



ОНЛАЙН-ОБРАЗОВАНИЕ



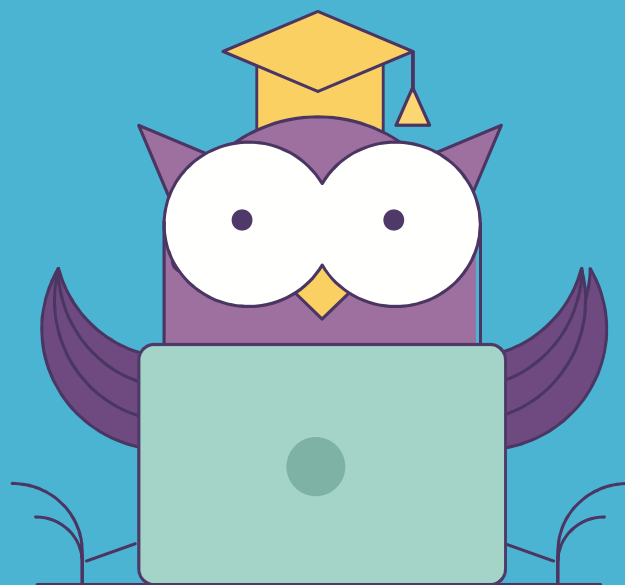
Подходы к проектированию базы данных

Курс «Разработчик MS SQL Server»

Занятие № 16



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

По окончании вебинара вы сможете:

- Объяснить когда какой подход можно использовать
- Использовать `system versioned tables`
- Использовать визуальные средства проектирования



Визуальные средства

Подходы к проектированию

System versioned tables

HierarchyId

Выводы

- Зачем делают нормализацию\денормализацию?
- Что такое функциональная зависимость? Можете привести пример?
- Зачем нужен первичный ключ в таблице?

1. Проект БД по вашей идее. Свой проект или кусочек из прода, который вы хотите пересмотреть
2. Порядка 10-15 таблиц, можно меньше или больше, это не жестко заданное требование
3. Желательно использовать SQL Server 2017, если это не прод проект с жестко заданной версией БД
4. Делаем структуру БД, с описанием, схемой
5. В конце модуля небольшая презентация, с объяснением интересных частей проекта и обсуждением принятых решений и вариантов

1. Локализация – язык\время
2. Масштабирование. Насколько быстро будет расти приложение, для которого вы делаете проект?
3. Безопасность и юридические нормы
4. Вероятность миграции в облако, если вы On-Premise
5. Политики по хранению данных – будете ли вы что-то удалять или предпочтете архивирование?

1. Для чего вы создаете БД?
2. Какие основные функции будут в вашем приложении?
3. Если у вас есть какая-то БД, какие основные проблемы с ней есть?
4. Какие основные таблицы вы будете создавать?

1. Lucidchart
2. SQL Server Database Modeler
3. SAP PowerDesigner
4. ErWin data modeler
5. DbDesigner.net

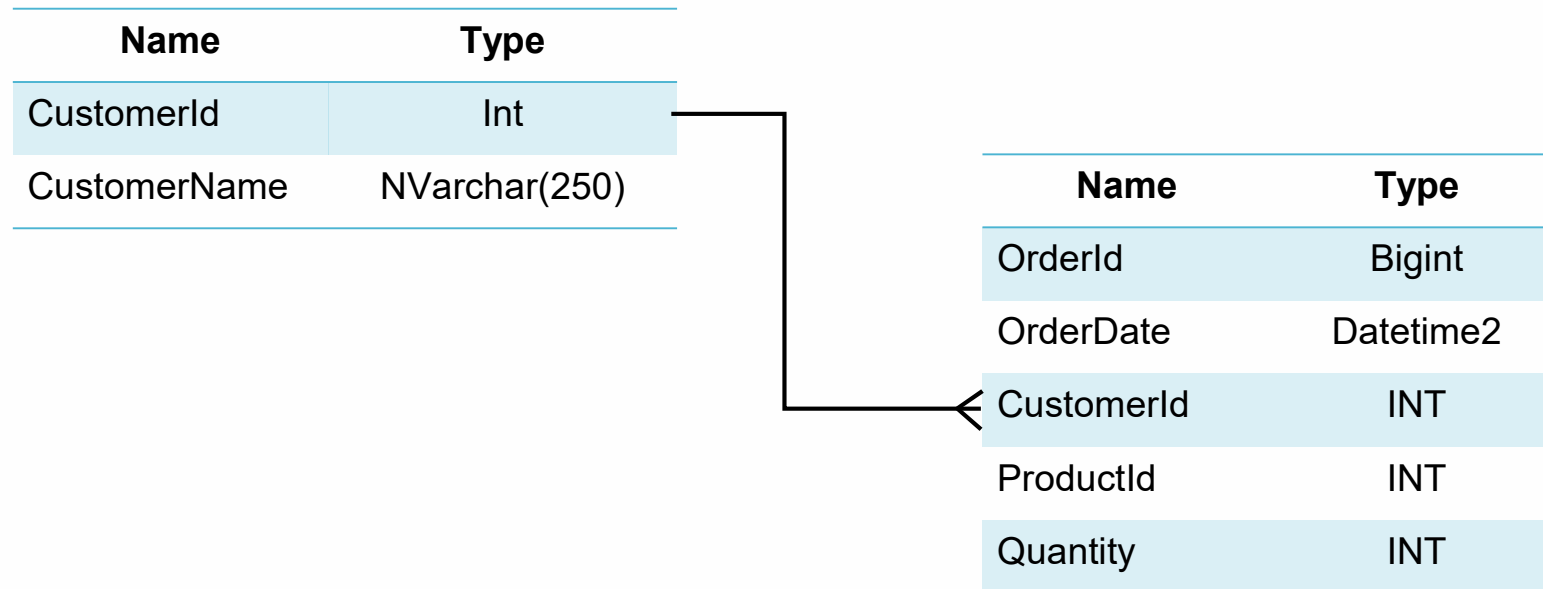
1. Int, Tinyint, Bigint
2. Date, Time, Datetime2, Datetimeoffset
3. Деньги – только заданная точность Money, Decimal(19,4) или Int
4. Float для научных данных
5. Varchar vs Char или **Nvarchar** и **Nchar** для юникода

Collation для особенно для мультязычных проектов!
SQL or Windows?

- CI – Case Insensitive
- AS – Accent Sensitive
- KS – Kana-sensitive иероглифы Hiragana и Katakana
- WS – Width Sensitive
- VSS – Variation-selector-sensitive

Example: SQL_Latin1_General_CP1_CI_AS

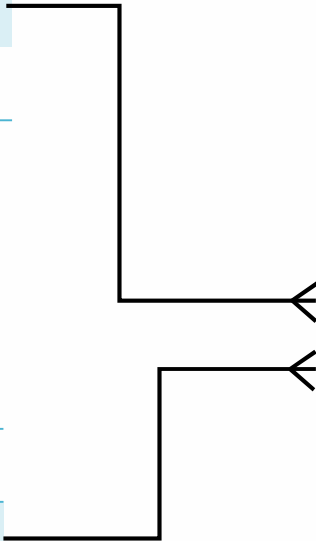
- **Prefix:** SQL_ – SQL Collation
- **Sort Rules:** Latin1_General – English
- **Code Page:** CP1 – Code Page 1252
- **Case Sensitivity:** Insensitive
- **Accent Sensitivity:** Sensitive



Name	Type
CustomerId	Int
CustomerName	NVarchar(250)

Name	Type
SubscriptionId	INT
SubscriptionName	Nvarchar(100)

Name	Type
SubscriptionCustomerId	Bigint
CustomerId	INT
SubscriptionId	INT



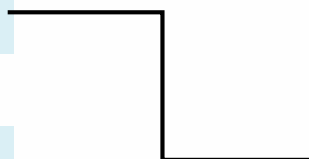
У вас есть широкая (много полей) таблица.
В этой таблице чаще всего используются 7 из 40 полей.
Остальные редко читаются и изменяются.

Вы бы хотели иметь эту таблицу поуже, чтобы она занимала меньше места в кэше. Что можно сделать?

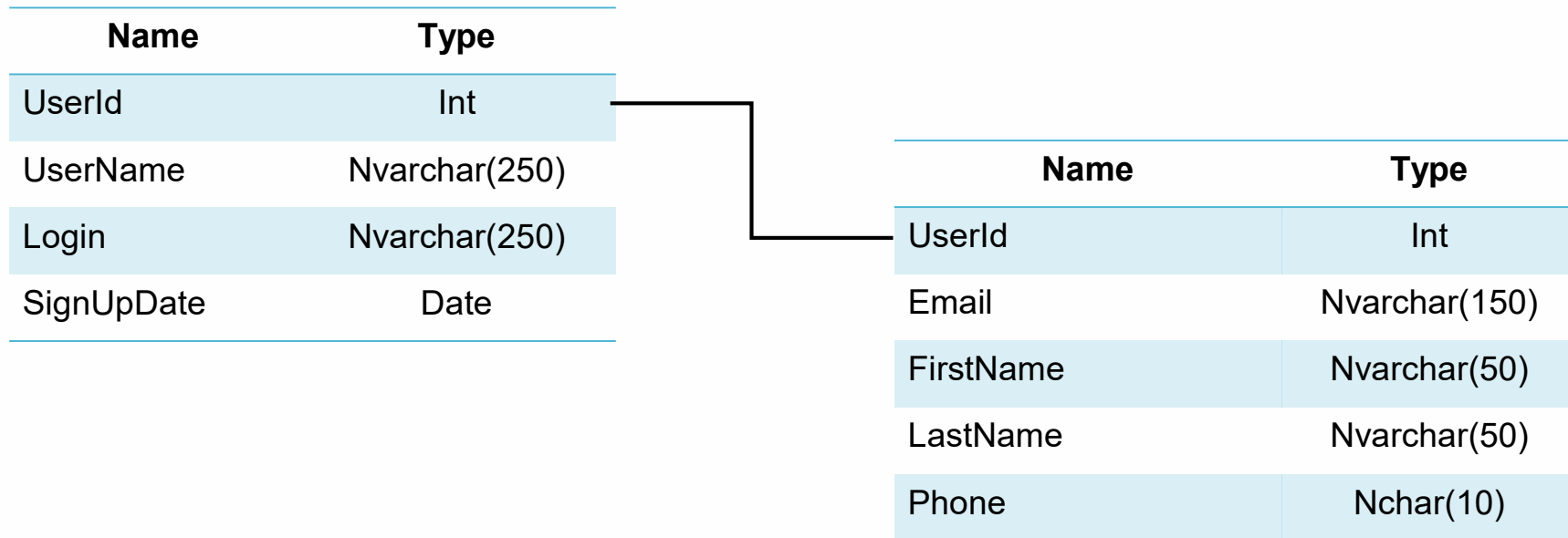
Разбить таблицу на 2 с отношением 1 к 1

Name	Type
UserId	Int
UserName	Nvarchar(250)
LastLoginDate	Date
LoginFailCount	Tinyint

Name	Type
UserId	Int
DateEmailConfirmed	Datetime2
FirstName	Nvarchar(50)
LastName	Nvarchar(50)
AccountType	Tinyint



Для поддержки требований по хранению персональных данных их лучше вынести в отдельную таблицу



Таблицы с часто меняющейся структурой. У вас есть таблица, например банкомат и его метрики работы и оборудования. Как бы вы сделали структуру?

Вариант 1. Для неленивых AtmStatus все в поля

Name	Type
AtmId	Bigint
AtmLocation	Int
CashType	Int
CashStatus	Int
CashErrorCode	NVARCHAR(50)
CashErrorMessage	NVARCHAR(500)
ModemType	Int
ModemStatus	Int
PrinterType	Int
PrinterStatus	Int

Плюсы

Удобно для выборки

Минусы

Каждый раз добавлять новые поля

Вариант 2. AtmStatus таблица с параметрами

Name	Type
ParamId	Int
ParamDescription	NVARCHAR(500)

Name	Type
AtmId	Bigint
ParamId	Int
ParamValue	NVARCHAR(500)

Name	Type
AtmId	Bigint
ParamId	Int
ParamValueInt	Int
ParamValueText	NVARCHAR(500)
ParamValueDecimal	Decimal(19,4)
ParamValueDatetime	Datetime2

Плюсы

Не нужно менять структуру таблицы

Минусы

Немного усложняются выборки

Нужно заводить новые параметры в таблице-описания

Вариант 3. В xml или json

Name	Type
AtmId	Bigint
AtmInfo	Json

Плюсы

Не нужно менять структуру
Вообще ничего не нужно делать при
добавлении новых данных
Удобство выборки

Минусы

Уровень свободы структуры

Вариант 4. Oldschool

Антипаттерн

НЕ Храните данные в строке через разделитель

Name	Type
AtmId	Bigint
AtmInfo	Nvarchar(MAX)

Плюсы

Не нужно менять структуру
Вообще ничего не нужно делать при добавлении новых данных

Минусы

Уровень свободы структуры
Уровень ошибок в данных
Неудобно делать выборки на SQL

Таблицы для ведения истории (temporal tables) по изменению данных в таблице.

Начиная с SQL Server 2016

- 2 таблицы – основная и архивная с одинаковыми полями
- Обязательно 2 поля SysStartTime и SysEndTime

Плюшки в виде выборок из архивной таблицы.

Вариант 1.

Рекурсивные выборки
в CTE/функциях

Employee	Manager
Пупкин	NULL
Иванов	Пупкин
Петров	Иванов
Сидоров	Иванов

Вариант 2.

Таблица развертки
(bridge)

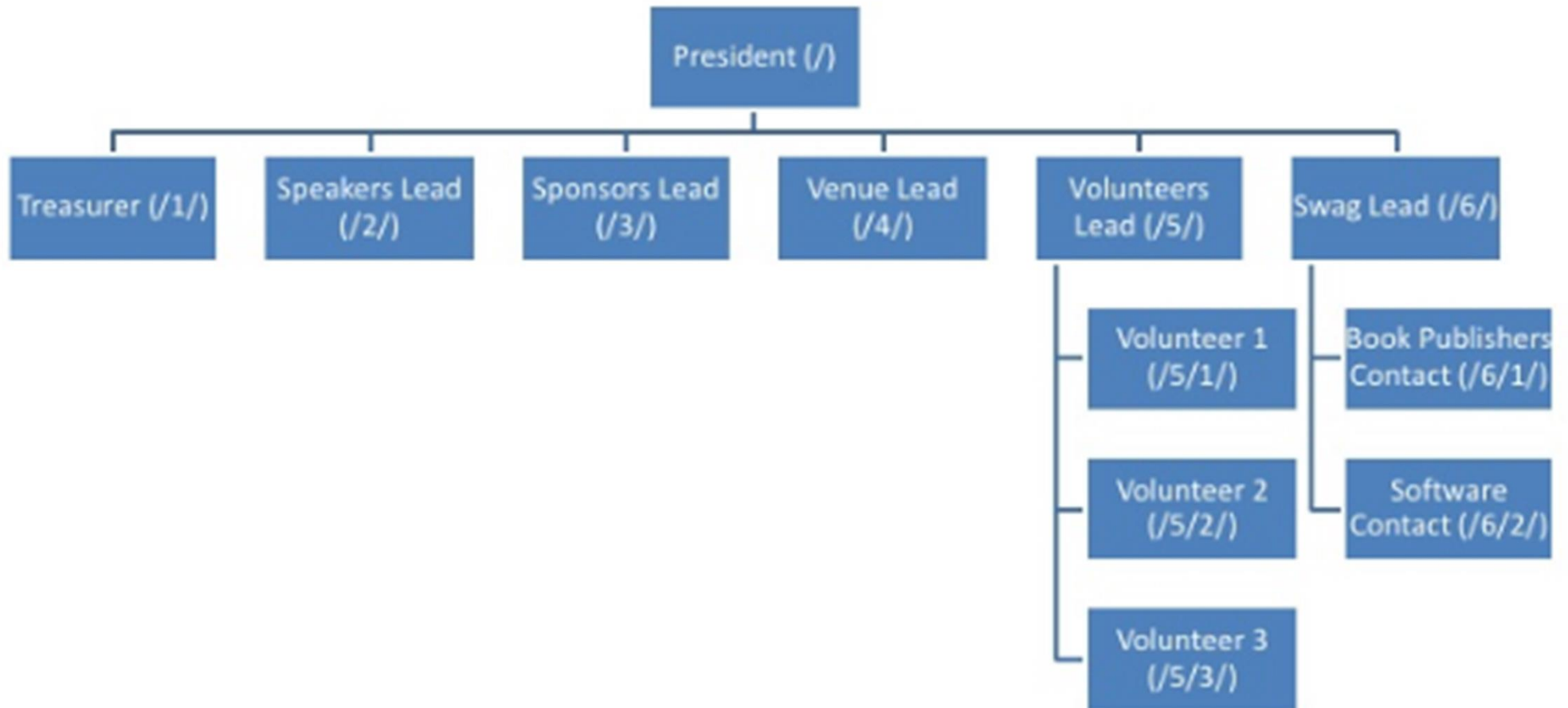
Employee	Manager	NodeLevel	LevelFromParent
Пупкин	NULL	0	0
Иванов	Пупкин	1	1
Петров	Иванов	2	1
Петров	Пупкин	2	2
Сидоров	Иванов	2	1
Сидоров	Пупкин	2	2

Тип данных с SQL Server 2008 для отображения и хранения иерархий

Встроенные методы работы с иерархией

Нет отслеживания корректности иерархии (при удалении родительского узла)

Не следит за уникальностью



https://www.slideshare.net/sadukie/intro-to-the-new-data-types-in-sql-2008?qid=af6f5fd2-b7e6-49ad-9834-36a5350f46d1&v=&b=&from_search=5

Используется, когда оба набора имеют индекс

Может использовать tempdb если в первом наборе много дубликатов

Лучший вариант для больших наборов данных



Merge Join

Рефлексия

О чем мы говорили сегодня?

- Что такое Collation?
- Зачем делать таблицы со связью 1 к 1?
- Зачем используются System Versioned Tables?



Рефлексия

Напишите, пожалуйста, свое впечатление о вебинаре.

- Отметьте 3 пункта, которые вам запомнились с вебинара.
- Что вы будете применять в работе из сегодняшнего вебинара?



Заполните, пожалуйста,
опрос в ЛК о занятии



Спасибо
за внимание!

До встречи в **Slack** и на вебинаре

