



WERKGROEP GAP		387
18 mei 2010	Not.	10

Aanwezig: Koen Meersman, Broes De Cat, Tom Haepers, Tim Mallezie, Bart Boone, Johan Vervloet

Verontschuldigd: Peter Bertels

Programmeerweekend praktisch

- Waar
 - Heibrand: We programmeren in 'De Vliering' en slapen in 'De Lagune'.
- Wanneer
 - Iedereen welkom zaterdagmorgen vanaf 9.00 uur.
 - We stoppen zondagmiddag om 18.00 uur.
- Meebrengen: de typische Heibranddingen
 - Slaapzak
 - Onderlaken
 - Kussensloop
 - 'weekendgerief' voor een kort weekend
- Ondertussen geverifieerd: er is Trappist bij de laver :-)

Korte inleiding Kipadmin

Om een beetje een zicht te krijgen op wat er gesynchroniseerd moet worden tussen Kipadmin en GAP: een overzicht van hoe de Kipadmin database qua leden in elkaar zit.¹²

Er zit weinig tot geen lijn in de namen van tabellen en velden. Dat is wat historisch gegroeid. Je kan er wel een beetje uit afleiden

- De tabelnamen volledig in hoofdletters zijn de tabellen van 5 jaar geleden.
- De tabelnamen met het prefix 'kip' zijn tabellen waar ik een werkjaar aan heb toegevoegd, of die genormaliseerd zijn
- De tabelnamen in kleine letters zonder 'kip' en met een schema, zijn namen na een 2^{de} normalisatieronde, en zouden min of meer de SQL-coding standaarden van GAP moeten volgen. Van die 2^{de} normalisatieronde is uiteindelijk niet veel in huis gekomen.

Wat in het schema doorstreept is, is niet relevant voor GAP. Sommige zaken zijn ook overbodig in Kipadmin, maar omdat de client zodanig hard gekoppeld is aan de database, is het moeilijk die te verwijderen.

Personen

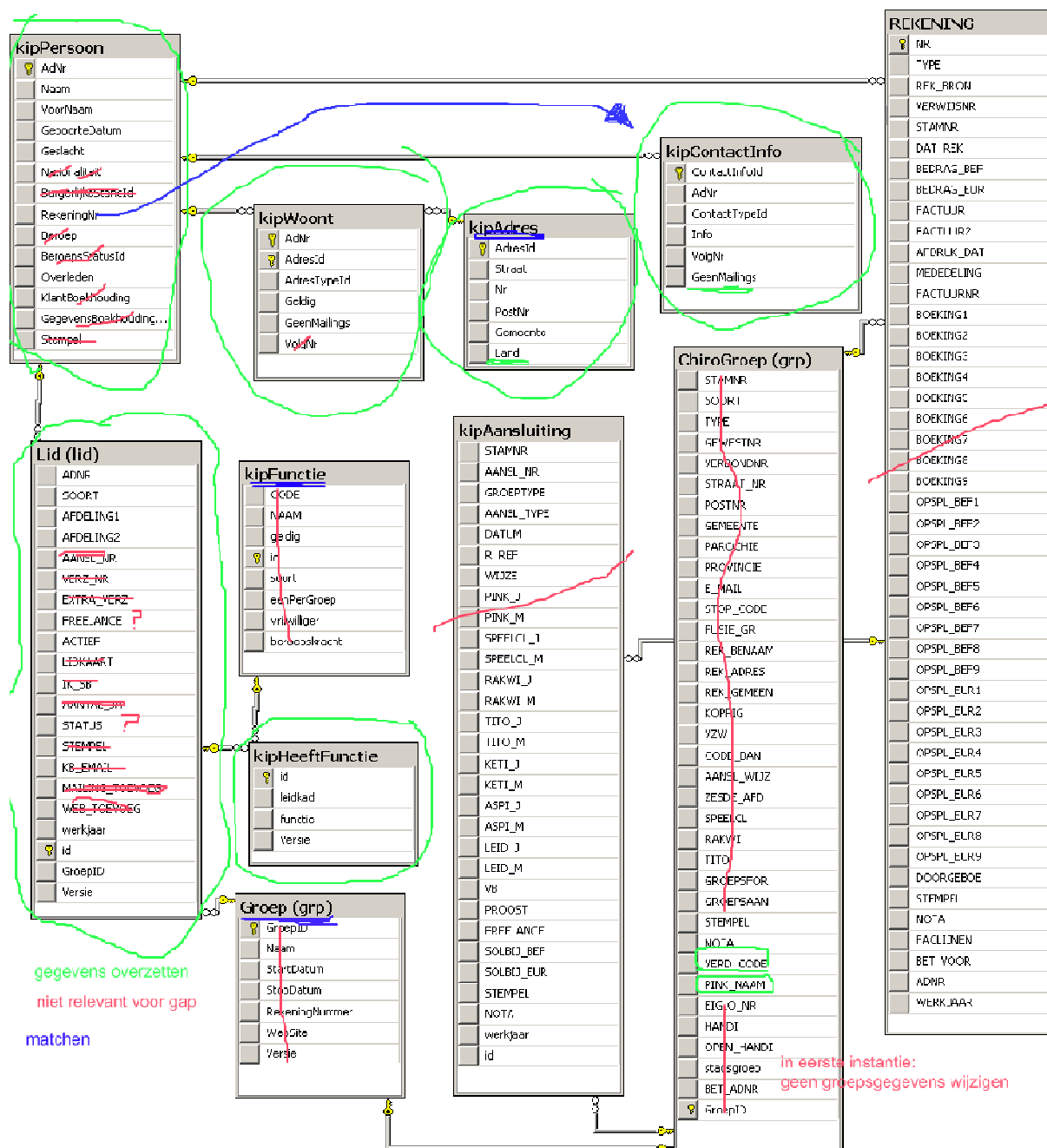
De tabel 'kipPersoon' bevat de personen. Net zoals in GAP heeft een persoon mogelijk meerdere adressen, en meerdere communicatievormen. De adressen zijn net genormaliseerd: straat, postnummer, en gemeente zijn gewoon strings.

Leden en groepen

Keti's, aspi's, leiding en kaders zitten allemaal als persoon in de database, en via de tabel lid.Lid worden ze aan een groep (grp.Groep) gekoppeld. Net zoals in GAP is er een verschil tussen grp.Groep en grp.ChiroGroep: er wordt al bijgehouden wie aansluit bij welke oud-leidingsploeg; oud-leidingsploegen zijn geen groepen, maar 'satellieten'. (De tabel 'satelliet' is niet weergegeven op de tekening, maar bestaat wel.)

¹ De applicatie werkt in praktijk niet rechtstreeks op deze tabellen, maar voornamelijk op views die er uitzien als de tabellen van 5 jaar geleden. Bijgevolg zijn lang niet alle 'features' van de database beschikbaar via het programma. Sommige views moeten hiervoor lijnen concatenen tot 1 string, wat de client heel traag maakt.

² Blijkbaar had ik de laatste versie van het schema niet bewaard? Ik heb het dan maar aangevuld op basis van wat ik mij herinner.



Iedere keer een groep een aansluiting doorstuurt (met het oude programma), komt er een lijn bij in de tabel 'kipAansluiting', die voornamelijk tellingen bevat. In het record 'lid.Lid' van een lid, geeft het veld 'aansl_nr' het nummer van de aansluiting weer via dewelke het lid aangesloten is.³

Bij het begin van een nieuw werkjaar worden alle leden van het vorige werkjaar opnieuw lid, maar met 'aansl_nr' 0. De filosofie daarachter is dat we ervan uitgaan dat iedereen lid blijft, tot de eerste aansluiting binnenkomt: dan weten we we daadwerkelijk aangesloten is.⁴

Bij elke aansluiting hoort een rekening (die aan de basis ligt van een factuur). Rekeningen kunnen gekoppeld zijn aan een persoon of aan een groep (via de velden AdNr of StamNr); voor een aansluiting is de rekening aan de groep gekoppeld.

³ De nummering is per groep en per werkjaar. De eerste aansluiting van een groep in een bepaald werkjaar heeft dus altijd nummer 1.

⁴ Met het nieuwe programma kan dit misschien gebruikt worden voor de leden waarvan de probeerperiode nog niet voorbij is?

Communicatie GAP-Kipadmin

Soorten communicatie

1. Initiele import. Gegevens worden uit het oude Chirogroepprogramma en uit Kipadmin geïmporteerd in GAP. Dit is zo goed als rond.
2. Uitwisseling gegevens GAP <-> Kipadmin wanneer GAP operationeel is. In eerste instantie werken we voornamelijk aan de richting GAP->Kipadmin (in de andere richting gaan enkel ad-nummers). We willen hiervoor message queueing gebruiken.

Message queueing

Wanneer GAP iets te vertellen heeft aan Kipadmin, stuurt die een bericht via een message queue naar een service. De service leest de berichten uit de queue's, en voert de nodige aanpassingen uit in Kipadmin.

Kort over queueing:

- Zit standaard in Windows; te activeren via 'add windows components'
- Als de service om een of andere reden niet draait, blijven de berichten in de queue.
- Als het afhandelen van een bericht niet lukt, wordt de transaction 'gerollbackt', en komt de message terug in de queue terecht.
- Er is een systeem van retry queue en bad queue.

Praktisch

We gebruiken WCF voor message queueing. Dat is iets beperkter dan 'native' message queueing, maar met WCF hebben we al ervaring.

Het stuk aan de kipadminkant komt sowieso op de takenlijst van Johan.

TODO's

- Functioneel: welke use-cases geven aanleiding tot een bericht van GAP? Wat komt er terug?
- Experimenteren met message queue's
- Opzoeken: hoe kan Delphi communiceren met andere processen? (COM?) (Dit wordt pas relevant voor communicate Kipadmin->Gap)