

Tilastokeskuksen opiskelijatiedonkeruu yliopistot

Ks. <http://stat.fi/keruu/ylio/>

"Yliopisto-opiskelijatiedonkeruu sisältää vain tutkintoon johtavan koulutuksen opiskelijat, jotka ovat ilmoittautuneet [päivämäärä] mennessä läsnä tai poissa oleviksi."

Poiminta

- Alustus
- Perusjoukon valinta
 - Opiskeluoikeuden valinta
- Tietueet
 - Yliopisto (kk)
 - Henkilötunnus (ht)
 - Uusi opiskelija (fuksi)
 - Ilmoittautumislaji keväällä (olotamm)
 - Ilmoittautumislaji syksyllä (olosyys)
 - Kirjoihintulo yliopistoon (kirtu)
 - Aineenopettajan pedagogiset opinnot (sv)
 - Opiskelumuoto (om)
 - Opiskelukunta (kkun)
 - Sukupuoli (sp)
 - Kansalaisuus (kansaly)
 - Erikoistumiskoulutus
 - Äidinkieli (aikieliy)
 - Asuinkunta (askuntay)
 - Sukunimi (snmi)
 - Etunimet (enimi)
 - Hyväksytty avoimen yliopiston opintojen perusteella (avo)
 - Lukukausimaksukokeilu ja maksullinen tilauskoulutus (lkmmt)
 - Opiskeluoikeuden alkaminen nykyisessä tutkinnossa (opoik)
 - Koulutuskoodi (koulk)
 - Opintopistetiedot (opes, opek, opker)
- Tulos
- Tarkistus
- Muutoshistoria
- 27.5.2019 :: HT4 tarkistuksen lisäys :: wiki-versio 31
- 12.10.2018:: Ohjauksenala tyhjä ja tutkintokoodin tarkistus :: wiki-versio 30
- 13.10.2017:: Erikoislääkäri kirjoilletulo päivämäärä (kirtu-muuttuja). :: wiki-versio 29
- 11.10.2017 ::olotamm3 käsittelyyn muutos tarkistuksessa :: wiki-versio 28
 - 17.10.2016 :: Lisätty erikoistumiskoulutus ohjauksenala poiminta, poikkeuspoiminta ei ilmoittautuneille erikoistumisopiskelijoille. Opiskeluluoituksen mukaa avo ja om muuttajat, avo= avoimen väylän kautta valitut, om=maisteri ohjelma, korvaa vanhan "muu" maisterin jota ei enään ole:: wikiversio 26
 - 22.1.2016 :: Lisätty perusjoukkoon mukaan erikoistumiskoulutusta suorittavat :: wikiversio 25
 - 30.09.2015 :: Päivitetty "Aineenopettajan pedagogiset opinnot (sv)" poimintaa, uudet pätevyyskoodit "ik","il","im" (entiset "1" ja "904") :: wikiversio 22
 - 8.12.2014 :: Tulos CSV-muotoiseksi :: wikiversio 21
 - 17.11.2014 :: Poistettu turhia sarakkeita ja lisätty avaintiedot :: wikiversio 19
 - 3.10.2014 :: Opiskelijatiedonkeruu 2014 :: wikiversio 17

Alustus

Dokumentaation tueksi on tuotu SQL-kielisiä kyselyitä, joissa taustalla on temp-tietokantataulu #tkopisk johon tiedot kerätään, joka vastaa tietuekuvaukseltaan hyvin lähelle sitä mikä Tilastokeskuksen tiedonkeruissa halutaan, sekä yksirivinen temp-tietokantataulu #param, jossa on tieto nyt käsittelyssä olevasta korkeakoulusta sekä tiedonkeruun kannalta oleellinen tilastointipäivä (20.9.VUOSI) ja lisäksi on erikseen vielä vuosi-tieto kyselyitä yksinkertaistamaan.

SQL

```
-- Temp-taulu tietojen valintaa varten
-- NB! param-taulussa saa olla vain yksi rivi!
IF OBJECT_ID('tempdb..#param') IS NOT NULL
BEGIN
    truncate table #param
```

```

        drop table #param
        if @debug<>0 print convert(varchar,getdate(),120)+' DROP temp-
taulu param''
END
;
create table #param (
    vuosi varchar(4),
    org varchar(30),--KK-lyhenne (DB-nimi)
    kk varchar(2),--Yliopisto (positiot 1-2)
    tilpvm date,--Tilastointipäivä
)
if @debug<>0 print convert(varchar,getdate(),120)+' CREATE temp-taulu
param''
;
insert into #param (vuosi,org,kk,tilpvm)
select
    @vuosi as vuosi
    ,UPPER(DB_NAME()) as org
    ,(case
        when UPPER(DB_NAME()) = 'AA' then '03'
        when UPPER(DB_NAME()) = 'AYO' then '25'
        when UPPER(DB_NAME()) = 'HY' then '01'
        when UPPER(DB_NAME()) = 'ISY' then '26'
        when UPPER(DB_NAME()) = 'JY' then '06'
        when UPPER(DB_NAME()) = 'LAY' then '21'
        when UPPER(DB_NAME()) = 'LTY' then '14'
        when UPPER(DB_NAME()) = 'OY' then '04'
        when UPPER(DB_NAME()) = 'SHH' then '10'
        when UPPER(DB_NAME()) = 'TAIY' then '28'
        when UPPER(DB_NAME()) = 'TaY' then '05'
        when UPPER(DB_NAME()) = 'TTY' then '15'
        when UPPER(DB_NAME()) = 'TY' then '27'
        when UPPER(DB_NAME()) = 'VY' then '13'
        else 'NA'
    end
    ) as kk
    ,@vuosi+'-09-20' as tilpvm
;
--select * from #param

-- Temp-taulu tiedonkeruun tietoja varten
IF OBJECT_ID('tempdb..#tkopisk') IS NOT NULL
BEGIN
    truncate table #tkopisk
    drop table #tkopisk
    if @debug<>0 print convert(varchar,getdate(),120)+' DROP temp-
taulu tkopisk''
END
;
create table #tkopisk (
    kk varchar(2),--Yliopisto (positiot 1-2)
    ht varchar(11),--Henkilötunnus (positiot 3-13)

```

```

fuksi varchar(1),--Tietueen tunnus (positio 14)
olotamm varchar(1),--Ilmoittautumislaji keväällä (positio 15)
olosyys varchar(1),--Ilmoittautumislaji syksyllä (positio 16)
kirtu varchar(5),--Kirjoihintulo yliopistoon (positiot 17-21)
--tdk varchar(2),--Tiedekunta/osasto (positiot 22-23)
--tutk varchar(3),--Tutkinto (positiot 24-26)
--ko varchar(4),--Koulutusohjelma (positiot 27-30)
--paine varchar(4),--Pääaine (positiot 31-34)
sv varchar(3),--Aineenopettajan pedagogiset opinnot (positiot 35-
37)
--opalay varchar(2),--Opintoala (positiot 38-39)
om varchar(1),--Opiskelumuoto (positio 40)
kkun varchar(3),--Opiskelukunta (positiot 41-43)
sp varchar(1),--Sukupuoli (positio 44)
kansaly varchar(3),--Kansalaisuus (positiot 45-47)
aikieliy varchar(1),--Äidinkieli (positio 48)
askuntay varchar(3),--Asuinkunta (positiot 49-51)
snimi nvarchar(30),--Sukunimi (positiot 52-81)
enimi nvarchar(15),--Etunimet (positiot 82-96)
avo varchar(1),--Hyväksytty avoimen yliopiston opintojen
perusteella (positio 97)
opes varchar(3),--Edellisenä syyslukukautena (1.8.-31.12.)
suoritettujen opintopisteiden määrä (positiot 98-100)
opek varchar(3),--Edellisenä kevätlukukautena (1.1.-31.7.)
suoritettujen opintopisteiden määrä (positiot 101-103)
opker varchar(4),--Edellisen lukuvuoden loppuun (31.7.) mennessä
suoritettujen kaikkien opintopisteiden määrä (positiot 104-106)
-- NB muutettu 28.4.2014 4
merkkiin, vaikka 2013 oli vielä 3 merkkiä!
lkmmt varchar(1),--Lukukausimaksukokeilu ja maksullinen
tilauskoulutus (positio 107)
opoik varchar(5),--Opiskeluoikeuden alkaminen nykyisessä
tutkinnossa (positiot 108-112)
koulk varchar(6),--Koulutuskoodi (positiot 113-118)
--
opiskelijaavain varchar(100),
opiskeluoikeusavain varchar(100),
opiskelija_id int,
opiskeluoikeus_id int
)
CREATE NONCLUSTERED INDEX IX_tkopisk_opiskelija_id ON #tkopisk
(opiskelija_id)
CREATE NONCLUSTERED INDEX IX_tkopisk_opiskeluoikeus_id ON #tkopisk
(opiskeluoikeus_id)
if @debug<>0 print convert(varchar,getdate(),120)+' ' CREATE temp-taulu
tkopisk''
;

```

Poimitaan aputauluun perustiedot opiskelijoista (henkilöistä). Perusjoukon poiminta tehdään tarkoituksella vain henkilö- ja opiskelija-taulu pohjaisesti. Kun opiskelijat on saatu valittua, valitaan kullekin opiskelijalle opiskeluoikeus ensisijaisuuden perusteella.

Opiskelija tulee poimituksi, mikäli:

- on tilastointipäivänä aktiivinen opiskeluoikeus, joka
- on tutkintoon (opiskeluoikeuden tyyppi 1-7) johtava TAI erikoistumiskoulutus (tyyppi 19) ensisijainen opiskeluoikeus, sekä
- on läsnä- tai poissaolo lukukausi-ilmoittautuminen syksylle (ilmoittautumispäivä 20.9. mennessä)
 - katsotaan lisäksi, ettei mukana ole tutkinnon suorittaneita ajalla 1.8.-19.9.
- Erikoistumiskoulutus joille ei löydy ilmoittautumistietoja lisätty poikkeuspoiminta jotta saadaan mukaan niiden korkeakoulujen joilta tämä puuttuu

Samalla täytetään tiedot:

- Korkeakoulu (kk)
- Henkilötunnus (ht)
- Sukupuoli (sp)
- Sukunimi (snimi)
- Etunimet (enimi)
- Kansalaisuus (kansaly) :: nb! vain suomalaiset
- Äidinkieli (aikieliy)
- Asuinkunta (askuntay)
- Kirjoitetulo (kirtu) :: tässä päätellään lukukausi kuukauden perusteella: suurempi tai yhtäsuuri kuin elokuu => syyslukukausi, muuten kevätlukukausi

SQL

```
---
-- Lisätään perustiedot (henkilö, opiskelija)
-- * täytyy olla tutkinto-oikeus
-- * täytyy olla ilmoittautuminen syksylle (20.9. mennessä)
-- * täytyy olla opiskeluoikeuden tila on 1=aktiivinen (20.9.)
-- - katsotaan erikseen myös ne, jotka ovat suorittaneet tutkinnon ajalla
1.8.-19.9.,
-- että eivät tule mukaan opiskeluoikeuden tilasta riippumatta
-- 2016-01-22: Lisätty perusjoukkoon myös Erikoistumiskoulutus (19) mutta
koskeeko heitä ilmoittautuminen? - AMK-puolelta otettu mallia...
---
INSERT into #tkopisk (kk,ht,sp,snimi,enimi,kansaly,aikieliy,askuntay,kirtu,
opiskelijaavain,opiskelija_id)
SELECT
  kk = (select top 1 kk from #param)
,ht = h.Henkilotunnus
,sp = (select top 1 k.Koodi from Sukupuoli k where k.ID=h.SukupuoliID)
,snimi = substring(h.Sukunimi,1,30)
,enimi = substring(h.Etunimet,1,15)
,kansaly =
--nb! vain suomalaiset tässä (jää null!)
(select top 1 m.Koodi
 from Henkilon_maa hm
 inner join Maa m on m.ID=hm.MaaID
 inner join Maan_rooli mr on mr.ID=hm.Maan_rooliID
 where hm.HenkiloID=h.ID --sido
 and mr.Koodi='1' --Kansalaisuus
 and m.Koodi='246' --Suomi
```

```

)
,aikieliy = (select top 1
  case k.Koodi when 'fi' then '1' when 'sv' then '2' else '3' end
  from Kieli k where k.ID=h.KieliID)
,askuntay = coalesce((select top 1 k.Koodi from Kunta k where k.ID=h.
KuntaID),'999')
,kirtu = (cast(year(p.Kirjoihintulopaivamaara) as varchar)
  +(case
    -- 8-12 (elo-joulu): syksy
    when month(p.Kirjoihintulopaivamaara)>=8 then '2'
    -- 1-7 (tammi-heinä): kevät
    else '1'
    end)
)
,opiskelijaavain = p.Avain
,opiskelija_id = p.ID
FROM Henkilo h
inner join dbo.Opiskelija p on p.HenkiloID=h.ID
WHERE 1=1
-- tutkinto-oikeus
and p.ID in (
  select o.OpiskelijaID
  from Opiskeluoikeus o
  inner join Opiskeluoikeuden_tyyppi ot on ot.ID=o.
Op_oikeuden_tyyppiID
  inner join Op_oikeuteen_liittyva_tila lt on lt.OpiskeluoikeusID=o.
ID
  inner join Opiskeluoikeuden_tila kt on kt.ID=lt.Op_oikeuden_tilaID
  where 1=1
  -- tutkinto-tyyppinen, erikoistumiskoulutus
  and ot.Koodi in ('2','4','5','6','7','19')
  -- voimassa tilpvm
  and (select tilpvm from #param) between o.Alkamispaivamaara and
coalesce(o.Paattymispaivamaara,'9999-01-01')
  -- tila
  and kt.Koodi = '1' --aktiivinen
  and (select tilpvm from #param) between lt.Alkamispaivamaara and
coalesce(lt.Paattymispaivamaara,'9999-01-01')
  -- EI tutkinnon suorittaneita mukaan, vaikka opiskeluoikeuden tila
olisikin jäänyt päivittämättä
  -- (tämän opiskeluoikeuden)
  and o.ID NOT IN (
    select s.OpiskeluoikeusID
    from Opintosuoritus s
    inner join Tutkintonimike sk on sk.ID=s.TutkintonimikeID
    -- liitetään mukaan myös opiskeluoikeusjakso, jotta
tiedetään kyseessä olevan SAMA tutkinto
    inner join Opiskeluoikeusjakso oj on oj.OpiskeluoikeusID=s.
OpiskeluoikeusID
    inner join Tutkintonimike ok on ok.ID=oj.TutkintonimikeID
    where s.OpiskelijaID=p.ID
    and Opintosuorituksen_lajiID in (select id from
Opintosuorituksen_laji where Koodi='1')--tutkinto

```

```

        and Suorituspaivamaara between (select vuosi from #param)
+''-08-01'' and (select vuosi from #param)+''-09-19''
        --sama tutkinto
        and sk.ID=ok.ID
    )
)
-- ilmo
and p.ID in (
    select OpiskelijaID
    from Lukukausi_ilmoittautuminen l
    inner join Lukukausi_ilm_tila lt on lt.ID=l.
Lukukausi_ilmoittautumisen_tilaID
    where lt.Koodi in (''1'', ''2'', ''3'') --läsnä, poissa, poissa ei
kuluta
    and l.Ilmoittautumispaivamaara <= (select tilpvm from #param)
    and (select tilpvm from #param) between l.Alkamispaivamaara and
coalesce(l.Paattymispaivamaara, ''9999-01-01'')
)
;

-- Erikoistumiskoulutusopiskelija joilla ei ole läsnäolotietoja

INSERT into #tkopisk (kk,ht,sp,snimi,enimi,kansaly,aikieliy,askuntay,kirtu,
opiskelijaavain,opiskelija_id,opiskeluoikeusavain,opiskeluoikeus_id,opoik)
SELECT
    kk = (select top 1 kk from #param)
,ht = h.Henkilotunnus
,sp = (select top 1 k.Koodi from Sukupuoli k where k.ID=h.SukupuoliID)
,snimi = h.Sukunimi
,enimi = h.Etunimet
,kansaly =
--nb! vain suomalaiset tässä (jää null!)
(select top 1 m.Koodi
    from Henkilon_maa hm
    inner join Maa m on m.ID=hm.MaaID
    inner join Maan_rooli mr on mr.ID=hm.Maan_rooliID
    where hm.HenkiloID=h.ID --sido
    and mr.Koodi='1' --Kansalaisuus
    and m.Koodi='246' --Suomi
)
,aikieliy = (select top 1
    case k.Koodi when 'fi' then '1' when 'sv' then '2' else '3' end
    from Kieli k where k.ID=h.KieliID)
,askuntay = coalesce((select top 1 k.Koodi from Kunta k where k.ID=h.
KuntaID), '999')
,kirtu = (cast(year(o.alkamispaivamaara) as varchar)
    +(case
        -- 8-12 (elo-joulu): syksy
        when month(p.Kirjoihintulopaivamaara)>=8 then '2'
        -- 1-7 (tammi-heinä): kevät
        else '1'
    end)

```

```

)
,opiskelijaavain = p.Avain
,opiskelija_id = p.ID
,opiskeluoikeusavain = o.avain
,opiskeluoikeus_id = o.ID
,opoik = (cast(year(o.alkamispäivä) as varchar)
+(case
-- 8-12 (elo-joulu): syksy
when month(p.Kirjoituspäivä)>=8 then '2'
-- 1-7 (tammi-heinä): kevät
else '1'
end)
)

FROM Henkilo h
INNER JOIN Opiskelija p ON p.HenkiloID=h.ID
INNER JOIN Opiskeluoikeus o ON o.OpiskelijaID = p.id
INNER JOIN Opiskeluoikeuden_tyyppi ot ON ot.ID = o.
Op_oikeuden_tyyppiID
INNER JOIN Op_oikeuteen_liittyvä_tila lt ON lt.OpiskeluoikeusID =
o.ID
INNER JOIN Opiskeluoikeuden_tila kt ON kt.ID = lt.
Op_oikeuden_tilaID

WHERE p.id NOT IN (SELECT T.opiskelija_id FROM #tkopisk T)
-- erikoistumiskoulutus, eikä edellisessä jossa haettiin
ilmoittautumisia
AND ot.Koodi in ('19')

-- voimassa tilpvm
AND (select tilpvm from #param) between o.Alkamispäivä and coalesce(o.
Paattymispäivä,'9999-01-01')
-- tila
AND kt.Koodi = '1' --aktiivinen

AND (select tilpvm from #param) between lt.Alkamispäivä and coalesce
(lt.Paattymispäivä,'9999-01-01')

```

Opiskeluoikeuden valinta

Opiskeluoikeus päätellään hyvin samoin kuin opiskelija valinnassa yllä, mutta per opiskelija siten että opiskeluoikeus on:

- tutkintoon johtava (opiskeluoikeuden tyyppi 1-7) ja
- voimassa tilastointipäivänä, sekä lisäksi
- tilastointipäivänä voimassaoleva tila on aktiivinen ja
- ensisijaisuus on voimassa tilastointipäivänä

Samalla täytetään opiskeluoikeustietoja:

- Opiskeluoikeuden alkaminen nykyisessä tutkinnossa (opoik)

SQL

```
-- valitaan yksi per opiskelija
update T
set opiskeluoikeus_id=o.ID, opiskeluoikeusavain=o.avain
  , opoik=(cast(year(o.Alkamispaivamaara) as varchar)
          +(case
              -- 8-12 (elo-joulu): syksy
              when month(o.Alkamispaivamaara)>=8 then '2'
              -- 1-7 (tammi-heinä): kevät
              else '1'
            end)
          )
from #tkopisk T
inner join Opiskeluoikeus o on o.OpiskelijaID=T.opiskelija_id
inner join Opiskeluoikeuden_tyyppi ot on ot.ID=o.Op_oikeuden_tyyppiID
inner join Op_oikeuteen_liittyva_tila lt on lt.OpiskeluoikeusID=o.ID
inner join Opiskeluoikeuden_tila kt on kt.ID=lt.Op_oikeuden_tilaID
where l=1
-- tutkinto-tyyppinen, erikoistumiskoulutus
and ot.Koodi in ('1','2','3','4','5','6','7','19')
-- voimassa tilpvm
and (select tilpvm from #param) between o.Alkamispaivamaara and coalesce(o.
Paattymispaivamaara,'9999-01-01')
-- tila
-- NB! Katsotaan nyt vain aktiivinen (ei päättynyt). Riski on juuri 1.8.
tutkinnon suorittaneet -- ehkä?
and kt.Koodi = '1' --aktiivinen
and (select tilpvm from #param) between lt.Alkamispaivamaara and coalesce
(lt.Paattymispaivamaara,'9999-01-01')
-- ensisijainen tilpvm
and o.ID in (
  select oe.OpiskeluoikeusID from Op_oikeuden_ensisijaisuus oe
  -- NB! vain tilanpäivä! päättyneillä ensisijaisuuden täytyy siis
  jatkoa?
  where (select tilpvm from #param) between oe.Alkamispaivamaara and
  coalesce(oe.Paattymispaivamaara,'9999-01-01')
)
;
```

Päivitetään sitten muita opiskeluoikeustietoja:

- Avoimen väylä (avo), opiskeluoikeuden luokitus koodilla 4= Myönnetty avoimen opintojen perusteella.

SQL

```
-- avo
-- Myönnetty avoimen opintojen perusteella. Avoimen väylä muutos 17.10.2016
UPDATE      T
SET        avo = '1'      -- avoimen perusteella vain uusista opiskelijoista
FROM      #tkopisk T
          INNER JOIN      Opiskeluoikeus o ON o.ID = T.opiskeluoikeus_id
          INNER JOIN      Opiskeluoikeusjakso oj ON oj.OpiskeluoikeusID = T.
opiskeluoikeus_id
          LEFT JOIN      Tutkintonimike kt on kt.ID = oj.TutkintonimikeID
          INNER JOIN      Op_oik_jaks_liittyva_luokittelu OOJL ON oj.id = OOJL.
OpiskeluoikeusjaksoID
          INNER      JOIN Opiskeluoikeuden_luokittelu OL ON OL.ID = OOJL.
Opiskeluoikeuden_luokitteluID
WHERE      OL.koodi = '4'      -- Myönnetty avoimen opintojen
perusteella. Avoimen väylä
AND      (SELECT tilpvm FROM #param) BETWEEN oj.Alkamispaivamaara AND
COALESCE(oj.Paattymispaivamaara, '9999-01-01')
AND      YEAR(o.alkamispaivamaara) = cast(@vuosi as int)
```

Päivitetään sitten seuraavat opiskeluoikeuskohtaiset tiedot, joista tarkempaa selitystä myöhemmin tässä dokumentissa:

- Koulutuskoodi (koulk); erikoistumiskoulutukselle päätellään arvo 999999
- Opiskelukunta (kkun)
- Lukukausimaksukokeilu ja maksullinen tilauskoulutus (lkmmt)

SQL

```
-- koulk, kkun, lkmmt
update T
set koulk=kt.Koodi --!
  , kkun=ku.Koodi
  , lkmmt=
  (case (select k.Koodi from Rahoituslahde k where k.ID=oj.
RahoituslahdeID)
  when '4' then '2' -- 4=Maksullinen tilauskoulutus
  when '5' then '1' -- 5=Lukukausimaksukokeilu
  else '' end)
from #tkopisk T
inner join Opiskeluoiikeus o on o.ID=T.opiskeluoiikeus_id
inner join Opiskeluoiikeusjakso oj on oj.OpiskeluoiikeusID=T.
opiskeluoiikeus_id
left join Tutkintonimike kt on kt.ID=oj.TutkintonimikeID
left join Kunta ku on ku.ID=oj.KuntaID
where (select tilpvm from #param) between oj.Alkamispaivamaara and coalesce
(oj.Paattymispaivamaara, '9999-01-01')
;
-- päivitetään tyhjän arvon tilalle vielä tutkintokeruuta mukailten
999999, kun kyse ei ole tutkinnonsuoritusoikeudesta
update T
set koulk='999999'
from #tkopisk T
inner join Opiskeluoiikeus o on o.ID=T.opiskeluoiikeus_id
inner join Opiskeluoiikeuden_tyyppi ot on ot.ID=o.Op_oikeuden_tyyppiID
where l=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
and ot.Koodi in ('19')
-- ja ei ollut arvoa vielä
and koulk is null
;
```

Tietueet

Yliopisto (kk)

Yliopiston tunnuskoodi on parametri-aputaulun luomisen yhteydessä päätelty arvo (kovakoodattu). Parametri-taulusta tieto saatiin kullekin riville jo ensimmäisissä poiminnoissa.

Henkilötunnus (ht)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa.

Uusi opiskelija (fuksi)

Päätellään kirjoihintulopäivämäärästä. Jos kirjoihintulopäivämäärä on tilastointipäivän mukaisen vuoden puolella uusi opiskelija saa arvon 1=kyllä, muutoin arvo 0=ei uusi opiskelija.

SQL

```
update T
set fuksi='1'
from #tkopisk T
where SUBSTRING(kirtu,1,4)=(select vuosi from #param)
;
update T
set fuksi='0'
from #tkopisk T
where fuksi is null
;
```

Ilmoittautumislaji keväällä (olotamm)

Mikäli syksyn uusi opiskelija kirjoilletulopäivämäärän perusteella, arvoksi tulee tyhjä. Muutoin katsotaan suoraan ko. opiskelijan lukukausi-ilmoittautumistieto siten, että lukukausi-ilmoittautumisen tila 1=läsnä on suoraan sama arvo tuloksessa ja arvot 2=poissa ja 3=poissa, ei kuluta aikaa saavat tuloksessa arvon 2=poissa.

NB! Lukukausi-tiedon päättelemiseksi käytetään lukukauden viimeistä päivää eli päivämäärää 31.7. (heinäkuun 31. päivä).

SQL

```
-- olotamm :: kevään ilmo tilanne
-- : uusilla tyhjä/'0'
update T
set olotamm='''
from #tkopisk T
where kirtu=(select vuosi from #param)+'2'
;
update T
set olotamm='1'
from #tkopisk T
where opiskelija_id in (
    select OpiskelijaID
    from Lukukausi_ilmoittautuminen l
    inner join Lukukausi_ilm_tila lt on lt.ID=l.
Lukukausi_ilmoittautumisen_tilaID
    where lt.Koodi='1' --läsnä
    and (select vuosi from #param)+'-07-31' between l.
Alkamispaivamaara and coalesce(l.Paattymispaivamaara,'9999-01-01')
    -- EI tilpvm
)
and olotamm is null
;
update T
set olotamm='2'
from #tkopisk T
where opiskelija_id in (
    select OpiskelijaID
    from Lukukausi_ilmoittautuminen l
    inner join Lukukausi_ilm_tila lt on lt.ID=l.
Lukukausi_ilmoittautumisen_tilaID
    where lt.Koodi in ('2','3') --poissa, poissa ei kuluta
    and (select vuosi from #param)+'-07-31' between l.
Alkamispaivamaara and coalesce(l.Paattymispaivamaara,'9999-01-01')
    -- EI tilpvm
)
and olotamm is null
;
-- ei ilmoa keväällä, tyhjäksi
update T
set olotamm='''
from #tkopisk T
where olotamm is null
;
```

Ilmoittautumislaji syksyllä (olosyys)

Katsotaan suoraan ko. opiskelijan lukukausi-ilmoittautumistieto siten, että lukukausi-ilmoittautumisen tila 1=läsnä on suoraan sama arvo tuloksessa ja arvot 2=poissa ja 3=poissa, ei kuluta aikaa saavat tuloksessa arvon 2=poissa. Lisäksi katsotaan, että ilmoittautuminen on tapahtunut ennen tilastointipäivää.

NB! Lukukausi-tiedon päättelemiseksi käytetään suoraan tilastointipäivää eli päivämäärää 20.9. (syyskuun 20. päivä).

SQL

```
-- olosyys :: syksyn ilmo tilanne
update T
set olosyys='1'
from #tkopisk T
where opiskelija_id in (
    select OpiskelijaID
    from Lukukausi_ilmoittautuminen l
    inner join Lukukausi_ilm_tila lt on lt.ID=l.
Lukukausi_ilmoittautumisen_tilaID
    where lt.Koodi='1' --läsnä
    and (select tilpvm from #param) between l.Alkamispaivamaara and
coalesce(l.Paattymispaivamaara,'9999-01-01')
    -- tilpvm
    and l.Ilmoittautumispaivamaara <= (select tilpvm from #param)
)
and olosyys is null
;
update T
set olosyys='2'
from #tkopisk T
where opiskelija_id in (
    select OpiskelijaID
    from Lukukausi_ilmoittautuminen l
    inner join Lukukausi_ilm_tila lt on lt.ID=l.
Lukukausi_ilmoittautumisen_tilaID
    where lt.Koodi in ('2','3') --poissa, poissa ei kuluta
    and (select tilpvm from #param) between l.Alkamispaivamaara and
coalesce(l.Paattymispaivamaara,'9999-01-01')
    -- tilpvm
    and l.Ilmoittautumispaivamaara <= (select tilpvm from #param)
)
and olosyys is null
;
```

Kirjohintulo yliopistoon (kirtu)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa. Tieto on siis suoraan opiskelijan tieto kirjohintulopäivä, josta päivämäärästä pätee llään lukukausi kuukauden perusteella: suurempi tai yhtäsuuri kuin elokuu => syyslukukausi, muuten kevätlukukausi

Aineenopettajan pedagogiset opinnot (sv)

Päätely aineenopettajan pedagogisista opinnoista tehdään sen perusteella liittyykö valittuun opiskeluoikeuteen opiskeluoikeusjaksoa johon liittyy pätevyys-tieto joka on joku seuraavista:

- "ii" ja "ik" =Opettaja
- "im"=Erilliset opettajan pedagogiset opinnot

Jos liittyy asetetaan '001', muuten tyhjä.

[Opettaja pätevyyskoodit](#)

SQL

```
-- sv
update T
set sv='001'
from #tkopisk T
inner join Opiskeluoiikeusjakso oj on oj.OpiskeluoiikeusID=T.
opiskeluoiikeus_id
inner join Opiskeluoiikeusjakson_patevyys jp on jp.OpiskeluoiikeusjaksoID=oj.
ID
inner join Patevyys p on p.ID=jp.PatevyysID
where (select tilpvm from #param) between oj.Alkamispaiivamaara and coalesce
(oj.Paattymispaiivamaara,'9999-01-01')
and (select tilpvm from #param) between jp.Alkamispaiivamaara and coalesce
(jp.Paattymispaiivamaara,'9999-01-01')
and p.Koodi in ('ik','il','im') --opettaja, erill opet pedagog opinn
and sv is null
;
-- tyhjät: '' vai '000'?
update T
set sv=''
from #tkopisk T
where sv is null
;
```

Opiskelumuoto (om)

Maisteriohjelma -tieto opiskeluoiikeusluokituksen koodi arvo 5= Maisteriohjelma

SQL

```
-- om muuttuja

-- maisteriohjelma muutos 17.10.2016
UPDATE      T
SET        om = '1'
FROM        #tkopisk T
           INNER JOIN      Opiskeluoiikeus o ON o.ID = T.opiskeluoiikeus_id
           INNER JOIN      Opiskeluoiikeusjakso oj ON oj.OpiskeluoiikeusID = T.
opiskeluoiikeus_id
           LEFT JOIN       Tutkintonimike kt on kt.ID = oj.TutkintonimikeID
           INNER JOIN      Op_oik_jaks_liittyva_luokittelu OOJL ON oj.id = OOJL.
OpiskeluoiikeusjaksoID
           INNER          JOIN Opiskeluoiikeuden_luokittelu OL ON OL.ID = OOJL.
Opiskeluoiikeuden_luokitteluID
WHERE      OL.koodi = '5'           -- Entinen (Muu) maisteriohjelma
AND        (SELECT tilpvm FROM #param) BETWEEN oj.Alkamispaiivamaara AND
COALESCE(oj.Paattymispaiivamaara,'9999-01-01')
AND        YEAR(o.alkamispaiivamaara) = cast(@vuosi as int);
```

Opiskelukunta (kkun)

Tieto sellaisenaan tietovarannosta valituksi tulleesta opiskeluoikeudesta tilastointipäivänä voimassaolleesta opiskeluoikeusjaksosta.

Sukupuoli (sp)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa.

Kansalaisuus (kansaly)

Perusjoukon keräämisen yhteydessä lisättiin suomalaiset. Päivitetään erikseen kansalaisuus vielä niille, joilta kansalaisuus puuttuu, mutta on tietovarannossa jokin muu kuin Suomi. Jos monta kansalaisuutta, valikoituu kansalaisuus "satunnaisesti".

Lopuksi korjataan vielä puuttuvat tuntemattomiksi.

SQL

```
-- kansaly
-- : muut kuin suomi
update T
set kansaly=(
  select top 1 m.Koodi
  from Henkilon_maa hm
  inner join Maa m on m.ID=hm.MaaID
  inner join Maan_rooli mr on mr.ID=hm.Maan_rooliID
  inner join Opiskelija p on p.HenkiloID=hm.HenkiloID
  where p.ID=T.opiskelija_id --sido
  and mr.Koodi='1' --Kansalaisuus
)
from #tkopisk T
where kansaly is null
;
-- tuntemattomia
-- : kansaly
update T
set kansaly='999'
from #tkopisk T
where kansaly is null
;
```

Erikoistumiskoulutus

Erikoistumiskoulutus, opiskeluoikeustyyppi 19, Tilastokeskus koulutustyyppi koodi = 7.

Ohjauksenala ei tutkintoon johtavalle koulutukselle (koulutuskoodi 999999). Lisätään OPH koulutuskoodi mikäli on tuotu.

Erikoistumiskoulutus

```
-- Erikoistumiskoulutus, ei-tutkintokoulutus 999999
UPDATE T
SET T.ohjauksenAla = K.koodi,
    T.koulu = '999999',
    T.opiskeluoikeusavain = o.avain,
    T.opiskeluoikeus_id = o.id
```

```

FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.Op_oikeuden_tyyppiID
    INNER JOIN koulutusala K ON K.id = O.koulutusalaid
WHERE     1=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND K.versio = 'ohjausala'

-- Erikoistumiskoulutus, 999999 eikä ole tuotu ohjauksen alaa?
UPDATE T
SET T.koulu = '999999',
    T.opiskeluoiikeusavain = o.avain,
    T.opiskeluoiikeus_id = o.id
FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.Op_oikeuden_tyyppiID
    INNER JOIN koulutusala K ON K.id = O.koulutusalaid
WHERE     1=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND K.versio <> 'ohjausala'

-- Erikoistumiskoulutuksen koulutuskoodi jos on
UPDATE T
SET T.ekkoodi = E.koodi
FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.Op_oikeuden_tyyppiID
    INNER JOIN koulutusala K ON K.id = O.koulutusalaid
    INNER JOIN op_oik_liittyva_erikoistumiskoulutus OE ON OE.
opiskeluoiikeusid = o.id
    INNER JOIN Erikoistumiskoulutus E ON E.id = OE.erikoistumiskoulutusid
WHERE     1=1
-- erikoistumisopinnot (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND K.versio = 'ohjausala'

-- Ei läsnäoloa, tulkitaan aktiivinen tila läsnäoloksi
Erikoistumiskoulutuksen tyyppi 19
UPDATE T
SET T.olosyys = '1'
FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.Op_oikeuden_tyyppiID
    INNER JOIN koulutusala K ON K.id = O.koulutusalaid
WHERE     1=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND K.versio = 'ohjausala'

```



```

AND T.olosyys IS NULL

UPDATE T
SET T.olosyys = '1'
FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.Op_oikeuden_tyyppiID
    INNER JOIN koulutusala K ON K.id = O.koulutusalaid
WHERE      1=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND K.versio <> 'ohjausala'
AND T.olosyys IS NULL

UPDATE T
SET T.kirtu = T.opoik
FROM #tkopisk T
    INNER JOIN Opiskeluoiikeus o ON o.id = T.opiskeluoiikeus_id
    INNER JOIN Opiskeluoiikeuden_tyyppi ot ON ot.ID = o.
Op_oikeuden_tyyppiID
WHERE      1=1
-- erikoistumiskoulutus (siis EI tutkinto-tyyppinen)
AND ot.Koodi in ('19')
AND T.kirtu IS NULL
AND T.opoik IS NOT NULL

```

Äidinkieli (aikieliy)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa. Puuttuvat muutetaan arvoon 3=muu.

Asuinkunta (askuntay)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa. Puuttuvat muutettiin samalla tuntemattomiksi.

Sukunimi (snmi)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa.

Etunimet (enimi)

Lisättiin tietovarannossa oleva tieto jo perusjoukon valinnassa.

Hyväksytty avoimen yliopiston opintojen perusteella (avo)

Mikäli kyseessä on uusi opiskelija (opiskeluoiikeuden alkaminen nykyisessä tutkinnossa (OPOIK) sisältää tilastovuoden), katsotaan opiskeluoiikeudesta avoimen väylä -tieto onko se kyllä vai ei. Jos kyllä niin 1=kyllä, muuten 0=ei.

Lukukausimaksukokeilu ja maksullinen tilauskoulutus (lkmmt)

Tieto sellaisenaan tietovarannosta tilastointipäivänä voimassaoleesta opiskeluoiikeusjaksosta rahoituslähde-tiedosta.

Opiskeluoiikeuden alkaminen nykyisessä tutkinnossa (opoik)

Tieto sellaisenaan tietovarannon tiedosta opiskeluoikeuden alkamispäivä. Päätellään lukukausi kuukauden perusteella: suurempi tai yhtäsuuri kuin elokuu => syyslukukausi, muuten kevätlukukausi.

Koulutuskoodi (koulk)

Tieto sellaisenaan tietovarannosta tilastointipäivänä voimassaoleesta opiskeluoikeusjaksosta tutkintonimike-tiedosta.

Opintopistetiedot (opes, opek, opker)

- Edellisenä syyslukukautena (1.8.-31.12.) suoritettujen opintopisteiden määrä (opes)
- Edellisenä kevätlukukautena (1.1.-31.7.) suoritettujen opintopisteiden määrä (opek)
- Edellisen lukuvuoden loppuun (31.7.) mennessä suoritettujen kaikkien opintopisteiden määrä (opker)

Mikäli valittu opiskeluoikeus on jatkotutkintoon johtava (opiskeluoikeuden tyyppi = 5, 6 tai 7), opintopisteinä ilmoitetaan 0:aa.

Opintopistekertymiin valitaan lehtitason (ei sisällä toisia suorituksia) muita opintosuorituksia (laji=2), ja mukaan otetaan myös hyväksiluetut opintosuoritukset sillä poikkeuksella että hyväksilukeminen (päivämäärä) tulee olla tehty viimeistään tilannepäivänä, joiden suorituspäivämäärä kuuluu ko. tietueen aikaväliin:

- Edellinen syksy on 1.8. - 31.12. vuodelta tilastointivuosi-1
- Edellinen kevät on 1.1. - 31.7. tilastointivuodelta
- Kertymä 31.7. tilastointivuodelta mennessä

Jos kertymään ei tullut mitään arvoa, kertymäksi asetetaan 0.

SQL

```
---
-- Opinnot, suoritukset, opintopisteet
---

-- Opintopisteitä varten tehdä oma erillinen aputaulu
-- Temp-taulu tietojen valintaa varten
-- NB! param-taulussa saa olla vain yksi rivi!
IF OBJECT_ID('tempdb..#pisteet') IS NOT NULL
BEGIN
    truncate table #pisteet
    drop table #pisteet
    print convert(varchar,getdate(),120)+' DROP temp-taulu pisteet'
END
;
create table #pisteet (
    opiskelija_id int not null,
    opiskeluoikeus_id int,
    opintosuoritus_id int not null,
    laajuus decimal(18,6) not null,
    suoritus_pvm date not null,
    hyväksiluettu_pvm date,
    es bit,
    ek bit,
    ker bit
)
print convert(varchar,getdate(),120)+' CREATE temp-taulu pisteet'
;
insert into #pisteet (opiskelija_id,opiskeluoikeus_id,opintosuoritus_id,
    laajuus,suoritus_pvm,hyväksiluettu_pvm,es,ek,ker)
    select
        opiskelija_id = s.OpiskelijaID
        ,opiskeluoikeus_id = s.OpiskeluoikeusID
```

```

,opintosuoritus_id = s.ID
,laajuus = s.Laajuus
,suoritus_pvm = s.Suorituspaivamaara
,hyvaksiluettu_pvm = shl.Hyvaksilukupaivamaara
,es = case when s.Suorituspaivamaara
    between cast(cast(#param.vuosi as int)-1 as varchar)+'-08-01' and
cast(cast(#param.vuosi as int)-1 as varchar)+'-12-31'
    then 1 else 0 end
,ek = case when s.Suorituspaivamaara between #param.vuosi+'-01-01' and
#param.vuosi+'-07-31'
    then 1 else 0 end
,ker = case when s.Suorituspaivamaara <= #param.vuosi+'-07-31'
    then 1 else 0 end
from virta.dbo.Opintosuoritus s
inner join virta.dbo.Opintosuorituksen_laji sl on sl.ID=s.
Opintosuorituksen_lajiID and sl.Koodi='2' --muu opintosuoritus
-- HYVÄKSILUETTU
left join virta.dbo.Opsuorituksen_hyvaksiluku shl on shl.
OpintosuoritusID=s.ID
cross join #param -- saa olla vain yksi rivi!
where 1=1
-- lehtitaso, tai ei ainakaan vanhempi
--and s.ID not in (select ss.OpintosuoritusID from virta.dbo.
Op_suor_sisaltyvyys ss)
-- Muutos 2017-09-16
AND s.ID NOT IN (

    SELECT      ss.OpintosuoritusID
    FROM        virta.dbo.Op_suor_sisaltyvyys ss

    -- Sisältyvydet Lajeja 3 ja 4 ei huomioida
    WHERE
        ss.OpintosuoritusID2 NOT IN (

            SELECT sss.ID
            FROM virta.dbo.Opintosuoritus sss

                INNER JOIN virta.dbo.Opintosuorituksen_laji sssl ON sssl.
ID = sss.Opintosuorituksen_lajiID

                WHERE      sssl.Koodi in ('3','4')

        )

    )

-- rajoita perusjoukkoon
and s.OpiskelijaID in (select T.opiskelija_id from #tkopisk T)
;
CREATE NONCLUSTERED INDEX IX_pisteet_opiskelija_id ON #pisteet
(opiskelija_id)
INCLUDE (opiskeluoikeus_id,opintosuoritus_id,laajuus,suoritus_pvm)
;

```

```

CREATE NONCLUSTERED INDEX IX_pisteet_suoritus_pvm ON #pisteet
(suoritus_pvm)
INCLUDE ([opiskelija_id],[opiskeluoikeus_id],[laajuus])
;
CREATE NONCLUSTERED INDEX IX_pisteet_hyvaksiluettu_pvm ON #pisteet
(hyvaksiluettu_pvm)
INCLUDE ([opiskelija_id],[opiskeluoikeus_id],[laajuus],[suoritus_pvm])
;
--select * from #pisteet
--
-- "opintopistekysymykset eivät koske jatkotutkintoja opiskelevia, joille
opintopistetietoja ei merkitä"
-- jatko-oikeus: merkitään tyhjää/'0':
update T
set opes='0'
    , opek='0'
    , opker='0'
from #tkopisk T
where T.opiskeluoikeus_id IN (
    select o.id from Opiskeluoikeus o
    inner join Opiskeluoikeuden_tyyppi ot on ot.ID=o.
Op_oikeuden_tyyppiID
    and ot.Koodi in (
        '5', --Lääkärien erikoistumiskoulutus
        '6', --Lisensiaatintutkinto
        '7'  --Tohtorintutkinto
    )
    where o.OpiskelijaID=T.opiskelija_id --sido,nopeuta
)
;
update T
set opes=(
    select cast(cast(round(sum(
        case when s.es = 1 then s.Laajuus else 0 end),0) as int) as
varchar)
    from #pisteet s
    where s.opiskelija_id=T.opiskelija_id
    --and s.opiskeluoikeus_id=T.opiskeluoikeus_id
    group by s.opiskelija_id
)
from #tkopisk T
where opes is null
;
update T
set opek=(
    select cast(cast(round(sum(
        case when s.ek = 1 then s.Laajuus else 0 end),0) as int) as
varchar)
    from #pisteet s
    where s.opiskelija_id=T.opiskelija_id
    --and s.opiskeluoikeus_id=T.opiskeluoikeus_id
    group by s.opiskelija_id
)
)

```

```

from #tkopisk T
where opek is null
;
update T
set opker=(
    select cast(cast(round(sum(
        case when s.ker = 1 then s.Laajuus else 0 end),0) as int) as
varchar)
    from #pisteet s
    where s.opiskelija_id=T.opiskelija_id
    --and s.opiskeluoikeus_id=T.opiskeluoikeus_id
    group by s.opiskelija_id
)
from #tkopisk T
where opker is null
;

update T
set opes=''0''
from #tkopisk T
where opes is null
;
update T
set opek=''0''
from #tkopisk T
where opek is null
;
update T
set opker=''0''
from #tkopisk T
where opker is null
;

print convert(varchar,getdate(),120)+' opintosuoritustiedot'
-- 2017-10-13 Jos Kirtu tyhjä opiskeluoikeuden alkamispaivamaara
-- kirtu
UPDATE T
SET kirtu=(cast(year(o.Alkamispaivamaara) as varchar)
    +(case
        -- 8-12 (elo-joulu): syksy
        when month(o.Alkamispaivamaara)>=8 then '2'
        -- 1-7 (tammi-heinä): kevät
        else '1'
        end)
)
FROM #tkopisk T
inner join virta.dbo.Opiskeluoikeus o on o.ID=T.opiskeluoikeus_id
WHERE kirtu IS NULL
AND T.koulu IN ('775101','775201','775999')
;

```

```
-- 2018-10-16
-- Tämä korjaa TAY case kun erikoislääkäarin oikeus alkaa syksyllä, aiempi
läsnäolo 1,2 kevällä eri opiskeluoikeus
update T
set olotamm='0'
from #tkopisk T
where kirtu=(select vuosi from #param)+'2'
;
```

Tulos

Lopullisessa tuloksessa muokataan vielä etunollat kohdilleen, jolloin itse asiassa moni tyhjä merkkijono muuttuu siis myös nolla-arvoksi.

SQL

```
-- CSV
select
  ISNULL(kk, '') as "Yliopisto" -- (kk) (positiot 1-2)
, ISNULL(ht, '') "Henkilötunnus" -- (ht) (positiot 3-13)
, ISNULL(fuksi, '') "Uusi opiskelija" -- (fuksi) (positio 14)
, ISNULL(olotamm, '') "Ilmoittautumislaji keväällä" -- (olotamm) (positio
15)
, ISNULL(olosyys, '') "Ilmoittautumislaji syksyllä" -- (olosyys) (positio
16)
, ISNULL(kirtu, '') "Kirjoihintulo yliopistoon" -- (kirtu) (positiot 17-
21)
, ISNULL(sv, '') "Aineenopettajan pedagogiset opinnot" -- (sv)
(positiot 35-37)
, ISNULL(om, '') "Opiskelumuoto" -- (om) (positio 40)
, ISNULL(kkun, '') "Opiskelukunta" -- (kkun) (positiot 41-43)
, ISNULL(sp, '') "Sukupuoli" -- (sp) (positio 44)
, ISNULL(kansaly, '') "Kansalaisuus" -- (kansaly) (positiot 45-47)
, ISNULL(aikieliy, '') "Äidinkieli" -- (aikieliy) (positio 48)
, ISNULL(askuntay, '') "Asuinkunta" -- (askuntay) (positiot 49-51)
, ISNULL(snimi, '') "Sukunimi" -- (snimi) (positiot 52-81)
, ISNULL(enimi, '') "Etunimet" -- (enimi) (positiot 82-96)
, ISNULL(avo, '') "Hyväksytty avoimen yliopiston opintojen
perusteella" -- (avo) (positio 97)
, ISNULL(opes, '') "Edellisenä syyslukukautena (1.8.-31.12.)
suoritettujen opintopisteiden määrä" -- opes (positiot 98-100)
, ISNULL(opek, '') "Edellisenä kevätlukukautena (1.1.-31.7.)
suoritettujen opintopisteiden määrä" -- opek (positiot 101-103)
, ISNULL(opker, '') "Edellisen lukuvuoden loppuun (31.7.) mennessä
suoritettujen kaikkien opintopisteiden määrä" -- opker (positiot 104-106)
, ISNULL(lkmmmt, '') "Lukukausimaksukokeilu ja maksullinen
tilauskoulutus" -- (lkmmmt) (positio 107)
, ISNULL(opoik, '') "Opiskeluoikeuden alkaminen nykyisessä tutkinnossa"
-- (opoik) (positiot 108-112)
, ISNULL(koulk, '') "Koulutuskoodi" -- (koulk) (positiot 113-118)
-- korkeakoulujen pyynnöstä, lisäkentät
, T.opiskelijaavain "Opiskelija-avain"
, T.opiskeluoikeusavain "Opiskeluoikeusavain"
from #tkopisk T
where T.opiskeluoikeus_id is not null
order by ht
;
```

Tarkistus

Suoritetaan tarkistuksia poimituihin tietoihin liittyen. Pohjaksi on otettu Tilastokeskuksen aiemmin tekemän tarkistusohjelman kuvaus.

NB! Tarkistuksia tulee jalostaa vielä. Nyt on kuitenkin jotakin pohjaa. Lisäksi tarkistuksia haluttaneen lisää. Dokumentointia ei vielä ole yritettykään tehdä kattavaksi, sillä niin paljon tarkennusta vaativia kohtia on nyt.

SQL

```
---
-- TARKISTUS
---
--!!!!!!! KOPIOITU AMK-SEKTORILTA !!!!!!--
/*
  Tiedonkeruuaineiston tarkistusohjelman suorittamat tarkistukset

  Amk-opiskelijatiedonkeruun aikana 20.9.-30.9. TK:n palvelimella toimii
  automaattinen
  tarkistusohjelma, joka tarkistaa aineistot yleisimpien virheiden osalta
  ja toimittaa tarkistuksesta
  palautetulosteen amk:n yhdyshenkilölle. Seuraavassa on kuvattu ohjelman
  tekemät tarkistukset ja
  niiden nimet palautetulosteessa. Kaikki tulosteessa mainitut kohdat
  eivät ole välttämättä virheitä,
  mutta ne tulevat listalle mahdollisina virheellisinä tietoina. Sellaisia
  ovat mm. epäilyttävän suuret
  opintopistekertymät tai puutteelliset henkilötunnukset.

  TUNN = tunn (oppilaitostunnus) on virheellinen
  onko oppilaitostunnus jokin voimassaolevista 5-numeroisista tunnuksista

  AVAIN = koulk (koulutuskoodi) on virheellinen
  Verrataan koulutusluokituksen koulutuskodeihin

  AVAIN2 = oltunn - koulk (ammattikorkeakoulu - koulutuskoodi) -
  koodikombinaatio on virheellinen
  Ammattikorkeakoulu - koulutuskoodi -kombinaatiota verrataan vuoden 2013
  kombinaatioihin

  AVAIN3 = koultyp on 3 tai 5 ja koulutusohjelma puuttuu

  KOULTYP = koulutustyyppi on virheellinen

  KOULTYP1 = muuttujan arvo ei ole 1, 2, 3, 5 eikä 6

  KOULTYP2 = jos koultyp on 1, 2 tai 6, niin koulutuskoodi ei saa olla
  999999

  KOULTYP3= jos koultyp on 3 tai 5, niin koulutuskoodin täytyy olla 999999
  ja koulutusohjelman tulee
  olla sallittu

  KKIELI = kkieli (koulutuksen opetuskieli) -koodiarvo on virheellinen
  koulutuksen opetuskieli suomi (fi), ruotsi (sv), englanti (en) tai saksa
  (de)

  KKUN = kkun (koulutuksen sijaintikunta) -koodiarvo on virheellinen
  Verrataan toimilupapäätöksiin, onko amk:lla lupa järjestää koulutusta
  kyseisessä kunnassa.
```


Ammattikorkeakoulun koulutustehtävässä määrättävä sijaintipaikka koskee ammattikorkeakoulututkintoon johtavaa nuorten koulutuksena järjestettävää koulutusta ja ylempään ammattikorkeakoulututkintoon johtavaa koulutusta.

KKUN2 = kkun2 (kunta) -koodiarvo on virheellinen
Verrataan voimassa olevaan kuntaluokitukseen (koulutustyyppi 2, 3 ja 5).

KIRTUPV = kirjoihintuloajankohta on virheellinen

KIRTUPV2 = kirjoihintulovuosi ei saa olla suurempi kuin tilastovuosi, kirjoihintulokuukausi saa arvon 1-12, päivä arvon 1-31

OLOTAMM = olotamm (kevään kirjoillaolo) on virheellinen

OLOTAMM1 = kevään läsnäolotieto voi olla 0, 1 tai 2

OLOTAMM2 = jos tilastovuosi=kirjoihintulovuosi ja kirjoihintulolukukausi on syksy, kevään läsnäolotiedon on oltava 0

OLOTAMM3 = kevään läsnäolotieto voi olla amk:n vanhoilla opiskelijoilla joko 1 tai 2

OLOTAMM4 = kirjoihintulolukukausi on kevät, mutta kevään läsnäolotieto on 0 (olotamm=0)

OLOSYYSS = olosyys (läsnäolo 20.9.) on virheellinen
syksyn läsnäolotieto voi olla 0, 1 tai 2

TU = sama opiskelija on aineistossa kahteen kertaan

TU1 = tarkistetaan onko joillakin havainnoilla samat arvot joka muuttujalla - täydellinen tuplatietue

TU2 = tarkistetaan onko joillakin havainnoilla samat oppilaitostunnus-koulutustyyppi-koulutuskoodi - kombinaatiot

OPES = ed. syyslukukaudella suoritettut opintopisteet on virheellinen lukukauden aikana suoritettu yli 99 opintopistettä

OPEK = ed. kevätlukukaudella suoritettut opintopisteet on virheellinen lukukauden aikana suoritettu yli 99 opintopistettä

OPKER = opker (ed. kevätlukukauden loppuun mennessä kertyneet opintopisteet) on virheellinen
opintopistekertymä yli 300 tai ed. kevät- ja syyslukukausien opintopisteet yhteensä ovat enemmän
kuin kertymä plus yksi (=pyöristykset voivat nostaa summaa yhdellä).

HT = henkilötunnus (tai sukupuoli) on virheellinen

HT1 = Puutteelliset henkilötunnukset - loppuosa puuttuu

HT2 = henkilötunnukset, joiden jakojäännös ei täsmää - loppuosa virheellinen

HT3 = sukupuolitieto virheellinen

HT4 = Hetu-välimerkki jos 2000-luvulla syntynyt ei '-' vaan 'A'

AIKIELI = äidinkielitieto virheellinen
verrataan äidinkieltä sallittujen kielten listaan

ASKUN = asuinkuntatieto virheellinen
verrataan kuntatietoa sallittujen kuntien listaan

KANSAL = kansalaisuustieto virheellinen
verrataan kansallisuustietoa sallittujen kansallisuuksien listaan

LASNALK = läsnäolo- tai poissaololukukausissa virhe

LASNALK1 = (läsnäololukukaudet + poissaololukukaudet) <= (3 - 1.
kirjoihintulolukukausi) +
(tilastovuosi - 1.kirjoihintulovuosi) * 2
Esim. jos opiskelija on tullut kirjoille 1. kerran syyslukukaudella
2010, ja tilastovuosi on 2012:
läsnä+poissa yhteensä voi olla maksimissaan: (3 - 2) + (2012-2010) * 2 =
5

LASNALK2 = läsnä- tai poissaolokausia täytyy olla vähintään 1.

OPLAAJ = opintojen laajuus virheellinen

OPLAAJ1 = muodollinen virhe (voi olla 1 - 300)

KIRTU1PV = Ensimmäinen kirjoihintuloajankohta ko. tutkintoon virheellinen

KIRTU1PV = kirjoihintulokuukausi voi saada arvon 1-12, päivä arvon 1-31

KIRTU1P2 = ei voi olla suurempi kuin kirjoihintulo tähän tutkintoon
tässä amk:ssa

OPETTAJA = Lastentarhanpettajan koulutus muodoltaan virheellinen

OPETT1 = voi olla 1, 2, tai tyhjä.

OPETT2 = opettajankoulutus ei ole sosionomi-tutkinnolla

OPETT3 = opettajatieto syksyn uudella opiskelijalla

RAHLAHDE = Rahoituslähde muodoltaan virheellinen

RAHLAHDE= voi olla 1, 2, 3, 4 tai 5.

RAHLAHD4= rahoituslähde on 4 ja maksullisen tilauskoulutuksen sijaintimaatieto puuttuu

EIMUKAAN = Opiskelijalla ei ilmoittautumistietoa 20.9. eikä ole aloittanut
vuonna 2014
Ei kuulu perusjoukkoon

TILMAA = Maksullisen tilauskoulutuksen sijaintimaatieto virheellinen
Verrataan sijaintimaatietoa sallittujen maiden listaan

OHJAUKSENALA= voi olla 1 - 12 jos koulutuskoodi 999999

```
*/
--/*
IF OBJECT_ID('tempdb..#tarkistus') IS NOT NULL
BEGIN
    truncate table #tarkistus
    drop table #tarkistus
    print convert(varchar,getdate(),120)+' DROP temp-taulu'
END
;
create table #tarkistus (
    tunn varchar(5),
    avain varchar(20),
    ---koultyp varchar(4),
    --koultyp1 varchar(1),
    --koultyp2 varchar(8),
    kkieli varchar(2),
    kkun varchar(10), --muutettu
    kkun2 varchar(3),
    kirtupv varchar(10), --muutettu
    kirtupv2 varchar(8),
    olotamm varchar(10), --muutettu
    olotamm1 varchar(8),
    olotamm2 varchar(8),
    olotamm3 varchar(8),
    olotamm4 varchar(8),
    olosyys varchar(10), --muutettu(lisäys)
    ---tu varchar(1),
    tu1 varchar(1),
    tu2 varchar(1),
    opes varchar(4),
    opek varchar(4),
    opker varchar(4),
    ---ht varchar(11),
    ht1 varchar(11),
    ht2 varchar(11),
    ht3 varchar(11),
    ht4 varchar(11),
    aikieli varchar(2),
```

```

askun varchar(3),
kansal varchar(3),
---lasnalk varchar(4),
--lasnalk1 varchar(10),
--lasnalk2 varchar(10),
---oplaaj varchar(4),
--oplaaj1 varchar(4),
--kirtulpv varchar(8),
--kirtulp2 varchar(8),
---opettaja varchar(1),
--opett1 varchar(2),
--opett2 varchar(8),
--opett3 varchar(10),
--rahlahde varchar(1),
--
opoik varchar(10), --lisäys
ohjauksenAla varchar(2),

--
opiskelijaavain varchar(100),
opiskeluoikeusavain varchar(100),
opiskelija_id int,
opiskeluoikeus_id int,
db varchar(8),
)
print convert(varchar,getdate(),120)+' CREATE temp-taulu #tarkistus'
;

-- truncate table #tarkistus
insert into #tarkistus
select
  tunn = '--(case when kk not in (select koodi from virta..Organisaatio)
then kk else '' end)
,avain = (case
  when koulk IS NULL AND opiskeluoikeusavain IS NOT NULL then
'puuttuu'
  when koulk not in (select koodi from virta.dbo.Tutkintonimike)
then koulk
-- lisätty tarkistus että tutkintonimike alkaa numerolla 6,7,8 tai 9
  when cast(substring(koulk,1,1) as int) not between 6 and 9
then koulk

  when koulk LIKE '%00' then koulk + '_00 loppuinen'

  else ''
end

)
--,avain2 = --amk-koulk kombinaatiot?
--,avain3 = --koulutusohjelma?
--,koultyp = --otsikko?
--,koultyp1 = (case when koultyp not in ('1','2','3','5','6') then koultyp
else '' end)

```

```

--,koulytyp2 = (case when koulytyp in ('1','2','6') and koulyk='999999' then
koulytyp+' '+koulyk else '' end)
--,koulytyp3 = --koulytususohjelma?
,kkieli = '--(case when kiel not in ('fi','sv','en','de') then kiel else
'' end)
,kkun = (case when coalesce(kkun,'')='' then '#puuttuu#' else '' end)
,kkun2 = (case
  when l=1 --koulytyp in ('2','3','5')
  and kkun not in (select koodi from virta..Kunta cross join #param where
#param.tilpvm between coalesce(Alkamispaivamaara,'1900-1-1') and coalesce
(Paattymispaivamaara,'9999-1-1'))
  then kkun else '' end)
,kirtupv = (case when coalesce(kirtu,'')='' then '#puuttuu#' else '' end)
,kirtupv2 = (case
  when cast(substring(kirtu,1,4) as int) > (select year(tilpvm) from
#param) then kirtu
--  when cast(substring(kirtu,5,2) as int) not between 1 and 12 then kirtu
--  when cast(substring(kirtu,7,2) as int) not between 1 and 31 then kirtu
  when l=1 then ''
  else '' end)
,olotamm = (case when coalesce(olotamm,'')='' then '#puuttuu#' else '' end)
,olotamm1 = (case when olotamm not in ('0','1','2') then olotamm else ''
end)
,olotamm2 = (case
  when cast(substring(kirtu,1,4) as int) = (select year(tilpvm) from
#param)
  --and cast(substring(kirtu,5,2) as int) >= 8
  and cast(substring(kirtu,5,1) as int) >= 2
  and olotamm<>'0' then kirtu+' '+olotamm
  else '' end)
,olotamm3 = (case
  when(
    (cast(substring(kirtu,1,4) as int) < (select year(tilpvm) from
#param))
    --or (cast(substring(kirtu,1,4) as int) = (select year(tilpvm) from
#param) and cast(substring(kirtu,5,2) as int) < 8))
    or (cast(substring(kirtu,1,4) as int) = (select year(tilpvm) from
#param) and cast(substring(kirtu,5,1) as int) < 2))
    and olotamm not in ('0','1','2') then kirtu+' '+olotamm
  else '' end)
,olotamm4 = (case
  --when (cast(substring(kirtu,1,4) as int) = (select year(tilpvm) from
#param) and cast(substring(kirtu,5,2) as int) < 8)
  when (cast(substring(kirtu,1,4) as int) = (select year(tilpvm) from
#param) and cast(substring(kirtu,5,1) as int) < 2)
  and olotamm = '0' then kirtu+' '+olotamm
  else '' end)
,olosyys = (case
  when coalesce(olosyys,'')='' then '#puuttuu#'
  when olosyys not in ('0','1','2') then olosyys
  else '' end)
--,tu = --otsikko?
,tul = (case

```

```

when exists (
  select 1 from #tkopisk t2 where t2.opiskelijaavain=#tkopisk.
opiskelijaavain and t2.kk = #tkopisk.kk
  group by kk, --koultyp, kiel,
  kkun, ht, snimi, enimi, sp, aikieliy, askuntay, kansaly, kirtu,
olotamm, olosyys
  , opes, opek, opker, --lasnalk, poissalk, --laajuus, kirtulpv, lastu,
  --rahlahde,
  koulk
  having count(*)>1
) then '1'
else '' end)
,tu2 = (case
  when exists (
    select 1 from #tkopisk t2 where t2.opiskelijaavain=#tkopisk.
opiskelijaavain and t2.kk = #tkopisk.kk
    group by kk, --koultyp,
    koulk, ht, snimi, enimi, sp
    having count(*)>1
  ) then '1'
  else '' end)
,opes = (case when cast(opes as int) > 300 then opes else '' end)
,opek = (case when cast(opek as int) > 300 then opek else '' end)
,opker = (case when cast(opker as int) > 500 then opker else '' end)
--,ht = --otsikko?
,ht1 = (case when len(ht)<11 then ht else '' end)
,ht2 = (case
  when len(ht)=11
  and substring(ht,11,1) <>
    case when cast(SUBSTRING(ht,1,6) + SUBSTRING(ht,8,3) as int) % 31 = 0
then '0'
    else SUBSTRING('123456789ABCDEFHJKLMNPRSTUVWXY',cast(SUBSTRING(ht,1,6)
+ SUBSTRING(ht,8,3) as int) % 31,1) end
  then ht
  else '' end)
,ht3 = (case
  when len(ht)=11 and sp='1' and cast(SUBSTRING(ht,10,1) as int) % 2 = 0
then ht
  when len(ht)=11 and sp='2' and cast(SUBSTRING(ht,10,1) as int) % 2 = 1
then ht
  else '' end)
-- 2000 luvulla syntyneet välimerkki, tulisi olla A
,ht4 = (CASE
  WHEN LEN(ht)<11 AND SUBSTRING(ht,7,1) = '-' AND (CAST(SUBSTRING(ht,
5,2) AS int) < 10) THEN ht
  ELSE ''
  END)
,aikieli = (case
-- when aikieliy not in (select koodi from virta..Kieli cross join #param
where #param.tilpvm between coalesce(Alkamispaivamaara,'1900-1-1') and
coalesce(Paattymispaivamaara,'9999-1-1'))
-- then aikieliy
  when 1=1 then ''

```

```

    else '' end)
,askun = (case
    when askuntay not in (select koodi from virta..Kunta cross join #param
where #param.tilpvm between coalesce(Alkamispaivamaara,'1900-1-1') and
coalesce(Paattymispaivamaara,'9999-1-1'))
    then askuntay
    else '' end)
,kansal = (case
    when kansaly not in (select koodi from virta..Maa cross join #param
where #param.tilpvm between coalesce(Alkamispaivamaara,'1900-1-1') and
coalesce(Paattymispaivamaara,'9999-1-1'))
    then kansaly
    else '' end)
--,lasnalk = --otsikko?
--,lasnalk1 = (case
-- when cast(lasnalk as int)+cast(poissalk as int)
-- > ((3 - (case when cast(substring(kirtulpv,5,2) as int)<8 then 1
else 2 end))
-- + (((select year(tilpvm) from #param) - cast(substring(kirtulpv,
1,4) as int)) * 2))
-- then lasnalk+''+poissalk+'<'+
-- cast((3 - (case when cast(substring(kirtulpv,5,2) as int)<8 then 1
else 2 end))
-- + (((select year(tilpvm) from #param) - cast(substring(kirtulpv,
1,4) as int)) * 2) as varchar)
-- else '' end)
--,lasnalk2 = (case
-- when cast(lasnalk as int)+cast(poissalk as int) < 1 then
lasnalk+''+poissalk+'<1'
-- else '' end)
--,oplaaj = --otsikko?
--,oplaaj1 = (case when cast(laajuuus as int) > 300 then laajuuus else ''
end)
--,kirtulpv = (case
-- when cast(substring(kirtulpv,5,2) as int) not between 1 and 12 then
kirtulpv
-- when cast(substring(kirtulpv,7,2) as int) not between 1 and 31 then
kirtulpv
-- else '' end)
--,kirtulpv2 = (case when cast(kirtulpv as int) > cast(kirtupv as int) then
kirtulpv else '' end)
--,opettaja = --otsikko?
--,opett1 = (case when coalesce(lastu,'') not in ('','1','2','3','4','5')
then lastu else '' end)
--,opett2 = (case when lastu in ('1','3','4','5') and koulk in
('671201','771301') then koulk+''+lastu else '' end)
--,opett3 = (case
-- when lastu in ('1','3','4','5')
-- and substring(kirtupv,1,4) = (select vuosi from #param) and cast
(substring(kirtupv,5,2) as int) >= 8
-- then kirtupv+''+lastu
-- else '' end)
--,rahlahde = (case when rahlahde not in ('1','2','3','4','5') then

```

```

rahlahde else '' end)
--
,opoik = (case when coalesce(opoik,'')='' then '#puuttuu#' else '' end)
--,ohjauksenAla = (CASE WHEN koulu='999999' AND ohjauksenAla NOT IN
('1','2','3','4','5','6','7','8','9','10','11','12') THEN ohjauksenAla +
'+ohjauksenAla ELSE '' END)
,ohjauksenAla = (CASE WHEN koulu='999999' AND ohjauksenAla NOT IN
('1','2','3','4','5','6','7','8','9','10','11','12') THEN ohjauksenAla +
'+ohjauksenAla
                                WHEN koulu='999999' AND ohjauksenAla IS NULL THEN
'99' ELSE '' END)
--
,opiskelijaavain
,opiskeluoikeusavain
,opiskelija_id
,opiskeluoikeus_id
,db = (select Asetus from Yhteiset.Asetukset.Ohjaus_Parametrit where
Toiminto='YO_tunnus' and Arvo=kk)
from #tkopisk
;

/*
select * from #tarkistus
where '<>
tunn+avain+
kkieli+kkun+kkun2+kirtupv+kirtupv2+
olotamm+olotamm1+olotamm2+olotamm3+olotamm4+
olosyys+tul+tu2+
opes+opek+opker+
ht1+ht2+ht3+
aikieli+askun+kansal+
opoik
;
--*/
--select * from #tkopisk where opiskeluoikeusavain='108782275'

-- transponoi
delete from TK_tarkistus where tiedonkeruu='tk_opisk_yo';
insert into TK_tarkistus (virhekoodi,virhe,arvo,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,tiedonkeruu)
select 'tunn' as VIRHEKOODI, 'oppilaitostunnus on virheellinen' as VIRHE,
tunn AS ARVO,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>tunn
union select 'avain', 'koulutuskoodi on virheellinen tai puuttuu',avain,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where '<>avain
union select 'kkieli', 'koulutuksen opetuskieli -koodiarvo on
virheellinen',kkieli,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>kkieli
union select 'kkun', 'koulutuksen sijaintikunta -koodiarvo on
virheellinen',kkun,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>kkun
union select 'kkun2', 'kunta-koodiarvo on virheellinen ( koulutustyyppi 2,

```



```
3 ja 5 )',kkun2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>kkun2
union select 'kirtupv', 'kirjoihintuloajankohta on virheellinen',kirtupv,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where ''<>kirtupv
union select 'kirtupv2', 'kirjoihintulovuosi ei saa olla suurempi kuin
tilastovuosi, kirjoihintulokuukausi ei ole 1-12 tai päivä ei ole 1-31',
kirtupv2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>kirtupv2
union select 'olotamm', 'kevään kirjoillaolo on virheellinen',olotamm,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where ''<>olotamm
union select 'olotamm1', 'kevään läsnäolotieto voi olla 0, 1 tai 2',
olotamm1,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>olotamm1
union select 'olotamm2', 'jos tilastovuosi=kirjoihintulovuosi ja
kirjoihintulolukukausi on syksy, kevään läsnäolotiedon on oltava 0',
olotamm2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>olotamm2
union select 'olotamm3', 'kevään läsnäolotieto voi olla vanhoilla
opiskelijoilla joko 1 tai 2',olotamm3,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
''<>olotamm3
union select 'olotamm4', 'kirjoihintulolukukausi on kevät, mutta kevään
läsnäolotieto on 0',olotamm4,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
''<>olotamm4
union select 'olosyys', 'olosyys (läsnäolo 20.9.) on virheellinen. syksyn
läsnäolotieto voi olla 0, 1 tai 2',olosyys,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where ''<>olosyys
union select 'tul', 'joillakin havainnoilla samat arvot joka muuttujalla -
täydellinen tuplatietue',tul,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
''<>tul
union select 'tu2', 'sama oppilaitostunnus-koulutustyyppi-koulutuskoodi -
kombinaatio useita kertoja',tu2,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
''<>tu2
union select 'opes', 'ed. syyslukukaudella suoritettu liikaa
opintopisteitä',opes,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>opes
union select 'opek', 'ed. kevätlukukaudella suoritettu liikaa
opintopisteitä',opek,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where ''<>opek
union select 'opker', 'opintopistekertymä liian suuri tai ed. kevät- ja
syyslukukausien opintopisteet yhteensä ovat enemmän kuin kertymä plus yksi
(=pyöristykset voivat nostaa summaa yhdellä).',opker,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where ''<>opker
union select 'ht1', 'Puutteellinen henkilötunnus - loppuosa puuttuu',ht1,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where ''<>ht1
```

```

union select 'ht2', 'henkilötunnus, jonka jakojäännös ei täsmää - loppuosa
virheellinen',ht2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>ht2
union select 'ht3', 'sukupuolitieto virheellinen',ht3,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where '<>ht3
union select 'ht4', 'Väärä välimerkki lopussa. Jos 2000 luvulla syntynyt
tulee olla A ei -',ht4,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>ht4
union select 'aikieli', 'äidinkielitieto virheellinen',aikieli,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where '<>aikieli
union select 'askun', 'asuinkuntatieto virheellinen',askun,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where '<>askun
union select 'kansal', 'kansalaisuustieto virheellinen',kansal,
opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where '<>kansal
---union select 'lasnalk1', '(länäololukukaudet + poissaololukukaudet) <=
(3 - 1.kirjoihintulolukukausi) + (tilastovuosi - 1.kirjoihintulovuosi) *
2',lasnalk1,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>lasnalk1
---union select 'lasnalk2', 'länä- tai poissaolokausia täytyy olla
vähintään 1.',lasnalk2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>lasnalk2
---union select 'oplaaj1', 'Opintojen laajuus virheellinen; muodollinen
virhe (voi olla 1 - 300)',oplaaj1,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
'<>oplaaj1
---union select 'kirtulpv', 'Ensimmäinen kirjoihintuloajankohta ko.
tutkintoon virheellinen; kirjoihintulokuukausi voi saada arvon 1-12, päivä
arvon 1-31',kirtulpv,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,
opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where '<>kirtulpv
---union select 'kirtulp2', 'Ensimmäinen kirjoihintuloajankohta ko.
tutkintoon virheellinen; ei voi olla suurempi kuin kirjoihintulo tähän
tutkintoon tässä amk:ssa',kirtulp2,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
'<>kirtulp2
---union select 'opett1', 'Lastentarhanpettajan koulutus muodoltaan
virheellinen; voi olla 1, 2, 3, 4, 5 tai tyhjä.',opett1,opiskelijaavain,
opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where '<>opett1
---union select 'opett2', 'opettajankoulutus ei ole sosionomi-tutkinnolla',
opett2,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where '<>opett2
---union select 'opett3', 'opettajatieto syksyn uudella opiskelijalla',
opett3,opiskelijaavain,opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,
db,'tk_opisk_yo' from #tarkistus where '<>opett3
---union select 'rahlahde', 'Rahoituslähde muodoltaan virheellinen. Voi
olla 1, 2, 3, 4 tai 5.',rahlahde,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from #tarkistus where
'<>rahlahde
union select 'opoik', 'OPOIK puuttuu',opoik,opiskelijaavain,

```

```

opiskeluoikeusavain,opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' from
#tarkistus where ''<>opoik
union SELECT 'ohjauksenAla', 'jos koulutuskoodi 999999 ja koulutustyyppi
on 5, 6 tai 7 niin ohjauksenala on annettava koodiarvolla 1 - 12 ja se ei
voi olla tyhjä',ohjauksenAla,opiskelijaavain,opiskeluoikeusavain,
opiskelija_id,opiskeluoikeus_id,db,'tk_opisk_yo' FROM #tarkistus WHERE
''<>ohjauksenAla

;
print convert(varchar,getdate(),120)+' tarkistustiedot viety tauluun
TK_tarkistus'
;

```

Muutoshistoria

27.5.2019 :: HT4 tarkistuksen lisäys :: wiki-versio 31

- Lisätty tarkistus hetutomille, syntymäajan välimerkki, muodossa pppkvv-. Hetu-välimerkki jos henkilö 2000-luvulla syntynyt ei saa olla '-' vaan 'A', syntymäaika tulee olla muodossa pppkvvA

12.10.2018:: Ohjauksenala tyhjä ja tutkintokoodin tarkistus :: wiki-versio 30

Tarkistuskäsittelyyn lisätty avain muuttuun tutkintokoodin ensimmäisen numeron tarkastelu että alkaa numeroilla 6,7,8 tai 9 ja ohjauksenala päättelyyn koulk = '999999' niin ohjauksenala ei saa olla tyhjä.

13.10.2017:: Erikoislääkäri kirjoilletulo päivämäärä (kirtu-muuttuja). :: wiki-versio 29

Erikoislääkärit ('775101','775201','775999') kirjoilletulopvm opiskeluoikeuden alkamisvpm jos tyhjä, kirtu-muuttuja.

11.10.2017 ::olotamm3 käsittelyyn muutos tarkistuksessa :: wiki-versio 28

- lisätty olotamm3 tarkistukseen myös olotamm= 0 vertailu, jos kevät lukukausi-ilmoittautumista ei ole ja olotamm on 0.

5.10.2017 ::Tu1 ja tu2 käsittelyyn muutos tarkistuksessa :: wiki-versio 27

- tu1 ja tu2 käsittelyyn lisätty korkeakoulu päättely, koska samalla opiskelija-avaimella voi olla useita tuloksia eri korkeakouluista.

17.10.2016 :: Lisätty erikoistumiskoulutus ohjauksenala poiminta, poikkeuspoiminta ei ilmoittautuneille erikoistumisopiskelijoille.

Opiskeluluoituksen mukaa avo ja om muuttujat, avo= avoimen väylän kautta valitut, om=maisteri ohjelma, korvaa vanhan "muu" maisterin jota ei enään ole:: wikiversio 26

22.1.2016 :: Lisätty perusjoukkoon mukaan erikoistumiskoulutusta suorittavat :: wikiversio 25

30.09.2015 :: Päivitetty "Aineenopettajan pedagogiset opinnot (sv)" poimintaa, uudet pätevyyskoodit "ik", "il", "im" (entiset "1" ja "904") :: wikiversio 22

8.12.2014 :: Tulos CSV-muotoiseksi :: wikiversio 21

17.11.2014 :: Poistettu turhia sarakkeita ja lisätty avaintiedot :: wikiversio 19

- Poistetut sarakkeet
 - Tiedekunta/osasto TDK
 - Tutkinto TUTK
 - Koulutusohjelma KO
 - Pääaine PAINE
 - Opintoala OPALAY
- Lisätty avaintiedot
 - Opiskelija-avain
 - Opiskeluoikeusavain

