

hyväksymispäivä arvosana

arvostelija

Tietokantojen historia ennen SQL:ää

Eila Helena Salmela

Helsinki 4.5.2007

Tietojenkäsittelyn historia -seminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1	Johdanto	1
2	Mihin tietokantaa tarvitaan?	2
3	Tietokantajärjestelmien terminologiaa	3
4	1950-luku: Tietokantojen varhaiset vuodet	5
5	1960-luku: IBM ja Apollo-ohjelma	6
6	1970-luku: historiasta kohti nykyaikaa	8
7	Yhteenveto	10
	Lähteet	10

1 Johdanto

Ensimmäiset tietokoneet käyttivät tiedon käsittelyyn tiedostoja, joista tietoa luettiin ja joihin kirjoitettiin. Kullakin tietokoneohjelmalla oli siten omat tiedostonsa, joita ne käsittelivät. 1950-luvun lopulla ja 1960-luvulla alettiin tiedostaa tarve pystyä jakamaan tietoa eri ohjelmien ja käyttäjien kesken, minkä seurauksena lähti käyntiin tietokantateknologian kehittäminen [Nor07].

Ensimmäiset keskitetyt tiedon määrittelyjärjestelmät, GE:n Mark I Report Generator vuonna 1956 ja IBM:n Information Retrieval vuonna 1958 olivat tiedonhallintajärjestelmän prototyyppejä [FS76]. 1960-luvulla tietokantakehitys sai uutta voimaa Nasan Apollo-projektista, jonka tavoitteena oli saada ensimmäinen ihminen kuuhun vuosikymmenen loppuun mennessä. IBM huolehti projektin tietojärjestelmistä yhdessä American Rockwellin ja Caterpillar tractorin kanssa. IBM:n 1960-luvun kehitystyön tuloksena syntyi IMS (Information Management System), jonka voidaan katsoa aloittaneen varsinaisen tietokannanhallintajärjestelmien vallankumouksen [LHH⁺05].

Ensimmäinen tietokantastandardi saatiin 1971, kun *Codasylin* (Conference on Data Systems Languages) asettama Data Base Task Group sai kehitettyä määritellyn verkkotietokannan hallintajärjestelmälle [VC71]. 1970-luvulla kehitettiin myös hierarkkinen tietokantamalli. Relaatiomallin kehitys lähti käyntiin vuonna 1970 julkaistusta ensimmäisestä aiheeseen liittyvästä artikkelista [Cod70]. Relaatiomalli löi itsensä varsinaisesti läpi vasta 1980-luvun alussa SQL:n myötä.

Näinä tietokantojen ensimmäisinä vuosikymmeninä käytettiin englanninkielisiä termejä data bank, data management system, data base erikseen kirjoitettuna tai myöhemmin data-base [Nor07].

2 Mihin tietokantaa tarvitaan?

Ensimmäiset tietokoneohjelmat käyttivät tiedostomuotoista tietoa. Kullakin ajettavalla ohjelmalla oli omat tiedostonsa ja ohjelmakoodiin oli koodattava tiedon käsittelyn toimenpiteet. Jos useampi ohjelma käytti samaa tietoa, siitä oli oltava oma kopionsa kullekin ohjelmalle. Yhden ohjelman päivittäessä tietoa oli huolehdittava, että kaikki tarvitsevat ohjelmat saivat päivitetyn version tiedostosta [Pat].

Sovellusten kehittyessä ja tietomäärien kasvaessa havaittiin tarve pystyä käsittelemään tietoa tehokkaammin. Tieto haluttiin saada organisaatioissa tarvitsijoiden käyttöön joustavasti, turvallisesti ja oikeellisesti. Samoin havaittiin, että koska useat ohjelmat käsitelivät samaa tietoa, oli tehokkaampaa tarjota eri ohjelmille pääsy samoihin tietoihin kuin kopioida tiedostot useisiin paikkoihin ja ylläpitää sitten tiedostojen yhdenmukaisuutta [Mar76].

Tiedostonkäsittelyjärjestelmässä (File Processing System) tiedot ovat tiedostoissa tietyssä formaatissa. Jokaiseen tiedostoa käsittelevään ohjelmaan on koodattu kyseisen tiedoston käsittelysäännöt. *Tietokannan käsittelyjärjestelmässä* (Database Processing System) kaikki käsiteltävä tieto on tallennettu keskitettyyn tietokantaan. Tietoa käsitellään tietokannan hallintajärjestelmän kautta, joka huolehtii sovelluksen lähettämien pyyntöjen toteuttamisen tietokantaan [Pat].

Tietokantaan tallennetusta tiedosta on saatavilla huomattavasti enemmän informaatiota kuin tiedostoista, koska tietoa pystytään yhdistelemään toisin kuin erillisten tiedostojen sisältämää tietoa. Tiedon toisteisuus on vähäisempää ja siten tieto säilyy yhdenmukaisempana ja oikeellisempana. Tämä helpottaa myös tiedon ylläpitoa. Yhtenäiset nimeämiskäytännöt ja eheysäännöt huolehtivat tiedon eheyden säilymisestä. Siinä missä tiedosto on kokoelma tietoa, jonka käsittely ja muokkaaminen on täysin käyttäjän vallassa, tietokanta on eräänlainen tosielämän malli, jonka oletetaan laadittujen sääntöjen mukaan pitävän itse huolta itsestään. Esimerkiksi yri-

tyksen tietokantaan ei pelkästään tallenneta yrityksen sisällä tapahtuneita tapah-
tumia vaan tietokanta kuvaa yrityksen rakennetta ja tehtävänkulkua. Jos henkilö
irtisanoutuu, ja hänet poistetaan aktiivisten työntekijöiden rekisteristä, oletamme,
ettei kyseinen henkilö nouse enää palkanmaksulistoille vaan tietokanta osaa hoitaa
tarvittavat prosessit itse. Siten proseduraalisia osia sisältävä tietokanta voidaankin
rinnastaa tietokoneohjelmaan ennemmin kuin passiiviseen tietokokoelmaan [Min74].

Tietokannan haittapuolina voidaan pitää paitsi tarvittavien ohjelmistojen ja laitteis-
tojen kalleutta, myös tietokannan perustamisen ja ylläpidon työläyttä. Tietokannan
rakenne on suunniteltava ja toteutettava, tarvittava tieto on ladattava tietokantaan,
tiedon saantitavat on määriteltävä ja tiedon ylläpidosta: poistoista, lisäyksistä, päi-
vityksistä on huolehdittava tiedon eheys säilyttäen. Lisäksi on huolehdittava tieto-
turvasta ja varmuuskopioista jotta ongelmatilanteista toipuminen on mahdollista
[Pat].

Codasylin raportissa [VC71] todettiin jo vuonna 1971, että tietokannan perustami-
nen ja ylläpito on yksi tärkeimmistä investoinneista missä tahansa usean käyt-
täjän järjestelmässä. Koska henkilöstökulut ovat usein suuremmat kuin laitteisto-
ja ohjelmistokulut, rajapinta, joka hyödyttää sekä asiantuntijoita että loppukäyt-
täjiä, on kannattava hankinta. Henkilöstön kouluttaminen tietojärjestelmän käyt-
töön todettiin välttämättömäksi tehtäväksi, sillä se, miten tietokantaa käyttävät
henkilöt toimivat, vaikuttaa suuresti myös käyttäjän ja tietokannan välissä olevaan
rajapintaan.

3 Tietokantajärjestelmien terminologiaa

Tietokanta (data-base) on Naftaly Minskyn [Min74] mukaan jonkin tosielämän jär-
jestelmän malli, jolla on seuraavat ominaisuudet:

- Malli sisältää suuren määrän koodattua informaatiota.
- Mallin elinikä on pitkä, useista päivistä useisiin vuosiin.
- Mallia voidaan tarkastella ja käsitellä milloin tahansa sen elinkaaren aikana.
- Malli muuttuu ensisijaisesti vastauksena ulkopuolelta siihen kohdistuviin operaatioihin.

Aiemmin todettiin, että tietokanta muistuttaa enemmän tietokoneohjelmaa kuin passiivista tietokokoelmaa. Sovellus toimii yleensä välineenä, jonka avulla saadaan aikaan tietty lopputulos. Sen sisältöä tai toimintaa ei haluta tutkia, ja sen elinikä suorituksen alusta suorituksen loppuun on suhteellisen lyhyt. Nämä ominaisuudet erottavat selkeimmin sovelluksen tietokannasta.

Tietokannan hallintajärjestelmä (data-base management system) on rajapintaohjelmisto, jonka avulla tietokannan tietoja manipuloidaan [FS76]. Tietokannan hallintajärjestelmän suhde tietokantaan on vastaava kuin tietokoneohjelman suhde ohjelmakielen kääntäjään. Tietokannan hallintajärjestelmä käyttää *määrittelykieltä* (data definition language, DDL) tietokannan tietojen kuvaamiseen ja *käsittelykieltä* (data manipulation language, DML) olemassa olevan tiedon muokkaamiseen. Aivan kuten ohjelmointikielen ilmaisuvoima vaikuttaa ohjelmistojen rakenteeseen, myös DDL:n kehitys on ollut erittäin tärkeässä roolissa tietokannan hallintajärjestelmien kehityksessä mahdollistamalla useiden käyttäjien ja sovellusten käytettävissä olevien tietokantojen kuvaamisen [Min74].

Tallennetun tiedon määrittelykieli (stored-data definition language) kehitettiin kuvaamaan tallennetun tiedon sekä fyysisiä että loogisia ominaisuuksia. Työn aloitti vuonna 1971 Codasylin työryhmä *SDDTTG* (Stored-Data Definition and Translation Task Group), jonka kehittämä tiedonmäärittelykieli sisälsi kolme osaa: määrittelyn tiedon fyysiselle ja loogiselle rakenteelle ja edellisten yhteistoiminnalle [FST72].

4 1950-luku: Tietokantojen varhaiset vuodet

1950-luvulla ja pitkälti vielä 1960-luvullakin oli normaali käytäntö, että tietokoneohjelma tallensi ja käsitteli tarvitsemansa tiedot ja tiedostot [Nor07]. Tietokannan hallintajärjestelmien syntyaikana voidaan pitää 1950-luvun jälkipuoliskoa, jolloin ensimmäiset tietokoneet olivat tulleet liike-elämän käyttöön ja jolloin aloitettiin keskustelu *yleistetyistä rutiineista* (generalized routines) [FS76]. Näillä tarkoitettiin yksinkertaisia, rutiiniluontoisia tehtäviä, kuten esimerkiksi proseduuria, joka pystyi järjestämään tiedostoja. Proseduuria ohjattiin käynnistuksen yhteydessä annettavilla parametreilla. Ajatusta kehitettiin edelleen pohtimalla, voitaisiinko muita vastaavia tehtäviä, kuten tiedostojen ylläpitoa tai raporttien tuottamista automatisoida samoin menetelmin. Tämä todettiin mahdolliseksi, mutta hintana oli operatiivisen tehokkuuden heikkeneminen tai vaihtoehtoisesti tarve hankkia enemmän resursseja, kuten muistia ja prosessoritehoa. Tietokonelaitteistojen hinta/teho-suhteen voimakas lasku ja päällekkäisten tehtävien väheneminen yleistettyjen rutiinien käytön mukana mahdollistivat kehityksen käynnistymisen [FS76].

Ensimmäiset *raporttigeneraattorit* (report generator), jotka tuottivat kohtuullisella ohjelmointityöllä ja yksinkertaisilla määrittelyillä suurista tietoaaineistoista selkeitä raportteja, olivat General Electric Companyn Mark I -raporttigeneraattori ja lajittelurutiini IBM 702-koneeseen vuonna 1956 ja Mark I:stä edelleen kehitetty raporttien ja tiedostojen ylläpitoon soveltuva toiminnallisuus, Mark II, joka julkaistiin vuonna 1957 [FS76].

Massachusetts Institute of Technology (MIT) julkisti ensimmäisen varsinaisen tiedonmäärittely -toiminnallisuuden nimellä COMPOOL 1950-luvun alkupuolella. Sillä määriteltiin attribuutteja SAGE -ilmapuolustusjärjestelmän sadoille reaaliaikaisille ohjelmille [FS76]. Samaan aikaan kehitettiin ohjelmointikieliä, kuten FACT, GECOM ja PL2, jotka kaikki sisälsivät jonkinlaisen tiedonmäärittelykielen. 1950-luvun lopul-

la DDL-kehitys yhdistettiin Cobol-ohjelmointikieleen. Cobolin keskitetty *Data Division* -toiminto erotti tiedon kuvauskielen (DDL) tietoa käsittelevistä prosesseista. Koska Data Division alunperin kuvasi tiedon joko reikäkorteille tai nauhalle tallennetuksi, sovellusten käyttöönottajat havaitsivat pian ongelman eri valmistajien yhteensopimattomilla tavoilla tallettaa tieto fyysisesti. Tämä käynnisti tutkimuksen fyysisten tietorakenteiden kuvauksen ja tallennetun tiedon määrittelykielen yhteensovittamisesta tiedon siirrettävyyden helpottamiseksi [FS76].

Näin 1950-luvulla kehitetty tiedon määrittelykieli kävi riittämättömäksi tiedon tallennusvälineiden kehittymisen myötä 1960-luvulla, mikä johti tallennetun tiedon kuvausmenetelmien huomattavaan kehittymiseen 1970-luvulla.

5 1960-luku: IBM ja Apollo-ohjelma

Tietojenkäsittelyn ja tietokantojen lisääntyvä kaupallinen käyttö toi myös uusia tekniikoita markkinoille. Yksityisyyden säilyttämisen ja laadun varmistamisen lisäksi kehitettiin tiedon saatavuutta. 1950-luvulla käyttöön otetut reikäkortit ja nauha-asetat saivat 1960-luvulla haastajakseen magneettiset levyasetat, jotka mahdollistivat *tiedon suorahaun* (random access) kun magneettinauhoilta tieto oli haettava hitaampaa *peräkkäishakua* (serial access) käyttäen. General Electric Company (GE) julkisti vuonna 1961 järjestelmän nimeltä *Integrated Data Store* eli IDS, joka oli ensimmäinen magneettista levyasetaa käyttävä tietokannan hallintajärjestelmä ja toimi suunnannäyttäjänä verkkotietokantojen hallintajärjestelmien kehityksessä. Järjestelmä sisälsi myös muita uusia piirteitä, kuten tietokantakaaviot (schema) ja lokijärjestelmän [Fun99].

Codasyl perusti työryhmän kehittämään tietokantastandardia. Työryhmä julkaisi raportin Cobolin laajennoksesta tietokantoihin vuonna 1969 ja edelleen verkkomallin tietokannanhallintajärjestelmän määrittelyn vuonna -71 [Nor07]. Työryhmä teki

suosituksen verkkomallisten tietokantojen tiedonmäärittelykielen (DDL) syntaksista ja semantiikasta ja määritteli Cobolin tiedonkäsittelykielen (DML) lauseet. Suurin osa 1960-luvun kehittäjistä käytti verkkomallista tietokantaratkaisua, joka myöhemmin tuli tunnetuksi Codasylin lähestymistapana. Suurimmat tietokonevalmistajat, kuten Univac, Siemens, Honeywell ja DEC julkistivat Codasyl-lähestymistapaa tukevia tietokannanhallintajärjestelmiä.

Codasylin menetelmän suurin ongelma oli proseduraalinen, tietue kerrallaan tapahutuva tiedon käsittely. Ohjelmoijan oli navigoitava läpi tietokannan seuraten osoitimia tietueesta toiseen. Tietokannan uudelleenorganisointi aiheutti sen, että tietokantaa käyttävä ohjelma oli kirjoitettava uudelleen. IBM:n hierarkkisilla kannoilla oli sama ongelma. Siksi tietokannan läpikäyntiä kutsuttiinkin joskus 'navigoinniksi' [Fun99] s. 162.

1960-luvun tietokantakehityksen tärkeimpiä vaikuttajia oli kuitenkin USA:n presidentin, John F. Kennedyn puhe toukokuussa 1961, jossa hän haastoi Yhdysvaltain teollisuuden viemään ihmisen turvallisesti kuuhun ja takaisin vuosikymmenen loppuun mennessä. Aloitettun Apollo-projektin rahoitus oli yltäkylläinen, mikä takasi parhaat mahdolliset olosuhteet tietojärjestelmäkehitykselle [LHH⁺05].

American Rockwell voitti tarjouskilpailun avaruusaluksen rakentamisesta ja aloitti yhteistyössä IBM:n ja Caterpillar tractorin kanssa Apollo-projektin tietojärjestelmien kehittämistyön vuonna 1965 [LHH⁺05]. Vuosien 1966-67 aikana IBM:n tiimi kehitti järjestelmän nimeltä ICS/DL/I (Information Control System and Data Language/Interface). Ensimmäinen versio julkaistiin vuonna 1967. Vuonna 1969 ICS nimettiin IMS:ksi (Information Management System) ja samalla se annettiin kaupalliseen levitykseen. IBM:stä on myös huomattava, että toisin kuin muut suuret tietokonevalmistajat, se käytti tietokantajärjestelmissään verkkomallin sijasta hierarkkista tietokantamallia.

IBM:n IMS aloitti tietokannanhallintajärjestelmien vallankumouksen. Siinä toteutettiin ensimmäisen kerran periaate, jonka mukaan ohjelmakoodi tulee pitää irrallaan datasta. DL/I toimi standardoituna rajapintana, jota kutsumalla sovellusohjelmat pääsivät käsiksi tietokannan tietoihin ja IMS huolehti kannan tietojen käsittelystä ja palauttamisesta [LHH⁺05]. Tämä rajapinnan ja tietokannan tietojen hallinnan erottaminen aloitti uuden aikakauden sovellusohjelmoinnissa. Nyt pystyttiin keskittymään ohjelmakoodissa halutun tiedon käsittelyyn tarvitsematta huolehtia tiedon saantipoluista ja ylläpidosta. Koska tiedon hallinta eriytettiin ohjelmakoodista, se mahdollisti myös useiden sovellusten pääsyn samaan aikaan käsittelemään samoja tietoja. Näin tuli tarpeettomaksi säilyttää useita kopioita samoista tiedoista. Myöhemmin DL/I-rajapinnasta kehitettiin IMS:n transaktioiden hallintajärjestelmä (IMS Transaction manager) IMS:n versioon 4.

1960-luvun lopulla DL/I:ta käytettiin hierarkkisesti rakennettujen tietokantojen rajapintana. Järjestelmällä ei ollut omaa tiedonkäsittelykieltä vaan sille annettavat komennot kirjoitettiin ohjelmointikielellä, kuten Cobolilla tai PL/I:llä, ja komentoihin liitettiin parametreja, jotka kuvasivat halutun tietokantaoperaation.

6 1970-luku: historiasta kohti nykyaikaa

1970-luku oli vielä hierarkkisten ja verkkotietokantojen kulta-aikaa, jolloin sovellusten ja toteutusten määrä lisääntyi nopeasti. Magneettiset levyasemat syrjäyttivät nauha-asemat tallennusvälineenä ja niin tallennuskapasiteetti kuin tallennettavan tiedon määrä lisääntyivät nopeasti. Vaihtoehtoisten toteutusten lisääntyminen ja hintojen lasku edesauttoivat tietotekniikan yleistymistä. Jo 1960-luvulla alkanut tuoteperheiden kehittyminen jatkui. Uusien sovellusten kehittäjät lainasivat surutta aiemmista toteutuksista. Myös rinnakkaisia toteutuksia syntyi, koska kehittäjät siirtyivät organisaatiosta toiseen ja kehittivät edelleen aiempia toteutuksia [FS76].

Tallennusvälineiden kehittyminen synnytti tiedon käsittelykielille uusia vaatimuksia. Codasylin vuonna 1969 perustaman työryhmän (Stored-Data Definition and Translation Task Group) tavoitteena oli luoda kieli tallennetun tiedon kuvaamiseen. Alustava raportti julkaistiin vuonna 1970 ja myöhempi vuonna 1972. IBM kehitti monitasoisen tiedonkuvaustoiminnallisuuden (Data Independent Access Model, DIAM). Kuvaus aloitetaan tietotasolta, sen jälkeen esitetään looginen tietomalli, koodataan tiedon esitystapa ja lopuksi lisätään fyysisen tallennusvälineen kuvaus ja tapa, jolla looginen rakenne liitetään fyysiseen [FS76].

IBM:llä työskentelevä matemaatikko Edgar F. Codd julkaisi ensimmäisen kirjoituksensa relaatiomallista vuonna 1970 [Cod70]. Kirjoituksessa esitetyt relaatiomallin keskeiset hyödyt olivat tiedon riippumattomuus käytettävästä tallennusvälineestä ja laitteistosta sekä mahdollisuus automaattiseen navigointiin kannassa eli korkean tason ei-proseduraalinen kieli datan käsittelyyn [Cod70].

Vaikka Coddin esittämä malli herätti suurta mielenkiintoa ja vaikutti tutkimustyöhön, sitä ei kuitenkaan ensimmäisinä vuosina otettu kaupallisessa mielessä vakavasti vaan sitä pidettiin lähinnä älyllisenä haasteena [Fun99]. Erityisesti IBM oli panostanut IMS:ään, josta oli kehittynyt menestyvä, luotettava ja tuottoisa myyntiartikkeli. IBM kuitenkin kehitti relaatiomallista prototyypin, System R:n, jonka ensimmäinen toimiva versio julkaistiin vuosina 1978-1979. Samaan aikaan UC-Berkeley'n tiedemiehet aloittivat relaatiokannan prototyypin kehittämisen. Järjestelmää nimeltä Ingres kehitettiin yliopistopiireissä useita vuosia [Fun99].

Ingresin kyselykieli oli QUEL ja System R:n SQL. Myös useita muita relaatiomallin toteutuksia kehitettiin 1970-luvun lopulla. SQL-kyselykieli vakiintui muutamassa vuodessa standardiksi huolimatta siitä, ettei IBM havainnut kielen potentiaalia. Vuonna 1977 Larry Ellison perusti Oraclen ja alkoi kehittää ja myydä SQL-yhteensopivia tuotteita ennen kuin IBM oli tuonut yhtään SQL-tuotetta markkinoille [Fun99].

Vuonna 1980 IBM julkisti ensimmäisen kaupalliseen tuotantoon tehdyn relaatiomallin mukaisen tietokantatuotteen. SQL/DS (Structured Query Language/Data System). Samana vuonna IBM toi ensimmäiset PC:t markkinoille ja Ashton-Tate julkaisi dBase II -käyttöjärjestelmän mikrotietokoneille. Tässä voidaan katsoa siirtymän historiasta nykyaikaan; ensimmäinen SQL-standardi valmistui vuonna 1985 ja relaatiokantojen voittokulku on jatkunut 1980-luvulta tähän päivään.

7 Yhteenveto

Tietokantojen kehityshistoria on lyhyt mutta kiihkeä. 1950-luvun ensimmäisistä, yhdelle laitteistolle ja ohjelmalle kehitetyistä malleista edettiin 1960-luvun Apollo-projektin siivittämänä nopeasti joustavampiin, eri ympäristöissä ja laitteistoissa toimiviin ratkaisuihin. Laitteistojen kehittyminen on antanut sysäyksen myös tietokantojen ja tietokannanhallintajärjestelmien kehittämiseksi. Ohjelmointikielien ovat kehittyneet käsi kädessä tietokantajärjestelmien kanssa.

Apollo-projektin jälkeen 1970-luku toi mukanaan ensimmäiset prototyypit relaatiokannoista, mutta vasta Oraclen voimakas panostus SQL-yhteensopiviin tuotteisiin 1980-luvun alussa sai aikaan lopullisen läpimurron. Relaatiokantojen läpimurto ja mikrotietokoneiden tulo markkinoille siirsivät kehityksen historiasta moderniin aikaan.

Lähteet

Cod70 Edgar F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6):377–387, 1970.

- FS76 P. Fry and E. H. Sibley. Evolution of Data-Base Management Systems. *Computing Surveys*, 3(1):7–42, 1976.
- FST72 James P. Fry, Diane P. Smith, and Robert W. Taylor. An approach to stored data definition and translation. In *Proceedings of 1972 ACM-SIGFIDET workshop on Data description, access and control*, pages 13–55. ACM Press, 1972.
- Fun99 Funding a Revolution: Government Support for Computing Research. Project report, pages 157–168, National Research Council, Washington, D.C., 1999.
- LHH⁺05 R. Long, R. Hain, M. Harrington, D. Meltz, and G. Nicholls. IBM's Information Management System: Then and Now. *IBM Press*, 8(1):50–57, 2005.
- Mar76 James Martin. *Principles of Data-Base Management*. Prentice-Hall Inc., 1976.
- Min74 N. Minsky. Another Look at Data-Bases. *ACM SIGMOD Record*, 6(4):9–17, 1974.
- Nor07 Ken North. Excellence in Database Technology, 2007. WWW-osoite: http://ourworld.compuserve.com/homepages/Ken_North/db_hall.htm.
- Pat M. Pattinson. Database Design: Overview of DB Processing. WWW-osoite: http://www.csuchico.edu/acms/mpattins/Skip_Lees_DB1.htm.
- VC71 Hare Van Court. A Special Report on the SIGBDP Forum 'The New Data Base Task Group Report'. *ACM SIGMIS Database*, 3(3):1–11, 1971.