

Klaas Brant over IBM's DB2 Universal Database

# U bent natuurlijk nooit zomaar een nummer...

Heeft u ooit wel eens het gevoel gehad dat u zomaar een nummer bent? Dan zit u er ver naast. Onze wereld hangt van nummers aan elkaar: sof-nummers, polisnummers, klantnummers, factuurnummers om er maar een paar te noemen. Sommige nummers zijn er niet zomaar, er is over nagedacht.

Van een sof-nummer kun je zo zien of deze wel geldig is. Maar veel nummers zijn puur sequentieel; soms is dat wettelijk voorgeschreven zoals bij factuurnummers. Bij die sequentiële nummers beginnen de problemen pas echt. Men kan er hier twee methodes op nahouden: zelf generen of dat door de databases laten doen.

Zelf generen heeft voordelen maar vaak een enorm nadeel. Het algoritme om de nummers te generen heeft meestal locking-problemen tot gevolg, in ieder geval wel bij DB2. Ergens in de database ligt in een record opgeslagen dat het hoogste tot nu toe uitgeven nummer bevat en daar wordt het lastig; iedereen wil dit record lezen én updaten! In boeken vol hints en tips komt het er meestal op neer dat het pakken en wegwezen is. De meeste applicaties voelen niks voor een tussentijdse commit. Het alternatief is om alle *updates* en *inserts* zo veel mogelijk uit te stellen. Aan het einde van het algoritme, vlak voor de commit, moet alles zo kort mogelijk na elkaar komen. Niet echt de ideale manier van programmeren, dus toch maar de nummering aan de database over laten?

DB2 nummers laten generen kan op twee manieren. De eerste methode is een identity column, een extra kenmerk dat men kan toevoegen aan één numerieke column van een table, bijvoorbeeld startwaarde, eindwaarde en increments. Het idee is dat de database bij iedere insert een nieuw nummer toekent aan deze column, maar dit is niet ideaal. De methode is niet meer uit te schakelen en er zijn toch echt momenten dat dit gewenst is. Bijvoorbeeld als de table-definities aangepast moeten worden, is er een *drop en create* nodig. Standaard wordt een script als *unload, aanpassen en reload* toegepast en daar ontstaan problemen. Als bij de unload meneer



Janssen probleem 10345 had, dan wilt u natuurlijk dat hij dat na de reload nog steeds heeft. Mooi niet zegt DB2, "ik ga over de nummering" en begint na een drop bij een reload vrolijk opnieuw te nummeren. Als identity columns het niet zijn, dan gaan we kijken

naar de alternatieve methode in DB2; sequences. Mainframe-gebruikers moeten wachten op Z/OS v8.

De gebruikers op de "kleinere" platforms Windows, Linux en Unix hebben deze variant al wel tot hun beschikking. Het komt er op neer dat de database ergens een nummer bijhoudt, maar dat dit nummer niet gekoppeld is aan een table.

Een sequence is dus een nieuw soort object in de database. Zo hoort het natuurlijk ook. Het feit dat u een nieuw factuurnummer nodig heeft in een applicatie, wil nog niet zeggen dat u in de database wil inserten. Omdat de constraints nu niet meer op de tabellen liggen kunt u hiermee doen en laten wat u wilt. Met speciale SQL-constructies kan men nog steeds direct een nieuwe waarde in de database inserten en wordt eigenlijk een soortgelijke functionaliteit bereikt als met de identity column. De sequences vormen dus een veel elegantere oplossing die veel van de identity-problemen weet op te lossen. Het is dan ook de nieuwe standaard voor sequentiële nummering, tenzij er speciale eisen zijn. Is de nummering niet sequentieel dan kunnen we alleen maar terugvallen op eigen methodes van genereren.

Dus mocht u ooit nog eens denken dat u maar een nummer bent, dan is het op zijn minst een nummer dat of met de nodige problemen is gegenereerd of met een slim algoritme is gegenereerd of het gevolg is van een goed database-ontwerp. Waarschijnlijk zult u het nooit weten...

Klaas Brant (kbrant@kbce.nl) is DB2-specialist en directeur van KBCE b.v. Meer informatie over DB2 is te vinden op [www.kbce.nl](http://www.kbce.nl) en [www.db2-times.com](http://www.db2-times.com)