Young People's Learning Agency

Paper

Learner Responsive Success Rates Methodology for 2009/10 – Specification and Supporting Documentation

Date of publication 28 July 2010
Audience The Data Service and the Skills Funding Agency

Publication intent NOT PROTECTIVELY MARKED

Changes since previous version

1 The table below lists changes made to the specification since the 2008/09 version issued 9 June 2010.

Description	Reason for Change	Impact of Change	Date
Reinstate the A27 field in the minimum level of matching when merging together different ILR years. This is for records that are returned in 2009/10 onwards	To allow more accurate matching of records. This was not included in the 2008/09 QSR methodology due to data quality issues	Enables more accurate reporting of success rates.	20/07/2010
Removal of methodology for 2004/05 and addition of methodology for 2009/10	To take account of 2009/10 data	Reporting reflects current year	20/07/2010
Inclusion of syntax	Updated with syntax	None	01/12/2010

The Young People's Learning Agency
Cheylesmore House Quinton Road Coventry CV1 2WT
T 0845 337 2000 **www.ypla.gov.uk**

Purpose

2 The purpose of this paper is to provide the business rules and a Learner Responsive success rates methodology for 2009/10 to the Data Service.

Background

3 This methodology is developed and tested by the YPLA and the Skills Funding Agency Strategic Analysis & Modelling teams to enable the provision of accurate advice and information to the Data Service, which will be used by the Data Service to generate the end to end process to calculate and report qualification success rates for 2009/10.

Summary

4 The Learner Responsive success rates are calculated from base 'ILR' Learner Responsive data submitted by providers who offer Learner Responsive programmes. Success rates are broadly defined as the number of learning aims that have been successfully completed against the number of learning aims that were expected to have been completed.

5 In more detail, the success rate calculation is :

The numerator (successfully completed aims) is defined as the number of the aims in the denominator which have been successfully completed in a given year.

The denominator (the cohort) is the number of aims that are expected to be completed in a given year.

Learner Responsive success rates are based on the expected end year.

For example, the 2009/10 success rate calculation is the number of Learner responsive funded aims due to be completed in 2009/10 that have been successfully completed divided by the number of Learner responsive funded aims that are due to be completed in 2009/10, less exclusions. (**p_achieved_funded / p_count_funded**).

NOTE: Some types of aims are excluded from success rate calculations altogether. The provisional list of exclusions is provided in paragraph 7.

6 In relation to the given year, learning aims that are expected to have been completed that are not yet completed will be considered as a non achieved aim. These are consequently added into the cohort.

Exclusions

7 Some learning aims are excluded from success rate calculations:

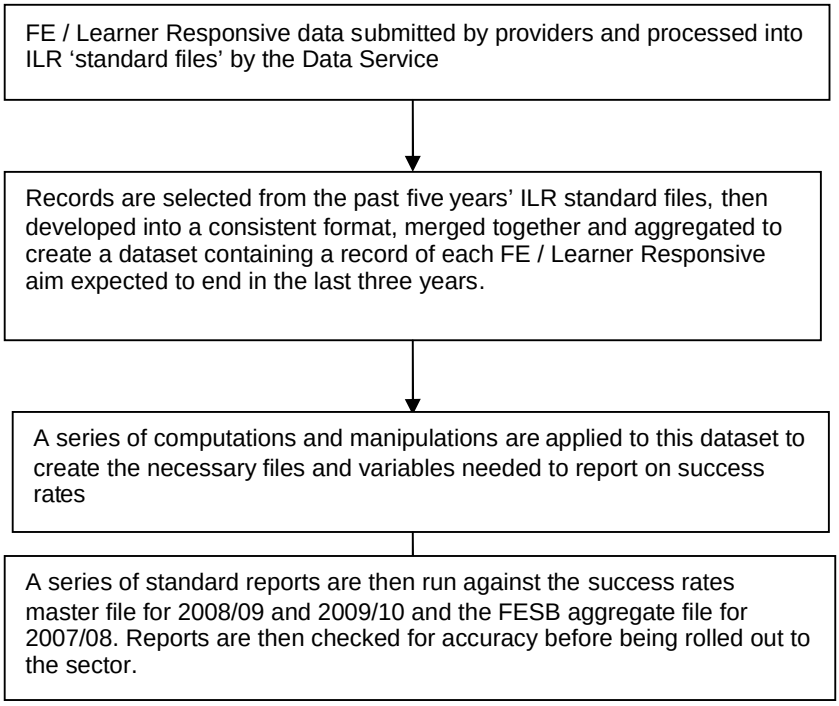
- University for Industry (UfI)
- Offenders' Learning and Skills Service (Olass)

- Qualifcation and Credit Framework Units (QCF) (**applicable to 2008/09 and 2009/10 only**)
- Train to Gain (TtG) (**applicable to 2006/07 only**)
- Employer Training Pilot (ETP)
- FE NVQ's (**applicable to 2006/07 and 2007/08 only**)
- Diplomas and Programme Led Pathways (PLP's) programme aims (**applicable to 2008/09 and 2009/10 only**)
- National Voluntary Training Provider (NVTP) (**applicable to 2008/09 and 2009/10 only**)
- ALR Flexibilities (**applicable to 2008/09 and 2009/10 only**)
- Entry 2 Employment (E2E) (**applicable to 2008/09 and 2009/10 only**)
- Adult Learner Accounts (ALA) (**applicable to 2007/08, 2008/09 and 2009/10 only**)
- First Steps (**applicable to 2008/09 and 2009/10 only**)
- Keyskills
- Functional Skills
- Additional Units
- Unitisation Qualifications
- Diagnostic Tests
- Tutorial Support
- Complementary Studies
- Aims that do not receive LSC funding
- Transfers
- Provision being taken by under 16 year olds

The Success Rates Process

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

9 This document lays out in detail the proce



ss used in taking ILR standard files and manipulating them into a master file on which the standard reports can be run. **Annex A** provides a glossary of the variables used in the success rates process. **Annex B** provides code in SQL to generate the masterfile. SQL code is structured in a specific way, the code gives the results the methodology describes, however it does not read in the same way as the methodology.

NOTE: The process uses variables and definitions that are governed and consistent with those published.

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

10 In order to create a dataset that contains a record of each learning 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. This is because of a significant change in the way ILR data is collected.

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.

Stage 1a – Manipulating the data for years (2005/06 – 2007/08)

11 **2005/06 Data** – Match the ***2005/06 F05 Learner*** standard file to the ***2005/06 F05 Aims*** standard file by the variables **L01** and **L03**. Next, match to the ***2005/06 F05 Aims DLF Lookup*** by the variables **L01 L03** and **A05**. Then,

match to the ***2005/06 F05 Funding Aims*** file by the variables **L01 L03** and **A05**. Next, match to the ***2009/10 LAD Hierarchy*** file by the variable **A09**.

12 **2006/07 Data** – Match the ***2006/07 F05 Learner*** standard file to the ***2006/07 F05 Aims*** standard file by the variables **L01** and **L03**. Next, match to the ***2006/07 F05 Aims DLF Lookup*** by the variables **L01 L03** and **A05**. Then, match to the ***2006/07 F05 Funding Aims*** file by the variables **L01 L03** and **A05**. Next, match to the ***2009/10 LAD Hierarchy*** file by the variable **A09**.

13 **2007/08 Data** – Match the ***2007/08 F05 Learner*** standard file to the ***2007/08 F05 Aims*** standard file by the variables **L01** and **L03**. Next, match to the ***2007/08 F05 Aims DLF Lookup*** by the variables **L01 L03** and **A05**. Then, match to the ***2007/08 F05 Funding Aims*** file by the variables **L01 L03** and **A05**. Next, match to the ***2009/10 LAD Hierarchy*** file by the variable **A09**.

14 Apply the following business rules and exclusions to **each year above**:

- Select records which are expected to end in or after 2007/08 (**p_expendyr>=2007**)
- Select records which are LSC funded (**l_fund=1, 2**) (applies to 2006/07 only)
- Exclude adult FE NVQs delivered in the workplace. The governed definition excludes cases where **A10** does equal **20** and **A18** does equal any of **12, 13, 15, 16, 22 and 23** and where the age of the learner as at 31 August banded (**l_ageb**) is not 16–18 (**l_ageb=1,2**)
- Exclude ETP (Select cases where **A46a and A46b <>17**)
- Exclude Additional Units (Select cases where **additional_unit <> Y**)
- Exclude Unitisation Qualifications (Select cases where **unitisation_qual <> Y**)
- Exclude UfI Qualifications (Select cases where **A46a and A46b <>1**)
- Exclude Diagnostic Tests (Select cases where **diagnostic_test <> Y**)
- Exclude Tutorial Support and Complementary Studies (Select cases where **tutorial_comp_studies <> Y**)

In order to align earlier year's datasets with 2008/09 rename the variable **a_exp_b** to **a_inyr_expected_glh** and **tot_fund** to **a_total_payment_y2d.**

Calculate derived variables for each year's dataset in order to identify whether the aims have received a sln payment (**a_to_date_sln_payment**), the total payment for the aim for the current year (**a_total_payment_y2d**), the total payments for the lifetime of the aim (**a_total_payment**), the total

SLN for the aim for the current year (**a_total_sln_y2d**) and the total SLN for the lifetime of the aim (**a_total_sln**)**.**

Stage 1b – Merging history files

15 The business rules specified above must be applied to the previous years' datasets. Once the business rules have been applied, the datasets are merged and then aggregated in order to create a dataset that contains one record for each learning aim. This is to ensure that where a learning aim spans several years then it is counted only once.

16 In order to identify which year the learning aim is from, add a variable that shows which academic year standard file the aim is being taken (**In_FE_0506, In_FE_0607** etc.).

Stage 1c – Mergers

17 When two colleges merge, the success rate methodology restates historical data under the new merged college. Where a provider has merged with another provider, the old provider numbers are changed to the new merged provider number. This is to allow comparisons across years. The original provider number is also retained in the master trim file, where appropriate (**L01_orig**).

A list of colleges who have merged is provided by the Data Service.

Stage 2 – Manipulating the data for 2008/09 onwards

18 The business rules specified in the next steps must be applied to the 2008/09 and 2009/10 datasets .

19 Match the ***2008/09 LR L05 Learner*** standard file to the ***2008/09 LR L05 Aims*** standard file by the variables **L01** and **L03**. Next, match to the ***2008/09 L05 Aims DLF*** by the variables **L01 L03** and **A05**. Then, match to the ***2008/09 LAD Hierarchy*** file by the variable **A09**. This is to allow for additional reporting fields not held in the standard files.

20 Match the ***2009/10 LR L05 Learner*** standard file to the ***2009/10 LR L05 Aims*** standard file by the variables **L01** and **L03**. Next, match to the ***2009/10 L05 Aims DLF*** by the variables **L01 L03** and **A05**. Then, match to the ***2009/10 LAD Hierarchy*** file by the variable **A09**. This is to allow for additional reporting fields not held in the standard files.

21 Then apply the following business rules and exclusions:

- E2E (Select cases where **A15<>9**)
- Select cases which are expected to end in or after 2007/08 (**p_expendyr >=2007**)

- ETP (Select cases where **A46a and A46b <>17**)
- Exclude UfI Qualifications (Select cases where **A46a and A46b <>1**)
- NVTP (Select cases where NOT **(A46a = 102 and A46b = 999 and A10 = 22 and A14 = 15 and L47 = 2**)
- ALA (Select cases where **A46a and A46b <> 82, 88 or 89**)
- ALR Flexibilities (Select cases where **A46a and A46b <> 108 or 109**)
- Diplomas and PLP's (Select cases where **A04 = 30**).
- QCF units (excluded thru additional_unit)
- First Steps (Select cases where **A10<>80 and A58 <> 05**)
- Exclude Additional Units (Select cases where **additional_unit <> Y**)
- Exclude Unitisation Qualifications (Select cases where **unitisation_qual <> Y**)
- Exclude Diagnostic Tests (Select cases where **diagnostic_test <> Y**)
- Exclude Tutorial Support and Complementary Studies (Select cases where **tutorial_comp_studies <> Y**)

Compute a variable that shows which academic year 'standard file' the aim is being taken from to ensure the latest record for that aim is taken (**In_LR_0809 or In_LR_0910**).

Select or calculate derived variables for each year's dataset in order to identify whether the aims have received a sln payment (**a_to_date_sln_payment**), the total payment for the aim for the current year (**a_total_payment_y2d**), the total payments for the lifetime of the aim (**a_total_payment**), the total SLN for the aim for the current year (**a_total_sln_y2d**) and the total SLN for the lifetime of the aim (**a_total_sln**).

Stage 3 – Merging the files from pre and post 2008/09

22 Files from stage 1b and 2 can now be added together to create one large dataset which contains a record of every aim recorded in the ILR over the past five years. Various combinations of matching variables are used to ensure the correct records are matched, and if the minimum of **L01**, **L03**, **A09**, A27 **p_startyr** and **p_expendyr** match across years it is assumed to be the same record. For records that ended before 2009/10 a lower level of matching is used that does not include A27.

23 After merging the files derived variables are calculated to determine whether the aim generated sln or received a payment in the current or in previous years (**a_to_date_sln_payment**), the total payment for the aim for the current year (**a_total_payment_y2d**), the total payments for the lifetime of

the aim (**a_total_payment**), the total SLN for the aim for the current year, (**a_total_sln_y2d**), the total SLN for the lifetime of the aim (**a_total_sln**), the expected guided learning hours for the current year (**a_total_inyr_expected_glh),** the expected guided learning hours for the lifetime of the aim and flags to identify the academic years in which the **aim** appeared (**In_FE_0506**, **In _FE_0607**, **In_FE_0708**, **In_LR_0809, In_LR_0910** and **L01_orig**)**.**

Stage 3a – Manipulating the data to create master files that can be used for success rates reporting

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

25 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).

Stage 3b – Missing mapcodes

26 If the **map_code_code** variable is missing then set the **map_code_code** to equal the **A09** field.

Stage 3b – Compute transfers

27 If completion status is equal to (4) the learner has transferred to a new learning aim. That is, the learner has withdrawn from his learning aim and as a direct result has at the same time started studying for another learning aim within the same provider (**A34=4**), **p_trans** is set to 1.

Stage 3c – Compute achievers

28 If learning outcome is equal to (1) achieved (**A35= 1**), **p_achieved** is set to 1.

Stage 3d – Compute completed aims

29 If completion status is equal to (4) the learner has completed the learning activities leading to the learning aim (**A34=2**), **p_complete** is set to 1.

Stage 3e – Identifying the year the aim was started, when it was completed and when it was planned to be completed

30 Start Year of the 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 2009/10 **p_startyr** has the value of 2009.

31 Actual End Year of the 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 2009/10 **p_actendyr** has the value of 2009.

32 Expected End Year of the 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 (**A28**) falls within the academic year 2009/10 **p_expendyr** has the value of 2009.

Stage 3f – 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 3g – Compute a_duration

34 Compute a **p_years** variable to be used in creating the duration variable in the next step. Subtract the planned end year (**p_expendyr)** from the start year (**p_startyr**) and add 1 year, this is capped at a maximum of 4 years.

35 Compute a duration variable which flags the learning aim over the academic years.

If the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 24 weeks flag the duration as short (**a_duration=1**) (**Short**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 1 year and **p_years** is equal to 1 then flag the duration as 1 year 1 academic year (**a_duration=2**) (**1 year_1ay)**.

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 1 year and **p_years** is equal to 2 then flag the duration as 1 year 2 academic years (**a_duration=3**) (**1 year_2ay**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 2 years and **p_years** is equal to 2 then flag the duration as 2 year 2 academic years (**a_duration=4**) (**2year_2ay**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 2 years and **p_years** is equal to 3 then flag the duration as 2 year 3 academic years (**a_duration=5**) (**2year_3ay**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 3 years and **p_years** is equal to 3 then flag the duration as 3 year 3 academic years (**a_duration=6**) (**3 year_3ay**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is less than or equal to 3 years and **p_years** is equal to 4 then flag the duration as 3 year 4 academic years (**a_duration=7**) (**3 year_4ay**).

Else if the difference between the planned end date (**A28**) and the start date (**A27**) is greater than **3** years then flag the duration as 4 year or more 4 academic years (**a_duration=8**) (**4year or more 4ay or more**).

Stage 3h – Compute short duration

36 Compute a duration variable which flags the aim as being very short, short or a long aim (**shortdur**).

If the aim is less than 5 weeks, flag as very short, 5 to 24 weeks as short and greater than or equal to 24 weeks as long. The shortdur is derived using the difference between the learning start date (**A27**) and planned end date (**A28**).

Stage 3i – Derive the age of a learner as at 31 August of start year of aim

37 Compute a variable (**a_age_31augstyr**) to calculate the age of a learner as at 31 August of the start year (**p_startyr**) of the learning aim using the date of birth (**L11**).

Stage 3j – Compute the banded age of the learner as at the 31 August of the start year of the aim

38 Band the field (**a_age_31augstyr**) as sysmis equals “missing age”.

0 thru 15 equals “Under 16”, 16 thru 18 equals “16-18”, 19 thru 20 equals “19-20”, 21 thru 24 equals “21-24”, 25 thru 59 equals “25-59” and 60 thru 120 equals “60 and Over” into (**a_age_31augstyr_b**).

Stage 3k – Compute age band of the learner as at the 31 August of start year of aim for reporting purposes

39 Band the field (**a_age_31augstyr_b**) as “Under 16” and “16-18” as “16-18” and all else as “19+” into (**a_age_31augstyr_band**).

Stage 3l – Compute Key Skills

40 If the **learning aim type code** is equal to “1327”, **a_keyskills** is set to 1.

Stage 3m – Compute Functional_Skills

41 If the **learning aim type code** is equal to "1439", **a_functional_skills** is set to 1.

Stage 3n – Compute Olass (Offenders in Custody)

42 If National learning aim monitoring (**A46a**) is equal to 34 or National learning aim monitoring (**A46b**) is equal to 34, **a_olass** is set to 1.

Stage 3o – Calculating the key variables for the success rates measure

43 The success rate measure is based on the expected end year (**p_expendyr**). The key variable for the success rate measure is the successful completion of programme aims on or before the planned end date.

44 The specific calculation is **p_achieved_funded / p_count_funded**.

The **p_achieved_funded** and **p_count_funded** variables are computed by selecting the range of the **p_expendyr** to be between 2007 and 2009 AND the learning aim must not be a transfer (**p_trans=0**) AND the learning aim must not be olass (**a_olass=0**) AND the learning aim must not be keyskills (**a_keyskills=0**) and the learning aim must not be functional skills (**a_functional_skills=**0) AND the age of the learner at the 31st of August of the start aim must not be under 16 (**a_age_31augstyr <>1**) AND the learning aim must be a funded aim. If these criteria are met then compute **p_count_funded=1**. If the criteria are met and if **A35** equals 1 **p_achieved_funded = 1.**

45 A learning aim is counted as funded if **a_to_date_sln_payment =1**. Note that to take account of anomalies in the way that ILR data is returned, learning aims that are expected to last at least 168 days that are withdrawn from within 42 days, are flagged as qualifying for an SLN payment **(A_ToDate_Qualifying_SLN_Period = 1)** yet have received no payment **(A_Total_Payment_Y2D = 0 or is null)** are considered as **not funded**

Stage 3p – Tidying up the raw data

45 All the derived fields with 'system missing' or 'null' values need to be set to 0.

Stage 3q – Compute Source Year variable

46 Compute a source year variable. This field can be used in order to identify source year of aim if for further analysis is required. If the field **In_LR_0910** is equal to 1 then set **source_year** equal to 2009/10. If the field **In_LR_0809** is equal to 1 then set **source_year** equal to 2008/09. If **In_FE_0708** is equal to 1 then set **source_year** equal to 2007/08. If **In_FE_0607** is equal to 1 then

set **source_year** equal to 2006/07. If **In_FE_0506** is equal to 1 then set **source_year** equal to 2005/06.

Stage 3r – Compute Year variable

47 Compute (**year**) is equal to 2009/2010.

Stage 4 – Preparing the final datasets for success rates reporting

48 Match in the provider name, provider LLSC and provider region into the dataset using the variable **L01**. The provider name should be taken from the latest year's provider lookup file (for example, 'ILR0910_UPIN_TO_LLSC').

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

50 The dataset is now ready for use in standard reporting.

Changes to Learner Reference Numbers between years

51 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.

52 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.

53 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: Keyskills (a_keyskills) and Functional skills (a_functional_skills) are flagged rather than excluded in the master trim. It is therefore necessary to exclude these when calculating success rates.

Author Strategic Analysis
Date created 20 July 2010
Publication no.
Document ref.
Version

Annex A

```
SELECT '-------------IMPORTANT-----------------'
UNION
SELECT 'CHECK USING CORRECT LAD HIERARCHY TABLE'
UNION
SELECT 'CHECK THAT WE ARE USING THE LATEST MERGERS FILE'

--
SELECT      2005 Year_ID
            , LA.L01
            , LA.L03
            , LA.A09
            , LA.A15
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A35
            , LA.A34
--Calculate FENVQ
            , CASE WHEN LA.A10 = 20
                              AND LA.A18 IN(12,13,15,16,22,23)
                              AND L.L_AGEB NOT IN(1,2) THEN 1
                                                                  ELSE 0
              END FE_NVQ
            , CASE 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
            , LA.A05
            , DLF.A_TODATE_QUALIFYING_SLN_PERIOD
            , DLF.A_TO_DATE_SLN_PAYMENT
            , Tot_Fund_Factor A_TOTAL_PAYMENT_Y2D
            , DLF.A_TOTAL_SLN_Y2D
            , LA.a_exp_b a_inyr_expected_glh
            , 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
INTO  #LA
FROM  ILR0506_F05_AIMS LA
            JOIN ILR0506_F05_AIMS_DLF_LOOKUP DLF ON DLF.L01 = LA.L01

                                 AND DLF.L03 = LA.L03

                                 AND DLF.A05 = LA.A05
            JOIN ILR0506_F05_LEARNER L ON LA.L01 = L.L01

                       AND LA.L03 = L.L03
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
            LEFT JOIN ILR0506_F05_FUNDING_AIMS F ON F.L01 = LA.L01
```

```
                                    AND F.L03 = LA.L03

                                    AND F.A05 = LA.A05
WHERE
--Only include Aims with a planned end date of 2007 or Greater
            CASE 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 >= 2007
--Exclude Ufi
            AND 1 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))  --Ufi
--Exclude ETP
            AND 17 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude additional units
            AND LAD.Additional_Unit != 'Y'
--Exclude Unitisation Quals
            AND LAD.Unitisation_Qual != 'Y'
--Exclude Diagnostic Tests
            AND LAD.Diagnostic_Test != 'Y'
--Exclude Tutorial Support and Complimentary Studies
            AND LAD.Tutorial_Comp_Studies != 'Y'
GO

INSERT INTO #LA
SELECT      2006 Year_ID
            , LA.L01
            , LA.L03
            , LA.A09
            , LA.A15
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A35
            , LA.A34
--Calculate FENVQ
            , CASE WHEN LA.A10 = 20
                            AND LA.A18 IN(12,13,15,16,22,23)
                            AND L.L_AGEB NOT IN(1,2) THEN 1
                                                          ELSE 0
              END FE_NVQ
            , CASE 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
            , LA.A05
            , DLF.A_TODATE_QUALIFYING_SLN_PERIOD
            , DLF.A_TO_DATE_SLN_PAYMENT
            , Tot_Fund_Factor A_Total_Payment_Y2D
            , DLF.A_TOTAL_SLN_Y2D
            , LA.a_exp_b a_inyr_expected_glh
            , 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
FROM  ILR0607_F05_AIMS LA
            JOIN ILR0607_F05_AIMS_DLF_LOOKUP DLF ON DLF.L01 = LA.L01

                                    AND DLF.L03 = LA.L03

                                    AND DLF.A05 = LA.A05
            JOIN ILR0607_F05_LEARNER L ON LA.L01 = L.L01

                        AND LA.L03 = L.L03
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
            LEFT JOIN ILR0607_F05_FUNDING_AIMS F ON F.L01 = LA.L01

                                    AND F.L03 = LA.L03

                                    AND F.A05 = LA.A05
WHERE
--Only include Aims with a planned end date of 2007 or Greater
            CASE 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 >= 2007
--Exclude Ufi
            AND 1 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude ETP
            AND 17 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude TtG
            AND LA.A_TTGAIN <> 2
--Exclude additional units
            AND LAD.Additional_Unit != 'Y'
--Exclude Unitisation Quals
            AND LAD.Unitisation_Qual != 'Y'
--Exclude Diagnostic Tests
            AND LAD.Diagnostic_Test != 'Y'
--Exclude Tutorial Support and Complimentary Studies
            AND LAD.Tutorial_Comp_Studies != 'Y'
GO

INSERT INTO #LA
SELECT      2007 Year_ID
            , LA.L01
            , LA.L03
            , LA.A09
            , LA.A15
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A35
            , LA.A34
--Calculate FENVQ
            , CASE WHEN LA.A10 = 20
                              AND LA.A18 IN(12,13,15,16,22,23)
                              AND L.L_AGEB NOT IN(1,2) THEN 1
                                                                  ELSE 0
              END FE_NVQ
```

```
            , CASE 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
            , LA.A05
            , DLF.A_TODATE_QUALIFYING_SLN_PERIOD
            , DLF.A_TO_DATE_SLN_PAYMENT
            , Tot_Fund_Factor a_Total_Payment_Y2D
            , DLF.A_TOTAL_SLN_Y2D
            , LA.a_exp_b a_inyr_expected_glh
            , 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
FROM  ILR0708_F05_AIMS LA
            JOIN ILR0708_F05_AIMS_DLF_LOOKUP DLF ON DLF.L01 = LA.L01

                                 AND DLF.L03 = LA.L03

                                 AND DLF.A05 = LA.A05
            JOIN ILR0708_F05_LEARNER L ON LA.L01 = L.L01

                       AND LA.L03 = L.L03
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
            LEFT JOIN ILR0708_F05_FUNDING_AIMS F ON F.L01 = LA.L01

                                 AND F.L03 = LA.L03

                                 AND F.A05 = LA.A05
WHERE
--Only include Aims with a planned end date of 2007 or Greater
            CASE 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 >= 2007
--Exclude ETP
            AND 17 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Adult Learner Accounts
            AND 88 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 89 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 82 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Ufi
            AND 1 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))  --Ufi
--Exclude additional units
            AND LAD.Additional_Unit != 'Y'
--Exclude Unitisation Quals
            AND LAD.Unitisation_Qual != 'Y'
--Exclude Diagnostic Tests
            AND LAD.Diagnostic_Test != 'Y'
--Exclude Tutorial Support and Complimentary Studies
            AND LAD.Tutorial_Comp_Studies != 'Y'
GO
```

```
INSERT INTO #LA
SELECT      2008 Year_ID
            , LA.L01
            , LA.L03
            , LA.A09
            , LA.A15
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A35
            , LA.A34
            , 0 FE_NVQ
            , CASE 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
            , LA.A05
            , LA.A_IY_QUALIFYING_SLN_Period
            , LA.A_TO_DATE_SLN_PAYMENT
            , DLF.A_Total_Payment_Y2D
            , DLF.A_TOTAL_SLN_Y2D
            , LA.A_INYR_EXPECTED_GLH
            , 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
FROM  ILR0809_L05_AIMS LA
            JOIN ILR0809_L05_LEARNER L ON LA.L01 = L.L01

                        AND LA.L03 = L.L03
            JOIN ILR0809_L05_AIMS_DLF DLF ON    DLF.L01 = LA.L01

                              AND DLF.L03 = LA.L03

                              AND DLF.A05 = LA.A05
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
WHERE
--Only include Aims with a planned end date of 2007 or Greater
            CASE 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 >= 2007
            AND
--Exclude NVTP
            CASE WHEN   LA.A10 = 22
                              AND 102 IN(ISNULL(LA.A46A,0))
                              AND 999 IN(ISNULL(LA.A46B,0))
                              AND LA.A14 = 15
                              and L.L47 = 2      THEN 0
                                                      ELSE 1
            END = 1
```

```
--Exclude ETP
            AND 17 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Adult Learner Accounts
            AND 82 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 88 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 89 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))

--Exclude Flexibilities
            AND 108 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 109 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Ufi
            AND 1 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude E2E
            AND LA.A15 != 9
--Exclude whole diplomas and PLPs
            AND   LA.A04 = 30
--Exclude First Steps
            AND CASE WHEN LA.A10 = 80 AND LA.A58 = 5 THEN 0 ELSE 1 END =
1
--Exclude additional units
            AND LAD.Additional_Unit != 'Y'
--Exclude Unitisation Quals
            AND LAD.Unitisation_Qual != 'Y'
--Exclude Diagnostic Tests
            AND LAD.Diagnostic_Test != 'Y'
--Exclude Tutorial Support and Complimentary Studies
            AND LAD.Tutorial_Comp_Studies != 'Y'
GO


INSERT INTO #LA
SELECT      2009 Year_ID
            , LA.L01
            , LA.L03
            , LA.A09
            , LA.A15
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A35
            , LA.A34
            , 0 FE_NVQ
            , CASE 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
            , LA.A05
            , LA.A_IY_QUALIFYING_SLN_Period
            , LA.A_TO_DATE_SLN_PAYMENT
            , DLF.A_Total_Payment_Y2D
            , DLF.A_TOTAL_SLN_Y2D
            , LA.A_INYR_EXPECTED_GLH
            , 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
FROM  ILR0910_L05_AIMS LA
            JOIN ILR0910_L05_LEARNER L ON LA.L01 = L.L01

                        AND LA.L03 = L.L03
            JOIN ILR0910_L05_AIMS_DLF DLF ON    DLF.L01 = LA.L01

                              AND DLF.L03 = LA.L03

                              AND DLF.A05 = LA.A05
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
WHERE
--Only include Aims with a planned end date of 2007 or Greater
            CASE 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 >= 2007
            AND
--Exclude NVTP
            CASE WHEN   LA.A10 = 22
                              AND 102 IN(ISNULL(LA.A46A,0))
                              AND 999 IN(ISNULL(LA.A46B,0))
                              AND LA.A14 = 15
                              and L.L47 = 2      THEN 0
                                                      ELSE 1
            END = 1
--Exclude ETP
            AND 17 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Adult Learner Accounts
            AND 82 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 88 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 89 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))

--Exclude Flexibilities
            AND 108 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
            AND 109 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude Ufi
            AND 1 NOT IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
--Exclude E2E
            AND LA.A15 != 9
--Exclude whole diplomas and PLPs
            AND   LA.A04 = 30
--Exclude First Steps
            AND CASE WHEN LA.A10 = 80 AND LA.A58 = 5 THEN 0 ELSE 1 END =
1
--Exclude additional units
            AND LAD.Additional_Unit != 'Y'
--Exclude Unitisation Quals
            AND LAD.Unitisation_Qual != 'Y'
--Exclude Diagnostic Tests
            AND LAD.Diagnostic_Test != 'Y'
--Exclude Tutorial Support and Complimentary Studies
            AND LAD.Tutorial_Comp_Studies != 'Y'
GO
```

```
--Add in Mergers based on L01 Mergers file and L03 Mapping returns
SELECT      L.*
            , CASE      WHEN M.L01 IS NULL            THEN  L.L01
                                                      ELSE
      M.L01_New
             END L01_New
            , CASE      WHEN M2.L03 IS NULL           THEN  L.L03
                                                      ELSE
      M2.L03_New
             END L03_New
INTO  #LA_Merged
FROM  #LA L
            LEFT JOIN LR_L01_Mergers M ON M.L01 = L.L01

            AND Mrg_Academic_Year < 9

            AND M.L01 != M.L01_New
            LEFT JOIN LR_L03_Mergers M2 ON M2.L01 = L.L01

                  AND M2.L03 = L.L03

                  AND ISNULL(M2.L03_New, '') != ''
GO

--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.P_ExpEndYr =
M2.P_ExpEndYr
                                          AND M.L01 = M.L01_New
                                          AND M.L03 = M.L03_New
WHERE M2.L01 != M2.L01_New
            OR M2.L03 != M2.L03_New
GO

--Produce a temporary store
EXEC usp_DropTable 'Temp.QSR_LR_Merged_0506_0910'

SELECT      *
--Generate a unique id within each year
            , RANK() OVER(PARTITION BY    Year_ID
--Sequence using the following
                                  ORDER BY Year_ID
                                          , L01
                                          , L03
                                          , A09
                                          , A27
                                          , A28
--Ensure anything with an actual end date or is a transfer
--is seeded first
```

```
                                                , CASE      WHEN A34 = 4
THEN 2
                                                            WHEN A31 IS
NOT NULL THEN 1

       ELSE 0
                                                  END DESC
                                                , A05) [Rank]
--Create matching control fields
            , CAST(0 As integer)    Matched
            , CAST(0 As integer)    MatchYear
            , CAST(0 As integer)    MatchedRank
INTO  Temp.QSR_LR_Merged_0506_0910
FROM  #LA_Merged
WHERE L01_New != -1
GO

--Call matching process for 2004 data, passing in start and end year
--(see later section for details of process)
--Call matching process for 2005 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2005, 2009, 0
GO

--Call matching process for 2006 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2006, 2009, 0
GO

--Call matching process for 2007 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2007, 2009, 0
GO

--Call matching process for 2008 data, passing in start and end year
--However ensure that match is constrained to match on startdate
EXEC QSR.usp_LR_QSR_MatchProcess 2008, 2009, 1
GO

--Call matching process for 2008 data, passing in start and end year
--However ensure that match is constrained to match on startdate
EXEC QSR.usp_LR_QSR_MatchProcess 2009, 2010, 1
GO


--Generate a row count
SELECT 'Dummy row forced'

--Force Match Year and Match Rank in prior matches to
--record the last found match
--Iterate until no more updates occur
WHILE @@RowCount != 0
BEGIN
      UPDATE      A
      SET         MatchYear = B.MatchYear
                  , A.MatchedRank = B.MatchedRank
                  , A.FE_NVQ = CASE WHEN B.Year_ID < 2008 THEN B.FE_NVQ
ELSE A.FE_NVQ END
      FROM  Temp.QSR_LR_Merged_0506_0910 A
                  JOIN  Temp.QSR_LR_Merged_0506_0910 B ON   B.Year_ID =
A.MatchYear
```

```
                                        AND B.Rank = A.MatchedRank
      WHERE A.MatchedRank != 0
                  AND B.MatchedRank != 0
END

--Identify all un-matched records and record
--In addition calulate the total GLH and payment values by summing all
--values from the associated matched records and the unmatched record
SELECT      A.Year_ID
            , A.L01_New L01
            , A.L03_New L03
            , A.A09
            , A.P_StartYr
            , A.P_ExpEndYr
            , A.FE_NVQ
            , A.L01     L01_Orig
            , A.L03     L03_Orig
            , A.A05
            , B.A_TODATE_QUALIFYING_SLN_PERIOD
            , B.A_TO_DATE_SLN_PAYMENT
            , B.A_TOTAL_PAYMENT
            , B.A_TOTAL_SLN
            , B.A_TOTAL_INYR_EXPECTED_GLH
            , A.A_Total_Payment_Y2D
            , A.Matched
            , A.A27
            , A.A28
            , A.A31
            , A.A34
INTO  #Latest
FROM  Temp.QSR_LR_Merged_0506_0910 A
            JOIN (      SELECT      Case WHEN MatchYear = 0 THEN Year_ID
ELSE MatchYear END Year_ID
                                    , Case WHEN MatchedRank = 0 THEN Rank
ELSE MatchedRank END Rank
                                    , Max(FE_NVQ)     FE_NVQ
                                    , Max(A_TO_DATE_SLN_PAYMENT)
A_TO_DATE_SLN_PAYMENT
                                    , SUM(ISNULL(A_Total_Payment_Y2D,0))
A_TOTAL_PAYMENT
                                    , SUM(ISNULL(A_TOTAL_SLN_Y2D, 0))
A_TOTAL_SLN
                                    , SUM(ISNULL(A_INYR_EXPECTED_GLH, 0))
A_TOTAL_INYR_EXPECTED_GLH
                                    ,
Max(ISNULL(A_TODATE_QUALIFYING_SLN_PERIOD, 0))
A_TODATE_QUALIFYING_SLN_PERIOD
                        FROM  Temp.QSR_LR_Merged_0506_0910 A
                        GROUP BY Case WHEN MatchYear = 0 THEN Year_ID
ELSE MatchYear END
                                    , Case WHEN MatchedRank = 0 THEN Rank
ELSE MatchedRank END) B ON    A.Year_ID = B.Year_ID

                                                                AND
A.Rank = B.Rank
```

```
--Enforce six week rule for records not eligible for funding and not
received funding
DELETE
FROM  #Latest
WHERE CASE WHEN   Datediff(d, A27, ISNULL(A31, A28))+1 < 6*7
                              AND Datediff(d, A27, A28)+1 >= 24*7
                              AND A34 = 3             THEN 0
                                                          ELSE 1
            END = 0
            AND A_ToDate_Qualifying_SLN_Period = 1
            AND ISNULL(A_Total_Payment_Y2D, 0) = 0
GO

SELECT      *
INTO  LR_QSR_Latest_Recs_Matches_vA27
FROM  #Latest
GO

--Remove FE NVQ records after matching
DELETE
FROM  #Latest
WHERE FE_NvQ = 1
            AND Year_ID Between 2005 AND 2007
GO

--***0506 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT      2005 Year_ID
            , LAT.L01
            , LAT.L01_Orig
            , LAT.L03
            , LAT.L03_Orig
            , LA.A05
            , LA.A09
            , LA.A10
            , LA.A14
            , LA.A15
            , LA.A18
            , LA.A21
            , LA.A23
            , LA.A26
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A32
            , LA.A34
            , LA.A35
            , LA.A36
--          , LA.A37
--          , LA.A38
            , LA.A46a
            , LA.A46b
            , LA.A49
            , LA.A50
            , LA.A53
            , LA.A_ATYPE
            , Cast(LA.A_DPLLSC As Real) A_DPLLSC
--          , LA.A_NVQLEV
```

```
              , LA.A_OPROV
              , L.L_ATYPE
              , L.L_FUND
              , L.L_LREG
              , L.L11
              , L.L12
              , L.L13
              , L.L14
              , L.L15
              , L.L16
              , L.L17
              , L.L22
              , L.L25
              , L.L29
              , L.L32
              , L.L34A
              , L.L34B
              , L.L34C
              , L.L34D
              , L.L35
              , CAST(L.L44 as Real) L44
              , L.L_LLLSC
              , L.L_PLLSC
              , L.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
                        , LA.a_exp_b A_INYR_EXPECTED_GLH
              , CASE 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

              , 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

              , LAT.A_TODATE_QUALIFYING_SLN_PERIOD
              , DLF.L_FUND_ACTIVE
              , DLF.A_IY_SLN_PAYMENT
              , DLF.A_PRIOR_SLN_PAYMENT
              , DLF.A_TOTAL_SLN_Y2D
              , DLF.A_FULLY_FUNDED
              , CAST(NULL As Real) L46
              , CAST(NULL As Real) A56
              , LAT.A_Total_Payment_Y2D
              , LAT.A_TO_DATE_SLN_PAYMENT
```

```
            , LAT.A_Total_Payment
            , LAT.A_TOTAL_SLN
            , LAT.A_TOTAL_INYR_EXPECTED_GLH
INTO  #0506_F05_Aims_Data_v1
FROM  ILR0506_F05_LEARNER L
            JOIN ILR0506_F05_AIMS LA ON LA.L01 = L.L01
                                                AND LA.L03 = L.L03
            JOIN ILR0506_F05_AIMS_DLF_LOOKUP DLF ON DLF.L01 = LA.L01

                                AND DLF.L03 = LA.L03

                                AND DLF.A05 = LA.A05
            JOIN (SELECT L01
                        , L01_Orig
                        , L03
                        , L03_Orig
                        , A05
                        , A_TODATE_QUALIFYING_SLN_PERIOD
                        , A_Total_Payment_Y2D
                        , A_TO_DATE_SLN_PAYMENT
                        , A_Total_Payment
                        , A_TOTAL_SLN
                        , A_TOTAL_INYR_EXPECTED_GLH
                  FROM      #Latest
                  WHERE     Year_ID = 2005) LAT ON LAT.L01_Orig =
LA.L01
                                                                AND

LAT.L03_Orig = LA.L03
                                                                AND

LAT.A05 = LA.A05

GO

--***0607 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT      2006 Year_ID
            , LAT.L01
            , LAT.L01_Orig
            , LAT.L03
            , LAT.L03_Orig
            , LA.A05
            , LA.A09
            , LA.A10
            , LA.A14
            , LA.A15
            , LA.A18
            , LA.A21
            , LA.A23
            , LA.A26
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A32
            , LA.A34
            , LA.A35
            , LA.A36
            , LA.A46a
            , LA.A46b
```

```
              , LA.A49
              , LA.A50
              , LA.A53
              , LA.A_ATYPE
              , LA.A_DPLLSC
              , LA.A_OPROV
              , L.L_ATYPE
              , L.L_FUND
              , L.L_LREG
              , L.L11
              , L.L12
              , L.L13
              , L.L14
              , L.L15
              , L.L16
              , L.L17
              , L.L22
              , L.L25
              , L.L29
              , L.L32
              , L.L34A
              , L.L34B
              , L.L34C
              , L.L34D
              , L.L35
              , L.L44
              , L.L_LLLSC
              , L.L_PLLSC
              , L.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
              , LA.a_exp_b A_INYR_EXPECTED_GLH
              , CASE 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

              , 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

              , LAT.A_TODATE_QUALIFYING_SLN_PERIOD
              , DLF.L_FUND_ACTIVE
              , DLF.A_IY_SLN_PAYMENT
              , DLF.A_PRIOR_SLN_PAYMENT
              , DLF.A_TOTAL_SLN_Y2D
```

```
            , DLF.A_FULLY_FUNDED
            , L.L46
            , LA.A56
            , LAT.A_Total_Payment_Y2D
            , LAT.A_TO_DATE_SLN_PAYMENT
            , LAT.A_Total_Payment
            , LAT.A_TOTAL_SLN
            , LAT.A_TOTAL_INYR_EXPECTED_GLH
INTO  #0607_F05_Aims_Data_v1
FROM  ILR0607_F05_LEARNER L
            JOIN ILR0607_F05_AIMS LA ON LA.L01 = L.L01
                                                      AND LA.L03 = L.L03
            JOIN ILR0607_F05_AIMS_DLF_LOOKUP DLF ON   DLF.L01 = LA.L01

      AND DLF.L03 = LA.L03

      AND DLF.A05 = LA.A05
            JOIN (SELECT L01
                        , L01_Orig
                        , L03
                        , L03_Orig
                        , A05
                        , A_TODATE_QUALIFYING_SLN_PERIOD
                        , A_Total_Payment_Y2D
                        , A_TO_DATE_SLN_PAYMENT
                        , A_Total_Payment
                        , A_TOTAL_SLN
                        , A_TOTAL_INYR_EXPECTED_GLH
                  FROM      #Latest
                  WHERE     Year_ID = 2006) LAT ON LAT.L01_Orig =
LA.L01
                                                            AND
LAT.L03_Orig = LA.L03
                                                            AND
LAT.A05 = LA.A05
GO


--0708 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT      2007 Year_ID
            , LAT.L01
            , LAT.L01_Orig
            , LAT.L03
            , LAT.L03_Orig
            , LA.A05
            , LA.A09
            , LA.A10
            , LA.A14
            , LA.A15
            , LA.A18
            , LA.A21
            , LA.A23
            , LA.A26
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A32
```

```
              , LA.A34
              , LA.A35
              , LA.A36
              , LA.A46a
              , LA.A46b
              , LA.A49
              , LA.A50
              , LA.A53
              , LA.A_ATYPE
              , LA.A_DPLLSC
              , LA.A_OPROV
              , L.L_ATYPE
              , L.L_FUND
              , L.L_LREG
              , L.L11
              , L.L12
              , L.L13
              , L.L14
              , L.L15
              , L.L16
              , L.L17
              , L.L22
              , L.L25
              , L.L29
              , L.L32
              , L.L34A
              , L.L34B
              , L.L34C
              , L.L34D
              , L.L35
              , L.L44
              , L.L_LLLSC
              , L.L_PLLSC
              , L.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
              , LA.a_exp_b A_INYR_EXPECTED_GLH
              , CASE 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

              , 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
```

```
            , LAT.A_TODATE_QUALIFYING_SLN_PERIOD
            , DLF.L_FUND_ACTIVE
            , DLF.A_IY_SLN_PAYMENT
            , DLF.A_PRIOR_SLN_PAYMENT
            , DLF.A_TOTAL_SLN_Y2D
            , DLF.A_FULLY_FUNDED
            , L.L46
            , LA.A56
            , LAT.A_Total_Payment_Y2D
            , LAT.A_TO_DATE_SLN_PAYMENT
            , LAT.A_Total_Payment
            , LAT.A_TOTAL_SLN
            , LAT.A_TOTAL_INYR_EXPECTED_GLH
INTO  #0708_F05_Aims_Data_v1
FROM  ILR0708_F05_LEARNER L
            JOIN ILR0708_F05_AIMS LA ON LA.L01 = L.L01
                                                AND LA.L03 = L.L03
            JOIN ILR0708_F05_AIMS_DLF_LOOKUP DLF ON   DLF.L01 = LA.L01

      AND DLF.L03 = LA.L03

      AND DLF.A05 = LA.A05
            JOIN (SELECT L01
                        , L01_Orig
                        , L03
                        , L03_Orig
                        , A05
                        , A_TODATE_QUALIFYING_SLN_PERIOD
                        , A_Total_Payment_Y2D
                        , A_TO_DATE_SLN_PAYMENT
                        , A_Total_Payment
                        , A_TOTAL_SLN
                        , A_TOTAL_INYR_EXPECTED_GLH
                  FROM      #Latest
                  WHERE     Year_ID = 2007) LAT ON LAT.L01_Orig =
LA.L01
                                                      AND
LAT.L03_Orig = LA.L03
                                                      AND
LAT.A05 = LA.A05
GO

--0809 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT      2008 Year_ID
            , LAT.L01
            , LAT.L01_Orig
            , LAT.L03
            , LAT.L03_Orig
            , LA.A05
            , LA.A09
            , LA.A10
            , LA.A14
            , LA.A15
            , LA.A18
            , LA.A21
            , LA.A23
            , LA.A26
```

```
              , LA.A27
              , LA.A28
              , LA.A31
              , LA.A32
              , LA.A34
              , LA.A35
              , LA.A36
              , LA.A46a
              , LA.A46b
              , LA.A49
              , LA.A50
              , LA.A53
              , LA.A_ATYPE
              , LA.A_DPLLSC
              , LA.A_OPROV
              , L.L_ATYPE
              , L.L_FUND
              , L.L_LREG
              , L.L11
              , L.L12
              , L.L13
              , L.L14
              , L.L15
              , L.L16
              , L.L17
              , L.L22
              , L.L25
              , L.L29
              , L.L32
              , L.L34A
              , L.L34B
              , L.L34C
              , L.L34D
              , L.L35
              , L.L44
              , L.L_LLLSC
              , L.L_PLLSC
              , L.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
              , LA.A_INYR_EXPECTED_GLH
              , CASE 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

              , 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
           , LAT.A_TODATE_QUALIFYING_SLN_PERIOD
           , -1 L_FUND_ACTIVE
           , LA.A_IY_SLN_PAYMENT
           , LA.A_PRIOR_SLN_PAYMENT
           , DLF.A_TOTAL_SLN_Y2D
           , LA.A_FULLY_FUNDED
           , L.L46
           , LA.A56
           , LAT.A_Total_Payment_Y2D
           , LAT.A_TO_DATE_SLN_PAYMENT
           , LAT.A_Total_Payment
           , LAT.A_TOTAL_SLN
           , LAT.A_TOTAL_INYR_EXPECTED_GLH
INTO  #0809_F05_Aims_Data_v1
FROM  ILR0809_L05_LEARNER L
           JOIN ILR0809_L05_AIMS LA ON LA.L01 = L.L01

      AND LA.L03 = L.L03
           JOIN ILR0809_L05_AIMS_DLF DLF ON    DLF.L01 = LA.L01

                              AND DLF.L03 = LA.L03

                              AND DLF.A05 = LA.A05
           JOIN (SELECT L01
                              , L01_Orig
                              , L03
                              , L03_Orig
                              , A05
                              , A_TODATE_QUALIFYING_SLN_PERIOD
                              , A_Total_Payment_Y2D
                              , A_TO_DATE_SLN_PAYMENT
                              , A_Total_Payment
                              , A_TOTAL_SLN
                              , A_TOTAL_INYR_EXPECTED_GLH
                 FROM      #Latest
                 WHERE     Year_ID = 2008) LAT ON LAT.L01_Orig =
LA.L01
                                                                  AND
LAT.L03_Orig = LA.L03
                                                                  AND
LAT.A05 = LA.A05
GO

--0809 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT      2009 Year_ID
           , LAT.L01
           , LAT.L01_Orig
           , LAT.L03
           , LAT.L03_Orig
           , LA.A05
           , LA.A09
           , LA.A10
           , LA.A14
           , LA.A15
```

```
              , LA.A18
              , LA.A21
              , LA.A23
              , LA.A26
              , LA.A27
              , LA.A28
              , LA.A31
              , LA.A32
              , LA.A34
              , LA.A35
              , LA.A36
              , LA.A46a
              , LA.A46b
              , LA.A49
              , LA.A50
              , LA.A53
              , LA.A_ATYPE
              , LA.A_DPLLSC
              , LA.A_OPROV
              , L.L_ATYPE
              , L.L_FUND
              , L.L_LREG
              , L.L11
              , L.L12
              , L.L13
              , L.L14
              , L.L15
              , L.L16
              , L.L17
              , L.L22
              , L.L25
              , L.L29
              , L.L32
              , L.L34A
              , L.L34B
              , L.L34C
              , L.L34D
              , L.L35
              , L.L44
              , L.L_LLLSC
              , L.L_PLLSC
              , L.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
              , LA.A_INYR_EXPECTED_GLH
              , CASE 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

              , 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
            , LAT.A_TODATE_QUALIFYING_SLN_PERIOD
            , -1 L_FUND_ACTIVE
            , LA.A_IY_SLN_PAYMENT
            , LA.A_PRIOR_SLN_PAYMENT
            , DLF.A_TOTAL_SLN_Y2D
            , LA.A_FULLY_FUNDED
            , L.L46
            , LA.A56
            , LAT.A_Total_Payment_Y2D
            , LAT.A_TO_DATE_SLN_PAYMENT
            , LAT.A_Total_Payment
            , LAT.A_TOTAL_SLN
            , LAT.A_TOTAL_INYR_EXPECTED_GLH
INTO  #0910_F05_Aims_Data_v1
FROM  ILR0910_L05_LEARNER L
            JOIN ILR0910_L05_AIMS LA ON LA.L01 = L.L01

      AND LA.L03 = L.L03
            JOIN ILR0910_L05_AIMS_DLF DLF ON    DLF.L01 = LA.L01

            AND DLF.L03 = LA.L03

            AND DLF.A05 = LA.A05
            JOIN (SELECT L01
                              , L01_Orig
                              , L03
                              , L03_Orig
                              , A05
                              , A_TODATE_QUALIFYING_SLN_PERIOD
                              , A_Total_Payment_Y2D
                              , A_TO_DATE_SLN_PAYMENT
                              , A_Total_Payment
                              , A_TOTAL_SLN
                              , A_TOTAL_INYR_EXPECTED_GLH
                  FROM      #Latest
                  WHERE     Year_ID = 2009) LAT ON LAT.L01_Orig =
LA.L01
                                                                  AND
LAT.L03_Orig = LA.L03
                                                                  AND
LAT.A05 = LA.A05
GO



--Drop table #0506_0910_FE_Data_v1
--Amalgamate all record from across years
SELECT      L01
            , L01_Orig
            , L03
            , L03_Orig
```

```
              , A09
              , A05
              , A10
              , A14
              , A15
              , A18
              , A21
              , A23
              , A26
              , A27
              , A28
              , A31
              , A32
              , A34
              , A35
              , A36
--            , A37
--            , A38
              , A46a
              , A46b
              , A49
              , ISNULL(A50, -1) A50
              , A53
              , A56
              , A_ATYPE
              , A_DPLLSC
--            , A_NVQLEV
              , A_OPROV
              , L_ATYPE
              , L_FUND
              , L_LREG
              , L11
              , L12
              , L13
              , L14
              , L15
              , L16
              , L17
              , L22
              , L25
              , L29
              , L32
              , L34A
              , L34B
              , L34C
              , L34D
              , L35
              , L44
              , L46
              , L_LLLSC
              , L_PLLSC
              , L_PREG
              , A_FE_PROVMIX_MATRIX
              , A_FE_PROVMIX_MATRIX_SUMM
              , L_FUND_ACTIVE
              , A_IY_SLN_PAYMENT
              , A_PRIOR_SLN_PAYMENT
              , A_TODATE_QUALIFYING_SLN_PERIOD
              , A_TOTAL_SLN_Y2D
```

```
            , A_FULLY_FUNDED
            , A_TO_DATE_SLN_PAYMENT
            , A_INYR_EXPECTED_GLH
            , Year_ID
            , p_ExpEndYr
            , P_StartYr
            , P_ActEndYr
--Calculate planned days
            , DateDiff(d, LA.A27, DateAdd(d, 1, LA.A28)) Days
--Calculate Planned course length (yrs)
            , CASE WHEN DateDiff(d,LA.A27,LA.A28)+1 < 24*7 THEN 'Short'
                   WHEN DateDiff(d,LA.A27,LA.A28)+1 <=
DateDiff(d,LA.A27,DateAdd(yy,1,LA.A27)) THEN '1 Yr'
                   WHEN DateDiff(d,LA.A27,LA.A28)+1 <=
DateDiff(d,LA.A27,DateAdd(yy,2,LA.A27)) THEN '2 Yr'
                   WHEN DateDiff(d,LA.A27,LA.A28)+1 <=
DateDiff(d,LA.A27,DateAdd(yy,3,LA.A27)) THEN '3 Yr'
                   ELSE '4 Yr+'
              END NoOfYears
--Calculate No of Academic Years
              , CASE    WHEN DateDiff(d,LA.A27,LA.A28)+1 < 24*7 THEN 1
                        WHEN (LA.p_expendyr-LA.p_startyr) + 1 > 4
THEN 4

            ELSE (LA.p_expendyr-LA.p_startyr) + 1
                 END AcademicYrs

--Calculate Age as at 31st aug
            , DateDiff(YY, LA.L11, CAST('31-Aug-' + CAST(p_startyr as
varchar) as DateTime))
                  - CASE      WHEN DatePart(MM, CAST('31-Aug-' +
CAST(p_startyr as varchar) as DateTime)) < DatePart(MM, L11)
      THEN 1
                              WHEN DatePart(MM, CAST('31-Aug-' +
CAST(p_startyr as varchar) as DateTime)) = DatePart(MM, L11)
                                    AND DatePart(DD, CAST('31-Aug-' +
CAST(p_startyr as varchar) as DateTime)) < DatePart(DD, L11) THEN 1


                                    ELSE 0


                  END A_AGE_31AugStYr
--Transfers
            , CASE      WHEN LA.A34 = 4 THEN 1 ELSE 0 END P_Trans
--Completers
            , CASE      WHEN LA.A34 = 2 THEN 1 ELSE 0 END P_Complete
--Achievers
            , CASE      WHEN LA.A35 = 1 THEN 1 ELSE 0 END P_Achieved
--Leavers
            , CASE      WHEN LA.A31 IS NULL THEN 0 ELSE 1 END P_Leavers
--Redundant (No Ufi records should exists at this stage)
            , CASE      WHEN 1 IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
THEN 1 ELSE 0 END A_UFI
--Indentify OLASS
            , CASE      WHEN 34 IN(ISNULL(LA.A46A,0), ISNULL(LA.A46B,0))
THEN 1 ELSE 0 END A_OLASS
            , A_TOTAL_SLN
            , A_TOTAL_PAYMENT
```

```
            , A_TOTAL_PAYMENT_Y2D
            , A_TOTAL_INYR_EXPECTED_GLH
--Identify Funded records
            , CASE WHEN P_ExpEndYr <= 2009
                              AND LA.A34 != 4
                              AND 34 NOT IN(ISNULL(LA.A46A,0),
ISNULL(LA.A46B,0)) --A_OLASS
                              AND A_TO_DATE_SLN_PAYMENT=1 THEN 1 ELSE 0
              END FUNDED
INTO  #0506_0910_FE_Data_v1
FROM  (SELECT     * FROM      #0506_F05_Aims_Data_v1
            UNION ALL
            SELECT      * FROM      #0607_F05_Aims_Data_v1
            UNION ALL
            SELECT      *     FROM  #0708_F05_Aims_Data_v1
            UNION ALL
            SELECT      *     FROM  #0809_F05_Aims_Data_v1
            UNION ALL
            SELECT      *     FROM  #0910_F05_Aims_Data_v1) LA
GO

--Produce MasterTrim file
SELECT      '2009/10' [Year]
            , 'F05' AS Collection
            , LA.L01
--Original L01 as recorded in the final F05 return
            , LA.L01_Orig
            , LA.L03
--Original L03 as recorded in the final F05 return
            , LA.L03_Orig
            , LA.A09
            , LA.A05
            , LA.A10
            , LA.A14
            , LA.A18
            , LA.A21
            , LA.A23
            , LA.A27
            , LA.A28
            , LA.A31
            , LA.A32
            , LA.A34
            , LA.A35
            , LA.A36
            , LA.A46a
            , LA.A46b
            , LA.A49
            , LA.A50
            , LA.A53
            , LA.A56
            , LA.A_ATYPE
            , LA.A_OPROV
            , LA.L_ATYPE
            , LA.L_FUND
            , LA.L_LREG
            , LA.L11
            , LA.L12
            , LA.L13
            , LA.L14
```

```
              , LA.L15
              , LA.L16
              , LA.L17
              , LA.L22
              , LA.L25
              , LA.L29
              , LA.L32
              , LA.L34A
              , LA.L34B
              , LA.L34C
              , LA.L34D
              , LA.L35
              , LA.L44
              , LA.L46 L46_Orig
              , P.L46
              , LA.L_LLLSC
              , LA.L_PLLSC
              , LA.L_PREG
              , LA.A_FE_PROVMIX_MATRIX
              , LA.A_FE_PROVMIX_MATRIX_SUMM
              , LA.A_IY_SLN_PAYMENT
              , LA.A_TOTAL_PAYMENT_Y2D
              , A_TOTAL_SLN
              , A_TOTAL_PAYMENT
              , LA.A_PRIOR_SLN_PAYMENT
              , LA.A_TO_DATE_SLN_PAYMENT
              , LA.A_TODATE_QUALIFYING_SLN_PERIOD
              , LA.A_TOTAL_SLN_Y2D
              , LA.A_FULLY_FUNDED
              , LA.P_Trans
              , LA.P_Achieved
              , LA.P_Complete
              , LA.P_StartYr
              , LA.P_ActEndYr
              , LA.p_ExpEndYr
              , LA.P_Leavers
              , P.PRV_NAME
              , LA.A_INYR_EXPECTED_GLH
              , LA.A_TOTAL_INYR_EXPECTED_GLH
              , P.PRV_LLSC
              , P.PRV_REG
              , P.PRV_TYPE
              , LA.A_UFI
              , LA.A_OLASS
--Calculate Duration
              , CASE WHEN NoOfYears = 'Short' THEN      1     --'Short'
                     WHEN NoOfYears = '1 Yr' AND AcademicYrs = 1 THEN 2
       --'1 Yr 1 AY'
                     WHEN NoOfYears = '1 Yr' AND AcademicYrs = 2 THEN 3
       --'1 Yr 2 AY'
                     WHEN NoOfYears = '2 Yr' AND AcademicYrs = 2 THEN 4
       --'2 Yr 2 AY'
                     WHEN NoOfYears = '2 Yr' AND AcademicYrs = 3 THEN 5
       --'2 Yr 3 AY'
                     WHEN NoOfYears = '3 Yr' AND AcademicYrs = 3 THEN 6
       --'3 Yr 3 AY'
                     WHEN NoOfYears = '3 Yr' AND AcademicYrs = 4 THEN 7
       --'3 Yr 4 AY'
```

```
             ELSE 8           --'4year or more 4ay or more'
            END A_Duration
--Calculate short Duration
            ,     CASE WHEN Days <= 34 THEN 0         --'Very Short'
                       WHEN Days < 24*7 THEN 1       --'Short'
                                                ELSE 2    --'Long'
                  END    shortdur
            ,     LA.A_AGE_31AugStYr
                       --'DV - Age of the learner as at 31 August
of start year'
--Identify Age Banding
            ,     CASE WHEN A_AGE_31AugStYr Between 0 AND 15 THEN 1
      --'Under 16'
                       WHEN A_AGE_31AugStYr Between 16 AND 18 THEN 2
      --'16-18'
                       WHEN A_AGE_31AugStYr Between 19 AND 20 THEN 3
      --'19-20'
                       WHEN A_AGE_31AugStYr Between 21 AND 24 THEN 4
      --'21-24'
                       WHEN A_AGE_31AugStYr Between 25 AND 59 THEN 5
      --'25-59'
                       WHEN A_AGE_31AugStYr Between 60 AND 120 THEN 6
      --'60 AND over'

            ELSE 9      --'missing age'
                  END    A_AGE_31AugStYr_B
            ,     CASE WHEN A_AGE_31AugStYr Between 0 AND 18 THEN 1
      --'16-18'

         ELSE 2         --'19+'
                  END    A_AGE_31AugStYr_Band
--Calculate P_COUNT_FUNDED
            , CASE WHEN Funded = 1
                       AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15 THEN 1

                                    ELSE 0
              END P_COUNT_FUNDED
--Calculate P_ACHIEVED_FUNDED
            , CASE WHEN Funded = 1
                       AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15
                       AND P_Achieved = 1 THEN 1 ELSE 0
              END P_ACHIEVED_FUNDED
--Calculate P_COMPLETE_FUNDED
            , CASE WHEN Funded = 1
                       AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15
                       AND P_Complete = 1 THEN 1 ELSE 0
              END P_COMPLETE_FUNDED
--Calculate P_COUNT_ALL
            , CASE WHEN P_ExpEndYr <= 2009
                       AND LA.A34 != 4
                       AND 34 NOT IN(ISNULL(LA.A46A,0),
ISNULL(LA.A46B,0)) --A_OLASS
                       AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15
```

```
                                AND A_TODATE_QUALIFYING_SLN_PERIOD=1 THEN 1
ELSE 0
              END P_COUNT_ALL
--Calculate P_ACHIEVED_ALL
            , CASE WHEN P_ExpEndYr <= 2009
                                AND LA.A34 != 4
                                AND 34 NOT IN(ISNULL(LA.A46A,0),
ISNULL(LA.A46B,0)) --A_OLASS
                                AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15
                                AND A_TODATE_QUALIFYING_SLN_PERIOD=1
                                AND P_Achieved = 1 THEN 1 ELSE 0
              END P_ACHIEVED_ALL
--Calculate P_COMPLETE_ALL
            , CASE WHEN P_ExpEndYr <= 2009
                                AND LA.A34 != 4
                                AND 34 NOT IN(ISNULL(LA.A46A,0),
ISNULL(LA.A46B,0)) --A_OLASS
                                AND ISNULL(A_AGE_31AugStYr, 99) NOT Between
0 AND 15
                                AND A_TODATE_QUALIFYING_SLN_PERIOD=1
                                AND P_Complete = 1 THEN 1 ELSE 0
              END P_COMPLETE_ALL
            , CASE Year_ID    WHEN 2004 THEN '2004/05'
                                    WHEN 2005 THEN '2005/06'
                                    WHEN 2006 THEN '2006/07'
                                    WHEN 2007 THEN '2007/08'
                                    WHEN 2008 THEN '2008/09'
                                    WHEN 2009 THEN '2009/10'
                                    WHEN 2010 THEN '2010/11'
                                    ELSE 'Err'
              END Source_Year
            , CASE WHEN Year_ID = 2004 THEN 1 ELSE 0 END  In_FE_0405
            , CASE WHEN Year_ID = 2005 THEN 1 ELSE 0 END  In_FE_0506
            , CASE WHEN Year_ID = 2006 THEN 1 ELSE 0 END  In_FE_0607
            , CASE WHEN Year_ID = 2007 THEN 1 ELSE 0 END  In_FE_0708
            , CASE WHEN Year_ID = 2008 THEN 1 ELSE 0 END  In_LR_0809
            , CASE WHEN Year_ID = 2009 THEN 1 ELSE 0 END  In_LR_0910
            , LAD.ACADEMIC_YEAR_CODE
--Identify Keyskills
            , CASE WHEN LAD.[LEARNING_AIM_TYPE_CODE] = '1327' THEN 1 ELSE
0 END A_KEYSKILLS
--Identify Functional skills
            , CASE WHEN LAD.[LEARNING_AIM_TYPE_CODE] = '1439' THEN 1 ELSE
0 END A_FUNCTIONAL_SKILLS
            , LAD.MAP_CODE_DESC
--If no mpacode use the aim code
            , CASE WHEN ISNULL([MAP_CODE_CODE], '') = '' THEN
LAD.[LEARNING_AIM_REF] ELSE LAD.[MAP_CODE_CODE] END MAP_CODE_CODE
            , LAD.INSP_CODE_CODE
            , LAD.AWARDING_BODY_CODE
            , LAD.LEARNING_AIM_TYPE_CODE
            , LAD.NOTIONAL_LEVEL_V2_CODE
            , LAD.NOTIONAL_NVQ_LEVEL_CODE
            , LAD.AREA_OF_LEARNING_CODE
            , LAD.NOTIONAL_NVQ_WIDTH
            , LAD.LEVEL2_ENTITLEMENT_CAT_CODE
            , LAD.LEVEL2_PERCENTAGE
            , LAD.LEVEL3_ENTITLEMENT_CAT_CODE
```

```
            , LAD.LEVEL3_PERCENTAGE
--Calculate FullLevel 2 aims
            , CASE WHEN LAD.[LEVEL3_ENTITLEMENT_CAT_CODE] IN('1', '2',
'3') AND LAD.[LEVEL3_PERCENTAGE] >= 100   THEN 0
                   WHEN LAD.[LEVEL2_ENTITLEMENT_CAT_CODE] IN('1', '4')
AND LAD.[LEVEL2_PERCENTAGE] >= 100                    THEN 1



                ELSE 0
              END A_FullLevel2
--Calculate FullLevel 3 aims
            , CASE WHEN LAD.[LEVEL3_ENTITLEMENT_CAT_CODE] IN('1', '2',
'3') AND LAD.[LEVEL3_PERCENTAGE] >= 100 THEN 1


               ELSE 0
              END A_FullLevel3
            , LAD.KEY_SKILL_CODE
            , LAD.SSA_TIER1_CODE
            , LAD.SSA_TIER2_CODE
            , LAD.SKILLS_FOR_LIFE
            , LAD.SKILLS_FOR_LIFE_TYPE_CODE
            , LAD.PROGRAMME_WGT_FACTOR_CODE
            , LAD.LSC_LR_WGT_FACTOR_1618_CODE
            , LAD.LSC_LR_WGT_FACTOR_ADULT_CODE
            , LAD.ADDITIONAL_UNIT
            , LAD.UNITISATION_QUAL
            , LAD.DIAGNOSTIC_TEST
            , LAD.TUTORIAL_COMP_STUDIES
            , ROW_NUMBER() OVER(ORDER BY LA.L01) Transaction_ID
INTO  LR_MasterTrim_0910
FROM  #0506_0910_FE_Data_v1 LA
            LEFT JOIN LAD_Hierarchy LAD ON LAD.Learning_AIM_Ref = LA.A09
            LEFT JOIN ILR0910_UPIN_TO_LLSC P ON P.L01 = LA.L01
GO


CREATE Procedure system.usp_TTG_QSR_MatchProcess (@Year_ID int,
@MaxYear_ID int, @ConstrainMatch int)

AS

DECLARE @NextYear_ID int
DECLARE     @RowCheck int

--Reset Match flags for specified Year
UPDATE      A
SET         Matched = 0
            , MatchYear = 0
            , MatchedRank = 0
FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
WHERE Year_ID >= @Year_ID
            AND Matched != 0

--Set Map Year
SELECT @NextYear_ID = @Year_ID + 1
```

```
--Loop around while Map Year is less than or equal to the max year
WHILE @NextYear_ID <= @MaxYear_ID
BEGIN
	--Variable to ensure loop continues if any record is updated
	SELECT	@RowCheck = 1

	WHILE @RowCheck != 0
	BEGIN

		--Perform Matching
		SELECT	A.Year_ID Year1, A.InYearRank InYearRank1,
B.Year_ID Year2, B.InYearRank InYearRank2
		INTO	#Match
		FROM	(	SELECT	Year_ID, L01_New, L03_New, A09, A27,
A28, A31, Max(InYearRank) InYearRank
				FROM	QSR_P12.Raw.QSR_TTG_Merged_0607_0910
A
				WHERE A.Year_ID IN(@Year_ID)
					AND Matched = 0
				GROUP BY Year_ID, L01_New, L03_New, A09,
A27, A28, A31) A
			JOIN (	SELECT	A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A28, A.A31, Max(A.InYearRank) InYearRank
				FROM
	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
					JOIN (	SELECT
	DISTINCT L01_New, L03_New, A09, A27, A28, A31
					FROM
	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
					WHERE
	A.Year_ID IN(@Year_ID)

	AND Matched = 0) B ON	A.L01_New = B.L01_New

				AND	A.L03_New = B.L03_New

				AND	A.A09 = B.A09

				AND	A.A27 = B.A27

				AND	A.A28 = B.A28

				AND	A.A31 = B.A31
				WHERE A.Year_ID IN(@NextYear_ID)
					AND A.Matched = 0
				GROUP BY A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A28, A.A31) B ON	A.L01_New = B.L01_New


	AND	A.L03_New = B.L03_New


	AND	A.A09 = B.A09


	AND	A.A27 = B.A27


	AND	A.A28 = B.A28
```

```
	AND	A.A31 = B.A31

		SELECT @RowCheck = @@RowCount

		UPDATE		A
		SET		Matched = 1
				, MatchYear = B.Year2
				, MatchedRank = B.InYearRank2
		FROM	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
				JOIN #Match B ON	A.Year_ID = B.Year1
							AND A.InYearRank =
B.InYearRank1


		UPDATE		A
		SET		Matched = 1
				, MatchYear = @Year_ID
				, MatchedRank = B.InYearRank1
		FROM	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
				JOIN #Match B ON	A.Year_ID = B.Year2
							AND A.InYearRank =
B.InYearRank2


		DROP Table #Match

	END

	SELECT		@RowCheck = 1

	WHILE @RowCheck != 0
	BEGIN

		SELECT		A.Year_ID Year1, A.InYearRank InYearRank1,
B.Year_ID Year2, B.InYearRank InYearRank2
		INTO	#Match2
		FROM	(	SELECT		Year_ID, L01_New, L03_New, A09, A27,
A28, Max(InYearRank) InYearRank
				FROM	QSR_P12.Raw.QSR_TTG_Merged_0607_0910
A
				WHERE A.Year_ID IN(@Year_ID)
					AND Matched = 0
				GROUP BY Year_ID, L01_New, L03_New, A09,
A27, A28) A
			JOIN (		SELECT		A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A28, Max(A.InYearRank) InYearRank
				FROM
	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
							JOIN (		SELECT
	DISTINCT L01_New, L03_New, A09, A27, A28
							FROM
	QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
							WHERE
	A.Year_ID IN(@Year_ID)

	AND Matched = 0) B ON	A.L01_New = B.L01_New
```

```
                                    AND   A.L03_New = B.L03_New
                                    AND   A.A09 = B.A09
                                    AND   A.A27 = B.A27
                                    AND   A.A28 = B.A28
                                    WHERE A.Year_ID IN(@NextYear_ID)
                                          AND A.Matched = 0
                                    GROUP BY A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A28) B   ON    A.L01_New = B.L01_New

      AND   A.L03_New = B.L03_New

      AND   A.A09 = B.A09

      AND   A.A27 = B.A27

      AND   A.A28 = B.A28

            SELECT @RowCheck = @@RowCount

            UPDATE      A
            SET         Matched = 1
                        , MatchYear = B.Year2
                        , MatchedRank = B.InYearRank2
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match2 B ON A.Year_ID = B.Year1
                                          AND A.InYearRank =
B.InYearRank1

            UPDATE      A
            SET         Matched = 1
                        , MatchYear = @Year_ID
                        , MatchedRank = B.InYearRank1
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match2 B ON A.Year_ID = B.Year2
                                          AND A.InYearRank =
B.InYearRank2

            DROP Table #Match2

      END

---
      SELECT      @RowCheck = 1

      WHILE @RowCheck != 0
      BEGIN
```

```
            SELECT      A.Year_ID Year1, A.InYearRank InYearRank1,
B.Year_ID Year2, B.InYearRank InYearRank2
            INTO  #Match3
            FROM  (     SELECT      Year_ID, L01_New, L03_New, A09, A27,
A31, Max(InYearRank) InYearRank
                        FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910
A
                        WHERE A.Year_ID IN(@Year_ID)
                              AND Matched = 0
                        GROUP BY Year_ID, L01_New, L03_New, A09,
A27, A31) A
                  JOIN (      SELECT      A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A31, Max(A.InYearRank) InYearRank
                              FROM
      QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                                    JOIN (      SELECT
      DISTINCT L01_New, L03_New, A09, A27, A.A31
                                                FROM
      QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                                                WHERE
      A.Year_ID IN(@Year_ID)

      AND Matched = 0) B ON   A.L01_New = B.L01_New

                              AND   A.L03_New = B.L03_New

                              AND   A.A09 = B.A09

                              AND   A.A27 = B.A27

                              AND   A.A31 = B.A31

                              WHERE A.Year_ID IN(@NextYear_ID)
                                    AND A.Matched = 0
                              GROUP BY A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, A.A31) B ON      A.L01_New = B.L01_New


                  AND   A.L03_New = B.L03_New


                  AND   A.A09 = B.A09


                  AND   A.A27 = B.A27


                  AND   A.A31 = B.A31

            SELECT @RowCheck = @@RowCount

            UPDATE      A
            SET         Matched = 1
                        , MatchYear = B.Year2
                        , MatchedRank = B.InYearRank2
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match3 B ON A.Year_ID = B.Year1
                                          AND A.InYearRank =
B.InYearRank1
```

```
            UPDATE      A
            SET         Matched = 1
                        , MatchYear = @Year_ID
                        , MatchedRank = B.InYearRank1
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match3 B ON A.Year_ID = B.Year2
                                                AND A.InYearRank =
B.InYearRank2


            DROP Table #Match3

      END

      SELECT      @RowCheck = 1

      WHILE @RowCheck != 0
      BEGIN

            SELECT      A.Year_ID Year1, A.InYearRank InYearRank1,
B.Year_ID Year2, B.InYearRank InYearRank2
            INTO  #Match4
            FROM  (     SELECT      Year_ID, L01_New, L03_New, A09, A27,
Max(InYearRank) InYearRank
                        FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910
A
                        WHERE A.Year_ID IN(@Year_ID)
                                    AND Matched = 0
                        GROUP BY Year_ID, L01_New, L03_New, A09,
A27) A
                  JOIN (      SELECT      A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27, Max(A.InYearRank) InYearRank
                              FROM
      QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                                                JOIN (      SELECT
      DISTINCT L01_New, L03_New, A09, A27
                                                      FROM
      QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                                                      WHERE
      A.Year_ID IN(@Year_ID)

      AND Matched = 0) B ON   A.L01_New = B.L01_New

                                    AND   A.L03_New = B.L03_New

                                    AND   A.A09 = B.A09

                                    AND   A.A27 = B.A27
                              WHERE A.Year_ID IN(@NextYear_ID)
                                          AND A.Matched = 0
                              GROUP BY A.Year_ID, A.L01_New,
A.L03_New, A.A09, A.A27) B ON A.L01_New = B.L01_New

                                                      AND

      A.L03_New = B.L03_New
```

```
                                                        AND    A.A09 =
B.A09
                                                        AND    A.A27 =
B.A27


            SELECT @RowCheck = @@RowCount

            UPDATE      A
            SET         Matched = 1
                        , MatchYear = B.Year2
                        , MatchedRank = B.InYearRank2
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match4 B ON A.Year_ID = B.Year1
                                                        AND A.InYearRank =
B.InYearRank1

            UPDATE      A
            SET         Matched = 1
                        , MatchYear = @Year_ID
                        , MatchedRank = B.InYearRank1
            FROM  QSR_P12.Raw.QSR_TTG_Merged_0607_0910 A
                        JOIN #Match4 B ON A.Year_ID = B.Year2
                                                        AND A.InYearRank =
B.InYearRank2

            DROP Table #Match4

      END

--Increment Map Year

      SELECT @NextYear_ID = @NextYear_ID + 1

END

RETURN(0)
GO
```