

Ed-Fi 2.x Course Transcript 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://mappedu.ed-fi.org/#/entity/c210049f-4ed2-e811-80d9-005056b36109/info?dataStandardId=db3697f4-4dd2-e811-80d9-005056b36109>

Scenarios

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

Schema

<https://schema.ed-fi.org/datahandbook-v22/Ed-Fi-UDM-Handbook-Index.html#/CourseTranscript552>

Root

HIS

Join

HIS, TDF, GRC

SQL

Scheduled SQ

HIS_Functions.GetEdFiDataTable_CourseTranscripts

```
SELECT [HIS].[PID], [HIS].[YR], [HIS].[TE], [T].TermStateCode, [S].[CA], [S].[CR], [S].[SC], Sum([HIS].[CR])
SessionAttemptedCredits, Sum([HIS].[CC]) SessionEarnedCredits, [HIS].[CN], [HIS].[CC], [HIS].[ST], [HIS].
[MK], [GRC].[CC] AS CourseAttemptResult
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]
LEFT JOIN [GRC] ON [HIS].[ST] = [GRC].[SC] AND [HIS].[MK] = [GRC].[MK]
WHERE [HIS].[DEL] = 0 AND [S].StuCount = 1 AND [S].[SC] = @SC AND [HIS].[ST] IN (993,994)
GROUP BY [HIS].[PID], [HIS].[YR], [HIS].[TE], [S].[CA], [S].[CR], [S].[SC], [T].TermStateCode, [HIS].[CN],
[HIS].[CC], [HIS].[ST], [HIS].[MK], [GRC].[CC]
```

Real-Time

HIS_Functions.GetEdFiDataTable_CourseTranscripts

```
SELECT [HIS].[PID], [HIS].[YR], [HIS].[TE], [T].TermStateCode, [S].[CA], [S].[CR], [S].[SC], Sum([HIS].[CR])
SessionAttemptedCredits, Sum([HIS].[CC]) SessionEarnedCredits, [HIS].[CN], [HIS].[CC], [HIS].[ST], [HIS].
[MK], [GRC].[CC] AS CourseAttemptResult
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]
LEFT JOIN [GRC] ON [HIS].[ST] = [GRC].[SC] AND [HIS].[MK] = [GRC].[MK]
WHERE [HIS].[DEL] = 0 AND [S].StuCount = 1 AND [S].[SC] = @SC AND [HIS].[ST] IN (993,994)
AND [HIS].[DTS] >= @TimeSince
GROUP BY [HIS].[PID], [HIS].[YR], [HIS].[TE], [S].[CA], [S].[CR], [S].[SC], [T].TermStateCode, [HIS].[CN],
[HIS].[CC], [HIS].[ST], [HIS].[MK], [GRC].[CC]
```

Mapping

Parent	Properties	Type	Required	Table	Field	Other	Code / Notes
--------	------------	------	----------	-------	-------	-------	--------------

\	alternativeCourseCode	String	Optional				
\	alternativeCourseTitle	String	Optional				
\	attemptedCreditConversion	Double	Optional				
\	attemptedCredits	Double	Required	HIS	HIS.CR		GetNullableDouble(s("CR"))
\	attemptedCreditType	String	Optional				
\	courseAttemptResultType	String	Identity	GRC	GRC.CC		attemptPass = GetNullableBoolean(rowHIS!CourseAttemptResult) If(attemptPass = True, "Pass", "Fail")
\	courseReference	CourseReference	(YES)				
courseReference	code	String	Identity	HIS	HIS.CN		GetString(s("CN"))
courseReference	educationOrganizationId	Integer	Identity			District ID	OdsConfiguration. DistrictId
courseReference	whenTakenGradeLevelDescriptor	String	Optional				
\	courseRepeatCodeType	String	Optional				
\	courseTitle	String	Optional				
\	earnedAdditionalCredits	List(Of CourseTranscript EarnedAdditionalCredits)	Optional				
earnedAdditionalCredits[n]	additionalCreditType	String	Optional				
earnedAdditionalCredits[n]	credits	Double	Optional				
\	earnedCreditConversion	Double	Optional				
\	earnedCredits	Double	Required	HIS	HIS.CC		GetNullableDouble(s("CC"))
\	earnedCreditType	String	Optional				
\	finalLetterGradeEarned	String	Required	HIS	HIS.MK		if IsNumeric(rowHIS!MK) = false GetString(rowHIS!MK)
\	finalNumericGradeEarned	Double	Required	HIS	HIS.MK		if IsNumeric(rowHIS!MK) = true GetNullableDouble(rowHIS!MK)
\	methodCreditEarnedType	String	Optional				
\	schoolReference	SchoolReference	Required				
schoolReference	link	Link	Optional				
schoolReference	schoolId	Integer	Required	HIS	HIS.ST		GetNullableInteger(s("ST"))
\	studentAcademicRecordReference	StudentAcademicRecordReference	Identity				
studentAcademicRecordReference	educationOrganizationId	Integer	Identity			District ID	OdsConfiguration. DistrictId
studentAcademicRecordReference	link	Link	Optional				
studentAcademicRecordReference	schoolYear	Integer	Identity	HIS	HIS.YR		GetSchoolYearFromInt(s("YR")) see function below
studentAcademicRecordReference	studentUniqueld	String	Identity	HIS	HIS.PID		GetString(s("PID"))
studentAcademicRecordReference	termDescriptor	String	Identity	HIS	HIS.TE		GetEdFicCodeValue(TermDescriptor, "TDF", "ST0", rowHIS!TermStateCode)
\	whenTakenGradeLevelDescriptor	String	Optional				

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
```