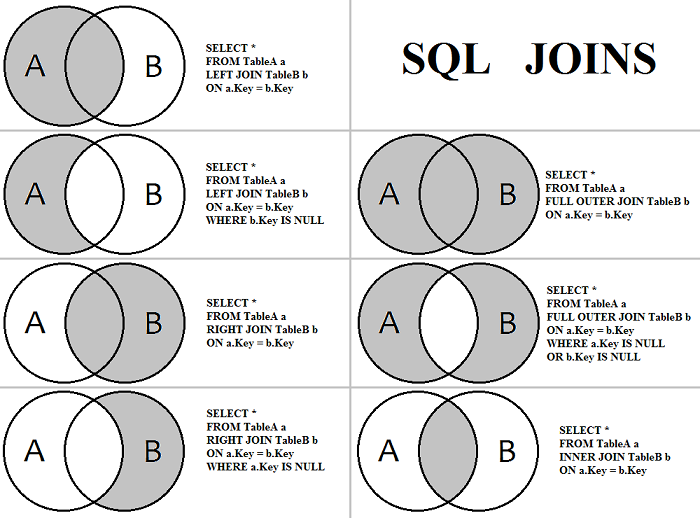
**Лабораторная работа № 4. Многотабличные SELECT-запросы**

**КУРНОСЕНКО СОФЬЯ**



**ЗАДАНИЯ**

*1.**Ознакомиться с* [*приложением*](#_Приложение_1_1)*. Изучить сценарии на языке T-SQL, содержащие операторы для создания и заполнения таблиц базы данных* ***UNIVER****.*

*Выполнить сценарии. Убедиться, что они работают корректно.*

*На основе таблиц* ***AUDITORIUM\_ TYPE*** *и* ***AUDITORIUM*** *сформировать перечень кодов аудиторий (столбец* ***AUDITORUM.AUDITORIUM****) и соответствующих им наименований типов аудиторий (столбец* ***AUDITORIUM\_ TYPE.AUDITORIUM\_ TYPENAME****). Примечание: использовать соединение таблиц INNER JOIN.*

***→ Теория (методичка)***

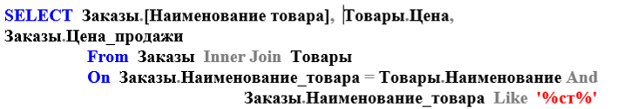
Соединение таблиц INNER JOIN (внутреннее соединение) наиболее часто используемый вид соединения реляционных таблиц.

На основании таблиц **Товары** и **Заказы** сформировать перечень товаров с ценой исходной и ценой продажи:

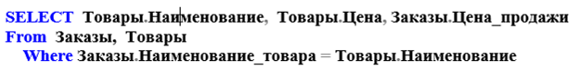


Результирующий набор создается следующим образом: выполняется декартово произведение (каждая строка одной таблицы соединяется с каждой строкой другой) для таблиц **Товары** и **Заказы**; из полученного результата выбираются строки, удовлетворяющие указанному условию; из всех столбцов результирующего набора выбираются столбцы, указанные в списке SELECT.

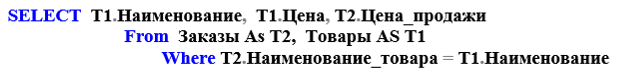
На основании таблиц **Товары** и **Заказы** сформировать перечень товаров с ценой исходной и ценой продажи, при этом выбрать товары, которые содержат буквосочетание ‘**ст**’:



Перечень товаров с ценой исходной и ценой продажи на основании таблиц **Товары** и **Заказы** безприменения INNER JOIN можно получить с помощью запроса:

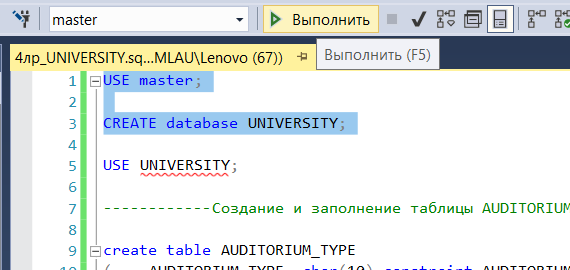


В таком запросе можно использовать псевдонимы:

Псевдонимы (здесь T1 и T2) указываются через ключевое слово AS или сразу после элемента списка (AS может быть опущено).

***→ Выполнение заданий***

Сначала выполняю часть скрипта для создания БД UNIVERSITY:



Затем запускаю сценарий создания и заполнения таблиц из приложения (исправленный):

--USE master;

--CREATE database UNIVERSITY;

USE UNIVERSITY;

------------Создание и заполнение таблицы AUDITORIUM\_TYPE

create table AUDITORIUM\_TYPE

( AUDITORIUM\_TYPE char(10) constraint AUDITORIUM\_TYPE\_PK primary key,

AUDITORIUM\_TYPENAME varchar(30)

);

insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛК', 'Лекционная');

insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-К', 'Компьютерный класс');

insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛК-К', 'Лекционная с уст. проектором');

insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-X', 'Химическая лаборатория');

insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-СК', 'Спец. компьютерный класс');

-------------Создание и заполнение таблицы AUDITORIUM

create table AUDITORIUM

( AUDITORIUM char(20) constraint AUDITORIUM\_PK primary key,

AUDITORIUM\_TYPE char(10) constraint AUDITORIUM\_AUDITORIUM\_TYPE\_FK foreign key

references AUDITORIUM\_TYPE(AUDITORIUM\_TYPE),

AUDITORIUM\_CAPACITY integer constraint AUDITORIUM\_CAPACITY\_CHECK default 1 check (AUDITORIUM\_CAPACITY between 1 and 300), -- вместимость

AUDITORIUM\_NAME varchar(50)

);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY)

values ('206-1', '206-1','ЛБ-К', 15);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY)

values ('301-1', '301-1', 'ЛБ-К', 15);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('236-1', '236-1', 'ЛК', 60);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('313-1', '313-1', 'ЛК-К', 60);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('324-1', '324-1', 'ЛК-К', 50);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('413-1', '413-1', 'ЛБ-К', 15);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('423-1', '423-1', 'ЛБ-К', 90);

insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,

AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )

values ('408-2', '408-2', 'ЛК', 90);

------Создание и заполнение таблицы FACULTY

create table FACULTY

( FACULTY char(10) constraint FACULTY\_PK primary key,

FACULTY\_NAME varchar(50) default '???'

);

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ХТиТ', 'Химическая технология и техника');

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ЛХФ', 'Лесохозяйственный факультет');

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ИЭФ', 'Инженерно-экономический факультет');

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ТТЛП', 'Технология и техника лесной промышленности');

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ТОВ', 'Технология органических веществ');

insert into FACULTY (FACULTY, FACULTY\_NAME )

values ('ИТ', 'Факультет информационных технологий');

------Создание и заполнение таблицы PROFESSION

create table PROFESSION

( PROFESSION char(20) constraint PROFESSION\_PK primary key,

FACULTY char(10) constraint PROFESSION\_FACULTY\_FK foreign key

references FACULTY(FACULTY),

PROFESSION\_NAME varchar(100), QUALIFICATION varchar(50)

);

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-40 01 02', 'Информационные системы и технологии', 'инженер-программист-системотехник' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-47 01 01', 'Издательское дело', 'редактор-технолог' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-36 06 01', 'Полиграфическое оборудование и системы обработки информации', 'инженер-электромеханик' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ХТиТ', '1-36 01 08', 'Конструирование и производство изделий из композиционных материалов', 'инженер-механик' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ХТиТ', '1-36 07 01', 'Машины и аппараты химических производств и предприятий строительных материалов', 'инженер-механик' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-75 01 01', 'Лесное хозяйство', 'инженер лесного хозяйства' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-75 02 01', 'Садово-парковое строительство', 'инженер садово-паркового строительства' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-89 02 02', 'Туризм и природопользование', 'специалист в сфере туризма' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИЭФ', '1-25 01 07', 'Экономика и управление на предприятии', 'экономист-менеджер' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИЭФ', '1-25 01 08', 'Бухгалтерский учет, анализ и аудит', 'экономист' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТТЛП', '1-36 05 01', 'Машины и оборудование лесного комплекса', 'инженер-механик' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТТЛП', '1-46 01 01', 'Лесоинженерное дело', 'инженер-технолог' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-48 01 02', 'Химическая технология органических веществ, материалов и изделий', 'инженер-химик-технолог' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-48 01 05', 'Химическая технология переработки древесины', 'инженер-химик-технолог' );

insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-54 01 03', 'Физико-химические методы и приборы контроля качества продукции', 'инженер по сертификации' );

------Создание и заполнение таблицы PULPIT

create table PULPIT

( PULPIT char(20) constraint PULPIT\_PK primary key,

PULPIT\_NAME varchar(100),

FACULTY char(10) constraint PULPIT\_FACULTY\_FK foreign key

references FACULTY(FACULTY)

);

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY )

values ('ИСиТ', 'Информационных систем и технологий ','ИТ' )

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛВ', 'Лесоводства','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛУ', 'Лесоустройства','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛЗиДВ', 'Лесозащиты и древесиноведения','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛКиП', 'Лесных культур и почвоведения','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ТиП', 'Туризма и природопользования','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛПиСПС','Ландшафтного проектирования и садово-паркового строительства','ЛХФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ТЛ', 'Транспорта леса', 'ТТЛП')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЛМиЛЗ','Лесных машин и технологии лесозаготовок','ТТЛП')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ТДП','Технологий деревообрабатывающих производств', 'ТТЛП')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ТиДИД','Технологии и дизайна изделий из древесины','ТТЛП')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ОХ', 'Органической химии','ТОВ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ХПД','Химической переработки древесины','ТОВ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ТНВиОХТ','Технологии неорганических веществ и общей химической технологии ','ХТиТ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ПиАХП','Процессов и аппаратов химических производств','ХТиТ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('ЭТиМ', 'Экономической теории и маркетинга','ИЭФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('МиЭП', 'Менеджмента и экономики природопользования','ИЭФ')

insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)

values ('СБУАиА', 'Статистики, бухгалтерского учета, анализа и аудита', 'ИЭФ')

------Создание и заполнение таблицы TEACHER

create table TEACHER

( TEACHER char(10) constraint TEACHER\_PK primary key,

TEACHER\_NAME varchar(100),

GENDER char(1) CHECK (GENDER in ('м', 'ж')),

PULPIT char(20) constraint TEACHER\_PULPIT\_FK foreign key

references PULPIT(PULPIT)

);

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('СМЛВ', 'Смелов Владимир Владиславович', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('АКНВЧ', 'Акунович Станислав Иванович', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('КЛСНВ', 'Колесников Виталий Леонидович', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('БРКВЧ', 'Бракович Андрей Игоревич', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('ДТК', 'Дятко Александр Аркадьевич', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('УРБ', 'Урбанович Павел Павлович', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('ГРН', 'Гурин Николай Иванович', 'м', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('ЖЛК', 'Жиляк Надежда Александровна', 'ж', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('МРЗ', 'Мороз Елена Станиславовна', 'ж', 'ИСиТ');

insert into TEACHER (TEACHER, TEACHER\_NAME,GENDER, PULPIT )

values ('БРНВСК', 'Барановский Станислав Иванович', 'м', 'ЭТиМ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('НВРВ', 'Неверов Александр Васильевич', 'м', 'МиЭП');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('РВКЧ', 'Ровкач Андрей Иванович', 'м', 'ЛВ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('РЖК', 'Рожков Леонид Николаевич ', 'м', 'ЛВ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('БЗБРДВ', 'Безбородов Владимир Степанович', 'м', 'ОХ');

insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )

values ('НСКВЦ', 'Насковец Михаил Трофимович', 'м', 'ТЛ');

------Создание и заполнение таблицы SUBJECT

create table SUBJECT

( SUBJECT char(10) constraint SUBJECT\_PK primary key,

SUBJECT\_NAME varchar(100) unique,

PULPIT char(20) constraint SUBJECT\_PULPIT\_FK foreign key

references PULPIT(PULPIT)

);

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('СУБД', 'Системы управления базами данных', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT)

values ('БД', 'Базы данных','ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ИНФ', 'Информационные технологии','ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ОАиП', 'Основы алгоритмизации и программирования', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ПЗ', 'Представление знаний в компьютерных системах', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ПСП', 'Программирование сетевых приложений', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('МСОИ', 'Моделирование систем обработки информации', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ПИС', 'Проектирование информационных систем', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('КГ', 'Компьютерная геометрия ','ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('КМС', 'Компьютерные мультимедийные системы', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT)

values ('ДМ', 'Дискретная математика', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('МП', 'Математическое программирование','ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ЛЭВМ', 'Логические основы ЭВМ', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('ООП', 'Объектно-ориентированное программирование', 'ИСиТ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ЭП', 'Экономика природопользования','МиЭП');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ЭТ', 'Экономическая теория','ЭТиМ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('ИГ', 'Инженерная геодезия ','ЛУ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('ЛВ', 'Лесоводство', 'ЛЗиДВ') ;

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ОХ', 'Органическая химия', 'ОХ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )

values ('ВТЛ', 'Водный транспорт леса','ТЛ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('ТиОЛ', 'Технология и оборудование лесозаготовок', 'ЛМиЛЗ');

insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )

values ('ТОПИ', 'Технология обогащения полезных ископаемых ','ТНВиОХТ');

------Создание и заполнение таблицы GROUPS

create table GROUPS

( IDGROUP integer identity(1,1) constraint GROUP\_PK primary key,

FACULTY char(10) constraint GROUPS\_FACULTY\_FK foreign key

references FACULTY(FACULTY),

PROFESSION char(20) constraint GROUPS\_PROFESSION\_FK foreign key

references PROFESSION(PROFESSION),

YEAR\_FIRST smallint check (YEAR\_FIRST<=YEAR(GETDATE())),

);

insert into GROUPS (FACULTY, PROFESSION, YEAR\_FIRST )

values ('ХТиТ','1-36 01 08', 2013), --1 группа

('ХТиТ','1-36 01 08', 2012), --2 группа...

('ХТиТ','1-36 07 01', 2011),

('ХТиТ','1-36 07 01', 2010),

('ТОВ','1-48 01 02', 2012),

('ТОВ','1-48 01 02', 2011),

('ТОВ','1-48 01 05', 2013),

('ТОВ','1-54 01 03', 2012),

('ЛХФ','1-75 01 01', 2013),

('ЛХФ','1-75 02 01', 2012),

('ЛХФ','1-75 02 01', 2011),

('ЛХФ','1-89 02 02', 2012),

('ЛХФ','1-89 02 02', 2011),

('ТТЛП','1-36 05 01', 2013),

('ТТЛП','1-36 05 01', 2012),

('ТТЛП','1-46 01 01', 2012),

('ИЭФ','1-25 01 07', 2013),

('ИЭФ','1-25 01 07', 2012),

('ИЭФ','1-25 01 07', 2010),

('ИЭФ','1-25 01 08', 2013),

('ИЭФ','1-25 01 08', 2012); --21 группа

------Создание и заполнение таблицы STUDENT

create table STUDENT

( IDSTUDENT integer identity(1000,1) constraint STUDENT\_PK primary key,

IDGROUP integer constraint STUDENT\_GROUP\_FK foreign key

references GROUPS(IDGROUP),

NAME nvarchar(100),

BDAY date,

STAMP timestamp,

INFO xml,

FOTO varbinary

);

insert into STUDENT (IDGROUP, NAME, BDAY)

values (2, 'Силюк Валерия Ивановна', '12.07.1994'),

(2, 'Сергель Виолетта Николаевна', '06.03.1994'),

(2, 'Добродей Ольга Анатольевна', '09.11.1994'),

(2, 'Подоляк Мария Сергеевна', '04.10.1994'),

(2, 'Никитенко Екатерина Дмитриевна', '08.01.1994'),

(3, 'Яцкевич Галина Иосифовна', '02.08.1993'),

(3, 'Осадчая Эла Васильевна', '07.12.1993'),

(3, 'Акулова Елена Геннадьевна', '02.12.1993'),

(4, 'Плешкун Милана Анатольевна', '08.03.1992'),

(4, 'Буянова Мария Александровна', '02.06.1992'),

(4, 'Харченко Елена Геннадьевна', '11.12.1992'),

(4, 'Крученок Евгений Александрович', '11.05.1992'),

(4, 'Бороховский Виталий Петрович', '09.11.1992'),

(4, 'Мацкевич Надежда Валерьевна', '01.11.1992'),

(5, 'Логинова Мария Вячеславовна', '08.07.1995'),

(5, 'Белько Наталья Николаевна', '02.11.1995'),

(5, 'Селило Екатерина Геннадьевна', '07.05.1995'),

(5, 'Дрозд Анастасия Андреевна', '04.08.1995'),

(6, 'Козловская Елена Евгеньевна', '08.11.1994'),

(6, 'Потапнин Кирилл Олегович', '02.03.1994'),

(6, 'Равковская Ольга Николаевна', '04.06.1994'),

(6, 'Ходоронок Александра Вадимовна', '09.11.1994'),

(6, 'Рамук Владислав Юрьевич', '04.07.1994'),

(7, 'Неруганенок Мария Владимировна', '03.01.1993'),

(7, 'Цыганок Анна Петровна', '12.09.1993'),

(7, 'Масилевич Оксана Игоревна', '12.06.1993'),

(7, 'Алексиевич Елизавета Викторовна','09.02.1993'),

(7, 'Ватолин Максим Андреевич', '04.07.1993'),

(8, 'Синица Валерия Андреевна', '08.01.1992'),

(8, 'Кудряшова Алина Николаевна', '12.05.1992'),

(8, 'Мигулина Елена Леонидовна', '08.11.1992'),

(8, 'Шпиленя Алексей Сергеевич', '12.03.1992'),

(9, 'Астафьев Игорь Александрович', '10.08.1995'),

(9, 'Гайтюкевич Андрей Игоревич', '02.05.1995'),

(9, 'Рученя Наталья Александровна', '08.01.1995'),

(9, 'Тарасевич Анастасия Ивановна', '11.09.1995'),

(10, 'Жоглин Николай Владимирович', '08.01.1994'),

(10, 'Санько Андрей Дмитриевич', '11.09.1994'),

(10, 'Пещур Анна Александровна', '06.04.1994'),

(10, 'Бучалис Никита Леонидович', '12.08.1994'),

(11, 'Лавренчук Владислав Николаевич','07.11.1993'),

(11, 'Власик Евгения Викторовна', '04.06.1993'),

(11, 'Абрамов Денис Дмитриевич', '10.12.1993'),

(11, 'Оленчик Сергей Николаевич', '04.07.1993'),

(11, 'Савинко Павел Андреевич', '08.01.1993'),

(11, 'Бакун Алексей Викторович', '02.09.1993'),

(12, 'Бань Сергей Анатольевич', '11.12.1995'),

(12, 'Сечейко Илья Александрович', '10.06.1995'),

(12, 'Кузмичева Анна Андреевна', '09.08.1995'),

(12, 'Бурко Диана Францевна', '04.07.1995'),

(12, 'Даниленко Максим Васильевич', '08.03.1995'),

(12, 'Зизюк Ольга Олеговна', '12.09.1995'),

(13, 'Шарапо Мария Владимировна', '08.10.1994'),

(13, 'Касперович Вадим Викторович', '10.02.1994'),

(13, 'Чупрыгин Арсений Александрович','11.11.1994'),

(13, 'Воеводская Ольга Леонидовна', '10.02.1994'),

(13, 'Метушевский Денис Игоревич', '12.01.1994'),

(14, 'Ловецкая Валерия Александровна','11.09.1993'),

(14, 'Дворак Антонина Николаевна', '01.12.1993'),

(14, 'Щука Татьяна Николаевна', '09.06.1993'),

(14, 'Коблинец Александра Евгеньевна','05.01.1993'),

(14, 'Фомичевская Елена Эрнестовна', '01.07.1993'),

(15, 'Бесараб Маргарита Вадимовна', '07.04.1992'),

(15, 'Бадуро Виктория Сергеевна', '10.12.1992'),

(15, 'Тарасенко Ольга Викторовна', '05.05.1992'),

(15, 'Афанасенко Ольга Владимировна', '11.01.1992'),

(15, 'Чуйкевич Ирина Дмитриевна', '04.06.1992'),

(16, 'Брель Алеся Алексеевна', '08.01.1994'),

(16, 'Кузнецова Анастасия Андреевна', '07.02.1994'),

(16, 'Томина Карина Геннадьевна', '12.06.1994'),

(16, 'Дуброва Павел Игоревич', '03.07.1994'),

(16, 'Шпаков Виктор Андреевич', '04.07.1994'),

(17, 'Шнейдер Анастасия Дмитриевна', '08.11.1993'),

(17, 'Шыгина Елена Викторовна', '02.04.1993'),

(17, 'Клюева Анна Ивановна', '03.06.1993'),

(17, 'Доморад Марина Андреевна', '05.11.1993'),

(17, 'Линчук Михаил Александрович', '03.07.1993'),

(18, 'Васильева Дарья Олеговна', '08.01.1995'),

(18, 'Щигельская Екатерина Андреевна','06.09.1995'),

(18, 'Сазонова Екатерина Дмитриевна', '08.03.1995'),

(18, 'Бакунович Алина Олеговна', '07.08.1995')

------Создание и заполнение таблицы PROGRESS

create table PROGRESS

( SUBJECT char(10) constraint PROGRESS\_SUBJECT\_FK foreign key

references SUBJECT(SUBJECT),

IDSTUDENT integer constraint PROGRESS\_IDSTUDENT\_FK foreign key

references STUDENT(IDSTUDENT),

PDATE date,

NOTE integer check (NOTE between 1 and 10)

);

insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)

values ('ОАиП', 1001, '01.10.2013',8),

('ОАиП', 1002, '01.10.2013',7),

('ОАиП', 1003, '01.10.2013',5),

('ОАиП', 1005, '01.10.2013',4);

insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)

values ('СУБД', 1014, '01.12.2013',5),

('СУБД', 1015, '01.12.2013',9),

('СУБД', 1016, '01.12.2013',5),

('СУБД', 1017, '01.12.2013',4);

insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)

values ('КГ', 1018, '06.5.2013',4),

('КГ', 1019, '06.05.2013',7),

('КГ', 1020, '06.05.2013',7),

('КГ', 1021, '06.05.2013',9),

('КГ', 1022, '06.05.2013',5),

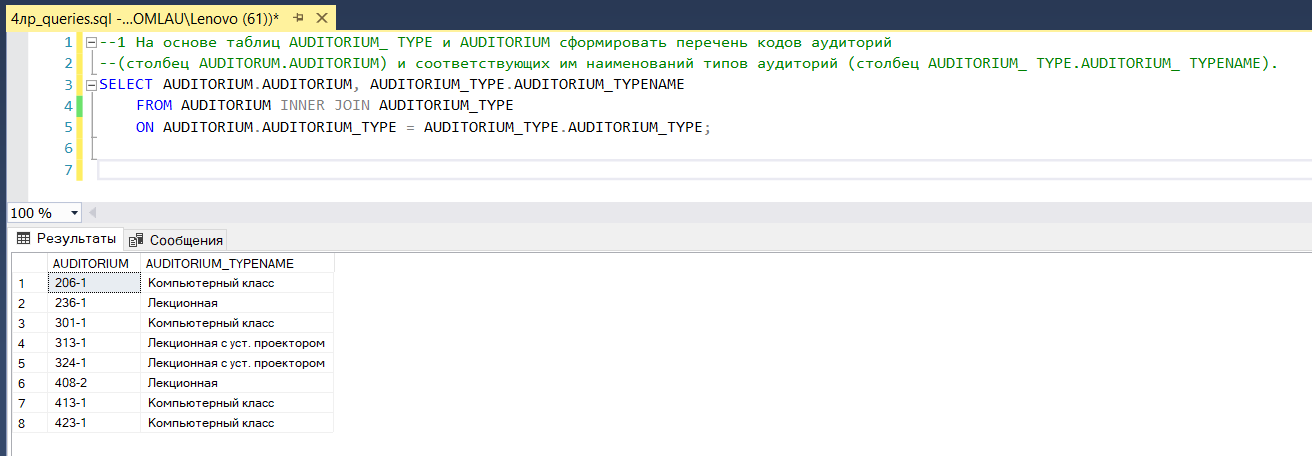
('КГ', 1023, '06.05.2013',6);

INNER JOIN запрос:

SELECT AUDITORIUM.AUDITORIUM, AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME

FROM AUDITORIUM INNER JOIN AUDITORIUM\_TYPE

ON AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE;



*2. На основе таблиц AUDITORIUM\_TYPE и AUDITORIUM сформировать перечень кодов аудиторий (столбец AUDITORIUM.AUDITORIUM) и соответствующих им наименований типов аудиторий (столбец AUDITORIUM\_ TYPE.AUDITO-RIUM\_TYPENAME).*

*При этом следует выбрать только те аудитории, в наименовании которых присутствует подстрока компьютер.*

*Примечание: использовать соединение таблиц INNER JOIN и предикат LIKE.*

***→ Выполнение заданий***

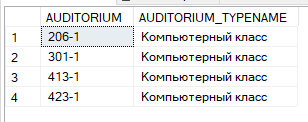
SELECT AUDITORIUM.AUDITORIUM, AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME

FROM AUDITORIUM INNER JOIN AUDITORIUM\_TYPE

ON AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE AND

AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME LIKE '%компьютер%';

Результат:



*3.**Написать два SELECT-запроса, формирующих результирующие наборы аналогичные запросам из заданий 1 и 2, но без применения INNER JOIN.*

***→ Выполнение заданий***

SELECT AUDITORIUM.AUDITORIUM, AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME

FROM AUDITORIUM, AUDITORIUM\_TYPE

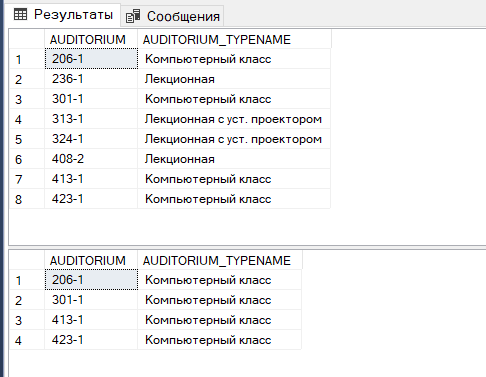
WHERE AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE;

SELECT AUDITORIUM.AUDITORIUM, AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME

FROM AUDITORIUM, AUDITORIUM\_TYPE

WHERE AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE AND

AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME LIKE '%компьютер%';



*4.**На основе таблиц* ***PRORGESS****,* ***STUDENT****,* ***GROUPS****,* ***SUBJECT****,* ***PULPIT*** *и* ***FACULTY*** *сформировать перечень студентов, получивших экзаменационные оценки (столбец* ***PROGRESS.NOTE****) от 6 до 8.*

*Результирующий набор должен содержать столбцы:* ***Факультет****,* ***Кафедра****,* ***Специальность****,* ***Дисциплина****,* ***Имя Студента****,* ***Оценка****. В столбце* ***Оценка*** *должны быть записаны экзаменационные оценки прописью:* ***шесть****,* ***семь****,* ***восемь****.*

*Результирующий набор отсортировать в порядке возрастания по столбцам* ***FACULTY.FACULTY****,* ***PULPIT.PULPIT****,* ***PROFESSION.PROFESSION****,* ***STUDENT. STUDENT\_NAME*** *и в порядке убывания по столбцу* ***PROGRESS.NOTE****.*

*Примечание: использовать соединение INNER JOIN, предикат BETWEEN и выражение CASE.*

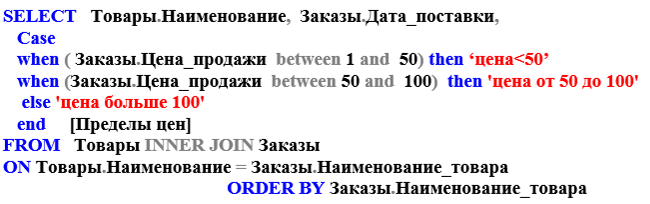
***→ Теория (методичка)***

На основании таблиц **Товары** и **Заказы** получить перечень товаров, дату поставки и сформировать столбец **Пределы цен** можно с помощью запроса, записанного ниже.

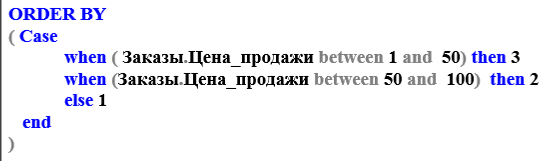
Здесь в выражении CASE каждое предложение WHEN содержит логическое выражение. Эти выражения проверяются на истинность сверху вниз, и при первом успешном сравнении формируется результирующее значение, указанное за ключевым словом THEN. В том случае, если ни одно из логических WHEN-выражений не принимает истинного значения, в качестве результата CASE формируется значение, указанное в предложении ELSE.

Результирующий набор содержит три столбца, наименования которых: **Наименование**, **Дата\_поставки**, **Пределы цен**.

В последнем столбце выводятся тексты из выражения CASE. Реализована сортировка по полю **Заказы.Наименование\_товара** в порядке возрастания.



Пусть надо в предыдущем запросе вывести результаты таким образом, чтобы сначала выводились названия товаров с ценой больше 100, затем с ценой в пределах от 50 до 100 и, наконец, с ценой, меньшей 50. Это можно сделать с помощью изменения секции ORDER BYпредыдущего запроса:



***→ Выполнение заданий***

SELECT FACULTY.FACULTY\_NAME AS Факультет,

PULPIT.PULPIT\_NAME AS Кафедра,

PROFESSION.PROFESSION\_NAME AS Специальность,

SUBJECT.SUBJECT\_NAME AS Дисциплина,

STUDENT.NAME AS [Имя студента],

CASE PROGRESS.NOTE

WHEN 6 THEN 'шесть'

WHEN 7 THEN 'семь'

WHEN 8 THEN 'восемь'

END AS Оценка

FROM PROGRESS

INNER JOIN STUDENT ON PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT

INNER JOIN GROUPS ON STUDENT.IDGROUP = GROUPS.IDGROUP

INNER JOIN SUBJECT ON SUBJECT.SUBJECT = PROGRESS.SUBJECT

INNER JOIN PULPIT ON SUBJECT.PULPIT = PULPIT.PULPIT

INNER JOIN FACULTY ON PULPIT.FACULTY = FACULTY.FACULTY

INNER JOIN PROFESSION ON FACULTY.FACULTY = PROFESSION.FACULTY

WHERE PROGRESS.NOTE BETWEEN 6 AND 8

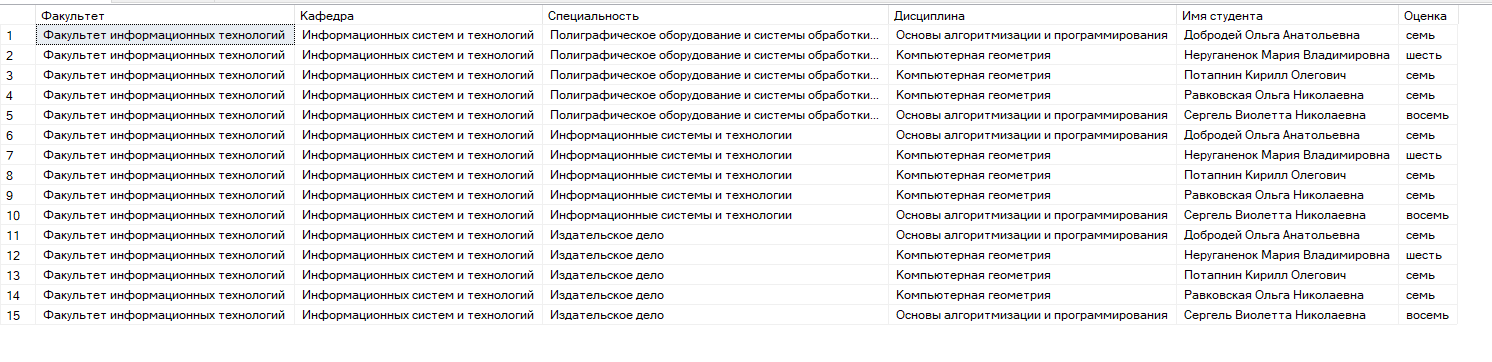
ORDER BY FACULTY.FACULTY,

PULPIT.PULPIT,

PROFESSION.PROFESSION,

STUDENT.NAME,

PROGRESS.NOTE DESC;

******

*5.**Переписать запрос, реализующий задание 4 таким образом, чтобы в результирующем наборе сортировка по экзаменационным оценкам была следующей: сначала выводились строки с оценкой* ***7****, затем строки с оценкой* ***8*** *и далее строки с оценкой* ***6****.*

*Примечание: использовать выражение CASE в секции ORDER BY.*

***→ Выполнение заданий***

SELECT FACULTY.FACULTY\_NAME AS Факультет,

PULPIT.PULPIT\_NAME AS Кафедра,

PROFESSION.PROFESSION\_NAME AS Специальность,

SUBJECT.SUBJECT\_NAME AS Дисциплина,

STUDENT.NAME AS [Имя студента],

CASE PROGRESS.NOTE

WHEN 6 THEN 'шесть'

WHEN 7 THEN 'семь'

WHEN 8 THEN 'восемь'

END AS Оценка

FROM PROGRESS

INNER JOIN STUDENT ON PROGRESS.IDSTUDENT = STUDENT.IDSTUDENT

INNER JOIN GROUPS ON STUDENT.IDGROUP = GROUPS.IDGROUP

INNER JOIN SUBJECT ON SUBJECT.SUBJECT = PROGRESS.SUBJECT

INNER JOIN PULPIT ON SUBJECT.PULPIT = PULPIT.PULPIT

INNER JOIN FACULTY ON PULPIT.FACULTY = FACULTY.FACULTY

INNER JOIN PROFESSION ON FACULTY.FACULTY = PROFESSION.FACULTY

WHERE PROGRESS.NOTE BETWEEN 6 AND 8

ORDER BY

(CASE

WHEN PROGRESS.NOTE = 7 THEN 1

WHEN PROGRESS.NOTE = 8 THEN 2

WHEN PROGRESS.NOTE = 6 THEN 3

END),

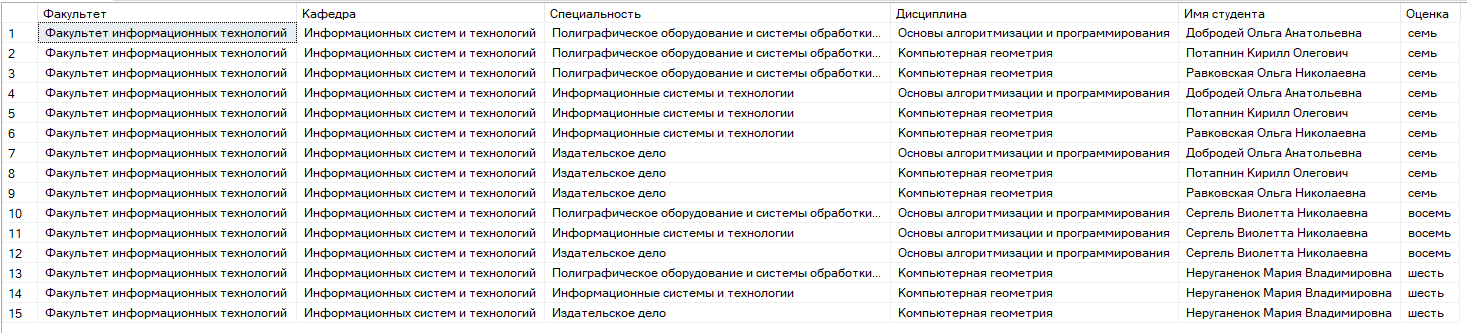
FACULTY.FACULTY,

PULPIT.PULPIT,

PROFESSION.PROFESSION,

STUDENT.NAME,

PROGRESS.NOTE DESC;

**

*6.**На основе таблиц* ***PULPIT*** *и* ***TEACHER*** *получить полный перечень кафедр (столбец* ***PULPIT.PULPIT\_ NAME)*** *и преподавателей (столбец* ***TEACHER.TEA-CHER\_NAME****) на этих кафедрах. Результирующий набор должен содержать два столбца:* ***Кафедра*** *и* ***Преподаватель****. Если на кафедре нет преподавателей, то в столбце* ***Преподаватель*** *должна быть выведена строка* ***\*\*\*.***

*Примечание: использовать соединение таблиц LEFT OUTER JOIN и функцию* ***isnull****.*

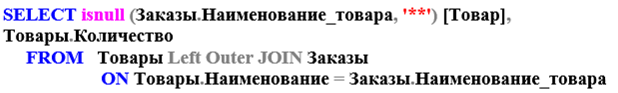
***→ Теория (методичка)***

Внешнее соединение OUTER JOIN двух таблиц формирует набор строк, состоящий из двух частей: результат внутреннего соединения двух таблиц (inner join) и строки из одной из двух таблиц, которые не смогли соединиться. Значения в столбцах, соответствующих незаполненной (несоединенной) части строки будет NULL.

Имеется два вида внешнего соединения: LEFT OUTER JOIN – левое внешнее соединение и RIGHT OUTER JOIN – правое внешнее соединение.

Левое внешнее соединение включает в набор несоединенные строки таблицы, имя которой записано слева от ключевых слов LEFT OUTER JOIN, а правое внешнее соединение – несоединенные строки таблицы, имя которой записано справа от RIGHT OUTER JOIN.

Пусть требуется получить список заказанных товаров с указанием их количества на складе. Столбец с наименованиями товара озаглавить − «Товар». Если наименование товара в таблице **Заказы** пропущено (в поле находится NULL), то в соответствующем поле столбца «Товар» вывести “\*\*”.



Встроенная функция **isnull** принимает два параметра и проверяет их значения на NULL слева направо. Функция возвращает первое значение, не равное NULL

***→ Выполнение заданий***

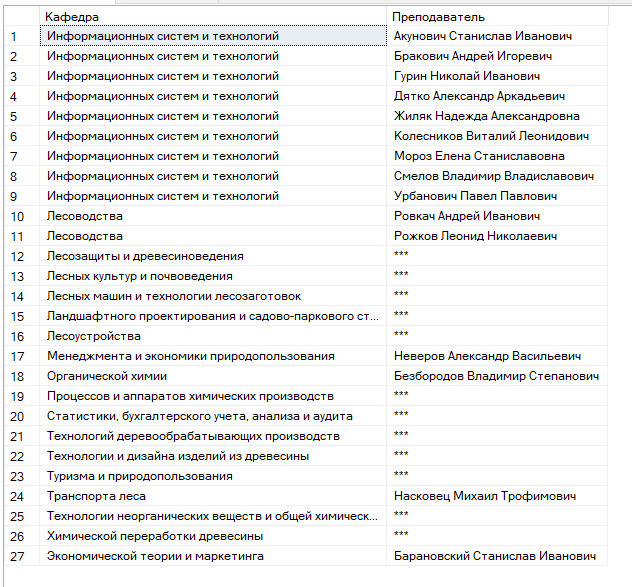
После нахождения всех комбинаций PULPIT.PULPIT\_NAME, TEACHER.TEACHER\_NAME и отбора тех, что подходят по условию PULPIT.PULPIT = TEACHER.PULPIT, мы дополняем выборку не отобранными ранее строками таблицы слева (PULPIT в данном случае), но так как больше не остается преподавателей, подходящих по условию, то в этих строках мы видим \*\*\* (замена null в нашем случае).

SELECT PULPIT.PULPIT\_NAME AS Кафедра,

isnull(TEACHER.TEACHER\_NAME, '\*\*\*') AS Преподаватель

FROM PULPIT LEFT JOIN TEACHER

ON PULPIT.PULPIT = TEACHER.PULPIT;

******

***7.*** *В запросе, реализующем пункт 6, поменять порядок таблиц в выражении LEFT OUTER JOIN. Объяснить: почему в столбце* ***Кафедра*** *не может быть значения NULL.*

*Переписать запрос таким образом, чтобы получился аналогичный результат, но применялось соединение таблиц RIGHT OUTER JOIN.*

***→ Выполнение заданий***

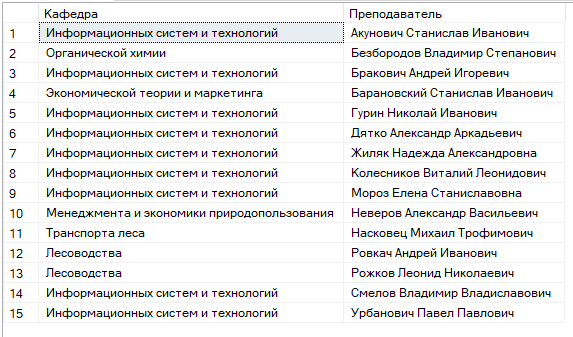
В запросе, реализующем пункт 6, поменять порядок таблиц в выражении LEFT OUTER JOIN

SELECT PULPIT.PULPIT\_NAME AS Кафедра,

isnull(TEACHER.TEACHER\_NAME, '\*\*\*') AS Преподаватель

FROM TEACHER LEFT JOIN PULPIT

ON PULPIT.PULPIT = TEACHER.PULPIT;

******

***Вопрос****: почему в столбце* ***Кафедра*** *не может быть значения NULL?*

***Ответ****: потому что, согласно логике, каждый преподаватель должен быть закреплен за определенной кафедрой (но не на каждой кафедре должен быть преподаватель).*

Переписать запрос таким образом, чтобы получился аналогичный результат, но применялось соединение таблиц RIGHT OUTER JOIN.

SELECT PULPIT.PULPIT\_NAME AS Кафедра,

isnull(TEACHER.TEACHER\_NAME, '\*\*\*') AS Преподаватель

FROM TEACHER RIGHT JOIN PULPIT

ON PULPIT.PULPIT = TEACHER.PULPIT;

*8.**Показать на примере, что соединение FULL OUTER JOIN двух таблиц:*

*− является коммутативной операцией;*

*− является объединением LEFT OUTER JOIN и RIGHT OUTER JOIN соединений этих таблиц;*

*− включает соединение INNER JOIN этих таблиц.*

*Примечание: создать две таблицы, заполнить их данными. Разработать SELECT-запросы, реализующие задания.*

*Создать три новых запроса:*

*− запрос, результат которого содержит данные левой (в операции FULL OUTER JOIN) таблицы и не содержит данные правой;*

*− запрос, результат которого содержит данные правой таблицы и не содержащие данные левой;*

*− запрос, результат которого содержит данные правой таблицы и левой таблиц;*

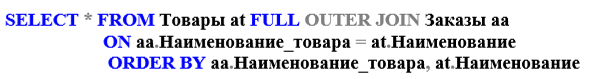
*Примечание: использовать в запросах выражение IS NULL и IS NOT NULL.*

***→ Теория (методичка)***

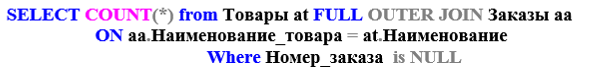
Операция является *коммутативной*, если формируемый результирующий набор не зависит от порядка, в котором указаны исходные таблицы.

FULL OUTER JOIN определяет объединение правого и левого соединения/

Рассмотрим два запроса для таблиц **Товары** и **Заказы**. В первом запросе выводятся значения, полученные в результате полного внешнего объединения таблиц:



Во втором запросе выводится количество значений, которые не смогли соединиться:



***→ Выполнение заданий***

Создаю две новые таблице чисто для примера в БД UNIVERSITY и заполняю их:

--создаем две таблицы для примеров

CREATE TABLE Товары(

ИД int identity(1,1) PRIMARY KEY,

Название nvarchar(50)

);

CREATE TABLE Заказы(

ИД int identity(16,1) PRIMARY KEY,

Товар int FOREIGN KEY

references Товары (ИД),

Дата\_заказа datetime,

);

--заполняем таблицы

INSERT INTO Товары

VALUES

('Кимоно'),

('Волшебный кинжал'),

('Электрогитара'),

('Деревянный меч'),

('DVD "Гинтама"'),

('Набор для убийства'),

('Проклятие'),

('Почка');

INSERT INTO Заказы

VALUES

(1, '20220308'),

(7, '20220308'),

(5, '20220308'),

(3, '20220308'),

(7, '20220428'),

(7, '20220308'),

(1, '20220308'),

(7, '20220308'),

(8, '20210408'),

(3, '20220308'),

(1, '20210304'),

(3, '20220308'),

(7, '20220308'),

(5, '20220211'),

(NULL, '20200308'),

(NULL, '20220211');

*Показать на примере, что соединение FULL OUTER JOIN двух таблиц:*

*− является коммутативной операцией;*

SELECT Товары.Название, Заказы.Дата\_заказа

FROM Товары FULL JOIN Заказы

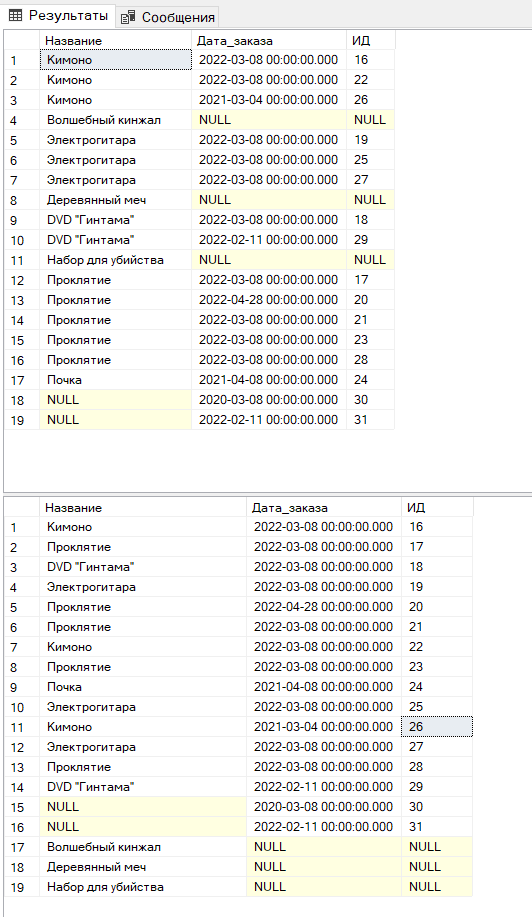
ON Товары.ИД = Заказы.Товар;

SELECT Товары.Название, Заказы.Дата\_заказа

FROM Заказы FULL JOIN Товары

ON Товары.ИД = Заказы.Товар;

Оба запроса возвращают одинаковые строки (вверху первый, внизу второй):

**

*− является объединением LEFT OUTER JOIN и RIGHT OUTER JOIN соединений этих таблиц;*

SELECT Товары.Название, Заказы.Дата\_заказа, Заказы.ИД

FROM Товары LEFT JOIN Заказы

ON Товары.ИД = Заказы.Товар

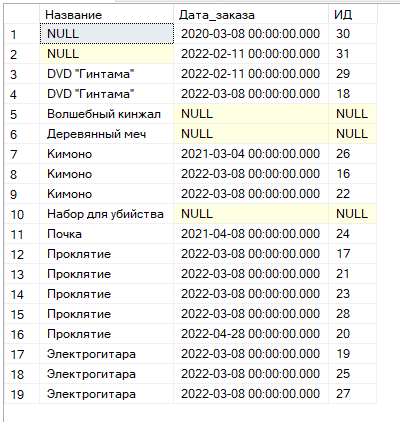
UNION

SELECT Товары.Название, Заказы.Дата\_заказа, Заказы.ИД

FROM Товары RIGHT JOIN Заказы

ON Товары.ИД = Заказы.Товар;

Пришлось включить в выборку поле ИД заказа, потому что оператор UNION оставляет в итоговой выборке только уникальные записи. А так как без ИД заказа остаются только записи вида «Название товара – Дата заказа», а один и тот же товар мог быть заказан в один и тот же день, то мы имеем на выходе идентичные записи. Этой идентичности и позволяет избежать поле ИД заказа.

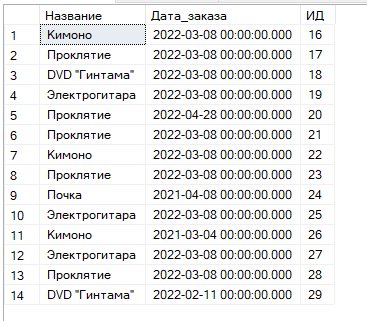


*− включает соединение INNER JOIN этих таблиц.*

SELECT Товары.Название, Заказы.Дата\_заказа, Заказы.ИД

FROM Товары INNER JOIN Заказы

ON Товары.ИД = Заказы.Товар;

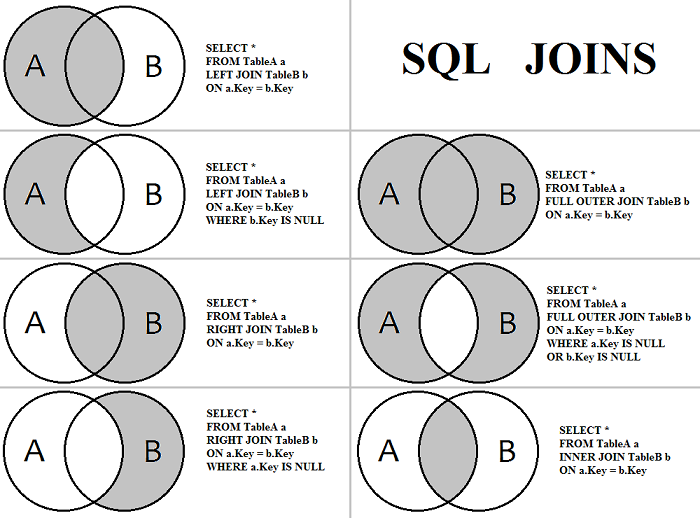
****

Чтобы убедится в правдивости утверждения, о том что FULL JOIN включает соединение INNER JOIN, нужно сравнить результаты запроса этого пункта с результатами предыдущего запроса.

*Создать три новых запроса:*

*− запрос, результат которого содержит данные левой (в операции FULL OUTER JOIN) таблицы и не содержит данные правой;*

Если нужно так:



--содержит все значения таблицы слева (LEFT JOIN короче)

SELECT Товары.ИД as ИД\_товара,

Товары.Название as Название\_товара,

Заказы.ИД as ИД\_заказа

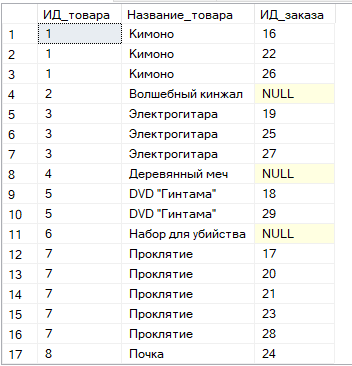
FROM Товары FULL JOIN Заказы

ON Товары.ИД = Заказы.Товар

WHERE Товары.ИД IS NOT NULL;

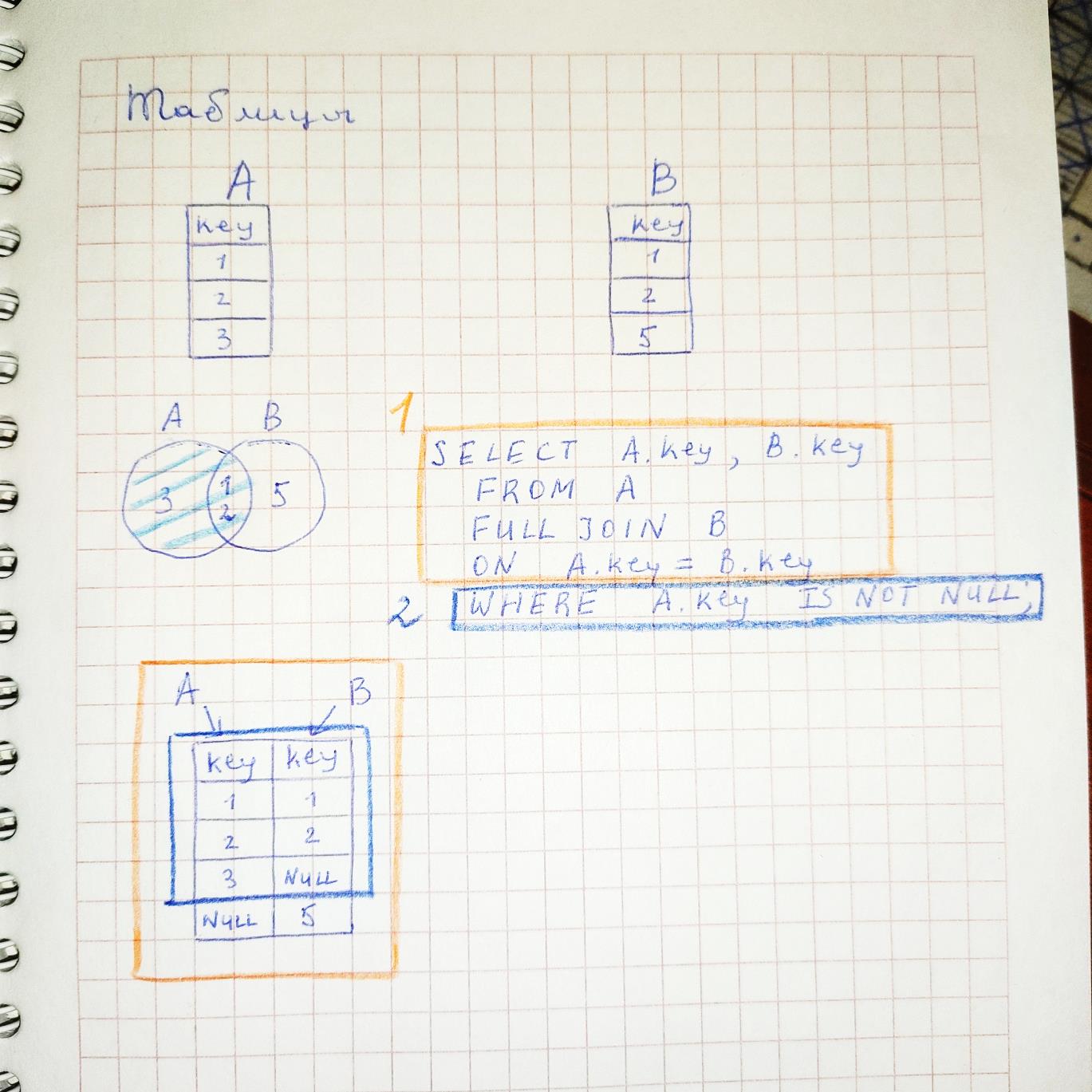
Последняя строка указывает, что если Товары.ИД = NULL, то нужно исключить такую запись из итоговой выборки, так как она вообще не относится к левой таблице (Товары). В плане, в таблице Товары столбец ИД является первичным ключом и значения NULL иметь не может. Так что запись NULL в выборке данного запроса на месте идентификатора товара говорит о том, что в столбце Заказы.Товар не было найдено соответствующей записи из таблицы Товары.

В результате мы видим все записи таблицы слева + соответствующие им значения из правой таблицы:

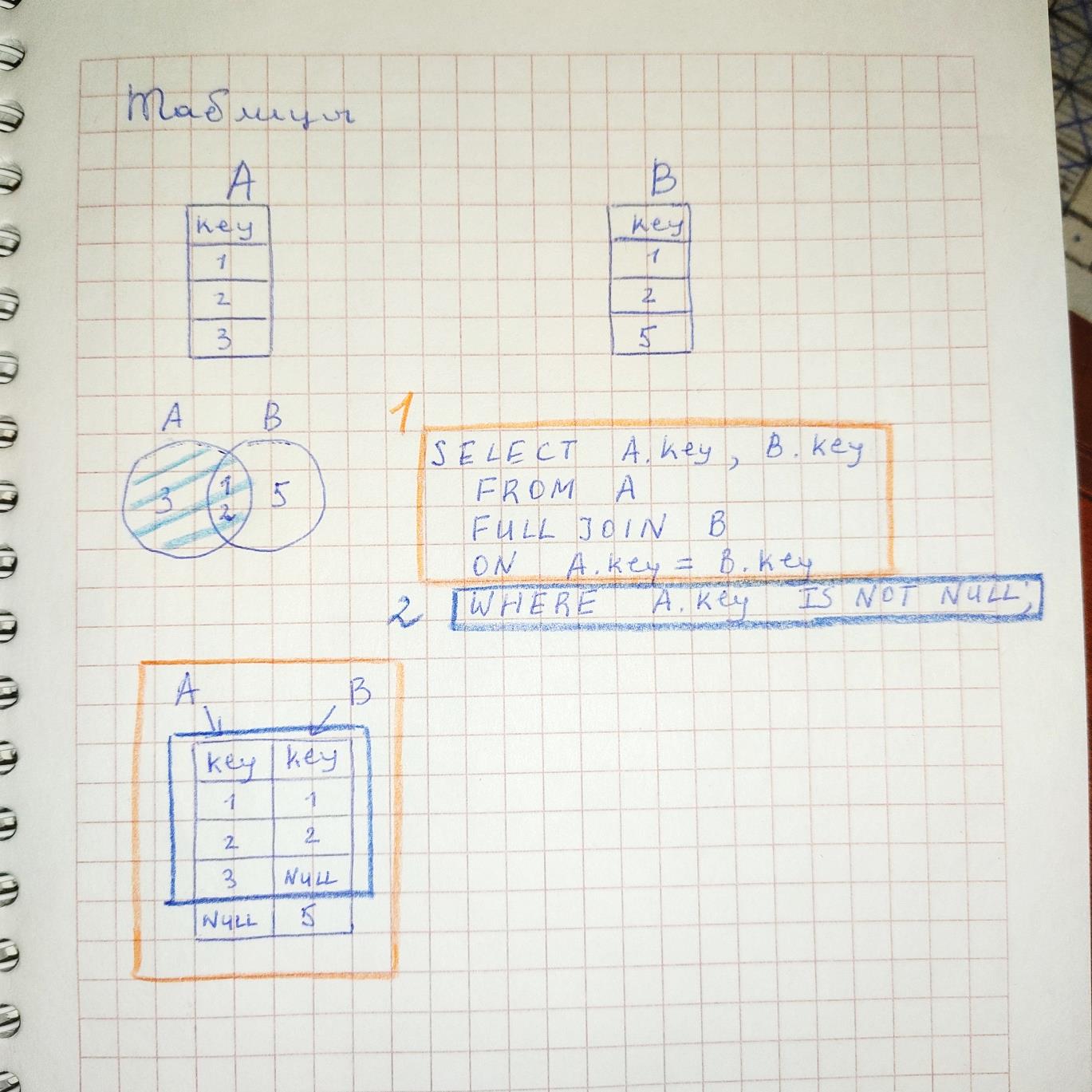


**Объяснение:**

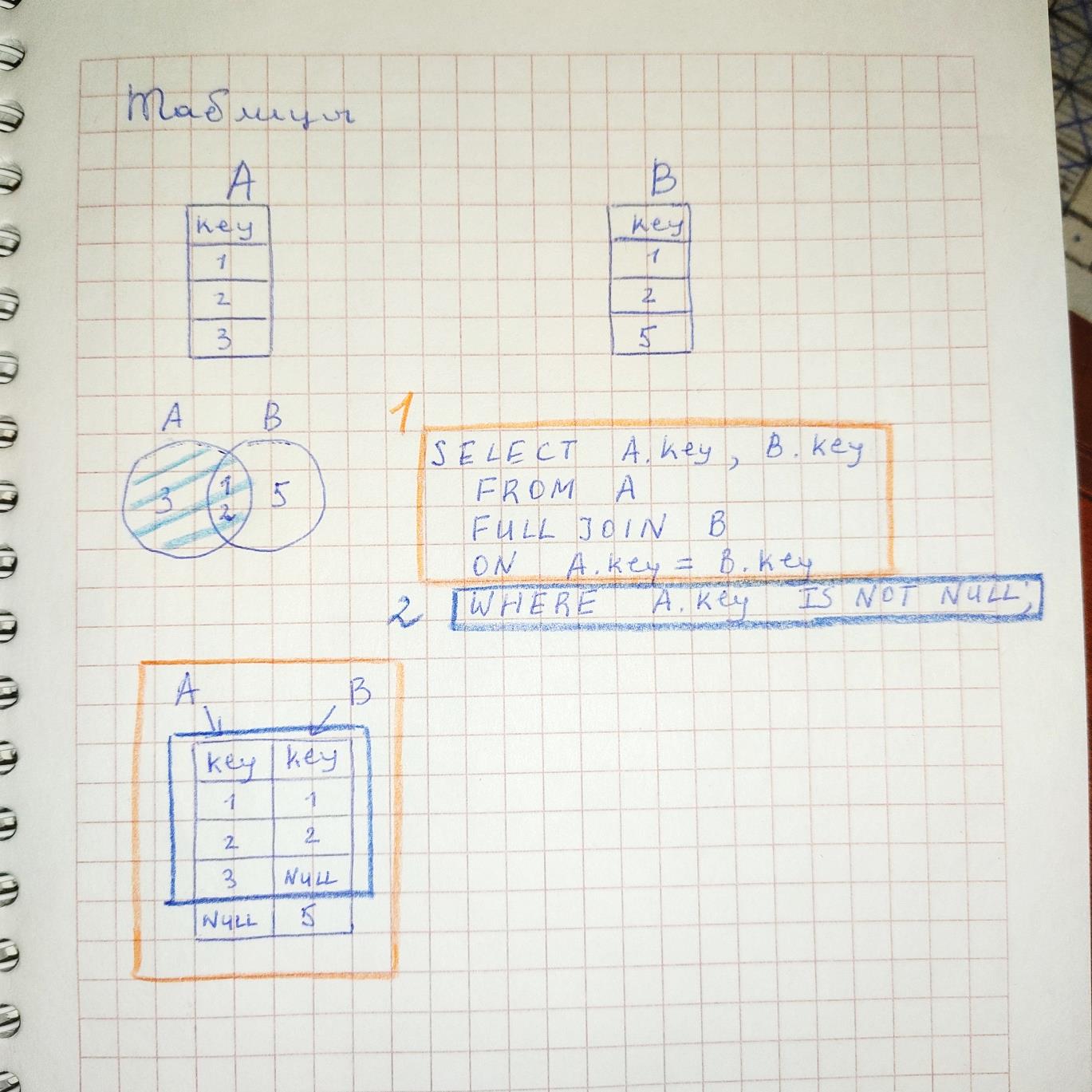
Допустим, у нас есть две таблицы: A со столбцом key, B со столбцом key.



Мы хотим получить LEFT JOIN, используя FULL JOIN. Пишем такой запрос:

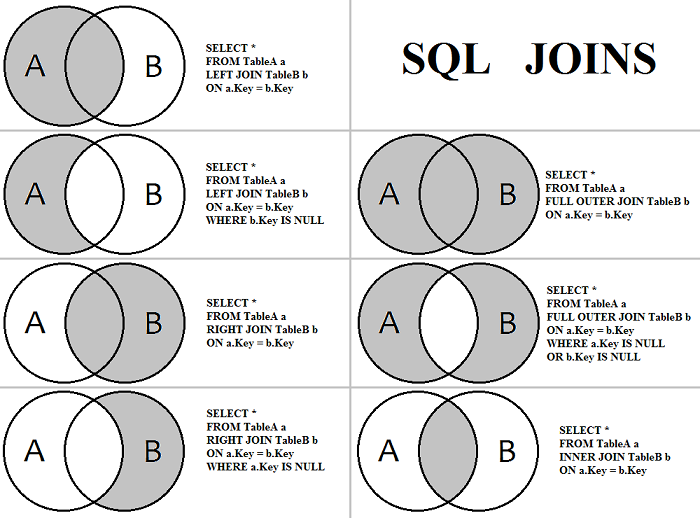


Первая часть этого запроса добивается того, чтобы, соблюдающие условие A.key = B.key, соответствия значений из двух таблиц попали в итоговую выборку. Если же значению из таблицы A не было найдено, подходящее по условию, значение из таблицы B, то вместо этого значения мы пишем NULL. То же самое для таблицы B (оранжевым обведена выборка первого блока с предыдущего условия):



Вторая же часть запроса отбрасывает значения, которые не содержатся в таблице A, то есть те NULL’ы, которые были проставлены на месте значений таблицы A, так как в таблице A не было найдено ни одного соответствия для некоторых значений из таблицы B (предыдущий рисунок (синяя обводка)).

Если нужно так:



--содержит все значения таблицы слева, кроме тех, что соответствуют значениям из таблицы справа (LEFT JOIN с исключением INNER JOIN'а)

SELECT Товары.ИД as ИД\_товара,

Товары.Название as Название\_товара,

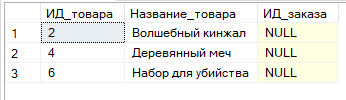
Заказы.ИД as ИД\_заказа

FROM Товары FULL JOIN Заказы

ON Товары.ИД = Заказы.Товар

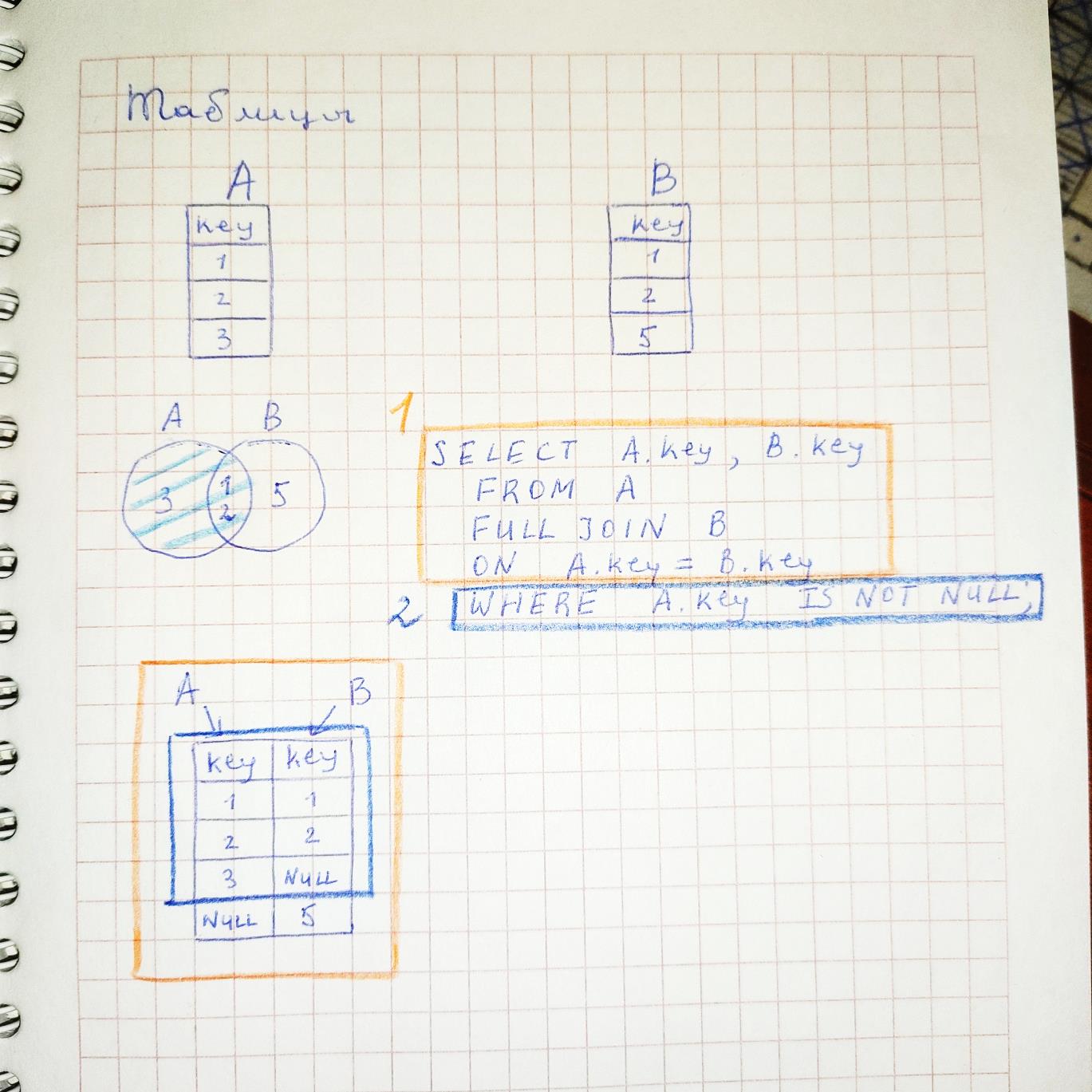
WHERE Товары.ИД IS NOT NULL AND Заказы.Товар IS NULL;

В результате мы видим все записи таблицы слева, кроме тех, что соответствуют значениям из таблицы справа:

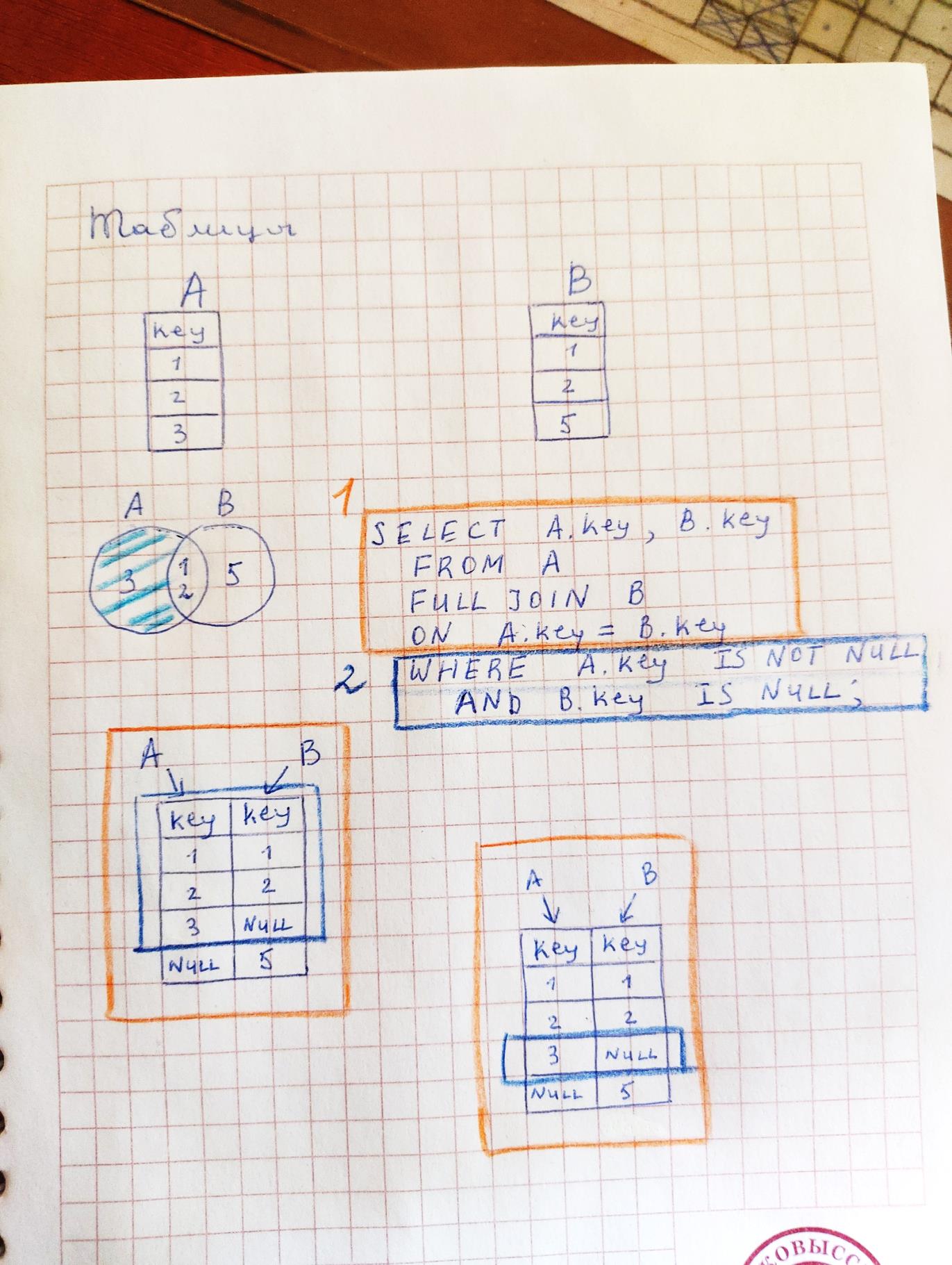


**Объяснение:**

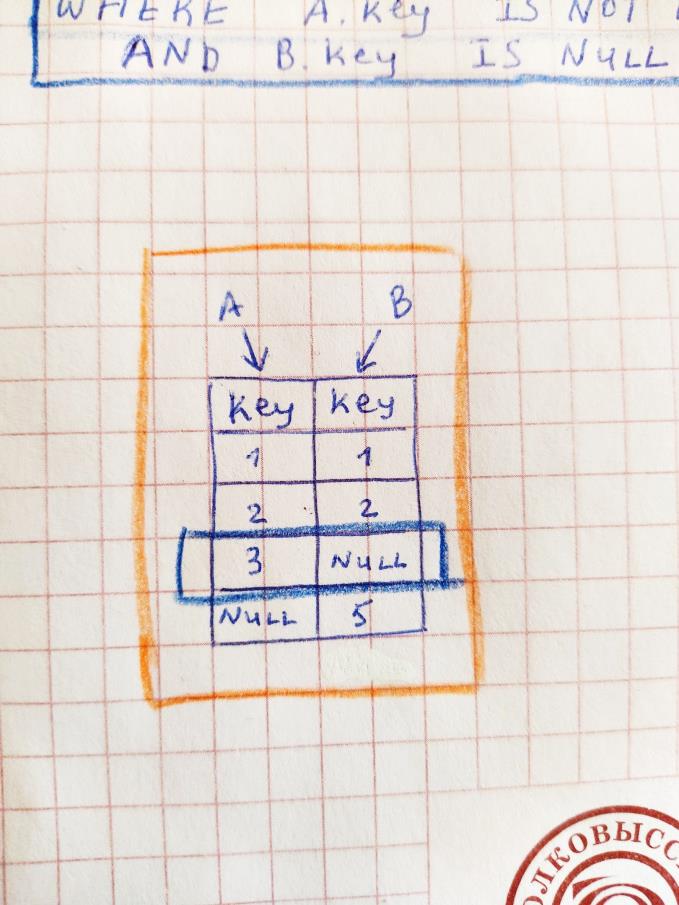
Допустим, у нас есть две таблицы: A со столбцом key, B со столбцом key.



Мы хотим получить LEFT JOIN, используя FULL JOIN. Пишем такой запрос:



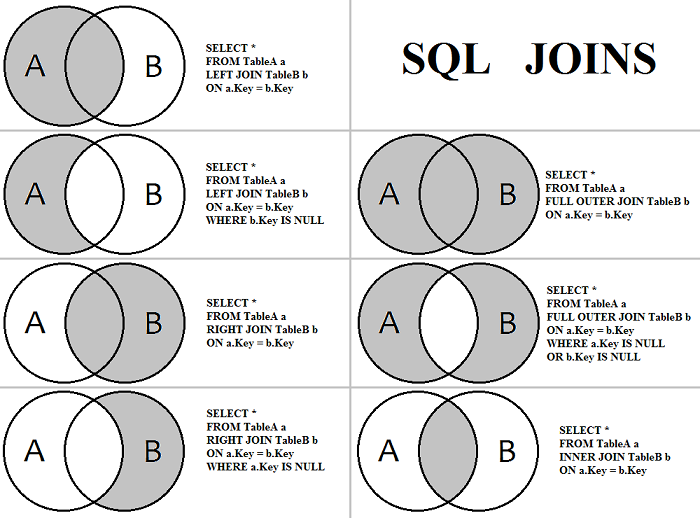
Первая часть этого запроса добивается того, чтобы, соблюдающие условие A.key = B.key, соответствия значений из двух таблиц попали в итоговую выборку. Если же значению из таблицы A не было найдено, подходящее по условию, значение из таблицы B, то вместо этого значения мы пишем NULL. То же самое для таблицы B (оранжевым обведена выборка первого блока с предыдущего условия):



Вторая же часть запроса оставляет только те строки, где у нас имеются значения из таблицы A, но соответствующих значений из таблицы B им не найдено (предыдущий рисунок (синяя обводка)).

*− запрос, результат которого содержит данные правой таблицы и не содержит данные левой;*

Если нужно так:



--содержит все значения таблицы справа (RIGHT JOIN короче)

SELECT Товары.ИД as ИД\_товара,

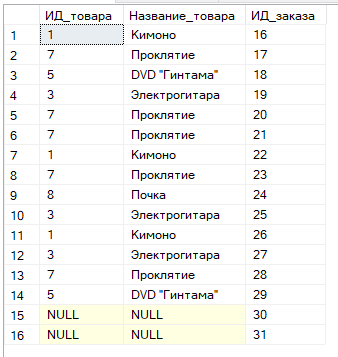
Товары.Название as Название\_товара,

Заказы.ИД as ИД\_заказа

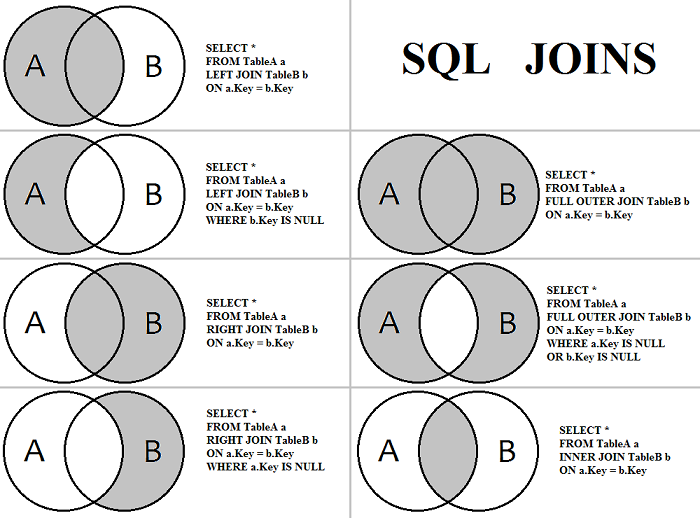
FROM Товары FULL JOIN Заказы

ON Товары.ИД = Заказы.Товар

WHERE Заказы.ИД IS NOT NULL;



Если нужно так:



--содержит все значения таблицы справа, кроме тех, что соответствуют значениям из таблицы слева (RIGHT JOIN с исключением INNER JOIN'а)

SELECT Товары.ИД as ИД\_товара,

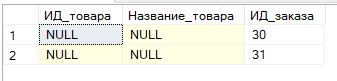
Товары.Название as Название\_товара,

Заказы.ИД as ИД\_заказа

FROM Товары FULL JOIN Заказы

ON Товары.ИД = Заказы.Товар

WHERE Заказы.ИД IS NOT NULL AND Товары.ИД IS NULL;

**

*− запрос, результат которого содержит данные правой таблицы и левой таблиц;*

SELECT Заказы.ИД AS [ИД заказа],

Заказы.Дата\_заказа AS [Дата заказа],

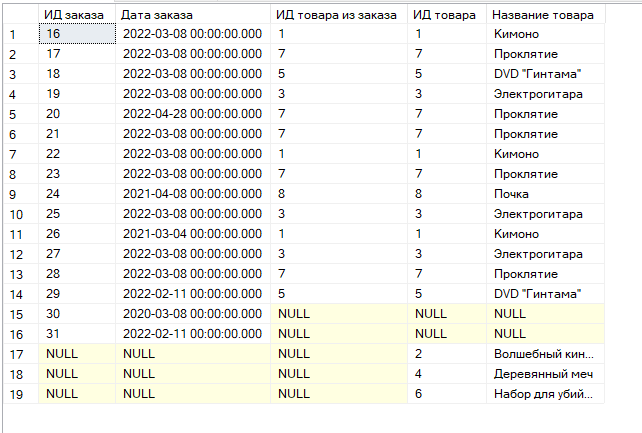
Заказы.Товар [ИД товара из заказа],

Товары.ИД AS [ИД товара],

Товары.Название AS [Название товара]

FROM Заказы FULL JOIN Товары

ON Товары.ИД = Заказы.Товар;

**

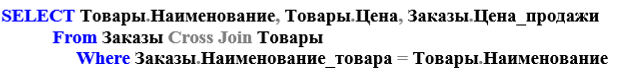
*Примечание: использовать в запросах выражение IS NULL и IS NOT NULL.*

*9.**Разработать SELECT-запрос на основе CROSS JOIN-соединения таблиц* ***AUDITORIUM\_TYPE*** *и* ***AUDITORIUM****, формирующего результат, аналогичный результату, полученному при выполнении запроса в задании 1.*

***→ Теория (методичка)***

При использовании соединения CROSS JOIN каждая строка одной таблицы соединяется с каждой строкой другой таблицы.

Например, на основании таблиц **Товары** и **Заказы** сформировать перечень товаров с ценой исходной и ценой продажи можно с помощью запроса:

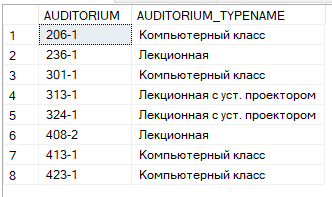


***→ Выполнение заданий***

SELECT AUDITORIUM.AUDITORIUM, AUDITORIUM\_TYPE.AUDITORIUM\_TYPENAME

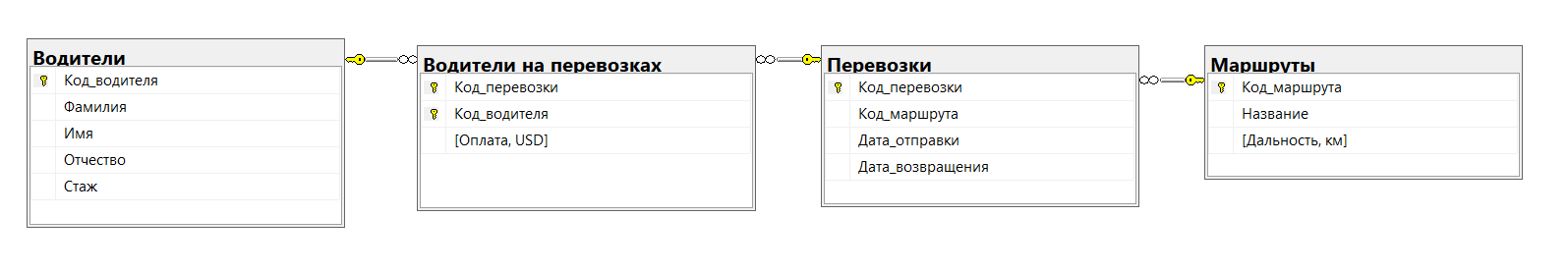
FROM AUDITORIUM CROSS JOIN AUDITORIUM\_TYPE

WHERE AUDITORIUM.AUDITORIUM\_TYPE = AUDITORIUM\_TYPE.AUDITORIUM\_TYPE;

******

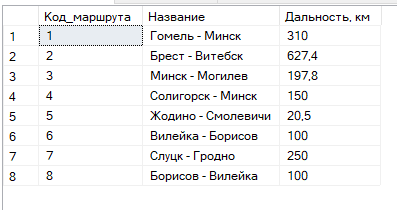
*10. Разработать и выполнить аналогичные запросы для базы данных* ***X\_MyBASE****.*

Со второй лабораторной (на третьей мы её тоже создавали) остается БД Ku\_MyBase\_2. Диаграмма этой БД:

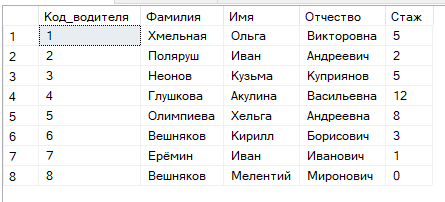


В каждой таблице по минимуму данных, так что дополняю таблицы немного. На выходе имею такие таблицы:

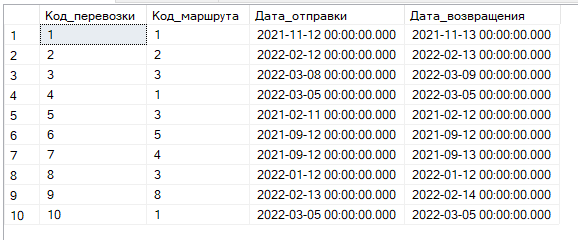
**Маршруты**

****

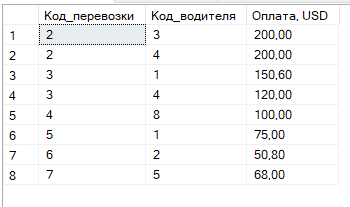
**Водители**

****

**Перевозки**

****

**Водители на перевозках**

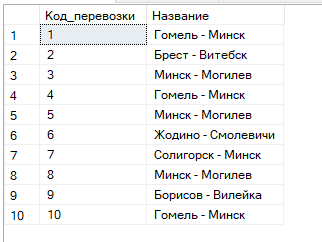
****

--1 (перевозки и соответствующие им маршруты)

SELECT Перевозки.Код\_перевозки, Маршруты.Название

FROM Перевозки INNER JOIN Маршруты

ON Перевозки.Код\_маршрута = Маршруты.Код\_маршрута;



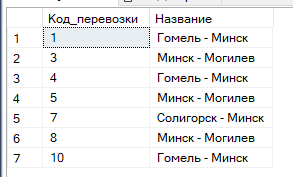
--2 (перевозки и соответствующие им маршруты, при этом в маршруте должно быть слово "Минск")

SELECT Перевозки.Код\_перевозки, Маршруты.Название

FROM Перевозки INNER JOIN Маршруты

ON Перевозки.Код\_маршрута = Маршруты.Код\_маршрута

WHERE Маршруты.Название LIKE '%Минск%';



--3 (Написать два SELECT-запроса, формирующих результирующие наборы аналогичные запросам из заданий 1 и 2, но без применения INNER JOIN)

SELECT Перевозки.Код\_перевозки, Маршруты.Название

FROM Перевозки, Маршруты

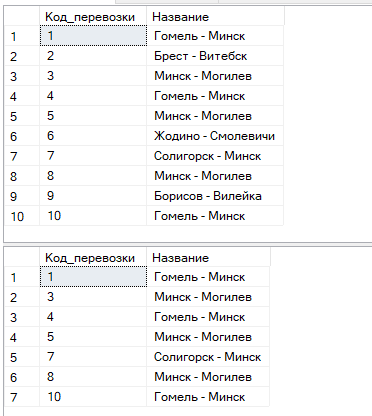
WHERE Перевозки.Код\_маршрута = Маршруты.Код\_маршрута;

SELECT Перевозки.Код\_перевозки, Маршруты.Название

FROM Перевозки, Маршруты

WHERE Перевозки.Код\_маршрута = Маршруты.Код\_маршрута

AND Маршруты.Название LIKE '%Минск%';



--4 (На основе таблиц "Водители" и "Водители на перевозках" сформировать перечень водителей, оплата которых равна 100, 75 и 200.

--Результирующий набор должен содержать столбцы: фамилия, имя, отчество водителя и оплата. Оплату записать прописью.

--Результирующий набор отсортировать в порядке возрастания по столбцам фио и в порядке убывания по оплате)

--Примечание: использовать соединение INNER JOIN, предикат BETWEEN (в нашем случае не нужен) и выражение CASE.

SELECT Водители.Фамилия,

Водители.Имя,

Водители.Отчество,

CASE [Водители на перевозках].[Оплата, USD]

WHEN 75 THEN 'семьдесят пять'

WHEN 100 THEN 'сто'

WHEN 200 THEN 'двести'

END

FROM Водители INNER JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя

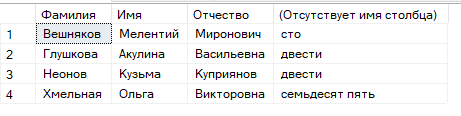
WHERE [Водители на перевозках].[Оплата, USD] IN (75, 100, 200)

ORDER BY Водители.Фамилия,

Водители.Имя,

Водители.Отчество,

[Водители на перевозках].[Оплата, USD] DESC;



--5 (Переписать запрос, реализующий задание 4 таким образом, чтобы в результирующем наборе сортировка по оплате была следующей:

--сначала выводились строки с оплатой 75, затем строки с оплатой 200 и далее строки с оплатой 100. Примечание: использовать выражение CASE в секции ORDER BY.)

SELECT Водители.Фамилия,

Водители.Имя,

Водители.Отчество,

CASE [Водители на перевозках].[Оплата, USD]

WHEN 75 THEN 'семьдесят пять'

WHEN 100 THEN 'сто'

WHEN 200 THEN 'двести'

END

FROM Водители INNER JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя

WHERE [Водители на перевозках].[Оплата, USD] IN (75, 100, 200)

ORDER BY

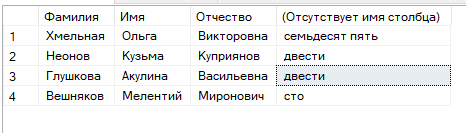
(CASE [Водители на перевозках].[Оплата, USD]

WHEN 75 THEN 1

WHEN 100 THEN 3

WHEN 200 THEN 2

END);



--6 (Получить полный перечень маршрутов, а также коды перевозок соответствующих им и ФИО с кодами водителей на этих перевозках.

--При этом, если ФИО пустое, то заменить его на "\*\*\*"

SELECT Маршруты.Название,

Перевозки.Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM Маршруты LEFT JOIN Перевозки

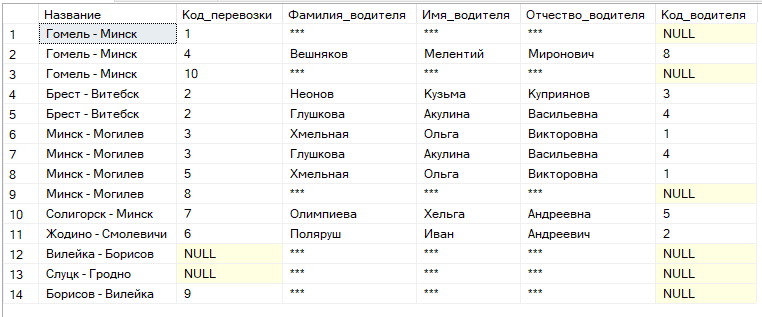
ON Маршруты.Код\_маршрута = Перевозки.Код\_маршрута

LEFT JOIN [Водители на перевозках]

ON Перевозки.Код\_перевозки = [Водители на перевозках].Код\_перевозки

LEFT JOIN Водители

ON [Водители на перевозках].Код\_водителя = Водители.Код\_водителя;



--7 (Повторить результат предыдущего запроса, используя RIGHT JOIN)

SELECT Маршруты.Название,

Перевозки.Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM Водители RIGHT JOIN [Водители на перевозках]

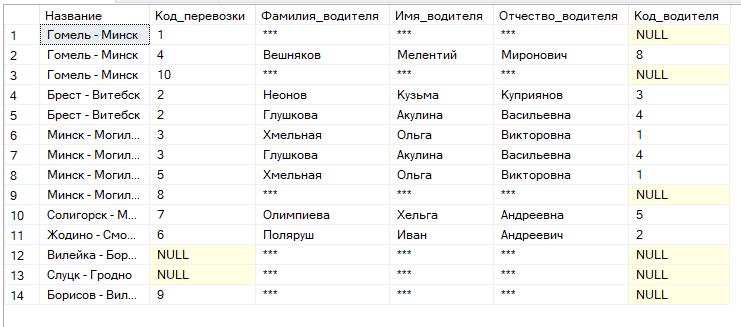
ON [Водители на перевозках].Код\_водителя = Водители.Код\_водителя

RIGHT JOIN Перевозки

ON Перевозки.Код\_перевозки = [Водители на перевозках].Код\_перевозки

RIGHT JOIN Маршруты

ON Маршруты.Код\_маршрута = Перевозки.Код\_маршрута;



--8 (Показать на примере, что соединение FULL OUTER JOIN двух таблиц: )

--8.1 (является коммутативной операцией;)

--следующие два запроса выводят соответствие между перевозками и водителями на них, если какой-либо перевозке не соответствует

--водитель, то в поле перевозки мы увидим NULL, и то же самое, если водителю не соответствует ни одной перевозки

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM Водители FULL JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

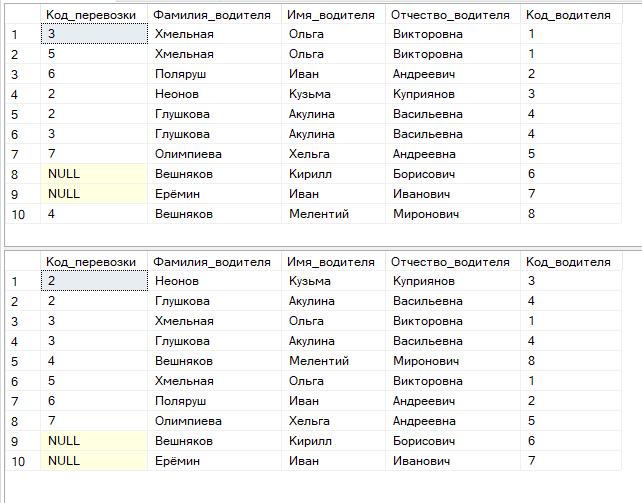
isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM [Водители на перевозках] FULL JOIN Водители

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;



--8.2 (является объединением LEFT OUTER JOIN и RIGHT OUTER JOIN соединений этих таблиц;)

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM [Водители на перевозках] LEFT JOIN Водители

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя

UNION

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

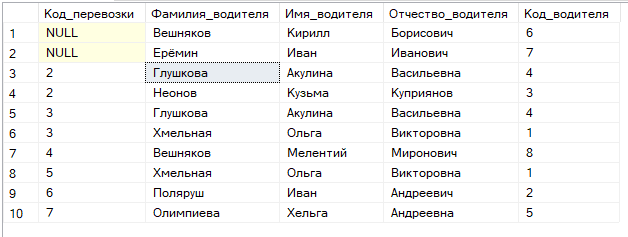
isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM [Водители на перевозках] RIGHT JOIN Водители

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;



--8.3 (включает соединение INNER JOIN этих таблиц.)

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

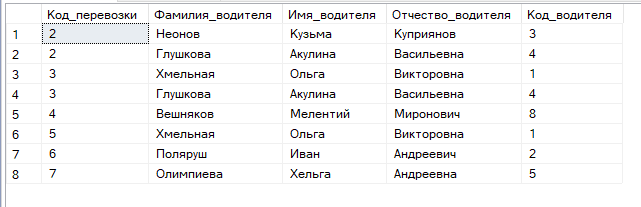
isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM Водители INNER JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;



--8 Создать три новых запроса:

--8.4 запрос, результат которого содержит данные левой (в операции FULL OUTER JOIN) таблицы и не содержит данные правой;

--LEFT JOIN с исключением INNER JOIN'а

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

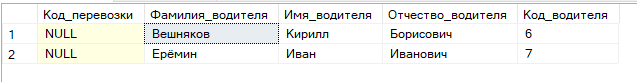
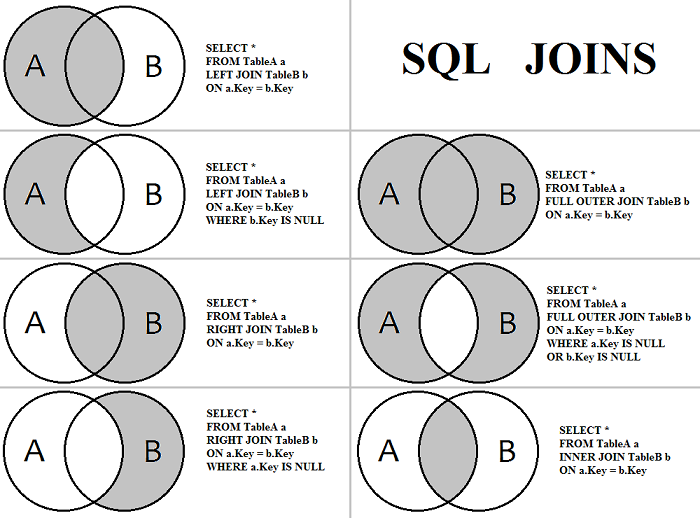
FROM Водители FULL JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя

WHERE Водители.Код\_водителя IS NOT NULL

AND [Водители на перевозках].Код\_водителя IS NULL

AND [Водители на перевозках].Код\_перевозки IS NULL;



--8.5 запрос, результат которого содержит данные правой таблицы и не содержащие данные л-вой;

--RIGHT JOIN с исключением INNER JOIN'а

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

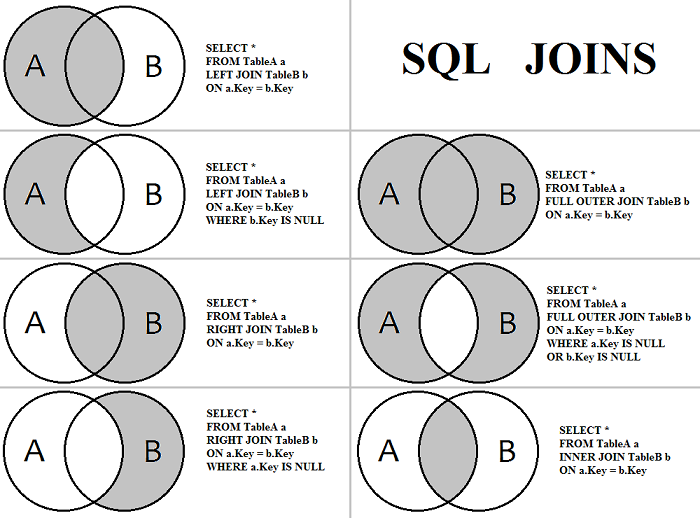
FROM Водители FULL JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя

WHERE Водители.Код\_водителя IS NULL

AND [Водители на перевозках].Код\_водителя IS NOT NULL

AND [Водители на перевозках].Код\_перевозки IS NOT NULL;



--8.6 запрос, результат которого содержит данные правой таблицы и левой таблиц;

SELECT [Водители на перевозках].Код\_перевозки,

isnull(Водители.Фамилия, '\*\*\*') AS Фамилия\_водителя,

isnull(Водители.Имя, '\*\*\*') AS Имя\_водителя,

isnull(Водители.Отчество, '\*\*\*') AS Отчество\_водителя,

Водители.Код\_водителя

FROM Водители FULL JOIN [Водители на перевозках]

ON Водители.Код\_водителя = [Водители на перевозках].Код\_водителя;

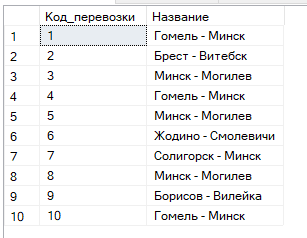


--9 (Разработать SELECT-запрос на основе CROSS JOIN-соединения таблиц, формирующего результат, аналогичный результату, полученному при выполнении запроса в задании 1.)

SELECT Перевозки.Код\_перевозки, Маршруты.Название

FROM Перевозки CROSS JOIN Маршруты

WHERE Перевозки.Код\_маршрута = Маршруты.Код\_маршрута;



### Приложение 1

**Таблицы базы данных UNIVER**

|  |  |  |
| --- | --- | --- |
| **Имя таблицы** | **Столбцы таблицы** | **Наименования столбцов, свойства столбцов** |
| FACULTY  (Факультеты) | FACULTY | код факультета, PK, char(10), not null |
| FACULTY\_NAME | наименование факультета, varchar(50), default ‘???’ |
| PROFESSION  (Специальности) | PROFESSION | код специальности, PK, char(20). not null |
| FACULTY | код факультета, FK(FACULTY), char(10), not null |
| PROFESSION\_NAME | наименование специальности, varchar(100), null |
| QUALIFICATION | квалификация, varchar(50), null |
| PULPIT  (Кафедры) | PULPIT | код кафедры, PK, char(20), not null |
| PULPIT\_NAME | наименование кафедры, varchar(100), null |
| FACULTY | код факультета, FK(FA-CULTY), char(10), not null |
| TEACHER  (Преподаватели) | TEACHER | код преподавателя, PK, char(10), not null |
| TEACHER\_NAME | фамилия, имя, отчество преподавателя, varchar(100),null |
| GENDER | пол, char(1), GENDER in (‘м’,’ж’) |
| PULPIT | код кафедры, FK(PULPIT), char(10), not null |
| SUBJECT  (Дисциплины) | SUBJECT | код дисциплины, PK, char(10), not null |
| SUBJECT\_NAME | наименование дисциплины, varchar(100), null, unique |
| PULPIT | код кафедры, FK(PULPIT), char(20), not null |
| AUDITORIUM\_ TYPE (Типы учебных аудиторий) | AUDITORIUM\_TYPE | код типа аудитории, PK, char(10), not null |
| AUDITORIUM\_ TYPENAME | наименование типа аудитории, varchar(30), null |
| AUDITORIUM  (Учебные аудитории) | AUDITORIUM | код аудитории, PK, char(20). not null |
| AUDITORIUM\_TYPE | код типа аудитории, FK(AUDITORIUM\_TYPE), char(10), not null |
| AUDITORIUM\_CAPACITY | вместимость, int, default 1, check between 1 and 300 |
| AUDITORIUM\_NAME | наименование аудитории, varchar(50), null |
| GROUP  (Студенческие группы) | IDGROUP | идентификатор группы, PK, int, not null |
| FACULTY | код факультета, FK(FACULTY), char(10), not null |
| PROFESSION | код специальности, FK(PROFESSION) , char(20), not null |
| YEAR\_FIRST | год поступления, smallint, < текущий год +2 |
| COURSE | курс, вычисляемое поле, tinyint, вычисляется на основе текущей даты и значения YEAR\_FIRST |
| STUDENT  (Студент) | IDSTUDENT | Код студента, РK(STUDENT), int, identity (1000,1) |
| IDGROUP | идентификатор группы, FK(GROUP), int, not null |
| NAME | фамилия, имя, отчество, nvarchar(100) |
| BDAY | дата рождения, date |
| STAMP | штамп времени, timestamp |
| INFO | дополнительная информация, xml, по умолчанию null |
| FOTO | фотография, varbinary(max), по умолчанию null |
| PROGRESS  (Оценки на экзамене) | SUBJECT | предмет, FK (SUBJECT), char(10) |
| IDSTUDENT | Код студента, FK(STUDENT), int |
| PDATE | дата экзамена, date |
| NOTE | оценка, int (between 1 and 10) |

**Содержимое таблицы FACULTY**

|  |  |
| --- | --- |
| FACULTY | FACULTY\_NAME |
| ТТЛП | Технологии и техника лесной промышленности |
| ТОВ | Технологии органических веществ |
| ХТиТ | Химические технологии и техника |
| ИЭФ | Инженерно-экономический |
| ЛХ | Лесохозяйственный |
| ИДиП | Издательское дело и полиграфия |
| ИТ | Информационных технологий |

**Содержимое таблицы PROFESSION**

|  |  |  |  |
| --- | --- | --- | --- |
| PROFESSION | FACULTY | PROFESSION\_NAME | QUALIFICATION |
| 1-36 06 01 | ИДиП | Полиграфическое оборудование и системы обработки информации | инженер-электромеханик |
| 1-36 07 01 | ХТиТ | Машины и аппараты химических производств и предприятий строительных материалов | инженер-механик |
| 1-40 01 02 | ИТ | Информационные системы и технологии | инженер-программист-системотехник |
| 1-46 01 01 | ТТЛП | Лесоинженерное дело | инженер-технолог |
| 1-47 01 01 | ИДиП | Издательское дело | редактор-технолог |
| 1-48 01 02 | ТОВ | Химическая технология органических веществ, материалов и изделий | инженер-химик-технолог |
| 1-48 01 05 | ТОВ | Химическая технология переработки древесины | инженер-химик-технолог |
| 1-54 01 03 | ТОВ | Физико-химические методы и приборы контроля качества продукции | инженер по сертификации |
| 1-75 01 01 | ЛХФ | Лесное хозяйство | инженер лесного хозяйства |
| 1-75 02 01 | ЛХФ | Садово-парковое строительство | инженер садово-паркового строительства |
| 1-89 02 02 | ЛХФ | Туризм и природопользование | специалист в сфере туризма |

**Содержимое таблицы PULPIT**

|  |  |  |  |
| --- | --- | --- | --- |
| PULPIT | PULPIT\_NAME | | FACULTY |
| РИТ | | Редакционно-издательских тенологий | ИДиП |
| СБУАиА | | Статистики, бухгалтерского учета, анализа и аудита | ИЭФ |
| ТДП | | Технологий деревообрабатывающих производств | ТТЛП |
| ТиДИД | | Технологии и дизайна изделий из древесины | ТТЛП |
| ТиП | | Туризма и природопользования | ЛХФ |
| ТЛ | | Транспорта леса | ТТЛП |
| ТНВиОХТ | | Технологии неорганических веществ и общей химической технологии | ХТиТ |
| ТНХСиППМ | | Технологии нефтехимического синтеза и переработки полимерных материалов | ТОВ |
| ХПД | | Химической переработки древесины | ТОВ |
| ХТЭПиМЭЕ | | Химии, технологии электрохимических производств и материалов электронной техники | ХТиТ |
| ЭТиМ | | Экономической теории и маркетинга | ИЭФ |

**Содержимое таблицы TEACHER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TEACHER | | TEACHER\_NAME | GENDER | PULPIT | |
| НСКВ | Носков Михаил Трофимович | NULL | | ТТЛП | |
| ПРКП | Прокопенко Николай Иванович | NULL | | ТНХСиППМ | |
| МРЗВ | Морозова Елена Степановна | NULL | | ИСиТ | |
| РВКС | Ровкас Андрей Петрович | NULL | | ОВ | |
| РЖКВ | Рыжиков Леонид Николаевич | NULL | | ЛВ | |
| РМНВ | Романов Дмитрий Михайлович | NULL | | ИСиТ | |
| СМЛВ | Смелов Владимир Владиславович | NULL | | ИСиТ | |
| КРЛВ | Крылов Павел Павлович | NULL | | ИСиТ | |
| ЧРН | Чернова Анна Викторовна | NULL | | ХПД | |
| МХВ | Мохов Михаил Сергеевич | NULL | | ПОиСОИ | |

**Содержимое таблицы SUBJECT**

|  |  |  |  |
| --- | --- | --- | --- |
| SUBJECT | | SUBJECT\_NAME | PULPIT |
| ПЗ | Представление знаний в компьютерных системах | ИСиТ | |
| ПИС | Проектирование информационных систем | ИСиТ | |
| ПСП | Программирование сетевых приложений | ИСиТ | |
| ПЭХ | Прикладная электрохимия | ХТЭПиМЭЕ | |
| СУБД | Системы управления базами данных | ИСиТ | |
| ТиОЛ | Технология и оборудование лесозаготовок | ЛМиЛЗ | |
| ТРИ | Технология резиновых изделий | ТНХСиППМ | |
| ЭП | Экономика природопользования | МиЭП | |
| ЭТ | Экономическая теория | ЭТиМ | |

**Содержимое таблицы AUDITORIUM**

|  |  |  |  |
| --- | --- | --- | --- |
| AUDITORIUM | AUDITORIUM\_ TYPE | AUDITORIUM\_ CAPACITY | AUDITORIUM\_ NAME |
| 301-1 | ЛБ-К | 15 | 301-1 |
| 304-4 | ЛБ-К | 90 | 304-4 |
| 313-1 | ЛК-К | 60 | 313-1 |
| 314-4 | ЛК | 90 | 314-4 |
| 320-4 | ЛК | 90 | 320-4 |
| 324-1 | ЛК-К | 50 | 324-1 |
| 413-1 | ЛБ-К | 15 | 413-1 |
| 423-1 | ЛБ-К | 90 | 423-1 |

**Содержимое таблицы AUDITORIUM\_TYPE**

|  |  |
| --- | --- |
| AUDITORIUM\_TYPE | AUDITORIUM\_TYPENAME |
| ЛБ-X | Химическая лаборатория |
| ЛБ-К | Компьютерный класс |
| ЛБ-СК | Спец. компьютерный класс |
| ЛК | Лекционная |
| ЛК-К | Лекционная с уст. проектором |

**Содержимое таблицы GROUP**

|  |  |  |  |
| --- | --- | --- | --- |
| IDGROUP | FACULTY | PROFESSION | YEAR\_FIRST |
| 22 | ЛХФ | 1-75 02 01 | 2011 |
| 23 | ЛХФ | 1-89 02 02 | 2012 |
| 24 | ЛХФ | 1-89 02 02 | 2011 |
| 25 | ТТЛП | 1-36 05 01 | 2013 |
| 26 | ТТЛП | 1-36 05 01 | 2012 |
| 27 | ТТЛП | 1-46 01 01 | 2012 |
| 28 | ИЭФ | 1-25 01 07 | 2013 |
| 29 | ИЭФ | 1-25 01 07 | 2012 |
| 30 | ИЭФ | 1-25 01 07 | 2010 |
| 31 | ИЭФ | 1-25 01 08 | 2013 |
| 32 | ИЭФ | 1-25 01 08 | 2012 |

**Содержимое таблицы STUDENT**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IDSTUDENT | IDGROUP | NAME | BDAY | STAMP | INFO | FOTO |
| 1000 | 22 | Пугач Михаил Трофимович | 12/01/1996 |  |  |  |
| 1001 | 23 | Авдеев Николай Иванович | 19/07/1996 |  |  |  |
| 1002 | 24 | Белова Елена Степановна | 22/05/1996 |  |  |  |
| 1003 | 25 | Вилков Андрей Петрович | 08/12/1996 |  |  |  |
| 1004 | 26 | Грушин Леонид Николаевич | 11/11/1995 |  |  |  |
| 1005 | 27 | Дунаев Дмитрий Михайлович | 24/08/1996 |  |  |  |
| 1006 | 28 | Клуни Иван Владиславович | 15/09/1996 |  |  |  |
| 1007 | 29 | Крылов Олег Павлович | 16/10/1996 |  |  |  |

**Содержимое таблицы PROGRESS**

|  |  |  |  |
| --- | --- | --- | --- |
| SUBJECT | IDSTUDENT | PDATE | NOTE |
| ОАиП | 1000 | 12/01/2014 | 4 |
| ОАиП | 1001 | 19/01/2014 | 5 |
| ОАиП | 1003 | 08/01/2014 | 9 |
| БД | 1008 | 11/01/2014 | 8 |
| БД | 1010 | 15/01/2014 | 4 |
| СУБД | 1013 | 16/01/2014 | 7 |
| СУБД | 1014 | 27/01/2014 | 6 |

[В начало практикума](#_Содержание)

**Сценарии создания таблиц базы данных UNIVER**

(данные произвольные)

|  |  |
| --- | --- |
| drop table PROGRESS  drop table STUDENT  drop table GROUPS  drop table SUBJECT  drop table TEACHER  drop table PULPIT  drop table PROFESSION  drop table FACULTY  drop table AUDITORIUM  drop table AUDITORIUM\_TYPE  ------------**Создание и заполнение таблицы AUDITORIUM\_TYPE**  create table AUDITORIUM\_TYPE  ( AUDITORIUM\_TYPE char(10) constraint AUDITORIUM\_TYPE\_PK primary key,  AUDITORIUM\_TYPENAME varchar(30)  )  insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛК', 'Лекционная');  insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-К', 'Компьютерный класс');  insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛК-К', 'Лекционная с уст. проектором');  insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-X', 'Химическая лаборатория');  insert into AUDITORIUM\_TYPE (AUDITORIUM\_TYPE, AUDITORIUM\_TYPENAME ) values ('ЛБ-СК', 'Спец. компьютерный класс');    -------------**Создание и заполнение таблицы AUDITORIUM**  create table AUDITORIUM  ( AUDITORIUM char(20) constraint AUDITORIUM\_PK primary key,  AUDITORIUM\_TYPE char(10) constraint AUDITORIUM\_AUDITORIUM\_TYPE\_FK foreign key  references AUDITORIUM\_TYPE(AUDITORIUM\_TYPE),  AUDITORIUM\_CAPACITY integer constraint AUDITORIUM\_CAPACITY\_CHECK default 1 check (AUDITORIUM\_CAPACITY between 1 and 300), -- вместимость  AUDITORIUM\_NAME varchar(50)  )  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY)  values ('206-1', '206-1','ЛБ-К', 15);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY)  values ('301-1', '301-1', 'ЛБ-К', 15);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('236-1', '236-1', 'ЛК', 60);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('313-1', '313-1', 'ЛК-К', 60);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('324-1', '324-1', 'ЛК-К', 50);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('413-1', '413-1', 'ЛБ-К', 15);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('423-1', '423-1', 'ЛБ-К', 90);  insert into AUDITORIUM (AUDITORIUM, AUDITORIUM\_NAME,  AUDITORIUM\_TYPE, AUDITORIUM\_CAPACITY )  values ('408-2', '408-2', 'ЛК', 90);  ------**Создание и заполнение таблицы FACULTY**  create table FACULTY  ( FACULTY char(10) constraint FACULTY\_PK primary key,  FACULTY\_NAME varchar(50) default '???'  );  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ХТиТ', 'Химическая технология и техника');  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ЛХФ', 'Лесохозяйственный факультет');  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ИЭФ', 'Инженерно-экономический факультет');  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ТТЛП', 'Технология и техника лесной промышленности');  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ТОВ', 'Технология органических веществ');  insert into FACULTY (FACULTY, FACULTY\_NAME )  values ('ИТ', 'Факультет информационных технологий');  ------**Создание и заполнение таблицы PROFESSION**  create table PROFESSION  ( PROFESSION char(20) constraint PROFESSION\_PK primary key,  FACULTY char(10) constraint PROFESSION\_FACULTY\_FK foreign key  references FACULTY(FACULTY),  PROFESSION\_NAME varchar(100), QUALIFICATION varchar(50)  );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-40 01 02', 'Информационные системы и технологии', 'инженер-программист-системотехник' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-47 01 01', 'Издательское дело', 'редактор-технолог' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИТ', '1-36 06 01', 'Полиграфическое оборудование и системы обработки информации', 'инженер-электромеханик' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ХТиТ', '1-36 01 08', 'Конструирование и производство изделий из композиционных материалов', 'инженер-механик' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ХТиТ', '1-36 07 01', 'Машины и аппараты химических производств и предприятий строительных материалов', 'инженер-механик' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-75 01 01', 'Лесное хозяйство', 'инженер лесного хозяйства' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-75 02 01', 'Садово-парковое строительство', 'инженер садово-паркового строительства' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ЛХФ', '1-89 02 02', 'Туризм и природопользование', 'специалист в сфере туризма' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИЭФ', '1-25 01 07', 'Экономика и управление на предприятии', 'экономист-менеджер' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ИЭФ', '1-25 01 08', 'Бухгалтерский учет, анализ и аудит', 'экономист' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТТЛП', '1-36 05 01', 'Машины и оборудование лесного комплекса', 'инженер-механик' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТТЛП', '1-46 01 01', 'Лесоинженерное дело', 'инженер-технолог' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-48 01 02', 'Химическая технология органических веществ, материалов и изделий', 'инженер-химик-технолог' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-48 01 05', 'Химическая технология переработки древесины', 'инженер-химик-технолог' );  insert into PROFESSION(FACULTY, PROFESSION, PROFESSION\_NAME, QUALIFICATION) values ('ТОВ', '1-54 01 03', 'Физико-химические методы и приборы контроля качества продукции', 'инженер по сертификации' );  ------**Создание и заполнение таблицы PULPIT**  create table PULPIT  ( PULPIT char(20) constraint PULPIT\_PK primary key,  PULPIT\_NAME varchar(100),  FACULTY char(10) constraint PULPIT\_FACULTY\_FK foreign key  references FACULTY(FACULTY)  );  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY )  values ('ИСиТ', 'Информационных систем и технологий ','ИТ' )  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛВ', 'Лесоводства','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛУ', 'Лесоустройства','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛЗиДВ', 'Лесозащиты и древесиноведения','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛКиП', 'Лесных культур и почвоведения','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ТиП', 'Туризма и природопользования','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛПиСПС','Ландшафтного проектирования и садово-паркового строительства','ЛХФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ТЛ', 'Транспорта леса', 'ТТЛП')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЛМиЛЗ','Лесных машин и технологии лесозаготовок','ТТЛП')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ТДП','Технологий деревообрабатывающих производств', 'ТТЛП')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ТиДИД','Технологии и дизайна изделий из древесины','ТТЛП')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ОХ', 'Органической химии','ТОВ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ХПД','Химической переработки древесины','ТОВ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ТНВиОХТ','Технологии неорганических веществ и общей химической технологии ','ХТиТ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ПиАХП','Процессов и аппаратов химических производств','ХТиТ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('ЭТиМ', 'Экономической теории и маркетинга','ИЭФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('МиЭП', 'Менеджмента и экономики природопользования','ИЭФ')  insert into PULPIT (PULPIT, PULPIT\_NAME, FACULTY)  values ('СБУАиА', 'Статистики, бухгалтерского учета, анализа и аудита', 'ИЭФ')    ------**Создание и заполнение таблицы TEACHER**  create table TEACHER  ( TEACHER char(10) constraint TEACHER\_PK primary key,  TEACHER\_NAME varchar(100),  GENDER char(1) CHECK (GENDER in ('м', 'ж')),  PULPIT char(20) constraint TEACHER\_PULPIT\_FK foreign key  references PULPIT(PULPIT)  );  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('СМЛВ', 'Смелов Владимир Владиславович', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('АКНВЧ', 'Акунович Станислав Иванович', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('КЛСНВ', 'Колесников Виталий Леонидович', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('БРКВЧ', 'Бракович Андрей Игоревич', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('ДТК', 'Дятко Александр Аркадьевич', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('УРБ', 'Урбанович Павел Павлович', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('ГРН', 'Гурин Николай Иванович', 'м', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('ЖЛК', 'Жиляк Надежда Александровна', 'ж', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('МРЗ', 'Мороз Елена Станиславовна', 'ж', 'ИСиТ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('БРТШВЧ', 'Барташевич Святослав Александрович', 'м','ПОиСОИ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('АРС', 'Арсентьев Виталий Арсентьевич', 'м', 'ПОиСОИ');  insert into TEACHER (TEACHER, TEACHER\_NAME,GENDER, PULPIT )  values ('БРНВСК', 'Барановский Станислав Иванович', 'м', 'ЭТиМ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('НВРВ', 'Неверов Александр Васильевич', 'м', 'МиЭП');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('РВКЧ', 'Ровкач Андрей Иванович', 'м', 'ЛВ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('ДМДК', 'Демидко Марина Николаевна', 'ж', 'ЛПиСПС');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('БРГ', 'Бурганская Татьяна Минаевна', 'ж', 'ЛПиСПС');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('РЖК', 'Рожков Леонид Николаевич ', 'м', 'ЛВ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('ЗВГЦВ', 'Звягинцев Вячеслав Борисович', 'м', 'ЛЗиДВ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('БЗБРДВ', 'Безбородов Владимир Степанович', 'м', 'ОХ');  insert into TEACHER (TEACHER, TEACHER\_NAME, GENDER, PULPIT )  values ('НСКВЦ', 'Насковец Михаил Трофимович', 'м', 'ТЛ'); | ------**Создание и заполнение таблицы SUBJECT**  create table SUBJECT  ( SUBJECT char(10) constraint SUBJECT\_PK primary key,  SUBJECT\_NAME varchar(100) unique,  PULPIT char(20) constraint SUBJECT\_PULPIT\_FK foreign key  references PULPIT(PULPIT)  )  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('СУБД', 'Системы управления базами данных', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT)  values ('БД', 'Базы данных','ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ИНФ', 'Информационные технологии','ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ОАиП', 'Основы алгоритмизации и программирования', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ПЗ', 'Представление знаний в компьютерных системах', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ПСП', 'Программирование сетевых приложений', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('МСОИ', 'Моделирование систем обработки информации', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ПИС', 'Проектирование информационных систем', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('КГ', 'Компьютерная геометрия ','ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ПМАПЛ', 'Полиграф. машины, автоматы и поточные линии', 'ПОиСОИ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('КМС', 'Компьютерные мультимедийные системы', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ОПП', 'Организация полиграф. производства', 'ПОиСОИ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT)  values ('ДМ', 'Дискретная математика', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('МП', 'Математическое программирование','ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ЛЭВМ', 'Логические основы ЭВМ', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ООП', 'Объектно-ориентированное программирование', 'ИСиТ');  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ЭП', 'Экономика природопользования','МиЭП')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ЭТ', 'Экономическая теория','ЭТиМ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('БЛЗиПсOO','Биология лесных зверей и птиц с осн. охотов.','ОВ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ОСПиЛПХ','Основы садово-паркового и лесопаркового хозяйства', 'ЛПиСПС')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ИГ', 'Инженерная геодезия ','ЛУ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ЛВ', 'Лесоводство', 'ЛЗиДВ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ОХ', 'Органическая химия', 'ОХ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ТРИ', 'Технология резиновых изделий','ТНХСиППМ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ВТЛ', 'Водный транспорт леса','ТЛ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ТиОЛ', 'Технология и оборудование лесозаготовок', 'ЛМиЛЗ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME,PULPIT )  values ('ТОПИ', 'Технология обогащения полезных ископаемых ','ТНВиОХТ')  insert into SUBJECT (SUBJECT, SUBJECT\_NAME, PULPIT )  values ('ПЭХ', 'Прикладная электрохимия','ХТЭПиМЭЕ')    **------Создание и заполнение таблицы GROUPS**  create table GROUPS  ( IDGROUP integer identity(1,1) constraint GROUP\_PK primary key,  FACULTY char(10) constraint GROUPS\_FACULTY\_FK foreign key  references FACULTY(FACULTY),  PROFESSION char(20) constraint GROUPS\_PROFESSION\_FK foreign key  references PROFESSION(PROFESSION),  YEAR\_FIRST smallint check (YEAR\_FIRST<=YEAR(GETDATE())),  )  insert into GROUPS (FACULTY, PROFESSION, YEAR\_FIRST )  values ('ИДиП','1-40 01 02', 2013), --1  ('ИДиП','1-40 01 02', 2012),  ('ИДиП','1-40 01 02', 2011),  ('ИДиП','1-40 01 02', 2010),  ('ИДиП','1-47 01 01', 2013),---5 гр  ('ИДиП','1-47 01 01', 2012),  ('ИДиП','1-47 01 01', 2011),  ('ИДиП','1-36 06 01', 2010),-----8 гр  ('ИДиП','1-36 06 01', 2013),  ('ИДиП','1-36 06 01', 2012),  ('ИДиП','1-36 06 01', 2011),  ('ХТиТ','1-36 01 08', 2013),---12 гр  ('ХТиТ','1-36 01 08', 2012),  ('ХТиТ','1-36 07 01', 2011),  ('ХТиТ','1-36 07 01', 2010),  ('ТОВ','1-48 01 02', 2012), ---16 гр  ('ТОВ','1-48 01 02', 2011),  ('ТОВ','1-48 01 05', 2013),  ('ТОВ','1-54 01 03', 2012),  ('ЛХФ','1-75 01 01', 2013),--20 гр  ('ЛХФ','1-75 02 01', 2012),  ('ЛХФ','1-75 02 01', 2011),  ('ЛХФ','1-89 02 02', 2012),  ('ЛХФ','1-89 02 02', 2011),  ('ТТЛП','1-36 05 01', 2013),  ('ТТЛП','1-36 05 01', 2012),  ('ТТЛП','1-46 01 01', 2012),--27 гр  ('ИЭФ','1-25 01 07', 2013),  ('ИЭФ','1-25 01 07', 2012),  ('ИЭФ','1-25 01 07', 2010),  ('ИЭФ','1-25 01 08', 2013),  ('ИЭФ','1-25 01 08', 2012) ---32 гр    ------**Создание и заполнение таблицы STUDENT**  create table STUDENT  ( IDSTUDENT integer identity(1000,1) constraint STUDENT\_PK primary key,  IDGROUP integer constraint STUDENT\_GROUP\_FK foreign key  references GROUPS(IDGROUP),  NAME nvarchar(100),  BDAY date,  STAMP timestamp,  INFO xml,  FOTO varbinary  )  insert into STUDENT (IDGROUP,NAME, BDAY)  values (2, 'Силюк Валерия Ивановна', '12.07.1994'),  (2, 'Сергель Виолетта Николаевна', '06.03.1994'),  (2, 'Добродей Ольга Анатольевна', '09.11.1994'),  (2, 'Подоляк Мария Сергеевна', '04.10.1994'),  (2, 'Никитенко Екатерина Дмитриевна', '08.01.1994'),  (3, 'Яцкевич Галина Иосифовна', '02.08.1993'),  (3, 'Осадчая Эла Васильевна', '07.12.1993'),  (3, 'Акулова Елена Геннадьевна', '02.12.1993'),  (4, 'Плешкун Милана Анатольевна', '08.03.1992'),  (4, 'Буянова Мария Александровна', '02.06.1992'),  (4, 'Харченко Елена Геннадьевна', '11.12.1992'),  (4, 'Крученок Евгений Александрович', '11.05.1992'),  (4, 'Бороховский Виталий Петрович', '09.11.1992'),  (4, 'Мацкевич Надежда Валерьевна', '01.11.1992'),  (5, 'Логинова Мария Вячеславовна', '08.07.1995'),  (5, 'Белько Наталья Николаевна', '02.11.1995'),  (5, 'Селило Екатерина Геннадьевна', '07.05.1995'),  (5, 'Дрозд Анастасия Андреевна', '04.08.1995'),  (6, 'Козловская Елена Евгеньевна', '08.11.1994'),  (6, 'Потапнин Кирилл Олегович', '02.03.1994'),  (6, 'Равковская Ольга Николаевна', '04.06.1994'),  (6, 'Ходоронок Александра Вадимовна', '09.11.1994'),  (6, 'Рамук Владислав Юрьевич', '04.07.1994'),  (7, 'Неруганенок Мария Владимировна', '03.01.1993'),  (7, 'Цыганок Анна Петровна', '12.09.1993'),  (7, 'Масилевич Оксана Игоревна', '12.06.1993'),  (7, 'Алексиевич Елизавета Викторовна','09.02.1993'),  (7, 'Ватолин Максим Андреевич', '04.07.1993'),  (8, 'Синица Валерия Андреевна', '08.01.1992'),  (8, 'Кудряшова Алина Николаевна', '12.05.1992'),  (8, 'Мигулина Елена Леонидовна', '08.11.1992'),  (8, 'Шпиленя Алексей Сергеевич', '12.03.1992'),  (9, 'Астафьев Игорь Александрович', '10.08.1995'),  (9, 'Гайтюкевич Андрей Игоревич', '02.05.1995'),  (9, 'Рученя Наталья Александровна', '08.01.1995'),  (9, 'Тарасевич Анастасия Ивановна', '11.09.1995'),  (10, 'Жоглин Николай Владимирович', '08.01.1994'),  (10, 'Санько Андрей Дмитриевич', '11.09.1994'),  (10, 'Пещур Анна Александровна', '06.04.1994'),  (10, 'Бучалис Никита Леонидович', '12.08.1994')  insert into STUDENT (IDGROUP,NAME, BDAY)  values (11, 'Лавренчук Владислав Николаевич','07.11.1993'),  (11, 'Власик Евгения Викторовна', '04.06.1993'),  (11, 'Абрамов Денис Дмитриевич', '10.12.1993'),  (11, 'Оленчик Сергей Николаевич', '04.07.1993'),  (11, 'Савинко Павел Андреевич', '08.01.1993'),  (11, 'Бакун Алексей Викторович', '02.09.1993'),  (12, 'Бань Сергей Анатольевич', '11.12.1995'),  (12, 'Сечейко Илья Александрович', '10.06.1995'),  (12, 'Кузмичева Анна Андреевна', '09.08.1995'),  (12, 'Бурко Диана Францевна', '04.07.1995'),  (12, 'Даниленко Максим Васильевич', '08.03.1995'),  (12, 'Зизюк Ольга Олеговна', '12.09.1995'),  (13, 'Шарапо Мария Владимировна', '08.10.1994'),  (13, 'Касперович Вадим Викторович', '10.02.1994'),  (13, 'Чупрыгин Арсений Александрович','11.11.1994'),  (13, 'Воеводская Ольга Леонидовна', '10.02.1994'),  (13, 'Метушевский Денис Игоревич', '12.01.1994'),  (14, 'Ловецкая Валерия Александровна','11.09.1993'),  (14, 'Дворак Антонина Николаевна', '01.12.1993'),  (14, 'Щука Татьяна Николаевна', '09.06.1993'),  (14, 'Коблинец Александра Евгеньевна','05.01.1993'),  (14, 'Фомичевская Елена Эрнестовна', '01.07.1993'),  (15, 'Бесараб Маргарита Вадимовна', '07.04.1992'),  (15, 'Бадуро Виктория Сергеевна', '10.12.1992'),  (15, 'Тарасенко Ольга Викторовна', '05.05.1992'),  (15, 'Афанасенко Ольга Владимировна', '11.01.1992'),  (15, 'Чуйкевич Ирина Дмитриевна', '04.06.1992'),  (16, 'Брель Алеся Алексеевна', '08.01.1994'),  (16, 'Кузнецова Анастасия Андреевна', '07.02.1994'),  (16, 'Томина Карина Геннадьевна', '12.06.1994'),  (16, 'Дуброва Павел Игоревич', '03.07.1994'),  (16, 'Шпаков Виктор Андреевич', '04.07.1994'),  (17, 'Шнейдер Анастасия Дмитриевна', '08.11.1993'),  (17, 'Шыгина Елена Викторовна', '02.04.1993'),  (17, 'Клюева Анна Ивановна', '03.06.1993'),  (17, 'Доморад Марина Андреевна', '05.11.1993'),  (17, 'Линчук Михаил Александрович', '03.07.1993'),  (18, 'Васильева Дарья Олеговна', '08.01.1995'),  (18, 'Щигельская Екатерина Андреевна','06.09.1995'),  (18, 'Сазонова Екатерина Дмитриевна', '08.03.1995'),  (18, 'Бакунович Алина Олеговна', '07.08.1995')  ------**Создание и заполнение таблицы PROGRESS**  create table PROGRESS  ( SUBJECT char(10) constraint PROGRESS\_SUBJECT\_FK foreign key  references SUBJECT(SUBJECT),  IDSTUDENT integer constraint PROGRESS\_IDSTUDENT\_FK foreign key  references STUDENT(IDSTUDENT),  PDATE date,  NOTE integer check (NOTE between 1 and 10)  )  insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)  values ('ОАиП', 1001, '01.10.2013',8),  ('ОАиП', 1002, '01.10.2013',7),  ('ОАиП', 1003, '01.10.2013',5),  ('ОАиП', 1005, '01.10.2013',4)  insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)  values ('СУБД', 1014, '01.12.2013',5),  ('СУБД', 1015, '01.12.2013',9),  ('СУБД', 1016, '01.12.2013',5),  ('СУБД', 1017, '01.12.2013',4)  insert into PROGRESS (SUBJECT, IDSTUDENT, PDATE, NOTE)  values ('КГ', 1018, '06.5.2013',4),  ('КГ', 1019, '06.05.2013',7),  ('КГ', 1020, '06.05.2013',7),  ('КГ', 1021, '06.05.2013',9),  ('КГ', 1022, '06.05.2013',5),  ('КГ', 1023, '06.05.2013',6) |