

## **Learner Responsive Success Rates Methodology for 2008/09 – Specification and Supporting Documentation**

**Date of issue**                      January 27, 2010

**LSC office**                              National Office

### **Purpose**

- 1     The purpose of this paper is to provide the business rules and a Learner Responsive success rates methodology for 2008/09.

### **Background**

- 2     This methodology is developed and tested by the LSC Strategic Analysis & Modelling team 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 2008/09.
- 3     The Learner Responsive success rates methodology and the success rate reporting requirements of the FE sector have changed in the past year as a result of the new data collection systems put in place by the Information Authority as a result of the LSC's move to demand-led funding from 2008/09 and as a result of the move by OfSTED, the LSC, BIS, DCSF and other partner organisations to harmonise the different success rate methodologies and reporting methods used across the FE sector.
- 4     The changes in the above paragraph are unprecedented in recent years and have required extensive work to manipulate previous year's data in accordance with agreed business rules, definitions and processes to align with 08/09 in order that the multiple ILR files can be merged.

## Summary

5 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.

6 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 2008/09 success rate calculation is the number of aims due to be completed in 2008/09 that have been successfully completed divided by the number of aims that are due to be completed in 2008/09, 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 9.

7 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

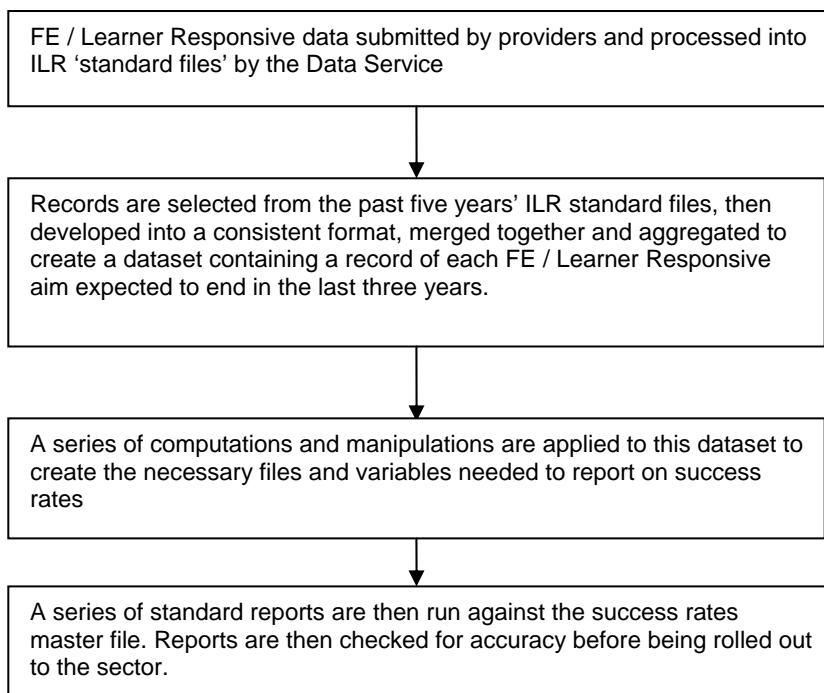
8 Some learning aims are excluded from success rate calculations:

- University for Industry (Ufi)
- Offenders' Learning and Skills Service (Olass)
- Qualification and Credit Framework Units (QCF) (**applicable to 2008/09 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 only**)
- National Voluntary Training Provider (NVTP) (**applicable to 2008/09 only**)
- Flexibilities (**applicable to 2008/09 only**)
- Entry 2 Employment (E2E) (**applicable to 2008/09 only**)

- Adult Learner Accounts (ALA) (**applicable to 2007/08 and 2008/09 only**)
- First Steps (**applicable to 2008/09 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

- 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 a glossary of the variables used in the success rates process.

***\*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**

- 11 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 now 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 previous years (2004/05 – 2007/08)**

- 12 **2004/05 Data** – Match the **2004/05 F05 Learner** standard file to the **2004/05 F05 Aims** standard file by the variables **L01** and **L03**. Then, match to the **2004/05 F05 Funding Aims** file by the variables **L01 L03** and **A05**. Next, match the **2008/09 LAD Hierarchy** file by the variable **A09**.

In order to align the 2004/05 dataset with the 2008/09 methodology, it is necessary to compute a variable that identifies those learning aims that would qualify for funding under the new methodology as, **a\_to\_date\_qualifying\_sln\_period** .

Long learning aims must be active for at least a 6 week qualifying period. If the difference between the learning start date (**A27**) and the learning actual end date (**A31**) is greater than 6 weeks then the aim should be flagged as 1 (**a\_to\_date\_qualifying\_sln\_period = 1**).

However, long aims greater than 24 weeks where the learner has withdrawn (the record must be flagged as withdrawn i.e. **A34 = 3**) within 6 weeks and where no funding payment has been made will be removed from the success rates.

Aim length is calculated using Actual Start Date (**A27**) and Planned End Date (**A28**).

Length of stay is calculated using Actual Start Date (**A27**) and Actual End Date (**A31**).

In order to flag funded learners, compute the variable **a\_to\_date\_sln\_payment** equals 1 if (**A10=20**).

- 13 **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 **2008/09 LAD Hierarchy** file by the variable **A09**.
- 14 **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 **2008/09 LAD Hierarchy** file by the variable **A09**.
- 15 **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 **2008/09 LAD Hierarchy** file by the variable **A09**.
- 16 Apply the following business rules and exclusions to **each year above**:
- Select records which are expected to end in or after 2006/07 (**p\_expendyr>=2006**)
  - 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 Ufl 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 previous 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

- 17 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.
- 18 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\_0405**, **In\_FE\_0506** etc.).

### Stage 1c – Mergers

- 19 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

- 20 The business rules specified in the next steps must be applied to the current year's dataset (2008/09).
- 21 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.
- 22 Then apply the following business rules and exclusions:
  - E2E (Select cases where **A15<>9**)
  - Select cases which are expected to end in or after 2006/07 (**p\_expendyr >=2006**)
  - ETP (Select cases where **A46a and A46b <>17**)

- Exclude Ufl 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**)
- 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**).

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

- 23 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 based on **L01, L03, A09, A27, p\_startyr** and **p\_expendyr**.
- 24 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\_0405, In\_FE\_0506, In\_FE\_0607, In\_FE\_0708, In\_LR\_0809** and **L01\_orig**).

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

- 25 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.
- 26 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**

- 27 If the **map\_code\_code** variable is missing then set the **map\_code\_code** to equal the **A09** field.

### **Stage 3b – Compute transfers**

- 28 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**

- 29 If learning outcome is equal to (1) achieved (**A35= 1**), **p\_achieved** is set to 1.

### **Stage 3d – Compute completed aims**

- 30 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**

- 31 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 2007/08 **p\_startyr** has the value of 2007.
- 32 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 2007/08 **p\_actendyr** has the value of 2007.

- 33 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 2007/08 **p\_expendyr** has the value of 2007.

### Stage 3f – Identify those who have left their learning aim

- 34 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

- 35 Compute a **p\_years** variable to be used in creating the duration variable in the next step. Take the minimum of the planned end year (**p\_expendyr**) and start year (**p\_startyr**) and add 1, but a maximum of 4. Adding the one takes into account aims that have started and are planned to complete in the same year.
- 36 Compute a duration variable which flags the learning aim over the academic years.

Compute a\_duration is equal to -1.

If the difference between the planned end date (**A28**) and the start date (**A27**) is less than 24 weeks flag the duration as short (**a\_duration=1**) (**Short**).

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** equals to 1 (**p\_years=1**) then flag the duration as 1 year 1 academic year (**a\_duration=2**) (**1 year\_1ay**).

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** equals to 2 (**p\_years=2**) then flag the duration as 1 year 2 academic years (**a\_duration=3**) (**1 year\_2ay**).

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** equals to 3 (**p\_years=3**) then flag the duration as 1 year 3 academic years (**a\_duration=4**) (**1 year\_3ay**).

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 less than or equal to 2 (**p\_years<=2**) then flag the duration as 2 year 2 academic years (**a\_duration=5**) (**2year\_2ay**).

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** equals 3 (**p\_years=3**) then flag the duration as 2 year 3 academic years (**a\_duration=6**) (**2year\_3ay**).

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** equals to 4 (**p\_years=4**) then flag the duration as 2 year 4 academic years (**a\_duration=7**) (**2year\_4ay**).

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 less than or equal to 3 (**p\_years<=3**) then flag the duration as 3 year 3 academic years (**a\_duration=8**) (**3 year\_3ay**).

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** equals 4 (**p\_years=4**) then flag the duration as 3 year 4 academic years (**a\_duration=9**) (**3 year\_4ay**).

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** equals 5 (**p\_years=5**) then flag the duration as 3 year 5 academic years (**a\_duration=10**) (**3 year\_5ay**).

If the difference between the planned end date (**A28**) and the start date (**A27**) is greater than or equal to 4 years then flag the duration as 4 year or more 4 academic years (**a\_duration=11**) (**4year or more 4ay or more**).

### **Stage 3h – Compute short duration**

37 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**

38 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**

39 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**

40 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**

41 If the **learning aim type code** is equal to “1327”, **a\_keyskills** is set to 1.

### **Stage 3m – Compute Functional\_Skills**

42 If the **learning aim type code** is equal to “1439”, **a\_functional\_skills** is set to 1.

### **Stage 3n – Compute Olass (Offenders in Custody)**

43 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**

44 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.

45 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 2006 and 2008 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 31<sup>st</sup> of August of the start aim must not be under 16 (**a\_age\_31augstyr <>1**) AND the learning aim must be a funded aim (**a\_to\_date\_sl\_n\_payment=1**). If these criteria are met then compute **p\_achieved\_funded** is **A35** equals 1.

### **Stage 3p – Tidying up the raw data**

46 All the derived fields with ‘system missing’ or ‘null’ values need to be set to 0.

### **Stage 3q – Compute Source Year variable**

47 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\_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. If **In\_FE\_0405** is equal to 1 then set **source\_year** equal to 2004/05.

### **Stage 3r – Compute Year variable**

48 Compute (**year**) is equal to 2008/2009.

#### **Stage 4 – Preparing the final datasets for success rates reporting**

- 49 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, 'ILR0809\_UPIN\_TO\_LLSC').
- 50 Once that step has been taken, the main success rates masterfile (known as the 'mastertrim' file) is ready.
- 51 The dataset is now ready for use in standard reporting.

#### **Changes to Learner Reference Numbers between years**

- 52 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.
- 53 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.
- 54 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.***

## **Annex A – Fields in the Mastertrim**

### **\*ILR Standard File Variables from Learner and Aims File**

**L01** – Provider number  
**L03** – Learner reference number  
**A09** – Learning aim reference  
**A05** – Learning aim data set sequence  
**A10** – LSC funding stream  
**A14** – Reason for full funding / co-funding of learning aim  
**A18** – Main delivery method  
**A21** – Franchised out and partnership arrangement  
**A23** – Delivery location postcode  
**A27** – Learning start date  
**A28** – Learning planned end date  
**A31** – Learning actual end date  
**A32** – Guided learning hours  
**A34** – Completion status  
**A35** – Learning outcome  
**A36** – Learning outcome grade  
**A37** – Number of units completed  
**A38** – Number of units to achieve full qualification  
**A46a** – National learning aim monitoring  
**A46b** - National learning aim monitoring  
**A49** – Special projects and pilots  
**A50** – Reason learning ended  
**A53** – Additional learning needs  
**A56** – UK provider reference number  
**A\_ATYPE** – Type of aim  
**A\_NVQLEV** - Notional NVQ Level of the Aim  
**A\_OPROV** - Aim other provision  
**L\_ATYPE** - In Year Aim Type for a Learner  
**L\_FUND** - Sources of funding for the learner  
**L\_LREG** - Home Region of Learner Postcode  
**L11** – Date of birth  
**L12** - Ethnicity  
**L13** - Gender  
**L14** – Learning difficulties  
**L15** - Disability  
**L16** – Learning difficulty  
**L17** – Home postcode  
**L22** – Current postcode  
**L25** – LSC number of funding LSC  
**L29** – Additional learning support  
**L32** – Eligibility of disadvantage uplift  
**L34A** – Learner support reason  
**L34B** - Learner support reason  
**L34C** - Learner support reason  
**L34D** - Learner support reason

**L35** – Prior attainment level  
**L44** – NES delivery LSV number  
**L46** – UK provider reference number  
**L46\_orig** – UK provider reference number  
**L\_LLLSC** - Home LSC of Learner Postcode  
**L\_PLLSC** - LLSC based on the 1-1 'owner' relationship UPIN to LLSC  
**L\_PREG** - Region of the Owning Provider  
**A\_SSA\_T1** - LAD SSA Tier 1 Code  
**A\_SSA\_T2** - LAD SSA Tier 2 Code  
**A\_FE\_PROVMIX\_MATRIX** - FE provision mix matrix  
**A\_FE\_PROVMIX\_MATRIX\_SUMM** - Provision Mix Matrix Summary  
**A\_INYR\_EXPECTED\_GLH** - Expected current year guided learning hours

\*DLF Aims Lookup

**A\_IY\_SLN\_PAYMENT** – Does the aim generate SLN or receive an in year payment, up to the current period  
**A\_PRIOR\_SLN\_PAYMENT** – Has the aim generated SLN or received a payment in previous years  
**A\_TO\_DATE\_SLN\_PAYMENT** – Learner generated SLN or received a payment in previous years or in this year  
**A\_TODATE\_QUALIFYING\_SLN\_PERIOD** – Aim passes the qualifying period in any year to be eligible to generate SLN  
**A\_TOTAL\_SLN\_Y2D** – Total SLN year to date  
**A\_FULLY\_FUNDED** – Aim fully funded  
**A\_TOTAL\_PAYMENT\_Y2D** – The total payment for the aim for the current year

\*Variables Derived as part of the Methodology

**In\_FE\_0405** – Aim is in the FE dataset in 2004/05  
**In\_FE\_0506** – Aim is in the FE dataset in 2005/06  
**In\_FE\_0607** – Aim is in FE dataset in 2006/07  
**In\_FE\_0708** – Aim is in the FE dataset in 2007/08  
**In\_LR\_0809** – Aim is in the Learner Responsive dataset in 2008/09  
**A\_DURATION** – Calculates the expected duration of the learning aim  
**SHORTDUR** – Calculates short learning aims  
**P\_TRANS** – transfer to another learning aim (**A34=4**) (*excluded from success rate calculation*)  
**P\_ACHIEVED** – learning aim has been achieved (**A35=1**)  
**P\_COMPLETE** – learning aim has been completed (**A34=2**)  
**P\_STARTYR** – Calculates the start year of the learning aim  
**P\_ACTENDYR** - Calculates the actual end year of the learning aim  
**P\_EXPENDYR** – Calculates the expected end year of the learning aim  
**P\_LEAVERS** – Calculates if the learner has left the learning aim (**date in the A31 field**)  
**P\_COUNT\_FUNDED** – cohort for funded aims (*denominator in calculating success rates, excludes transfers, Olass and under 16 year olds*)  
**P\_COUNT\_ALL** – Cohort for unfunded and funded aims (*excludes transfers, Olass and under 16 year olds*)

**A\_AGE\_31AugStYr** – Age of learner at the 31<sup>st</sup> of August of the start of the aim  
**A\_AGE\_31AugStYr\_b** - Age of learner at the 31<sup>st</sup> of August of the start of the aim banded  
**A\_AGE\_31AugStYr\_Band** - Age of learner at the 31<sup>st</sup> of August of the start of the aim banded by 16-18 and 19+  
**YEAR** – Current academic year for the mastertrim (2008/09)  
**SOURCE\_YEAR** – The relevant year which the learning aim is sourced from, for example if source year is 2007/08 the record is from the 2007/08 dataset  
**A\_KEYSILLS** – Learning aim type ‘1327’. Keyskills flag (***excluded from success rate calculation***)  
**A\_FUNCTIONAL\_SKILLS** – Learning aim type ‘1439’. Functional Skills flag (***excluded from success rate calculation***)  
**A\_OLASS** – Offenders flag (***excluded from success rate calculation***)  
**A\_TOTAL\_EXPECTED\_GLH** – Total guided learning hours over the lifetime of the aim  
**P\_ACHIEVED\_ALL** - Achievers for unfunded and funded aims (***excludes transfers, Olass and under 16 year olds***)  
**P\_ACHIEVED\_FUNDED** – Achievers for funded aims (numerator for success rate calculation, ***excludes transfers, Olass and under 16 year olds***)  
**A\_TOTAL\_SLN** – The total SLN for the lifetime of the aim  
**A\_TOTAL\_PAYMENT** – The total payments for the lifetime of the aim  
**P\_COMPLETE\_FUNDED** - Completed the learning aim (funded only)  
**L01\_orig** – Original upin before merger was applied  
**L46\_orig** – Original UK provider reference number

\*Upin to LLSC Lookup

**PRV\_NAME** – Provider name  
**PRV\_LLSC** – Provider LSC  
**PRV\_REG** – Provider region  
**PRV\_TYPE** – Provider type

\*2008/09 Lad Hierarchy Fields

**ACADEMIC\_YEAR\_CODE** – The teaching year to which the annual values relate  
**MAP\_CODE\_DESC** -  
**MAP\_CODE\_CODE** - A code used to group learning aims together in a hierarchy for the purpose of analysis / benchmarking  
**INSP\_CODE\_CODE** – A code used to group learning aims together in a hierarchy for Inspection purposes  
**AWARDING\_BODY\_CODE** - A code to identify the awarding body associated with each learning aim record  
**LEARNING\_AIM\_TYPE\_CODE** - Code used by the Learner Information Suite to assign funding units to certain listed learning aims  
**NOTIONAL\_LEVEL\_V2\_CODE** – A level on the NVQ scale for all learning aims. Enables learning aims to be analysed against the NVQ scale  
**NOTIONAL\_NVQ\_LEVEL\_CODE** – A level on the NVQ scale for all learning aims. Enables learning aims to be analysed against NVQ scale  
**AREA\_OF\_LEARNING\_CODE** – A code to classify the learning aim using the Adult Learning Inspectorate (ALI) areas of learning

**NOTIONAL\_NVQ\_WIDTH** – Gives the proportion of a full NVQ that the amount of material covered in a learning aim represents

**LEVEL2\_ENTITLEMENT\_CAT\_CODE** – A code to indicate the category or family to which the learning aims belongs for the purpose of assessing how far the aim contributes to a Full Level 2

**LEVEL2\_PERCENTAGE** – The percentage of Full Level 2 which a learning aim contributes

**LEVEL3\_ENTITLEMENT\_CAT\_CODE** - A code to indicate the category or family to which the learning aims belongs for the purpose of assessing how far the aim contributes to a Full Level 3

**LEVEL3\_PERCENTAGE** - The percentage of Full Level 3 which a learning aim contributes

**SSA\_TIER1\_CODE** – The broad (Tier 1) classification of the subject of learning

**SSA\_TIER2\_CODE** – The more specialised classification (Tier 2) of the subject of a learning aim

**SKILLS\_FOR\_LIFE** - Indicates the learning aim is considered a skill for life relating to achievements in literacy, numeracy and language and counts towards Skills for Life national targets

**SKILLS\_FOR\_LIFE\_TYPE\_CODE** – A sub-category identifying the type of basic skill (e.g. Adult literacy, ESOL)

**ADDITIONAL\_UNIT** – Learning aim type X901

**UNITISATION\_QUAL** – Learning aim type 8008 and 8009

**DIAGNOSTIC\_TEST** – Learning aim type 8011

**TUTORIAL\_COMP\_STUDIES** – A09 is CMISC001 or XESF0001

## Mastertrim SQL Syntax (SQL Server 2005 compliant)

```
SELECT 2004 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
--Calculate Expected End Year
      , 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 A05
--Not a derived variable in 2004/05 there calculate
      , CASE WHEN datediff(d, a27,a31)/7 >=6 THEN 1 ELSE 0 END
A_TODATE_QUALIFYING_SLN_PERIOD
--Not a derived variable in 2004/05 there calculate
      , CASE WHEN datediff(d, a27,a31)/7 >=6 AND LA.A10 = 20 THEN 1 ELSE 0
END A_TO_DATE_SLN_PAYMENT
      , Tot_Fund A_TOTAL_PAYMENT_Y2D
      , Cast(0 As Numeric(20, 7))A_TOTAL_SLN_Y2D
      , LA.a_exp_b a_inyr_expected_glh
--Calculate P_StartYear
      , 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
--Insert into a temporary table
INTO #LA
FROM ILR0405_F05_AIMS LA
      JOIN ILR0405_F05_LEARNER L ON LA.L01 = L.L01
      AND LA.L03 = L.L03
      LEFT JOIN HIER_20091127_0809_PSV LAD ON LAD.Learning_AIM_Ref =
LA.A09
      LEFT JOIN ILR0405_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 2006 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 >= 2006
--Funded by LSC or Other LSC
      AND L.L_FUND IN(1,2)
--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 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

--Append records into a temporary table
INSERT INTO #LA
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
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 HIER_20091127_0809_PSV 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 2006 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 >= 2006
--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_AGE8 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 HIER_20091127_0809_PSV 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 2006 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 >= 2006
--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_AGE8 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 HIER_20091127_0809_PSV 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 2006 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 >= 2006
--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 HIER_20091127_0809_PSV LAD ON LAD.Learning_AIM_Ref =
LA.A09
WHERE
--Only include Aims with a planned end date of 2006 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 >= 2006
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

--Enforce six week rule for records not eligible for funding and not received
funding
DELETE
FROM #LA
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

--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 Mergers M ON M.L01 = L.L01
                        AND Mrg_Academic_Year < 8
                        AND M.L01 != M.L01_New
      LEFT JOIN Mergers_L03 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 = M2.L01_New
                                AND M.L03 = M2.L03_New

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

GO

--Produce a temporary store
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 MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809
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)
EXEC QSR.usp_LR_QSR_MatchProcess 2004, 2008, 0

GO

--Call matching process for 2005 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2005, 2008, 0

GO

--Call matching process for 2006 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2006, 2008, 0

GO

--Call matching process for 2007 data, passing in start and end year

```

```

EXEC QSR.usp_LR_QSR_MatchProcess 2007, 2008, 1
GO

--Call matching process for 2008 data, passing in start and end year
EXEC QSR.usp_LR_QSR_MatchProcess 2008, 2009, 1
GO

--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 MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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 calculate 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
INTO #Latest
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
MatchYear END
GROUP BY Case WHEN MatchYear = 0 THEN Year_ID ELSE
, Case WHEN MatchedRank = 0 THEN Rank
```

```

--Remove FE NVQ records after matching
DELETE
FROM #Latest
WHERE FE_NvQ = 1
      AND Year_ID Between 2004 AND 2007
GO

--0405 FE Data.
--Get all record details from the relevant year based on the
--unmatched records
SELECT 2004 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
, 0 L_FUND_ACTIVE
, 0 A_IY_SLN_PAYMENT
, 0 A_PRIOR_SLN_PAYMENT
, 0 A_TOTAL_SLN_Y2D
, 0 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 #0405_F05_Aims_Data_v1
FROM ILR0405_F05_LEARNER L
      JOIN ILR0405_F05_AIMS LA ON LA.L01 = L.L01
                                AND LA.L03 = L.L03
      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 = 2004) LAT ON LAT.L01_Orig = LA.L01
                                AND
LAT.L03_Orig = LA.L03
                                AND LAT.A05
= LA.A05

```

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.A37
, LA.A38
, LA.A46a
, LA.A46b
, LA.A49
, LA.A50
, LA.A53
, LA.A_ATYPE
, LA.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
, 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)

```

```

LA.A31)-1
        WHEN DatePart(m, LA.A31) < 8
        THEN DatePart(yy,

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
--
        , DLF.L_ACTIVE
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.A37
, LA.A38
, LA.A46a
, LA.A46b
, LA.A49
, LA.A50
, LA.A53
, LA.A_ATYPE
, LA.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
, 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.A37
      , LA.A38
      , LA.A46a
      , LA.A46b
      , LA.A49
      , LA.A50

```

```

, LA.A53
, LA.A_ATYPE
, LA.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
, 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

--Drop table #0405_0809_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 Wks
, DateDiff(Wk, LA.A27, DateAdd(d, 1, LA.A28)) Wks
--Calculate Yrs days
, CASE WHEN DATEPART(day, LA.A27) > DATEPART(day, DateAdd(d, 1,
LA.A28)) THEN DATEDIFF(month, LA.A27, DateAdd(d, 1, LA.A28)) - 1
                                                                ELSE DATEDIFF(month,
LA.A27, DateAdd(d, 1, LA.A28)) END / 12 Yrs
--Calculate Planned course length (yrs)
, CASE WHEN (p_expendyr-p_startyr) + 1 > 4 THEN 4 ELSE (p_expendyr-
p_startyr) + 1 END P_Years
--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

```

```

--Identify 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 <= 2008
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 #0405_0809_FE_Data_v1
FROM (SELECT * FROM #0405_F05_Aims_Data_v1
UNION ALL
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) LA
GO

--Produce MasterTrim QSR file
SELECT '2008/09' [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.A37
, LA.A38
, LA.A46a
, LA.A46b
, LA.A49
, LA.A50
, LA.A53
, LA.A56
, LA.A_ATYPE
, LA.A_NVQLEV
, 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 short Duration
, CASE WHEN Days <= 34 THEN 0 --'Very Short'
      WHEN Days <= 167 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'
--'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 <= 2008
                                AND LA.A34 != 4
                                AND 34 NOT IN(ISNULL(LA.A46A,0)),
ISNULL(LA.A46B,0)) --A_OCLASS
                                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 <= 2008
                                AND LA.A34 != 4
                                AND 34 NOT IN(ISNULL(LA.A46A,0)),
ISNULL(LA.A46B,0)) --A_OCLASS
                                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 <= 2008
                                AND LA.A34 != 4
                                AND 34 NOT IN(ISNULL(LA.A46A,0)),
ISNULL(LA.A46B,0)) --A_OCLASS
                                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'
                                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
        , 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 mpcode 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
INTO LR_MasterTrim_0809
FROM #0405_0809_FE_Data_v1 LA
        LEFT JOIN HIER_20091127_0809_PSV LAD ON LAD.Learning_Aim_Ref =
LA.A09
        LEFT JOIN ILR0809_UPIN_TO_LLSC P ON P.L01 = LA.L01
GO

```

```

CREATE Procedure QSR.usp_LR_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    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.Rank Rank1, B.Year_ID Year2, B.Rank Rank2
        INTO #Match
        FROM (
            SELECT Year_ID, L01_New, L03_New, A09, A27, A28, A31,
Max(Rank) Rank
            FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.Rank) Rank
            FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
            JOIN ( SELECT DISTINCT
L01_New, L03_New, A09, A27, A28, A31
            FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.Rank2
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match B ON A.Year_ID = B.Year1
AND A.Rank = B.Rank1

UPDATE A
SET Matched = 1
, MatchYear = @Year_ID
, MatchedRank = B.Rank1
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match B ON A.Year_ID = B.Year2
AND A.Rank = B.Rank2

DROP Table #Match

END

SELECT @RowCheck = 1

WHILE @RowCheck != 0
BEGIN

SELECT A.Year_ID Year1, A.Rank Rank1, B.Year_ID Year2, B.Rank Rank2
INTO #Match2
FROM ( SELECT Year_ID, L01_New, L03_New, A09, A27, A28,
Max(Rank) Rank
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.Rank) Rank
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN ( SELECT DISTINCT
L01_New, L03_New, A09, A27, A28
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.Rank2
FROM    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN #Match2 B ON   A.Year_ID = B.Year1
        AND A.Rank = B.Rank1

UPDATE A
SET      Matched = 1
        , MatchYear = @Year_ID
        , MatchedRank = B.Rank1
FROM    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN #Match2 B ON   A.Year_ID = B.Year2
        AND A.Rank = B.Rank2

DROP Table #Match2

END

---
SELECT @RowCheck = 1

WHILE @RowCheck != 0
BEGIN

    SELECT A.Year_ID Year1, A.Rank Rank1, B.Year_ID Year2, B.Rank Rank2
    INTO #Match3
    FROM (
        SELECT Year_ID, L01_New, L03_New, A09, A27, P_ExpEndYr,
A31, Max(Rank) Rank
        FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
            WHERE A.Year_ID IN(@Year_ID)
            AND Matched = 0
            GROUP BY Year_ID, L01_New, L03_New, A09, A27,
P_ExpEndYr, A31) A
        JOIN ( SELECT A.Year_ID, A.L01_New, A.L03_New, A.A09,
A.A27, A.P_ExpEndYr, A.A31, Max(A.Rank) Rank
        FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
            JOIN ( SELECT DISTINCT
L01_New, L03_New, A09, A27, P_ExpEndYr, A.A31
        FROM
            MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        WHERE
            A.Year_ID IN(@Year_ID)
            AND Matched = 0) B ON A.L01_New = B.L01_New

```

```

B.L03_New
AND A.A09 = B.A09
AND A.A27 = B.A27
AND A.P_ExpEndYr = B.P_ExpEndYr
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.P_ExpEndYr, 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.P_ExpEndYr = B.P_ExpEndYr
AND A.A31 = B.A31
SELECT @RowCheck = @@RowCount
UPDATE A
SET Matched = 1
, MatchYear = B.Year2
, MatchedRank = B.Rank2
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match3 B ON A.Year_ID = B.Year1
AND A.Rank = B.Rank1
UPDATE A
SET Matched = 1
, MatchYear = @Year_ID
, MatchedRank = B.Rank1
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match3 B ON A.Year_ID = B.Year2
AND A.Rank = B.Rank2
DROP Table #Match3
END
SELECT @RowCheck = 1
WHILE @RowCheck != 0
BEGIN
SELECT A.Year_ID Year1, A.Rank Rank1, B.Year_ID Year2, B.Rank Rank2
INTO #Match4
FROM (
SELECT Year_ID, L01_New, L03_New, A09, A27, P_ExpEndYr,
Max(Rank) Rank
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
WHERE A.Year_ID IN(@Year_ID)
AND Matched = 0
GROUP BY Year_ID, L01_New, L03_New, A09, A27,
P_ExpEndYr) A

```

```

        JOIN ( SELECT A.Year_ID, A.L01_New, A.L03_New, A.A09,
A.A27, A.P_ExpEndYr, Max(A.Rank) Rank
                FROM
                MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
                JOIN ( SELECT DISTINCT
L01_New, L03_New, A09, A27, P_ExpEndYr
                        FROM
                        MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.P_ExpEndYr = B.P_ExpEndYr
                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.P_ExpEndYr) 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.P_ExpEndYr = B.P_ExpEndYr

SELECT @RowCheck = @@RowCount

UPDATE A
SET      Matched = 1
        , MatchYear = B.Year2
        , MatchedRank = B.Rank2
FROM    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN #Match4 B ON A.Year_ID = B.Year1
        AND A.Rank = B.Rank1

UPDATE A
SET      Matched = 1
        , MatchYear = @Year_ID
        , MatchedRank = B.Rank1
FROM    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN #Match4 B ON A.Year_ID = B.Year2
        AND A.Rank = B.Rank2

DROP Table #Match4

END

IF @ConstrainMatch = 0
BEGIN
    SELECT @RowCheck = 1

    WHILE @RowCheck != 0
    BEGIN

        SELECT A.Year_ID Year1, A.Rank Rank1, B.Year_ID Year2, B.Rank
Rank2
        INTO #Match5

```

```

FROM ( SELECT Year_ID, L01_New, L03_New, A09,
P_StartYr, P_ExpEndYr, A31, Max(Rank) Rank
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
WHERE A.Year_ID IN(@Year_ID)
AND Matched = 0
GROUP BY Year_ID, L01_New, L03_New, A09,
P_StartYr, P_ExpEndYr, A31) A
JOIN ( SELECT A.Year_ID, A.L01_New, A.L03_New,
A.A09, A.P_StartYr, A.P_ExpEndYr, A.A31, Max(A.Rank) Rank
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN ( SELECT DISTINCT L01_New,
L03_New, A09, P_StartYr, P_ExpEndYr, A.A31
FROM
MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.P_StartYr = B.P_StartYr
AND A.P_ExpEndYr = B.P_ExpEndYr
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.P_StartYr,
A.P_ExpEndYr, A.A31) B ON A.L01_New =
B.L01_New
AND A.L03_New =
B.L03_New
AND A.A09 = B.A09
AND A.P_StartYr =
B.P_StartYr
AND A.P_ExpEndYr =
B.P_ExpEndYr
AND A.A31 = B.A31
SELECT @RowCheck = @@RowCount
UPDATE A
SET Matched = 1
, MatchYear = B.Year2
, MatchedRank = B.Rank2
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match5 B ON A.Year_ID = B.Year1
AND A.Rank =
B.Rank1
UPDATE A
SET Matched = 1
, MatchYear = @Year_ID
, MatchedRank = B.Rank1
FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
JOIN #Match5 B ON A.Year_ID = B.Year2
AND A.Rank =
B.Rank2

```

```

DROP Table #Match5

END

SELECT @RowCheck = 1

WHILE @RowCheck != 0
BEGIN

    SELECT A.Year_ID Year1, A.Rank Rank1, B.Year_ID Year2, B.Rank
Rank2
    INTO #Match6
    FROM ( SELECT Year_ID, L01_New, L03_New, A09,
P_StartYr, P_ExpEndYr, Max(Rank) Rank
    FROM
    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
    WHERE A.Year_ID IN(@Year_ID)
    AND Matched = 0
    GROUP BY Year_ID, L01_New, L03_New, A09,
P_StartYr, P_ExpEndYr) A
    JOIN ( SELECT A.Year_ID, A.L01_New, A.L03_New,
A.A09, A.P_StartYr, A.P_ExpEndYr, Max(A.Rank) Rank
    FROM
    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
    JOIN ( SELECT DISTINCT
L01_New, L03_New, A09, P_StartYr, P_ExpEndYr
    FROM
    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 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.P_StartYr = B.P_StartYr

    AND A.P_ExpEndYr = B.P_ExpEndYr
    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.P_StartYr, A.P_ExpEndYr) B ON
    A.L01_New = B.L01_New

    AND A.L03_New = B.L03_New

    AND A.A09 = B.A09

    AND A.P_StartYr = B.P_StartYr

    AND A.P_ExpEndYr = B.P_ExpEndYr

    SELECT @RowCheck = @@RowCount

    UPDATE A
    SET
    Matched = 1
    , MatchYear = B.Year2
    , MatchedRank = B.Rank2
    FROM MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
    JOIN #Match6 B ON A.Year_ID = B.Year1
    AND A.Rank =
B.Rank1

```

```

UPDATE A
SET      Matched = 1
        , MatchYear = @Year_ID
        , MatchedRank = B.Rank1
FROM    MI_Warehouse_Staging.Temp.QSR_LR_Merged_0405_0809 A
        JOIN #Match6 B ON  A.Year_ID = B.Year2
                        AND A.Rank =
B.Rank2

DROP Table #Match6

END

END

--Increment Map Year
SELECT @NextYear_ID = @NextYear_ID + 1

END

```