

Relaatiomalli kilpailijoinen 1960-70 luvuilla

Risto Kuusterä

Helsinki 13.5.2005

Tietojenkäsittelytieteen historia -seminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto	1
2 Tietomalleista	2
2.1 Relaatiomalli	3
2.2 Verkkomalli	4
2.3 Hierarkiamalli	6
3 Järjestelmät	7
3.1 Integrated Data Store (IDS)	8
3.2 Information Management System (IMS)	8
3.3 System R	9
4 Yhteenveto	12
Lähteet	14

1 Johdanto

Esitelmässä käsitellään tietokantojen historiaa kolmen tietomallin ja tietokantajärjestelmän kautta. Esittelyssä ovat verkko- hierarkia- ja relaatiomallit. Järjestelmistä huomion kohteena ovat ensimmäisenä onnistuneena toteutuksensa pidettävää IDS, sekä IBM:n hierarkiamallinen IMS ja saman yrityksen prototyyppijärjestelmä System R. Käsiteltävä aikakausi alkaa 1960-luvulta ja päättyy 1980-luvun alkuun.

Erilaisia tietokantoja oli toki ollut jo ennen 1960-lukua: Natsi-Saksan käytössä olleen IBM:n valmistaman reikäkorttipohjaisen väestötietokannan väitetään mahdollistaneen toisen maailmansodan aikaiset juutalaisvainot [Bla04]. Tarve digitaalisissa tietokoneissa toimiville tietokannoille syntyi, kun käytössä olleiden kortistojärjestelmien ylläpitämiseen tarvittavat työvoimakustannukset kasvoivat ja tietotekniikka kehittyi riittävästi mahdollistaen järjestelmien kehittämisen. Ensimmäiset tietokonepohjaiset tietokantajärjestelmät kehitettiin puolustusministeriöiden sekä tiedustelupalveluiden käyttöön. Tietokannanhallintajärjestelmien historian voidaan katsoa alkaneen kuitenkin vasta 1950-luvulla, jolloin tietokoneita käytettiin ensimmäisen kerran yritysten liiketoiminnan tukena. Tämä johtuu siitä, että sotilaalliseen käyttöön suunnitellut järjestelmät poikkesivat olennaisesti yritysten tarpeisiin kehitetyistä. [FS76]

Varhaisissa tietojärjestelmissä oli tyypillistä, että kullakin sovelluksella oli oma tietosisältönsä sekä oma tapansa tallettaa se. Merkittävä muutos tässä ajattelussa tapahtui 1960- ja 70-luvun taitteessa, jolloin järjestelmien kasvaessa oli tarve tallettaa tieto sellaisessa muodossa, että useammat järjestelmät voisivat hyödyntää sitä. Eräs merkittävimmistä varhaisien tietokantojen kanssa työskennelleistä henkilöistä, Charles Bachman, vertasikin ajattelumallin muutosta tietojenkäsittelytieteellisessä mielessä yhtä merkittäväksi kuin Nikolas Kopernikuksen ja Galileo Galilein aikaansaamaa muutosta maakeskeisestä maailmankuvasta aurinkokeskeiseksi maailmankuvaksi. [Bac75]

2 Tietomalleista

Jotta relaatiomallin nousu tunnetuimmaksi tavaksi mallintaa tietokanta, on syytä esittää sen keksimisen aikana markkinoilla olleet kilpailijat: verkko- ja hierarkiamallit. Toisin kuin relaatiomallin kohdalla, niiden osalta mallista puhuminen on hiukan harhaanjohtavaa. Tämä johtuu siitä, että mallinnettavat tietokannat olivat olemassa jo ennen mallien formalisointia. Relaatiomalli sen sijaan kehitettiin teoreettiselta pohjalta ja vasta mallin julkaisemisen jälkeen oli vuorossa tuotteiden kehitys.

Mallien kehitysproesseista johtuen niihin pohjautuvien tietokantojen käyttö poikkeaa toisistaan. Aikaisemmin kehitettyjen ja aikansa järjestelmien tapaan verkko- ja hierarkiamallin tietokannat ovat hyvin laitteistoläheisiä ja niiden käyttäjien vastuulla oli mahdollisimman tehokkaiden kyselyiden kirjoittaminen. Sen sijaan relaatiomallissa käyttäjä ei yleisesti ottaen ole tietoinen tietokannan sisäisistä tietorakenteista ja kyselyoptimointi on tietokannanhallintajärjestelmän vastuulla.

Relaatiomallin esitelleen paperin [Cod70] jälkeen sen kehittänyt Ted Codd ja hierarkiamallin puolesta puhunut Charles Bachman julkaisivat muiden tutkijoiden säestämänä liudan artikkeleita [Bac75, DC75, MMC76], joissa malleja vertailtiin. Kiista mallien paremmuudesta oli kiivasta, kunkin tahon pitäessä kiinni mielipiteistään. Mielenkiintoinen sivujuonne aiheessa oli 70- ja 80-luvuilla tehty tutkimus mallien yhdistämisestä siten, että samaa tietokantaa olisi voitu käyttää sekä relaatio- verkko- että hierarkiamallisen tietokannan tavoin. Tutkimusten ansiosta nykyisin monia ei-relaatiopohjaisia tietokantoja käytetään SQL-kielellä. [Lie82]

Artikkeleissaan Bachman oli sitä mieltä, että vaikka relaatiomallilla oli hieno teoreettinen pohja ja se oli hyvin dokumentoitu, ei sitä voitu pitää käyttökelpoisena ilman käytännön kokemuksia. Bachman myönsi puutteet verkkomallin dokumentoinnissa, mutta piti hyvin suurena vahvuutena kymmenen vuoden käyttökokemusta sadoista järjestelmistä, jotka oli toteutettu eri ympäristöihin. Bachmanin oli myös vaikea hyväksyä relaatiotietokannoissa

välttämättömiä keinotekoisia yksilöiviä tunnisteita. [Bac75]

Malliaan puolustanut Codd piti [DC75] relaatiomallin suurimpana etuna sen helppokäyttöisyyttä: hän näki että tulevaisuudessa tietokoneiden käyttäjäkunta ei koostuisi pelkästään ohjelmoijista, vaan että tietokantoihin oli taattava pääsy myös tavalliselle kadunmiehelle.

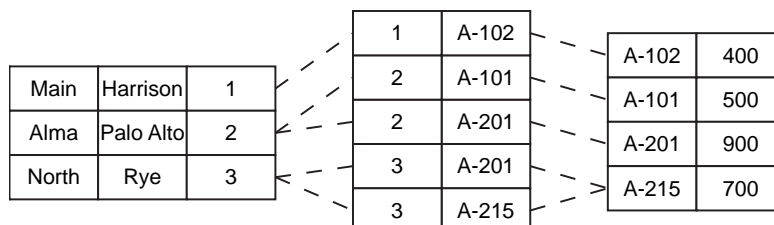
Codd oli myös osoittanut matemaattisesti, että monimutkaisuudestaan huolimatta verkkomallin tiedonkäsittelykieli ei ollut yhtään ilmaisuvoimaisempi kuin relaatiomallin kyselykieli. Hän piti relaatiomallin vahvana etuna sitä, että sen käyttäjien ei tarvinnut tuntea järjestelmän sisäisiä tallennusrakenteita kyetäkseen käyttämään sitä tehokkaasti. Tämä mahdollistaisi myös relaatiotietokannan dynaamisemman käytön. Lisäksi Codd uskoi, että relaatiotietokannassa mahdollinen taulujen ja sarakkeiden rajoittaminen ainoastaan tiettyjen käyttäjien käytettäväksi oli ominaisuus, jota ei verkkomallin tietokannassa voinut olla. [DC75]

2.1 Relaatiomalli

Relaatiomallin kehitti IBM:llä työskennellyt matemaatikko / kemisti Ted Codd vuonna 1969 [Cod70]. Relaatiomalli poikkesi julkaisuhetkellä käytössä olleista malleista niin paljon, että siihen aikaan hierarkiamallisen IMS:n parissa työskennelleet tietokantaexpertit arvioivat (myöhemmin ajatellen yllättävän tarkasti) että menisi kymmenen vuotta ennen kuin malliin pohjautuva järjestelmä olisi valmis [MBB⁺97].

Koska relaatiomallin pitäisi olla lukijoille tuttu, sitä ei esitellä tässä. Sen sijaan sitä käytetään esimerkkinä kuvassa 1, jossa on esitetty yksinkertainen pankin tietokanta. Tietokannasta esitellään myöhemmin verkko- ja hierarkiamalliset versiot. Tietokannassa on tiedot asiakkaista (nimi, katuosoite ja kaupunki) sekä näiden tileistä (yksilöllinen tilinumero ja tilin saldo). Järjestelmässä yhdellä henkilöllä voi olla monta tiliä ja tilillä monta omistajaa. Relaatiotietokanta koostuu kolmesta taulusta: CUSTOMER, ACCOUNT ja CUSTO-

MER_ACCOUNT. Asiakkaan tiedot, CUSTOMER_NAME, CUSTOMER_STREET ja CUSTOMER_CITY, ovat CUSTOMER-tilussa. Lisäksi taulussa on sarake ID, johon on talletettu asiakkaan yksilöivä tunnus. ACCOUNT-tilu koostuu attribuuteista ACCOUNT_NUMBER ja BALANCE. Välitilu CUSTOMER_ACCOUNT liittyy asiakkaan ja tämän omistamaan tilin attribuuttien ID ja ACCOUNT-NUMBER avulla.



Kuva 1: Relaatiomallin mukainen tietokanta

Jotta eri mallien mukaisien tietokantojen käsittelyä olisi helppo verrata on seuraavassa esitetty esimerkkitietokannassa ajettava SQL-kysely, jolla noudetaan kaikkien Harrisonissa asuvien asiakkaiden nimet:

```
SELECT customer_name FROM customer WHERE city="Harrison";
```

2.2 Verkkomalli

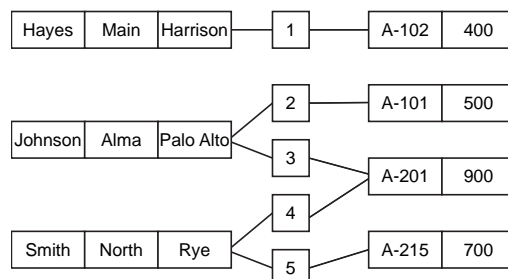
Verkkomalli kehitettiin Charles Bachmanin suunnitteleman tietokantajärjestelmä IDS:n pohjalta. Mallin sekä siihen liittyvän tiedonkäsittelykielen standardoi CODASYL:n (Conference On DATA SYstems Language) tietokantaryhmä DBTG (Database Task Group) 1960- ja 1970-lukujen taitteessa [FS76].

Relaatiomalli monimutkaisemmassa verkkomallissa tietokannan tietoalkiot ja niiden väliset suhteet erotettu eri rakenteisiin. *Attribuutteja* sisältävät *tietueet* (record) liitetään toisiinsa *joukoilla*¹ (set). Joukot mallintavat tietueiden keskinäisiä suhteita sisältämällä osoittimia tietueisiin. CODASYL:n standardissa joukot mallintavat aina 1:N-suhteita siten, et-

¹Termi ei liity joukko-opilliseen synonyymiinsä

tä joukon *omistaja*-osoitin (owner) viittaa yhteen tietueeseen ja *jäsen*-osoittimet viittaavat mielivaltaiseen määrään tyypiltään yhteneviä tietoalkioita. [SKS05a]

Kuvassa 2 DEPOSITOR-joukot (numeroitu 1-4), sisältävät osoittimet CUSTOMER- ja ACCOUNT-tietueisiin.



Kuva 2: Verkkomallin mukainen tietokanta [SKS05a]

Verkkomallissa tietoja hallitaa *isäntäkielellä* (host language) kirjoitetuilla sovelluksilla. Sovelluksissa isäntäkielen (esimerkeissä Pascalia, alunperin Gecom / Cobol) sekaan on kirjoitettu CODASYL:n määrittelemiä tietokantakomentoja. Järjestelmä pitää yllä kutakin sovellusta omaa *käyttäjän muistialuetta* (user work area), jossa on muistipaikat tietokannasta viimeksi noudetuille tietueille sekä joukko osoittimia (current of record type ja current of set type), jotka viittaavat suoraan ja joukkojen osoittimien välityksellä käsiteltäviin tietueisiin. Muistialueella on lisäksi *tilamuuttujia* (status flag), jotka ilmaisevat suoritettujen tietokantaoperaatioiden tilan. Tärkein tilamuuttuja on DB-STATUS, jossa on tallessa edellisestä tietokantaoperaatiosta mahdollisesti tullut virhekoodi. Mikäli edellinen tietokantaoperaatio onnistui, on tämän tilamuuttujan arvo 0. [SKS05a]

Tärkeimmät CODASYL-DBTG:n määrittämistä tietokantakomennoista ovat:

find: Etsii ehtojen mukaisen tietoalkion ja asettaa muistialueen osoittimet löydetyn osoittamaan tietoalkion osoitteeseen.

get: Siirtää osoittimen osoittaman tietueen käyttäjän muistialueelle.

store: Lisää määritetyn tietueen levyille.

modify: Tallettaa muistialueella olevan muokatun tietueen muutokset levyille.

erase: Poistaa osoittimen osoittaman tietueen levyltä.

Edellisessä kappaleessa esitetty SQL-kysely voitaisiin suoritetaan verkkomallin tietokannassa oheisella tavalla. Mikäli aihe kiinnostaa, on Daten ja Coddin artikkelissa [DC75] paljon hurjempia esimerkkejä.

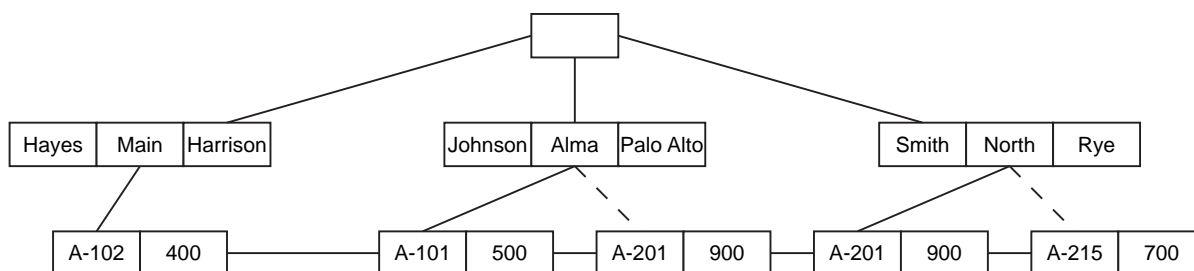
```
customer.customer.city := "Harrison";
find any customer using customer.city;
while DB-status = 0 do
    begin
        get customer;
        print(customer.customer_name);
        find duplicate customer using customer_city;
    end;
```

2.3 Hierarkiamalli

Hierarkiamalli on hyvin samankaltainen kuin verkkomalli, sillä siinäkin tietokanta koostuu erillistä tietueista ja *linkeistä* (link), jotka ovat yhteyksiä kahden tietueen välillä. Toisin kuin verkkomallissa, jossa tietokanta on malliltaan epäsäännöllinen verkko, muodostaa hierarkiamallisen tietokannan rakenne puun. Puumuoto tarkoittaa siitä, että jokaisella tietueella (juurta lukuunottamatta) on yksi vanhempi (parent) ja mielivaltainen määrä lapsia (child), jotka voivat olla minkä tahansa tyyppisiä tietueita. Koska hierarkiamalli kehitettiin Apollo-ohjelman yhteydessä kehitetyn IMS-järjestelmän pohjalta, on mallin teoriaan liittyviä artikkeleita hyvin niukasti saatavilla. [SKS05b]

Kuvassa 3 on hierarkiamallinen tietokanta. Sama tietokanta voitaisiin mallintaa myös toisella tavalla, sillä on järjestelmän suunnittelijan vastulla määrittää tietokannan hierarkia. Kustakin vanhemmasta on linkit ensimmäiseen lapseen, ja niissä tiedot vanhemmista sekä linkki seuraavaan sisarukseen tai serkkuun. [SKS05b]

Hierarkiamallisen tietokannan käsittely on hyvin samantyyppistä kuin verkkomallisen. Ohjelmia varten on käyttäjän muistialue, jossa käsitellään levyiltä noudettuja tietueita kä-



Kuva 3: Hierarkiamallin mukainen tietokanta [SKS05b]

sitellään. Verkkomalliin verrattuna komennoissa on se ero, että hierarkiamallissa **get**-komento sekä hakee ja siirtää halutun tietueen työmuistiin. Aiemmin esitellyn kyselyn suorittaminen hierarkisessa tietokannassa tapahtuisi seuraavalla tavalla:

```

get first customer
  where customer.customer_city = "Harrison"
while DB-status = 0 do
  begin
  print (customer.customer_name)
  get next customer
    where customer.customer_city = "Harrison"
  end

```

3 Järjestelmät

Seuraavassa esitellään kolme eri tietomalleihin pohjautuvaa tietokantaa: Charles Bachmanin kehittämää IDS, sekä IBM:n hierarkiamallinen IMS ja relaatiomallinen prototyyppitietokanta System R. Kaksi ensimmäistä ovat merkittäviä paitsi innovaatioidensa, myös aikanaan laajan käyttönsä ansiosta. System R:n merkitys perustuukin pääasiassa projektin aikana kehitettyyn SQL-kielen.

Vaikka relaatiomallin tietokannat ovat nykyisin yleisiä, kannattaa huomioida, että vielä vuonna 1992 yli 70% kaikista teollisuuden ja hallinnon tietokannosta toimi hierarkia- tai verkkomallin mukaisesti. [Nav92]

3.1 Integrated Data Store (IDS)

Merkittävin verkkomallinen tietokanta oli IDS (Integrated Data Store). Charles Bachmanin kehitti IDS:n General Electricille² 1961-64. Verkkomallisessa tietokantaa käytettiin korkean tason kieleen (aluksi Gecom, myöhemmin Cobol) lisätyillä komennoilla. IDS:a pidetään yleisesti ensimmäisenä onnistuneesti toteutettuna tietokantajärjestelmänä.

IDS sisälsi monia aikanaan edistyksellisiä ominaisuuksia. Yhteiskäyttöisessä IDS:ssa oli elvytysjärjestelmä, joka piti tietokannast atallessa kahta versiota, joita vertaamalla elvytys tapahtui (Shadow paging). Tietoalkioita oli mahdollista noutaa paitsi pääavaimen, myös muiden attribuuttien perusteella. [FS76]

IDS:n pohjalta kehitetty IDMS (Integrated Data Management System) on edelleen käytössä monissa suurkoneissa toimivissa raskaissa järjestelmissä. IDMS on Computer Associates:n tuote, jota operoidaan SQL-kielellä.

3.2 Information Management System (IMS)

Hierarkiamallisista tietokannoista tunnetuin on Apollo-ohjelman tavaranhallintaan kehitetty IBM:n IMS (Information Management System). Järjestelmän kantaisät ovat vuonna 1965 kehitetyt GUAM (Generalized Update Access Method) ja RATS (Remote-Access Terminal System), jotka toimivat IBM:n 7010-tietokoneessa. Näistä IBM:n kehitti yhdessä Rockwell Internationalin ja pian projektista vetäytyneen Caterpillar Tractor Corp.:n kanssa IMS:n (alkuperäiseltä nimeltään Information Control System (ICS)). [FS76, Bla98, SKS05b]

IMS:n vahvuus oli sen tehokkaassa transaktionkäsittely. Alkuperäisen IMS:n rinnakkaisuudenhallintajärjestelmä oli ensimmäisiä laatuaan. Monta samanaikaista lukuoperaatiota sallinut järjestelmä ei mahdollistanut rinnakkaisia kirjoitusoperaatioita. Ankara rinnak-

²Honeywell osti myöhemmin GE:n tietotekniikkaosaston ja Bachman jatkoi työtään siellä

kaisuudenhallinta oli aikanaan perusteltua, sillä tietokannoilla oli vain vähän yht'aikaisia käyttäjiä, jotka suorittivat pitkiä eräajoja. Tähän tuli kuitenkin muutos 1970-luvun aikana, jolloin tietokantojen yht'aikaisten käyttäjien määrä kasvoi ja transaktoista tuli lyhyempiä. IMS:sta kehitettiin enemmän keskusmuistitietokannan kaltainen järjestelmä, joka piti muistissa mahdollisimman paljon tietoa, tallettaen muutokset levyille ainoastaan *tarkastuspisteissä* (check point). [SKS05b]

IMS on edelleen tuotannossa oleva järjestelmä ja sitä käytetään laajalti raskaissa sovelluksissa, kuten voimalaitosten ja pankkien järjestelmissä. [Bla98]

3.3 System R

Relaatiotietokantojen kehityksen kannalta merkittävin yksittäinen järjestelmä on IBM:n San Josen tutkimuskeskuksessa 1970-luvulla kehitetty System R. Projektin aikana luotiin SQL-kielen ensimmäinen versio ja kehitettiin tietokantojen samanaikaisuuden hallintaan sekä tietoturvaan liittyviä käytäntöjä, jotka ovat yhä käytössä. System R:stä ei kehittynyt kaupallista menestystuotetta, mutta siihen liittyviä tieteellisiä artikkeleita julkaistiin runsaasti. Nämä paperit auttoivat muita relaatiotietokantojen kehittäjiä työssään ja mahdollistivat menestyksen Oraclen johdolla [MBB⁺97].

System R ei suinkaan ollut ensimmäinen relaatiotietokanta, sillä jo vuosina 1970 ja 1971 MIT:ssä oli kehitetty kaksi Multics-järjestelmässä toimivaa tietokantajärjestelmää MADAM (MacAIMS Data Management System) ja RDMS (Relational Data Management System) [Kim79]. Myös ensimmäinen kaupallinen relaatiotietokanta, omaa kyselykieltään käyttänyt Honeywellin Multics Relational Data Store (MRDS), julkaistiin ennen System R:n valmistumista kesällä 1976. Multics-käyttöjärjestelmässä toimineen järjestelmän kehittivät James Weeldreyer ja Oris Friesen. [McJ]

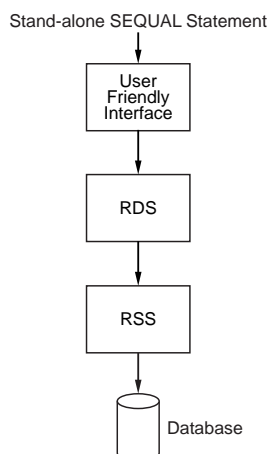
System R:n kehitys alkoi vuoden 1974 loppupuolella ja projekti päättyi vuonna 1979. Projekti koostui kolmesta vaiheesta: yhden käyttäjän prototyypin rakentamisesta, varsinaisen

monen käyttäjän järjestelmän toteuttamisesta sekä kolmen asiakkaan luona tapahtuneesta testauksesta. Projektin alkaessa oli tiedossa, että relaatiomallissa tiedon riippumattomuus tietokannan sisäisistä rakenteista on huomattavasti parempi kuin toisissa malleissa. Myös kehitettävän korkeamman abstraktiotason kyselykielen uskottiin parantavan järjestelmän käyttäjien työn tehokkuutta. Vallalla oli kuitenkin epäily siitä, olisiko tietokannanhallintajärjestelmän mahdollista muokata käyttäjän syöttämät korkean tason kyselyt tarpeeksi tehokkaiksi, jotta järjestelmän suorituskyky olisi riittävä. Epäilyt olivat samankaltaisia, kuin kymmenen vuotta aiemmin, jolloin korkeamman tason ohjelmointikieliä oli kehitetty ja ne olivat löyneet itsensä läpi. Eräs kehittäjien visiosta oli kehittää väline tavallisten loppukäyttäjien tiedonhallintaan. SQL:n läpimurto välikielenä sovelluksien ja tietokannan välissä tapahtuikin puolivahingossa. [MBB⁺97, CAB⁺81]

Prototyypin rakennus, Phase Zero, ajoittui vuosille 1974 ja 1975. Raymond Lorien kehittämän yhden käyttäjän *relaatioisaantimenetelmä*³ (relational access method) XRM:n (Extended Relational Memory) päälle rakennettu prototyyppi tarjosi mahdollisuuden tehdä tietokantakyselyjä järjestelmää varten kehitetyllä SEQUEL-kielillä (Structured English Query Language). SEQUEL:n pohjana oli Roy Boycen ja Don Chamberlinin matemaattisia notaatioita sisältänyt SQUARE-kieli [BCHK75], jonka avainsanat oli käännetty englanniksi. Koska SEQUEL oli Hawker Siddeley Aircraft Companyn tavaramerkki, jouduttiin kielen nimi muuttamaan myöhemmin SQL:ksi. Prototyypillä ei ollut kaupallista merkitystä, mutta sen avulla onnistuttiin varsinaiselle järjestelmälle hankkimaan muutama asiakas. Myöhemmin prototyypistä kehitettiin IBM:n SQL/Data System -tietokantajärjestelmä.

Varsinainen System R toteutettiin IBM:n System/370-tietokoneelle vuosina 1975-77. Järjestelmä koostui kahdesta alijärjestelmästä (kuva 4), jotka olivat kyselynkäsitteljä RDS (Relational Data System) ja saantimenetelmä RSS (Research Storage System). Järjestelmässä oli lisäksi itsenäinen käyttöliittymä UFI (User-Friendly Interface), joka mahdollisti kyselyjen suorittamisen terminaalissa. [CAB⁺81]

³Saantimenetelmä on tietokannan osajärjestelmä, joka noutaa levyiltä muistiin määritetyt monikot



Kuva 4: System R:n arkkitehtuuri [Kim79]

Kyselynkäsittelijä huolehti järjestelmän tiedon autentikoinnista muokkaamalla kyselyt muotoon, jossa tietokannan käyttäjillä oli pääsy ainoastaan heille sallittuihin tietoihin. System R mahdollisti myös tietokantänäkymien käytön. Kyselynkäsittelijä RDS:n tehtävänä oli käsitellä ensin käyttäjän syöttämät kyselyt sellaiseen muotoon, että kyselyjen tulokset olivat järjestelmän autentikointipolitiikan mukaisia. Tämän jälkeen RDS käänsi kyselyt System/370:n konekielen rutiineiksi. Kyselyjen kääntäminen konekieliseksi oli merkittävästi tehokkaampaa kuin niiden tulkkaminen.

Alusta lähtien monen käyttäjän käyttöön tarkoitettun RSS:n suunnittelu oli alkanut jo prototyypin rakennusvaiheessa. Alijärjestelmä piti sisällään järjestelmän lukitus- ja elvytystoiminnot. Kehitetty lukitusjärjestelmä oli hierarkinen ja siinä oli käytössä aikomuslukot. Nykyisetkin tietokannanhallintajärjestelmät käyttävät samaa lukituskäytäntöä. Sen sijaan tietokannan sivuista levyllä kahta versiota pitänyt elvytyskäytäntö (Shadow paging) on korvattu sofistikoituneimmilla menetelmillä [MHL⁺92].

System R:n testausvaiheessa vuosina 1978-79 järjestelmää käytettiin paitsi IBM:n sisäisessä käytössä myös Boeingilla ja suihkumoottoreita valmistaneessa Pratt & Whitneyllä sekä lääkevalmistaja W.E. Upjohnilla, jossa järjestelmä oli käytössä vielä pitkään projektin päättymisen jälkeen. Varsinaisia suorituskyky mittauksia ei testausvaiheessa tehty ja

merkittävin testauksessa tehty huomio oli se, että SQL-kieli todettiin toimivaksi ja helpokäyttöiseksi myös oikeassa käytössä. Järjestelmän tehokkuus ei kuitenkaan ole voinut olla vertailukelpoinen kilpailevien järjestelmien kanssa, sillä System R:stä ei koskaan tehty kaupallista versiota. Testikokemusten perusteella SQL-kieleen lisättiin joitakin ominaisuuksia, tärkeimpinä ulkoliitokset sekä merkkijonojen vertailua helpottavan LIKE-operaattorin. [MBB⁺97, CAB⁺81]

Projektien päätyttyä oli selvinnyt, että relaatiomallinen tietokanta oli mahdollista toteuttaa. Projektin puitteissa kehitetystä SQL-kielestä tuli ANSI-standardi 1986. SQL-kieltä hyödyntäneitä järjestelmiä julkaistiin jo aiemmin, ja esimerkiksi Oracle⁴ oli kehittänyt oman SQL:ää tukeneen tietokannanhallintajärjestelmänsä, Oracle versio 2:n, jo vuonna 1979. Oraclen kehityksessä käytettiin apuna runsaasti System R -projektin aikana julkaistuja papereita, mutta yksinkertaisena järjestelmänä se toimi huomattavasti esikuvaansa vaatimattomammalla alustalla, Digitalin PDP-11 minikoneella. Konekielellä kirjoitettu järjestelmä ei ollut kovin vakaa ja siitä puuttuivat esimerkiksi transaktiot, mutta yksinkertainen järjestelmälle löytyi käyttäjäkunta yritysten johtoportaista ja yrityksen myöhemmät järjestelmät osoittautuivat onnistuneemmiksi. [MBB⁺97, Nau04, Jac03]

4 Yhteenveto

Kunkin esitellyn järjestelmän vaikutukset on nähtävillä vielä nykypäivänä: IDS:n kehityksessä merkittävin innovaatio oli tiedon siirtäminen sovelluksista monen käyttäjän tietokantaan, IMS:n yhteydessä kehitetyt transaktionhallinta sekä elvytyskäytännöt ovat selvästi nykyjärjestelmien edeltäjiä ja System R:n ohessa kehitetty SQL-kieli on merkittävin nykyisistä tietokannankäsittelykielistä.

Tietokantojen kehitys on noudattanut samaa kehitystä kuin ohjelmointikielten: ne ovat kehittyneet hyvin laiteläheisistä kohti abstraktimpia ja käyttäjän kannalta helpompia jär-

⁴Alunperin Relational Software Inc

jestelmiä. Ohjelmointikielet ovat aina olleet tässä kehityksessä edellä ja siellä olio-ajattelu on jo vallalla, joten tulevaisuudessa saattaa olla edessä siirtyminen relaatiomallista olio-pohjaiseen tapaan mallintaa tietokanta. Tämä on sinällään ironista, sillä oliomalli voidaan nähdä verkkomallin jälkeläisenä.

Lähteet

- Bac75 Bachman, C. W., The data structure set model. *Proceedings of the 1974 ACM SIGFIDET workshop on Data description, access and control*, 1975, sivut 1–10.
- BCHK75 Boyce, R. F., Chamberlin, D. D., Hammer, M. M. ja King III, W. F., Specifying queries as relational expressions. *Communications of the ACM*, 18,11(1975).
- Bla98 Blackman, K. R., Technical note – IMS celebrates thirty years as an IBM product. *IBM Systems Journal*, 37,4(1998).
- Bla04 Black, E., IBM and the holocaust (websivu, tarkastettu 24.3.), 2004. URL www.ibmmandtheholocaust.com.
- CAB⁺81 Chamberlin, D. D., Astrahan, M. M., Blasgen, M. W., Gray, J. N., King, W. F., Lindsay, B. G., Lorie, R., Mehl, J. W., Price, T. G., Putzolu, F., Sellinger, P. G., Schkolnick, M., Slutz, D. R., Traiger, I. L., Wade, B. W. ja Youst, R. A., A history and evaluation of System R. *Communications of the ACM*, 24,10(1981), sivut 632–646.
- Cod70 Codd, E. F., A relational model of data for large shared data banks. *Communications of the ACM*, 13,6(1970), sivut 377–387.
- DC75 Date, C. J. ja Codd, E. F., The relational and network approaches: comparison of the application programming interfaces. *Proceedings of the 1974 ACM SIGFIDET workshop on Data description, access and control*, 1975, sivut 83–113.
- FS76 Fry, J. P. ja Sibley, E. H., Evolution of data-base management systems. *ACM Computing Surveys*, 8,1(1976), sivut 7–42.
- Jac03 Jacobs, K., Oracle database changed the world. *Oracle Magazine*, 2,3(2003).
- Kim79 Kim, W., Relational database systems. *ACM Computing Surveys*, 11,3(1979), sivut 185–210.
- Lie82 Lien, Y. E., On the equivalence of database models. *Journal of the Association for Computing Machinery*, 29,2(1982), sivut 333–362.
- MBB⁺97 McJones, P., Bamford, R., Blasgen, M., Cheng, D. C. J., Daudenarde, J.-J., Finkelstein, S., Gray, J., Jolls, B., Lindsay, B., Lorie, R., Mehl, J., Miller, R., Mohan, C., Nauman, J., Pong, M., Price, T., Putzolu, F., Schkolnick, M., Selinger, B., Selinger, P., Slutz, D., Traiger, I., Wade, B. ja Yost, B., The 1995 SQL reunion: People, projects, and politics. Tekninen raportti, Digital, 1997.

- McJ McJones, P., Multics Relational Data Store (MRDS) (websivu, tarkastettu 24.3.). URL www.mcjones.org/System_R/mrds.html.
- MHL⁺92 Mohan, C., Haderlen, D., Lindsay, B., Piresh, H. ja Schwarz, P., ARIES: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems*, 17,1(1992), sivut 94–162.
- MMC76 Michaels, A. S., Mittman, B. ja Carlson, C. R., A comparison of the relational and CODASYL approaches to data-base management. *ACM Computing Surveys*, 8,1(1976), sivut 125–151.
- Nau04 Naudé, F., FAQ about oracle corporation (websivu, tarkastettu 24.3.), 2004. URL www.orafaq.com/faqora.htm.
- Nav92 Navathe, S. B., Evolution of data modeling for databases. *Communications of the ACM*, 35,9(1992), sivut 112–123.
- SKS05a Silberschatz, A., Korth, H. F. ja Sudarshan, S., Appendix A: Network model (julkaistu ainoastaan webissä tarkastettu 24.3.). *Database System Concepts Fifth Edition*, 2005, URL www.cs.yale.edu/homes/avi/db-book/db5/index.html.
- SKS05b Silberschatz, A., Korth, H. F. ja Sudarshan, S., Appendix B: Hierarchical model (julkaistu ainoastaan webissä, tarkastettu 24.3.). *Database System Concepts Fifth Edition*, 2005, URL www.cs.yale.edu/homes/avi/db-book/db5/index.html.