



Leading learning and skills

Further Education Success Rates reporting for the academic year 2007/08

Methodology document

Date of issue 7 July 2009

Background

1. This document gives a detailed description of how the success rates files are calculated.
2. A separate set of guidance notes are also available to accompany Qualification Success Rate Reports.

Introduction

3. Success rates are a key indicator of the quality of provision delivered in the FE sector. They are used by providers to drive quality improvements as well as being used by the LSC in performance management and to inform commissioning decisions and Ofsted to support inspections.
4. This document outlines the source data files that are used to calculate 2007/08 FE success rates, including those submitted by providers. It also explains how variables are derived to support the success rate calculation process and how these are transformed into the base data files that underpin the reports visible to providers.
5. The detail behind the success rates calculations is contained in Annex A, which offers the technical detail behind the calculation.

Source data

6. This section lists which data sets have been used in calculating success rates.
 - Individualised Learning Record (ILR) Learning Aim dataset
 - ILR Learner dataset
 - Learning Aims Hierarchy table
 - Institution Lookup table

Each of these are covered in more detail below:

ILR Learning Aim dataset

7. What is it – The ILR Learning Aim dataset is an aggregation of all the most recent submissions of F05 learning aim data submitted by FE funded providers to the Data Service (previously the LSC). Five years worth of data are used, from 2003/04 up to and including 2007/08.
8. Where it comes from – The data is provided to the Data Service (previously the LSC) by FE funded providers. A number of data fields are used to generate Success Rate files. Detailed definitions including descriptions of each field is included in Annex B.

ILR Learner dataset

9. What is it – The ILR learner dataset is extracted from the same data returns used to produce the ILR learning aim dataset. The ILR Learner dataset is an aggregation of all the most recent submissions of F05 learner data submitted by FE funded providers to the Data Service (previously the LSC). Five years worth of data are used, from 2003/04 up to and including 2007/08.
10. Where it comes from – The data is provided to the Data Service (previously the LSC) by FE funded providers. A number of data fields are used to generate Success Rate files. Detailed definitions including descriptions of each field is included in Annex B.

Learning Aims Hierarchy Table

11. What is it – The purpose of this table is to group similar qualifications together for reporting purposes. For example, qualifications that are the same subject, but may have a different learning aim reference (A09) due to being a different syllabus, are grouped together for success rate reporting purposes.
12. Where it comes from – The Learning Aims Hierarchy table is contained within the Learning Aims Database (LAD). Detailed definitions including descriptions of each field is included in Annex B.

Institution Lookup Table

13. What is it – The Institution Lookup table contains information regarding providers to support the calculation of success rates, primarily for demographical comparison. It also contains information that categorises providers in terms of relative deprivation.
14. Where it comes from – The information is extracted from an LSC application called the Provider and Organisation Information Management System (PIMS). This links to the UK Register of Learning Providers (UKRLP), and is administered by staff in local and regional LSC teams. Information regarding mergers is collected separately from local and regional LSC teams.

Derived variables

15. In calculating success rates it is necessary to derive a number of variables. These are derived by applying standard algorithms to the ILR data returned by providers. Again, these are detailed in Annex B.

Output files

16. This section explains which data sets are outputs used in the production of success rate data. Two files are produced:

1. Success rates aims master file
2. Success rates demographics master file

17. Each of these files performs a different purpose. The success rates aims master file contains information about learning aims, but not demographic information. Conversely, the success rates demographics master file contains information about learner demographics, but is not at learning aim level. The reasoning behind this approach is that one combined file would be exceptionally large, and therefore storage and processing time would become significant issues.

18. Both files are used to calculate different elements of the qualification success rate reports.

Detailed definition

19. The format and content of the data collected on learners has changed from year to year. The pre-processing of the ILR collections to form the master files is partly to ensure that they are of comparable format to the latest year.

20. There is a certain amount of functionality within the code for the FE Success Rates Methodology that is concerned with handling exceptions, inconsistencies and peculiarities within the data. The details of this are not covered here but may be seen within the sample code.

Initial processing of source data

21. Learning aims information is matched in from the hierarchy table. The hierarchy file is an output from the Learning Aim Database (LAD) that includes a grouping variable map code and other data fields from the LAD for each learning aim.

22. In 2005/06 all mapcodes were revised to reflect level and sector skill areas and these continue to be updated each year to group new learning aims on the LAD.

23. The actual start year, expected end year and actual end year is calculated for each learning aim.

Merge five years worth of ILR files together.

24. The separate files are merged such that any entries for each year, where the entry appears in more than one file for the same learner at the same institution with the same aim, then the latest year's entry is the only one kept. This is to ensure that where a learning aim spans several years then it is counted only once.

Drop cases that are not required for reporting.

25. Entries are dropped where not required for analysis. There are a number of reasons for this, however the predominant ones are

- where the expected end year is not one of the last three years of the sequence i.e. 2007/08, 2006/07 or 2005/06
- where a learner withdraws from the aim before 1 November of the first year
- UFI qualifications and offender learners in custody qualifications
- Qualifications classified as unitised qualifications (such as Unitisation Level 2, Preparation for Life and Work) and additional units

Produce a series of categorical derived variables

26. A number of derived variables are calculated and stored in the output datasets. These variables have values that are categorised or banded.

- a) The learner's age at 31 August of the year in which they started the aim is used to categorise into various age groups. The 16-18 age band includes learners who are under 16 when they started the aim.
- b) The expected duration of the learning programme is used to categorise into various duration bands.
- c) The qualification type is used to categorize the aim in a number of different ways based on both level and duration.

Create a series of scaled derived variables

27. There are a number of measures that are required to calculate success rates.

- a) The number of learning aims started
- b) The number transferred to another learning aim within the same institution.
- c) The number that have achieved the learning aim
- d) The number that have completed the learning aim
- e) The number that are continuing the learning aim
- f) The number where there is a known outcome for the learning aim
- g) The number of high grade results
- h) The number of full level 2 qualifications in their own right
- i) The number of full level 3 qualifications in their own right

Remove non LSC funded learner cohorts

28. Success Rates are calculated for aims where there is at least one learning aim in a cohort that is funded by the LSC. A cohort is generally a group of learners with the same aim, the same duration category, expected to end in the same year and provided by the same institution. The reason for this inclusion of

some non-LSC funded learners is that OFSTED inspections cover all aims where there are any LSC funded learners.

29. Records are removed for learning aims where the whole cohort is not LSC funded. This means that some non-LSC funded learners are included in success rates.

Aggregate to create qualification master file.

30. The data is aggregated to produce a source file for further analysis by learning aim characteristics at institution level.
31. The institution and LSC office details are then matched in from the institution lookup file and saved as the Success rate Aim Master File.

Aggregate to create demographic master file.

32. The data is aggregated to produce a source file for further analysis by demographics institution level.
33. The institution and LSC office details are then matched in from the institution lookup file and saved as the Success Rate Demographics Master File.

Sample code

34. The SPSS code attached in Annex A is provided to explain how the success rates have been calculated. This is intended for technical users and can be applied to data held by providers to produce success rates that confirm with the LSC methodology. We would advise providers to allow a reasonable degree of tolerance with success rates where an attempt is made to replicate published figures.

Enquiries

35. If you would like any more information about the success rate methodology please contact our Service Desk on 0870 2670001 or email servicedesk@thedata-service.org.uk

Annex A: FE Success Rate Sample Code 2007/08

1 The following SPSS code is provided to illustrate how FE Success Rates are calculated

Extract relevant history files and add fields required for matching and calculations

```
ILR FE F05 2007/08
ILR FE F05 2006/07
ILR FE F05 2005/06
ILR FE F05 2004/05
ILR FE F05 2003/04
```

Remove spaces and zeroes in learner enrolment number

```
String triml03 (A12).
Compute Triml03=ltrim(L03).

COMPUTE no_zeros = 0.

LOOP.
COMPUTE no_zeros = no_zeros + 1.
END LOOP IF (SUBST(triml03,no_zeros,1)<>"0").

STRING new_l03 (A12).
COMPUTE new_l03 = SUBSTR(triml03,no_zeros, (13 - no_zeros)).
```

Compute a new variable for matching learners across years

```
COMPUTE Newref = CONCAT(STRING(I01,F6.0)," ",new_l03).
exe.
```

Calculate age at start of aim – CAUTION This is different in some cases to the derived variable in the data dictionary.

```
DO IF (NOT SYSMIS (L11)).
do IF ( XDATE.MONTH(A27) LE 7).
+ do IF ( XDATE.MONTH(L11) LE 8).
+ COMPUTE L_AGE=(XDATE.YEAR(A27)-1) - XDATE.YEAR(L11).
+ ELSE.
+ COMPUTE L_AGE=(XDATE.YEAR(A27)-2)-XDATE.YEAR(L11).
+ end IF.
Else if ( XDATE.MONTH(A27) GT 7).
+ do IF ( XDATE.MONTH(L11) LE 8).
+ COMPUTE L_AGE=XDATE.YEAR(A27) - XDATE.YEAR(L11).
+ ELSE.
```

```

+ COMPUTE L_AGE=(XDATE.YEAR(A27)-1)-XDATE.YEAR(L11).
+ end IF.
end IF.
end IF.

RECODE L_AGE (SYSMIS=9) (0 thru 15=1) (16 thru 18=2) (19 thru 20=3)
      (21 thru 24=4) (25 thru 59=5) (60 thru 120=6) (ELSE=9) into L_AGEB.

VARIABLE LABELS
      L_AGE 'DV - Age of the learner as at 31 August of start year'
      L_AGEB 'DV - Age of the learner as at 31 August of start year banded'.

VALUE LABELS L_AGEB
      1 'Under 16'
      2 '16-18'
      3 '19-20'
      4 '21-24'
      5 '25-59'
      6 '60 AND over'
      9 'missing age'.

COMPUTE agegroup = 0.
IF (L_AGEB = 1 or L_AGEB = 2) agegroup = 1.

VARIABLE LABELS agegroup 'age group of student'.
VALUE LABELS agegroup 1 '16-18' 0 '19 & over'.
exe.

```

Match in hierarchy table

```

SORT CASES by A09.
MATCH FILES /FILE=*
      /TABLE='C:\Hierarchy Table.sav'
      /BY A09.

IF mapcode = ("      ") mapcode = a09.

```

If mapcode is missing then recode mapcode to equal qual aim ref.

Save output for each year

```

SORT CASES by newref, mapcode, startyr, expendyr.

SAVE OUTFILE = 'C:\ILR0405_F05_FESB_Initial Extract.sav'.
SAVE OUTFILE = 'C:\ILR0304_F05_FESB_Initial Extract.sav'.
SAVE OUTFILE = 'C:\ILR0203_F05_FESB_Initial Extract.sav'.

```

```
SAVE OUTFILE = 'C:\ILR0102_F05_FESB_Initial Extract.sav'.
```

Extract relevant 2007/08 data from ILR FE F05 2007/08

Select relevant aims only

```
Select if ANY(A10,20,70,80,99).
```

```
RECODE A10 (20=1) (80=1) (70=1)(ELSE=0) into fundstat.  
RECODE A10(ELSE=2002) into F05.
```

Calculate newref to ensure matching works okay
Match in INST code on UPIN.
Save, dropping unwanted variables.

Creation of match variable

```
String triml03 (A12).  
Compute Triml03=ltrim(L03).  
  
COMPUTE no_zeros = 0.  
  
LOOP.  
COMPUTE no_zeros = no_zeros + 1.  
END LOOP IF (SUBST(triml03,no_zeros,1)<>"0").  
  
STRING new_l03 (A12).  
COMPUTE new_l03 = SUBSTR(triml03,no_zeros, (13 - no_zeros)).  
  
STRING newref (A18).  
COMPUTE newref = CONCAT(STRING(I01,F6.0)," ",new_l03).  
exe.
```

* Match in institution codes from UPINs.

```
Sort cases by L01.
```

```
Match files file=*  
/table='C:\INST_LOOKUP.sav'  
/by L01.
```

Calculation of Start year, expected end year and actual end year.

```
compute actendyr = 0.
```

Do if (A31 >= date.dmy(1, 8, 2002) and
A31 <= date.dmy(31, 7, 2003)).
compute actendyr=2002.

else if (A31 >= date.dmy(1, 8, 2001) and
A31 <= date.dmy(31, 7, 2002)).
compute actendyr=2001.

else if (A31 >= date.dmy(1, 8, 2000) and
A31 <= date.dmy(31, 7, 2001)).
compute actendyr=2000.

else if (A31 >= date.dmy(1, 8, 1999) and
A31 <= date.dmy(31, 7, 2000)).
compute actendyr=1999.

else if (A31 >= date.dmy(1, 8, 2003) and
A31 <= date.dmy(31, 7, 2004)).
compute actendyr=2003.

else if (A31 >= date.dmy(1, 8, 2004) and
A31 <= date.dmy(31, 7, 2005)).
compute actendyr=2004.

else if (A31 >= date.dmy(1, 8, 2005) and
A31 <= date.dmy(31, 7, 2006)).
compute actendyr=2005.

else if (A31 >= date.dmy(1, 8, 2006) and
A31 <= date.dmy(31, 7, 2007)).
compute actendyr=2006.

else if (A31 >= date.dmy(1, 8, 2007) and
A31 <= date.dmy(31, 7, 2008)).
compute actendyr=2007.

else if (A31 >= date.dmy(1, 8, 2008) and
A31 <= date.dmy(31, 7, 2009)).
compute actendyr=2008.

else if (A31 >= date.dmy(1, 8, 2009) and
A31 <= date.dmy(31, 7, 2010)).
compute actendyr=2009.

else if (A31 >= date.dmy(1, 8, 2010) and
A31 <= date.dmy(31, 7, 2011)).
compute actendyr=2010.

end if.

Execute.

compute expendyr = 0.

Do if (A28 >= date.dmy(1, 8, 2002) and
A28 <= date.dmy(31, 7, 2003)).
compute expendyr=2002.

else if (A28 >= date.dmy(1, 8, 2001) and
A28 <= date.dmy(31, 7, 2002)).
compute expendyr=2001.

else if (A28 >= date.dmy(1, 8, 2000) and
A28 <= date.dmy(31, 7, 2001)).
compute expendyr=2000.

else if (A28 >= date.dmy(1, 8, 1999) and
A28 <= date.dmy(31, 7, 2000)).
compute expendyr=1999.

else if (A28 >= date.dmy(1, 8, 2003) and
A28 <= date.dmy(31, 7, 2004)).
compute expendyr=2003.

else if (A28 >= date.dmy(1, 8, 2004) and
A28 <= date.dmy(31, 7, 2005)).
compute expendyr=2004.

else if (A28 >= date.dmy(1, 8, 2005) and
A28 <= date.dmy(31, 7, 2006)).
compute expendyr=2005.

else if (A28 >= date.dmy(1, 8, 2006) and
A28 <= date.dmy(31, 7, 2007)).
compute expendyr=2006.

else if (A28 >= date.dmy(1, 8, 2007) and
A28 <= date.dmy(31, 7, 2008)).
compute expendyr=2007.

else if (A28 >= date.dmy(1, 8, 2008) and
A28 <= date.dmy(31, 7, 2009)).
compute expendyr=2008.

else if (A28 >= date.dmy(1, 8, 2009) and
A28 <= date.dmy(31, 7, 2010)).

```
compute expendyr=2009.
```

```
else if (A28 >= date.dmy( 1, 8, 2010) and  
A28 <= date.dmy(31, 7, 2011)).  
compute expendyr=2010.
```

```
else if (A28 >= date.dmy( 1, 8, 2011) and  
A28 <= date.dmy(31, 7, 2012)).  
compute expendyr=2011.
```

```
end if.  
Execute.
```

```
compute startyr = 0.
```

```
Do if (A27 >= date.dmy( 1, 8, 2002 ) and  
A27 <= date.dmy(31, 7, 2003)).  
compute startyr=2002.
```

```
else if (A27 >= date.dmy( 1, 8, 2001) and  
A27 <= date.dmy(31, 7, 2002)).  
compute startyr=2001.
```

```
else if (A27 >= date.dmy( 1, 8, 2000) and  
A27 <= date.dmy(31, 7, 2001)).  
compute startyr=2000.
```

```
else if (A27 >= date.dmy( 1, 8, 1999) and  
A27 <= date.dmy(31, 7, 2000)).  
compute startyr=1999.
```

```
else if (A27 >= date.dmy( 1, 8, 2003 ) and  
A27 <= date.dmy(31, 7, 2004)).  
compute startyr=2003.
```

```
else if (A27 >= date.dmy( 1, 8, 2004 ) and  
A27 <= date.dmy(31, 7, 2005)).  
compute startyr=2004.
```

```
else if (A27 >= date.dmy( 1, 8, 2005 ) and  
A27 <= date.dmy(31, 7, 2006)).  
compute startyr=2005.
```

```
else if (A27 >= date.dmy( 1, 8, 2006) and  
A27 <= date.dmy(31, 7, 2007)).  
compute startyr=2006.
```

```
else if (A27 >= date.dmy( 1, 8, 2007) and
```

```

A27 <= date.dmy(31, 7, 2008)).
compute startyr=2007.

else if (A27 >= date.dmy( 1, 8, 2008) and
A27 <= date.dmy(31, 7, 2009)).
compute startyr=2008.

else if (A27 >= date.dmy( 1, 8, 2009) and
A27 <= date.dmy(31, 7, 2010)).
compute startyr=2009.

else if (A27 >= date.dmy( 1, 8, 2010) and
A27 <= date.dmy(31, 7, 2011)).
compute startyr=2010.

end if.
Execute.

```

Calculate age

```

DO IF (NOT SYSMIS (L11)).

do IF ( XDATE.MONTH(A27) LE 7).
+ do IF ( XDATE.MONTH(L11) LE 8).
+ COMPUTE L_AGE=(XDATE.YEAR(A27)-1) - XDATE.YEAR(L11).
+ ELSE.
+ COMPUTE L_AGE=(XDATE.YEAR(A27)-2)-XDATE.YEAR(L11).
+ end IF.
Else if ( XDATE.MONTH(A27) GT 7).
+ do IF ( XDATE.MONTH(L11) LE 8).
+ COMPUTE L_AGE=XDATE.YEAR(A27) - XDATE.YEAR(L11).
+ ELSE.
+ COMPUTE L_AGE=(XDATE.YEAR(A27)-1)-XDATE.YEAR(L11).
+ end IF.
end IF.
end IF.

RECODE L_AGE (SYSMIS=9) (0 thru 15=1) (16 thru 18=2) (19 thru 20=3)
(21 thru 24=4) (25 thru 59=5) (60 thru 120=6) (ELSE=9) into L_AGEB.

VARIABLE LABELS
L_AGE 'DV - Age of the learner as at 31 August of start year'
L_AGEB 'DV - Age of the learner as at 31 August of start year banded'.

VALUE LABELS L_AGEB
1 'Under 16'

```

```
2 '16-18'  
3 '19-20'  
4 '21-24'  
5 '25-59'  
6 '60 AND over'  
9 'missing age'.
```

```
COMPUTE agegroup = 0.  
IF (L_AGEB = 1 or L_AGEB = 2) agegroup = 1.
```

```
VARIABLE LABELS agegroup 'age group of student'.  
VALUE LABELS agegroup 1 '16-18' 0 '19 & over'.  
exe.
```

Add mapcodes to 2007/08 file

```
SORT CASES by A09.  
MATCH FILES /FILE=*  
/TABLE='C:\Hierarchy Table.sav'  
/BY A09.
```

```
IF map_code_code = (" ") map_code_code = a09.
```

** If mapcode is missing then recode mapcode to equal qual aim ref.

```
SORT CASES by newref, map_code_code, startyr, expendyr.
```

Save output for ILR FE F05 200708

Step C

MERGE ALL files TOGETHER

If a student has the same newref) and the same qualification aim code (mapcode) in two or more files, then the qualification in the 2007/08 file takes precedence over the 2006/07 file which in turn takes over the 2005/06 file etc.

```
Get file ='C:\ILR0708_F05_FESB_Initial Extract.sav'.
```

```
MATCH FILES FILE = *  
/FILE = 'C:\ILR0607_F05_FESB_Initial Extract.sav'  
/FILE = 'C:\ILR0506_F05_FESB_Initial Extract.sav'  
/FILE = 'C:\ILR0405_F05_FESB_Initial Extract.sav'  
/FILE = 'C:\ILR0304_F05_FESB_Initial Extract.sav'  
/BY newref map_code_code startyr expendyr.
```

Code to recognise BTEC Nation Dips with three letters in A36. This code identifies high grades and uses the two character code "HI".

```
STRING temp (A3).  
COMPUTE temp = a36.  
EXECUTE.  
DELETE VARIABLES a36.
```

```
STRING a36 (A2).  
DO IF (((SUBSTR(temp,3,1) = "D") OR (SUBSTR(temp,3,1) = "M") OR  
(SUBSTR(temp,3,1) = "P"))  
AND (SUBSTR(temp,1,1) <> "P") AND (SUBSTR(temp,2,1) <> "P") AND  
(SUBSTR(temp,3,1) <> "P")).  
COMPUTE A36 = "HI".  
ELSE.  
COMPUTE A36=TEMP.  
END IF.  
  
EXECUTE.  
  
DELETE VARIABLES temp.
```

Remove UFI qualifications

```
SELECT IF A46A<>1.  
SELECT IF A46B<>1.
```

compute actendyr = 0.

Drop cases that are not required for analysis.

Code to ignore qualifications withdrawn before 1 November in first year of qualification adapted for new cyglh code.

This code was changed in 2005/06 by removing the 2 lines below. It removes learners who leave a long course within 1 day of starting.

```
**SELECT IF NOT (A_EXP_A <=0 and (startyr=endyr)).  
**SELECT IF NOT (A_EXP_A <=0.01 AND (A31 < DATE.DMY(1,11,startyr)) AND  
A34= 3 AND ANY (A35,0,3)).
```

** The above code was changed in 2005/06 to resolve a query

```
SELECT IF NOT (A_EXP_A <=0.01 AND (A31 < DATE.DMY(1,11,startyr)) AND  
A34= 3 AND ANY (A35,0,3)).
```

```
SELECT IF NOT (A_EXP_A <=0.01 AND (STARTYR = ACTENDYR) AND (A34=  
3) AND ANY (A35,0,3) AND (CTIME.DAYS(A31) - CTIME.DAYS(A27) <= 1)).
```

**Drop all qualifications that are additional units, diagnostic tests, tutorial support and unitised qualifications.

```
SELECT IF mapqty<>"X901".
SELECT IF mapqty<>"8009".
SELECT IF mapqty<>"8008".
SELECT IF
NOT(ANY(MAPCODE,'B14X218','B14X0219','B14X0221','B14X0269','BUUX0220'
)).
SELECT IF NOT(ANY(a09,"77777777","CMISC001","XESF0001")).
```

Keep only cases where expected end year is 2007/08, 2006/07 or 2005/06.

```
SELECT IF ((expendyr = 2007) OR (expendyr = 2006) OR (expendyr = 2005)).
```

Add a marker for UFI qualifications.

```
COMPUTE Ufi=0.
IF (A46A=1) OR (A46B=1) OR (awarding_body_code="UFI") ufi =1.

VARIABLE LABELS Ufi 'Ufi quals'.
VALUE LABELS Ufi 0 'not Ufi qual' 1 'Ufi qual'.
exe.
```

Remove OLASS learners Olass = 1 means they are OLASS learners and should be excluded

```
COMPUTE OLASS=0.
IF A46a = 34 OLASS = 1.
IF A46b = 34 OLASS =1.

SELECT IF OLASS = 0.
```

Calculate the expected duration of the learning programme.
Determine the number of academic years the qualification spans, this is capped at 4 years.

```
DO IF (NOT ((MISSING (A28)) AND (MISSING (A31)))).
COMPUTE noayears = ((expendyr - startyr) + 1).
  ELSE IF ((MISSING (A28)) and (MISSING (A31))).
  COMPUTE noayears = 1.
END IF.
IF (noayears > 4) noayears = 4.
VARIABLE LABELS noayears 'Number of academic years qual spans'.
```

Calculate the expected duration (of the learning programme) as the number of days.

This uses expected end date where this is available and actual end date where this is not shown. Where both actual and expected end dates are missing, the qualification is assumed to have a short expected. Duration (99 days used to ensure qual is categorised as short (5 - 24 weeks).

Note: we do not want qualifications with zero cyglh to have an expected duration, therefore code these as missing (-9); these should be the only quals with an expected duration of -9.

```
COMPUTE expdur = -9.
DO IF (NOT (MISSING(A28))).
COMPUTE expdur = CTIME.DAYS(A28) - CTIME.DAYS(A27) + 1.
  ELSE IF (NOT (MISSING (A31))).
    COMPUTE expdur = CTIME.DAYS(A31) - CTIME.DAYS(A27) + 1.
      ELSE IF ((MISSING (A28)) and (MISSING (A31))).
        COMPUTE expdur = 99.
END IF.
VARIABLE LABELS expdur 'expected duration of the qualification'.
VALUE LABELS expdur -9 'missing'.
```

Define duration category - combination of 'noyears' and 'expdur', taking due account of the impact of leap years.

If there is a leap year during the expected duration of the course an extra day needs to be added to 'expdur'.

```
COMPUTE leap =0.
DO IF (A27 <= DATE.DMY(28,2,2000)) AND (A28 > DATE.DMY(28,2,2000)).
COMPUTE leap=1.
  ELSE IF (A27 <= DATE.DMY(28,2,2004)) AND (A28 > DATE.DMY(28,2,2004)).
    COMPUTE leap=1.
COMPUTE leap=1.
  ELSE IF (A27 <= DATE.DMY(28,2,2008)) AND (A28 > DATE.DMY(28,2,2008)).
    COMPUTE leap=1.

END IF.

COMPUTE dur = 0.

DO IF leap=0.
  DO IF (expdur = -9).
    COMPUTE dur = -9.
  ELSE IF (expdur >= 1 and expdur <= 167).
    COMPUTE dur = 1.
    ELSE IF (expdur >= 168 and expdur <= 365 and noyears = 1).
      COMPUTE dur = 2.
```

```

ELSE IF(expdur >= 168 and expdur <= 365 and noayears = 2).
COMPUTE dur = 3.
ELSE IF (expdur >= 366 and expdur <= 730 and noayears <= 2).
COMPUTE dur = 4.
ELSE IF (expdur >= 366 and expdur <= 730 and noayears = 3).
COMPUTE dur = 5.
ELSE IF (expdur >= 731 and expdur <= 1095 and noayears <= 3).
COMPUTE dur = 6.
ELSE IF (expdur >= 731 and expdur <= 1095 and noayears = 4).
COMPUTE dur = 7.
ELSE IF (expdur >= 1096).
COMPUTE dur = 8.
END IF.
END IF.

DO IF leap=1.
DO IF (expdur = -9).
COMPUTE dur = -9.
ELSE IF (expdur >=1 and expdur <= 167).
COMPUTE dur = 1.
ELSE IF (expdur >= 168 and expdur <= 366 and noayears = 1).
COMPUTE dur = 2.
ELSE IF (expdur >= 168 and expdur <= 366 and noayears = 2).
COMPUTE dur = 3.
ELSE IF (expdur >= 367 and expdur <= 731 and noayears <= 2).
COMPUTE dur = 4.
ELSE IF (expdur >= 367 and expdur <= 731 and noayears = 3).
COMPUTE dur = 5.
ELSE IF (expdur >= 732 and expdur <= 1096 and noayears <= 3).
COMPUTE dur = 6.
ELSE IF (expdur >= 732 and expdur <= 1096 and noayears = 4).
COMPUTE dur = 7.
ELSE IF (expdur >= 1097).
COMPUTE dur = 8.
END IF.
END IF.

VARIABLE LABELS dur 'category qualifications lie in'.

VALUE LABELS dur
-9 'missing'
1 'Short'
2 '1year_1ay'
3 '1year_2ay'
4 '2year_2ay'
5 '2year_3ay'
6 '3year_3ay'
7 '3year_4ay'

```

8 '4year or more 4ay or more'.

Code to split short quals into 'Short (< 5 wks)' and 'Short (5 - 24 wks)'.

```
COMPUTE shortdur =0.  
IF (expdur >=1) AND (expdur <35) shortdur=1.  
IF (expdur >=35) AND (expdur<=167) shortdur=2.  
  
VALUE LABELS shortdur  
  0 'Long' 1 'Short (<5wks)' 2 'Short (5-24 wks)'.  
exe.
```

Create groupings of learning aims.

```
RECODE LEARNING_AIM_TYPE_CODE ('0001','1432', '1433','1434','1435'=3)  
( '0002', '1430', '1431'=1) ('0003'=5) ('1414','1415','1416','1417','1418'=7)  
( '0017','0028'=10) ('0029','0030','0886'=10) ('0035'=7) ('0036'=15)  
( '0031','0032'=18)('1327'=30) ('1413'=31)(ELSE=19) INTO qtype2.  
  IF ((LEARNING_AIM_TYPE_CODE = '0004') OR  
(LEARNING_AIM_TYPE_CODE = '0005')) AND (NOTIONAL_LEVEL_V2_CODE  
= '1') qtype2 = 17.  
  
IF (SUBSTR(A09,1,5)='X9000') qtype2=20.  
IF (A09 = '55555555' or A09 = '66666666') qtype2 = 20.  
IF (A09 = '77777777' or A09 = '88888888') qtype2 = 20.  
IF (A09 = '99999000' or A09 = '99999111') qtype2 = 20.  
IF (A09 = '99999222' or A09 = '99999333') qtype2 = 20.  
IF (A09 = '99999999') qtype2 = 20.  
  
IF (SUBSTR(A09,1,4)='X9CE') qtype2=20.  
IF (SUBSTR(A09,1,4)='X9CN') qtype2=20.  
IF (SUBSTR(A09,1,5)='X9G01') qtype2=2.  
IF (SUBSTR(A09,1,5)='X9G02') qtype2=4.  
IF (SUBSTR(A09,1,5)='X9G03' OR SUBSTR(A09,1,4)='X9G3') qtype2=6.  
IF (SUBSTR(A09,1,4)='X9GQ') qtype2=8.  
IF (SUBSTR(A09,1,4)='X9NQ') qtype2=16.  
IF (SUBSTR(A09,1,4)='X9VQ') qtype2=9.  
  
VARIABLE LABELS qtype2 'reduced type of qual'.  
  
VALUE LABELS qtype2  
  1 'A'  
  2 'A generic'  
  3 'AS'  
  4 'AS generic'
```

5 'GCSE'
 6 'GCSE generic'
 7 'GNVQ and AVCE'
 8 'GNVQ generic'
 9 'GNVQ/NVQ units'
 10 'GNVQ precursors'
 15 'NVQ'
 16 'NVQ generic'
 17 'Access'
 18 'HND/HNC'
 19 'Other'
 20 ' Other generic'
 30 'Key Skills'
 31 'A2'.

Compute a field to identify the spreadsheet on which each qualification will appear in the BM qual level data.

```
COMPUTE ssheet=0.
DO IF ((qtype2 >=1 and qtype2 <=6) or qtype2=31).
COMPUTE ssheet=1.
  ELSE IF (qtype2 >= 7 and qtype2 <= 10).
  COMPUTE ssheet=2.
  ELSE IF (qtype2=15 or qtype2=16).
  COMPUTE ssheet=3.
  ELSE IF (qtype2 >= 17 and dur=1).
  COMPUTE ssheet=4.
  ELSE.
  COMPUTE ssheet=5.
END IF.
```

VARIABLE LABELS ssheet 'spreadsheet number'.

VALUE LABELS ssheet
 1 'A AS and GCSEs including A2'
 2 'GNVQ and GNVQ pre'
 3 'NVQs'
 4 'Other short'
 5 'Other not short'.

Recode qtype2 into qtype3 for the summaries. This is a slightly aggregated version of qtype2, and even broader qualification type so to speak. It also combines the spreadsheet it is on for other qualifications so we have one DV with the two pieces of information instead.

RECODE qtype2 (2=1) (4=3) (6=5) (8=7) (16=15) (ELSE=COPY) into qtype3.

IF (qtype2 = 17 and ssheet = 4) qtype3 = 21.
 IF (qtype2 = 18 and ssheet = 4) qtype3 = 22.
 IF (qtype2 = 19 and ssheet = 4) qtype3 = 23.
 IF (qtype2 = 20 and ssheet = 4) qtype3 = 24.
 IF (qtype2 = 17 and ssheet = 5) qtype3 = 25.
 IF (qtype2 = 18 and ssheet = 5) qtype3 = 26.
 IF (qtype2 = 19 and ssheet = 5) qtype3 = 27.
 IF (qtype2 = 20 and ssheet = 5) qtype3 = 28.
 IF (qtype2 = 30 and ssheet = 4) qtype3 = 29.
 IF (qtype2 = 30 and ssheet = 5) qtype3 = 30.

VARIABLE LABELS qtype3 'reduced version of qtype2 with ss for other'.

VALUE LABELS qtype3

1 'A level inc generic'
 3 'AS level inc generic'
 5 'GCSE inc generic'
 7 'GNVQ/AVCE inc generic'
 9 'GNVQ/NVQ units'
 10 'GNVQ precursors'
 15 'NVQs inc generic'
 21 'Access short'
 22 'HND/HNC short'
 23 'Other short'
 24 ' Other generic short'
 25 'Access long'
 26 'HND/HNC long'
 27 'Other long'
 28 ' Other generic long'
 29 'Key Skills short'
 30 'Key Skills long'
 31 'A2 quals'.

Create a high level group for use in SFR that combines length with notional level - area1.

```

RECODE map_nl
("E"="1")("4"="H")("5"="H")("6"="H")("7"="H")("M"="X")("U"="X")("
"="X")(else=copy).

```

```

RECODE map_ssa1 (" "="U")(ELSE =COPY).

```

```

RECODE ssa_tier2_code (" " = 'U')(ELSE =COPY).

```

```

COMPUTE area1 = 0.

```

```

IF (qtype2 = 30) area1 =7.

```

```

IF (dur =1 AND NOT (Notional_level_v2_code = "X") AND NOT (qtype2=30))
area1 = 6.

```

```

IF (dur > 1 AND Notional_level_v2_code = "1" AND NOT (qtype2=30)) area1 = 1.
IF (dur > 1 AND Notional_level_v2_code = "2" AND NOT (qtype2=30)) area1 = 2.
IF (dur > 1 AND (Notional_level_v2_code = "3") AND NOT (qtype2=30)) area1 = 3.
IF (dur > 1 AND Notional_level_v2_code = "H" AND NOT (qtype2=30)) area1 = 4.
IF (dur > 1 AND Notional_level_v2_code = "X" AND NOT (qtype2=30)) area1 = 5.

```

VARIABLE LABELS area1 'high level aggregation of learning aims'.

VALUE LABELS area1

```

0 'Short nl X'
1 'Long nl 1'
2 'Long nl 2'
3 'Long nl 3'
4 'Long nl H'
5 'Long nl X'
6 'Short nl 123H'
7 'Key Skills'.

```

Calculate a more detailed version of Area 1 variable which shows longs and shorts by broad qualification type for supporting data tables - area2.

```

COMPUTE area2 = 0.

```

```

IF (dur > 1 AND ssheet = 2 AND map_nl = "1") area2 = 1.
IF (dur > 1 AND ssheet = 3 AND map_nl = "1") area2 = 2.
IF (dur > 1 AND ssheet = 5 AND map_nl = "1" AND NOT (qtype2=30)) area2 = 3.
IF (dur > 1 AND ssheet = 1 AND map_nl = "2") area2 = 4.
IF (dur > 1 AND ssheet = 2 AND map_nl = "2") area2 = 5.
IF (dur > 1 AND ssheet = 3 AND map_nl = "2") area2 = 6.
IF (dur > 1 AND ssheet = 5 AND map_nl = "2" AND NOT (qtype2=30)) area2 = 7.
IF (dur > 1 AND ssheet = 1 AND map_nl = "3") area2 = 8.
IF (dur > 1 AND ssheet = 2 AND map_nl = "3") area2 = 9.
IF (dur > 1 AND ssheet = 3 AND map_nl = "3") area2 = 10.
IF (dur > 1 AND ssheet = 5 AND map_nl = "3" AND NOT (qtype2=30)) area2 = 11.
IF (dur > 1 AND ssheet = 2 AND map_nl = "H") area2 = 12.
IF (dur > 1 AND ssheet = 3 AND map_nl = "H") area2 = 13.
IF (dur > 1 AND ssheet = 5 AND map_nl = "H" AND NOT (qtype2=30)) area2 = 14.
IF (dur > 1 AND ssheet = 1 AND map_nl = "X") area2 = 15.
IF (dur > 1 AND ssheet = 2 AND map_nl = "X") area2 = 16.
IF (dur > 1 AND ssheet = 3 AND map_nl = "X") area2 = 17.
IF (dur > 1 AND ssheet = 5 AND map_nl = "X" AND NOT (qtype2=30)) area2 = 18.
IF (dur = 1 AND map_nl = "1" AND NOT (qtype2=30)) area2 = 19.
IF (dur = 1 AND map_nl = "2" AND NOT (qtype2=30)) area2 = 20.
IF (dur = 1 AND map_nl = "3" AND NOT (qtype2=30)) area2 = 21.
IF (dur = 1 AND map_nl = "H" AND NOT (qtype2=30)) area2 = 22.
IF (dur = 1 AND map_nl = "X" AND NOT (qtype2=30)) area2 = 23.
IF (qtype2 = 30) area2 = 24.

```

VARIABLE LABELS area2 'area of supporting data'.

VALUE LABELS area2

- 1 'Long GNVQ nl 1'
- 2 'Long nvq nl 1'
- 3 'Long other nl 1'
- 4 'Long GCSEs nl 2'
- 5 'Long GNVQ nl 2'
- 6 'Long NVQ nl 2'
- 7 'Long other nl 2'
- 8 'Long AAS (including A2) nl 3'
- 9 'Long GNVQ nl 3'
- 10 'Long NVQ nl 3'
- 11 'Long other nl 3'
- 12 'Long GNVQ nl H'
- 13 'Long NVQ nl H'
- 14 'Long other nl H'
- 15 'Long AASGCSE nl X'
- 16 'Long GNVQ nl X'
- 17 'Long NVQ nl X'
- 18 'Long other nl X'
- 19 'Short all quals nl 1'
- 20 'Short all quals nl 2'
- 21 'Short all quals nl 3'
- 22 'Short all quals nl H'
- 23 'Short all quals nl X'
- 24 'Key Skills'.

exe.

Create a series of derived variables - count, trans, achieve, complete, continue, known outcome, high grade.

COMPUTE count = 1.

COMPUTE trans = 0.

IF (A34 = 4) trans = 1.

VARIABLE LABELS trans 'student has transferred onto another qual'.

COMPUTE achieved = 0.

IF (A35 = 1) OR (A35=6) OR (A35=7) achieved = 1.

VARIABLE LABELS achieved 'qual is achieved'.

COMPUTE complete = 0.

IF (A34 = 2) complete = 1.

VARIABLE LABELS complete 'qual is completed'.

COMPUTE contin = 0.

IF (A34 = 1) contin = 1.

COMPUTE compv = 0.

```
IF (A34 = 2) AND (A35 =1 OR A35 = 2 OR A35 = 3 OR A35 >= 6) compv = 1.  
VARIABLE LABELS compv 'completed and valid outcome'.
```

```
COMPUTE exam = 0.  
IF (A35 = 4) exam = 1.  
VARIABLE LABELS exam 'exam taken but result not yet known'.
```

```
COMPUTE learn = 0.  
IF (A35 = 5) learn = 1.  
VARIABLE LABELS learn 'learning complete but exam not yet taken'.
```

```
STRING grade (A8).  
RECODE A36 ("="="NA") ('A', '*', '*A', 'A*', 'AA', 'AB', 'AC', 'AD',  
'AE', 'AF', 'AG', 'B', 'BB', 'BA', 'BC', 'BD', 'BE', 'BF', 'BG'='HIGH_AB')  
( 'C', 'CA', 'CB', 'CC', 'CD', 'CE', 'CF', 'CG', 'ME', 'DS', 'DI', 'HI' = 'HIGH')  
(ELSE=COPY) into grade.
```

```
COMPUTE gradek = 1.  
IF (grade = "NA") gradek = 0.  
VARIABLE LABELS gradek 'qual has a known grade'.
```

```
COMPUTE aa2as =0.  
IF(qtype3 = 1 OR qtype3 =3 OR qtype3 =31 OR (LEARNING_AIM_TYPE_CODE  
='1414') or (LEARNING_AIM_TYPE_CODE ='1415') OR  
(LEARNING_AIM_TYPE_CODE ='1416') OR (LEARNING_AIM_TYPE_CODE  
='1417') OR (LEARNING_AIM_TYPE_CODE ='1418')) aa2as=1.
```

```
COMPUTE high = 0.  
COMPUTE high = 0.  
IF (achieved = 1 AND grade = "HIGH_AB" AND aa2as=1 ) high = 1.  
IF (achieved =1 AND aa2as <> 1 AND grade ="HIGH") high=1.  
IF (achieved =1 AND aa2as <> 1 AND grade = "HIGH_AB") high=1.  
IF (achieved = 1 AND LEARNING_AIM_TYPE_CODE = "1424" AND (grade =  
"DD" OR grade = "DM" OR grade = "MD" OR grade = "MM")) high = 1.  
IF (achieved = 1 AND (LEARNING_AIM_TYPE_CODE = "1418" OR  
LEARNING_AIM_TYPE_CODE = "1415") AND A36 = "DD") high =1.  
VARIABLE LABELS high 'qual achieved and high grade'.
```

```
Compute Full2 =0.  
RECODE FullLevel2 ("1", "3" =1) ("2", "4", "U", "X"=0) INTO Full2.
```

```
RECODE full3 ("=" = "U") (ELSE = COPY).
```

Code to merge in the Skills for Life and Basic Skills markers.

Code to identify learner cohorts without aggregating the file and thus losing the learner and learning aim reference.

Need to identify members of the same cohort. Having pre-sorted the variables used to define a cohort, the variable CHANGE is used to identify where any learner is in a different cohort from the learner in the row above. This is achieved by setting the change variable to 0 for all cases and then the IF command sets the change variable to 1 where the learner is in the same cohort as the case in the row above. Therefore cases where change = 0 are the first case of the new cohort.

```
COMPUTE change =0.
IF (caseno = 1) change = 0.
IF (I01 = LAG(I01,1)
  AND ssheet = LAG(ssheet,1)
  AND qtype2 = LAG(qtype2,1)
  AND qtype3 = LAG(qtype3,1)
  AND map_code_code = LAG(map_code_code,1)
  AND awarding_body_code = LAG(awarding_body_code,1)
  AND map_code_desc = LAG(map_code_desc,1)
  AND notional_level_v2_code = LAG(notional_level_v2_code,1)
  AND dur=LAG(dur,1)
  AND expendyr = LAG(expendyr,1)
  AND agegroup = LAG(agegroup,1)
  AND ufi = LAG(ufi,1)
  AND area1 = LAG(area1,1)
  AND area2 = LAG(area2,1)
  AND ssa_tier1_code = LAG(ssa_tier1_code,1)
  AND ssa_tier2_code = LAG(ssa_tier2_code ,1)) change = 1
```

The COHORT variable sequentially numbers each cohort. The start of a new cohort is identified by cases where change = 0. Because this piece of code uses the LAG command, there are problems with the first row. By using 'caseno' it is possible to force the correct value in the first case - all other cases work from the LAG command without problem. Only cohorts with funded learners are included.

```
COMPUTE cohort = 0.
IF (caseno = 1) cohort = 1.
IF (caseno > 1 AND change = 0) cohort = LAG(cohort,1) + 1.
IF (caseno > 1 AND change = 1) cohort = LAG(cohort,1).

SAVE OUTFILE = 'C:\0506Cohort_temp.sav'
/COMPRESSED.

AGGREGATE OUTFILE = 'C:\0506fundstat_agg.sav'
/BREAK cohort
/sum_fund = SUM(fundstat).

GET FILE = 'C:\0506Cohort_temp.sav'.

MATCH FILES / FILE = *
```

```
/TABLE ='C:\506fundstat_agg.sav'  
/BY cohort.
```

```
SELECT IF (sum_fund <> 0).
```

Save output. This is the preaggregated file

[Qual Masterfile](#)

Aggregate to create qual master file.

```
SELECT IF (olass = 0).
```

```
SORT CASES BY I01 ssheet qtype2 qtype3 mapcode map_aol mapawb maptitle  
map_nl dur expendyr agegroup Ufi area1 area2 map_ssa1 full2 full3 shortdur  
olass cove.
```

```
AGGREGATE OUTFILE=*  
/PRESORTED
```

```
/BREAK= I01 ssheet qtype2 qtype3 mapcode map_aol mapawb maptitle map_nl  
dur expendyr agegroup Ufi area1 area2 map_ssa1 full2 full3 shortdur olass cove
```

```
/count = sum(count)
```

```
/nov = sum(nov)
```

```
/fundstat = sum(fundstat)
```

```
/trans=sum(trans)
```

```
/contin=sum(contin)
```

```
/complete = sum(complete)
```

```
/achieved = sum(achieved)
```

```
/gradek=sum(gradek)
```

```
/high = sum(high)
```

```
/compv=sum(compv)
```

```
/exam=sum(exam)
```

```
/learn=sum(learn)
```

```
/expgh=sum(A_EXP_A).
```

```
SORT CASES BY I01.
```

Save output file as qual master file.

Aggregate to create [demographic master file](#).

Certain aims are removed as not used in analysis. Ethnicity is re-coded to ISR format.

```
SELECT IF (area1 <> 7 and Ufi = 0).
```

```
RECODE L14 (MISSING = 9) (else = copy).
```

```
STRING ethncity (A20).
```

```
RECODE L12 (11,01 = "Bangladeshi") (12,06 = "Indian") (13, 07 = "Pakistani") (14,  
09 = "Other Asian") (15, 02 = "Black African")
```

```
(16, 03 = "Black Caribbean") (17, 04 = "Black Other") (18, 05 = "Chinese") (19,  
20, 21, 22 = "Mixed") (08, 23, 24, 25 = "White")  
(98 = "Any other") (99 = "Unknown") (else = "Unknown") INTO ethncity.
```

```
SORT CASES by L01, area1, shortdur, expendyr, agegroup, L13, L14, ethncity,  
cove,map_ssa1,bs_partpn.
```

```
AGGREGATE OUTFILE=*
```

```
  /PRESORTED
```

```
  /BREAK=L01, area1, shortdur, expendyr, agegroup, L13, L14,  
ethncity,cove,map_ssa1,bs_partpn
```

```
  /count = sum(count)
```

```
  /trans=sum(trans)
```

```
  /contin=sum(contin)
```

```
  /complete = sum(complete)
```

```
  /achieved = sum(achieved)
```

```
  /compv=sum(compv).
```

Save output file as the demographics file.

Annex B Glossary of variables used in FE success rates output files

Field Name	Description
L01	Provider number
Ssheet	Year of Qualification
Qtype2	Qualification type
Qtype3	Aggregated qualification type
Inspcode	Aggregate inspection code – No longer used for new aims
Mapcode	Aggregated qualification code
Map_aol	Area of learning – No longer used for new aims
Mapawb	Awarding body
Maptitle	Aggregated qualification title
Map_nl	Notional level of the qualification – based on LAD
map_ssa1	Sector subject area code – tier 1
Expendyr	College year in which the qualification is expected to end
Dur	Duration category of the qualification
Agegroup	Age group of the student (16-18 or 19+)
Ufi	Whether the qualification is flagged as University for Industry
Area1	Primary level data variable
Area2	Supporting level of data variable
Shorddur	Duration of learning aim – sub divides short aims
Inst	Abbreviated institution name
Name	Institution name
Type	Grouped institution type
Type2	Institution type
LLSCCode	Local LSC code number - based on provider location
LLSCReg	LSC region – based on provider location
LLSCNme	Local LSC name – based on provider location
Col_Type	Grouped institution type code number
TypeName	Details of the college in full
Full2	Learning aims that are full level 2 (only present from 2004/05 onwards) – From LAD
Full3	Learning aims that are full level 3 (only present from 2004/05 onwards)

	onwards) – From LAD
BS_PARTPN	Learning aims that meet criteria for Skills for Life (only present from 2004/05 onwards)
Count	Total starters
Nov	Number of starts on or before 1 November of the start year
Fundstat	Number of starts which are council funded
Trans	Number of transfers onto another qualification
Contin	Number of enrolments continuing
Complete	Number of enrolments completed
Achieved	Number of enrolments achieved
gradek	Number of enrolments where the grade is known
High	Number of enrolments that have a known grade that is 'high'
Compv	Number of enrolments completed with a known outcome
Exam	Number of enrolments with outcome 'exam has been taken but result is not known'
Learn	Number of enrolments with outcome 'learning activities complete exam has not yet been taken'
DidMatch	Learner on this learning aim was located in current year's data and the previous year's data
ExpMatch	Learner on this learning aim should be found in following year's ILR data
COVE	Number of aims studied in COVEs
OLASS	Whether an aim is from an OLASS pilot
ExpGLH	Number of expected guided learning hours

LAD Hierarchy file

1 This will contain the 'Map Code' and other values for each learning aim.

Purpose

2 To assist providers in producing analyses of success rates and benchmarking, a downloadable file of 'Hierarchy Data' will be made available via the LAD.

Detailed definition

3 The file contains an entry for each learning aim with the fields taken from the learning aim table and all annual values table using a join on the learning aim reference column. The table is available in a variety of formats from the LSC providers' website <http://providers.lsc.gov.uk/lad/> where the field names are the same as the LAD field names.

Table 1 Fields in the Hierarchy Table

LAD Field name	Description	Hierarchy table field name
ACADEMIC_YEAR_CODE	The teaching year to which the annual values relate.	
LEARNING_AIM_REF (A09)	The reference number by which the LSC refers to this learning aim.	A09
MAP_CODE_DESC	Description of mapcode	maptitle
MAP_CODE_CODE	A code used to group learning aims together in a hierarchy for the purpose of analysis / benchmarking	mapcode
INSP_CODE_CODE	A code used to group learning aims together in a hierarchy for Inspection purposes – No longer used but included for old aims only	inspcode
AWARDING_BODY_CODE	A code to identify the awarding body associated with each learning aim record.	mapawb
LEARNING_AIM_TYPE_CODE	A code used by the Learner Information Suite to assign funding units to certain listed learning aims (e.g. NVQs and	mapqty

	GNVQs).	
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 Levels revised 2004 (v2) to expand levels 4 and 5 to levels 4 – 8.	map_nl
NOTIONAL_NVQ_LEVEL_CODE	A level on the NVQ scale for all learning aims. Enables learning aims to be analysed against the NVQ scale. Levels 4 and 5 continue to reflect their original (i.e. pre 0405) values. Not used on 2005/06	
AREA_OF_LEARNING_CODE	A code to classify the learning aim using the Adult Learning Inspectorate (ALI) areas of learning. AOL has been replaced by SSA and will not be used in future	Map_aol
NOTIONAL_NVQ_WIDTH	Gives the proportion of a full NVQ that the amount of material covered in a learning aim represents. For example, a GCSE has a notional NVQ width of 20%, since five GCSEs are equivalent to a single NVQ level 2.	
BASIC_SKILL	Indicates that the Learning Aim is considered a basic skill relating to achievements in literacy, numeracy and language and counts towards basic skills national targets. For 05/06 this field will be populated with the same data as the skills_for_life field	
BASIC_SKILL_TYPE_CODE	A sub-category identifying the type of basic skill (e.g. Adult literacy, ESOL). For 05/06 this field will be populated with the same data as the skills_for_life_type_code field	

LEVEL2_ENTITLEMENT_CAT_CODE	A code to indicate the category or 'family' to which the learning aim belongs for the purpose of assessing how far the aim contributes to a Full level 2	Full 2
LEVEL2_PERCENTAGE	The percentage of a Full Level 2 which a learning aim contributes. 100% indicates a Level 2 qualification in its own right	
LEVEL3_ENTITLEMENT_CAT_CODE	A code to indicate the category or 'family' to which the learning aim belongs for the purpose of assessing how far the aim contributes to a Full level 3	Full3
LEVEL3_PERCENTAGE	The percentage of a Full Level 3 which a learning aim contributes. 100% indicates a Level 3 qualification in its own right.	
KEY_SKILL_CODE	Identifies whether a learning aim is considered to be a 'key skill' for funding calculation purposes in a particular teaching year.	
SSA_TIER1_CODE	The broad (Tier 1) classification of the subject of a learning within the QCA Sector Subject Area classification system	Map_ssa1
SSA_TIER2_CODE	The more specialised classification (Tier 2) of the subject of a learning aim within the QCA Sector Subject Area classification system. Represented as Tier1 code <.> Tier 2 code (e.g. 05.1)	
SKILLS_FOR_LIFE	Indicates that 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). For 05/06 this field will be populated with the same data as the skills_for_life_type_code field