

**ВАРИАНТ 7. ВЫПОЛНИЛА МАРТЫНЮК ОЛЕСЯ, 18ПИ-2**

Уровень 1

1. Данна 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)
)

```

## Query Editor

```

1  CREATE TABLE vessel (
2      Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР
3      Name text NOT NULL, -- НАЗВАНИЕ (выражается строковыми данными)
4      Port text NOT NULL, -- ПОРТ ПРИПИСКИ (выражается строковыми данными)
5      Exemption integer NOT NULL, CHECK (Exemption >= 0) -- ЛЬГОТА (выражается числовыми данными)
6  );
7  CREATE TABLE places ( -- МЕСТА ПОГРУЗКИ
8      Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР
9      Pier text NOT NULL, -- ПРИЧАЛ (выражается строковыми данными)
10     Port text NOT NULL, -- ПОРТ (выражается строковыми данными)
11     Allocation integer NOT NULL, CHECK (Allocation >= 0) -- ОТЧИСЛЕНИЯ НА ПОГРУЗКУ (выражается
12 );
13
14  CREATE TABLE cargo ( -- ГРУЗ
15      Id SERIAL PRIMARY KEY, -- ИДЕНТИФИКАТОР
16      Name text NOT NULL, -- НАЗВАНИЕ (выражается строковыми данными)
17      Port text NOT NULL, -- ПОРТ СКЛАДИРОВАНИЯ (выражается строковыми данными)
18      Price integer NOT NULL, CHECK (Price >= 0), -- СТОИМОСТЬ, РУБ (выражается числовыми данным
19      Amount integer NOT NULL, CHECK (Amount > 0) -- МАКС. КОЛ-ВО (выражается числовыми данными, не
20 );
21
22  CREATE TABLE shipment ( -- ПОГРУЗКА
23      Register SERIAL PRIMARY KEY, -- НОМЕР ВЕДОМОСТИ
24      Weekday text NOT NULL, -- ДАТА (выражается строковыми данными)
25      Vessel integer NOT NULL, CHECK (Vessel > 0), -- СУДНО (выражается числовыми данными, не м
26      Place integer NOT NULL CHECK (Place > 0), -- МЕСТО ПОГРУЗКИ (выражается числовыми данными,
27      Cargo integer NOT NULL, CHECK (Cargo > 0), -- ГРУЗ (выражается числовыми данными, не может
28      Amount integer NOT NULL, CHECK (Amount > 0), -- КОЛ-ВО (выражается числовыми данными, не м
29      Price integer NOT NULL -- СТОИМОСТЬ, РУБ (выражается числовыми данными)
30 )
31

```

## Messages

CREATE TABLE

Query returned successfully in 75 msec.

**2.** 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. Шалляпин', '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)
```

### Query Editor

```
1 INSERT INTO vessel (id, name, port, exemption) VALUES (001, 'Балтимор', 'Одесса', 3), (002
2
3 INSERT INTO places (id, pier, port, allocation) VALUES (001, 'Северный', 'Одесса', 3), (002
4
5 INSERT INTO cargo (id, name, port, price, amount) VALUES (001, 'Рис', 'Одесса', 100000, 76
6
7 INSERT INTO shipment (register, weekday, vessel, place, cargo, amount, price) VALUES (70204,
8
```

### Messages

INSERT 0 16

Query returned successfully in 54 msec.

3. 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
```

### Query Editor

```
1 SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ_ПРИПИСКИ, ЛЬГОТА FROM СУДНО
```

#### Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА
1 001	Балтимор	Одесса	3
2 002	Генуя	Одесса	3
3 003	ТПР-123	Владивосток	5
4 004	Ф. Шаляпин	Мурманск	6
5 005	Рейн	Калининград	4
6 006	Россия	Владивосток	5

```
SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, ПРИЧАЛ, ПОРТ, ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ FROM МЕСТА_ПОГРУЗКИ
```

### Query Editor

```
1 SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, ПРИЧАЛ, ПОРТ, ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ FROM МЕСТА_ПОГРУЗКИ
```

#### Data Output

ИДЕНТИФИКАТОР	ПРИЧАЛ	ПОРТ	ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ
1 001	Северный	Одесса	3
2 002	Южный	Одесса	4
3 003	N1	Владивосток	2
4 004	N2	Владивосток	2
5 005	N3	Владивосток	2
6 006	Основной	Калининград	4

```
SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ_СКЛАДИРОВАНИЯ, СТОИМОСТЬ, МАКС_КОЛВО FROM ГРУЗ
```

Query Editor

```
1 SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ_СКЛАДИРОВАНИЯ, СТОИМОСТЬ, МАКС_КОЛВО FROM ГРУЗ
```

Data Output

	ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО
1	001	Рис	Одесса	100000	700
2	002	Зерно	Одесса	80000	890
3	003	Хлопок	Одесса	300000	400
4	004	Сахар	Владивосток	140000	600
5	005	Соль	Мурманск	120000	700
6	006	Скобяные и...	Калининград	300000	140
7	007	Древесина	Мурманск	400000	260
8	008	Уголь	Владивосток	400000	400

`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`

Query Editor

```
1 SELECT НОМЕР_ВЕДОМОСТИ, ДАТА, LPAD(СУДНО::text, 3, '0') as СУДНО, LPAD(МЕСТО_ПОГРУЗКИ::text, 3, '0')
2 as МЕСТО_ПОГРУЗКИ, LPAD(ГРУЗ::text, 3, '0') as ГРУЗ, КОЛВО, СТОИМОСТЬ FROM ПОГРУЗКА
```

Data Output

	НОМЕР_ВЕДОМОСТИ	ДАТА	СУДНО	МЕСТО_ПОГРУЗКИ	ГРУЗ	КОЛВО	СТОИМОСТЬ
1	70204	Понедельник	001	005	002	100	8000000
2	70205	Понедельник	003	003	006	4	1200000
3	70206	Вторник	001	005	007	2	800000
4	70207	Вторник	002	005	001	20	2000000
5	70208	Вторник	005	005	002	3	240000
6	70209	Среда	003	003	006	4	1200000
7	70210	Среда	004	001	001	70	7000000
8	70211	Среда	004	002	006	1	300000
9	70212	Среда	004	002	001	10	1000000
10	70213	Четверг	001	006	003	20	6000000
11	70214	Четверг	003	004	002	2	16000
12	70215	Четверг	004	003	004	30	4200000
13	70216	Суббота	003	002	005	10	1200000
14	70217	Суббота	002	003	008	20	8000000
15	70218	Суббота	001	001	001	20	2000000
16	70219	Суббота	005	006	004	10	1400000

#### 4. Создать запросы для вывода:

- a) всех различных названий судов и их льгот;

`SELECT DISTINCT НАЗВАНИЕ, ЛЬГОТА FROM СУДНО`

Query Editor

```
1 SELECT DISTINCT НАЗВАНИЕ, ЛЬГОТА FROM СУДНО
```

Data Output

	НАЗВАНИЕ	ЛЬГОТА
1	Россия	5
2	Ф. Шаляпин	6
3	Балтимор	3
4	Генуя	3
5	Рейн	4
6	ТПР-123	5

- b) всех различных адресов приписки судов;

`SELECT DISTINCT ПОРТ_ПРИПИСКИ FROM СУДНО`

Query Editor

```
1 SELECT DISTINCT ПОРТ_ПРИПИСКИ FROM СУДНО
```

Data Output

ПОРТ_ПРИПИСКИ
Мурманск
Одесса
Калининград
Владивосток

с) всех различных портов

```
SELECT DISTINCT ПОРТ FROM МЕСТА_ПОГРУЗКИ
UNION
SELECT ПОРТ_СКЛАДИРОВАНИЯ FROM ГРУЗ
UNION
SELECT ПОРТ_ПРИПИСКИ FROM СУДНО
```

Query Editor

```
1 SELECT DISTINCT ПОРТ FROM МЕСТА_ПОГРУЗКИ
2 UNION
3 SELECT ПОРТ_СКЛАДИРОВАНИЯ FROM ГРУЗ
4 UNION
5 SELECT ПОРТ_ПРИПИСКИ FROM СУДНО
6
```

Data Output

ПОРТ
Мурманск
Калининград
Владивосток
Одесса

5. Создав запрос получить следующую информацию:

а) название и стоимость грузов, находящихся в количестве менее 500 единиц;

```
SELECT НАЗВАНИЕ, СТОИМОСТЬ FROM ГРУЗ WHERE МАКС_КОЛВО < 500
```

Query Editor

```
1 SELECT НАЗВАНИЕ, СТОИМОСТЬ FROM ГРУЗ WHERE МАКС_КОЛВО < 500
2
```

Data Output

НАЗВАНИЕ	СТОИМОСТЬ
Хлопок	300000
Скобяные и...	300000
Древесина	400000
Уголь	400000

б) 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%'
```

Query Editor

```
1 SELECT DISTINCT ПОРТ FROM МЕСТА_ПОГРУЗКИ WHERE ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 5 OR ПРИЧАЛ LIKE '%N%'
```

Data Output

ПОРТ
text
1 Владивосток

с) названия судов из Одессы.

`SELECT DISTINCT НАЗВАНИЕ FROM СУДНО WHERE ПОРТ_ПРИПИСКИ = 'Одесса'`

Query Editor

```
1 SELECT DISTINCT НАЗВАНИЕ FROM СУДНО WHERE ПОРТ_ПРИПИСКИ = 'Одесса'
```

Data Output

НАЗВАНИЕ
text
1 Балтимор
2 Генуя

6. На основании данных о погрузке вывести все данные в таком формате:

а) 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
```

Query Editor

```
1 SELECT НОМЕР_ВЕДОМОСТИ, ДАТА, НАЗВАНИЕ, СТОИМОСТЬ FROM ПОГРУЗКА, СУДНО WHERE
2 ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР
3 ORDER BY CASE
4 WHEN ДАТА = 'Понедельник' THEN 1
5 WHEN ДАТА = 'Вторник' THEN 2
6 WHEN ДАТА = 'Среда' THEN 3
7 WHEN ДАТА = 'Четверг' THEN 4
8 WHEN ДАТА = 'Пятница' THEN 5
9 WHEN ДАТА = 'Суббота' THEN 6
10 WHEN ДАТА = 'Воскресенье' THEN 7
11 END
12 , НАЗВАНИЕ
13
```

Data Output

НОМЕР_ВЕДОМОСТИ	ДАТА	НАЗВАНИЕ	СТОИМОСТЬ	
integer	text	text	integer	
1	70204	Понедельник	Балтимор	800000
2	70205	Понедельник	ТПР-123	1200000
3	70206	Вторник	Балтимор	800000
4	70207	Вторник	Генуя	2000000
5	70208	Вторник	Рейн	240000
6	70209	Среда	ТПР-123	1200000
7	70212	Среда	Ф. Шалляпин	1000000
8	70210	Среда	Ф. Шалляпин	700000
9	70211	Среда	Ф. Шалляпин	300000
10	70213	Четверг	Балтимор	6000000
11	70214	Четверг	ТПР-123	16000
12	70215	Четверг	Ф. Шалляпин	4200000
13	70218	Суббота	Балтимор	2000000
14	70217	Суббота	Генуя	8000000
15	70219	Суббота	Рейн	1400000
16	70216	Суббота	ТПР-123	1200000

**b)** 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

```

Query Editor

```

1 SELECT СУДНО.НАЗВАНИЕ AS НАЗВАНИЕ_СУДНА, ПОГРУЗКА.ДАТА, МЕСТА_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА_ПОГРУЗКИ.ПОРТ, ГРУЗ.НАЗВАНИЕ AS НАЗВАНИЕ_ГРУЗА
2 FROM СУДНО, ПОГРУЗКА, МЕСТА_ПОГРУЗКИ, ГРУЗ WHERE
3 ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР
4

```

Data Output

	НАЗВАНИЕ_СУДНА	ДАТА	ПРИЧАЛ	ПОРТ	НАЗВАНИЕ_ГРУЗА
1	Балтимор	Понедельник	N3	Владивосток	Зерно
2	ТПР-123	Понедельник	N1	Владивосток	Скобяные изделия
3	Балтимор	Вторник	N3	Владивосток	Древесина
4	Генуя	Вторник	N3	Владивосток	Рис
5	Рейн	Вторник	N3	Владивосток	Зерно
6	ТПР-123	Среда	N1	Владивосток	Скобяные изделия
7	Ф. Шаляпин	Среда	Северный	Одесса	Рис
8	Ф. Шаляпин	Среда	Южный	Одесса	Скобяные изделия
9	Ф. Шаляпин	Среда	Южный	Одесса	Рис
10	Балтимор	Четверг	Основной	Калининград	Хлопок
11	ТПР-123	Четверг	N2	Владивосток	Зерно
12	Ф. Шаляпин	Четверг	N1	Владивосток	Сахар
13	ТПР-123	Суббота	Южный	Одесса	Соль
14	Генуя	Суббота	N1	Владивосток	Уголь
15	Балтимор	Суббота	Северный	Одесса	Рис
16	Рейн	Суббота	Основной	Калининград	Сахар

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

```

Query Editor

```

1 SELECT СУДНО.НАЗВАНИЕ AS НАЗВАНИЕ_СУДНА, ПОГРУЗКА.ДАТА, МЕСТА_ПОГРУЗКИ.ПРИЧАЛ, ГРУЗ.НАЗВАНИЕ AS НАЗВАНИЕ_ГРУЗА
2 FROM СУДНО, ПОГРУЗКА, МЕСТА_ПОГРУЗКИ, ГРУЗ WHERE
3 ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР
4

```

Data Output

	НАЗВАНИЕ_СУДНА	ДАТА	ПРИЧАЛ	НАЗВАНИЕ_ГРУЗА
1	Балтимор	Понедельник	N3	Зерно
2	ТПР-123	Понедельник	N1	Скобяные изделия
3	Балтимор	Вторник	N3	Древесина
4	Генуя	Вторник	N3	Рис
5	Рейн	Вторник	N3	Зерно
6	ТПР-123	Среда	N1	Скобяные изделия
7	Ф. Шаляпин	Среда	Северный	Рис
8	Ф. Шаляпин	Среда	Южный	Скобяные изделия
9	Ф. Шаляпин	Среда	Южный	Рис
10	Балтимор	Четверг	Основной	Хлопок
11	ТПР-123	Четверг	N2	Зерно
12	Ф. Шаляпин	Четверг	N1	Сахар
13	ТПР-123	Суббота	Южный	Соль
14	Генуя	Суббота	N1	Уголь
15	Балтимор	Суббота	Северный	Рис
16	Рейн	Суббота	Основной	Сахар

**7. Вывести:**

- 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;

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_ПРИПЛИСКИ AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND
МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 3 ORDER BY МЕСТА_ПОГРУЗКИ.ПОРТ

```

Получились следующие данные:

Query Editor

```
1 SELECT СУДНО.НАЗВАНИЕ, МЕСТА_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА_ПОГРУЗКИ.ПОРТ FROM
2 СУДНО, МЕСТА_ПОГРУЗКИ, ПОГРУЗКА WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
3 МЕСТА_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ_ПРИПИСКИ AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 3
4 ORDER BY МЕСТА_ПОГРУЗКИ.ПОРТ
5 |
```

Data Output

НАЗВАНИЕ	ПРИЧАЛ	ПОРТ
1 Балтимор	Основной	Калининград
2 Ф. Шаляпин	Южный	Одесса
3 Ф. Шаляпин	Южный	Одесса
4 ТПР-123	Южный	Одесса

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
```

Query Editor

```
1 SELECT DISTINCT СУДНО.НАЗВАНИЕ, МЕСТА_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА_ПОГРУЗКИ.ПОРТ FROM
2 СУДНО, МЕСТА_ПОГРУЗКИ, ПОГРУЗКА WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР
3 AND МЕСТА_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ_ПРИПИСКИ AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР
4 AND МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 3 ORDER BY МЕСТА_ПОГРУЗКИ.ПОРТ
5 |
6 |
```

Data Output

НАЗВАНИЕ	ПРИЧАЛ	ПОРТ
1 Балтимор	Основной	Калининград
2 ТПР-123	Южный	Одесса
3 Ф. Шаляпин	Южный	Одесса

**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.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
```

Query Editor

```
1 SELECT DISTINCT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ AS ПОРТ_ПОГРУЗКИ FROM
2 СУДНО, ГРУЗ, ПОГРУЗКА, МЕСТА_ПОГРУЗКИ WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР
3 AND ГРУЗ.СТОИМОСТЬ > 50000 AND CASE
4 WHEN ПОГРУЗКА.ДАТА = 'Понедельник' THEN 1
5 WHEN ПОГРУЗКА.ДАТА = 'Вторник' THEN 2
6 WHEN ПОГРУЗКА.ДАТА = 'Среда' THEN 3
7 WHEN ПОГРУЗКА.ДАТА = 'Четверг' THEN 4
8 WHEN ПОГРУЗКА.ДАТА = 'Пятница' THEN 5
9 WHEN ПОГРУЗКА.ДАТА = 'Суббота' THEN 6
10 WHEN ПОГРУЗКА.ДАТА = 'Воскресенье' THEN 7
11 END <= 3 AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР
12 |
```

Data Output

НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ПОРТ_ПОГРУЗКИ
1 Рейн	Калининград	Владивосток
2 Балтимор	Одесса	Владивосток
3 Генуя	Одесса	Владивосток
4 ТПР-123	Владивосток	Владивосток
5 Ф. Шаляпин	Мурманск	Одесса

- c)** 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
```

Query Editor

```
1 SELECT ГРУЗ.НАЗВАНИЕ, ГРУЗ.СТОИМОСТЬ FROM ГРУЗ, СУДНО, ПОГРУЗКА, МЕСТА_ПОГРУЗКИ WHERE
2 ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.НАЗВАНИЕ = 'Генуя' AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ =
3 МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 2
```

Data Output

НАЗВАНИЕ	СТОИМОСТЬ

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

- d)** 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
```

Query Editor

```
1 SELECT SUBQ.НАЗВАНИЕ, SUBQ.ПОРТ_ПРИПИСКИ FROM ( SELECT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ FROM СУДНО JOIN ПОГРУЗКА
2 JOIN МЕСТА_ПОГРУЗКИ ON ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР
3 GROUP BY СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ
4 ) AS SUBQ GROUP BY SUBQ.НАЗВАНИЕ, SUBQ.ПОРТ_ПРИПИСКИ HAVING COUNT(SUBQ.НАЗВАНИЕ) > 1
```

Data Output

НАЗВАНИЕ	ПОРТ_ПРИПИСКИ
Балтимор	Одесса
Рейн	Калининград
Ф. Шаляпин	Мурманск
ТПР-123	Владивосток

- 8.** 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
```

## Query Editor

```

1 UPDATE ПОГРУЗКА SET СТОИМОСТЬ = СТОИМОСТЬ * (100 - СУДНО.ЛЬГОТА)/100 FROM СУДНО WHERE СУДНО.ИДЕНТИФИКАТОР = ПОГРУЗКА.СУДНО;
2 SELECT НОМЕР_ВЕДОМОСТИ, ДАТА, LPAD(СУДНО::text, 3, '0') AS СУДНО, LPAD(МЕСТО_ПОГРУЗКИ::text, 3, '0')
3 AS МЕСТО_ПОГРУЗКИ, LPAD(ГРУЗ::text, 3, '0') AS ГРУЗ, КОЛВО, СТОИМОСТЬ FROM ПОГРУЗКА
4

```

Data Output

НОМЕР_ВЕДОМОСТИ	ДАТА	СУДНО	МЕСТО_ПОГРУЗКИ	ГРУЗ	КОЛВО	СТОИМОСТЬ
[PK] integer	text	text	text	text	integer	integer
1	70204	Пон...	001	005	002	100
2	70205	Пон...	003	003	006	4
3	70206	Втор...	001	005	007	2
4	70207	Втор...	002	005	001	20
5	70208	Втор...	005	005	002	3
6	70209	Среда	003	003	006	4
7	70210	Среда	004	001	001	70
8	70211	Среда	004	002	006	1
9	70212	Среда	004	002	001	10
10	70213	Четв...	001	006	003	20
11	70214	Четв...	003	004	002	2
12	70215	Четв...	004	003	004	30
13	70216	Субб...	003	002	005	10
14	70217	Субб...	002	003	008	20
15	70218	Субб...	001	001	001	20
16	70219	Субб...	005	006	004	10

9. 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

```

## Query Editor

```

1 ALTER TABLE ПОГРУЗКА ADD ОТЧИСЛЕНИЯ integer CHECK (ОТЧИСЛЕНИЯ >= 0);
2 UPDATE ПОГРУЗКА SET ОТЧИСЛЕНИЯ = ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ FROM МЕСТА_ПОГРУЗКИ WHERE МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР = ПОГРУЗКА.МЕСТО_ПОГРУЗКИ;
3
4 SELECT НОМЕР_ВЕДОМОСТИ, ДАТА, LPAD(СУДНО::text, 3, '0') AS СУДНО, LPAD(МЕСТО_ПОГРУЗКИ::text, 3, '0')
5 AS МЕСТО_ПОГРУЗКИ, LPAD(ГРУЗ::text, 3, '0') AS ГРУЗ, КОЛВО, СТОИМОСТЬ, ОТЧИСЛЕНИЯ FROM ПОГРУЗКА

```

Data Output

НОМЕР_ВЕДОМОСТИ	ДАТА	СУДНО	МЕСТО_ПОГРУЗКИ	ГРУЗ	КОЛВО	СТОИМОСТЬ	ОТЧИСЛЕНИЯ
[PK] integer	text	text	text	text	integer	integer	integer
1	70204	Пон...	001	005	002	100	7760000
2	70205	Пон...	003	003	006	4	1140000
3	70206	Втор...	001	005	007	2	776000
4	70207	Втор...	002	005	001	20	1940000
5	70208	Втор...	005	005	002	3	230400
6	70209	Среда	003	003	006	4	1140000
7	70210	Среда	004	001	001	70	6580000
8	70211	Среда	004	002	006	1	282000
9	70212	Среда	004	002	001	10	940000
10	70213	Четв...	001	006	003	20	5820000
11	70214	Четв...	003	004	002	2	15200
12	70215	Четв...	004	003	004	30	3948000
13	70216	Субб...	003	002	005	10	1140000
14	70217	Субб...	002	003	008	20	7760000
15	70218	Субб...	001	001	001	20	1940000
16	70219	Субб...	005	006	004	10	1344000

Уровень 2

10. Используя операцию IN (NOT IN) реализовать следующие запросы:

a) 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'))))

```

## Query Editor

```

1 SELECT * FROM СУДНО WHERE ИДЕНТИФИКАТОР NOT IN (SELECT DISTINCT СУДНО FROM ПОГРУЗКА
2 WHERE МЕСТО_ПОГРУЗКИ IN (SELECT ИДЕНТИФИКАТОР FROM МЕСТА_ПОГРУЗКИ WHERE ПОРТ IN ('Владивосток'))))

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПISКИ	ЛЬГОТА
[PK] integer	text	text	integer
1	6	Россия	Владивосток
			5

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

**b)** 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)
```

Query Editor

```
1 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)
```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО
1	Рис	Одесса	100000	700
2	Хлопок	Одесса	300000	400
3	Уголь	Владивосток	400000	400

**c)** запрос задания 7.а и 7.с.

7.а: 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
```

Query Editor

```
1 SELECT DISTINCT СУДНО.НАЗВАНИЕ, МЕСТА_ПОГРУЗКИ.ПРИЧАЛ, МЕСТА_ПОГРУЗКИ.ПОРТ FROM  
2 СУДНО, МЕСТА_ПОГРУЗКИ, ПОГРУЗКА WHERE  
3 ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ_ПРИПИСКИ  
4 AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND  
5 ПОГРУЗКА.МЕСТО_ПОГРУЗКИ IN (SELECT ИДЕНТИФИКАТОР FROM МЕСТА_ПОГРУЗКИ WHERE ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 3 )ORDER BY МЕСТА_ПОГРУЗКИ.ПОРТ
```

Data Output

НАЗВАНИЕ	ПРИЧАЛ	ПОРТ
Балтимор	Основной	Калининград
ТПР-123	Южный	Одесса
Ф. Шалляпин	Южный	Одесса

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)
```

Query Editor

```
1 SELECT НАЗВАНИЕ, СТОИМОСТЬ FROM ГРУЗ WHERE ИДЕНТИФИКАТОР IN  
2 (SELECT ГРУЗ FROM ПОГРУЗКА, СУДНО, МЕСТА_ПОГРУЗКИ WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.НАЗВАНИЕ = 'Генуя'  
3 AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ > 2)  
4
```

Data Output

НАЗВАНИЕ	СТОИМОСТЬ
----------	-----------

**11.** Используя операции ALL-ANY реализовать следующие запросы:

a) 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))
```

Query Editor

```
1 SELECT DISTINCT ПОРТ_ПРИПИСКИ FROM СУДНО  
2 WHERE ИДЕНТИФИКАТОР IN (SELECT СУДНО FROM ПОГРУЗКА WHERE СТОИМОСТЬ >= ALL (SELECT СТОИМОСТЬ FROM ПОГРУЗКА))
```

Data Output

ПОРТ_ПРИПИСКИ
Одесса

b) 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 = 'Владивосток' AND КОЛВО > 20)
```

Query Editor

```
1 SELECT * FROM ГРУЗ WHERE ИДЕНТИФИКАТОР = ANY(SELECT ГРУЗ FROM ПОГРУЗКА, СУДНО  
2 WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.ПОРТ_ПРИПИСКИ = 'Владивосток' AND КОЛВО > 20)
```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО

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

c) 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)
```

Query Editor

```
1 SELECT ПРИЧАЛ FROM МЕСТА_ПОГРУЗКИ WHERE  
2 ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ <= ALL (SELECT ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ FROM МЕСТА_ПОГРУЗКИ) AND  
3 ИДЕНТИФИКАТОР IN (SELECT DISTINCT МЕСТО_ПОГРУЗКИ FROM ПОГРУЗКА WHERE СТОИМОСТЬ > 500000 GROUP BY МЕСТО_ПОГРУЗКИ HAVING COUNT(*) >= 2)
```

Data Output

ПРИЧАЛ
N1
N3

d) 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)

```

Query Editor

```

1 SELECT DISTINCT СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ AS ПОРТ_ПОГРУЗКИ FROM
2 СУДНО, ГРУЗ, ПОГРУЗКА, МЕСТА_ПОГРУЗКИ WHERE ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND
3 ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
4 ПОГРУЗКА.ГРУЗ = ANY(SELECT ИДЕНТИФИКАТОР FROM ГРУЗ WHERE СТОИМОСТЬ > 50000) AND
5 ПОГРУЗКА.ДАТА = ANY(SELECT ДАТА FROM ПОГРУЗКА WHERE CASE
6 WHEN ПОГРУЗКА.ДАТА = 'Понедельник' THEN 1
7 WHEN ПОГРУЗКА.ДАТА = 'Вторник' THEN 2
8 WHEN ПОГРУЗКА.ДАТА = 'Среда' THEN 3
9 WHEN ПОГРУЗКА.ДАТА = 'Четверг' THEN 4
10 WHEN ПОГРУЗКА.ДАТА = 'Пятница' THEN 5
11 WHEN ПОГРУЗКА.ДАТА = 'Суббота' THEN 6
12 WHEN ПОГРУЗКА.ДАТА = 'Воскресенье' THEN 7
13 END <= 3)
14

```

Data Output

НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ПОРТ_ПОГРУЗКИ
1 Балтимор	Одесса	Владивосток
2 Генуя	Одесса	Владивосток
3 Рейн	Калининград	Владивосток
4 ТПР-123	Владивосток	Владивосток
5 Ф. Шалляпин	Мурманск	Одесса

## 12. 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

```

Query Editor

```

1 SELECT ПОРТ_ПРИПИСКИ AS ПОРТ_ПОГРУЗКИ_ПРИПИСКИ FROM СУДНО
2 UNION
3 SELECT ПОРТ FROM МЕСТА_ПОГРУЗКИ
4

```

Data Output

ПОРТ_ПОГРУЗКИ_ПРИПИСКИ
1 Мурманск
2 Калининград
3 Владивосток
4 Одесса

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

```

SELECT ИДЕНТИФИКАТОР, ПОРТ_ПРИПИСКИ AS ПОРТЫ FROM СУДНО
UNION
SELECT ИДЕНТИФИКАТОР, ПОРТ FROM МЕСТА_ПОГРУЗКИ

```

Query Editor

```

1 SELECT ИДЕНТИФИКАТОР, ПОРТ_ПРИПИСКИ AS ПОРТЫ FROM СУДНО
2 UNION
3 SELECT ИДЕНТИФИКАТОР, ПОРТ FROM МЕСТА_ПОГРУЗКИ
4
5
Data Output


| ИДЕНТИФИКАТОР | ПОРТЫ       |
|---------------|-------------|
| 1             | Мурманск    |
| 2             | Одесса      |
| 3             | Одесса      |
| 4             | Владивосток |
| 5             | Владивосток |
| 6             | Калининград |
| 7             | Владивосток |
| 8             | Калининград |
| 9             | Владивосток |


```

13. 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.

а) 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
)

```

Query Editor

```

1 SELECT * FROM МЕСТА_ПОГРУЗКИ WHERE EXISTS (
2     SELECT * FROM (SELECT МЕСТО_ПОГРУЗКИ FROM ПОГРУЗКА, СУДНО
3     WHERE ПОГРУЗКА.ДАТА IN ('Понедельник', 'Вторник', 'Среда', 'Четверг')
4     AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND (СУДНО.ПОРТ_ПРИПИСКИ = 'Одесса')
5     GROUP BY МЕСТО_ПОГРУЗКИ
6     HAVING COUNT(DISTINCT ПОГРУЗКА.СУДНО) = (SELECT COUNT(ИДЕНТИФИКАТОР) FROM СУДНО WHERE ПОРТ_ПРИПИСКИ = 'Одесса')
7     UNION SELECT МЕСТО_ПОГРУЗКИ FROM ПОГРУЗКА, СУДНО
8     WHERE ПОГРУЗКА.ДАТА IN ('Понедельник', 'Вторник', 'Среда', 'Четверг')
9     AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND (СУДНО.ПОРТ_ПРИПИСКИ = 'Мурманск')
10    GROUP BY МЕСТО_ПОГРУЗКИ
11    HAVING COUNT(DISTINCT ПОГРУЗКА.СУДНО) = (SELECT COUNT(ИДЕНТИФИКАТОР) FROM СУДНО WHERE ПОРТ_ПРИПИСКИ = 'Мурманск'))
12    AS QBAR WHERE QBAR.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР
13 )
14
Data Output


| ИДЕНТИФИКАТОР | ПРИЧАЛ   | ПОРТ      | ОТЧИСЛЕНИЯ_НА_ПОРТ |
|---------------|----------|-----------|--------------------|
| 1             | Северный | Одесса    | 3                  |
| 2             | Южный    | Одесса    | 4                  |
| 3             | Н1       | Владив... | 2                  |
| 4             | Н3       | Владив... | 2                  |


```

б) 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
)

```

Query Editor

```

1  SELECT * FROM СУДНО WHERE EXISTS (
2      SELECT * FROM (
3          SELECT СУДНО.ИДЕНТИФИКАТОР FROM СУДНО, ПОГРУЗКА
4              WHERE СУДНО.ИДЕНТИФИКАТОР = ПОГРУЗКА.СУДНО AND
5                  ПОГРУЗКА.СУДНО NOT IN
6                  (
7                      SELECT ПОГРУЗКА.СУДНО FROM ПОГРУЗКА
8                          GROUP BY ПОГРУЗКА.СУДНО, ПОГРУЗКА.МЕСТО_ПОГРУЗКИ
9                          HAVING COUNT(*) > 1
10                 )
11             GROUP BY СУДНО.ИДЕНТИФИКАТОР HAVING COUNT(СУДНО.ИДЕНТИФИКАТОР) = (SELECT COUNT(ПРИЧАЛ) FROM МЕСТА_ПОГРУЗКИ)
12         ) AS QBAR WHERE QBAR.ИДЕНТИФИКАТОР = СУДНО.ИДЕНТИФИКАТОР
13     )
14

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА

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)

```

Query Editor

```

1  INSERT INTO ПОГРУЗКА (НОМЕР_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ)
2      VALUES (70220, 'Среда', 002, 001, 001, 20, 1940000),
3      (70221, 'Среда', 002, 002, 001, 20, 1940000),
4      (70222, 'Среда', 002, 004, 001, 20, 1940000),
5      (70223, 'Среда', 002, 006, 001, 20, 1940000);
6
7  SELECT * FROM СУДНО WHERE EXISTS (
8      SELECT * FROM (
9          SELECT СУДНО.ИДЕНТИФИКАТОР FROM СУДНО, ПОГРУЗКА
10             WHERE СУДНО.ИДЕНТИФИКАТОР = ПОГРУЗКА.СУДНО AND
11                 ПОГРУЗКА.СУДНО NOT IN
12                 (
13                     SELECT ПОГРУЗКА.СУДНО FROM ПОГРУЗКА
14                         GROUP BY ПОГРУЗКА.СУДНО, ПОГРУЗКА.МЕСТО_ПОГРУЗКИ
15                         HAVING COUNT(*) > 1
16                 )
17             GROUP BY СУДНО.ИДЕНТИФИКАТОР HAVING COUNT(СУДНО.ИДЕНТИФИКАТОР) = (SELECT COUNT(ПРИЧАЛ) FROM МЕСТА_ПОГРУЗКИ)
18         ) AS QBAR WHERE QBAR.ИДЕНТИФИКАТОР = СУДНО.ИДЕНТИФИКАТОР
19     )
20

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА
1	2 Генуя	Одесса	3

**с)** 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
)

```

Query Editor

```

1 SELECT * FROM ГРУЗ WHERE EXISTS(
2   SELECT WBAR.ГРУЗ FROM СУДНО,
3     SELECT QBAR.ГРУЗ, COUNT(QBAR.ГРУЗ) AS ИТОГ FROM (
4       SELECT DISTINCT ПОГРУЗКА.ГРУЗ, ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ, СУДНО.ПОРТ_ПРИПИСКИ FROM ПОГРУЗКА, ГРУЗ, СУДНО
5         WHERE ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ != СУДНО.ПОРТ_ПРИПИСКИ AND
6           ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР
7     ) AS QBAR
8     GROUP BY QBAR.ГРУЗ
9   ) AS WBAR
10  WHERE ГРУЗ.ИДЕНТИФИКАТОР = WBAR.ГРУЗ
11  GROUP BY WBAR.ГРУЗ, WBAR.ИТОГ HAVING COUNT(DISTINCT СУДНО.ПОРТ_ПРИПИСКИ) - 1 = WBAR.ИТОГ
12 )
13

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО

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)

```

Query Editor

```

1 INSERT INTO ПОГРУЗКА (НОМЕР_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ)
2   VALUES (70220, 'Среда', 004, 001, 002, 20, 1940000),
3   (70221, 'Среда', 006, 002, 002, 20, 1940000);
4
5
6 SELECT * FROM ГРУЗ WHERE EXISTS(
7   SELECT WBAR.ГРУЗ FROM СУДНО,
8     SELECT QBAR.ГРУЗ, COUNT(QBAR.ГРУЗ) AS ИТОГ FROM (
9       SELECT DISTINCT ПОГРУЗКА.ГРУЗ, ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ, СУДНО.ПОРТ_ПРИПИСКИ FROM ПОГРУЗКА, ГРУЗ, СУДНО
10        WHERE ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ != СУДНО.ПОРТ_ПРИПИСКИ AND
11          ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР
12     ) AS QBAR
13     GROUP BY QBAR.ГРУЗ
14   ) AS WBAR
15  WHERE ГРУЗ.ИДЕНТИФИКАТОР = WBAR.ГРУЗ
16  GROUP BY WBAR.ГРУЗ, WBAR.ИТОГ HAVING COUNT(DISTINCT СУДНО.ПОРТ_ПРИПИСКИ) - 1 = WBAR.ИТОГ
17 )
18

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО
1	2	Зерно	Одесса	80000
				890

d) 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)

```

Query Editor

```

1 SELECT * FROM ГРУЗ WHERE EXISTS
2 (SELECT * FROM ПОГРУЗКА, МЕСТА_ПОГРУЗКИ, СУДНО WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
3 МЕСТА_ПОГРУЗКИ.ПОРТ = 'Одесса' AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.ПОРТ_ПРИПИСКИ = 'Владивосток'
4 AND ДАТА IN ('Вторник', 'Среда', 'Четверг') AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР)
5

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО

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)`

Query Editor

```

1 INSERT INTO ПОГРУЗКА (НОМЕР_ВЕДОМОСТИ, ДАТА, СУДНО, МЕСТО_ПОГРУЗКИ, ГРУЗ, КОЛВО, СТОИМОСТЬ) VALUES (70220, 'Среда', 003, 002, 001, 20, 1940000)
2
3 SELECT * FROM ГРУЗ WHERE EXISTS
4 (SELECT * FROM ПОГРУЗКА, МЕСТА_ПОГРУЗКИ, СУДНО WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
5 МЕСТА_ПОГРУЗКИ.ПОРТ = 'Одесса' AND ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND СУДНО.ПОРТ_ПРИПИСКИ = 'Владивосток'
6 AND ДАТА IN ('Вторник', 'Среда', 'Четверг') AND ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР)
7

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_СКЛАДИРОВАНИЯ	СТОИМОСТЬ	МАКС_КОЛВО
1	1 Рис	Одесса	100000	700

#### 14. Реализовать запросы с использованием агрегатных функций:

- a) 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)
)

```

Query Editor

```

1 SELECT LPAD(СУДНО.ИДЕНТИФИКАТОР::text, 3, '0') as ИДЕНТИФИКАТОР, СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, СУДНО.ЛЬГОТА FROM СУДНО
2 WHERE СУДНО.ЛЬГОТА < (SELECT AVG(ЛЬГОТА) FROM СУДНО)
3 AND СУДНО.ИДЕНТИФИКАТОР =
4      SELECT SUBQ.ИДЕНТИФИКАТОР FROM ( SELECT СУДНО.ИДЕНТИФИКАТОР, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ FROM СУДНО JOIN ПОГРУЗКА ON ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР
5 JOIN МЕСТА_ПОГРУЗКИ ON ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР
6 GROUP BY СУДНО.ИДЕНТИФИКАТОР, СУДНО.ПОРТ_ПРИПИСКИ, МЕСТА_ПОГРУЗКИ.ПОРТ
7 ) AS SUBQ GROUP BY SUBQ.ИДЕНТИФИКАТОР HAVING COUNT(SUBQ.ИДЕНТИФИКАТОР) = (SELECT COUNT(DISTINCT ПОРТ) FROM МЕСТА_ПОГРУЗКИ)
8
9

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА
1 001	Балтимор	Одесса	3

- b) определить число причалов в порту Владивосток;

`SELECT COUNT(ПОРТ) FROM МЕСТА_ПОГРУЗКИ WHERE ПОРТ = 'Владивосток'`

### Query Editor

```
1 SELECT COUNT(ПОРТ) FROM МЕСТА_ПОГРУЗКИ WHERE ПОРТ = 'Владивосток'
```

#### Data Output

	count
	bigint
1	3

**с)** 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'
```

### Query Editor

```
1 SELECT AVG(ПОГРУЗКА.СТОИМОСТЬ) FROM ПОГРУЗКА, МЕСТА_ПОГРУЗКИ
2 WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
3 МЕСТА_ПОГРУЗКИ.ПОРТ = 'Одесса'
```

#### Data Output

	avg
	numeric
1	2176400.000000000000

**д)** 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
```

### Query Editor

```
1 SELECT SUM(ПОГРУЗКА.СТОИМОСТЬ) FROM ПОГРУЗКА, МЕСТА_ПОГРУЗКИ, СУДНО
2 WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
3 ПОГРУЗКА.СУДНО = СУДНО.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ПОРТ != СУДНО.ПОРТ_ПРИПИСКИ
4
```

#### Data Output

	sum
	bigint
1	37176400

**15.** Используя средства группировки реализовать следующие запросы:

**а)** 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 ТАБ
WHERE СУДНО.ИДЕНТИФИКАТОР = ТАБ.СУДНО AND ТАБ.МАКСИМУМ <= 100000
```

### Query Editor

```
1 SELECT LPAD(СУДНО.ИДЕНТИФИКАТОР::text, 3, '0') AS ИДЕНТИФИКАТОР, СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, СУДНО.ЛЬГОТА
2 FROM СУДНО, (SELECT СУДНО, MAX(СТОИМОСТЬ) AS МАКСИМУМ FROM ПОГРУЗКА WHERE ДАТА = 'Понедельник' OR
3 ДАТА = 'Вторник' GROUP BY СУДНО, ДАТА) AS ТАБ
4 WHERE СУДНО.ИДЕНТИФИКАТОР = ТАБ.СУДНО AND ТАБ.МАКСИМУМ <= 100000
5
```

#### Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА
text	text	text	integer

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

Query Editor

```

1 SELECT LPAD(СУДНО.ИДЕНТИФИКАТОР::text, 3, '0') AS ИДЕНТИФИКАТОР, СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, СУДНО.ЛЬГОТА
2 FROM СУДНО, (SELECT СУДНО, MAX(СТОИМОСТЬ) AS МАКСИМУМ FROM ПОГРУЗКА WHERE ДАТА = 'Понедельник' OR ДАТА = 'Вторник'
3 GROUP BY СУДНО) AS ТАБ
4 WHERE СУДНО.ИДЕНТИФИКАТОР = ТАБ.СУДНО AND ТАБ.МАКСИМУМ <= 100000
5

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА

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 ТАБ
WHERE СУДНО.ИДЕНТИФИКАТОР = ТАБ.СУДНО AND ТАБ.МАКСИМУМ <= 2000000

```

Query Editor

```

1 SELECT LPAD(СУДНО.ИДЕНТИФИКАТОР::text, 3, '0') AS ИДЕНТИФИКАТОР, СУДНО.НАЗВАНИЕ, СУДНО.ПОРТ_ПРИПИСКИ, СУДНО.ЛЬГОТА
2 FROM СУДНО, (SELECT СУДНО, MAX(СТОИМОСТЬ) AS МАКСИМУМ FROM ПОГРУЗКА WHERE ДАТА = 'Понедельник' OR ДАТА = 'Вторник'
3 GROUP BY СУДНО, ДАТА) AS ТАБ
4 WHERE СУДНО.ИДЕНТИФИКАТОР = ТАБ.СУДНО AND ТАБ.МАКСИМУМ <= 2000000
5
6

```

Data Output

ИДЕНТИФИКАТОР	НАЗВАНИЕ	ПОРТ_ПРИПИСКИ	ЛЬГОТА
1 002	Генуя	Одесса	3
2 001	Балтийск	Одесса	3
3 003	ТПР-123	Владивосток	5
4 005	Рейн	Калининград	4

**b)** 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

```

Query Editor

```

1 SELECT ПОГРУЗКА.ДАТА, COUNT(ПОГРУЗКА.ДАТА) AS КОЛВО FROM ПОГРУЗКА, МЕСТА_ПОГРУЗКИ
2 WHERE ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND МЕСТА_ПОГРУЗКИ.ПОРТ = 'Владивосток'
3 GROUP BY ДАТА ORDER BY CASE
4 WHEN ПОГРУЗКА.ДАТА = 'Понедельник' THEN 1
5 WHEN ПОГРУЗКА.ДАТА = 'Вторник' THEN 2
6 WHEN ПОГРУЗКА.ДАТА = 'Среда' THEN 3
7 WHEN ПОГРУЗКА.ДАТА = 'Четверг' THEN 4
8 WHEN ПОГРУЗКА.ДАТА = 'Пятница' THEN 5
9 WHEN ПОГРУЗКА.ДАТА = 'Суббота' THEN 6
10 WHEN ПОГРУЗКА.ДАТА = 'Воскресенье' THEN 7
11 END
12

```

Data Output

ДАТА	КОЛВО
1 Понедельник	2
2 Вторник	3
3 Среда	1
4 Четверг	2
5 Суббота	1

с) 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

```

Query Editor

```

1 SELECT DISTINCT ПОГРУЗКА.ГРУЗ AS ИДЕНТИФИКАТОР_ГРУЗА, ГРУЗ.НАЗВАНИЕ, МЕСТА_ПОГРУЗКИ.ПОРТ
2 AS ПОРТ_ПОГРУЗКИ, ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ, SUM(ПОГРУЗКА.СТОИМОСТЬ*(1+МЕСТА_ПОГРУЗКИ.ОТЧИСЛЕНИЯ_НА_ПОГРУЗКУ/100)) AS ИТОГ
3 FROM ПОГРУЗКА, ГРУЗ, МЕСТА_ПОГРУЗКИ
4 WHERE ПОГРУЗКА.ГРУЗ = ГРУЗ.ИДЕНТИФИКАТОР AND ПОГРУЗКА.МЕСТО_ПОГРУЗКИ = МЕСТА_ПОГРУЗКИ.ИДЕНТИФИКАТОР AND
5 ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ = МЕСТА_ПОГРУЗКИ.ПОРТ
6 GROUP BY ПОГРУЗКА.ГРУЗ, ГРУЗ.НАЗВАНИЕ, МЕСТА_ПОГРУЗКИ.ПОРТ, ГРУЗ.ПОРТ_СКЛАДИРОВАНИЯ
7

```

Data Output

ИДЕНТИФИКАТОР_ГРУЗ	НАЗВАНИЕ	ПОРТ_ПОГРУЗКИ	ПОРТ_СКЛАДИРОВАНИЯ	ИТОГ
1	1 Рис	Одесса	Одесса	9460000
2	4 Сахар	Владивосток	Владивосток	3948000
3	8 Уголь	Владивосток	Владивосток	7760000

д) 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

Query Editor

```
1 SELECT LPAD(ИДЕНТИФИКАТОР::text, 3, '0') AS ИДЕНТИФИКАТОР, НАЗВАНИЕ, ПОРТ_СКЛАДИРОВАНИЯ, СТОИМОСТЬ, МАКС_КОЛВО, QTAB.КОЛВ
2 FROM ГРУЗ, (SELECT ГРУЗ, COUNT(DISTINCT ПОГРУЗКА.СУДНО) AS КОЛВО FROM ПОГРУЗКА
3 WHERE CASE
4 WHEN ПОГРУЗКА.ДАТА = 'Понедельник' THEN 1
5 WHEN ПОГРУЗКА.ДАТА = 'Вторник' THEN 2
6 WHEN ПОГРУЗКА.ДАТА = 'Среда' THEN 3
7 WHEN ПОГРУЗКА.ДАТА = 'Четверг' THEN 4
8 WHEN ПОГРУЗКА.ДАТА = 'Пятница' THEN 5
9 WHEN ПОГРУЗКА.ДАТА = 'Суббота' THEN 6
10 WHEN ПОГРУЗКА.ДАТА = 'Воскресенье' THEN 7
11 END > 1
12 GROUP BY ГРУЗ) AS QTAB WHERE ИДЕНТИФИКАТОР = QTAB.ГРУЗ
13
```

Data Output

	ИДЕНТИФИКАТОР text	НАЗВАНИЕ text	ПОРТ_СКЛАДИРОВАНИЯ text	СТОИМОСТЬ integer	МАКС_КОЛВО integer	КОЛВО bigint
1	001	Рис	Одесса	100000	700	3
2	002	Зерно	Одесса	80000	890	2
3	003	Хлопок	Одесса	300000	400	1
4	004	Сахар	Владивосток	140000	600	2
5	005	Соль	Мурманск	120000	700	1
6	006	Скобяные и...	Калининград	300000	140	2
7	007	Древесина	Мурманск	400000	260	1
8	008	Уголь	Владивосток	400000	400	1

Конец.

надеюсь, это 10  
я старалась как никогда