

Ed-Fi 3.x Grading Period Entity Mapping

Overview

This document defines the mapping of Aeries data to the Ed-Fi data standard.



Document Version 1.0

Last Updated 07 Nov 2021

Status

Ready for v3 certification.

Description

This entity represents the time span for which grades are reported.

Note

If the school is a straight elementary school then get the data from the TRM table, else get the data from the GRP table.

Grading Period Links

Mapping v3.2.0-c

<https://mappingedu.ed-fi.org#!/entity/32016848-e7ba-ea11-80df-005056b36109/info?dataStandardId=7ec6133d-e7ba-ea11-80df-005056b36109>

Schema v3.2.0-c

<https://schema.ed-fi.org/datahandbook-v320c#!/GradingPeriod561>

API v3.4.0

<https://api.ed-fi.org/v3.4.0/docs/index.html#/gradingPeriods>

Scenarios v3

<https://techdocs.ed-fi.org/display/EDFICERT/v3+Education+Organization+Calendar+%3E+Grading+Periods+Scenarios>

Natural Key

- GradingPeriodDescriptor
- PeriodSequence
- SchoolId
- SchoolYear

Used By

- Grade.GradingPeriod (as required)
- GradebookEntry.GradingPeriod (as optional)
- ReportCard.GradingPeriod (as required)
- Session.GradingPeriod (as optional collection)
- StudentCompetencyObjective.GradingPeriod (as required)
- StudentLearningObjective.GradingPeriod (as required)

Entity Mapping

Parent	Properties	Type	Required	Table	Field	Other	Code / Notes
\	beginDate	Date	Required	GRP	B1 -12	BeginDate	GetNullableDate(s("BeginDate"))
\	endDate	Date	Required	GRP	E 1-12	EndDate	GetNullableDate(s("EndDate"))
\	periodSequence	Integer	Identity		SEQ 1-12		GetNullableInteger(s("SEQ"))

\	gradingPeriodDescriptor	GradingPeriodDescriptor	Identity	GRP	C 1-12	StateCode	GetEdFiCodeValue(GradingPeriodDescriptor, "GRP", "C1", s("StateCode"))
\	schoolReference	SchoolReference	Identity				
schoolReference	schoolId	Integer	Identity	GRP	SC		GetNullableInteger(s("SC"))
\	schoolYearTypeReference	SchoolYearTypeReference	Identity				
schoolYearTypeReference	schoolYear	Integer	Identity	GRP	BeginDate		GetSchoolYear(s("BeginDate")) see function below
\	totalInstructionalDays	Integer	Required	GRP	SC, BeginDate, EndDate		GetTotalInstructionalDays(drGRP!SC, drGRP!BeginDate, drGRP!EndDate) see function below

Code

```
Public Shared Function GetTotalInstructionalDays(schoolCode As Integer,
startDate As Date, endDate As Date, ByVal connectionString As String)
As Integer?
```

```
    Return DAY_Functions.GetTeachingDays(schoolCode, "", startDate,
endDate, connectionString)
```

```
End Function
```

```
Public Shared Function GetTeachingDays(ByVal SC As Integer, ByVal TR As
String, ByVal StartDate As Date, ByVal EndDate As Date, ByVal
ConnectionString As String) As Integer
```

```
    Dim intDays As Integer = 0
```

```
    ' Ensure Date-Range is defined
```

```
    If Not StartDate = Nothing AndAlso IsDate(StartDate) AndAlso
Not EndDate = Nothing AndAlso IsDate(EndDate) Then
```

```
        Dim strSQL As String = "SELECT DY, DT, HO, T1, T2, T3,
T4, T5, T6, T7, T8, T9, T10, T11, T12, T13,"
```

```
        strSQL &= " T14, T15, T16, T17, T18, T19, T20, T21,
T22, T23, T24, T25, T26"
```

```
        strSQL &= " FROM DAY WHERE DEL = 0 AND SC = @SC AND DT
>= @SD AND DT <= @ED ORDER BY DY"
```

```
        Dim intTR As Integer = 0
```

```
        If TR > " " Then
```

```
            intTR = Asc(UCase(TR)) - 64
```

```
            If intTR > 26 OrElse intTR < 1 Then intTR = 0
```

```
        End If
```

```
        Dim drDAY As SqlDataReader = ConnectDB.GetDataReader
(strSQL, connectionString, New SqlParameter() {
```

```
            MakeParam("@SC", SqlDbType.SmallInt, 0, SC),
```

```

MakeParam("@SD", SqlDbType.Date, 0, StartDate), MakeParam("@ED",
SqlDbType.Date, 0, EndDate))})

        While drDAY.Read
            If intTR > 0 AndAlso intTR <= 26 Then
                If isSchoolDay(drDAY("T" & intTR)) Then
intDays += 1
                Else
                    If isSchoolDay(drDAY!HO) Then intDays
+= 1
                End If
            End While

            drDAY.Close()

        End If

        Return intDays

End Function

''' <summary>
''' GetSchoolYear - Ed-Fi uses the end date of a full school year.
Aeries calculates the full year based on the start date based on the
month.
''' </summary>
''' <param name="SchoolDate"></param>
''' <returns></returns>
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

''' <summary>
''' GetSchoolYear - Get value for Version 2 of Ed-Fi
''' </summary>
''' <param name="SchoolDate"></param>
''' <returns></returns>
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

''' <summary>
''' GetSchoolYear - This function takes the startDate of a school year,
calculates the endDate based on Aeries logic, and returns the endDate
year
''' This function is not really Ed-Fi specific and probably belongs in
an Aeries helper class.
''' </summary>
''' <param name="startDate"></param>
''' <returns></returns>
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

```