AM Notice: AM 2015/05
Date of Issue: Thursday 26th March 2015

Analysis Method Notice

Local Health Board Provider Spells

This notice describes an Analysis Method that has been developed for use in the production of published national outcome indicators, performance measures and/or currencies, which are derived directly from NHS Wales data.

The Analysis Method has been reviewed by the Analysis Methodologies Group and its output submitted to the Welsh Information Standards Board (WISB) for potential accreditation.

*It should be noted that, where the data flow on which the analysis is being undertaken has not been reviewed by WISB (see 'Status of WISB Data Standards Assurance' below), accreditation of the analysis method **cannot** be interpreted as an approval of the underlying data standards or the quality of the data used.*

It is recognised that formal review and/or assurance of the data flow may have been undertaken by other bodies, where those data are being formally published; for example, as Official Statistics'. In such circumstances, users of this method are advised to contact the relevant organisations should they require further information on the underlying quality of the specified data source.

For further details about the group, including Terms of Reference and membership, please visit the following website:

http://howis.wales.nhs.uk/sites3/page.cfm?orgid=742&pid=56696

WISB Reference: ISRN Ref. 2013/028

Please address enquiries about this Analysis Method the NHS Wales Informatics Service Data Standards Team.

E-mail: data.standards@wales.nhs.uk / Tel: 029 2050 2539

WISB Analysis Method Appraisal Assessment	**Accredited with Caveats** WISB suggests that modifications are made to the Analysis Method as outlined in the appraisal Outcome.
WISB Analysis Method Appraisal Outcome(s)	**Outcome**

Status of Data Standards Assurance	**Not WISB Reviewed** Some or all of the data used in this Analysis Method do not have standards approved via the Information Standards Assurance Process. This may include data flows that predate the establishment of WISB.
WISB Decision	n/a
Data Standards Assurance Outcome(s)	n/a

Indicator
n/a

Target:
n/a

Rationale / Context
When reporting Admitted Patient Care data set (APC ds) activity (i.e. inpatient, day case etc.), NHS Wales organisations are required to submit activity on the basis of a 'hospital spells' – i.e. continuous periods of admitted patient care within the same hospital site. This approach differs to that of England, whereby similar activity is reported on the basis of a 'provider spell' – i.e. a continuous period of inpatient care under a single provider, or organisation (i.e. LHB or Trust).

In recent years, there have been regular occasions when users of Welsh healthcare data (e.g. Welsh Government) have requested a 'provider spell' analysis of NHS Wales APC ds activity data. However, there has never been a clearly understood and documented methodology for how 'provider spell' analyses are to be undertaken.

Terminology

Hospital spells – these are continuous periods of inpatient care within the same hospital. With regards to the Patient Episode Database for Wales (PEDW), it is assumed that episodes submitted by the same Local Health Board with the same spell number are part of the same hospital spell.

Provider spells – these are continuous periods of inpatient care under a single provider (i.e. LHB or Trust).

Person spells – these are continuous periods of inpatient care for a single patient which could take place under any number of different providers.

Super spells – this is a non-specific term which has been used interchangeable over recent years to describe either a provider or person spell. It is important to stress that there is no **one** "super spell" algorithm, but the term is used generically to cover the wide range of algorithms described above, which link together individual hospital spells. Given its lack of specificity, it is recommended that the term 'super spell' should never be used to describe one of the more specific types of spell described above.

The algorithms above all attempt to track a patient through a continuous period of inpatient care, where hospital spells for the same individual are in close proximity to one another. Most 'spell' methodology variants also include additional linking criteria around hospital transfers, where the administrative codes supplied have to be consistent with a hospital transfer having taken place. For example, this is not the case for the PRISM implementation of a provider spells methodology, where the purpose is less about the calculation of Key Performance Indicators (KPIs) and more about risk estimation, and the enforcement of these additional criteria is less relevant.

NHS England issued technical guidance in 2005 around the derivation of so-called "Continuous Inpatient (CIP) Spells" for HES[1], and on these foundations the methodologies for hospital mortality and readmission indicators were developed. The majority of this was taken on board jointly by (then) Health Solutions Wales and NHS Performance Management within Welsh Government (WG) in their initial development of a "provider spell" algorithm for the Annual Quality Framework length of stay indicators.

The information contained hereafter only seeks to describe the methodology for the generation of 'provider spell' analyses (as described above) for use in relevant 2013/14 Welsh Government Tier 1 Measures.

[1] http://ratings2005.healthcarecommission.org.uk/Downloads/MoreInformationPageDocs/Construction%20of%20CIP%20Spells%202005.pdf

'Provider spell' algorithms are currently used in a number of settings such as the 2013/14 Welsh Government 'Tier 1' measures (e.g. Chronic Conditions Admissions / Multiple Admissions), the Predictive Risk Stratification Model (PRISM) predictive tool and in the calculation of hospital mortality and readmission rates in England and elsewhere.

Data Source

Admitted Patient Care Dataset

Definitions:

Definitional Guidance:

The algorithm takes into consideration the "mapped forward" Provider Codes (or 'Current Provider Codes' to use PEDW parlance) in order to take account of changes to organisational codes. For example, where one or more of the component hospital spells in a Cardiff & Vale provider spell took place during a period of organisational change (e.g. during 2009), the provider code in the first hospital spell might be RWM, whereas the subsequent hospital spell might have the provider code as 7A4.

However, where significant organisational changes (such as in 2009) resulted in the merger of previous organisations, such as was the case with Betsi Cadwaladr University, Hywel Dda, Abertawe Bro Morgannwg University and Cwm Taf LHBs, the above approach may link a pair of hospitals to be part of the same provider spell, even though *at the time of the original hospital transfer*, the pair of hospital spells took place within different providers.

The following data items and terms are relevant

Data Items:

Admission Method
Discharge Destination
Hospital Provider Spell Number
Organisation Code (Code of Provider)
Patient Classification
Source of Admission

Terms:

Hospital Provider Spell

Detailed Specification

Hospital spells are considered to be linked if all the following conditions are met:

1. The Organisation Code (Code of Provider) is the same for each spell;
   AND
2. They relate to the same person, i.e. same NHS number or same Case Record Number, if records received from the same instance of the Patient Administrative System (PAS);
   AND
3. The admission date of the subsequent hospital spell takes place within 1 day of the discharge date of the earlier spell – i.e. Admission Date (subsequent spell) minus Discharge Date (earlier spell) is between 0 and 1 day;
   AND
4. There is some suggestion of an **intra**-LHB hospital transfer between the two spells – i.e.
   (Discharge destination of earlier spell IN {55, 56, 57}
   OR
   Source of Admission of subsequent spell IN {55, 56, 57}
   OR
   Admission Method of subsequent spell = 81)

Practical Considerations

In order to establish which hospital spells are the "earlier" and "subsequent" spells, a rank order of all spells for each person must first be established.

For this algorithm, the rank order for each person is derived according to the following ordering system:

1. Admission Date (Ascending);
2. Discharge Date (Ascending);
3. Unique Hospital Spell Identifier (i.e. an ascending alphanumeric ordering of 'Organisation Code (Code of Provider)' and 'Hospital Provider Spell Number', concatenated together).

While the inclusion of these variables may seem slightly arbitrary, it is important to note that most are only included to **force** a unique ordering of hospital spells in those rare instances where both hospital spells have a zero length of stay, starting and ending on the same day.

Calculation:

Numerator
n/a

Denominator
n/a

Reporting Format / Frequency

n/a

Areas for Future Development

The following points reflect considerations raised by either the Analysis Methodologies Group or WISB in terms of aspects of the Analysis Method that require further investigation or development.

For a full breakdown of the issues considered, please refer to the formal WISB Outcome for this Analysis Method, which can be access via the Information Standards Assurance website:

http://howis.wales.nhs.uk/sites3/page.cfm?orgid=742&pid=52532

This section details any areas the Analysis Methodologies Group felt needed further consideration / review by the 'sponsor' of the method.

Inclusion of patients whose subsequent admission date is 1 or 2 days after the discharge date of the previous admission.

- Following a review of the data extraction criteria by the Analysis Methodologies group, which included a small review of individual patient data by some of the LHB members of the group, the methodology was amended to exclude those patients whose subsequent admission date is 2 days after the discharge date of the previous admission. Members recognised the justification for including the 1 day patients – recognisable data quality issues could be responsible for the recording of incorrect admission and discharge dates for genuine hospital transfers. However, members felt less confident that the 2 day patients were genuine transfers

- The group recognised that a further review of this issue was required. This should take the form of a structured review of patient case notes against the data contained within PEDW. Such a review should be used to inform a data quality review in relevant LHBs.

Restriction of methodology to inpatients (i.e. 'Patient Classification' = 1) only

- The original methodology reviewed by the group was restricted to the creation of provider spells for inpatients only. These were identified via the Patient Classification data item within the APC ds. Only those episodes where the Patient Classification = 1 were included in the methodology. The Analysis Methodologies Group recognised that a provider spell could quite easily be initiated (or continued) via other types of admission. For example:

    - A day case admission where complications arise and the patient is required to be admitted to another hospital site for further treatment.

    - An inpatient in a community hospital being transferred to an acute site (e.g. for a day case or regular day admission) and then being transferred back to the community hospital site.

- In light of the above observations, the restriction of the methodology to inpatients only was removed.

Ranking of provider spells

- The original methodology included the following five steps in the applying a ranking order to the individual hospital spells that make up one provider spell. These were originally as follows:

1. Admission Date (Ascending);
2. Discharge Date (Ascending);
3. Admission Method (Ascending value of 2 digit code;
4. Source of Admission (Ascending value of 2 digit code)
5. Unique Hospital Spell Identifier (e.g. ascending alphanumeric ordering of Provider Code and Spell Number, concatenated together).

- Whilst it was recognised that the purpose of the ranking is to **force** a unique ordering of hospital spells in those rare instances where both hospital spells have a zero length of stay, starting and ending on the same day, the rationale for the inclusion of 'Admission Method' and 'Source of Admission' in this ranking methodology was queried by the Welsh Information Standards Board (WISB) and the Analysis Methodology Group. They were originally included on the basis that, in ascending order, the values of the two data items had a logical 'flow' in terms of the routes patients usually move through secondary care (i.e. elective → emergency / home → nursing or residential homes). On analysing the data, it was shown that a small number of records were being impacted by this part of the ranking methodology. Therefore, given the concerns as to the inclusion of these two data items, this element of the ranking was removed.

Reporting of provider spells within the APC ds

- The Analysis Methodology Group recommended a longer term review into the business requirement for the reporting of NHS Wales APC ds activity data by hospital spells vs. provider spells.

Appendix A – Additional Information

n/a

Appendix B – SQL Code (where applicable)

Important: The following code is intended for information purposes only. It will contain references to specific references (servers, data item descriptions etc.) that are applicable within the NHS Wales Informatics Service only and therefore will not be suitable for direct application to local (LHB) data.

```
ALTER procedure [IPU].[Provider_spells]

as

/*

INDEX OF TABLES:

##provider_spells_step_1 - Base table of all ordinary inpatient spells
##provider_spells_step_2 - Table with 'uncleansed' discharge destinations from Monty
2
##provider_spells_step_3 - Spell details for all spells with a relevant transfer
code, along with all other spells for the same patient within the 2 day time frame,
along with all overlapping and duplicate spells
##provider_spells_step_4 - The result table of the cursor used to link spells in to
provider spells
##provider_spells_step_5 - The final look up table, with all other stand-alone spells
added
*/


-- STEP ONE: Create a base subset of apc_spell_subset (##provider_spells_step_1) with
the records we are interested in
-- This will enable us to update a handful of spells which have a null discharge
date, but shouldn't have
-- This occurs where a multi episode spells spans a trust merger, the spell is in
apc_spell twice, once
-- with a null discharge date and once with a valid discharge date

-- Create the table ##provider_spells_step_1

Select distinct
s.curr_prov_unit_code,
s.prov_unit_code,
s.spell_no,
s.admission_date,
s.discharge_date,
s.admission_method,
s.admission_source,
s.discharge_destination,
s.intended_management,
s.crn,
coalesce(p.nhs_no,case when s.curr_prov_unit_code<>'eng' then
s.curr_prov_unit_code+s.crn else s.prov_unit_code+s.crn end) nhs_no -- Use NHS number
if a valid one can be found, else prov_unit_code+crn

into ##provider_spells_step_1

from warehousedb.dbo.apc_spell_subset s

join warehousedb.dbo.patient_nhs p on
s.prov_unit_code=p.prov_unit_code
and s.spell_no=p.spell_no

where 1=2 and
submitted_patient_class in ('1','2')  -- Ordinary admissions and day cases only
and s.admission_date >= '01-April-2005'
```

```
-- Populate the table ##provider_spells_step_1

Insert into ##provider_spells_step_1

Select distinct
s.curr_prov_unit_code,
s.prov_unit_code,
s.spell_no,
s.admission_date,
s.discharge_date,
s.admission_method,
s.admission_source,
s.discharge_destination,
s.intended_management,
s.crn,
coalesce(p.nhs_no,case when s.curr_prov_unit_code<>'eng' then
s.curr_prov_unit_code+s.crn else s.prov_unit_code+s.crn end) nhs_no -- Use NHS number
if a valid one can be found, else prov_unit_code+crn

from warehousedb.dbo.apc_spell_subset s

join warehousedb.dbo.patient_nhs p on
s.prov_unit_code=p.prov_unit_code
and s.spell_no=p.spell_no

where
submitted_patient_class in ('1','2') -- Ordinary admissions and day cases only
and s.admission_date >= '01-April-2005'

-- (About 3 mins)

-- Index the table

create clustered index base_ind on ##provider_spells_step_1 (prov_unit_code,spell_no)

-- There are some spells which have a null discharge date, but which should not.
-- These are multi episode spells which span a trust merger, the spell is in
apc_spell twice, once
-- with a null discharge date and once with a valid discharge date

-- We want to set the discharge date of the premerged spell to be the discharge date
of the post merged spell
-- Match on spell number and admission date

Update a

set discharge_date = b.discharge_date

from ##provider_spells_step_1 a

join ##provider_spells_step_1 b on
a.curr_prov_unit_code = b.curr_prov_unit_code and -- Same merged provider
a.spell_no = b.spell_no and                                    -- Same spell
a.admission_date = b.admission_date                -- Same admission date

where a.discharge_date is null -- Pre-merged spell has a null discharge date
and a.prov_unit_code not like '7%' -- Pre-merged spell
and b.discharge_date is not null -- Post merged spell has a valid discharge date

-- BASE TABLE, ##provider_spells_step_1, CREATED

-- STEP TWO:

-- Create a table, ##provider_spells_step_2, of relevant discharge destinations from
```

```
Monty 2 (to remove effect of incorrectly cleansed discharge destinations)

Select distinct s.prov_unit_code,s.spell_no,s.discharge_destination

into ##provider_spells_step_2

from [IPU\NHSNational].monty2.dbo.dq_spell_subset s

where s.admission_date >= '01-April-2005'
and
(admission_source between '55' and '57'
or admission_method in ('29','81')
or discharge_destination between '55' and '57') -- There is a code indicating a
transfer
and submitted_patient_class in ('1','2') -- Ordinary admissions and day cases only

-- Index the table:

create clustered index ind1 on ##provider_spells_step_2 (prov_unit_code,spell_no)


-- STEP THREE:    Select spell details for those patients appearing in the uncleansed
discharge destination tables in to ##provider_spells_step_3

select distinct
s.curr_prov_unit_code,
s.prov_unit_code,
s.spell_no,
s.admission_date,
s.discharge_date,
s.admission_method,
s.admission_source,
d.discharge_destination,
s.intended_management,
s.crn,
s.nhs_no

into ##provider_spells_step_3

from   ##provider_spells_step_1 s

join ##provider_spells_step_2 d on
s.prov_unit_code=d.prov_unit_code
and s.spell_no=d.spell_no

where s.admission_date >= '01-Apr-2005'
and
(s.admission_source between '55' and '57'
or s.admission_method in ('29','81')
or d.discharge_destination between '55' and '57') -- There is a code indicating a
transfer

-- Index the table:

create clustered index ind1 on ##provider_spells_step_3(prov_unit_code,spell_no)
create index ind2 on ##provider_spells_step_3(nhs_no)


-- ADD ALL OTHER SPELLS FOR THE PERSON IN SECOND STAGE TABLE, APART FROM SPELLS THAT
ALREADY EXIST IN THE SECOND STAGE TABLE

insert into ##provider_spells_step_3
(curr_prov_unit_code,
prov_unit_code,
spell_no,
```

```
admission_date,
discharge_date,
admission_method,
admission_source,
discharge_destination,
intended_management,
crn,
nhs_no)


select distinct
p.curr_prov_unit_code,
p.prov_unit_code,
p.spell_no,
p.admission_date,p.discharge_date,
p.admission_method,p.admission_source,
p.discharge_destination,
p.intended_management,
p.crn,
p.nhs_no

from ##provider_spells_step_3 g

join ##provider_spells_step_1 p on
g.nhs_no=p.nhs_no
and g.prov_unit_code+g.spell_no<>p.prov_unit_code+p.spell_no

left outer join ##provider_spells_step_3 g2
on g2.prov_unit_code=p.prov_unit_code
and g2.spell_no=p.spell_no

where g2.spell_no is null
and (p.admission_date between g.discharge_date and (coalesce(g.discharge_date,'31-
December-2030')+1)
or p.discharge_date between g.admission_date-1 and g.admission_date
or p.admission_date between g.admission_date and coalesce(g.discharge_date,'31-
December-2030')
or p.discharge_date between g.admission_date and coalesce(g.discharge_date,'31-
December-2030')
)


-- ADD ALL INPATIENT STANDALONE SPELLS (i.e. not already in ##script2_step2) WHICH
OVERLAP IN TO
-- ##provider_spells_step_3.  THESE ARE SPELLS WHICH DO NOT HAVE THE RELEVANT
ADMISSION SOURCE/METHOD/DISCHARGE DEST SO WILL NOT BE IN THIS TABLE YET

Insert into ##provider_spells_step_3

Select distinct
s.curr_prov_unit_code,
s.prov_unit_code,
s.spell_no,
s.admission_date,s.discharge_date,
s.admission_method,s.admission_source,
s.discharge_destination,
s.intended_management,
s.crn,
s.nhs_no

from ##provider_spells_step_1 s

join ##provider_spells_step_1 p on
p.nhs_no=s.nhs_no and
s.prov_unit_code+s.spell_no <> p.prov_unit_code+p.spell_no -- Same person, different
```

```
spell number

left outer join ##provider_spells_step_3 z on -- Not already in the step 2 table
s.prov_unit_code = z.prov_unit_code and
s.spell_no = z.spell_no

where z.prov_unit_code is null
and

(

            (s.admission_date = p.admission_date
            and s.discharge_date = p.discharge_date
            and datediff(day, s.admission_date, s.discharge_date) > 0) -- 1
or

(((s.admission_date = p.admission_date and s.discharge_date < p.discharge_date)
or (s.admission_date = p.admission_date and s.discharge_date > p.discharge_date))
and datediff(day, s.admission_date, s.discharge_date) <> 0
and datediff(day, p.admission_date, p.discharge_date) <> 0) -- 2

or

(((s.discharge_date = p.discharge_date and s.admission_date < p.admission_date)
or (s.discharge_date = p.discharge_date and s.admission_date > p.admission_date))
and datediff(day, s.admission_date, s.discharge_date) <> 0
and datediff(day, p.admission_date, p.discharge_date) <> 0) -- 3

or

((s.admission_date > p.admission_date and s.discharge_date < p.discharge_date)
or (s.admission_date < p.admission_date and s.discharge_date > p.discharge_date)) --
4

or

((s.admission_date < p.admission_date and s.discharge_date < p.discharge_date and
s.discharge_date > p.admission_date)
or (s.admission_date > p.admission_date and s.discharge_date > p.discharge_date and
s.admission_date < p.discharge_date)) -- 5

)

-- (About 2 mins)

-- 1: Exact duplicate spells of > 0 day duration
-- 2: Admission dates the same, both spells over 0 day duration
-- 3: Discharge dates the same, both spells over 0 day duration
-- 4: One spell completely contained in the other - i.e. has an admission date after,
and a discharge date before, the other
-- 5: Overlapping spell



-- DELETE SPELLS WITH A NULL DISCHARGE DATE (ONLY INTERESTED IN FINISHED SPELLS),
NULL PATIENT ID's, AND ANY PATIENT ID's IN TABLE ONLY ONCE (no point putting these
through the cursor!)

delete ##provider_spells_step_3
where discharge_date is null

delete ##provider_spells_step_3
where nhs_no is null

delete ##provider_spells_step_3
```

```
 from ##provider_spells_step_3
where nhs_no in (select nhs_no from ##provider_spells_step_3
                                group by nhs_no having count(*)=1)

-- PRE-CURSOR TABLE, ##provider_spells_step_3, CREATED


-- STEP FOUR: THE CURSOR!

-- Create a table, ##provider_spells_step_4, for the results of the cursor:

Create table ##provider_spells_step_4 (
prov_unit_code char(3) null,
spell_no char (12) null,
prov_spell char (15) null,
prov_spell_order int null,
person_spell char (15) null,
person_spell_order int null,
max_prov_spell int null,
max_person_spell int null)


-- Declare variables

declare @prov_unit_code char(3),@spell_no char(12),@curr_prov_unit_code char(3),@crn
varchar(14),@nhs_no char(10),
@admission_method char(2),@admission_source char(2),@discharge_destination char(2),
@admission_date datetime,@discharge_date datetime

-- Declare variables for previous curr_prov, crn, nhs_no, disch_dest etc

declare @curr_prov_unit_code_prev char(3),@crn_prev varchar(14),@nhs_no_prev
char(10),
@discharge_destination_prev char(2),
@admission_date_prev datetime,@discharge_date_prev datetime

-- Declare variables for prov and person spell ID's and orders

declare @prov_spell char(15),@prov_spell_prev char(15), @prov_spell_order int
declare @person_spell char(15),@person_spell_prev char(15), @person_spell_order int

-- Declare cursor for curr_prov, prov, spell, crn, nhs_no, adm_meth etc
-- Order by nhs_no, admission_date, disch_date

declare curs1 cursor for

select prov_unit_code,spell_no,case when curr_prov_unit_code <> 'Eng' then
curr_prov_unit_code                 else prov_unit_code end,crn,nhs_no,
admission_method,admission_source,discharge_destination,
admission_date,coalesce(discharge_date,'31-December-2030')

from ##provider_spells_step_3

order by nhs_no,admission_date,coalesce(discharge_date,'31-December-2030'),
admission_method,admission_source,prov_unit_code,spell_no

-- Open the cursor:

open curs1

-- Fetch the first row from the cursor into the variables for prov_unit_code,
spell_no, curr_prov_unit_code etc

fetch next from curs1
into
```

```
@prov_unit_code,@spell_no,@curr_prov_unit_code,@crn,@nhs_no,
@admission_method,@admission_source,@discharge_destination,@admission_date,@discharge
_date


while @@fetch_status=0  -- While there are still rows to fetch
begin

if @nhs_no=@nhs_no_prev  -- If nhs_no = previous nhs no
 and (@admission_date between @admission_date_prev and @discharge_date_prev+1 --
Admission date falls between the previous spells admission date and one day after
previous spells discharge date
 and
 (@discharge_destination_prev between '55'and '57'
 or @admission_source between '55'and '57'
 or @admission_method in ('29','81'))) -- or current spell has a relevant admission
code
 or @admission_date between @admission_date_prev and @discharge_date_prev-1 -- Or
second admission starts before first ends, i.e. overlapping spell

begin
  if (@discharge_destination_prev between '55' and '57' or @admission_source between
'55'and '57'
  or @admission_method in ('29','81') or @admission_date between @admission_date_prev
and @discharge_date_prev-1)
  and @curr_prov_unit_code=@curr_prov_unit_code_prev -- If it has relevant
admission/discharge code, or it is overlapping and the curr_prov_unit_codes match,
then it is both a person and a prov spell:

    begin --person and prov
    set @prov_spell_order=@prov_spell_order+1 -- Add 1 to the prov spell order
    set @prov_spell=@prov_spell_prev -- The prov spell is the same as the previous
prov spell
    set @person_spell_order=@person_spell_order+1 -- Add 1 to the person spell order
    set @person_spell=@person_spell_prev -- The person spell is the same as the
previous person spell

 -- Then insert the data in to the prov/person spell look up table:

    insert into ##provider_spells_step_4
(prov_unit_code,spell_no,prov_spell,prov_spell_order,person_spell,person_spell_order)
    select
@prov_unit_code,@spell_no,@prov_spell,@prov_spell_order,@person_spell,@person_spell_o
rder
    end -- End of the both person and prov spell bit!

  else  -- Otherwise it will be a person spell not a prov spell
   begin
   set @prov_spell_order=1 -- Set prov spell order = 1
   set @prov_spell=@curr_prov_unit_code+@spell_no -- Set prov_spell ID
   set @person_spell_order=@person_spell_order+1 -- Add one to the person spell order
   set @person_spell=@person_spell_prev  -- Set person spell ID to be the previous
person spell ID


-- Then insert the data in to the prov/person spell look up table:

   insert into ##provider_spells_step_4
(prov_unit_code,spell_no,prov_spell,prov_spell_order,person_spell,person_spell_order)
   select
@prov_unit_code,@spell_no,@prov_spell,@prov_spell_order,@person_spell,@person_spell_o
rder
   end --person not prov
 end
```

```
else -- Otherwise it is neither a person spell or a prov spell

begin
 set @prov_spell_order=1 -- Set prov spell order = 1
 set @prov_spell=@curr_prov_unit_code+@spell_no -- Set prov_spell ID
 set @person_spell_order=1 -- Set person spell order = 1
 set @person_spell=@curr_prov_unit_code+@spell_no -- Set person_spell ID


-- Then insert the data in to the prov/person spell look up table:

insert into ##provider_spells_step_4
(prov_unit_code,spell_no,prov_spell,prov_spell_order,person_spell,person_spell_order)
 select
@prov_unit_code,@spell_no,@prov_spell,@prov_spell_order,@person_spell,@person_spell_o
rder
 end


-- Then set the current variables to the previous ones:

 set @person_spell_prev=@person_spell
 set @prov_spell_prev=@prov_spell
 set @curr_prov_unit_code_prev=@curr_prov_unit_code
 set @crn_prev=@crn
 set @nhs_no_prev=@nhs_no
 set @discharge_destination_prev=@discharge_destination
 set @admission_date_prev=@admission_date
 set @discharge_date_prev=@discharge_date

-- And fetch next from the cursor in to the current variables:

fetch next from curs1
into
@prov_unit_code,@spell_no,@curr_prov_unit_code,@crn,@nhs_no,
@admission_method,@admission_source,@discharge_destination,@admission_date,@discharge
_date

end

close curs1
deallocate curs1

-- 54 secs!

-- TABLE WITH RESULTS OF CURSOR, ##provider_spells_step_4, CREATED

-- SET MAX PROV AND MAX PERSON SPELL FLAGS:

update ##provider_spells_step_4
set max_prov_spell=1
from ##provider_spells_step_4  p
join (select prov_spell,max(prov_spell_order) as max_spell from
##provider_spells_step_4
                              group by prov_spell) m
on
p.prov_spell=m.prov_spell
and
p.prov_spell_order=m.max_spell

update ##provider_spells_step_4
set max_person_spell=1
from ##provider_spells_step_4  p
join (select person_spell,max(person_spell_order) as max_spell from
```

```
##provider_spells_step_4
                                group by person_spell) m
on
p.person_spell=m.person_spell
and
p.person_spell_order=m.max_spell


-- Cursor up table!

create clustered index ind_prov on ##provider_spells_step_4 (prov_unit_code,spell_no)


-- STEP FIVE: Add spells not already in ##provider_spells_step_4 in to
##provider_spells_step_5

select  distinct

      coalesce(prov_spell, s.prov_unit_code+s.spell_no) as provider_spell,
    coalesce(person_spell, s.prov_unit_code+s.spell_no) as person_spell,
      s.prov_unit_code,
      s.spell_no,
      s.admission_date,
      s.discharge_date,
      s.admission_method,
      prov_spell_order,
      person_spell_order,
      max_prov_spell,
      max_person_spell

into ##provider_spells_step_5

from ##provider_spells_step_1 s

      left outer join ##provider_spells_step_4 l on
      s.prov_unit_code=l.prov_unit_code
      and s.spell_no=l.spell_no

where s.admission_date >= '01-Apr-2005'

-- Delete person spells where the max person spell has a discharge destination
indicating that the person spell is unfinished, i.e. discharge destination in ('55',
'56', '57', '98')

Select distinct z.person_spell

into ##step5_temp

from ##provider_spells_step_5 z

join [IPU\NHSNational].monty2.dbo.dq_spell_subset s on
s.prov_unit_code = z.prov_unit_code and
s.spell_no = z.spell_no

where (max_person_spell = 1 or person_spell_order is null) -- Max spell or only spell
in person spell
and s.discharge_destination in ('55', '56', '57', '98')

-- Delete person spells

Delete ##provider_spells_step_5

where person_spell in

(Select distinct person_spell from ##step5_temp)
```

```
-- Put data in to permanent table:

truncate table userobj.analysis.Provider_spells_11_12

Insert into userobj.analysis.Provider_spells_11_12
Select * from ##provider_spells_step_5


drop table ##provider_spells_step_1
drop table ##provider_spells_step_2
drop table ##provider_spells_step_3
drop table ##provider_spells_step_4
drop table ##provider_spells_step_5


-- END OF PROVIDER SPELLS SCRIPT
```