Waardegebonden toegang in DataPA® OpenAnalytics

Auteur Arjaan den Ouden

Datum 4 december 2013

Status Definitief

Versie 1.0

1 Inleiding

DataPA® OpenAnalytics is dé oplossing voor complexe rapportages en dashboards op basis van data in Progress® OpenEdge® databases. Het stelt eindgebruikers in staat om zelf rapporten en dashboards te genereren en de daarvoor benodigde gegevens op te vragen uit de business databases. Naast de uitgebreide rapport- en dashboardfunctionaliteit biedt DataPA volledige ontsluiting van gegevens in Microsoft® Office Excel® en Access®. Door de uitgebreide licentiestructuur is het ook mogelijk om gebruikers zelfstandig queries te laten maken op basis van voorgedefinieerde views op de database(s).

Voor toegang tot gegevens in een administratiesysteem zijn vaak restricties van toepassing. Gebruikers hebben alleen toegang tot gegevens die voor hun deel van de werkzaamheden noodzakelijk zijn. Dit wordt “waardegebonden toegang genoemd”. Tooling voor rapporten en dashboards moet deze restricties respecteren. Zo ook DataPA. Dit document beschrijft de beschikbare mogelijkheden in DataPA om rekening te houden met de waardegebonden toegang tot een administratiesysteem.

NB: Dit document is geen handleiding voor de installatie of het gebruik van DataPA.

2 Algemene werking van DataPA

2.1 Inleiding

De werking van DataPA is op meerdere plaatsen uitgebreid gedocumenteerd:

- DataPA website: http://datapa.com/
- DataPA documentatie: http://datapa.com/support/online-documentation
- Document “OTAP voor DataPA® OpenAnalytics”, GYBA, september 2013: http://www.gyba.nl/downloads/DataPA_OTAP.pdf

Voor zover niet relevant voor waardegebonden toegang wordt de werking van DataPA in dit document niet verder toegelicht.

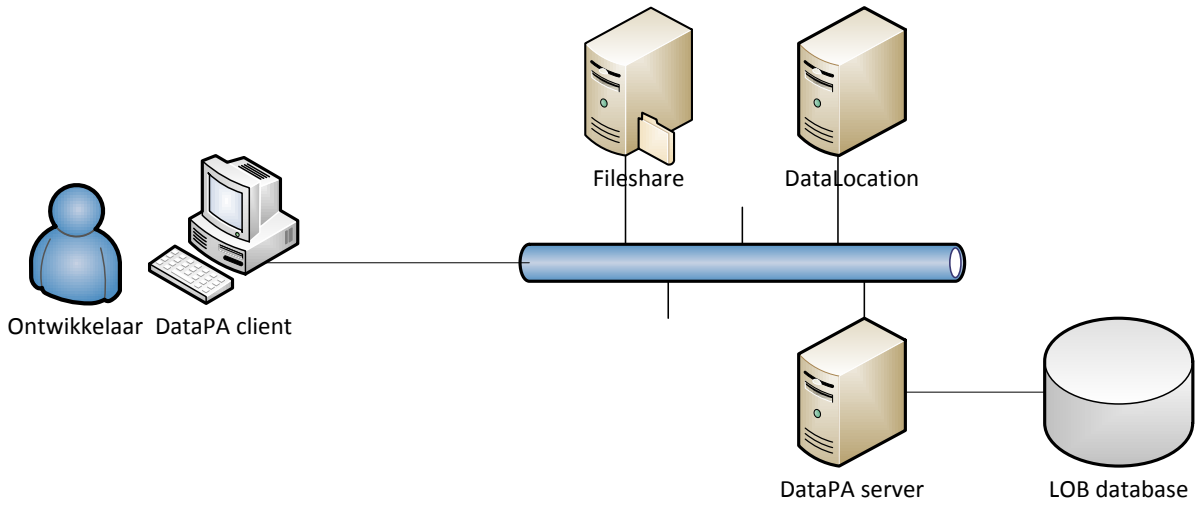
2.2 Client / server

DataPA is een client/server applicatie. De clients data op bij een Progress AppServer die is ingericht voor DataPA. De AppServer heeft connectie met de databases en voeren de queries op die database uit. De resultaten worden geretourneerd aan de client die deze verwerkt tot rapporten, in dashboards, Excel of Access tabellen.

Ook de configuratie van DataPA wordt op een AppServer onderhouden. Dit is niet noodzakelijkerwijs dezelfde AppServer als degene die de queries op de databases uitvoert.

2.3 Minimale DataPA “keten”

Met de minimale DataPA keten bedoelen we de minimale set aan onderdelen die nodig is om één complete omgeving neer te zetten. De minimale keten is weergegeven in **Fout! Verwijzingsbron niet gevonden.**.



Figuur 1 – Minimale keten

- De gebruiker gebruikt de DataPA client applicaties.
- De DataPA client applicaties connecteren via het netwerk met de DataLocation server voor user authentication, licentiebeheer en het beheer van configuratie/metadata. De DataLocation server is een Progress Appserver.
- De DataPA client applicaties connecteren via het netwerk met de DataPA server om queryopdrachten af te geven. De DataPA server is een Progress Appserver.
- De DataPA server connecteert met de business applicatie database voor het ophalen van gegevens.
- De DataPA server retourneert de opgehaalde gegevens naar de DataPA client.
- De DataPA client genereert een rapport, dashboard, excel werkboek of access tabel met de ontvangen gegevens en toont deze aan de gebruiker.

NB

De rollen van de DataLocation en DataPA server kunnen door één en dezelfde AppServer worden uitgevoerd. Voor de duidelijkheid zijn deze rollen hier apart afgebeeld.

In de configuratie van het systeem (de business applicatie) kunnen connecties naar meerdere DataPA servers worden ingericht. Bijvoorbeeld naar alle OTAP-omgevingen van de business applicatie.

3 Toegang tot data

3.1 Inleiding

Zodra DataPA is geconfigureerd met een systeem, verbindingen en onderwerpen, kan een gebruiker queries op de database afvuren. Zonder aanpassingen heeft iedere gebruiker van DataPA via elke connectie toegang tot alle gedefinieerde systemen en alle onderwerpen. Ook hebben queries op basis van de onderwerpen toegang tot alle records in de ontsloten tabellen.

Dit kan tot ongewenste situaties leiden: gebruikers hebben in de business applicatie alleen toegang tot subsets van de data waarvoor zijn geautoriseerd, maar via DataPA hebben ze alsnog inzicht in gehele set aan data.

DataPA kan zodanig ingeregeld worden dat deze situatie niet optreedt. In de volgende paragrafen worden de mogelijkheden hiertoe beschreven. Dit zijn:

- Gebruikers toegang tot systemen of instanties van systemen (via de connecties) ontzeggen; zie 3.3.
- Gebruikers toegang tot bepaalde onderwerpen ontzeggen; zie 3.4.
- Autorisatie in de business applicatie toepassen op DataPA; zie 3.5.

Alle mogelijkheden kunnen in willekeurige combinatie met elkaar gebruikt.

In alle gevallen is het noodzakelijk dat DataPA de gebruiker ‘kent’. Hiervoor zijn 2 mogelijkheden die in paragraaf 3.2 worden uitgewerkt:

- DataPA gebruikt windows user account
- DataPA laat gebruikers expliciet inloggen

3.2 Gebruikers in DataPA

3.2.1 Windows user account

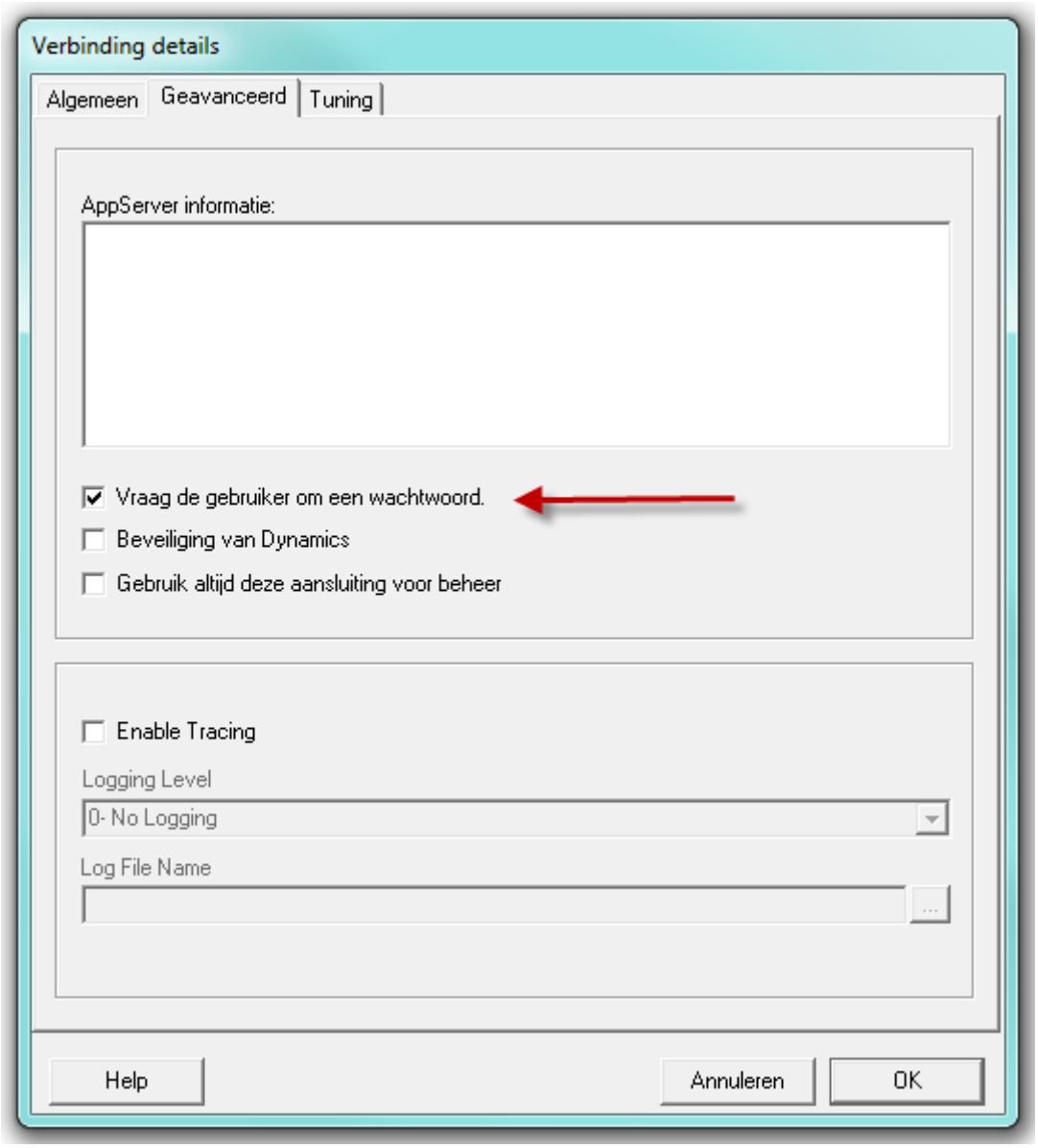
DataPA werkt altijd in de context van een gebruiker. Bij het opstarten van DataPA is dat het Windows gebruikers account. Dit account wordt o.a. gebruikt om de juiste licentie voor de betreffende gebruiker toe te passen.

Zonder verder configuratie wordt het Windows user account gebruikt in de mogelijkheden die in de paragrafen 3.3 tot 3.5 worden beschreven.

3.2.2 Gebruikers in laten loggen

Voor iedere connectie die voor een systeem in DataPA wordt geconfigureerd, kan aangegeven worden dat de gebruiker moet inloggen om de betreffende connectie te gebruiken. Zie Figuur 2. Zodra

DataPA een connectie met een AppServer nodig heeft, wordt de gebruiker om een zijn gebruikersnaam en wachtwoord gevraagd. Zie Figuur 3.



Figuur 2 – Vraag gebruikers om een wachtwoord bij gebruik van een verbinding

Figuur 3 – Invoeren van gebruikersnaam en wachtwoord

De gebruikersnaam en het wachtwoord worden doorgegeven aan de connect-procedure van de betreffende AppServer. De connect-procedure is vervolgens verantwoordelijk om te bepalen of de gegevens correct zijn en of de gebruiker toegang tot deze AppServer heeft. Als dit niet het geval is, moet de connect-procedure een error retourneren. Zie Figuur 4 voor een voorbeeld van de connect-procedure.

```
DEFINE INPUT  PARAMETER cUsername      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cPassword      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cAppServerInfo AS CHARACTER   NO-UNDO.

IF NOT ValidateUser(cUsername, cPassword) THEN RETURN ERROR.
```

Figuur 4 – Voorbeeld connect-procedure

Als een gebruiker op deze manier is ingelogd, wordt de ingevoerde gebruikersnaam gebruikt in verder beschreven mogelijkheden. Als inloggen niet is aangevinkt, wordt de Windows gebruikersnaam gebruikt.

3.3 Filteren van systemen en connecties

Als een gebruiker wel alle records van de tabellen in een geconnecteerd systeem mag inzien, maar geen toegang tot alle (instanties van) systemen mag hebben, moet het filter voor systemen en connecties gebruikt worden.

Om dit filter te gebruiken, moet aan twee voorwaarden zijn voldaan:

- De DataLocation voor DataPA is AppServer
- Op de AppServer is een PAFilterConnections.p procedure geïmplementeerd

Als aan deze voorwaarden is voldaan, zal de PAFilterConnections procedure uitgevoerd worden, zodra DataPA wordt opgestart. In de procedure zijn alle beschikbare systemen met hun connecties beschikbaar in een temp-table. De procedure krijgt de gebruikersnaam als input parameter. Door

systemen en/of connecties uit de temp-table te verwijderen, worden ze onbereikbaar voor de gebruiker. De logica om de records te verwijderen kan naar eigen inzicht opgebouwd worden. Zie Figuur 5 voor een voorbeeld van een PAFilterConnection procedure.

```
DEFINE TEMP-TABLE ttConnections
    FIELD cName           AS CHARACTER
    FIELD cConnectionName AS CHARACTER
    FIELD bPrimary        AS LOGICAL
    FIELD bAdmin          AS LOGICAL.

DEFINE INPUT  PARAMETER cUsername   AS CHARACTER NO-UNDO.
DEFINE INPUT-OUTPUT PARAMETER TABLE FOR ttConnections.

FOR EACH ttConnections:

    IF NOT UserCanAccess(cUsername, cName, cConnectionName) THEN DO:
        DELETE ttConnections.
    END.
END.
```

Figuur 5 – Voorbeeld van PAFilterConnections.p

3.4 Filteren van onderwerpen

Als een gebruiker wel alle records van de tabellen in een geconnecteerd systeem mag inzien, maar niet alle onderwerpen mag gebruiken, moet het filter voor onderwerpen, velden, links en zoekvelden gebruikt worden.

Om dit filter te gebruiken, moet aan vier voorwaarden zijn voldaan:

- De DataLocation voor DataPA is AppServer
- De DataPA configuratie wordt opgeslagen in een database
- Op de AppServer is een PADBStoreFile.p procedure geïmplementeerd
- Op de AppServer is een PADBGetFile procedure geïmplementeerd

Als aan deze voorwaarden is voldaan, kan de PADBGetFile procedure gebruikt worden om op basis van de gebruikersnaam configuratie voor onderwerpen, velden, links en zoekvelden beschikbaar te stellen aan de ingelogde gebruiker. De procedure krijgt de gebruikersnaam als input parameter.

Het uitwerken van deze optie gaat te ver voor dit document. Een goed voorbeeld kan gevonden worden in de kennisbank op de DataPA website:

http://support.datapa.com/ics/support/default.asp?deptID=5554&task=knowledge&questionID=104

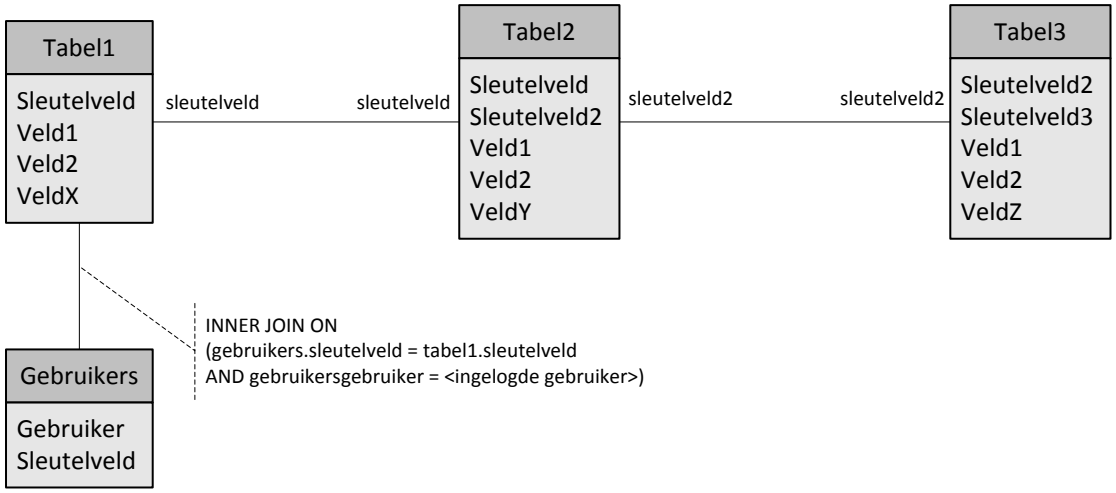
3.5 Autorisatie van business applicatie toepassen

Als een gebruiker op basis van toegangsrechten niet alle records van de tabellen in een geconnecteerd systeem mag inzien, moeten de onderwerpen hierop aangepast worden.

Om dit mogelijk te maken, moet er aan drie voorwaarden voldaan worden:

- De hoofdtabel in een onderwerp moet een sleutel bevatten waarop gefilterd kan worden. Bijvoorbeeld een contractnummer.
- In het onderwerp moet een inner join op basis van de sleutel in de voorwaarde hierboven gemaakt worden met de tabel met toegangsrechten. Deze tabel moet gefilterd kunnen worden op een gebruiker.
- Er moet een functie aanwezig zijn die de gebruikersnaam in DataPA koppelt aan de gebruikersnaam in de business applicatie zodat de tabel in de voorgaande voorwaarde gefilterd kan worden.

Schematisch moet een onderwerp er dan uit zien als weergegeven in Figuur 6.



Figuur 6 – Schematische weergave van een onderwerp met filtering op gebruikersnaam

De tabellen Tabel1, Tabel2 en Tabel3 zijn op reguliere wijze gelinked in het onderwerp. Hiervoor zijn de relevante join gebruikt. Extra in het onderwerp is de INNER JOIN naar de tabel Gebruikers. De inner join zorgt ervoor dat het totale onderwerp alleen maar records oplevert die in beide ‘paden’ voorkomen. In casu komt het er daarmee op neer dat als de Gebruiker niet geselecteerd kan worden met bijbehorende Sleutelveld in de tabel Gebruikers de records uit Tabel1 met dat sleutelveld niet opgehaald worden. Alle onderliggende gegevens in Tabel2 en Tabel3 zijn daarmee ook afgeschermd.

Let op Als van deze methode gebruik gemaakt wordt, moeten ALLE onderwerpen op deze manier zijn ingericht. Als dit niet het geval is, kunnen de gegevens mogelijk via een andere combinatie van onderwerpen alsnog opgehaald worden.

Om de inner join op gebruikersnaam mogelijk te maken, moet bij het uitvoeren van een query op dit onderwerp wel de gebruikersnaam ingevoegd kunnen worden. Hiervoor moeten twee zaken geregeld worden:

- Gebruikersnaam moet beschikbaar zijn in de sessie van de AppServer
- Er moet een functie zijn die de gebruikersnaam dynamisch invult in de query

In de volgende paragrafen wordt de uitwerking hiervan gedaan.

3.5.1 Gebruikersnaam beschikbaar in de sessie van de AppServer

Als eerste moet de AppServer die de query gaat uitvoeren de gebruikersnaam kennen. Hiervoor kan de connect-procedure die in 3.2.2 gebruikt worden. Door deze procedure met 1 regel uit te breiden, wordt de gebruikersnaam in de session van de AppServer bewaard. Zie Figuur 7.

```
DEFINE INPUT  PARAMETER cUsername      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cPassword      AS CHARACTER   NO-UNDO.
DEFINE INPUT  PARAMETER cAppServerInfo AS CHARACTER   NO-UNDO.

IF NOT ValidateUser(cUsername, cPassword) THEN RETURN ERROR.

SESSION:SERVER-CONNECTION-CONTEXT = cUsername.
```

Figuur 7 – Connect-procedure uitgebreid met opslaan van gebruikersnaam in AppServer sessie

Merk op dat hiervoor niet per se de inlogprocedure in DataPA ingeregeld moet worden. De connect-procedure krijgt altijd de gebruikersnaam uit de DataPA context binnen. Als er niet is ingelogd op de DataPA connection, is dat dus de Windows gebruikersnaam.

Let op Mogelijk is er aanvullende logica in de connect-procedure nodig om de ontvangen DataPA gebruikersnaam om te zetten naar de bijbehorende gebruikersnaam die bekend is in de business applicatie tabel “Gebruikers”. Dat is voor het overzicht hier achterwege gelaten.

3.5.2 Functie voor invullen van gebruikersnaam in de query

Om de gebruikersnaam die nu in de AppServer sessie bekend is, te ‘injecteren’ in de query die daadwerkelijk op de database wordt uitgevoerd, is een dynamische functie nodig. Deze moet ingeregeld worden op de AppServer en gebruikt worden in de relatie tussen de tabellen Tabel1 en Gebruikers die gebruikt worden in het onderwerp.

Als eerste moet een functie geschreven worden die de gebruikersnaam uit de sessie retourneert. Zie Figuur 8. Deze functie moet op de AppServer geplaatst worden. In dit voorbeeld wordt de functie opgeslagen als “startup_functions.p”.

```
FUNCTION GetCurrentUser RETURNS CHARACTER:

    RETURN SESSION:SERVER-CONNECTION-CONTEXT.

END FUNCTION.
```

Figuur 8 – functie definitie voor het ophalen van de gebruikersnaam

In de startup-procedure van de AppServer moet de functie persistent geladen worden, zodat DataPA de functie kent en kan gebruiken. Dit gebeurt op de gebruikelijke wijze in startup.p. Zie Figuur 9.

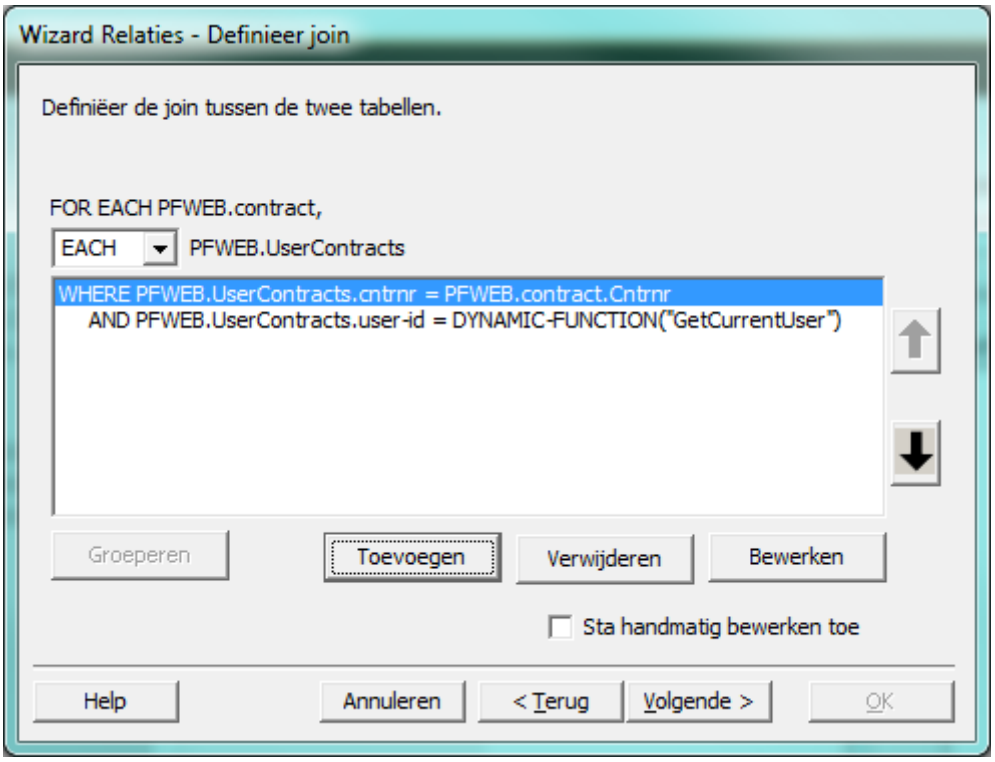
```
DEFINE INPUT  PARAMETER cStartup AS CHARACTER   NO-UNDO.

DEFINE VARIABLE hProc AS HANDLE      NO-UNDO.

RUN startup_functions.p PERSISTENT SET hProc.
SESSION:ADD-SUPER-PROCEDURE(hProc).
```

Figuur 9 – Startup-procedure

Als laatste moet de INNER JOIN tussen de hoofdtabel (Tabel1 in ons voorbeeld) en de tabel met gebruikers (Gebruikers) gemaakt worden. Zie Figuur 10.



Figuur 10 – Join tussen hoofdtabel en gebruikerstabel

Zodra dit is ingeregeld en opgeslagen en de AppServer is herstart, krijgt een gebruiker alleen nog de records in het resultaat waarvan de sleutel via de tabel Gebruikers is gekoppeld aan zijn gebruikersnaam.

NB Bovenstaande geldt niet alleen voor DataPA gebruikers met de rol “gebruiker”, maar voor iedereen, dus ook beheerders. Mogelijk moeten de rechten van beheerders voor beheer- en testdoeleinden in de business applicatie uitgebreid worden om überhaupt resultaten van queries te krijgen.

Dit geldt voor alle mogelijkheden die in dit document zijn beschreven.