

--Key points

We have Navigate processes on a different server to EBS

Navigate table which is updated daily = tblusers

--1 We have a table on the ebs server that contains the courses
and
tutors-----

select * from ebsDatabasename.dbo.navigate_courses

eg

course_code	tutor_email	tutor_first_name	tutor_surname
ABC	Smith	Ann	annsmith@college.ac.uk
DEF	Jackson	Lee	leejackson@college.ac.uk

so this can be populated with the course codes that are required for
Navigate, and the tutor details for each of those courses

--2 We then have a SQL view that uses this
table-----

USE [ebs server name]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER VIEW [dbo].[vw_navigate_active_learners] (
studentnumber,
studentfirstname,
studentsurname,
studentemail,
studentusername,
tutorfirstname,
tutorsurname,
tutoremail,
attribute1code,
attribute1,
attribute2code,
attribute2,
attribute3code,

```

        attribute3,
        attribute4code,
        attribute4,
        attribute5code,
        attribute5,
        attribute6code,
        attribute6,
        externalid
    ) AS
    (
        SELECT distinct
        P.PERSON_CODE,
        case when P.MIDDLE_NAMES is null then P.FORENAME else P.FORENAME+'
'+P.MIDDLE_NAMES end,
        P.SURNAME,
        cast(p.person_code as nvarchar)+'@collegename.ac.uk', --change to
        specific college
        cast(p.person_code as nvarchar)+'@collegename.ac.uk', --change to
        specific college
        navcc.tutor_first_name,
        navcc.tutor_surname,
        navcc.tutor_email,
        'Department',
        UI0.OWNING_ORGANISATION+' - '+ou.fes_full_name,
        'Course',
        UI.FES_UNIT_INSTANCE_CODE+' '+UI0.CALOCC_OCCURRENCE_CODE,
        'Group',
        isnull(TG.NAME, 'NO GROUP'),
        'EHCP/SEND',
        CASE WHEN z.has_healthcare_plan = 'Y' THEN 'yes' ELSE 'no' END,
        'Ethnicity',
        isnull(E.fes_long_description, 'not known/not provided'),
        'Year started',
        '20'+SubString(UI0.CALOCC_OCCURRENCE_CODE,1,2),
        P.PERSON_CODE
    FROM PEOPLE_UNITS PU with (nolock)
    INNER JOIN UNIT_INSTANCE_OCCURRENCES UI0 with (nolock)
    ON UI0.UI0_ID = PU.UI0_ID
    inner join UNIT_INSTANCES UI with (nolock)
    on UI.FES_UNIT_INSTANCE_CODE = UI0.FES_UINS_INSTANCE_CODE
    INNER JOIN ebsDatabasename.dbo.navigate_courses navcc with (nolock)
    ON navcc.course_code = pu.unit_instance_code
    INNER JOIN PEOPLE P with (nolock)
    ON P.PERSON_CODE = PU.PERSON_CODE
    left outer JOIN ORGANISATION_UNITS OU with (nolock)
    ON ou.organisation_code = UI0.OWNING_ORGANISATION
    left JOIN PEOPLE_ILR z with (nolock)
    ON z.person_code = P.PERSON_CODE

    left outer JOIN (

```

```

SELECT put.people_units_id,
tg.NAME
FROM PEOPLE_UNIT_TUTORGROUPS put with (nolock)
inner join TUTORGROUPS tg with (nolock)
on put.tutorgroup_id = tg.ID
) tg
ON tg.people_units_id = pu.id
left OUTER JOIN (
SELECT *
FROM VERIFIERS with (nolock)
WHERE rv_domain = 'ETHNICITY') E
ON E.low_value = P.ethnicity
WHERE
(uio.calocc_occurrence_code LIKE '2123%' OR uio.calocc_occurrence_code
LIKE '22%' ) --the years being chosen would need to be changed
AND pu.progress_status in ('A','F') and pu.unit_type = 'R')
GO

```

```

--3      We then have a SQL job that uses this view - several parts to
it-----
-----
-

```

```

--part A - pull the data onto the Navigate server
drop table if exists outputdatabase.dbo.tbl_navigate_active_learners
select *
into outputdatabase.dbo.tbl_navigate_active_learners
from [ebs server name].ebsDatabasename.dbo.vw_navigate_active_learners

```

```

create nonclustered index studentnumber on
outputdatabase.dbo.tbl_navigate_active_learners (studentnumber)
create nonclustered index attribute2 on
outputdatabase.dbo.tbl_navigate_active_learners (attribute2)
create nonclustered index tutoremail on
outputdatabase.dbo.tbl_navigate_active_learners (tutoremail)

```

```

--part B - insert new records - so if records from original view do
not exist in Naviagte table, they are moved into it
INSERT INTO tblusers
([StudentNumber],[StudentFirstName],[StudentSurname],[StudentEmail],
[StudentUsername],[TutorFirstName],[TutorSurname],[TutorEmail],
[Attribute1],[Attribute1Code],[Attribute2],[Attribute2Code],
[Attribute3],[Attribute3Code],[Attribute4],[Attribute4Code],
[Attribute5],[Attribute5Code],[Attribute6],[Attribute6Code],
[ExternalID],[DateProcessed],[Active])
SELECT distinct
StudentNumber,StudentFirstName,StudentSurname,StudentEmail,StudentUser
name,TutorFirstName,TutorSurname,TutorEmail,Attribute1,Attribute1Code,
Attribute2,
Attribute2Code,Attribute3,Attribute3Code,Attribute4,Attribute4Code,Att

```

```

        Attribute5,Attribute5Code,Attribute6,Attribute6Code,ExternalID ,NULL ,'1'
FROM outputdatabase.dbo.tbl_navigate_active_learners ebs
where not exists (
select * from tblusers nav
where nav.StudentNumber = ebs.StudentNumber
and nav.Attribute2 COLLATE DATABASE_DEFAULT = ebs.Attribute2 COLLATE
DATABASE_DEFAULT);

```

--part C - turn off non-active records - so if the records no longer exist in the refresh of the original view then they are made inactive and [DateProcessed] made null so the process picks up the changes

```

UPDATE tblusers
SET [Active] = 0,
    [DateProcessed] = null,
    [DateUpdated] = getdate()
WHERE [ImportID] in (
select nav.ImportID from tblusers nav
where nav.active = 1
and not exists (
select *
FROM outputdatabase.dbo.tbl_navigate_active_learners ebs
where ebs.StudentNumber = nav.StudentNumber
and ebs.Attribute2 COLLATE DATABASE_DEFAULT = nav.Attribute2 COLLATE
DATABASE_DEFAULT))

```

/*

part D - update existing records - so matching on the person code and attribute2 (course), if any of the below fields have changed then it updates those to the new value and

puts [DateProcessed] to be null so the process picks up the changes

*/

```

UPDATE tblusers
SET [StudentFirstName] = ebs.StudentFirstName
, [StudentSurname] = ebs.StudentSurname
, [Attribute3] = ebs.Attribute3
, [Attribute4] = ebs.Attribute4
, [Attribute5] = ebs.Attribute5
, [Attribute6] = ebs.Attribute6
, [Active] = 1
, [DateProcessed] = null
, [DateUpdated] = getdate()
from tblusers nav1
inner join outputdatabase.dbo.tbl_navigate_active_learners ebs
on ebs.StudentNumber = nav1.StudentNumber
and ebs.Attribute2 COLLATE DATABASE_DEFAULT = nav1.Attribute2 COLLATE
DATABASE_DEFAULT
and ebs.TutorEmail COLLATE DATABASE_DEFAULT = nav1.TutorEmail COLLATE
DATABASE_DEFAULT
WHERE nav1.ImportID in (
SELECT [ImportID]

```

```
FROM tblusers nav
inner join outputdatabase.dbo.tbl_navigate_active_learners ebs
on ebs.StudentNumber = nav.StudentNumber
and ebs.Attribute2 COLLATE DATABASE_DEFAULT = nav.Attribute2 COLLATE
DATABASE_DEFAULT
and ebs.TutorEmail COLLATE DATABASE_DEFAULT = nav.TutorEmail COLLATE
DATABASE_DEFAULT
where concat(nav.StudentNumber,
nav.Attribute2,nav.Tutoremail,nav.StudentFirstName,nav.StudentSurname,
nav.Attribute3,nav.Attribute4,nav.Attribute5,nav.Attribute6,nav.Active
) COLLATE DATABASE_DEFAULT
<>
concat(ebs.StudentNumber,
ebs.Attribute2,ebs.Tutoremail,ebs.StudentFirstName,ebs.StudentSurname,
ebs.Attribute3,ebs.Attribute4,ebs.Attribute5,ebs.Attribute6,1) COLLATE
DATABASE_DEFAULT
)
```