

Ed-Fi 2.x Session Entity Mapping

Overview

Use this page to help understand and define the mapping of Aeries data to Ed-Fi data



Status

Ready for certification

Document Version 1.5

Last Updated 25 Feb 2020

Note

We start by getting a list of terms from the TRM table. From each row we get the start data, end date, and term.

Then we check to see if the school is an elementary school or secondary school. If an elementary school pull the data from the TRM table, else get it from the GRP table.

Links

Entity

<https://mappingedu.ed-fi.org/#/entity/18caf4aa-4ed2-e811-80d9-005056b36109/info?dataStandardId=db3697f4-4dd2-e811-80d9-005056b36109>

Scenarios

<https://techdocs.ed-fi.org/display/EDFICERT/Education+Organization+Calendar+%3E+Sessions+Scenarios>

Schema

<https://schema.ed-fi.org/datahandbook-v22/Ed-Fi-UDM-Handbook-Index.html#/Session582>

SQL

Root Table

TRM

Scheduled

TRM_Functions.GetEdFiDataTable_Sessions

```
SELECT [SC], [D1] AS BeginDate, [D2] AS EndDate, [TM] AS StateCode, [TM] AS SEQ, 'Elementary' AS SchoolType
FROM [TRM]
WHERE [DEL] = 0
```

Real-Time

TRM_Functions.GetEdFiDataTable_Sessions

```
SELECT [SC], [D1] AS BeginDate, [D2] AS EndDate, [TM] AS StateCode, [TM] AS SEQ, 'Elementary' AS SchoolType
FROM [TRM]
WHERE [DEL] = 0 AND [DTS] >= @TimeSince
```

Elementary Schools

needed for session grading period

Root Table

TRM

SQL

Scheduled

TRM.Functions.GetEdFiDataTable_GradingPeriods

```
SELECT [SC], [D1] AS BeginDate, [D2] AS EndDate, [TM] AS StateCode, [TM] AS SEQ, 'Elementary' AS SchoolType
FROM [TRM]
WHERE [DEL] = 0 AND [SC] = @SC
```

Real-Time

TRM.Functions.GetEdFiDataTable_GradingPeriods

```
SELECT [SC], [D1] AS BeginDate, [D2] AS EndDate, [TM] AS StateCode, [TM] AS SEQ, 'Elementary' AS SchoolType
FROM [TRM]
WHERE [DEL] = 0 AND [SC] = @SC AND [DTS] >= @TimeSince
```

Secondary Schools

needed for session grading period

Root Table

GRP

SQL

Scheduled

GRP_Functions.GetEdFiDataTable_GradingPeriods

```
WITH cteGRP ([SC], [KY], [DTS]
, [B1], [B2], [B3], [B4], [B5], [B6], [B7], [B8], [B9], [B10], [B11], [B12]
, [E1], [E2], [E3], [E4], [E5], [E6], [E7], [E8], [E9], [E10], [E11], [E12]
, [C1], [C2], [C3], [C4], [C5], [C6], [C7], [C8], [C9], [C10], [C11], [C12], RowNumber) AS
(
    SELECT [SC], [KY], [DTS]
    , [B1], [B2], [B3], [B4], [B5], [B6], [B7], [B8], [B9], [B10], [B11], [B12]
    , [E1], [E2], [E3], [E4], [E5], [E6], [E7], [E8], [E9], [E10], [E11], [E12]
    , [C1], [C2], [C3], [C4], [C5], [C6], [C7], [C8], [C9], [C10], [C11], [C12]
    , ROW_NUMBER() OVER(PARTITION BY [SC] ORDER BY [KY]) [RowNumber]
    FROM [GRP]
    WHERE [DEL] = 0 AND [SC] = @SC
)
SELECT [SC], [KY], [B1] 'BeginDate', [E1] 'EndDate', 1 [SEQ], [C1] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C1] > ' '
UNION
SELECT [SC], [KY], [B2] 'BeginDate', [E2] 'EndDate', 2 [SEQ], [C2] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C2] > ' '
UNION
SELECT [SC], [KY], [B3] 'BeginDate', [E3] 'EndDate', 3 [SEQ], [C3] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C3] > ' '
UNION
SELECT [SC], [KY], [B4] 'BeginDate', [E4] 'EndDate', 4 [SEQ], [C4] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C4] > ' '
UNION
SELECT [SC], [KY], [B5] 'BeginDate', [E5] 'EndDate', 5 [SEQ], [C5] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C5] > ' '
UNION
SELECT [SC], [KY], [B6] 'BeginDate', [E6] 'EndDate', 6 [SEQ], [C6] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C6] > ' '
UNION
SELECT [SC], [KY], [B7] 'BeginDate', [E7] 'EndDate', 7 [SEQ], [C7] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C7] > ' '
UNION
SELECT [SC], [KY], [B8] 'BeginDate', [E8] 'EndDate', 8 [SEQ], [C8] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C8] > ' '
UNION
SELECT [SC], [KY], [B9] 'BeginDate', [E9] 'EndDate', 9 [SEQ], [C9] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C9] > ' '
UNION
SELECT [SC], [KY], [B10] 'BeginDate', [E10] 'EndDate', 10 [SEQ], [C10] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C10] > ' '
UNION
SELECT [SC], [KY], [B11] 'BeginDate', [E11] 'EndDate', 11 [SEQ], [C11] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C11] > ' '
UNION
SELECT [SC], [KY], [B12] 'BeginDate', [E12] 'EndDate', 12 [SEQ], [C12] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C12] > ' '

```

Real-Time

GRP_Functions.GetEdFiDataTable_GradingPeriods

```
WITH cteGRP ([SC], [KY], [DTS]
, [B1], [B2], [B3], [B4], [B5], [B6], [B7], [B8], [B9], [B10], [B11], [B12]
, [E1], [E2], [E3], [E4], [E5], [E6], [E7], [E8], [E9], [E10], [E11], [E12]
, [C1], [C2], [C3], [C4], [C5], [C6], [C7], [C8], [C9], [C10], [C11], [C12], RowNumber) AS
(
    SELECT [SC], [KY], [DTS]
    , [B1], [B2], [B3], [B4], [B5], [B6], [B7], [B8], [B9], [B10], [B11], [B12]
    , [E1], [E2], [E3], [E4], [E5], [E6], [E7], [E8], [E9], [E10], [E11], [E12]
    , [C1], [C2], [C3], [C4], [C5], [C6], [C7], [C8], [C9], [C10], [C11], [C12]
    , ROW_NUMBER() OVER(PARTITION BY [SC] ORDER BY [KY]) [RowNumber]
    FROM [GRP]
    WHERE [DEL] = 0 AND [SC] = @SC
)
SELECT [SC], [KY], [B1] 'BeginDate', [E1] 'EndDate', 1 [SEQ], [C1] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C1] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B2] 'BeginDate', [E2] 'EndDate', 2 [SEQ], [C2] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C2] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B3] 'BeginDate', [E3] 'EndDate', 3 [SEQ], [C3] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C3] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B4] 'BeginDate', [E4] 'EndDate', 4 [SEQ], [C4] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C4] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B5] 'BeginDate', [E5] 'EndDate', 5 [SEQ], [C5] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C5] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B6] 'BeginDate', [E6] 'EndDate', 6 [SEQ], [C6] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C6] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B7] 'BeginDate', [E7] 'EndDate', 7 [SEQ], [C7] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C7] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B8] 'BeginDate', [E8] 'EndDate', 8 [SEQ], [C8] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C8] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B9] 'BeginDate', [E9] 'EndDate', 9 [SEQ], [C9] 'StateCode', 'Secondary' AS SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C9] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B10] 'BeginDate', [E10] 'EndDate', 10 [SEQ], [C10] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C10] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B11] 'BeginDate', [E11] 'EndDate', 11 [SEQ], [C11] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C11] > ' ' AND [DTS] >= @TimeSince
UNION
SELECT [SC], [KY], [B12] 'BeginDate', [E12] 'EndDate', 12 [SEQ], [C12] 'StateCode', 'Secondary' AS
SchoolType
FROM cteGRP
WHERE RowNumber = 1 AND [C12] > ' ' AND [DTS] >= @TimeSince
```

Mappings

These are the actual mappings used in code

Parent	Properties	Type	Required	Table	Field	Other	Code / Notes
\	beginDate	Date	Required	TRM	BeginDate		GetNullableDate(s("D1"))
\	endDate	Date	Required	TRM	EndDate		GetNullableDate(s("D2"))
\	name	String	Required	TRM	D1,D2,SC,TM		schoolYear & " " & termDescriptor
\	termDescriptor	TermDescriptor TermMapType	Identity	TRM	TM		GetTermDescriptor(rowTRM!SC, rowTRM!TM) see function below
\	schoolYearType Reference	SchoolYearType Reference	Identity				
schoolYearType Reference	schoolYear	Integer	Identity	TRM	BeginDate		GetSchoolYear(rowTRM!BeginDate) see function below
\	gradingPeriods	List(Of SessionGrading Period)	Required				
gradingPeriods	gradingPeriods(1)	SessionGrading Period	Required				
gradingPeriods	gradingPeriod Reference	GradingPeriod Reference	Required				
gradingPeriods	schoolId	Integer	Required	TRM or GRP	SC		GetNullableInteger(rowGRPTRM!SC)
gradingPeriods	beginDate	Date	Required	TRM or GRP	BeginDate		GetNullableDate(rowGRPTRM!BeginDate)
gradingPeriods	descriptor	GradingPeriod MapType GradingPeriod Descriptor	Required	TRM or GRP	StateCode	Mapped	GetEdFiCodeValue(GradingPeriodDescriptor, "GRP", "C1", rowGRPTRM!StateCode)
\	schoolReference	SchoolReference	Identity				
schoolReference	schoolId	Integer	Identity	TRM	SC		GetNullableInteger(s("SC"))
\	totalInstructional Days	Integer	Required	TRM	SC, BeginDate, EndDate	Calculated	GetTotalInstructionalDays(drTRM!SC, rowTRM!BeginDate, rowTRM!EndDate) see function below
\	academicWeeks	List(Of SessionAcademic Weeks)	Optional				
academicWeeks[n]	academicWeeks[n]	SessionAcademic Weeks	Optional				
academicWeeks[n]	academicWeek Reference	AcademicWeek Reference	Optional				
academicWeeks[n]	link	Link	Optional				
academicWeeks[n]	schoolId	Integer	Optional				
academicWeeks[n]	weekIdentifier	String	Optional				

Code

```
Public Shared Function GetTermDescriptor(ByVal SC As Integer, ByVal TM As String, ByVal ConnectionString As String) As String
```

```
    Select Case TM
        Case "Y"          Return "Year Round"
        Case "S"          Return "Spring Semester"
        Case "F"          Return "Fall Semester"
    End Select
```

```
    If Trimester(SC, ConnectionString) Then
        Select Case TM
            Case "1"      Return "First Trimester"
            Case "2"      Return "Second Trimester"
            Case "3"      Return "Third Trimester"
        End Select
    Else
```

```
        Select Case TM
            Case "1"      Return "First Quarter"
            Case "2"      Return "Second Quarter"
            Case "3"      Return "Third Quarter"
            Case "4"      Return "Fourth Quarter"
        End Select
    End If
```

```
    Return TM
End Function
```

```
Public Shared Function GetSchoolYear(ByVal SchoolDate As String) As Integer?
```

```
    Dim schoolYear As Integer? = Nothing
    Dim schoolDateValue As Date

    If Date.TryParse(SchoolDate, schoolDateValue) Then
        schoolYear = EdFiV2Functions.GetSchoolYearFromDate(schoolDateValue)
    End If

    Return schoolYear
```

```
End Function
```

```
Public Shared Function GetSchoolYearFromDate(ByVal SchoolDate As Date) As Integer
```

```
    ' Ed-Fi sometimes uses a single year as 'school year'. It's actually, just the end year of a full
    school year in the form YYYY-YYYY
    Return GetEndYear(SchoolDate)
```

```
End Function
```

```
Public Shared Function GetEndYear(ByVal startDate As Date) As Integer
```

```
    Dim endYear As Integer = Nothing
    Dim startMonth As Integer = Month(startDate)

    If startMonth >= 7 Then
        ' July-Dec = "this year" (i.e. 9/1/2012 = 2012-2013, so 2013)
        endYear = Year(startDate) + 1
    Else ' startMonth < 7
        ' Jan-June = "last year" (i.e. 3/1/2012
        = 2011-2012, so 2012)
        endYear = Year(startDate)
    End If

    Return endYear
```

```
End Function
```