**Отчет по дисциплине “Базы данных”**

Лабораторная №3

Цель работы: Ознакомиться с [приложением](#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. 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**). На основе таблиц **PRORGESS**, **STUDENT**, **GROUPS**, **SUBJECT**, **PULPIT** и **FACULTY** сформировать перечень студентов, получивших экзаменационные оценки (столбец **PROGRESS.NOTE**) от 6 до 8.

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**. 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.

Ободов Павел

2 курс 2 группа

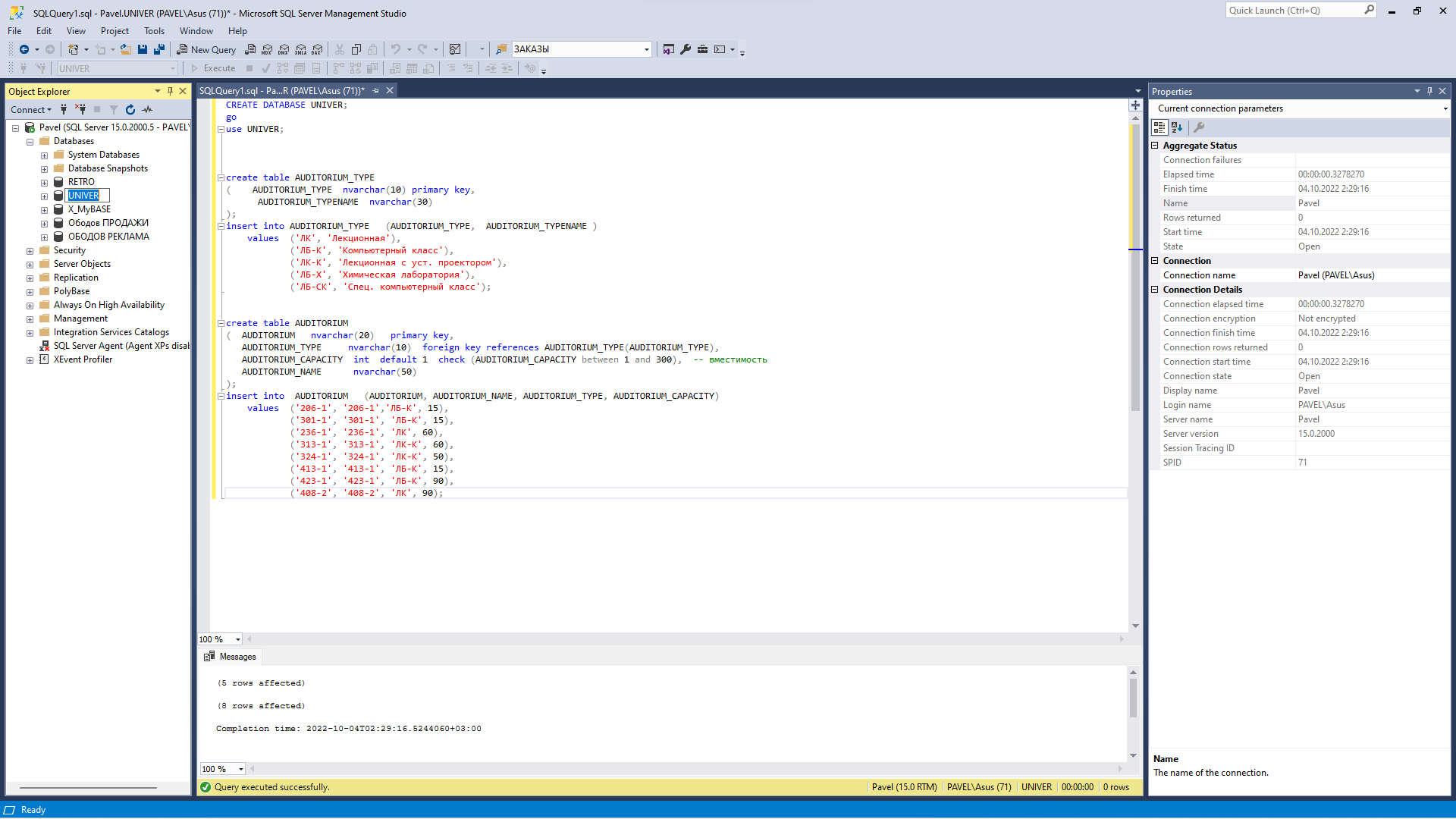


Рисунок 1

Рисунок 1 – Создание таблицы **UNIVER**. Заполнение таблиц AUDIOROOM и AUDIOROOM\_TYPE.

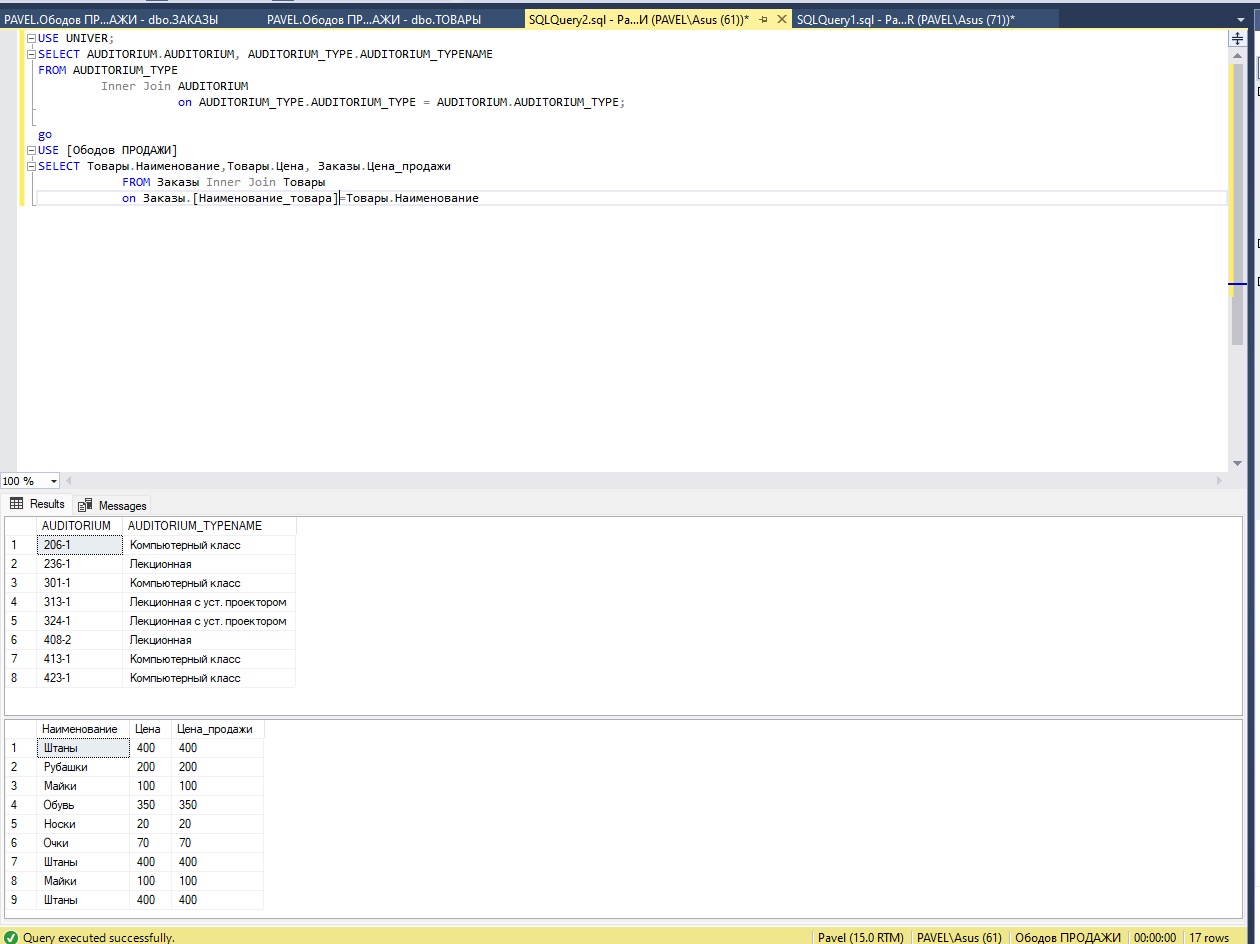


Рисунок 2

Рисунок 2 - 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. INNER JOIN - **объединение записей из двух таблиц по связующему полю, если оно содержит одинаковые значения в обеих таблицах**.

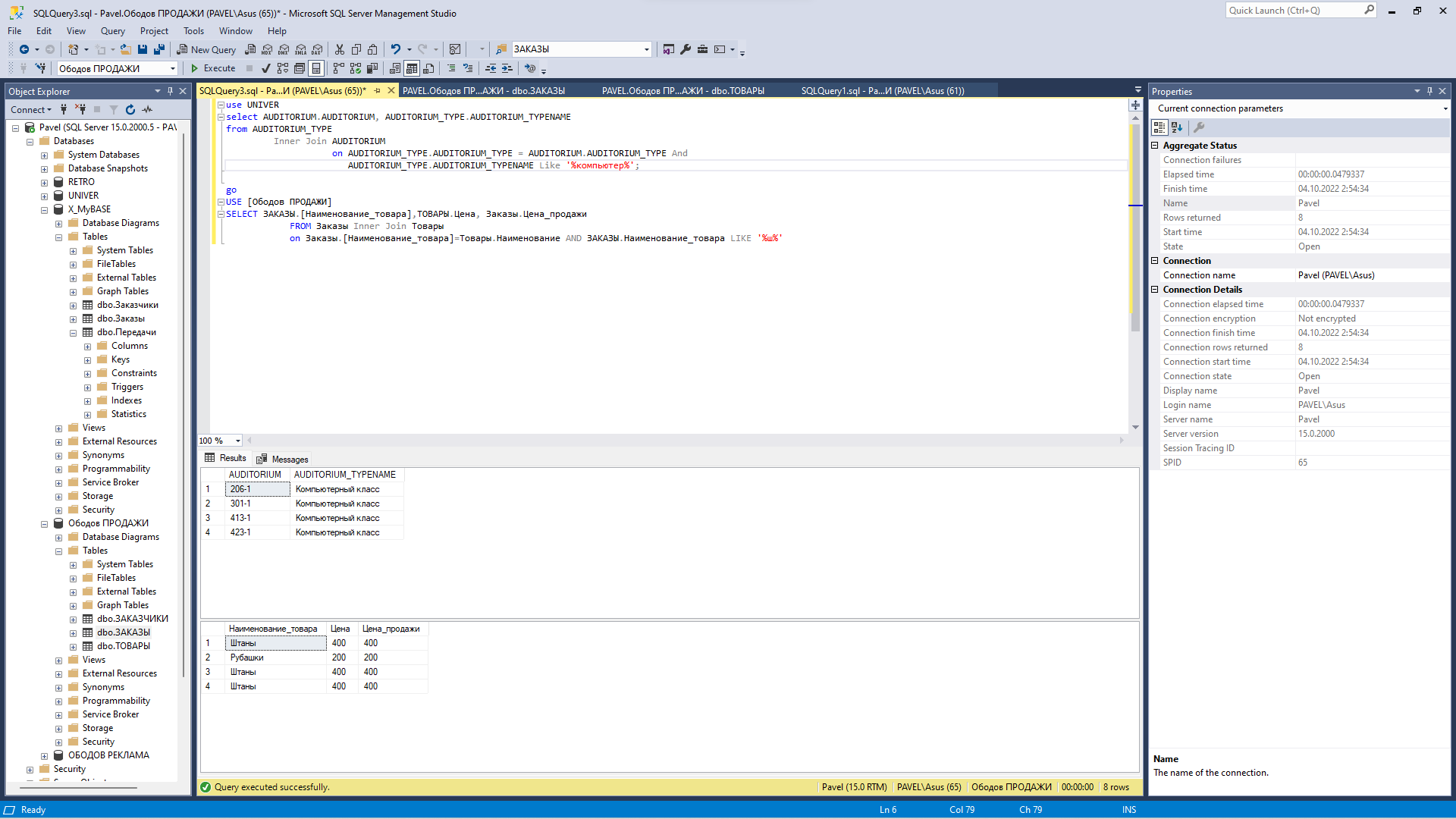


Рисунок 3

Рисунок 3 - 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**.

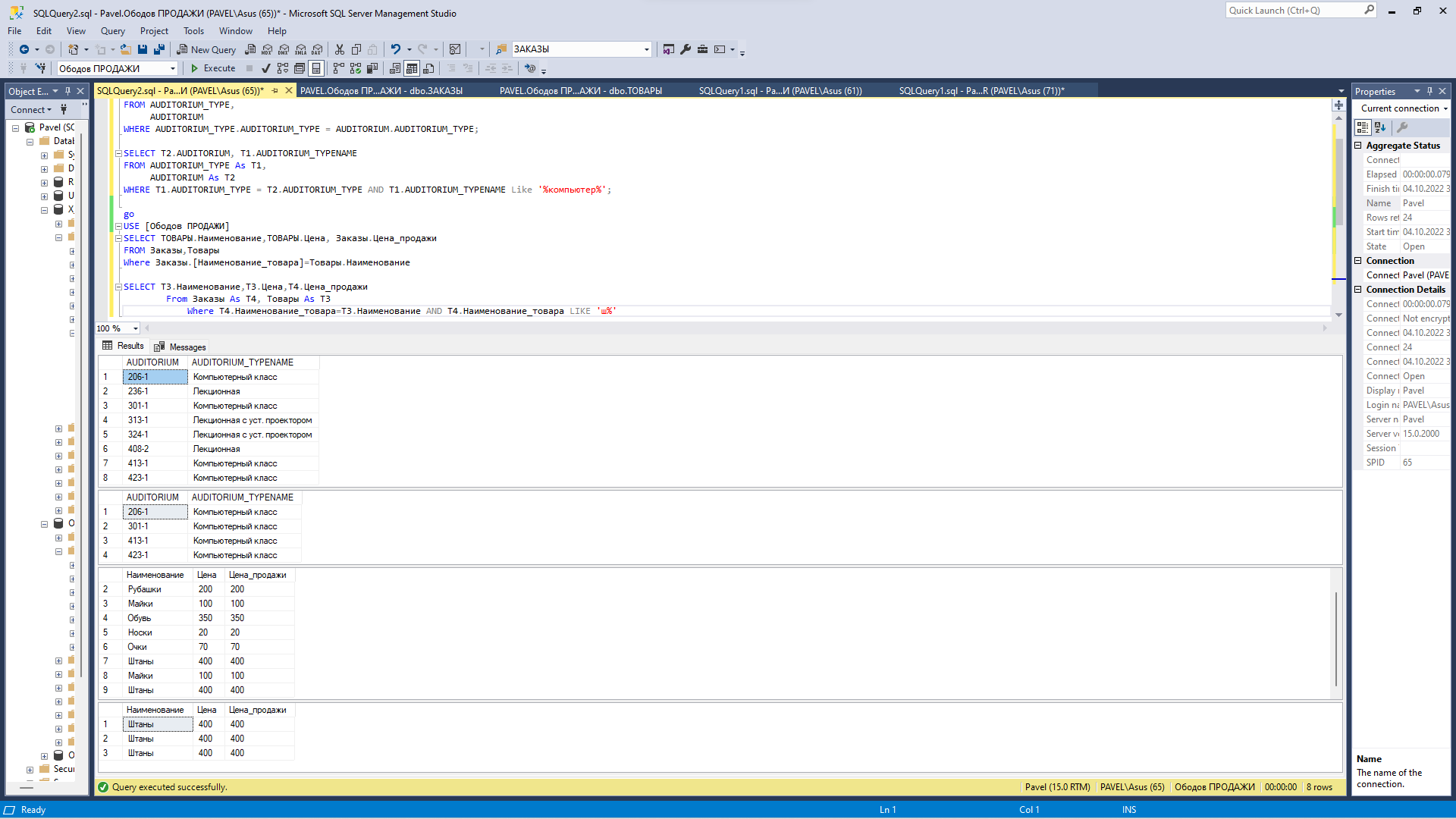


Рисунок 4

Рисунок 4 – Создание запроса используя Alias(AS). Используя псевдонимы таблиц.

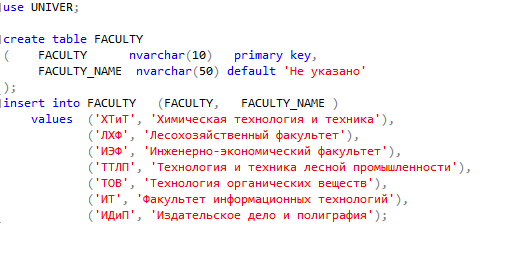


Рисунок 5

Рисунок 5 - Создание и заполнение таблицы факультетов в БД UNIVER.

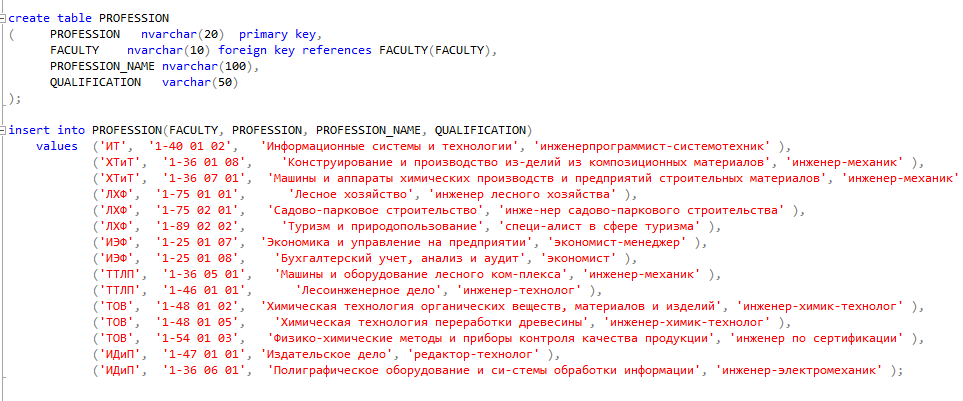


Рисунок 6

Рисунок 6 – Создание и заполнение таблицы профессий преподавателей в БД UNIVER.

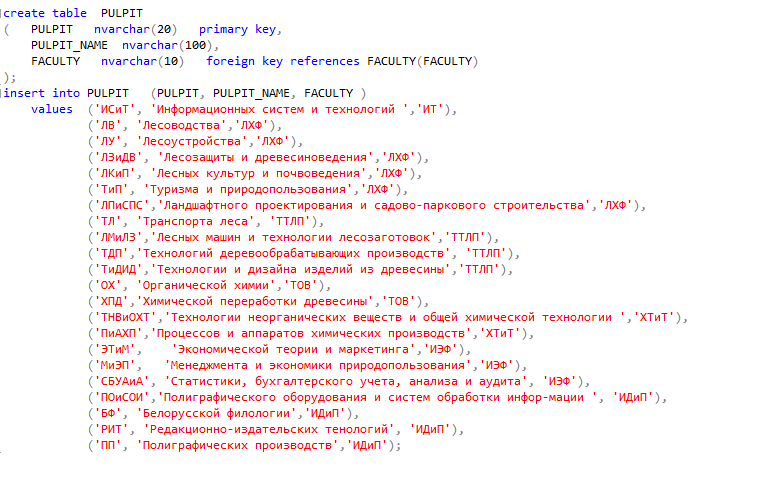


Рисунок 7



Рисунок 8

Рисунок 8 – Создание и заполнение таблицы преподаватели.



Рисунок 9

Рисунок 9 – Создание и заполнение таблицы предметов.

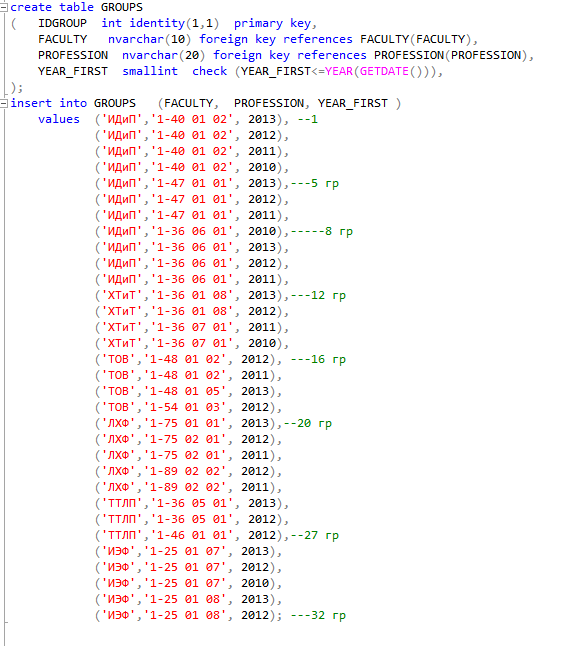


Рисунок 10

Рисунок 10 – Создание и заполнение таблицы групп студентов.

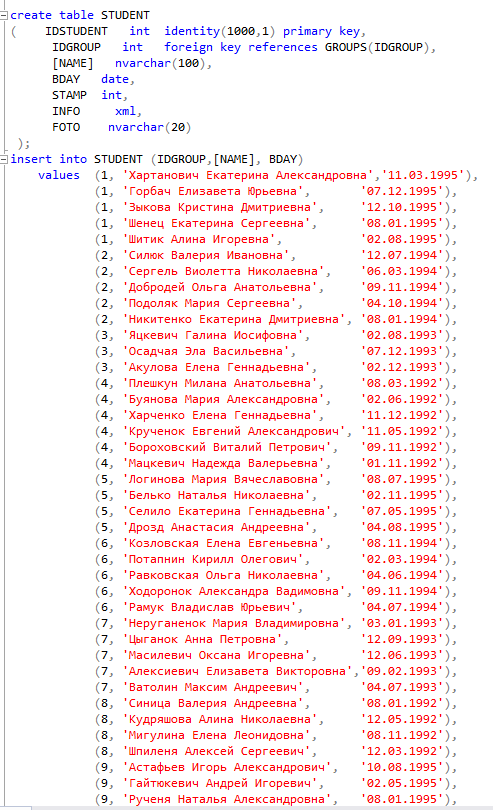


Рисунок 11

Рисунок 11 – Создание и заполнение таблицы Студенты.

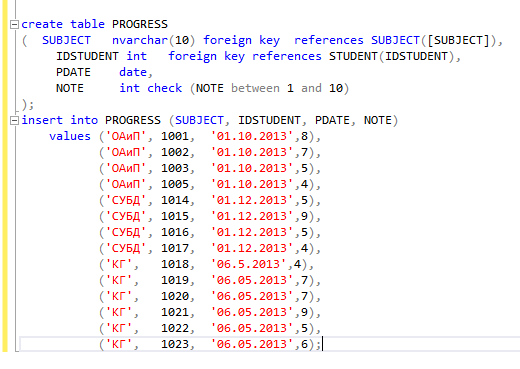


Рисунок 12

Рисунок 12 – Создание и заполнение таблицы экзаменов студентов.

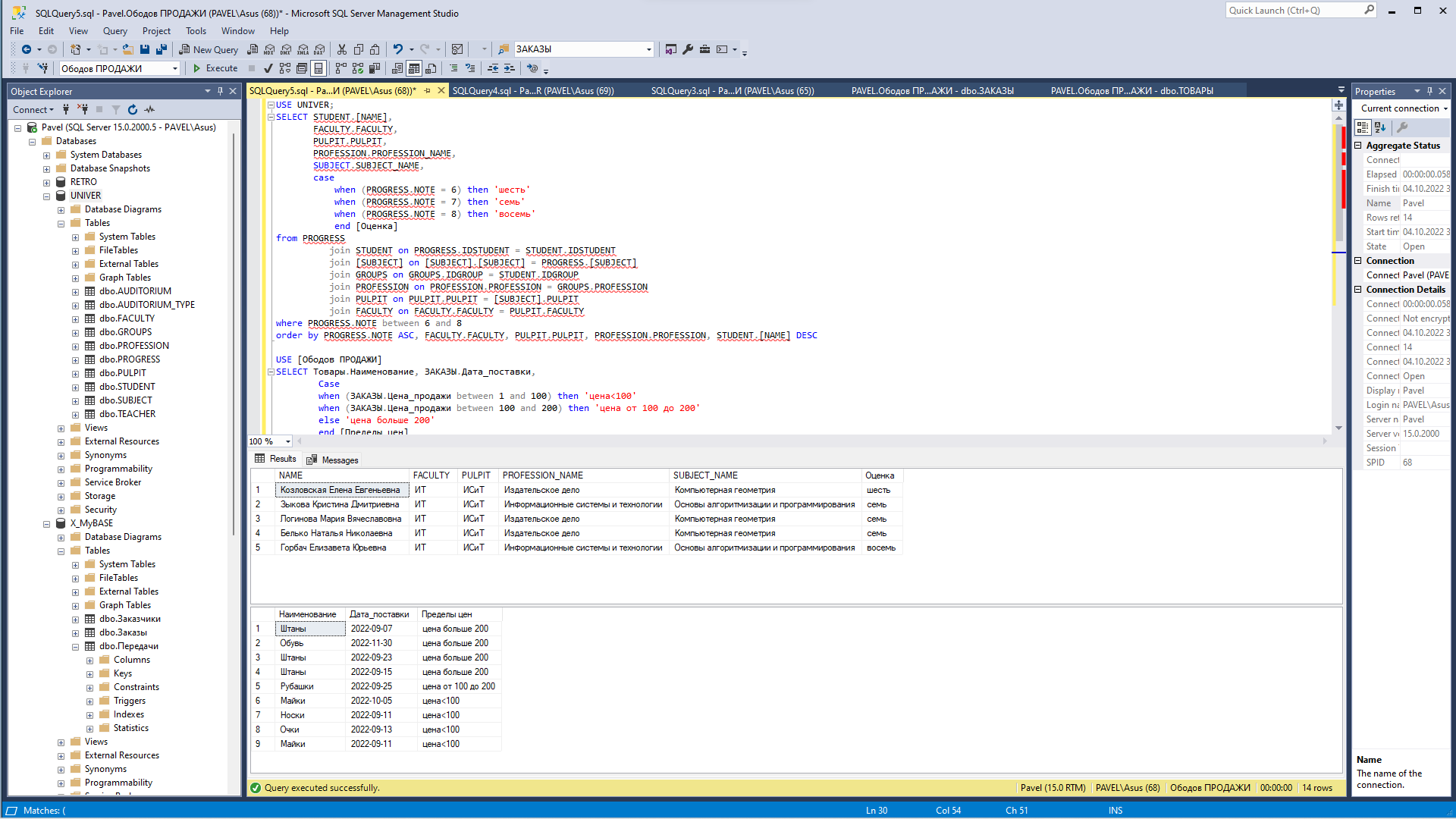


Рисунок 13

Рисунок 13 – Создание запроса по выводу ответок студентов используя выражение CASE в секции ORDER BY. Также использование JOIN для соединения столбцов таблиц в новую таблицу. Также сортировка по экзаменационным оценкам: сначала выводились строки с оценкой **6**, затем строки с оценкой **7** и далее строки с оценкой **8**.

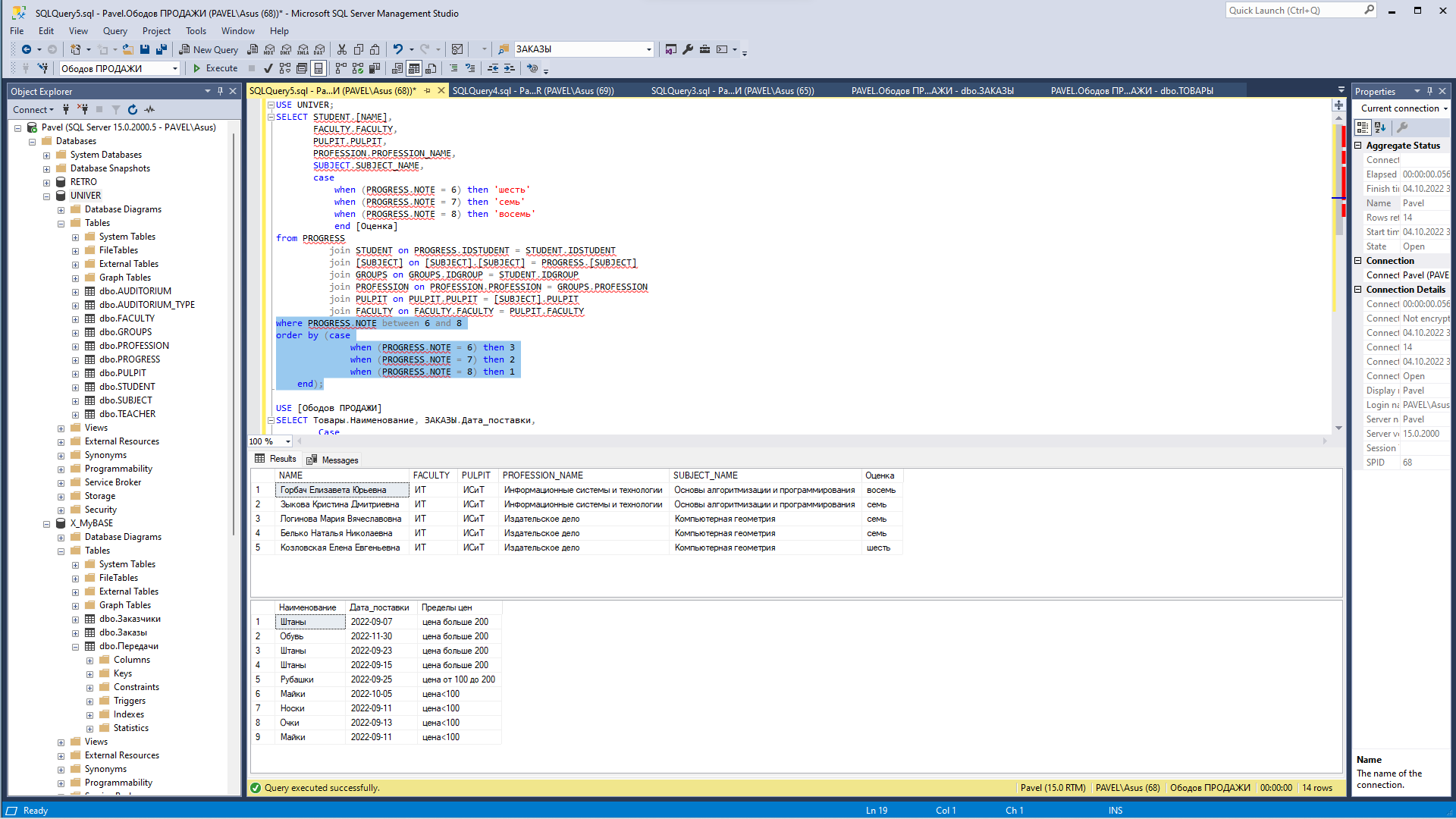


Рисунок 14

Рисунок 14 – использование предиката BETWEEN и выражения CASE.

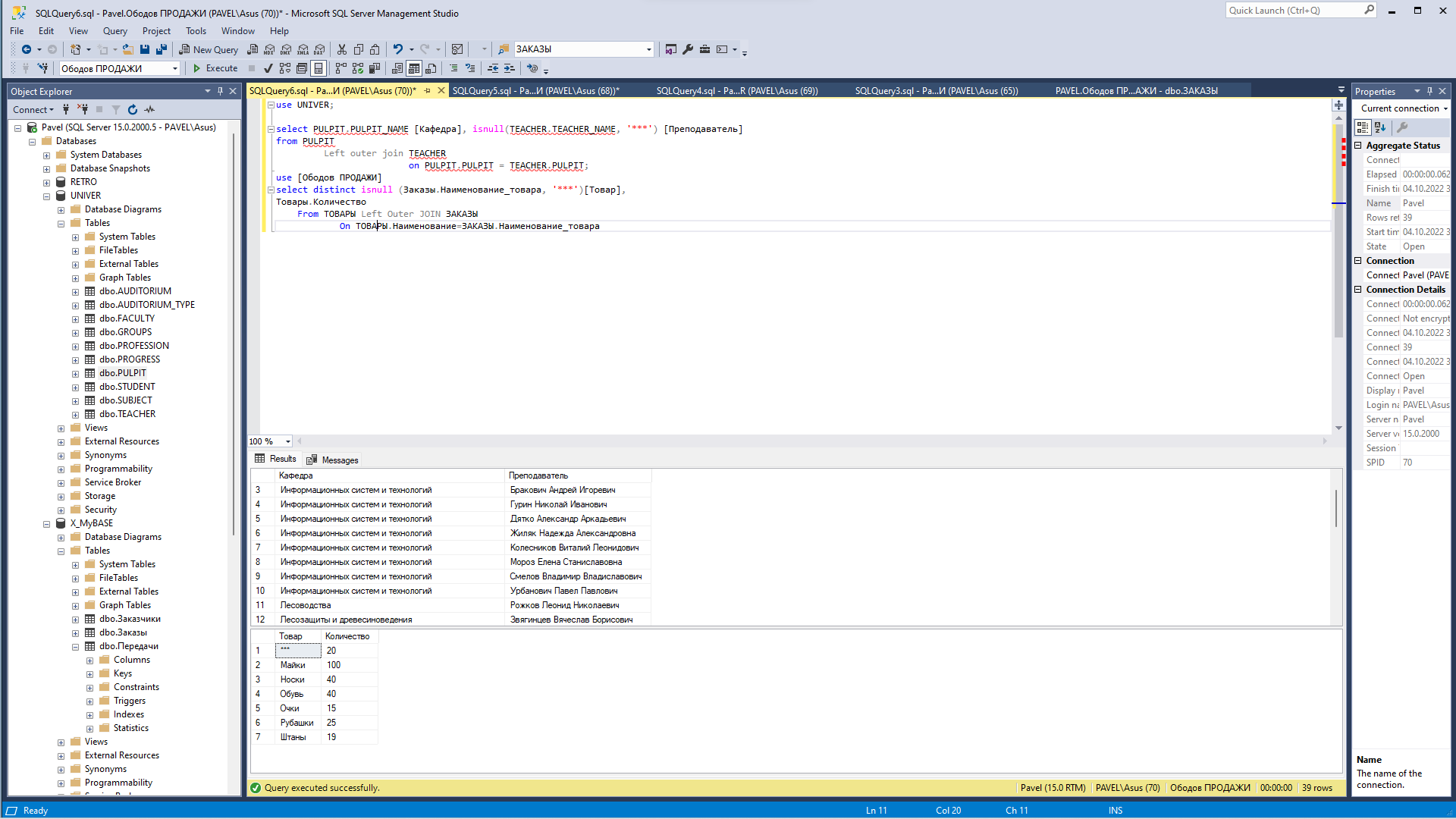


Рисунок 15

Рисунок 15 - 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**.

Рисунок 16

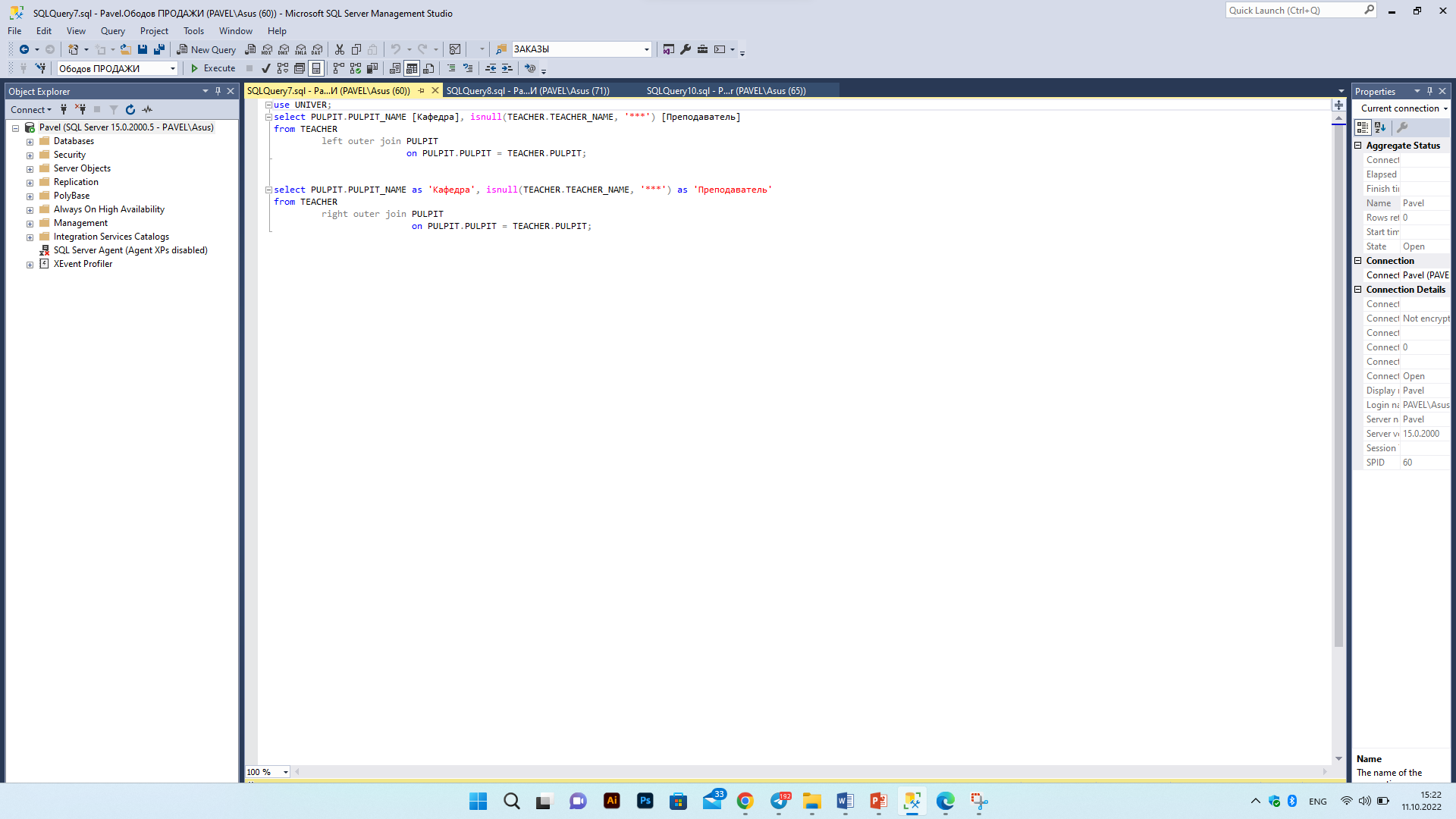


Рисунок 16 – изменение запроса таким образом, чтобы получился аналогичный результат, но применялось соединение таблиц RIGHT OUTER JOIN.

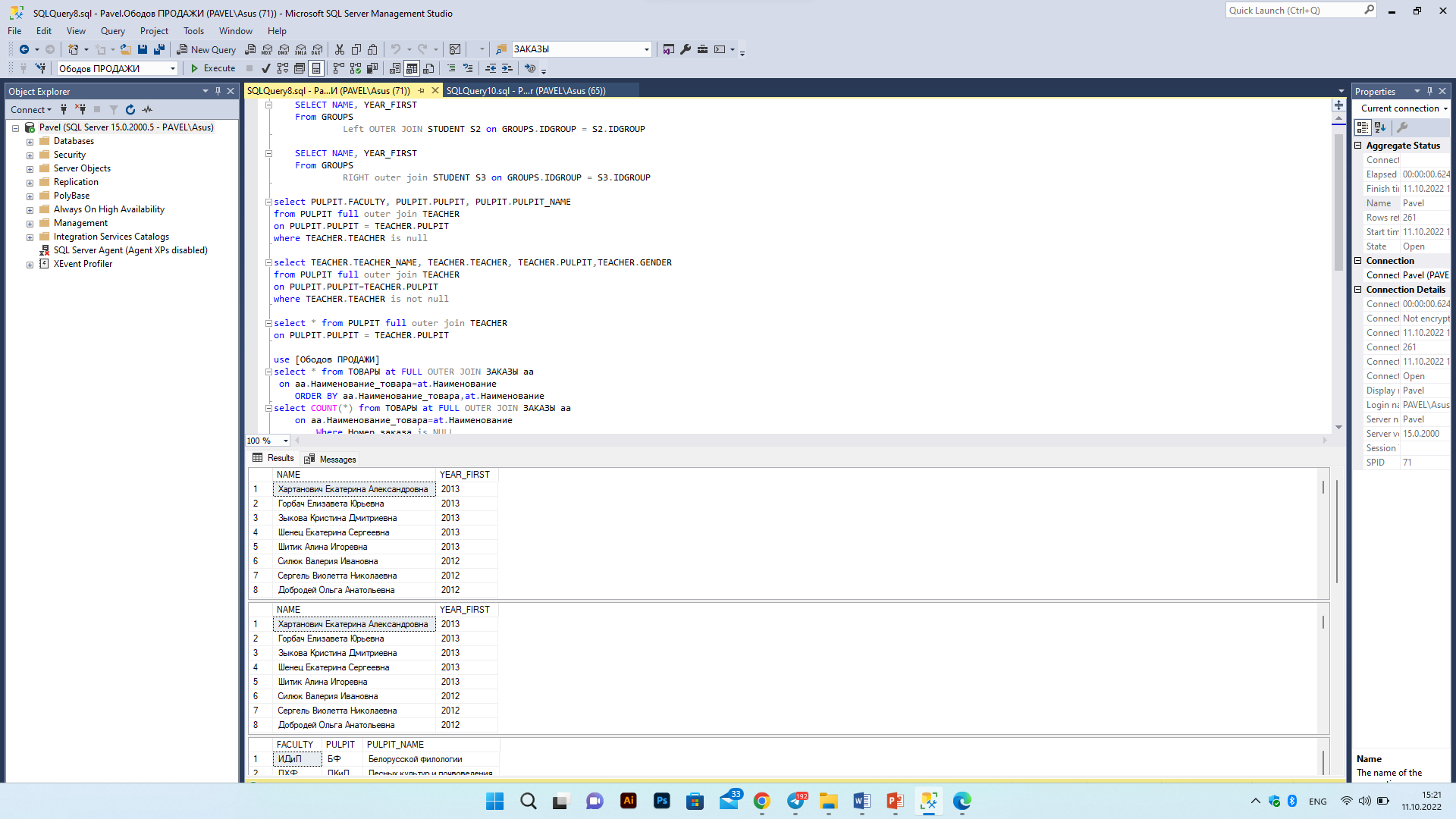


Рисунок 16

Рисунок 16 – 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.

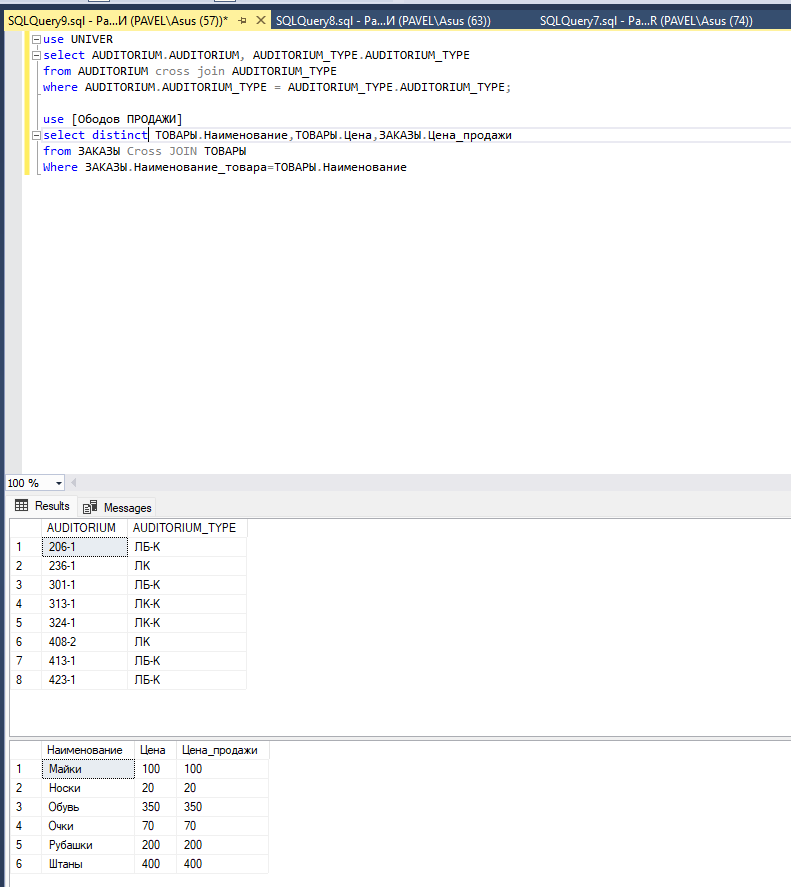


Рисунок 17

Рисунок 17 - Разработать SELECT-запрос на основе CROSS JOIN-соединения таблиц **AUDITORIUM\_TYPE** и **AUDITORIUM**, формирующего результат, аналогичный результату, полученному при выполнении запроса в задании 1.



Рисунок 18

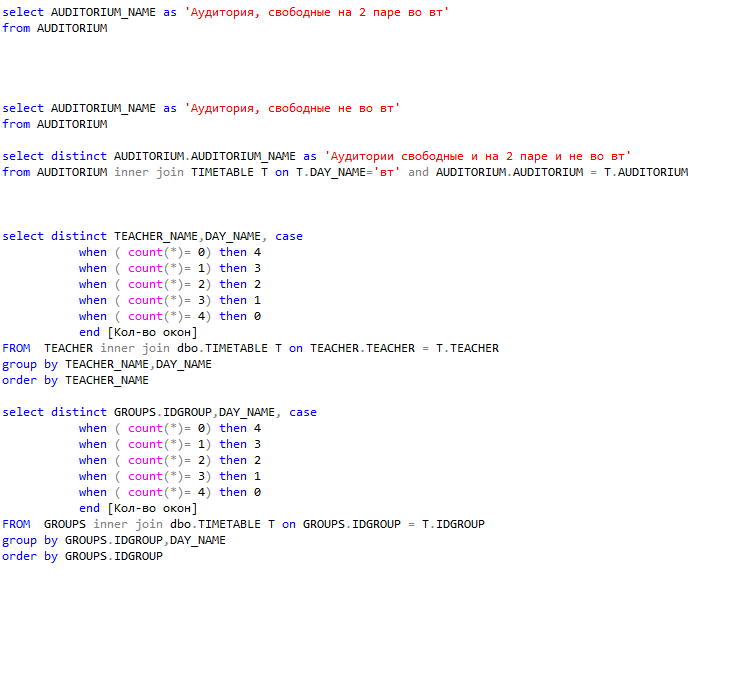


Рисунок 19

Рисунки 18 и 19 – Создание таблицы **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.

**ВЫВОД**: Я ознакомиться с приложением. Изучил сценарии на языке T-SQL, содержащие операторы для создания и заполнения таблиц базы данных **UNIVER**. (Inner Join, LIKE, CASE, ORDER BY, OUTER JOIN, RIGHT OUTER JOIN, LEFT OUTER JOIN, FULL OUTER JOIN, CROSS JOIN). Использовал псевдонимы в таблицах. Научился соединять таблицы внешним и внутренним способом. Выводить данные левой таблицы и правой.