1. Создание таблиц.

Таблица «Пилоты»:

CREATE TABLE pilot (

id VARCHAR(10) PRIMARY KEY,

surname VARCHAR(50) NOT NULL,

squadron VARCHAR(50) NOT NULL,

hours\_limit INT NOT NULL);

Таблица «Штурман»:

CREATE TABLE navigator (

id VARCHAR(10) PRIMARY KEY,

surname VARCHAR(50) NOT NULL,

squadron VARCHAR(50) NOT NULL,

hours\_limit INT NOT NULL);

Таблица «Рейс»:

CREATE TABLE trip (

id VARCHAR(10) PRIMARY KEY,

destination VARCHAR(50) NOT NULL,

staff VARCHAR(50) NOT NULL,

flight\_time INT NOT NULL,

diff\_factor INT NOT NULL);

Таблица «Полёт»:

CREATE TABLE flights (

record VARCHAR(10) PRIMARY KEY,

date\_month VARCHAR(50) NOT NULL,

pilot VARCHAR(10) NOT NULL,

navigator VARCHAR(10) NOT NULL,

trip VARCHAR(10) NOT NULL,

dep\_count INT NOT NULL,

hours\_count INT NOT NULL);

Зададим внешние ключи.

Родительская таблица «Пилоты» и дочерняя «Полёт»

ALTER TABLE flights

ADD CONSTRAINT fk\_pilot

FOREIGN KEY (pilot)

REFERENCES pilot(id)

ON UPDATE CASCADE;

Родительская таблица «Штурман» и дочерняя «Полёт»

ALTER TABLE flights

ADD CONSTRAINT fk\_navigator

FOREIGN KEY (navigator)

REFERENCES navigator(id)

ON UPDATE CASCADE;

Родительская таблица «Рейс» и дочерняя «Полёт»

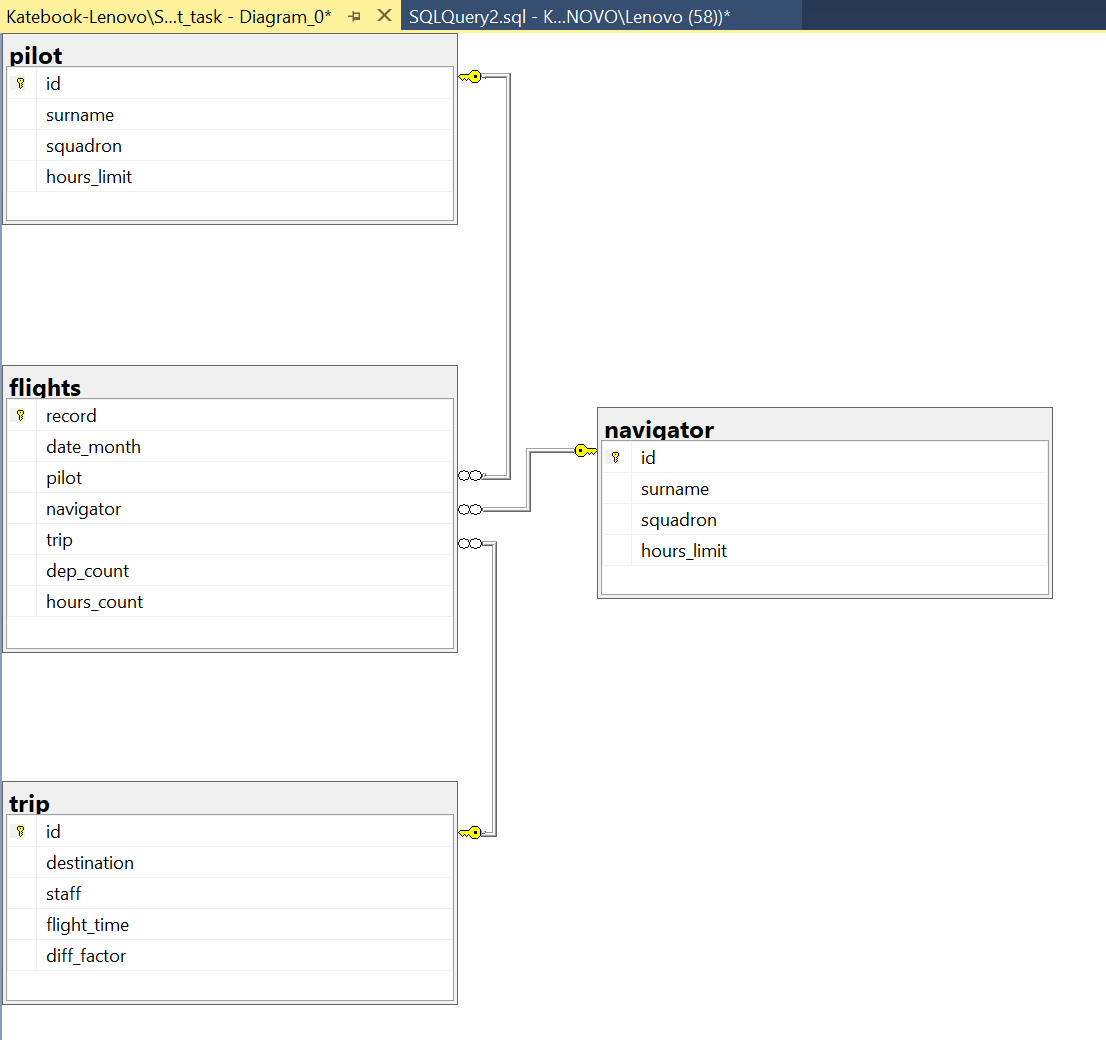
ALTER TABLE flights

ADD CONSTRAINT fk\_trip

FOREIGN KEY (trip)

REFERENCES trip(id)

ON UPDATE CASCADE;



1. Внесение в созданные таблицы записей:

Для таблицы «Пилоты»:

INSERT INTO pilot (id, surname, squadron, hours\_limit)

VALUES ('001', 'Lyapin', 'Nishegorodski', 50),

('002', 'Ivanov', 'Moscowski1', 48),

('003', 'Polivsev', 'Moscowski2', 32),

('004', 'Sviatov', 'Moscowski3', 49),

('005', 'Shybenkov', 'Volgogradski', 50);

Для таблицы «Штурман»:

INSERT INTO navigator (id, surname, squadron, hours\_limit)

VALUES ('001', 'Sidelnikov', 'Moscowski1', 48),

('002', 'Bobylev', 'Moscowski2', 48),

('003', 'Samarin', 'Moscowski3', 50),

('004', 'Platonav', 'Nishegorodski', 49),

('005', 'Kylenkov', 'Volgogradski', 50),

('006', 'Ovchinnikov', 'S.-Peterbyrgski', 49);

Для таблицы «Рейс»:

INSERT INTO trip (id, destination, staff, flight\_time, diff\_factor)

VALUES ('001', 'Moscow', 'Nishegorodski', 0, 1),

('002', 'Perm', 'Nishegorodski', 0, 3),

('003', 'Volgograd', 'Volgogradski', 0, 2),

('004', 'Habarovsk', 'Moscowski2', 0, 4),

('005', 'S.-Peterbyrg', 'Moscowski2', 0, 2),

('006', 'Ischevsk', 'Moscowski3', 0, 2),

('007', 'Myrmansk', 'Moscowski1', 0, 2);

Для таблицы «Полёт»:

INSERT INTO flights (record, date\_month, pilot, navigator, trip, dep\_count, hours\_count)

VALUES ('12500', 'January', '001', '001', '004', 3, 12),

('12501', 'January', '001', '004', '003', 4, 8),

('12502', 'January', '003', '002', '007', 3, 6),

('12503', 'February', '002', '004', '003', 4, 8),

('12504', 'February', '004', '003', '003', 5, 10),

('12505', 'March', '001', '001', '003', 4, 8),

('12506', 'April', '001', '004', '002', 3, 9),

('12507', 'April', '002', '005', '007', 4, 8),

('12508', 'April', '005', '006', '004', 3, 12),

('12509', 'May', '002', '003', '004', 4, 16),

('12510', 'June', '005', '005', '001', 4, 4),

('12511', 'July', '001', '001', '004', 4, 16),

('12512', 'August', '001', '002', '002', 4, 12),

('12513', 'September', '005', '006', '007', 3, 6),

('12514', 'December', '001', '001', '001', 4, 4),

('12515', 'December', '002', '005', '005', 6, 12),

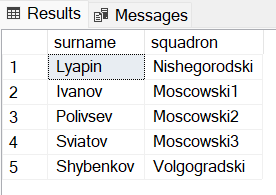
('12516', 'December', '001', '004', '004', 4, 16);

1. Создав запрос получить следующую информацию:
   1. фамилии и авиаотряд для пилотов, имеющих лимит часов полёта более 30;

SELECT surname, squadron

FROM pilot

WHERE hours\_limit > 30;

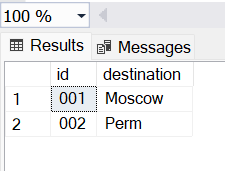


* 1. место назначения для тех рейсов, где используется обслуживающий персонал Нижегородского авиаотряда;

SELECT id, destination

FROM trip

WHERE staff LIKE 'Nishegorodski';



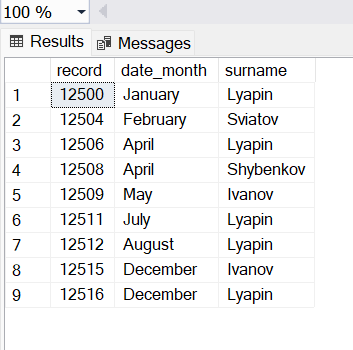
* 1. номер записи, дату и фамилию пилота для тех записей о полётах, где число часов превысило 8.

SELECT f.record, f.date\_month, p.surname

FROM flights AS f

RIGHT JOIN pilot AS p ON f.pilot = p.id

WHERE f.hours\_count > 8;



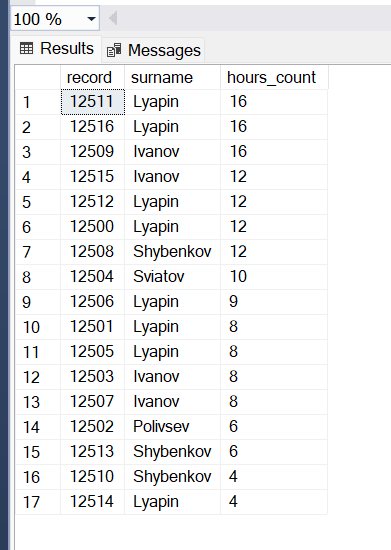
1. На основании данных о полётах вывести все данные в таком формате:
   1. номер записи, фамилия пилота, количество часов. Отсортировать по количеству часов;

SELECT f.record, p.surname, f.hours\_count

FROM flights AS f

RIGHT JOIN pilot AS p ON f.pilot = p.id

ORDER BY f.hours\_count DESC;



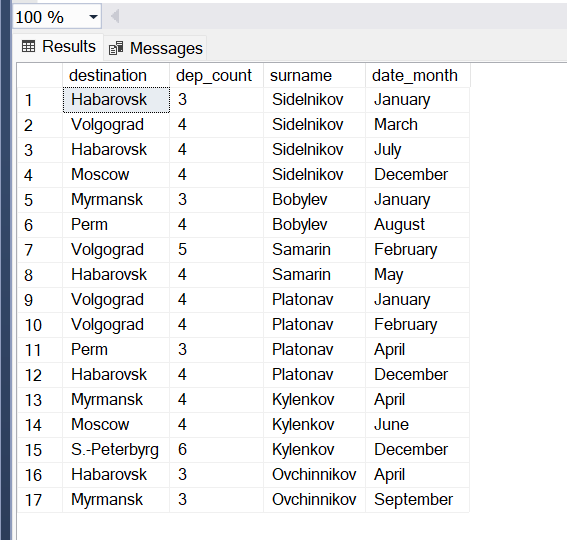
* 1. место назначения рейса, число вылетов, фамилия штурмана, дата.

SELECT t.destination, f.dep\_count, n.surname, f.date\_month

FROM flights AS f

RIGHT JOIN trip AS t ON f.trip = t.id

RIGHT JOIN navigator AS n ON f.navigator = n.id;



1. Вывести:
   1. фамилии штурманов, летавших в Хабаровск или с пилотами того же авиаотряда;

SELECT n.surname

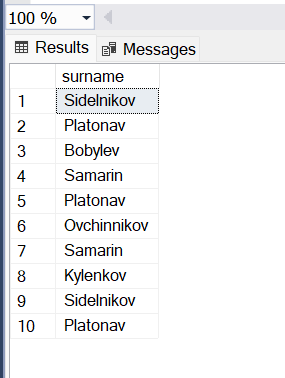
FROM flights AS f

LEFT JOIN navigator AS n ON f.navigator = n.id

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN trip AS t ON f.trip = t.id

WHERE t.destination LIKE 'Habarovsk' OR p.squadron = n.squadron;



* 1. фамилии и лимит часов для тех пилотов, которые работали на рейсах со временем полета более 3 часов не ранее февраля;

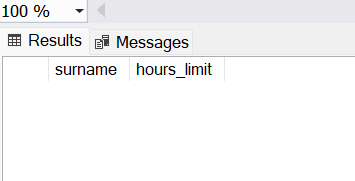
SELECT p.surname, p.hours\_limit

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN trip AS t ON f.trip = t.id

WHERE t.flight\_time > 3 AND f.date\_month NOT LIKE 'January';



***Запрос не вывел данные (по условию время полёта во всех строках = 0).***

* 1. фамилию и авиаотряд штурмана, с которым летал пилот Поливцев более одного раза.

WITH t1 AS (

SELECT pilot,

navigator,

COUNT(f.pilot) AS ranks

FROM flights AS f

GROUP BY f.pilot, f.navigator

)

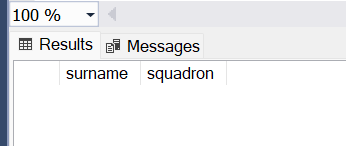
SELECT n.surname, n.squadron

FROM t1

LEFT JOIN navigator AS n ON t1.navigator = n.id

LEFT JOIN pilot AS p ON t1.pilot = p.id

WHERE p.surname LIKE 'Polivsev' AND t1.ranks > 1;



***Запрос не вывел данные (по условию пилот Поливцев летал за всё время 1 раз).***

* 1. идентификатор и место назначения для тех рейсов, на которых работали штурманы из С.-Петербургского авиаотряда. Добавить в вывод фамилии таких штурманов и отсортировать по месту назначения.

SELECT t.id, t.destination, n.surname

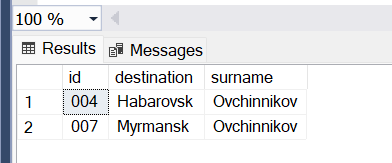
FROM flights AS f

LEFT JOIN navigator AS n ON f.navigator = n.id

LEFT JOIN trip AS t ON f.trip = t.id

WHERE n.squadron LIKE 'S.-Peterbyrgski'

ORDER BY t.destination;



1. Создать запрос для модификации всех значений столбца с суммарным числом часов полета, чтобы он учитывал коэффициент сложности рейса. Вывести новые значения.

***Изменим столбец hours\_count в таблице flights по такому принципу:***

***f.hours\_count + f.hours\_count \* f.hours\_count \* t.diff\_factor / 100***

UPDATE f

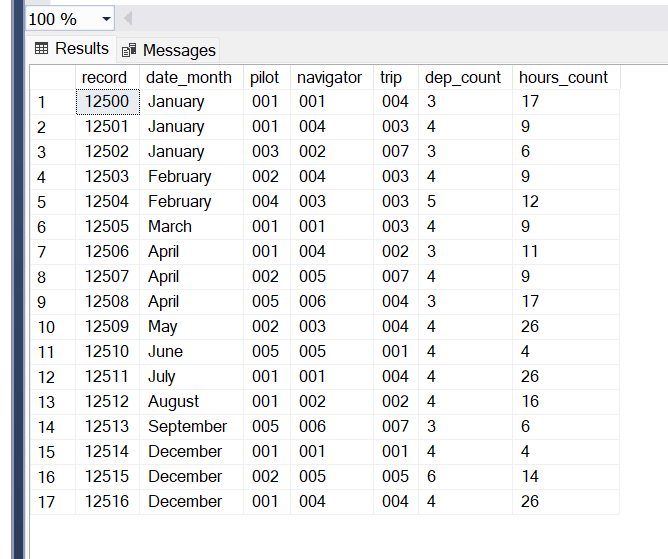
SET f.hours\_count = f.hours\_count + f.hours\_count \* f.hours\_count \* t.diff\_factor / 100

FROM flights f

INNER JOIN trip t ON f.trip = t.id;

SELECT \*

FROM flights;



1. Расширить таблицу с данными о полётах столбцом, содержащим процент налетанных каждым пилотом в рейсе часов от лимита часов. Создать запрос для ввода конкретных значений во все строки таблицы.

Поняла «данные о полётах, содержащие процент налетанных каждым пилотом в рейсе часов от лимита часов» следующим образом:

WITH t2 AS (

SELECT \*,

CONVERT(decimal(10,2), (CAST(f.hours\_count AS decimal(5,2)) / CAST(p. hours\_limit AS decimal(5,2)) \* 100)) perc\_hours

FROM flights AS f

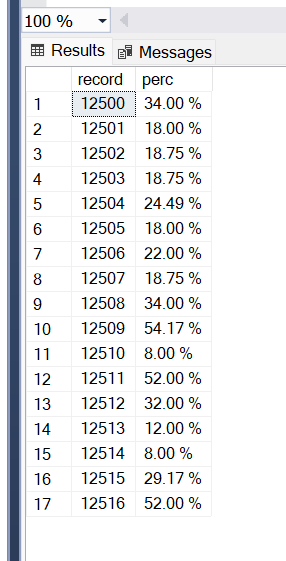
LEFT JOIN pilot AS p ON f.pilot = p.id

)

SELECT record,

CAST(t2.perc\_hours AS VARCHAR(6)) + ' %' AS perc

FROM t2;



Поняла «создать запрос для ввода конкретных значений во все строки таблицы» следующим образом:

ALTER TABLE flights

ADD perc\_fly\_hours VARCHAR(10) NULL;

UPDATE flights

SET perc\_fly\_hours = '34.00 %'

WHERE record = '12500';

… … …

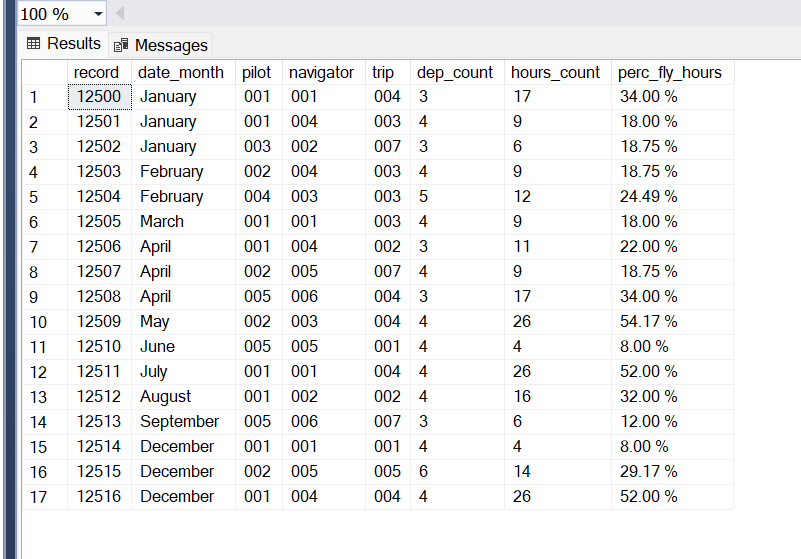
UPDATE flights

SET perc\_fly\_hours = '52.00 %'

WHERE record = '12516';

SELECT \*

FROM flights;



1. Используя операцию IN (NOT IN) реализовать следующие запросы:
   1. кто из пилотов имеет такое же максимальное количество часов, что и какого-либо штурмана из Нижегородского авиаотряда;

WITH t3 AS (

SELECT MAX(hours\_limit) AS max\_hour

FROM navigator

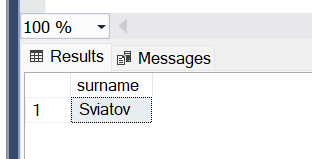
WHERE squadron IN ('Nishegorodski')

)

SELECT p.surname

FROM pilot AS p

RIGHT JOIN t3 ON p.hours\_limit = t3.max\_hour;



* 1. кто из пилотов не летал в Хабаровск до декабря.

WITH t3 AS (

SELECT \*,

SUM(dec\_true) OVER (PARTITION BY surname ORDER BY date\_month) AS ranc

FROM (

SELECT f.record, f.date\_month, t.destination, p.surname,

CASE

WHEN f.date\_month IN ('December') THEN 1

ELSE 0

END AS dec\_true

FROM flights AS f

LEFT JOIN trip AS t ON f.trip = t.id

LEFT JOIN pilot AS p ON f.pilot = p.id

) AS a

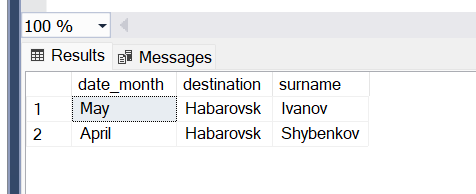
WHERE destination IN ('Habarovsk')

)

SELECT date\_month, destination, surname

FROM t3

WHERE ranc NOT IN (1);



1. Используя операцию UNION получить авиаотряды, где работают пилоты и обслуживающий рейсы персонал.

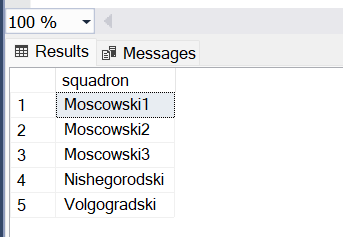
SELECT squadron

FROM pilot

UNION

SELECT staff

FROM trip;



10.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 пары пилот-штурман, работавшие на всех рейсах;

***Уникальное количество рейсов в таблице flights - 6, уникальное количество в таблице trip - 7 (на рейсе 007 в Мурманск не летал ни один пилот).***

WITH t5 AS (

SELECT CONCAT ('pilot ', pilot\_surname, ' - navigator ', navig\_surname) AS pair\_pn,

trip

FROM (

SELECT p.surname AS pilot\_surname, n.surname AS navig\_surname, trip

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id

) AS pairs

)

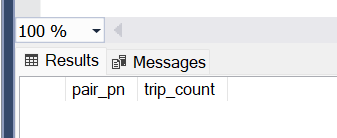
SELECT pair\_pn,

COUNT(DISTINCT(trip)) AS trip\_count

FROM t5

GROUP BY pair\_pn

HAVING COUNT(DISTINCT(trip)) > 6;



***В данных нет пары пилот – штурман, которые работали бы на всех рейсах. Если БД дополнить недостающими рейсами для какой-либо пары пилот-штурман, то она появится в запросе.***

* 1. какие рейсы обслуживаются пилотами только Нижегородского авиаотряда, а штурманами Московского авиаотряда;

WITH t10 AS (SELECT f.trip, p.squadron AS pilot\_squa, n.squadron AS navig\_squa

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id),

t11 AS (

SELECT \*

FROM t10

WHERE EXISTS (SELECT \* FROM pilot

JOIN navigator ON pilot.squadron = navigator.squadron

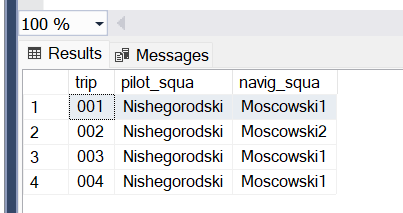
WHERE pilot.squadron = t10.pilot\_squa

HAVING t10.pilot\_squa LIKE 'Nishegorodski' AND t10.navig\_squa IN ('Moscowski1', 'Moscowski2', 'Moscowski3'))

)

SELECT DISTINCT(trip), pilot\_squa, navig\_squa

FROM t11;



* 1. найти штурманов, которые работали с пилотами Нижегородского авиаотряда на всех рейсах со временем полета более двух часов; Необходимо заполнить столбец triptime в таблице trip для того, чтобы запрос не выдавал пустое множество.

UPDATE trip

SET flight\_time = 1

WHERE id = '001';

UPDATE trip

SET flight\_time = 3

WHERE id = '002';

UPDATE trip

SET flight\_time = 4

WHERE id = '003';

UPDATE trip

SET flight\_time = 9

WHERE id = '004';

UPDATE trip

SET flight\_time = 5

WHERE id = '005';

UPDATE trip

SET flight\_time = 4

WHERE id = '006';

UPDATE trip

SET flight\_time = 5

WHERE id = '007';

WITH t12 AS (SELECT n.surname AS navig\_surname, p.surname AS pilot\_surname, p.squadron, t.flight\_time

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id

LEFT JOIN trip AS t ON f.trip = t.id

WHERE t.flight\_time > 2)

SELECT \*

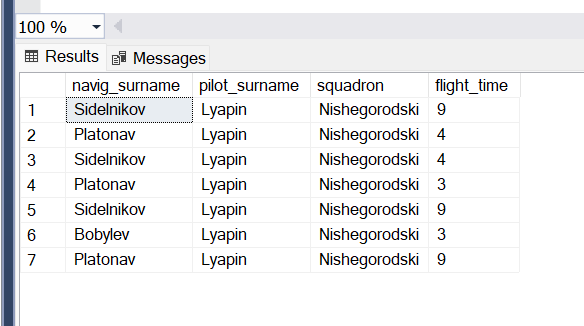
FROM t12

WHERE EXISTS (SELECT \* FROM pilot

JOIN navigator ON pilot.squadron = navigator.squadron

WHERE pilot.squadron = t12.squadron

HAVING t12.squadron LIKE 'Nishegorodski');



* 1. найти пилотов, работавших на всех рейсах, на которых авиаотряд обслуживающего персонала и пилота различны, со всеми штурманами, имеющими лимит часов более 48.

SELECT p.surname AS pilot\_surname, p.squadron, t.staff, n.surname AS navig\_surname, n.hours\_limit

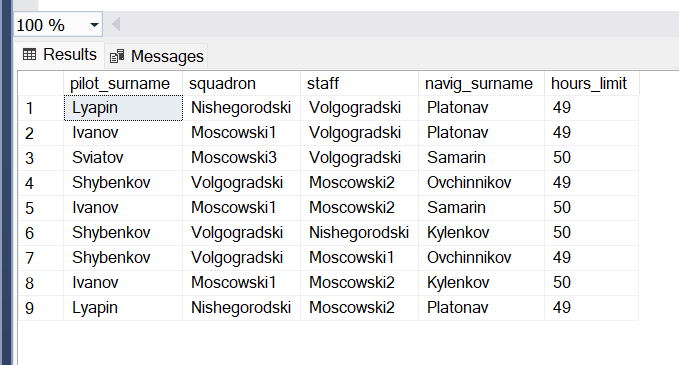
FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id

LEFT JOIN trip AS t ON f.trip = t.id

WHERE n.hours\_limit > 48 AND p.squadron != t.staff;

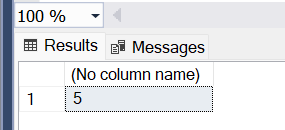


11.Реализовать запросы с использованием агрегатных функций:

* 1. определить число различных авиаотрядов, в которых работали пилоты;

SELECT COUNT(DISTINCT(squadron))

FROM pilot;



* 1. найти средний коэффициент сложности для рейсов, на которых работали пилоты из авиаотряда Московский1;

SELECT AVG(diff\_factor)

FROM (

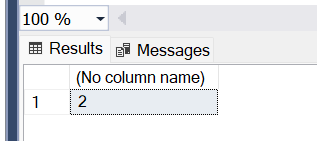
SELECT p.squadron, t.diff\_factor

FROM flights AS f

LEFT JOIN trip AS t ON f.trip = t.id

LEFT JOIN pilot AS p ON f.pilot = p.id

WHERE p.squadron LIKE 'Moscowski1') AS m1;



* 1. найти общее количество налетанных часов для пилотов, работавших на всех рейсах;

***Уникальное количество рейсов в таблице flights - 6, уникальное количество в таблице trip - 7 (на рейсе 007 в Мурманск не летал ни один пилот).***

WITH t7 AS (

SELECT \*,

DENSE\_RANK() OVER (PARTITION BY pilot ORDER BY trip) AS ranc

FROM (

SELECT pilot, trip, hours\_count

FROM flights AS f

) AS t6

)

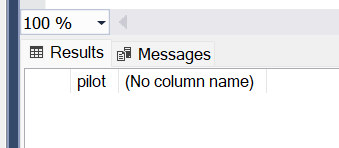
SELECT pilot,

MAX(ranc)

FROM t7

GROUP BY pilot

HAVING MAX(ranc) = 6;



***Не найдены пилоты, летавшие на всех рейсах.***

* 1. найти число вылетов в Хабаровск для пилота, со штурманом Самариным более раза.

WITH t8 AS (

SELECT p.surname AS pilot\_surname, n.surname AS navig\_surname, t.destination,

DENSE\_RANK() OVER (PARTITION BY p.surname ORDER BY t.destination) AS ranc

FROM flights AS f

LEFT JOIN trip AS t ON f.trip = t.id

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id

WHERE n.surname LIKE 'Samarin'

)

SELECT pilot\_surname, navig\_surname,

MAX(count\_pilot) AS count\_fly

FROM (

SELECT \*,

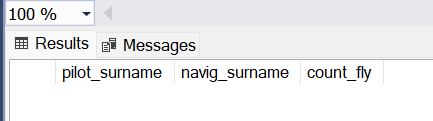
COUNT(ranc) OVER (PARTITION BY pilot\_surname) AS count\_pilot

FROM t8

WHERE destination LIKE 'Habarovsk') AS m

GROUP BY pilot\_surname, navig\_surname

HAVING MAX(count\_pilot) > 1;



***Нет пилотов, которые летали бы более раза со штурманом Самариным.***

12.Используя средства группировки реализовать следующие запросы:

* 1. для каждого пилота вывести по месяцам названия городов, куда он летал более раза;

WITH t8 AS (

SELECT p.surname, f.date\_month, t.destination,

COUNT(\*) OVER (PARTITION BY p.surname ORDER BY t.destination) AS count\_dest

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN trip AS t ON f.trip = t.id

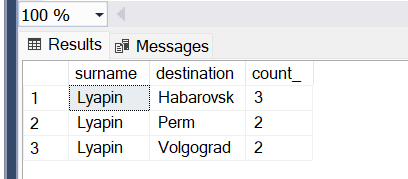
)

SELECT surname, destination, COUNT(\*) AS count\_

FROM t8

GROUP BY surname, destination

HAVING COUNT(\*) > 1;



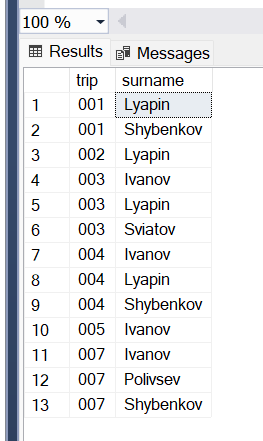
* 1. вывести для каждого рейса фамилии пилотов, на нем работавших;

SELECT f.trip, p. surname

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

GROUP BY f.trip, p.surname;



* 1. определить те месяцы, когда в Хабаровск производилось более трех вылетов;

SELECT f.date\_month, t.destination, COUNT(t.destination)

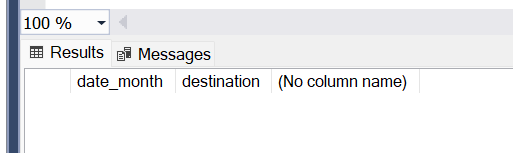
FROM flights AS f

LEFT JOIN trip AS t ON f.trip = t.id

WHERE t.destination LIKE 'Habarovsk'

GROUP BY f.date\_month, t.destination

HAVING COUNT(t.destination) > 3;



***Таких месяцев нет.***

* 1. определить пилотов и месяца, в которых данный пилот налетал более 30 часов.

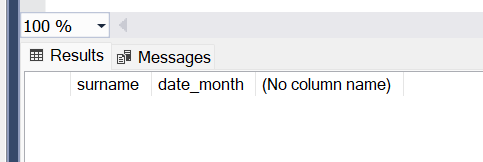
SELECT p.surname, f.date\_month, SUM(hours\_count)

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

GROUP BY p.surname, f.date\_month

HAVING SUM(hours\_count) > 30;



***Все пилоты суммарно за месяц налетали меньше 30 часов.***

13. Используя операцию HAVING реализовать запрос выводящий сумму часов полетов пары пилота и штурмана за весь год большую чем 10 часов.

WITH t5 AS (

SELECT CONCAT ('pilot ', pilot\_surname, ' - navigator ', navig\_surname) AS pair\_pn,

hours\_count

FROM (

SELECT p.surname AS pilot\_surname, n.surname AS navig\_surname, hours\_count

FROM flights AS f

LEFT JOIN pilot AS p ON f.pilot = p.id

LEFT JOIN navigator AS n ON f.navigator = n.id

) AS pairs

)

SELECT pair\_pn, SUM(hours\_count) AS total\_hours

FROM t5

GROUP BY pair\_pn

HAVING SUM(hours\_count) > 10;

