Visual Studio LIVE! | San Diego

EXPERT SOLUTIONS FOR .NET DEVELOPERS

Level: Intermediate

Modern SQL Server Security Features for Developers

Leonard Lobel
Chief Technology Officer
Sleek Technologies

Code Again for the First Time!

Visual Studio 25 YEARS OF CODING INNOVATION

About Me

sleek technologies

TALLAN

Microsoft Most Valuable Professional

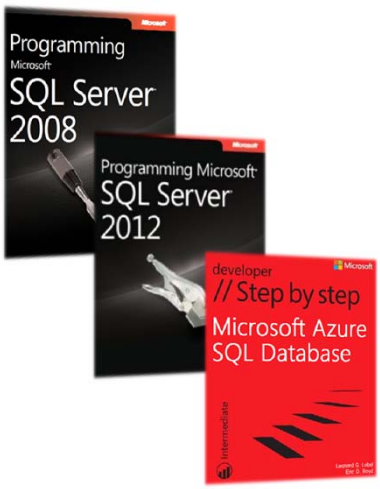
pluralsight

Leonard Lobel

- CTO & Co-Founder
  - Sleek Technologies, Inc.
- Principal Consultant
  - Tallan, Inc.
- Microsoft MVP
  - Data Platform
- Trainer/Speaker/Author
- Programming since 1979

Contact

- Email: lenni.lobel@sleektech.com
- Blog: lennilobel.wordpress.com
- Twitter: @lennilobel



Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Download Slides and Code

http://bit.ly/vslchicago2018_sqlsecurity

(all lower case!)

Visual Studio LIVE!

Agenda

- **Three new SQL Server 2016 security features**
  - Dynamic Data Masking (DDM)
  - Row Level Security (RLS)
  - Always Encrypted (AE)

Visual Studio LIVE!

Dynamic Data Masking (DDM)

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Introducing Dynamic Data Masking (DDM)

- Limit exposure to sensitive data by masking
  - Full – entire column is masked
  - Partial – show starting and/or ending characters of the column data, mask the rest with a custom string
  - Email – show the first character of the column data, mask the rest with XXX@XXXX.com
  - Random – entire column is replaced by random values
- Reveals masked data to queries
  - Data in the database is not changed
- Enforced at the database level
  - No impact at the application level

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Masking Table Columns

```
CREATE TABLE Customer(
  FirstName varchar(20)
   MASKED WITH (FUNCTION='partial(1, "...", 0)'),
  LastName varchar(20),
  Phone varchar(12)
   MASKED WITH (FUNCTION='default()'),
  Email varchar(200)
   MASKED WITH (FUNCTION='email()'),
  Balance money
   MASKED WITH (FUNCTION='random(1000, 5000)'))

ALTER TABLE Customer
 ALTER COLUMN LastName
 ADD MASKED WITH (FUNCTION='default()')
```

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Masking Different Data Types

Masking Function	Behavior	Strings	Numbers	Dates	Other Types

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Masking Different Data Types

Masking Function	Behavior	Strings	Numbers	Dates	Other Types
default()	Show **xxxx** mask (strings), or minimum value (other types)	Yes	Yes	Yes	Yes

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

2016+

Masking Different Data Types

Masking Function	Behavior	Strings	Numbers	Dates	Other Types
default()	Show **xxxx** mask (strings), or minimum value (other types)	Yes	Yes	Yes	Yes
partial(*a*, 'x', *b*)	Show first *a* characters, custom mask, and last *b* characters	Yes	No	No	No

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Masking Different Data Types

Masking Function	Behavior	Strings	Numbers	Dates	Other Types
default()	Show **xxxx** mask (strings), or minimum value (other types)	Yes	Yes	Yes	Yes
partial(*a*, 'x', *b*)	Show first *a* characters, custom mask, and last *b* characters	Yes	No	No	No
email()	Show first character and **XXX@XXXX.com**	Yes	No	No	No

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Masking Different Data Types

Masking Function	Behavior	Strings	Numbers	Dates	Other Types
default()	Show **xxxx** mask (strings), or minimum value (other types)	Yes	Yes	Yes	Yes
partial(*a*, 'x', *b*)	Show first *a* characters, custom mask, and last *b* characters	Yes	No	No	No
email()	Show first character and **XXX@XXXX.com**	Yes	No	No	No
random(*a*, *b*)	Show random value between *a* and *b*	No	Yes	No	No

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Discovering Masked Columns

- sys.columns
  - is_masked
  - masking_function
- sys.masked_columns
  - Inherits from sys.columns
  - Filters to show only masked columns
    - WHERE is_masked = 1

Visual Studio LIVE!

Mask Permissions

- DDM is based on user permissions
- Create a table with masked columns
  - No special permission required
- Add, replace, or remove a column mask
  - Requires **ALTER ANY MASK** permission
- View unmasked data in masked columns
  - Requires **UNMASK** permission
- Updating data in a masked column
  - No special permission

Visual Studio LIVE!

Dynamic Data Masking (DDM)

demo

Visual Studio LIVE!

DDM Limitations and Considerations

- DDM cannot be used with
  - FILESTREAM columns
  - COLUMN_SET, or a sparse column that's part of a COLUMN_SET
  - Computed columns
    - But will return masked data if it depends on a masked column
  - Key for FULLTEXT index
  - Encrypted columns (Always Encrypted)
- Masking is a one-way street
  - Once masked, the actual data can never be obtained
  - An ETL process from a source with masked columns results in an irreversible data loss when loaded into the target environment

Visual Studio LIVE!

Row Level Security (RLS)

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Introducing Row-level Security (RLS)

- Restrict access to individual rows in a table
  - Create predicate functions (inline TVF)
  - Write custom logic to control user access to every row
- Security policy
  - Bind the functions to tables as a filter or block predicate
  - SQL Server filters and blocks user access to individual rows
  - Can enable/disable the policy as desired

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Filter and Block Predicates

- Filter predicate
  - SELECT, UPDATE, DELETE
    - Can't select, update, or delete rows that violate the predicate
- Block predicate
  - AFTER INSERT, AFTER UPDATE
    - Can't insert or update rows to values that would violate the predicate
  - BEFORE UPDATE, BEFORE DELETE
    - Can't update or delete rows that violate the predicate
    - Implied when combined with filter predicate

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values
Filter	No	Yes	Yes

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values
Filter	No	Yes	Yes
AFTER INSERT block	Yes	No	Yes

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values
Filter	No	Yes	Yes
AFTER INSERT block	Yes	No	Yes
AFTER UPDATE block	Yes	Yes	No

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values
Filter	No	Yes	Yes
AFTER INSERT block	Yes	No	Yes
AFTER UPDATE block	Yes	Yes	No
BEFORE UPDATE block	No	N/A	N/A

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

Predicate	SELECT/UPDATE/DELETE rows that violate the predicate	INSERT rows with violating values	UPDATE rows to violating values
Filter	No	Yes	Yes
AFTER INSERT block	Yes	No	Yes
AFTER UPDATE block	Yes	Yes	No
BEFORE UPDATE block	No	N/A	N/A
BEFORE DELETE block	No	N/A	N/A

Creating Security Predicate Functions

- Write a security predicate function
  - Ordinary inline table-valued function (TVF)
    - Must be schema-bound
  - Accept any parameters of any type
    - Map these parameters to column values
- Implement your own custom logic in T-SQL
  - Examine the row via the columns passed in as parameters
    - Determine if access should be allowed or denied
  - Return a scalar 1 (allow) or nothing at all (deny)
  - Encapsulate logic inside WHERE clause of a single SELECT statement inside the TVF

2016+

Creating Security Predicate Functions

```
CREATE FUNCTION sec.fn_MySecurityPredicate(@Parm1 AS int, ...)
 RETURNS TABLE
 WITH SCHEMABINDING
AS
  -- SQL Server passes in column values of each row via parameters

  RETURN
    SELECT 1 AS Result
    WHERE ...
      -- Custom logic here examines the parameters (column values)
      -- passed in, and determines the row's accessibility
```

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy

- Create a security policy
  - Add filter and block predicates to the policy
- Bind each predicate function to a table
  - Map table columns to the TVF parameters
    - SQL Server will call the TVF to determine the accessibility of each row

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

RLS Security Policy Examples

- With filter predicate

```
CREATE SECURITY POLICY sec.MySecurityPolicy
 ADD FILTER PREDICATE sec.fn_MySecurityPredicate(Col1, ...)
  ON dbo.MyTable
 WITH (STATE = ON)
```

- With AFTER INSERT and AFTER UPDATE block predicates

```
CREATE SECURITY POLICY sec.MySecurityPolicy
 ADD BLOCK PREDICATE sec.fn_MySecurityPredicate(Col1, ...)
  ON dbo.MyTable AFTER INSERT,
 ADD BLOCK PREDICATE sec.fn_MySecurityPredicate(Col1, ...)
  ON dbo.MyTable AFTER UPDATE,
 WITH (STATE = ON)
```

Visual Studio LIVE!

Getting started with
Row-Level Security (RLS)

Visual Studio LIVE!

Identifying Users for RLS

- Credentials supplied for the database connection
  - SQL Server login (username and password)
  - Windows authentication
  - Obtain the username from DATABASE_PRINCIPAL_ID
- Different strategy required for n-tier applications
  - Typically, all users connect to the database using the same service account from the application tier
  - DATABASE_PRINCIPAL_ID is the same for every user
- Solution: Use new SESSION_CONTEXT feature
  - Store the application level user ID as a readonly value in session context

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Using Row-Level Security
in n-tier applications

demo

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Always Encrypted

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Traditional SQL Server Encryption Features

- Column (cell-level) encryption
  - Uses certificates or symmetric keys
- Database (page-level) and backup encryption
  - Transparent Data Encryption (TDE)
  - Uses TDE certificate with database encryption keys (DEKs)
- Keys and certificates are stored in the database
  - Risk of security breach at the database level
- Data is only encrypted “at rest”
  - Risk of security breach while “in flight”

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Introducing Always Encrypted

- Always Encrypted in SQL Server 2016
  - Based on keys managed outside the database
  - Keys are never revealed to SQL Server
- Separating those who *own* the data from those who *manage* it
  - Uses client side drivers to encrypt/decrypt on the fly
- SQL server is incapable of decrypting on its own
  - Data is always encrypted in flight
- Enable Always Encrypted
  - Use T-SQL or the Always Encrypted Wizard in SSMS

Encryption Types

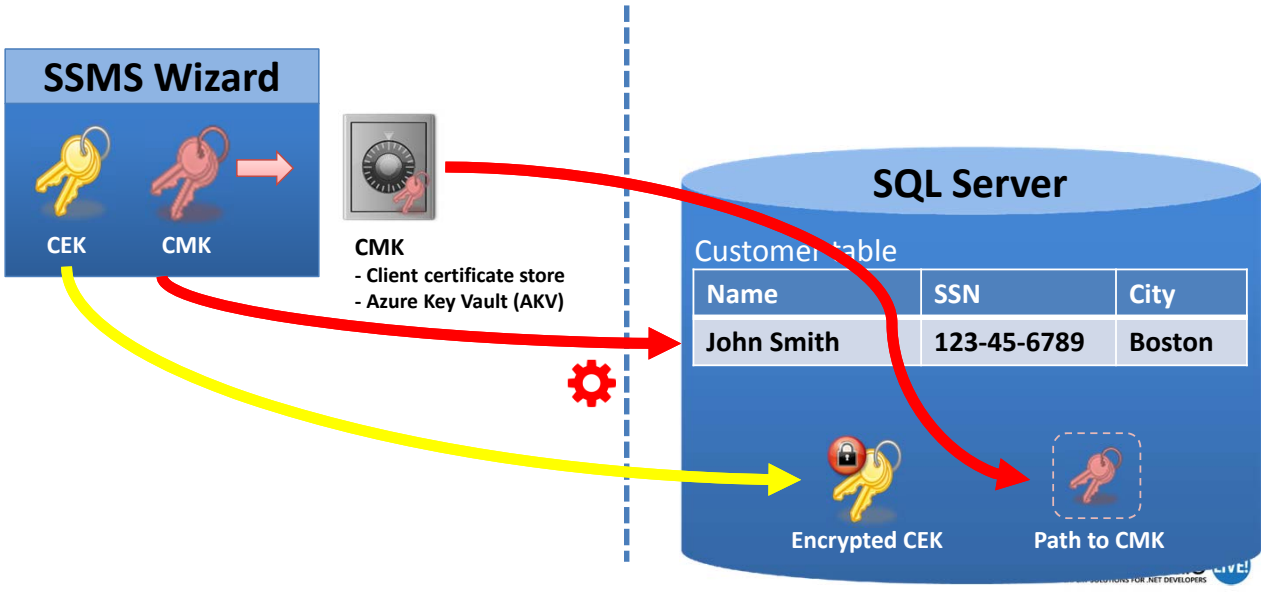
- Randomized
  - Unpredictable, more secure
  - No support for equality searches, joins, grouping, indexing
  - Use for data that is returned but not queried
- Deterministic
  - Predictable, less secure
  - Use for data that must be queried
  - Easier to guess by examining encryption patterns
    - Increased risk for small value sets (e.g., True/False)

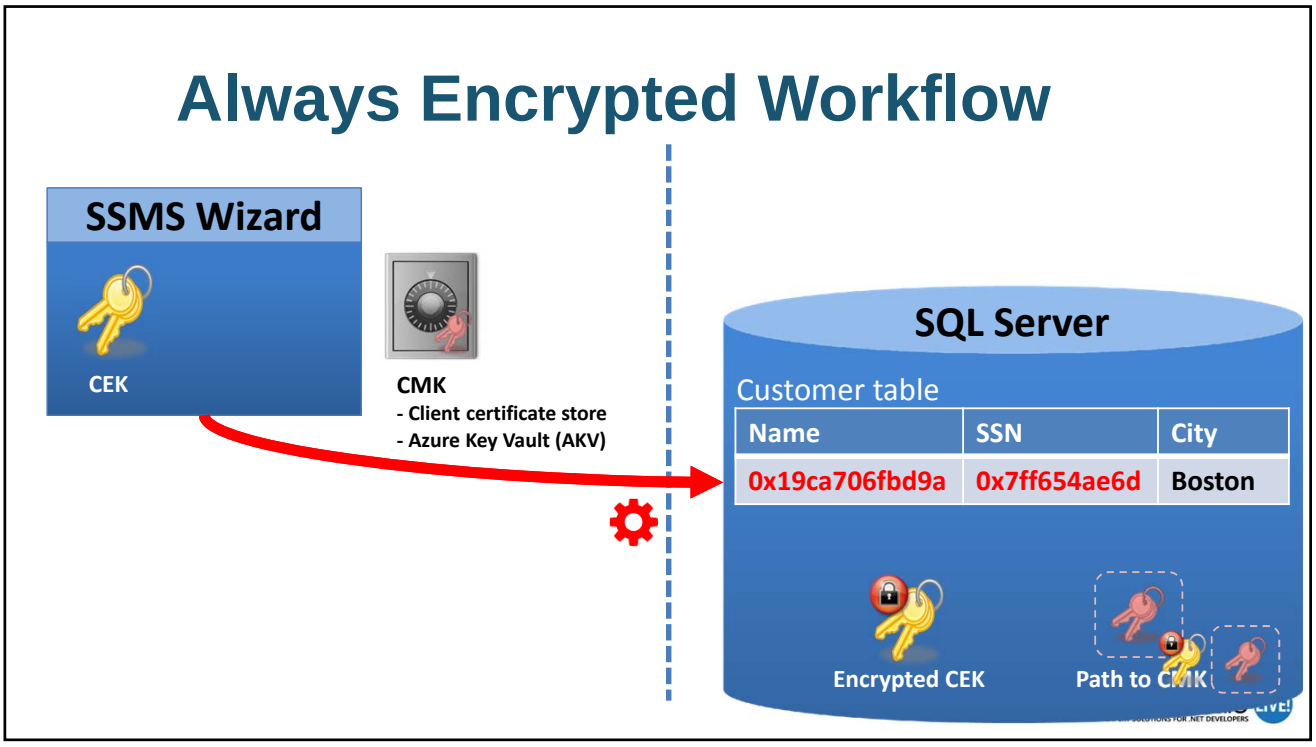
Encryption Keys

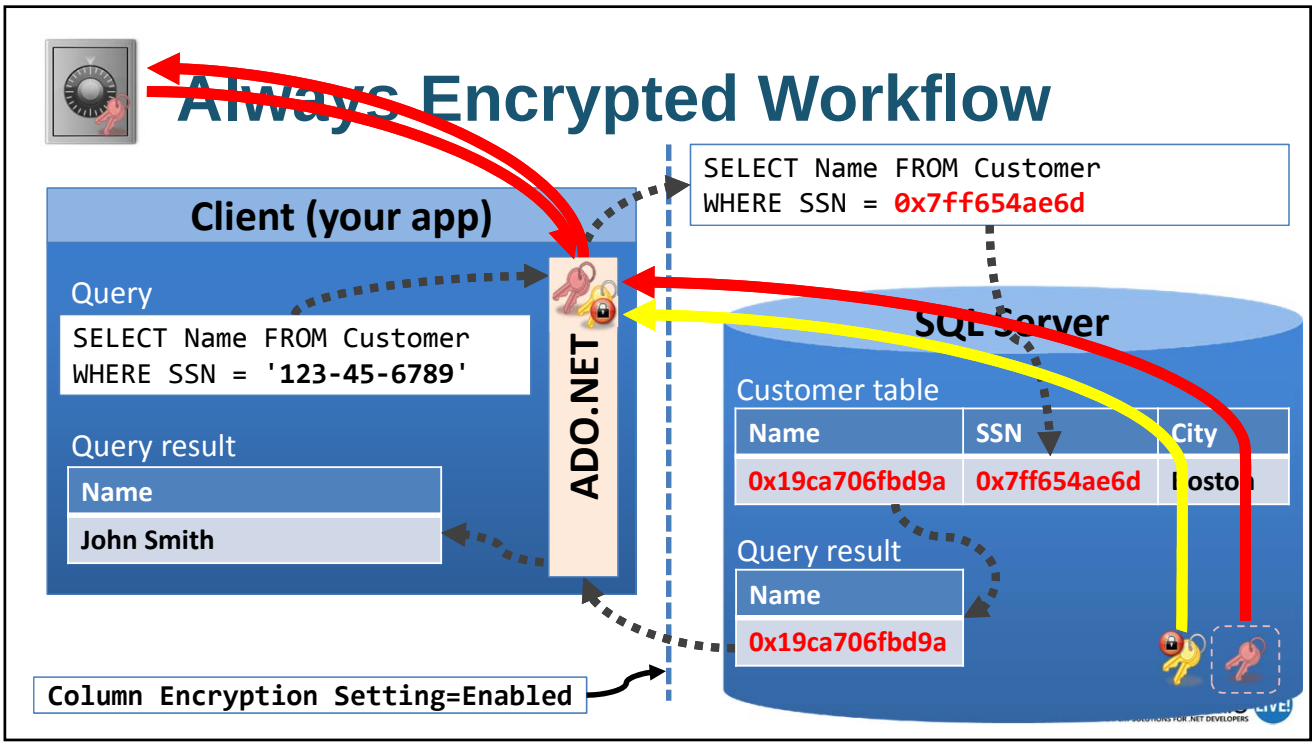
- Column Encryption Keys (CEK)
  - Used to encrypt values in specific columns
  - Encrypted versions of each CEK is stored in the database
- Column Master Keys (CMK)
  - Used to encrypt all the CEKs
  - Must be stored externally in a secure key store
    - Key store providers: Azure Key Vault, Certificate store, HSM

Visual Studio LIVE!

Always Encrypted Workflow




Always Encrypted Workflow
SSMS Wizard
CEK
CMK
- Client certificate store
- Azure Key Vault (AKV)
SQL Server
Customer table
Name
SSN
City
0x19ca706fbd9a
0x7ff654ae6d
Boston
Encrypted CEK
Path to CMK


Always Encrypted Workflow
Client (your app)
Query
SELECT Name FROM Customer
WHERE SSN = '123-45-6789'
Query result
Name
John Smith
ADO.NET
SELECT Name FROM Customer
WHERE SSN = 0x7ff654ae6d
SQL Server
Customer table
Name
SSN
City
0x19ca706fbd9a
0x7ff654ae6d
Query result
Name
0x19ca706fbd9a
Column Encryption Setting=Enabled

Always Encrypted Wizard (SSMS)

- Creates CMK in either:
  - Local Windows Certificate Store
  - Azure Key Vault
- Creates CEKs
  - Then encrypts them from the CMK
- Deploys to database:
  - Encrypted CEKs
  - Path to CMK
- Runs encryption migration
  - Queries the unencrypted table
  - Encrypts client-side (within SSMS)
  - Creates new encrypted temp table
  - Swaps in the new temp table to replace the old unencrypted table

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

2016+

Always Encrypted Catalog Views

- **`sys.column_master_keys`**
  - Identifies each CMK
  - Contains external path to CMK location
- **`sys.column_encryption_keys`**
  - Identifies each CEK
- **`sys.column_encryption_key_values`**
  - Contains CMK-encrypted values of each CEK
- **`sys.columns`**
  - New metadata columns to identify encrypted columns

Visual Studio LIVE! EXPERT SOLUTIONS FOR .NET DEVELOPERS

Always Encrypted

demo

Visual Studio LIVE!

CMK Rotation

- CEKs encrypt all your sensitive data
  - Which is why they are encrypted by a CMK
- The CMK encrypts all your CEKs
  - When the CMK is compromised, all your sensitive data is compromised
- Solution: Rotate the CMK
  - Create a new CMK
  - Re-encrypt the CEKs with the new CMK
  - PowerShell script available at
    - https://blogs.msdn.microsoft.com/sqlsecurity/2015/08/13/always-encrypted-key-rotation-column-master-key-rotation/
  - SQL Server Management Studio has integrated GUI support

Visual Studio LIVE!

AE Limitations and Considerations

- Unsupported in SSDT
- Unsupported data types
  - xml, rowversion, image, ntext, text, sql_variant, hierarchyid, geography, geometry
- Also not supported for
  - FILESTREAM, ROWGUIDCOL, sparse, or partitioning columns
  - Fulltext indexes
  - Columns with default constraints
- Unsupported with randomized encryption
  - Columns referenced by unique constraints
  - Primary key columns

AE Limitations and Considerations (cont.)

- Client code
  - .NET 4.6 only
  - ODBC and JDBC updates to come
- Entity Framework 6 considerations
  - http://blogs.msdn.com/b/sqlsecurity/archive/2015/08/27/using-always-encrypted-with-entity-framework-6.aspx
- Performance
- Troubleshooting becomes more complex
- Additional management to install certificates on all clients
- And more...
  - http://blogs.sqlsentry.com/aaronbertrand/t-sql-tuesday-69-always-encrypted-limitations/

Questions?

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Thank You!

- Contact me
  - lenni.lobel@sleektech.com
- Visit my blog
  - lennilobel.wordpress.com
- Follow me on Twitter
  - @lennilobel
- Thanks for coming! ☺

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS