

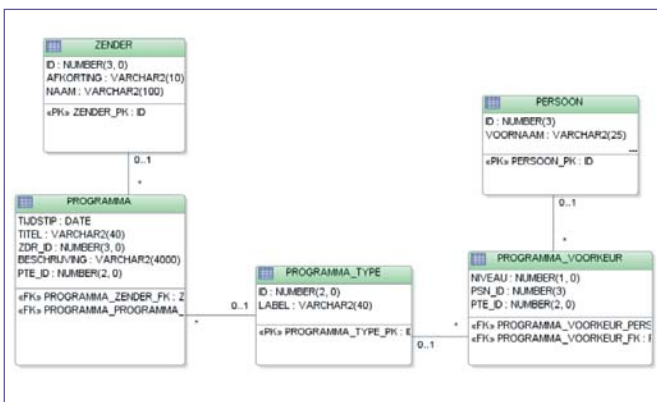
De ideale televisie-avond van Madelon

Als de kat van huis is... Op avonden dat zij alleen thuis is (dan in ieder geval) kijkt Madelon graag naar de televisie. Een uitdaging daarbij is het samenstellen van de ideale televisieavond. Waar kan ze het beste naar gaan kijken opdat haar televisie-genot optimaal is?

Hoe moeilijk is het niet om te kiezen wat je gaat kijken op televisie? Als je alleen maar naar de dertien meest voor de hand liggende kanalen kijkt, dan is er keuze uit maar liefst 162 verschillende programma's. Je kunt er hooguit vijf procent van kijken. Hoe maak je die keuze?

In deze puzzel gaan we kijken hoe we een SQL Query kunnen ontwikkelen die de keuze voor ons maakt. Wij geven op wat onze programma-smaak is, en de query bepaalt de ideale televisie-avond. Een tip voor John de Mol: misschien een idee voor een leuke nieuwe show, 'SQL with the Stars'?

Het datamodel voor deze puzzel bestaat uit een vijftal tabellen (zie afbeelding). Belangrijk om te weten: de voorkeur ligt vast in de tabel PROGRAMMA_VOORKEUR op een schaal van 0 tot en met 5. Hierbij is 0 totaal niet interessant en is 5 het summum van televisiekijken.



Het datamodel voor deze puzzel bestaat uit een vijftal tabellen.

Uitdagingen

1. Schrijf een query die laat zien wat het aanbod is op Nederland 1 vanavond.

```
select to_char( pga.tijdstip, 'HH24:MI' ) begintijd
      , pga.titel
      , pga.beschrijving
from programma pga
inner
join zender zdr
  on (pga.zdr_id = zdr.id)
where zdr.afkorting = 'NED 1'
      and trunc( pga.tijdstip ) = to_date( '13-02-2007' , 'DD-MM-YYYY'
      )
order by pga.tijdstip
```

Deze eerste query is redelijk eenvoudig, het enige wat opvalt, is de ANSI-join syntax. Met name de ANSI OUTER JOIN maken het schrijven van queries een stuk eenvoudiger. Door gebruik te maken van deze ANSI-join syntax krijg je ook een duidelijke scheiding tussen Join-condities en de filtervoorwaarden. Een ander voorbeeld van eenvoudiger syntax is de FULL OUTER JOIN. Voorheen was het alleen met een UNION en een tweetal SELECT statements mogelijk om de data van twee datasets te bekijken, een (+) mag maar op één tabel worden gebruikt. Met een FULL OUTER JOIN is dit een stuk eenvoudiger, er is slechts één SELECT statement voor nodig.

2. Welke programma's zijn in zijn geheel (start en eind tussen 21.00 en 22.00 uur) te zien tussen 21.00 en 22.00 uur?

Om deze vraag te kunnen beantwoorden, moet je in eerste instantie weten wat de begin- en eindtijden van de programma's zijn. In de PROGRAMMA-tabel staan echter alleen maar de begintijden. De begintijd van het ene programma is de eindtijd van het vorige programma. De begin- en eindtijden kunnen bepaald worden door gebruik te maken van de Analytische Functie LEAD:

```
Lead (pga.tijdstip, 1) over (partition by pga.zdr_id order by pga.tijd-
stip) eindtijd
```

Met LEAD kunnen we 'vooruit kijken' in de resultaatset, we kunnen informatie verkrijgen van rijen die nog moeten komen, die volgens de ORDER BY later komen dan de huidige rij. Allereerst bepaal je waarnaar je wilt kijken, in ons geval het tijdstip. Dit is het eerste argument. Als tweede argument kun je aangeven hoeveel rijen je terug wilt kijken. Aangezien wij alleen geïnteresseerd zijn in de vorige rij is dit 1. Dit argument is optioneel. Er is nog een derde optioneel argument waar je een default waarde kan opgeven voor het geval er geen vorige rij is. De PARTITION clause geeft aan hoe de result set is opgedeeld, in ons geval per zender. En de ORDER BY clause geeft, niet verrassend, hoe de sortering is binnen de PARTITION. Nu de begin- en eindtijden van de programma's bekend zijn, is de query eenvoudig te schrijven:

```
select begintijd
, to_char (eindtijd, 'HH24:MI') eindtijd
, zdr.afkorting zender
, titel programma
from (select to_char (pga.tijdstip, 'HH24:MI') begintijd
, lead (pga.tijdstip, 1) over (partition by pga.zdr_id
order by pga.tijdstip
) eindtijd
, pga.titel
, pga.zdr_id
from programma pga
where pga.tijdstip >= to_date('13-02-2007 21:00', 'DD-MM-YYYY
HH24:MI')
)
, zender zdr
where eindtijd <= to_date('13-02-2007 22:00', 'DD-MM-YYYY HH24:MI')
and zdr.id = zdr_id
order by begintijd, eindtijd;
```

BEGIN	EINDT	ZENDER	PROGRAMMA
21:00	21:30	BBC 2	Dr Alice Roberts: Don't Die Young
21:00	22:00	BBC 1	Holby City : I Feel Pretty
21:05	21:45	NED 1	Bureau Van den Heuvel:
21:20	22:00	NED 2	Het elfde uur:
21:30	22:00	BBC 2	Bill Oddie Back in the USA : Arizona
21:30	22:00	NED 3	PREMtime:

Wat is het aanbod van speelfilms vanavond?

Hoewel er alleen naar de speelfilm wordt gevraagd, hebben we ook de begin- en eindtijd bij gezet. Ook hier is er gebruik gemaakt van de Analytische functie LEAD. De ANSI-join syntax is nu inmiddels een oude bekende.

```
select *
from (select to_char (pga.tijdstip, 'HH24:MI') begintijd
, to_char (lead (pga.tijdstip,1) over (partition by pga.zdr_id
order by tijdstip
)
, 'HH24:MI') eindtijd
, zdr.afkorting zender
, pga.titel
```

```
, pte.label programma_type
from programma pga
join zender zdr
on (pga.zdr_id = zdr.id)
join programma_type pte
on (pte.id = pga.pte_id)
where trunc(pga.tijdstip) = to_date('13-02-2007', 'DD-MM-YYYY')
)
where programma_type = 'Speelfilm'
order by begintijd
```

BEGIN	EINDT	ZENDER	TITEL	PROGRAMMA_TYPE
21:35	23:05	Veronica	Loaded Weapon.	Speelfilm

Natuurlijk zijn er nog vele andere vragen te verzinnen, zoals het aantal nieuwsprogramma's, per zender en in het totaal. Of "Wat zijn de langste programma's van de avond?" En natuurlijk: "Hoe ziet de ideale televisieavond voor Madelon eruit, gezien haar voorkeuren?"

Antwoorden op deze vragen, inclusief de scripts voor het creëren en met data vullen van de tabellen, kun je vinden op de weblog van AMIS: <http://technology.amis.nl/blog/?p=2045>.

Alex Nuijten is als consultant werkzaam bij AMIS.