Лабораторная работа №5

Использование оператора SELECT

USE BSTU

go

/\*1\*/ SELECT \* FROM Faculty;

/\*2\*/ SELECT Teacher, Pulpit FROM Teacher;

/\*3\*/ SELECT Teacher\_name FROM Teacher WHERE Pulpit = 'ИСиТ';

/\*4\*/ SELECT Teacher\_name FROM Teacher WHERE Pulpit = 'ИСиТ' OR Pulpit = 'ПОиСОИ';

/\*5\*/ SELECT Teacher\_name FROM Teacher WHERE Pulpit = 'ИСиТ' AND Gender = 'ж';

/\*6\*/ SELECT Teacher\_name FROM Teacher WHERE Pulpit = 'ИСиТ' AND Gender <> 'ж';

/\*7\*/ SELECT DISTINCT Pulpit FROM Teacher;

/\*8\*/ SELECT Auditorium\_name, Auditorium\_capacity FROM Auditorium ORDER BY Auditorium\_capacity DESC;

/\*9\*/ SELECT DISTINCT TOP(2) Auditorium\_name, Auditorium\_capacity FROM Auditorium ORDER BY Auditorium\_capacity DESC;

/\*10\*/ SELECT Subjects, Note FROM Progress WHERE Note <= 10 AND Note >= 8;

SELECT Subjects, Note FROM Progress WHERE Note BETWEEN 8 AND 10;

/\*11\*/ SELECT Subjects FROM Subjects WHERE Pulpit IN ('ИСиТ', 'ПОиСОИ', 'ТЛ');

/\*12\*/ SELECT Profession\_name, Qualification FROM Profession WHERE Qualification LIKE ('%химик%');

/\*13\*/ CREATE TABLE #Students

(

ID\_student int PRIMARY KEY IDENTITY(1, 1), /\*код студента\*/

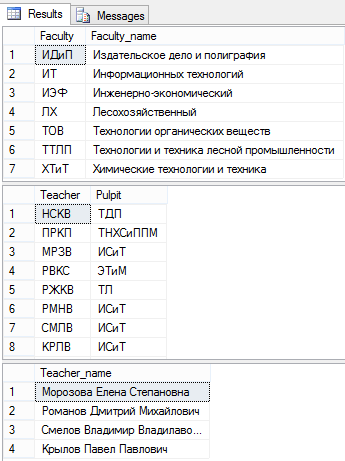
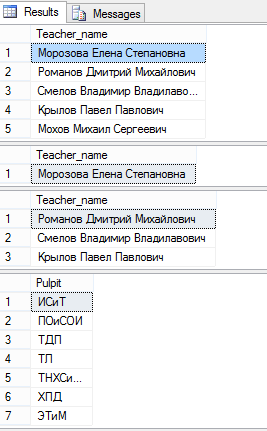
Name nvarchar(100), /\*фамилия, имя, отчество студента\*/

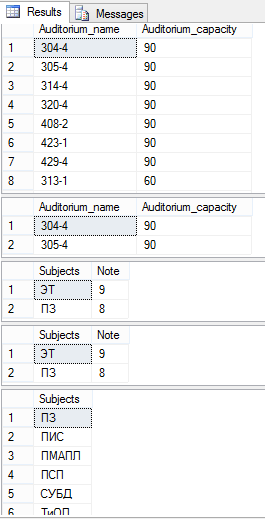
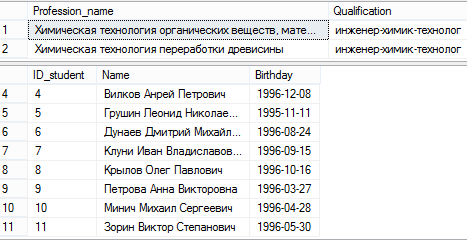
Birthday date, /\*дата рождения\*/

);

INSERT INTO #Students (Name, Birthday) SELECT Name, Birthday FROM Students;

SELECT \* FROM #Students;



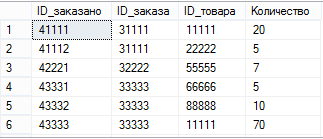


Разработать и выполнить аналогичные запросы для базы данных X\_MyBASE.

use MyBase;

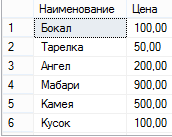
--1--

SELECT \*FROM ЗАКАЗАНО;



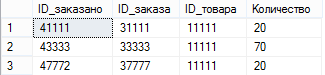
--2--

SELECT Наименование, Цена from ТОВАРЫ;



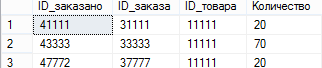
--3--

SELECT \* from ЗАКАЗАНО Where ID\_товара='11111';



--4--

SELECT \* from ЗАКАЗАНО Where ID\_товара='11111' or ID\_товара='12222';



--5--

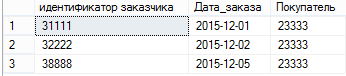
SELECT \* from ЗАКАЗЫ Where Покупатель='23333' and Дата\_заказа='2015-12-03';



--6--

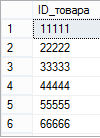
SELECT id\_заказа[идентификатор заказчика],Дата\_заказа,Покупатель

from ЗАКАЗЫ Where Покупатель='23333' and Дата\_заказа<>'2015-12-03';



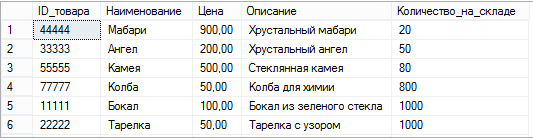
--7--

SELECT Distinct ID\_товара FROM ЗАКАЗАНО;



--8--

SELECT \* FROM ТОВАРЫ order by Количество\_на\_складе;



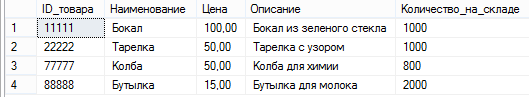
--9--

SELECT distinct Top(2) \* FROM ТОВАРЫ order by Количество\_на\_складе desc;



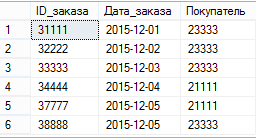
--10--

SELECT distinct \* from ТОВАРЫ Where Количество\_на\_складе Between '100'And '2000';



--11--

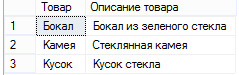
SELECT distinct \* from ЗАКАЗЫ Where Покупатель In ('23333','21111');



--12--

SELECT distinct Наименование[Товар], Описание[Описание товара]

from ТОВАРЫ where Описание like '%стекл%';



--13--

CREATE table #ТОВ

( ИМЯ nvarchar(50), ЦЕНА money );

INSERT INTO #ТОВ (ИМЯ, ЦЕНА)

SELECT Наименование, Цена FROM ТОВАРЫ

SELECT \* FROM #ТОВ

drop table #ТОВ;



Лабораторная работа №6

Многотабличные SELECT-запросы

use BSTU;

/\*1\*/SELECT Aud.Auditorium, Aud.Auditorium\_type, Aud.Auditorium\_capacity, Aud.Auditorium\_name, Aud\_T.Auditorium\_type, Aud\_T.Auditorium\_type\_name

FROM Auditorium\_type Aud\_T INNER JOIN Auditorium Aud

ON Aud\_T.Auditorium\_type = Aud.Auditorium\_type;

/\*2\*/SELECT Aud.Auditorium, Aud.Auditorium\_type, Aud.Auditorium\_capacity, Aud.Auditorium\_name, Aud\_T.Auditorium\_type, Aud\_T.Auditorium\_type\_name

FROM Auditorium\_type Aud\_T INNER JOIN Auditorium Aud

ON Aud\_T.Auditorium\_type = Aud.Auditorium\_type AND Aud\_T.Auditorium\_type\_name LIKE '%компьютер%';

/\*3\*/SELECT Auditorium.Auditorium, Auditorium.Auditorium\_type, Auditorium.Auditorium\_capacity, Auditorium.Auditorium\_name, Auditorium\_type.Auditorium\_type, Auditorium\_type.Auditorium\_type\_name

FROM Auditorium, Auditorium\_type

WHERE Auditorium.Auditorium\_type = Auditorium\_type.Auditorium\_type;

SELECT Aud.Auditorium, Aud.Auditorium\_type, Aud.Auditorium\_capacity, Aud.Auditorium\_name, Aud\_T.Auditorium\_type, Aud\_T.Auditorium\_type\_name

FROM Auditorium AS Aud, Auditorium\_type AS Aud\_T

WHERE Aud.Auditorium\_type = Aud\_T.Auditorium\_type;

/\*4\*/SELECT DISTINCT F.Faculty [Факультет], P.Pulpit [Кафедра], G.Profession [Специальность], S.Subjects\_name [Дисциплина], Stud.Name [Имя студента],

CASE

WHEN Pr.Note = 6 THEN 'шесть'

WHEN Pr.Note = 7 THEN 'семь'

WHEN Pr.Note = 8 THEN 'восемь'

END [Оценка]

FROM Pulpit P INNER JOIN Faculty F ON P.Faculty = F.Faculty,

Groups G INNER JOIN Profession Prof ON G.Profession = Prof.Profession,

Subjects S,

Students Stud INNER JOIN Progress Pr ON Stud.ID\_student = Pr.ID\_student

WHERE Pr.Note BETWEEN 6 AND 8

/\*5\*/SELECT F.Faculty [Факультет], P.Pulpit [Кафедра], G.Profession [Специальность], S.Subjects\_name [Дисциплина], Stud.Name [Имя студента],

CASE

WHEN Pr.Note = 6 THEN 'шесть'

WHEN Pr.Note = 7 THEN 'семь'

WHEN Pr.Note = 8 THEN 'восемь'

END [Оценка]

FROM Pulpit P INNER JOIN Faculty F ON P.Faculty = F.Faculty,

Groups G INNER JOIN Profession Prof ON G.Profession = Prof.Profession,

Subjects S,

Students Stud INNER JOIN Progress Pr ON Stud.ID\_student = Pr.ID\_student

WHERE Pr.Note BETWEEN 6 AND 8

ORDER BY (CASE

WHEN Pr.Note = 8 THEN 1

WHEN Pr.Note = 7 THEN 2

WHEN Pr.Note = 6 THEN 3

END)

/\*6\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Pulpit P LEFT OUTER JOIN Teacher T ON T.Pulpit = P.Pulpit;

/\*7\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Teacher T LEFT OUTER JOIN Pulpit P ON T.Pulpit = P.Pulpit;

/\*8\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Teacher T RIGHT OUTER JOIN Pulpit P ON T.Pulpit = P.Pulpit;

SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Pulpit P RIGHT OUTER JOIN Teacher T ON T.Pulpit = P.Pulpit;

/\*9.1\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Teacher T FULL OUTER JOIN Pulpit P ON T.Pulpit = P.Pulpit;

/\*9.2\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Teacher T LEFT OUTER JOIN Pulpit P ON T.Pulpit = P.Pulpit;

/\*9.3\*/SELECT ISNULL(T.Teacher\_name,'\*\*\*') [Преподаватель], P.Pulpit\_name [Кафедра]

FROM Teacher T RIGHT OUTER JOIN Pulpit P ON T.Pulpit = P.Pulpit;

USE КравчукПродажи;

/\*10.1\*/SELECT ISNULL(T.Наименование, '\*\*\*'), T.Цена, ISNULL(Z.Заказчик, '\*\*\*')

FROM Товары T LEFT OUTER JOIN Заказы Z ON T.Цена = Z.[Цена продажи]

ORDER BY T.Наименование;

/\*10.2\*/SELECT ISNULL(T.Наименование, '\*\*\*'), T.Цена, ISNULL(Z.Заказчик, '\*\*\*')

FROM Товары T RIGHT OUTER JOIN Заказы Z ON T.Цена = Z.[Цена продажи]

ORDER BY Z.Заказчик;

/\*10.3\*/SELECT ISNULL(T.Наименование, '\*\*\*'), T.Цена, ISNULL(Z.Заказчик, '\*\*\*')

FROM Товары T FULL OUTER JOIN Заказы Z ON T.Цена = Z.[Цена продажи]

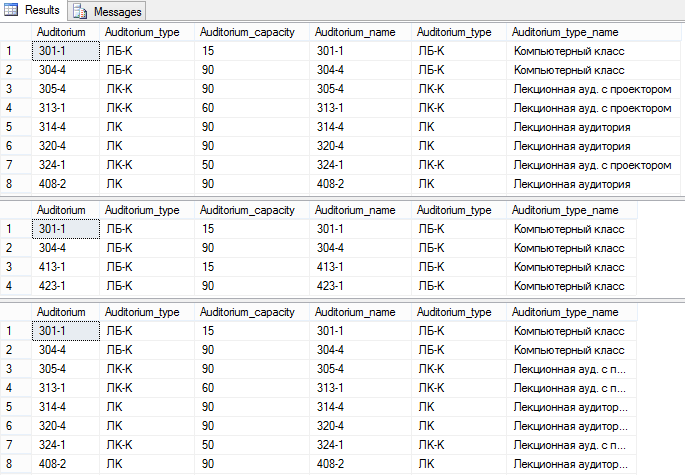
ORDER BY T.Цена;

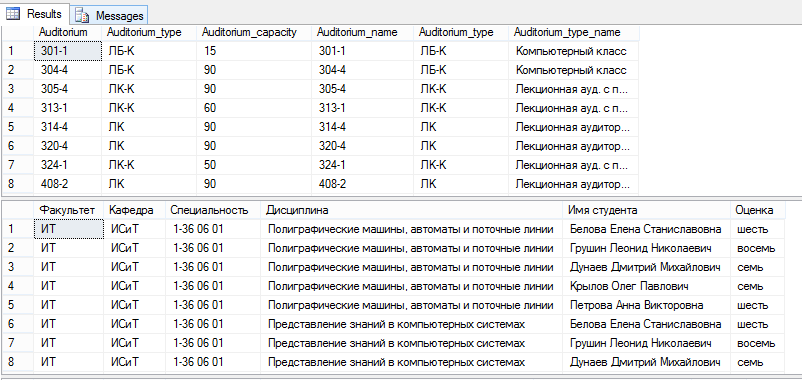
USE BSTU;

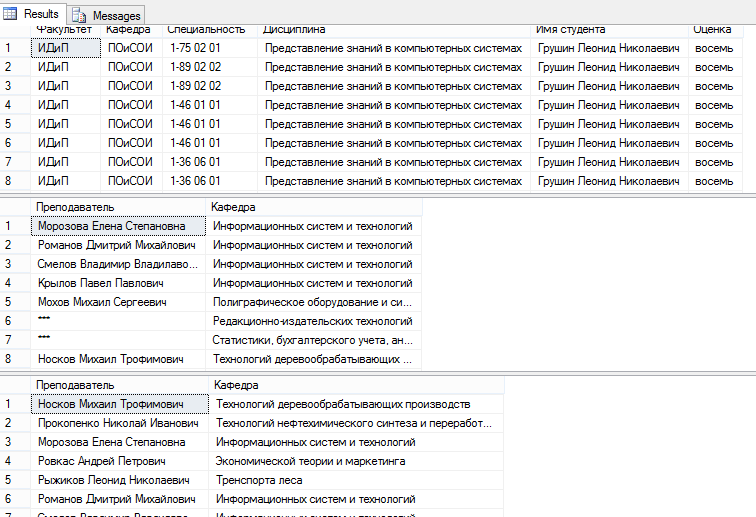
/\*11\*/SELECT Aud.Auditorium, Aud.Auditorium\_type, Aud.Auditorium\_capacity, Aud.Auditorium\_name, Aud\_T.Auditorium\_type, Aud\_T.Auditorium\_type\_name

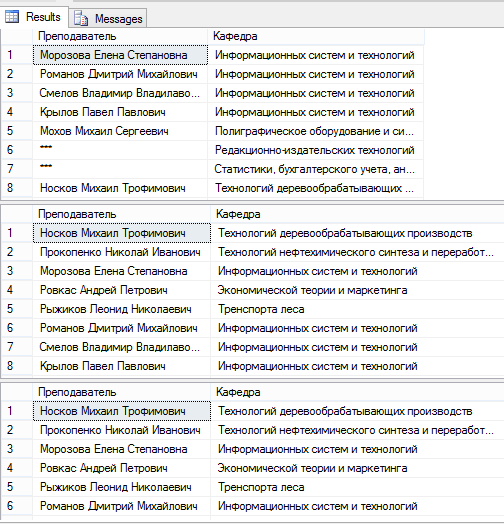
FROM Auditorium\_type Aud\_T CROSS JOIN Auditorium Aud

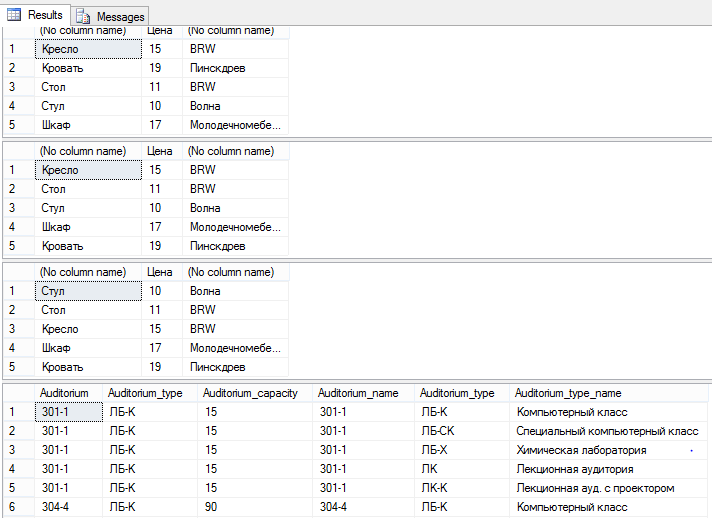
ORDER BY Aud.Auditorium;











12. Разработать и выполнить аналогичные запросы для базы данных **X\_MyBASE**.

USE MyBase

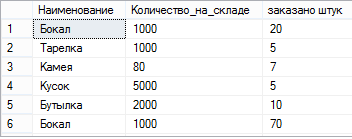
use lab4

--1--

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.Количество\_на\_складе, ЗАКАЗАНО.Количество[заказано штук]

From ТОВАРЫ Inner Join ЗАКАЗАНО

on ТОВАРЫ.ID\_товара=ЗАКАЗАНО.ID\_товара;



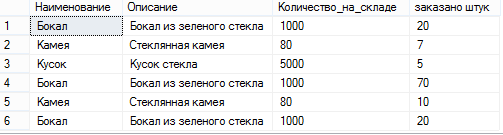
--2--

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.Описание, ТОВАРЫ.Количество\_на\_складе, ЗАКАЗАНО.Количество[заказано штук]

From ТОВАРЫ Inner Join ЗАКАЗАНО

on ТОВАРЫ.ID\_товара=ЗАКАЗАНО.ID\_товара

and ТОВАРЫ.Описание LIKE '%стекл%';



--3-- то же самое, но без inner join

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.Количество\_на\_складе, ЗАКАЗАНО.Количество[заказано штук]

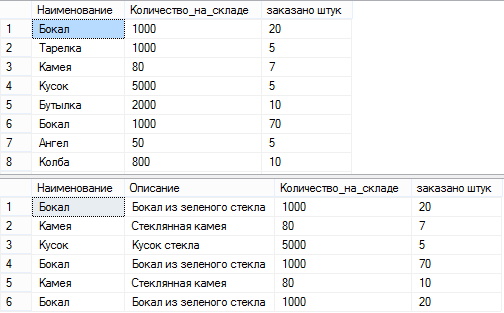
From ТОВАРЫ, ЗАКАЗАНО

where ТОВАРЫ.ID\_товара=ЗАКАЗАНО.ID\_товара;

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.Описание, ТОВАРЫ.Количество\_на\_складе, ЗАКАЗАНО.Количество[заказано штук]

From ТОВАРЫ, ЗАКАЗАНО

where ТОВАРЫ.ID\_товара=ЗАКАЗАНО.ID\_товара and ТОВАРЫ.Описание LIKE '%стекл%';



--4-- выводит все заказы каждого покупателя (с информацией о покупателе)

SELECT ЗАКАЗЧИКИ.Заказчик as [имя заказчика],

ЗАКАЗЧИКИ.Адрес as [адрес заказчика],

Case

when ( ЗАКАЗЧИКИ.ID\_заказчика='21111') then 'единицы'

when ( ЗАКАЗЧИКИ.ID\_заказчика='22222') then 'двойки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='23333') then 'тройки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='24444') then 'четверки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='25555') then 'пятерки'

end [айди заказчика],

ЗАКАЗЫ.Дата\_заказа as [дата заказа],

ЗАКАЗЫ.ID\_заказа

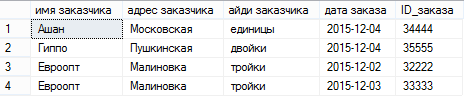
FROM ЗАКАЗЧИКИ INNER JOIN ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель and

ЗАКАЗЫ.Дата\_заказа between '2015-12-02' and '2015-12-04'

order by

ЗАКАЗЧИКИ.ID\_заказчика asc



--5--

SELECT ЗАКАЗЧИКИ.Заказчик as [имя заказчика],

ЗАКАЗЧИКИ.Адрес as [адрес заказчика],

Case

when ( ЗАКАЗЧИКИ.ID\_заказчика='21111') then 'единицы'

when ( ЗАКАЗЧИКИ.ID\_заказчика='22222') then 'двойки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='23333') then 'тройки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='24444') then 'четверки'

when ( ЗАКАЗЧИКИ.ID\_заказчика='25555') then 'пятерки'

end [айди заказчика],

ЗАКАЗЫ.Дата\_заказа as [дата заказа],

ЗАКАЗЫ.ID\_заказа

FROM ЗАКАЗЧИКИ INNER JOIN ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель and

ЗАКАЗЫ.Дата\_заказа between '2015-12-02' and '2015-12-04'

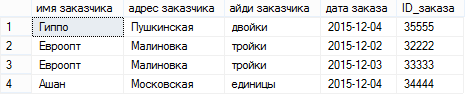
order by ( Case when ( ЗАКАЗЧИКИ.ID\_заказчика='21111') then 4

when ( ЗАКАЗЧИКИ.ID\_заказчика='22222') then 1

when ( ЗАКАЗЧИКИ.ID\_заказчика='23333') then 3

when ( ЗАКАЗЧИКИ.ID\_заказчика='24444') then 5

when ( ЗАКАЗЧИКИ.ID\_заказчика='25555') then 2 end )



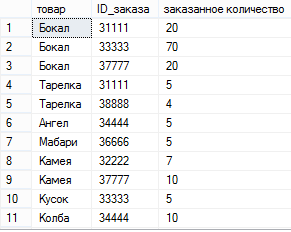
--6-- сколько единиц каждого товара куплено в разных заказах

SELECT ТОВАРЫ.Наименование[товар], ЗАКАЗАНО.ID\_заказа,

isnull(ЗАКАЗАНО.Количество, NULL)[заказанное количество]

FROM ТОВАРЫ Left Outer JOIN ЗАКАЗАНО

ON ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара



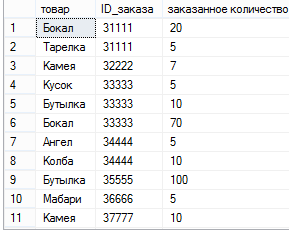
--7-- поменяли местами таблицы в left outer join

SELECT ТОВАРЫ.Наименование[товар], ЗАКАЗАНО.ID\_заказа,

isnull(ЗАКАЗАНО.Количество, NULL)[заказанное количество]

FROM ЗАКАЗАНО Left Outer JOIN ТОВАРЫ

ON ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара



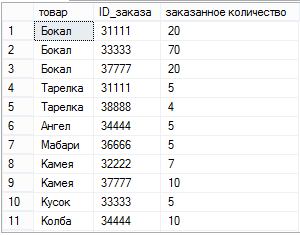
--8-- как и в п. 7, но с right outer join (опять поменяли местами таблицы)

SELECT ТОВАРЫ.Наименование[товар], ЗАКАЗАНО.ID\_заказа,

isnull(ЗАКАЗАНО.Количество, NULL)[заказанное количество]

FROM ЗАКАЗАНО RIGHT Outer JOIN ТОВАРЫ

ON ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара



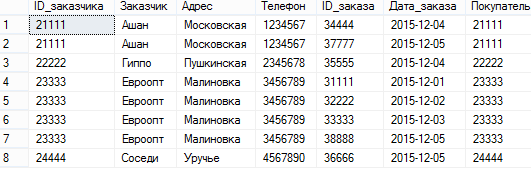
--9-- full outer join

--все заказы каждого заказчика

SELECT \* from ЗАКАЗЧИКИ FULL OUTER JOIN ЗАКАЗЫ

on ЗАКАЗЫ.Покупатель = ЗАКАЗЧИКИ.ID\_заказчика

Order by ЗАКАЗЧИКИ.Заказчик

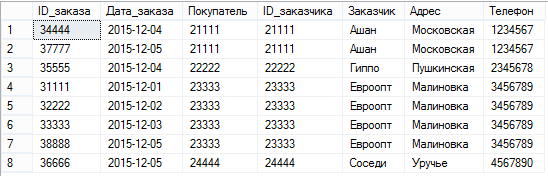


--коммутативность

SELECT \* from ЗАКАЗЫ FULL OUTER JOIN ЗАКАЗЧИКИ

on ЗАКАЗЫ.Покупатель = ЗАКАЗЧИКИ.ID\_заказчика

Order by ЗАКАЗЧИКИ.Заказчик



--соеднеие left и right outer join'ов

SELECT \* from ЗАКАЗЧИКИ LEFT OUTER JOIN ЗАКАЗЫ

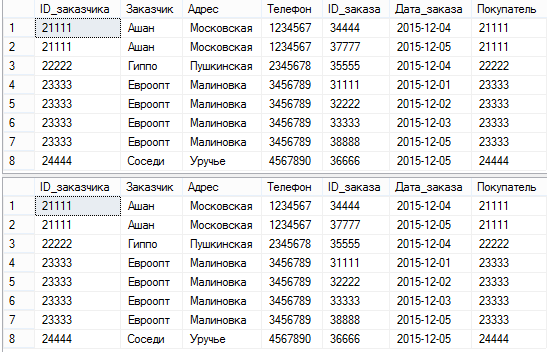
on ЗАКАЗЫ.Покупатель = ЗАКАЗЧИКИ.ID\_заказчика

Order by ЗАКАЗЧИКИ.Заказчик

SELECT \* from ЗАКАЗЧИКИ RIGHT OUTER JOIN ЗАКАЗЫ

on ЗАКАЗЫ.Покупатель = ЗАКАЗЧИКИ.ID\_заказчика

Order by ЗАКАЗЧИКИ.Заказчик

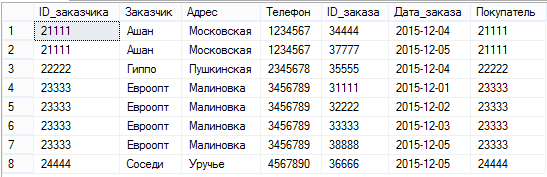


--включает inner join этих таблиц

SELECT \* from ЗАКАЗЧИКИ INNER JOIN ЗАКАЗЫ

on ЗАКАЗЫ.Покупатель = ЗАКАЗЧИКИ.ID\_заказчика

Order by ЗАКАЗЧИКИ.Заказчик



--10-- все "заказано" для каждого товара

SELECT ТОВАРЫ.ID\_товара, ТОВАРЫ.Количество\_на\_складе, ТОВАРЫ.Наименование,

ТОВАРЫ.Описание, ТОВАРЫ.Цена from ТОВАРЫ FULL OUTER JOIN ЗАКАЗАНО

on ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара

Order by ТОВАРЫ.Наименование

SELECT ЗАКАЗАНО.ID\_заказа, ЗАКАЗАНО.ID\_заказано, ЗАКАЗАНО.ID\_товара,

ЗАКАЗАНО.Количество from ТОВАРЫ FULL OUTER JOIN ЗАКАЗАНО

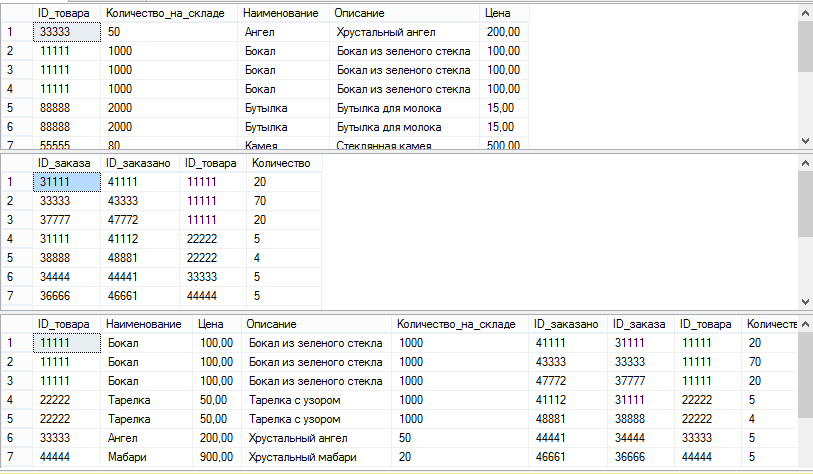
on ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара

where ТОВАРЫ.Количество\_на\_складе is not null

SELECT \* from ТОВАРЫ FULL OUTER JOIN ЗАКАЗАНО

on ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара

where ТОВАРЫ.Количество\_на\_складе is not null



--11-- cross join (каждая строка одной таблицы соединяется с каждой строкой другой таблицы)

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.Количество\_на\_складе, ЗАКАЗАНО.Количество[заказано штук]

From ТОВАРЫ Cross Join ЗАКАЗАНО

where ТОВАРЫ.ID\_товара=ЗАКАЗАНО.ID\_товара;



Лабораторная работа №7

Использование подзапросов

USE BSTU;

/\*1\*/

SELECT Pulpit.Pulpit\_name, Faculty.Faculty FROM Pulpit, Faculty

WHERE Pulpit.Faculty = Faculty.Faculty and Pulpit.Faculty IN

(SELECT Faculty FROM Profession

WHERE (Profession\_name LIKE '%технология%' OR Profession\_name LIKE '%технологии%'));

/\*2\*/

SELECT Pulpit.Pulpit\_name, Faculty.Faculty FROM Pulpit INNER JOIN Faculty

ON Pulpit.Faculty = Faculty.Faculty and Pulpit.Faculty IN

(SELECT Faculty FROM Profession

WHERE (Profession\_name LIKE '%технология%' OR Profession\_name LIKE '%технологии%'));

/\*3\*/

SELECT Pulpit.Pulpit\_name, Faculty.Faculty FROM Pulpit INNER JOIN Faculty

ON Pulpit.Faculty = Faculty.Faculty INNER JOIN Profession ON Pulpit.Faculty = Profession.Faculty

WHERE (Profession.Profession\_name LIKE '%технология%' OR Profession.Profession\_name LIKE '%технологии%');

/\*4\*/ /\*ERROR\*/

SELECT Auditorium\_name, Auditorium\_capacity FROM Auditorium A

WHERE Auditorium\_capacity = (SELECT TOP (1) Auditorium\_capacity FROM Auditorium AA

WHERE AA.Auditorium\_name = Auditorium\_name ORDER BY Auditorium\_capacity DESC);

/\*5\*/

SELECT Faculty.Faculty\_name FROM Faculty

WHERE NOT EXISTS (SELECT \* FROM Pulpit, Faculty WHERE Pulpit.Faculty = Faculty.Faculty);

/\*6\*/

SELECT TOP(1)

(SELECT avg(NOTE) FROM Progress WHERE Subjects LIKE 'ЭТ') [Экономическая теория],

(SELECT avg(NOTE) FROM Progress WHERE Subjects LIKE 'ПЗ') [Представление знаний в комьютерных системах],

(SELECT avg(NOTE) FROM Progress WHERE Subjects LIKE 'СУБД') [Системы управления базами данных]

FROM Progress;

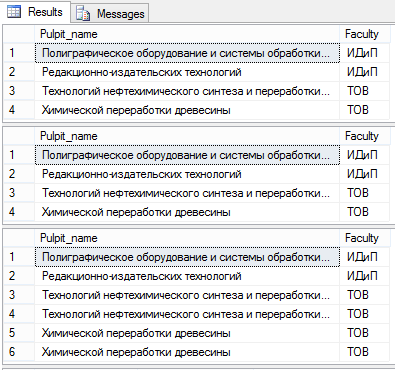
/\*7\*/

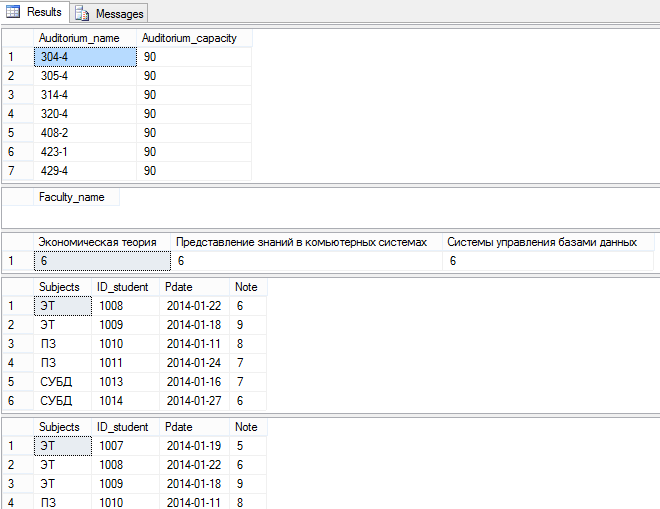
SELECT \* FROM Progress

WHERE Note > ALL (SELECT Note FROM Progress WHERE Note <= 5);

/\*8\*/

SELECT \* FROM Progress

WHERE Note > ANY (SELECT Note FROM Progress WHERE Note <= 5); 



9. Разработать и выполнить аналогичные запросы для базы данных X\_MyBASE.

USE MyBase

--1-- список товаров, если их хотя бы раз заказал Ашан (id 21111)

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.ID\_товара

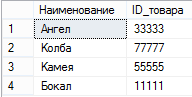
FROM ЗАКАЗЫ, ЗАКАЗАНО, ТОВАРЫ

Where ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара and

ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

and ЗАКАЗЫ.Покупатель In (Select ЗАКАЗЫ.Покупатель FROM ЗАКАЗЫ

Where (ЗАКАЗЫ.Покупатель = '21111'))



--2-- --3-- то же самое, но без подзапроса (с inner join'ами)

SELECT ТОВАРЫ.Наименование, ТОВАРЫ.ID\_товара

FROM ЗАКАЗАНО inner join ТОВАРЫ

on ТОВАРЫ.ID\_товара = ЗАКАЗАНО.ID\_товара inner join ЗАКАЗЫ

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

where ЗАКАЗЫ.Покупатель In (Select ЗАКАЗЫ.Покупатель FROM ЗАКАЗЫ

Where (ЗАКАЗЫ.Покупатель = '21111'))

--4-- самое малое количество единиц товара за один заказ (для каждого товара)

SELECT distinct ТОВАРЫ.Наименование, ТОВАРЫ.ID\_товара, ЗАКАЗАНО.Количество

FROM ЗАКАЗАНО, ТОВАРЫ

where ЗАКАЗАНО.Количество = ( select top(1) ЗАКАЗАНО.Количество from ЗАКАЗАНО

where ЗАКАЗАНО.ID\_товара = ТОВАРЫ.ID\_товара

order by ТОВАРЫ.ID\_товара desc)

--5-- список заказчиков, которые не сделали ни одного заказа

SELECT ЗАКАЗЧИКИ.Заказчик, ЗАКАЗЧИКИ.ID\_заказчика

from ЗАКАЗЧИКИ

where NOT EXISTS (SELECT \* FROM ЗАКАЗЫ

where ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель)

--6-- строка, содержащая средние значения количества заказанных единиц товара (для каждого товара)

SELECT top 1

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '11111' ) [бокал],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '12222' ) [тарелка],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '13333' ) [ангел],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '14444' ) [мабари],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '15555' ) [камея],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '16666' ) [кусок],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '17777' ) [колба],

(select avg(ЗАКАЗАНО.Количество) from ЗАКАЗАНО where ЗАКАЗАНО.ID\_товара = '18888' ) [бутылка]

from ЗАКАЗАНО



--7-- ALL - список заказов, в которых количество единиц проданного товара выше, чем в любом другом заказе

SELECT ТОВАРЫ.Наименование, ЗАКАЗАНО.Количество FROM ЗАКАЗАНО, ТОВАРЫ

where ЗАКАЗАНО.Количество>=ALL(SELECT ЗАКАЗАНО.Количество FROM ЗАКАЗАНО)

and ЗАКАЗАНО.ID\_товара = ТОВАРЫ.ID\_товара



--8-- ANY - список всех заказов, кроме тех, у которых количество единиц проданного товара ниже, чем в любом другом заказе

SELECT ТОВАРЫ.Наименование, ЗАКАЗАНО.Количество FROM ЗАКАЗАНО, ТОВАРЫ

where ЗАКАЗАНО.Количество>ANY(SELECT ЗАКАЗАНО.Количество FROM ЗАКАЗАНО)

and ЗАКАЗАНО.ID\_товара = ТОВАРЫ.ID\_товара



Лабораторная работа №8

Группировка данных

use BSTU;

/\*1\*/

SELECT

max(Auditorium\_capacity) [Максимальная вместимость],

min(Auditorium\_capacity) [Минимальная вместимость],

avg(Auditorium\_capacity) [Средняя вместимость],

sum(Auditorium\_capacity) [Суммарная вместимость],

count(\*) [Общее количество аудиторий]

FROM Auditorium;

/\*2\*/

SELECT

max(Auditorium\_capacity) [Максимальная вместимость],

min(Auditorium\_capacity) [Минимальная вместимость],

avg(Auditorium\_capacity) [Средняя вместимость],

sum(Auditorium\_capacity) [Суммарная вместимость],

count(\*) [Общее количество аудиторий]

FROM Auditorium INNER JOIN Auditorium\_type ON Auditorium\_type.Auditorium\_type = Auditorium.Auditorium\_type

GROUP BY Auditorium\_type.Auditorium\_type;

/\*3\*/

SELECT \* FROM

(

SELECT CASE WHEN Note = 10 THEN '10'

WHEN Note BETWEEN 8 AND 9 THEN '8-9'

WHEN Note BETWEEN 6 AND 7 THEN '6-7'

WHEN Note BETWEEN 4 AND 5 THEN '4-5'

END [Оценки], count (\*) [Количество]

FROM Progress GROUP BY

CASE WHEN Note = 10 THEN '10'

WHEN Note BETWEEN 8 AND 9 THEN '8-9'

WHEN Note BETWEEN 6 AND 7 THEN '6-7'

WHEN Note BETWEEN 4 AND 5 THEN '4-5'

END

) as P ORDER BY CASE [Оценки]

WHEN '10' THEN 0

WHEN '8-9' THEN 1

WHEN '6-7' THEN 2

WHEN '4-5'THEN 3

END;

/\*4\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT)),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

GROUP BY F.Faculty, G.Profession, G.Year\_First, P.Note;

/\*5\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE P.Subjects LIKE 'ПЗ' OR P.Subjects LIKE 'СУБД'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note

/\*6\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

/\*7\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП'

GROUP BY CUBE (F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note);

/\*8\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ИДиП'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note

UNION ALL

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ИДиП'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

/\*9\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП' OR F.Faculty LIKE 'ЛХ'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note

INTERSECT /\*пересечение двух исходных наборов строк\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ИДиП' OR F.Faculty LIKE 'ЛХ'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

/\*10\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ТТЛП' OR F.Faculty LIKE 'ЛХ'

GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note

EXCEPT /\*разность двух исходных наборов строк\*/

SELECT F.Faculty [Факультет], G.Profession [Специальность], P.Subjects [Предмет],

YEAR(GETDATE()) - G.Year\_First [Курс], ROUND(AVG(CAST(P.Note AS FLOAT(4))),2) [Средняя оценка]

FROM Faculty F INNER JOIN Groups G ON F.Faculty = G.Faculty

INNER JOIN Students S ON G.ID\_group = S.ID\_group

INNER JOIN Progress P ON S.ID\_student = P.ID\_student

WHERE F.Faculty LIKE 'ИДиП' OR F.Faculty LIKE 'ЛХ'

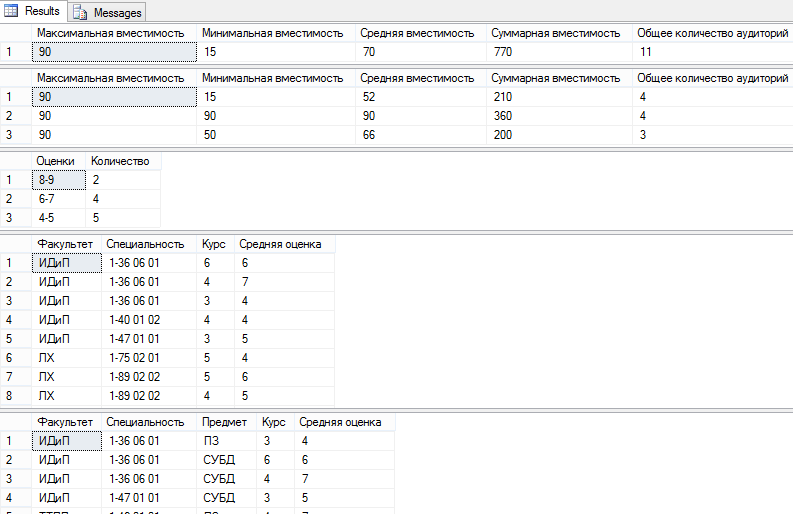
GROUP BY F.Faculty, G.Profession, P.Subjects, G.Year\_First, P.Note;

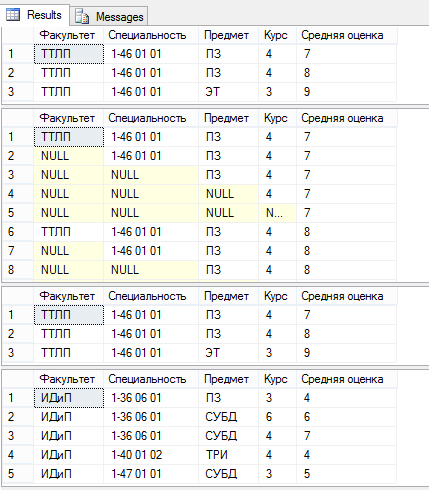
/\*11\*/

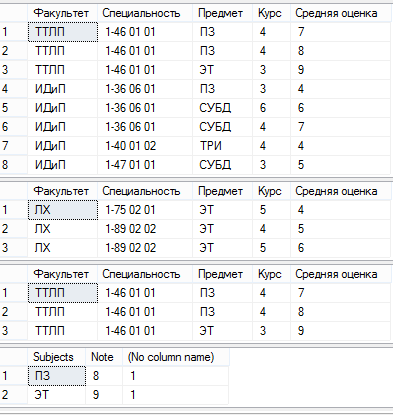
SELECT P1.Subjects, P1.Note, (SELECT count(\*) FROM Progress p2 WHERE P2.Subjects = P1.Subjects AND P2.Note = P1.Note) FROM Progress P1

GROUP BY P1.Subjects, P1.Note

HAVING P1.Note = 8 OR P1.Note = 9







12. Разработать и выполнить аналогичные запросы для базы данных X\_MyBASE.

USE MyBase;

--1-- данные о ценах на товары в целом

SELECT MAX(Цена) [MAX],

MIN(Цена) [MIN],

AVG(Цена) [AVG],

SUM(Цена) [SUM],

count(\*) [COUNT]

from ТОВАРЫ



--2-- данные об аудиториях каждого типа

SELECT ТОВАРЫ.Наименование,

max(Количество) [Макс. количество],

min(Количество) [Мин. количество],

avg(Количество) [Ср. количество],

sum(Количество) [Сум. количество],

count(\*) [Кол-во заказов этого товара]

From ЗАКАЗАНО inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара = ТОВАРЫ.ID\_товара

group by ТОВАРЫ.Наименование



--3-- количество товаров для диапазонов цен

SELECT \*

FROM (select Case

when цена between 5 and 15 then '5-15'

when Цена between 50 and 100 then '50-100'

when Цена between 200 and 500 then '200-500'

else '900'

end [Цены], COUNT (\*) [Количество товаров из этого диапазона цен]

FROM ТОВАРЫ Group by Case

when цена between 5 and 15 then '5-15'

when Цена between 50 and 100 then '50-100'

when Цена between 200 and 500 then '200-500'

else '900'

end ) as T

ORDER BY Case [Цены]

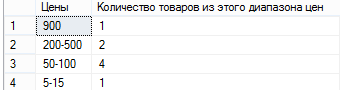
when '5-15' then 4

when '50-100' then 3

when '200-500' then 2

else 1

end



--4-- средняя цена в каждом заказе и какому заказчику этот заказ принадлежит

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

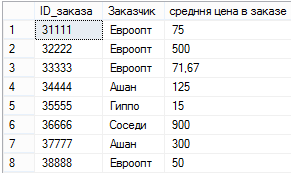
on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик

ORDER BY ЗАКАЗЫ.ID\_заказа asc



--5-- п.4 только для Евроопта и Ашана

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

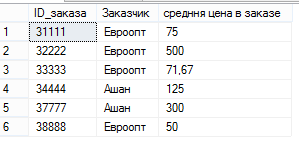
inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт' or ЗАКАЗЧИКИ.Заказчик='Ашан'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик

ORDER BY ЗАКАЗЫ.ID\_заказа asc



--6-- п.5 только для Евроопта

use lab4

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

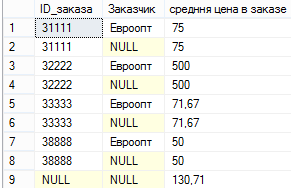
on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт'

GROUP BY rollup (ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик)



--7-- п.6 с cube-группировкой

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

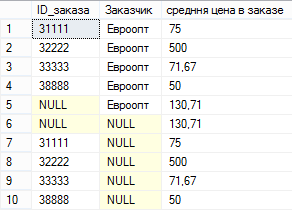
on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт'

GROUP BY cube (ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик)



--8—union(строки не могут повторяться) Евроопт и Ашан

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик

UNION

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

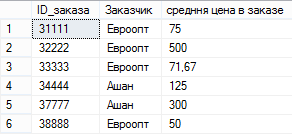
on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Ашан'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик



--9-- intersect (нет общих строк)

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик

INTERSECT

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Ашан'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик



--10-- except( (ничего не меняется, так как нет общих строк)

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Евроопт'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик

EXCEPT

SELECT ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик, round(avg(cast(ТОВАРЫ.Цена as float(4))),2) [средння цена в заказе]

From ЗАКАЗЧИКИ inner join ЗАКАЗЫ

on ЗАКАЗЧИКИ.ID\_заказчика = ЗАКАЗЫ.Покупатель

inner join ЗАКАЗАНО

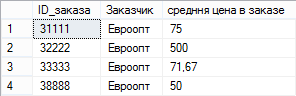
on ЗАКАЗАНО.ID\_заказа = ЗАКАЗЫ.ID\_заказа

inner join ТОВАРЫ

on ЗАКАЗАНО.ID\_товара=ТОВАРЫ.ID\_товара

where ЗАКАЗЧИКИ.Заказчик='Ашан'

GROUP BY ЗАКАЗЫ.ID\_заказа, ЗАКАЗЧИКИ.Заказчик



--11-- having - количество товаров, которые в заказе заказали 10 и 20 раз

select s1.ID\_товара, s1.Количество,

(select count(\*) from ЗАКАЗАНО s2

where s1.ID\_товара=s2.ID\_товара and s1.Количество=s2.Количество) [количество]

from ЗАКАЗАНО s1

group by s1.ID\_товара, s1.Количество

having s1.Количество=10 or s1.Количество=20

