

Alternatieven voor reporting- en windowingfuncties

Max

Met aggregaatsfunctie count (en self-join):

```
SELECT t1.hasc,t1.iso,t1.gebruik
select x.name,x.weight
from competitors x
join competitors y on y.weight>=x.weight
group by x.name,x.weight
having count(1)=1
```

Met minus (en self-join):

```
select name,weight
from competitors
where weight is not null
minus
SELECT x.name, x.weight
from competitors x
join competitors x2 on x2.weight>x.weight
where x.weight is not null
group by x.name,x.weight
order by 2 desc
```

Met set operatoren (en self-join):

```
with x as (select name,weight from Competitors
          WHERE gender='M' and weight is not null
          )
SELECT name, weight
FROM x
where name not in (
    select x.name from x
    join x x2 on x2.weight >x.weight)
```

Mediaan

Met de ranking-, reporting- en windowingfuncties count en row_number

Gemeente met middelste hoogte (19 gemeentes: 10^e, 20 gemeentes: 10^e en 11^e)

```
select coalesce (lev2, lev1), elevation
,2*row_number() over(partition by coalesce (lev2, lev1) order by elevation)
-1-count(1) over(partition by coalesce (lev2, lev1) order by elevation)
,case when 2*row_number() over(partition by coalesce (lev2, lev1) order by elevation) -1-count(1)
over(partition by coalesce (lev2, lev1) order by elevation) between -1 and 1 then 'x' else '' end
from cities
```

where iso = 'BE' and elevation is not null
order by coalesce(lev2,lev1),elevation;

Met autojoin en aggregaatsfunctie count

```
select x.gender, x.weight MEDIAAN
from Competitors x
join Competitors y on y.gender=x.gender
where x.weight is not null
  and y.weight is not null
group by x.weight, x.gender
having abs(count (case when y.weight > x.weight then 1 end)
  - count (case when y.weight < x.weight then 1 end))
  <= count (case when y.weight = x.weight then 1 end)
```

Nota: de counts kunnen vervangen worden door sum(case .. then 1 else 0 end) om null waarden te vermijden.

Met subqueries

```
select gender, weight MEDIAAN
from Competitors x
where weight is not null
group by weight,gender
having abs((select count(weight) from Competitors
  where weight>x.weight and gender=x.gender)
  -(select count(weight) from Competitors
  where weight < x.weight and gender=x.gender))
  <= (select count(weight) from Competitors
  where weight=x.weight and gender=x.gender)
```

Met de windowing- en rankingfunctie row_number en aggregaatsfunctie count

```
Select elevation, 2*row_number() over(order by elevation) -1-count(1)over()
from cities
where iso='BE' and elevation is not null
Group by x.elevation
```

Alternatieven voor rankingfuncties

Rank

Met join en count:

Top 3 per discipline en gender:

```
select r1.discipline,r1.gender,r1.name,r1.points,count(case when r2.points>r1.points then 1 end)+1
from ranking r1
```

```

join ranking r2 on r1.discipline=r2.discipline and r1.season=r2.season and r1.gender=r2.gender and
r2.points>=r1.points
where r1.season=1990
group by r1.discipline,r1.gender,r1.name,r1.points
having count(case when r2.points>r1.points then 1 end)<=3
order by discipline,gender,points desc

```

Met reportingsfunctie en aggregaatsfunctie:

```

select gender, season, name, sum(points)
, case when max(sum(points)) over(partition by gender, season)= sum(points) then 'X' else '' end
from ranking where season > 1991 and discipline is not null
group by gender, season, name
order by gender, season, name, sum(points) desc;

```

Top N problem

Top 3 met selfjoin en aggregaatsfunctie count

```

select x.season, x.discipline, x.name, x.points, count(case when y.points>x.points then 1 end)
from ranking x
join ranking y on x.season=y.season and x.discipline= y.discipline and x.gender = y.gender
where x.gender = 'L' and x.discipline is not null
having count(case when y.points>x.points then 1 end)<4

```

Top 2 met CTE, aggregaatsfunctie sum en set operator not in

```

with x as (
    select name, sum(points) tel
    from ranking
    where season =2006 and gender='M'
    group by name
)
select name, tel
from x
where name not in (
    select a.name
    from x a
    join x b on b.tel>a.tel
    join x c on c.tel>b.tel
)
order by 2 desc

```

Nota: niet performant

Enkel de 3e plaats, mbv vorige query

```

with x as (
    select name, sum(points) tel
    from ranking
    where season =2006 and gender='M'
    group by name
)
select name, tel
from x
where name in (
    select a.name
    from x a
    join x b on b.tel > a.tel
    join x c on c.tel > b.tel
    where a.name not in (
        select z.name
        from x z
        join x f on f.tel > z.tel
        join x g on g.tel > f.tel
        join x h on h.tel > g.tel
    )
)
order by 2 desc

```

Nota: zeker niet performant 😊

Varia

Toon de opeenvolgende races van de skiërs, zonder analytische functies!

```

with x as (
    select re.name naam,ra.rid,ra.racedate
    from results re
    join races ra on re.rid = ra.rid
    where re.rank = 1
)
select x1.naam naam
,x1.racedate date1,x1.rid rid1
,x2.racedate date2,x2.rid rid2
,x2.racedate-x1.racedate verschil
from x x1
    join x x2 on x1.naam = x2.naam and x2.racedate> x1.racedate
    left join x x3 on x1.naam = x3.naam and x3.racedate between x1.racedate+1 and X2.racedate-1
where x3.racedate is null

```

Intersect simuleren met Join

```
Select name, r.elevation, r.number, avg(c.elevation), count(1)
```

From regio's r

Join cities c on name = coalesce(lev2,lev1) and iso = 'BE '

where substring(hasc,1,2)= 'BE 'and level=2

group by name, r.elevation, r.number

having r.elevation <> avg(c.elevation) or number<> count(1) –deze having is optioneel

nota: geen idee wat hier zo speciaal aan is en welke intersect query deze zou simuleren