4.6 SimpleQL

DEPRICATED

Denna sida tillhör en äldre version av tjänsten.

Aktuell version av detta dokument hittar ni på:
https://confluence.cgiostersund.se/display/PU/SimpleQL

Innehåll

1. Bakgrund

SimpleQL är framtagen för att enkelt kunna söka personuppgifter via framtaget RIV-TA kontrakt genom att använda en sedan tidigare känd syntax och hålla nere komplexiteten. Valet föll på en SQL-baserade syntax för igenkänningsfaktorn men med anpassningar för att ge det stöd som behövs inom användningsfallen för personuppgiftstjänsten med låg komplexitet.

Även om syntaxen baseras på SQL så är det viktigt att poängtera att frågorna **aldrig** ställs direkt mot databasen. Dvs det finns **ingen** risk för SQL-injection. Alla söksträngar som kommer in till Personuppgiftstjänsten går först igenom validering innan den går genom en parser som översätter den till en syntax som kan användas mot Personuppgiftstjänsten databas (som ej är SQL-baserad).

2. Beskrivning

I en söksträng ingår alltid två *keywords*. Dessa är **FROM** samt **WHERE** och är de enda *keywords* som används och de är alltid obligatoriska. Tillsammans med dessa anges vilka noder man vill söka på och vilka dessa noder är begränsas enligt beskrivning i detta dokument. Till noderna anges sedan operatorer och värden. Precis som med noderna så är operatorerna begränsade och specificeras i detta dokument.

Med *noder* så avses de noder som ingår i typen *PersonRecordType* som är en del av domänen *strategicresourcemanagement:persons:person*. (Dock förekommer förenklingar och anpassningar där lämpligt).

Alla sökningar görs utan hänsyn till gemen/versal skrivning (case insensitive).

OBS! Tjänsten returnerar **samma** uppgifter gällande sekretessmarkerade personer som motsvarande tjänstekontrakt gör.

I språket ingår fördefinierade värden för vilken typ identitet som man vill söka på. Mer info om fördefinierade värden nedan.

3. Användning / Begränsning

Nedan anges vilka keywords, noder, samt operatorer som är giltiga för PU-tjänsten.

En generisk söksträng följer följande mönster:

```
FROM <rootnod att utgå ifrån> WHERE <nod> <operator> '<värde>' AND (<nod> <operator> '<värde>' OR <nod>
<operator> '<värde>')
```

'<värde>' skall alltid anges inom enkla citationstecken alternativt "<värde>"

3.1. Testpersoner

I både Produktionsmiljö och Testmiljö finns personposter som är flaggade som testperson. Dessa ligger i en separat databas och är editerbara av personal på Inera Test. För att få med dessa i resultatet av en sökning måste man ange detta i sin fråga genom att sätta flaggan *includetestidentities* till *true*.

Obs! Notera att detta även gäller testmiljöer där det även här finns personposter som inte är flaggade som testpersoner tillsammans med de som är det.

3.2. Keywords

Namn	Beskrivning
FROM	Anger vilken nod som skall vara root-nod och som samtliga noder specificerade i WHERE satsen är relativa emot.
WHERE	Anger noder, operatorer och värden för sökningen

3.3. Logiska operatorer

Namn	Beskrivning	Exempel
AND	Används för att sätta samma flera villkor där alla villkor måste vara uppfyllda.	`<nod1> = '<värde>' AND <nod2> = '<värde>'`
OR	Används för att sätta samman flera villkor där ett eller flera villkor måste vara uppfyllda. Frågor innehållande OR ska anges med parenteser.	`(<nod1> = '<värde>' OR <nod1> = '<värde>')`

3.4. Datatyper

Datatyp	Nod-exempel	Värde-exempel	
String	`personrecord.name.givenname`	`'Anna'`	
Numeric	`personrecord.addressinformation.residentialaddress.postalcode`	`'83132'`	

PartialDate	`personrecord.birth.dateofbirth`	`'1912'`
		`'1912-12'`
		`'1912-12-12'`	

OID (IIType.root)	`personrecord.personalidentity.root`	`'1.2.752.129.2.1.3.1'` `'PNR'`	Kan anges som antingen det exakta OID-värdet (som då hanteras som en String), men det kan kan även anges som ett fördefinierat värde i enlighet med Fördefinierade värden
Boolean	`protectedpersonindicator`	'true' 'false'	

3.5. Jämförande operatorer

Namn	För datatyp	Beskrivning	Exempel
=	Alla	Anger att värdet för noden skall matchas exakt (dock case insensitive).	`<nod> = 'daniel'`
!=	Alla	Anger att värdet för noden **inte** skall matcha exakt (case sensitive).	<nod> != 'kalle'

LIKE	String
	
	PartialDate
		Anger att värdet för noden innehåller 1 wildcard i form av "%" på slutet.
		
		Används för att söka på ofullständigt värde.
			`<string nod> LIKE 'Andersss%'`
			
			`<partialdate nod> OBS! SE NEDAN`

IN	Alla	Anger att värdet på noden ska finnas i en specificerad värdemängd.	`<nod> IN ('daniel','anna','noa')`
NOT IN	Alla	Anger att värdet på noden inte ska finnas i en specificerad värdemängd.	<nod> NOT IN ('daniel', 'anna', 'noa')

BETWEEN	Numeric
	
	Date
	
	PartialDate
	
	PNR, SNR (Hanteras som numeriska värden i detta fall)
		Anger att värdet på noden ska finns mellan två värden.
		
		Frågor innehållande OR ska anges med parenteser.
			`(<numeric nod> BETWEEN '132' AND '321')`
			
			`(<personid nod> BETWEEN '197804010000' AND '197804319999')`
			
			`<partialdate nod> OBS! SE NEDAN`

>
 	
<
 	
>=
 	
<=
	Numeric
	
	Date
	
	PartialDate
	
	PNR, SNR (Hanteras som numeriska värden i detta fall)
		Anger att värdet för noden ska vara:
		
		• *större än*
		• *mindre än*
		• *större eller lika med*
		• *mindre eller lika med*
		
		angivet värde.
			`<numeric nod> > '321'`
			
			`<personid nod> > '197804000000'`
			
			`<date nod> <= '1983-03-04'`
			
			`<partialdate nod> OBS! SE NEDAN`

IS NULL
 	
IS NOT NULL
	Alla	Anger att noden måste vara satt, kontra att noden inte får vara satt.	`<nod> IS NULL`
			
			`<nod> IS NOT NULL`

3.6. Syntax

Exempelfrågor	
=	`FROM PersonRecord WHERE personalIdentity.extension = "191212121212";`
!=	FROM PersonRecord WHERE personalIdentity.extension != "191212121212";
LIKE	`FROM PersonRecord.name WHERE givenName LIKE "Ad%";`
IN	`FROM PersonRecord.addressInformation.residentialAddress WHERE postalCode IN ("83100","83134","83433");`
NOT IN	FROM PersonRecord WHERE PopulationRegistrationLocality.CountyCode NOT IN ("01","04") AND Version >= "20210929162541"
BETWEEN	`FROM PersonRecord.populationRegistrationLocality WHERE (populationRegistrationDate BETWEEN "1990-01-01" AND "1999-12-31");`
>	`FROM PersonRecord.immigration WHERE immigrationDate > "1990";`
<	`FROM PersonRecord.immigration WHERE immigrationDate < "1990";`
>=	`FROM PersonRecord.immigration WHERE immigrationDate >= "1990";`
<=	`FROM PersonRecord.immigration WHERE immigrationDate <= "1990";`
IS NULL	`FROM PersonRecord WHERE name.middleName IS NULL;`
IS NOT NULL	`FROM PersonRecord WHERE name.middleName IS NOT NULL;`
OR	`FROM PersonRecord WHERE (personalIdentity.extension = "191212121212" OR personalIdentity.extension = "181212121212");`

AND	`FROM PersonRecord WHERE personalIdentity.extension = "191212121212" AND gender = "1" AND includetestidentities = "true";`
OR / AND	`FROM PersonRecord WHERE (personalIdentity.extension LIKE "19%" OR personalIdentity.extension LIKE "20%") AND gender = "1";`

3.6.1. PartialDate

När det gäller jämförelse på PartialDate så är det inte självklart hur dessa skall jämföras då mängden data som fältet innehåller kan variera. Följande regelverk gäller för de olika operanderna.

Operator	Beskrivning	Exempel

=
 	
!=
 	
IN
 	
NOT IN
	Samtliga värden är giltiga att ange och dessa matchas exakt.
	
	Anges exempelvis '1912-01' som värde, så ges det enbart träff på de som har exakt detta värde. Dvs träff ges ej på de med värde '1912' eller '1912-01-03'.
		`personrecord.birth. dateofbirth = '1912-01'`
		`personRecord.birth. dateOfBirth != '1912-01'`
		`personrecord.birth. dateofbirth IN ('1912','1913')`
		`personrecord.birth. dateofbirth NOT IN ('1912','1913')`

LIKE	Matchar alla värden som börjar med ett visst värde. Kan exempelvis användas för att söka på alla som har 1912 som angivet år, oavsett ytterligare kvalificerare. Sökning på "LIKE '1912%'" matchar '1912', '1912-02', '1912-03-17' o.s.v.	`personrecord.birth. dateofbirth LIKE '1912%'`

>
 	
<
 	
>=
 	
<=
 	
BETWEEN
	Sökningen matcher enbart de partialDate som har samma antal värdesiffror som anges i frågan. Om villkor ställs med ett värde innehållande **YYYY**, så ges enbart träff på de som har **YYYY** lagrat i databasen. De med värden **YYYY-MM** samt **YYYY-MM-DD** matchas inte.
	
	Sökning på "> 1912-01" matchar alla partialDate som är lagade på formatet **YYYY-MM** och som har ett värde större än "1912-01".
	
	Om man vill åstadkomma en sökning som finner alla partialDate från 1950 till 1959-12-31 oavsett vilket format de har i databasen så ställs en fråga med tre "OR" villkor.
	
	((nod BETWEEN '1950' AND '1959') OR (nod BETWEEN '1950-01' AND '1959-12') OR (nod BETWEEN '1950-01-01' AND '1959-12-31'))
	
	Här blir det dubbel parantes då både OR och BETWEEN operatorer ska omges av parenteser.
		`personrecord.birth. dateofbirth >= '1950-01-01'`
		
		`personrecord.birth. dateofbirth >= '1978'`
		
		`(personrecord.birth. dateofbirth BETWEEN '1950-01-01' AND '1959-12-31')`

3.7. Noder

Nod namn	Datatyp	Kommentar
`primaryidentity`	boolean	Specialnod/flagga som endast kan anges en gång i samma fråga. Noden får inte vara omgiven av parenteser. Måste användas med andra noder dock inte bara med andra specialnoder som t.ex includeTestIdentities. Ifall denna sätts till *true* kommer endast personposter som är huvudidentitet inkluderas i svaret. Giltiga värden är true/false.
`includetestidentities`	boolean	Specialnod/flagga som endast kan anges en gång i samma fråga. Noden får inte vara omgiven av parenteser. Måste användas med andra noder dock inte bara med andra specialnoder som t.ex primaryIdentity. Om satt kommer även personposter flaggade som testpersoner att sökas och inkluderas i svaret. Giltiga värden är true/false.

maxResultsPerFile	Numeric	Specialnod som endast kan anges en gång i samma fråga. Noden får inte vara omgiven av parenteser. Måste användas med andra noder dock inte bara med andra specialnoder som t.ex primaryIdentity. Anger ifall man önskar att sökresultatet delas upp i flera filer och i sådana fall hur många personposter som max skall läggas i varje resultatfil. Ifall ej satt så kommer resultatet att levereras i endast en (1) fil oavsett hur stort resultatet är. Har endast effekt i "ByOrder"-kontrakten.
gender	String	Kodverk 1 för Man 2 för Kvinna
protectedpersonindicator	boolean	Sekretessmarkering hos Skatteverket
protectedPopulationRecord	boolean	Skyddad folkbokföring hos Skatteverket
personrecord.personalidentity.root	String	OID eller fördefinierat värde för ID-typ. Se Fördefinierade värden nedan samt 4.6 Identitetsformat för möjliga värden.
personrecord.personalidentity.extension	String /Numeric	Personnummer, Samordningsnummer eller nummer för Reservidentitet.
personrecord.version	Numeric	Tidsstämpel då personposten senast uppdaterades i Personuppgiftstjänstens databas.
personrecord.name.givenname	String	
personrecord.name.middlename	String	
personrecord.name.surname	String	
personrecord.birth.dateofbirth	String /PartialDate	
personrecord.birth.placeofbirthsweden.birthcountycode	String	
personrecord.birth.placeofbirthsweden.birthparish	String	
personrecord.birth.birthabroad.placeofbirthabroad.placeofbirthabroad	String	
personrecord.birth.birthabroad.countryofbirth	String	
personrecord.populationregistrationlocality.countycode	String	
personrecord.populationregistrationlocality.municipalitycode	String	
personrecord.populationregistrationlocality.parishcode	String	
personrecord.populationregistrationlocality.populationregistrationdate	String /PartialDate	
personrecord.addressinformation.nationalkeys.propertyid	String	

personrecord.addressinformation.nationalkeys.addressplaceid	Numeric	
personrecord.addressinformation.nationalkeys.apartmentid	Numeric	
personrecord.addressinformation.nationalkeysuuid.propertyid	String	
personrecord.addressinformation.nationalkeysuuid.addressplaceid	String	
personrecord.addressinformation.nationalkeysuuid.apartmentid	String	
personrecord.addressinformation.specialpostaladdress.postaladdress1	String	
personrecord.addressinformation.specialpostaladdress.postaladdress2	String	
personrecord.addressinformation.specialpostaladdress.careof	String	
personrecord.addressinformation.specialpostaladdress.postalcode	Numeric	
personrecord.addressinformation.specialpostaladdress.city	String	
personrecord.addressinformation.residentialaddress.postaladdress1	String	
personrecord.addressinformation.residentialaddress.postaladdress2	String	
personrecord.addressinformation.residentialaddress.careof	String	
personrecord.addressinformation.residentialaddress.postalcode	Numeric	
personrecord.addressinformation.residentialaddress.city	String	
personrecord.addressinformation.addressabroad.postaladdress1	String	
personrecord.addressinformation.addressabroad.postaladdress2	String	
personrecord.addressinformation.addressabroad.postaladdress3	String	
personrecord.addressinformation.addressabroad.countrycode	String	
personrecord.addressinformation.addressabroad.addressabroaddate	String /ParitalDate	
personrecord.addressinformation.addressabroad.votingdate	String /PartialDate	

personrecord.addressinformation.district.districtcode	String	
personrecord.contactinformation.contacttype	String	
personrecord.contactinformation.use	String	
personrecord.contactinformation.value	String	
personrecord.contactperson.contactrelationshiptype	String	
personrecord.contactperson.givenname	String	
personrecord.contactperson.middlename	String	
personrecord.contactperson.surname	String	
personrecord.contactperson.contactpersonaddress.postaladdress1	String	
personrecord.contactperson.contactpersonaddress.postaladdress2	String	
personrecord.contactperson.contactpersonaddress.careof	String	
personrecord.contactperson.contactpersonaddress.postalcode	Numeric	
personrecord.contactperson.contactpersonaddress.city	String	
personrecord.contactperson.contactpersoncontactinformation.contacttype	String	
personrecord.contactperson.contactpersoncontactinformation.use	String	
personrecord.contactperson.contactpersoncontactinformation.value	String	
personrecord.confirmedidentity.typeofidentification	String	
personrecord.confirmedidentity.identificationnumber	String	
personrecord.confirmedidentity.attachmentid	String	
personrecord.confirmedidentity.issuersofid	String	
personrecord.administrativeinformation.categoryofperson	String	
personrecord.administrativeinformation.accountcode	String	
personrecord.deregistration.deregistrationreasoncode	String	
personrecord.deregistration.deregistrationdate	String /PartialDate	

personrecord.maritalstatus.maritalstatuscode	String	
personrecord.maritalstatus.maritalstatusdate	String /PartialDate	
personrecord.immigration.immigrationdate	String /PartialDate	
personrecord.immigration.immigrationidentity. personalidentitynumber	String	
personrecord.citizenship.citizenshipcountrycode. countrycode	String	
personrecord.citizenship.citizenshipdate	String /PartialDate	
personrecord.citizenship.status	String	
personrecord.relationship.relationshipid. personalidentity.extension	String	
personrecord.relationship.relationshipid. dateofbirth	PartialDate	
personrecord.relationship.relationshiptype	String	
personrecord.relationship.status	String	
personrecord.attachment.id	String	
personrecord.attachment.mediatype	String	
personrecord.populationRegistrationRecord. historicalRecords.historicalRegistration. municipalityCode	String	
personrecord.populationRegistrationRecord. historicalRecords.historicalRegistration.countyCode	String	
personrecord.populationRegistrationRecord. historicalRecords.historicalRegistration. localRegistrationTime	Numeric	Specialnod som ej finns inkluderad som informationsmängd i tjänstekontrakten. Tidsstämpel då den historiska folkbokföringen lästes in från Navet för första gången. Finns endast på historiska poster ändrade efter februari 2019.
personrecord.populationRegistrationRecord.historicalRecords. historicalRegistration.localRegistrationEndTime	Numeric	Specialnod som ej finns inkluderad som informationsmängd i tjänstekontrakten. Tidsstämpel då den historiska folkbokföringen slutade att gälla till förmån för en ny historisk folkbokföringspost. Finns endast på historiska poster ändrade efter december 2021.

3.8. Fördefinierade värden

Dessa används för att ange vilken typ av personidentitet som sökningen avser. De kan anges istället för den specifika OID (eller gruppering av OID) som har motsvarande betydelse.

Namn	Beskrivning	Motsvarar	Exempel

PNR	Typen av identitet är personnummer.	`'1.2.752.129.2.1.3.1'`	`personalidentity.root = '1.2.752.129.2.1.3.1'`
			
			`personalidentity.root = 'PNR'`

SNR	Typen av identitet är samordningsnummer.	`'1.2.752.129.2.1.3.3'`	`personalidentity.root = '1.2.752.129.2.1.3.3'`
			
			`personalidentity.root = 'SNR'`

NRID	Typen av identitet är reservidentitet som följer formatet för nationell reservidentitet.	`'1.2.752.74.9.1'`	`personalidentity.root = '1.2.752.74.9.1'`
			
			`personalidentity.root = 'NRID'`

LRID	Typen av identititet är lokalt reservid, men inte specifikt vilket.	`Alla som inte är av ovan angivna värden.`	`personalidentity.root IN ('1.1.1','2.2.2',.....)`
			
			`personalidentity.root = 'LRID'`

4. Exempel på söksträngar

Personer med förnamn Johan och efternamn Andersson.

```
FROM PersonRecord.Name
        WHERE GivenName = 'Johan'
        AND SurName = 'Andersson';
```

Personer födda 1978 med förnamn Anna och som är folkbokförd i kommun 0183 (notera uppdelning av län och kommun enligt SKV's data).

```
FROM PersonRecord
        WHERE PersonalIdentity.Extension LIKE '1978%'
        AND Name.GivenName = 'Anna'
        AND PopulationRegistrationLocality.CountyCode = '01'
        AND PopulationRegistrationLocality.MunicipalityCode = '83';
```

Personer boende inom postnummer 83132 som är födda i Maj 2014.

```
FROM PersonRecord
        WHERE AddressInformation.ResidentialAddress.PostalCode = '83132'
        AND PersonalIdentity.extension LIKE '201405%';
```

Personer boende i län 04 på gata Trollrunan och med efternamn Hansson.

```
FROM PersonRecord
        WHERE PopulationRegistrationLocality.CountyCode = '04'
        AND AddressInformation.ResidentialAddress.PostalAddress2 LIKE 'Trollrunan%'
        AND Name.SurName = 'Hansson';
```

Personer där könet är man på deras huvudidentitet.

```
FROM PersonRecord
        WHERE Gender  = '1'
        AND PrimaryIdentity = 'true';
```

Personer med meborgarskap i Finland .

```
FROM PersonRecord.Citizenship.citizenshipCountryCode
        WHERE countryCode  = 'FI';
```

5. Exempel på Request

Följande exempel visar ett request mot tjänsten SearchPersonsForProfile. Enbart body av requestet är representerat. Sökningen efterfrågar följande:

Alla personer i län 01 som har förnamn Johan och ett efternamn som börjar på Trulls. I svaret efterfrågas data i form av svarsprofil 2.

```
<SearchPersonsForProfile>
  <query>FROM PersonRecord WHERE Name.GivenName = 'Johan' AND Name.SurName LIKE 'Trulls%' AND
PopulationRegistrationLocality.CountyCode = '01';</query>
  <queryLanguage>SimpleQL</queryLanguage>
  <profile>P2</profile>
</SearchPersonsForProfile>
```