

Objektorienteeritud (OO), XML, NoSQL ja NewSQL andmebaasisüsteemid

Teema 9



Rakendustest

- ♦ **Registripõhised infosüsteemid** vajavad **aknapõhiseid andmebaasirakendusi** (*window on data applications*).
- ♦ Kui on ka palju teisi rakendusi, mis vajavad tööks andmeid ja seega mingil kujul andmebaasi/andmebaase.

Keerukate rakenduste näited

- ♦ *Computer Aided Design* (CAD)
- ♦ *Computer Aided Manufacturing* (CAM)
- ♦ *Computer Aided Software Engineering* (CASE)
- ♦ *Computer Aided Method Engineering* (CAME)
- ♦ Multimeedia registrid

Keerukate rakenduste näited (2)

- ♦ Dokumendiregistrid ja elektrooniline kirjastamine
- ♦ Geograafilised infosüsteemid e geoinfosüsteemid (GIS)
- ♦ Teadusuuringute andmebaasid
 - Andmebaasid geneetilise info hoidmiseks
- ♦ Sotsiaalvõrgustikud

Keerukate rakenduste näited (3)

- ♦ Teadmusbaasid
 - Ettevõtte infoportaal
 - Ettevõtte Teadmiste Portaal
- ♦ Mõned funktsionaalsused, millest oleks abi kõigi eelnevate rakenduste juures
 - Hägusad päringud
 - Leia kliendid, kes **suure tõenäosusega** võiksid meilt osta meie **peagi** müügile tulevaid **ülialkeid** kaupu
 - Tehisintellekt (AI – Artificial Intelligence)

Eelnevate rakenduste üldised nõuded andmebaasidele

- ♦ Objektide, mille andmeid soovitakse registreerida, *struktuur on keeruline*.
- ♦ Andmebaasi *struktuur muutub sageli*.
- ♦ Objektide andmete terviklikkuse kontrolliks on vaja jõustada *keerukaid reegleid*.
- ♦ *Suured andmemahud*.
- ♦ Andmeid tuleb *reaalajas* juurde (nt sensorid, veebiliiklus, tehinginfo börsidelt)

Eelnevate rakenduste üldised nõuded andmebaasidele (2)

- ♦ Vajalik on *integreerida* erinevatest allikatest pärit andmeid, millel erinev vorming.
- ♦ Andmebaasis toimunud sündmus (nt objekti muutmine) peab võimaldama käivitada tegevusi (nt kasutajate teavitamine).
- ♦ Andmetega toimuvad *keerukad operatsioonid*.

Eelnevate rakenduste üldised nõuded andmebaasidele (3)

- ♦ *Kauakestvad transaktsioonid*, mille käigus ka teised kasutajad peavad objekti andmetele ligi pääsema.
- ♦ Vajalik on *versioonihaldus*.

Relatsiooniline mudel

- ♦ **Levinud väide:** relatsioonilist mudelit kasutavad andmebaasisüsteemid ei ole sellist tüüpi rakenduste loomiseks sobilikud.
- ♦ **Õppejõu arvamus:** Tegelikult on puudused SQLis (relatsioonilise mudeli üks võimalik **realisatsioon**) ja SQLi pakkuvates andmebaasisüsteemides, mitte relatsioonilises mudelis!!!

SQL-andmebaasisüsteemide (mitte relatsioonilise mudeli) puudused

- ♦ Tabelite viimine kõrgele normaliseerituse astmele muudab andmebaasi kasutamise *aeglaseks*.
- ♦ Andmebaasisüsteemid võimaldavad kasutada piiratud hulka *lihtsaid andmetüüpe*.
- ♦ *CLOB/BLOB* tüüpi andmete põhjal ei saa teha keerukaid päringuid.
- ♦ Ei saa defineerida uusi *operaatoreid*.
- ♦ Keerukas on teha päringuid *graafi* (sealhulgas puu) struktuuri esitavate andmete põhjal.

SQL-andmebaasisüsteemide (mitte relatsioonilise mudeli) puudused (2)

- ♦ Andmete ning neid kasutavate programmide *struktuurne eraldatus*
 - rakenduses klassid/objektid, andmebaasis tabelid/read – objektid tuleb salvestamiseks "laiali lõhkuda"
 - Probleeme aitaks ületada kui:
 - Klass = Tüüp
 - Objekt = Väärtus

Näiteks saate defineerida tüübi *t_Tellimus* ning saate luua andmebaasis tabeli *Tellimus*, milles on *t_Tellimus* tüüpi veerg.

SQL-andmebaasisüsteemide (mitte relatsioonilise mudeli) puudused (3)

- ♦ Andmebaasisüsteemis ja rakenduste loomise vahendites erinevad toetatud tüübid.
- ♦ Pole võimalik piisavalt täpselt realiseerida *üldistuseseid* ja *osa-terviku seoseid*.
 - Raske realiseerida vastavat tüüpi seosega seotud kitsendusi.
 - [Moodul]-<*>-----1..*-[Rutiin]
 - Sisuliselt võimatu luua *deklaratiivset* kitsendust, mis tagaks, et iga mooduliga oleks alati seotud vähemalt üks rutiin.
 - Raske realiseerida vastavat tüüpi seostega seotud operatsioone (lugemine, lisamine, muutmine, kustutamine).

SQL-andmebaasisüsteemide (mitte relatsioonilise mudeli) puudused (4)

- ♦ Keeruline ja/või kulukas (litsentsitasud) tööle panna *arvutite klastritel*.
 - Kasulik päringute jõudluse ja transaktsioonide läbilaskevõime suurendamisel kui ei soovita võtta kasutusel üha võimsamaid arvuteid.
 - Selle asemel võiks kasutada palju „tavalisi“ arvuteid, mida vajadusel klastrisse juurde lisada (horisontaalne skaleeritavus).



SQLi (mitte relatsioonilise mudeli) puudused

- ♦ Lopsakas (barokne), kohmakas, mõnikord mitte-intuiitiivne andmebaasikeele süntaks.

Kokkuvõte

- ♦ Tüübid e domeenid, operaatorid ja deklaratiivsed kitsendused pole tänapäeva SQL-andmebaasisüsteemides piisavalt hästi toetatud.
- ♦ **Lahendus:** Juurutada täielikult relatsiooniline andmemudel.

Selle asemel ... kolmas põlvkond

- ♦ Esimene põlvkond
 - Hierarhilised andmebaasisüsteemid
 - Võrkstruktuuriga andmete andmebaasisüsteemid
- ♦ Teine põlvkond
 - Relatsioonilised andmebaasisüsteemid (õigemini küll **SQL-andmebaasisüsteemid**)
- ♦ **Kolmas põlvkond**
 - Objektorienteeritud andmebaasisüsteemid (OO andmebaasisüsteemid)
 - Objekt-relatsioonilised andmebaasisüsteemid
 - XML-andmebaasisüsteemid
 - NoSQL andmebaasisüsteemid (**neljas põlvkond!?**)
 - NewSQL andmebaasisüsteemid (**viies põlvkond!??**)

Objektorienteerituse põhimõisted

- ♦ **Objekt** (eksemplar) on tarkvaraline ühendus muutujatest ja nendega seotud meetoditest. Kõik, mida tarkvaraobjekt teab (seisund) ja oskab teha (käitumine), on väljendatud muutujate ja meetodite abil sama objekti sees.
- ♦ **Klass** on prototüüp või näidis, mis defineerib kõigile mingit tüüpi objektidele ühtsed muutujad ja meetodid.

Objektorienteerituse põhimõisted (2)

- ♦ **Kapseldamine** on objektide muutujate varjamine meetoditega.

Andmebaasid II 2017 © Erki Eessaar

Näide

- ♦ Klass: Jalgratas
- ♦ Muutujad (atribuudid):
 - käikude arv,
 - hetkel kasutatav käik,
 - hetke kiirus,
 - hetke pedaalide pöörlemiskiirus.
- ♦ Meetodid:
 - kiirenda,
 - pidurda,
 - vaheta käiku,
 - anna kella.

10.01.2018 Teema 9 19

Andmebaasid II 2017 © Erki Eessaar

Analoogia

Klass – koogivorm

Objekt – kook

[Edasi](#)

10.01.2018 Teema 9 20

Andmebaasid II 2017 © Erki Eessaar

Objektorienteerituse põhimõisted – igal objektil on

- ♦ **Privaatne mälu.** Objekti muutujad, mille väärtus määrab objekti seisundi.
 - Praktikas võib klass lubada ka avalikke muutujaid.
- ♦ **Avalik liides.** Meetodid, mille kaudu on võimalik nende muutujate väärtuseid lugeda ja muuta. Meetod on defineeritud oma *signatuuriga* – meetodi nimi, meetodi täitmiseks vajalikele andmetele vastavad parameetrid.
 - Meetodi väljakutsel tuleb igale kohustuslikule parameetrile seada vastavusse parameetri tüüpi kuuluv väärtus.

10.01.2018 Teema 9 21

Andmebaasid II 2017 © Erki Eessaar

Objektorienteerituse põhimõisted (3)

- ♦ Tarkvaraobjektid mõjutavad üksteist ja suhtlevad omavahel **teateid** vahetades.
- ♦ Objekt saadab teisele objektile teate, et see rakendaks mõne oma **meetodi**.
- ♦ Igal meetodil on null või rohkem **parameetrit**.
 - Objektile saadetud teates on esitatud parameetrite väärtused (argumendid).

10.01.2018 Teema 9 22

Andmebaasid II 2017 © Erki Eessaar

Objektorienteerituse põhimõisted (4)

- ♦ OO süsteemides on lubatud klasside vaheline **pärimine**, kus alamklassid pärivad ülaklassi seisundi ja käitumise.
 - Alamklassid saavad lisada päritud muutujatele ja meetoditele oma spetsiifilisi muutujaid ja meetodeid.
 - Alamklassid saavad päritud meetodeid asendada ja realiseerida spetsiifilisi lahendusi nende meetodite jaoks.

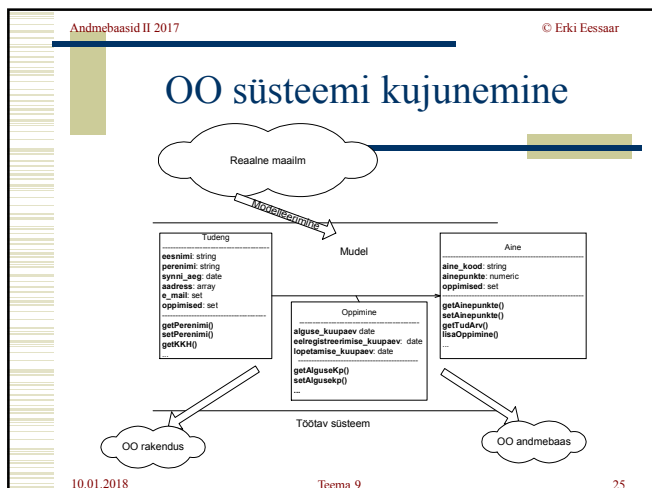
10.01.2018 Teema 9 23

Andmebaasid II 2017 © Erki Eessaar

Objektorienteerituse põhimõisted (5)

- ♦ Igal objektil on unikaalne **objekti identifikaator (ID)**, mis tähendab objekti **füüsilist aadressi mälus** ja mida kasutatakse objektile viitamiseks.
- ♦ Kaks objekti võivad olla atribuutide väärtuste poolest võrdsed, kuid sellele vaatamata olla erinevad objektid, sest neil on erinev OID.
 - Öeldakse, et kaks objekti on võrdsed, kuid erineva identiteediga.

10.01.2018 Teema 9 24



- Andmebaasid II 2017 © Erki Eessaar
- ## OO põhimõtete ja andmebaasisüsteemide integreerimise strateegiaid
- ♦ Laiendada olemasolevat OO programmeerimiskeelt (nt Java, C++, Smalltalk).
 - ♦ Objektorienteeritud andmebaasisüsteemid.
 - Luua algusest peale uus andmemudel ja andmebaasi programmeerimise keel.
 - ♦ Objekt-relatsioonilised andmebaasisüsteemid.
 - Laienda olemasolevat andmebaasi programmeerimise keelt (nt SQL) objektorienteerituse võimalustega (SQL:1999 ja hilisemad SQL standardi versioonid).
- 10.01.2018 Teema 9 26

- Andmebaasid II 2017 © Erki Eessaar
- ## OO andmebaasisüsteemid
- ♦ OO programmeerimiskeeles kirjutatud programmi töö käigus luuakse klasside põhjal **objekte**, mis asuvad arvuti **muutmälus**.
 - ♦ OO andmebaasisüsteem peaks võimaldama nende objektide valutut **püsisalvestamist** (**materialiseerimist**), ilma, et tuleks hakata tegelema nende "tõlkimisega" relatsioonilise andmebaasi/SQL-andmebaasi jaoks sobivale kujule ja andmete jagamisega erinevate tabelite vahel.
- 10.01.2018 Teema 9 27

- Andmebaasid II 2017 © Erki Eessaar
- ## OO andmebaaside omadused
- ♦ Objektide püsivalt salvestamise võimalus.
 - ♦ **Objekti ID** (süsteemi genereeritud, unikaalne, kasutajale nähtamatu objekti **füüsiline aadress mälus**).
 - ♦ Objektide vahelised seosed **viitade** kaudu.
 - ♦ Objekti struktuur suvalise keerukusega. Objektide hierarhia on tegelikult realiseeritud viitade kaudu.
- 10.01.2018 Teema 9 28

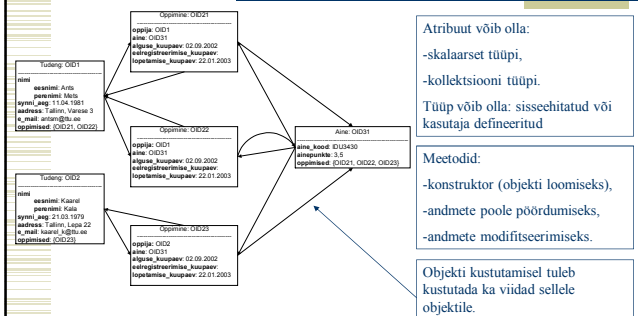
- Andmebaasid II 2017 © Erki Eessaar
- ## Objektide hierarhia, viidad – näide
- ♦ Igas osakonnas on null või rohkem töötajat.
 - ♦ Iga töötaja töötab täpselt ühes osakonnas.
 - ♦ Klass EMP (Töötaja)
 - ♦ Klass DEPT (Osakond)
 - Klassi DEPT muutuja **EMPS** on EMP kollektsiooni tüüpi.
 - Muutuja **EMPS** tegelik väärtus on **viitade kogum** EMP objektidele.
- | OID | #DEPT | ENAME | BUDGET | EMPS |
|-----|-------|-----------|---------|---------|
| ddd | 1 | Marketing | 5000005 | eee aaa |
- | OID | #EMP | ENAME | SALARY |
|-----|------|-------|----------|
| eee | 200 | Smith | 30000 \$ |
- | OID | #EMP | ENAME | SALARY |
|-----|------|---------|----------|
| aaa | 220 | Jackson | 35000 \$ |
- Edasi
- 10.01.2018 Teema 9 29

- Andmebaasid II 2017 © Erki Eessaar
- ## OO andmebaaside omadused (2)
- ♦ Muutujate **kapseldamine** meetodite abil.
 - ♦ Meetodite **polümorfism**.
 - Sama nimega, kuid erineva realiseerimisega meetodid.
 - ♦ **Versioonihaldus** – andmebaas peab suutma talletada sama objekti erinevaid versioone.
 - ♦ **Skeemi evolutsiooni** võimaldamine:
 - Klasside loomine pärimise kaudu.
- 10.01.2018 Teema 9 30

Object Data Standard

- ♦ 1999. aastal pakkus Object-Data Management Group välja *Object Data Standardi*, mis kirjeldab OO andmebaasisüsteemi arhitektuuri ning koosneb.
 - Objektudel.
 - Object Definition Language (ODL).
 - Object Query Language (OQL).
 - Deklaratiivne keel andmete manipuleerimiseks.
 - Spetsifikatsioon OO andmebaasi kasutamiseks C++, Java ja Smalltalk keeles.

Andmebaasi kuuluvate objektide näide



OQL päringute näiteid

- ♦ Leia esimese õppimise OID üliõpilase (kelle matrikli_nr="990999") õppimiste hulgast:
 - first(element(SELECT t FROM t IN tudeng WHERE t.matrikli_nr="990999").Teostab);
- ♦ Leia kõik Tallinnas elavad tudengid:
 - SELECT t.matrikl_nr FROM t IN tudeng WHERE t.aadress.linn="Tallinn";

Probleemid OO andmebaasisüsteemidega

- ♦ Teoreetiliste (matemaatiliste) *aluste* puudumine.
 - Konsensuse puudumine.
 - Kõigi poolt tunnustatud standardite puudumine.
 - Object Data Standardi viimane versioon aastast 2001.
 - 2006-2007. aastal üritati edutult uut versiooni välja töötada.
- ♦ Andmebaasi ja rakenduse omavaheline *tihe sidusus*.

Probleemid OO andmebaasisüsteemidega (2)

- ♦ Paljude OO andmebaasisüsteemide **nõrgad kohad**.
 - Vaated.
 - Deklaratiivsed kitsendused.
 - Deklaratiivne andmekäitluskeel.
 - *Ad hoc* päringud
 - pähe karanud küsimustele kohe ja kiiresti vastuse saamiseks
 - Keerukate päringute kiire täitmine.
 - Objektide lukustamine.
 - Turvalisuse tagamine.
 - Objektide vaheliste viidete haldamine.
 - Süsteemikataloog.

Objektorienteeritud andmebaasisüsteemide näited

- ♦ Caché
- ♦ Db4o
- ♦ ObjectStore
- ♦ Versant Object Database
- ♦ Matisse
- ♦ ObjectDB
- ♦ Objectivity/DB
- ♦ GemStone/S
- ♦ Perst

Tänapäeval on objektorienteeritud andmebaasisüsteem *nišitoode*. <https://db-engines.com/en/ranking/object+oriented+dbs>

Andmebaasid II 2017 © Erki Eessaar

XML- andmebaasisüsteemid

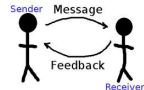
```
<aine>
  <aine_kood>IDU0220</aine_kood>
  <nimetus>Andmebaasid I</nimetus>
  <ainepunkte>5</ainepunkte>
  <loenguid>1</loenguid>
  <laboreid>0</laboreid>
  <harjutusi>1</harjutusi>
</aine>
```

- Andmed esitatakse XML dokumentidena.
- Bernstein et al.: The Asilomar Report on Database Research, ACM SIGMOD Record 27, No. 4 (December 1998).
 - "Unfortunately, XML is likely to generate chaos for database systems. XML's evolving query language is reminiscent of the procedural query language prevalent 25 years ago. XML is also driving the development of client-side caches that will support updates which is leading the XML designers into a morass of distributed transaction issues."

10.01.2018 Teema 9 37

Andmebaasid II 2017 © Erki Eessaar

XML- andmebaasisüsteemid (2)



- Süsteemid vahetavad teateid.
- Teade võib olla XML dokument.
- Kui süsteem saab XML dokumendi, siis:
 - võib selle sõeluda, eraldada andmeväärtused ning salvestada need oma andmestruktuuridesse (nt tabelid),
 - salvestada XML dokumendi,
 - kasutada eelmise kahe kombinatsiooni.

10.01.2018 Teema 9 38

Andmebaasid II 2017 © Erki Eessaar

Kuidas XML dokumente SQL-andmebaasis salvestada?

- Tekstitüüpi väärtusena tekstitüüpi (nt CLOB tüüpi) veerus.
- SQL standardis defineeritud XML tüüp ja operaatorid XML tüüpi andmetega töötamiseks – dokumendi saab salvestada tabelisse.
 - Oracle võimaldab lisaks luua XML tüübi põhjal XML tüüpi tabeli.

10.01.2018 Teema 9 39

Andmebaasid II 2017 © Erki Eessaar

Native XML

- Olemas spetsiaalsed XML dokumentide talletamiseks/haldamiseks mõeldud andmebaasisüsteemid (*Native XML Database Management Systems*)(NXD).
 - XML dokument seal fundamentaalne salvestusühik.

10.01.2018 Teema 9 40

Andmebaasid II 2017 © Erki Eessaar

Arengusuunad

10.01.2018 Teema 9 41

Andmebaasid II 2017 © Erki Eessaar

NoSQL

- Nimi võeti kasutusele 11.06.2009 ühel tarkvaraarendajate kohtumisel San Franciscos.
- Vaidlus, kas tähendab „No **to** SQL“ või „Not **Only** SQL“.
 - Variant 2 paremini kooskõlas suundumusega *Polyglot Persistence* – isegi üks ja sama rakendus võib kasutada erinevatel andmemudelitel põhinevaid andmekogusid; erinevat tüüpi andmete jaoks sobivad erinevad andmemudelid ja süsteemid.

10.01.2018 Teema 9 42

Andmebaasid II 2017 © Erki Eessaar

Oht üle pingutada



What the hell have you built.

- Did you just pick things at random?
- Why is Redis talking to MongoDB?
- Why do you even use MongoDB?



Allikas:
<https://twitter.com/codinghorror/status/347070841059692545>

10.01.2018 Teema 9 43

Andmebaasid II 2017 © Erki Eessaar

Agregaat

- Agregaat on kogum seotud objekte, mida soovitakse käsitleda ühe tervikuna.
- Näide:


```

      {"isik": {
        "isikukood": "39009090123",
        "eesnimi": "Mart",
        "perenimi": "Mets",
        "auto": [
          {"auto_nr": "333ABC",
            "mark": "Volvo"},
          {"auto_nr": "335SER",
            "mark": "Volga"}]}
      
```



Rakendus otsib agregaatide võtme (antud juhul isikukood) järgi ning muudab agregaatide (see peab olema terviklik atomaarne operatsioon).

Serverite klastril paikneva andmekogu korral agregaatide ei lõhuta – läheb tervikuna ühte serverisse.

10.01.2018 Teema 9 44

Andmebaasid II 2017 © Erki Eessaar

Agregaat (2)

- Agregaadi struktuuri kavandamisel tuleb mõelda sellele kuidas tüüpiliselt agregaat otsitakse – tuleb mõelda *pöördusteele*.
- Eelmise slaidi lahendus on optimeeritud otsimisele *isikukoodi* alusel.
- Kui põhiliselt toimub otsimine *auto numbri* järgi, siis peaks struktuur olema teistsugune – auto andmetega koos salvestatakse omanik.

10.01.2018 Teema 9 45

Andmebaasid II 2017 © Erki Eessaar

Agregaat (3)

- Mida teha, kui *Isik* ja *Auto* vahel oleks M:N seosetüüp? Võimalus – kaks eraldi agregaatide.
 - Isik ja tema kasutatud autod.
 - Auto ja seda kasutanud isikud.
 - Vaja kahte, et saaks registreerida autot isikut ja isikuta autot.
- Tulemuseks **kontrollimatu andmete liiasus**.
 - Sama väide registreeritud mitmes kohas, andmebaasisüsteem ei taga kooskõla.

10.01.2018 Teema 9 46

Andmebaasid II 2017 © Erki Eessaar

Agregaat (4)

- Mõni süsteem võib pakkuda ka võimalust defineerida agregaatide vahelisi seoseid, kuid see tingib *jõudluse* probleeme.
- Agregaatide struktuur, mis sobib ühele rakendusele ei ole kõige sobivam teisele.
- Ka sama rakenduse korral võib sobilik agregaatide struktuur ajas muutuda.

10.01.2018 Teema 9 47

Andmebaasid II 2017 © Erki Eessaar

Agregaat – kokkuvõte



- NoSQL süsteemid üritavad pakkuda kooskasutatavate andmete (agregaatide) kiire lugemise võimalust. **Hinnaks** on
 - andmete liiasus,
 - koopiad võivad olla mittekooskõlalised,
 - keerulised ja aeganõudvad päringud, kui peaks kokku panema mitu agregaatide (st agregaat pole täpselt selline nagu rakendusel vaja),
 - uute agregaatide lisamine kiire, kuid olemasolevate andmete muutmine võtab aega.

10.01.2018 Teema 9 48

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemide "andmemudelid"

- ♦ Agregaatidel (kogumitel) põhinevad.
 - Võtme-väärtuse paarid.
 - Agregaadiks on võtmega seotud väärtus.
 - Andmebaasisüsteem ei näe agregaatide „sisse“.
 - Dokumendid.
 - Agregaadiks on dokument.
 - Andmebaasisüsteem näeb agregaatide „struktuuri“, saab muuta andmeid agregaadis, otsida andmeid agregaadist.

10.01.2018 Teema 9 49

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemide "andmemudelid" (2)

- Veeruperekonnad.
 - Veeruperekond on nagu tabel selle vahega, et igal real võib olla erinev struktuur, st koosneda erinevatest veergudest.
 - Agregaadiks on veeruperekonda kuuluv rida.
 - Andmebaasisüsteem näeb agregaatide „struktuuri“, saab muuta andmeid agregaadis, otsida andmeid agregaadist.
- ♦ Graafidel põhinevad.

10.01.2018 Teema 9 50

Andmebaasid II 2017 © Erki Eessaar

Andmete esitus – võti väärtus paarid ja dokumendid

Võti **Väärtus**

Võti-väärtus paarid

Isik1 39009090123, Mart, Mets, 333ABC, Volvo, 335SER, Volga

Dokumendid

```
{
  "Isik": {
    "Isikukood": "39009090123",
    "Eesnimi": "Mart",
    "Perenimi": "Mets",
    "auto": {
      "auto_nr": "333ABC",
      "mark": "Volvo",
      "auto_nr": "335SER",
      "mark": "Volga"
    }
  }
}
```

Kuidas seda väärtust kasutada teab kasutaja (rakendus). Andmebaasisüsteemi jaoks see must-kast.

Andmebaasis saab teha päringuid ja andmemuudatusi, viidates dokumendis olevatele atribuutidele (võtmetele).

10.01.2018 Teema 9 51

Andmebaasid II 2017 © Erki Eessaar

Andmete esitus – veeruperekond

Primaar võti

Veerg

Veeruperekond

Rida

Rida

Võti-väärtus paaride hierarhia. Erinevatel ridadel võib olla erinev struktuur. Andmebaasis saab teha päringuid ja andmemuudatusi, viidates ridades olevatele võtmetele.

Cassandra on kõik read, kus on primaarvõtme esimeses veerus samasugune väärtus, koos ühel serveril. Primaarvõtme ülejäänud veergude alusel on read kettal sorteeritud.

10.01.2018 Teema 9 52

Andmebaasid II 2017 © Erki Eessaar

Andmete esitus – omaduste graaf

Sõlm

Serv

Isik

Auto

Auto

Isik: "Isikukood": "39009090123", "Eesnimi": "Mart", "Perenimi": "Mets"

Auto: "auto_nr": "333ABC", "mark": "Volvo"

Auto: "auto_nr": "335SER", "mark": "Volga"

10.01.2018 Teema 9 53

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemide näiteid (ühel andmemudelil põhinevad)

- ♦ Võtme-väärtuse paarid
 - Redis, Memcached, Hazelcast
- ♦ Dokumendid
 - MongoDB, AmazonDynamoDB, Couchbase
- ♦ Veeruperekonnad
 - Cassandra, HBase
- ♦ Graafid
 - Neo4j, Titan

Pikem nimekiri: <https://db-engines.com/en/ranking>

10.01.2018 Teema 9 54

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemide ühiseid omadusi

- ♦ Turule tulnud 21. sajandi alguses.
- ♦ Ei kasuta SQLi.
- ♦ Enamasti avatud lähtekoodiga.

10.01.2018 Teema 9 55

Andmebaasid II 2017 © Erki Eessaar

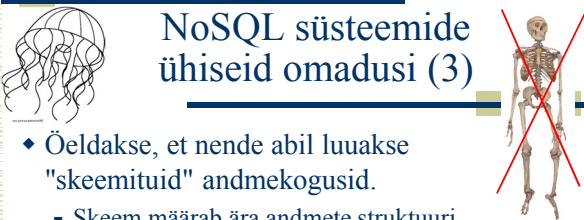
NoSQL süsteemide ühiseid omadusi (2)

- ♦ Paljude puhul lähtunud vajadusest panna süsteem tööle arvutite klastril e kobaral.
 - Võimaldab lisada süsteemi järk-järgult uusi servereid.
 - *Shared Nothing Partitioning* (vt Teema 4).
 - Töökindluse huvides on võimalik andmete tiražeerimine sama kobara erinevatele serveritele (vt Teema 12).

10.01.2018 Teema 9 56

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemide ühiseid omadusi (3)



- ♦ Öeldakse, et nende abil luuakse "skeemituid" andmekogusid.
 - Skeem määrab ära andmete struktuuri.
 - Andmebaasisüsteem *ei nõua* skeemi defineerimist ning ei saa skeemi kirjelduse puudumisel ka kontrollida andmete skeemile vastavust.
 - <https://martinfowler.com/articles/schemaless/>

10.01.2018 Teema 9 57

Andmebaasid II 2017 © Erki Eessaar

Skeemitusest

- ♦ Halb väljend, sest **skeem on olemas** – lihtsalt *suure tõenäosusega* pole seda andmebaasis ilmutatult kirjeldatud.
- ♦ "Skeem" on *ilmutamata kujul rakenduse koodis* – rakendus peab teadma andmete struktuuri, et nendega midagi peale hakata.
 - Kui andmetel ei ole skeemi (struktuuri), siis ei ole võimalik ka neid kasutada.

10.01.2018 Teema 9 58

Andmebaasid II 2017 © Erki Eessaar

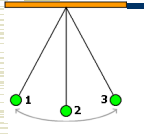
Skeemitusest (2)

- ♦ Õigem oleks eristada:
 - kirjutamise skeem (*schema-on-write*),
 - hulk reegleid, millele peavad andmed vastama ja mille täidetust kontrollib andmebaasisüsteem andmete andmebaasi lisamisel.
 - lugemise skeem (*schema-on-read*),
 - andmete lugeja arusaam andmete struktuurist, mis on vajalik andmetest arusaamiseks.
- ♦ NoSQL "skeemitus" tähendab, et andmetel:
 - ei ole kirjutamise skeemi,
 - on lugemise skeem.

10.01.2018 Teema 9 59

Andmebaasid II 2017 © Erki Eessaar

Skeemitusest (3)



- ♦ See, et NoSQL andmebaasisüsteemides **ei pea kirjutamise skeemi** defineerima, on andmebaasisüsteemide loojate **valik**, mitte andmemudelitest tulenev **paratamatus**.
- ♦ See valik on **muutumas** ja aina suuremas hulgas NoSQL süsteemides on võimalik kirjutamise skeemi aina rohkem jõustada.
 - Ühte äärmusesse liikunud pendel liigub teisele poole.

10.01.2018 Teema 9 60

Andmebaasid II 2017 © Erki Eessaar


Kirjutamise skeem – dokumendid

- ♦ Dokumentidele saab skeeme defineerida.
 - *JSON skeem* (<http://json-schema.org/>)
 - Varasemast ajast *XML skeem* (<https://www.w3.org/standards/xml/schema>)
- ♦ MongoDB.
 - Alates MongoDB 3.2 saab defineerida dokumendi **valideerimisreegleid** ja soovi korral uusi dokumente nende suhtes kontrollida.

10.01.2018 Teema 9 61

Andmebaasid II 2017 © Erki Eessaar

Kirjutamise skeem – dokumendid (2)



- Alates MongoDB 3.6 saab MongoDB andmebaasis defineerida **JSON skeemi** ja kontrollida dokumentide vastavust sellele.
- Valideerimine toimub dokumentide lisamisel või muutmisel.
- Valideerimise võimaluse saab lisada olemasolevale dokumentide kogumile – eelnevalt registreeritud dokumentide kontrollimiseks tuleb neid muuta.

10.01.2018 Teema 9 62

Andmebaasid II 2017 © Erki Eessaar

Kirjutamise skeem – dokumendid (3)

SQL-andmebaasisüsteemid – täielikult jõustatud deklaratiivsetest kitsendustest ei saa mööda minna. Võib olla võimalik kitsenduse seisundit muuta, et seda ei kontrollitaks lisatavate andmete või enne kitsenduse lisamist andmebaasis olnud andmete korral.

- ♦ MongoDB.
 - Lubab määrata, kas valideerimise ebaõnnestumine (uus dokument ei vasta reeglitele) tähendab andmemuudatuse **tagasilükkamist** või ainult **hoiatuse** logimist ja kasutajale edastamist.
 - Lubab lõppkasutajatel andmemuudatusi tehes valideerimisest **mööda minna**.
 - Ei luba valideerimist mõnedes andmebaasides.

10.01.2018 Teema 9 63

Andmebaasid II 2017 © Erki Eessaar

Kirjutamise skeem – dokumendid (4)

- ♦ Sõltumata sellest, kas dokumentide oodatavat struktuuri saab andmebaasis kirjeldada või mitte, kontrollib dokumendipõhine andmebaasisüsteem et dokument oleks **trimmis** e reeglipõhine.
 - Nagu siin saab valideerida, kas tegu on trimmis JSON dokumendiga: <https://jsonformatter.curiousconcept.com/>

10.01.2018 Teema 9 64

Andmebaasid II 2017 © Erki Eessaar

Skeemitusest – kokkuvõte


- ♦ Eeliseid
 - Ei pea kulutama aega skeemi loomisele ja muutmisele. Neid tegevusi kutsutakse mõnikord üldolevalt "rituaaliks" või "tseremooniaks".
 - Rakendused ei pea andmeid salvestamiseks **teisendama/tükeldama** – salvestavad sama struktuuriga andmeid nagu ise kasutavad.
- ♦ Probleme
 - Andmete kasutajad (rakendused) peavad oskama arvestada andmete **erineva struktuuriga**.
 - Tekitab kiusatuse liiga **sageli** andmete struktuuri muuta.
 - Andmete struktuur **ebaselge**, andmete kasutamises on lihtsam teha **vigu**.
 - Välisel **vaatlejal** raskem aru saada, milliseid andmeid süsteemis hallatakse.
 - Andmebaasisüsteemil on andekäitluse operatsioonide **optimeerimiseks** vähem infot.

10.01.2018 Teema 9 65

Andmebaasid II 2017 © Erki Eessaar

NoSQL süsteemid (4)

Paljudel ettevõtte infosüsteemidel teistsugused vajadused.



- ♦ Paljud on kasvanud välja **sotsiaalmeedia** süsteemide vajadustest:
 - suured ja kiiresti kasvavad andmehulgad,
 - uusi andmeid tuleb peale väga suurel hulgal,
 - andmete struktuur pole rangelt paigas
 - väga suur samaaegsete kasutajate hulk ning vajadus säilitada ka sellisel juhul hea jõudlus,
 - lõdvenenud nõuded transaktsioonidele on aktsepteeritavad.

10.01.2018 Teema 9 66

NoSQL süsteemid (5)

- ◆ Tänapäeval tekib suurandmeid ka mujal kui sotsiaalmeedia süsteemides:
 - erinevate seadmete poolt edastatav asukohainfo,
 - süsteemide kasutamise logid, sh IP aadressid,
 - erinevatelt sensoritelt pärit mõõteinfo,
 - raadiosagedustuvastuse tulemused.
- ◆ Nende puhul on NoSQL süsteemi kasutamine kaalumist väärt.

Dolumendipõhise NoSQL süsteemi analoog – Maurus

- ◆ Erinevad üliõpilased eelistaksid **erinevat kataloogide struktuuri** (teemade kaupa, nädalate kaupa, kombinatsioon eelmistest).
- ◆ Kui teha ainele **mitu aktiivset lehte** (NoSQLi promotav **andmete dubleerimine**), kus igaühes erinev kataloogide struktuur, siis muutuks dokumentide uuendamine väga aeganõudvaks.
 - Võimalik, et samal ajahetkel on samast dokumendist erinevatel lehtedel erinev versioon (viiteajaga terviklikkuse saavutamine; vt Teema 12).

Hierarhiate näited

- **Kasutatakse** (10) (a)
- **Teostavad kõik projektid tagamaks** (0) (a)
- **Näitame** (2) (a)
- **Mäeld** (1) (a)
- **Teo** laetumise (sisse-astumise) jend (1) (a)
- **Teostav teinend** (vabatahtlik, etim) **vastutamine** (1) (a)
- **Teostav teinend** (kannatuste suurendamiseks) (4) (a)
- **Longitud** (9) (a)
- **Teostav teinend** ja **teostav teinend** (2) (a)
- **CASE** **Enterprise Architect** (või 11 baasi) **vedes** (1) (a)
- **EA** **moduliseerimise** **teinend** (1) (a)
- **CASE** **Rational Rose** **vedes** (1) (a)
- **PostgreSQL** **vedes** (**PostgreSQL 9.3** baasi) **vedes** (3) (a)
- **PostgreSQL** **+3E** **Access** **vedes** (**MS Access 2013 baasi**) (1) (a)
- **PostgreSQL** **+ FHP** **vedes** (1) (a)
- **Oracle** **vedes** (**Oracle 12 baasi**) (1) (a)
- **Oracle** **AFXP** **vedes** (**AFXP 4 baasi**) **400** (a)
- **Tuuletud** (1) (a)

Hierarhia, võttes aluseks teadmiste kontrollide/ülesannete tüübid

Hierarhia, võttes aluseks nädalad

Dolumendipõhise NoSQL süsteemi analoog – Maurus (2)

- ♦ Kuna erinevate ainete lehtedel on kataloogide struktuur erinev ja seda saab ka semestri sees jooksvalt kergesti muuta (NoSQLi promotav **paindlikus**), siis kasutajatel vajalik pidev vaimne ümberlülitumine (kurnav, segab keskendumist).

Dolumendipõhise NoSQL süsteemi analoog – Maurus (3)

- ♦ Mida teha, kui **dokument** peaks olema **mitmes kataloogis**, mis on samal või erinevatel lehtedel?
 - **Dubleerida** dokumenti erinevates kohtades.
 - Dokumendi muutmise võtab palju rohkem aega.
 - Võimalik, et samal ajahetkel on samast dokumendist erinevatel lehtedel erinev versioon.
 - Võimalik, et osades kohtades ununeb muudatus tegemata.

Dolumendipõhise NoSQL süsteemi analoog – Maurus (4)

- Kasutada **viitamist**.
 - Viitan leheküljelt **366** lehekülje **346** kataloogile *Andmebaasid 2/Andmebaasisüsteemide juhendid ja koodinäited/Slaidid andmebaasiobjektide kohta* dokumendile **"Domeenid, vaated, sünonüümid"**
 - Kui muudan leheküljel **346** dokumendi nime, kataloogide struktuure või nimesid, siis lähevad viited **katki**.
 - Muutja peab ise kõik viited üles otsima ja ära parandama või leppima katkiste viidete negatiivsete tagajärgedega.

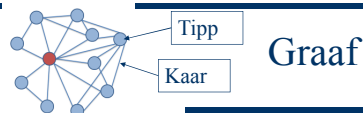
Juhtumianalüüs agregaatimise probleemide kohta

- ♦ Sarah Mei, Why You Should Never Use MongoDB, 11.11.2013.
- ♦ <http://www.sarahmei.com/blog/2013/11/11/why-you-should-never-use-mongodb/>
 - Üks ja sama objekt (nt isik) võib agregaadis olla esindatud mitmes erinevas kohas.
 - Dubleeri andmeid. ☹
 - Pane viited eraldi dokumenti ja teosta ühendused rakenduse koodis. ☹

10.01.2018

Teema 9

73



- ♦ Mittetühi hulk T koos selle elementide (graafi **tippude** e **sõlmede**) järjestatud või järjestamata paaride hulgaga S .
- ♦ Tippude järjestamata paari $(a,b) \in S$ nimetatakse graafi servaks e orienteerimata servaks, järjestatud paari **kaareks** ehk orienteeritud servaks.

Kaasik, Ü., 1992.
Matemaatikaleksikon. Eesti Entsüklopeediakirjastus.

10.01.2018

Teema 9

74

Graafi andmemudelid – *Resource Description Framework (RDF)*

- ♦ Osa püüdlusest luua semantiline veeb.
- ♦ Graafi andmed esitatakse kolmikutena – **subjekt-predikaat-objekt**.
 - Erki Eessaar õpetab Andmebaasid I
- ♦ Standartne päringukeel SPARQL (W3C standard).
 - Süntaks meenutab SQLi.
- ♦ Andmebaasisüsteemi näide: MarkLogic.

10.01.2018

Teema 9

75

Graafi andmemudelid – Omaduste graaf (*property graph*)

- ♦ Nii graafi sõlmed kui servad võivad olla seotud atribuut-väärtus (võti-väärtus) paaridena esitatud omadustega.
- ♦ Andmebaasisüsteemi näide: Neo4j.
 - Selles saab kasutada *Cypher* päringukeelt.
 - Ei paku regulaarse tee päringute võimalust (2017. seisuga).
 - ♦ Regulaarse tee päring leiab teega ühendatud sõlmed, kus teel olevate servadega seotud andmed rahuldavad mingit regulaaravaldist.

10.01.2018

Teema 9

76

Omaduste graafi andmemudel (2)

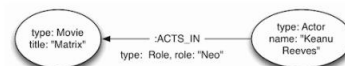
- ♦ Püüdlus töötada välja ühtne päringukeel – PGQL (*Property Graph Query Language*), mis toetaks kõiki tüüpilisi graafi operatsioone sh regulaarse tee päring.
 - Süntaks ja funktsionaalsus on SQLi sarnane.
 - <http://pgql-lang.org/>

10.01.2018

Teema 9

77

NoSQL – graafidel põhinevad süsteemid (Neo4j näitel)



Allikas: <https://neo4j.com/developer/cypher-query-language/>

- ♦ Leia filmi "Matrix" (mille id="603") kõik osatäitjad sorteerituna nime järgi:
 - `START m=node:node_auto_index(id="603")`
`MATCH m<-[:ACTS_IN]-actor RETURN actor.name order by actor.name;`

10.01.2018

Teema 2

78

NoSQL – graafidel põhinevad süsteemid (Neo4j näitel) (2)

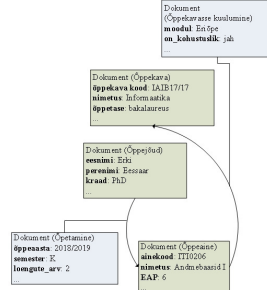
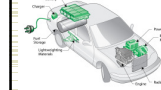
- ♦ Alates versioonist 2.0 *võimalus* defineerida **kirjutamise skeem**, millele vastavust jõustab andmebaasisüsteem.
 - Saab määrata, et
 - nimega n sõlme või kaare korral on atribuudi a väärtuse registreerimine kohustuslik
 - andmebaasis ei tohi olla kahte sõlme, millel on samasugune nimi ja atribuudi a väärtus
 - Otsingute kiirendamiseks saab luua indekseid.

10.01.2018

Teema 9

79

Hübriidsüsteemid (multi model systems)



- ♦ Kombineerivad mitu andmemudelit.
- ♦ Näide: ArangoDB
 - Andmebaas on dokumentide **graaf**.
 - Sõlmed ja kaared on JSON **dokumentid**.
 - Dokument on kogum **võtme-väärtuse** (atribuutide-atribuudi väärtuste) **paare**.

10.01.2018

Teema 9

80

Hübriidsüsteemid – ArangoDB näitel (2)

- ♦ Deklaratiivne andmebaasikeel AQL, mis mõeldud kõigi toetatud andmemudelite jaoks.
 - Süntaks meenutab SQLi.
- ♦ Andmebaasiserveris talletatud rutiinide asemel JavaScriptil põhinev mikroteenuste kirjutamise raamistik (nimega Foxx).
 - Võib näiteks kasutada:
 - kasutajate autentimiseks ja autoriseerimiseks,
 - dokumentide skeemile vastavuse kontrolliks.

10.01.2018

Teema 9

81

NoSQL süsteemid – eeliseid

- ♦ *Võimalikud* eelised võrreldes SQL-andmebaasisüsteemidega:
 - parem operatsioonide *töökiirus* väga suurtel andmehulkadel,
 - *horisontaalne skaleeritavus* (võime kasvada järk-järgult uute serverite lisamise abil).

10.01.2018

Teema 9

82

NoSQL – probleeme

- ♦ Selgelt defineeritud ning matemaatilise aluspõhjaga andmemudeli puudumine ja *madalam abstraktsioonitase*.
 - muudab raskeks väljaspool konkreetset rakendust (millega seoses NoSQL süsteem on kasutusele võetud) *andmete ligipääsemise* ning nendest andmetest *arusaamise*.
- ♦ *Väga palju* erinevaid tehnilisi lahendusi, mis pidevas muutuses.

10.01.2018

Teema 9

83



Lukustamise näide (MongoDB 2.6)

- ♦ "MongoDB uses a readers-writer [1] lock that allows concurrent reads access to a **database** but gives exclusive access to a single write operation."
 - Muutmisoperatsioon lukustab terve andmebaasi.
- ♦ MongoDB 3.1 lukustab andmete muutmiseks kollektiooni (agregaadi).

10.01.2018

Teema 9

84

Imperatiivne vs. deklaratiivne (3)

- On aina rohkem NoSQL süsteeme, mille jaoks töötatakse välja **deklaratiivseid keeli**.
- Sageli meenutavad need SQLi (iroonia – NoSQL tõlgendus algselt "No to SQL").
 - Cassandra (veeruperekondade andmemudel) keel **CQL** – Cassandra Query Language.
 - Couchbase (dokumendipõhine andmemudel) keel – **N1QL** – SQL for JSON.

10.01.2018

Teema 9

91

Cassandra Query Language

♦ Peaaegu nagu SQL

Evans, E., 2011. CQL: SQL for Cassandra. [WWW] <https://www.slideshare.net/jericevans/cql-sql-in-cassandra> (06.12.2017)

```
Column col = new Column(ByteBuffer.wrap("name".getBytes()));
col.setValue(ByteBuffer.wrap("value".getBytes()));
col.setTimestamp(System.currentTimeMillis());

ColumnOrSuperColumn cosc = new ColumnOrSuperColumn();
cosc.setColumn(col);

Mutation mutation = new Mutation();
Mutation.setColumnOrSuperColumn(cosc);

List mutations = new ArrayList<Mutation>();
mutations.add(mutation);

Map mutations_map = new HashMap<ByteBuffer, Map<String, List<Mutation>>>();
Map cf_map = new HashMap<String, List<Mutation>>>();
cf_map.set("Standard1", mutations);
mutations.put(ByteBuffer.wrap("key".getBytes()), cf_map);
```



```
INSERT INTO Standard1 (KEY, col0)
VALUES (key, value0)
```

10.01.2018

Teema 9

92

```
SELECT b.name, DATE_PART_STR(a.year, "year") as year,
a.name as award
FROM awards a INNER JOIN books b
ON KEYS a.book_id
ORDER BY b.name, year, award
```

N1QL

Allikas: <https://www.couchbase.com/products/n1ql>

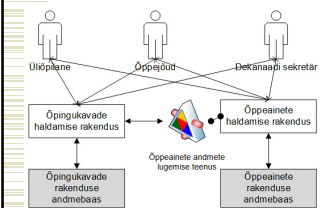
- Võimaldab Couchbase süsteemis midagi, mida mitmetes dokumendipõhistes NoSQL süsteemides ei saa teha.
 - Andmebaasis saab teha **deklaratiivse** päringu, millega ühendatakse (*join*) kokku erinevates dokumentides olevad andmed.
 - Ühendamist ei pea tegema rakenduses või rakenduse ja andmebaasisüsteemi vahel istuvas vahevaras või imperatiivselt andmebaasis.

10.01.2018

Teema 9

93

Tarkvara integratsioon NoSQL süsteemide kasutamise korral



Kuna andmebaasid on "skeemitud", siis mitme rakenduse poolt sama andmebaasi kasutamise realiseerimine on keerukas.

- Igal rakendusel oma andmebaas.
- Rakendus pakub teistele rakendustele oma andmebaasi kasutamise *teenuseid*.
- Muutunud aktuaalseks seoses *NoSQL süsteemide* populaarsuse kasvuga.

10.01.2018

Teema 9

94

Tarkvara integratsioon NoSQL süsteemide kasutamise korral (2)

- Tervik süsteem koosneb ühest või mitmest rakendusest ja ühest või mitmest andmebaasist.
- Erinevad rakendused võivad olla realiseeritud *erinevates programmeerimiskeeltes*.
- Erinevad andmebaasid võivad olla realiseeritud *erinevatel andmebaasisüsteemidel*.
- Iga *rakendus* realiseerib ühe või mitu **funktsionaalset allsüsteemi**.
- Iga *andmebaas* realiseerib ühe või mitu **registrit**.
- Andmebaasioperatsioonid** on aluseks *veebiteenuste* leidmisele.

10.01.2018

Teema 9

95

Terviklikkusest

- Enamik NoSQL süsteeme võimaldavad andmete jaotamist mitme serveri vahel.
- Andmete **terviklikkus** andmete **hajusa** paiknemise korral:
 - andmed rahuldavad andmebaasis jõustatud kitsendusi,
 - kui andmetest on koopiad, siis erinevates serverites olevad koopiad on ühesugused.

10.01.2018

Teema 9

96

Terviklikkusest (2)

- ♦ C transaktsioonide ACID omadustes tähendab esimest.
 - Andmed vastavad kitsendustele.
- ♦ Terviklikkus NoSQLi tähenduses.
 - "Skeemitus" andmebaasis kitsendusi ei jõustata.
 - Sageli võimaldavad lähenemist, et andmete koopiad erinevates serverites sünkroniseeritakse viiteajaga (*eventual consistency*).

NoSQL ja transaktsioonid

- ♦ Transaktsioonide puhul ei toetata täies mahus ACID omadusi.
 - Näiteks MongoDB korral kehtivad need vaid siis, kui operatsioon loeb ühte dokumenti ühel serveril.
- ♦ Süsteeme iseloomustab akronüüm BASE – *Basically Available, Soft state, Eventual consistency*.

NoSQL – transaktsioonid (MongoDB näitel)

- ♦ MongoDB ei paku automaatselt võimalust kasutada ACID omadustega transaktsioone, mis *koondavad mitme dokumendi muutmise*.
- ♦ MongoDBs on võimalik saavutada midagi *ACID-laadset*, kuid eeldab andmeid kasutava rakenduse kirjutajalt *programmeerimistööd*.
 - Isoleerituse tagamiseks *\$isolated* operaator.
 - Atomaarsuse ja terviklikkuse tagamiseks kahefaasilise kinnitamise protokoll (vt teema 12).

NoSQL – transaktsioonid (MongoDB näitel) (2)

- ♦ Väikimisi isolatsioonitase READ UNCOMMITTED
 - Näide
 - *Kasutaja1* ja *Kasutaja2* alustavad operatsioone ajahetkel *t*, kui andmebaasis on dokumendid *d1* ja *d2*.
 - *Kasutaja1* muudab ühe operatsiooniga dokumendi *d1* ja *d2* (asendab dokumentidega *d1'* ja *d2'*).
 - *Kasutaja2* tahab samal ajal lugeda dokumente *d1* ja *d2*.
 - *Kasutaja2* loeb tegelikult *d1'* ja *d2*.
 - *Kasutaja1* rullib muutuse tagasi, kuid *Kasutaja2* teeb otsuse loetud mittekooskõlaliste ja lõpuks üldse andmebaasi mitte jõudnud andmete põhjal.



Eventual consistency

- ♦ Vähemalt kahel korral on seda omadust kuritarvitades õnnestunud varastada bitcoin krüptoraha.
 - <http://hackingdistributed.com/2014/04/06/another-one-bites-the-dust-flexcoin/>
 - <https://bitcointalk.org/index.php?topic=499580>
 - Praktiliselt samal hetkel üritati raha välja võtta samalt kontolt, mitu korda ja summaarselt rohkem kui kontol oli raha.

Eventual consistency (2)

- ♦ Väljavõtud registreeriti erinevatel serveritel, mis koheselt omavahel andmeid ei sünkroniseerinud.
 - Väljavõtte oleks tulnud töödelda üksteise järel, mitte samaaegselt.
- ♦ Järeldused
 - Paljud süsteemid vajavad tugevamate garantiidega transaktsioone.
 - Iga ülesande jaoks tuleb valida õige lahendus.

NoSQL ja ACID

- Enamik NoSQL süsteeme ei paku ACID omadustega transaktsioone, kuid see **ei pea** nii olema.
- HyperDex* on NoSQL süsteem, mis koos laiendusega *Warp* toetab ACID transaktsioone (<http://hyperdex.org/warp/>)
 - Andmed on jaotatud erinevate serverite vahel.
 - MongoDB READ UNCOMMITTED näite olukorda ei saa tekkida.

10.01.2018

Teema 9

103

NoSQL ja standardid

- Turul on kümneid või isegi sadu NoSQL süsteeme.
- Puuduvad neid reguleerivad standardid.
 - V.a JSON avatud standard, mis reguleerib dokumentide struktuuri ja mille alusel dokumendipõhised NoSQL süsteemid kontrollivad, kas dokument on trimmis.

10.01.2018

Teema 9

104

NoSQL ja standardid (2)

Risk: Võtan kasutusele NoSQL süsteemi ja see "upub".

- Keskelt väljatöötatud standardile eelistatakse lähenemist, et kehvad süsteemid upuvad, kuid parimad jäävad ellu ja muutuvad *de facto* standardiks.
- Leitakse, et:
 - standarditele võib mõelda siis, kui tehnoloogia on piisavalt küpsenud,
 - kesksed standardid on "relv", millega vanad turu osalised saavad end uute tulijate eest kaitsta.

10.01.2018

Teema 9

105

Veel üks viis kuidas NoSQL süsteeme kasutada

- Andmebaas SQL-andmebaasisüsteemis.
- Selle peale ehitatud vahemälu (*cache*), kasutades mõnda dokumendipõhist NoSQL süsteemi (nt MongoDB).
 - Seal esitatakse rakendusele andmeid denormaliseeritud agregaatidena.
 - Vahemälu tuleb andmebaasi põhjal värskendada.

Kui SQL-andmebaasisüsteem toetab hetktõmmiseid ja JSONi, siis saab seda kõike teha SQL-andmebaasis.

10.01.2018

Teema 9

106

Lisamaterjali

- Hea kõrgtaseme ülevaade NoSQL süsteemide kohta.
 - Sadalage, P.J., Fowler, M. (2013) *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley. 164 p.
 - http://www.estor.ee/record=b3046556*est
- Michael Stonebrakeri ettekanne suurandmete ja nendega töötamise vahendite (sh NoSQL süsteemid) kohta.
 - <https://www.youtube.com/watch?v=KRcecxGxvQ>

10.01.2018

Teema 9

107

NewSQL

- Võimaldavad endiselt kasutada *SQLi*.
- Transaktsioonide puhul tagatakse *ACID* omadused.
- Sisemiselt on need andmebaasisüsteemid disainitud suurte andmehulkade töötlemist, jõudlust ja skaaleritavust silmas pidades.

10.01.2018

Teema 9

108

Tehniliste uuenduste näiteid

- ◆ Töö käigus kogu andmebaasi muutmälus hoidmine (vt teema 3).
- ◆ Andmete veerupõhine salvestamine (vt teema 4).
- ◆ Andmete salvestamine ja töötlemine serverite klastril *Shared Nothing Partitioning* põhimõttest lähtuvalt (vt teema 4).

10.01.2018

Teema 9

109

NewSQL (2)

- ◆ Uus pole mitte SQL, vaid vahendeid loovad firmad ja süsteemide sisemised põhimõtted.
 - ◆ SQL võimaluste tugi ei pruugi olla nii täielik kui "traditsioonilistes" SQL-andmebaasisüsteemides.
 - ◆ Süsteemide näited:
 - SAP HANA
 - VoltDB, NuoDB
- Neemre, P., 2013. *PostgreSQLi ja kahe NewSQL andmebaasisüsteemi omavaheline võrdlus*. Bakalaureusetöö. TTÜ Informaatikainstituut.

10.01.2018

Teema 9

110

OldSQL annab vastulöögi

- ◆ Andmete veerupõhine salvestamine (vt teema 4).
 - Microsoft SQL Server 2014 realiseerib andmete reapõhise salvestamise. Samas on selles realiseeritud indeksi tüüp, mis salvestab andmed veerupõhiselt.
- ◆ Võimalus salvestada andmeid agregaatidena või agregaatide dünaamiliselt päringuga moodustada.

10.01.2018

Teema 9

111



OldSQL annab vastulöögi – agregaatide salvestamine

- CREATE TABLE Tellimus (tellimus_id INTEGER NOT NULL PRIMARY KEY, tellimus_dok **JSONB** NOT NULL);
- INSERT INTO tellimus (tellimus_id, tellimus_dok) VALUES (1, '{"tellimuse_nr": 1600, "viide": "ABULL-20140421", "tellija": "Alexis Bull", "luba_osalist_tarnimist": true}');
- /*Leian sellised tellimused, mis lubavad osalist tarnimist*/
- SELECT * FROM Tellimus WHERE tellimus_dok->>"luba_osalist_tarnimist" = 'true';
- JSON (JavaScript Object Notation) on avatud standardil põhinev formaat andmeobjektide inimestele (ja tarkvarale) loetaval kujul esitamiseks.
- Näide PostgreSQL (10) andmebaasisüsteemis.

10.01.2018

Teema 9

112

Näide
PostgreSQL
10

OldSQL annab vastulöögi – agregaatide moodustamine

- `SELECT row_to_json(bar) AS tulemus FROM (SELECT isikukood, eesnimi, perenimi, synni_aeg, on_aktuaalne, telefon, (SELECT array_to_json(array_agg(row_to_json(foo))) FROM (SELECT koristamine_id, alguse_aeg, ruumi_number, lopu_aeg, kommentaar, registreerimise_aeg FROM Koristamine AS Ko WHERE K.isikukood=Ko.isikukood ORDER BY alguse_aeg ASC) AS foo) AS koristamised FROM Koristaja AS K) AS bar:`

[illegible]

10.01.2018

Teema 9

113

Sulgudes on muutus eelmisest aastast

Andmebaasisüsteemide populaarsuse indeks (<https://db-engines.com/en/ranking>) (6. detsember 2017)

Tüüp	Kõrgeim koht pingereas	Andmebaasisüsteem kõrgeimal kohal	Süsteemide koguarv pingereas
Võtme-väärtuse paarid (NoSQL)	8 (+1)	Redis	49 (+3)
Dokumendid (NoSQL)	5 (0)	MongoDB	39 (+1)
Verguperekonnad (NoSQL)	9 (-2)	Cassandra	10 (+3)
Graafid (NoSQL)	21 (0)	Neo4j	27 (+5)
SQL	1	Oracle	126 (+4) (k.a NewSQL)
OO	75 (+7)	Caché	17 (0)
XML	36 (-3)	MarkLogic	8 (+1)

10.01.2018

Teema 9

114

Andmebaasid II 2017 © Erki Eessaar

Shpychka, V., 2017. Jõudluse analüüs ning veebirakenduse loomise keerukus Couchbase ja JSONB tüübiga PostgreSQL põhjal. Bakalaureusetöö. TTÜ Tarkvarateaduse instituut [WWW] <https://digi.lib.ttu.ee/i/?8244>

NoSQL pole imevits

Shpychka töö **oluline piirang** – kõik andmed olid ühes serveris (mitte jaotatud serverite kobaras vahel).

- Kuigi suurlaudade teemal (kus NoSQL annab väidetavalt eelise) on palju **haipi**, on suur hulk andmekogusid ikkagi **väikeandmed**.
- Ka NoSQL süsteemid pole jõudluse **imevits** (enne valimist tuleb ise testida).
 - "An initial expectation was that Couchbase will show better performance with large data, although the experiment showed that I was mistaken. Couchbase performance is much lower compared to PostgreSQL performance on any data size and any operation. During the experiment, I discovered problems related to the indexing in Couchbase." (Shpychka, 2017) (PostgreSQL 9.6 and Couchbase community edition 4.5.1)

10.01.2018 Teema 9 115

Andmebaasid II 2017 © Erki Eessaar

SQLi tulevik

- Tulevikus sulavad SQL-andmebaasisüsteemid ja dokumendipõhised süsteemid ilmselt kokku.
 - Vt näitena PostgreSQLi JSON tugi.
 - SQL:2016 lisas standardisse JSON toe.
 - Sama on varem juhtunud XMLiga.
 - SQL standard ja SQL-andmebaasisüsteemid lisasid XML toe ning XML-andmebaasisüsteemid muutusid nišitooteks.
 - SQL ja andmete salvestamine serverite klastriks pole üksteist välistavad.
 - <https://code.facebook.com/posts/190251048047090/myrocks-a-space-and-write-optimized-mysql-database/>

10.01.2018 Teema 9 116

Andmebaasid II 2017 © Erki Eessaar

Andmete tähtsus


- Andmetel ja nendega tehtavatel tegevustel on erinevad **tähtsusklassid**.
 - Kui andmeid on hästi palju, need pole eriti tähtsad ja soov oleks need kiiresti kuhugi ladustada, võib klassikaline NoSQL süsteem omal kohal.
 - Kui tuleb jälgida andmete vastavust reeglitele ja transaktsioonide lõdvenenud nõuded tekitaksid kahju, siis on paremad SQL-süsteemid.

10.01.2018 Teema 9 117

Andmebaasid II 2017 © Erki Eessaar

Mikk, S.M., 2017. Graafide esitamine SQL-andmebaasides. Magistritöö. TTÜ Tarkvarateaduse instituut [WWW] <https://digi.lib.ttu.ee/i/?7979>

Graafipõhised andmebaasid



- Kuigi ka relatsioonilistes/SQL-andmebaasides saab esitada graafide andmeid ning nende põhjal otsinguid teha on **eraldi** süsteemidena **suurim perspektiiv graafipõhistel** süsteemidel.
 - Paljud (aga mitte kõik!) ülesanded on taandatavad tööle graafidega.
 - Sobilikud tööks andmetega võrkude/võrgustike kohta.

10.01.2018 Teema 9 118

Andmebaasid II 2017 © Erki Eessaar

Graafiandmed SQL-andmebaasis

- Mikk (2017) toob **mustite** vormis välja **16** andmebaasi disaini graafide esitamiseks SQL-andmebaasides, lisaks veel kaks disaini mis olid sobilikud ainult graafi erijuhu – puu – andmebaasis esitamiseks.
- SQL-andmebaasi disaini valik peaks sõltuma:
 - andmetega tehtavatest operatsioonidest,
 - salvestatavate graafide omadustest (tsükliline, suunatud, suunamata, multigraaf, kaalutud).

10.01.2018 Teema 9 119

Andmebaasid II 2017 © Erki Eessaar

Ajaseeriade andmebaasisüsteemid

11.12.2017 12:34	27
11.12.2017 12:36	25,5
11.12.2017 12:38	27,1
11.12.2017 12:40	26,95
11.12.2017 12:42	25,3
11.12.2017 12:44	25,7
11.12.2017 12:46	26,45
11.12.2017 12:48	27,1
11.12.2017 12:50	27

- Ajaseeriade andmete näited.
 - Mõõtmistulemused (sensorid, asjade internet).
 - Tehingute tulemused (automaatne kauplemine börsidel).
 - Kasutaja käitumine veebis (mis hetkel kuhu klõpsas).

Andmehulgad suured ja kiiresti kasvavad. Andmed tuleb ära panna ja hiljem analüüsida

10.01.2018 Teema 9 120

Ajaseeriade andmebaasisüsteemid (2)

- Spetsialiseerunud andmebaasisüsteemid ajaseeriatega töötamiseks.
- Põhinevad ühel või mitmel andmemudelil.
 - Võti-väärtus paaride mudel.
 - OpenTSDB
 - Dokumendipõhine andmemudel.
 - InfluxDB
 - SQL aluseks olev andmemudel.
 - Timescale

Loengud erinevate süsteemide kohta:
<http://db.cs.cmu.edu/seminar2017/>

10.01.2018

Teema 9

121

Õppejõu meelisvisioon tuleviku andmebaasisüsteemidest

- Andmebaasisüsteemi välisel ja kontseptuaalsel tasemel *relatsioonilise mudeli täielik* realisatsioon (lisaks läbimõeldud ja kasutajasõbralik uute tüüpide ja operaatorite loomise võimalus).
- SQLi asendamine paremini disainitud ja kasutajasõbralikumaks andmebaasikeeleks.

10.01.2018

Teema 9

122

Õppejõu meelisvisioon tuleviku andmebaasisüsteemidest (2)

- Andmebaasisüsteemi sisemisel tasemel rakendada jõudluse ja transaktsioonide läbilaskevõime suurendamiseks *NewSQL* süsteemide tehnilisi lahendusi.
- Sellist süsteemi võiks nimetada **relatsiooniliseks NoSQL süsteemiks**.

10.01.2018

Teema 9

123

Lisalugemist

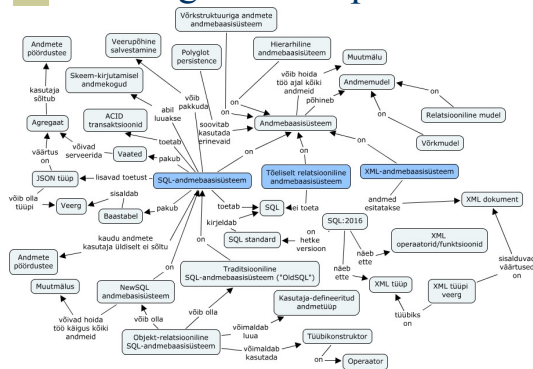
- Detailne nägemus selle kohta, milliseid võimalusi peaks pakkuma tõeliselt relatsiooniline andmebaasisüsteem.
- Date, C.J., Darwen, H. (2007). *Databases, types, and the relational model*. 3rd ed. China Machine Press.
- http://www.estor.ee/record=b2287548*est

10.01.2018

Teema 9

124

Mõningad teema põhimõisted

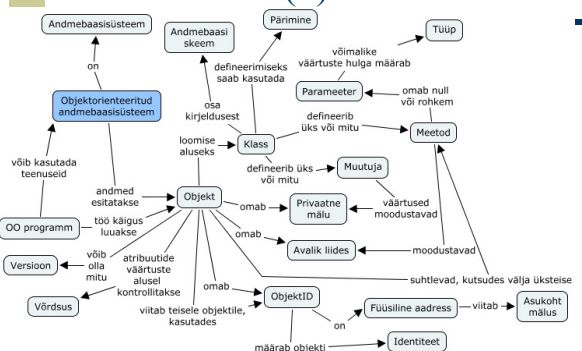


10.01.2018

Teema 9

125

Mõningad teema põhimõisted (2)

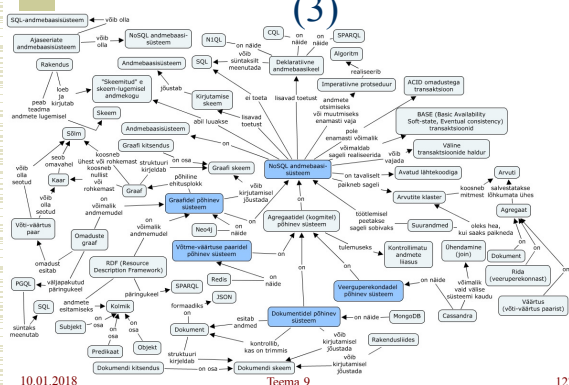


10.01.2018

Teema 9

126

Mõningad teema põhimõisted (3)



10.01.2018

Teema 9

127