Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**Лабораторная работа №4**

Многотабличные SELECT-запросы

Выполнил:

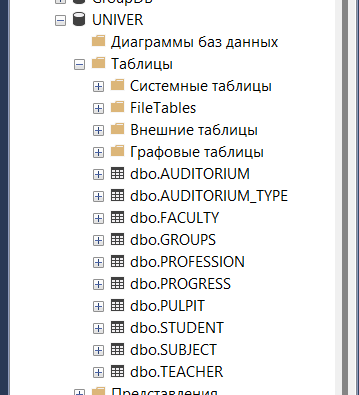
Студент 2 курса 7 группы ФИТ

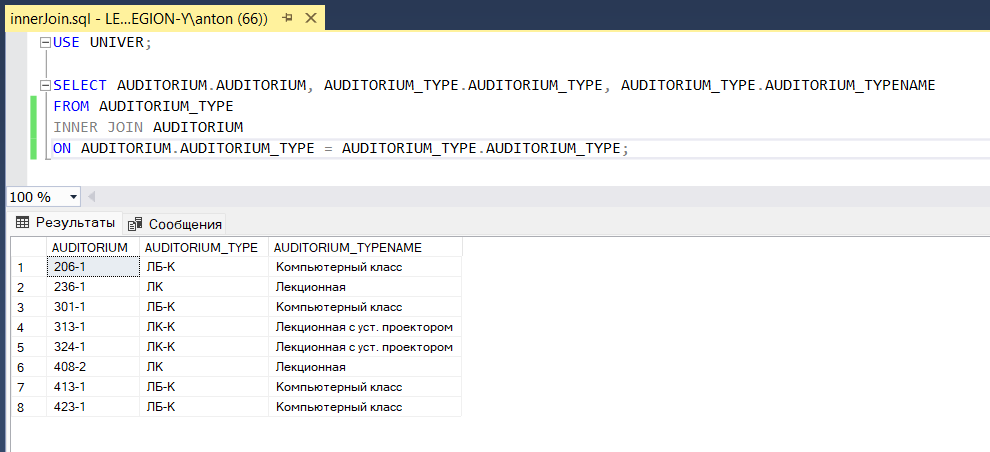
Адамович А.М.

**2022 г.**

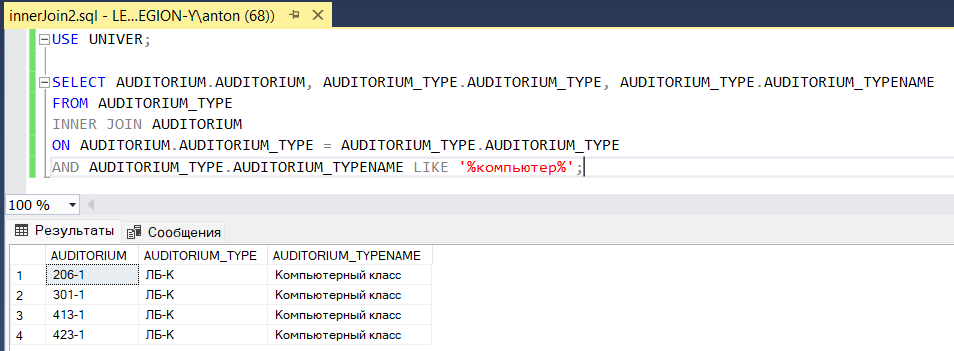
***Задание 1.*** Ознакомиться с [приложением](file:///D:\Универ%202%20курс\Университет%204%20семестр\Базы%20данных\Методички\ПрактикумБД%202022.doc#Прил1). Изучить сценарии на языке T-SQL, содержащие операторы для создания и заполнения таблиц базы данных **UNIVER**. Выполнить сценарии. Убедиться, что 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.

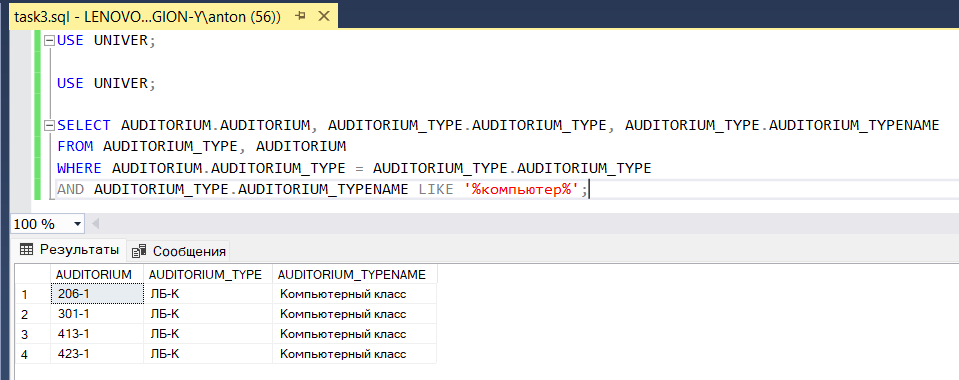




***Задание 2.*** На основе таблиц **AUDITORIUM\_TYPE** и **AUDITORIUM** сформировать перечень кодов аудиторий (столбец **AUDITORIUM.AUDITORIUM**) и соответствующих им наименований типов аудиторий (столбец **AUDITORIUM\_** **TYPE.AUDITORIUM\_TYPENAME**). При этом следует выбрать только те аудитории, в наименовании которых присутствует подстрока компьютер. Примечание: использовать соединение таблиц **INNER** **JOIN** и предикат **LIKE**.



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

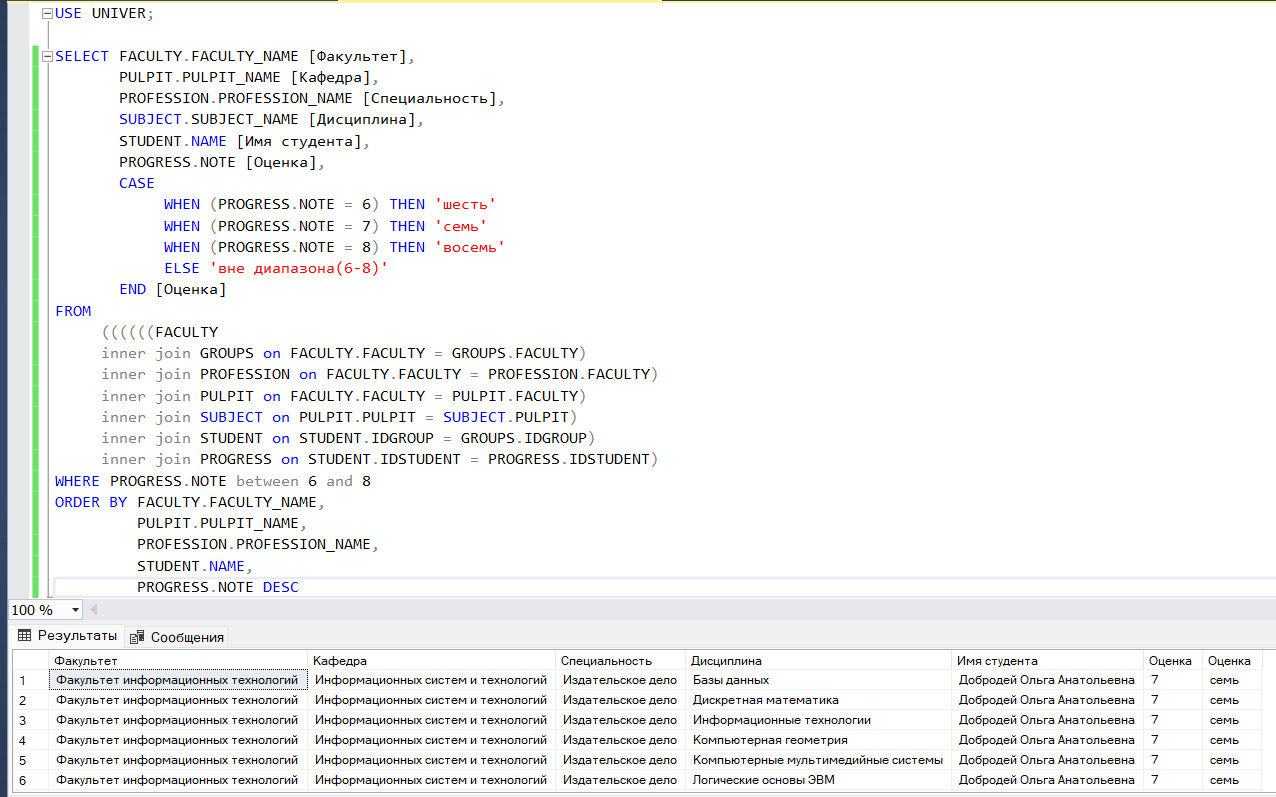


***Задание 4.*** На основе таблиц **PRORGESS**, **STUDENT**, **GROUPS**, **SUBJECT**, **PULPIT** и **FACULTY** сформировать перечень студентов, получивших экзаменационные оценки (столбец **PROGRESS.NOTE**) от 6 до 8.

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

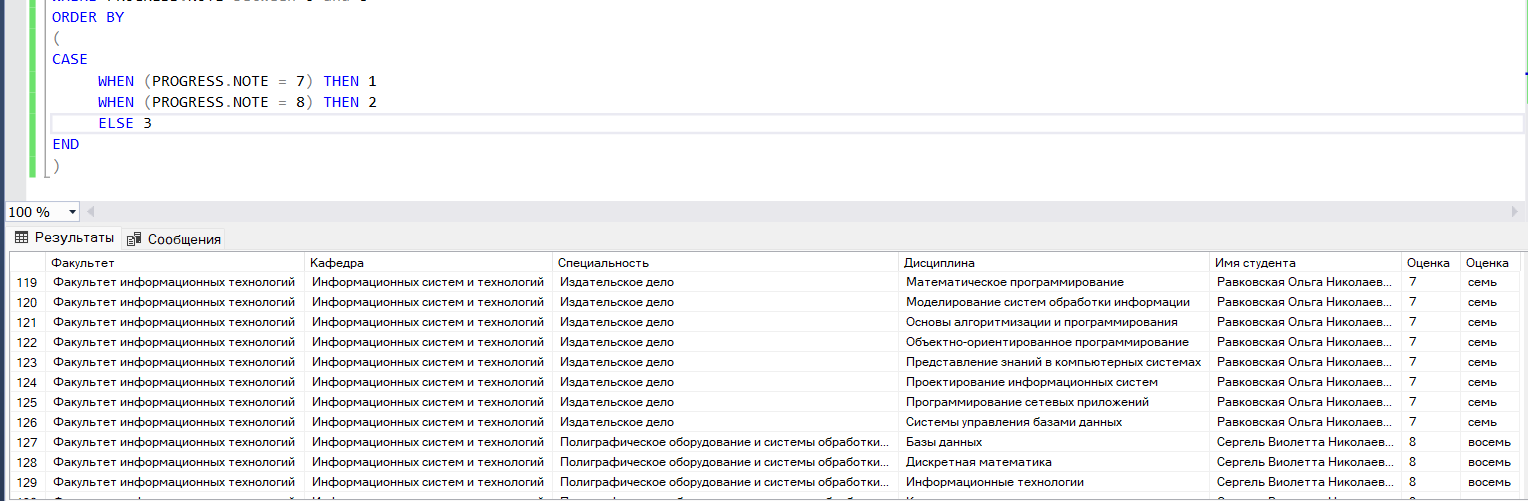
Результирующий набор отсортировать в порядке возрастания по столбцам **FACULTY.FACULTY**, **PULPIT.PULPIT**, **PROFESSION.PROFESSION**, **STUDENT. STUDENT\_NAME** и в порядке убывания по столбцу **PROGRESS.NOTE**.

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



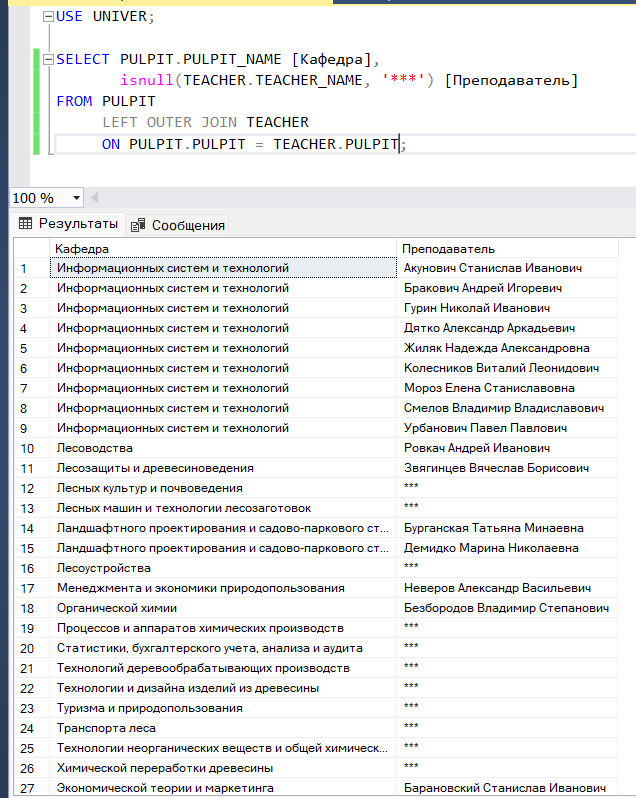
***Задание 5.*** Переписать запрос, реализующий задание 4 таким образом, чтобы в результирующем наборе сортировка по экзаменационным оценкам была следующей: сначала выводились строки с оценкой **7**, затем строки с оценкой **8** и далее строки с оценкой **6**.

Примечание: использовать выражение CASE в секции ORDER BY.



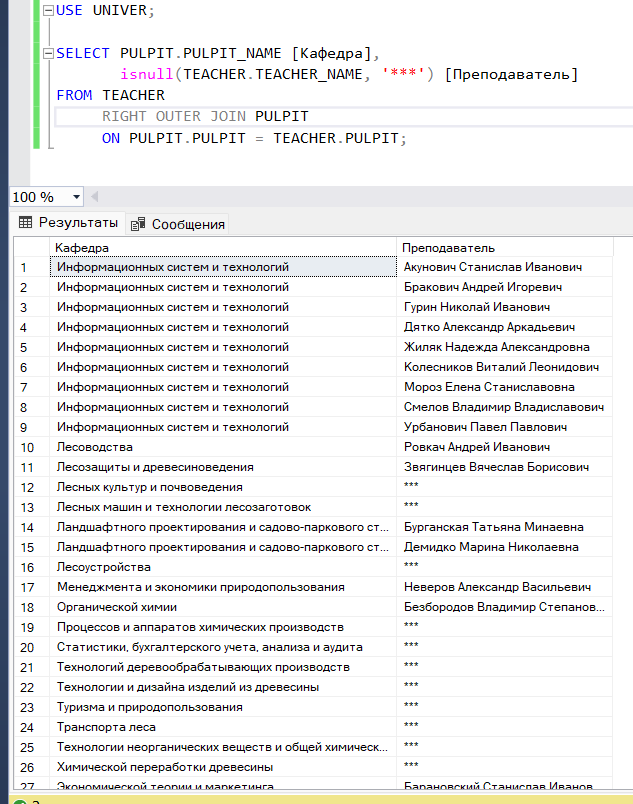
***Задание 6.*** На основе таблиц **PULPIT** и **TEACHER** получить полный перечень кафедр (столбец **PULPIT.PULPIT\_ NAME)** и преподавателей (столбец **TEACHER.TEA-CHER\_NAME**) на этих кафедрах. Результирующий набор должен содержать два столбца: **Кафедра** и **Преподаватель**. Если на кафедре нет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

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



***Задание 7.*** В запросе, реализующем пункт 6, поменять порядок таблиц в выражении LEFT OUTER JOIN.

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



***Задание 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.

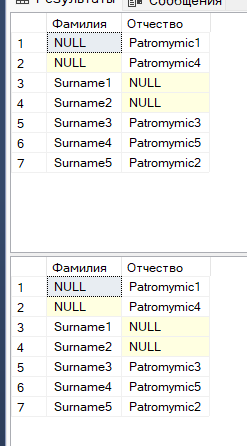
*Создать три новых запроса:*

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

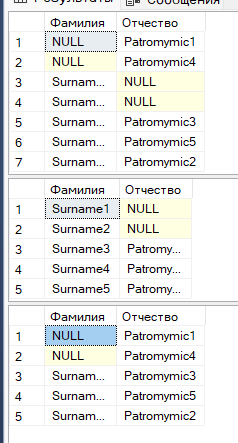
− запрос, результат которого содержит данные правой таблицы и не содержащие данные левой;

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

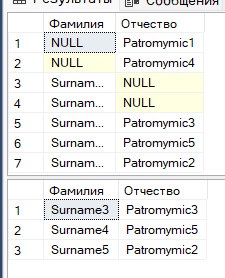
Примечание: использовать в запросах выражение IS NULL и IS NOT NULL.



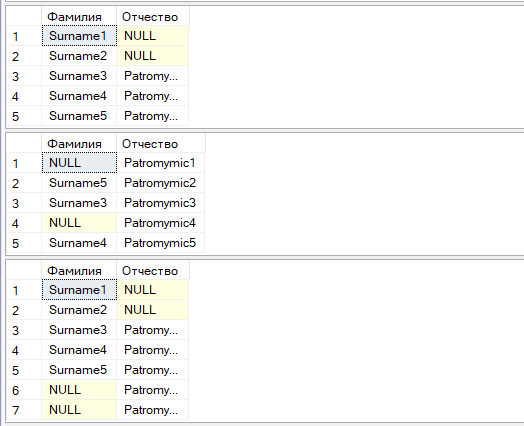
Коммутативна



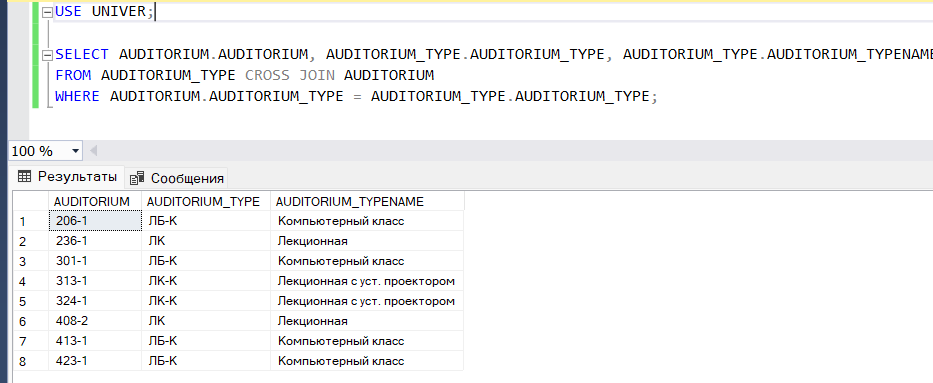
Является объединением



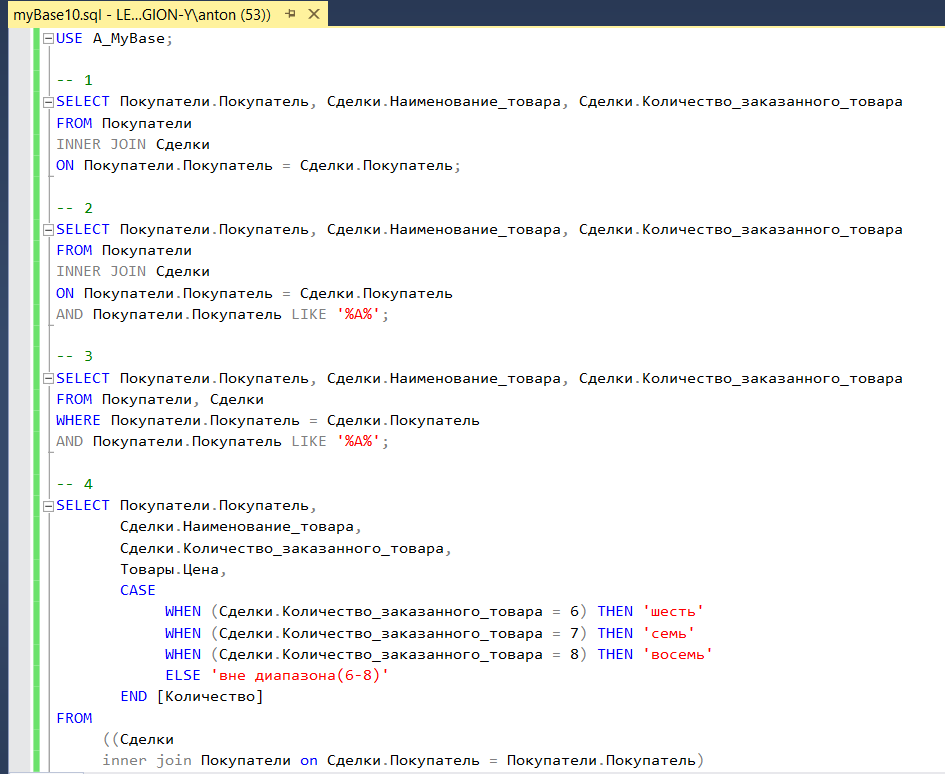
Включает соединение INNER JOIN таблиц



***Задание 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.



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



***Задание 11\*.*** Создать таблицу **TIMETABLE** (Группа, аудитория, предмет, преподаватель, день недели, пара), установить связи с другими таблицами, заполнить данными. Написать запросы на наличие свободных аудиторий на определенную пару, на определенный день недели, наличие «окон» у преподавателей и в группах.

