



Koppeling P8 met BRK

Documentatie API versie 1.0

Versie: 1.0

Datum: 21 maart 2016

Kenmerk:

Opdrachtgever: I-Real

Contactpersoon: Aarjan Langereis



1. INHOUDSOPGAVE

2. INLEIDING.....	2
2.1. VERSIEBEHEER	2
2.2. DISTRIBUTIE	2
2.3. DOCUMENTATIE	2
3. REST-SERVICE	3
4. REST-COLLECTIONS	3
4.1. /KADASTRALEPERCELEN.....	3
4.2. /KADASTRAALPERCEEL	4
4.3. /KADASTRALESUBJECTPERCELEN	7
4.4. /SUBJECTEN	8
5. DATA VIEWS.....	10
5.1. ONROERENDE ZAKEN.....	10
5.2. KADASTRALE PERCLEEN	10
5.3. RECHTEN	11
5.4. SUBJECTEN	11
5.5. AANTEKENINGEN.....	13
5.6. ONTSTAAN UIT/ OVERGEGAAN IN	14

2. INLEIDING

In versie 2.1 van P8 is de Kadaster-module toegevoegd. Deze module maakt het mogelijk om kadastrale data, welke is opgeslagen in een IMKAD-compatible database. De koppeling tussen P8 en de IMKAD-database gaat op basis van REST-services. De IMKAD-database maakt de data beschikbaar via deze REST-services en P8 koppelt vervolgens op deze services.

Het doel van dit document is om de REST-services, welke P8 nodig heeft voor deze koppeling, te beschrijven. Het document beschrijft zowel de collections als de views die deze collections van data voorzien.

Dit document is bedoeld voor applicatieontwikkelaar welke ervaring hebben met het opzetten van REST-services. De voorbeelden van de views zijn op basis van een Oracle database welke gevuld is door de BRK applicatie van Giskit.

2.1. VERSIEBEHEER

Datum	Versie	Auteur	Omschrijving
20-08-2015	0.1	Aarjan Langereis (Ala)	Initiële versie
21-3-2016	1.0	Aarjan Langereis (Ala)	-

2.2. DISTRIBUTIE

Naam	0.1	0.2	1.0
I-Real (intern)	✓		✓
P8 Klanten	✓		✓

2.3. DOCUMENTATIE

Naam	Omschrijving
Wiki pagina over REST	https://en.wikipedia.org/wiki/Representational_state_transfer
Voorbeelden database views	Views_IMKAD_tbv_API.sql
Automatisch gegenereerde documentatie van de API	API documentation.htm

3. REST-SERVICE

De data uit de IMKAD-compatible database wordt middels een REST-service aangeboden aan P8. Hierbij hoeft alleen de GET-request ondersteund te worden. PUT, POST, etc zullen door P8 niet gebruikt worden. De eerste implementatie van de P8 REST-client gaat er vanuit dat er geen gebruikersnaam/wachtwoord nodig is voor de service. Eventuele afscherming zal door middel van een firewall, waarin het IP-adres van de P8 server is vrijgegeven, moeten geschieden. In de roadmap is deze uitbreiding al wel opgenomen natuurlijk.

Digitale documentatie van de API zoals P8 deze verwacht is bijgevoegd bij deze handleiding.

4. REST-COLLECTIONS

Voor de koppeling tussen P8 en de BRK database zijn nu vier collections gedefinieerd. Mogelijk zullen dit er in de toekomst meer worden als behalve de BRK ook nog andere gegevens ontsloten worden. Hieronder een korte toelichting per collection. De volledige beschrijving is te vinden in de gegenereerde HTML-file welke als bijlage bij deze handleiding is gevoegd.

4.1. /KADASTRALEPERCELEN

Deze collection is bedoeld om een hele lijst aan percelen op te vragen (dit in tegenstelling tot /Kadastraalperceel). In deze collection staan alleen een aantal basis gegevens die op een overzicht getoond kunnen worden.

Parameter	Type	Omschrijving
kadastrale_percelen[]	array of objects (KadastraalPerceel)	
kadastrale_percelen[][kadastrale_code]	string	Volledige kadastrale code
kadastrale_percelen[][gemeente_code]	string	Gemeentecode
kadastrale_percelen[][sectie]	string	Sectie
kadastrale_percelen[][perceelnummer]	integer	Perceelnummer
kadastrale_percelen[][oppervlakte]	integer	Oppervlakte
kadastrale_percelen[][geom]	string	Geometrie in WKT

Betrokken database views:

- VIEW_API_KADASTRALE_PERCELEN

Voorbeeld antwoord:

```
{
  "kadastrale_percelen":[
    {
```

```

    "kadastrale_code":"ASN00AA1",
    "gemeente_code":"ASN00",
    "sectie":"AA",
    "perceelnummer":1,
    "oppervlakte":14476,
    "geom":"aa"
  },
  {
    "kadastrale_code":"ASN00AA2",
    "gemeente_code":"ASN00",
    "sectie":"AA",
    "perceelnummer":2,
    "oppervlakte":5660,
    "geom":"aa"
  },
  {
    "kadastrale_code":"ASN00AA3",
    "gemeente_code":"ASN00",
    "sectie":"AA",
    "perceelnummer":3,
    "oppervlakte":6980,
    "geom":"aa"
  },
],
"offset":"0",
"limit":"10",
"total_item_count":"26526"
}

```

4.2. /KADASTRAALPERCEEL

Om meer details van een enkel perceel te krijgen is een aparte collection gemaakt. P8 zal deze collection altijd aanroepen met een kadastrale code.

Parameter	Type	Omschrijving
gemeente_code	string	Gemeentecode
sectie	string	Sectie
perceelnummer	integer	Perceelnummer
oppervlakte	integer	Oppervlakte
adres	string	adres
postcode	string	postcode
woonplaats	string	woonplaats
gemeente	string	gemeente
cultuur	string	Cultuur
rechten[]	array of objects (KadastraalPerceelRecht)	Rechten
rechten[][subject]	object (KadastraalSimpleSubject)	Gekoppeld subject
rechten[][subject][naam]	string	Presentatie naam van het subject
rechten[][subject][type]	string	Type subject NatPers NietNatPers IGGbaPers NIGGbaPers
rechten[][subject][persoonid]	string	Uniek id van subject

rechten[][subject][woonplaats]	string	Woonplaats subject
rechten[][aandeel]	string	Aandeel
rechten[][recht_soort]	string	Recht soort
rechten[][datum_ingang]	DateTime	Datum ingang
rechten[][datum_eind]	DateTime	Datum ingang
aantekeningen[]	array of objects (KadastraalPerceelAante kening)	Aantekeningen
aantekeningen[][aantekening_op]	string	aantekening op
aantekeningen[][aantekening_aar d]	string	Aard van aantekening
aantekeningen[][aantekening_rec ht_type]	string	Recht type van een aantekening
aantekeningen[][einddatum]	DateTime	Eind datum
aantekeningen[][subject]	object (KadastraalSimpleSubjec t)	Subject
aantekeningen[][subject][naam]	string	Presentatie naam van het subject
aantekeningen[][subject][type]	string	Type subject NatPers NietNatPers IGGbaPers NI GGbaPers
aantekeningen[][subject][persoo nid]	string	Uniek id van subject
aantekeningen[][subject][woonpl aats]	string	Woonplaats subject
ontstaan_uit[]	array of objects (KadastraalPerceelMutati e)	Aantekeningen
ontstaan_uit[][gemeente_code]	string	Gemeentecode
ontstaan_uit[][sectie]	string	Sectie
ontstaan_uit[][perceelnummer]	integer	Perceelnummer
ontstaan_uit[][aard]	string	Sectie
ontstaan_uit[][begindatum]	DateTime	Begin datum
overgegaan_in[]	array of objects (KadastraalPerceelMutati e)	Aantekeningen

Betrokken database views:

- VIEW_API_KADASTRALE_PERCELEN
- VIEW_API_RECHTEN (+VIEW_API_SUBJECTEN)
- VIEW_API_AANTEKENINGEN (+VIEW_API_SUBJECTEN)
- VIEW_API_PERCEEL_ONTSTAAN_UIT (+VIEW_API_KADASTRALE_PERCELEN)
- VIEW_API_PERCEEL_OVERGEGAAN_IN (+VIEW_API_KADASTRALE_PERCELEN)

Voorbeeld antwoord:

```
{
  "gemeente_code": "ASN00",
```

```

"sectie":"T",
"perceelnummer":4425,
"oppervlakte":265,
"adres":"DE GAARDE 20",
"postcode":"9402LJ",
"woonplaats":"ASSEN",
"gemeente":"Assen",
"cultuur":"Wonen",
"rechten":[
  {
    "subject":{
      "naam":"Kuiper",
      "type":"IGGbaPers",
      "persoonid":"NL.KAD.Persoon.57598661",
      "woonplaats":"ASSEN"
    },
    "aandeel":"1\2",
    "recht_soort":"Eigendom (recht van)",
    "datum_ingang":"2009-10-31",
  },
  {
    "subject":{
      "naam":"Postma",
      "type":"IGGbaPers",
      "persoonid":"NL.KAD.Persoon.253295359",
      "woonplaats":"ASSEN"
    },
    "aandeel":"1\2",
    "recht_soort":"Eigendom (recht van)",
    "datum_ingang":"2009-10-31T00:00:00+0100",
  }
],
"ontstaan_uit":[
  {
    "gemeente_code":"ASN00",
    "sectie":"Q",
    "perceelnummer":4591,
    "aard":"Ver Nummering",
    "begindatum":"2009-10-31T00:00:00+0100"
  }
],
"overgegaan_in":[
  {
    "gemeente_code":"ASN00",
    "sectie":"Q",
    "perceelnummer":4591,
    "aard":"Ver Nummering",
    "begindatum":"2009-10-31T00:00:00+0100"
  }
]
}

```

4.3. /KADASTRALESUBJECTPERCELEN

Via deze collection vraagt P8 de percelen op waar een bepaald subject rechten op heeft.

Parameter	Type	Omschrijving
kadastrale_percelen[]	array of objects (KadastraalSubjectPerceel)	
kadastrale_percelen[][kadastrale_code]	string	Volledige kadastrale code
kadastrale_percelen[][gemeente_code]	string	Gemeentecode
kadastrale_percelen[][sectie]	string	Sectie
kadastrale_percelen[][perceelnummer]	integer	Perceelnummer
kadastrale_percelen[][oppervlakte]	integer	Oppervlakte
kadastrale_percelen[][geom]	string	Geometrie in WKT
kadastrale_percelen[][aandeel]	string	aandeel
kadastrale_percelen[][rechtsoort]	string	rechtsoort
kadastrale_percelen[][datum_ingang]	DateTime	Datum ingang
kadastrale_percelen[][datum_eind]	DateTime	Datum ingang

Betrokken database views:

- VIEW_API_KADASTRALE_PERCELEN
- VIEW_API_RECHTEN

Voorbeeld antwoord:

```
{
  "kadastrale_percelen": [
    {
      "kadastrale_code": "ASN00AA1",
      "gemeente_code": "ASN00",
      "sectie": "AA",
      "perceelnummer": 1,
      "oppervlakte": 14476,
      "geom": "aa",
      "aandeel": "1\\2",
      "recht_soort": "Eigendom (recht van)",
      "datum_ingang": "2009-10-31",
    },
    {
      "kadastrale_code": "ASN00AA2",
      "gemeente_code": "ASN00",
      "sectie": "AA",
      "perceelnummer": 2,
      "oppervlakte": 5660,
      "geom": "aa",
      "aandeel": "1\\2",
      "recht_soort": "Eigendom (recht van)",
      "datum_ingang": "2009-10-31",
    },
    {
      "kadastrale_code": "ASN00AA3",
      "gemeente_code": "ASN00",
      "sectie": "AA",
      "perceelnummer": 3,
```



```

    "oppervlakte":6980,
    "geom":"aa"
    "aandeel":"1\2",
    "recht_soort":"Eigendom (recht van)",
    "datum_ingang":"2009-10-31",
  },
],
"offset":"0",
"limit":"10",
"total_item_count":"26526"
}

```

4.4. /SUBJECTEN

Om de details van een subject (natuurlijk of niet-natuurlijk) op te vragen wordt deze collection gebruikt

Parameter	Type	Description
subjectid	string	Subject ID
type	string	Type subject
natuurlijk_subject	object (KadastraalNatuurlijkSubject)	Natuurlijk subject
natuurlijk_subject[voornaam]	string	Voornaam
natuurlijk_subject[tussenvoegsel]	string	Tussenvoegsel(s)
natuurlijk_subject[achternaam]	string	Achternaam
natuurlijk_subject[geslacht]	string	Geslacht (man/vrouw)
natuurlijk_subject[geboorte_datum]	DateTime	Geboortedatum
natuurlijk_subject[geboorte_plaats]	string	Geboorteplaats
natuurlijk_subject[geboorte_land]	string	Geboorteland
natuurlijk_subject[overlijdens_datum]	DateTime	Overlijdensdatum
natuurlijk_subject[is_overleden]	boolean	Is overleden
natuurlijk_subject[adres]	string	Adres
natuurlijk_subject[woonplaats]	string	Woonplaats
natuurlijk_subject[postcode]	string	Postcode
niet_natuurlijk_subject	object (KadastraalNietNatuurlijkSubject)	Niet natuurlijk subject
niet_natuurlijk_subject[kvk]	string	Kamer van Koophandel nummer
niet_natuurlijk_subject[bedrijfsnaam]	string	Bedrijfsnaam
niet_natuurlijk_subject[plaats]	string	Vestigingsplaats

Betrokken database views:

- VIEW_API_SUBJECT_DETAILS

Voorbeeld antwoord:

```

{
  "subjectid":"NL.KAD.Persoon.58006525",
  "type":"KadastraalNatuurlijkSubject",
  "natuurlijk_subject":{
    "voornaam":"Ralf",

```

```
"achternaam": "Ringenaldus",  
"geslacht": "M",  
"geboorte_datum": "1976-09-01T00:00:00+0100",  
"geboorte_plaats": "ASSEN",  
"adres": "Bosgras 4",  
"woonplaats": "ASSEN",  
"postcode": "9408LS"  
}  
}
```

Of

```
{  
  "subjectid": "NL.KAD.Persoon.58148858",  
  "type": "KadastraalNietNatuurlijkSubject",  
  "niet_natuurlijk_subject": {  
    "bedrijfsnaam": "Gemeente Assen",  
    "plaats": "ASSEN"  
  }  
}
```

5. DATA VIEWS

In dit hoofdstuk wordt ingegaan op de views die de data aanleveren aan de REST-service. De collections binnen de service maken gebruik van deze views. De queries hieronder zijn gebaseerd op een database die is afgeleid van het IMKAD model. Het is mogelijk dat ze voor uw database moeten worden aangepast

5.1. ONROERENDE ZAKEN

Deze view geeft alle onroerende zaken terug. De onroerende zaken staan in onze voorbeeld database verdeeld over twee tabellen: percelen met geometrie en percelen zonder. Deze query voegt deze samen zodat alle percelen worden teruggegeven.

Voorbeeld query:

```
create or replace view VIEW_HULP_BRK_ONROERENDEZAKEN as
select GEMC, SECTIE, PRCNR, KADGROOTTE, BRKID, TOESTDATUM, PERCNR17, WKT from BRK_PERCGEOM
union all
select GEMC, SECTIE, PRCNR, KADGROOTTE, BRKID, TOESTDATUM, PERCNR17, NULL from
BRK_PERCZGEOM;
```

5.2. KADASTRALE PERCLEEN

De basis van kadastrale percelen, zoals code, geometrie en adres staan in deze view.

Voorbeeld query:

```
create or replace view VIEW_API_KADASTRALE_PERCELEN as
SELECT
  PERC.BRKID as onroerendezaakid,
  PERC.GEMC as gemeente_code,
  PERC.SECTIE as sectie,
  PERC.PRCNR as perceelnummer,
  PERC.KADGROOTTE as oppervlakte,
  PERC.WKT as geom,
  CASE WHEN ADR.HUISNR = -1 then "
    ELSE ADR.ORNAAM || ' ' || ADR.HUISNR || ADR.HUISNRTOE END as adres,
  ADR.POSTCODE as postcode,
  ADR.WPNAAM as woonplaats,
  ADR.GEMN as gemeente,
  case when ADR.CULBEB is not null then ADR.CULBEB
    ELSE LOC.CULBEB end cultuur
FROM VIEW_HULP_BRK_ONROERENDEZAKEN PERC
  left join BRK_KADBINADRLOC ADR on PERC.BRKID = ADR.BRKOBJID and PERC.TOESTDATUM =
ADR.TOESTDATUM
  left join BRK_LOC LOC on PERC.BRKID = LOC.BRKOBJID and PERC.TOESTDATUM = LOC.TOESTDATUM;
```

5.3. RECHTEN

Op basis van een onroerendezaakid of subjectid kunnen de rechten die te naam gesteld zijn worden opgevraagd.

Voorbeeld query:

```
SELECT
  RECHT.BRKID as zakelijkrechtid,
  RECHT.AARDRE as recht_soort,
  TNS.BRKID as ternaamstellingid,
  TNS.AANDTEL || '/' || TNS.AANDNOEM as aandeel,
  TNS.VPERSTYPE,
  TNS.VPERSID as subjectid,
  to_char(TNS.TOESTDATUM, 'DD-MM-YYYY') as datum_ingang,
  to_char(TNS.EINDDATUM, 'DD-MM-YYYY') as datum_eind,
  RUSTOPID as onroerendezaakid,
  PERC.GEMC as gemeente_code,
  PERC.SECTIE as sectie,
  PERC.PRCNR as perceelnummer,
  PERC.KADGROOTTE as oppervlakte,
  PERC.WKT as geom
FROM VIEW_HULP_BRK_ONROERENDEZAKEN PERC
  inner join BRK_ZAKELIJKERECHT RECHT on (PERC.BRKID = RECHT.RUSTOPID)
  inner join BRK_TENAAMSTELLING TNS on (RECHT.BRKID = TNS.VANZRID);
```

5.4. SUBJECTEN

Hiervoor zijn twee aparte views aangemaakt. Dit is deels vanwege performance, maar ook omdat dit het overzicht ten goede komt.

Voorbeeld query subjecten (basis):

```
create or replace view VIEW_API_SUBJECTEN as
SELECT
  PERS.BRKID as subjectid,
  PERS.NAAM as naam,
  PERS.TYPE as type,
  ADR.WPNAAM as woonplaats
FROM (
  SELECT 'IGGbaPers' as type, BRKID, GESLNAAM as NAAM FROM BRK_IGGBAPERS
  UNION
  SELECT 'NatPers' as type, BRKID, GESLNAAM as NAAM FROM BRK_NATPERS
  UNION
  SELECT 'NietNatPers' as type, BRKID, NAAM FROM BRK_NIETNATPERS
  UNION
  SELECT 'NIGGbaPers' as type, BRKID, GESLNAAM as NAAM FROM BRK_NIGGBAPERS
) PERS
LEFT JOIN BRK_KADBINADR ADR on (PERS.BRKID = ADR.BRKOBJID);
```

Voorbeeld query subjecten (details):

```
create or replace view VIEW_API_SUBJECTEN_DETAILS as
SELECT
  PERS.SUBJECTID,
```

```

PERS.TYPE,
PERS.VOORNAMEN,
PERS.TUSSENVOEGSEL,
PERS.ACHTERNAAM,
PERS.GESLACHT,
PERS.GEBOORTE_DATUM,
PERS.GEBOORTE_PLAATS,
PERS.GEBOORTE_LAND,
PERS.OVERLIJDENS_DATUM,
PERS.IS_OVERLEDEN,
PERS.NNS_NAAM,
PERS.NNS_PLAATS,
PERS.NNS_RECHTSVORM,
ADR.ORNAAM || ' ' || ADR.HUISNR || ADR.HUISNRTOE as adres,
ADR.POSTCODE as postcode,
ADR.WPNAAM as woonplaats
FROM (
  SELECT
    'IGGbaPers' as type,
    BRKID as subjectid,
    VOORNAMEN as voornamen,
    VOORV as tussenvoegsel,
    GESLNAAM as achternaam,
    GESLCODE as geslacht,
    to_char(GEBDAT, 'DD-MM-YYYY') as GEBOORTE_DATUM,
    GEBPLAATS as geboorte_plaats,
    GEBLND as geboorte_land,
    to_char(OVERLDAT, 'DD-MM-YYYY') as OVERLIJDENS_DATUM,
    case when OVERLDAT != to_date('01-01-0001','DD-MM-YYYY') then 'true'
      else 'false' end is_overleden,
    NULL as nns_NAAM,
    NULL as nns_plaats,
    null as nns_rechtsvorm
  FROM BRK_IGGBAPERS
  UNION
  SELECT
    'NatPers' as type,
    BRKID,
    VOORNAMEN,
    VOORV,
    GESLNAAM,
    GESLCODE,
    GEBPLAATS,
    GEBLND,
    to_char(GEBDAT, 'DD-MM-YYYY') as GEBOORTEDATUM,
    to_char(OVERLDAT, 'DD-MM-YYYY') as OVERLIJDENS DATUM,
    case when OVERLDAT != to_date('01-01-0001','DD-MM-YYYY') then 'true'
      else 'false' end is_overleden,
    NULL as NAAM,
    NULL as STATZETEL,
    null as RV
  FROM BRK_NATPERS
  UNION
  SELECT
    'NietNatPers' as type,
    BRKID,
    NULL as VOORNAMEN,
    NULL as VOORV,
    NULL as GESLNAAM,
    NULL as GESLCODE,
    NULL as GEBPLAATS,

```

```

NULL as GEBLND,
NULL as GEBOORTEDATUM,
NULL as OVERLIJDENSdatum,
NULL as is_overleden,
NAAM,
STATZETEL,
RV
FROM BRK_NIETNATPERS
UNION
SELECT
'NIGGbaPers' as type,
BRKID,
VOORNAMEN,
VOORV,
GESLNAAM,
GESLCODE,
GEBPLAATS,
GEBLND,
to_char(GEBDAT, 'DD-MM-YYYY') as GEBOORTEDATUM,
to_char(OVERLDAT, 'DD-MM-YYYY') as OVERLIJDENSdatum,
case when OVERLDAT != to_date('01-01-0001','DD-MM-YYYY') then 'true'
     else 'false' end is_overleden,
NULL as NAAM,
NULL as STATZETEL,
null as RV
FROM BRK_NIGGBAPERS
) PERS
LEFT JOIN BRK_KADBINADR ADR on (PERS.subjectid = ADR.BRKOBJID);

```

5.5. AANTEKENINGEN

Aantekeningen waar subjecten of tenaamstellingen in voorkomen, staan in onderstaande view.

Voorbeeld query:

```

create or replace view VIEW_API_AANTEKENINGEN as
select
  AANT.BRKID as aantekeningid,
  AANT.AARD as AantekeningAard,
  to_char(AANT.TOESTDATUM, 'DD-MM-YYYY') as datum_ingang ,
  to_char(AANT.EINDDATUM, 'DD-MM-YYYY') as datum_eind,
  ABKO.RELID as onroerendezaakid,
  ABKO.TYPE as aantekeningop,
  ABP.RELID as PERSOONID,
  ABR.RELID as TENAAMSTELLINGID
from BRK_AANTBETRKO ABKO
  left join BRK_AANTEKENING AANT ON ABKO.BRKOBJID = AANT.BRKID AND ABKO.TOESTDATUM =
AANT.TOESTDATUM
  left join BRK_AANTBETRRECHT ABR on ABR.BRKOBJID= AANT.BRKID AND ABKO.TOESTDATUM =
ABR.TOESTDATUM
  left join BRK_AANTBETRPERS ABP on ABP.BRKOBJID = AANT.BRKID AND ABKO.TOESTDATUM =
ABP.TOESTDATUM;

```

5.6. ONTSTAAN UIT/ OVERGEGAAN IN

Hier ook weer twee views om het e.a. makkelijker te maken tijdens het opzetten van de REST-services

Voorbeeld query (ontstaan uit):

```
create or replace view VIEW_API_PERCEEL_ONTSTAAN_UIT as
select
  FIL.BRKOBJID as onroerendezaakid,
  FIL.AARD,
  FIL.OVGGROOTTE,
  FIL.OZID as ontstaan_uit_onroerendezaakid,
  to_char(FIL.TOESTDATUM, 'DD-MM-YYYY') as datum_ingang
from BRK_OZFILIATIE FIL
where FIL.ONTSTUIT = 'True';
```

Voorbeeld query (overgegaan in):

```
create or replace view VIEW_API_PERCEEL_OVERGEGAAN_IN as
select
  FIL.BRKOBJID as onroerendezaakid,
  FIL.AARD,
  FIL.OVGGROOTTE,
  FIL.OZID as overgegaan_in_onroerendezaakid,
  to_char(FIL.TOESTDATUM, 'DD-MM-YYYY') as datum_ingang
from BRK_OZFILIATIE FIL
where FIL.ONTSTUIT = 'False';
```