ВАРИАНТ 7. ВЫПОЛНИЛА МАРТЫНЮК ОЛЕСЯ, 18ПИ-2

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 |

*CREATE TABLE vessel (*

*Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР*

*Name text NOT NULL, -- НАЗВАНИЕ (выражается строковыми данными)*

*Port text NOT NULL, -- ПОРТ ПРИПИСКИ (выражается строковыми данными)*

*Exemption integer NOT NULL, CHECK (Exemption >=0 ) -- ЛЬГОТА (выражается числовыми данными, не может быть < 0)*

*);*

*CREATE TABLE places ( -- МЕСТА ПОГРУЗКИ*

*Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР*

*Pier text NOT NULL, -- ПРИЧАЛ (выражается строковыми данными)*

*Port text NOT NULL, -- ПОРТ (выражается строковыми данными)*

*Allocation integer NOT NULL, CHECK (Allocation >= 0) -- ОТЧИСЛЕНИЯ НА ПОГРУЗКУ (выражается числовыми данными, не может быть < 0)*

*);*

*CREATE TABLE cargo ( -- ГРУЗ*

*Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР*

*Name text NOT NULL, -- НАЗВАНИЕ (выражается строковыми данными)*

*Port text NOT NULL, -- ПОРТ СКЛАДИРОВАНИЯ (выражается строковыми данными)*

*Price integer NOT NULL, CHECK (Price >= 0), -- СТОИМОСТЬ, РУБ (выражается числовыми данными, не может быть < 0)*

*Amount integer NOT NULL, CHECK (Amount > 0) -- МАКС. КОЛ-ВО (выражается числовыми данными, не может быть <= 0)*

*);*

*CREATE TABLE shipment ( -- ПОГРУЗКА*

*Register SERIAL PRIMARY KEY, -- НОМЕР ВЕДОМОСТИ*

*Weekday text NOT NULL, -- ДАТА (выражается строковыми данными)*

*Vessel integer NOT NULL, CHECK (Vessel > 0), -- СУДНО (выражается числовыми данными, не может быть <= 0)*

*Place integer NOT NULL CHECK (Place > 0), -- МЕСТО ПОГРУЗКИ (выражается числовыми данными, не может быть <= 0)*

*Cargo integer NOT NULL, CHECK (Cargo > 0), -- ГРУЗ (выражается числовыми данными, не может быть <= 0)*

*Amount integer NOT NULL, CHECK (Amount > 0), -- КОЛ-ВО (выражается числовыми данными, не может быть <= 0)*

*Price integer NOT NULL -- СТОИМОСТЬ, РУБ (выражается числовыми данными)*

*)*

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

*INSERT INTO vessel (id, name, port, exemption) VALUES (001, 'Балтимор', 'Одесса', 3);*

*INSERT INTO vessel (id, name, port, exemption) VALUES (002, 'Генуя', 'Одесса', 3);*

*INSERT INTO vessel (id, name, port, exemption) VALUES (003, 'ТПР-123', 'Владивосток', 5);*

*INSERT INTO vessel (id, name, port, exemption) VALUES (004, 'Ф. Шаляпин', 'Мурманск', 6);*

*INSERT INTO vessel (id, name, port, exemption) VALUES (005, 'Рейн', 'Калининград', 4);*

*INSERT INTO vessel (id, name, port, exemption) VALUES (006, 'Россия', 'Владивосток', 5);*

*INSERT INTO places (id, pier, port, allocation) VALUES (001, 'Северный', 'Одесса', 3);*

*INSERT INTO places (id, pier, port, allocation) VALUES (002, 'Южный', 'Одесса', 4);*

*INSERT INTO places (id, pier, port, allocation) VALUES (003, 'N1', 'Владивосток', 2);*

*INSERT INTO places (id, pier, port, allocation) VALUES (004, 'N2', 'Владивосток', 2);*

*INSERT INTO places (id, pier, port, allocation) VALUES (005, 'N3', 'Владивосток', 2);*

*INSERT INTO places (id, pier, port, allocation) VALUES (006, 'Основной', 'Калининград', 4);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (001, 'Рис', 'Одесса', 100000, 700);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (002, 'Зерно', 'Одесса', 80000, 890);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (003, 'Хлопок', 'Одесса', 300000, 400);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (004, 'Сахар', 'Владивосток', 140000, 600);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (005, 'Соль', 'Мурманск', 120000, 700);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (006, 'Скобяные изделия', 'Калининград', 300000, 140);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (007, 'Древесина', 'Мурманск', 400000, 260);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (008, 'Уголь', 'Владивосток', 400000, 400);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70204, 'Понедельник', 001, 005, 002, 100, 8000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70205, 'Понедельник', 003, 003, 006, 4, 1200000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70206, 'Вторник', 001, 005, 007, 2, 800000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70207, 'Вторник', 002, 005, 001, 20, 2000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70208, 'Вторник', 005, 005, 002, 3, 240000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70209, 'Среда', 003, 003, 006, 4, 1200000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70210, 'Среда', 004, 001, 001, 70, 7000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70211, 'Среда', 004, 002, 006, 1, 300000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70212, 'Среда', 004, 002, 001, 10, 1000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70213, 'Четверг', 001, 006, 003, 20, 6000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70214, 'Четверг', 003, 004, 002, 2, 16000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70215, 'Четверг', 004, 003, 004, 30, 4200000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70216, 'Суббота', 003, 002, 005, 10, 1200000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70217, 'Суббота', 002, 003, 008, 20, 8000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70218, 'Суббота', 001, 001, 001, 20, 2000000);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70219, 'Суббота', 005, 006, 004, 10, 1400000)*

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

Далее я решила изменить названия таблиц и их столбцов на те, что были в исходном файле. Сделала я это с помощью команд:

*ALTER TABLE vessel RENAME COLUMN exemption TO ЛЬГОТА;*

*ALTER TABLE vessel RENAME TO СУДНО;*

и тд. Полный список команд в файлах.

*SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ\_ПРИПИСКИ, ЛЬГОТА FROM СУДНО*

Изображение выглядит как снимок экрана

Автоматически созданное описание

*SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, ПРИЧАЛ, ПОРТ, ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ FROM МЕСТА\_ПОГРУЗКИ*

Изображение выглядит как снимок экрана

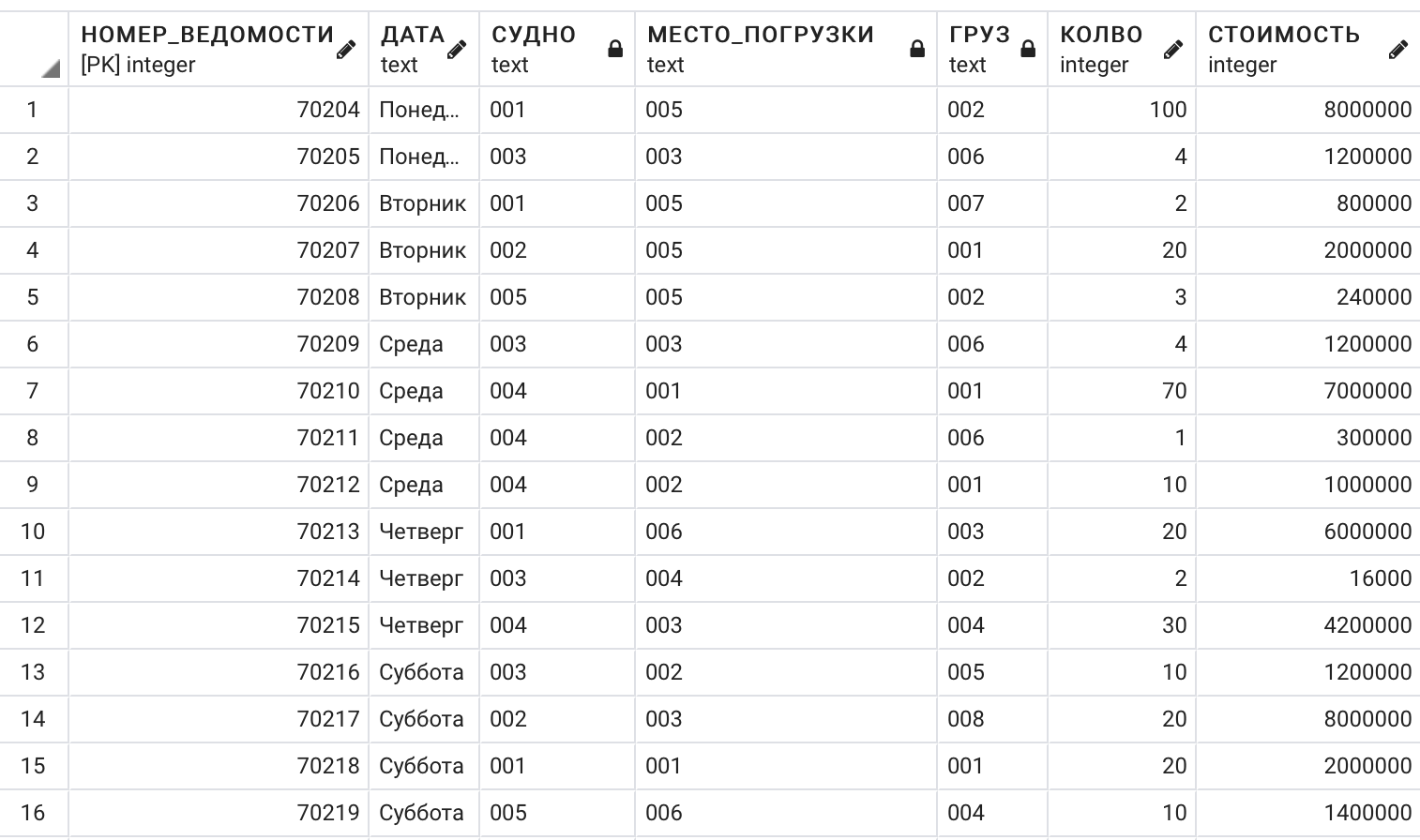
Автоматически созданное описание

*SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ\_СКЛАДИРОВАНИЯ, СТОИМОСТЬ, МАКС\_КОЛВО FROM ГРУЗ*

Изображение выглядит как снимок экрана

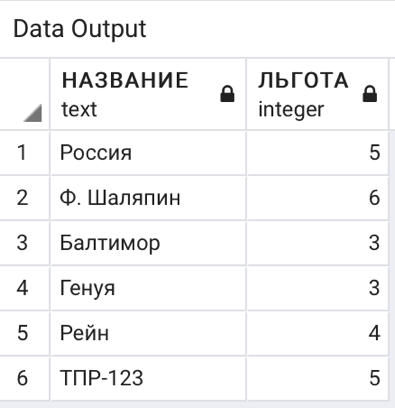
Автоматически созданное описание

*SELECT НОМЕР\_ВЕДОМОСТИ, ДАТА, LPAD(СУДНО::text, 3, '0') as СУДНО, LPAD(МЕСТО\_ПОГРУЗКИ::text, 3, '0') as МЕСТО\_ПОГРУЗКИ, LPAD(ГРУЗ::text, 3, '0') as ГРУЗ, КОЛВО, СТОИМОСТЬ FROM ПОГРУЗКА*



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

*SELECT DISTINCT НАЗВАНИЕ, ЛЬГОТА FROM СУДНО*



* 1. **всех различных адресов приписки судов;**

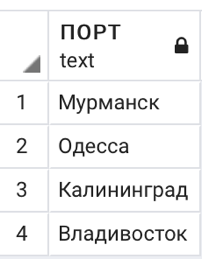
*SELECT DISTINCT ПОРТ\_ПРИПИСКИ FROM СУДНО*

Изображение выглядит как снимок экрана

Автоматически созданное описание

* 1. **всех различных портов**

*SELECT DISTINCT ПОРТ FROM МЕСТА\_ПОГРУЗКИ UNION SELECT ПОРТ\_СКЛАДИРОВАНИЯ FROM ГРУЗ UNION SELECT ПОРТ\_ПРИПИСКИ FROM СУДНО*



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

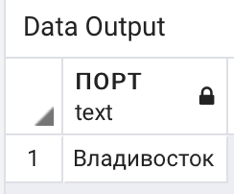
*SELECT НАЗВАНИЕ, СТОИМОСТЬ FROM ГРУЗ WHERE МАКС\_КОЛВО < 500*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

*SELECT DISTINCT ПОРТ FROM МЕСТА\_ПОГРУЗКИ WHERE ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ > 5 OR ПРИЧАЛ LIKE '%N%'*



* 1. **названия судов из Одессы.**

*SELECT DISTINCT НАЗВАНИЕ FROM СУДНО WHERE ПОРТ\_ПРИПИСКИ = 'Одесса'*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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**;

*SELECT НОМЕР\_ВЕДОМОСТИ, ДАТА, НАЗВАНИЕ, СТОИМОСТЬ FROM ПОГРУЗКА, СУДНО WHERE*

*ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР*

*ORDER BY CASE*

*WHEN ДАТА = 'Понедельник' THEN 1*

*WHEN ДАТА = 'Вторник' THEN 2*

*WHEN ДАТА = 'Среда' THEN 3*

*WHEN ДАТА = 'Четверг' THEN 4*

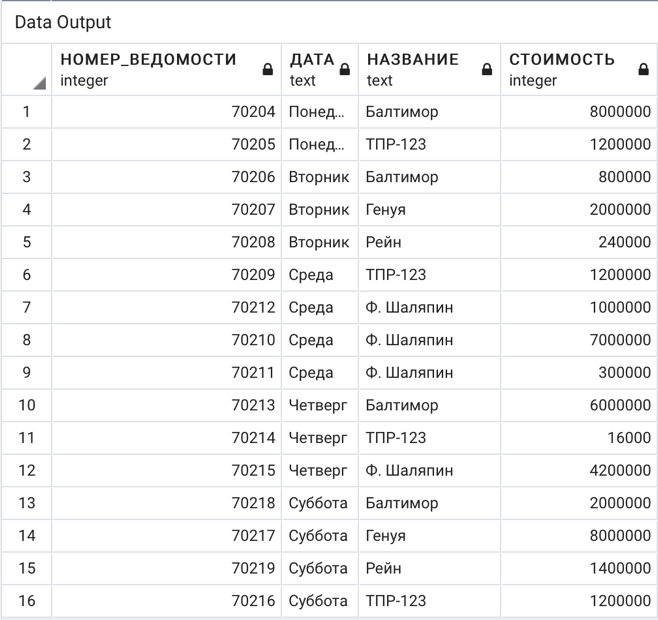
*WHEN ДАТА = 'Пятница' THEN 5*

*WHEN ДАТА = 'Суббота' THEN 6*

*WHEN ДАТА = 'Воскресенье' THEN 7*

*END*

*, НАЗВАНИЕ*

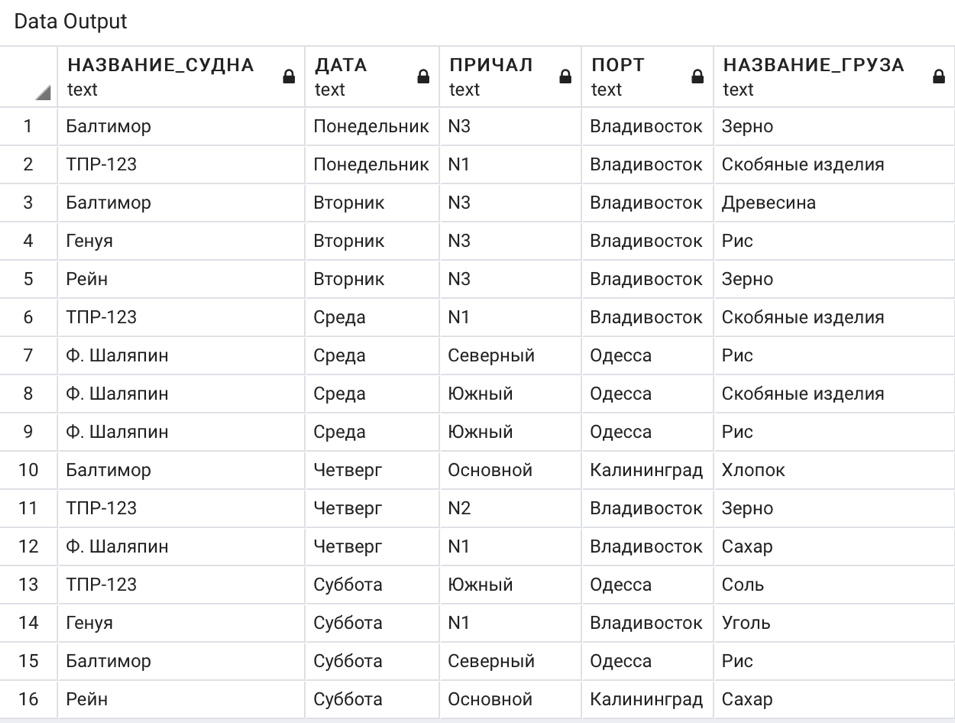


* 1. **название судна, дата, название места погрузки, груз.**

Если под «местом погрузки» подразумевались **и причал, и порт,** то запрос такой:

*SELECT СУДНО.НАЗВАНИЕ AS НАЗВАНИЕ\_СУДНА, ПОГРУЗКА.ДАТА, МЕСТА\_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА\_ПОГРУЗКИ.ПОРТ, ГРУЗ.НАЗВАНИЕ AS НАЗВАНИЕ\_ГРУЗА FROM СУДНО, ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ, ГРУЗ WHERE*

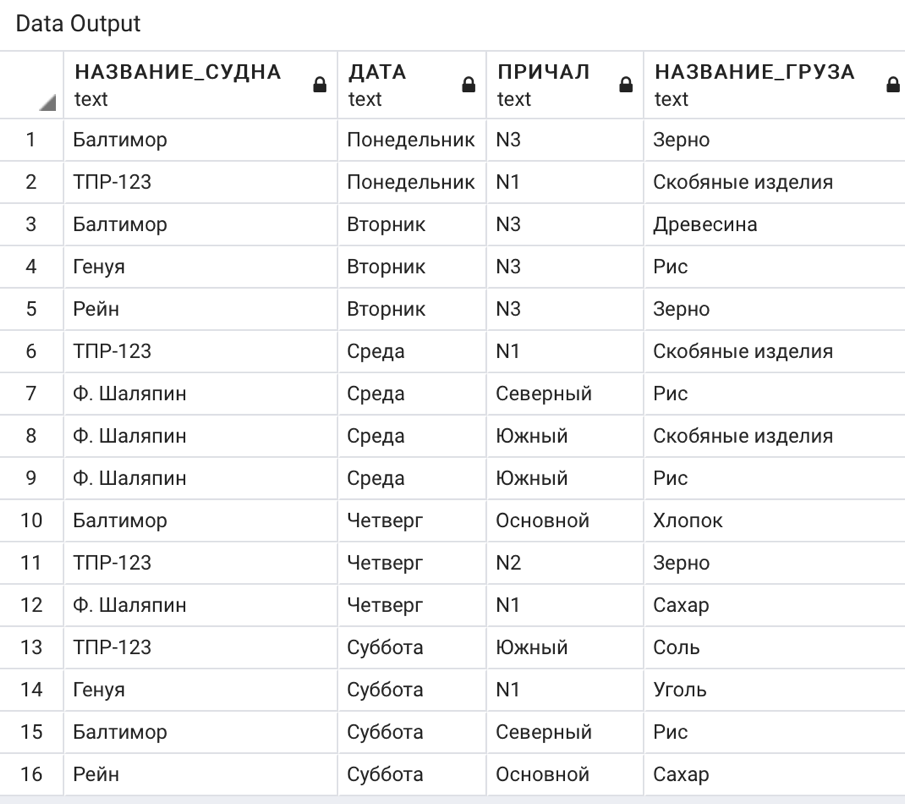
*ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР*



Если под «местом погрузки» подразумевался **только причал**, то запрос такой:

*SELECT СУДНО.НАЗВАНИЕ AS НАЗВАНИЕ\_СУДНА, ПОГРУЗКА.ДАТА, МЕСТА\_ПОГРУЗКИ.ПРИЧАЛ, ГРУЗ.НАЗВАНИЕ AS НАЗВАНИЕ\_ГРУЗА FROM СУДНО, ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ, ГРУЗ WHERE*

*ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР*



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. Отсортировать по портам**;

При таком запросе:

*SELECT СУДНО.НАЗВАНИЕ, МЕСТА\_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА\_ПОГРУЗКИ.ПОРТ FROM*

*СУДНО, МЕСТА\_ПОГРУЗКИ, ПОГРУЗКА WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ\_ПРИПИСКИ AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ > 3 ORDER BY МЕСТА\_ПОГРУЗКИ.ПОРТ*

Получились следующие данные:

Изображение выглядит как снимок экрана

Автоматически созданное описание

Ф. Шаляпин **дублируется**, так как везет в Южный два различных груза. Чтобы убрать дублирующиеся строки, я изменила запрос:

*SELECT DISTINCT СУДНО.НАЗВАНИЕ, МЕСТА\_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА\_ПОГРУЗКИ.ПОРТ FROM*

*СУДНО, МЕСТА\_ПОГРУЗКИ, ПОГРУЗКА WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ\_ПРИПИСКИ AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ > 3 ORDER BY МЕСТА\_ПОГРУЗКИ.ПОРТ*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

В базе данных нет информации про месяцы погрузки. Для сохранения логики, изменю условие на «не позднее среды». В таком случае получится следующий запрос:

*SELECT DISTINCT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ\_ПРИПИСКИ, МЕСТА\_ПОГРУЗКИ.ПОРТ AS ПОРТ\_ПОГРУЗКИ FROM*

*СУДНО, ГРУЗ, ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР AND ГРУЗ.СТОИМОСТЬ > 50000 AND CASE*

*WHEN ПОГРУЗКА.ДАТА = 'Понедельник' THEN 1*

*WHEN ПОГРУЗКА.ДАТА = 'Вторник' THEN 2*

*WHEN ПОГРУЗКА.ДАТА = 'Среда' THEN 3*

*WHEN ПОГРУЗКА.ДАТА = 'Четверг' THEN 4*

*WHEN ПОГРУЗКА. ДАТА = 'Пятница' THEN 5*

*WHEN ПОГРУЗКА. ДАТА = 'Суббота' THEN 6*

*WHEN ПОГРУЗКА.ДАТА = 'Воскресенье' THEN 7*

*END <= 3 AND ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

*SELECT ГРУЗ.НАЗВАНИЕ, ГРУЗ.СТОИМОСТЬ FROM ГРУЗ, СУДНО, ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ WHERE*

*ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.НАЗВАНИЕ = 'Генуя' AND ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ > 2*

*Изображение выглядит как снимок экрана, птица

Автоматически созданное описание*

Data Output пуст, так как «Геную» (ID = 002) грузили на причалах с ID = 005 и ID = 003. На этих причалах отчисления на погрузку = 2.

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

*SELECT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ\_ПРИПИСКИ FROM СУДНО WHERE СУДНО.ИДЕНТИФИКАТОР IN (SELECT СУДНО FROM (SELECT DISTINCT СУДНО, (SELECT ПОРТ FROM МЕСТА\_ПОГРУЗКИ WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ=МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР) AS PORT FROM ПОГРУЗКА) AS SUBQ GROUP BY СУДНО HAVING COUNT(\*) > 1)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

Уровень 2

1. Используя операцию IN (NOT IN) реализовать следующие запросы:
   1. найти суда не производившие погрузки на причалах Владивостока;
   2. найти грузы, которые грузили на суда из Одессы после понедельника в количестве более 10 единиц;
   3. запрос задания 7.а и 7.с.
2. Используя операции ALL-ANY реализовать следующие запросы:
   1. определить порт приписки судна, производившего самую дорогую погрузку в среду;
   2. какие грузы грузились на суда из Владивостока в количестве более 20 единиц;
   3. найти причал с наименьшими отчислениями среди тех, на кторых проходила погрузка со стоимостью более 500000руб. не менее двух раз;
   4. запрос задания 7.b.
3. Используя операцию UNION получить порты приписки судов и места складирования грузов.
4. 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;
   2. какие суда проходили погрузку на всех причалах не более раза;
   3. найти грузы, которые грузились на все суда с иным портом приписки чем порт складирования этого груза;
   4. какие грузы грузлись на суда из Владивостока только причалах Одесского порта со вторника по четверг.
5. Реализовать запросы с использованием аггрегатных функций:
   1. найти среди тех судов, которые производили погрузку во всех портах, такие, которые имеют льготы меньше среднего;
   2. определить число причалов в порту Владивосток;
   3. найти среднюю стоимость погрузки на причалах Одесского порта;
   4. найти суммарную стоимость погрузок, произведенных судами в чужих портах.
6. Используя средства группировки реализовать следующие запросы:
   1. определить суда, у которых стоимость самой дорогой погрузки в понедельник или во вторник не превышала 100000;
   2. определить для каждого дня недели число произведенных погрузок на причалах Владивостока;
   3. для каждой пары “груз-место погрузки” с совпадающим портом складирования и портом погрузки, вывести суммарную величину стоимостей погрузок;
   4. для каждого груза определить количество различных судов, на которые грузили этот груз после понедельника.