

Indexen

wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0
Unported Licentie

INDEX

- Doel: beïnvloeden van de verwerkingstijd!



INDEX

- Doel: beïnvloeden van de verwerkingstijd!

OS:

- Rijen worden in bestanden opgeslagen
- Een bestand bestaat uit pagina's
- Als een rij opgehaald wordt :
 - de betreffende pagina wordt opgehaald
 - de betreffende rij wordt opgehaald

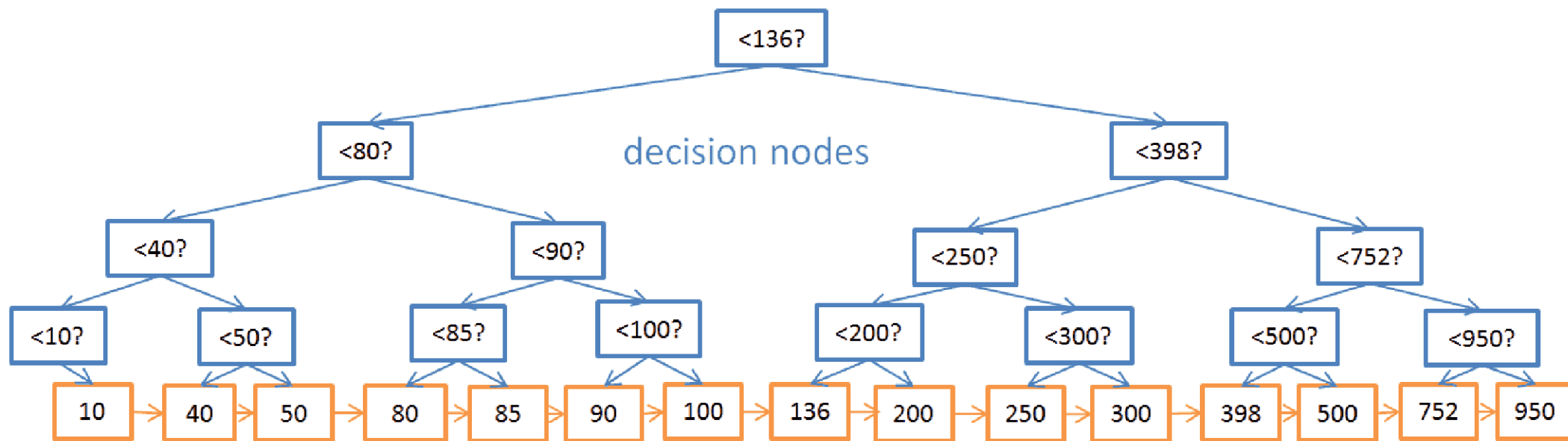
Werking van een INDEX

- 2 methodes voor het opzoeken:
 - Sequentiële zoekmethode: rij voor rij
 - Tijdrovend en inefficiënt
- Geïndexeerde zoekmethode: index (B-tree)
 - Boom
 - Knooppunten
 - Leafpage (bevat referentie naar pagina+rij)
 - 2 methodes:
 - Zoeken van rijen met een bepaalde waarde
 - Doorlopen van de hele tabel via een gesorteerde kolom (geclusterde index)

B-tree (default)

- Binaire zoek boom (node kan 2+leaves hebben)
- Best voor $> < =$ operatoren

B+ Tree / Database index

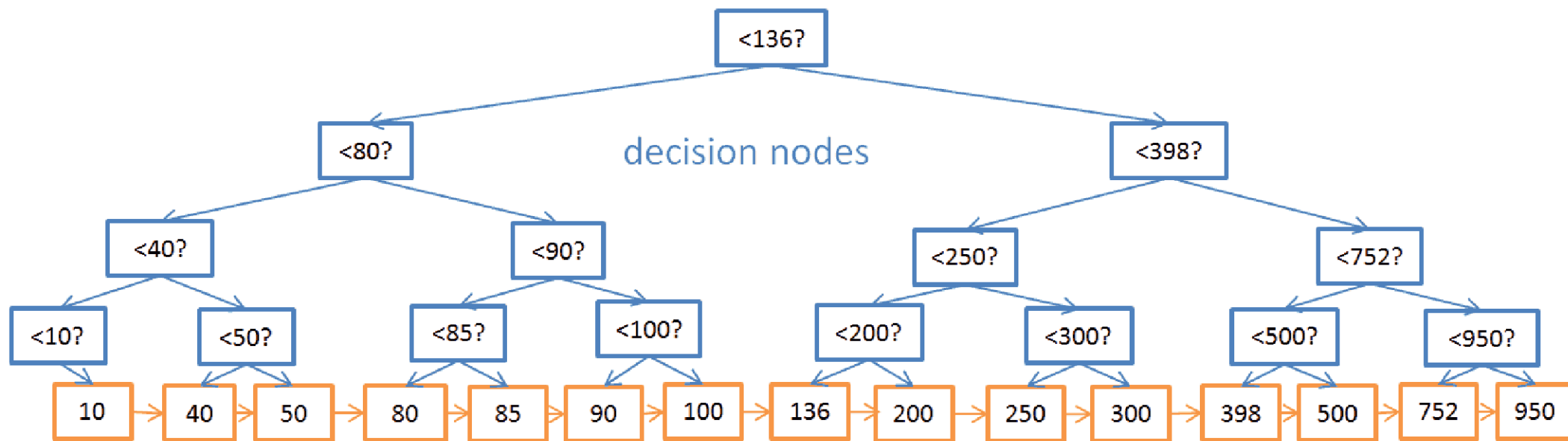


nodes with a pointer to a row in the associated table
they also have a link to their successor in the B+ Tree

B-tree (voorbeeld)

- Sequentiel: max 15 stappen
- Via boom: max 4 stappen, namelijk $\lceil \log_2 15 \rceil + 1$

B+ Tree / Database index



nodes with a pointer to a row in the associated table
they also have a link to their successor in the B+ Tree

Werking van een INDEX

- Opmerkingen:
 - Wanneer tabel word aangepast: word index aangepast
 - Index: ook op niet-unieke kolom
 - Op één tabel:
 - meerdere indexen
 - één geclusterde index
 - Samengestelde index
- Opgelet:
 - Index neemt opslagruimte in beslag
 - Als index vol is: reorganisatie van de index
- Meerdere indexvormen zijn mogelijk

Planner / Optimiser

- Conceptueel: bv denken in lussen of geneste lessen bij joins, de volgorde van verwerking van een SELECT instructie
- Concreet: intern kan dit helemaal anders verwerkt worden
- SQL is een declaratieve (programmeer)taal, geen imperatieve (programmeer)taal.
 - Idee: Je zegt wat er (logisch) moet gebeuren, niet (expliciet) hoe het moet gebeuren.
 - 1 van de sterke punten van SQL
- Redeneer conceptueel en groei

Optimiser

- Zoekt de beste strategie
 - Verwachte verwerkingstijd
 - Aantal rijen
 - Indexen
 - Interne statistieken
 - ..

CREATE INDEX

- Geen ANSI of ISO specificatie
- Vendor specificatie :
<https://www.postgresql.org/docs/current/sql-createindex.html>

CREATE INDEX

- PostgreSQL:

```
CREATE INDEX spelers_postcode_idx
    ON spelers (postcode asc);
CREATE UNIQUE INDEX spelers_naam_vl_idx
    ON spelers (naam, voorletters);
-- UNIQUE !
CREATE INDEX spelers_naam_vl_partial_idx
    ON spelers (naam, voorletters)
    WHERE spelersnr < 100;
CLUSTER spelers USING speler_naam_vl_idx;
```

DELETE/UPDATE

Wat is het effect van een geclusterde index?

REINDEX

- Vendor specificatie:

<https://www.postgresql.org/docs/current/sql-reindex.html>

- PostgreSQL:

- `REINDEX INDEX een_index;`
- `REINDEX TABLE een_tabel;`
- `REINDEX DATABASE een_database;`

INDEX management

- CREATE
- ALTER
 - ALTER INDEX groot_idx
SET TABLESPACE ergens_anders;
- DROP
- Meeste SQL-producten :
 - automatische creatie van index op primaire en secundaire sleutels bij het maken van de tabel
 - naam wordt afgeleid uit de naam van de tabel en de betreffende kolommen

Wanneer INDEXeren?

- Index:
 - Voordeel: index versnelt verwerking
 - Nadeel:
 - index neemt opslagruimte
 - elke mutatie vraagt aanpassing van index
=> verwerking vertraagt

Welke kolommen?

- Richtlijnen voor keuze van kolommen :
 - Unieke index op kandidaatsleutels
 - Index op refererende sleutels
 - Index op kolommen waarop (veel) geselecteerd wordt
 - Grootte van de tabel
 - Kardinaliteit (verschillende waarden) van de tabel
 - Distributie (verdeling) van de waarden
 - Index op een combinatie van kolommen
 - Index op kolommen waarop gesorteerd wordt
 - ..

Speciale indexvormen

- Multi-tabelindex :
= index op kolommen in meerdere tabellen
- Virtuele-kolomindex :
= index op een expressie
- Selectieve index
= index op een selectie van de rijen
- Hash-index
= index op basis van het adres van de pagina
- Bitmapindex
= interessant als er veel dubbele waardes zijn

Nieuwere indexvormen: GIST en SP-GIST

- GiST (Generalized Search Tree)
 - gebalanceerd
 - template voor verschillende index schema's
 - Voor “clusters” volgens een afstandsmaat
 - Bv vergelijken van intervallen, GIS, bevat, dichtste burens, tekst
- SP-GiST (space-partitioned GiST)
 - Hoeft niet gebalanceerd te zijn
 - Geen overlap tussen de clusters (GiST)

Nieuwere indexvormen: Gin en Brin

- GIN (Generalized Inverted Index)
 - Interessant bij veel dubbele waarden
 - Er worden meerdere opzoekwaarden tegelijk aangemaakt (handig voor rij, tekst, json,..)
- BRIN : voor grote geclusterde tabellen
 - Klein, minder performant tenzij:
 - min/max waarde per blok

Conclusie

Een index op elke tabel en iedere kolom?

Samengevat

- B-tree:
 - goeie standaard, $< \leq = \geq >$
- Hash:
 - alternatief voor 1 rij, $=$
- GIN:
 - efficiënt bij dubbels, meerdere opzoekwaarden per veld, $<@ @> = \&\&$
- GiST:
 - veel mogelijkheden, minder performant, $<< \&< .. <<| \&<| <@ \sim = \&\&$
- Sp-GiST:
 - niet overlappende GiST, $<< >> \sim = <@ <<| |>>$
- BRIN:
 - grote geclusterde data, $< \leq = \geq >$

INDEX catalog_table

- PostgreSQL: pg_index
- <http://www.postgresql.org/docs/current/static/catalog-pg-index.html>
- <http://www.postgresql.org/docs/current/static/internals.html>
- Referenties: Index Internals, Heikki Linnakangas (Pivotal)
- <https://www.pexels.com/photo/blurred-book-book-pages-literature-46274/>