Задание 1

select Группа.Поток from Группа

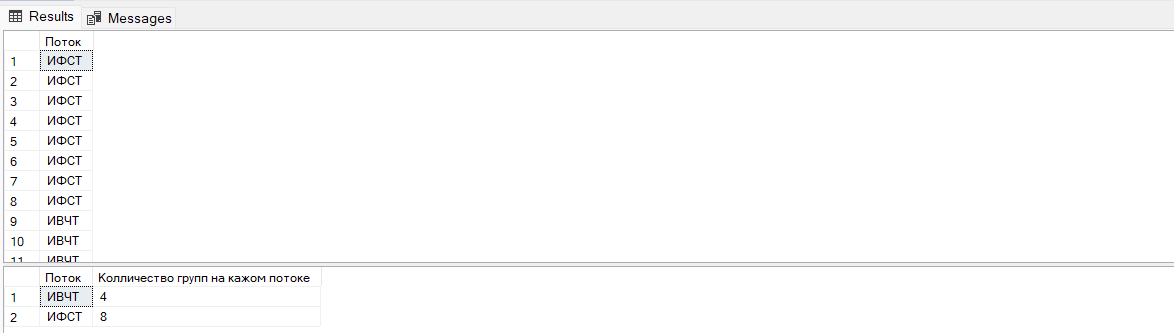
go

select Группа.Поток, COUNT(\*) as 'Колличество групп на кажом потоке'

from Группа

group by Поток

go



Задание 2

select Группа.Поток, Группа.Курс from Группа

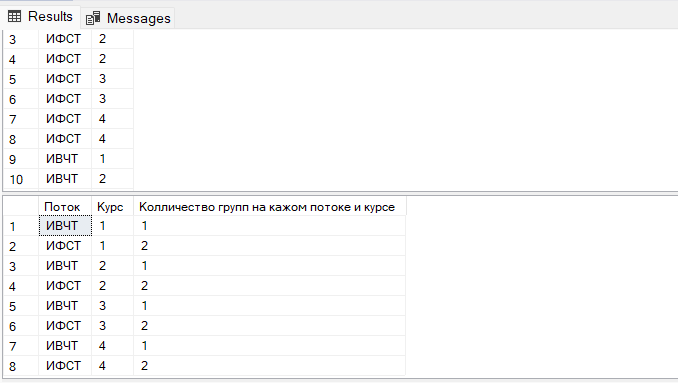
go

select Группа.Поток, Группа.Курс, COUNT(\*) as 'Колличество групп на кажом потоке и курсе'

from Группа

group by Поток, Курс

go



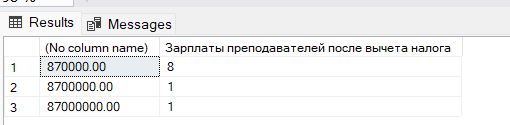
Задание 3

select Преподаватель.Зарплата \* 0.87, COUNT(\*) AS 'Зарплаты преподавателей после вычета налога'

from Преподаватель

group by Преподаватель.Зарплата \* 0.87

go



Задание 4

/\* Задание 4\* Проблема кортежей заключается в повторяемости группируемых

значений при их разном смысле. Например, одинаковые имена пользователей\*/

select Преподаватель.Фамилия, Преподаватель.Имя, COUNT(\*) as 'Куратор у групп'

from Преподаватель

join Группа on Группа.ID\_куратора = Преподаватель.ID

group by Преподаватель.Фамилия, Преподаватель.Имя

go

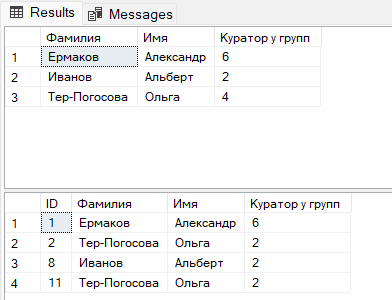
select Преподаватель.ID, Преподаватель.Фамилия, Преподаватель.Имя, COUNT(\*) as 'Куратор у групп'

from Преподаватель

join Группа on Группа.ID\_куратора = Преподаватель.ID

group by Преподаватель.ID, Преподаватель.Фамилия, Преподаватель.Имя

go



Задание 5

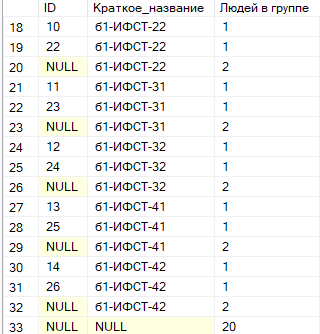
select Студент.ID, Группа.Краткое\_название, COUNT(\*) as 'Людей в группе'

from Студент

join Группа on Студент.Id\_группы = Группа.ID

group by ROLLUP(Группа.Краткое\_название,Студент.ID)

go



Задание 6

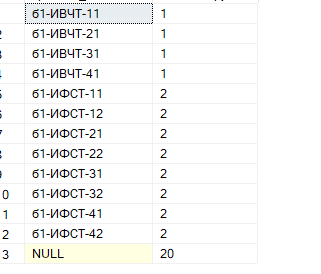
select Группа.Краткое\_название, COUNT(\*) as 'Людей в группе'

from Студент

join Группа on Студент.Id\_группы = Группа.ID

group by CUBE(Группа.Краткое\_название)

go



Задание 7

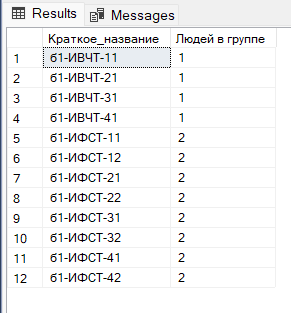
select Группа.Краткое\_название, COUNT(\*) as 'Людей в группе'

from Студент

join Группа on Студент.Id\_группы = Группа.ID

group by GROUPING SETS(Группа.Краткое\_название)

go



Задание 8

Ключевое отличие ROLLUP и CUBE состоит в иерархии результата запроса. В ROLLUP присутствует структура с промежуточными итогами, CUBE же генерирует все возможные пересечения

Задание 9

select Группа.Краткое\_название, ISNULL(CAST(Студент.ID as nvarchar(50)),

case when GROUPING(Студент.ID)=1 and GROUPING(Группа.Краткое\_название)=0

then N'Промежуточный итог' else N'Общий итог' end),

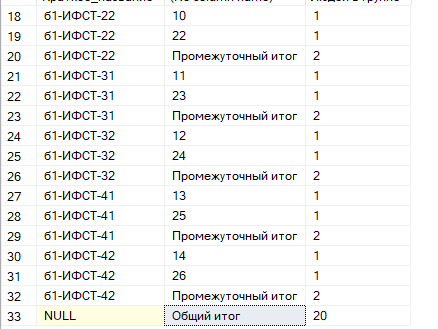
COUNT(\*) as N'Людей в группе'

from Студент

join Группа on Студент.Id\_группы = Группа.ID

group by ROLLUP(Группа.Краткое\_название,Студент.ID)

go



Задание 10

select Id\_студента, Результат,

ROW\_NUMBER() OVER (ORDER BY Результат) as row\_number,

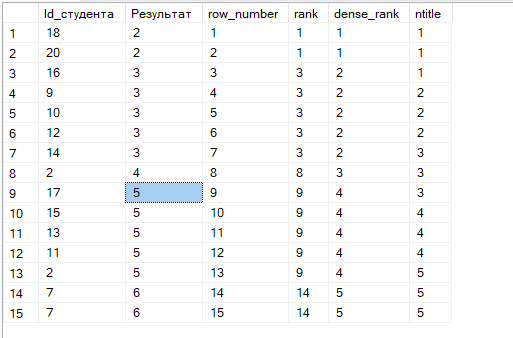
RANK() OVER (ORDER BY Результат) as rank,

DENSE\_RANK() OVER (ORDER BY Результат) as dense\_rank,

NTILE(5) OVER (ORDER BY Результат) as ntitle

from [Результат экзамена] where Id\_предмета = 1

go



Задание 11

select DISTINCT

[Результат экзамена].Id\_студента,

COUNT([Результат экзамена].Id\_студента) OVER (PARTITION BY [Результат экзамена].Id\_студента) as 'Количетсво Оценок',

SUM([Результат экзамена].результат) OVER (PARTITION BY [Результат экзамена].Id\_студента) as 'Сумма',

AVG([Результат экзамена].результат) OVER (PARTITION BY [Результат экзамена].Id\_студента) as 'Среднее',

MAX([Результат экзамена].результат) OVER (PARTITION BY [Результат экзамена].Id\_студента) as 'Макс',

MIN([Результат экзамена].результат) OVER (PARTITION BY [Результат экзамена].Id\_студента) as 'Мин'

from [Результат экзамена]



Задание 12

WITH rec (Id, Id\_начальника, Роль, "level") AS (

select

N1.[Id],

N1.Id\_начальника,

N1.Роль,

0

from Преподаватель N1

where N1.Id\_начальника is null

UNION ALL

select

N2.[Id],

N2.Id\_начальника,

N2.Роль,

N."level" + 1

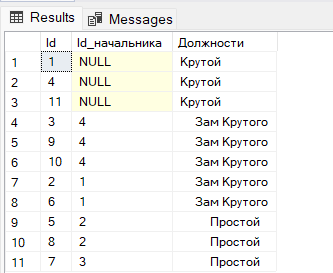
from Преподаватель N2

inner join rec N on N.Id = N2.Id\_начальника

)

select N.Id, N.Id\_начальника, (SPACE(N."level" \* 5) + N.РОЛЬ) AS Должности

from rec N Order by Должности DESC



Задание 13

MERGE INTO [Студенты и Преподаватели] N1

USING (select \* from Студент) N2

on ((N1.Фамилия = N2.Фамилия) and (N1.Отчество = N2.Отчество) and (N1.Фамилия = N2.Фамилия))

when not matched then

insert (Фамилия, Имя, Отчество, Статус)

values (N2.Фамилия, N2.Имя, N2.Отчество, N'студент');

MERGE INTO [Студенты и Преподаватели] N1

USING (select \* from Преподаватель) N2

on ((N1.Фамилия = N2.Фамилия) and (N1.Отчество = N2.Отчество) and (N1.Фамилия = N2.Фамилия))

when not matched then

insert (Фамилия, Имя, Отчество, Статус)

values (N2.Фамилия, N2.Имя, N2.Отчество, N'преподаватель');

select \* from [Студенты и преподаватели]