Migrate a 'surrogate' key rather than the code itself?

Do we have different 'Status' codes for Segment, Segment Poing, Segment Description, etc, or can we get away with one 'Status' table?

Segment Status
- Segment Status Id: INTEGER
- Segment Status Code: NCHAR(1)
- Segment Status Description: NVARCHAR(100)

Should we use a GUID (global unique identifier) in case we ever want to merge data with another state?

[Is 'SEG' a SQL Reserved Word?]

Segment
- Segment Identifier: INTEGER
- Segment Local Identifier: NVARCHAR(9)
- Segment Transport Mode Code: NVARCHAR(2)
- Segment Create Date: DATE
- Segment Update Date: DATE
- Segment Validate Date: DATE
- Segment Retire Date: DATE
- Segment Status Code: NCHAR(1)
- Segment Object Code: NCHAR(1)
- Segment Length Number: DECIMAL(9,2)
- Segment Geometry: LARGE BINARY
- Segment Status Id: INTEGER (FK)
- Horizontal Accuracy Measurement Method Identifier: INTEGER (FK)

Should we use "Road Authority" or "Mode Authority"?

Mode Authority
- Mode Authority Identifier: INTEGER
- Mode Authority Code: NVARCHAR()
- Mode Authority Name: NVARCHAR(50)
- Mode Authority Level Description: NVARCHAR(300)

Road Authority
- Road Authority Identifier: INTEGER
- Road Authority Code: NVARCHAR()
- Road Authority Name: NVARCHAR(50)
- Road Authority Level Description: NVARCHAR(300)

Transport Mode Type
- Transport Mode Type Identifier: INTEGER
- Transport Mode Type Code: NVARCHAR()
- Transport Mode Type Description: NVARCHAR(100)

Need the following fields in City View

CityName
FIPSCityIdentifier

D:NOA

Horizontal Accuracy Measurement Method
- Horizontal Accuracy Measurement Method Identifier: INTEGER
- Horizontal Accuracy Measurement Method Code: NCHAR(3)
- Horizontal Accuracy Measurement Method Code Description: NVARCHAR(255)
- Horizontal Accuracy Measurement Method Datum Description: NVARCHAR(255)
- Horizontal Accuracy Measurement Method Project Description: NVARCHAR(255)

Need the following fields in County View

CountyName
FIPSCountyIdentifier

Segment Description
- Segment Description Identifier: INTEGER
- Segment Description Local Identifier: NVARCHAR(9)
- Segment Local LRS Description: NVARCHAR(25)
- Segment Path Description: NVARCHAR(255)
- Segment Description Create Date: DATE
- Segment Description Update Date: DATE
- **Segment Description Validate Date: DATE**
- Segment Description Retire Date: DATE
- Segment Description Status Code: NCHAR(1)
- Segment Description Local Length Number: DECIMAL(9,2)
- Segment Local Functional Class Code: NVARCHAR(2)
- Segment Description Begin Milepoint: DECIMAL(5,2)
- Segment Description End Milepoint: DECIMAL(5,2)
- Segment Description Left Low Address: NVARCHAR(10)
- Segment Description Left Low Zip Code: NVARCHAR(10)
- Segment Description Right Low Address: NVARCHAR(10)
- Segment Description Right Low Zip Code: NVARCHAR(10)
- Segment Description Left High Address: NVARCHAR(10)
- Segment Description Left High Zip Code: NVARCHAR(10)
- Segment Description Right High Address: NVARCHAR(10)
- Segment Description Right High Zip Code: NVARCHAR(10)
- Length Accuracy Measurement Method Identifier: INTEGER (FK)

D:NOA

Segment Description Road Segment Name
- Segment Description Identifier: INTEGER (FK)
- Road Segment Name Identifier: INTEGER (FK)

D:NOA

Road Segment Name
- Road Segment Name Identifier: INTEGER
- Road Segment Name: NVARCHAR(50)

D:NOA

Segment Name Prefix
- Road Segment Name Identifier: INTEGER (FK)
- Road Segment Name Prefix Identifier: INTEGER (FK)

D:NOA

Road Segment Name Prefix
- Road Segment Name Prefix Identifier: INTEGER
- Road Segment Name Prefix: NVARCHAR(15)

D:NOA

Segment Description Road Segment Number
- Segment Description Identifier: INTEGER (FK)
- Road Segment Number Identifier: INTEGER (FK)

D:NOA

Road Segment Number
- Road Segment Number Identifier: INTEGER
- Road Segment Number: NVARCHAR(15)

D:NOA

Segment Name Suffix
- Road Segment Number Identifier: INTEGER (FK)
- Road Segment Name Suffix Identifier: INTEGER (FK)

D:NOA

Road Segment Name Suffix
- Road Segment Name Suffix Identifier: INTEGER
- Road Segment Name Suffix: NVARCHAR(15)

Made the following changes to table structure:

Field name changes:
'...Validation Date' to Validate Date.

Added fields below:

...Left Low Zip Code
...Right Low Zip Code
...Left High Zip Code
...Right High Zip Code

ISSUE to address: We do NOT currently have a StateCounty table or view.

StateCounty

Segment Point
- Segment Point Identifier: INTEGER
- Segment Point Agreement Indicator: NCHAR(1)
- Segment Point Object Code: NCHAR(1)
- Segment Point Local Identifier: NVARCHAR(9)
- Segment Point Location Description: NVARCHAR(255)
- Segment Point Status Code: NCHAR(1)
- Segment Point Latitude Number: DECIMAL(10,6)
- Segment Point Longitude Number: DECIMAL(10,6)
- Segment Point Create Date: DATE
- Segment Point Update Date: DATE
- Segment Point Validate Date: DATE
- Segment Point Retire Date: DATE

D:NOA

Made the following changes to table structure:

Field name changes:
'...Validation Date' to Validate Date.

Need the following fields in State View

Statename
FipsStateIdentifier

ISSUE to address: Our WSDOT Common State table does NOT currently have FipsStateIdentifier.

Exactly what is the purpose of this table?

Why is Begin/End Milepoint in the description table?
If it's here, that means a 'Segment' can 'move'; i.e. be from
mile 50-55, to 49-54. Is that the intent?

D:NOA

Length Accuracy Measurement Method
- Length Accuracy Measurement Method Identifier: INTEGER
- Length Accuracy Measurement Method Code: NCHAR(3)
- Length Accuracy Measurement Method Code Description: NVARCHAR(50)

Segment Point -Aviation
- Segment Point Aviation Identifier: INTEGER
- Segment Point Identifier: NVARCHAR()
- Airport Identifier: NVARCHAR(4)
- **Surface Type: NVARCHAR(3)**
- Instrument Approach: BOOLEAN
- ARC Code: NVARCHAR(4)
- Surface Width: DECIMAL(4)
- Elevation: DECIMAL(6,1)
- FAA Classification: NVARCHAR(30)
- State Classification: NVARCHAR(10)
- Airport Name: NVARCHAR(100)
- Tower: BOOLEAN
- AWAS: BOOLEAN
- Owner: NVARCHAR(30)
- Terminal: BOOLEAN

Is this 'SurfaceType' the same as in Segment Surface Type?

Segment Point Agreement
- Segment Point Agreement Identifier: INTEGER
- Segment Point Agreement Document Description: NVARCHAR(255)
- Segment Point Agreement Survey Description: NVARCHAR(255)
- Segment Point Identifier: INTEGER (FK)

Segment Point Type
- Segment Point Type Identifier: INTEGER
- Segment Point Type Code: NVARCHAR()
- Segment Point Type Description: NVARCHAR(100)

Surface Type
- Surface Type Id: INTEGER
- Surface Type Code: NCHAR(1)
- Surface Type Description: NVARCHAR(100)

Migrate a 'surrogate' key rather than the code itself?

Segment Surface Type
- Segment Surface Type Identifier: INTEGER
- Segment Identifier: NVARCHAR()
- Segment Surface Type Code: NCHAR(1)
- Surface Begin Milepoint Number: DECIMAL(6,3)
- Surface End Milepoint Number: DECIMAL(6,3)
- Surface Left Low Address: NVARCHAR(10)
- Surface Left Low Zip Code: NVARCHAR(10)
- Surface Right Low Address: NVARCHAR(10)
- Surface Right Low Zip Code: NVARCHAR(10)
- Surface Left High Address: NVARCHAR(10)
- Surface Left High Zip Code: NVARCHAR(10)
- Surface Right High Address: NVARCHAR(10)
- Surface Right High Zip Code: NVARCHAR(10)

Why are these milepoints DC(6,3) but those in Segment Description DC(5,2)?

Segment Lanes
- Segment Lanes Identifier: INTEGER
- Segment Identifier: NVARCHAR()
- Segment Lanes Code: NCHAR(1)
- Lanes Begin Milepoint Number: DECIMAL(6,3)
- Lanes End Milepoint Number: DECIMAL(6,3)
- Lanes Left Low Address: NVARCHAR(10)
- Lanes Left Low Zip Code: NVARCHAR(10)
- Lanes Right Low Address: NVARCHAR(10)
- Lanes Right Low Zip Code: NVARCHAR(10)
- Lanes Left High Address: NVARCHAR(10)
- Lanes Left High Zip Code: NVARCHAR(10)
- Lanes Right High Address: NVARCHAR(10)
- Lanes Right High Zip Code: NVARCHAR(10)

Made the following changes to table structure in the 3 'event' tables (Segment Surface Type, Segment Lanes, Segment Speed Limit):

Added fields below:

...Left Low Zip Code
...Right Low Zip Code
...Left High Zip Code
...Right High Zip Code

Segment Ferry
- Segment Ferry Identifier: INTEGER
- Segment Identifier: NVARCHAR()
- Ferry Hours Not Available: NVARCHAR(30)
- Ferry System-Wide Restrictions: NVARCHAR(30)
- Ferry Route Load Restrictions: NVARCHAR(30)
- Ferry Route Length Restrictions: NVARCHAR(30)
- Ferry Route Crossing Time: DECIMAL(3,0)
- Ferry Route Begin Milepoint Number: DECIMAL(6,3)
- Ferry Route End Milepoint Number: DECIMAL(6,3)

Segment Speed Limit
- Segment Speed Limit Identifier: INTEGER
- Segment Identifier: NVARCHAR()
- Segment Speed Code: NCHAR(1)
- Speed Begin Milepoint Number: DECIMAL(6,3)
- Speed End Milepoint Number: DECIMAL(6,3)
- Speed Left Low Address: NVARCHAR(10)
- Speed Left Low Zip Code: NVARCHAR(10)
- Speed Right Low Address: NVARCHAR(10)
- Speed Right Low Zip Code: NVARCHAR(10)
- Speed Left High Address: NVARCHAR(10)
- Speed Left High Zip Code: NVARCHAR(10)
- Speed Right High Address: NVARCHAR(10)
- Speed Right High Zip Code: NVARCHAR(10)

If this are meant to be 'Event' tables, and defined 'separately' from Segments, then should Segment Identifier be associated with an instance?
What about when an 'Event' overlaps multiple segments?
If Segment Identifier is NOT included then what do we need to store to identify what road/etc. the event is associated with?

= Suggested Additional Entity

General Questions:

1. Should we be consistent in field naming?

'...Validate Date' - vs - '...Validation Date'

2. Is there some business reason there are no '...Type/Status' tables
   a. Surface Type
   b. Status - 'SegmentStatus', 'SegmentPointStatus'

3. Are there any issues with using WSDOT Database standards?
   a. No Underscores in Field Names
   b. nvarchar/nchar instead of varchar/char
   c. abbreviating of Physical Field Names; ex. Descr for Description