

Train to Gain Success Rates Methodology for 2009/10 – Specification and Supporting Documentation

Date of Publication 16th November 2010

Office National Office

Changes since previous version (Period 12)

- The table below lists any changes made to the specification since the Period 12 reports were compiled.

Period	Description	Reason for Change	Impact of Change	Date
15	Addition of the 90-day period after the expected end date, for end of year reporting	To ensure timely success is reported correctly. ie Period 15 includes achievements up to, and including, Period 3 2010/11 (90 days)	Accurate reporting of timely success, capturing achievements up to 90 days after the expected end date	November 2010
15	Addition of the following variables in the mastertrim file:- prv_type (Provider Type) pst_lea (LEA based on A23 delivery location – ONS postcode file)	Requested by Ofsted	This will not result in any change to the Period 15 QSR provider reports	November 2010

Period	Description	Reason for Change	Impact of Change	Date
15	Addition of the following variables to the mastertrim file:- a_ssa_t1 (Sector Subject Area Tier 1) a_ssa_t2 (Sector Subject Area Tier 2)	For internal purposes	This will not result in any change to the Period 15 QSR provider reports	November 2010

Purpose

- The purpose of this paper is to provide the business rules and the Train to Gain success rates methodology for 2009/10 to the Data Service.

Background

- The Train to Gain success rates methodology and the success rate reporting requirements of the FE sector have changed in the past year as a result of the move to demand-led funding, the new data collection systems put in place by *the information authority* to support that move, and the desire of OfSTED, the LSC, BIS, DCSF and partner organisations to harmonise the different success rate methodologies and reporting methods used across the Train to Gain sector.

Summary

- As part of the Data Harmonisation agenda, a decision was made to align Train to Gain and Apprenticeship success rates for 2009/10. From 2009/10, success in the Train to Gain sector will be measured using the same measures used for the Apprenticeship sector. More detailed information is available on *the Information Authority* [website](#).
- There are two separate measures of success rates: the **overall success rate** and the **timely success rate**. Both are calculated from base 'ILR' data submitted by providers who offer Train to Gain programmes. In broad terms, the success rate of a provider is defined as the proportion of learning aims that were successfully achieved against the number of learning aims that were completed in the relevant year (either successfully or unsuccessfully).

6. The **overall success rate** (based on the **hybridendyr**) measures the proportion of learning aims that are completed at any time, irrespective of when the aim was due to be completed. Specifically the measure is defined:

The proportion of learning aims that have achieved successfully at any time (**p_ach_overall**) / The cohort of completed aims (**p_count_overall**) expressed as a percentage.

7. The **timely success rate** (based on the **expendyr**) measures the proportion of learning aims that are achieved by their planned end date or within 90 days. Those learning aims completed more than 90 days after the planned end date are not counted as successes in the timely measure. Specifically the measure is defined:

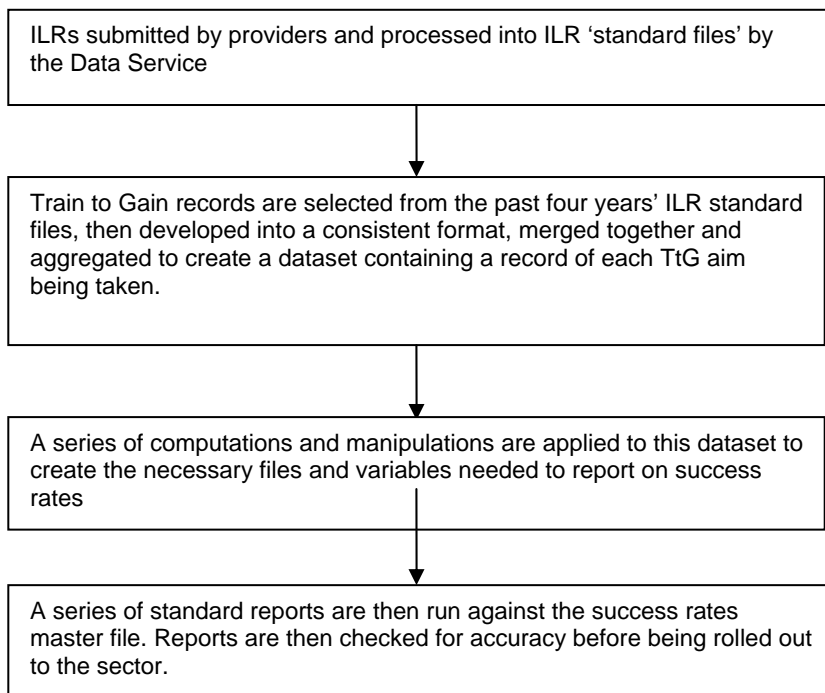
The proportion of programmes that have achieved successfully within 90 days of their planned end date (**p_ach_timely**) / The cohort of learning aims expected to leave (**p_count_timely**) expressed as a percentage.

Exclusions

8. Some learning aims are excluded from success rate calculations:
- Any transfer to a different learning aim within the same provider will be excluded from success rate calculation (**p_trans**).
 - There is a period of six weeks (period of grace) from the start date (**A27**), during which time a learner can leave the learning aim without achieving. These learning aims are excluded from the success rate calculation.
 - Planned breaks are excluded from success rate calculation (**p_plan_break**).

The Success Rates Process

9. The process in broad terms for deriving and reporting success rates is outlined below:



10. This document lays out in detail the process used in taking ILR standard files and manipulating them into a master file on which the standard reports can be run. **Annex A** provides technical detail of how the process is run in SQL. **Annex B** provides a glossary of the variables used in the success rates process. **Annex A also** provides technical detail of how the 2006/07 and 2007/08 historical data is manipulated to be compatible with post-2008/09 collection(s).

Stage 1 – Creating a file that contains a record of each qualification

11. In order to create a dataset that contains a record of each qualification aim, it is necessary to initially treat the data collected before 2008/09 in a separate manner to the way the data collected from 2008/09 onward is treated. This is because of a significant change in the way ILR data is collected. Most notably prior to 2008/09, there was a separate ILR collection for Train to Gain. In addition, FE NVQ's offered by FE colleges but delivered in the workplace are now recorded as Train to Gain programmes where as previously they had been recorded as FE programmes. They are now collected alongside other Employer Responsive data.

Stage 1a – Manipulating the data for 2008/09 onwards to select only Train to Gain aims

12. In order to capture the 90-day timely cap correctly, achievements from Periods 1 to 3 2010/11 will also be included in the post-2008/09 dataset.

13. The only records that need to be kept are the Train to Gain qualification aims. For 2008/09, these are selected from the Employer Responsive Aims file(s) by picking up records where (**a_ttgain**) = 2.

Due to definition changes in the (**a_ttgain**) variable from 2009/10 onwards, ESP aims (**A46a or A46b**) = 083 have been included in addition to where (**a_ttgain**) = 2. This addition effectively applies the 2008/09 definition of (**a_ttgain**), which includes ESP aims.

14. The standard file is matched to the final 2009/10 Learning Aims Database (LAD) using (**A09**) as the match variable. This is to enable reporting on variables such as (**fulllevel2**) and (**fulllevel3**) as they are not in the standard files.
15. It is important to add a variable that shows which academic year 'standard file' the TtG aim is being taken to ensure the latest record for that aim is taken (**In_1011**).
16. A variable (**p_prog_status**) will need to be computed in order to be consistent with other Employer Responsive success rate methodology (Apprenticeships).

(**p_prog_status**) is set to 7 (transfers) unless the following conditions apply.

If learning actual end date (**A31**) is missing or -1 then (**p_prog_status**) is set to 0 (continuing).

If achievement date (**A40**) is greater than 0 then (**p_prog_status**) is set to 1 (achieved).

If learning actual end date (**A31**) is greater than 0 and learning outcome (**A35**) is 4 or 5 (exam taken / assessment completed but result not yet known or learning activities are complete but the exam has not yet been taken and there is an intention to take the exam / assessment) and achievement date (**A40**) is missing then (**p_prog_status**) is set to 3 (aim complete awaiting confirmation of achievement).

If learning actual end date (**A31**) is greater than 0 and achievement date (**A40**) is missing then (**p_prog_status**) is set to 6 (no achievement).

If completion status (**A34**) is 3 (the learner has withdrawn from the learning activities leading to the learning aim) (**p_prog_status**) is set to 13 (withdrawn).

If completion status (**A34**) is 4 or 5 (the learner has transferred to a new learning aim. That is, the learner has withdrawn from this learning aim and as a direct result has at the same time started studying for another learning aim within the same provider or changes in learning within the same programme type and sector subject area / funding category and remaining with the existing provider) (**p_prog_status**) is set to 8 (transfer to a new programme at the same provider).

If reason learning ended (**A50**) is 2 (learner transferred to another employer / provider / local LSC in the same programme type) then (**p_prog_status**) is set to 9 (transfer to new provider).

If reason learning ended (**A50**) is 7 (learner transferred between providers due to intervention by the LSC) then (**p_prog_status**) is set to 10 (transfer to a new provider forced by LSC intervention).

If completion status (**A34**) is 6 (learner has temporarily withdrawn from the aim due to an agreed break in learning) then (**p_prog_status**) is set to 11 (learner is taking a planned break from learning).

The Transact-SQL syntax is laid out in **Annex A**.

17. Once the TtG aims have been selected and laid out in the paragraphs above, the dataset is ready to be added to the 'pre-2008/09' datasets.

Stage 1b – Merging the files from pre and post 2008/09

18. Files from stage 1a can now be added together with pre 08/09 files to create one large dataset which contains a record of every TtG aim (aside from duplicates) recorded in the ILR since 2006/07.

As per paragraph 15, it is important to add a variable that identifies which academic years standard file the TtG aim is coming from.

****PLEASE NOTE:** The 2006/07 and 2007/08 history file is a one off file which has been created from numerous data sources such as TTG ILR standard files and FE ILR standard files (where NVQ's in the workplace used to be recorded) and Train to Gain data collected by the National Employer Service (NES) that was not submitted through the ILR. This file should not be amended. For information, the SQL syntax used to create this file is outlined in **Annex A**.*

Stage 1c – Mergers

19. When two colleges merge, the success rate methodology restates historical data under the new merged college. Where a college has merged with another college, the old college numbers are changed to the new merged college number. This is to allow easier and accurate comparisons across years.

This method currently only affects FE colleges, that also provide Train to Gain provision, that have merged prior to the beginning of the 2009/10 academic year.

The original college number is also retained in the mastertrim file, where appropriate (**L03_orig**). A list of FE colleges providing Train to Gain provision, which have merged, is provided by the Data Service.

Stage 2 – Manipulating the data to create master files that can be used for success rates reporting

20. There are a series of aggregations and calculations that need to be applied to this dataset before the data can be easily manipulated into standard MI reports.

21. As the same aims can be recorded in ILRs across more than one year, the first task is to work through the data to pick up the latest TtG aim record, removing those records which have been superseded.
22. As previous years files are merged the mastertrim flags the source year that the learning aim record has been taken from.
23. It is useful to identify which providers were 'live' in each year – this will identify which providers need to be reported against in which years (**p_live**).
24. Once this process is complete, a series of derived variables are calculated and used either directly or indirectly in the success rates calculations. During these calculations, there are also further manipulations of the dataset (either recoding of variables or dropping of records that are excluded from calculations). These are outlined in Stage 2c – Stage 2i. The SQL syntax in **Annex A** is very useful in understanding the processes that take place here.

Stage 2a – Compute fulllevel2 and 3 variables

25. As the standard files do not calculate the (**fulllevel2**) and (**fulllevel3**) variables required for reporting purposes these variables are derived using the (**a_l2wid**) and (**a_l3wid**) fields. If (**a_l2wid**) >=100 then flag aim as (**fulllevel2**). If (**a_l3wid**) >=100 then flag aim as fulllevel3. See **Annex A** for the SQL calculation.

Stage 2b – Deriving the age band variable

26. The age bands used when reporting success rates are '16-18', '19-24' and '25+'. These are derived from the (**a_agemtb**) variable that defines the age at start of the TtG programme. '16-18' is defined as (**a_agemtb** = 1 or 2), '19-24' is defined as (**a_agemtb**) = 3 or 4 and all other values of (**a_agemtb**) are set to '25+'.

Stage 2c – Identify achievements, transfers within a provider and planned breaks.

27. The key variables in deriving the success rate calculation. A TtG aim has been achieved (**p_achieve**) if the programme status (**p_prog_status**) = 1.
Internal transfer (**p_trans**) records are excluded from success rate calculations. These records can be identified where the programme status (**p_prog_status**) = 7, 8, 10.
Learners on a planned break (**p_plan_break**) are also excluded from success rate calculations. They can be identified where the programme status (**p_prog_status**) = 11.

Stage 2d – Identifying the year the TtG aim was started, when it was completed and when it was planned to be completed

28. Start Year of the TtG aim (**p_startyr**): If the learning start date (**A27**) falls within an academic year then it is assigned the value of the first calendar year of the academic period. For example, if (**A27**) falls within the academic year 2007/08, **p_startyr** has the value of 2007.
29. Actual End Year of the TtG aim (**p_actendyr**): If the learning actual end date (**A31**) falls within an academic year then it is assigned the value of the first calendar year of the academic period. For example, if (**A31**) falls within the academic year 2007/08 **p_actendyr** has the value of 2007.
30. Expected End Year of the TtG aim (**p_expendyr**): If the expected end year (**A28**) falls within an academic year then it is assigned the value of the first calendar year of the academic period. For example, if (**A31**) falls within the academic year 2007/08 **p_expendyr** has the value of 2007.

Stage 2e – Identify learning aims that are recorded in the ILR as being completed after the current reporting period and recalculate the stats of these learning aims

31. Some providers record an actual end date, achieve date, transfer date and planned break of a learning aim which is after the last date of the current period (**p_period_end**). These learning aims are excluded from the success rate calculation.
32. For example, if a learner is to complete in period 9 and success rates are being reported for period 6, the learning aim needs to be flagged so it is not included in the success rate calculation.

Stage 2f – Identify those who have left their learning aim

33. Those learners who have left their learning aim often form part of the count of learners in success rate calculations (**p_leavers**). In initial calculations, a leaver can be identified if they have a date entered in the actual learning end date (**A31**).

Stage 2g - Six week rule

34. Learners on TtG aims have a grace period of six weeks from the start date. Any TtG learning aim which has started (**A27**) and ended (**A31**) within six weeks is excluded from the success rate calculation unless they have achieved the programme.

Stage 2h – Calculating the key variables for the overall success rates measure

35. The overall success rate measure is based on the hybrid end year. The key component of the overall success rate measure is the successful completion of programme aims in the hybrid end year where the learners are flagged as

having finished their learning (leave date in the (A31) field), but are not a transfer or on a planned break.

Stage 2i – Calculating the key variables for the timely success rates measure

36. The timely success rate measure is based on the expected end year. The key component of the timely success rate measure is the successful completion of programme aims on or before the planned end date (or within 90 days of the planned end date) where the learners are flagged as having finished their learning (leave date in the (A31) field), but are not a transfer or on a planned break.
37. A change has been made to capture timely success correctly for year end reporting. For learning aims with an expected end date in period 15 2009/10, achievements are included up to 90 days after that date (2010/11 Periods 1 to 3).

Stage 2j – Tidying up the raw data

38. All the derived fields with 'system missing' or 'null' values need to be set to 0.
39. Compute a year variable for the current year for reporting purposes (2009).

Stage 3 – Preparing the final datasets for success rates reporting

40. Once the derived variables calculations and dataset manipulations have been carried out, there are a few tasks to carry out before the datasets are ready for the success rate reports.

The first step is to match in the provider name into the dataset. The provider name should be taken from the latest year's provider lookup file (eg 'ILR0910_UPIN_TO_LLSC'). Once that step has been taken, the main success rates master file (known as the 'mastertrim' file) is ready.

41. For reporting purposes the dataset can be aggregated up to provider level and split by key variables (**p_hybridendyr**, **p_expendyr**) summing on the remaining key variables (**p_ach_overall**, **p_count_overall**, **p_ach_timely**, **p_count_timely**). It is important to note that the following exclusions also need to be applied:
 - A46a or A46b = 83 (ESP funded aims)
 - **p_trans** = 1 (transfers)
 - **p_plan_break** = 1 (planned breaks)
42. The technical definition for the timely and overall success rate calculation is outlined in **Annex A**.
43. The dataset is now ready for use in standard reporting.

Changes to Learner Reference Numbers between years

44. The learner's reference code is assigned by the provider. The learner reference number stored in the field (**L03**) should be retained by the learner for any period of study with the provider and also during any period of absence. It should not be re-used for a different learner.
45. The learner reference number is used as a key identifier of the learner for data reporting between years and in particular for the calculation of success rates. Changes to the learner reference should be avoided if at all possible between years for continuing learners.
46. If a provider does unavoidably have to change the learner reference numbers used, for example because of a change to their MIS system, they should ensure that the Data Service are informed of this change so that mapping information between the old and new numbers can be obtained.

NOTE: The L03 Lookup is matched to each year in order to identify changes to learner reference numbers between years. The lookup file replaces the original L03 with the new L03 if there is a new L03 in the lookup file.

Additional Guidance

47. Moving to a 90-day cap after the planned end date of a learning aim will result in a data lag for the period of the latest year for which success rates have been calculated. For example, period 6 will not provide a true reflection of the timely success rate until period 9 data is received, in order to take into account the 90-day cap.

To report timely success rates more accurately at year end, reports for period 15 will include data from Periods 1 to 3 of 2010/2011. This will capture achievements that occur within the 90-day period following the planned end date.

NOTE: In order to resolve the definition error relating to FE NVQ's offered by FE colleges but delivered in the workplace a fix has been put in place. This fix has been approved by the Data Harmonisation Group.

Before the final stage in creating the TtG mastertrim if an FE NVQ aim has been flagged as being in 07/08 but not in 08/09 and A18=15 or 16 and the aim is not flagged as a leaver the learning aim is removed.

Annex A - SQL Syntax used to create the variables used in success rate calculations

```

---Set the reporting year

DECLARE @Year_ID int

SET @Year_ID = 2009

***Sort out and combine 0607 FE, NES and TiG Data.

EXEC dbo.uspDropTable 'TTG_0607_F05_Aims_Data'

---Fe 0607 Data, FE in the workplace from 2008/09 considered to be TTG
SELECT      2006 Year_ID
            , CAST('FE' as varchar(3)) DataSource
            , 12 Period
            , LA.101, LA.103, LA.a05, LA.a09, LA.a15, LA.a18, LA.a23,
LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b,
LA.a50, CAST(-1 As int) A63
            , LA.111, LA.112, LA.113, LA.114, LA.115, LA.116, LA.117,
LA.122, LA.125, LA.135, LA.a_agem, LA.a_agemtb, LA.a_ssa_t1
            , LA.a_ssa_t2, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct
            , LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype
            , CASE          WHEN LA.A15 = 3                      THEN
100
                                WHEN LA.A15 = 10                  THEN 0
                                WHEN LAD.A_L2CAT IN('1', '4')    THEN A_L2PCT

            ELSE 0
            END A_L2WID
            , CASE          WHEN LA.A15 = 2                      THEN
100
                                WHEN LA.A15 = 10                  THEN 0
                                WHEN LAD.A_L3CAT IN('1', '2', '3') THEN A_L3PCT

            ELSE 0
            END A_L3WID
            , LA.L_DPLA
INTO  dbo.TTG_0607_F05_Aims_Data
FROM  dbo.ILR0607_F05_AIMS LA
      JOIN  dbo.LAD_0607 LAD ON LAD.A09 = LA.A09
WHERE  LA.A10=20
      AND  LA.A18 IN(12,13,15,16,22,23)
      AND  LA.L_AGEb NOT IN(1,2)

EXEC  dbo.uspDropTable 'TTG_0607_T_Aims_Data'

---TTG 0607 Data
SELECT      2006 Year_ID
            , CAST('TtG' as varchar(3)) DataSource
            , LA.Period

```

```

, LA.l01, LA.l03, LA.a05, LA.a09, LA.a15, LA.a18, LA.a23,
LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b,
LA.a50, CAST(-1 As int) A63
, LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem, LA.a_agemtb, LA.a_ssa_t1
, LA.a_ssa_t2, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct
, LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype
, CASE WHEN LA.A15 = 3 THEN
100
WHEN LA.A15 = 10 THEN 0
WHEN LAD.A_L2CAT IN('1', '4') THEN A_L2PCT
ELSE 0
END A_L2WID
, CASE WHEN LA.A15 = 2 THEN
100
WHEN LA.A15 = 10 THEN 0
WHEN LAD.A_L3CAT IN('1', '2', '3') THEN A_L3PCT
ELSE 0
END A_L3WID
, LA.L_DPLA
INTO dbo.TTG_0607_T_Aims_Data
FROM dbo.ILR0607_T_AIMS LA
JOIN dbo.LAD_0607 LAD ON LAD.A09 = LA.A09
--Fix to remove duplicate records 0607 only
LEFT JOIN ( SELECT L01, L03, A09, A05, Min(Collection)
Collection
FROM dbo.ILR0607_T_AIMS LA
WHERE A_TTGAIN=2
GROUP BY L01, L03, A09, A05
Having Count(*) = 2
AND Max(Collection) !=
Min(Collection)) D ON LA.L01 = D.L01
AND LA.L03 = D.L03
AND LA.A09 = D.A09
AND LA.A05 = D.A05
AND LA.Collection = D.Collection
WHERE LA.A_TTGAIN=2
AND D.Collection IS NULL
EXEC dbo.uspDropTable 'TtG_QSR_0607_NES_Aims_Data_Pre'
--NES 0607 Data
SELECT 2006 Year_ID
, CAST('NES' as varchar(3)) DataSource
, LA.Period
, LA.l01, LA.l03
, LA.a09, A15, NULL A18, LA.a23, LA.a27, LA.a28, LA.a31,
LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b, LA.a50, CAST(-1 As int)
A63
, LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem
, CASE WHEN A_ageSt Between 0 AND 17 THEN 1

```

```

        WHEN A_ageSt = 18 THEN 2
        WHEN A_ageSt In(19,20) THEN 3
        WHEN A_ageSt Between 21 AND 24 THEN 4
        WHEN A_ageSt Between 25 AND 59 THEN 5
        WHEN A_ageSt Between 60 AND 120 THEN 6
    ELSE 9
    END A_ageStB
    , LAD.a_ssa_t1, LAD.a_ssa_t2, CASE WHEN a_pwfc='F' or a_sfl=1
THEN 1 ELSE 0 END A_SFLP
    , CASE
        WHEN A_FUN_ST IN('01','02','03','06') THEN 0
            WHEN A_AWARDB = 'NONE' THEN 1
            WHEN A_AWARDB != 'NONE' THEN 2
    ELSE 0
    END A_OPROV
    , LAD.a_notion
    , 0 a_fe_provmix_matrix_summ
    , LAD.a_l2cat, LAD.a_l2pct
    , LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype
    , CASE
        WHEN LA.A15 = 3 THEN
100
            WHEN LA.A15 = 10 THEN 0
            WHEN LAD.A_L2CAT IN('1','4') THEN A_L2PCT
    ELSE 0
    END A_L2WID
    , CASE
        WHEN LA.A15 = 2 THEN
100
            WHEN LA.A15 = 10 THEN 0
            WHEN LAD.A_L3CAT IN('1','2','3') THEN A_L3PCT
    ELSE 0
    END A_L3WID
    , A_SUBPRG
    , A_FUN_ST
    , A09_TITL
    , 0 A_FE_PROVMIX_MATRIX
    , P.LAUACODE L_DPLA
    , CAST(0 as int) Sequence
INTO dbo.TtG_QSR_0607_NES_Aims_Data_Pre
FROM dbo.NES0607_AIMs LA
    LEFT JOIN dbo.LAD_0607 LAD ON LAD.A09 = LA.A09
    LEFT JOIN dbo.PostCode0607 P ON P.PostCode = LA.A23

CREATE CLUSTERED INDEX IDX_0607_NES_Aims_Data_Pre ON
dbo.TtG_QSR_0607_NES_Aims_Data_Pre(L01, L03, A35, A34)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0607_NES_Aims_Data_Pre',
'Sequence'

EXEC dbo.uspDropTable 'TTG_0607_NES_Data'

--Get the NES 0607 Data (Columns) that we are interested in
SELECT
    Year_ID
    , DataSource
    , LA.Period
    , LA.l01, LA.l03, (LA.Sequence-P2.Sequence)*-1 A05, LA.a09,
LA.a15, LA.a18, LA.a23, LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40,
LA.a44, LA.a46a, LA.a46b, LA.a50, CAST(-1 As int) A63

```

```

, LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem, LA.a_agemtb, LA.a_ssa_t1
, LA.a_ssa_t2, LA.a_sflp, LA.a_oprov, a_notion

, A_FE_PROVMIX_MATRIX A_FE_PROVMIX_MATRIX_SUMM
, a_l2cat, a_l2pct
, a_l3cat, a_l3pct
, a_latype
, A_L2WID
, A_L3WID
, LA.L_DPLA
INTO dbo.TTG_0607_NES_Data
FROM dbo.TtG_QSR_0607_NES_Aims_Data_Pre LA
JOIN ( SELECT L01, L03, MIN(Sequence)-1 Sequence
FROM dbo.TtG_QSR_0607_NES_Aims_Data_Pre
GROUP BY L01, L03) P2 ON LA.L01 = P2.L01
AND LA.L03 =
P2.L03

EXEC dbo.uspDropTable 'TtG_QSR_0607'

--Merge the 0607 FE, TtG and NES Data
SELECT *
--Create matching control fields
, CAST(0 As int) Rank
, CAST(0 as int) Sequence
INTO dbo.TtG_QSR_0607
FROM ( SELECT L.*, ISNULL(M.L03_New, L.L03) L03_New
FROM( SELECT *, L03 L03_orig
FROM dbo.TTG_0607_NES_Data
WHERE L03 IS NOT NULL
UNION
SELECT *, L03 L03_orig
FROM dbo.TTG_0607_T_Aims_Data
UNION
SELECT *, L03 L03_orig
FROM dbo.TTG_0607_F05_Aims_Data) L
--This table holds the L03 Learner Reference no changes that have been
advised to us by providers
LEFT JOIN dbo.ER_L03_Mergers M
ON L.L01 = M.L01

AND L.L03 = M.L03

AND ISNULL(M.L03, '') != ''

AND M.L03 != M.L03_New) LA

CREATE CLUSTERED INDEX IDX_TTG_MT_0607 ON dbo.TtG_QSR_0607(L01, L03_New,
A09, A27, A05 DESC, DataSource DESC)

--Syntax detail provided at the end
EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0607' , 'Sequence'

--Reseed the sequence to start at 1 for all unique occurrences of L01,
L03_New, A09, A27
UPDATE P
SET Rank = P.Sequence-P2.Sequence

```

```

FROM dbo.TtG_QSR_0607 P
      JOIN (
SELECT      L01, L03_New, A09, A27,
MIN(Sequence)-1 Sequence
FROM      dbo.TtG_QSR_0607
GROUP BY L01, L03_New, A09, A27) P2 ON P.L01 =
P2.L01

      AND P.L03_New = P2.L03_New

      AND P.A09 = P2.A09

      AND P.A27 = P2.A27

```

***Sort out and combine 0708 FE, NES and TtG Data.

```
EXEC dbo.uspDropTable 'TTG_0708_F05_Aims_Data'
```

```
--FE 0708 Data
```

```

SELECT      2007 Year_ID
           , CAST('FE' as varchar(3)) DataSource
           , 12 Period
           , LA.l01, LA.l03, LA.a05, LA.a09, LA.a15, LA.a18, LA.a23,
LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b,
LA.a50, CAST(-1 As int) A63
           , LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem, LA.a_agemb, LA.a_ssa_t1
           , LA.a_ssa_t2, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct
           , LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype
           , CASE      WHEN LA.A15 = 3                      THEN
100
                           WHEN LA.A15 = 10                 THEN 0
                           WHEN LAD.A_L2CAT IN('1', '4')   THEN A_L2PCT

      ELSE 0
           END A_L2WID
           , CASE      WHEN LA.A15 = 2                      THEN
100
                           WHEN LA.A15 = 10                 THEN 0
                           WHEN LAD.A_L3CAT IN('1', '2', '3') THEN A_L3PCT

      ELSE 0
           END A_L3WID
           , LA.L_DPLA
INTO      dbo.TTG_0708_F05_Aims_Data
FROM      dbo.ILR0708_F05_AIMS LA
      JOIN dbo.LAD_0708 LAD ON LAD.A09 = LA.A09
WHERE     LA.A10=20
           AND LA.A18 IN(12,13,15,16,22,23)
           AND LA.L_AGEB NOT IN(1,2)

```

```
EXEC dbo.uspDropTable 'TTG_0708_T_Aims_Data'
```

```
--TtG 0708 Data
```

```

SELECT      2007 Year_ID
           , CAST('TtG' as varchar(3)) DataSource
           , LA.Period

```

```

, LA.l01, LA.l03, LA.a05, LA.a09, LA.a15, LA.a18, LA.a23,
LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b,
LA.a50, CAST(-1 As int) A63
, LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem, LA.a_agemtb, LA.a_ssa_t1
, LA.a_ssa_t2, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct
, LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype
, CASE WHEN LA.A15 = 3 THEN
100
WHEN LA.A15 = 10 THEN 0
WHEN LAD.A_L2CAT IN('1', '4') THEN A_L2PCT
ELSE 0
END A_L2WID
, CASE WHEN LA.A15 = 2 THEN
100
WHEN LA.A15 = 10 THEN 0
WHEN LAD.A_L3CAT IN('1', '2', '3') THEN A_L3PCT
ELSE 0
END A_L3WID
, LA.L_DPLA
INTO dbo.TTG_0708_T_Aims_Data
FROM dbo.ILR0708_T_AIMS LA
JOIN dbo.LAD_0708 LAD ON LAD.A09 = LA.A09
WHERE LA.A_TTGAIN=2

EXEC dbo.uspDropTable 'TtG_QSR_0708_NES_Aims_Data_Pre'

--NES 0708 Data
SELECT 2007 Year_ID
, CAST('NES' as varchar(3)) DataSource
, LA.Period
, LA.l01, LA.l03
, LA.a09, A15, NULL A18, LA.a23, LA.a27, LA.a28, LA.a31,
LA.a34, LA.a35, LA.A40, LA.a44, LA.a46a, LA.a46b, LA.a50, CAST(-1 As int)
A63
, LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agem
, CASE WHEN A_agem Between 0 AND 17 THEN 1
WHEN A_agem = 18 THEN 2
WHEN A_agem In(19,20) THEN 3
WHEN A_agem Between 21 AND 24 THEN 4
WHEN A_agem Between 25 AND 59 THEN 5
WHEN A_agem Between 60 AND 120 THEN 6
ELSE 9
END A_agemtb
, LAD.a_ssa_t1, LAD.a_ssa_t2, CASE WHEN a_pwfc='F' or a_sfl=1
THEN 1 ELSE 0 END A_SFLP
, CASE WHEN A_FUN_ST IN('01', '02', '03', '06') THEN 0
WHEN A_AWARDB = 'NONE' THEN 1
WHEN A_AWARDB != 'NONE' THEN 2
ELSE 0
END A_OPROV
, LAD.a_notion
, 0 a_fe_provmix_matrix_summ
, LAD.a_l2cat, LAD.a_l2pct
, LAD.a_l3cat, LAD.a_l3pct, LAD.a_latype

```

```

100          , CASE          WHEN LA.A15 = 3                      THEN
                                WHEN LA.A15 = 10                  THEN 0
                                WHEN LAD.A_L2CAT IN('1', '4')    THEN A_L2PCT
ELSE 0
END A_L2WID
100          , CASE          WHEN LA.A15 = 2                      THEN
                                WHEN LA.A15 = 10                  THEN 0
                                WHEN LAD.A_L3CAT IN('1', '2', '3') THEN A_L3PCT
ELSE 0
END A_L3WID

, 0 A_FE_PROVMIX_MATRIX
, P.LAUACODE          L_DPLA
, CAST(0 as int) Sequence
INTO  dbo.TtG_QSR_0708_NES_Aims_Data_Pre
FROM  dbo.NES0708_AIMs LA
      JOIN  dbo.LAD_0708 LAD ON LAD.A09 = LA.A09
      LEFT JOIN  dbo.PostCode0708 P ON P.PostCode = LA.A23

CREATE CLUSTERED INDEX IDX_0708_NES_Aims_Data_Pre ON
dbo.TtG_QSR_0708_NES_Aims_Data_Pre(L01, L03, A35, A34)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0708_NES_Aims_Data_Pre',
'Sequence'

EXEC dbo.uspDropTable 'TTG_0708_NES_Data'

--NES 0708 Data
SELECT      Year_ID
           , DataSource
           , LA.Period
           , LA.l01, LA.l03, (LA.Sequence-P2.Sequence)*-1 A05, LA.a09,
LA.a15, LA.a18, LA.a23, LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40,
LA.a44, LA.a46a, LA.a46b, LA.a50, CAST(-1 As int) A63
           , LA.l11, LA.l12, LA.l13, LA.l14, LA.l15, LA.l16, LA.l17,
LA.l22, LA.l25, LA.l35, LA.a_agemt, LA.a_agemtb, LA.a_ssa_t1
           , LA.a_ssa_t2, LA.a_sflp
           , LA.a_oprov, a_notion

           , A_FE_PROVMIX_MATRIX A_FE_PROVMIX_MATRIX_SUMM
           , a_l2cat, a_l2pct
           , a_l3cat
           , a_l3pct
           , a_latype
           , A_L2WID
           , A_L3WID
           , LA.L_DPLA
INTO  dbo.TTG_0708_NES_Data
FROM  dbo.TtG_QSR_0708_NES_Aims_Data_Pre LA
      JOIN (
          SELECT      L01, L03, MIN(Sequence)-1 Sequence
          FROM  dbo.TtG_QSR_0708_NES_Aims_Data_Pre
          GROUP BY L01, L03) P2 ON LA.L01 = P2.L01
                                AND LA.L03 =
P2.L03

```

```

EXEC dbo.uspDropTable 'TtG_QSR_0708'

--Merge the 0607 FE, TtG and NES Data
SELECT      *
            , CAST(0 As int) Rank
            , CAST(0 as int) Sequence
INTO  dbo.TtG_QSR_0708
FROM  (
        SELECT      L.*, ISNULL(M.L03_New, L.L03) L03_New
        FROM(
                SELECT      *, L03 L03_orig
                FROM  dbo.TTG_0708_NES_Data
                WHERE L03 IS NOT NULL
                UNION
                SELECT      *, L03 L03_orig
                FROM  dbo.TTG_0708_T_Aims_Data
                UNION
                SELECT      *, L03 L03_orig
                FROM  dbo.TTG_0708_F05_Aims_Data) L
        LEFT JOIN  dbo.ER_L03_Mergers M
ON L.L01 = M.L01

                                                                AND L.L03 =
M.L03

                                                                AND
ISNULL(M.L03, '') != ''

                                                                AND M.L03 !=
M.L03_New) LA

CREATE CLUSTERED INDEX IDX_TTG_MT_0708 ON dbo.TtG_QSR_0708(L01, L03_New,
A09, A27, A05 DESC, DataSource DESC)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0708' , 'Sequence'

UPDATE      P
SET          Rank = P.Sequence-P2.Sequence
FROM  dbo.TtG_QSR_0708 P
        JOIN (
                SELECT      L01, L03_New, A09, A27,
                MIN(Sequence)-1 Sequence
                FROM  dbo.TtG_QSR_0708
                GROUP BY L01, L03_New, A09, A27) P2 ON P.L01 =
P2.L01

                AND P.L03_New = P2.L03_New

                AND P.A09 = P2.A09

                AND P.A27 = P2.A27

--Get 2009 data

EXEC dbo.uspDropTable 'TtG_QSR_0910_ER_Data'

--TtG & ESP 0910 ER Data
SELECT      2009 Year_ID

```

```

, CAST('ER' as varchar(3)) DataSource
, LA.*
, CAST(0 As int) Rank
, CAST(0 as int) Sequence
INTO dbo.TtG_QSR_0910_ER_Data
FROM (SELECT      LA.Period, LA.L01, LA.L03, LA.A05, LA.a09, LA.a15,
LA.a18, LA.a23, LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44,
LA.a46a, LA.a46b, LA.a50, LA.A63, LA.L11
, LA.L12, LA.L13, LA.L14, LA.L15, LA.L16, LA.L17,
LA.L22, LA.L25, LA.L35, LA.a_agemt, LA.a_agemtb, LA.a_ssa_t1, LA.a_ssa_t2
, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct, LAD.a_l3cat
, LAD.a_l3pct, LAD.a_latype, LA.a_l2wid,
LA.a_l3wid
, LA.L03 L03_orig
, ISNULL(M.L03_New, LA.L03) L03_New
, LA.L_DPLA
FROM      dbo.ILR0910_E_AIMS LA
--sp_Tables 'ILR0910_E_AIMS%'
LEFT JOIN dbo.ER_L03_Mergers M ON LA.L01 = M.L01

AND LA.L03 = M.L03

AND ISNULL(M.L03, '') != ''

AND M.L03 != M.L03_New
JOIN      dbo.LAD_0910 LAD ON LAD.A09 = LA.A09
WHERE LA.A_TTGAIN =2
--Added to ensure any changes between 0809 and 0910 to update an aim
to/from ESP are managed
OR 83 IN(ISNULL(LA.A46a, 0), ISNULL(LA.A46b, 0)) LA

CREATE CLUSTERED INDEX IDX_TTG_MT_0910 ON dbo.TtG_QSR_0910_ER_Data(L01,
L03_New, A09, A27, A05 DESC)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0910_ER_Data', 'Sequence'

UPDATE      P
SET          Rank = P.Sequence-P2.Sequence
FROM      dbo.TtG_QSR_0910_ER_Data P
JOIN      (      SELECT      L01, L03_New, A09, A27,
MIN(Sequence)-1 Sequence
FROM      dbo.TtG_QSR_0910_ER_Data
GROUP BY L01, L03_New, A09, A27) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

AND P.A09 = P2.A09

AND P.A27 = P2.A27

EXEC      dbo.uspDropTable 'TtG_QSR_0809_ER_Data'

--TtG 2008 Data
SELECT      2008 Year_ID

```

```

, CAST('ER' as varchar(3)) DataSource
, LA.*
, CAST(0 As int) Rank
, CAST(0 as int) Sequence
INTO dbo.TtG_QSR_0809_ER_Data
FROM (SELECT LA.Period, LA.L01, LA.L03, LA.A05, LA.a09, LA.a15,
LA.a18, LA.a23, LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44,
LA.a46a, LA.a46b, LA.a50, CAST(-1 As int) A63, LA.L11
, LA.L12, LA.L13, LA.L14, LA.L15, LA.L16, LA.L17,
LA.L22, LA.L25, LA.L35, LA.a_agemt, LA.a_agemtb, LA.a_ssa_t1, LA.a_ssa_t2
, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct, LAD.a_l3cat
, LAD.a_l3pct, LAD.a_latype, LA.a_l2wid,
LA.a_l3wid
, LA.L03 L03_orig
, ISNULL(M.L03_New, LA.L03) L03_New
, LA.L_DPLA
FROM dbo.ILR0809_E_Aims LA
LEFT JOIN dbo.ER_L03_Mergers M ON LA.L01 = M.L01
AND LA.L03 = M.L03
AND ISNULL(M.L03, '') != ''
AND M.L03 != M.L03_New
JOIN dbo.LAD_0809 LAD ON LAD.A09 = LA.A09
WHERE LA.A_TTGAIN =2) LA

CREATE CLUSTERED INDEX IDX_TTG_MT_0809 ON dbo.TtG_QSR_0809_ER_Data(L01,
L03_New, A09, A27, A05 DESC)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_0809_ER_Data', 'Sequence'

UPDATE P
SET Rank = P.Sequence-P2.Sequence
FROM dbo.TtG_QSR_0809_ER_Data P
JOIN (SELECT L01, L03_New, A09, A27,
MIN(Sequence)-1 Sequence
FROM dbo.TtG_QSR_0809_ER_Data
GROUP BY L01, L03_New, A09, A27) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

AND P.A09 = P2.A09

AND P.A27 = P2.A27

--Get the data submitted as at 2010/11 period 3 in order to calculate the
Timely 90 day cap accurately
EXEC dbo.uspDropTable 'TtG_QSR_1011_ER_Data'

--TtG & ESP 1011 ER Data
SELECT 2010 Year_ID
, CAST('ER' as varchar(3)) DataSource
, LA.*
, CAST(0 As int) Rank

```

```

, CAST(0 as int) Sequence
INTO dbo.TtG_QSR_1011_ER_Data
FROM (SELECT LA.Period, LA.L01, LA.L03, LA.A05, LA.a09, LA.a15,
LA.a18, LA.a23, LA.a27, LA.a28, LA.a31, LA.a34, LA.a35, LA.A40, LA.a44,
LA.a46a, LA.a46b, LA.a50, LA.A63, L.l11
, L.l12, L.l13, L.l14, L.l15, L.l16, L.l17,
L.l22, -1 l25, L.l35, LA.a_agemt, LA.a_agemtb, LA.a_ssa_t1, LA.a_ssa_t2
, LA.a_sflp, LA.a_oprov, LAD.a_notion,
LA.a_fe_provmix_matrix_summ, LAD.a_l2cat, LAD.a_l2pct, LAD.a_l3cat
, LAD.a_l3pct, LAD.a_latype, LA.a_l2wid,
LA.a_l3wid
, LA.L03 L03_orig
, ISNULL(M.L03_New, LA.L03) L03_New
, L.L_DP_LA L_DPLA
FROM dbo.ILR1011_ER_AIMS LA
JOIN dbo.ILR1011_ER_LEARNER as L ON LA.L01 = L.L01

AND LA.L03 = L.L03
LEFT JOIN dbo.ER_L03_Mergers M ON LA.L01 = M.L01

AND LA.L03 = M.L03

AND ISNULL(M.L03, '') != ''

AND M.L03 != M.L03_New
--Still use the 0910 LAD Definition
JOIN dbo.LAD_0910 LAD ON LAD.A09 = LA.A09
WHERE (LA.A_TTGAIN =2
--Added to ensure any changes between 0809 and 1011 to update an aim
to/from ESP are managed
OR 83 IN(ISNULL(LA.A46a, 0), ISNULL(LA.A46b, 0)))
AND LA.A28 < '1-Aug-2010') LA

CREATE CLUSTERED INDEX IDX_TTG_MT_0910 ON dbo.TtG_QSR_1011_ER_Data(L01,
L03_New, A09, A27, A05 DESC)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_1011_ER_Data', 'Sequence'

UPDATE P
SET Rank = P.Sequence-P2.Sequence
FROM dbo.TtG_QSR_1011_ER_Data P
JOIN (SELECT L01, L03_New, A09, A27,
MIN(Sequence)-1 Sequence
FROM dbo.TtG_QSR_1011_ER_Data
GROUP BY L01, L03_New, A09, A27) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

AND P.A09 = P2.A09

AND P.A27 = P2.A27

-----
-----
-----

```

```

----- Data collected into staging tables and sequenced and Ranked ---
-----
-----
-----

--Mergers
SELECT      *
INTO #MergerList
FROM  dbo.LR_L01_Mergers
WHERE Mrg_Academic_Year < 9
      AND L01 != L01_New

--Ensure multiple mergers i.e. where a upin mergers and then that upin
also merges, are handled
--Generate a row count
SELECT 'Dummy row forced'

--Iterate until no more updates occur
WHILE @@RowCount != 0
BEGIN
    UPDATE      ML2
    SET         ML2.L01_New = ML.L01_New
    FROM #MergerList ML
            JOIN #MergerList ML2 ON ML.L01 = ML2.L01_New

END

--Get the combined data set from all years only interested in the latest
occurrence of an L01, L03, A09, A27
SELECT      L.*
            , CASE      WHEN M.L01 IS NULL      THEN L.L01
                        ELSE
M.L01_New
                        END L01_New
INTO #LA_Merged
FROM (      SELECT      *      FROM  dbo.TtG_QSR_0607  WHERE [Rank] = 1
        UNION
        SELECT      *      FROM  dbo.TtG_QSR_0708  WHERE [Rank]
= 1
        UNION
        SELECT      *      FROM  dbo.TtG_QSR_0809_ER_Data
WHERE [Rank] = 1
        UNION
        SELECT      *      FROM  dbo.TtG_QSR_0910_ER_Data WHERE
Rank = 1
        UNION
        SELECT      *      FROM  dbo.TtG_QSR_1011_ER_Data WHERE
Rank = 1
        ) L
LEFT JOIN #MergerList M ON M.L01 = L.L01
                        AND M.L01 !=
M.L01_New

--Delete duplicate records due to Merger,
--i.e. where the new provider has returned the same L03
--remove the same record submitted by the old L01

```

```

DELETE      M2
FROM      #LA_Merged M
          JOIN #LA_Merged M2 ON      M.Year_ID = M2.Year_ID
          AND M.L01_New =
M2.L01_New
          AND M.L03_New =
M2.L03_New
          AND M.A09 = M2.A09
          AND M.A27 = M2.A27
          AND M.L01 = M.L01_New
          AND M.L03 = M.L03_New

WHERE M2.L01 != M2.L01_New
      OR M2.L03 != M2.L03_New

--Define the Years that the records appear in
SELECT      L01_New L01
          , L03_New L03
          , A09
          , A27
          , Max(CASE WHEN Year_ID = 2006 THEN 1 ELSE 0 END) P_In_0607
          , Max(CASE WHEN Year_ID = 2007 THEN 1 ELSE 0 END) P_In_0708
          , Max(CASE WHEN Year_ID = 2008 THEN 1 ELSE 0 END) P_In_0809
          , Max(CASE WHEN Year_ID = 2009 THEN 1 ELSE 0 END) P_In_0910
          , Max(CASE WHEN Year_ID = 2010 THEN 1 ELSE 0 END) P_In_1011
          , Max(CASE WHEN DataSource = 'NES' AND Year_ID = 2006 THEN 1
ELSE 0 END) In_NES_0607
          , Max(CASE WHEN DataSource = 'NES' AND Year_ID = 2007 THEN 1
ELSE 0 END) In_NES_0708
          , Max(CASE WHEN DataSource = 'FE' AND Year_ID = 2006 THEN 1
ELSE 0 END) In_FE_0607
          , Max(CASE WHEN DataSource = 'FE' AND Year_ID = 2007 THEN 1
ELSE 0 END) In_FE_0708
          , Max(CASE WHEN DataSource = 'TtG' AND Year_ID = 2006 THEN 1
ELSE 0 END) In_TtG_0607
          , Max(CASE WHEN DataSource = 'TtG' AND Year_ID = 2007 THEN 1
ELSE 0 END) In_TtG_0708
          , Max(CASE WHEN DataSource = 'ER' AND Year_ID = 2008 THEN 1
ELSE 0 END) In_ER_0809
          , Max(CASE WHEN DataSource = 'ER' AND Year_ID = 2009 THEN 1
ELSE 0 END) In_ER_0910
          , Max(CASE WHEN DataSource = 'ER' AND Year_ID = 2010 THEN 1
ELSE 0 END) In_ER_1011
          , Max(CASE WHEN DataSource = 'FE' THEN 1 ELSE 0 END) In_NES
INTO      #DataSource
FROM      (SELECT      L.*
          , CASE      WHEN M.L01 IS NULL      THEN
L.L01
          ELSE      M.L01_New
          END L01_New
          FROM      (      SELECT      L01, L03_New, A09, A27,
Year_ID, DataSource      FROM      dbo.TtG_QSR_0607
          UNION
          SELECT      L01, L03_New, A09, A27,
Year_ID, DataSource      FROM      dbo.TtG_QSR_0708
          UNION
          SELECT      L01, L03_New, A09, A27,
Year_ID, DataSource      FROM      dbo.TtG_QSR_0809_ER_Data

```

```

                                UNION
                                SELECT      L01, L03_New, A09, A27,
Year_ID, DataSource      FROM  dbo.TtG_QSR_0910_ER_Data
                                UNION
                                SELECT      L01, L03_New, A09, A27,
Year_ID, DataSource      FROM  dbo.TtG_QSR_1011_ER_Data
                                ) L
                                LEFT JOIN #MergerList M ON M.L01 = L.L01

                                AND M.L01 != M.L01_New
                                ) LA
GROUP BY L01_New
        , L03_New
        , A09
        , A27

--Identify each year the L01 has submitted data
SELECT      L01
        , p_live_0607=max(p_In_0607)
        , p_live_0708=max(p_In_0708)
        , p_live_0809=max(p_In_0809)
        , p_live_0910=max(p_In_0910)
        , Live=max(p_In_0910)

INTO #L01
FROM #DataSource
GROUP BY L01

--Get the latest Period
DECLARE @p_period_end SmallDateTime
DECLARE @p_year_per int
DECLARE @Period int

EXEC dbo.uspDropTable 'TtG_QSR_Data'

SELECT @p_period_end = CAST(CASE      WHEN Period = 1 THEN '1-Sep-' +
CAST(Year_ID as varchar(4))
                                WHEN Period = 2 THEN '1-Oct-' + CAST(Year_ID as
varchar(4))
                                WHEN Period = 3 THEN '1-Nov-' + CAST(Year_ID as
varchar(4))
                                WHEN Period = 4 THEN '1-Dec-' + CAST(Year_ID as
varchar(4))
                                WHEN Period = 5 THEN '1-Jan-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 6 THEN '1-Feb-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 7 THEN '1-Mar-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 8 THEN '1-Apr-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 9 THEN '1-May-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 10 THEN '1-Jun-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period = 11 THEN '1-Jul-' + CAST(Year_ID+1 as
varchar(4))
                                WHEN Period >= 12 THEN '1-Aug-' + CAST(Year_ID+1 as
varchar(4))
                                END as SmallDateTime)

```

```

        , @p_year_per = Year_ID
        , @Period = Period
FROM (
    SELECT      MIN(Period) Period, Max(Year_ID) Year_ID
    FROM      #LA_Merged LA
    WHERE Year_ID = @Year_ID) LA2

SELECT      DateAdd(d,-1,@p_period_end) P_Period_End
        , @Period Period
        , @p_year_per p_year_per
INTO #Period

EXEC dbo.uspDropTable 'TtG_QSR_Data'

--Calculate derived variables
SELECT      LA.*
        , CASE      WHEN A_L2WID>=100 THEN 1 ELSE 0 END A_FULLLEVEL2
        , CASE      WHEN A_L3WID>=100 THEN 1 ELSE 0 END A_FULLLEVEL3
        , CASE      WHEN a_agemtb IN(1,2) THEN 1 --'16-18'
                    WHEN a_agemtb IN(3,4) THEN 2 --'19-24'
                    ELSE 3 --'25+'
        END age_band
        , CASE WHEN p_prog_status = 1
                AND ISNULL(A31, P.P_Period_End) <= DateAdd(d, 90,
P.P_Period_End) THEN 1
                ELSE 0
        END p_achieve
        , CASE WHEN p_prog_status IN(7,8,10) AND A31 <=
P.P_Period_End THEN 1 ELSE 0 END p_Trans
        , CASE WHEN p_prog_status IN(11) AND A31 <= P.P_Period_End
THEN 1 ELSE 0 END p_plan_break
        , CASE WHEN DatePart(m, LA.A27) >= 8      THEN DatePart(yy,
LA.A27)
                WHEN DatePart(m, LA.A27) < 8      THEN
DatePart(yy, LA.A27)-1
        ELSE 0
        END P_StartYr
        , CASE --WHEN A31 > P.P_Period_End      THEN 2009.5
                WHEN DatePart(m, LA.A31) >= 8  THEN DatePart(yy,
LA.A31)
                WHEN DatePart(m, LA.A31) < 8  THEN DatePart(yy,
LA.A31)-1
        ELSE 0
        END P_ActEndYr
        , CASE WHEN A28 > P.P_Period_End      THEN 2009.5
                WHEN DatePart(m, LA.a28) >= 8  THEN DatePart(yy,
LA.a28)
                WHEN DatePart(m, LA.a28) < 8  THEN
DatePart(yy, LA.a28)-1
        ELSE 0
        END P_ExpEndYr
        , P.p_year_per
        , P.P_Period_End p_period_end
        , CASE WHEN a31 IS NOT NULL AND A31 <= P.P_Period_End THEN 1
ELSE 0 END p_leavers

```

```

        , CAST(0 as int) [LastRecord]
--Generate a unique id within each year
        , CAST(0 as int) Sequence2
INTO  dbo.TtG_QSR_Data
FROM  (SELECT
        --See Note A
        , CASE
            WHEN L.A34 = 6
            THEN 11
            WHEN L.A50 = 7
            THEN 10
            WHEN L.A50 = 2
            THEN 9
            WHEN L.A34 IN(4,5)
            THEN 8
            WHEN L.A34 IN(3)
            THEN 13
            WHEN L.A31 IS NOT NULL
            AND CASE WHEN L.A35 = 1 THEN
L.A31 ELSE L.A40 END IS NULL THEN 6
            WHEN L.A31 IS NOT NULL
            AND L.A40 IS NOT NULL
            AND L.A35 IN(3,4)
            THEN 3
            WHEN CASE WHEN L.A35 = 1 THEN L.A31
            ELSE L.A40 END IS NOT NULL THEN 1
            WHEN ISNULL(L.A31, -1) = -1 THEN 0
        ELSE 7
        END P_PROG_STATUS
        FROM  #LA_Merged L) LA
        CROSS JOIN #Period P

CREATE CLUSTERED INDEX IDX_ER_TtG_Data ON  dbo.TtG_QSR_Data(L01_New,
L03_New, A09, A27, Year_ID DESC)

EXEC [usp_CreateIdentityColumn] 'TtG_QSR_Data' , 'Sequence2'

UPDATE  P
SET      [LastRecord] = P.Sequence2-P2.Sequence2
FROM  dbo.TtG_QSR_Data P
        JOIN (
        SELECT      L01_New, L03_New, A09, A27,
MIN(Sequence2)-1 Sequence2
        FROM  dbo.TtG_QSR_Data
        GROUP BY L01_New, L03_New, A09, A27) P2 ON
P.L01_New = P2.L01_New

        AND P.L03_New = P2.L03_New

        AND P.A09 = P2.A09

        AND P.A27 = P2.A27

--Remove records where the Learner has left within six weeks without an
achievement
DELETE
FROM  dbo.TtG_QSR_Data
WHERE datediff(d, a27,a31) < 6*7
        AND p_achieve = 0

```

```

If Object_ID ('TtG_MasterTrim_0910', 'U') IS NOT NULL
    DROP Table TtG_MasterTrim_0910

SELECT      '2009/10' [Year], (SELECT Period FROM #Period) Period
           , D.L01_New L01, D.L01 L01_Orig, U.prv_name, U.Prv_type,
D.L03_New L03, D.L03_Orig, D.A09, D.A18, D.A23, D.A27, D.A28, D.A31,
D.A34, D.A35, D.A44, D.A46a, D.A46b, ISNULL(D.A50, -1) A50, D.A63
           , D.L11, D.L12, D.L13, D.L14, D.L15, D.L16, D.L25, D.L35,
D.L17, D.L22, D.a_ssa_t1, D.a_ssa_t2
           , ISNULL(D.a_sflp, 0) a_sflp, D.p_prog_status, D.a_oprov,
D.a_notion, D.a_fe_provmix_matrix_summ
           , ISNULL(D.a_l2cat, 0) a_l2cat
           , ISNULL(D.a_l2pct, 0) a_l2pct
           , ISNULL(D.a_l3cat, 0) a_l3cat
           , ISNULL(D.a_l3pct, 0) a_l3pct
           , D.a_latype, D.a_l2wid, D.a_l3wid
           , D.a_agemt, D.a_agemtb, D.age_band
           , D.a_fulllevel2, D.a_fulllevel3
           , DS.in_nes_0607, DS.in_nes_0708, DS.in_fe_0607,
DS.in_fe_0708, DS.in_ttg_0607, DS.in_ttg_0708, DS.in_er_0809,
DS.in_er_0910
           , DS.p_in_0910, DS.p_in_0809, DS.p_in_0708, DS.p_in_0607
           , DL.p_live_0607, DL.p_live_0708, DL.p_live_0809,
DL.p_live_0910, DL.live
           , D.p_leavers, D.p_achieve, D.p_trans, D.p_plan_break
           , D.p_startyr, D.p_expendyr, D.p_actendyr
           , CASE WHEN p_expendyr >= p_actendyr THEN p_expendyr ELSE
p_actendyr END p_hybridendyr
           , CASE WHEN p_leavers =1
                   AND p_trans=0
                   AND p_plan_break=0
                   AND CASE WHEN p_expendyr >= p_actendyr
THEN p_expendyr
                   ELSE p_actendyr END BETWEEN 2004 AND 2009 THEN 1
                   ELSE 0 END P_COUNT_OVERALL
           , CASE WHEN p_leavers =1
                   AND p_trans=0
                   AND p_plan_break=0
                   AND CASE WHEN p_expendyr >= p_actendyr
THEN p_expendyr
                   ELSE p_actendyr END BETWEEN 2004 AND 2009 THEN
P_ACHIEVE
                   ELSE 0 END P_ACH_OVERALL
           , CASE WHEN p_trans=0
                   AND p_plan_break=0
                   AND p_expendyr BETWEEN 2004 AND 2009 THEN 1
                   ELSE 0 END P_COUNT_TIMELY
           , CASE WHEN p_trans=0
                   AND p_plan_break=0
                   AND p_expendyr BETWEEN 2004 AND 2009
                   AND DateDiff(d,D.A28,D.A31) <= 90
                   THEN P_ACHIEVE
                   ELSE 0 END P_ACH_TIMELY
           , D.p_year_per, D.p_period_end
           , D.L_DPLA

```

```

INTO TtG_MasterTrim_0910
FROM dbo.TtG_QSR_Data D
      JOIN dbo.ILR0910_UPIN_TO_LLSC U ON D.L01_New = U.L01
      JOIN #DataSource DS ON D.L01_New = DS.L01
                                AND D.L03_New = DS.L03
                                AND D.A09 = DS.A09
                                AND D.A27 = DS.A27
      JOIN #L01 DL ON DL.L01 = D.L01_New
WHERE CASE WHEN In_FE_0708 = 1 and In_ER_0809 = 0 and A18 in(15,16) and
p_leavers=0 THEN 1 ELSE 0 END = 0
--Only include the latest record from the latest return
      AND [LastRecord] = 1

```

Note A:

```

/*
Variable labels P_PROG_STATUS 'P_PROG_STATUS Status of Programme'.
Value labels P_PROG_STATUS -1 "Not Applicable/ Not Known"
  0 "Continuing"
  1 "Programme achieved"
  3 "Programme complete awaiting confirmation of achievements"
  6 "No achievement"
  7 "Transferred"
  8 "Transfer to a new programme at the same provider"
  9 "Transfer to a new provider"
 10 "Transfer to a new provider forced by LSC intervention"
 11 "Learner is taking a planned break from learning"
 13 "Withdrawn".
*/

```

Annex B – Glossary of variables used in success rate calculations

L01 – Provider Number

L03 – Learner reference number

A09 – Learning Aim Reference

A23 – Delivery location postcode

A27 – Learning Start Date

A28 – Learning planned end date

A31 – Learning actual end date

A34 – Completion Status

A35 – Learning Outcome

A44 – Employer Identifier

A46a – National Learning Aim Monitoring

A46b – National Learning Aim Monitoring

A63 – National Skills Academy (from 09/10 onwards)

A50 – Reason Learning Ended

A34 – Completion Status

A AGESTB – Age of the learner as at the start date of aim banded

A LEAVEP – Period left in

P PROG STATUS – To determine status at programme level, for example, “framework achieved”

PERIOD – Period data contains

L11 – Date of birth

L12 – Ethnicity

L13 – Sex

L14 – Learning difficulties and / or disabilities and / or health problems

L15 – Disability

L16 – Learning Difficulty

L25 – LSC

L35 – Prior Attainment Level

L17 – Home postcode

L22 – Current Postcode

A SSA T1 – SSA Tier 1 Code

A SSA T2 – SSA Tier 2 Code

A SFLP – Skills for Life participation flag

A OPROV – DV – Is the aim other provision

A NOTION – Notional NVQ level

A FE PROV MIX MATRIX SUMM – CFE provision mix matrix summary

A L2CAT – Full level 2 category

A L2PCT – Full level 2 %

A L3CAT – Full level 3 category

A L3PCT – Full level 3 %

A LATYPE – Learning Aim Type

A L2WID – Level 2 width for full level 2 and GCSE aims

A L3WID – Level 3 width for full level 3 and AS A2 aims

A AGEST – Age of learner at the start date of the aim

In NES 0708 – Learner is in the 07/08 NES dataset

In NES 0607 – Learner is in the 06/07 NES dataset
In FE 0708 – Learner is in the 07/08 FE dataset
In FE 0607 – Learner is in the 06/07 FE dataset
In TtG 0708 – Learner is in the 07/08 TtG dataset
In TtG 0607 – Learner is in the 06/07 TtG dataset
In ER 0809 – Learner is in the 08/09 ER dataset
In ER 0910 – Learner is in the 09/10 ER dataset
P IN 0910 – Learner is in the 09/10 dataset
P IN 0809 – Learner is in the 08/09 dataset
P IN 0708 – Learner is in the 07/08 dataset
P IN 0607 – Learner is in the 06/07 dataset
P LIVE 0910 – Provider is live in 09/10
P LIVE 0809 – Provider is live in 08/09
P LIVE 0708 - Provider is live in 07/08
P LIVE 0607 - Provider is live in 06/07
LIVE - Provider is live in the current year
PERIOD – Period collection date contains
A FULLLEVEL2 – Full level 2 aim
A FULLEVEL3 – Fulllevel 3 aim
P ACHIEVE - If p_prog_status is set to 1, then record is flagged as achieved.
P TRANS - If p_prog_status is set to 7,8,10 or 12 then record is flagged as a transfer. Excluded from success rate calculations.
P PLAN BREAK - If p_prog_status is set to 11 or 12, then record is flagged as planned break. Excluded from success rate calculations.
P STARTYR - Variable to flag the year the record started the framework
P ACTENDYR – Variable to flag the year the record actually ended the framework
P EXPENDYR - Variable to flag the year the record is expected. This is equal to the startyr.
AGE BAND – A_agestb variable banded into 16-18, 19-24 and 25+ for reporting purposes
P YEAR PER - Identify leavers who have left after the end date of the current period, to exclude them from the success rate calculation.
P PERIOD END – Excludes anyone who has left after the current year
P LEAVERS – If field A31 (learning actual end date) is not missing then flag the learner as a leaver (leaver =1). If not then learner is continuing (leaver =0).
P HYBRIDENDYR - Maximum of expendyr or actendyr
P COUNT OVERALL - If hybridendyr is in the past 5 years and record is flagged as a leaver, not as a transfer or on a planned break then the record is counted towards overall success rate (cohort is leavers – trans – planned breaks).
P ACH OVERALL – Learning aim achieved overall is equal to aim achieved in the cohort of overall measure
P COUNT TIMELY - If expendyr is in the past 5 years and record is not flagged as a transfer or on a planned break then the record is counted towards timely success rate (cohort is leavers + continuers – trans – plan breaks).
P ACH TIMELY – Learning aim achieved timely is equal to aim achieved in the cohort of timely measure
PRV NAME – Provider name
L01 Orig – Original L01 before L01 Lookup (change in provider UPIN) is applied

L03 Orig – Original L03 before L03 Lookup (change in learner reference) is applied

PRV TYPE – Provider Type

PST LEA – Local Authority based on A23 Delivery Location postcode – ONS postcode file