Соглашение о кодировании

# SQL

1. Имя таблицы должно соответствовать имени хранимых в ней сущностей во множественном числе. (Users, Albums, Properties).
2. В БД проекте перед объявлением таблиц необходимо описать на русском языке их назначение.
3. Имя поля таблицы должно быть написано в стиле CamelCase и в единственном числе UserId, ActiveOrganizationId, а не UsersId, UserID, userid, userId.

«ID», «UserID» - неверно,

«Id», «UserId» - верно.

1. Имена входных параметров хранимых процедур должны начинаться с заглавной буквы.
2. Имя поля не должно дублировать название таблицы.

[Users].[UserId] – неверно,

[Users].[Id] – верно, [Users].[PhotoId] – верно.

1. Запрещено копирование существующего uniqueidentifier для создания нового uniqueidentifier, путем изменения некоторых символов. Исключение - допустима нумерация в первых 8 символах:

00000001-63A8-4897-8ADD-F3A8CE234194

00000002-63A8-4897-8ADD-F3A8CE234194

допустимо, но

C912C116-EC**B**2-4BF9-9354-C616CFD40A1B

C912C116-EC**8**2-4BF9-9354-C616CFD40A1B

не допустимо!

Для формирования нового uniqueidentifier можно использовать Visual Studio, online-ресурсы или спец. команду SQL.

1. Имя столбцов с одинаковыми по значению данными желательно давать во всех таблицах одинаковое.

[Users].[Address] и [Organizations].[LocationAddress] – неверно,

[Users].[Address] и [Organizations].[Address] – верно.

1. Наименование констрейнтов должны быть вида:

«FK

\_[Схема таблицы с которой ссылка] *(опционально, если предполагаются несколько таблиц с одинаковыми именованиями)*

\_[Таблица с которой ссылка]

\_[Схема таблицы на которую ссылка] *(опционально, если предполагаются несколько таблиц с одинаковыми именованиям)*

\_[Таблица на которую ссылка]

\_[Доп. Информация] *(опционально, если предполагается несколько ссылок между двумя таблицами)*»

Примеры:

[FK\_Staff\_Appointments\_Staff\_Employees]

[FK\_Staff\_Appointments\_StaffUnits]

[FK\_Staff\_Appointments\_Import\_History\_Create]

[FK\_Staff\_Appointments\_Import\_History\_Update]

# Работа с репозиторием кода

1. Перед созданием комита необходимо в Visual Studio или иной среде разработки сохранить все изменения (ctrl+S).
2. Перед заливкой кода убедиться, что измененный проект собирается. Если необходимо, что бы проект довел до рабочего состояния другой сотрудник, необходимо создать специальную ветку кода. В ветку «develop» допускается заливка только корректного кода. Для названия ветки желательно использовать номер заявки ИИС.
3. Выполнение задач по изменению функционала, которые потребуют внесения изменений в структуру БД, не совместимых с текущей версией серверного кода также следует производить в специальной ветке.
4. При создании комита необходимо удостоверить, что вы осознаёте, что вы делаете и узнаете все добавляемые изменений. Нельзя заливать автоматически созданные строки и/или файлы, предназначение которых вам не ясно. Обратитесь к коллегам или непосредственному руководителю.

# 4 Работа с системой распределения задач

1. Закрывать заявку следует только после того, как она полностью выполнена и код залит в основную ветку разработки - develop.
2. Если задача требует доработки, например со стороны специалиста по вёрстке, то не следует её закрывать. Сообщите своему руководителю о необходимости перевода задачи на соответствующего специалиста.

# послесловие

Помните! Код чаще читается, чем пишется. Соблюдение стандартов кодирования способствует созданию более качественного и унифицированного кода, его лучшему восприятию, а также облегчает коллективную разработку.