрузки” с совпадающим портом складирования и портом погрузки, вывести суммарную величину стоимостей погрузок;



* 1. для каждого груза определить количество различных судов, на которые грузили этот груз после понедельника.

