**САМОСТОЯТЕЛЬНАЯ РАБОТА ПО МОДУЛЮ №2**

**Курс: «Программирование и администрирование СУБД MS SQL Server»**

**Тема: «Хранимые процедуры, триггеры, пользовательские функции», «Таблицы и представления»**

**Цель работы:** Закрепить знания полученные студентами по созданию триггеров, хранимых процедур, пользовательских функций и представлений. Выявить основные преимущества использования триггеров в таблицах, а также пользовательских функций, хранимых процедур и многотабличных запросов в представлениях. Научить их основным подходам проектирования многофункциональных многотабличных баз данных.

**Данные:** Таблицы и запросы, содержащиеся в данной работе.

**Результат:** Многотабличная база данных “MySchool” полученная в результате выполнения всех пунктов заданий, перечисленных в данной работе.

**Время выполнения:** 1 час 20 минут

**Шаг №1: Создание базы данных**

Создайте базу данных “MySchool” средствами MS SQL Server Management Studio. Выберите ее базой данных по умолчанию для создания следующих запросов.

**Шаг №2: Создание таблиц**

Создайте следующие таблицы c следующим набором полей:

1. Students (Студенты)

* ID (int, not null)
* Name (nvarchar(50), not null)
* Surname (nvarchar(50), not null)
* MiddleName (nvarchar(50))
* DateOfBirth (datetime, not null)
* Email (nvarchar(50), not null)
* Gender (nvarchar(10), not null)
* PhoneNumber (nvarchar(20), not null)
* HomePhoneNumber (nvarchar(20), not null)

Примечание: на столбец “ID” необходимо наложить ограничение (Constraint) первичного ключа (Primary Key).

Примечание: на столбец “Gender” необходимо наложить ограничение (Constraint) диапазона значений (Check) “Male” или “Female”.

1. Teachers (Преподаватели)

* ID (int, not null)
* Name (nvarchar(50), not null)
* Surname (nvarchar(50), not null)
* MiddleName (nvarchar(50))
* Email (nvarchar(50), not null)
* Title (nvarchar(10), not null)
* PhoneNumber (nvarchar(20), not null)
* HomePhoneNumber (nvarchar(20), not null)
* DateOfHiring (datetime, not null)

Примечание: на столбец “ID” необходимо наложить ограничение (Constraint) первичного ключа (Primary Key)

Примечание: на столбец “Title” необходимо наложить ограничение (Constraint) диапазона значений (Check) “Mr”, “Mrs”, “Ms”, “Miss”, “Dr”, “Sir” или “Madam”

1. Subjects (Предметы)

* ID (int, not null)
* Name (nvarchar(50), not null)
* Comments (nvarchar(50, not null)

Примечание: на столбец “ID” необходимо наложить ограничение (Constraint) первичного ключа (Primary Key)

1. Assessments (Компоненты для предметов)

* ID (int, not null)
* Name (nvarchar(50), not null)
* ShortName (nvarchar(20), not null)
* Comments (nvarchar(50))
* Weighting (int, not null)
* ThresholdPassMark (int, not null)
* AllowedAttempts (int, not null)

Примечание: на столбец “ID” необходимо наложить ограничение (Constraint) первичного ключа (Primary Key)

Примечание: на столбец “Weighting” необходимо наложить ограничение (Constraint) диапазона значений (Check) от 0 до 100

Примечание: на столбец “ThresholdPassMark” необходимо наложить ограничение (Constraint) диапазона значений (Check) от 1 до 12

Примечание: на столбец “AllowedAttempts” необходимо наложить ограничение (Constraint) диапазона значений (Check) от 1 до 3

1. StudentOnSubjects (Студенты и предметы)

* StudentID (int, not null, foreign key)
* SubjectID (int, not null, foreign key)

Примечание: на столбцы “StudentID” и “SubjectID” необходимо наложить ограничение (Constraint) внешнего ключа (Foreign Key) со ссылками на таблицы “Students” и “Subjects” соответственно.

1. TeachersOnSubjects (Преподаватели и предметы)

* TeacherID (int, not null, foreign key)
* SubjectID (int, not null, foreign key)

Примечание: на столбцы “TeacherID” и “SubjectID” необходимо наложить ограничение (Constraint) внешнего ключа (Foreign Key) со ссылками на таблицы “Teachers” и “Subjects” соответственно.

1. AssessmentsOnSubjects (Компоненты и предметы)

* AssessmentID (int, not null, foreign key)
* SubjectID (int, not null, foreign key)

Примечание: на столбцы “AssessmentID” и “SubjectID” необходимо наложить ограничение (Constraint) внешнего ключа (Foreign Key) со ссылками на таблицы “Assessments” и “Subjects” соответственно.

1. StudentsOnAssessments (Студенты и компоненты)

* StudentID (int, not null, foreign key)
* AssessmentID (int, not null, foreign key)
* Mark (int)
* Status (nvarchar(5))
* Rank (nvarchar(3))
* Attempt (int, not null)
* Comments (nvarchar(50))

Примечание: на столбцы “StudentID” и “AssessmentID” необходимо наложить ограничение (Constraint) внешнего ключа (Foreign Key) со ссылками на таблицы “Students” и “Assessments” соответственно.

Примечание: Примечание: на столбец “Mark” необходимо наложить ограничение (Constraint) диапазона значений (Check) от 1 до 12

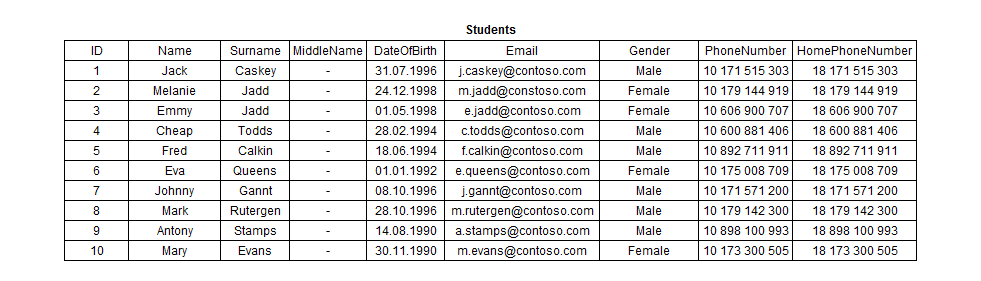
Примечание: Примечание: на столбец “Attempt” необходимо наложить ограничение (Constraint) диапазона значений (Check) от 1 до 3

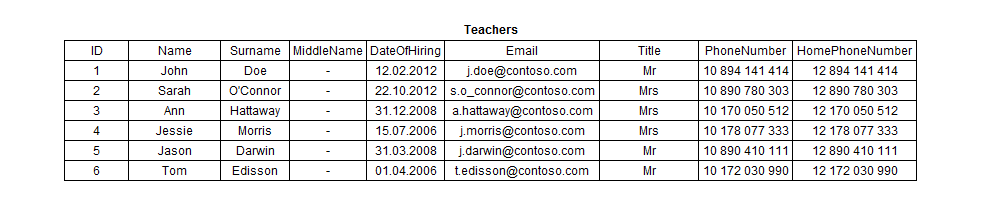
1. StudentProgression (Успеваемость студентов)

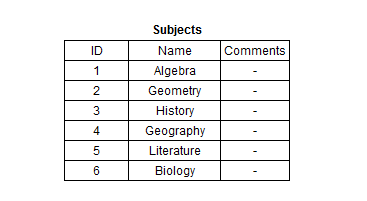
* StudentID (int, not null, foreign key)
* SubjectID (int, not null, foreign key)
* Mark (int)
* Status (nvarchar(5))
* Rank (nvarchar(3))
* Attempt (int, not null)
* Comments (nvarchar(50))

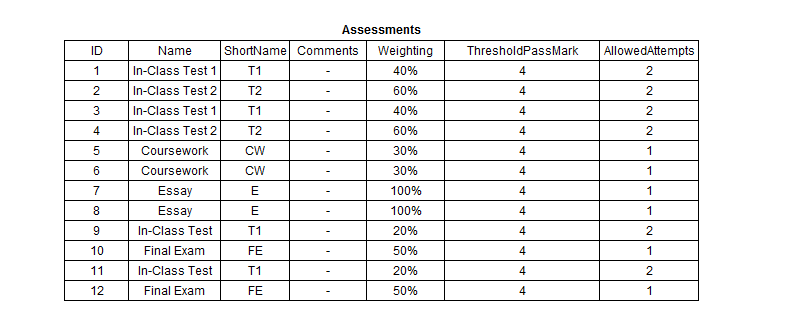
**Шаг №3. Заполнение таблиц**

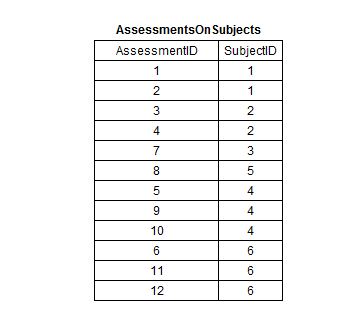
Заполните таблицы данными:

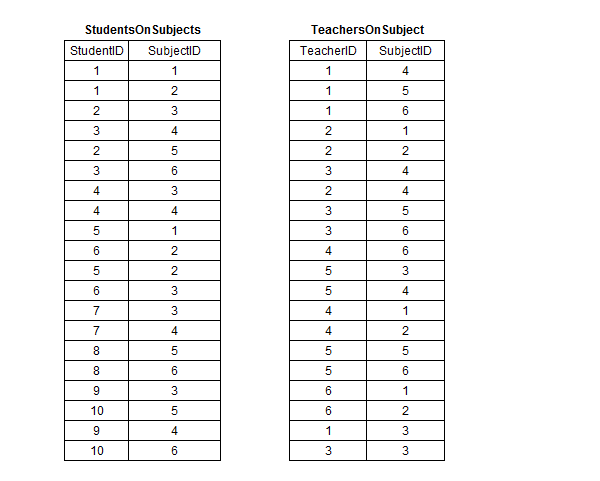


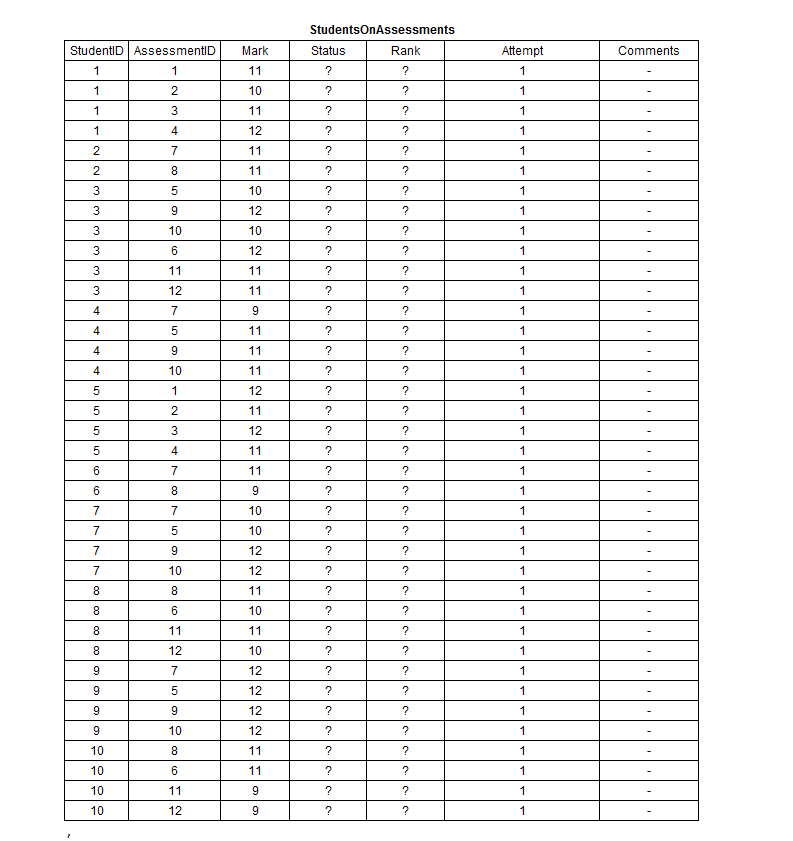












**Шаг №4: Создание пользовательских функций**

Создайте следующие пользовательские функции:

1. CalculateAge (Вычисление возраста) – данная функция должна принимать один параметр (DateOfBirth) и вычислять возраст на основании переданного значения параметра и текущей даты и возвращать результат в виде целого числа.
2. CalculateStatus (Вычисление статуса предмета) – данная функция должна принимать один параметр (Mark) и вычислять статус предмета на основании переданного значения согласно следующему условию: если оценка находится в диапазоне от 0 до 39, то статус равен “F” или “Fail”, если оценка находится в диапазоне от 40 до 100, то статус равен “P” или “Pass”, при это результат должен быть возвращен в виде строки.
3. CalculateRank (Вычисление ранга предмета) – данная функция должна принимать один параметр (Mark) и вычислять ранг предмета на основании переданного значения согласно следующему условию: если оценка находится в диапазоне от 0 до 39, то ранг равен “F/E”, если от 40 до 59 – “D”, от 60 до 74 – “C”, от 75 до 89 – “B”, от 90 до 99 – “A” и 100 – “A+”.

**Шаг №5: Создание триггеров**

Создайте следующие триггеры:

1. StudentProgressionOnInsert – данный триггер отслеживает добавление записей в таблицу “StudentProgression” и вычисляет значения для столбцов “Mark”, “Rank” и “Status” автоматически (причем значение столбца “Mark” должно быть вычислено на основе данных из таблиц “AssessmentsOnSubjects” и “StudentsOnAssessments” таким образом, чтобы оценки полученные за все компоненты и перемноженные на их вес формировали оценку за весь предмет, то есть, оценка за 1-й компонент – 12, оценка за 2-й компонент – 10, вес 1-го компонента – 20%, вес 2-го компонента – 80%, оценка за весь предмет составит 12 × 0,2 + 10 × 0,8 = 10,4 ≈ 10).
2. StudentProgressionOnUpdate – данный триггер отслеживает обновление записей в таблице “StudentProgression” и вычисляет значения для столбцов “Mark”, “Rank” и “Status” автоматически аналогично описанному выше триггеру.

**Шаг №6. Создание хранимых процедур**

Создайте следующие хранимые процедуры:

1. GetStudentsWithHighestMarks – данная хранимая процедура формирует список из тех студентов, у которых средняя оценка по всем предметам превышает 10 баллов.
2. GetStudentsWithLowestMarks – данная хранимая процедура формирует список из тех студентов, у которых средняя оценка по всем предметам ниже или равна 4 баллам.

**Шаг №7. Создание представлений**

Создайте следующие представления:

1. ShowStudentProgression – данное представление отображает успеваемость каждого студента по всем предметам (обязательно отображение столбцов “Mark”, “Rank” и “Status”)
2. ShowStudentInfo – данное представление отображает всю информацию о каждом студенте (обязательно отображение столбца “Age”, который использует функцию “CalculateAge”, а также столбца “DisplayName”, который отображает полное имя студента)
3. ShowTeachersInfo – данное представление отображает всю информацию о каждом преподавателе (здесь обязательно отображение столбца “DisplayName”, который отображает полное имя преподавателя вместе с титулом, например: “Dr. Mark Jay Adams”)

Результаты выполненной работы необходимо оформить в виде отчета, который должен содержать в себе копии сформированных запросов, а также скриншоты каждого результата выполнения этих запросов. Отсутствие этих частей в загруженном задании расценивается как невыполнение задания.

Желаем удачи!