

# 1. Relatsiooniliste andmebaaside ja andmemudelite areng läbi ajaloo

Kuna käesolev kursus käsitleb ainult relatsioonilisi andmemudeleid ja relatsioonilisi andmebaase, siis ei vaadelda all pool järgneva ajaloolise ülevaate käigus NoSQL (Not Only SQL databases) arenguga seonduvat vaid keskendutakse ainult relatsiooniliste andmemudelite ja andmebaaside arenguloole. Samas, kuna NoSQL andmebaasid tekkisid SQL (so relatsiooniliste) andmebaaside arengu tulemusena, siis võib kõike seda vaadata ka kui eellugu NoSQL baaside arendamise algusele. Samas ei saa öelda, et NoSQL andmebaasid oleks JÄRGMINE arenguetapp. NoSQL andmebaaside areng toimub paralleelselt SQL andmebaaside arenguga. SQL ja NoSQL andmebaaside kasutusvaldkonnad on täiesti erinevad.

## 1.1. Kaasaaegsete andmete modelleerimise ja andmebaasisüsteemide eelne periood

Uue uurimis- ja rakendusvaldkonna tekkimine ei toimu iseenesest, vaid selle tekkimisele eelneb mitmete teiste valdkondade areng, mis lõpuks kanaliseerub uue ainevalla tekkimisega. Osa nendest eeldusi loovatest ainevaldadest võib käsitleda kui tekkinud uue ainevalla eelkäijaid osa aga kui selle uue ainevalla tekkimise põhjustajaid. Seejuures enamikel juhtudel ei ole uue ainevalla tekkimisel esialgu tegemist millegi kvalitatiivselt täiesti uuega, vaid esialgu rakendatakse vanu teadmisi lihtsalt mingil uuel tasemel.

Samuti on ka andmete modelleerimise ja andmebaasisüsteemide ainevallaga. Kaasaaegse andmete modelleerimise eelkäijaks võib lugeda erinevaid ankeetide ja kartoteekide koostamise meetodikaid, samas kui kaasaaegsete andmebaasisüsteemide eelkäijaks on ankeetide kogumid ja kartoteegid. Nende teooriate ja rakenduste suurimad arendajad ja kasutajad on olnud luureorganid ja kriminalistika asutused. Siinkohal on sobiv tuua ka mõned näited.

1881 aastal alustati Prantsusmaal Alphonse Bertilloni juhtimisel kurjategijate registreerimist ja kartoteekimist antropomeetrilisel meetodil, kus andmete süstematiseerimise aluseks oli kurjategijate kehaosade mõõtmised. Koos registri loomise ja pidamisega käis pidevalt andmete süstematiseerimise ja andmeotsingu meetodikate arendustöö. Kui registri arendamise algusaegadel piirduti inimese identifitseerimiseks 14 kehaosa mõõdetega siis aja jooksul lisandusid sinna foto, isiku struktureeritud kirjeldus ja lõpuks isegi mõne etteantud sõrmede jäljed. Andmete keerukamaks muutumisel muutusid keerukamaks ka otsingumeetodid – lisaks mõõtmistulemustele pidi olema lõpuks võimalik

**andmete modelleerimise ja andmebaasisüsteemide tekkimine**

**ankeetide ja kartoteekide koostamise meetodikad**

**ankeetide kogumid ja kartoteegid**

**1881 Alphonse Bertillion antropomeetria**

leida kartoteegist isiku kaarti ka kirjelduse üksikute elementide järgi ja ka sõrmejälgede järgi. 1914-ks aastaks, kui antropomeetria asendati Prantsusmaal daktüloskoopiaga ja kogu kartoteegi struktuur läks ümberkorraldamisele, oli kartoteegis sadu tuhandeid kaarte.

USA-s loodi 1930. aastal FBI juurde isikusamastamise register, mis põhines daktüloskoopial. Selle registri loomise juures on tähelepanuväärne see, et registrit ei loodud tühjalt kohalt, vaid sinna koondati alates 1911-st aastast osariikide ja linnade politseijaoskondades olevad kartoteegid. Üle toomise ja liitmise käigus tehti ära suur töö väikestes registrites olevate andmete struktuuride ühtlustamiseks, andmete süstematiseerimiseks ja suurt andmehulka toetava otsingusüsteemi väljatöötamiseks. Kui registri loomise algusaegadel toimus kogu töö käsitsi, siis aastaks 1956, millal registri maht oli juba ca 14 miljoni kaardini (ca 140 miljonit sõrmejälge) toetas kaartide kartoteeki paigutamist ja sealt nende leidmist juba arvutisüsteem. Kuna arvutid ei olnud veel oma võimalustelt sellised, mis võimaldaks kogu informatsiooni salvestada elektrooniliselt, siis oli andmete adresseerimissüsteem lahutatud andmetest enestest – arvudeks kodeeritud tunnuste alusel leidis arvuti kartoteegikaardi aadressi kartoteegis.

Toodud näited ei kirjelda muidugi esimesi registreid või kartoteeke, kuid need näitavad ilmekalt seda, milliste rakendusteni oli maailm jõudnud enne seda kui hakkasid arenema kaasaegne andmete modelleerimine ja andmebaasisüsteemid.

Veel tänapäevalgi kasutatakse toleaegset kartoteekide “lähedasi sugulasi”. Nii on näiteks ka Eestis ilmselt igas väikeses raamatukogus kartoteegikaartidel põhinev raamatute otsingusüsteem, mille selliseks kujunemine millisenä me seda täna teame, on pika arendus- ja süstematiseerimistöö tulemus.

## **1.2. Kaasaegse andmete modelleerimise ja andmebaasisüsteemide tekkimise eeldused**

Elektrooniliste andmebaaside (esialgu küll kartoteekide) loomisele andsid peamise tõuke kolm asja (antud järjekorras) – kaasaegse struktuuriga arvuti loomine, aine magnetilistel omadustel baseeruva püsimalu loomine ja andmete lugemise-kirjutamise otsepöördus (mitte-järjestik) süsteemide loomine. Siin ei saa muidugi tähtsa katalüsaatorina jätta mainimata seda,

**1911 USA osariikide  
politsei kartoteegid**

**1930 FBI  
Isikusamastamise register**

**1956 14 milj. kaarti selles  
registris  
üksteisest lahutatud  
adresseerimissüsteem ja  
andmed**

**väikeste  
raamatukogude kaart-  
kartoteegid**

**kaasaegne arvuti**

**magnetiline püsimalu**

et selleks ajaks oli loodud hulk strateegilise tähtsusega kartoteeke, mille pidamine vanal tehnilisel ja struktuursel platvormil hakkas üle jõu käima.

Kaasaaegse arvuti esimese põlvkonna (1945-1956) tekkimisel oli suur osakaal teisel maailmasõjal, kus riigid püüdsid arvutite abil saavutada strateegilist üleolekut – sõja ajal tehtud pingutused kanaliseerusid vahetult pärast sõda uue põlvkonna arvutite loomisega, milles rakendatud põhimõtted määrasid arvutite arendamise trendid järgmiseks neljakümneks aastaks (EDVAC - University of Pennsylvania 1945, EDSAC - Cambridge University 1949, UNIVAC I - Remington Rand, 1951). Peamisi põhimõttelisi muutusi oli kaks. Esiteks kirjeldati arvuti arhitektuuris keskujuhtseadme (protsessori) mõiste, mis võimaldas arvutit juhtida läbi ühe andmevoo. Teiseks hakati nii täidetavat programmi kui ka programmi juhtimiseks või töötlemiseks vajalikke andmeid hoidma samas mälus. See kõik lõi aluse kaubanduslikuks arvutite tootmiseks.

1945. aastal loodi uus andmekandja, magnetlint, mis vähehaaval hakkas välja vahetama perfokaarte ja perfolinte. See oli esimene andmekandja, mis võimaldas andmete otsimist. Siiski ei olnud see otsingumehhanism veel eriti täiuslik, kuna võimalik oli ainult järjestik-otsing. Suure tähtsusega oli aga see, et “ruumiühikusse salvestatud andmete hulk” kasvas mõõtnatult ja andmete fragmenteerimisega paljudele paralleelselt töös olevatele lintidele suudeti tagada ka juba piisav operatiivsus andmete otsimisel.

Esimeste kaasaaegsete andmebaasisüsteemide loomiseni oli aega veel umbes paar aastat, kui William. C. McGee 1959. aastal publitseeris oma artikli “*Generalization: Key to Successful Electronic Data Processing*” ajakirjas Journal of the ACM (Volume 6, Number 1, January 1959 lk. 1-23, ACM – Association for Computing Machinery). Artiklis ei anta küll veel konkreetseid andmete üldistamise põhimõtteid vaid piirduakse üldkontseptuaalse filosoofiaga, kuid see on esimene suunda näitav teeviit kaasaaegse andmete modelleerimise põhimõtete poole.

Samal aastal võtab IBM kasutusele oma magnetkettasüsteemi Ramac (Random Access Method of Accounting and Control) mudel 305, mis on maailma esimene magnetkettasüsteem ja mis koosneb 50-st umbes 60 cm diameetriga kettast, millede mõlemale küljele saab salvestada informatsiooni. Informatsiooni salvestamise tiheduseks on 2000 bitti

**otsepöördus andmete lugemisel / kirjutamisel**

**II maailmasõda**

**1945 EDVAC**

**1949 EDSAC**

**1951 UNIVAC**

**protsessor**

**dünaamiline mälu**

**1945 magnetlint**

**1959 William C. McGee  
esimene artikkel andmete  
modelleerimisest**

**1959 IBM Ramac model 305**

ruuttollil kogumahuga 5MB. Olulisimaks murranguks on siin siiski andmete otsepöördus lugemis/kirjutusrežiimi esmakordne rakendamine ja andmete kirjutamise/lugemise suur kiirus.

### **1.3. Kaasaaegse andmete modelleerimise ja andmebaasisüsteemide tekkimine**

Aastal 1961 töötati korporatsioonis General Electric Co. välja andmebaasi juhtimissüsteem IDS (Integrated Data Store), mida loetakse esimeseks elektrooniliseks andmebaasi juhtimise süsteemiks. Projekti juht oli Charles Bachman. Siin ei saa muidugi rääkida veel andmebaasi juhtimissüsteemist tänapäevases tähenduses, kuna enamik andmebaasi funktsioone oli käsitsi kodeeritud, andmebaasina oli käsitletav ainult üks fail ja ta töötas ainult General Electric Co. arvutitel ja lahendas ainult selle firma konkreetseid vajadusi. Sellest sai tõuke CODASYL (Conference on Data Systems Languages) grupi kokkukutsumine, mis koosnes vabatahtlikest ja mille eesmärgiks oli andmesüsteemide efektiivsema analüüsi, disaini ja rakendamise vahendite ja meetodikate väljatöötamine. Grupp moodustati 1959 aastal ja töötas kuni 1985-nda aastani. Grupi peamiseks ülesandeks määratleti standardse, erinevatel arvutitel kasutatava programmeerimiskeele loomine. Selleks programmeerimiskeeleks sai COBOL ja selle raames formuleeriti ka võrk-andmemudeli põhilised kontseptsioonid.

1968 aastal tuli IBM välja oma IMS (Information Management System) kontseptsiooniga mis formuleeris hierarhiliste andmemudelite põhialused. Ja seejärel kohe (1969) sama mudeli täiendusega (IDM DB/DC), mis kirjeldas meetodit võrk-vaadete ehitamiseks hierarhilisele andmemudelile. Mõlemad lahendused olid mõeldud kasutamiseks IBM SYSTEM/360 *mainframe*'del.

Kuni siiani olid andmebaasid kõik ühe protsessi poolt kasutatavad. Eelmise sajandi 70-ndate aastate lõpus lõi IBM koos American Airlines'ga süsteemi SABRE, kus läbi kommunikatsioonivõrgu said andmetele korraga ligi juba paljud kasutajad.

Vaatamata suhteliselt tormilisele erinevate andmebaaside loomisele eelmise sajandi 70-ndate aastate lõpus, ei olnud ikka veel tekkinud kommertsiaalsetel alustel müüdavat, ristvarast eraldiseisvat andmebaasisüsteemi. Sellele aluse panemiseks oli jällegi suur teene

**1961 General Electric Co.  
IDS (Integrated Data Store)**

**Charles Bachman**

**1959-1985 CODASYL group**

**COBOL  
võrk-andmemudelid**

**1968 IBM - IMS**

**1969 IBM - IDM DB/DC**

**1978 IBM**

**SABRE  
kommunikatsioonivõrk**

**1970 Edgar F. Codd  
relatsiooniline mudel**

firmal IBM. IBM uurija Edgar F. Codd pakkus 1970 aastal välja relatsioonilise andmebaasimudeli, kus andmeid säilitatakse tabelites mille vahele ehitatakse relatsioonid. IMS mudelit täiendati relatsioonilise andmemudeli printsiipidega ja selle alusel arendati välja andmebaasisüsteem SYSTEM/R, mida müüdi koos IBM *mainframe*'dega kuni 1980-nda aastani. IBM süsteemi SYSTEM/R kohta avaldatud informatsiooni võtsid oma uurimis- ja arendustöodes aluseks California Ülikooli teadlased Michael Stonebraker ja Eugene Wong ning lõid nende arenduste tulemusena oma andmebaasisüsteemi, millele andsid nimeks Ingres ja millel olid kõik kommertsialiseerimiseks vajalikud olulised omadused. See produkt kommertsialiseeriti lõpuks firmade Oracle Corp. ja Ingres Corp. poolt.

**1977 IBM - SYSTEM/R**

**~1980 Ingres**

Eelmise sajandi 70-ndate keskel toimus veel üks andmete modelleerimise tähtsündmusi, millest ei saa mööda vaadata. 1976. aastal esitles Peter P. Chen olemi-suhte diagrammi, mis on kaasaajal üks kõige levinumaid andmete modelleerimise meetodikaid. Huvitav on siinkohal Chen'i väide, et selle teooria kirjapanemiseks oli ta kogu elu, kuni selle hetkeni kogu aeg õigel ajal õiges kohas ja tal jäi ainult elu enese poolt ettelükatud meetoodika kirja panna.

**1976 Peter P. Chen - ERD  
(olemi-suhte diagramm)**

Aastast 1979 tasub ära märkida veel sündmus, mis märkis ühe tõusva produkti ja trendi sünni. 70.ndate aastate jooksul töötati välja mitmeid erinevaid päringukeeli – SQUARE, SEQUEL, QBE, QEL jne. 1979. aastal tõi Oracle turule esimese kommertsiaalse andmebaasisüsteemi, mis kasutas andmemanipuleerimiskeelena SQL-keelt (Oracle Corp. teisend standardist SEQUEL). Tõusvaks produktiks sai Oracle DBMS ja tõusvaks trendiks SQL-keel.

**1979 Oracle corp.  
Oracle DBMS ja SQL-keel**

60-ndate aastate lõpust hakkas arenema veel üks süsteemide grupp, mis tänapäeval on suurte andmebaaside kasutamise lahutamise koostisosa. Esialgu nimetati neid otsuse toetussüsteemideks (DSS – Decision Supporting System).ja nende esmane eesmärk oli andmekäsitluse lihtsustamine ja parem kasutamine otsuste tegemise toetamisel. Sisuliselt olid tegemist andmete käsitlemise analüütiliste vahendite arendustega. Selliseid alamsüsteeme loodi küll kogu eelmise sajandi 70-ndate aastate jooksul kuid esimese kommertsiaalse lahenduseni jõuti alles 1970 aastal. Selleks oli süsteem EXPRESS.

**1970 DDS EXPRESS**

## 1.4. Arendused 1980-ndatel aastatel

Eelmise sajandi 80-ndate aastate alguseks oli põhiline platvorm andmete modelleerimise ja andmebaasisüsteemide arenemiseks loodud ja tundus, et mingisuguseid erilisi lööke ei edasi ega tagasi ei tohiks tulla. Selle situatsiooni lõi aga segamine personaalarvutite turule tulek. Üpris pea loodi esimene personaalarvutile mõeldud relatsiooniline andmebaasisüsteem DBase. Kiiresti järgnesid DBase II, Paradox, Fox, FoxPro, DBase III, Dbase IV jne. See muutis olukorda pöördeliselt – andmebaasid said kättesaadavaks suurele kasutajate hulgale ja andmete modelleerimise meetodikad ja vahendid vajalikuks paljudele. Arendama asuti järjest uusi ja mugavamaid kasutajaliideseid.

Mõne aja pärast lisandusid lokaalvõrkude loomise vahendid – riistvara ja tarkvara. Oma hüppe tegid kaasa ka andmebaasisüsteemid – kõigile elujõulistele andmebaasisüsteemidele lisati andmete mitme kasutaja poolt samaaegse kasutamise omadused. See ei osutunudki aga nii lihtsaks kui esialgu tundus – hakkasid tekkima konfliktsituatsioonid andmete samaaegsete kasutajate vahel ja see tingis andmete ühiskasutuse ja lukustusteooriate arenemise.

1981 aastal tuli James Martin juhtimisel töötanud uurimissgrupp (JMA – James Martin Associates) välja infosüsteemide loomise meetodika IE (Information Engineering) esimese versiooniga. Lisaks J. Martinile kuulusid gruppi veel Clive Finkelstein ja Tong Carter. IE esimeses versioonis keskenduti andmeanalüüsile (infooloogilised skeemid, normaliseerimine, andmetele juurdepääsu-analüüs) ja andmevoo diagrammidele. Aasta hiljem tuldi välja tervikliku IE meetodikaga. Selles meetodikas sisaldas ka andmete seos-maatriks, mis sellest tehnoloogiast lahutatuna on ka iseseisvat rakendust leidnud. 1984 aastal liitus grupiga Ian MacDonald töötati välja meetodikad andmeanalüüsi, andmevoogude diagrammide ja objekt-modelleerimise ühendamiseks. Tehti ka esimesed arendused analüüsi ja modelleerimist toetava tarkvara loomiseks.

1985-ndal aastal publitseeriti SQL-keele esialgne standard – keel, mille Oracle Corp. oli kasutusele võtnud kui oma andmebaasisüsteemi andmemanipuleerimiskeele, oli vahepeal läbi teinud standardiseerimise protsessi ja valmis kasutamiseks suuremas hulgas produktides. Praeguseks on toona kinnitatud standardit mitmeid kordi muudetud ja ANSI klassifikatsiooni järgi on viimane standard SQL99, mida arvestavad

**1980 Ashton Date - DBase  
DBase II,  
Paradox  
Fox, Foxpro  
Dbase III, DBase IV**

**~1982 LAN  
andmete ühiskasutus**

**lukustusmeetodite areng**

**1981 J. Martin & JMA  
Inforamtion Engineering (IE)**

**1984-... esimesed ERD Case-  
süsteemid**

**1985 esimene SQL-keele  
standard**

oma uutes versioonides kõik tuntumad andmebaasisüsteemid.

1985 aasta oli nii mõneski mõttes huvitav aasta Lisaks SQL-keele esmase standardi esitlemisele rakendati samal aastal ka esimene äri-intelligentsi (business intelligence) süsteem. Firma Metaphor Computer Systems Inc. valmistas Procter & Gamble Co. jaoks süsteemi, mis ühendas analüüsiks müügiinformatsiooni ja turu jälgimise informatsiooni. Samal aastal alustas Pilot Software Inc. süsteemi Command Center müüki – turule oli tulnud esimene Client/Server arhitektuuriga süsteem. Seoses sellega hakkasid ka kohe arenema eraldatud (distributed) struktuuriga andmebaaside teooriad ja õigepea ka andmete replikeerimist käsitlevad teooriad ja toetavad süsteemid. Paraku ei ole lõpuni korrektselt toimivate ja kõiki kasutamisevajadusi rahuldavate replikeerimissüsteemide loomiseni jõutud veel tänasel päevalgi.

Samuti samal aastal muudeti Ingres projekt Postgre projektiks, eesmärgiga luua objekt-orienteeritud andmebaasisüsteem. Siiski ei saanud Postgre mitte esimeseks objektorienteeritud andmebaasisüsteemiks. Selle au näppas ära 1986 aastal firma Graphael Inc., kui ta tuli turule andmebaasisüsteemiga Gbase.

Aastal 1988 defineerisid firma IBM uurijad Barry Devlin ja Paul Murphy mõiste andmeladu (*information warehouse*) ja sellest ajast alates algas andmeladude ehitamine. Aastal 1991, avaldas W. H. Inmon raamatu "*Building the Data Warehouse*". Sellest alates arvatakse andmeladude praktilise teoreetilise arengu alguseks. Selles raamatus esitati peamised printsiibid andmelattu andmete kollektioneerimise, andmete selekteerimise ja valimite koostamise tehnoloogiate kohta.

80-ndate aastate trendidest on mainimata veel ekspertsüsteemid – teadmiste süsteemid, mis baseeruvad tehisintellektil (*artificial intelligence*) ja on suunatud vajaliku informatsiooni leidmisele, süstematiseerimisele ja analüüsile üldjuhul suhteliselt kitsas ainevallas. Ekspertsüsteemide loomisel on kaks sihifunktsiooni – inimese intelligentsi uurimine luues teadmiste baase ja nende käsitlemise mehhanisme ja praktiliste rakenduste loomine erinevate valdkondade töötajate teabevajaduste rahuldamiseks. Andmebaasisüsteemide areng ja seega ka informatsiooni talletamise ning analüüsimise vahendite kasutatavuse

**1985 - Business Intelligence**

**1985 - klient / teenindaja  
(Client / Server) arhitektuur**

**1985 Ingres -> Postgre**

**1986 objektorienteeritud  
andmebaaside arengu algus**

**1988 Barry Devlin  
andmelad**

**1980-... ekspertsüsteemid  
tehisintellekti rakenduste esile  
tõusmine**

paranemisega tekkis soodne pinnas seda tüüpi teooriate arenemiseks ja rakenduste loomiseks.

80-ndate teisel poolel tuleb turule uus liik andmete modelleerimist toetavaid süsteeme - CASE-süsteemid. Siin on peamiselt kolm põhjust, miks see just sellel ajal juhtub. Esiteks on modelleeritavad andmestruktuurid läinud nii suureks, et nende "käsitsi modelleerimine" on muutunud pea võimatuks. Teiseks tunduvalt muutunud, tänu paremaks muutunud arendusvahenditele, infosüsteemide dünaamika ja andmenudeleid on vaja pidevalt ringi teha, samas tagades sünkroonselt adekvaatse dokumentatsiooni üha suuremaks muutuvates arendustiimides. Kolmas ja mitte vähe oluline põhjus on see, et graafilised info ehituse tehnikad on arenenud piisavalt kaugele selleks, et luua graafilisel andmestruktuuride käsitlusel põhinevaid arendusvahendeid.

Esimesed andmete modelleerimise CASE-süsteemid luuakse traditsioonilistesse graafilisse töökeskkonda, milleks on Macintosh oma MacOs operatsioonisüsteemiga. Enamikus toetavad nad JMA grupi poolt loodud IE metoodikat. Üks tuntumaid selliseid vahendeid oli CASE-süsteem DEFT, mis võimaldas luua ERD-mudeleid (Entity Relationship Model) ja DFD-mudeleid (Data Flow Diagram), kommenteerida neid erinevatel tasemetel ja trükkida välja dokumentatsiooni. Üsna pea, koos MS Windows' operatsioonisüsteemi arenguga tulid vastavad arendused ka PC maailma. Tuntumateks kaubamärkideks olid ERWin, PBWin (Logic Works), System Architect (Popkin Software) ja SilverRun (Magna Solutions). Nende turule tulemise hetkel oli kõikidel andmete modelleerimise CASE-süsteemidel oluliselt paranenud dokumentatsiooni genereerimise võimalused. Lisaks sellele oli lisandunud andmebaaside automaatgenereerimise ja muudatuste sünkroniseerimise omadused.

## 1.5. Arendused 1990-ndatel aastatel

1990-ndate aastate arengud on kõik suuremal või vähemal määral seotud interneti, kommunikatsiooni, töökindluse ja multimeediaga.

Interneti arenemine pööras tagasi mõned leheküljed andmekäsitluse ajaloos. SQL-päringukeele kõrvale tõusis taas kirje-taseme pöördus (RLA – row level access), kui SQL-keelset tunduvalt ebamugavam kuid samal ajal tunduvalt kiirem andmekäsitluse meetod. See omakorda on upitanud

**1985-... ERD Case süsteemide leviku algus**

**Andmemudelite Case süsteemide massiline levik:**

DEFT

ERWin, PBWin  
System Architect  
SilverRun

**1990- ... Internet, turvalisus töökindlus, multimeedia**

**SQL-keel vs. RLA**



tõusuteele objektorienteeritud andmebaasid, mille kasutus kuidagi edeneda ei tahtnud, kuna siin on meetodite taha vastavalt vajadusele võimalik maskida just kasutuskeskkonna ja viisi poolt määratud pöördusmeetodid. Kuigi need kasvud ei ole sellised mida tegelikult loota võiks.

Kasutajate hulga ja töödeldava informatsiooni mahu pidevast kasvust tingituna on oluliseks suunaks ka mitme protsessoriga arvutite tulek turule ja sellest tingituna ka andmebaasisüsteemide arenemine kasutamaks seda jõudlust. See nõuab omakorda täiesti uute optimeerimismetoodikate väljatöötamist ja rakendamist. Kogu 90-ndaid kattev suur trend ongi SQL päringute uute optimeerimismeetodite väljatöötamine ja rakendamine.

Kommunikatsioonitehnika arenemisest tulenevalt kasvab 90-ndatel mitmete erinevate andmebaaside ja isegi erinevate andmebaasisüsteemide abil lahendatud andmebaaside kooskasutused. Arenevad replikeerimistehnikad, andmete distantvarundus, süsteemide distantdiagnoosika jms.

Kuni siiani võib tunduda, et eelmise sajandi 90-ndad aastad ei andnudki andmete modelleerimise ja andmebaasisüsteemide arengusse midagi kardinaalselt uut, tegeldi vaid juba olemasolevate teooriate ja tarkvarakomponentide täiustamisega. Tegelikult tuli juurde vähemalt kolm uut asja – UML-süsteemid (Unified Modeling Language), spetsiaalsed andmebaasisüsteemid andmeladude jaoks ja andmete kaevandamine.

Spetsiaalsed andmelaole orienteeritud andmebaasisüsteemide vajadus ilmnes eelmise sajandi 90-ndate aastate teisel poolel kui andmelaondusega oli tegeletud juba piisavalt kaua selleks, et andmeladude andmemahud oleksid piisavalt suured probleemide tekitamiseks. Selgus et olemasolevate andmebaasisüsteemide füüsilised struktuurid ja päringute optimeerimise meetodid ei sobi nii suurte andmemahude käsitlemiseks ja küll kõigi reeglite järgi “kokku kuhjatud” andmetest ei ole mingit kasu, kuna puudusid vahendid, mis aktsepteeritava ajaga suudaksid analüüsida andmeid ja esitada analüüsi tulemused soovitud kujul. Seepärast töötati välja täiesti uue sisemise füüsilise arhitektuuriga süsteemid koos uute päringu optimeerimise meetoditega mis on orienteeritud andmete lugemisele andmebaasist.

**Andmekäsitluse uued  
optimeerimise meetodid**

**eraldatud andmebaasid  
replikeerimistehnikad  
andmete distant-varundus  
distantdiagnoosika**

**UML-süsteemide teke  
andmeladude teke  
andmete kaevandamine**

**spetsiaalsed andmebaasi  
süsteemid andmeladude  
 jaoks**

**Sybase IQ**

**Oracle DB Data Warehouse**

Lisaks sellele piirati ka oluliselt andmete lisamise ja andmete kustutamise võimalusi. Uuendamise operatsioon puudub nendes süsteemides üldse. Tuntumad andmelao andmebaasisüsteemid, mis tollel ajaperioodil välja töötati olid Sybase IQ ja Oracle DB Data Warehousing.

Andmete kaevandamise all mõistetakse sellist tegevust mis on suunatud suure hulga andmete automaatsele või poolautomaatsele uurimisele ja analüüsile selleks et leida tähendust omavaid mustreid, mudeleid ja reegleid. See andmeanalüüsi suund on otseselt tingitud andmeladude arengust, kuna sellisel määral andmeid, mille alusel andmekaevandamist saab teha on tavaliselt ainult andmeladudes. Selle teooria peamised arendused liideti erinevate äri-intelligentsi toodetega.

Suure muudatuse modelleerimisse tõi UML-süsteemide tulek. Suures osas on nad edasiarendused eelmise sajandi 80-ndatel aastatel loodud ERD- ja DFD-mudelite joonistamise CASE-vahenditest. Siiski oli siingi uustulnukaid ja paraku on just nemad olnud innovatiivsemad ja ka edukamad. UML-süsteem moodustab tervikliku kogumi erinevaid kirjeldus-elemente, mille abil esitatakse infosüsteemi spetsifikatsioon ja millest hiljem on võimalik genereerida suur osa infosüsteemist. UML-süsteemides sisalduvad ka andmete modelleerimise vahendid, mis on küll tihedalt seotud kõikvõimalike muude kirjeldustega (andmevoo-diagrammid, oleku-diagrammid, struktuuri-diagrammid, protsessivoo-diagrammid, kasutuslood jne.). Tuntumad UML-süsteemid on firma Select Inc. Poolt loodud süsteem SELECT ja firma Rational poolt loodud toodete pere. Viimase poolt välja töötatud RUP-metoodika (Rational Unified Process) analüüsi läbiviimiseks, infosüsteemi spetsifitseerimiseks ja dokumenteerimiseks kujunes teatavates arendusvaldkondades *de facto* standardiks.

## 1.6. Areng 21. sajandil

21-sajandi algusaastate jooksul ei ole tekkinud juurde uusi suundi andmete modelleerimise ja andmebaasisüsteemide alal. Samas on jõudsalt edasi arenenud mitmed juba varem alguse saanud suunad. Enamik arenguid on juhitud interneti, kommunikatsioonitehnoloogia, ning nendest tulenevast plahvatuslikult suurenenud kasutajate arvust ja infotöötuse hajumisest ühest keskusest paljudesse erinevatesse punktidesse.

**andmete kaevandamine**

**mustrite otsimine**

**UML-süsteemide levik**

**RUP standardi teke**

**ei midagi uut - vaid olemas oleva edasi arendamine**

### 1.6.1. Andmete modelleerimise vahendite areng

Andmete modelleerimise meetodikate ja tehnikate areng ei ole olnud eriti aktiivne ja on väljendunud peamiselt juba varem turul olnud CASE- ja UML-süsteemide uusi omadusi sisaldavate uute versioonide turule tulemisega. Täiustatud on analüüsi/spetsifitseerimis-vahendite erinevate mudelite vaheliste seoste kirjeldusi, grupitöö organiseerimise vahendeid (seal hulgas versioonihaldust ja sisu haldust) ja mudelite testimise vahendeid. Kogu muutuste suund on dünaamilisuse poole, kus iga kasutajate grupp saab kirjeldada metamudelina just endale vajaliku struktuuriga modelleerimiskeskonna ja siis teha sellega tööd maksimaalse efektiivsusega.

**UML- ja CASE - süsteemide uued omadused grupitöö vahendid versioonihaldus mudelite testimise vahendid**

### 1.6.2. Andmebaasisüsteemide arendused

Andmebaasisüsteemide arendamine on olnud tunduvalt aktiivsem. Seda tingituna nii tugevast konkurentsist kui ka turu ja muu infotöötlusmaailma arengu poolt esitatavatele nõuetest. Vaatame toimuvaid arenguid gruppide kaupa

#### 1.6.2.1. Andmete kaitse

Koos interneti ja interneti kaudu kasutatavate rakenduste laialdase levikuga on kiireid ja radikaalseid arendusi nõudnud andmete kaitse. Siin võib vaadata kolme arendussuunda – andmetele juurdepääsu reguleerimine, baasisolevate salastamine ja andmete klassifitseerimine turvaseme järgi.

Andmetele juurdepääsu reguleerimisel on praegu arvestavaim meetod PKI (Public Key Infrastructure) sertifikaadil X.509, versioon 3 (interneti autoriseerimis standard) üle TLS-i (Transport Layer Security) põhinev autoriseerimismeetod.

Andmete salastamisel on juhtivaimaks lahenduseks andmete valikuline krüpteerimine (Selective Data Encryption) . See tagab tundlike andmete salastatuse ka juhul, kui peaks toimuma andmete autoriseerimata lugemine. Siin rakendatavate printsiipide abil püütakse optimeerida andmete krüpteerimise mahtusid - krüpteeritavateks määratakse ainult tundlikke andmeid. Välja on töötatud krüpteerimisstandardid DES (Data Encryption Standard) ja 3DES (Triple-DES).

**1. andmebaasi klastrid koormuse jagamine**

**2. Online Data Evolution**

**3. andmete säilimise tagamine**

**andmete varundamise uued meetodid**

Suuremaks probleemiks andmete turvamisel on olnud andmete horisontaalsete kasutusõiguste kirjeldamine – ligipääsu piiramine vastavalt kasutajate õigustele andmebaasi tabelite erinevatele ridadele. Secure Data Sharing - tehnoloogia pakub lahendust selle probleemile. Siin on võimalik kirjed jagada turvatasemete vahel ja vastavalt kasutaja turvatasemele lubada juurdepääsu ühe või teise turvatasemega kirjetele.

### 1.6.2.2. Töökindlus ja käideldavus

Koos pidev-teenusena (24 tundi päevas, 7 päeva nädalas, 365 päeva aastas – 24-7-365) läbi interneti ja ka lokaalvõrgu rakendustena pakutavate virtuaalsete infoteenuste tulekuga on oluliselt kasvanud vajadus andmebaasisüsteemide töökindlusele. Infosüsteemi mistahes komponendi (ka andmebaasi) kõrge käideldavuse taseme märkimiseks on võetud kasutusele termin *high availability*. Selle mõiste alla käivad järgmised arendused.

Andmebaasi klastrid (Database Clusters) on andmebaasi käsitlemise meetodika, kus sama andmebaasiga töötab mitu omavahel ühendatud (interconnected) paralleelserverit (andmebaasimootorit, mis asuvad erinevates füüsilistes arvutites). Klaster-mudel on võimalik jagada klastris olevate serverite vahel koormust ja mõne serveri töötamast lakkamisel anda tema poolt tehtavad toimingud üle klastrisse ühendatud ülejäänud serverite vahel. Samuti on võimalik käivitada klastrisse uusi servereid ilma juba seal töötavaid serverid ümber startimata.

Pidevalt töötavate (24-7-365) andmebaasisüsteemide kasutamisel on olnud suureks probleemiks ka see, et andmebaasistruktuuride muutmiseks ja muudeks hooldustöödeks tuleb tavaliselt peatada kõigi kasutajate töö andmebaasiga (logoff). Uute *Online Data Evolution* meetodite rakendamisel saab muudatusi teha samal ajal kui toimub aktiivne töö andmebaasiga. See parandab oluliselt juurdepääsu andmetele paljudes missiooni-kriitilistes süsteemides .

Uued meetodid on kasutusele võetud andmete säilimise tagamiseks (Data Protection). Kui seniseid varundus-meetodeid (backup/restore) võib nimetada passiivseteks ja perioodilisteks, siis uusi tehnoloogiaid võib kirjeldada kui aktiivseid ja pidevaid. Siin ei toimu andmete varundamine enam range tsükliisusega vaid on rohkem asünkroonse

## 4. andmekäsitluse automaat-korreksioon

**Andmete käideldavus:**  
*(high availability)*

**1. andmebaasi klastrid**  
**koormuse jagamine**

**2. *Online Data Evolution***

**3. andmete**  
**säilimise tagamine**

**andmete varundamise uued**  
**meetodid**

iseloomuga ja protsessid käivituvad "ise" vastavalt mingite tingimuste täitumisele. Samuti on ka andmete taastamist oluliselt automatiseeritud. Andmete hävimisel on süsteemid ise suutelised analüüsima vea suurust ja taastama hävinud andmed.

Suur hulk andmebaasisüsteemide tööaja kasulikku ressursi kulub vigaste käsituslausete töötamisele. Selle vältimiseks on hakatud ehitama intelligentseid süsteeme, kus süsteemi loojad saavad kirjeldada andmebaasi käsitlemise korralduste veakorrektsiooni reeglid ja seega juba enne käsitluskorralduste andmebaasi mootorile täitmiseks saatmist parsimist teha korraldustele eelkorrektsiooni. Nende abil vabastatakse suur hulk seni veatöötamiseks kulutatud ressursi.

### **1.6.2.3. Transaktsioonikäsitlus (*Transaction Handling*)**

Üsna pikka aega kasutati andmete lukustamisel andmebaasi lehekülje (page locking) lukustamist. Selle lukustusmeetodi puhul lukustatakse koos kirje lukustamisega terve andmebaasi lehekülg. Sellise lukustusmeetodi põhiliseks miinuseks on see, et koos mingi tuntud kirje lukustamisega lukustatakse teatud hulk tundmatuid kirjeid. Seda sellepärast, et kasutaja jaoks ei ole teada, millised kirjed on ühe andmebaasi lehekülje peal. Kasutaja jaoks on see sisuliselt juhuslik komplekt kirjeid. Selle meetodi kasutamine oli tingitud peamiselt tehnilistest võimalustest. Kui kasutajate hulk või andmebaasikasutamise intensiivsus on väiksed, siis ei ole selles meetodis midagi halba. Kuid koos interneti tulekuga kasvas nii andmebaaside kasutajate arv kui ka kasutamise intensiivsus hüppeliselt. Selle tulemusena muutus lehekülje lukustusmeetod kohmakas ja infosüsteemide tööd pärssivaks – "juhuslikult" ja üldsegi mitte vajalikult lukus olevate kirjete suur arv hakkas takistama andmetele juurdepääsu. Kuna vahepeal on arenenud ka tehnilised võimalused, siis on lehekülje lukustamist asunud välja vahetama kirje lukustamine (row level locking). Siin pannakse lukku töö poolest ainult need kirjed, mille lukustamiseks on põhjust. Lisaks sellele on hakatud kasutama ka andmete nn. "räpase lugemise" (dirty read) meetodikat, kus kui kasutaja kes on aktsepteerinud sellise lugemise taseme saab lugeda ka lukus kirjeid, kuid ainult kirje seda kujutist, mis viimasel oli enne kirje lukustamist. Arendatud on ka kombinatoorseid lukustusmeetodikaid, kus X-, S- ja U-lukkude kombineerimisel "räpase lugemise" meetodikaga saadakse päris häid tulemusi andmete

#### **4. andmekäsitluse automaatkorrektsioon**

#### **AB lehekülje lukustamine**

#### **AB kirjete lukustamine räpased lugemised**

#### **X-, S- ja U-lukud**

kasutamise konkurentsuse vähendamisel.

Vaated (Views) on andmebaasisüsteemide koosseisudes eksisteerinud juba paarkümmend aastat. Andmebaasi jõudluse tõstmise vajadusest tingituna on tehtud muutusi siingi. On välja arendatud materialiseeritud vaadete (materialized views) kontseptsioon. Kui siiani oli vaade defineeritav kui "virtuaalne tabel", mis moodustati päringu täitmise ajaks, siis materialiseeritud vaadet hoitakse füüsiliselt andmebaasis ja muudetakse kui muutub mõni vaadet moodustavatest komponentidest. See võimaldab hoida oluliselt kokku kallist aega, mis läheb iga kord "kaduma" virtuaalse vaate moodustamisel. Seda meetodikat on mõtet kasutada suure ridade arvuga ja tihti kasutatavate vaadete korral.

Nii kaua kui on olnud kasutusel SQL-keel, on olnud andmebaasisüsteemide arendamist läbivaks probleemiks päringute optimeerimine. Cartesia produkti (Cartesian Product) elimineerimine optimeerimisprotsessist on jäänud aegade taha, kuid ka täna leidub palju parandamist SQL-korralduste täituralgortimide optimeerimise juures. Oluliseks on muutunud teooriad, mis võimaldavad suhteliselt kiiresti elimineerida optimeerimisalgoritmide ebaefektiivsed harud. Optimeerimisalgoritmide pidev parendamine on oluline, kuna pahatihti võtab SQL-lausetate optimeerimine tunduvalt rohkem aega kui ebaefektiivselt optimeeritud lausetate täitmine.

#### 1.6.2.4. Sisu haldus (Content management)

Täistekst-indekseerimine ja täistekst-otsing ei ole üldse enam uued leiutised. Siiski on ka siin tehtud ära märkimisväärne arendustöö. Selle on tinginud üha suuremate koguste dokumentide salvestamine andmebaasidesse ja vajadus neid leida dokumendi sisu alusel. Tavalisele täistekst otsingule ("fraas") on lisandunud uued otsingu strateegiad "täistekst: boolean" (*full-text boolean*), "täpne fraas" (*exact phrase*), "lähendus" (*proximity*), "lõigu otsimine" (*section searching*), väärkirjutised (*misspellings*), "tõkestamine" (*stemming*), "mask" (*wildcard*), "sõnaraamat" (*thesaurus*), "sõnade ekvivalentsus" (*word equivalence*) ja scoring.

Siiski on see just see koht kus peab natuke viitama NoSQL andmebaasidele – just see andmebaaside klass on tänapäeval sisuhalduse suuresti üle võtnud. Sisuhaldusülesannete puhul tuleb

materialiseeritud vaated

SQL-keele optimeerimis-  
algoritmide areng

täistekst indeksid

täistekst otsing

kasutada pigem NoSQL andmebaase kui et SQL andmebaase. Siiski on ka viimastega võimalik lahendada sisuhaldusülesandeid, kuigi see ei ole nii mugav ja efektiivne.

Omaette nähtuseks on andmestruktuurid, mida on hakatud tähistama mõistega "*Semi-structured databases*" (pool-struktureeritud andmebaasid). Tegelikult ei ole siin midagi uut vaid lihtsalt kahte vana ja hästi tuntud asja on hakatud kasutama uues kombinatsioonis ja saavutatud seega uus tulemus. Tihti ei soovita salvestada dokumente vahetult andmebaasi. Samas on aga oluline siduda neid dokumente omavahel ja teistegi andmetega. Sellisel juhul lahendatakse asi nii, et dokumentide päised salvestatakse andmebaasi ja nende dokumendipäiste kaudu moodustatakse ka kõik vajalikud seosed teise dokumentide ja muude objektidega. Dokumendid ise salvestatakse tavalisse failisüsteemi jagatud teekidesse (Shared Folders). Andmebaasi kirje päisesse kirjutatakse dokumendi URL. Probleemid millega siin peamiselt tegeletakse on seoste püsivus, seoste taastamine ja andmekaitse meetodikad.

Loomulikult ei ole mööda saadud ka XML-st - XML-jadade käsitlus on muutunud kõikide tuntud andmebaaside lahutamatuks koostisosaks. Lisaks sellele on tekkinud andmebaasid, kus nii andmete kirjeldusi kui andmeid endid hoitakse XML-kujul.

Oluliseks on muutumas kõikvõimalikud elektrooniliste maakaartide süsteemid ja sellel baseeruvad infosüsteemid (maakaardid, turismiinfo, maakatastrid, geoloogia süsteemid, ehitiste registrid jne.). Oluline on siin graafilise info kihilisus, graafilise informatsiooni tükeldamine ja graafilise informatsiooni sidumine andmevõrgustikuga. Ainus arvestatav tegija selles valdkonnas on Oracle corp. oma andmebaasisüsteemiga alates versioonist 8i. Need spetsiaalsed vahendid võimaldavad luua tavalistest geograafilistest infosüsteemidest (GIS) kuni traadita lokaliseerimissüsteemideni ja lokaliseerimist nõudvate E-kommertsit teenusteni. Seda probleemvaldkonda tähistatakse mõistega *location based services*.

**"pool-struktureeritud"  
andmed ja andmebaasid**

**XML**

**geoinfosüsteemid (GIS)**

**lokaliseerimissüsteemid**

### **1.6.2.5. Andmelaod ja äriintelligents**

Toimub hoogne äriintelligentsi süsteemide arendamine. Täiendatakse metamudeleid ja metamudelite ehitamise vahendeid. Viimased muutuvad järjest visuaalsemaks. Äriintelligentsi süsteemidega liidetakse ka andmekaevandamine ja tekkivad vahendid, mis võimaldavad andmekaevandamise tulemusi pool-automaatselt ja ka juba osaliselt automaatselt muuta äriintelligentsi kirjeldavateks metamudeliteks.

Kuna on aru saadud, et nii äri-intelligentsi kasutamise kui ka andmekaevandamise aluseks saab olla võimalikult lai ja korrektselt koostatud andmeladu, siis arenevad kiiresti edasi andmete kolleksioneerimise ja teisendamise süsteemid. Kuna kolleksioneeritavate andmete voogude hulk kasvab pidevalt ja nende töötlemine andmelao jaoks vajalikule kujule nõuab järjest rohkem ajalist ressursi, siis arenevad siingi meetodikad ja vahendid andmevoogude ja andmevoogude siseselt paralleeltöötamise suunas. Andmete teisendusmudelite kirjelduskeeled muutuvad järjest paindlikumaks ja visuaalsemaks.

### **1.6.2.6 Andmete integreerimine**

Hajusandmetöötluse üks suuremaid probleeme on üksteisest eemalolevate samatähenduslike ja sama struktuuriga või eritähenduslike kuid protseduurilist seost omavate andmebaaside vahelised andmete ülekanded. Esimesel juhul selleks, et hoida kahte või enam andmebaasi omavahel sünkroonis, teisel juhul selleks et liigutada vajalik hulk (koond)andmeid ühest andmebaasist teise. Selleks kasutatakse juba pikemat aega andmete replikeerimise meetodeid. Paraku ei ole veel siiani mistahes protsessi toetavat replikeerimissüsteemi loodud. Kui replikatsiooni vahendite arengu algusaastatel murti pead seose terviklikkusest (*referential integrity*) tingitud replikatsiooniprobleemide üle, siis tänapäeval on peamiseks probleemiks väga võimsaks muutunud transaktsioonide haldamise mehhanismid, mille tulemusena ei ole võimalik enam korrata teises serveris kõiki tegevusi just selles samas järjestuses kui seda tehti replikeeritavas serveris. Suurimaks puuduseks on siin replikeerimistegevuste korrektse toimimise järelkontrolli võimaluse puudumine.

Viimastel aegadel on paljud kohad, kus seni kasutati mingi keerukusega replikatsiooni asendatud teatapärase (MQ - Message Queue)

**metamudelite arendamine**

**andmekaevandamine**

**tagasiside**

**andmete kolleksioneerimine**

**andmete teisendamine**

**andmete paralleeltöötlus**

**modelleerimise visuaalsus**

**hajusandmetöötlus**

**andmete sünkroniseerimine**

**andmete edastamine**

**andmete replikeerimine**

**Message Queueing**



infovahetuse ja sünkroniseerimise mehhanismidega. Siin saadetakse erinevate süsteemide (ka andmebaasimootorite) vahel asünkroonseid teateid just sellisel hetkel kui see vajalikuks osutub. Teadetega koos liiguvad andmed ja tegevuskorraldused. Suureks plussiks replikatsioonisüsteemide ees on siin see, et MQ kaudu saadetud teadete järjekord ei muutu, teate adressaadile kohalejõudmine ja töötlemine tagatakse süsteemsete vahenditega. Seega puudub ka järelkontrolli vajadus. Suurt "võimsust" lisab sellele meetodile XML RPC (*Remote Procedure Call*) kasutamine.

Andmete integreerimiseks võib nimetada ka seost oleviku ja mineviku vahel. Väljaarendamisel on tehnoloogiad, mis võimaldaksid mineviku ajahetke etteandmisel näha andmebaasi just sellisena nagu ta tollel hetkel oli – sellisena nagu tollel hetkel olid nii struktuur kui ka andmed. Neid meetodikaid ja vahendeid tähistatakse mõistega "Minevikupilt" (Flashback). Sellele tehnoloogiale üritatakse veel liita automaatset SQL-lauset korrigeerimise mehhanismi, mis suudaks kehtiva andmebaasistruktuuri jaoks kirjutatud SQL-korraldused teisendada tollel ajahetke andmebaasi struktuurile vastavaks.

### 1.6.2.7 Andmebaaside haldamine

Kui eelmise sajandi 90-ndatel aastatel olid võtmesõna "ise-haldav" (self-managing) kasutatav vaid mõnede keskmiste ja väikeste andmebaasisüsteemide puhul (tuntuim Centura corp. andmebaasisüsteem SqlBase), siis praeguseks hetkeks on saanud see trendiks ka suurte andmebaasisüsteemide hulgas. Lisandunud on veel ka mõiste "ise häälestuv" (self-tuning) ". Ühelt poolt on see aja märk – kõike mida saab automaatseks teha, seda ka automaatseks tehakse. Teiselt poolt on aga suurte andmebaaside haldamine muutunud nii keeruliseks, et andmebaasiadministraatoritelt on vaja osa koormust maha võtta. Lisaks sellele on osadel puhkudel arvuti poolt väljakalkuleeritud lahend igal juhul täpsem kui inimese poolt arvatud lahendus. Suurt tööd on siin teinud jällegi andmebaasisüsteemide loojate lipulaev Oracle corp.

Uueks suunaks on andmebaasiserverite ressursside jaotamine ja prioriseerimine vastavalt äriprioriteetidele. Peamiselt vaadeldakse siin protsessorite ressursi jagamist erinevate kasutajate, kasutajagruppide ja protsesside vahel, määrates ka paralleelprotsessorite kasutamise

XML RPC

Flashback

"ise-haldavad"  
andmebaasisüsteemid

"ise-häälestuvad"  
andmebaasisüsteemid

ressursside prioriseerimine

tasemed. Oluline on siinkohal nende protsesside ressursinõudluse automaatne mahasurumine, mis “tuntakse ära” kui “nii kui nii pikalt töötavad protsessid” ja nende arvelt süsteemi üldise käideldavuse tõstmine.

Jõudsasti arenevad andmebaasisüsteemide distantstdiagnostika ja distantshalduse vahendid.

**andmebaaside diagnostika ja distantshaldus**