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)*

*)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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), (002, 'Генуя', 'Одесса', 3), (003, 'ТПР-123', 'Владивосток', 5), (004, 'Ф. Шаляпин', 'Мурманск', 6), (005, 'Рейн', 'Калининград', 4), (006, 'Россия', 'Владивосток', 5);*

*INSERT INTO places (id, pier, port, allocation) VALUES (001, 'Северный', 'Одесса', 3), (002, 'Южный', 'Одесса', 4), (003, 'N1', 'Владивосток', 2), (004, 'N2', 'Владивосток', 2), (005, 'N3', 'Владивосток', 2), (006, 'Основной', 'Калининград', 4);*

*INSERT INTO cargo (id, name, port, price, amount) VALUES (001, 'Рис', 'Одесса', 100000, 700), (002, 'Зерно', 'Одесса', 80000, 890), (003, 'Хлопок', 'Одесса', 300000, 400), (004, 'Сахар', 'Владивосток', 140000, 600), (005, 'Соль', 'Мурманск', 120000, 700), (006, 'Скобяные изделия', 'Калининград', 300000, 140), (007, 'Древесина', 'Мурманск', 400000, 260), (008, 'Уголь', 'Владивосток', 400000, 400);*

*INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70204, 'Понедельник', 001, 005, 002, 100, 8000000), (70205, 'Понедельник', 003, 003, 006, 4, 1200000), (70206, 'Вторник', 001, 005, 007, 2, 800000), (70207, 'Вторник', 002, 005, 001, 20, 2000000), (70208, 'Вторник', 005, 005, 002, 3, 240000), (70209, 'Среда', 003, 003, 006, 4, 1200000), (70210, 'Среда', 004, 001, 001, 70, 7000000), (70211, 'Среда', 004, 002, 006, 1, 300000), (70212, 'Среда', 004, 002, 001, 10, 1000000), (70213, 'Четверг', 001, 006, 003, 20, 6000000), (70214, 'Четверг', 003, 004, 002, 2, 16000), (70215, 'Четверг', 004, 003, 004, 30, 4200000), (70216, 'Суббота', 003, 002, 005, 10, 1200000), (70217, 'Суббота', 002, 003, 008, 20, 8000000), (70218, 'Суббота', 001, 001, 001, 20, 2000000), (70219, 'Суббота', 005, 006, 004, 10, 1400000)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

*SELECT DISTINCT НАЗВАНИЕ, ЛЬГОТА FROM СУДНО*

Изображение выглядит как снимок экрана

Автоматически созданное описание

* 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*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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%'*

Изображение выглядит как снимок экрана

Автоматически созданное описание

* 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*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

* 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*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

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 SUBQ.НАЗВАНИЕ, SUBQ.ПОРТ\_ПРИПИСКИ FROM ( SELECT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ\_ПРИПИСКИ, МЕСТА\_ПОГРУЗКИ.ПОРТ FROM СУДНО JOIN ПОГРУЗКА ON ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР*

*JOIN МЕСТА\_ПОГРУЗКИ ON ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР*

*GROUP BY СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ\_ПРИПИСКИ, МЕСТА\_ПОГРУЗКИ.ПОРТ*

*) AS SUBQ GROUP BY SUBQ.НАЗВАНИЕ, SUBQ.ПОРТ\_ПРИПИСКИ HAVING COUNT(SUBQ.НАЗВАНИЕ) > 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.**

*UPDATE ПОГРУЗКА SET СТОИМОСТЬ = СТОИМОСТЬ \* (100 - СУДНО.ЛЬГОТА)/100 FROM СУДНО WHERE СУДНО.ИДЕНТИФИКАТОР = ПОГРУЗКА.СУДНО;*

*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*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

ALTER TABLE ПОГРУЗКА ADD ОТЧИСЛЕНИЯ integer CHECK (ОТЧИСЛЕНИЯ >= 0);

UPDATE ПОГРУЗКА SET ОТЧИСЛЕНИЯ = ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ FROM МЕСТА\_ПОГРУЗКИ WHERE МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР = ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ

Изображение выглядит как снимок экрана

Автоматически созданное описание

Уровень 2

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

*SELECT \* FROM СУДНО WHERE ИДЕНТИФИКАТОР NOT IN (SELECT DISTINCT СУДНО FROM ПОГРУЗКА WHERE МЕСТО\_ПОГРУЗКИ IN (SELECT ИДЕНТИФИКАТОР FROM МЕСТА\_ПОГРУЗКИ WHERE ПОРТ IN ('Владивосток')))*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

Это единственное судно, которое ни разу не проводило погрузки на причалах Владивостока (ID причалов: 003, 004, 005)

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

*SELECT \* FROM ГРУЗ WHERE ИДЕНТИФИКАТОР IN (SELECT DISTINCT ГРУЗ FROM ПОГРУЗКА WHERE СУДНО IN (SELECT ИДЕНТИФИКАТОР FROM СУДНО WHERE ПОРТ\_ПРИПИСКИ IN ('Одесса')) AND ДАТА IN ('Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота', 'Воскресенье') AND КОЛВО > 10)*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

* 1. **запрос задания 7.а и 7.с.**

7.a: **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**;

*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*

*ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ IN (SELECT ИДЕНТИФИКАТОР FROM МЕСТА\_ПОГРУЗКИ WHERE ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ > 3 )ORDER BY МЕСТА\_ПОГРУЗКИ.ПОРТ*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

*SELECT НАЗВАНИЕ, СТОИМОСТЬ FROM ГРУЗ WHERE ИДЕНТИФИКАТОР IN*

*(SELECT ГРУЗ FROM ПОГРУЗКА, СУДНО, 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

Автоматически созданное описание

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

*SELECT DISTINCT ПОРТ\_ПРИПИСКИ FROM СУДНО WHERE ИДЕНТИФИКАТОР IN (SELECT СУДНО FROM ПОГРУЗКА WHERE СТОИМОСТЬ >= ALL (SELECT СТОИМОСТЬ FROM ПОГРУЗКА))*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

*SELECT \* FROM ГРУЗ WHERE ИДЕНТИФИКАТОР = ANY(SELECT ГРУЗ FROM ПОГРУЗКА, СУДНО WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.ПОРТ\_ПРИПИСКИ = 'Владимир' AND КОЛВО > 20)*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

Data Output пуст, так как грузы в количестве > 20 грузились только на суды из Одессы и Мурманска

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

*SELECT ПРИЧАЛ FROM МЕСТА\_ПОГРУЗКИ WHERE*

*ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ <= ALL(SELECT ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ FROM МЕСТА\_ПОГРУЗКИ) AND*

*ИДЕНТИФИКАТОР IN (SELECT DISTINCT МЕСТО\_ПОГРУЗКИ FROM ПОГРУЗКА WHERE СТОИМОСТЬ > 500000 GROUP BY МЕСТО\_ПОГРУЗКИ HAVING COUNT(\*) >= 2)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

* 1. **запрос задания 7.b.**

**7.b.: 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.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND*

*ПОГРУЗКА.ГРУЗ = ANY(SELECT ИДЕНТИФИКАТОР FROM ГРУЗ WHERE СТОИМОСТЬ > 50000) AND*

*ПОГРУЗКА.ДАТА = ANY(SELECT ДАТА FROM ПОГРУЗКА WHERE 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)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

Если требуется порты, где **и приписка судов, и место складирования грузов**:

*SELECT ПОРТ\_ПРИПИСКИ AS ПОРТ\_ПОГРУЗКИ\_ПРИПИСКИ FROM СУДНО*

*UNION*

*SELECT ПОРТ FROM МЕСТА\_ПОГРУЗКИ*

Изображение выглядит как снимок экрана

Автоматически созданное описание

Если требуется **всё вместе** (но зачем?), то так:

*SELECT ИДЕНТИФИКАТОР, ПОРТ\_ПРИПИСКИ AS ПОРТЫ FROM СУДНО*

*UNION*

*SELECT ИДЕНТИФИКАТОР, ПОРТ FROM МЕСТА\_ПОГРУЗКИ*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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;**

*SELECT \* FROM МЕСТА\_ПОГРУЗКИ WHERE EXISTS (*

*SELECT \* FROM (SELECT МЕСТО\_ПОГРУЗКИ FROM ПОГРУЗКА, СУДНО*

*WHERE ПОГРУЗКА.ДАТА IN ('Понедельник', 'Вторник', 'Среда', 'Четверг')*

*AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND (СУДНО.ПОРТ\_ПРИПИСКИ = 'Одесса')*

*GROUP BY МЕСТО\_ПОГРУЗКИ*

*HAVING COUNT(DISTINCT ПОГРУЗКА.СУДНО) = (SELECT COUNT(ИДЕНТИФИКАТОР) FROM СУДНО WHERE ПОРТ\_ПРИПИСКИ = 'Одесса')*

*UNION SELECT МЕСТО\_ПОГРУЗКИ FROM ПОГРУЗКА, СУДНО*

*WHERE ПОГРУЗКА.ДАТА IN ('Понедельник', 'Вторник', 'Среда', 'Четверг')*

*AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND (СУДНО.ПОРТ\_ПРИПИСКИ = 'Мурманск')*

*GROUP BY МЕСТО\_ПОГРУЗКИ*

*HAVING COUNT(DISTINCT ПОГРУЗКА.СУДНО) = (SELECT COUNT(ИДЕНТИФИКАТОР) FROM СУДНО WHERE ПОРТ\_ПРИПИСКИ = 'Мурманск'))*

*AS QBAR WHERE QBAR.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР*

*)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

Если идти изнутри: сначала я выбирала те ID судов, у которых было несколько погрузок на одни и те же причалы. Потом выбирала те суды, которые не входили в этот список. Потом выбирала ID, который повторялся такое же количество раз, сколько существует причалов. Запихнула все в exists и готово. Можно было сделать проще, но я не додумалась, как.

*SELECT \* FROM СУДНО WHERE EXISTS (*

*SELECT \* FROM (*

*SELECT СУДНО.ИДЕНТИФИКАТОР FROM СУДНО, ПОГРУЗКА*

*WHERE СУДНО.ИДЕНТИФИКАТОР = ПОГРУЗКА.СУДНО AND*

*ПОГРУЗКА.СУДНО NOT IN*

*(*

*SELECT ПОГРУЗКА.СУДНО FROM ПОГРУЗКА*

*GROUP BY ПОГРУЗКА.СУДНО, ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ*

*HAVING COUNT(\*) > 1*

*)*

*GROUP BY СУДНО.ИДЕНТИФИКАТОР HAVING COUNT(СУДНО.ИДЕНТИФИКАТОР) = (SELECT COUNT(ПРИЧАЛ) FROM МЕСТА\_ПОГРУЗКИ)*

*) AS QBAR WHERE QBAR.ИДЕНТИФИКАТОР = СУДНО.ИДЕНТИФИКАТОР*

*)*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

В результате ничего нет, потому что нет подходящих судов. Добавила временные записи с Генуей и недостающими причалами и появился нужный ответ.

INSERT INTO ПОГРУЗКА (НОМЕР\_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО\_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ)

VALUES (70220, 'Среда', 002, 001, 001, 20, 1940000),

(70221, 'Среда', 002, 002, 001, 20, 1940000),

(70222, 'Среда', 002, 004, 001, 20, 1940000),

(70223, 'Среда', 002, 006, 001, 20, 1940000)

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

SELECT \* FROM ГРУЗ WHERE EXISTS(

SELECT WBAR.ГРУЗ FROM СУДНО, (

SELECT QBAR.ГРУЗ, COUNT(QBAR.ГРУЗ) AS ИТОГ FROM (

SELECT DISTINCT ПОГРУЗКА.ГРУЗ, ГРУЗ.ПОРТ\_СКЛАДИРОВАНИЯ, СУДНО.ПОРТ\_ПРИПИСКИ FROM ПОГРУЗКА, ГРУЗ, СУДНО

WHERE ГРУЗ.ПОРТ\_СКЛАДИРОВАНИЯ != СУДНО.ПОРТ\_ПРИПИСКИ AND

ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР

) AS QBAR

GROUP BY QBAR.ГРУЗ

) AS WBAR

WHERE ГРУЗ.ИДЕНТИФИКАТОР = WBAR.ГРУЗ

GROUP BY WBAR.ГРУЗ, WBAR.ИТОГ HAVING COUNT(DISTINCT СУДНО.ПОРТ\_ПРИПИСКИ) - 1 = WBAR.ИТОГ

)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Добавим необходимые для выполнения условия записи (временные):

INSERT INTO ПОГРУЗКА (НОМЕР\_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО\_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ)

VALUES (70220, 'Среда', 004, 001, 002, 20, 1940000),

(70221, 'Среда', 006, 002, 002, 20, 1940000)

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

SELECT \* FROM ГРУЗ WHERE EXISTS

(SELECT \* FROM ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ, СУДНО WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND

МЕСТА\_ПОГРУЗКИ.ПОРТ = 'Одесса' AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.ПОРТ\_ПРИПИСКИ = 'Владивосток'

AND ДАТА IN ('Вторник', 'Среда', 'Четверг') AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР)

Изображение выглядит как снимок экрана

Автоматически созданное описание

Data Output пуст. Для получения информации я **временно** добавила новую запись

*INSERT INTO ПОГРУЗКА (НОМЕР\_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО\_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ) VALUES (70220, 'Среда', 003, 002, 001, 20, 1940000)*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

Здесь я использовала код из задания 7.d (только вместо >1 написала = кол-ву портов)

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

WHERE СУДНО.ЛЬГОТА < (SELECT AVG(ЛЬГОТА) FROM СУДНО)

AND СУДНО.ИДЕНТИФИКАТОР = (

SELECT SUBQ.ИДЕНТИФИКАТОР FROM ( SELECT СУДНО.ИДЕНТИФИКАТОР, СУДНО.ПОРТ\_ПРИПИСКИ, МЕСТА\_ПОГРУЗКИ.ПОРТ FROM СУДНО JOIN ПОГРУЗКА ON ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР

JOIN МЕСТА\_ПОГРУЗКИ ON ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР

GROUP BY СУДНО.ИДЕНТИФИКАТОР, СУДНО.ПОРТ\_ПРИПИСКИ, МЕСТА\_ПОГРУЗКИ.ПОРТ

) AS SUBQ GROUP BY SUBQ.ИДЕНТИФИКАТОР HAVING COUNT(SUBQ.ИДЕНТИФИКАТОР) = (SELECT COUNT(DISTINCT ПОРТ) FROM МЕСТА\_ПОГРУЗКИ)

)

Изображение выглядит как снимок экрана

Автоматически созданное описание

* 1. **определить число причалов в порту Владивосток;**

*SELECT COUNT(ПОРТ) FROM МЕСТА\_ПОГРУЗКИ WHERE ПОРТ = 'Владивосток'*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

*SELECT AVG(ПОГРУЗКА.СТОИМОСТЬ) FROM ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ*

*WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ПОРТ = 'Одесса'*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

*SELECT SUM(ПОГРУЗКА.СТОИМОСТЬ) FROM ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ, СУДНО*

*WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND*

*ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ\_ПРИПИСКИ*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

Насколько я поняла, в таблице нужно оставить самую дорогую погрузку за день (в нашем случае за понедельник и вторник отдельно). Получается следующий запрос:

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

*FROM СУДНО, (SELECT СУДНО, MAX(СТОИМОСТЬ) AS МАКСИМУМ FROM ПОГРУЗКА WHERE ДАТА = 'Понедельник' OR ДАТА = 'Вторник' GROUP BY СУДНО, ДАТА) AS TAB*

*WHERE СУДНО.ИДЕНТИФИКАТОР = TAB.СУДНО AND TAB.МАКСИМУМ <= 100000*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

Если имелась в виду самая дорогая погрузка за оба дня, то запрос поменяется лишь в группировке: *GROUP BY СУДНО, ДАТА –> GROUP BY СУДНО.*

Изображение выглядит как снимок экрана

Автоматически созданное описание

Чтобы были хоть какие-то результаты, я увеличила лимит с 100000 до 2000000:

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

*FROM СУДНО, (SELECT СУДНО, MAX(СТОИМОСТЬ) AS МАКСИМУМ FROM ПОГРУЗКА WHERE ДАТА = 'Понедельник' OR ДАТА = 'Вторник' GROUP BY СУДНО, ДАТА) AS TAB*

*WHERE СУДНО.ИДЕНТИФИКАТОР = TAB.СУДНО AND TAB.МАКСИМУМ <=* 2000000

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

*SELECT ПОГРУЗКА.ДАТА, COUNT(ПОГРУЗКА.ДАТА) AS КОЛВО FROM ПОГРУЗКА, МЕСТА\_ПОГРУЗКИ*

*WHERE ПОГРУЗКА.МЕСТО\_ПОГРУЗКИ = МЕСТА\_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА\_ПОГРУЗКИ.ПОРТ = 'Владивосток'*

*GROUP BY ДАТА ORDER BY 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*

Изображение выглядит как снимок экрана

Автоматически созданное описание

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

*SELECT DISTINCT ПОГРУЗКА.ГРУЗ AS ИДЕНТИФИКАТОР\_ГРУЗА, ГРУЗ.НАЗВАНИЕ, МЕСТА\_ПОГРУЗКИ.ПОРТ AS ПОРТ\_ПОГРУЗКИ, ГРУЗ.ПОРТ\_СКЛАДИРОВАНИЯ, SUM(ПОГРУЗКА.СТОИМОСТЬ\*(1+МЕСТА\_ПОГРУЗКИ.ОТЧИСЛЕНИЯ\_НА\_ПОГРУЗКУ/100)) AS ИТОГ*

*FROM ПОГРУЗКА, ГРУЗ, МЕСТА\_ПОГРУЗКИ*

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

*ГРУЗ.ПОРТ\_СКЛАДИРОВАНИЯ = МЕСТА\_ПОГРУЗКИ.ПОРТ*

*GROUP BY ПОГРУЗКА.ГРУЗ, ГРУЗ.НАЗВАНИЕ, МЕСТА\_ПОГРУЗКИ.ПОРТ, ГРУЗ.ПОРТ\_СКЛАДИРОВАНИЯ*

*Изображение выглядит как снимок экрана

Автоматически созданное описание*

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

Вывела таблицу с информацией о грузах + добавила к выводу столбец «КОЛВО» с количеством различных судов.

*SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') AS ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ\_СКЛАДИРОВАНИЯ, СТОИМОСТЬ, МАКС\_КОЛВО, QTAB.КОЛВО FROM ГРУЗ, (SELECT ГРУЗ, COUNT(DISTINCT ПОГРУЗКА.СУДНО) AS КОЛВО FROM ПОГРУЗКА*

*WHERE 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 > 1*

*GROUP BY ГРУЗ) AS QTAB WHERE ИДЕНТИФИКАТОР = QTAB.ГРУЗ*

Изображение выглядит как снимок экрана

Автоматически созданное описание

Конец.

надеюсь, это 10

я старалась как никогда