

Apprenticeship 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)

1. 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 2009/10 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	Correction to ensure where learners have returned from a planned break, the restarted aim is included in the success rate calculation	To fix a known issue in some cases where more than one framework record exists, planned break aims were previously being selected instead of the restarted aim	In these cases, the restarted aim will correctly be included in the mastertrim file instead of the planned break aim. Planned break and transferred aims will still be included in the mastertrim file, but continue to be	November 2010

			excluded from success rate calculations	
15	<p>Addition of the following variables to the mastertrim file:-</p> <p>prv_type (Provider Type)</p> <p>pst_lea (LEA based on A23 delivery location – ONS postcode file)</p>	Requested by Ofsted	This will not result in any change to the Period 15 QSR provider reports	November 2010
15	<p>Addition of the following variables to the mastertrim file:-</p> <p>a_ssa_t1 (Sector Subject Area Tier 1)</p> <p>a_ssa_t2 (Sector Subject Area Tier 2)</p>	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 Apprenticeship success rates methodology for 2009/10 to the Data Service.

Background

- The Apprenticeship success rates methodology and the success rate reporting requirements of the FE sector have changed from 2008/09 onwards 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 Apprenticeships sector.

Summary

4. 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, the 'timely' element of the timely success rate has moved from a 31 to 90-day cap after the planned end date. More detailed information is available on *the Information Authority website*.
5. There are two separate measures of success rates used in the Apprenticeship sector: the **overall success rate** and the **timely success rate**. Both are calculated from base 'ILR' data submitted by providers who offer Apprenticeship programmes. In broad terms, the success rate of a provider is defined as the proportion of learners who achieved an Apprenticeship against those who completed the Apprenticeship in the relevant year (either successfully or unsuccessfully).
6. The **overall success rate** (based on the **hybridendyr**) measures the proportion of Apprenticeships that are completed at any time, irrespective of when the Apprenticeship was due to be completed. Specifically the measure is defined:

The proportion of programmes that have achieved successfully at any time (**p_frm_ach_overall**) / The cohort of leavers who have finished their learning (**p_count_overall**) expressed as a percentage
7. The **timely success rate** (based on the **expendyr**) measures the proportion of Apprenticeships that are achieved by their planned end date or within 90 days. Those Apprenticeships completed more than 90 days after the planned end date are not counted as successes in the timely measure. Specifically the measure is defined:

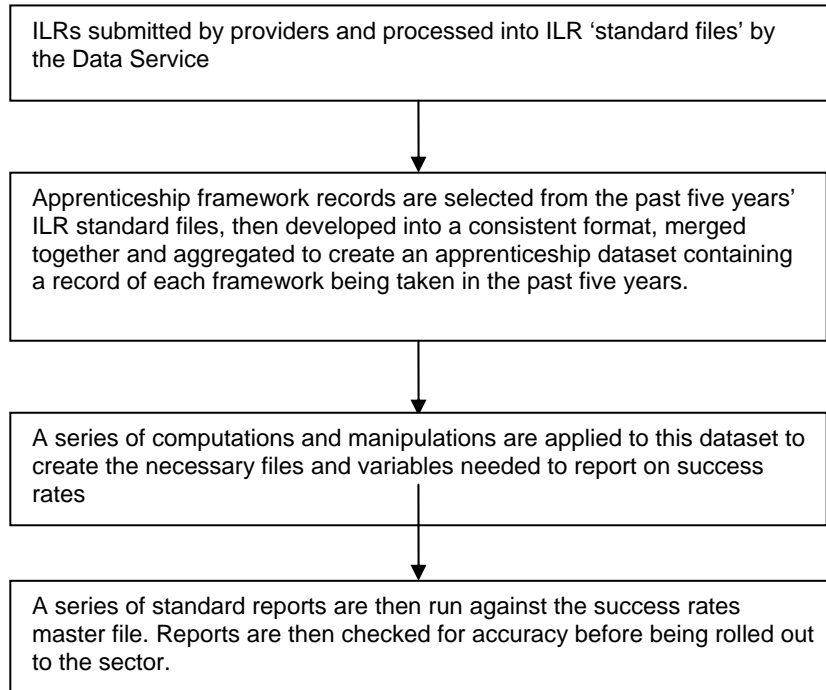
The proportion of programmes that have achieved successfully within 90 days of their planned end date (**p_frm_ach_timely**) / The cohort of learning aims expected to leave (**p_count_timely**) expressed as a percentage.

Exclusions

8. Learners excluded from success rate calculations:
 - Any transfer to a different programme 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 programme without achieving and be 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.

Stage 1 – Creating a file that contains a record of each programme aim

11. In order to create a dataset that contains a record of each Apprenticeship programme 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, the ILR did not collect an Apprenticeship framework record – this record had to be derived during the ILR 'standard file' process. In addition, Apprenticeship data is now collected alongside other Employer Responsive data, such as Train to Gain, which had previously been collected through a separate collection.

Stage 1a – Manipulating the data from 2008/09 onwards to select only programme 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 Apprenticeship programme aims. These are selected from the Employer Responsive Aims file(s) by picking up records where (**A04 = 35** and **A15 = 2, 3 or 10**).
14. It is important to add a variable that shows which academic year 'standard file' the framework record is being taken from (For example, **In_0809**).
15. It is possible for multiple programme aims to be included in ILR returns in the same year. It is necessary to group records so that for each learner (**L03**) in a provider (**L01**) there is only one record for each programme type (**A15**) and sector framework (**A26**) for each academic year (from 2008/09 onwards). In cases where there are multiple records, there are a set of rules that need to be applied to make sure the correct values of other fields are carried through to the master file. This is laid out in **Annex A**.
16. Once the programme aims have been selected and grouped as laid out in the three paragraphs above, the 2008/09, 2009/10 and 2010/11 datasets are ready to be added to the 'pre-2008/09' datasets produced in stage 1b below.

Stage 1b – Manipulating the data collected before 2008/09 to select only framework records

****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.***

The L03 lookup file is maintained and updated by the Data Service.

17. Similar to Stage 1a, only Apprenticeship framework records need to be kept. However as the framework records are derived through the ILR standard files process rather than collected, there is no need to aggregate the data to remove duplicate records. The framework records are selected from the Work-Based Learning (WBL) framework and aims files by picking up records where (**A10**) =50.
18. It is necessary to pick up at least five years' worth of records. So for the 2009/10 academic year add together framework records from the 2004/05, 2005/06, 2006/07 and 2007/08 framework and aims files in stage 1b.
19. It is important to add a variable that shows which academic year 'standard file' the framework record is being taken from (**In_0708**).
20. The variable (**a_status**) held in these files has a different name in the ILR standard files for 2008/09 onwards. It is now called (**p_prog_status**).

21. Once the framework records from each of the previous years in scope have been added together, the next step is to add the files from stage 1a and stage 1b together. Note that the variables that need to be kept at the end of stage 1b are the same as those in stage 1a.

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

22. Files from stage 1a and stage 1b can now be added together to create one large dataset which contains a record of every framework aim (aside from duplicates) recorded in the ILR over the past five years.

Stage 1d – Mergers

23. 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, which also provide Apprenticeship 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 Apprenticeship 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

24. There are a series of aggregations and manipulations that need to be applied to this framework dataset before the data can be easily manipulated into standard MI reports.
25. As the same Apprenticeship framework record can be recorded in ILRs across more than one year, the first task is to work through the data to pick up the latest framework record, removing those records which have been superseded.
26. In cases where a learner has returned from a planned break, it is essential that the restarted aim is included in the mastertrim file, and not the planned break aim.

The variable (**priority**) ensures this, by reordering a learner's framework records where more than one exists. Subsequently, the record with the higher (**A05**) value will be included in the mastertrim file. Any planned break and transferred aims will only be included in the mastertrim file if a learner has no other framework records.

27. It is useful at this point to also identify which providers were 'live' in each year – this is useful in identifying which providers need to be reported against in which years (**p_live**).
28. 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 2a – Stage 2j. The SQL syntax in **Annex A** is very useful in understanding the processes that take place here.

Stage 2a – Deriving the age band variable

29. 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 Apprenticeship 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 2b – Identify achievements, transfers within a provider and planned breaks.

30. These are key variables in deriving the final variables used in the overall and timely success rate calculations summarised in paragraphs 5 and 6. A framework has been achieved (**p_frm_ach**) if the programme status (**p_prog_status**) = 1.

Internal transfer records are excluded from success rate calculations (**p_trans**). These records can be identified where the programme status (**p_prog_status**) = 7, 8, 10.

Planned break aims are also excluded from success rate calculations (**p_plan_break**). They can be identified where the programme status (**p_prog_status**) = 11 or 12.

Stage 2c – Identifying the year the Apprenticeship was started, when it was completed and when it was planned to be completed

31. Start Year of the Apprenticeship (**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.
32. Actual End Year of the Apprenticeship (**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.
33. Expected End Year of the Apprenticeship (**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 2d – Identify those who have left their Apprenticeship

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

Stage 2e – Identify the last day in each month and recalculate leavers and expected end year

35. Identify leavers whose actual end date (A31) is after the last date of the current period. These learners are excluded from the success rate calculation.
36. For example, if a learner is to complete in period 9 and we are currently in period 6, the learner needs to be flagged so are not included in the success rate calculation.

Stage 2f – Calculating the hybrid end year

37. The overall success measure is based on the hybrid end year. This is the maximum of the actual learning end year (**p_actendyr**) or the planned learning end year (**p_expendyr**).

Stage 2g - Six week rule

38. Learners doing Apprenticeships have a grace period of six weeks from the start date. Any programme aim which has started (A27) and ended (A31) within six weeks is excluded from the success rate calculation unless they have achieved the framework.

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

39. 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

40. 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.

Stage 2j – Tidying up the raw data

41. All the derived fields with system missing variable need to be set to 0.
42. Compute a year variable for the current year for reporting purposes (**2009**).

Stage 3 – Preparing the final datasets for success rates reporting

43. 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.
44. 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'). Note that it is not necessary to match in provider names for those providers who are not 'live' in the latest year as the LSC does not need to report on those providers.

Once that step has been taken, the main success rates master file (known as the 'mastertrim' file) is ready.

45. For reporting purposes the dataset can be aggregated to provider level and key variables (**p_hybridendyr, p_expendyr, a15, age_band, a26**) summing on the remaining key variables **p_frm_ach, p_trans, p_leavers, p_plan_break, p_frm_ach_overall, p_count_overall, p_frm_ach_timely, p_count_timely**.
46. The technical definition for the timely and overall success rate calculation is outlined in **Annex A**.
47. The dataset is now ready for use in standard reporting.

Changes to Learner Reference Numbers between years

48. 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.
49. 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.
50. 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.

Additional Guidance

51. 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 9 will not provide a true reflection of the timely success rate until period 12 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: The L01 field for all learning aims before 01/04/08 for upin 110023 have been set to (-1) as per request from the Data Service, and excluded from the mastertrim file.

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

```
EXEC          dbo.uspDropTable 'APPS_QSR_1011'

--Get the data submitted as at 2010/11 period 3 in order to calculate the
Timely 90 day cap accurately
SELECT        Year_ID = cast(2009 as int),
              AC_Year = cast('2010' as varchar(50)),
              L01 = cast(AIMS.L01 as int),
              L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),
              L01_Orig = cast(AIMS.L01 as int),
              L03_Orig = cast(AIMS.L03 as varchar(50)),
              L11 = cast(L.L11 as datetime),
              L.L12,
              L.L13,
              L.L14,
              L.L15,
              L.L16,
              L.L17,
              L.L22,
              -1 L25,
              '-1' L44,
              AIMS.A04,
              AIMS.A05,
              AIMS.A09,
              AIMS.A10,
              AIMS.A15,
              AIMS.A16,
              AIMS.A23,
              AIMS.A34,
              AIMS.A35,
              AIMS.A40,
              AIMS.A44,
              AIMS.A49,
              ISNULL(AIMS.A26, -1) A26,
              AIMS.A27,
              AIMS.A28,
              AIMS.A31,
              AIMS.A46a,
              AIMS.A46b,
              AIMS.A63,
              AIMS.a_agestb,
              AIMS.a_leavep,
              AIMS.A50,
              L.L39,
              AIMS.p_prog_status,
              15 period,
              L.l_age,
              Case WHEN AIMS.A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
              Sequence = cast(0 as int),
              Rnk = cast(0 as int)
INTO        dbo.APPS_QSR_1011
FROM        dbo.ILR1011_ER_AIMS as AIMS
           JOIN dbo.ILR1011_ER_LEARNER as L ON AIMS.L01 = L.L01
           AND AIMS.L03 = L.L03
```

```

--This table hold the L03 Learner Reference no changes that have been
advised to us by providers
LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01
                                AND AIMS.L03 = M.L03

                                AND ISNULL(M.L03,

'' ) != ''

                                AND M.L03 !=

M.L03_New
--Only table the Programme Level Apprenticeship Data
WHERE AIMS.A04 = 35
      and AIMS.A15 in (2,3,10)
--Exclude aims started prior to 01-04-2008 for UPIN 110023 due to an
issue with issuing this upin
      and not (aims.L01 = 110023
              and AIMS.a27 < convert(datetime,'01-04-2008',
105))
--Only include aims with a planned end date prior to 2010/11
      AND AIMS.a28 < '1-Aug-2010'

--Index for performance
CREATE CLUSTERED INDEX IDX_APPS_QSR_1011 ON dbo.APPS_QSR_1011(L01,
L03_New, A15, A26, Priority, A05 desc)

--Syntax detail provided at the end
EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_1011', 'Sequence'

--Reseed the sequence to start at 1 for all unique occurances of L01,
L03_New, A15, A26
UPDATE P
SET Rnk = P.Sequence-P2.Sequence
FROM  dbo.APPS_QSR_1011 P
      JOIN ( SELECT L01, L03_New, A15, A26, MIN(Sequence)-1
Sequence
          FROM  dbo.APPS_QSR_1011
          GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 = P2.L01
                                AND P.L03_New
= P2.L03_New
                                AND P.A15 =
P2.A15
                                AND P.A26 =
P2.A26

EXEC dbo.uspDropTable 'APPS_QSR_0910'

--Get data returned in 2009/10
SELECT Year_ID = cast(2009 as int),
       AC_Year = cast('2009' as varchar(50)),
       L01 = cast(AIMS.L01 as int),
       L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

       L01_Orig = cast(AIMS.L01 as int),
       L03_Orig = cast(AIMS.L03 as varchar(50)),
       L11 = cast(L11 as datetime),
       L12,
       L13,
       L14,

```

```

L15,
L16,
L17,
L22,
L25,
L44,
A04,
A05,
A09,
A10,
A15,
A16,
A23,
A34,
A35,
A40,
A44,
A49,
ISNULL(A26, -1) A26,
A27,
A28,
A31,
A46a,
A46b,
A63,
a_agestb,
a_leavep,
A50,
L39,
p_prog_status,
period,
l_age,
Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
Sequence = cast(0 as int),
Rnk = cast(0 as int)
INTO dbo.APPS_QSR_0910
FROM dbo.ILR0910_E_AIMS as AIMS
LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01
AND AIMS.L03 =
M.L03
AND ISNULL(M.L03,
'' ) != ''
AND M.L03 !=
M.L03_New
WHERE A04 = 35
and a15 in (2,3,10)
and not ( aims.L01 = 110023
and a27 < convert(datetime,'01-04-2008', 105))

CREATE CLUSTERED INDEX IDX_APPS_QSR_0910 ON dbo.APPS_QSR_0910(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0910', 'Sequence'

UPDATE P
SET Rnk = P.Sequence-P2.Sequence
FROM dbo.APPS_QSR_0910 P

```

```

        JOIN (
Sequence      SELECT      L01, L03_New, A15, A26, MIN(Sequence)-1
                FROM    dbo.APPS_QSR_0910
                GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 = P2.L01
                AND P.L03_New
= P2.L03_New
                AND P.A15 =
P2.A15
                AND P.A26 =
P2.A26

EXEC dbo.uspDropTable 'APPS_QSR_0809'

SELECT      Year_ID = cast(2008 as int),
            AC_Year = cast('2008' as varchar(50)),
            L01 = cast(AIMS.L01 as int),
            L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

            L01_Orig = cast(AIMS.L01 as int),
            L03_Orig = cast(AIMS.L03 as varchar(50)),
            L11 = L11,
            L12,
            L13,
            L14,
            L15,
            L16,
            L17,
            L22,
            L25,
            L44,
            A04,
            A05,
            A09,
            A10,
            A15,
            A16,
            A23,
            A34,
            A35,
            A40,
            A44,
            A49,
            ISNULL(A26, -1) A26,
            A27,
            A28,
            A31,
            A46a,
            A46b,
            A63 = null,
            a_agemtb,
            a_leavep,
            A50,
            L39,
            p_prog_status,
            period,
            l_age,
            Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
            Sequence = cast(0 as int),

```

```

        Rnk = cast(0 as int)
INTO  dbo.APPS_QSR_0809
FROM  dbo.ILR0809_E_AIMS as AIMS
        LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01

AND AIMS.L03 = M.L03

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

AND M.L03 != M.L03_New
WHERE A04 = 35
        and a15 in (2,3,10)
        and not (aims.L01 = 110023 and a27 < convert(datetime, '01-04-
2008', 105))

CREATE CLUSTERED INDEX IDX_APPS_QSR_0809 ON dbo.APPS_QSR_0809(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0809', 'Sequence'

UPDATE      P
SET         Rnk = P.Sequence-P2.Sequence
FROM        dbo.APPS_QSR_0809 P
            JOIN (      SELECT          L01, L03_New, A15, A26,
MIN(Sequence)-1 Sequence
                        FROM    dbo.APPS_QSR_0809
                        GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

AND P.A15 = P2.A15

AND P.A26 = P2.A26

EXEC dbo.uspDropTable 'APPS_QSR_0708'

SELECT      Year_ID = cast(2007 as int),
AC_Year = cast('2007' as varchar(50)),
L01 = cast(AIMS.L01 as int),
L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

L01_Orig = cast(AIMS.L01 as int),
L03_Orig = cast(AIMS.L03 as varchar(50)),
L11 = L11,
L12,
L13,
L14,
L15,
L16,
L17,
L22,
L25,
L44,
A04,
A05,

```

```

        A09,
        A10,
        A15,
        A16,
        A23,
        A34,
        A35,
        A40,
        A44,
        A49,
        ISNULL(A26, -1) A26,
        A27,
        A28,
        A31,
        A46a,
        A46b,
        A63 = null,
        a_agemtb,
        a_leavep,
        A50,
        L39,
        p_prog_status = a_status,
        period,
        l_age,
        Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
        Sequence = cast(0 as int),
        Rnk = cast(0 as int)
INTO  dbo.APPS_QSR_0708
FROM  dbo.ILR0708_W_FRAMEWORKS_AND_AIMS as AIMS
      LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01
                                           AND AIMS.L03 = M.L03
                                           AND ISNULL(M.L03, '')
                                           AND M.L03 !=
M.L03_New
WHERE a10 = 50
      and not (aims.L01 = 110023 and a27 < convert(datetime, '01-04-
2008', 105))

CREATE CLUSTERED INDEX IDX_APPS_QSR_0708 ON dbo.APPS_QSR_0708(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0708', 'Sequence'

UPDATE      P
SET         Rnk = P.Sequence-P2.Sequence
FROM        dbo.APPS_QSR_0708 P
           JOIN (      SELECT      L01, L03_New, A15, A26,
MIN(Sequence)-1 Sequence
                FROM        dbo.APPS_QSR_0708
                GROUP BY    L01, L03_New, A15, A26) P2 ON P.L01 =
P2.L01
                                           AND
P.L03_New = P2.L03_New
                                           AND
P.A15 = P2.A15
                                           AND
P.A26 = P2.A26

```

```

EXEC      dbo.uspDropTable 'APPS_QSR_0607'

SELECT    Year_ID = cast(2006 as int),
          AC_Year = cast('2006' as varchar(50)),
          L01 = cast(AIMS.L01 as int),
          L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

          L01_Orig = cast(AIMS.L01 as int),
          L03_Orig = cast(AIMS.L03 as varchar(50)),
          L11 = L11,
          L12,
          L13,
          L14,
          L15,
          L16,
          L17,
          L22,
          L25,
          L44,
          A04,
          A05,
          A09,
          A10,
          A15,
          A16,
          A23,
          A34,
          A35,
          A40,
          A44,
          A49,
          ISNULL(A26, -1) A26,
          A27,
          A28,
          A31,
          A46a,
          A46b,
          A63 = null,
          a_agestb,
          a_leavep,
          A50,
          L39,
          p_prog_status = a_status,
          period,
          l_age,
          Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
          Sequence = cast(0 as int),
          Rnk = cast(0 as int)
INTO      dbo.APPS_QSR_0607
FROM      dbo.ILR0607_W_FRAMEWORKS_AND_AIMS as AIMS
          LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01

          AND AIMS.L03 = M.L03

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

          AND M.L03 !=

M.L03_New

```

```

WHERE a10 = 50 and not (aims.L01 = 110023 )

CREATE CLUSTERED INDEX IDX_APPS_QSR_0607 ON dbo.APPS_QSR_0607(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0607', 'Sequence'

UPDATE      P
SET         Rnk = P.Sequence-P2.Sequence
FROM        dbo.APPS_QSR_0607 P
           JOIN (      SELECT      L01, L03_New, A15, A26,
MIN(Sequence)-1 Sequence
                FROM    dbo.APPS_QSR_0607
                GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 =
P2.L01
AND
P.L03_New = P2.L03_New
AND P.A15
= P2.A15
AND P.A26
= P2.A26

EXEC dbo.uspDropTable 'APPS_QSR_0506'

SELECT      Year_ID = cast(2005 as int),
AC_Year = cast('2005' as varchar(50)),
L01 = cast(AIMS.L01 as int),
L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

L01_Orig = cast(AIMS.L01 as int),
L03_Orig = cast(AIMS.L03 as varchar(50)),
L11 = L11,
L12,
L13,
L14,
L15,
L16,
L17,
L22,
L25,
L44,
A04,
A05,
A09,
A10,
A15,
A16,
A23,
A34,
A35,
A40,
A44,
A49,
ISNULL(A26, -1) A26,
A27,
A28,
A31,
A46a,

```

```

        A46b,
        A63 = null,
        a_agemtb,
        a_leavep,
        A50,
        L39,
        p_prog_status = a_status,
        period,
        l_age,
        Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
        Sequence = cast(0 as int),
        Rnk = cast(0 as int)
INTO  dbo.APPS_QSR_0506
FROM  dbo.ILR0506_W_FRAMEWORKS_AND_AIMS as AIMS
      LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01
                                     AND AIMS.L03 = M.L03
                                     AND ISNULL(M.L03,
'' ) != ''
                                     AND M.L03 !=
M.L03_New
WHERE a10 = 50 and not (aims.L01 = 110023)

CREATE CLUSTERED INDEX IDX_APPS_QSR_0506 ON dbo.APPS_QSR_0506(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0506', 'Sequence'

UPDATE      P
SET         Rnk = P.Sequence-P2.Sequence
FROM        dbo.APPS_QSR_0506 P
           JOIN (      SELECT      L01, L03_New, A15, A26,
MIN(Sequence)-1 Sequence
                FROM    dbo.APPS_QSR_0506
                GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

AND P.A15 = P2.A15

AND P.A26 = P2.A26

EXEC dbo.uspDropTable 'APPS_QSR_0405'

SELECT      Year_ID = cast(2004 as int),
AC_Year = cast('2004' as varchar(50)),
L01 = cast(AIMS.L01 as int),
L03_New = cast(isnull(M.L03_New, AIMS.L03) as varchar(50)),

L01_Orig = cast(AIMS.L01 as int),
L03_Orig = cast(AIMS.L03 as varchar(50)),
L11 = L11,
L12,
L13,
L14,
L15,
L16,

```

```

L17,
L22,
L25,
L44,
A04,
A05,
A09,
A10,
A15,
A16,
A23,
A34,
A35,
A40,
A44,
A49,
ISNULL(A26, -1) A26,
A27,
A28,
A31,
A46a,
A46b,
A63 = null,
a_agemtb,
a_leavep,
A50,
L39,
p_prog_status = a_status,
period,
l_age,
Case WHEN A34 IN(4, 6) THEN 1 ELSE 0 END Priority,
Sequence = cast(0 as int),
Rnk = cast(0 as int)
INTO dbo.APPS_QSR_0405
FROM dbo.ILR0405_W_FRAMEWORKS_AND_AIMS as AIMS
LEFT JOIN dbo.ER_L03_Mergers M ON AIMS.L01 = M.L01

AND AIMS.L03 = M.L03

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

AND M.L03 != M.L03_New
WHERE a10 = 50 and not (aims.L01 = 110023)

CREATE CLUSTERED INDEX IDX_APPS_QSR_0405 ON dbo.APPS_QSR_0405(L01,
L03_New, A15, A26, Priority, A05 desc)

EXEC [dbo].[usp_CreateIdentityColumn] 'APPS_QSR_0405', 'Sequence'

UPDATE P
SET Rnk = P.Sequence-P2.Sequence
FROM dbo.APPS_QSR_0405 P
JOIN (
SELECT L01, L03_New, A15, A26,
MIN(Sequence)-1 Sequence
FROM dbo.APPS_QSR_0405
GROUP BY L01, L03_New, A15, A26) P2 ON P.L01 =
P2.L01

AND P.L03_New = P2.L03_New

```

```
AND P.A15 = P2.A15
```

```
AND P.A26 = P2.A26
```

```
-----  
-----  
-----  
----- 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, A15, A26
```

```
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.APPS_QSR_0405 WHERE Rnk = 1  
  
      UNION  
      SELECT * FROM dbo.APPS_QSR_0506 WHERE Rnk = 1  
  
      UNION  
      SELECT * FROM dbo.APPS_QSR_0607 WHERE Rnk = 1  
  
      UNION  
      SELECT * FROM dbo.APPS_QSR_0708 WHERE Rnk = 1  
  
      UNION
```

```

SELECT * FROM dbo.APPS_QSR_0809 WHERE Rnk = 1

UNION
SELECT * FROM dbo.APPS_QSR_0910 WHERE Rnk = 1

UNION
SELECT * FROM dbo.APPS_QSR_1011 WHERE Rnk = 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.A15 = M2.A15
                  AND M.A26 = M2.A26
                  AND M.L01_Orig = M.L01_New
                  AND M.L03_Orig = M.L03_New

WHERE M2.L01_Orig != M2.L01_New
      OR M2.L03_Orig != M2.L03_New

--Define the Years that the records appear in
SELECT L01_New L01
      , L03_New L03
      , A15
      , A26
      , Max(CASE WHEN AC_Year = '2004' THEN 1 ELSE 0 END) P_In_0405
      , Max(CASE WHEN AC_Year = '2005' THEN 1 ELSE 0 END) P_In_0506
      , Max(CASE WHEN AC_Year = '2006' THEN 1 ELSE 0 END) P_In_0607
      , Max(CASE WHEN AC_Year = '2007' THEN 1 ELSE 0 END) P_In_0708
      , Max(CASE WHEN AC_Year = '2008' THEN 1 ELSE 0 END) P_In_0809
      , Max(CASE WHEN AC_Year = '2009' THEN 1 ELSE 0 END) P_In_0910
      , Max(CASE WHEN AC_Year = '2010' THEN 1 ELSE 0 END) P_In_1011

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, A15, A26,
AC_Year FROM
      dbo.APPS_QSR_0405
      UNION

```

```

        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_0506
        UNION
        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_0607
        UNION
        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_0708
        UNION
        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_0809
        UNION
        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_0910
        UNION
        SELECT          L01, L03_New, A15, A26, AC_Year
FROM      dbo.APPS_QSR_1011
        ) L
        LEFT JOIN #MergerList M ON M.L01 = L.L01
        AND M.L01 !=
M.L01_New
        ) LA
GROUP BY L01_New
        , L03_New
        , A15
        , A26

--Identify each year the L01 has submitted data
SELECT      L01
        , p_live_0405=max(p_In_0405)
        , p_live_0506=max(p_In_0506)
        , 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)
        , p_live_1011=max(p_In_1011)
        , Live=max(p_In_0910)

INTO #L01
FROM #DataSource
GROUP BY L01

--Work out the latest Year and Period
DECLARE @p_period_end SmallDateTime
DECLARE @p_year_per int
DECLARE @Period int

-----
-----

EXEC dbo.uspDropTable 'App_QSR_Data'

--Get the latest Period
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))

```

```

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 + 1 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-' +
                                     END as SmallDateTime)
, @p_year_per = Year_ID
, @Period = Period
FROM ( SELECT MIN(Period) Period, Max(Year_ID) Year_ID
FROM #LA_Merged LA) LA2

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

--Calculate derived variables
SELECT LA.*
, CASE WHEN a_agesb IN(1,2) THEN 1 --'16-18'
        WHEN a_agesb 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_frm_ach
, CASE WHEN p_prog_status IN(7,8,10) AND ISNULL(A31,
P.P_Period_End) <= P.P_Period_End THEN 1 ELSE 0 END p_Trans
, CASE WHEN p_prog_status IN(11,12) AND ISNULL(A31,
P.P_Period_End) <= 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 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
    , P.Period LastestPeriod
    , 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.App_QSR_Data
FROM (SELECT * FROM #LA_Merged L) LA
    CROSS JOIN #Period P

CREATE CLUSTERED INDEX IDX_App_QSR_Data ON dbo.App_QSR_Data(L01_New,
L03_New, A15, A26, Year_ID DESC)

EXEC [dbo].[usp_CreateIdentityColumn] 'App_QSR_Data' , 'Sequence2'

UPDATE P
SET [LastRecord] = P.Sequence2-P2.Sequence2
FROM dbo.App_QSR_Data P
    JOIN ( SELECT L01_New, L03_New, A15, A26,
MIN(Sequence2)-1 Sequence2
    FROM dbo.App_QSR_Data
    GROUP BY L01_New, L03_New, A15, A26) P2 ON
    P.L01_New = P2.L01_New

    AND P.L03_New = P2.L03_New

    AND P.A15 = P2.A15

    AND P.A26 = P2.A26

--Remove records where the Learner has left within six weeks without an
achievement
DELETE FROM dbo.App_QSR_Data
WHERE (datediff(dd, a27,a31)+1 <= 42)
    AND (p_frm_ach = 0)

-----
-----
----- SELECT the Data into the MASTER TRIM table -----
-----
-----

EXEC dbo.uspDropTable 'Apps_MasterTrim_0910'

SELECT L01 = D.L01_New,
L03 = D.L03_New,
A15 = D.A15,
A23 = D.A23,

```

```

A26 = D.A26 ,
A27 = D.A27 ,
A28 = D.A28 ,
A31 = D.A31 ,
A46A = D.A46a ,
A46B = D.A46b ,
A44 = D.A44 ,
A63 = D.A63 ,
A_AGESTB = D.a_agemtb ,
A_LEAVEP = D.a_leavep ,
P_PROG_STATUS = D.p_prog_status ,
period = D.period ,
L_AGE = D.l_age ,
L11 = D.L11 ,
L12 = D.L12 ,
L13 = D.L13 ,
L14 = D.L14 ,
L15 = D.L15 ,
L16 = D.L16 ,
L22 = D.L22 ,
L25 = D.L25 ,
L44 = D.L44 ,
L17 = D.L17 ,
p_in_1011 = DS.p_in_1011 ,
p_in_0910 = DS.p_in_0910 ,
p_in_0809 = DS.p_in_0809 ,
p_in_0708 = DS.p_in_0708 ,
p_in_0607 = DS.p_in_0607 ,
p_in_0506 = DS.p_in_0506 ,
p_in_0405 = DS.p_in_0405 ,
p_live_0910 = DL.p_live_0910 ,
p_live_0809 = DL.p_live_0809 ,
p_live_0708 = DL.p_live_0708 ,
p_live_0607 = DL.p_live_0607 ,
p_live_0506 = DL.p_live_0506 ,
p_live_0405 = DL.p_live_0405 ,
p_frm_ach = D.p_frm_ach ,
p_trans = D.p_trans ,
p_plan_break = D.p_plan_break ,
p_startyr = D.p_startyr ,
p_actendyr = D.p_actendyr ,
p_expendyr = D.p_expendyr ,
age_band = D.age_band ,
p_year_per = D.p_year_per ,
p_period_end = D.p_period_end ,
p_leavers = D.p_leavers ,
p_hybridendyr = CASE WHEN p_expendyr >= p_actendyr THEN
p_expendyr ELSE p_actendyr 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 1
ELSE 0 END ,
P_FRM_ACH_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_FRM_ACH
ELSE 0 END,
P_COUNT_TIMELY = CASE WHEN p_trans=0
AND
p_plan_break=0
AND
p_expendyr BETWEEN 2004 AND 2009 THEN 1
ELSE 0
END,
P_FRM_ACH_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_FRM_ACH
ELSE 0
END,
PRV_NAME = U.PRV_NAME,
PRV_Type = U.PRV_Type,
L46 = U.L46,
[YEAR] = '2009/10',
L01_orig = D.L01_orig,
L03_orig = D.L03_orig,
ISNULL(PC.Pst_LEA, -1) Pst_LEA,
ISNULL(SSA.A_SSA_T1, 'NA') A_SSA_T1,
ISNULL(SSA.A_SSA_T2, 'NA') A_SSA_T2
INTO dbo.Apps_MasterTrim_0910
FROM dbo.App_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.A15 = DS.A15
AND D.A26 = DS.A26
JOIN #L01 DL ON DL.L01 = D.L01_New
LEFT JOIN dbo.Postcode_OLDC_201011 PC ON PC.pst_PostCode =
D.A23
LEFT JOIN dbo.A26_A_SSA_T1_T2_LOOKUP SSA ON SSA.A26 = D.A26
--Only include the latest record from the latest return
WHERE [LastRecord] = 1
GO

CREATE PROCEDURE [dbo].[usp_CreateIdentityColumn]

@TableName as varchar(200),
@ColumnName as varchar(100)

AS

declare @SQLStmt as varchar(4000)
declare @columnnamecheck as varchar(200)

```

```

--Check to see if the column already exists
SET @columnnamecheck = (
    SELECT
        FROM
            dbo.syscolumns.Name
            dbo.sysobjects
            INNER JOIN
                dbo.syscolumns ON dbo.sysobjects.id = dbo.syscolumns.id
            WHERE
                dbo.sysobjects.name
                    =@TableName
                    AND
                        dbo.syscolumns.Name = @ColumnName)

-- drop column if already present
if( @columnnamecheck is not null )
BEGIN

    SET @SQLStmt = 'ALTER TABLE ' + @TableName + ' DROP COLUMN ' +
@ColumnName
    exec(@SQLStmt)
END
/*Create the SQL statement to add a column to the table*/

select @SQLStmt = 'ALTER TABLE ' + @TableName + '
    ADD ' + @ColumnName + ' int IDENTITY'

exec(@SQLStmt)

RETURN(0)

```

Annex B – Glossary of variables used in success rate calculations

L01 – Provider Number

L03 – Learner reference number

A15 – Programme Type

A23 – Delivery location postcode

A26 – Framework code

A27 – Learning Start Date

A28 – Learning planned end date

A31 – Learning actual end date

A46A/B – National aim monitoring

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

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

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

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

L44 – NES delivery LSC number

L AGE – Age of learner as at 31st August

L17 – Home postcode

P IN 0910 – Programme aim is in the 09/10 dataset

P IN 0809 – Programme aim is in the 08/09 dataset

P IN 0708 – Framework record is in the 07/08 dataset

P IN 0607 – Framework record is in the 06/07 dataset

P IN 0506 – Framework record is in the 05/06 dataset

P IN 0405 – Framework record is in the 04/05 dataset

P LIVE 0910 – Provider is live in 09/10

P LIVE 0708 – Provider is live in 07/08

P LIVE 0607 – Provider is live in 06/07

P LIVE 0506 – Provider is live in 05/06

P LIVE 0405 – Provider is live in 04/05

P FRM ACH – If p_prog_status is set to 1, then record is flagged as framework 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_FRM_ACH_OVERALL – Framework achieved overall is equal to framework 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_FRM_ACH_TIMELY – Framework achieved timely is equal to framework 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

A_SSA_T1 – SSA Tier 1 Code

A_SSA_T2 – SSA Tier 2 Code

PRV_TYPE – Provider Type

PST_LEA – Local Authority based on A23 Delivery Location postcode – ONS postcode file