



(12) **PATENT**

(19) **NO**

(11) **331821**

(13) **B1**

NORGE

(51) **Int Cl.**

G06F 17/30 (2006.01)

G06F 12/00 (2006.01)

Patentstyret

(21)	Søknadsnr	20033790	(86)	Int.inng.dag og søknadsnr	2002.02.19 PCT/GB2002/00706
(22)	Inng.dag	2003.08.26	(85)	Videreføringsdag	2003.08.26
(24)	Løpedag	2002.02.19	(30)	Prioritet	2001.02.26, GB, 0104823
(41)	Alm.tilgj	2003.08.26			
(45)	Meddelt	2012.04.10			
(73)	Innehaver	Bernard Consulting Ltd, Geneva Place, Waterfront Drive, PO Box 3469, VG- ROAD TOWN, TORTOLA, De britiske jomfruøyene			
(72)	Oppfinner	Duncan Gunther Pauly, c/o Coppereye Ltd, Box House Box, GB-SN138AA WILTSHIRE, Storbritannia			
(74)	Fullmektig	Onsagers AS, Postboks 1813 Vika, 0123 OSLO, Norge			

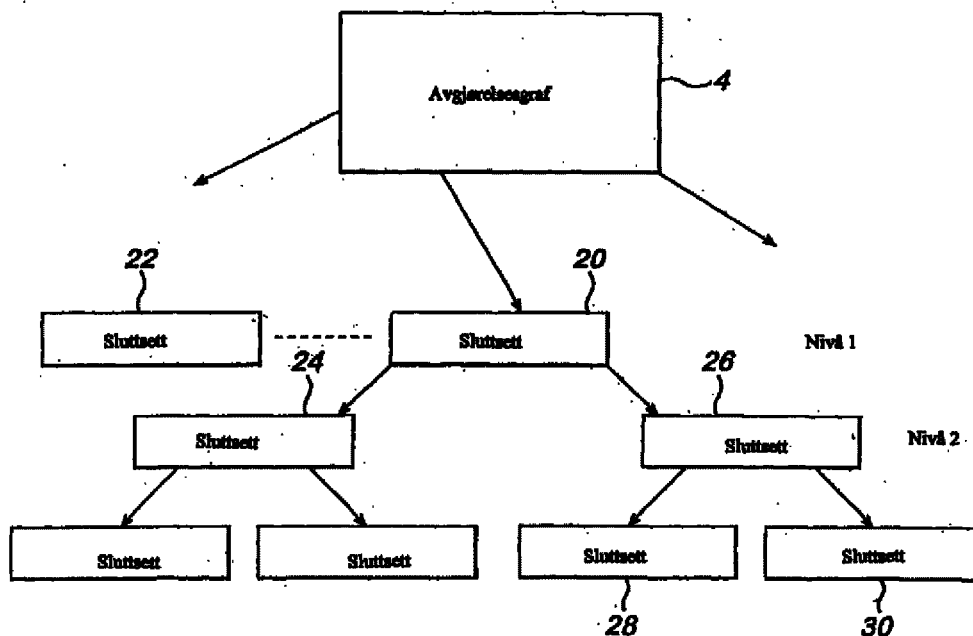
(54) **Benevnelse** **Organisering av data i en database**

(56) **Anførte publikasjoner**

JAGADISH H V et al. *Incremental organization for data recording and warehousing*, PROCEEDINGS OF VLDB 97: 23RD INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, ATHEN GREECE 26-19 August 1997 (1997.08.26) (1997.08.29), pages 16-25, MORGAN KAUFMANN PUBLISHERS, SAN FRANCISCO, CA USA.

(57) **Sammendrag**

En database er tilveiebragt ved hvilke sluttsett (20, 22, 24, 26, 28 og 30) er delt inn i hierarkiske serier av nivåer (nivå 1, nivå 2, nivå 3). Data blir lagt til et sluttsett ved det første nivået (nivå 1) inntil tiden hvor sluttsettet er fullt. Data blir deretter flyttet fra sluttsettet (20) til dens underordnede sluttsett (24, 26) for derved å redusere mengden av disktilgang nødvendig for å legge til data til databasen siden multiple oppføringer blir flyttet ved relativt få disktilgangsoperasjoner.



Den foreliggende oppfinnelsen omhandler en metode for organisering av data i en database, og til en databaseimplementering av en slik metode.

5 Typiske databaser, enten de er av typen beskrevet i søkerens tilsvarende egne britiske patentsøknad GB 0029238.3, eller om de er av andre kjente typer, slik som "B-tree" struktur har en avgjørelsesgraf (eng.: decision graph) eller annen indeks som peker mot sluttsettet (eng.: conclusion set) som lagrer data som matcher søkekriteriet. I tillegg og/eller alternativt kan sluttsettet lagre pekere som peker til lokasjonen til dataene som matcher søkenøkkelen.

10 I enhver database av en rimelig størrelse, blir sluttsettet lagret på massemedia, som for tiden typisk betyr harddisker. Harddiskinnretninger har en tendens til å være betraktelig tregere enn halvlederminnet og følgelig kan databaseytelsen være et kompromiss ved å måtte utføre disse innmatnings/utmatnings (I/O) operasjonene på disse masselagringsinnretningene. Selv ved en minimal indeksoperasjon (eng.: index overhead) må en database typisk utføre to I/O-operasjoner som del av lese, 15 modifisering og skrivesyklusen for å sette inn data i indeksen og sluttsettet.

I publikasjon "Jagadish H V et al. "Incremental organization for data recording and warehousing" Proceedings of VLDB 97: 23RD International Conference on very large Databases, Athen Greece 26-19 August 1997 (1997.08.26)-(1997.08.29), pages 16-25, Morgan Kaufmann Publishers, San Francisco, CA USA" beskrives et 20 system og en metode for organisering av datalagring i en database, hvor sluttsettet er arrangert i en hierarkisk struktur.

Den foreliggende oppfinnelsen har en annen tilnærming som kjennetegners ved å fordele sluttsettene gjennom hele avgjørelsetreet kan grenfaktoren (dvs. antallet sluttsett generert i neste nivå i hierarkiet når innholdet i et sluttsett blir flyttet) 25 enkelt bli valgt ved å velge graden ved hvilket sluttsett blir "fordelt" gjennom avgjørelsesgrafen. Det vil si "mellomliggende avgjørelsessatt avstand" av avgjørelsetrehierarkiet (eller antall hierarkinivåer *uten* sluttsett mellom de *med* sluttsett) kan enkelt bli valgt. Dette er det ikke mulig å gjøre med systemet beskrevet i den nevnte publikasjonen.

30 I henhold til et første aspekt ved den foreliggende oppfinnelsen, er det tilveiebragt en metode for organisering av data i en database, hvor sluttsettet for databasen er arrangert i en hierarkisk struktur, og hvor sluttsettet er arrangert slik at enheter blir ført inn i et valgt sluttsett ved første nivå av betydning inntil antall enheter når en terskelverdi for valgte sluttsett, og da vil innholdet av det valgte sluttsettet migrere 35 til et underordnet sluttsett, for derved å tømme det valgte sluttsettet.

Det er dermed mulig å tilveiebringe en modifisert sluttsettstruktur som betraktelig reduserer antall av sluttsett som nærmest følger utgangen av avgjørelsesgrafen.

Videre er det også mulig å fordele sluttsettet innenfor avgjørelsesgrafen.

En tidligere kjent database vil ha et enkelt "lag" av sluttsett tilgjengelig fra avgjørelsesgrafen. Enhver ny enhet med data innført i sluttsettet vil trenge i det minste to I/O-operasjoner for å innbefatte de dataene (dersom dataene bare kan
5 tilhøre ett sluttsett, og muligens flere I/O-operasjoner dersom de innførte dataene kan tilhøre flere enn ett sluttsett).

Ved å organisere sluttsettene i en hierarkisk struktur, kan antall sluttsett som er øyeblikkelig tilgjengelig fra avgjørelsesgrafen bli svært redusert. Det er faktisk mulig å beholde det mest betydningsfulle hierarkiet (dvs. toppnivået) sluttsettet i et
10 raskt minne, slik som et halvlederminne. Dette er spesielt riktig i de utførelsene av oppfinnelsen hvor sluttsettet er fordelt gjennom avgjørelsesgrafen.

Ved å ha toppnivåsluttsettet i halvlederminnet, vil ingen I/O-kostnader bli påført når data føres inn i databasen. Det er dermed mulig å tilveiebringe en signifikant forbedring i databaseytelsen under nøkkel- og datainnføringsoperasjoner.

15 Fordelaktig vil høynivåsluttsettet effektivt hurtigbufre data inntil tiden hvor sluttsettet blir fullt eller antall av innføringer overstiger et forhåndsbestemt nivå. Sluttsettet blir deretter tømt ved å flytte dets innhold til underordnede sluttsett. Under flytteprosessen blir dataene sortert med hensyn til et søkekriterium, som er en søkenøkkel, slik at dataene kan forventes å være tilfeldig fordelt mellom
20 øyeblikkelige underordnede sluttsett. Denne fyllings- og flytteprosessen kan bli repetert for et flertall av hierarkiske nivåer innenfor sluttsettstrukturen.

Flyttingen av data kan kreve, og krever faktisk ofte, overføring av data mellom ett eller flere sluttsett på en massemedielagringsenhet. Dermed blir disk lese og disk skrive operasjoner pådratt, men nå oppstår disse for sluttsettet som helhet i stedet
25 for hver individuelle enhet innenfor sluttsettet og følgelig blir I/O kostnader pr. enhet redusert mye.

Fordelaktig blir, under nøkkelhenting eller sletting, det passende toppnivåsluttsett og hvert underordnede sluttsett hvilket avgjørelseskriterie matcher søkenøkkelens undersøkt for å se om matchende data er lagret der. Dermed vil
30 databaseforespørseloperasjonen økes sammenlignet med tidligere kjente databaser, men dette er akseptabelt i noen databasestrukturer hvor antall av innsettinger er stort, men antall forespørsler er relativt lavt.

I de utførelsene av oppfinnelsen hvor sluttsettene blir fordelt gjennom avgjørelsesgrafen, kan mellomsluttsettavstanden bli vurdert for å hindre sluttsettene
35 fra å oppstå for ofte.

Fordelaktig er en sluttsettavstandsparameter definert av et heltall Q . Q kan ha et tall større enn eller lik null. Dermed kan f.eks. en database bli laget hvor:

- 5 $Q = 0$: Et sluttsett kan bli dannet ved hver grennode.
- $Q = 1$: Et sluttsett kan bli dannet ved enhver annen grennode.
- $Q = 2$: Et sluttsett kan bli dannet ved hver andre grennode.
- Og så videre.

10 Disse reglene kan bli opprettholdt gjennom avgjørelsesgrafen inntil et slutt Q -nivå av avgjørelsesgrafen blir nådd. I disse lagene blir reglene som omhandler den hierarkiske avstanden mellom sluttsettene ikke gjennomførbare og blir dermed ikke strengt anvendt.

15 I henhold til et annet aspekt ved den foreliggende oppfinnelsen, er det tilveiebragt en database hvor sluttsettet for databasen er arrangert i en hierarkisk struktur, og hvor sluttsettet er arrangert slik at enhetene blir innført inn i et valgt sluttsett ved et første nivå av betydning inntil antall av enheter der når en terskelverdi, og deretter blir innholdt av det valgte sluttsettet flyttet til underordnede sluttsett, for derved å tømme det valgte sluttsettet.

20 I henhold til et tredje aspekt ved den foreliggende oppfinnelsen er det tilveiebragt et datamaskinprogramprodukt for å sørge for at dataprosessoren opererer i henhold til det første aspektet ved den foreliggende oppfinnelsen.

Beskyttelsesomfanget til den foreliggende oppfinnelsen er definert av de selvstendige kravene 1 og , mens ytterligere aspekter ved oppfinnelsen fremkommer fra tilhørende uselvstendige krav.

25 Den foreliggende oppfinnelsen vil bli videre beskrevet ved hjelp av eksempler, med referanse til tilhørende tegninger hvor:

Fig. 1 skjematisk illustrerer en database som har et sluttsett arrangert på en konvensjonell måte;

Fig. 2 skjematisk illustrerer en database som har et sluttsett arrangert i henhold til den foreliggende oppfinnelsen; og

30 Fig. 3 skjematisk illustrerer en database som har et sluttsett fordelt innenfor avgjørelsesgrafen, og utgjør en utførelse av den foreliggende oppfinnelsen.

35 Databasen vist i fig. 1 har en indeks 2 som omfatter avgjørelsesgrafen 4 og et flertall av sluttsett 6, 8, 10, 12 og 14. Hvert sluttsett blir nådd ved én, og kun én, sti gjennom avgjørelsesgrafen. Imidlertid peker hver avslutning da til relevante oppføringer innenfor et datalager 16.

Avgjørelsesgrafen 4 omfatter et flertall av avgjørelsesnoder hvor en søkenøkkel blir tilpasset ved avgjørelseskriteriet for å definere hvilken sti som bør bli tatt gjennom avgjørelsesgrafen. Den interne organisasjonen av taster innenfor avgjørelsesgrafen utgjør ikke en del av den foreliggende oppfinnelsen, og trenger derfor ikke å bli

5 beskrevet i detalj her. Imidlertid kan tidligere kjente indekseringsstrukturer, slik som B-tree indeks bli utnyttet innenfor avgjørelsesgrafen.

I arrangementet vist i fig. 1, har alle avgjørelsessettene 6, 8, 10, 12 og 14 lik signifikans og derved er ingen sluttsett mer hierarkisk signifikant enn andre sluttsett og det kan derfor være mange hundre eller faktisk tusener av sluttsett.

10 I arrangementet vist i fig. 2 er sluttsettene arrangert i en hierarkisk struktur. I arrangementet illustrert finnes det tre nivåer av sluttsett hvor nivå 1 er det hierarkisk mest signifikante, og nivå 3 er det hierarkisk minst signifikante. Dermed er det i dette arrangementet bare en fjerdedel av antall nivå 1 sluttsett som det er nivå 3

15 sluttsett, og igjen i dette eksemplet markerer ett nivå 1 sluttsett oppføringen til seks andre sluttsett. Det er klart at ettersom antall av nivåer øker, vil for et gitt antall sluttsett ved det minst signifikante nivået antall første nivå sluttsett bli progressivt forminsket.

Anta nå at det er ønskelig å innføre en oppføring i databasen. Avgjørelsesgrafen blir navigert i henhold til innføringsnøkkelen for oppføringen, som vil være tilfelle i

20 tidligere kjente databaser, for å finne hvilket sluttsett som oppføringen tilhører. I databasen illustrert i fig. 1, vil dette lede til et sluttsett som er unikt definert. Imidlertid i den foreliggende oppfinnelsen vil resultere dette i at et sluttsett 20 av et antall av nivå 1 sluttsett (hvor bare to 20 og 22 er vist for klarhets skyld) blir identifisert. Fordelaktig er nivå 1 sluttsettene 20 og 22 også inneholdt i fast minne,

25 dvs. enten i fast massemedialagringsinnretning eller enda bedre i halvlederminnet, slik at tidsoperasjonen (eng.: time overhead) for innføring av datasluttsett 20 er mindre sammenlignet med tiden nødvendig for å innføre data i ett av sluttsettene 6, 8, 10, 12 og 14 til en konvensjonell database. Dersom nivå 1 sluttsettene 20 og 22 er holdt på et halvlederminne er det ingen I/O kostnader innbefattet i å skrive til dem.

30 I løpet av tiden, ettersom mer og mer data blir innført i databasen, begynner sluttsettet 20 å fylles. Når antall oppføringer i sluttsettet når et forhåndsbestemt tall, korresponderende til at sluttsettet er fullt, blir oppføringene i sluttsettet 20 migrert ned til rett underliggende (eng.: subordinate) sluttsett 24 og 26 som tilhører til nivå 2 av den hierarkiske strukturen.

35 Avgjørelsen på hvilken av de nedre nivåsluttsettene 22 og 24 som mottar en oppføring fra nivå 1 sluttsettet 20 blir bestemt ved å fortsette navigasjonsreglene som eksisterer innenfor avgjørelsesgrafen 4. Dermed vil, f.eks. dersom avgjørelsesgrafen 4 har regler basert på individuelle bitverdier i økende rekkefølge

av bittallet, regelen for å flytte data fra toppnivåsluttsettet til det andre nivåsluttsettet 24 og 26 bruke neste bit i søkenøkkelen for å bestemme hvilket av sluttsettene 24 eller 26 som skal være mottaker av hver enhet av data. Under flytteprosessen, blir sluttsettet 20 tømt.

- 5 Prosessen med å flytte fra det Nte nivået til et N+1 nivå oppstår ved hvert nivå innenfor sluttsetthierarkiet ettersom hvert sluttsett der fylles opp. Dermed straks sluttsett 26 blir fullt, vil det i tur flytte sine data til dens underordnede sluttsett 28 og 30 på det tredje nivået av hierarkiet. I dette eksemplet er det tredje nivået det laveste nivået til hierarkiet og sluttsettene 28 og 30 er ikke i stand til å videreføre sine data ned til underordnede sluttsett. Imidlertid, dersom fire eller flere nivåer av sluttsett ble innbefattet innenfor denne hierarkiske strukturen, vil sluttsettene 28 og 30 kunne flytte sine data til deres egne underordnede sluttsett når de blir fulle. Det kan forventes at, ved å anta en tilfeldig fordeling av nøkler, at under flyttingen vil halvparten av oppføringene i sluttsettet 20 gå til sluttsettet 24 og den andre halvparten vil gå til sluttsettet 26. Denne prosessen blir repetert for hvert nivå av flytting slik at alle oppføringene i store trekk blir likt fordelt gjennom det nedre nivået av sluttsett.

- I/O kostnadene ved flytting kan bli illustrert i operasjonen med flytting av data fra sluttsettet 26 til sluttsettet 28 og 30. I denne prosessen, må data leses fra sluttsettene 26, som trenger én lesing fra disk. Det nedre nivåsluttsett 28 og 30 må også ha sine data lest fra disk, noe som trenger to leseoperasjoner. Dataene fra sluttsettet 26 blir deretter sortert for å korrekt peke til destinasjonssettet 28 eller 30, oppføringene blir deretter oppdatert, og deretter blir dataene for de to nedre sluttsettene 28 og 30 skrevet tilbake til disken, noe som trenger to skriveoperasjoner. Deretter blir sluttsettet 26 tømt, noe som trenger én skriveoperasjon. Dette gir dermed en total kostnad på seks input/output operasjoner pr. migrasjon. Imidlertid kan denne enkle flyttingen påvirke flere hundre av oppføringer. Det skal også bemerkes at flyttingen fra toppnivåsluttsettet 20 også vil bli mindre fordi den typisk er i minnet og følgelig må bare to lesinger til settene 24 og 26 og skrivingene til disse settene 24 og 26 bli utført.

- Anta at den hierarkiske strukturen som er vist i fig. 2, hvor sluttsettet har S undernivåer (hvor $S = 2$ i dette tilfellet, $S = (L-1)$ hvor L er antallet av nivåer), antall I/O operasjoner nødvendig for å flytte oppføringen til det nederste nivået er $6S$. Imidlertid ettersom hver flytteoperasjon flytter et helt (fullt) sluttsett som inneholder E oppføringer, så vil I/O kostnadene pr. oppføring være gitt ved:

$$\text{I/O kostnad} = \frac{6S}{E}$$

derfor er E stor nok og/eller S liten nok slik at I/O kostnaden kan bli signifikant redusert sammenlignet med tidligere kjente systemer illustrert i fig. 1 som har en I/O kostnad på 2 pr. oppføring. Dermed vil sluttsettet ha hundre oppføringer, og sluttsettdybden på åtte, hvor I/O kostnaden pr. oppføring er $6 \times \frac{8}{100} = 0,48$.

- 5 Det skal bemerkes at for en fast indeksstørrelse, mens reduisering av S øker innføringsgjennomgangen, øker den også antall av sluttsett som må holdes i minnet for at fordelten skal bli realisert.

- 10 Ved etterspørring av indeksen til en spesiell nøkkel, må hvert sluttsett i en enkel sti som går fra minst signifikante sluttsettet bli forespurt. Dermed, dersom den hierarkiske strukturen består av L nivåer, må L sluttsett bli forespurt for å kunne finne alle resultatene som matcher nøkkelen, siden nøkkelen må være på ethvert av nivåene innenfor hierarkiet. Dermed vil dette indekseringsskjemaet øke gjennomgangen, eller lettheten ved at oppføringer kan bli lagt til indeksen på bekostning av degradering av forespørselsytelse. Imidlertid er denne avveiningen
- 15 akseptabel i enhver applikasjon som må akseptere store mengder data men forespør dem sjelden. Et eksempel på en slik applikasjon er et svindeldeteksjonssystem som må laste hver transaksjon, men som bare forespør de transaksjonene som omhandler en suspekt aktivitet.

- 20 Det er dermed mulig å tilveiebringe en forbedret databaseytelse ved å modifisere strukturen til sluttsettene i en hierarkisk struktur.

Oppfinneren har innsett at sluttsettene nødvendigvis ikke alle trenger å være av nederste nivå av databasestrukturen. Som bemerket over, i den fordelte strukturen vist i fig. 2, blir data innført i det første sluttsettet funnet, dvs. nivå 1 sluttsettene, og blir deretter progressivt flyttet til lavere nivåer.

- 25 Dermed kan innføringsoperasjonen bli videre redusert dersom sluttsettene er fordelt gjennom avgjørelsesindeksstrukturen.

Fig. 3 viser en modifisert database hvor sluttsettene er arrangert i en hierarkisk struktur og hvor strukturen strekker seg inn i avgjørelsesgraften.

- 30 Fig. 3 illustrerer en del av avgjørelsesgraften hvor avgjørelsesgraften har blitt konstruert med en mellomsluttsettavstand $Q = 3$, slik at et sluttsett 40, 50 og 60 er tillatt ved hver tredje forgreningsnode. Anta at vi forener databasen ved tiden hvor data inneholdt i den er slik at sluttsettet 40 akkurat har blitt laget, men at sluttsettene 50 og 60 ikke ennå har blitt laget.

- 35 Under bruk blir data lagt til databasen, og innholdet av disse dataene er slik at sluttsettet 40 blir fullt. Når dette oppstår blir informasjonen innenfor sluttsettet

omfordelt som helhet ved å lage en lavere nivånode 42, tilleggsnode 43, og fordele dataene fra sluttsettet 40 inn i et nytt temporært sluttsett 44 og 45 vist i utkastet i fig. 3. Fylling av databasen fortsetter inntil sluttsettet 40 igjen fylles. Den forsøker deretter å omfordele sitt innhold til sine underordnede sluttsett. Dette vil i sin tur

5 gjøre én av dem fulle og det blir dermed nødvendig å fjerne det fulle midlertidige sluttsettet fra utgangsstien til node 42 og å innføre én eller flere noder og tilleggssluttsett etter behov.

Dette kan fortsette inntil sluttsettet 50 er etablert som en utgang til node 46. Ved denne tiden korresponderer mellomsluttsettavstanden mellom sluttsettene 40 og 50

10 til avstanden Q spesifisert av utvikleren eller brukeren av databasen. Ved slike forhold, blir en sjekk utført for å se om et mellomsluttsett eksisterer i avgjørelsesstien som strekker seg mellom nodene 41 og 46. Ethvert sluttsett i denne avgjørelsesstigen blir fjernet og deres innhold blir fjernet fra sluttsettet 50.

Ettersom sluttsettet 50 fylles opp, kan nye temporære sluttsett bli laget inntil

15 sluttsettet 60 er laget, osv. Det blir dermed mulig for mellomsluttsettavstanden mellom sluttsettene som ikke er på det laveste nivået til avgjørelsesindeksen å holdes ved en spesifisert mellomsluttsettavstand. I arrangementet vist i fig. 3 representerer settene 62 og 70 de laveste nivåene til databasen, og sluttsettene 60, 62 og 70 kan representere nivåene 1, 2 og 3 av sluttsettene i arrangementet vist i fig.

20 2.

Som før resulterer masseomfordeling av sluttsettoppføringer til laveste nivå sluttsett i redusert disk input/output straff sammenlignet med skrivingen av individuelle enheter av data. Videre er dataoppføringskostnadene i strukturen vist i fig. 3 ytterligere redusert siden oppføringen er innført i det første sluttsettet funnet. Hver

25 oppføring fra et fullt sluttsett blir deretter flyttet til passende lavt nivå sluttsett som blir funnet ved å følge avgjørelsesgrafene fra det fulle sluttsettet via oppføringen som skal flyttes.

Ved forespørring av databasen, må avgjørelsesstigen for relevant nøkkel navigeres riktig til det laveste nivået sluttsett, og hvert sluttsett funnet på ruten må også bli

30 forespurt ettersom den kan inneholde data som matcher søkenøkkelkriteriet.

PATENTKRAV

1. Metode for organisering av lagring av data i en database (2), hvor sluttsettene (20, 24, 26, 28, 30; 40, 50, 60) for databasen er arrangert i en hierarkisk struktur med et flertall signifikante nivåer innbefattende en første signifikantnivå
5 (nivå 1) og et siste signifikantnivå av betydning, hvor sluttsettene lagrer data som matcher søkekriteriet eller pekere som peker til lokasjonen til dataene som matcher søkekriteriet og hvor sluttsettene er arrangert på en slik måte at enheter blir innført i et valgt sluttsett ved et første nivå av betydning (nivå 1) inntil antall enheter når en terskelverdi for valgte sluttsett, og deretter blir innholdene av valgte sluttsett flyttet
10 til underordnede sluttsett, for derved å tømme det valgte sluttsettet, og hvor det etterfølgende flyttingen av innholdet fra det valgte sluttsettet kan utføres videre innføringer i sluttsettet,
k a r a k t e r i s e r t v e d a t sluttsettene blir fordelt gjennom avgjørelsesgrafene til databasen, hvor avgjørelsesgrafene omfatter et flertall av grennoder ved hvilke en
15 søkenøkkel blir tilpasset avgjørelseskriteriet for å definere hvilken avgjørelsesti som bør bli tatt gjennom avgjørelsesgrafene, hvor hvert sluttsett blir nådd av én, og kun én avgjørelsesti gjennom avgjørelsesgrafene; hvor sluttsett blir dannet ved noen, men ikke alle av grennodene (41, 46); og grennodene ved hvilket det ikke er dannet sluttsett definerer avgjørelsestier som strekker seg mellom grennoder ved hvilke
20 sluttsett er dannet.
2. Metode i henhold til krav 1, hvor hvert sluttsett (20, 24, 26, 28, 30) har null, én eller to nærmeste underordnede sluttsett.
- 25 3. Metode i henhold til krav 1 eller 2, hvor under flytting av data i det valgte sluttsett blir sammenlignet med søkekriterier for å velge hvilken av de nærmeste underordnede sluttsettene som en oppføring skal overføres til.
- 30 4. Metode i henhold til krav 3, hvor databasen innbefatter en avgjørelsesgraf og søkekriteriet er effektivt en utvidelse av avgjørelsesgrafene.
5. Metode i henhold til et av de foregående kravene, hvor den hierarkiske strukturen til sluttsettene er arrangert i en struktur i lag.
- 35 6. Metode i henhold til krav 5, hvor strukturen i lag omfatter L-nivåer, med hvert sluttsett i det første til (L-1)te nivået som har to nærmeste underordnede sluttsett i den andre av Lte nivå respektivt.

7. Metode i henhold til krav 6, hvor når et gitt sluttsett i det Nte nivået, hvor N er et heltall fra 1 til (L-1) har et forhåndsbestemt antall av innføringer i seg, dens innhold flyttet til dens nærmeste underordnede sluttsett.
- 5
8. Metode i henhold til et av de foregående kravene, hvor sluttsettene har en hierarkisk betydning over en forhåndsbestemt verdi lagret, i bruk, i fast minne.
9. Metode i henhold til et hvert av de foregående kravene, hvor
- 10 avgjørelsesgrafene er konstruert for å opprettholde en spesifisert mellomsluttsettavstand gjennom hoveddelen av avgjørelsesgrafene.
10. Metode i henhold til et hvert av de foregående kravene, hvilket under flytting av data fra sluttsettet blir dataene flyttet til underordnede sluttsett på bakgrunn av
- 15 det spesifiserte datainnholdet.
11. Metode i henhold til et hvert av de foregående kravene, hvor innholdet av sluttsettet blir flyttet som helhet.
- 20 12. Metode i henhold til et hvert av de foregående kravene, hvor det når databasen forespørres etter data eller pekere som matcher et søkekriterium må hvert sluttsett i en enkel sti som går fra det minst signifikante sluttsettet bli forespurt, siden data eller pekere kan være på ethvert nivå av hierarkiet.
- 25 13. Metode i henhold til et hvert av de foregående kravene, hvor innholdet av de valgte sluttsett blir flyttet til et sluttsett på lavere nivå som blir funnet ved å følge avgjørelsesgrafene.
- 30 14. Metode i henhold til et hvert av de foregående kravene, hvor avgjørelsesgrafene er konstruert til å holdes ved en spesifisert mellomsluttsettavstand ved:
- å etablere et første sluttsett (40) ved det første nivået av betydning (nivå 1),
- å etablere en eller flere temporære sluttsett (44, 45); og
- å flytte innholdet av det første sluttsettet (40) til det midlertidige sluttsettet inntil et andre sluttsett (50) er etablert ved den spesifiserte mellomsluttsettavstanden fra det

første sluttsettet (40), hvor da det temporære sluttsettet blir fjernet og deres innhold blir flyttet til det andre sluttsettet.

- 5 15. Metode i henhold til krav 9, hvor avgjørelsesgrafene omfatter et flertall grennoder, og hvor avgjørelsesgrafene er konstruert slik at det opprettholdes en spesifisert mellomsluttsettavstand definert av heltall Q gjennom i store trekk hele avgjørelsesgrafene ved å danne et sluttsett ved annenhver grennode dersom $Q=1$, ved hver andre grennoder dersom $Q=2$, osv., hvor Q er større enn null.
- 10 16. Database hvor sluttsettene til databasen er arrangert i en hierarkisk struktur med et flertall av signifikantnivåer innbefattende et første signifikantnivå av betydning (nivå 1) og et siste signifikantnivå av betydning, hvor sluttsettene lagrer data som matcher søkekriteria eller pekere som peker til lokasjonen med data som matcher søkekriteriet, og hvor sluttsettene er arrangert slik at enheter blir innført i et
 - 15 valgt sluttsett ved et første nivå av betydning inntil antall av enheter når en terskelverdi for valgte sluttsett, og hvor deretter innholdet av valgte sluttsett flyttes til underordnede sluttsett, for derved å tømme det valgte sluttsettet, og hvor videre innføringer kan utføres i sluttsettet etter flytting av innholdet fra valgte sluttsettet, k a r a k t e r i s e r t v e d a t sluttsettene blir fordelt gjennom avgjørelsesgrafene
 - 20 (41, 42, 43, 46), hvor avgjørelsesgrafene omfatter et flertall av grennoder ved hvilke en søkenøkkel blir tilpasset avgjørelseskriteriet for å definere hvilken avgjørelsesti som bør bli tatt gjennom avgjørelsesgrafene, hvor hvert sluttsett blir nådd av én, og kun én avgjørelsesti gjennom avgjørelsesgrafene; hvor sluttsett blir dannet ved noen, men ikke alle av grennodene (41, 46); og hvor grennodene ved hvilket det ikke er
 - 25 dannet sluttsett definerer avgjørelsestier som strekker seg mellom grennoder ved hvilke sluttsett er dannet.
 - 30 17. Database i henhold til krav 16, hvor sluttsettene (40, 50, 60) blir fordelt gjennom en avgjørelsesgraf.
 18. Datamaskinprogramprodukt for å sørge for at en dataprosessor implementerer metoden i henhold til et av kravene 1-15.

Fig. 1

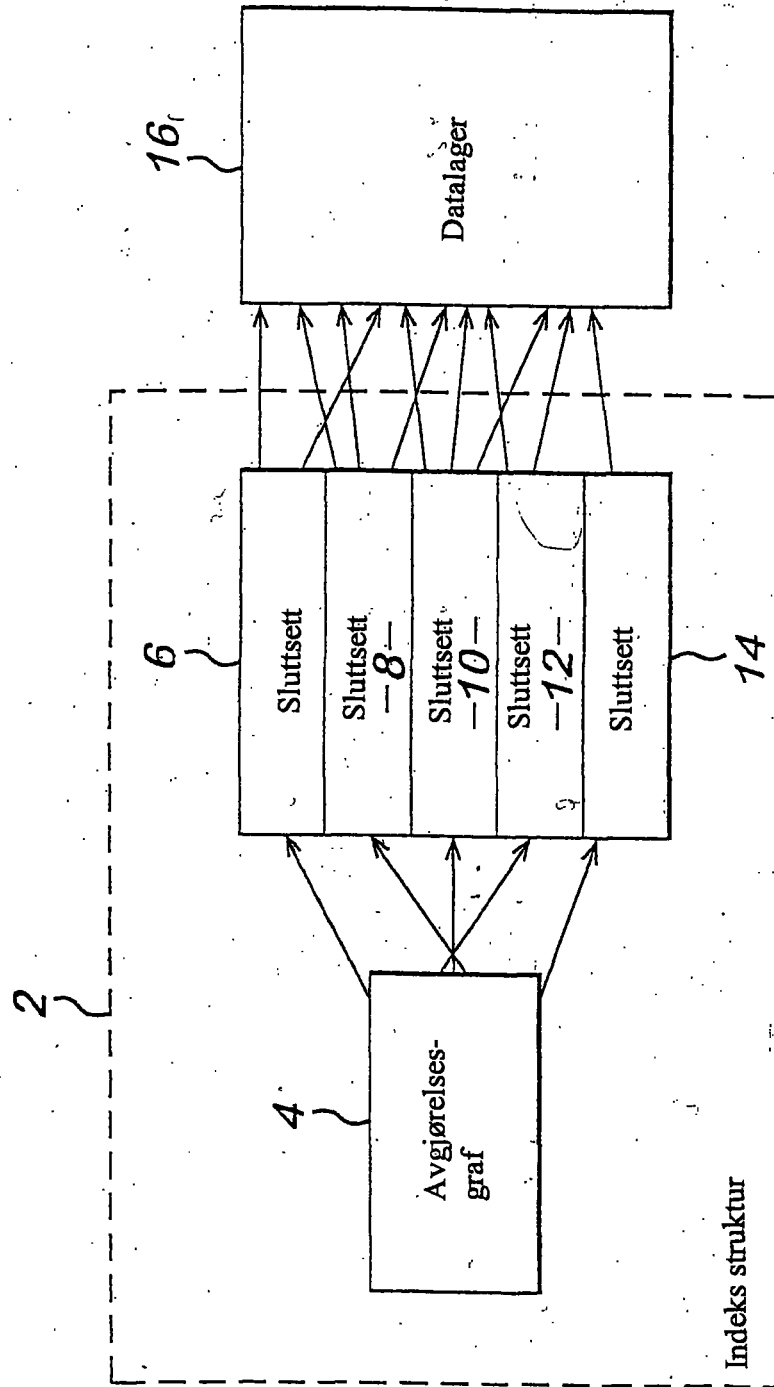


Fig. 2

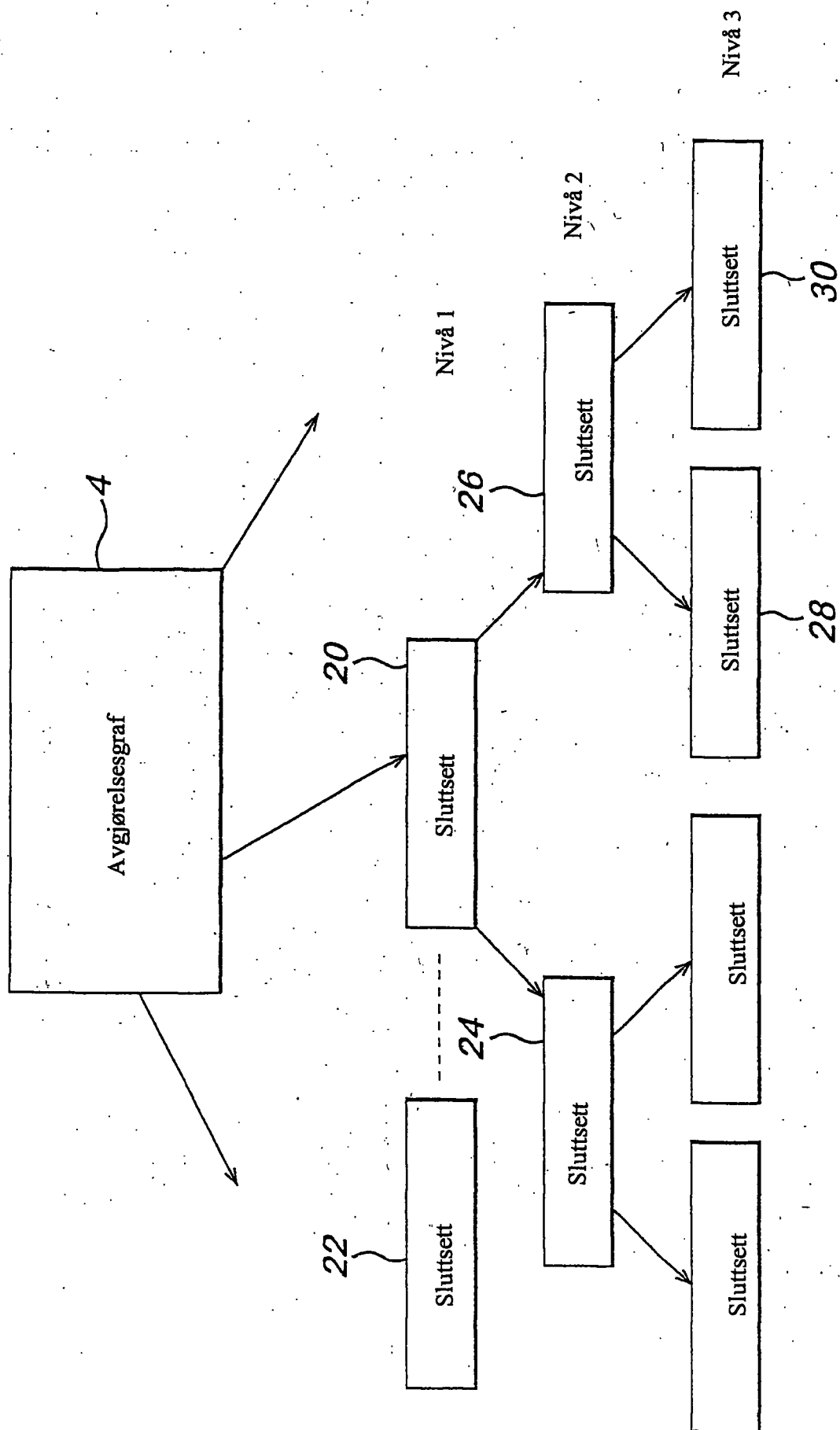


Fig. 3