

# Algemene Richtlijnen Optimalisaties

[wim.bertels@ucll.be](mailto:wim.bertels@ucll.be)

Naamsvermelding-NietCommercieel-GelijkDelen 4.0  
Unported Licentie

# 1. Inleiding

- Optimiser – Beste uitvoeringsplan?
  - Indexen – Query snelheid verhogen?
  - Inzicht – Wat gebeurt er intern?
- 
- Maar soms niet optimaal
- => Herformulering om tot een efficiënte verwerkingsstrategie te komen !

## 2. Vermijd de OR-operator

- OR : index wordt meestal niet gebruikt
- Alternatief (indien mogelijk) :
  - Vervangen door een conditie met IN
  - Vervangen door 2 selects met UNION
- Vb.

...  
where spelersnr = 15  
or       spelersnr = 29  
or       spelersnr = 55

=>   where spelersnr in (15, 29, 55)

### 3. Onnodig gebruik van UNION

- UNION : dezelfde tabel meerdere malen doorlopen
- Alternatief (indien mogelijk) :
  - Herformuleren waarbij alle voorwaarden in één select instructie geplaatst worden

## 4. Vermijd de NOT-operator

- NOT : index wordt niet gebruikt
- Alternatief (indien mogelijk) :
  - Vervang NOT door vergelijkingsoperatoren
- Vb.

...

where not (jaartoe > 1980)

## 4. Vermijd de NOT-operator

- Oplossing:

...

where not (jaartoe > 1980)

=> where jaartoe <= 1980

## 5. Isoleer kolommen in condities

- Kolom in een berekening of in een scalaire functie : index wordt niet gebruikt

- Alternatief (indien mogelijk) :

- Isoleer de kolom

- Vb.

...

where jaartoe + 10 = 1990

=> where jaartoe = 1980

## 6. Gebruik de BETWEEN-operator

- AND : gebruikt de index meestal niet
- Vb.

...

where jaartoe  $\geq$  1985  
and jaartoe  $\leq$  1990



## 6. Gebruik de BETWEEN-operator

- AND : gebruikt de index meestal niet
- Oplossing :
  - Gebruik van BETWEEN
- Vb.

...

where     jaartoe >= 1985  
and        jaartoe <= 1990

=>    where jaartoe between 1985 and 1990

# 7. Bepaalde vormen van LIKE-operator

- LIKE : index wordt niet gebruikt als patroon begint met % of \_

- Alternatief :

–Geen, tenzij..

- Vb.

...

where naam like '%sen'

=> ???

## 8. Redundante condities bij joins

- Redundante condities : om SQL te verplichten om een bepaald pad te kiezen
- Vb.

...

where boetes.spelersnr = spelers.spelersnr  
and boetes.spelersnr = 44

=>where boetes.spelersnr = spelers.pelersnr  
and boetes.spelersnr = 44  
and spelers.spelersnr = 44

## 9. Vermijd de HAVING-component

- Conditie in HAVING : index wordt niet gebruikt
- Alternatief (indien mogelijk)
  - Zoveel mogelijk condities in WHERE
- Vb. --hoort dit thuis in de having?

...

```
group by spelersnr  
having   spelersnr >= 40
```

```
=> where   spelersnr >= 40  
    group by spelersnr
```

## 10. SELECT-component : compact

- SELECT-component zo compact mogelijk
  - Onnodige kolommen weglaten uit SELECT
  - Bij gecorreleerde subquery met exists : één expressie bestaande uit één constante
- Vb.

```
select spelersnr, naam
from   spelers
where  exists (select  '1'
                  from    boetes
                  where    boetes.spelersnr =
                          spelers.spelersnr)
```

# 11. Vermijd DISTINCT

- DISTINCT : verwerkingstijd verlengd
- Alternatief (indien mogelijk)
  - Vermijden als het overbodig is
- Vb.

```
select distinct wedstrijdnr, naam  
from  wedstrijden, spelers  
where wedstrijden.spelersnr =  
        spelers.spelersnr
```

```
=> select wedstrijdnr, naam
```

## 12. ALL-optie bij set operatoren

- Zonder ALL : verwerkingstijd verlengd
- Data moeten gesorteerd worden om dubbels eruit te halen
- Vb.

```
Select  naam, voorletters
from    spelers
where   spelersnr = 10
union all
select  naam, voorletters
from    spelers
where   spelersnr = 18
```

# 13. Kies outer-joins boven UNION

- UNION : verwerkingstijd verlengd
- Alternatief (indien mogelijk)
  - Outer-join is beter



- Vb.

```
select    spelers.spelersnr, naam, bedrag
from      spelers, boetes
where     spelers.spelersnr = boetes.spelersnr
union
select    spelersnr, naam, null
from      spelers
where     spelersnr not in (select spelersnr
                             from boetes)
order by  1
```

- Vb. Oplossing

```
select    spelers.spelersnr, naam, bedrag
from      spelers, boetes
where     spelers.spelersnr = boetes.spelersnr
union
select    spelersnr, naam, null
from      spelers
where     spelersnr not in (select spelersnr
                           from boetes)
order by 1
```

```
=> select spelersnr, naam, bedrag
      from      spelers left outer join boetes
              using (spelersnr)
      order by 1
```

# 14. Vermijd datatype-conversies

- Converteren van datatypes : verwerkingstijd verlengd
- Alternatief (indien mogelijk) :
  - Datatype-conversie vermijden
- Vb.

```
select      *  
from        spelers  
where       spelersnr = '15'
```

=> where spelersnr = 15

# 15. Volgorde tabellen

- Volgorde van tabellen kan belangrijk zijn
- Afhankelijk van de juistheid van de interne statistieken
- Vb.

```
select  spelers.spelersnr, naam, teamnr
from    spelers, teams
where   spelers.spelersnr = teams.spelersnr
```

```
<=>    select      spelers.spelersnr, naam, teamnr
        from        teams, spelers
        where        spelers.spelersnr =
                    teams.spelersnr
```

# 16. Vermijd ANY- en ALL-operatoren

- ANY en ALL : index wordt niet gebruikt
- Alternatief (indien mogelijk)

—?

- Vb.

```
Select      spelersnr, naam, geb_datum
from        spelers
where       geb_datum <= all (select geb_datum
                             from spelers)
```

```
=> select spelersnr, naam, geb_datum  
    from spelers  
    where geb_datum = (select min(geb_datum)  
                       from spelers)
```

Vervang door min of max

# Hoe nakijken

- Explain
- Explain analyze