**Базы данных**

**Отчет по лабораторной работе №13, Лешук Дмитрий ПОИБМС 7-2**

**Цель:** Изучить особенности использования разработки функций на T-SQL.

Функция – это объект БД, представляющий собой поименованный код T-SQL. Для создания, удаления и изменения функций надо использовать операторы CREATE, DROP и ALTER соответственно. Отличие функций от хранимых процедур в ограничениях, накладываемых на код функции, в форме представления результата работы, а также в способе вызова. В функции не допускается применение DDL-операторов, DML-операторов, изменяющих БД (INSERT, DELETE, UPDATE), конструкций TRY/CATCH, а также использование транзакций.

Результатом выполнения функции является возвращаемое к точке вызова значение. Если функция возвращает единственное значение (число, строка, дата, время и пр.), то она называется скалярной. Функция, возвращающая таблицу, называется табличной. В зависимости от структуры кода, различают встроенные функции и многооператорные табличные функции.

В первом задании требуется разработать скалярную функцию с именем COUNT\_STUDENTS, которая вычисляет количество студентов на факультете, код которого задается параметром с именем @faculty. Использовать внутреннее соединение таблиц FACULTY, GROUPS, STUDENT – рисунок 1.1.

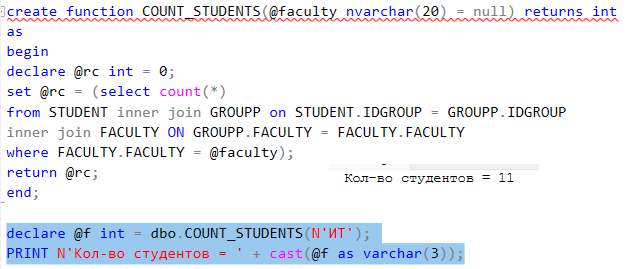


Рисунок 1.1 – Функция COUNT\_STUDENTS

Далее изменим функцию с помощью оператора ALTER так, чтобы она принимала 2 значения : факультет и код специальности – рисунок 1.2.

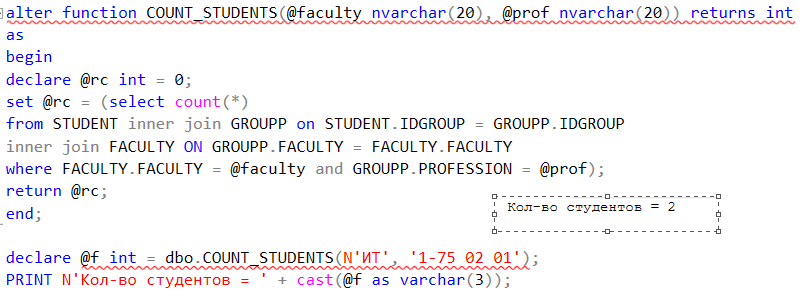


Рисунок 1.2- Модификация функции

Отличительной чертой функции является то, что мы можем использовать её в SELECT-запросе, указав определенные столбцы таблиц как аргументы функций, что и показано на рисунке 1.3. Данный запрос применяет функцию для каждой специальности каждого факультета и выводит количество студентов на данной специальности.

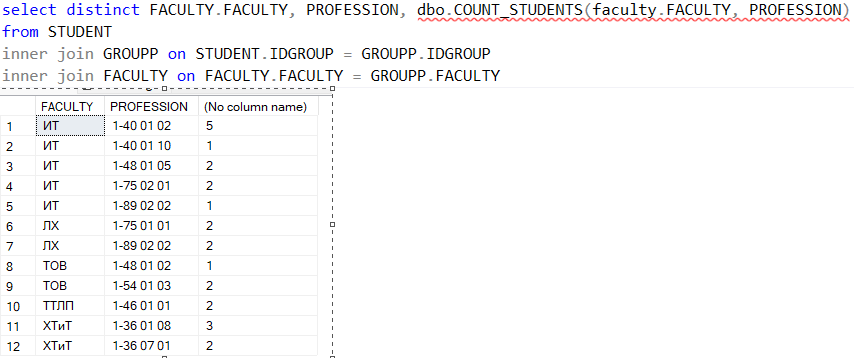


Рисунок 1.3 – Совместное использование функции и SELECT-запроса

Второе задание – создать скалярную функцию **FSUBJECTS,** принимающую один параметр – факультет, и возвращать строку со списком дисциплин на данном факультете. В дальнейшем – применить её в select-запросе – рисунок 1.4.

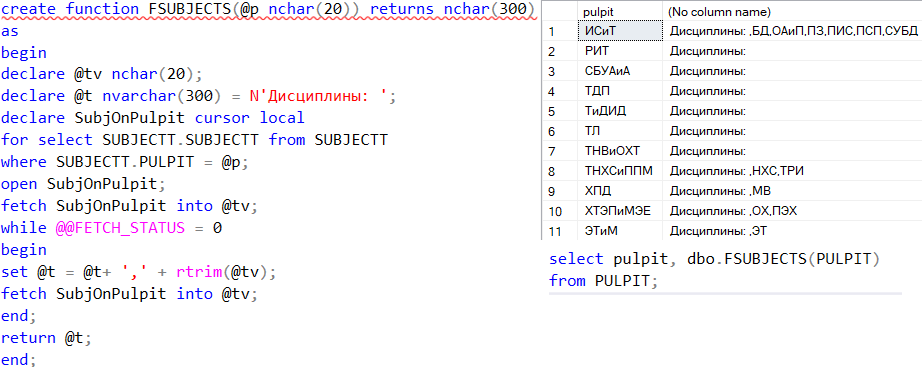


Рисунок 1.4 – функция FSUBJECT

Далее необходимо разработать табличную функцию FFACPUL, функция принимает два параметра, задающих код факультета (столбец FACULTY.FACULTY) и код кафедры (столбец PULPIT.PULPIT). Использует SELECT-запрос c левым внешним соединением между таблицами FACULTY и PULPIT.

Если оба параметра функции равны NULL, то она возвращает список всех кафедр на всех факультетах.

Если задан первый параметр (второй равен NULL), функция возвращает список всех кафедр заданного факультета.

Если задан второй параметр (первый равен NULL), функция возвращает результирующий набор, содержащий строку, соответствующую заданной кафедре.

Если заданы два параметра, функция возвращает результирующий набор, содержащий строку, соответствующую заданной кафедре на заданном факультете.

Если по заданным значениям параметров невозможно сформировать строки, функция возвращает пустой результирующий набор.

Код и результат выполнения представлен на рисунке 1.5.

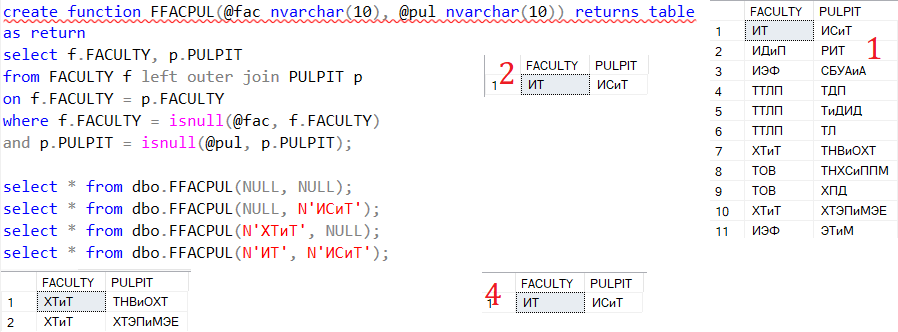


Рисунок 1.5 – использование ISNULL() в функции

Четвертое задание представляет собой следующее: Функция FTEACHER принимает один параметр, задающий код кафедры. Функция возвращает количество преподавателей на заданной параметром кафедре. Если параметр равен NULL, то возвращается общее количество преподавателей – рисунок 1.6.

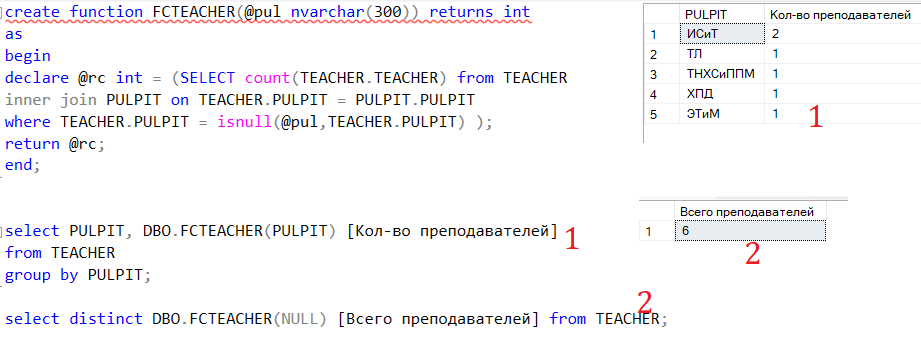


Рисунок 1.6 – функция FTEACHER

**Вывод:**

В результате данной лабораторной работы были изучены такие объекты базы данных, как функции. Изучены их особенности и синтаксис. Разработаны функции для решения типовых задач, табличные и скалярные. Опробовано использование встроенной функции ISNULL(), а также курсоров при написании пользовательских функций.