1. Вывести отсортированный по количеству перелетов (по убыванию) и имени (по возрастанию) список пассажиров, совершивших хотя бы 1 полет.

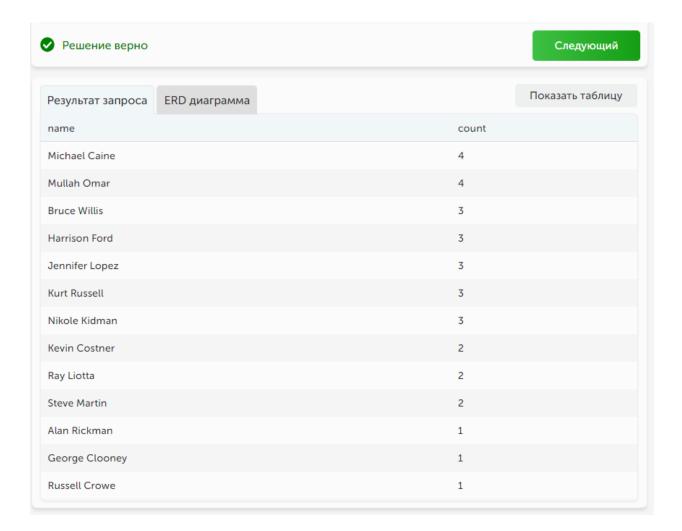https://sql-academy.org/ru/trainer/tasks/16

SELECT name, COUNT(name) AS count

FROM Pass_in_trip INNER JOIN Passenger ON Pass_in_trip.passenger = Passenger.id

GROUP BY Passenger.id

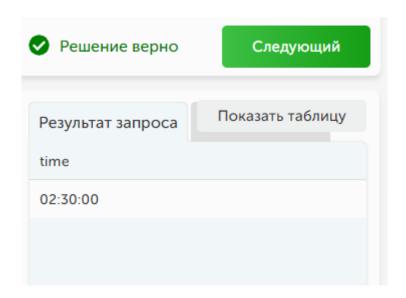HAVING COUNT(name) > 0

ORDER BY count DESC, name;

name	count
Michael Caine	4
Mullah Omar	4
Bruce Willis	3
Harrison Ford	3
Jennifer Lopez	3
Kurt Russell	3
Nikole Kidman	3
Kevin Costner	2
Ray Liotta	2
Steve Martin	2
Alan Rickman	1
George Clooney	1
Russell Crowe	1

2. Сколько времени обучающийся будет находиться в школе, учась со 2-го по 4-ый уч. предмет ?

SELECT DISTINCT TIMEDIFF((SELECT end_pair FROM Timepair WHERE id = 4),

(SELECT start_pair FROM Timepair WHERE id = 2)) as time

FROM Timepair;

3. Выведите список комнат, которые были зарезервированы в течение 12 недели 2020 года.
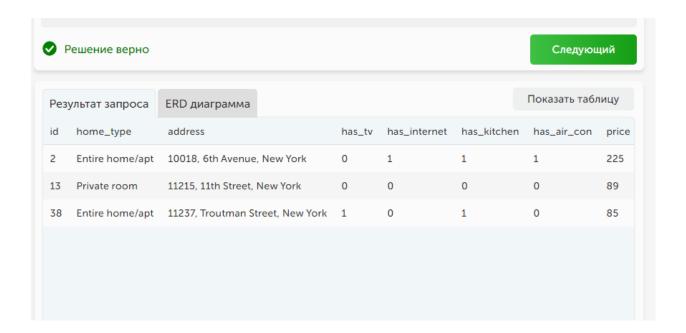
```sql
SELECT DISTINCT Rooms.*
FROM Rooms INNER JOIN Reservations ON Rooms.id = Reservations.room_id
WHERE WEEK(start_date, 1) = 12
    AND YEAR(start_date) = 2020;
```

- WEEK(start_date, 1) - т.к. полагаем, что неделя начинается с понедельника, а WEEK(date, 0) - с воскресенья

4. Какой(ие) кабинет(ы) пользуются самым большим спросом?

```
SELECT classroom
FROM Schedule
GROUP BY classroom
HAVING COUNT(classroom) = (
    SELECT COUNT(classroom)
    FROM Schedule
    GROUP BY classroom
    ORDER BY COUNT(classroom) DESC
    LIMIT 1);
```
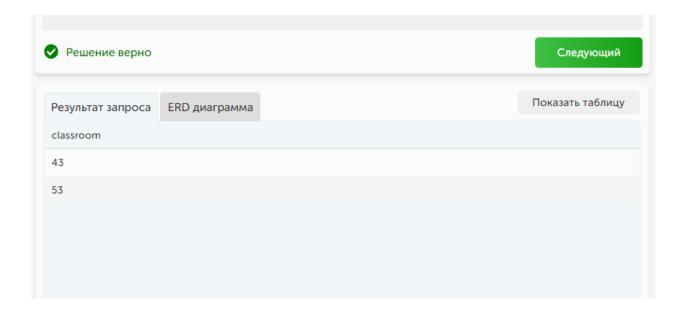
- получается 2 кабинета, т.к. значение COUNT(classroom) для них одинаковое и равно 5

5. Для каждой пары последовательных дат, dt1 и dt2, поступления средств (таблица Income_o) найти сумму выдачи денег (таблица Outcome_o) в полуоткрытом интервале (dt1, dt2].

Income_o

point	date	inc
1	2001-03-22 00:00:00.000	15000.0000
1	2001-03-23 00:00:00.000	15000.0000
1	2001-03-24 00:00:00.000	3400.0000
1	2001-04-13 00:00:00.000	5000.0000
1	2001-05-11 00:00:00.000	4500.0000
2	2001-03-22 00:00:00.000	10000.0000
2	2001-03-24 00:00:00.000	1500.0000
3	2001-09-13 00:00:00.000	11500.0000
3	2001-10-02 00:00:00.000	18000.0000

Outcome_o

point	date	out
1	2001-03-14 00:00:00.000	15348.0000
1	2001-03-24 00:00:00.000	3663.0000
1	2001-03-26 00:00:00.000	1221.0000
1	2001-03-28 00:00:00.000	2075.0000
1	2001-03-29 00:00:00.000	2004.0000
1	2001-04-11 00:00:00.000	3195.0400
1	2001-04-13 00:00:00.000	4490.0000
1	2001-04-27 00:00:00.000	3110.0000
1	2001-05-11 00:00:00.000	2530.0000
2	2001-03-22 00:00:00.000	1440.0000
2	2001-03-29 00:00:00.000	7848.0000
2	2001-04-02 00:00:00.000	2040.0000
3	2001-09-13 00:00:00.000	1500.0000
3	2001-09-14 00:00:00.000	2300.0000
3	2002-09-16 00:00:00.000	2150.0000

Результат правильного выполнения упражнения:

qty	dt1	dt2
.0000	2001-03-22 00:00:00.000	2001-03-23 00:00:00.000
3663.0000	2001-03-23 00:00:00.000	2001-03-24 00:00:00.000
22873.0400	2001-03-24 00:00:00.000	2001-04-13 00:00:00.000
5640.0000	2001-04-13 00:00:00.000	2001-05-11 00:00:00.000
1500.0000	2001-05-11 00:00:00.000	2001-09-13 00:00:00.000
2300.0000	2001-09-13 00:00:00.000	2001-10-02 00:00:00.000

Шаг 1 - выведем интервалы дат dt1, dt2 с помощью оконной + LEAD функции. Работаем с Income_o таблицей.

```
SELECT dt1, dt2

FROM (

    SELECT date AS dt1,

        LEAD (date, 1) over (ORDER BY date) AS dt2

    FROM (SELECT DISTINCT(date) FROM Income_o) AS temp

) AS temp2

WHERE dt2 IS NOT NULL
```

dt1	dt2
2001-03-22 00:00:00.000	2001-03-23 00:00:00.000
2001-03-23 00:00:00.000	2001-03-24 00:00:00.000
2001-03-24 00:00:00.000	2001-04-13 00:00:00.000
2001-04-13 00:00:00.000	2001-05-11 00:00:00.000
2001-05-11 00:00:00.000	2001-09-13 00:00:00.000
2001-09-13 00:00:00.000	2001-10-02 00:00:00.000

Шаг 2 – Верное решение, но скорее всего есть более оптимальное и короткое решение.

```
SELECT SUM(qty) AS qty, dt1, dt2

FROM

(SELECT ISNULL(qty, 0) AS qty, dt1, dt2

FROM

(SELECT dt1, dt2, qty

FROM (SELECT date AS dt1,

        LEAD(date, 1) OVER(ORDER BY date) AS dt2

        FROM (SELECT DISTINCT(date) FROM Income_o AS I_o) AS tmp1) AS tmp2

OUTER APPLY (SELECT SUM(out) AS qty

        FROM Outcome_o AS O_o

        GROUP BY date

        HAVING O_o.date > tmp2.dt1 AND O_o.date <= tmp2.dt2) AS tmp3

WHERE dt2 IS NOT NULL) AS tmp4) AS tmp5

GROUP BY dt1, dt2
```

```
SELECT SUM(qty) AS qty, dt1, dt2
FROM
(SELECT ISNULL(qty, 0) AS qty, dt1, dt2
FROM
(SELECT dt1, dt2, qty
FROM (SELECT date AS dt1,
             LEAD(date, 1) OVER(ORDER BY date) AS dt2
             FROM (SELECT DISTINCT(date) FROM Income_o AS I_o) AS tmp1) AS tmp2
OUTER APPLY (SELECT SUM(out) AS qty
             FROM Outcome_o AS O_o
             GROUP BY date
             HAVING O_o.date > tmp2.dt1 AND O_o.date <= tmp2.dt2) AS tmp3
WHERE dt2 IS NOT NULL) AS tmp4) AS tmp5
GROUP BY dt1, dt2
```
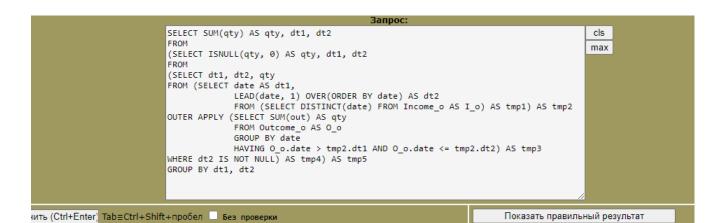
cls
max

нить (Ctrl+Enter) Tab≡Ctrl+Shift+пробел ☐ Без проверки

Показать правильный результат

Правильно.

Результат выполнения Вашего запроса:

qty	dt1	dt2
.0000	2001-03-22 00:00:00.000	2001-03-23 00:00:00.000
3663.0000	2001-03-23 00:00:00.000	2001-03-24 00:00:00.000
22873.0400	2001-03-24 00:00:00.000	2001-04-13 00:00:00.000
5640.0000	2001-04-13 00:00:00.000	2001-05-11 00:00:00.000
1500.0000	2001-05-11 00:00:00.000	2001-09-13 00:00:00.000
2300.0000	2001-09-13 00:00:00.000	2001-10-02 00:00:00.000

6. Составить отчет о битвах кораблей в два суперстолбца.

Classes							Ships		

Classes

class	type	country	numGuns	bore	displacement
Bismarck	bb	Germany	8	15.0	42000
Iowa	bb	USA	9	16.0	46000
Kongo	bc	Japan	8	14.0	32000
North Carolina	bb	USA	12	16.0	37000
Renown	bc	Gt.Britain	6	15.0	32000
Revenge	bb	Gt.Britain	8	15.0	29000
Tennessee	bb	USA	12	14.0	32000
Yamato	bb	Japan	9	18.0	65000

Ships

name	class	launched
California	Tennessee	1921
Haruna	Kongo	1916
Hiei	Kongo	1914
Iowa	Iowa	1943
Kirishima	Kongo	1915
Kongo	Kongo	1913
Missouri	Iowa	1944
Musashi	Yamato	1942
New Jersey	Iowa	1943
North Carolina	North Carolina	1941
Ramillies	Revenge	1917

Battles

name	date
#Cuba62a	1962-10-20 00:00:00.000
#Cuba62b	1962-10-25 00:00:00.000
Guadalcanal	1942-11-15 00:00:00.000
North Atlantic	1941-05-25 00:00:00.000
North Cape	1943-12-26 00:00:00.000
Surigao Strait	1944-10-25 00:00:00.000

Outcomes

ship	battle	result
Bismarck	North Atlantic	sunk
California	Guadalcanal	damaged
CAlifornia	Surigao Strait	ok
Duke of York	North Cape	ok
Fuso	Surigao Strait	sunk
Hood	North Atlantic	sunk
King George V	North Atlantic	ok
Kirishima	Guadalcanal	sunk
Prince of Wales	North Atlantic	damaged
Rodney	North Atlantic	OK

Работаем только с таблицей Battles

Шаг 1 – По сути это уже и есть решение, но нужно другое отображение данных.

SELECT ROW_NUMBER() OVER(ORDER BY date, name) AS rn, name, date

FROM Battles

ORDER BY date

rn	name	date
1	North Atlantic	1941-05-25 00:00:00.000
2	Guadalcanal	1942-11-15 00:00:00.000
3	North Cape	1943-12-26 00:00:00.000
4	Surigao Strait	1944-10-25 00:00:00.000
5	#Cuba62a	1962-10-20 00:00:00.000
6	#Cuba62b	1962-10-25 00:00:00.000

Результаты выполнения

правильного запроса:

rn_1	name_1	date_1	rn_2	name_2	date_2
1	North Atlantic	1941-05-25 00:00:00.000	4	Surigao Strait	1944-10-25 00:00:00.000
2	Guadalcanal	1942-11-15 00:00:00.000	5	#Cuba62a	1962-10-20 00:00:00.000
3	North Cape	1943-12-26 00:00:00.000	6	#Cuba62b	1962-10-25 00:00:00.000

Шаг 2 – магия (такие прихоти отображения данных не стоят трудностей этой обработки 'суперкод для суперстолбцов')

SELECT rn_1, name_1, date_1, rn_2, name_2, date_2

FROM (SELECT (CASE when quartile = 1 then tmp.rn end) AS rn_1,

(CASE when quartile = 1 then tmp.name end) AS name_1,

(CASE when quartile = 1 then tmp.date end) AS date_1,

LEAD((CASE when quartile = 2 then tmp.rn end), (SELECT COUNT(quartile)

FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp1

WHERE quartile = 1)) over (ORDER BY (CASE when quartile = 2 then tmp.rn end)) AS rn_2,

LEAD((CASE when quartile = 2 then tmp.name end), (SELECT COUNT(quartile)

FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp1

WHERE quartile = 1)) over (ORDER BY (CASE when quartile = 2 then tmp.rn end)) AS name_2,

LEAD((CASE when quartile = 2 then tmp.date end), (SELECT COUNT(quartile)

FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp1

WHERE quartile = 1)) over (ORDER BY (CASE when quartile = 2 then tmp.rn end)) AS date_2

FROM

(SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp) AS temp2

WHERE rn_1 IS NOT NULL

Правильно.

Результат выполнения Вашего запроса:

rn_1	name_1	date_1	rn_2	name_2	date_2
1	North Atlantic	1941-05-25 00:00:00.000	4	Surigao Strait	1944-10-25 00:00:00.000
2	Guadalcanal	1942-11-15 00:00:00.000	5	#Cuba62a	1962-10-20 00:00:00.000
3	North Cape	1943-12-26 00:00:00.000	6	#Cuba62b	1962-10-25 00:00:00.000

Пояснение к решению:

Левая часть и правая части:

SELECT tmp.rn AS rn_1, tmp.name AS name_1, tmp.date AS date_1 FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2) OVER(ORDER BY date) AS quartile FROM Battles) AS tmp WHERE quartile = 1	SELECT tmp.rn AS rn_2, tmp.name AS name_2, tmp.date AS date_2 FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2) OVER(ORDER BY date) AS quartile FROM Battles) AS tmp WHERE quartile = 2

rn_1	name_1	date_1
1	North Atlantic	1941-05-25 00:00:00.000
2	Guadalcanal	1942-11-15 00:00:00.000
3	North Cape	1943-12-26 00:00:00.000

rn_2	name_2	date_2
4	Surigao Strait	1944-10-25 00:00:00.000
5	#Cuba62a	1962-10-20 00:00:00.000
6	#Cuba62b	1962-10-25 00:00:00.000

Первая попытка соединения:

SELECT (CASE when quartile = 1 then tmp.rn end) AS rn_1,

(CASE when quartile = 1 then tmp.name end) AS name_1,

(CASE when quartile = 1 then tmp.date end) AS date_1,

(CASE when quartile = 2 then tmp.rn end) AS rn_2,

(CASE when quartile = 2 then tmp.name end) AS name_2,

(CASE when quartile = 2 then tmp.date end) AS date_2

FROM

(SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp

rn_1	name_1	date_1	rn_2	name_2	date_2
NULL	NULL	NULL	4	Surigao Strait	1944-10-25 00:00:00.000
NULL	NULL	NULL	5	#Cuba62a	1962-10-20 00:00:00.000
NULL	NULL	NULL	6	#Cuba62b	1962-10-25 00:00:00.000
1	North Atlantic	1941-05-25 00:00:00.000	NULL	NULL	NULL
2	Guadalcanal	1942-11-15 00:00:00.000	NULL	NULL	NULL
3	North Cape	1943-12-26 00:00:00.000	NULL	NULL	NULL

NULL просто так не убрать, добавим еще магии:

SELECT rn_1, name_1, date_1, rn_2, name_2, date_2

FROM (SELECT (CASE when quartile = 1 then tmp.rn end) AS rn_1,

(CASE when quartile = 1 then tmp.name end) AS name_1,

(CASE when quartile = 1 then tmp.date end) AS date_1,

LEAD((CASE when quartile = 2 then tmp.rn end), 3) over (ORDER BY (CASE when quartile = 2 then tmp.rn end)) AS rn_2,

LEAD((CASE when quartile = 2 then tmp.name end), 3) over (ORDER BY (CASE when quartile = 2 then tmp.date end)) AS name_2,

LEAD((CASE when quartile = 2 then tmp.date end), 3) over (ORDER BY (CASE when quartile = 2 then tmp.date end)) AS date_2

FROM

(SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp) AS temp2

WHERE rn_1 IS NOT NULL

Запрос:

```
SELECT rn_1, name_1, date_1, rn_2, name_2, date_2
FROM (SELECT (CASE when quartile  = 1 then tmp.rn end) AS rn_1,
(CASE when quartile  = 1 then tmp.name end) AS name_1,
(CASE when quartile  = 1 then tmp.date end) AS date_1,
LEAD((CASE when quartile  = 2 then tmp.rn end), 3) over (ORDER BY (CASE when
quartile  = 2 then tmp.rn end)) AS rn_2,
LEAD((CASE when quartile  = 2 then tmp.name end), 3) over (ORDER BY (CASE when
quartile  = 2 then tmp.date end)) AS name_2,
LEAD((CASE when quartile  = 2 then tmp.date end), 3) over (ORDER BY (CASE when
quartile  = 2 then tmp.date end)) AS date_2
FROM
(SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)
OVER(ORDER BY date) AS quartile
FROM Battles) AS tmp) AS temp2
WHERE rn_1 IS NOT NULL
```
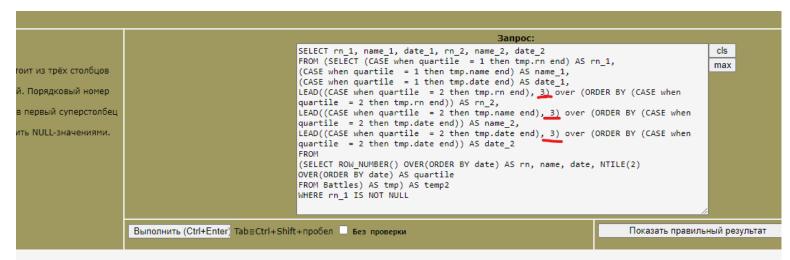
cls
max

оит из трёх столбцов
й. Порядковый номер
в первый суперстолбец
ить NULL-значениями.

Выполнить (Ctrl+Enter) Tab≡Ctrl+Shift+пробел ☐ Без проверки

Показать правильный результат

Неверно.

Ваш запрос вернул правильные данные на основной базе, но не прошёл тест на проверочной базе.
* Несовпадение данных (12)

Результат выполнения Вашего запроса:

rn_1	name_1	date_1	rn_2	name_2	date_2
1	North Atlantic	1941-05-25 00:00:00.000	4	Surigao Strait	1944-10-25 00:00:00.000
2	Guadalcanal	1942-11-15 00:00:00.000	5	#Cuba62a	1962-10-20 00:00:00.000
3	North Cape	1943-12-26 00:00:00.000	6	#Cuba62b	1962-10-25 00:00:00.000

Запрос ломается на проверочной базе из-за «3», отмеченной на картинке выше. Т.к. 3 = половине количества элементов для конкретной таблицы Battles. Поэтому ее стоит заменить на общее выражение:

SELECT COUNT(quartile)

FROM (SELECT ROW_NUMBER() OVER(ORDER BY date) AS rn, name, date, NTILE(2)

OVER(ORDER BY date) AS quartile

FROM Battles) AS tmp1

WHERE quartile = 1

И тогда получим итоговый запрос, написанный в шаге 2.