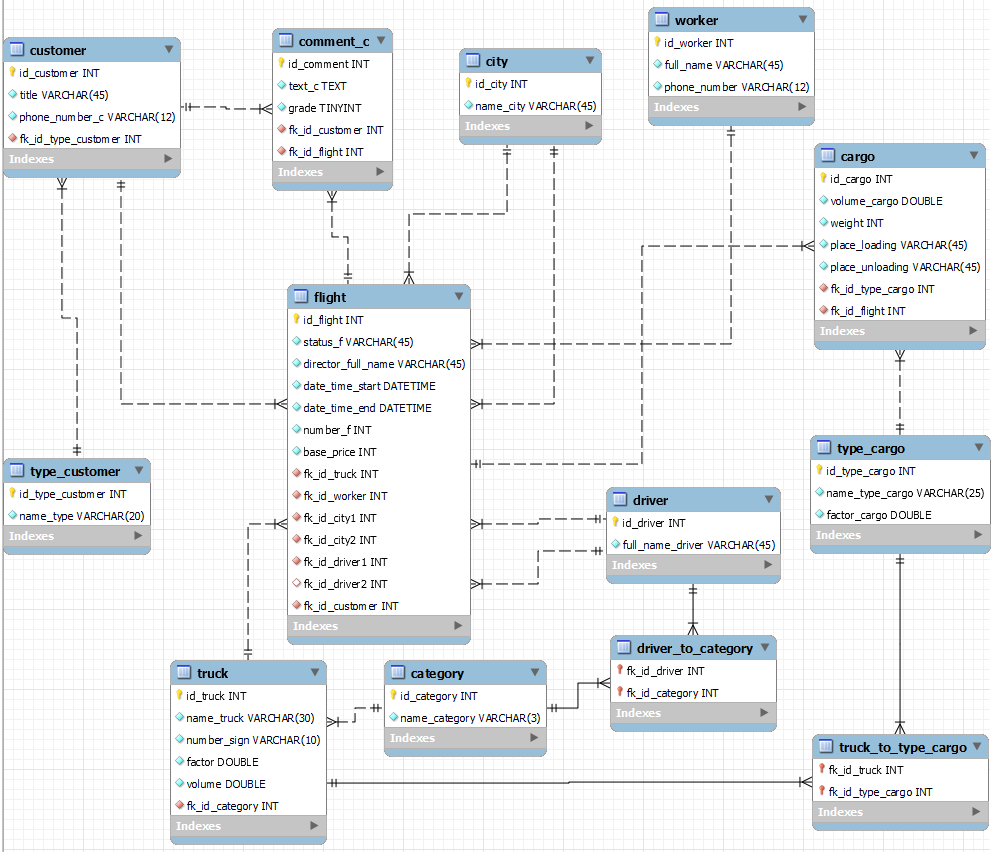
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №4  Реализация SQL-запросов базы данных  Тема: грузоперевозки (В-13) | Ф.И.О. | Козякин Андрей Дмитриевич |
| Группа | ИВТ-263 |
| Преподаватель | Соколов Александр |
| Дата сдачи |  |

**Цель работы:** создать 100 SQL-запросов.

**Ход работы**: Сначала рассмотрим схему БД из ЛР 3:



Модификация:

1) ТОП 3 типов груза за последние 3 месяца и их кол-во.

2) Вывести все комментарии (все столбцы), в них ФИО пользователя, его тип, номер рейса и 2 водителей.

3) Для каждого водителя вывести среднюю оценку из комментариев к рейсам, где они участвовали.

Код запросов:

/\*Запросы\*/

/\*Блок с DELETE\*/

/\*удалить все данные с таблицы bad\*/

DELETE FROM bad

WHERE

(bad\_id);

/\*удалить саму таблицу(она создана для примера)\*/

drop table bad;

/\*удалить заказчика, у которого в номере есть 666\*/

DELETE FROM customer

WHERE

(id\_customer

AND phone\_number\_c LIKE '%666%');

/\*удалить категорию E\*/

DELETE FROM category

WHERE

(id\_category AND name\_category = 'E');

/\*удалить строку у городов, где начало на "Мух"\*/

DELETE FROM city

WHERE

(id\_city AND name\_city LIKE 'Мух%');

/\*удалить строку данных среди типов заказчиков где название "Несуществующий тип"\*/

DELETE FROM type\_customer

WHERE

(id\_type\_customer

AND name\_type = 'Несуществующий тип');

/\*Удалить строку грузовика, если его коэффициент более 1.5\*/

DELETE FROM truck

WHERE

(id\_truck AND factor > 1.5);

/\*Блок с UPDATE\*/

UPDATE city

SET

name\_city = 'Волгоград'

WHERE

(id\_city AND name\_city = 'Волгорад');/\*исправление ошибки в имени города\*/

UPDATE worker

SET

full\_name = 'Степанов Данила Валерьевич'

WHERE

(id\_worker

AND full\_name = 'Степанов Данил Валерьевич');/\*исправление ошибки в фио у одного из работников\*/

UPDATE type\_cargo

SET

factor\_cargo = 1.9

WHERE

((id\_type\_cargo) AND (factor\_cargo > 1.8)

AND (factor\_cargo < 1.9));/\*повышение размера коэффициента типа груза\*/

UPDATE type\_customer

SET

name\_type = 'Физическое лицо'

WHERE

((id\_type\_customer)

AND name\_type LIKE 'Физ.%');/\*обновление названия одного типа заказчика\*/

UPDATE flight

SET

fk\_id\_truck = 3

WHERE

((number\_f = 10) AND (fk\_id\_truck = 2));/\*замена грузовика в 10 будущем рейсе\*/

UPDATE customer

SET

title = 'Перевозчик (не Стетхем)'

WHERE

(id\_customer AND title = 'Перевозчик');/\*изменить название заказчика\*/

/\*Блок с SELECT INTO или INSERT SELECT\*/

/\*пробная таблица, в которую мы перекопируем данные из truck (не все)\*/

CREATE TABLE clone\_truck (

name\_tr VARCHAR(30) NOT NULL,

number\_tr VARCHAR(10) NOT NULL,

factor\_tr DOUBLE NOT NULL

);

/\*перекопировали в клона 3 столбца из truck\*/

insert into clone\_truck (name\_tr, number\_tr, factor\_tr) select name\_truck, number\_sign, factor from truck;

drop table clone\_truck;/\*удаляем клона\*/

/\*клон рейсов (не полный)\*/

CREATE TABLE clone\_2 (

id INT NOT NULL PRIMARY KEY,

date\_st DATETIME NOT NULL,

date\_end DATETIME NOT NULL

);

/\*перекопировали в клона 3 столбца из flight\*/

insert into clone\_2 (id, date\_st, date\_end) select id\_flight, date\_time\_start, date\_time\_end from flight;

drop table clone\_2;/\*удалили таблицу клона рейсов\*/

/\*3 клон, совмещающий название и номер каждого заказчика\*/

CREATE TABLE clone\_cus (

clone\_title VARCHAR(45) NOT NULL,

clone\_phone\_number\_c VARCHAR(12) NOT NULL

);

insert into clone\_cus (clone\_title, clone\_phone\_number\_c)

select title, phone\_number\_c

from customer;

drop table clone\_cus;/\*удалили таблицу клона заказчиков\*/

/\*Блок с SELECT DISTINCT, WHERE, AND, OR, BETWEEN, IS NULL, AS\*/

select distinct name\_truck as 'Названия грузовика'

from truck;/\*вывод всех имён без повторов, daf 105 1 раз, хотя их 2 есть\*/

SELECT

\*

FROM

flight

WHERE

(base\_price > 46000);/\*вывести все рейсы (столбцы все), базовая стоимость которых больше 46000, только 1 подходит\*/

SELECT

\*

FROM

flight

WHERE

((date\_time\_start > '2022-01-28 15:02:42')

AND (date\_time\_start < '2022-03-31 13:13:32'));/\*вывести строки рейсов, у которых дата начала больше 1 даты и меньше 2 даты (даты указаны)\*/

SELECT DISTINCT

place\_unloading

FROM

cargo;/\*вывести все места прибытия груза без повторений, из 10 вывели меньше 10\*/

SELECT

\*

FROM

flight

WHERE

(fk\_id\_driver2 IS NULL);/\*вывести только те рейсы, где 1 водитель (2 null)\*/

select base\_price as 'Базовая стоимость',

date\_time\_start as 'Начало рейса',

date\_time\_end as 'Конец рейса'

from flight

where (base\_price between 33000 and 37000);/\*вывести столбцы рейсов с датами и ценой, у которых цена от 33000 до 37000, могут быть рейсы, равные левой или правой границе\*/

SELECT

\*

FROM

flight

WHERE

(fk\_id\_driver1 IS NOT NULL

AND fk\_id\_driver2 IS NOT NULL);/\*вывести рейсы, у которых оба водителя не null\*/

SELECT

full\_name, phone\_number

FROM

worker

WHERE

(id\_worker = 4);

SELECT DISTINCT

status\_f AS 'Статус рейса'

FROM

flight

LIMIT 2;/\*вывести статусы из всех рейсов (только 2 и без повторений)\*/

SELECT

\*

FROM

cargo

WHERE

((weight BETWEEN 5000 AND 6700)

OR (weight BETWEEN 7400 AND 9040));/\*вывести строки с грузами, вес которых принадлежит одному из промежутков\*/

SELECT

id\_customer, title

FROM

customer

WHERE

(fk\_id\_type\_customer IN (1));/\*вывод названия и id заказчиков, у которых id типа 1, можно через =\*/

SELECT

number\_f, fk\_id\_driver1, fk\_id\_driver2, status\_f

FROM

flight

WHERE

(status\_f IN ('В процессе'));/\*вывод столбцов строк рейсов, которые в процессе\*/

SELECT

phone\_number AS 'Nomer telefona', full\_name AS 'FIO'

FROM

worker;/\*просто выводим работников с их телефонами и новыыми подписями колонок\*/

SELECT

\*

FROM

flight

WHERE

(base\_price IN (46000));/\*выводим только те рейсы, где базовая стоимость 46000\*/

/\*Суперзапрос SELECT, DISTINCT, WHERE, AND/OR/NOT, IN, BETWEEN\*/

select status\_f as 'СТАТУС',

number\_f as 'НОМЕР',

base\_price as 'СТОИМОСТЬ БЕЗ КОЭФФИЦИЕНТОВ'

from flight

where ((base\_price between 39000 and 47000)

and ((date\_time\_start >= '2022-03-01 00:00'))

and (date\_time\_end <= '2022-05-25 23:00'))

order by base\_price asc;/\*вывести столбцы с номером, статусом и стоимостью рейсов, где ограничения стоимости и промежутки дат начала и конца\*/

/\*Блок с GROUP CONCAT\*/

select group\_concat('/',

name\_category,

'/')

from category;/\*вывести в одну строку все категории\*/

select group\_concat(volume\_cargo,

'\_')

from cargo

where (place\_unloading like '%METRO%');/\*вывести в строку объёмы грузов, у которых в месте погрузки есть вхождение METRO\*/

select group\_concat(' ',

title)

from customer

group by fk\_id\_type\_customer;/\*вывести заказчиков, объединив в строку по каждому id типа\*/

select id\_cargo,

group\_concat(volume\_cargo,

'/',

weight,

'/',

place\_loading,

'/',

place\_unloading)

as 'Объём/вес/место погрузки/место выгрузки'

from cargo

group by id\_cargo;/\*вывести таблицу грузов с помощью group concat\*/

/\*Блок с WITH\*/

with cte\_1 as

(select

\*

from worker)

select

\*

from cte\_1;/\*как бы отобразить общее табличное выражение из всей таблицы работников\*/

with cte\_2 as

(select name\_truck,

number\_sign

from truck)

select

\*

from cte\_2

where (name\_truck != 'Daf 105');/\*как бы отобразить общее табличное выражение из таблицы грузовиков, в общее только название и номер и где не Daf 105\*/

with cte\_3 as (select flight.number\_f,

customer.title from flight

join customer

on (flight.fk\_id\_customer = customer.id\_customer

and ((customer.title = 'ОРВС')

or (customer.title = 'Перевозчик')))

order by flight.number\_f) select

\*

from cte\_3;/\*сохранить в общее табличное выражение номер рейса(не id) и присоединённое поле заказчика, который ОРВС или Перевозчик\*/

/\*Блок с JOIN: INNER, OUTER (LEFT, RIGHT, FULL), CROSS, NATURAL\*/

SELECT

driver.full\_name\_driver, worker.full\_name

FROM

driver

CROSS JOIN

worker;/\*перекрёсное соединение столбцов фио водителей с фио работников, если нет условия, то можно для перекреста обычный join\*/

SELECT

c.title, t.name\_type

FROM

customer c

JOIN

type\_customer t ON (c.fk\_id\_type\_customer = t.id\_type\_customer);/\*соединить заказчиков к их типу, слева заказчик, справа их тип, все без id, where можно вместо on\*/

/\*INNER JOIN для временных таблиц\*/

CREATE TABLE first\_t (

id\_f INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

title\_f VARCHAR(15) NOT NULL,

digit\_f DOUBLE NOT NULL

);

CREATE TABLE second\_t (

id\_s INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

title\_s VARCHAR(15) NOT NULL,

digit\_s DOUBLE NOT NULL

);

insert into first\_t (title\_f, digit\_f) values

('Everyday', 12.5),

('Man', 16.15),

('g63', 121.41),

('user', 1.5);

insert into second\_t (title\_s, digit\_s) values

('Everyday', 12.5),

('MGW', 1.15),

('g63', 121.41),

('us', 1.5);

SELECT

f.title\_f, f.digit\_f, s.title\_s, s.digit\_s

FROM

first\_t f

INNER JOIN

second\_t s ON ((f.id\_f = s.id\_s)

AND (f.title\_f = s.title\_s

AND f.digit\_f = s.digit\_s));/\*присоединить те строки, которые равны по содержанию и id\*/

drop table first\_t;

drop table second\_t;

/\*Остальные JOIN\*/

SELECT

c.name\_category, t.name\_truck, t.number\_sign

FROM

category c

LEFT JOIN

truck t ON (c.id\_category = t.fk\_id\_category);/\*соединение столбцов по равенству внешних ключей категории, у D1E нет грузовиков, поэтому null\*/

SELECT

t.name\_truck, t.number\_sign, c.name\_type\_cargo

FROM

truck t

CROSS JOIN

type\_cargo c;/\*случай, когда все бы грузовики смогли бы возить каждый из типов груза\*/

SELECT

f.number\_f, f.base\_price, t.number\_sign, c.title

FROM

flight f

JOIN

truck t ON (f.fk\_id\_truck = t.id\_truck)

JOIN

customer c ON (f.fk\_id\_customer = c.id\_customer)

ORDER BY f.number\_f;/\*присоединить к 2 столбцам номера и стоимости рейса названия заказчиков и номер назначенного грузовика\*/

SELECT

m.fk\_id\_truck,

m.fk\_id\_type\_cargo,

t.name\_truck,

p.name\_type\_cargo

FROM

truck\_to\_type\_cargo m

RIGHT JOIN

truck t ON (m.fk\_id\_truck = t.id\_truck)

RIGHT JOIN

type\_cargo p ON (m.fk\_id\_type\_cargo = p.id\_type\_cargo);/\*вывод id грузовика и его типов груза со значениями этих id\*/

select flight.number\_f as 'Номер рейса',

flight.date\_time\_start as 'Дата начала',

flight.date\_time\_end as 'Дата окончания',

worker.full\_name as 'ФИО работника',

city.name\_city as 'Город отправления',

c.name\_city as 'Город прибытия'

from flight

join worker

on (worker.id\_worker = flight.fk\_id\_worker)

join city

on (flight.fk\_id\_city1 = city.id\_city)

join city c

on (flight.fk\_id\_city2 = c.id\_city)

order by flight.number\_f;/\*как бы новая таблица с номером и датами рейса, полными именами работника и 2 городами\*/

select cargo.weight as 'Вес груза',

type\_cargo.name\_type\_cargo as 'Тип груза',

flight.base\_price as 'Базовая стоимость всего рейса'

from cargo

join type\_cargo

on (cargo.fk\_id\_type\_cargo = type\_cargo.id\_type\_cargo)

join flight

on (cargo.fk\_id\_flight = flight.id\_flight)

order by type\_cargo.name\_type\_cargo;/\*вес груза с типом и стоимостью перевозки в рейсе, участвуют 3 таблицы\*/

SELECT

comment\_c.grade, comment\_c.text\_c, customer.title

FROM

comment\_c

JOIN

customer ON (comment\_c.fk\_id\_customer = customer.id\_customer)

ORDER BY comment\_c.grade DESC;/\*вывод коммента с оценкой и его автора(заказчика)\*/

SELECT

f.id\_flight,

f.status\_f,

f.director\_full\_name,

f.date\_time\_start,

f.date\_time\_end,

f.number\_f,

f.base\_price,

t.name\_truck,

w.full\_name,

y.name\_city,

n.name\_city,

i.full\_name\_driver,

v.full\_name\_driver,

c.title

FROM

flight f

LEFT JOIN

truck t ON (f.fk\_id\_truck = t.id\_truck)

LEFT JOIN

worker w ON (f.fk\_id\_worker = w.id\_worker)

LEFT JOIN

city y ON (f.fk\_id\_city1 = y.id\_city)

LEFT JOIN

city n ON (f.fk\_id\_city2 = n.id\_city)

LEFT JOIN

driver i ON (f.fk\_id\_driver1 = i.id\_driver)

LEFT JOIN

driver v ON (f.fk\_id\_driver2 = v.id\_driver)

LEFT JOIN

customer c ON (f.fk\_id\_customer = c.id\_customer);/\*вывести полностью таблицу рейсов, заменив внешние ключи конкретными соответствующими данными (не fk\_id)\*/

SELECT

customer.title, customer.phone\_number\_c, n.name\_type

FROM

customer

LEFT JOIN

type\_customer n ON (customer.fk\_id\_type\_customer = n.id\_type\_customer);

/\*Блок с GROUP BY, ORDER BY, COUNT, LIMIT и так далее\*/

SELECT

\*

FROM

flight

ORDER BY date\_time\_start DESC

LIMIT 6;/\*вывод 6 рейсов по дате начала по убыванию\*/

SELECT

\*

FROM

comment\_c

ORDER BY grade ASC;/\*вывод всех отзывов по оценке по возрастанию\*/

SELECT

id\_customer, title

FROM

customer

WHERE

(title LIKE CONCAT('%', 'ИП', '%'));/\*вывести из заказчиков поля с id и названием только тех, у кого есть вхождение "ИП" в названии\*/

SELECT

COUNT(1)

FROM

category;/\*вывод числа категорий\*/

SELECT

SUM(base\_price)

FROM

flight;/\*вывод суммы всей выручки за всё время рейсов\*/

SELECT

AVG(base\_price)

FROM

flight;/\*вывод средней базовой стоимости рейса\*/

SELECT

AVG(weight)

FROM

cargo;/\*вывод среднего веса среди всех грузов\*/

SELECT

MIN(volume)

FROM

truck;/\*вывод наименьшего объёма у грузовиков\*/

SELECT

MIN(name\_type\_cargo)

FROM

type\_cargo;/\*вывод самого короткого названия типа груза\*/

SELECT

COUNT(id\_worker), full\_name

FROM

worker

GROUP BY full\_name

ORDER BY full\_name DESC

LIMIT 3;/\*вывод 2 столбцов (кол-ва id у работника с таким фио и его фио), группируя при этом по фио по убыванию алфавита? 3 работников\*/

SELECT

number\_f, base\_price

FROM

flight

GROUP BY number\_f

HAVING (base\_price >= 32000

AND base\_price <= 45000)

ORDER BY base\_price ASC;/\*вывод рейсов(номера и базовой цены) по стоимости больше 32000 и меньше 45000 с сортировкой по возрастанию цены\*/

SELECT

\*

FROM

category

HAVING (name\_category LIKE CONCAT('%', 'E', '%'));/\*вывести всю таблицу с категориями где есть буква E, то есть все кроме кратегории C\*/

SELECT

full\_name, phone\_number

FROM

worker

WHERE

(full\_name LIKE CONCAT('Андреев', '%'));/\*вывести ФИО и номер телефона среди работников, у которых фамилия Андреев как вхождение\*/

SELECT

COUNT(fk\_id\_truck), fk\_id\_truck

FROM

flight

WHERE

(status\_f LIKE CONCAT('Выполнен'))

GROUP BY fk\_id\_truck;/\*вывод того, сколько каждый грузовик совершил рейсов, считаются только выполненные\*/

SELECT

\*

FROM

type\_cargo

WHERE

(name\_type\_cargo LIKE 'На%'

AND factor\_cargo > 1.3);/\*вывод всех строк типов груза, где начало на "На" и коэффициент типа более 1.3\*/

SELECT

full\_name

FROM

worker

WHERE

(full\_name LIKE '%ич')

LIMIT 2;/\*вывод фио работников, у которых конец на ич, то есть мужчин, также не всех, а только первых 2, без этого лимита 3\*/

select status\_f as 'Статус',

count(\*) as 'Раз'

from flight

group by status\_f

having count(\*) = (select min(nor)

from (select status\_f,

count(\*) as nor

from flight

group by status\_f) flight);/\*вывести самый(самые) нераспространённые статусы рейсов\*/

SELECT

grade AS 'Оценка', COUNT(\*) AS 'Раз'

FROM

comment\_c

GROUP BY grade

HAVING COUNT(\*) = (SELECT

MAX(nor)

FROM

(SELECT

grade, COUNT(\*) AS nor

FROM

comment\_c

GROUP BY grade) comment\_c);/\*вывести самую частую оценку из всех комментариев\*/

/\*Блок с UNION\*/

SELECT

full\_name

FROM

worker

UNION SELECT

full\_name\_driver

FROM

driver

ORDER BY full\_name;/\*вывести в 1 столбец работников и водителей\*/

SELECT

phone\_number\_c

FROM

customer

UNION SELECT

phone\_number

FROM

worker

ORDER BY phone\_number\_c;/\*вывести в столбец все номера заказчиков и работников по возрастанию\*/

/\*1 пример на новых таблицах, их потом удаляем\*/

CREATE TABLE t1 (

id1 INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

n1 VARCHAR(10) NOT NULL,

date1 DATE NOT NULL

);

CREATE TABLE t2 (

id2 INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

n2 VARCHAR(10) NOT NULL,

date2 DATE NOT NULL

);

insert into t1 (n1, date1) values

('sdf', '2022-12-13'),

('FWS', '2021-03-19'),

('D23', '2020-11-16');

insert into t2 (n2, date2) values

('sdf', '2022-12-13'),

('AF', '2020-02-04'),

('DDDDW', '2018-08-13');

SELECT

\*

FROM

t1

UNION ALL SELECT

\*

FROM

t2

ORDER BY n1;/\*вывести как бы 2 таблицы вертикально в одни столбцы (таблицы идентичные по колонкам), без all дубликаты не повторяются в выводе, по алфавиту названий n1 n2\*/

drop table t1;

drop table t2;

select name\_truck as 'Названия грузовиков и типов груза',

factor

from truck

union select name\_type\_cargo, factor\_cargo

from type\_cargo;/\*объединение 2 столбцов с коэффициентом и названием у truck и type\_cargo, без order by\*/

SELECT

\*

FROM

category

UNION ALL SELECT

\*

FROM

city

ORDER BY id\_category;/\*просто объединить 2 таблицы категорий и городов, просто как пример, порядок по id\*/

SELECT

id\_driver, full\_name\_driver

FROM

driver

WHERE

(id\_driver = 1)

UNION SELECT

id\_truck, name\_truck

FROM

truck

WHERE

(id\_truck = 1);/\*объединить в колонки названия грузовиков и фио водителей с их id, где у каждого id = 1\*/

SELECT

title

FROM

customer

UNION ALL SELECT

full\_name

FROM

worker

ORDER BY title; /\*вывести в столбец заказчиков и работников\*/

SELECT

volume\_cargo

FROM

cargo

UNION ALL SELECT

volume

FROM

truck

ORDER BY volume\_cargo DESC;/\*все объёмы грузов и грузовиков в 1 столбец\*/

/\*Аналитические запросы:

1. Показать количество отмененных рейсов за последний месяц v

2. Показать общую выручку с рейсов за 1 год. v

3. Вычислить самый распространённый тип груза за последние 3 месяца.

4. Узнать среднюю конечную стоимость рейса за 1 месяц в течение года. v

5. Выяснить, какой город является самым посещаемым в качестве города загрузки груза за 3 года.

\*/

select count(1) as 'Количество отменённых рейсов за апрель 2022 года'

from flight

where ((status\_f = 'Отменён')

and (date\_time\_start >= '2022-04-01 00:00')

and (date\_time\_end <= '2022-04-30 23:59'));/\*1. Показать количество отмененных рейсов за последний месяц\*/

select sum(base\_price) as 'Общая выручка с рейсов за 1 год'

from flight

where ((date\_time\_start >= '2022-01-01 00:00')

and (date\_time\_end <= '2022-12-31 23:59')

and (status\_f = 'Выполнен'));/\*2. Показать общую выручку с рейсов за 1 год. Без учёта коэффициентов типов груза и грузовиков\*/

/\*3. Вычислить ТОП3 типа груза за последние 3 месяца и их кол-во (1 - екпр 4 раза, 2 - Название типа 2 раза).\*/

/\*4. Узнать среднюю конечную стоимость рейса за 1 месяц в течение года. Условие, что смотрим на дату конца, которая в апреле 2022 и статус рейса не важен\*/

select avg(base\_price) as 'Средняя базовая стоимость рейса за апрель в течение 2022 года'

from flight

where ((date\_time\_end >= '2022-04-01 00:00')

and (date\_time\_end <= '2022-04-30 23:59'));

/\*5. Выяснить, какой город является самым посещаемым в качестве города загрузки груза за 3 года.\*/

/\*Справочные (оперативные запросы):

1. Показать расписание на текущие сутки. v

2. Показать расписание на ближайшие 3 дня. v

3. Отсортировать текущие рейсы за месяц по дате (от самой ближайшей). v

4. Показать рейсы в город прибытия (один и тот же). v

5. Отсортировать отзывы по оценке (по возрастанию). v

\*/

SELECT

\*

FROM

flight

WHERE

((date\_time\_start >= '2022-04-27 00:00')

AND (date\_time\_start <= '2022-04-27 23:59'));/\*1. Показать расписание на текущие сутки. (27.04.2022)\*/

SELECT

\*

FROM

flight

WHERE

((date\_time\_start >= '2022-04-01 00:00')

AND (date\_time\_start <= '2022-04-03 23:59'));/\*2. Показать расписание на ближайшие 3 дня.\*/

SELECT

\*

FROM

flight

WHERE

((date\_time\_start >= '2022-04-01 00:00')

AND (date\_time\_start <= '2022-04-30 23:59')

AND (date\_time\_end <= '2022-04-30 23:59'))

ORDER BY date\_time\_start;/\*3. Отсортировать текущие рейсы за месяц по дате (от самой ближайшей).\*/

SELECT

\*

FROM

flight

WHERE

(fk\_id\_city1 = 2);/\*4. Показать рейсы в город прибытия(выгрузки) (один и тот же).\*/

SELECT

\*

FROM

comment\_c

ORDER BY grade ASC;/\*5. Отсортировать отзывы по оценке (по возрастанию).\*/

/\*Вложенные SELECT с GROUP BY, ALL, ANY, EXISTS (3-5 шт.)\*/

SELECT

\*

FROM

flight

WHERE

fk\_id\_city1 = (SELECT

id\_city

FROM

city

WHERE

name\_city = 'Краснодар');/\*вывести рейсы, у которых город загрузки груза Краснодар\*/

SELECT

number\_f, base\_price, director\_full\_name

FROM

flight

WHERE

((fk\_id\_driver2 IS NULL)

AND (fk\_id\_customer = (SELECT

id\_customer

FROM

customer

WHERE

title = 'Перевозчик'))

AND fk\_id\_worker = (SELECT

id\_worker

FROM

worker

WHERE

full\_name = 'Кузьмина Оксана Валерьевна'));/\*рейсы, у которых 1 водитель, заказчик Перевозчик и работник Кузьмина Оксана Валерьевна\*/

SELECT

\*

FROM

flight

WHERE

EXISTS( SELECT

\*

FROM

city

WHERE

name\_city IS NULL);/\*вывести рейсы, у которых в городах нет имени, но есть id, такого нет, поэтому пустая таблица вывода\*/

SELECT

\*

FROM

flight

WHERE

fk\_id\_worker = ANY (SELECT

id\_worker

FROM

worker

WHERE

phone\_number = '89613338345');/\*вывести рейсы, у которых работник имеет указанный номер\*/

SELECT

name\_truck, number\_sign

FROM

truck

WHERE

fk\_id\_category = ALL (SELECT

id\_category

FROM

category

WHERE

name\_category = 'С1E');/\*искать номер и грузовик, у которого категория C1E\*/

/\*Запросы со строковыми функциями СУБД, с функциями работы с датами временем (форматированием дат), с арифметическими функциями (5-7 шт.)\*/

/\*пример на новой таблице\*/

CREATE TABLE strings (

id\_s INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

s1 VARCHAR(15) NOT NULL,

s2 VARCHAR(35) NOT NULL,

s3 VARCHAR(50) NOT NULL

);

insert into strings (s1, s2, s3) values

('FFF', '3dg', '234fd'),

('32321', ' Rfag', '34fd '),

(' tWdWf2 ', 'Name: ', 'John');

SELECT

LENGTH(s1)

FROM

strings

WHERE

id\_s = 1;/\*длина строки колоночки\*/

SELECT

REVERSE(s1)

FROM

strings

WHERE

id\_s = 3;/\*перевернутый вывод строки\*/

SELECT

LOWER(s1)

FROM

strings

WHERE

id\_s = 3;/\*значение колонки в малом регистре\*/

SELECT

REPEAT(s2, 4)

FROM

strings

WHERE

id\_s = 3;/\*4 раза повторить Name\*/

truncate table strings;/\*очистить таблицу без её удаления\*/

drop table strings;/\*удалить таблицу (можно сразу с данными без truncate), все строки ищутся по id через =\*/

/\*Примеры с БД грузоперевозок\*/

select name\_truck as 'Название грузовика',

length(name\_truck) as 'Длина в символах'

from truck

where id\_truck = 2;/\*нашли длину названия грузовика с определённым id\*/

select full\_name\_driver, length(full\_name\_driver) from driver;

/\*вывод имени заказчиков в большом регистре и перевёрнутые номера\* по алфавиту имени\*/

select upper(title) as 'Название',

concat('(', reverse(phone\_number\_c), ')')

from customer

order by title;

SELECT

\*

FROM

driver

WHERE

(LOWER(full\_name\_driver) LIKE CONCAT('%вА%'));/\*вывести водителей, имеющих в полном имени "ва", в независимости от регистров\*/

/\*вывести оценку, текст коммента и название автора-заказчика с использованием строковых функций и условия, что длина названия заказчика от 5 до 10 символов, сортируем по убыванию оценки\*/

SELECT

m.grade, UPPER(m.text\_c), CONCAT('|', customer.title, '|')

FROM

comment\_c m

JOIN

customer ON (LENGTH(customer.title) BETWEEN 5 AND 20

AND (m.fk\_id\_customer = customer.id\_customer))

ORDER BY grade DESC;

/\*Запрос на арифметику\*/

/\*сумма тех весов, где "База" есть в месте погрузки или выгрузки, а также столбцы с полным перечислением мест погрузки и выгрузки груза\*/

SELECT

SUM(cargo.weight) AS 'Сумма весов грузов, где фигурирует "База"',

GROUP\_CONCAT('/', cargo.place\_loading, '/'),

GROUP\_CONCAT('/', cargo.place\_unloading, '/')

FROM

cargo

WHERE

((LOWER(cargo.place\_loading) LIKE CONCAT('%БАЗА%'))

OR (LOWER(cargo.place\_loading) LIKE CONCAT('%БАЗА%')));

/\*///////////////////////////////////////////////////МОДИФИКАЦИЯ///////////////////////////////////////////////////\*/

/\*1) ТОП 3 типов груза за последние 3 месяца и их кол-во (1 - Насыпной (4 раза), 2 - Наливной (2 раза) -> пример.\*/

/\*запрос на всё, для проверок\*/

SELECT

\*

FROM

flight

JOIN

cargo ON cargo.fk\_id\_flight = flight.id\_flight

RIGHT JOIN

type\_cargo ON cargo.fk\_id\_type\_cargo = type\_cargo.id\_type\_cargo;

/\*Сам запрос\*/

SELECT

type\_cargo.name\_type\_cargo, COUNT(fk\_id\_flight) AS raz

FROM

cargo

JOIN

flight ON cargo.fk\_id\_flight = flight.id\_flight

RIGHT JOIN

type\_cargo ON cargo.fk\_id\_type\_cargo = type\_cargo.id\_type\_cargo

WHERE

flight.date\_time\_start IS NULL

OR (flight.date\_time\_start >= '2020-03-01 00:00'

AND flight.date\_time\_end <= '2022-05-15 00:00')

GROUP BY cargo.fk\_id\_type\_cargo

ORDER BY raz DESC

limit 3;

/\*2) Вывести все комментарии (все столбцы), в них ФИО пользователя, его тип, номер рейса и 2 водителей.\*/

SELECT

comment\_c.id\_comment,

comment\_c.text\_c,

comment\_c.grade,

customer.title,

type\_customer.name\_type,

flight.number\_f,

l.full\_name\_driver,

i.full\_name\_driver

FROM comment\_c

LEFT JOIN

customer ON (comment\_c.fk\_id\_customer = customer.id\_customer)

LEFT JOIN

type\_customer ON (customer.fk\_id\_type\_customer = type\_customer.id\_type\_customer)

LEFT JOIN

flight ON (comment\_c.fk\_id\_flight = flight.id\_flight)

LEFT JOIN

driver l ON (flight.fk\_id\_driver1 = l.id\_driver)

LEFT JOIN

driver i ON (flight.fk\_id\_driver2 = i.id\_driver);

/\*3) Для каждого водителя вывести среднюю оценку из комментариев к рейсам, где они участвовали.\*/

SELECT

AVG(ROUND(comment\_c.grade, 3)), d.full\_name\_driver

FROM comment\_c

LEFT JOIN

flight ON (comment\_c.fk\_id\_flight = flight.id\_flight)

LEFT JOIN

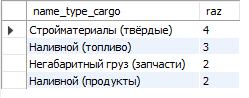
driver d ON d.id\_driver = flight.fk\_id\_driver1 OR d.id\_driver = flight.fk\_id\_driver2

GROUP BY d.id\_driver

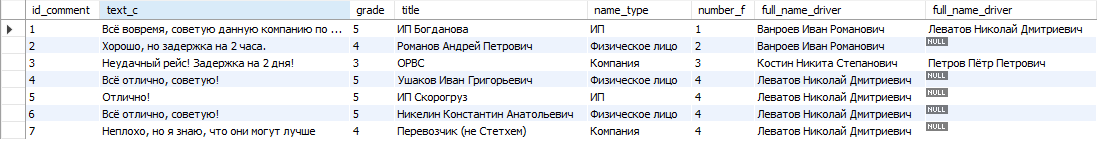
ORDER BY d.full\_name\_driver;

Примеры результатов запросов:

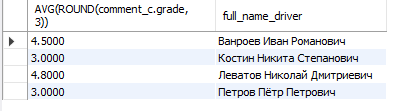
Модификация, 1 запрос:



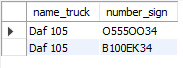
Модификация, 2 запрос:



Модификация, 3 запрос:



Искать номер и грузовик, у которого категория C1E:



Запрос как новая таблица с номером и датами рейса, полными именами работника и 2 городами:

