Отчёт 11 задание 1 вариант

Регулярные выражения

i)Создайте регулярное выражение, которое выберет слова только с латинскими буквами и цифрами:

Регулярное выражение: “\b[a-zA-Z\d]+\b”

Пояснение: граница слова, латинские буквы или цифры, которые повторяются 1 и более раз, граница слова.

j)Создайте регулярное выражение, которое выберет слова, имеющие хотя бы одну латинскую букву.

Регулярное выражение: “\b\S\*[A-Za-z]+\S\*\b”

Пояснение: граница слова, символы не пробела, встречающиеся 0 или более раз, латинская буква, встречающаяся не менее 1 раза, символы не пробела, встречающиеся 0 или более раз, граница слова.

l)Напишите регулярное выражение, которое выделяет все списки параметров в круглых скобках включая сами скобки, но не чисто цифры (скобки, в которых только цифры нужно пропустить)

Регулярное выражение: “(?!\(\d+\))\(.+\)”

Пояснение: выражение «Символ открывающей скобки, любой знак >=1 раз, символ закрывающей скобки», которое не соответствует выражению «Символ открывающей скобки, цифра >=1 раз, символ закрывающей скобки».

m)Ответьте на следующие вопросы:

a) Что означает символ @ в начале строкового литерала в C#? Как будет выглядеть литерал @"(\W\*)(\w+)(\W+)(\w+)(\W+)(\w+)(\W\*)" без использования символа @?

Символ @ используется, чтобы указать, что литерал стоит интерпретировать буквально. Без использования @ будет необходимо использовать двойные слэши.

б) Что означают шаблоны: \w\*, \W\*, \w+ и \W+?

\w\* - цифра, латинская буква или знак подчёркивания 0 или более раз.  
\W\* - не символ слова 0 или более раз.  
\w+ - Цифра, латинская буква или знак подчёркивания 1 или более раз.  
\W+ - не символ слова 1 или более раз.   
в) Что означает шаблон [^<>]\*?

Выражения не включают в себя символ < или > и длина таких выражений 0 или более символов.  
г) Как сделать именованную группу?  
С помощью (?<name>выражение)

д) Где в использованных выше выражениях использовались?  
Исчислители использовались везде (+, \*)  
Классы символов использовались везде (\w, \d и т.д.)  
Обычные символы использовались в пункте l (скобки).  
Обратные ссылки нигде не использовались.  
Подстановки нигде не использовались.

Практическая часть

1. Решение практической задачи:

Использованные для преобразования regex replace:

Все строки типа GO для MySQL не нужны;  
string s = Regex.Replace(orig, "GO", "");

Все строки типа SET … для MySQL не нужны;  
string s1 = Regex.Replace(s, "SET.+\n\*", "");

Все конструкции [].[<имя>] должны быть преобразованы в <имя>;  
string s2 = Regex.Replace(s1, @"\[.+\]\.\[(.+?)\]", "$1");

Все конструкции [<имя>] должны быть преобразованы в <имя>;  
string s3 = Regex.Replace(s2, @"\[(.+?)\]", "$1");

Вместо IDENTITY(1,1) применяется AUTO\_INCREMENT;  
string s4 = Regex.Replace(s3, @"IDENTITY\(1,1\)", "AUTO\_INCREMENT");

Тип данных timestamp должен быть преобразован в timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP;  
string s5 = Regex.Replace(s4, "timestamp", "timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP");

Все конструкции типа: CONSTRAINT […] PRIMARY KEY CLUSTERED (<параметры>) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] должны быть преобразованы в PRIMARY KEY (<параметры>);  
string s6 = Regex.Replace(s5, @"CONSTRAINT .+ PRIMARY KEY CLUSTERED\n\*(\([\s\S]+?\))WITH \(PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON\) ON PRIMARY", "$1");

Все конструкции типа: PRIMARY KEY NONCLUSTERED (<параметры>) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] должны быть преобразованы в PRIMARY KEY (<параметры>);  
string s7 = Regex.Replace(s6, @"CONSTRAINT .+ PRIMARY KEY CLUSTERED\n\*(\([\s\S]+?\))WITH \(PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON \) ON PRIMARY", "$1");

Все конструкции типа: UNIQUE NONCLUSTERED (<параметры>) WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] должны быть преобразованы в UNIQUE KEY (<параметры>)\*;  
string s8 = Regex.Replace(s7, @"CONSTRAINT .+ PRIMARY KEY CLUSTERED\n\*(\([\s\S]+?\))WITH \(PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON \) ON PRIMARY", "$1");

Все остальные ON [PRIMARY] … конструкции должны быть заменены на ;.  
string s9 = Regex.Replace(s8, @"ON PRIMARY", "?");

Типы (uniqueidentifier => char(32)), (xml, ntext, nvarchar(max) => text), (image, varbinary(max) => BLOB).  
string s10 = Regex.Replace(s9, "uniqueidentifier", "char(32)");

string s11 = Regex.Replace(s10, @"xml|ntext|nvarchar\(max\)", "text");

string s12 = Regex.Replace(s11, @"image|varbinary\(max\)", "BLOB");