

METSÄNTUTKIMUSLAITOKSEN
TIEDONANTOJA 157

Rovaniemen tutkimusasema



TUTKA-TIEDONHALLINTAOHJELMISTO TIETOKANNAN MUODOSTUS JA KÄYTTÖ

Erkki Kaila ja Markku Taipale



ROVANIEMI 1984

METSÄNTUTKIMUSLAITOKSEN TIEDONANTOJA 157

TUTKA - TIEDONHALLINTAOHJELMISTO
TIETOKANNAN MUODOSTUS JA KÄYTTÖ

ERKKI KAILA JA MARKKU TAIPALE

ROVANIEMI 1984

TUTKA-tiedonhallintaohjelmisto
Tietokannan muodostus ja käyttö

Metsäntutkimuslaitoksen Rovaniemen tutkimusasemalla on kehitetty TUTKA-tiedonhallintaohjelmisto. TUTKALLA voidaan tutkimuskohtaiset havaintoaineistot käsitellä hierarkkisina tietokantoina, joista voidaan poimia havaintomatriiseja. Tietokantojen tietorakenne soveltuu erityisesti luonnontieteellisten tutkimusaineistojen hallintaan. TUTKA-ohjelmisto on toteutettu VAX-11 FORTRAN -ohjelmointikielellä ja toimii tehokkaasti VT100 ja VDU210 päätteillä.

Tässä käsikirjassa selvitetään aluksi tietokannan hallintaan liittyviä peruskäsitteitä ja tietokannan muodostamista sekä käyttömahdollisuuksia. Toinen osa käsittää TUTKAN yleisten apuohjelmien käyttöohjeet ja kolmas datankäsittelyruutiinit.

TUTKA - a database management system.
Database design, construction and usage.

A database management system - TUTKA - has been developed at Rovaniemi Research Station of the Finnish Forest Research Institute. By means of TUTKA the data of individual research projects can be manipulated as hierarchical databases from which data matrices can be produced. The database structure is especially suitable for management of research data in science. TUTKA uses VAX-11 FORTRAN as the host language and operates most effectively with VT100 and VDU210 display terminals.

The first part of this handbook presents main concepts of database management as well as construction and usage of databases. The two other parts form a user's manual containing TUTKA's utilities and data manipulation language.

ISBN 951-40-0947-9

ISSN 0358-4283

Helsinki 1984. Valtion painatuskeskus

Alkusanat

Tähän käsikirjaan on koottu Metsäntutkimuslaitoksen Rovaniemen tutkimusasemalla vv. 1980 - 1984 rakennetun, yleiseen käyttöön tarkoitetun tiedonhallintaohjelmiston käyttöohjeet. Käsikirjan julkistaminen päättää metsäntutkimuksen tiedonhallinnan kehittämisessä vaiheen, jonka aikana tietokantojen hyväksikäytön tärkeys ja pääasialliset muodot ovat selkiytyneet. Työ liittyy METLÄN matemaattisen osaston tutkimusaiheeseen MAT20 - 025 "Tietokantojen soveltuvuus metsätalouden suunnittelun ja metsäntutkimuksen käyttöön". Työtä jatketaan itse tietokantojen toteutuksena.

Ohjelmiston suunnittelu ja ohjelmointi ovat tapahtuneet kirjoittajien toimesta. Erkki Kaila on hankkeen alullepanija sekä ohjelmiston hyväksikäytön ja toiminnan suunnittelija kokonaisuuden kannalta. TUTKA 1/1 ohjelmiston lopullinen toteutus on tapahtunut Markku Taipaleen toimesta. Lisäksi Jari Vanhatalo on osallistunut järkeistelmän kehittämistyöhön sen alkuvaiheessa. Useat testauksessa käytetyt tietokannat ovat yllämainittujen konstruoimia.

Rovaniemen tutkimusaseman tutkijoista tri Pentti Roiko-Jokela ja MML Hannu Saarenmaa ovat merkittävällä panoksella osallistuneet ohjelmiston rakentamisen ohjaukseen. Rovaniemen tutkimusaseman johtajat MMT Yriö Norokorpi ja tri Pentti Roiko-Jokela sekä professorit Risto Seppälä ja Pekka Kilkki ovat kaikki rakentamistyön eri vaiheissa vaikuttaneet siihen, että ohjelmiston kehittämisolot ovat olleet erinomaiset. Lisäksi FK Hilikka Kinnunen ja merkonomi Eila Kuirinlahti ovat huolehtineet käsikirjan kieliasun tarkistuksesta ja teknisestä toteutuksesta.

Kiitämme kaikkia mainittuja yhteistyöstä sekä Suomen Akatemiaa hankkeen osarahoituksesta.

Rovaniemellä 15.6.1984

Erkki Kaila

Markku Taipale

S I S Ä L L Y S L U E T T E L O

JOHDANTO	1
1. TUTKA-ohjelmisto	1
2. Käsikirjan tarkoitus	1
3. Käsikirjan sisältö	1
I T I E T O K A N T A, S E N M U O D O S T U S J A K Ä Y T T Ö	
1. KÄSITTEITÄ	2
1.1 Tutkimuksen tiedonhallinta	2
1.2 Määritelmiä	2
1.3 Esimerkki	8
2. TIETOKANNAN MUODOSTUS	9
2.1 Looqinen kuvaus	9
2.2 Fyysinen kuvaus	13
2.3 Tietokannan lataus	17
3. TIETOKANNAN KÄYTTÖ	21
3.1 Yleiset ohjelmat	21
3.1.1 Tietokannan ylläpito ja huolto	21
3.1.2 Raportointi- ja valintaohjelmat	22
3.2 Käyttäjän omat ohjelmat	23
3.2.1 Tietokantohjelmiston yleispiirteet ...	24
3.2.2 DML- eli datan käsittelyohjelmat	25
II T I E T O K A N N A N K Ä S I T T E L Y N A P U O H J E L M A T (T U T K A U T I L I T Y)	
1. APUOHJELMISTON YLEINEN KUVAUS	27
1.1 Ohjelmiston tarkoitus	27
1.2 Ohjelmiston looqinen rakenne	27
1.3 Ohjelmiston käynnistäminen ja päävalikko ...	28
2. TIETOKANNAN LUONTI (DDP)	31
2.1 Ohjelman tarkoitus	31
2.2 Ohjelman käyttöohje	31
2.2.1 Kaavion kuvaaminen DDP-ohjelmalla ...	31
2.2.2 Tietokannan generointi kaaviosta	36
2.3 Esimerkki	36
2.4 Rajoitukset	42
2.5 Virheilmoitukset	43
2.6 HELP-komento	45
3. TIETOKANNAN KAAVION MANIPULOINTI (DMP)	46
3.1 Ohjelman tarkoitus	46
3.2 Ohjelman käyttöohje	46
3.2.1 Kaavion käsittely	46
3.2.2 Käyttäjien lisääminen ja poistaminen ..	46
3.2.3 Termiluetteloiden käsittely	46
3.3 Esimerkki	47
3.4 Rajoitukset	47
3.5 Virheilmoitukset	47

4. TIETOKANNAN KAAVION JA TERMIUETTELOIDEN RAPORTOINTI (DRP)	51
4.1 Ohjelman tarkoitus	51
4.2 Ohjelman käyttöohje	51
4.3 Esimerkki	51
4.4 Virheilmoitukset	51
5. DATAN KERUU TIETOKANNASTA (DAP)	53
5.1 Ohjelman tarkoitus	53
5.2 Ohjelman käyttöohje	53
5.2.1 Ehtojen svöttö	53
5.2.2 Tulostustiedostojen määrääminen	54
5.2.3 Tulostettavat muuttujat	54
5.3 Esimerkki	55
5.4 Rajoitukset	57
5.5 Virheilmoitukset	57
6. TIETOKANNAN TIEDOSTOJEN DUMPPAUS (DUM)	59
6.1 Ohjelman tarkoitus	59
6.2 Ohjelman käyttöohje	59
6.2.1 Kaavion (sanakirjan ja lyhenneluettelon) dumppaus	59
6.2.2 Datatiedostojen dumppaus	59
6.2.3 Termiluetteloiden dumppaus	59
6.3 Esimerkki	60
6.4 Rajoitukset	63
6.5 Virheilmoitukset	63
7. TIETOKANNAN TUHOAMINEN JA POISTAMINEN REKISTERISTÄ (DDD)	64
7.1 Ohjelman tarkoitus	64
7.2 Käyttöohje	64
7.3 Esimerkki	64
7.4 Rajoitukset	64
7.5 Virheilmoitukset	64
III DATAN KÄSITTELYOHJELMAT (TUTKA DML)	
1. DATAN KÄSITTELYOHJELMISTO	66
1.1 Ohjelmien nimeämisperiaatteet	66
1.2 Ohjelmien käyttöönotto	66
1.3 Ohjelmien ryhmittely	66
1.4 Ohjelmakuvaukset	69
1.4.1 Yleisohjelmat	71
1.4.2 Tietosisältöä muuttavat ohjelmat	74
1.4.3 Tiedon saantiohjelmat	79
1.4.4 Hakuohjelmat	91
1.4.6 Käsitteilyosoitteiden asetusohjelmat ..	94
1.4.6 Looqiset testifunktiot	99
1.4.7 Apuohjelmat	101
2. ESIMERKKIOHJELMIA	102
3. RAJOITUKSET	108
4. VIRHEILMOITUKSET	108

VIITTEET 115

LIITTEET

- A Kapykanta-tietokannan tiedostot
- B Tietokannan fyysinen kuvaus
- C Loogisen kaavion kuvauslomakkeet
- D Hakemisto

JOHDANTO

1. TUTKA-ohjelmisto

TUTKA on lyhenne sanoista "TutkimusTietokantojen hallintajärjestelmä". Sillä tarkoitetaan Metsäntutkimuslaitoksessa kehitettyä, erityisesti metsäntutkimuksen havaintoaineistojen käsittelyyn soveltuvaa tiedonhallintaohjelmistoa. TUTKA mahdollistaa hierarkkisten tietokantojen muodostamisen numeerisista havaintoaineistoista sekä niiden käsittelyn. Ohjelmiston osat ovat: tietokannan kaavion ja datan käsittelyn yleiset apuohjelmat (TUTKA UTILITY) sekä datan käsittelyohjelmat (TUTKA DML). Ohjelmisto on rakennettu VAX 11/780 laitteistolle. Metsäntutkimuslaitoksessa TUTKA-ohjelmisto on sijoitettu sekä Helsingin että Rovaniemen laitteistoille. Se on toiminnassa myös Joensuun yliopiston VAX-tietokoneella.

2. Käsikirjan tarkoitus

Tällä käsikirjalla on kolme tarkoitusta:

1) Siinä esitetään suppeasti tietokantojen hallintaan liittyvää peruskäsitteistöä. Käsitteiden määrittelyllä pyritään sekä antamaan yleiskuva tiedonhallinnasta ja TUTKA-ohjelmiston käytöstä että helpottamaan ohjelmiston käytön opettelua.

2) Sillä johdatellaan käyttäjä (tutkija) tutkimusaineistoista muodostettavien tietokantojen rakentamiseen ja hyväksikäyttöön.

3) Se toimii TUTKA-ohjelmiston käyttöohjeena.

3. Käsikirjan sisältö

Käsikirjassa on kolme pääosaa. Ensimmäisessä luvussa esitetään tietokannan hallinnan peruskäsitteistöä. Luku on jaettu kappaleisiin, joissa kuvataan tietokannan muodostamista ja käyttöä. Lukijan kannalta uusiksi oletetut käsitteet määritellään niiden esiintyessä ensimmäistä kertaa. Määritelmät on otettu sellaisinaan lähdeeteoksista, koska käsitteiden suomentamisen on katsottu vähentävän niiden yksikäsitteisyyttä.

Toisessa luvussa esitetään kaavion ja datan käsittelyohjelmisto. Ohjelmiston osat esitetään yhden tietokannan osalta loogisessa käyttöjärjestyksessä tietokannan luonnista sen tuhoamiseen. Kunkin ohjelman osalta esitetään ohjelman tarkoitus, käyttöohje, esimerkkiajo, rajoitukset sekä ohjelman antamat virheilmoitukset. Esimerkit on havainnollistettu näyttöruuduilla.

Käsikirjan kolmas osa sisältää tietokannan data-tiedostojen käsittelyrutiinit eli ns. DML-ohjelmat. Ohjelmat on ryhmitelty käyttötarkoituksen mukaan seitsemäksi erilliseksi kokonaisuudeksi. Jokainen ohjelma kuvataan saman kaavan mukaisesti, mihin sisältyy ohjelman nimi, nimen selite, ohjelman toiminta, ohjelman kutsu, kutsuparametrit, ns. käsittelyosoitteiden tilaindikaattori, ohjelman antama paluukoodi sekä mahdollinen huomautus.

I TIETOKANTA. SEN MUODOSTUS JA KÄYTTÖ

1. KASITTEITA

1.1 Tutkimuksen tiedonhallinta

Tutkimuksen tiedonhallinnalla tarkoitetaan tässä tapahtumasarjaa, joka alkaa tieteellisten havaintojen merkitsemisestä, jatkuu havaintokokonaisuuksien käsittelytoimintoina ja päättyy tiettyä analyysitoimintoa varten muodostetun havaintomatriisin esittämiseen.

Metsäntutkimuslaitoksessa on viime aikoina puhuttu yhä enemmän tiedonhallinnan osaamisen ja välineistön tarpeellisuudesta tutkimustyössä. Yhä useammalla tutkimussuunnalla on tultu tilanteeseen, jossa tutkimustoimintaan tai erillisiin tutkimushankkeisiin liittyvät tiedostot haluttaisiin koota yhteen jonkinlaiseksi kokonaisjärjestelmäksi. Yhteenkokoamisella haluttaisiin parantaa havaintomatriisien saatavuutta aineistojen muodostamasta havaintomassasta ja dokumentoida olemassaolevaa mittaustietoa. Toistaiseksi laitoksessa ei ole ollut käytettävissä tiedonhallintaan soveltuvaa ohjelmistoa DATATRIEVE-ohjelmistoa lukuunottamatta, mutta parhaillaan (1984) on käynnissä selvitystyö sopivan ohjelmiston hankkimiseksi. Tietokannan hallintajärjestelmän hankinta on sisällytetty laitoksen tietojenkäsittelytoiminnan kehittämisuunnitelmaan.

TUTKA-ohjelmistolla on mahdollista muodostaa hierarkkisia tietokantoja erityisesti tutkimusaineistojen kannalta sopivaa tietorakennetta käyttäen. Ohjelmiston käyttö on pääpiirteittäin CODASYLin (Conference on Data System Languages) suositusten mukainen ja sellaisenaan sitä voidaan pitää prototyypijärjestelmänä "oikeasta" tiedonhallintajärjestelmästä. Ohjelmiston kehittämisessä on käytetty esimerkkinä FORTRAN-ohjelmointikielellä ohjelmoitua GPLAN/DMS-ohjelmistoa (HASEMAN ja WHINSTON 1977).

1.2 Määritelmiä

Tietokantojen muodostaminen tutkimusaineistoista on siis tähän asti ollut verrattain harvinaista. Tästä johtuen on TUTKA-ohjelmiston kehittämisen aikana toistuvasti kohdattu vaikeuksia tiedonhallintaan liittyvän käsitteistön yhteydessä. Englanninkielisille termeille on ollut vaikea löytää suomalaisia vastineita. Joissakin tapauksissa uusi käsite eroaa jo opitusta, tietueiden ja tiedostojen käsittelyyn liittyvästä käsitteestä niin vähän, että eroa on ensituntumalta vaikea ymmärtää. Jotkin toiset käsitteet ovat taas kokonaan uusia eivätkä ehkä selity käyttäjälle ennen kuin ne realisoivasta tiedonhallintaohjelmistosta on käyttökokemusta.

Tiedonhallinnan yhteydessä tiedolla tarkoitetaan poikkeuksetta tietokoneella käsiteltävissä olevaa tietoa, dataa. Tietokannan (database) katsotaan muodostuvan samaan asiakokonaisuuteen liittyvästä ohjelmallisesti hallitusta datatiedostojen joukosta, jolle on olemassa sekä ohjelmista että tiedostoista

erillinen kuvaus eli kaavio (schema). Yleisesti tulkittuna tietokanta saattaa merkitä myös "tietotausta"-sanana synonyymia eli asiakokonaisuuteen liittyvää käsitteistöä tai tietopohjaa (knowledge base). Se ei suoranaisesti vastaa tässä käytettyä suppeampaa tietokoneelle rakennetun tietokannan käsitettä. DATE (1981) määrittelee tietokannan seuraavasti:

Määritelmä

A database is a collection of stored operational data used by the application systems of some particular enterprise.

Tietokanta on määritelmän mukaan hyötykäyttöön tarkoitettu datakokoelma. Datakokoelman käyttö tapahtuu jotain tarkoitusta varten muodostetun yrityksen tietosysteemeissä.

DATEN tarkoittama yritys (particular enterprise) on tehdas, pankki, sairaala, yliopisto, tutkimuslaitos tai muu vastaava, toiminnallisesti suuri, instituutio. TUTKA-ohjelmiston yhteydessä mittasuhteet ovat huomattavasti suppeammat: tietokannan hyväksikäyttö ymmärretään tapahtuvaksi jonkin tarkoituksellisen toiminnon, erityisesti tutkimushankkeen, osana.

Tietokannan hallinta (database management) tarkoittaa tietokoneella ja tätä tarkoitusta varten laadituilla erityisohjelmilla tapahtuvaa tiedostojen hallintaa, jolle on ominaista

- 1) tietokannan kaikkien osien välisten suhteiden kuvaus
- 2) sovellusohjelmien ja datan välinen riippumattomuus (muutokset jommassakummassa eivät aiheuta muutoksia toisessa)
- 3) tiedon moninkertaisen tallennuksen välttäminen suurelta osin tai kokonaan
- 4) käyttäjän mahdollisuus käsitellä tietokantaa korkeatasoisen kielen avulla (korkeatasoisuudella tarkoitetaan tässä sovellusalakohtaista, käyttäjän omaan terminologiaan perustuvaa käsitteelykieltä).

MARTIN (1977) määrittelee tietokannan ja erillisten tietokantojen muodostaman järjestelmän seuraavasti:

Määritelmä

A data base may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy to serve multiple applications; the data are stored so that they are independent of programs which use the data; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within the data base. The

data is structured so as to provide a foundation for future application development. One system is said to contain a collection of data bases if they are entirely separate in structure.

Tietokanta voidaan siis määritellä keskinäisessä suhteessa olevien datajoukkojen kokonaisuudeksi. Kokonaisuus on talletettu useampaa käyttötarkoitusta varten välttämättä haitallista tai tarpeetonta moninkertaisuutta. Talletustapa on riippumaton dataa käyttävistä ohjelmista. Uuden tiedon vieminen tietokantaan sekä tietokannassa olevan tiedon modifiointi ja saanti voi tapahtua vain kontrolloidulla tavalla. Datan rakenne mahdollistaa tietokannan käytön muuttamattomana sen muodostamista seuraavien sovellutusten kehittämisessä. Järjestelmän, jonka tietokannat ovat rakenteellisesti täysin erilliset, sanotaan sisältävän tietokantakokoelman.

Tutkimustoimintaa tukevan tietokannan ja tietokantajärjestelmän perusajatuksena on lisätä tieteellisten havaintojen keskitettyä hallintaa ja hajautettua käyttöä. Tällä oletetaan saavutettavan seuraavat edut:

- talletetun tiedon moninkertaisuus vähenee
- tiedon yksikäsitteisyys lisääntyy
- dataan voidaan kohdistaa useita käyttötarkoituksia
- standardien käyttö edistyy
- tieto voidaan suojata rajoituksin
- tietokokonaisuuksien eheys säilyy
- dokumentoinnin aste kasvaa.

Tietokannan hallintajärjestelmän peruskomponentti on tietokannan kaavio, joka kuvaa tietokannan käyttäjän kannalta yksinkertaisella tavalla tietokannan tiedostorakenteen, tietue-tyyppien ja tietokenttien tietosisällön, tietue-tyyppien väliset suhteet, tietokannan loogiset varmistukset, käytön suojaukset jne. Kaikki kommunikointi tietokantaa käyttävien ohjelmien ja tietokannan tietosisällön välillä tapahtuu kaavion kautta. CODASYLin antama määritelmä tietokannalle perustuu kaavion (schema) käsitteeseen (MARTIN 1977):

Määritelmä

A data base consists of all the record occurrences, set occurrences and areas which are controlled by a specific schema. If an installation has multiple data bases, there must be a separate schema for each data base. Furthermore, the content of different data bases is assumed to be disjoint.

Tietokanta muodostuu täten kaikista tietue-esiintymistä, niiden joukkosuhteista sekä tilavarauksista, joita kontrolloidaan erityisellä kaaviolla. Jos järjestelmässä on useita tietokantoja, jokaista tietokantaa kohti täytyy olla erillinen

kaavio. Edelleen eri tietokantojen tietosisällöt oletetaan erillisiksi.

Kaavio tai "skeema" voidaan yksinkertaisesti ymmärtää tietokannan loogisen rakenteen kokonaiskuvauksena. CODASYLin tarkka määritelmä kaaviolle on (MARTIN 1977):

Määritelmä

A schema consists of DDL (Data Description Language) entries and is a complete description of all of the area, set occurrences, record occurrences and associated data-items and data-aggregates as they exist in the data base.

Kaavio koostuu DDL:n (kaavion kuvauskielen) lauseista ja on kaikkien tietokannan tilavarausten, setti- ja tietue-esiintymien sekä niihin liittyvien tietoalkioiden ja niistä muodostettujen yhdistelmien täydellinen kuvaus.

Määritelmässä mainittu DDL-kaavionkuvauskieli on korkeatasoinen, COBOL-ohjelmointikielen kaltainen määrittelykieli. DDL on COBOLin tapaan itsedokumentoiva kieli. DDL:ään sidottu määritelmä kuvaa huonosti TUTKalla totetettavan tietokannan kaaviota, koska TUTKAan ei ole rakennettu kyseisen kielen kääntäjää. Kaavion määritelmään ja muotoon palataan kappaleissa 2.1 ja 2.2, joissa kuvataan kaavion muodostamista tietokannan rakentamisessa.

Mainitulle "set"-käsitteelle on ollut erityisen vaikea löytää sopivaa suomenkielistä vastinetta. Käsitteet "joukkorelaatio" tai "tietuejoukko" kuvannevat asiasisältöä kohtuullisen hyvin asian jo ennestään tuntevalle henkilölle. Käytännössä on normaalkieleen pujahtanut selvä suomenkielen sana "setti", jolla koko käsitteistöön on helppo viitata mutta joka lyhydessään kuvaa vieläkin huonommin kyseistä ilmiötä. CODASYLin määritelmä setille on seuraava (MARTIN 1977):

Määritelmä

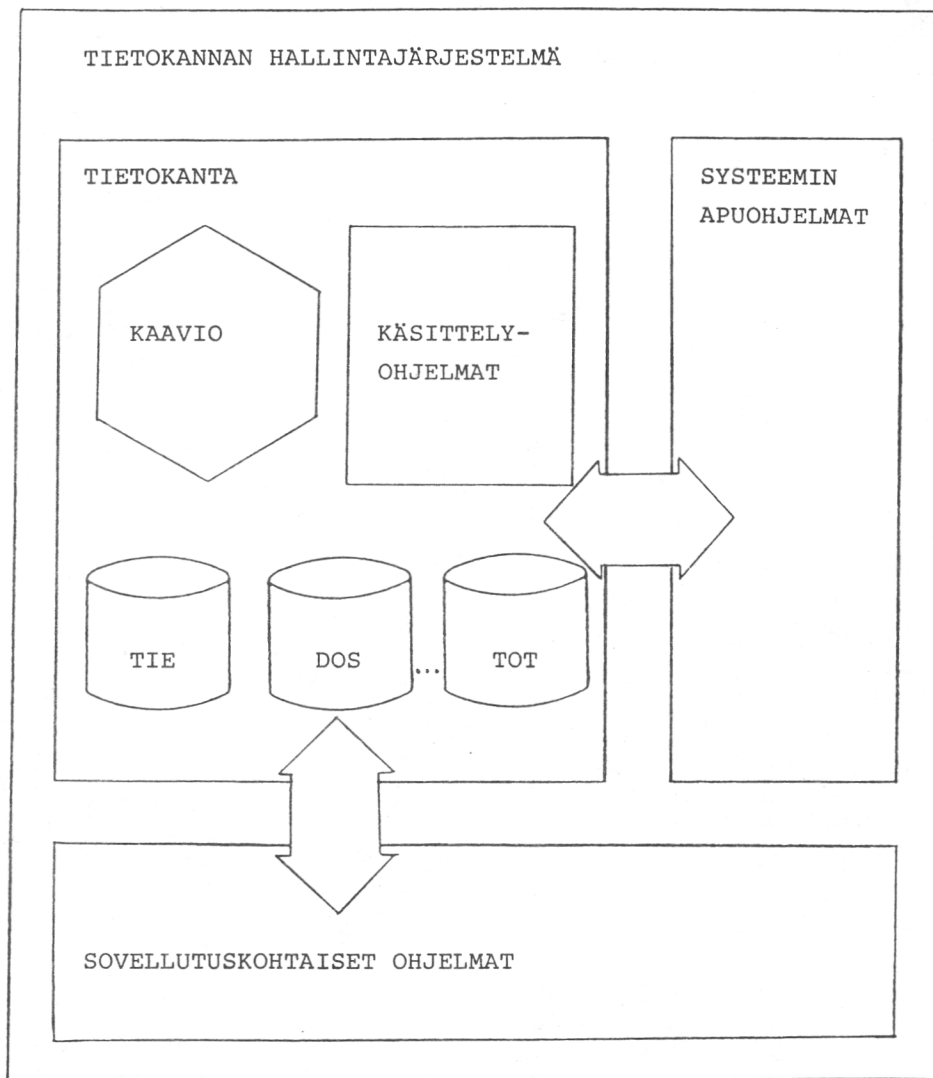
A set is a named collection of record types. As such, it establishes the characteristics of an arbitrary number of occurrences of the named set. Each set type specified in the schema must have one record type declared as its owner and one or more record types declared as its member records. Each occurrence of a set must contain one occurrence of its owner record and may contain an arbitrary number of occurrences of each of its member record types.

Setti on nimetty joukko tietuetyyppejä. Sellaisena siihen sisältyy rajoittamaton määrä yksilöitäviä setin esiintymiä. Jo-

kaiseen kaaviossa määriteltyn settityyppiin kuuluu tietue-tyyppi, joka nimetään setin isännäksi, ja yksi tai useampia tietuetyyppejä, jotka nimetään setin jäseniksi. Jokaiseen setin esiintymään kuuluu yksi isäntätietuetyypin esiintymä ja siihen liittyen rajoittamaton määrä kunkin jäsentietuetyypin esiintymiä.

Setin käsite liittyy hierarkkisiin ja verkkorakenteisiin tietokantoihin, erityisesti CODASYLin DBTG-tietokannanhallintajärjestelmään. Hierarkkisessa järjestelmässä voi kullakin jäsenellä (member) olla vain yksi isäntä (owner). Verkkorakenteisessa isäntiä voi olla monta. TUTKassa settityyppi ilmaisee yksinkertaisesti kahden tietuetyypin välisen suhteen. Siten setti koostuu yhdestä isäntätietuetyypistä ja yhdestä jäsentietuetyypistä.

Edelliset määritelmät sisältävät lähes kokonaisuudessaan sen käsitteistön, joka tulee hallita hierarkkisten tai verkkorakenteisten tietokantojen perusteiden ymmärtämiseksi. Eräs vaikeaselkoisimpia asioita saattaa olla tietuetyypin (record type) ja sen esiintymän (record occurrence) sekä sen kaltaisten vastaavuuksien (set type, set occurrence) välinen ero. Totuttu käsite "tietue" pyritään usein yhdistämään molempiin tai jompaan kumpaan, mutta yksikäsitteinen merkitys on jäänyt saavuttamatta. Asian ymmärtämistä helpottaa, jos tietuetyyppi käsitetään tietokannan kaavioon sisältyväksi loogiseksi kuvaukseksi ja sen esiintymät varsinaisiksi dataa sisältäviksi fyysisiksi tietueiksi, jotka on talletettu kullekin tyypille erikseen varatulle työalueelle (area) levyllä.

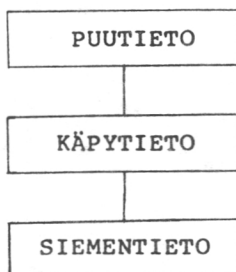


Kuva 1. Tietokannan hallintajärjestelmä ja tietokanta.

1.3 Esimerkki

Seuraavassa esitetään esimerkki hierarkkisesta tietokannasta ja sen loogisesta kuvaamisesta. Esimerkkietokannan fyysisen kaavion listaus on esitetty liitteessä A.

Oletetaan, että tutkijan tehtävänä on tutkia kuusen siementuotantoa. Tätä tarkoitusta varten hän kerää tietoa sopivan kuusiotannan perusteella. Kerättävät tiedot ovat sekä puu-, käpy- että siemenkohtaisia tunnuksia. Tiedoista muodostuu kolme erilaista tietuetyyppiä, joiden hierarkia on selvästi nähtävissä:



Kuva 2. Siementutkimuksen tietokannan looginen rakenne.

Kuvassa 2 nähdään kolmen tietuetyypin, PUUTIETO, KÄPYTIETO ja SIEMENTIETO, muodostama rakenne. Rakenteeseen kuuluu kaksi settityyppiä: PUUTIETO-KÄPYTIETO JA KÄPYTIETO-SIEMENTIETO. Setit voidaan tässä tapauksessa tulkita seuraavasti: jokaiseen PUUTIETO-tietuetyypin esiintymään liittyy joukko KÄPYTIETO-tietuetyypin esiintymiä. Samoin on alemmalla tasolla: jokaiseen KÄPYTIETO-tietuetyypin esiintymään liittyy joukko SIEMENTIETO-tietuetyypin esiintymiä. Jokaisen puun kävyt ja jokaisen kävyn siemenet ovat siis saavutettavissa ao. isäntätietueen kautta. Jokaista tietuetyyppiä varten on levyllä oma talletusalue (area, tiedosto), jonne talletetaan sekä esiintymien tietosisältö (havainnot, tietueet) että tietokannan rakennetta koskevaa tietoa.

Tähän esimerkkiin palataan loogisen ja fyysisen kaavion muodostamista ja tietokannan luontia kuvaavissa kappaleissa. Esimerkkietokannan TUTKA-versio on luotu nimelle KÄPYKANTA. Tietokantaan liittyviä listauksia ja raportteja on koottu liitteisiin A ja B.

2. TIETOKANNAN MUODOSTUS

2.1 Looginen kuvaus

Tietokannan muodostamisessa on kolme erillistä päävaihetta, joista loogisen kaavion muodostaminen on ensimmäinen. TUTKA-ohjelmiston käytön yhteydessä looginen kaavio käsitetään tietokannan rakennetta ja tietosisältöä kuvaavaksi käsittekaavioksi, joka on esitettävissä yksinkertaisilla graafisilla apuvälineillä. Tietosisällön kuvaamista varten on laadittu määramuotoiset lomakkeet (liite C).

MARTIN (1977) määrittelee loogisen kaavion (schema, logical data-base description) seuraavasti:

Määritelmä

A schema. A description of the overall data-base structure as perceived for the users, which is employed by the data-base management software.

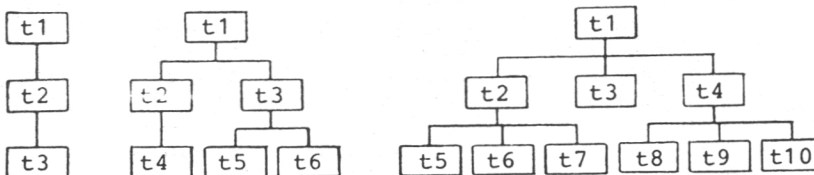
Tietokannan looginen kuvaus tai kaavio on siis tietokannan rakenteen kokonaiskuvaus käyttäjän näkökulmasta. Kuvaustapa on yhteensopiva tietokannan hallintaohjelmiston kanssa.

Looginen kaavio yhdistää tietokannan tietosisällön ja sen taustalla olevan "knowledge basen". Se muodostetaan yleisin käsittein. Yksittäisten tietueitten ja linkkien sijaan käytetään tietuetyyppejä ja niiden keskinäisiä relaatioita. Looginen kaavio on riippumaton tietokannan toteuttamistavasta. Hierarkkista rakennetta suunniteltaessa kaaviosta pyritään muodostamaan puu. Tietokannan ylimmälle tasolle sijoitetaan yksi tietuetyyppi juureksi (root) ja siihen kytketään alemmalta tasolta yksi tai useampia tietuetyyppejä. Juuri määritellään seuraavasti (MARTIN 1977):

Määritelmä

Root. The base node of a tree structure. Data in the tree may be accessed starting at its root.

TUTKA-tiedonhallintaohjelmistolla (versio 1/1) muodostetuissa tietokannoissa voidaan kullekin tietuetyypille määritellä enintään kolme jäsentä. Mahdollisia rakenteita ovat siis esimerkiksi:



Kuva 3. Esimerkkejä hierarkkisista kaaviorakenteista.

Kuvassa 3 ensimmäisenä esitetty yksinkertainen kaaviorakenne on yhtäläinen kuvan 2 esimerkkitietokannan rakenteen kanssa. Kärjimmäisenä oikealla esitetty kaaviorakenne on samalla esimerkki TUTKA-ohjelmistolla (versio 1/1) toteutettavissa olevasta maksimirakenteesta. Tietokantaan voidaan määritellä enintään kymmenen tietuetyyppejä.

Verkkorakenteisessa tietokannassa tietuetyypeillä voi olla myös useita isäntiä. Yhden juuren sijaan tietokantaan voidaan määritellä useita loogisia sisääntuloakohtia. Rengas- ja tähtirakenteet ovat mahdollisia.

Tietokannan rakenteen kuvaamisen jälkeen kuvataan jokaisen tietuetyypin tietosisältö. Kuvaaminen tapahtuu nimeämällä tietuetyypin tietokentät eli attribuutit (attribute), jotka MARTIN (1977) määrittelee seuraavasti:

Määritelmä

Attribute. A field containing information about an entity.

Tietuetyyppien kuvaamista varten laaditussa lomakkeessa on varattu sarake tietokentän nimelle, nimen lyhenteelle, kentän maksimi- ja miniarvolle sekä puuttuvan tiedon koodille (liite C).

Kappaleen 1.3 esimerkissä on kaikkiaan kolme tietuetyyppeä, joiden kuvaukset voisivat olla seuraavat:

PUUTTIETO	KÄPYTTIETO	SIEMENTIETO
1 pituus	1 kävyn pituus	1 siemenen kuivapaino
2 läpimitta	2 kävyn paino	2 lentimen pituus
3 kunto	3 kävyn väri	3 lentimen leveys
4 ikä dl3		

Esimerkin kolme tietuetyyppeä on siis kuvattu siten, että tietuetyyppeä PUUTTIETO on kuvattu neljällä sekä tietuetyypit KÄPYTTIETO ja SIEMENTIETO kolmella attribuutilla.

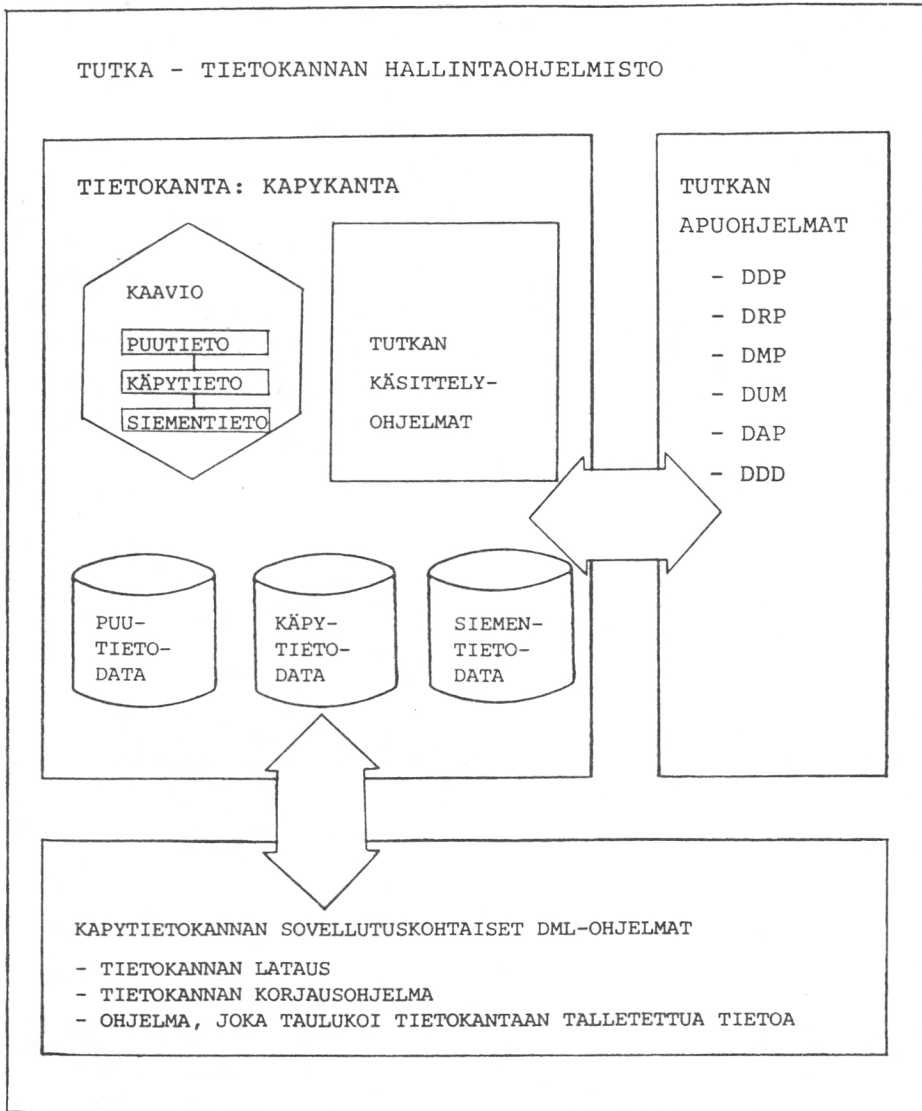
TUTKA-ohjelmisto on rakennettu erityisesti tutkimusaineistoista muodostettavien tietokantojen hallintaan. Tämän vuoksi tietokannan tietorakennevalikoima on verrattain suppea. Valitut rakenteet soveltuvat kuitenkin hyvin luonnontieteellisessä tutkimuksessa syntyvien havaintoaineistojen tiedonhallintaan. Tietorakenteiden valintaan on vaikuttanut osaltaan järjestelmän rakentamisessa käytetty ohjelmointikieli FORTRAN. Tietorakenteita on kaksi: 1) linkitetty lista numeerisia havaintovektoreita varten ja 2) indeksoidut luettelot havaintojen tulkintaa varten. Toista rakennetta käytetään myös tietokannan rakenteen kuvaamiseen iyyssisessä kaaviossa.

Havaintojen tulkinta tarkoittaa tässä sitä, että jokaisesta luokittelua edellyttävästä attribuutista muodostetaan tietokantaan talletettava luokitus. Luokitus muodostuu kaksiosaisista, \indeksin ja sanallisen tulkintavektorin muodostamista termeistä.

Esimerkkietokannan kaaviossa oli ainoastaan kaksi luokiteltavaa attribuuttia: kunto ja kävyn väri. Luokituksia vastaavat termiluettelot ovat esimerkiksi:

kunto	kävyn väri
1 terve	1 vihreä
2 sairas	2 punaruskea
3 kuollut	3 tumman ruskea
	4 harmaa
	5 musta

Termeistä muodostuvaa kokonaisuutta, termiluetteloa, on TUTKA-järjestelmän yhteydessä totuttu nimittämään "terminaaliksi". Termiluettelon muodostaminen luokiteltaville attribuuteille täydentää loogisen kaavion. Loogisen kaavion muodostamiseksi laadittu lomakkeisto sisältää myös tätä tarkoitusta varten laaditun lomakkeen (liite C).



Kuva 4. TUTKA-järjestelmä ja KAPYKANTA-tietokanta.

2.2 Fyysinen kuvaus

Fyysinen kuvaus eli fyysisen kaavion muodostaminen on ehkä enemmän TUTKA-sidonnainen käsite kuin looginen kuvaus. Sillä tarkoitetaan yksinkertaisesti kaavion toteuttamista tietokoneen muistilaitteilla. TUTKAN fyysinen kaavio muodostuu levytiedostoista ja keskusmuistiin luetuista taulukoista. MARTINin (1977) määritelmä fyysiselle kaaviolle (physical database description) on seuraava:

Määritelmä

A chart of the physical layout of the data on the storage devices. The view seen by the system programmers and the system designers who are concerned with performance and how data is positioned on the hardware, how it is indexed or located, and what compaction techniques are used.

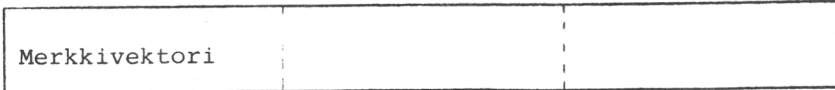
Tietokannan fyysinen kuvaus tai kaavio on siten tietokoneella tutkittavissa oleva "kartta" datan esitystavasta ja sijainnista fyysisillä muistilaitteilla. Fyysistä kaaviota tarkastellaan systeemiohjelmoijan tai -suunnittelijan näkökulmasta, joiden kannalta on oleellista kuinka systeemi toimii, kuinka data sijoittuu muistilaitteille, kuinka tieto indeksoidaan tai paikallistetaan ja millaisia pakkaustekniikoita käytetään.

MARTIN tuo edellä olevassa määritelmässään tietokannan käyttöjäröolin rinnalle tietokannan suunnittelijan ja systeemiohjelmoijan roolin. Vaikka tietokantojen käyttö esitetään helpoksi ja usein painotetaan käyttäjän riippumattomuutta talletetun tiedon rakenteesta ja muodosta, se ei vapauta tietokannan suunnittelijoita ja sovellutusohjelmien tekijöitä tietokannan fyysisen rakenteen tuntemisesta. Tämän johdosta TUTKA-tietokannan fyysinen tiedostorakenne ja tietokannan rakentuminen kuvataan yksityiskohtaisesti seuraavassa kappaleessa.

Tietokannan fyysisen kaavion muodostaminen tapahtuu TUTKA UTILITYn DDP-ohjelmalla (ks. luku I kappale 2). Ohjelma kysyy käyttäjältä kaikki loogiseen kaavioon määritellyt, tietokannan rakennetta, nimistöä ja tietuetyyppien tietosisältöä koskevat tiedot. Ohjelma on interaktiivinen ja se käyttää muodostuvan kaavion tietosisältöä myös ohjaustietona. Yksinkertaisen tietokannan kaavion muodostaminen tapahtuu DDP:llä melko nopeasti, mutta normaalin, 4-5 tietuetyyppiä sisältävän tietokannan fyysinen kuvaaminen kestää pari tuntia. DDP:n ajon yhteydessä ohjelmalle annettu syöttötieto kirjoittuu kaaviotiedostoon, joka on tunnistettavissa tyyppimerkinnästä .SCH:

DATABASE.SCH - DDP-ohjelman tuottama kaaviotiedosto.

Kaaviotiedoston tietueet ovat oletuspituuden mittaisia merkkivektoreita, jotka on talletettu formatoitua peräkkäisrakennetta käyttäen.



Kuva 5. Kaaviotiedoston tietorakenne.

Tiedostoa käytetään tarvittaessa tietokannan kaavion uudelleen generointiin. Sitä voidaan editoida normaalisti käytettävissä olevalla tekstieditorilla, joten pienten virheiden korjaaminen onnistuu kätevästi. Tietokannan rakenteen korjaaminen on myös mahdollista lokitiedoston korjaamisen avulla, mutta se edellyttää kokonaisuuden huomioonottoa ja suurta huolellisuutta.

Tietokannan fyysinen kaavio muodostuu TUTKA-ohjelmiston yhteydessä useista erillisistä tiedostoista. Kaavioon kuuluvien tiedostojen nimien alkuosa muodostuu tietokannan koko nimestä tai nimen kuudesta ensimmäisestä kirjaimesta. Tiedostot ovat

- DATABASE.NIM - sanakirja, käytetty nimistö
- DATABASE.LYH - sanakirja, käytetyt lyhenteet
- DATABASE.DEV - tiedostovaraukset
- DATABASE.AVA - vapaan tilan osoitteet
- DATABASE.LOG - huollon lokitiedosto
- DATABASE.KYS - kyselyvektoritiedosto.

Kaavion sanakirjan tietosisältö on jaettu kahteen tiedostoon. Ensimmäiseen (.NIM) on talletettu kaikki tietokannan rakennetta ja nimistöä koskevat tiedot. Toiseen (.LYH) on luetteloitu nimistön lyhenteet.

Molemmat tiedostot ovat formatoimattomia suorasaantitiedostoja. Niitä voi siis käsitellä ainoastaan TUTKAN apuohjelmilla. Nimistön ja rakenteen talletuksessa käytetyt tietorakenteet esitetään kuvissa 6 ja 7.

Attri- buutti	Lyhenne	Minimi	Maksimi	Puuttu- van tiedon koodi	Laatu	Tietue- tyyppi	Vakio- teksti "M"	Tyhjää		Datatie- dosto	MÄKREAN indeksi	MÄKREAN alku- sarake	MÄKREAN tulos- tusfor- maatti	Tyhjä
Attri- buutti	Lyhenne	Minimi	Maksimi	Puuttu- van tiedon koodi	Laatu	Tietue- tyyppi	Vakio- teksti "L"	Terminet- telotiedos- luku- määriä to	Termien merkien max. lu- kumäärä	Datatie- dosto	MÄKREAN indeksi	MÄKREAN alku- sarake	MÄKREAN tulos- tusfor- maatti	Tyhjä
Tietue- tyyppi	Lyhenne	Tie- tuiden luku- määriä	Taso	Tyhjä	Vakio- teksti "DIUF"	Vakio- teksti "RYRMIT"	Vakio- teksti "R"	Tietue- tyyppi- telotie- dosto	Tietue- tyyppien luku- määriä	Datatie- dosto	Ketjujen luku- määriä	1. Jäsen	2. Jäsen	3. Jäsen
Setti	Lyhenne	Tyhjää			Vakio- teksti "AF"	Vakio- teksti "SETIT"	Vakio- teksti "S"	Tyhjää		Jäsenen datatie- dosto	Lajitte- luovain- ten lu- kumäärä	1. Lajittelu- avain	2. Lajittelu- avain	3. Lajittelu- avain

Kuva 6. Sanakirjan tietorakenne.

Sanakirja on lajiteltu aakkosjärjestykseen nimikentän mukaan. Lyhenteet on lajiteltu aakkosjärjestykseen ja yhdistetty nimitykseen suoralla osoitteella.

Lyhenne	Osoite
---------	--------

Kuva 7. Lyhennesanakirjan tietorakenne.

Käyttäjän lokitiedostoon talletetaan tietokannan jokaisen käyttötapahtuman alkaessa ja päättyessä ajankohta- ja käyttäjätiedot. Tiedot talletetaan formatoituna merkkivektorina peräkkäisrakenteeseen, joten tietokannan käyttöä voidaan tarvittaessa seurata esimerkiksi listaamalla lokitiedosto.

O/C	Käyttö- päivä	Käyttö- aika	Käyttäjä
-----	------------------	-----------------	----------

Kuva 8. Lokitiedoston tietorakenne.

Termien tallettamisessa käytettyä tietorakennetta on käytetty hyväksi myös tietokannan rakennetta kuvaavan tiedon tallettamisessa kaavioon. Tietokannan tietue- ja settityypeistä sekä tietuetyyppien attribuuteista muodostetut luettelot on talletettu omiin "terminaaleihinsa":

DATABASE.RYH - tietuetyyppiluettelo
DATABASE.SET - settiluettelo.

Attribuuttiluettelot talletetaan tiedostoihin, joiden nimi muodostetaan tietokannan nimen kuusikirjaimisesta lyhenteestä ja R:llä alkavasta numeroidusta kirjainosasta. Numerointi tapahtuu tietuetyyppien esittelyjärjestyksessä.

DATABA.R01 - juuritietuetyypin attribuuttiluettelo
DATABA.R02 - 2. tason 1. tietuetyypin
attribuuttiluettelo
...
DATABA.R10 - alimman tason viimeisen tietuetyypin
attribuuttiluettelo
(jos tietokannassa on 10 tietuetyypistä).

Kaavion muodostamisen yhteydessä syntyvien termiluettelotiedostojen nimet muodostetaan liittämällä tietokannan nimen kuusikirjaimiseen lyhenteeseen juokseva numero.

DATABA.001 - 1. luokittelumuuttujan termitiedosto
 DATABA.002 - 2. luokittelumuuttujan termitiedosto

...

Tietuetyyppi-, setti-, attribuutti- sekä termiluetteloiden tietorakenne on sama. Tietue muodostuu kokonaislukuindeksistä ja merkkivektorista. Talletustapa on formatoimaton suora-saanti.

Indeksi	Merkkivektori	
---------	---------------	--

Kuva 9. Termitiedoston tietorakenne.

Merkkivektorin oletuspituus termeille on 75 merkkiä, mutta sitä voidaan muunnella tekemällä korjaus sanakirjaan ja luomalla ao. termiluettelotiedosto uudelleen. Merkkivektoriin voidaan määritellä loogisia kenttiä, joista yhtä voidaan käyttää lajitteluavaimena vaihtoehdoksi indeksille. Tietuetyyppi-, setti- ja attribuuttiluetteloiden merkkivektorin oletuspituus on 30 merkkiä.

2.3 Tietokannan lataus

Tietokannan muodostamisen viimeinen vaihe on tietokannan lataus. Toimenpiteellä tarkoitetaan datan vientiä varattuihin tiedostoihin tietokannan rakenteen mukaisesti. Datan syöttö voi tapahtua joko olemassa olevista tiedostoista tai pääteltä. Vaihe ei ole välttämätön. Tietokanta on valmis jo kaavion muodostamisen jälkeen, joskin se on tyhjä. Tutkimusaineistojen osalta on tietokantaan vietävä data yleensä syytä järjestää valmiiksi tallennettuun muotoon. Tietokannan lataamisen tai käytön yhteydessä syntyvät varsinaiset dataa sisältävät tiedostot, jokaista tietuetyyppiä kohden yksi. Tiedostojen nimet muodostetaan samaan tapaan kuin attribuuttiluetteleja varten muodostetut nimet. Tunnuksena käytetään nyt D-kirjainta. Numerointi vastaa tässäkin tapauksessa tietuetyyppien esittelyjärjestystä.

DATABA.D01 - juuritietuetyypin datatiedosto
 DATABA.D02 - jne.

...

DATABA.D10

Datatiedostot on toteutettu kahta erilaista tietorakennetta käyttäen. Tietueisiin talletetaan datan lisäksi sekä tiedoston sisäinen että tiedostojen välinen osoitteisto. Rakennetyypin mukaan tietueista käytetään joko nimitystä tunnustietue tai datatietue.

Tunnustietue muodostetaan jokaisen settityypin jokaista esiintymää kohti. Se sisältää neljä kenttää joihin talletetaan seuraavat tiedot:

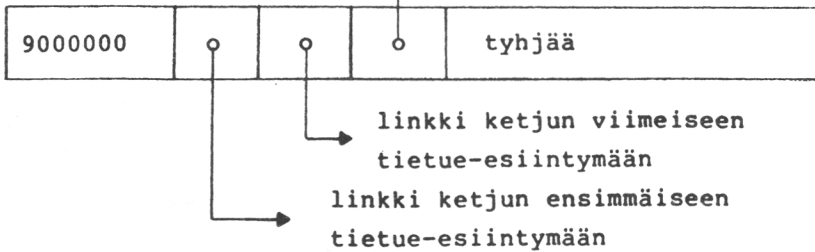
- tunnustietueen koodi 9000000 + settiin kuuluvien jäsentietue-esiintymien lukumäärä
- linkki setin ensimmäiseen jäsentietuetyypin tietue-esiintymään
- linkki setin viimeiseen jäsentietuetyypin tietue-esiintymään
- linkki setin isäntätietuetyypin tietue-esiintymään.

Datatietue-esiintymät on tunnustietueitten avulla linkitetty rengaslistaksi. Listan edellyttämä linkkikenttä on sijoitunut heti attribuuteille varattujen tietokenttien jälkeen. Tietuetyypin jäsenlinkit ovat viimeisinä jäsenten esittelyjärjestyksessä.

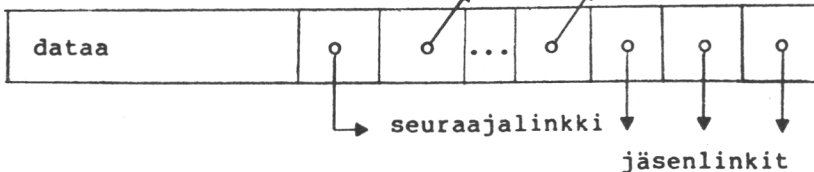
tunnus + ketjun

esiintymien lukumäärä

linkki isäntään



mahdolliset ketju-
linkit

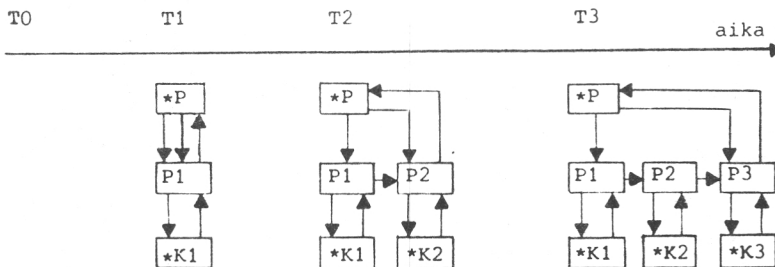


Kuva 10. Datatiedoston tietorakenteet: tunnustietue ja datatietue.

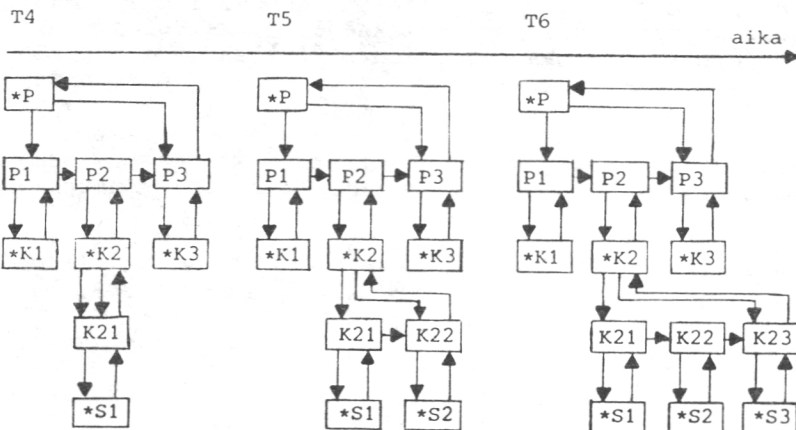
Listarakenteen seuraajalinkin ja jäsenlinkkien väliin on mahdollista sijoittaa vielä ns. ketjulinkkejä, jotka voidaan nimetä ja joiden avulla tietokantaa voidaan käyttää myös verkkorakenteen tapaan. Ketjulinkkien käsittelyohjelmien suunnittelu on kuitenkin vasta alullaan ja toteutus tehdään vain siinä tapauksessa, että TUTKA-tietokantojen käyttäjät katsovat sen tarpeelliseksi. Ketjutusominaisuuden mukaanotto lisää myös kaavion sanakirjaan uuden attribuuttityypin, ketjuattribuutin.

Latausohjelma rakennetaan DML-ohjelmia käyttäen. Ohjelmien toimintaa ja käyttöä kuvataan tarkemmin kappaleessa 3.2.2 ja luvussa III.

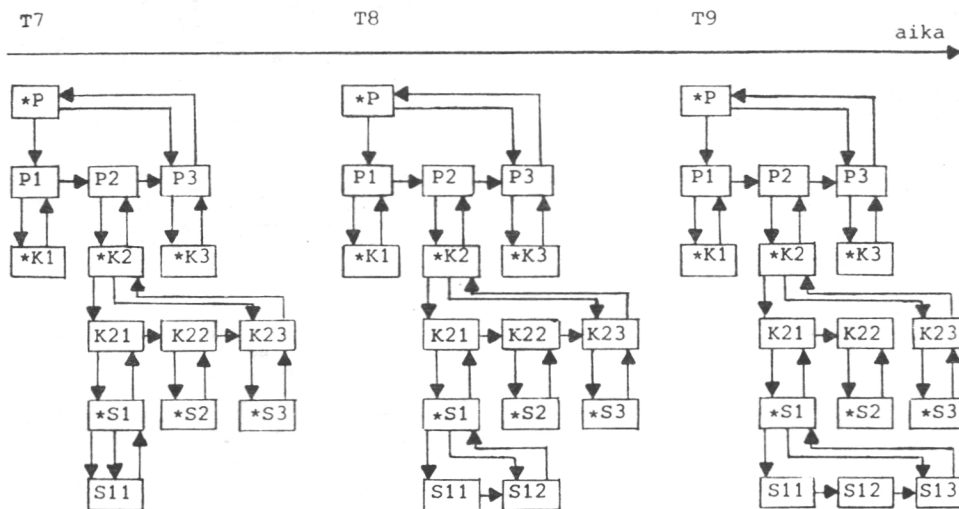
Seuraavassa kuvasarjassa on esitetty periaate, jonka mukaisesti tietokanta rakentuu latausohjelman ajon yhteydessä. Kuvasarjassa on käytetty KAPYKANTA-esimerkin mukaista tietokantaa. Eri tietuetyyppien tietue-esiintymät on merkitty kirjainsymboleilla. Tietokanta rakentuu vaihe vaiheelta siten, että ensin (kuva 11) muodostuu tyhjiin tietokantaan (T0) kolme PUUTIEETO-tietuetyypin esiintymää ja seuraavaksi (kuva 12) toiselle esiintymälle kolme KAPYTIETO-tietuetyypin esiintymää. Kuvissa on tietokannan rakennetietoa sisältävät tunnustietueet merkitty tähdellä. Ylimmällä tasolla ei ole kuin yksi tietuejoukko PUUTIEETO-tietuetyypin esiintymiä, joten sinne muodostuu vain yksi tunnustietue. Alemmilla tasoilla on kutakin ylempään tason tietue-esiintymää vastaavat tunnustietueet yksilöity indekseihin.



Kuva 11. Kolmen tietue-esiintymän luominen tyhjiin tietokantaan sen ylimmälle tasolle.



Kuva 12. Kolmen jäsentietuetyypin esiintymän lisääminen tietokannan toiselle tasolle.



Kuva 13. Kolmen jäsentietuetyypin esiintymän lisääminen tietokannan alimmalle tasolle. P-, K- ja S-tunnuksella merkityt tietue-esiintymät kirjoittuvat omiin tiedostoihinsa. Tunustietueet lomittuvat datatietueiden joukkoon.

3. TIETOKANNAN KÄYTTÖ

3.1 Yleiset ohjelmat

Metsäntutkimuksen tiedonhallinnan kehittämistarpeet liittyvät pääasiassa havaintoaineistojen dokumentointiin ja aineistojen joustavaan saantiin. Tietokantaan talletettavan tiedon oikeellisuutta ja tietokannan eheyttä korostavat kriteerit ovat olleet määrääviä TUTKA-tiedonhallintaohjelmistoa kehitettäessä. Tiedon talletukseen ja saantiin liittyvät tehokkuuskriteerit ovat olleet tähän asti toissijaisia.

Muodostettua tietokantaa voidaan käyttää joko yleisillä tiedonhallintaohjelmistoon kuuluvilla apuohjelmilla tai tietokantakohtaisilla, käyttäjän (ohjelmoija) itsensä rakentamalla ohjelmilla. Yleiset ohjelmat mahdollistavat tietokannan eriosia koskevien raporttien tuoton, datan viemisen tietokantaan, tietokannan tietosisällön editoinnin, tietokannan pakkaamisen ja uudelleenjärjestelyn sekä erilaisten kyselyjen tekemisen tietokantaan havaintomatriisin eristämiseksi tietokannan tietosisällöstä.

TUTKA-ohjelmiston (1/1-versio) yleisohjelmapaketti TUTKA UTILITY sisältää seuraavat ohjelmat:

- DDP -kaavion kuvaus
- DMP -tietokannan kaavion ja termiluetteloiden käsittely
- DRP -kaavioraportin kirjoitus
- DAP -kyselyohjelma havaintomatriisin muodostamiseksi
- DUM -tietokannan kaavion ja tiedostojen tulostus
- DDD -tietokannan tuhoaminen ja poistaminen rekisteristä.

Ohjelmien käyttöä on kuvattu yksityiskohtaisesti luvussa II.

3.1.1 Tietokannan ylläpito ja huolto

Tietokannan muodostaminen, käyttö ja ylläpito edellyttävät hyvää ohjelmallista tukea. Lataamisen jälkeen tietokantaa voidaan ryhtyä heti käyttämään. Käytön aikana tietokantaan lisätään tietueita, tietueita muutetaan ja poistetaan sekä tietosisältöön kohdistetaan hakuja. Tietosisällön muutokset ja poistot muuttavat tietueitten loogista järjestystä tietokannassa. Poistetut tietueet pysyvät levyllä tietokannan pakkaamiseen saakka. Aktiivinen tietokannan päivittäminen johtaa ennen pitkää tilanteeseen, jossa levytilan täyttöaste tulee epäedulliseksi. Loogisen rakenteen pirstoutuminen saattaa myös johtaa käytön hidastumiseen. Tietokanta kaipaa siis aika-ajoin toistuvaa huoltoa.

Tietokannan tietosisällön käsittelyyn on useimmissa järjestelmissä käytettävissä ns. tietokantaeditori, joka mahdollistaa tietokannassa tapahtuvan "navigoinnin", siis haluttujen tietueitten paikallistamisen erilaisia etenemistapoja käyttäen, sekä paikallistettujen tietueitten modifioinnin. TUTKAN osalta datan editoriohjelmisto on suunnitteilla, mutta sitä ei ole sisällytetty TUTKAN 1/1-versioon. Sen sijaan kaavion sa-

nakirjan ja termiluetteloiden käsittelyyn tarkoitettu listaus- ja korjausohjelma DMP (ks. luku II) on jo käytettävissä.

DMP-ohjelman toiminnot ovat:

- käyttöoikeuksien lisäys ja poisto
- tietuetyyppien, settien ja attribuuttien listaus, lisäys, korjaus ja poisto
- termien listaus, lisäys, korjaus ja poisto.

Käytön ja dokumentoinnin ohjelmisto täydentyy vielä listausohjelmien osalta. Tietokannan suhteellisen suuresta koosta huolimatta tulee tietosisällön osittainen tai koko "dumpaus" eli listaus aika-ajoin tarpeelliseksi. TUTKA-ohjelmiston DUM sisältää jo joitakin listausohjelmia:

- tietokannan tietue-esiintymien fyysisen rakenteen mukainen datan dumpaus (hard dump)
- sanakirjan ja lyhenneluettelon tietosisällön fyysisen rakenteen mukainen dumpaus
- termiluetteloiden fyysisen rakenteen mukainen dumpaus.

Fyysisen rakenteen mukainen dumpaus antaa tietokannan rakentajalle mahdollisuuden tarkastella tietosisältöä myös tunnus-tietueiden ja linkkien osalta.

Sanakirjan fyysisen rakenteen dumpausmahdollisuus on sisällytetty ohjelmistoon siksi, että vakavien virhetilanteiden satuessa kaavion tilaraportti saadaan mahdollisimman täydellisenä.

TUTKAN versio 1/2 tulee sisältämään loogisen rakenteen mukaisen datan dumpausmahdollisuuden.

3.1.2 Raportointi- ja valintaohjelmat

Raportointi- ja valintaohjelmilla tarkoitetaan ohjelmia, joiden avulla tietokannasta saadaan tarkoituksenmukaisia tulostuksia. Ohjelmat eivät muuta tietokannan tietosisältöä.

Raportointi tarkoittaa määrämuotoisten tulostusmallien mukaista tietokannan tietosisällön tulostamista. Raportointi palvelee toimintona jotain toista tarkoituksellista toimintoa. Tällöin jälkimmäisen toiminnon kannalta on merkityksellistä seurata tiettyjen muuttujien arvomuutoksia tietokannassa. Raportointi voi palvella myös tietokannan ylläpitoa: tietokannan tilaraportin tuottaminen säännöllisesti varmistaa luotettavan käytön. Raportointifunktio on tietokannan hyväksikäytön kannalta sitä tärkeämpi mitä useammin tietokantaan tehdään muutoksia.

TUTKAN kehittämisen yhteydessä raportointi on ajateltu sovel-lutuskohtaisesti ohjelmoitavaksi toiminnoksi. TUTKA UTILITYN DRP-ohjelmalla voidaan tulostaa vain ns. kaavioraportti.

Tutkimustoiminnan ja erityisesti metsien inventointeihin tai koetoimintaan liittyvien havaintovektoreiden tiedonhallinnassa ei tietokantaan kohdistettavilla muutoksilla ole suurta merkitystä. Pitkinä koesarjoina tehtävät toistuvat inventoinnit eivät tuota pelkkiä päivitystietoja tietokantaan, vaan jokainen tieteellinen havainto on yhtä merkityksellinen ilmiöiden ajallisia sekvenssejä luotaessa. Uudet havainnot lisätään uusina tietueina tietokantaan. Tiedonhallinnan ja tietokannan pääasialliseksi tehtäväksi jää tiedon oikeellisuuden varmistaminen, järjestyksen ylläpito havaintojen osalta ja tiedon pitäminen helposti saatavana.

Kyselyohjelmat mahdollistavat tietokannan tietosisällön selaimen erilaisten ehtojen rajaamana. Kyselyohjelmien ohjauksessa käytetyt kyselykielet (query languages) ovat yleensä korkeatasoisia ja niille on ominaista vahva loogisuus. Käyttäjän kannalta juuri kyselyohjelmat avaavat tietokantaan talletettuun tietoon monipuolisia näkökulmia. Tieteelliseen tutkimukseen liittyvä havaintovektoreista muodostettu tietokanta on tutkijalle tietovarasto, josta hän voi kyselyohjelmilla muodostaa tarvitsemansa havaintomatriisin. Matriisi on sisällöllisesti eheä: analyysin kannalta tarpeeton materiaali on karsittu ja aineistoa koskevat rajaukset ovat yksiselitteisesti ilmaistavissa.

TUTKA-ohjelmiston DAP (ks. luku II kappale 5) on verrattain pitkälle kehitetty kyselyohjelma. Se mahdollistaa havaintomatriisin valinnan havaintotietueilla täytetystä tietokannasta. DAP-ohjelmaa voidaan käyttää myös nopean palautetiedon saamiseksi tietokannan tietosisällöstä. Ohjelman tehokkuus on kuitenkin niin alhainen, että interaktiivisen palvelun saamiseksi tietokannan tietosisältö ja rakenne on koottava tarkoituksenmukaisesti. DAP-ohjelma kysyy käyttäjältä selväkielisiä ehtoja tietokannan tietosisällön rajaamiseksi. Ehtojen antamisen jälkeen käyttäjä voi vielä muotoilla syntyvää tulostiedostoa. DAP-ohjelman 1/1-versio muodostaa halutut tietokentät sisältävän matriisitiedoston lisäksi tiedoston kuvauslomakkeen, jossa myös annetut rajaukset on otettu huomioon.

3.2 Käyttäjän omat ohjelmat

Tietokannan hallintajärjestelmän valmishjelmat tarjoavat käyttäjälle yksinkertaisen tavan lähestyä tietokantaan talletettua tietomassaa. Kehittyneelle ja ohjelmointitaitoiselle käyttäjälle nämä eivät ole kuitenkaan tutkimustietokantojen yhteydessä riittävä väline täydellisen hyödyn saamiseksi tietokannasta. Valmishjelmat ovat yleisiä, ja yleisyys on usein ristiriidassa tehokkuustekijöiden kanssa. Tietokantansa tietosisällön ja rakenteen tunteva henkilö voi tarvittaessa käyttää oikoteitä navigoidessaan tietokannassa. Yleisiä ohjelmia käytettäessä tämä ei ole mahdollista.

HASEMAN ja WHINSTON (1977) sanovat kirjassaan (s. 115) seuraavasti:

"The programming user needs the capability to access specific records within the data base through the use of a collection of data base primitives or functions. These users must also be able to fetch, store, and modify data values stored within those records. The data base primitives, or functions are grouped together to form what is known as the data manipulation language (DML). (By definition, this DML must be an extension to a procedural host language.) The host language provides the necessary logical capability, while the DML extension provides for the selection of records and the access of data from those records. This class of user is concerned with the data base on the logical level of records and data items. Systems that primarily support this class of users are referred to as host language systems."

Tämä sisältää TUTKA-ohjelmiston käyttötavan perusajatuksen ja oletetun käyttäjän luonnehdinnan. Ohjelmoiva käyttäjä tarvitsee käyttöönsä valikoiman ohjelmoituja perustoimintoja, joilla tietokannan yksittäiset tietueet voidaan saavuttaa. Käyttäjän on myös kyettävä hakemaan, tallettamaan ja modifioimaan osoitettuihin tietueisiin talletettua tietoa. Perustoiminnot on koottu yhteen kokonaisuudeksi, jota nimitetään datan käsittelykieleksi (DML) (DML on käytettävissä olevan ohjelmointikielen eli ns. isäntäkielen laajennus). Isäntäkielen ominaisuuksiin kuuluvat tarvittava looginen tai laskennallinen suorituskyky (FORTRAN), kun taas DML-laajennus mahdollistaa tietokannan tietueiden valinnan ja käsittelyn. DML-ohjelmia muodostavan käyttäjäryhmän kannalta tietokannan loogisen rakenteen ja tietosisällön tuntemus riittää. Järjestelmiä, joiden käyttö tapahtuu tämän käyttäjäryhmän toimesta, kutsutaan isäntäkielijärjestelmiksi.

Käyttäjän omilla ohjelmilla tarkoitetaan juuri DML:llä muodostettuja ohjelmia, jotka palvelevat jotain tiettyä käyttötarkoitusta.

3.2.1 Tietokantaohjelmiston yleispiirteet

Jokaista muodostettua tietokantaa varten on syytä rakentaa oma tietokantakohtainen apuohjelmisto. Ohjelmisto alkaa muodostua heti tietokannan rakentamisen alkuvaiheessa, mikäli lataus suoritetaan tarkoitusta varten laadittua DML-ohjelmalla. Kunkin ohjelmiston erityispiirteet muodostuvat sovellutuskohdasta. Rakennettavien apuohjelmien tulee palvella seuraavia yleisiä toimintoja:

- tietokannan luonti ja päivitys
- tietokannan kaavion kuvaaminen
- tietokannan tietosisällön kuvaaminen
- tietokantaa hyväksikäyttävän toiminnon edellyttämät tuotanto-ohjelmat.

Tässä yhteydessä on huomionarvoista se, että tietokannan kaavion muuttaminen käyttäjän omilla ohjelmilla ei ole mahdollista.

Yleisten ohjelmien käytön kannalta suurimmaksi haittatekijäksi muodostuu ohjelmien tehottomuus. TUTKA-ohjelmiston soveltama tietorakenne, linkitetty lista, on kahdesta syystä hidas: 1) osoitetiedot eli linkit on talletettu datan kanssa samoihin tietueisiin ja 2) varsinaisen datan käsittelyn lisäksi DML-ohjelmat joutuvat käsittelemään myös tietokannan rakennetta kuvaavaa tietoa. Linkkien tallettaminen datan kanssa samaan tilaan monimutkaistaa käsittelyn logiikkaa ja rakennetietueiden käsittely moninkertaistaa levyhakuja määrän. Käyttäjän (ohjelmoijan) olisi ohjelmissaan kyettävä välttämään mahdollisimman pitkälle näiden kahden ominaisuuden vaikutuksia.

Käsittelyohjelmia voidaan nopeuttaa ottamalla huomioon keskeisimmät tiedontarpeet jo tietokannan kaavion suunnitteluvaiheessa. Toiseksi käyttäjä voi rakentaa ohjelmiinsa erilaisia hakutauluja, joihin talletetaan suoria, tietokannan fyysisiin tietueisiin osoittavia, hakuavaimia. Hakuavainten käsittely kannattaa suorittaa keskusmuistissa. Kolmas tapa edistää tietokannan nopeaa käyttöä on huolehtia siitä, että tietokannan talletusrakenne muodostuu mahdollisimman edulliseksi virtuaaliominaisuudella varustetulle tietokoneelle (VAX). Tämä tarkoittaa, että tunnistietueet ja datatietueet sijaitsevat mahdollisuuksien mukaan samoilla muistisivuilla, mikä tulee ottaa huomioon ennenkaikkea latausohjelman yhteydessä. Järjestelmän yleisohjelmiin tullaan lisäämään (versiossa 1/2) tietokannan tietosisältöä pakkaavat ohjelmat. Pakkausohjelmat ylläpitävät tietokannan optimirakennetta.

3.2.2 DML- eli datan käsittelyohjelmat

TUTKalla muodostetun tietokannan pääasiallinen käyttö tapahtuu DML-ohjelmilla. DML-aliohjelmilla ohjelmointi on perusteiltaan helppoa ja siihen liittyy vahva semanttinen ote käsiteltäviin tietoihin ja niiden takana oleviin ilmiöihin. DML:n isäntäkielenä on ohjelman prototyypivaiheessa VAX-11 FORTRAN. Varsinaiset DML-lauseet esitetään siis subroutine- tai function-aliohjelmakutsuina. Jotkin DML-aliohjelmien 1/1-version erityispiirteistä poikkeavat karkeasti standardista. Eräänä esimerkkinä tästä on merkkitiedon käsittely byte- eli logical*1-vektoreissa.

Toistaiseksi ohjelmoijalla on käytettävissään kaikkiaan 36 aliohjelmakutsua, joilla hän voi

- 1) avata ja sulkea tietokannan
- 2) luoda, muuttaa ja tuhota tietokannan tietue-esiintymiä ja termejä
- 3) saada käyttöönsä tietoa sekä tietokannan kaaviosta että tietue-esiintymistä
- 4) navigoida tietokannan rakenteessa
- 5) käyttää ulkoisia arvoja navigoinnin ohjaimina ja loogisina testiarvoina
- 6) seurata ohjelman etenemistä tietokannan rakenteessa.

DML-aliohjelmat muodostavat tietokannan käyttöön niukan mutta riittävän peruskäskyvalikoiman. Ohjelmien käyttöön perehtynyt henkilö kykenee nopeasti muodostamaan tarpeisiinsa soveltuvia makrorakenteita.

II TIETOKANNAN KASITTELYN APUOHJELMAT (TUTKA UTILITY)

1. APUOHJELMISTON YLEINEN KUVAUS

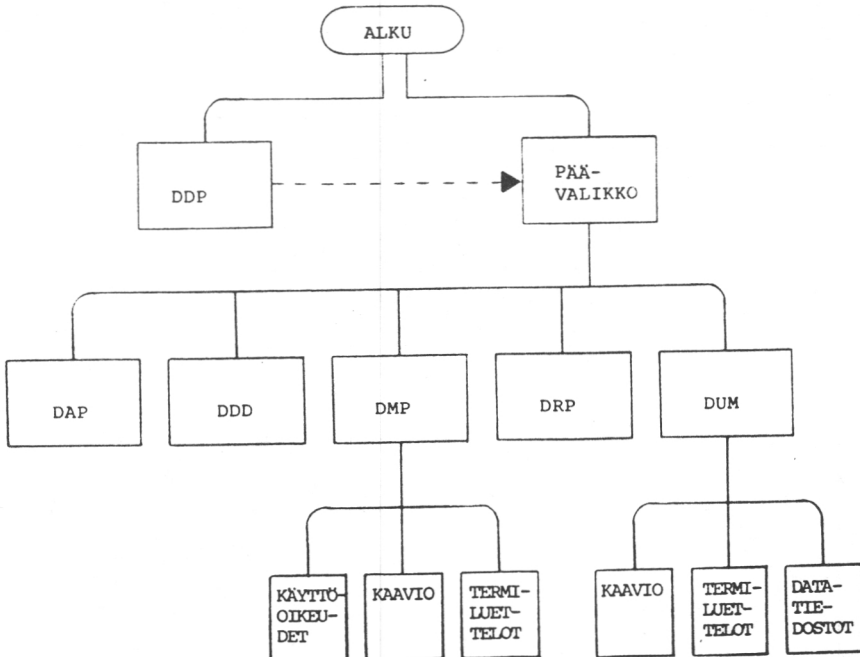
1.1 Ohjelmiston tarkoitus

TUTKA-tiedonhallintaohjelmiston apuohjelmien tarkoitus on palvella tietokannan käyttäjiä ja rakentajia yleisinä apuohjelminä. Ohjelmien avulla on mahdollista luoda, ylläpitää ja tuhota tietokantoja. Niiden perusominaisuus on yleisyys: mikä tahansa TUTKA-tiedonhallintaohjelmistolla toteutettava tietokanta voidaan luoda ja ylläpitää apuohjelmilla. Apuohjelmien käyttö on kannattavaa tietokantaa luotaessa, kaaviota ylläpidettäessä, dumpauksessa sekä yksinkertaisissa aineiston ha-uissa. Muussa yhteydessä käyttäjän omat tietokantakohtaiset DML-ohjelmat ovat tehokkaampia (ks. luku 1 kappale 4.2).

1.2 Ohjelmiston looginen rakenne

Apuohjelmat on koottu yhdeksi kokonaisuudeksi: haluttu ohjelma voidaan valita päävalikosta. Integrointi nopeuttaa ja helpottaa ohjelmiston käyttöä. Ohjelmisto koostuu useista hierarkkisesti järjestetyistä osista (kuva 14). Käyttäjä on yhteydessä eri osiin näyttöjen välityksellä. Kutakin ohjelmaa eli toimintoa edustaa yksi näyttö, josta voidaan palata ylemmälle tasolle, jatkaa alemmalle hierarkiatasolle tai suorittaa toimenpide.

Näyttöjen käsittely aiheuttaa laiteriippuvuutta. Ohjelma toimii tehokkaasti VT100 ja VDU210 päätteillä.



Kuva 14. Ohjelmiston looginen rakenne.

1.3 Ohjelmiston käynnistäminen ja päähakemisto

Ohjelmisto käynnistetään komennolla "TUTKA". Aluksi ohjelmalle annetaan sisääntulotietoa: käsiteltävän tietokannan nimi, käyttäjätunnus ja salasana.

Mikäli ajon tarkoitus on luoda tietokanta, vastataan systeemin eli tietokannan nimeä tiedusteltaessa kysymykseen

System >

C:llä (create), jolloin DDP-prosessori käynnistyy. Onnistuneen luonnin lopuksi palataan kysymykseen "System >". Ohjelmiston käynnistämiseen liittyvä näyttö on esitetty kuvassa 15. Käynnistyksen yhteydessä ohjelma lataa tietokannan ja tarkistaa sen rakenteen loogisuuden. Onnistuneen latauksen jälkeen ohjelma tyhjentää näytön ja tulostaa päähakemiston (kuva 16). Mikäli ohjelma havaitsee tietokannan rakenteessa virheitä, se antaa tarpeelliset virheilmoitukset ja siirtyy kuvissa 17 ja 18 esitettyjen näyttöjen kautta päähakemistoon.

```
TUTKA UTILITY 1/1
```

```
System >kapykanta
```

```
User/Password >Accepted. Now loading "kapykanta", please wait.
```

Kuva 15. Ohjelmiston käynnistäminen. Tietokannan lataaminen kestää kaavion koosta riippuen puolesta viiteen minuuttiin.

```
TUTKA UTILITY 1/1
```

```
kapykanta
```

```
Choose utility you want to use
```

```
Data Access Program ..... DAP
Delete Data Description ..... DDD
Data Editing Program ..... DEP
Database schema Manipulation Program ..... DMP
Database Packing Program ..... DPP
Database Reporting Program ..... DRP
Dumping Program ..... DUM
```

```
Please enter your choice >
```

Kuva 16. Päävalikko. Toimintatilasta haaraudutaan eri ohjelmiin tai lopetetaan apuohjelmiston käyttö.

```
TUTKA UTILITY 1/1      kapykanta

System >kapykanta
User/Password >Accepted. Now loading "kapykanta", please wait.
dbinit: List of attributes of record type "kapytieto" is incorrect
```

Kuva 17. Tietokannan rakenteessa on havaittu loogisia virheitä.

```
TUTKA UTILITY 1/1      kapykanta

                                LOAD DONE!
                                ERRORS DETECTED!
                                Structure of database is incorrect.

                                To continue type <cr> >
```

Kuva 18. Tietokannan rakenteellisista virheistä huomautetaan.

2. TIETOKANNAN LUONTI (DDP)

2.1 Ohjelman tarkoitus

DDP-ohjelmalla (Data Description Program) voidaan luoda tietokannan fyysinen kaavio. Samalla ohjelma tarkistaa kuvauksen oikeellisuuden sekä antaa tarvittavat virheilmoitukset.

2.2 Ohjelman käyttöohje

DDP-ohjelma käynnistyy, kun käyttäjä vastaa kuvan 15 kysymykseen

System >

kirjaimella C (create). Onnistuneen luonnin jälkeen palataan kuvan 15 näyttöön. DDP-ohjelmassa on kaksi peräkkäistä vaihtetta: 1) kaavion kuvaus tiedostoon (DATABASE.SCH) ja 2) fyysisen kaavion luonti em. tiedoston perusteella. Mikäli tietokannan kaaviota ei ole aikaisemmin luotu, kyselee ohjelma tarvittavat tiedot ja tallettaa ne tiedostoon, joka nimetään tietokannan nimellä ja .SCH-tyyppiseksi. Kun tämä tiedosto on luotu, DDP generoi siitä tietokannan fyysisen kaavion: sanakirjan, lyhennesanakirjan jne. Samalla DDP tarkistaa määrittysten loogisuuden ja informoi käyttäjää mahdollisista virhetilanteista. Mikäli virhetilanteita esiintyy siinä määrin, ettei niitä ole tarkoituksenmukaista korjata DDP-ohjelmalla, voidaan DATABASE.SCH-tiedosto korjata tavallisella editorilla. Tiedostoa voidaan käyttää tietokannan uudelleen generointiin DDP-ohjelmalla. Tällöin lisätään DATABASE.SCH-tiedostossa olevan tietokannan nimen perään vipu U kauttaviivalla nimestä erotettuna:

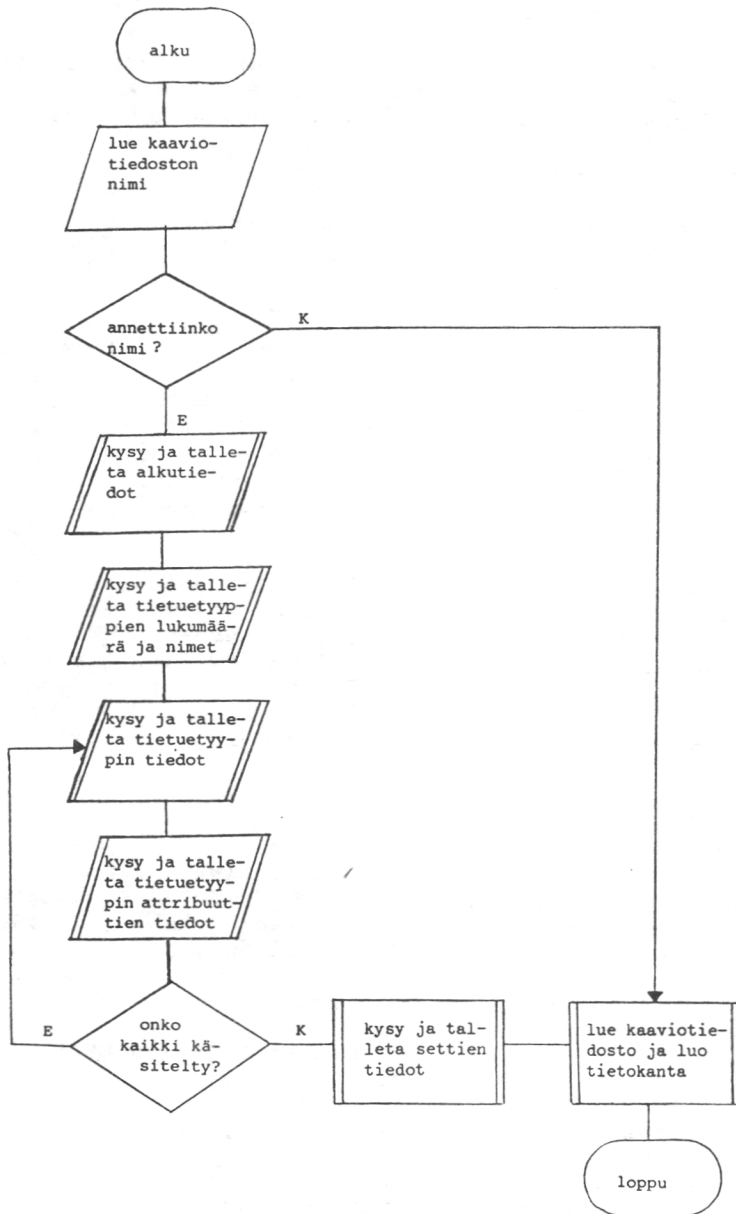
database/U.

Jos tietokanta on poistettu rekisteristä, vipua ei tarvita. Ennen uudelleen generointia on syytä tuhota edellisen generoinnin yhteydessä syntyneet, tietokantaan liittyvät tiedostot, ei kuitenkaan DATABASE.SCH-tiedostoa.

2.2.1 Kaavion kuvaaminen DDP-ohjelmalla

Tietokannan rakenne saadaan kuvatuksi ohjelman kyselemien tietojen avulla. Rakenteeseen kuuluvat

- tietuetyypit
- setit
- tietuetyyppien attribuutit
- luokiteltavien attribuuttien luokitukset
- muut attribuutteihin liittyvät tiedot: minimi, maksimi, jne.
- lajitteluvaimet eli attribuutit, joiden mukaan tietuetyypin tiedueet lajitellaan.



Kuva 19. DDP-ohjelman lohkokaaaviorakenne.

Seuraavassa esitetään ohjelman kysymykset ja niiden selitykset.

File of schema >

Mikäli kaavio luodaan ensimmäistä kertaa, painetaan return-näppäintä. Ohjelma alkaa tällöin kysellä tarvitsemiaan tietoja. Tiedot talletetaan tiedostoon DATABASE.SCH, missä "DATABASE" on tietokannan nimi. Mikäli kaavio on jo luotu ja tietokanta halutaan generoida nyt uudelleen, annetaan kaavion sisältävän tiedoston nimi (esim. File of schema >KAPYKANTA.SCH). Tällöin ohjelma lukee tarvitsemansa tiedot annetusta tiedostosta ja generoi tietokannan.

Name of database >

Kaavion luonti on alkanut. Aluksi kysytään luotavan tietokannan nimi. Nimi on 6-9 merkkiä pitkä.

User/Password >

Tietokannan omistaja kirjoittautuu käyttäjäksi ilmoittamalla haluamansa käyttäjä/salasana yhdistelmän. Omistajalle annetaan kaikki tietokannan käsittelyoikeudet, siis käyttöoikeus l. Käyttäjä/salasana yhdistelmän maksimipituus on 19 merkkiä.

Estimated size of dictionary >

Ohjelma tarvitsee sanakirjan arvioidun koon varatakseen tiedostoja. Koko voidaan ilmoittaa tietuetyyppien ja attribuuttien määrän summana. Ohjelma suorittaa varmistuksen lisäämällä annettuun kokoon n. 20 %.

Number of record types >

Ilmoitetaan tietokannan tietuetyyppien lukumäärä.

Name of record type n >

Ohjelma kysyy kaikkien tietuetyyppien nimet. Nimet annetaan yksi kerrallaan ylimmältä tasolta aloittaen.

Abbreviation of "record type" >

Annetaan lainausmerkeissä olevan tietuetyypin lyhenne. Lyhenne saa olla 1-4 merkkiä pitkä.

Stratum >

Annetaan käsiteltävän tietuetyypin taso tietokannassa.

Maximum number of records to be stored >

Annetaan tietuetyypin tietueiden maksimilukumäärä tietokannassa. Lukumääräarvio saa olla runsas, sillä myös tunnustietueiden varaama tila on huomioitava tässä vaiheessa.

Number of attributes >

Annetaan käsiteltävään tietuetyypiin liittyvien attribuuttien lukumäärä. Attribuutteja tulee olla vähintään 4.

Number of members >

Annetaan käsiteltävän tietuetyypin jäsenten lukumäärä. Jäseniä saa olla 0-3.

Name of first member of record type >

Name of second member of record type >

Name of third member of record type >

Ohjelma kyselee tietuetyypin jäsenien nimet. Nimet annetaan kokonaisina, lyhenteitä ei saa käyttää.

Name of attribute number n >

Annetaan tietuetyypiin liittyvien attribuuttien nimet yksi kerrallaan.

Abbreviation of attribute "attribute" >

Annetaan kyseessä olevan attribuutin lyhenne. Lyhenne saa olla 1-4 merkkiä pitkä.

Minimum >

Annetaan käsiteltävän attribuutin minimiarvo.

Maximum >

Annetaan käsiteltävän attribuutin maksimiarvo.

Code of missing data >

Annetaan käsiteltävään attribuuttiin liittyvä puuttuvan tiedon

koodi.

Unit of measurement >

Annetaan käsiteltävän attribuutin mittayksikkö (esim: m, v, kW tai MJ). Yksikkö saa olla 0-6 merkkiä pitkä.

Type of attribute >

Annetaan käsiteltävän attribuutin tyyppi. Mikäli attribuutti on mitattava suure kuten paino tai pituus, on tyyppi M. Mikäli attribuutti on luokiteltava suure kuten puulaji tai metsätyyppi, on tyyppi L. Tällöin on annettava jatkossa attribuutin luokitukset ja indeksit.

Output format >

Annetaan formaatti, jolla käsiteltävä attribuutti on tulostettavissa tietokannasta. Jos esimerkiksi attribuutin maksimiarvo on 100, annetaan formaatti i3. Formaattia annettaessa on syytä muistaa, että miinusmerkki vaatii oman kenttensä. TUTKAN versiossa 1/1 on mahdollista käsitellä datatiedoistoissa sekä kokonaisluku- että reaaliuuttujia.

Input of classes F/A/M >

Käsiteltävän attribuutin tyyppi on ollut L ja ohjelma haluaa nyt luokitukset. Luokitukset voidaan syöttää kolmella eri tavalla:

- | | |
|---------------|---|
| F (File) | Luokitus syötetään tiedostosta. Ohjelma kysyy silloin tiedoston nimen sekä formaatin, jolla indeksit ja luokkanimet eli termit voidaan lukea. |
| A (Automatic) | Luokat syötetään päätteltä jolloin ohjelma suorittaa indeksoinnin numerosta 1 alkaen. |
| M (Manual) | Samoin kuin edellä A, mutta käyttäjä antaa myös indeksit. |

Abbreviation of "set" >

Annetaan kyseisen setin lyhenne. Lyhenne saa olla 1-4 merkkiä pitkä.

Number of sortkeys of member of set >

Annetaan käsiteltävän setin jäsenen lajitteluavainten luku-

määrä. Lajitteluavaimet ovat attribuutteja, joiden mukaan jäsentietuetyypin tietueet lajitellaan. Maksimimäärä on 3.

```
Name of first  sortkey of member of set >
Name of second sortkey of member of set >
Name of third  sortkey of member of set >
```

Annetaan käsiteltävän setin jäsentietuetyypin lajitteluavainten nimet.

```
Number of sortkeys of record type "record type" >
```

Annetaan tietokannan ylimmän tason tietuetyypin lajitteluavainten lukumäärä.

2.2.2 Tietokannan generointi kaaviosta

Kun tietokannan kaavio on kuvattu DDP-ohjelmalla, ohjelma yrittää tietokannan generointia. Samalla tarkistetaan syntynyt kaavio. Korjaukset voidaan tehdä joko DMP-ohjelmalla tai editoimalla kaaviotiedostoa. Jos korjaus suoritetaan kaaviotiedostoon, tietokanta on generoitava uudelleen. Tämä tapahtuu vastaamalla DDP-ohjelman kysymykseen

```
File of schema >
```

kaaviotiedoston nimellä. Mikäli tietokannan generointi edellisellä kerralla onnistui kuitenkin siten, että tietokannan nimi (ym. tietoa) vietiin yleiseen rekisteriin, on uudelleen generoitaessa käytettävä vipua U tietokannan nimen perässä. Vipua lisätään DATABASE.SCH-tiedostossa olevan tietokannan nimen perään:

```
database/U.
```

Tällöin on syytä tuhota generointiyrityksessä syntyneet tietokantaan liittyvät tiedostot, ei kuitenkaan DATABASE.SCH:ta.

2.3 Esimerkki

Esimerkkinä luodaan tietokanta KAPYKANTA. Esimerkkiin liittyvät näytöt on esitetty kuvissa 20-30. Tietokannan luonnin ajan kuvaputki on liukuvan näytön tilassa. Tiedosto KAPYKANTA.SCH on esitetty liitteessä A.

```
TUTKA UTILITY 1/1
```

```
System >
```

```
If creating new system type: C
```

Kuva 20. Ohjelmiston käynnistämisen yhteydessä valitaan luodaanko uusi tietokanta vai käsitelläänkö olemassa olevaa tietokantaa.

```
TUTKA UTILITY 1/1
```

```
System >c
```

```
If creating new system type: C
```

Kuva 21. Vastauksella C käynnistetään kaavionluontiprosessi DDP.

TUTKA UTILITY 1/1

For help information type "?"

File of schema (if not type <cr>) >
 Name of database >kapykanta
 User/Password >m/m
 Estimated size of dictionary >80
 Number of record types >3

Now enter names of record types.
 Enter them hierarchically from top down
 Maximum length is 15, minimum is 5 characters

Name of record type number 1 >puutiето
 Name of record type number 2 >kapytieto
 Name of record type number 3 >sientieto

Now enter information and attributes of record type: "puutiето".

Abbreviation >

Kuva 22. Kaavionluonti on aloitettu. DDP-ohjelma on varustettu HELP-toiminnolla.

TUTKA UTILITY 1/1

Abbreviation >puu
 Stratum >1
 Maximum number of records to be stored >100
 Number of attributes >2
 Number of members >1
 Name of first member of record type >kapytieto

Now enter names of attributes of record type.

Name of attribute number 1. >puun pituus
 Name of attribute number 2. >puun ika

Now enter information of attribute "puun pituus"

Abbreviation >pit
 Minimum >1
 Maximum >30
 Code of missing data >-1
 Unit of measurement >m
 Type of attribute (M/L) >

Kuva 23. Tietuetyypit käsitellään hierarkkisessa järjestyksessä.

```
TUTKA UTILITY 1/1
```

```
Type of attribute (M/L) >m
Output format >i3
```

```
Now enter information of attribute "puun ika"
```

```
Abbreviation >ika
Minimum >1
Maximum >100
Code of missing data >-1
Unit of measurement >v
Type of attribute (M/L) >m
Output format >i3
```

```
Now enter information and attributes of record type: "kapytieto".
```

```
Abbreviation >kap
Stratum >2
Maximum number of records to be stored >1000
Number of attributes >2
Number of members >1
Name of first member of record type >
```

Kuva 24. Tietuetyyppiin kuuluvat attribuutit käsitellään peräkkäin, jonka jälkeen siirrytään seuraavaan tietuetyyppiin.

```
TUTKA UTILITY 1/1
```

```
Name of first member of record type >siementieto
```

```
Now enter names of attributes of record type.
```

```
Name of attribute number 1. >kavyn paino
Name of attribute number 2. >kavyn vari
```

```
Now enter information of attribute "kavyn paino"
```

```
Abbreviation >pain
Minimum >1
Maximum >300
Code of missing data >-1
Unit of measurement >g
Type of attribute (M/L) >m
Output format >i3
```

```
Now enter information of attribute "kavyn vari"
```

```
Abbreviation >vari
Minimum >
```

Kuva 25. Tietuetyypin KAPYTIETO attribuuttien syöttäminen.

TUTKA UTILITY 1/1

Minimum >1
 Maximum >3
 Code of missing data >-1
 Unit of measurement >
 Type of attribute (M/L) >1
 Output format >i2

Now enter classification of attribute.

Input of classes F/A/M >a
 Number of classes >3
 Name of class number 1 >ruskea
 Name of class number 2 >punaruskea
 Name of class number 3 >harmaa

Now enter information and attributes of record type: "siementieto".

Abbreviation >sie
 Stratum >3
 Maximum number of records to be stored >10000
 Number of attributes >

Kuva 26. Luokiteltavan attribuutin luokat voidaan syöttää kolmella eri tavalla.

TUTKA UTILITY 1/1

Number of attributes >2
 Number of members >0

Now enter names of attributes of record type.

Name of attribute number 1. >siemenen pituus
 Name of attribute number 2. >lentimen pituus

Now enter information of attribute "siemenen pituus"

Abbreviation >spit
 Minimum >1
 Maximum >10
 Code of missing data >-1
 Unit of measurement >mm
 Type of attribute (M/L) >m
 Output format >i2

Now enter information of attribute "lentimen pituus"

Abbreviation >

Kuva 27. Vuorossa olevan tietuetyypin attribuutit syötetään normaalisti.

```
TUTKA UTILITY 1/1
```

```
Now enter information of attribute "lentimen pituus"
```

```
Abbreviation >lpit
Minimum >1
Maximum >30
Code of missing data >-1
Unit of measurement >mm
Type of attribute (M/L) >m
Output format >i2
```

```
Now enter information of sets
```

```
Abbreviation of set "puutieto-kapytieto" >puka
Number of sortkeys of member of set >0
Abbreviation of set "kapytieto-sientieto" >kasi
Number of sortkeys of member of set >0
```

```
Now enter sortkeys of 1st record type
```

```
Number of sortkeys of record type "puutieto" >1
Name of first sortkey >puun ika
```

Kuva 28. Lopuksi annetaan settejä koskevat tiedot sekä määrätään attribuutit, joiden mukaan ao. tietuetyypin esiintymät lajitellaan.

```
TUTKA UTILITY 1/1
```

```
Now creating database, please wait.
```

Kuva 29. Kun kaikki tiedot on annettu, ryhtyy ohjelma generoimaan fyysistä tietokantaa.

```
TUTKA UTILITY 1/1
```

```
System >kapykanta
```

```
If creating new system type: C
```

Kuva 30. Mikäli generointi onnistuu, palataan alkutilaan. Tietokanta on nyt valmis, joskin tyhjä.

2.4 Rajoitukset

DDP-ohjelman ja TUTKAN käyttö yleensä sisältää muutamia rajoituksia, joihin on syytä tutustua ennen ohjelmien käyttöä.

Sanakirjan maksimikoko, siis attribuuttien ja tietuetyyppien lukumäärä on 400 TUTKAN versiossa 1/1. Tietuetyyppien maksimilukumäärä on 10.

Tietuetyyppien, attribuuttien ja settien nimien ja lyhenteiden on oltava yksiselitteisiä. Kahta samaa nimeä tai lyhennettä ei saa esiintyä.

Tietuetyyppien nimien tulee olla vähintään 5 merkkiä ja enintään 15 merkkiä. Nimen tulee alkaa kirjaimella.

Attribuuttien ja settien nimien tulee olla vähintään 5 merkkiä ja enintään 30 merkkiä. Nimen tulee alkaa kirjaimella.

Lyhenteet saavat olla enintään 4 merkkiä pitkiä. Lyhenteen tulee alkaa kirjaimella.

Tietokannan nimen tulee olla 6-9 merkkiä pitkä aakkosista tai numeroista koostuva merkkiyhdistelmä. Se ei saa alkaa numerolla eikä siinä saa olla skandinaavisia kirjaimia.

Nimissä ei yleensä tule käyttää muita kuin aakkosnumeerisia merkkejä. Esimerkiksi merkit '/' ja '-' ovat kiellettyjä. Nimen tulee alkaa kirjaimella.

Attribuutin mittayksikkö saa olla korkeintaan 6 merkin pituinen.

TUTKAN 1/1-versiossa ovat nimet "RYHMÄT" ja "SETIT" sekä lyhenteet "A", "R", "S" ja "ALL" varattuja.

2.5 Virheilmoitukset

*Abbreviation "abbr" is used duplicately

Lyhenne "abbr" on annettu kahdesti. Samaa lyhennettä tai nimeä ei saa käyttää useammin kuin kerran.

*Allowed types of dictionary records are M, L, R and S

Attribuutille ilmoitettu tyyppi ei ole sallittu. Sallittuja ovat tyypit M, L, R ja S. Näistä käyttäjä tarvitsee vain tyyppejä L ja M, (L = luokiteltava, M = mitattava).

*Attribute is not of type L.

File of classification cannot be created.

Luokittelutiedostoa yritetään luoda attribuutille, joka on tyyppiä M. Tiedostoa ei varata eikä luoda. Vian aiheuttavat todennäköisesti väärät tiedot kaaviotiedostossa (DATABASE.SCH).

*Dictionary is out of limits.

Consult the system manager.

Sanakirja on ylittänyt suurimman sallitun kokonsa tai on alimittainen. Tietokanta on generoitava uudelleen. Yhteydenotto TUTKA-ohjelmiston ylläpitäjään on tarpeen.

*Duplicate definition to "attribute".

Attribuutti "attribuutti" on nimetty kahdesti. Nimen muuttaminen editorilla DATABASE.SCH-tiedostoon korjaa virheen. Nimen tulee olla yksiselitteinen.

*Error, try again.

Syöttölause on virheellinen, yritä uudelleen.

*File not readable.

Annettu tiedosto on virheellinen tai puuttuu. Termiluetteiloita sisältävät syöttötiedostot on koottava samaan hakemis-

toon, jossa tietokantaa generoidaan.

*File empty or missing.

Annettu tiedosto on tyhjä tai puuttuu.

*Input error in given file.

Annetun tiedoston luvussa on tapahtunut konversiovirhe. Tiedosto sisältää esimerkiksi merkkitietoa, kun ohjelma yrittää lukea kokonaislukuja.

*Class too long.

Syötetty luokka on liian pitkä. Luokan maksimipituus on 75 merkkiä.

*Maximum number of databases already in use.
Consult the system manager.

Tietokantoja on luotu maksimimäärä. Maksimimäärää voidaan kasvattaa. Yhteydenotto TUTKA-ohjelmiston ylläpitäjään on tarpeen.

*Member type "member" is not defined in dictionary.

Käsiteltävän setin jäsentietuetyyppejä "member" ei ole määritelty sanakirjassa.

*Name of database incorrect.

Tietokannalle annettu nimi on virheellinen. Nimen tulee olla vähintään 6, enintään 9 kirjaimesta ja/tai numerosta muodostuva aakkosnumeerinen merkkisarja. Nimen tulee alkaa kirjaimella. Nimessä ei saa olla skandinaavisia kirjaimia.

*Name "name" already in use.

Tietokannalle annettu nimi on jo käytössä. TUTKA-tietokantojen nimien tulee olla yksiselitteisiä.

*Password "password" incorrect.

Annettu käyttäjätunnus/salasanayhdelmä on virheellinen. Käyttäjätunnus on erotettava salasanasta kauttaviivalla. Yhdelmän maksimipituus on 19 merkkiä.

*Please try again, failed in decoding.

Ohjelma ei ole onnistunut dekodauksessa ja vika on todennäköisesti syöttötietueessa. Syöttötieto on annettava uudelleen.

*Unable to create dictionary.

Sanakirjan luonti ei onnistu. Syy selviää aikaisemmasta virheilmoituksesta.

*Unable to store classification of attribute "attribute".

Attribuutin "attribuutti" luokittelutiedoston luonti ei onnistu.

*Use comma to separate index from class.

Käytä pilkkua erottimenä syöttäessäsi indeksiä ja luokkaa päätteeltä.

2.6 HELP-komento

Useimmat DDP:n kysymyksistä on varustettu HELP-komennolla. HELP-tulostus tuotetaan syöttämällä kysymykseen vastaukseksi kysymysmerkki:

File of schema >?

If you are now creating schema then press return.

If you have already created it and you will now generate the database again then enter name of file of schema.

File of schema >

3. TIETOKANNAN KAAVION MANIPULOINTI (DMP)

3.1 Ohjelman tarkoitus

DMP-ohjelmalla (Database schema Manipulation Program) voidaan korjailla generoidun tietokannan kaaviota ja termiluetteloja sekä lisätä tai poistaa tietokannan käyttäjiä. Käyttöoikeuksien muuttaminen on myös mahdollista.

3.2 Ohjelman käyttöohje

DMP-prosessori käynnistetään antamalla päävalikossa komento "DMP". Ohjelmaa voivat käyttää ainoastaan käyttöoikeuden l omaavat käyttäjät. Käynnistyksen jälkeen valitaan käsittelyn kohde. Valintaan liittyvä näyttö on esitetty kuvassa 31.

3.2.1 Kaavion käsittely

Vastaamalla kuvan 31 näytön kysymykseen "D" (dictionary) päästään kaavion sanakirjan manipulointitilaan. Aluksi määritetään joko sanakirjan fyysinen tietue tai sen attribuutin nimi, johon käsittely kohdistetaan. Tämän jälkeen päätteelle tulostuu ao. tietue, kukin kenttä eri riville. Rivit numeroidaan, ja käyttäjä voi osoittaa haluamansa rivin numerolla. Kursori siirtyy osoitetulle, riville ja kyseiseen kenttään voidaan antaa uusi tieto. Mikäli muutos kohdistuu attribuutin nimeen tai lyhenteeseen, suoritetaan sanakirjan uudelleenlaajittelu. Kaavion manipulointitilasta poistutaan painamalla return-näppäintä.

3.2.2 Käyttäjien lisääminen ja poistaminen

Vastaamalla kuvan 31 näytön kysymykseen "A" (authorities), siirrytään käyttöoikeuksien manipulointitilaan. Käyttäjä voi lisätä tai poistaa käyttöoikeuksia. Lisäys tapahtuu komennolla "A" (add) ja poisto komennolla "D" (delete). Toimintovaihtoehdon valinnan jälkeen ohjelma kysyy käyttäjätunnus/salasana -yhdistelmän, johon valittu toiminto kohdistuu. Lisäystilanteessa uudelle käyttäjälle annetaan käyttöoikeus seuraavasti:

- | | |
|------------|--|
| 1 | kaavion käsittely ja tietueiden tuhoaminen |
| 1, 2 | kirjoittaminen tietokantaan |
| 1, 2, 3 | tietokannan lukeminen |
| 1, 2, 3, 4 | valmisohjelmien käyttö. |

Käyttöoikeus 1 sisältää siis kaikki oikeudet. Käyttöoikeuksien lisäystilasta poistutaan return-näppäintä painamalla.

3.2.3 Termiluetteloiden käsittely

Termiluetteloja päästään manipuloimaan vastaamalla kuvassa 31 esitetyn näytön kysymykseen "C" (classifications). Tällöin ohjelma tulostaa päätteelle uuden sivun, josta valitaan haluttu toiminto: lisäys, poisto, muutos tai tutkiminen. Mikäli käyttäjä ei tiedä esimerkiksi poistettavan indeksin ja luokan fyysistä tietueen numeroa, on se ensin tutkittava käyttämällä toimintoa "E" (examine). Kun tietueen numero on sel-

villä valitaan haluttu käsittelytoiminto, esimerkiksi "D" (delete). Ohjelma kysyy tietueen numeron ja suorittaa toiminnon. Termiluetteloiden manipulointitilasta poistutaan painamalla return-näppäintä.

3.3 Esimerkki

Esimerkki koostuu kahdesta osasta: aluksi muutetaan tietue-tyypin KAPYTIETO attribuuttien lukumäärä 4:ksi. Sen jälkeen annetaan tietokannan käyttöoikeus Markku-nimiselle henkilölle. Salasanaksi hänelle annetaan MAS. Käyttäjä varustetaan käyttöoikeudella 1. Em. toimintoihin liittyvät näytöt on esitetty kuvissa 31-37.

3.4 Rajoitukset

Käyttäjän on itse huolehdittava DMP-ohjelman käytön yhteydessä kaavion loogisen rakenteen eheydestä. Esimerkiksi attribuutin nimeä muutettaessa korjaus on tehtävä sekä sanakirjaan että vastaavan tietue-tyypin attribuuttiluetteloon.

3.5 Virheilmoitukset

Itse DMP-ohjelman tuottamia virheilmoituksia on vain muutamia ja ne ovat yksiselitteisiä. Muilta osin virheilmoitukset ovat DML-ohjelmien antamia virheilmoituksia (ks. luku III kappale 4).

```

T U T K A   U T I L I T Y   1/1           kapykanta

You can choose one of following actions:

Manipulate authorities ..... A
Manipulate data of dictionary ..... D
Manipulate files of classifications ..... C

Please enter your choice >d

```

Kuva 31. DMP-ohjelmalla voidaan manipuloida käyttöoikeuksia, sanakirjaa ja termiluetteloita.

```

TUTKA UTILITY 1/1      kapykanta

Attribute or record number >kapytieto

```

Kuva 32. Sanakirjan manipuloinnissa määritetään käsiteltävä tietue joko numerolla tai attribuutin nimellä.

```

TUTKA UTILITY 1/1      kapykanta

Record number:      4

1. Record type ..... kapytieto
2. Abbreviation ..... kap
3. Maximum number of records ..... 1000
4. Stratum ..... 2
5. Type of file of data ..... DIUF
6. Object ..... RVHM[T
7. Type ..... R
8. File of attributes ..... kapyka.R02
9. Number of attributes ..... 2
10. Maximum length of name of attribute ... 75
11. File of data ..... kapyka.D02
12. Chains (not allowed in this version) .. 0
13. First member of record type ..... siementieto
14. Second member of record type .....
15. Third member of record type .....

Number of line to be manipulated >9

```

Kuva 33. Muutettava rivi ilmoitetaan rivin numerolla.

```

TUTKA UTILITY 1/1          kapykanta

Record number:    4

1. Record type ..... kapytieto
2. Abbreviation ..... kap
3. Maximum number of records ..... 1000
4. Stratum ..... 2
5. Type of file of data ..... DIUF
6. Object ..... RYHM[T
7. Type ..... R
8. File of attributes ..... kapyka.R02
9. Number of attributes ..... 3
10. Maximum lenght of name of attribute ... 75
11. File of data ..... kapyka.D02
12. Chains (not allowed in this version) .. 0
13. First member of record type ..... siementieto
14. Second member of record type .....
15. Third member of record type .....

Number of line to be manipulated >9

```

Kuva 34. Kun tietue on kunnossa, poistutaan painamalla return-näppäintä.

```

TUTKA UTILITY 1/1          kapykanta

You can choose one of following actions:

Manipulate authorities ..... A
Manipulate data of dictionary ..... D
Manipulate files of classifications ..... C

Please enter your choice >a

```

Kuva 35. Käyttöoikeuksia ryhdytään manipuloimaan komennolla A.

```
TUTKA UTILITY 1/1 kapykanta
```

```
Add User/Password-combination ..... A  
Delete User/Password-combination ... D
```

```
Please enter your choice >a
```

Kuva 36. Käyttäjää voidaan joko lisätä (A) tai poistaa (D).

```
TUTKA UTILITY 1/1 kapykanta
```

```
Enter User/Password-combination >markku/mas
```

```
Choose degree of authorization associated to given user
```

```
Possible degrees are:
```

```
1 ..... User is authorized to delete  
2 ..... User is authorized to write  
3 ..... User is authorized to read  
4 ..... User is authorized to execute
```

```
Enter degree of authorization associated to given user >2
```

Kuva 37. Uudelle käyttäjälle annetaan käyttöoikeusaste.

4. TIETOKANNAN KAAVION JA TERMILUETTELOIDEN RAPORTOINTI (DRP)

4.1 Ohjelman tarkoitus

DRP-ohjelmalla suoritetaan tietokannan kaavion määrämuotoinen raportointi. Raportissa kuvataan tietokannan fyysinen rakenne sekä käytetyt luokitukset. Raportti on tulostettavissa RNO-tekstinkäsittelyohjelmalla.

4.2 Ohjelman käyttöohje

Ohjelma käynnistetään päähakemistossa komennolla DRP. Ohjelma kysyy syöttötietoina lyhyen otsikkorivin raportille (nimi, projekti tai vastaava) sekä tiivistelmän raportin sisällöstä. Otsikkorivin maksimipituus on 40 merkkiä.

Ohjelma hakee raportin tarvitsemat tiedot tietokannan sanakirjasta sekä termiluetteloista. Tiedot tulostetaan kahteen tiedostoon. Raportit valmistetaan siten, että ne voidaan tulostaa RNO-ohjelmalla. Fyysisen kuvauksen sisältävä tiedosto nimetään tietokannan nimellä. Käytetyt luokitukset sisältävä tiedosto nimetään aina TERMIT-etuosalla. Molempien tiedostojen tyyppi on .RNO.

4.3 Esimerkki

Tuotetaan KAPYKANTA-tietokannan fyysinen kuvaus. Raporttien tuottamiseen liittyvät näytöt on esitetty kuvissa 38 ja 39. Raportit on esitetty liitteessä B.

4.4 Virheilmoitukset

Ohjelman tulostamat virheilmoitukset ovat DML-ohjelmien antamia virheilmoituksia (ks. luku III kappale 4).

```

TUTKA UTILITY 1/1      kapykanta

Project or name >KAPYKANTA/TUTKA

You can now enter short abstract of report.
Maximum lenght is 10 lines. Type <cr> to quit.

Line 1 >KAPYKANTA is a little database. It is used to demonstrate
Line 2 >the structure of databases consturated by TUTKA.
Line 3 >

```

Kuva 38. Ennen tietokannan raportoimista kysytään otsikkotason tietoa: PROJEKTI, TIETOKANNAN LAATIJA tms.

```

TUTKA UTILITY 1/1      kapykanta

Physical description of given database is in file:
      KAPYKANTA.RNO

Classifications used in this database are reported
in file:
      TERMIT.RNO

To continue type <cr> >

```

Kuva 39. Tietokannan raportointi on suoritettu.

5. DATAN KERUU TIETOKANNASTA (DAP)

5.1 Ohjelman tarkoitus

DAP-ohjelma (Data Access Program) kerää käyttäjän asettamien ehtojen mukaista dataa tietokannasta nimettyyn tiedostoon. Se tuottaa myös tulostetun tiedoston tietueenkuvauslomakkeen.

5.2 Ohjelman käyttöohje

Ohjelma etenee lineaarisesti alusta loppuun: hierarkiatasoja ei ole. Suoritettavat toiminnot jakautuvat kolmeen vaiheeseen:

- 1) ehtojen määrittäminen
- 2) tulostus- ja tietueenkuvaustiedoston määrittäminen
- 3) tulostettavien muuttujien määrittäminen.

Näyttö tyhjenee kunkin vaiheen jälkeen. Kuhunkin vaiheeseen liittyvien syöttötietojen loppuminen ilmaistaan lähettämällä linjalle tyhjä rivi return-näppäimellä. Kun ohjelma on saanut tarvittavat määrittäystiedot, alkaa aineiston keruu. Kun aineisto on haettu, palataan päävalikkoon.

5.2.1 Ehtojen syöttö

Ensimmäiseksi ohjelmalle syötetään ehdot. Ohjelma kysyy

```
Restriction 1 >...
Restriction 2 >...
jne.
```

Ehtojen syöttö lopetetaan painamalla return-näppäintä.

Yksittäisen ehtolauseen yleinen muoto on

```
Restriction n >tunnus.vertailuoperaattori.vertailuarvo
```

Ohjelma numeroi ehdot juoksevasti (n).

Tunnus annetaan joko lyhenteenä tai koko nimenä. (Nimi ja lyhenne annetaan kullekin tunnukselle tietokannan luonnin yhteydessä).

Vertailuoperaattori voidaan antaa joko selväkielisenä tai lyhenteenä. Kun vertailuoperaattori annetaan selväkielisenä, erottimena toimii blanko, jolloin sanat muodostavat lauseen. Sallittuja selväkielisiä vertailuoperaattoreita ovat

- 1) ON
- 2) ON YHTÄSUURI KUIN
- 3) EI OLE
- 4) ON ERISUURI KUIN
- 5) ON PIENEMPI KUIN
- 6) ON SUUREMPI KUIN
- 7) ON PIENEMPI TAI YHTÄSUURI KUIN
- 8) ON SUUREMPI TAI YHTÄSUURI KUIN.

Kun vertailuoperaattori annetaan lyhenteenä, erottimena käytetään pistettä (.). Sallitut lyhenteet ovat (mm. FORTRAN-kielestä tutut) .EQ., .NE., .LT., .GT., .LE. ja .GE. Molemmissa tapauksissa voidaan käyttää joko pieniä tai suuria kirjaimia.

Vertailuarvo voidaan antaa joko kokonaislukuina tai, mikäli kyseessä on luokiteltava tunnus (esim. puulaji), myös selväkielisenä. Selväkielinen muuttuja annetaan siinä muodossa kuin se on kirjoitettu haluttua luokkaa vastaavaan termiin. Lisäksi, mikäli vertailuarvo annetaan kokonaislukuna, voidaan useita vertailuarvoja antaa yhtäaikaan pilkuilla toisistaan erotettuina. Tällöin on kyse loogisesta disjunktioista eli "TAI"-ehdosta. Ehto on mahdollinen vain silloin, kun vertailuoperaattori on "ON", "ON YHTÄSUURI KUIN" tai ".EQ.".

Erillisten ehtolauseiden välillä vallitsee looginen konjunktio eli "JA"-ehto.

Esimerkkejä ehtojen syöttämisestä:

```

Restriction 1 >tilan numero on 1
Restriction 2 >puustokuvion numero on pienempi kuin 5
Restriction 3 >puulaji on mänty
Restriction 4 >ikä on 50, 51, 52
...
```

5.2.2 Tulostustiedostojen määrääminen

Tulostustiedostot määrätään vastaamalla heti ehtojen syöttämisen jälkeen kahteen kysymykseen:

```

Output file (OUT.DAT) >
File of record description (RCD.DCR) >
```

Oletustiedostoina ovat datatiedostolle OUT.DAT ja tietueenkuvaukselle RCD.DCR. Oletukset jäävät voimaan, jos vastauksena annetaan tyhjä rivi.

5.2.3 Tulostettavat muuttujat

Tulostettavat muuttujat annetaan tiedostojen määräämisen jälkeen. Muuttujat voidaan antaa joko omilla nimillään tai lyhenteillä, jotka tietokannan kaavioon on talletettu.

```
Variable 1 >tilan numero  
Variable 2 >keskipituus  
...
```

Tulostettavien muuttujien syöttäminen lopetetaan return-näppäimellä.

5.3 Esimerkki

Kerätään aineistoa tietokannasta TILAKANTA, johon on talletettu puusto- ja kasvupaikkakuviokohtaista tietoa. Tietokannasta etsitään tietty tila ja siltä tietty puustokuvio, sekä tutkitaan ko. kuviolla kasvavia mäntyjä. Esimerkkiin liittyvät näytöt on esitetty kuvissa 40, 41 ja 42. Haun tuloksena saadut tiedostot on esitetty liitteessä A.

```
TUTKA UTILITY 1/1          tilakanta  
  
Restriction 1 >tilan numero on 1  
Restriction 2 >pkuvion numero on pienempi kuin 5  
Restriction 3 >laji on mänty  
Restriction 4 >
```

Kuva 40. Tietokannasta kerätään ainoastaan annetut ehdot täyttävä data.

```
TUTKA UTILITY 1/1      tilakanta
```

```
Output file (OUT.DAT) >  
File of record description (RCD.DCR) >
```

Kuva 41. Tulostustiedostojen määrääminen. Oletukset jäävät voimaan, jos kysymyksiin vastataan painamalla return-näppäintä.

```
TUTKA UTILITY 1/1      tilakanta
```

```
Variable 1 >tilan numero  
Variable 2 >pkuvion numero  
Variable 3 >laji  
Variable 4 >pohjapinta ala  
Variable 5 >keskipituus  
Variable 6 >
```

Kuva 42. Tulostettavien muuttujien määrääminen. Tulostiedoston tietue muodostuu eri tietuetyyppien em. ehdot täyttävien esiintymien tietokenttien arvoista.

5.4 Rajoitukset

Ehdot ja tulostettavat muuttujat on annettava samalta hierarkkiselta polulta. Hierarkkinen polku tarkoittaa yksikäsitteisesti määrättyä tietuetyyppiyhdistelmää, joista 1) jokainen on tietokannan eri tasolta ja jossa 2) jokainen tietuetyyppi muodostaa jonkin toisen yhdistelmään kuuluvan tietuetyypin kanssa setin. Tietuetyypistä voidaan luonnollisesti valita mitä tahansa attribuutteja ehto- tai tulostusmuuttujiksi. Muita rajoituksia ei ole. Käyttäjän on kuitenkin syytä olla looginen ehtoja ja tulostettavia muuttujia asettaessaan.

5.5 Virheilmoitukset

Ohjelman virheilmoitukset ovat pääosaltaan DML-ohjelmien virheilmoituksia. Muutamia yksilöllisempiä virheilmoituksia annetaan lähinnä ehtojen syntaksin tarkistuksesta sekä kaaviossa etenemisen loogisuudesta. DML-ohjelmien tuottamien virheilmoitusten selitykset on dokumentoitu erikseen (ks. luku III kappale 4). DAP-ohjelman virheilmoitukset erottuvat muista '*'-merkillä ja selitetään tässä.

*Illegal access path.

Tulostus- tai ehtomuuttujia on annettu eri poluilta. Muuttujia saa antaa vain yhdeltä hierarkkiselta polulta.

*Maximum number of restrictions given.
Last restriction accepted.

Ehtoja on annettu maksimimäärä. Viimeinen ehto hyväksytään.

*Maximum number of variables given.
Last variable accepted.

Tulostettavia muuttujia on annettu maksimimäärä. Viimeinen muuttuja hyväksytään.

*Maximum number of restrictions is 100.
Last restriction omitted.

Ehtoja on annettu yli maksimimäärän, viimeinen ehto hylätään.

*Point missing.

Ehtolauseesta puuttuu erotinmerkki (piste).

*Restriction not logical.

Annettu ehto ei ole looginen tai sen syntaksi on väärä.

*Second point incorrectly placed or missing.

Toinen erotinmerkki on väärässä paikassa tai puuttuu.

*Too many restrictions.

TAI-ehdossa on annettu liian paljon vertailuarvoja.

*Syntax error in using OR-operator.

Ehtolauseen TAI-osa on virheellinen.

*Unknown attribute.

Annettu muuttuja on tuntematon.

*Unknown operator.

Vertailuoperaattori on tuntematon.

6. TIETOKANNAN TIEDOSTOJEN DUMPPAUS (DUM)

6.1 Ohjelman tarkoitus

DUM-ohjelmalla (Dumping program) voidaan tulostaa tietokannan kaavio (sanakirja ja lyhenneluettelo), datatiedostot sekä termiluettelot fyysisessä järjestyksessä. Ohjelma helpottaa mahdollisten virhelähteiden paikallistamista sekä tietueitten osoitteiden hyväksikäyttöä sovellusohjelmissa.

6.2 Ohjelman käyttöohje

DUM-ohjelma käynnistetään päävalikossa komennolla DUM. Tällöin ohjelma tulostaa uuden sivun, josta valitaan dumpkauksen kohde: sanakirja, datatiedostot tai termiluettelot. Päähakemistoon päästään takaisin return-näppäintä painamalla.

6.2.1 Kaavion (sanakirjan ja lyhenneluettelon) dumpkaus

Sanakirja ja lyhenneluettelo voidaan dumpata valitsemalla dumpauskohde käskyllä "D" (dictionary). Ohjelma tulostaa uuden sivun kuvaputkelle. Valittavana on yksi seuraavista toiminnoista:

- sanakirjan dumpkaus (1 tietue näytölle) D
- sanakirjan lyhyt dumpkaus (15 tietuetta näytölle) S
- lyhenneluettelon dumpkaus (15 tietuetta näytölle) A.

Dumpauskohteen valintatilaan päästään return-näppäimellä.

6.2.2 Datatiedostojen dumpkaus

Datatiedostojen tulostus käynnistyy komennolla "F" (files). Ohjelma avaa uuden näytön, jossa kysytään tulostettava tietue-tyyppi sekä tulostustiedosto. Siten datatiedostojen tulostus voidaan suunnata myös tiedostoon. Tulostus saadaan päätteelle vastaamalla tiedostokyselyyn "TT:" tai return-näppäimellä.

Dumpauskohteen valintatilaan siirrytään takaisin return-näppäintä käyttäen.

6.2.3 Termiluetteloiden dumpkaus

Termiluetteloiden dumpaustilaan päästään vastaamalla valintatilan kysymykseen "C" (classifications). Dumpkaus toimii täysin analogisesti DMP-ohjelman termiluetteloiden manipulointitilan examine-toiminnon kanssa. Ohjelmalle annetaan siis dumpattavan attribuutin nimi. Ohjelma tulostaa kuvaputkelle 15 luokkaa tai attribuuttia kerrallaan. Dumpauskohteen valintatilaan päästään painamalla return-näppäintä vastaukseksi kysymykseen

Attribute >

Jos kysymykseen vastataan kirjaimella R (tai S) saadaan näyttöön luettelo tietokannan tietuetyypeistä (tai seteistä). Luettelo tietuetyyppiin kuuluvista attribuuteista saadaan tietuetyypin nimellä tai lyhenteellä.

6.3 Esimerkki

Ensimmäisessä esimerkissä dumpataan KAPYKANNAN tietuetyypin KAPVTIETO rakenteellinen tieto sanakirjasta. Toimenpiteeseen liittyvät näytöt on esitetty kuvissa 43-45. Toisessa esimerkissä dumpataan TILAKANNAN tietuetyypin TILAT datatiedosto. Näytöt on esitetty kuvissa 46-48.

```

TUTKA UTILITY 1/1          kapykanta

You can choose one of following actions:

Dump dictionary or file of abbreviations ..... D
Dump data files ..... F
Dump files of classifications ..... C

Please enter your choice >d

```

Kuva 43. DUM-ohjelmalla voidaan dumpata sanakirjaa, lyhenne-
luetteloja tai termiluetteloita.

```
TUTKA UTILITY 1/1      kapykanta
```

Choose one of following possibilities:

```
Dump dictionary ..... D
Dump dictionary shortly ..... S
Dump file of abbreviations ..... A
```

Please enter your choice >d

```
First record (all = <cr>) >5
Last record (all = <cr>) >6
```

Kuva 44. Sanakirja voidaan tulostaa tyypistetyssä tai laajemmassa muodossa.

```
TUTKA UTILITY 1/1      kapykanta
```

Record number: 5

```
Set ..... kapytieto-siementieto
Abbreviation ..... kasi
Type of relation ..... AP
Object ..... SETIT
Type ..... S
File of data ..... kapyka.D03
Number of sortkeys ..... 0
First sortkey .....
Second sortkey .....
Third sortkey .....
```

Continue? >

Kuva 45. Laajassa dumpissa tulostetaan yksi sanakirjan tietue näytölle.

```
TUTKA UTILITY 1/1      kapykanta

You can choose one of following actions:

Dump dictionary or file of abbreviations ..... D
Dump data files ..... F
Dump files of classifications ..... C

Please enter your choice >f
```

Kuva 46. Datatiedostoja voidaan dumpata komennolla F.

```
TUTKA UTILITY 1/1      tilakanta

Record type to be dumped >tilat
Output file (users terminal) >
First record (all = <cr>) >1
Last record (all = <cr>) >5
```

Kuva 47. Datatiedostojen dumpaus voidaan suunnata tiedoston. Oletuksena on kuitenkin pääte.

TUTKA UTILITY 1/1				tilakanta				
1: 900000004	2	5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
2: 1	1	-1	101	99999	1	69640	5501	
1050	144	10	333	0	0	11	0	
12	0	13	0	0	3	3	3	
1								
3: 2	1	-1	202	99999	1	69622	5510	
1050	144	10	333	0	0	21	0	
22	0	23	0	0	4	4	4	
44								
4: 3	1	-1	303	99999	1	69609	5515	
1050	144	10	333	0	0	31	0	
32	0	33	0	0	5	5	5	
69								
5: 4	1	-1	404	99999	1	69615	5525	
1050	144	10	333	0	0	41	0	
42	0	42	0	0	6	6	1	
109								

Continue? >

Kuva 48. Näytölle mahtuu 5 oletustietuetta kerralla. Vasemman laitaa tulostuvat numerot ovat tietueiden fyysisiä osoitteita. Osoitteessa 1: nähdään tunnustietue; esillä olevassa setissä on neljä tietue-esiintymää, ensimmäinen osoitteessa 2: ja viimeinen osoitteessa 5:. Tunnustietueen isäntälinkkikentän arvo 0 ilmaisee, että tietuetyyppi kuuluu tietokannan ylimmälle tasolle.

6.4 Rajoitukset

DUM-ohjelmaan ei sisälly rajoituksia.

6.5 Virheilmoitukset

DUM-ohjelman virheilmoitukset ovat DML-ohjelmien virheilmoituksia (ks. luku III kappale 4).

7. TIETOKANNAN TUHOAMINEN JA POISTAMINEN REKISTERISTÄ (DDD)

7.1 Ohjelman tarkoitus

DDD-ohjelmalla (Delete Data Description) voidaan tuhota tietokantaan liittyvät tiedostot sekä poistaa se rekisteristä.

7.2 Käyttöohje

Ohjelmaan päästään päähakemistosta komennolla DDD. Ohjelmalohkoon pääsy on sallittu vain niille käyttäjille, joilla on kaikki tietokannan käyttöoikeudet. Varmistukseksi on käyttäjän vielä annettava komento "DELETE". Tällöin DDD-prosessori käynnistyy. Ohjelma tuhoaa ensin tietokantaan liittyvät tiedostot ja poistaa sen jälkeen tietokannan rekisteristä. Tietokannan generoinnissa tarvittavaa .SCH-tyyppistä tiedostoa ei tuhota. Suorituksen loputtua apuohjelmisto on käynnistettävä uudelleen, mikäli sitä vielä halutaan käyttää.

Tuhottu tietokanta on palautettavissa ainoastaan varmistusnauhoilta, jos sellaisia on, sekä kääntymällä TUTKA-ohjelmiston ylläpitäjän puoleen tietokannan uudelleen rekisteröintiä varten.

7.3 Esimerkki

Esimerkissä tuhoataan tietokanta KAPYKANTA. Kuvissa 49 ja 50 on esitetty tuhoamiseen liittyvät näytöt sekä ohjelman tarvittavat komennot.

7.4 Rajoitukset

Ohjelman käyttö on sallittu vain käyttäjille, joilla on l-käyttöoikeus.

7.5 Virheilmoitukset

Ohjelma tuhoaa vain tuhottavissa olevat tiedostot: virhetilanteita ei synny tiedostojen osalta. Suojatut tiedostot jäävät tuhoamatta. Ohjelma käyttää TUTKA-tiedonhallintaohjelmiston DML-aliohjelmia, joihin liittyvien virheilmoitusten selitykset on esitetty luvun III kappaleessa 4.

```
TUTKA UTILITY 1/1 kapykanta
```

```
WARNING: You are now destroying your database  
Only file of schema will be saved.
```

```
To destroy your database type DELETE >delete
```

Kuva 49. Tietokannan tuhoaminen.

```
TUTKA UTILITY 1/1 kapykanta
```

```
Now deleting database, please wait.
```

```
File "kapykanta.nim" deleted  
File "kapykanta.lyh" deleted  
File "kapykanta.dev" deleted  
File "kapykanta.lup" deleted  
File "kapykanta.ava" deleted  
File "kapykanta.kys" deleted  
File "kapykanta.log" deleted  
File "kapyka.ryh" deleted  
File "kapyka.set" deleted  
File "kapyka.001" deleted  
File "kapyka.r01" deleted  
File "kapyka.r02" deleted  
File "kapyka.r03" deleted
```

Kuva 50. Ohjelma luetteloit tuhotut tiedostot kuvaputkelle.

III DATAN KÄSITTELYOHJELMAT (TUTKA DML)

1. DATAN KÄSITTELYOHJELMISTO

Datan käsittelyohjelmisto (Data Manipulation Programs) tai yleisesti datan käsittelykieli (Data Manipulation Language) mahdollistaa kaavion määritysten mukaisen datan järjestelyn ja saannin sovellusohjelmissa. Yksinkertaisuuden vuoksi datan käsittely (data manipulation) lyhennetään jatkossa DML:ksi.

DML-ohjelmat muodostavat tehokkaan välineen, jolla ohjelmoiva käyttäjä voi käsitellä tietokantaansa. DML-ohjelmien käyttö on loogista. Niiden kautta välittyy vahva semanttinen ote käsiteltävään tietoon. Ohjelmien rakentaja ja käyttäjä saa täyden hyödyn tietokannan rakenteesta ja nimistöstä.

1.1 Ohjelmien nimeämisperiaatteet

TUTKA tiedonhallintajärjestelmässä on DML-ohjelmien nimeämisessä noudatettu seuraavia periaatteita:

1) Kaikki DML-ohjelmanimet alkavat kirjainparilla DB (Data Base), mikä helpottaa DML-ohjelmien erottamista muusta ohjelmakoodista.

2) Koska DML-ohjelmien isäntäkielenä on "englannin kaltainen" FORTRAN, on ohjelmanimet muodostettu ohjelman toimintaa kuvaavan englanninkielisen lauseen lyhenteestä. Kirjainparin DB lisäksi on käytetty 2 - 4 kirjainta ilmaisemaan ohjelman toimintaa. Käytännössä on osoittautunut hyödylliseksi muistaa kunkin ohjelman englanninkielinen selite, sillä se helpottaa sekä lähdekielisten ohjelmien ymmärtämistä että ohjelmointityötä.

1.2 Ohjelmien käyttöönotto

DML-ohjelmat liitetään käyttäjän ohjelmaan komennolla

```
$LINK ...DML/OPTIONS.
```

1.3 Ohjelmien ryhmittely

DML-ohjelmat ryhmitellään käyttötarkoituksen ja toiminnan perusteella seitsemään ryhmään:

1) Yleisohjelmat

Ohjelmat mahdollistavat tietokannan käsittelyn muilla DML-ohjelmilla. Ne huolehtivat erilaisten käyttötaulukoiden muodostamisesta ja ylläpidosta ohjelman suorituksen aikana. Ohjelmat myös lukevat ja tallettavat kaavioon tai aputiedostoihin talletetun tietokannan kulloisenkin tilan, statuksen. Yleisohjelmia ovat

2) Tietokannan tietosisältöä muuttavat ohjelmat

Tietokannan tietosisältöön voidaan vaikuttaa lisäämällä tai poistamalla tietueita tai muuttamalla yksittäisten tietokenttien arvoa. Toiminnot toteutetaan seuraavilla ohjelmilla:

- DBCR - tietuetyypin tietue-esiintymän luonti
- DBCRS - tietuetyypin tietue-esiintymän luonti ja datan talletus
- DBSFR - tietokentän arvon asetus
- DBDR - tietuetyypin tietue-esiintymän tuhoaminen
- DBMDT - termiluettelon editointi.

3) Tiedonsaantiohjelmat

Tiedonsaantiohjelmilla voidaan lukea tietokannan tietosisältöä. Talletettua dataa luetaan datatiedostoista sekä tietokannan tietosisältöä ja rakennetta kuvaavaa tietoa kaaviosta. Datan siirtoon käytetään lukuvektoria, joka on mitoitettava riittävän pitkäksi. Merkkitiedon siirrossa käytetään byte-vektoria. Tiedonsaantiohjelmia ovat

- DBGDR - tietueen tietosisällön luku
- DBGDF - tietokentän arvon luku
- DBGDT - tietojen luku termiluettelosta
- DBGIR - tietuetyyppiä koskevan tiedon luku sanakirjasta
- DBGII - tietokenttää koskevan tiedon luku sanakirjasta
- DBGIS - settiä koskevan tiedon luku sanakirjasta
- DBGNR - tietuetyyppien nimien luku R-termiluettelosta
- DBGNI - tietuetyypin tietokenttien nimien luku termiluettelosta
- DBGAN - kaikkien avoinnaolevien tietuetyyppien kaikkien nimien luku
- DBGKR - käsittelyosoitteen luku muuttujaan
- DBLOOK - tietuetyypin tietue-esiintymän tulostus päätteelle
- DBFIGU - tietokannan rakenteen kuvaaminen.

4) Hakuohjelmat

Hakuohjelmilla mahdollistetaan horisontaalinen siirtyminen tietokannan tietue-esiintymistä muodostetuissa listoissa. Siirtymiseen vaikuttaa kulloinenkin hakutilanne eli se missä kohden tietokantaa ollaan. Hakutilanne ennen ja jälkeen siirtymisen ilmaistään tietuetyypin tietue-esiintymien ryhmästä tai samaan settiin kuuluvien tietueiden muodostamasta joukosta esille otetuin tietue-esiintymien. Hakuohjelmia ovat

- DBFNR - loogisessa järjestyksessä seuraavan tietue-esiintymän haku
- DBFFM - setin jäsentietuetyypin ensimmäisen tietue-esiintymän haku
- DEFNM - setin jäsentietuetyypin loogisessa järjestyksessä seuraavan tietue-esiintymän haku.

5) Tietue-esiintymien käsittelyosoitteiden (currency indicators) asetusohjelmat

Ohjelmat mahdollistavat vertikaaliset siirtymiset tietokannan rakenteessa. Tietokannan kullekin tietuetyypille, kunkin setin isännälle ja jäsenelle on varattu yksi muistipaikka halutun käsittelyosoitteen tallettamiseksi. Käsittelyosoitetta voidaan muuttaa joko hakuohjelmilla tai käsittelyosoitteen päivitysohjelmilla. Käsittelyosoitteen osoittaman tietue-esiintymän sanotaan "olevan esillä" (current) tietuetyypin tietue-esiintymien ryhmästä tai setin jäsenesiintymien muodostamasta joukosta. Seuraavat ohjelmat vaihtavat esilläolevan tietue-, isäntä- tai jäsenesiintymän lähtötilanteen perusteella toiseksi:

- DBSRK - tietue-esiintymän osoittaminen käsittelyosoitteen perusteella
- DBSRO - tietue-esiintymän osoittaminen isäntätietuetyypin esilläolevan tietue-esiintymän perusteella
- DBSRM - tietue-esiintymän osoittaminen jäsentietuetyypin esilläolevan tietue-esiintymän perusteella
- DBSOR - isäntätietuetyypin esiintymän osoittaminen tietuetyypin esilläolevan tietue-esiintymän perusteella
- DBSMR - jäsentietuetyypin esiintymän osoittaminen tietuetyypin esilläolevan tietue-esiintymän perusteella.

6) Loogiset testifunktiot

Loogiset testifunktiot mahdollistavat tietokannan tietosäällön suoran vertailun ulkoisiin arvoihin. Niitä ovat

- DBLT - tietokentän arvo pienempi kuin vertailuarvo
- DBLE - tietokentän arvo pienempi tai yhtä suuri kuin vertailuarvo
- DBEQ - tietokentän arvo yhtä suuri kuin vertailuarvo
- DBNE - tietokentän arvo eri suuri kuin vertailuarvo
- DBGE - tietokentän arvo suurempi tai yhtä suuri kuin vertailuarvo
- DBGT - tietokentän arvo suurempi kuin vertailuarvo
- DBTEST - tietokentän arvon vertailu vertailuarvoon.

7) Apuohjelmat

Apuohjelmien tarkoitus on helpottaa ohjelmointityötä. Toistaiseksi valmiina on ainoastaan

- DBCHEK - Esilläolevien esiintymien käsittelyosoitteiden tulostus.

1.4 Ohjelmakuvaukset

Seuraavilla sivuilla on kuvattu käytössä olevat DML-ohjelmat. Kaikki ohjelmat, eräitä loogisia testifunktioita lukuunottamatta, on kuvattu taulukkomuodossa seuraavasti:

- 1) Ohjelma: Ilmaisee ohjelman nimen.
- 2) Selite: Kuvaa nimen muodostumisen loogikkaa.
- 3) Toiminta: Kuvaa ohjelman toimintaa.
- 4) Kutsu: Ohjelman kutsu FORTRAN-ohjelmissa.
- 5) Parametrit: Kuvaa kutsussa välitettävät parametrit.
- 6) Eo.status: Kuvaa tietuetyypin, isäntätyypin tai jäsenätyypin käsittelyosoitteen muuttamisen ohjelman vaikutuksesta.
- 7) Paluukoodit: Selittää parametrin ipk paluuarvot.
- 8) Huom! : Ohjelman toimintaa koskevat mahdolliset varoitukset ja huomautukset.

Edellä on esilläolevien tietue-, isäntä- ja jäsenesiintymien käsittelyosoitteiden muuttumisdynamiikka kuvattu kuusipaikkaisella tilataulukolla (eo. status). Taulukon I sarakkeella ilmaistaan ohjelman suorituksen aikana merkitsevät käsittelyosoitteet, 0 sarakkeella käsittelyosoitteet, jotka ohjelman suorituksen aikana ovat muuttuneet. Tietue, isäntä ja jäsen viittaavat ohjelman kutsussa olevaan tietuetyyppiin tai settiin.

Esim. 1: Ohjelmassa DBFNR on taulukko

	I	0
tietue	X	X
isäntä		
jäsen		

Tulkinta: DBFNR-ohjelma hakee ennen ohjelman suoritusta (I), esilläolevaksi tietue-esiintymäksi osoitetun tietueen loogisen seuraajan ja tekee siitä uuden (0) esilläolevan tietue-esiintymän. Siis, ohjelma päivittää nimetyn tietuetyypin käsittelyosoitteen arvoa.

Esim. 2: Ohjelmassa DBSRO on taulukko

	I	0
tietue		X
isäntä	X	
jäsen		

Tulkinta: Ohjelma tekee kutsussa mainitun settityyppin esilläolevasta isäntäesiintymästä isännän tietuetyyppin esilläolevan tietue-esiintymän. Siis, isäntä (I):n arvo kopioidaan kohtaan tietue (0).

Käsittelyosoitteiden käytön opettelu edellyttää pientä vaivanäköä. Koska tietuetyyppiä ja attribuuttia tarkastellaan useassa eri kohdassa hieman toisistaan poikkeavissa näkökulmissa, selvyuden vuoksi on otettu käyttöön niitä kuvaavia synonyymeja. Seuraavat käsitteet voidaan rinnastaa jatkossa:

tietuetyyppi ja kohde

sekä

attribuutti, määre, tunnus ja muuttuja.

1.4.1 Yleisohjelmat

Ohjelma: DBINIT

Selite: Initialize arrays and files for data base manipulation.

Toiminta: Ohjelma alustaa tietokannan käsittelyssä tarvittavat taulukot ja oletuslaitteet sekä lataa osan tietokannan kaaviosta keskusmuistiin. Ohjelma tarkistaa myös sanakirjan.

Kutsu: CALL DBINIT(tkanta,tunnus,ipk)

Parametrit: tkanta = Tietokannan nimi byte-vektorissa (9) tai literaalina.
 tunnus = Käyttäjätunnus ja salasana ioko byte-vektorissa (19) tai literaalina. Parametrissa voidaan antaa myös vakio-teksti: " ' ", jolloin ohjelma kysyy käyttäjältä käyttäjätunnuksen ja salasanan.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 0 = Joko tietokantaa ei ole luotu, tietokannan tietuetyyppiluettelo on virheellinen, käyttäjällä ei ole käyttöoikeutta tietokantaan tai sanakiriasta puuttuvat R- tai S-määrittelyt.
 -1 = Tietuetyyppiluettelossa on tietuetyyppi, jota ei ole määritelty sanakirjaan.
 -16 = Tietuetyypin attribuuttiluettelossa olevaa attribuuttia ei ole määritelty sanakirjaan.
 -17 = Tietuetyypin attribuuttiluettelon käsittely epäonnistui.
 -21 = Sanakirjan luku epäonnistui.
 -22 = Attribuuttiluettelon luku epäonnistui.

Huom! : Ohjelma on suoritettava aina ensimmäisenä DML-aliohjelmana sovellutusohjelmassa.

Ohjelma: DBOPEN

Selite: Open record type for data manipulation (DML-) program.

Toiminta: Ohjelma suorittaa DML-ohjelmassa tietuetyypin laitevaraukset ja alustaa käyttötaulukkoita.

Kutsu: CALL DBOPEN(kohde,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
0 = Avaus on suoritettu jo aiemmin ohjelmassa.
-1 = Tietuetyyppejä ei ole määritelty sanakirjaan.
-4 = Tietuetyyppejä ei ole tietokannan tietuetyypiluettelossa.
-13 = Tietuetyypin laitemäärittely ei onnistunut.
-99 = Joko indeksitiedoston avaus epäonnistui tai tietuetyypin talletustapa on outo tai formatoitu suorasaanti ei ole mahdollista.

Huom!: Ohjelma on suoritettava ennen käsittelyohjelmia. Tietuetyypin käsittelyosoite on ohjelman suorituksen jälkeen 1, ts. se osoittaa fyysisesti ensimmäiseen tietueeseen tietuetyypille varatussa tiedostossa. Joissakin tilanteissa tietoa voidaan käyttää hyväksi. Kuitenkin on muistettava, että fyysinen ja looginen järjestys eivät ole samat. Lisäksi, osoitteessa 1 on normaalitilanteessa aina tunnustietue.

Ohjelma: DBCLOS

Selite: Close record type(s) of data base and store status.

Toiminta: Ohjelma sulkee DML-ohjelmassa avatun tietuetyypin annetun parametrin perusteella. Mikäli parametrina annetaan vakioteksti 'ALL', ohjelma sulkee kaikki avatut tietuetyypit.

Kutsu: CALL DBCLOS(kohde,ipk)

Parametrit: kohde = Tietuetyypin nimi, lyhenne tai vakioteksti 'ALL' byte-vektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue
isäntä
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
-4 = Tietuetyyppejä ei ole avattu.
-5 = Tietuetyyppejä ei ole tietokannan tietuetyypiluettelossa.
-14 = Tietuetyypin laitemäärittäminen ei onnistu.
-21 = Sanakirjan luku ei onnistu.
-29 = Tietokannan statuksen talletus epäonnistui.

Huom!: Ohjelma on suoritettava kaikkien tietokannan tietosisältöä muuttavien ohjelmien jälkeen. DBCLOS-ohjelman käyttö vapauttaa I/O-kanavia.

1.4.2 Tietosisältöä muuttavat ohjelmat

Ohjelma: DBCR

Selite: Create record occurrence of record type.

Toiminta: Ohjelma muodostaa tietokantaan parametrissa annetun tietuetyypin tietue-esiintymän.

Kutsu: CALL DBCR(kohde,kos,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 kos = Muodostetun tietuetyypin käsittelyosoite.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä X
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppejä ei ole määritelty sanakirjaan.
 -3 = Tietuetyypin ja sen isännän muodostamaa settiä ei ole määritelty sanakirjaan.
 -4 = Tietuetyyppejä ei ole avattu.
 -5 = Tietuetyyppejä ei ole tietokannan tietuetyypiluettelossa.
 -8 = Tietuetyypin jäsentä ei ole tietokannan tietuetyypiluettelossa.
 -9 = Tietuetyypin isäntää ei ole avattu tai sitä ei ole tietokannan tietuetyypiluettelossa.
 -10 = Tietuetyypin jäsentä ei ole avattu.
 -11 = Jäsentietuetyyppejä ei ole nimetty kohteen tietuetyypin määrittelyssä sanakirjassa.
 -21 = Sanakirjan luku epäonnistui.
 -23 = Tietuetyypin esiintymän kirjoittaminen tiedostoon epäonnistui.
 -26 = Tietuetyypin tai sen jäsenen esiintymän luku tiedostosta epäonnistui.
 -27 = Isäntätietuetyypin esiintymän luku tiedostosta epäonnistui.
 -32 = Esilläoleva isäntätietuetyypin esiintymä on väärä tai tietueiden linkityksessä on tapahtunut virhe.
 -47 = Käyttäjällä ei ole kirjoitusoikeutta tietokantaan.

Huom!: Tyhjän tietueen lisäys tehdään aina listan loppuun. Ohjelman yhteydessä on aina suoritettava DBCLOS-ohjelma.

Ohjelma: DBCRS

Selite: Create record occurrence of record type and store data.

Toiminta: Ohjelma muodostaa tietokantaan parametrissa annetun tietuetyypin tietue-esiintymän ja tallettaa parametrissa annetun data-vektorin tietosisällön siihen.

Kutsu: CALL DBCRS(kohde,data,kos,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 data = Reaalilukuvektori.
 kos = Luodun tietueen käsittelyosoite.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä X
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppejä ei ole määritetty sanakirjaan.
 -3 = Tietuetyypin ja sen isännän muodostamaa settiä ei ole määritetty sanakirjaan.
 -4 = Tietuetyyppejä ei ole avattu.
 -5 = Tietuetyyppejä ei ole tietokannan tietuetyyppiluettelossa.
 -9 = Tietuetyypin isäntää ei ole avattu.
 -10 = Tietuetyypin jäsentä ei ole avattu.
 -11 = Jäsentietuetyyppejä ei ole nimetty kohteen tietuetyypin määrittelyssä sanakirjassa.
 -12 = Setin, jossa tietuetyyppi on jäsenenä, lajitteluavainta ei ole määritetty sanakirjaan.
 -21 = Sanakirjan luku ei onnistu.
 -23 = Tietuetyypin esiintymän kirjoittaminen tiedostoon epäonnistui.
 -26 = Tietuetyypin esiintymän luku tiedostosta epäonnistui.
 -27 = Isäntätietuetyypin esiintymän luku tiedostosta epäonnistui.
 -32 = Esilläoleva isäntätietuetyypin esiintymä on väärä
 -34 = Esilläoleva isäntätietuetyypin esiintymä ei sisällä dataa tai on aiemmin tuhottu esiintymä.
 -47 = Käyttäjällä ei ole kirjoitusoikeutta tietokantaan.

Huom! : Data-vektorin on oltava riittävän pitkä, jotta kaikki tietuetyyppeihin määritellyt kentät tulevat mukaan. Ohjelma suorittaa automaattisesti lajittelun. Ohjelman yhteydessä on aina suoritettava DBCLOS-ohjelma.

Ohjelma: DBMDT

Selite: Manipulate data of file of classification.

Toiminta: Ohjelmalla editoidaan termiluetteloiden tietoja. Ohjelmalle ilmoitetaan kohteen hakutapa ja toimenpide, joka kohteelle suoritetaan. Hakutapa voi olla: 1) peräkkäishaku, 2) suora-haku tai 3) puolitus-haku. Toimenpide voi olla: 1) paikannetun termin kohdalle lisätään uusi indeksi ja uusi termi, 2) paikannettu termi ja indeksi tuhoetaan ja termiluettelo pakataan, 3) paikannettu indeksi muutetaan, 4) paikannettu termi muutetaan, 5) indeksi ja termi muutetaan tai 6) termi lisätään termiluettelon loppuun ja indeksoidaan kasvattamalla suurinta termiluettelon indeksia 1:llä.

Kutsu: CALL DBMDT(maare,termi,utermi,ind,uind,laj,laj2,ipk)

Parametrit: maare = Attribuutin nimi byte-vektorissa (30) tai literaalina.
 termi = Termi byte-vektorissa (80).
 ind = Etsittävä indeksi tai suora osoite.
 uind = Lisättävän termin indeksi tai indeksi, joksi paikannettu indeksi muutetaan.
 utermi = Lisättävä termi byte-vektorissa (80) tai termi, joksi paikannettu termi muutetaan.
 laj = Hakutapa 1),2) tai 3).
 laj2 = Toimenpide 1),2),3),4),5) tai 6).
 ipk = Paluukoodi.

Eo. status: I 0
 tietue
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 2 = Haluttua termiä ei löytynyt. Mikäli hakutapa oli puolitus-haku, lisäys suoritettiin lajitellun järjestyksen osoittamaan paikkaan.
 0 = Annettua indeksia vastaavaa termiä ei löytynyt termiluettelosta.
 -2 = Attribuuttia ei ole määritelty sanakirjaan.
 -19 = Annettua termiä ei löydy.
 -21 = Sanakirjan luku ei onnistu.
 -22 = Termiluettelon luku ei onnistu.
 -47 = Käyttäjällä ei ole tarvittavaa käyttöoikeutta.

Ohjelma: DBSFR

Selite: Set value in data field based on current record.

Toiminta: Ohjelma asettaa parametrissa annetun arvon parametrissa nimettyyn tietokenttään.

Kutsu: CALL DBSFR(maare,iarvo,ipk)

Parametrit: maare = Attribuutin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 arvo = Muuttujan arvo, joka sijoitetaan määrään tietokenttään.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Attribuutin tietuetyyppiä ei ole määritetty sanakirjaan.
 -2 = Attribuuttia ei ole määritetty sanakirjaan.
 -3 = Settiä, jossa attribuutin tietuetyyppi on jäsenenä, ei ole määritetty sanakirjaan.
 -4 = Attribuutin tietuetyyppiä ei ole avattu DML-ohjelmassa.
 -11 = Attribuutin tietuetyyppiä ei ole nimetty sen ja sen isännän muodostaman setin määritelmässä sanakirjassa.
 -12 = Attribuutin tietuetyypin lajitteluavainta ei ole määritetty sanakirjaan.
 -21 = Sanakirjan luku ei onnistu.
 -23 = Attribuutin tietuetyypin esiintymän kirjoittaminen tiedostoon ei onnistu.
 -26 = Attribuutin tietuetyypin esiintymän luku tiedostosta ei onnistu.
 -27 = Setin, jossa attribuutin tietuetyyppi on jäsenenä, isännän tietue-esiintymän luku sanakirjasta ei onnistu.
 -31 = Esillä olevan attribuutin tietuetyypin esiintymä ei sisällä dataa.
 -43 = Tietokenttään sijoitettavan attribuutin arvo ei ole sallituissa rajoissa.
 -47 = Käyttäjällä ei ole kirjoitusoikeutta tietokantaan.

Huom!: Ohjelma muuttaa tietueiden loogista järjestystä tietokannassa, mikäli sijoitettava tietokenttä on lajitteluavain.

Ohjelma: DBDRR
 Selite: Delete record occurrence based on current record.

Toiminta: Ohjelma tuhoaa tietuetyypin tietue-esiintymän tietokannasta sen esilläolevan tietue-esiintymän perusteella.

Kutsu: CALL DBDRR(kohde,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne byte-vektorissa (30) tai literaalina.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä X
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppejä ei ole määritelty sanakirjaan.
 -9 = Tietuetyypin isäntää ei ole avattu DML-ohjelmassa.
 -3 = Tietuetyypin ja sen isännän muodostamaa settiä ei ole määritelty sanakirjaan.
 -11 = Tietuetyyppejä ei ole määritelty isäntänsä jäseneksi sanakirjassa.
 -21 = Sanakirjan luku ei onnistu.
 -23 = Tietuetyypin tiedoston päivittäminen ei onnistu.
 -26 = Tietuetyypin tiedoston luku ei onnistu.
 -27 = Isäntätietuetyypin esiintymän luku tiedostosta ei onnistu.
 -46 = Käyttäjällä ei ole tuhoamisoikeutta tietokantaan.
 -99 = Tietokannan rakenteesta on löytenyt virhe.

Huom! : Ylemmän tason tietuetyypin esiintymän tuhoaminen aiheuttaa myös alemman tason tietuetyyppien tietue-esiintymien tuhoutumisen.
 Tuhotut tietue-esiintymät menetetään lopullisesti tietokannan pakkaamisen yhteydessä.
 Mikäli ohjelma antaa paluukoodin arvon -99, on syytä ottaa tietokannan aikaisempi, kunnossa oleva versio käyttöön.

1.4.3 Tiedonsaantiohjelmat

Ohjelma: DBCDR

Selite: Get data of record occurrence based on current record.

Toiminta: Ohjelma lukee tietuetyypin esilläolevan tietue-
esiintymän data-vektoriin.

Kutsu: CALL DBCDR(kohde,data,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne byte-
vektorissa (30) tai literaalina.
data = Lukuvektori tietokannasta luettavaa
dataa varten.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä
iäsen

Paluukoodit: 1 = Onnistunut suoritus.
-1 = Tietuetyyppeä ei ole määriteltä sanakir-
iaan.
-4 = Tietuetyyppeä ei ole avattu DML-ohjelmassa.
-26 = Tietuetyypin esiintymän luku tiedostosta
epäonnistui.
-31 = Esille otettu tietue-esiintymä on väärä.
-48 = Käyttäjällä ei ole lukuoikeutta tietokan-
taan.

Huom! : Datavektorin käsittelyn yhteydessä on mielekäästä
käyttää hyväksi DBGII ohjelman antamia indeksejä.
Data-vektorin on oltava riittävän pitkä, liika
pituus ei haittaa.

Ohjelma: DBGDF

Selite: Get data of data field based on current record.

Toiminta: Ohjelma lukee paluumuuttujan arvoksi parametrissa annetun attribuutin tietokentän tietosisällön esilläolevasta tietueesta.

Kutsu: CALL DBGDF(maare,iarvo,ipk)

Parametrit: maare = Tietokentän nimi tai lyhenne bytevektorissa (30) tai literaalina.
 arvo = Muuttuja tietokentän paluuarvoa varten.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -2 = Attribuuttia ei ole määritelty sanakirjaan.
 -13 = Attribuutin tietuetyypin laitteen määrittäminen ei onnistu.
 -21 = Sanakirjan luku ei onnistu.
 -26 = Attribuutin tietuetyypin esiintymän luku ei onnistu.
 -48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

- Ohjelma: DBGDT
- Selite: Get data of file of classification.
- Toiminta: Ohjelma poimii tietokannan termiluettelon termin ja indeksin paluumuuttujiin. Termiä voidaan hakea annetulta sarakkeelta lähtien.
Hakutapa voi olla: 1) suora haku indeksin osoitteesta, 2) peräkkäishaku indeksin perusteella alkaen tietueesta IAS, 3) peräkkäishaku termin perusteella (haun alkuosoite parametrissa IND, vertailun alkusarake parametrissa IAS), 4) puolitus haku indeksin perusteella tai 5) puolitus haku termin perusteella (vertailun alkusarake parametrissa IND).
- Kutsu: CALL DBGDT(maare,termi,ind,laj,ias,ipk)
- Parametrit: maare = Attribuutin nimi tai lyhenne bytevektorissa (30) tai literaalina.
termi = Byte-vektori (80) palautettavalle termille tai haettava termi. Vertailukentän pituus on sama kuin termin merkkien määrä.
ind = Kokonaislukumuuttuja palautettavaa indeksia varten tai haettava indeksi.
laj = Kokonaislukumuuttuja tai -vakio, jossa haluttu toiminto 1, 2, 3, 4 tai 5.
ias = Vertailun alkusarake termin perusteella etsittäessä (kts. toimenpiteet).
ipk = Paluukoodi.
- Eo.status: I 0
tietue
isäntä
jäsen
- Paluukoodit: 1 = Onnistunut suoritus.
2 = Haluttua termiä ei löytynyt. Mikäli käytettiin puolitus haku, osoittaa indeksi termin sijaintikohtaa lajitellussa järjestyksessä.
0 = Etsittävää indeksia ei löytynyt.
-2 = Attribuuttia ei ole määritelty sanakirjaan.
-19 = Etsittävää termiä ei löytynyt.
-21 = Sanakirjan luku ei onnistu.
-22 = Termiluettelon luku ei onnistu.
-48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.
- Huom! : Ohjelmalla voidaan käsitellä luokittelumuuttujia, attribuuttiluetteloita ja tietokannan R- ja S-termiluetteloita sekä merkkitietomuuttujia.

Ohjelma: DBGIR

Selite: Get information of record type.

Toiminta: Ohjelma lukee sanakirjasta parametrissa annettuun tietuetyyppiin liittyvää rakenteellista tietoa paluumuuttujiin.

Kutsu: CALL DBGIR(kohde,lyh,isanta,set,jlkm,llkm,data,ipk)

Parametrit:

- kohde = Tietuetyypin nimi tai lyhenne joko byte-vektorissa (30) tai literaalina.
- lyh = Byte-vektori (4) palautettavaa lyhenettä varten.
- isanta = Byte-vektori (30) palautettavaa isäntätietuetyypin nimeä varten.
- set = Byte-vektori (30) palautettavaa setin nimeä varten.
- jlkm = Kokonaislukumuuttuja tietuetyypin jäsenten lukumäärää varten.
- llkm = Kokonaislukumuuttuja tietuetyypin lajitteluavainten lukumäärää varten.
- data = Byte-vektori (<200) jäsenten nimiä ja lajitteluavaimia varten.
Nimet on erotettu toisistaan kauttavii-voilla, viimeisenä kaksi kauttavii-voilla.
- ipk = Paluukoodi.

Eo.status: I 0

tietue
isäntä
jäsen

Paluukoodit:

- 1 = Onnistunut suoritus.
- 2 = Tietuetyypin taso on 1.
- 1 = Tietuetyyppeä ei ole määritelty sanakirjaan.
- 3 = Settiä, jossa tietuetyyppi on jäsenenä, ei ole määritelty sanakirjaan.
- 5 = Tietuetyyppeä ei ole tietokannan tietuetyyppiluettelossa.
- 7 = Tietuetyypin isäntää ei ole avattu DML-ohjelmassa.
- 19 = Kyseiselle attribuutille ei ole määritelty tietuetyyppeä sanakirjaan.
- 21 = Sanakirjan luku ei onnistu.
- 48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

Huom! : Tietuetyyppi on jäsenenä set-vektorissa palautetussa setin nimessä.

Ohjelma: DBGII

Selite: Get information of data item.

Toiminta: Ohjelma lukee sanakirjasta attribuutin arvoa ja tietokenttää koskevaa tietoa paluumuuttujiin.

Kutsu: CALL DBGII(maare,lyh,ind,isar,mi,ma,ipt,kohde, ita,laatu,fo,ipk)

Parametrit:

- maare = Attribuutin nimi tai lyhenne byte-vektorissa (30) tai literaalina.
- lyh = Byte-vektori (4) palautettavaa lyhennettä varten.
- ind = Kokonaislukumuuttuja tietokentän indeksia varten.
- isar = Kokonaislukumuuttuja tietokentän alkusaraketta varten.
- mi = Kokonaislukumuuttuja attribuutin minimiarvoa varten.
- ma = Kokonaislukumuuttuja attribuutin maksimiarvoa varten.
- ipt = Kokonaislukumuuttuja tietokentän pituutta varten.
- kohde = Byte-vektori (30) attribuutin tietue-tyyppiä varten.
- ita = Attribuutin tietue-tyypin taso tietokannassa.
- laatu = Byte-vektori (6) attribuutin laadulle.
- fo = Byte-vektori (4) tietokentän formaatille.
- ipk = Paluumuuttuja.

Eo.status: I 0

tietue
isäntä
jäsen

Paluukoodit:

- 1 = Onnistunut suoritus.
- 2 = Attribuuttia ei ole määritelty sanakirjaan.
- 21 = Sanakirjan luku ei onnistu.
- 48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

Huom!:

Ohjelmaa kannattaa käyttää yksilöitäessä DBGDR-ohjelmalla poimitun data-vektorin kenttiä attribuutteja vastaaviksi.

Ohjelma: DBGIS

Selite: Get information of set.

Toiminta: Ohjelma lukee sanakirjasta settiä koskevaa tietoa.

Kutsu: CALL DBGIS(set,lyh,isanta,jasen,jlkm,llkm,data,ipk)

Parametrit:

- set = Setin nimi tai lyhenne byte-vektorissa (30) tai literaalina.
- lyh = Byte-vektori (4) setin lyhenteelle.
- isanta = Byte-vektori (30) setin isännälle.
- jasen = Byte-vektori (30) setin jäsenelle.
- jlkm = Kokonaislukumuuttuja jäsentietuetyyp-
pien esiintymien lukumäärää varten.
- llkm = Kokonaislukumuuttuja jäsentietuetyypin
lajitteluavainten lukumäärää varten.
- data = Lajitteluavaimet byte-vektorissa (100).
Lajitteluavaimet on erotettu toisistaan
kauttaviivoilla, viimeisenä on kaksi
kauttaviivaa.
- ipk = Paluukoodi.

Eo.status: I 0

tietue
isäntä
jäsen

Paluukoodit:

- 1 = Onnistunut suoritus.
- 3 = Settiä ei ole määritelty sanakirjaan.
- 11 = Isäntätietuetyypin määritelmässä sanakir-
jassa ei ole nimetty jäsentietuetyyppeä.
- 21 = Sanakirjan luku ei onnistu.
- 27 = Isäntätietuetyypin esiintymän luku ei
onnistu.
- 28 = Jäsentietuetyypin esiintymän luku ei
onnistu.
- 48 = Käyttäjällä ei ole lukuoikeutta tieto-
kantaan.

Huom! : Sekä isäntä- että jäsentietuetyyppi on avattava
DBOPEN ohjelmalla.

Ohjelma: DBGNR

Selite: Get names of record type(s).

Toiminta: Ohjelma täyttää datavektorin DML-ohjelmassa avattujen tietuetyyppien nimillä.

Kutsu: CALL DBGNR(data,lkm,ipk)

Parametrit: data = Byte-vektori (<312) tietuetyyppien nimille. Nimet erotetaan toisistaan kauttaviivoilla, viimeisenä kaksi kauttaviivaa.
lkm = Vektorissa olevien nimien lukumäärä.
ipk = Paluukoodi.

Eo.status: I 0
tietue
isäntä
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
0 = DML-ohjelmassa ei ole avattu tietuetyyppejä.
-48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

Huom! : Data-vektorin on oltava riittävän pitkä, liika pituus ei ole haitaksi. Teoreettinen yläraja on 312.

Ohjelma: DBGNI

Selite: Get names of data items of record type.

Toiminta: Ohjelma lukee sanakirjasta parametrissa annetun tietuetyypin tietokenttien nimet ja sijoittaa ne bytevektorin tietosisällöksi.

Kutsu: CALL DBGNI(kohde,data,lkm,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 data = Byte-vektori tietokenttien nimille. Nimet erotetaan toisistaan kauttaviivoilla, viimeisenä kaksi kauttaviivaa.
 lkm = Kokonaislukumuuttuja nimien lukumäärää varten.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppeä ei ole määritelty sanakirjaan.
 -4 = Tietuetyyppeä ei ole avattu DML-ohjelmassa.
 -21 = Sanakirjan luku ei onnistu.
 -22 = Tietuetyypin attribuuttiluettelon luku tiedostosta ei onnistu.
 -48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

Huom! : Data-vektorin on oltava riittävän pitkä, liika pituus ei haittaa.

Ohjelma: DBGAN

Selite: Get all names of opened record types of data base.

Toiminta: Ohjelma kerää tietokannan DML-ohjelmassa avattujen tietuetyyppien nimet ja niiden tietokenttien nimet datavektoriin.

Kutsu: CALL DBGAN(data,lkm,ipk)

Parametrit: data = Byte-vektori, jossa nimet kauttaviivoilla erotettuna, viimeisenä kaksi kauttaviivaa.
 lkm = Nimien lukumäärä data-vektorissa.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -15 = Tietuetyyppiä ei ole määritelty sanakirjaan.
 -16 = Tietuetyypin attribuuttiluettelossa olevaa attribuuttia ei ole määritelty sanakirjaan.
 -21 = Sanakirjan luku ei onnistunut.
 -22 = Joko tietokannan tietuetyyppiluettelon tai jonkin tietuetyypin attribuuttiluettelon luku ei onnistunut.

Huom! : Ohjelma kerää tiedot vain DML-ohjelmassa avattuja tietuetyypeistä. Data-vektorin on oltava riittävän pitkä, liika pituus ei haittaa.

Ohjelma: DBGKR

Selite: Get key based on current record.

Toiminta: Ohjelma sijoittaa parametrissa annetun tietuetyypin esilläolevan tietueen käsittelyosoitteen paluumuuttujan arvoksi.

Kutsu: CALL DBGKR(kohde,kos,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 kos = Kokonaislukumuuttuja käsittelyosoitteen arvoa varten.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppejä ei ole määritelty sanakirjaan.
 -4 = Tietuetyyppejä ei ole avattu DML-ohjelmassa.
 -48 = Käyttäjällä ei ole lukuoikeutta tietokantaan.

Huom!: Käsittelyosoite on tietuetyypikohtainen.

Ohjelma: DBLOOK

Selite: Look up current record occurrence of record type.

Toiminta: Ohjelma tulostaa parametrissa annetun tietuetyypin esilläolevan tietue-esiintymän päätteelle.

Kutsu: CALL DBLOOK(kohde,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
-1 = Tietuetyypsiä ei ole määritelty sanakirjaan.
-5 = Tietuetyypsiä ei ole tietokannan tietuetyyppiluettelossa.
-21 = Sanakirjan luku ei onnistu.
-26 = Tietue-esiintymän luku tiedostosta ei onnistunut.
-41 = Tietuetyypin formaatti ei vastaa tietokannan tietosisältöä.
-49 = Käyttäjällä ei ole oikeutta tietokannan käyttöön.

Huom! : Käyttäjän on huolehdittava siitä, että tulostuslaite on asetettu oikeaan tilaan esimerkiksi yli 80 merkin mittaisten tietueiden tulostuksessa.

Ohjelma: DBFIGU

Selite: Show figure of the data base organization

Toiminta: Ohjelma kuvaa kutsuttaessa tietokannan tietue-
tyyppien keskinäiset suhteet (VT100, VDU-210)
pääteelle.

Kutsu: CALL DBFIGU

Parametrit:

Eo.status: I 0
tietue
isäntä
jäsen

Paluukoodit:

Huom!: Ohjelmaa kutsutaan DBINIT-ohjelman suorituksen
jälkeen. Ohjelma ei edellytä tietuetyyppien
avaamista DBOPEN-ohjelmalla.
Esim.:

```
CALL DBINIT('TESTBASE',' ',IPK)
CALL DBFIGU
STOP
END
```

Esimerkkiohjelma kysyy käyttäjältä käyttäjätun-
nuksen ja salasanan, avaa tietokannan ja piirtää
kuvan pääteelle.

1.4.4 Hakuohjelmat

Ohjelma: DBFNR

Selite: Find next record occurrence of record type.

Toiminta: Ohjelma etsii annetun tietuetyypin esilläolevan tietue-esiintymän loogisen seuraajan ja ottaa sen esille tietuetyypin tietueista.

Kutsu: CALL DBFNR(kohde,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne ioko bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X X
isäntä
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
2 = Esilläolevalla tietue-esiintymällä ei ole loogista seuraajaa setissä. Ylimmällä tasolla esilläoleva tietue on loogisesti viimeinen.
-1 = Tietuetyyppeä ei ole määritelty sanakirjaan.
-13 = Tietuetyypin tiedoston laitemäärittely epäonnistui.
-26 = Tietuetyypin tiedoston luku epäonnistui.

Huom!: Jos tietuetyypin esiintymien muodostamassa listassa ei kyetä etenemään, jää esilläolevaksi tietueeksi oletettu edeltäjä.

Ohjelma: DBFFM

Selite: Find first member occurrence based on current owner.

Toiminta: Ohjelma etsii tietokannasta annetun setin esilläolevan isäntätietue-esiintymän ensimmäisen jäsentietue-esiintymän ja ottaa sen esille settistä.

Kutsu: CALL DBFFM(set,ipk)

Parametrit: set = Setin nimi tai lyhenne joko bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue
isäntä X
jäsen X

Paluukoodit: 1 = Onnistunut suoritus.
2 = Isännällä ei ole jäsenesiintymiä.
-1 = Setin isäntätietuetyyppejä ei ole määriteltä sanakirjaan.
-3 = Settiä ei ole määriteltä sanakirjaan.
-11 = Jäsentietuetyyppejä ei ole nimetty tietuetyypin määrittelyssä sanakirjaan.
-13 = Isäntätietuetyypin tiedoston laitemäärittely ei onnistu.
-21 = Sanakirjan luku ei onnistu.
-27 = Isäntätietuetyypin esiintymän luku tiedostosta epäonnistui.
-28 = Jäsentietuetyypin esiintymän luku epäonnistui.

Huom!: Esillä olevan jäsenesiintymän arvoksi tulee 0 mikäli isäntäesiintymällä ei ole jäseniä.

Ohjelma: DBFNM

Selite: Find next member occurrence based on current member.

Toiminta: Ohjelma etsii parametrina annetun setin jäsen-tietueesiintymän loogisen seuraajan ja ottaa sen esille setistä.

Kutsu: CALL DBFNM(set,ipk)

Parametrit: set = Setin nimi tai lyhenne byte-vektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue
isäntä
jäsen X X

Paluukoodit: 1 = Onnistunut suoritus.
2 = Jäsenellä ei ole loogista seuraajaa setissä.
-3 = Settiä ei ole määritelty sanakirjaan.
-13 = Isäntä- tai jäsentietuetyypin tiedoston laitemäärityksessä epäonnistuttiin.
-21 = Sanakirjan tiedoston luku epäonnistui.
-27 = Isäntätietuetyypin esiintymän luku tiedostosta epäonnistui.
-28 = Jäsentietuetyypin esiintymän luku tiedostosta epäonnistui.

Huom!: Mikäli jäsenesiintymällä ei ole loogista seuraajaa setissä, jää oletettu edeltäjä esilläolevaksi jäsenesiintymäksi setissä.

1.4.5 Käsittelyosoitteiden asetusohjelmat

Ohjelma: DBSRK

Selite: Set record based on data base key.

Toiminta: Ohjelma ottaa esille tietuetyypin esiintymän, joka osoitetaan parametrissa annetulla käsittelyosoitteella.

Kutsu: CALL DBSRK(kohde,kos,ipk)

Parametrit: kohde = Tietuetyypin nimi tai lyhenne bytevektorissa (30) tai literaalina.
 kos = Käsittelyosoite kokonaislukuna tai kokonaislukumuuttujassa.
 ipk = Paluukoodi.

Eo.status: I 0
 tietue X
 isäntä
 jäsen

Paluukoodit: 1 = Onnistunut suoritus.
 -1 = Tietuetyyppejä ei ole määritelty sanakirjaan.
 -4 = Tietuetyyppejä ei ole avattu DML-ohjelmassa.

Huom!: Ohjelman oikea toiminta voidaan varmistaa DBCHEK-ohjelmalla.

Ohjelma: DBSRO

Selite: Set record occurrence based on current owner.

Toiminta: Ohjelma ottaa esille parametrissa annetun setin isäntätietuetyypin esiintymän esilläolevan tietue-esiintymän perusteella.

Kutsu: CALL DBSRO(set,ipk)

Parametrit: set = Setin nimi tai lyhenne joko bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä X
jäsen

Paluukoodit: 1 = Onnistunut suoritus.
-3 = Settiä ei ole määritelty sanakirjaan.
-6 = Settiä ei ole tietokannan settiluettelossa.
-9 = Annetun setin isäntätietuetyyppeä ei ole avattu.

Huom!: Ohjelman oikea toiminta voidaan varmistaa DBCHEK-aliohjelmalla.

Ohjelma: DBSRM

Selite: Set record occurrence based on current member.

Toiminta: Ohjelma ottaa esille tietuetyypin esiintymän, joka on esilläolevana jäsenesiintymänä annetussa setissä.

Kutsu: CALL DBSRM(set,ipk)

Parametrit: set = Setin nimi tai lyhenne joko bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä
jäsen X

Paluukoodit: 1 = Onnistunut suoritus.
-3 = Settiä ei ole määritelty sanakirjaan.
-6 = Settiä ei ole tietokannan settiluettelossa.
-10 = Setin jäsentietuetyyppejä ei ole avattu DML-ohjelmassa.

Huom!: Ohjelman oikea toiminta voidaan varmistaa DBCHEK-aliohjelmalla.

Ohjelma: DBSOR

Selite: Set owner based on current record.

Toiminta: Ohjelma ottaa annetun setin isäntäesiintymän esille esilläolevan tietue-esiintymän perusteella.

Kutsu: CALL DBSOR(set,ipk)

Parametrit: set = Setin nimi tai lyhenne byte-vektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä X
jäsen

Paluukoodit: 1 = Onnistunut suoritus
-3 = Settiä ei ole määritelty sanakirjaan.
-6 = Settiä ei ole tietokannan settiluettelossa.
-9 = Annetun setin isäntätietuetyyppejä ei ole avattu DML-ohjelmassa.

Huom! : Ohjelman oikea toiminta voidaan varmistaa DBCHEK-aliohjelmalla.

Ohjelma: DBSMR

Selite: Set member based on current record.

Toiminta: Ohjelma ottaa parametrissa annetun setin jäsenesiintymän esille tietuetyypin esilläolevan esiintymän perusteella.

Kutsu: CALL DBSMR(kohde,ipk)

Parametrit: kohde = Attribuutin nimi tai lyhenne bytevektorissa (30) tai literaalina.
ipk = Paluukoodi.

Eo.status: I 0
tietue X
isäntä
jäsen X

Paluukoodit: 1 = Onnistunut suoritus.
-3 = Annettua settiä ei ole määritelty tietokantaan.
-6 = Settiä ei ole tietokannan settiluettelossa.
-10 = Setin jäsentietuetyyppejä ei ole avattu DML-ohjelmassa.

Huom!: Ohjelman oikea toiminta voidaan varmistaa DBCHEK-aliohjelmalla.

1.4.6 Loogiset testifunktiot

- Ohjelmat:** DBEQ, DBLT, DBLE, DBNE, DBGE, DBGT
- Selite:** Value of data field equals to, is less than, is less than or equals to, differs from, is greater than or equals to, is greater than given value.
- Toiminta:** Ohjelmat vertaavat annetun attribuutin esillä olevan tietue-esiintymän ao. tietokentän arvoa parametrina annettuun vertailuarvoon ja asettavat loogisen funktion arvoksi .true., jos muuttujan arvo on yhtäsuuri, pienempi, pienempi tai yhtäsuuri, eri suuri, suurempi tai yhtäsuuri, suurempi kuin kentän arvo. Muussa tapauksessa funktio saa arvon .false..
- Kutsu:** Loogisena funktiona.
(Esim. if(DBEQ(maare,iarvo))then ...)
- Parametrit:** maare = Attribuutin nimi tai lyhenne joko bytevektorissa (30) tai literaalina.
iarvo = Vertailuarvo muuttujassa tai vakiona.
- Eo.status:**
- | | | |
|--------|---|---|
| | I | O |
| tietue | X | |
| isäntä | | |
| jäsen | | |
- Paluukoodit:**
- Huom!:** Ohjelmat on määriteltävä pääohjelmassa loogiseksi funktioiksi, esim. "logical DBEQ".

Ohjelma: DBTEST

Selite: Test data field of current record occurrence.

Toiminta: Ohjelma testaa parametrissa annetun attribuutin esillä olevan tietueen tietokentän arvoa parametrina annettuun vertailuarvoon.
 Looginen funktio saa arvokseen :true., jos

- 1) annettu vertailufunktio on 'EQ' ja vertailuarvo ja tietue-esiintymän tietokentän arvo ovat yhtä suuria
- 2) annettu vertailufunktio on 'NE' ja vertailuarvo ja tietue-esiintymän tietokentän arvo ovat eri suuria
- 3) annettu vertailufunktio on 'LE' ja tietue-esiintymän tietokentän arvo on pienempi tai yhtä suuri kuin vertailuarvo
- 4) annettu vertailufunktio on 'LT' ja tietue-esiintymän tietokentän arvo on pienempi kuin vertailuarvo
- 5) annettu vertailufunktio on 'GE' ja tietue-esiintymän tietokentän arvo on suurempi tai yhtä suuri kuin vertailuarvo
- 6) annettu vertailufunktio on 'GT' ja tietue-esiintymän tietokentän arvo on suurempi kuin vertailuarvo.

Muussa tapauksessa funktio saa arvon .false..

Kutsu: Loogisena funktiona.
 (Esim. if(DBTEST(maare,vf,iarvo))then ...)

Parametrit: maare = Attribuutin nimi tai lyhenne joko bytevektorissa (30) tai literaalina.
 vf = Vertailufunktio.
 Vakioteksti: LT, LE, EQ, GE, GT tai NE joko literaaleina tai byte-vektorina (2).
 iarvo = Vertailuarvo muuttujassa tai vakiona.

Eo.status: I 0
 tietue X
 isäntä
 jäsen

Huom!: Ohjelma on määriteltävä pääohjelmassa loogiseksi funktioksi lauseella "logical DBTEST".

1.4.7 Apuohjelmat

Ohielma: DBCHEK
 Selite: Check status of navigation
 Toiminta: Ohielma kuvaa päätteelle tietuetyyppien ja settien käsittelyosoitteet DML-ohjelman suorituksen jälkeen. Ohielman tulostus on muotoa:

```

====DBXXXX=====
      RT1      RT2      RT3
CRE      1      1      1
      SET1     SET2
COW      1      1
CME      1      1
=====

```

Tulostusmallin symbolit merkitsevät seuraavaa:
 DBXXXX - Sen DML-ohjelman nimi, joka on suoritettu viimeisenä, ennen DBCHEK-ohjelmaan tuloa.
 RT1 ine. = Tietuetyyppien lyhenteet.
 SET1 ine. = Settien lyhenteet.
 CRE = Current record.
 COW = Current owner.
 CME = Current member.

Kutsu: CALL DBCHEK

Parametrit:

Eo.status: I 0
 tietue X
 isäntä X
 jäsen X

Paluukoodit:

Huom!: Ohielmaa kannattaa käyttää varsinkin ladattaessa tietokantaa. Tulostus saadaan havainnolliseksi, kun taulukko lukitaan päätteen putkelle näytönkäsittelyohjelmia hyväksikäyttäen.

2. ESIMERKKIOHJELMIA

Esimerkkiohjelmissa on käytetty seuraavia kirjastorutiineja:

nolla = Täyttää parametrissa annetun vektorin täyttö-
merkillä.
move = Siirtää merkkijonon vektorista toiseen.
nulchg = Siirtää vektorin sisältämät merkit toiseen
vektoriin ja sijoittaa annetun täyttömerkin
0:n sisältäviin vektorin alkioihin.
merk = Etsii merkkivektorista annettua merkkiä.

```
c-- Esimerkkiohjelma 1.
c-- Erkki Kaila
c-- 5.10.82
c-- Ohjelma luo "taimisto"-tietokannan tiedostoon talletetusta
c-- datasta.
      integer data(20),lohko(2),koeala,koepuu(16),lukupu(4)
      call assign(30,'salla.har')
c...Alustetaan tietokanta ja avataan tietuetyypit
      call dbinit('taimisto','Erkki Kaila/e',ipk)
      call dbopen('lohko',ipk)
      call dbopen('koeala',ipk)
      call dbopen('koepuu',ipk)
      call dbopen('lukupuu',ipk)
      iv=9999999
      iv2=9999999
      lask=0
c...Luetaan syöttötietoja tehdään joitakin laskennallisia
muutoksia
100  read(30,200,end=900) data
200  format(3x,i2,i1,2i3,i1,6i3,4i2,2i1,i2,2i3)
      if(data(1).eq.iv) goto 10
      lask=lask+1
      iv=data(1)
      lohko(1)=lask
      lohko(2)=data(1)-(data(1)/10*10)
c...Tietuetyypin LOHKO tietue-esiintymä on pakattu
c   vektoriin lohko. Talletetaan se tietokantaan.
      call dbcrs('lohko',lohko,kos,ipk)
      if(ipk.lt.0) stop
c...Asetetaan talletettu LOHKO-tietue isännäksi seuraaville
c   jäsentietuetyypeille
      call dbsor('lohko-koeala',ipk)
      call dbsor('lohko-koepuu',ipk)
      type *,'lohko luotu',lask
10   if(data(2).eq.iv2) goto 20
```

```
c...Koealan tiedot pakattu, talletetaan tietue-esiintymä
      call dbcrs('koeala',data(2),kos,ipk)
      if(ipk.lt.0) stop
c...Asetetaan talletettu koealatietue isännäksi seuraaville
      lukupuille.
      call dbsor('koeala-lukupuu',ipk)
      iv2=data(2)
20    lukupu(1)=data(5)
      lukupu(2)=data(6)
      lukupu(3)=data(7)
      lukupu(4)=data(8)
c...Lukupuun tiedot pakattu. Talletetaan ne.
      call dbcrs('lukupuu',lukupu,kos,ipk)
      if(ipk.lt.0) stop
      if(data(19).eq.0) goto 100
      do 40 i=5,20
      koepuu(i-4)=data(i)
40    continue
c...Koepuun tiedot pakattu. Talletetaan ne.
      call dbcrs('koepuu',koepuu,kos,ipk)
      if(ipk.lt.0) stop
      goto 100
c...Suljetaan tietuetyypit
900   call dbclos('ALL',ipk)
      call exit
      end
```

```

c-- Esimerkkiohjelma 2.
c-- Erkki Kaila
c-- 5.10.82
c-- Ohjelma kysyy käyttäjältä koepuuta koskevan attribuutin
c-- ehdon ja arvon sekä hakee "taimisto"-tietokannan tietue-
c-- esiintymät, jotka täyttävät annetun ehdon ja tulostaa ne
c-- päätteelle.
c--
      byte v,vekt(30),op(2)
      Logical dbtest
c...Alustetaan tietokanta ja avataan tietuetyypit
      call dbinit('taimisto',' ',ipk)
      call dbopen('lohko',ipk)
      call dbopen('koeala',ipk)
      call dbopen('koepuu',ipk)
      call dbopen('lukupuu',ipk)
c
100      type 1000
1000     format(' anna ehto (attribuutti,oper,arvo)>','$)
c...Otetaan tietuetyyppi esille tietueavaimen perusteella
c      Tietueavain=1
      call dbsrk('lohko',1,ipk)
      read(5,1100) vekt,op,iarvo
1100     format(30a1/2a1/i5)
c...Edetään lohkotietueita pitkin
      do 10 i=1,18
          call dbfnr('lohko',ipk)
c...Otetaan jäsen esille
          call dbsor('lokp',ipk)
c...Etsitään ensimmäinen jäsentietue
          call dbffm('lokp',ipk)
c...Otetaan sen tietue-esiintymä esille
          call dbsrm('lokp',ipk)
c...Seuraava jäsentietue-esiintymä
1      call dbfnr('koepuu',ipk)
          if(ipk.ne.1) goto 10
c...Luetaan käsittelyosoite muuttujaan kos
          call dbgkr('koepuu',kos,ipk)
c...Onko ehto voimassa, jos on, suoritetaan kpinol
          if(dbtest(vekt,op,iarvo))
1      call kpinol(pi,kos,it,250)
          goto 1
10      continue
      do 20 i=1,it
c...Koepuu-tietue esille
          call dbsrk('koepuu',pi(i),ipk)
c...Tulostetaan tietue
          call dblook('koepuu',ipk)
20      continue
          call kysy(' jatkatko>',v,l,n)
          if(n.eq.0.or.v.eq.'e') stop
          goto 100
      end

```

```

c-- Esimerkkiohjelma 3.
c-- Erkki Kaila
c-- 5.10.82
c-- Ohjelma lukee sanakirjasta "taimisto"-tietokannan tieto-
c-- sisältöä ja tulostaa päätteelle käyttökelpoisen kaavio-
c-- raportin.
c--
      byte maare(30),lyhe(4),fo(4),data(5000),apu(30)
c...Tietokannan alustus ja tietuetyyppien avaukset
      call dbinit('taimisto',' ',ipk)
      call dbopen('lohko',ipk)
      call dbopen('kp',ipk)
      call dbopen('lp',ipk)
      call dbopen('koeala',ipk)
c...Luetaan termit
      call dbgni('lo',data,lkm,ipk)
c...Tulostetaan siististi päätteelle
      type 1050
1050   format(/// lohko//' ===')
      il=0
      do 10 ix=1,lkm
          ia=il+1
          il=merk(///,data,ia,5000)
          call nolla(maare,30,' ')
          call move(data,maare,ia,il-1,1,il-ia-1)
c...Luetaan sanakirjan tiedot
      call dbgii(maare,lyhe,ind,isar,mi,ma,ipt,
1         fo,ipk)
          call nulchg(maare,30,' ',apu)
          type 1000,apu,lyhe,ind,isar,mi,ma,ipt,fo
1000   format(lx,30al,lx,4al,3(lx,i5),lx,il0,lx,
1         i5,lx,4al)
10     continue
c...Luetaan termit
      call dbgni('koeala',data,lkm,ipk)
      type 1051
1051   format(///koeala//lx,11('='))
      il=0
      do 11 ix=1,lkm
          ia=il+1
          il=merk(///,data,ia,5000)
          call nolla(maare,30,' ')
          call move(data,maare,ia,il-1,1,il-ia-1)
c...Luetaan tietuetyypin tiedot sanakirjasta
      call dbgii(maare,lyhe,ind,isar,mi,ma,ipt,
1         fo,ipk)
          call nulchg(maare,30,' ',apu)
          type 1000,apu,lyhe,ind,isar,mi,ma,ipt,fo
11     continue
c...Ja termit
      call dbgni('koepuu',data,lkm,ipk)
      type 1052
1052   format(/// koepuu//' =====')
      il=0
      do 12 ix=1,lkm
          ia=il+1

```

```

        il=merk('/',data,ia,5000)
        call nolla(maare,30,' ')
        call move(data,maare,ia,il-1,1,il-ia-1)
c...tietuetyypin tiedot
        call dbgii(maare,lyhe,ind,isar,mi,ma,ipt,
1          fo,ipk)
        call nulchg(maare,30,' ',apu)
        type 1000,apu,lyhe,ind,isar,mi,ma,ipt,fo
12      continue
c...termit
        call dbgii('lukupuu',data,lkm,ipk)
        type 1053
1053    format(/// ' lukupuu'/' =====')
        il=0
        do 13 ix=1,lkm
            ia=il+1
            il=merk('/',data,ia,5000)
            call nolla(maare,30,' ')
            call move(data,maare,ia,il-1,1,il-ia-1)
            call dbgii(maare,lyhe,ind,isar,mi,ma,ipt,
1          fo,ipk)
            call nulchg(maare,30,' ',apu)
            type 1000,apu,lyhe,ind,isar,mi,ma,ipt,fo
13      continue
        call exit
        end

```

```

c
c-- Esimerkkiohjelma 4.
c-- Markku Taipale
c-- 10.6.84
c-- Aliohjelma hoitaa TILAKANTA-tietokannan omistaja-
c-- ryhmä-attribuutin merkkitiedon talletuksen termiluette-
c-- loon. Tietokanta ja tietuetyypit on avattu pääoh-
c-- jelmassa. Parametreina välitetään havaintoa vastaava
c-- omistajaryhmä (omry). Pääohjelmalle palautetaan indeksi,
c-- jolla merkkitieto on löydettävissä termiluettelosta.
c--
      subroutine update(omry,ind)
      byte omry(80)
      integer ind
c
c-- Aluksi tutkitaan onko ko. merkkitieto jo viety termi-
c-- luetteloon. Mikäli on, palautetaan heti sitä vastaava
c-- indeksi pääohjelmalle talletettavaksi datan mukana.
      index=1
      call dbgdt('omistajaryhmä',omry,index,3,1,ipk)
c...Paluukoodin arvolla 1 suoritus onnistui ja termi löytyi,
c...muuttujassa index palutetaan termiä vastaava indeksi.
      if(ipk.eq.1) then
          ind=index
          return
      end if
c...Termiä ei löytynyt, joten talletetaan se.
      call dbmdt('omistajaryhmä',' ',omry,0,index,1,6,ipk)
c...Suoritustapa 6 ilmoittaa, että termi lisättiin termi-
c...luettelon loppuun ja indeksoitiin 1:llä suurinta
c...termiluettelon indeksiä kasvattamalla.
c...Parametrissa index on termiin liittyvä indeksi.
      ind=index
      return
      end

```

3. RAJOITUKSET

Koska TUTKAN ohjelmat varaavat I/O-kanavia dynaamisesti, ei ohjelmien käytön aikana voi tarkkaan tietää mitkä kanavanumerot ovat vapaita. Kanavia 5 ja 6 voi käyttää käyttöjärjestelmän oletusten mukaisesti, mutta muut kanavat 1- 4 ja 7-20 ovat varattuja. Ongelmilta vältytään, jos omissa ohjelmissa varataan tiedostot kanavanumerosta 30 alkaen.

4. VIRHEILMOITUKSET

DML-ohjelmiston tuottamat virheilmoitukset ovat muotoa:

DBxxxx: Error message,

missä DBxxxx on ohjelma, jossa virhetilanteeseen päädyttiin ja "Error message" virheilmoitus. Seuraavassa esitetään ja selitetään datan käsittelyohjelmiston tuottamat virheilmoitukset. Lainausmerkeissä kirjoitetut nimet ovat muuttujia, jotka todellisissa virheilmoituksissa korvataan virhetilanteen aiheuttaneen setin, attribuutin jne. tietokannan kaaviossa määrättyllä nimellä.

DBxxxx: Attribute of record type "record type" has improper output format

Tietuetyypin attribuutille on määrätty kaaviossa tulostusformaatti, joka ei sovellu aktiivin tietue-esiintymän tulostukseen. Tietuetyypin attribuuttien formaatit voi muuttaa DMP-ohjelmalla. (Ks. luku II).

DBxxxx: Attribute "attribute" of record type "record type" is not defined in dictionary

Annetun tietuetyypin annettua attribuuttia ei ole määritelty sanakirjaan.

DBxxxx: Current member occurrence "record type" is improper

Esillä oleva jäsenesiintymä on kelvoton. Käsittelyosoite on väärä tai tietueen sisältö on viallinen.

DBxxxx: Current owner occurrence "record type" is improper

Esillä oleva isäntäesiintymä on kelvoton. Käsittelyosoite on väärä tai tietueen sisältö on viallinen.

DBxxxx: Current record occurrence of attribute "attribute" is improper

Esillä oleva tietue-esiintymä on kelvoton. Käsittelyosoite on väärä tai tietueen sisältö on viallinen.

DBxxxx: Dictionary is not readable

Tietokannan sanakirja puuttuu tai on viallinen.

DBxxxx: File of classification of attribute
"attribute" is not readable

Tiedosto, jossa on annetun attribuutin luokitukset, puuttuu tai on viallinen.

DBxxxx: File "database".ava is not readable

Tietokannan tilanvaraustiedosto puuttuu tai on viallinen.

DBxxxx: List of attributes of record type
"record type" is incorrect

Tiedosto, jossa on lista tietuetyypin attribuuteista on viallinen.

DBxxxx: List of classifications of attribute
"attribute" is incorrect

Tiedosto, jossa on annetun attribuutin luokitukset, on viallinen.

DBxxxx: Member type of set "set" is not catalogued

Annetun setin jäsentä ei ole luetteloitu. Vika löytyy todennäköisesti kaaviosta ja se voi olla korjattavissa DMP-ohjelmalla.

DBxxxx: Member type of set "set" is not open

Annetun setin jäsen on aukaistava ohjelmalla DBOPEN.

DBxxxx: Operator "operator" is unknown

Ohjelmalle syötetty vertailuoperaattori on tuntematon.

DBxxxx: Owner type of set "set" is not catalogued

Annetun setin isäntää ei ole luetteloitu. Vika löytyy todennäköisesti kaaviosta ja se voi olla korjattavissa DMP-ohjelmalla.

DBxxxx: Owner type of set "set" is not open

Annetun setin isäntä on aukaistava ohjelmalla DBOPEN.

DBxxxx: Record specifier of owner type of set "set"
is improper

Annetun setin isäntätietuetyypin tietueosoitin on saanut viallisen arvon.

DBxxxx: Record type "record type" is catalogued but not defined in dictionary

Annettu tietuetyyppi on luetteloitu, mutta sitä ei ole määritelty sanakirjassa.

DBxxxx: Record type "record type" is not catalogued

Annettua tietuetyyppeä ei ole luetteloitu.

DBxxxx: Record type "record type" is not defined in dictionary

Annettua tietuetyyppeä ei ole määritelty sanakirjassa.

DBxxxx: Record type "record type" is not open

Tietuetyyppi on aukaistava ohjelmalla DBOPEN.

DBxxxx: Record type "record type" of attribute "attribute" is not defined in dictionary

Tietuetyyppeä, johon annettu attribuutti kuuluu, ei ole määritelty sanakirjassa.

DBxxxx: Set type "set" is not defined in dictionary

Annettua settiä ei ole määritelty sanakirjassa.

DBxxxx: Set type "set" is not catalogued

Annettua settiä ei ole luetteloitu.

DBxxxx: Sortkey of set "set" is not defined in dictionary

Annetulle setille ei ole määritelty lajitteluavainta sanakirjassa.

DBxxxx: Unable to close record type "record type"

Annetun tietuetyypin sulkeminen ei onnistu.

DBxxxx: Unable to read record occurrence of member of set "set"

Annetun setin jäsenen tietue-esiintymän luku ei onnistu.

DBxxxx: Unable to read record occurrence of owner of set "set"

Annetun setin isäntätietuetyypin tietue-esiintymän luku ei onnistu.

DBxxxx: Unable to read record occurrence of record type "record type"

Annetun tietuetyypin tietue-esiintymän luku ei onnistu.

DBxxxx: Unable to store record occurrence of member of set "set"

Annetun setin jäsentietue-esiintymän talletus ei onnistu.

DBxxxx: Unable to store record occurrence of owner of set "set"

Annetun setin isäntätietue-esiintymän talletus ei onnistu.

DBxxxx: Unable to store record occurrence of record type "record type"

Annetun tietuetyypin tietue-esiintymän talletus ei onnistu.

DBxxxx: Unable to specify unit to record type "record type"

Annetun tietuetyypin laitemääritys ei onnistu. Tietuetyypin tiedosto puuttuu tai on viallinen.

DBxxxx: User is not authorized to delete

Käyttäjällä ei ole tuhoamiseen tarvittavaa käyttöoikeutta.

DBxxxx: User is not authorized to enter this database

Käyttäjällä ei ole sisääntuloon tarvittavaa käyttöoikeutta.

DBxxxx: User is not authorized to examine

Käyttäjällä ei ole listaamiseen tarvittavaa käyttöoikeutta.

DBxxxx: User is not authorized to write

Käyttäjällä ei ole kirjoittamiseen tarvittavaa käyttöoikeutta.

DBxxxx: Value of attribute "attribute" is out of limits specified by dictionary

Annetun attribuutin arvo ei ole sanakirjassa määriteltyjen minimi- ja maksimiarvojen välissä.

Seuraavat virheilmoitukset ovat edellisistä poiketen DML-ohjelmakohtaisia.

DBCLOSE: Unable to update file "xxxxxx.AVA"

Vapaan tilan osoitteet sisältävän tiedoston päivitys epäonnistui. Virheen esiintyessä on palautettava tietokannan edellinen, kunnossa oleva versio. Tiedostojen päivitys ei ole onnistunut.

DBFIGU: Unable to create picture

Tietokannan rakenne on liian kompleksinen kuvattavaksi näyttöpäätteelle.

DBFIGU: Unable to draw picture completely

Kuva tietokannan rakenteesta ei mahdu näyttöpäätteelle täydellisenä.

DBGDT : Given class "class" has not been used in classification of attribute "attribute"

Annetulla attribuutilla ei ole luokkanimeä termiluettelossa.

DBGDT : Given index "index" has not been used in classification of attribute "attribute"

Annetulla attribuutilla ei ole indeksiä termiluettelossa.

DBINIT: Authorization failure

Ohjelma ei hyväksy avauksen yhteydessä syötettyä käyttäjätunnusta tai salasanaa.

DBINIT: File of record types is incorrect

Tietuetyyppiluettelo on virheellinen.

DBINIT: File of sets is incorrect

Settiluettelo on virheellinen.

DBINIT: Given database "database" has not been created

Tietokantaa ei ole luotu.

DBOPEN: Formatted direct access is not allowed in this version

Formatoitu suorasaanti ei ole mahdollista tässä TUTKAN versiossa.

DBOPEN: Index record types are not allowed in this version

Indeksoitujen tiedostojen käyttö ei ole mahdollista tässä TUTKAN versiossa.

DBOPEN: Record type "record type" is already open

Tietuetyyppi on aukaistu aiemmin ohjelmassa.

DBOPEN: Unknown file organization specified to record type "record type"

Tietuetyypin talletustapa on outo. TUTKA 1/1 ohjelmisto hyväksyy ainoastaan formatoimattoman suorasaannin (DIUF).

VIITTEET

DATE, C. J. 1981. An Introduction to Database Systems. 3. painos. Addison-Wesley Systems, Programming Series, Reading, Mass. 547 s.

HASEMAN, William D. & WHINSTON, Andrew B. 1977. Introduction to Data Management. Richard D. Irwin Inc. Irwin Series in Information and Decision Sciences, Homewood, Ill. 423 s.

MARTIN, James. 1977. Computer Data-Base Organization. 2. painos. Prentice-Hall Series in Automatic Computation. 711 s.

LIITE A
KÄPYKANTA-TIETOKANNAN TIEDOSTOT

KAPYKANTA-tietokannan kaavion talletusmuoto peräkkäistiedostossa kapykanta.sch. Tiedosto muodostetaan DDP-ohjelmalla ja sitä voidaan käyttää myös DDP:n syöttötiedostona.

```

kapykanta
m/m
  80
  3
  1puutieto
  2kapytieto
  3siementieto
puutietopuu
  100  1
  2
  0  1
kapytieto
  1puun pituus
  2puun ika
puun pituuspit
  1  30  -1
m
M
i3
puun ikaika
  1  100  -1
v
  3
  1ruskea
  2punaruskea
  3harmaa
siementietosie
10000  3
  2
  0  0
  1siemenen pituus
  2lentimen pituus
siemenen pituuspit
  1  10  -1
mm
M
i2
lentimen pituuslpit
  1  30  -1
mm
M

```

```

M
i3
kapytietokap
  1000  2
  2
  0  1
siementieto
  1kavyn paino
  2kavyn vari
kavyn painopain
  1  300  -1
g
M
i3
kavyn varivari
  1  3  -1
L
i2
  2kapytieto-siementieto
puutieto-kapytietopuka
  0
kapytieto-siementietokasi
  0
  1
puun ika

```

KAPYKANTA-tietokannan kaavion tiedostot levyllä. Tiedostot muodostuvat DDP-ohjelman ajon yhteydessä. Esimerkkitietokannasta puuttuu vain tietuetyyppejä vastaavat datatiedostot.

dir

Directory DUA0:AMATMTA.BOARD.KAPYKANTAA

KAPYKA.001;1	1/3	31-JUL-1984 08:52
KAPYKA.R01;1	1/3	31-JUL-1984 08:52
KAPYKA.R02;1	1/3	31-JUL-1984 08:52
KAPYKA.R03;1	1/3	31-JUL-1984 08:52
KAPYKA.RYH;1	1/3	31-JUL-1984 08:52
KAPYKA.SET;1	1/3	31-JUL-1984 08:52
KAPYKANTA.AVA;2	1/3	31-JUL-1984 08:55
KAPYKANTA.DEV;1	2/3	31-JUL-1984 08:52
KAPYKANTA.KYS;1	20/21	31-JUL-1984 08:52
KAPYKANTA.LOG;1	2/3	31-JUL-1984 08:52
KAPYKANTA.LYH;1	2/3	31-JUL-1984 08:52
KAPYKANTA.NIM;1	40/42	31-JUL-1984 08:52
KAPYKANTA.SCH;2	4/6	31-JUL-1984 08:48

Total of 13 files, 77/99 files

LIITE B
TIETOKANNAN FYYSINEN KUVAUS
(DRP-ohjelman tulostusesimerkki)

PHYSICAL DESCRIPTION OF DATABASE:

KAPYKANTA

KAPYKANTA is a small database. It is used to demonstrate the structure of databases constructed by TUIKA.

TUTKA

31-JUL-84

DATABASE: KAPYKANTIA PROJECT: KAPYKANTIA/TUTKA

RECORD TYPE : puutieto
 ABBREVIATION : puu
 OWNER :
 STRATUM : 1
 MEMBER 1 : kapytieto
 SORTKEY 1 : puun ika
 FILE OF DATA : kapyka.D01
 FILE OF ATTRIBUTES: kapyka.R01

ATTRIBUTES:

ATTRIBUTE/ABBREVIATION	MIN	MAX	UNIT	T	IND	FORM
puun pituus						
pit	1	30 m		M	1	i3
puun ika						
ika	1	100 v		M	2	i3

TUTKA

31-JUL-84

DATABASE: KAPYKANTA PROJECT: KAPYKANIA/TUTKA

RECORD TYPE : kapytieto
 ABBREVIATION : kap
 OWNER : puutieto
 STRATUM : 2
 MEMBER 1 : siementieto
 FILE OF DATA : kapyka.D01
 FILE OF ATTRIBUTES: kapyka.R02

ATTRIBUTES:

ATTRIBUTE/ABBREVIATION	MIN	MAX	UNIT	T	IND	FORM
kavyn paino pain	1	300 g	M	1	i3	
kavyn vari vari	1	3	L	2	i2	

TUTKA

31-JUL-84

DATABASE: KAPYKANTA PROJECT: KAPYKANTA/TUTKA

RECORD TYPE : siementieto
 ABBREVIATION : sie
 OWNER : kapytieto
 STRATUM : 3
 FILE OF DATA : kapyka.D03
 FILE OF ATTRIBUTES: kapyka.R03

ATTRIBUTES:

ATTRIBUTE/ABBREVIATION	MIN	MAX	UNIT	T	IND	FORM
siemenen pituus						
spit	1	10 mm		M	1	12
lentimen pituus						
lpit	1	30 mm		M	2	12

TUTKA

(4)

31-JUL-84

DATABASE: KAPYKANTA PROJECT: KAPYKANTA/TUTKA

CLASSIFICATION OF ATTRIBUTE: kavyn vari

- 1 ruskea
- 2 punaruskea
- 3 harmaa

LIITE C
LOGISEN KAAVION KUVAUSLOMAKKEET

KAAVION KUVAUSLOMAKE

Laatija	Pvm	Kuvaus	N:o/Versio

LIITE D
HAKEMISTO

Attribuutti	10
CODASYL	2, 4, 5, 6
DAP	53
DATABASE.AVA	14
DATABASE.DEV	14
DATABASE.KYS	14
DATABASE.LOG	14
DATABASE.LYH	14
DATABASE.NIM	14
DATABASE.RYH	16
DATABASE.SCH	13, 31, 33, 36
DATABASE.SET	16
DBCHEK	68, 101
DBCLOS	67, 73, 111
DBCR	67, 74
DBCRS	67, 75
DBDR	67
DBDRR	78
DBEQ	68, 99
DBFFM	68, 92
DBFIGU	67, 90, 112
DBFNM	68, 93
DBFNR	68, 91
DBGAN	67, 87
DBGDF	67, 80
DBGDR	79
DBGDT	67, 81, 112
DBGE	68, 99
DBGII	67, 83
DBGIR	67, 82
DBGIS	67, 84
DBGKR	67, 88
DBGNI	67, 86
DBGNR	67, 85
DBGRD	67
DBGT	68, 99
DBINIT	67, 71, 112
DBLE	68, 99
DBLOOK	67, 89
DBLT	68, 99
DBMDT	67, 76
DBNE	68, 99
DBOPEN	67, 72, 112
DBSFR	67, 77
DBSMR	68, 98
DBSOR	68, 97
DBSRK	68, 94
DBSRM	68, 96
DBSRO	68, 95
DBTEST	68, 100
DDD	64
DDL	5
DDP	31

DML	24
DMP	36, 46
DRP	51
DUM	59
Dumppaus	22, 59
Eo.status	69
Fyysinen kaavio	13
Juuri	9
Kaavio	4, 5
Käyttöoikeus	46
Lataus	17
Looginen kaavio	9
Paluukoodit	69
Päävalikko	27, 28
Raportointi	22, 51
Setti	5
Termiluettelo	11
Tietokannan hallinta	3
Tietokanta	3
Tietokantakokoelma	4
Tulostus	54
Uudelleen generointi	36

ISBN 951-40-0947-9

ISSN 0358-4283