

# Ed-Fi 2.x Student Academic Record 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

## Links

### Entity

<https://mappingedu.ed-fi.org/#/entity/9fccf4aa-4ed2-e811-80d9-005056b36109/info?dataStandardId=db3697f4-4dd2-e811-80d9-005056b36109>

### Scenarios

<https://techdocs.ed-fi.org/display/EDFICERT/Student+Transcript+%3E+StudentAcademicRecord+Scenarios>

### Schema

<https://schema.ed-fi.org/datahandbook-v22/Ed-Fi-UDM-Handbook-Index.html#/StudentAcademicRecord586>

## Root Table

HIS

## Join

STU, TDF

## Scheduled SQL

## HIS\_Functions.GetEdFiDataTable\_StudentAcademicRecords

```
SELECT [HIS].[PID], [HIS].[YR], [HIS].[TE], [T].TermStateCode, [S].[CA], [S].[CR], [S].[SC], Sum([HIS].[CR])
SessionAttemptedCredits, Sum([HIS].[CC]) SessionEarnedCredits
FROM [HIS] INNER JOIN
(
    SELECT ROW_NUMBER() OVER (PARTITION BY ID ORDER BY CASE WHEN NOT [TG] > ' ' THEN ' ' ELSE 'X' END, CASE
    WHEN NOT [SP] > ' ' THEN ' ' ELSE 'X' END, [ED]) StuCount, [ID], [SC], [SN], [CA], [CR]
    FROM [STU] WHERE [DEL] = 0 AND [STU].[TG] = '' AND [STU].[ID] IN (SELECT [ENR].[ID] FROM [ENR] WHERE
    [ENR].[YR] = @YR)
)
[S] ON [HIS].[PID] = [S].[ID]
LEFT JOIN
(
    SELECT [TE], TermStateCode FROM
    (
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 0 [TE], [ST0] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST0] > ' ' GROUP
        BY [ST0]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 1 [TE], [ST1] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST1] > ' ' GROUP
        BY [ST1]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 2 [TE], [ST2] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST2] > ' ' GROUP
        BY [ST2]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 3 [TE], [ST3] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST3] > ' ' GROUP
        BY [ST3]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 4 [TE], [ST4] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST4] > ' ' GROUP
        BY [ST4]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 5 [TE], [ST5] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST5] > ' ' GROUP
        BY [ST5]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 6 [TE], [ST6] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST6] > ' ' GROUP
        BY [ST6]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 7 [TE], [ST7] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST7] > ' ' GROUP
        BY [ST7]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 8 [TE], [ST8] TermStateCode, Count
        (*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST8] > ' ' GROUP
        BY [ST8]
    ) [T]
    WHERE RowNum = 1
) [T] ON [HIS].[TE] = [T].[TE]
WHERE [HIS].[DEL] = 0 AND [S].StuCount = 1 AND [S].[SC] = @SC
GROUP BY [HIS].[PID], [HIS].[YR], [HIS].[TE], [S].[CA], [S].[CR], [S].[SC], [T].TermStateCode
```

## Real-Time SQL

## HIS\_Functions.GetEdFiDataTable\_StudentAcademicRecords

```
SELECT [HIS].[PID], [HIS].[YR], [HIS].[TE], [T].TermStateCode, [S].[CA], [S].[CR], [S].[SC], Sum([HIS].[CR])
SessionAttemptedCredits, Sum([HIS].[CC]) SessionEarnedCredits
FROM [HIS] INNER JOIN
(
    SELECT ROW_NUMBER() OVER (PARTITION BY ID ORDER BY CASE WHEN NOT [TG] > ' ' THEN ' ' ELSE 'X' END, CASE
WHEN NOT [SP] > ' ' THEN ' ' ELSE 'X' END, [ED]) StuCount, [ID], [SC], [SN], [CA], [CR]
    FROM [STU] WHERE [DEL] = 0 AND [STU].[TG] = '' AND [STU].[ID] IN (SELECT [ENR].[ID] FROM [ENR] WHERE
[ENR].[YR] = @YR)
)
[S] ON [HIS].[PID] = [S].[ID]
LEFT JOIN
(
    SELECT [TE], TermStateCode FROM
    (
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 0 [TE], [ST0] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST0] > ' ' GROUP
BY [ST0]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 1 [TE], [ST1] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST1] > ' ' GROUP
BY [ST1]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 2 [TE], [ST2] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST2] > ' ' GROUP
BY [ST2]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 3 [TE], [ST3] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST3] > ' ' GROUP
BY [ST3]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 4 [TE], [ST4] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST4] > ' ' GROUP
BY [ST4]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 5 [TE], [ST5] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST5] > ' ' GROUP
BY [ST5]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 6 [TE], [ST6] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST6] > ' ' GROUP
BY [ST6]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 7 [TE], [ST7] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST7] > ' ' GROUP
BY [ST7]
        UNION ALL
        SELECT ROW_NUMBER() OVER (ORDER BY Count(*) DESC) RowNum, 8 [TE], [ST8] TermStateCode, Count
(*) RcdCount FROM [TDF] WHERE [DEL] = 0 AND ([CD] = ' ' OR [CD] = '') AND [SC] = @SC AND [ST8] > ' ' GROUP
BY [ST8]
    ) [T]
    WHERE RowNum = 1
) [T] ON [HIS].[TE] = [T].[TE]
WHERE [HIS].[DEL] = 0 AND [S].StuCount = 1 AND [S].[SC] = @SC
AND [HIS].[DTS] >= @TimeSince
GROUP BY [HIS].[PID], [HIS].[YR], [HIS].[TE], [S].[CA], [S].[CR], [S].[SC], [T].TermStateCode
```

## Mapping

Parent	Properties	Type	Required	Table	Field	Other	Code / Notes
\	academicHonors	List(Of StudentAcademicRecordAcademicHonor)	Optional				

\	classRanking	StudentAcademicRecordClassRanking	Optional				
classRanking	classRank	String	Optional				
classRanking	classRankingDate	Date	Optional				
classRanking	percentageRanking	Integer	Optional				
classRanking	totalNumberInClass	Integer	Optional				
\	cumulativeAttemptedCreditConversion	Double	Optional				
\	cumulativeAttemptedCredits	Double	<b>Required</b>	STU	CA		GetNullableDouble(s("CA"))
\	cumulativeAttemptedCreditType	String	Optional				
\	cumulativeEarnedCreditConversion	Double	Optional				
\	cumulativeEarnedCredits	Double	<b>Required</b>	STU	CR		GetNullableDouble(s("CR"))
\	cumulativeEarnedCreditType	String	Optional				
\	cumulativeGradePointAverage	Double	Optional				
\	cumulativeGradePointsEarned	Double	Optional				
\	diplomas	List(Of StudentAcademicRecordDiploma)	Optional				
\	educationOrganizationReference	EducationOrganizationReference	Identity				
educationOrganizationReference	educationOrganizationId	Integer	<b>Identity</b>			District ID	OdsConfiguration.DistrictId
educationOrganizationReference	link	Link	Optional				
\	gradeValueQualifier	String	Optional				
\	projectedGraduationDate	Date	Optional				
\	recognitions	List(Of StudentAcademicRecordRecognition)	Optional				
\	reportCards	List(Of StudentAcademicRecordReportCard)	Optional				
\	schoolYearTypeReference	SchoolYearTypeReference	Identity				
schoolYearTypeReference	link	Link	Optional				
schoolYearTypeReference	schoolYear	Integer	<b>Identity</b>	HIS	YR		GetSchoolYearFromInt(s("YR"))
\	sessionAttemptedCreditConversion	Double	Optional				
\	sessionAttemptedCredits	Double	<b>Required</b>	HIS	Sum(HIS.CR)		GetNullableDouble(s("SessionAttemptedCredits"))
\	sessionAttemptedCreditType	String	Optional				
\	sessionEarnedCreditConversion	Double	Optional				
\	sessionEarnedCredits	Double	<b>Required</b>	HIS	Sum(HIS.CC)		GetNullableDouble(s("SessionEarnedCredits"))
\	sessionEarnedCreditType	String	Optional				
\	sessionGradePointAverage	Double	Optional				
\	sessionGradePointsEarned	Double	Optional				
\	studentReference	StudentReference	Identity				
studentReference	link		Optional				
studentReference	studentUniqueId		<b>Identity</b>	HIS	PID		GetString(s("PID"))
\	termDescriptor	String	<b>Required</b>	HIS	TermStateCode		GetEdFiCodeValue(TermDescriptor, "TDF", "ST0", rowHIS!TermStateCode)

Code

```
Public Shared Function GetSchoolYearFromInt(ByVal obj As Object) As Integer?

    Dim schoolYear As Integer? = Nothing
    Dim year As Integer = 0

    If obj IsNot Nothing AndAlso Integer.TryParse(obj, year) Then
        If year < 1000 Then
            ' 2 or 3 digit year i.e. 18 will return 2018 or 118 will return 2118
            schoolYear = 2000 + year
        ElseIf year < 9999 Then
            ' 4 digit year i.e. 2018 so return 2018 (or maybe 2118)
            schoolYear = year
        Else
            ' the data is corrupt so just return Nothing.
        End If
    End If

    Return schoolYear

End Function
```