

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Infotehnoloogia eriala

Roland Roose

**MySQL, Sybase ja Oracle andmebaaside varundamine
ja taaste**

Bakalaureusetöö (6 EAP)

Juhendaja: Vambola Leping

Autor: „.....“ mai 2013

Juhendaja: „.....“ mai 2013

Lubada kaitsmisele

Professor: „.....“ mai 2013

Sisukord

Sissejuhatus	4
1. Varundamisest ja taastest.....	5
1.1 Vajalikkus.....	5
1.2 Varundamine	5
1.2.1 Täielik varukoopia ja diferentsiaalne varukoopia	6
1.2.2 Füüsiline varukoopia ja loogiline varukoopia	6
1.2.3 <i>Online</i> varukoopia ja <i>offline</i> varukoopia	6
1.2.4 Kohalik varukoopia ja kaugvarukoopia.....	7
1.2.5 Indeksitest varukoopia tegemine	7
1.3 Planeerimine	7
2. Andmebaaside juhtimissüsteemide ülevaade	9
2.1 MySQL 5.6.....	9
2.1.1 Füüsiline struktuur.....	9
2.2 Oracle Database 11g.....	10
2.2.1 Füüsiline struktuur.....	11
2.3 Sybase.....	11
2.3.1 SQL Anywhere'i füüsiline struktuur	12
3. Varundamine ja taaste	13
3.1 MySQL 5.6.....	13
3.2 Oracle Database 11g.....	15
3.3 SQL Anywhere 12.....	18
4. Võrdlus	22
4.1 Dokumentatsioon.....	22
4.1.1 MySQL 5.6.....	22
4.1.2 Oracle Database 11g.....	22
4.1.3 Sybase SQL Anywhere 12.....	23

4.2 Funktsionaalsus	23
4.2.1 MySQL 5.6	23
4.2.2 Oracle Database 11g	23
4.2.3 Sybase SQL Anywhere 12.....	24
4.3 Õppimise kurv	24
4.3.1 MySQL 5.6	24
4.3.2 Oracle Database 11g	24
4.3.3 Sybase SQL Anywhere 12.....	24
4.4 Võrdluse tulemused	25
Kokkuvõte	26
Backing up and restoring MySQL, Sybase and Oracle databases.....	27
Viited	28

Sissejuhatus

Andmebaaside varundamisel ning taastamisel on oluline roll andmebaasis olevate andmete terviklikkuse ning kättesaadavuse tagamisel. Ilma hea varundamise ning taastamise plaanita võib tehniliste rikete, tarkvaraliste vigade või kasutajate eksimise tõttu kustunud või valeks muudetud andmete taastamine nõuda väga palju aega ning vaeva või olla üldse võimatu. Hea varundamise ning taastamise plaani loomiseks on oluline teada, milliseid võimalusi kasutatav andmebaaside juhtimissüsteem pakub.

Käesoleva bakalaureusetöö eesmärgiks on anda ülevaade kolme erineva andmebaaside juhtimissüsteemi Sybase SQL Anywhere 12, Oracle 11g ning MySQL 5.6 poolt andmebaaside varundamiseks ning taastamiseks pakutavatest võimalustest, võrrelda neid ja luua õppematerjale, mida oleks võimalik kasutada andmebaaside kursuse praktikumis. Kõikidest mainitud andmebaaside juhtimissüsteemidest on erinevaid versioone, kuid kuna enamik varundamise ning taastamise võimalused nendel versioonidel kattuvad, siis on erisuste korral andmebaaside juhtimissüsteemi versioon eraldi välja toodud. MySQL 5.6 korral on arvestatud, et kasutatakse vaikimisi määratud salvestusmootorit (*storage engine*) InnoDB.

Töö koosneb neljast peatükist. Esimeses peatükis kirjeldatakse andmebaaside varundamise ning taastamise vajalikkust, tuuakse välja erinevad varunduste liigid ning antakse ülevaade aspektidest, mida on vaja varundamise ning taastamise planeerimisel silmas pidada. Teises peatükis kirjeldatakse lühidalt erinevaid andmebaaside juhtimissüsteeme ning nende füüsilist struktuuri, mis aitab mõista millised failid varundamise ning taastamisega üldse seotud on. Kolmandas peatükis on toodud ülevaade võrreldavate andmebaaside juhtimissüsteemide varundamise ning taastamise võimalustest. Neljandas peatükis võrreldakse varukoopiate tegemise ning nende taastamiseks kasutamise võimalusi ja mugavust ning antakse andmebaaside juhtimissüsteemidele hinnang. Töö käigus loodud õppematerjalid asuvad tööga kaasas oleval CD-l.

1. Varundamisest ja taastest

1.1 Vajalikkus

Andmebaasi administraatori kõige tähtsamaks ülesandeks on andmete terviklikkuse ja kättesaadavuse tagamine, mille tõttu peab ta alati olema valmis taastama mingisuguse tõrke tõttu rikunud andmed. Mida paremini on varundamise ja taastamise plaan läbi mõeldud, seda kergem on andmebaasi administraatoril vajaduse korral efektiivselt reageerida. [1]

Tõrked tekivad süsteemis mitmetel põhjustel. Kõiki neid põhjuseid peaks varundamise ja taastamise planeerimisel arvestama. Kindlasti tuleb üritada tõrkeid ennetada, kasutades UPS-i või ketaste peegeldamist, kuid siiski tekivad ettenägematud vead, mille tõttu ei ole võimalik tagada andmete terviklikkust ja/või kättesaadavust. Kõik võimalikud andmebaasi tõrked, mille tagajärjel võib olla vajalik andmete taastamine, saab jagada kolme kategooriasse.

1. Süsteemitarkvara tõrked (*instance failures*) on operatsioonisüsteemi, andmebaaside juhtimissüsteemi (*database management system*) või mingi muu programmi vead. Seda tüüpi tõrked tavaliselt andmeid ei riku ning vajalik on ainult ABJS-i taaskäivitamine. Võib aga juhtuda, et andmed saavad kahjustada ning sellisel juhul on need vaja taastada.
2. Rakenduste (või transaktsioonide) tõrked (*application failures*) tekivad kui programme või skripte jooksutatakse valel ajal, vale sisendiga või vales järjekorras. Seda tüüpi tõrgete tulemusena tuleb andmete terviklikkuse taastamiseks enamasti andmebaas taastada. Mida kiiremini rakenduste tõrked tuvastatakse, seda vähem andmeid kahjustatakse.
3. Seadmete tõrked (*media failures*) tekivad kui kõvaketas või mälu on kahjustunud, failisüsteemis tekib viga või andmebaasi andmefailid on kustutatud. Selle tulemusena ei ole andmed enam kättesaadavad või terviklikud. [1]

1.2 Varundamine

Andmebaasi varundamise ja taastamise plaani fundamentaalseks osaks on andmetest varukoopiate tegemine. Varukoopiaid on mitut tüüpi: täielikud või diferentsiaalsed varukoopiad (*incremental backup* või *differential backup*), füüsilised või loogilised varukoopiad, *online* või *offline* varukoopiad, kohalikud varukoopiad või kaugvarukoopiad. Andmebaaside juhtimissüsteem võib varukoopiate tegemiseks kasutada operatsioonisüsteemi failisüsteemi käske. [1]

1.2.1 Täielik varukoopia ja diferentsiaalne varukoopia

Täielik varukoopia on andmebaasi täpne kujutis (*image*) mingist ajahetkest ehk kui tekib tõrge, mille tulemusena andmed ei ole enam terviklikud, on võimalik andmebaas tänu täielikule varukoopia le viia täpselt sellisesse seisusse, millises see oli varukoopia tegemise hetkel. Selle jaoks peab varukoopia sisaldama kõiki faile, mida võib vaja minna andmebaasi taastamiseks (andmefailid, logifailid, konfiguratsioonifailid). [1]

Diferentsiaalne varukoopia sisaldab ainult neid andmeid, mis on muutunud pärast viimast täieliku või diferentsiaalse varukoopia tegemist. Selle tõttu on diferentsiaalse varukoopia maht väiksem ning koostamine kiirem kui täieliku varukoopia puhul. Seda tüüpi varukoopia annab märgatava võidu kiiruses ja mahus juhul kui andmebaas on suur, kuid muudatusi tehakse vähe. Taastamine on diferentsiaalset varukoopiat kasutades aeglasem, kuna võib juhtuda, et ühte rida tuleb mitu korda uuendada. [1]

Mõned andmebaaside juhtimissüsteemid lubavad pärast varukoopia tegemist ühendada mitu diferentsiaalset varukoopiat või diferentsiaalsed varukoopiad ja täieliku varukoopia üheks varukoopiaks. Selle tulemusena muutub andmebaasi taastamine kiiremaks, kuna ridasid ei tule enam mitu korda uuendada. Varukoopiate ühendamine tuleb kindlasti ära teha enne kui tekib vajadus andmebaasi taastamiseks, muidu kiirusevõitu ei saavutata. [1]

1.2.2 Füüsiline varukoopia ja loogiline varukoopia

Füüsiline varukoopia koosneb andmebaasi tööks vajalikke andmeid sisaldavate kaustade ja failide koopiast. Selline varundamise viis sobib suurte ja tähtsaid andmeid sisaldavate andmebaaside jaoks, kuna tõrke korral on füüsilisest varukoopiast taastamine kiirem kui loogilisest. Samuti on füüsilise varukoopia tegemine kiirem kui loogilise varukoopia tegemine. [2]

Loogiline varukoopia sisaldab andmebaasi taastamiseks vajalikke `CREATE_DATABASE`, `CREATE_TABLE` ja `INSERT` päringuid, kuid ei sisalda logifaile, konfiguratsioonifaile ega muid andmebaasiga seotud faile. Selline varukoopia tegemise viis sobib väiksemate tabelite korral ning juhtudel, kus andmebaas on vaja taastada mõnes teises masinas. Loogilise varukoopia puhul on võimalik varukoopias olevaid andmeid või tabeleid enne taastamist muuta. [2]

1.2.3 Online varukoopia ja offline varukoopia

Online varukoopia ehk kuum varukoopia tehakse andmebaasi serveri töötamise ajal ning *offline* varukoopia ehk külm varukoopia siis, kui andmebaasi server ei tööta. Sooja

varukoopia korral server küll töötab, kuid tabelitesse ei saa andmeid lisada ega tabelites olevaid andmeid muuta. [2]

Online varukoopia tegemine on enamasti kasutajate jaoks vähem häirivam, kuna olenevalt varukoopia tegemise ajaks tabelile määratud piiragutest on neil siiski võimalik andmetele ligi pääseda. Administraator peab jälgima, et vajalikud tabelid oleksid lukustatud, nii et varukoopia tegemise ajal ei tehtaks ühtegi muutust. Mõned andmebaaside juhtimissüsteemid teevad seda automaatselt. [2]

Offline varukoopia tegemine on lihtsam, kuna andmed ei saa varukoopia tegemise ajal muutuda, kuid selline viis võib olla ebameeldivam kasutajatele, kes samal ajal andmetele ligi pääseda sooviks. Kui andmed peegeldatakse pidevalt ka teise andmebaasi serverisse, on võimalus teine server välja lülitada ning teha varukoopia seal olevatest andmetest, nii et kasutajad ei märka midagi. [2]

1.2.4 Kohalik varukoopia ja kaugvarukoopia

Nagu nimigi ütleb, saab kohaliku varukoopia teha ainult masinast, milles andmebaasi server käib ning kaugvarukoopia eemalasuvast masinast. Mõnel juhul saab varukoopia tegemise küll käivitada eemal asuvast masinast, kuid varukoopia tehakse siiski masinasse, milles töötab andmebaasi server. [2]

1.2.5 Indeksitest varukoopia tegemine

Mõned andmebaaside juhtimissüsteemid lubavad teha varukoopia indeksitest. Mõnede puhul on indeksitest koopiade tegemine kohustuslik ning mõned ei võimalda seda üldse. Kui indeksite varundamine on valikuline, peab andmebaasi administraator otsustama, kas ehitada indeksid uuesti või taastada varukoopiast. Indeksitest varukoopia tegemine annab ajalise võidu suurte andmebaaside taastamise korral, kuid varukoopia tegemine võtab kauem aega. [1]

1.3 Planeerimine

Andmebaasi administraator peab otsustama kui tihti ja milliseid varukoopiaid teha, kuidas neid teha ja nendest taastada ning mitu varukoopiat alles hoida. Selle jaoks tuleb arvesse võtta mitmeid tegureid nagu varundamiseks ja taastamiseks kuluv aeg, andmete olulisus ning andmete maht. [1]

Aja arvestamise juures peab teadma kui suur hulk andmeid varundatakse, kas andmed on krüpteeritud või pakitud, kui kaua läheb aega, et teises seadmes oleval varukoopiad

taastatava andmebaasiga masinasse toimetada ning taastamise ja varundamisega seotud andmebaaside juhtimissüsteemi iseärasusi. Mida sagedamini teha täielikke varukoopiaid, seda kiiremini toimub enamasti ka taastamine [1]

Varukoopiate säilitamisega seoses tasub arvestada, et varukoopia tegemisel võib sisse tulla mingi viga, mille tõttu sellest taastamine ei õnnestu. Seega peaks alles hoidma vähemalt kaks varukoopiat — kui viimasest taastamine ei õnnestu, saab kasutada eelviimast. [1]

Kõik andmed ei ole võrdselt tähtsad. Mõned andmebaasid ja tabelid sisaldavad andmeid, mis on ettevõtte efektiivseks tööks väga olulised. Teised andmebaasi objektid sisaldavad andmeid, mis ei ole nii tähtsad või mida saab erinevaid allikaid kasutades lihtsasti taastada. Andmete olulisuse mõistmiseks peaks vastama järgmistele küsimustele.

- Kui tihti andmeid päritakse?
- Kui tihti andmeid muudetakse?
- Kui olulised on andmed ettevõttele?
- Kas andmeid on lihtne uuesti luua?
- Mis aegadel andmetele juurdepääsu vaja on? Kas andmetele on vaja ööpäeva jooksul pidevat juurdepääsu?
- Kui palju läheb maksma andmebaasi töö seiskumine? [1]

2. Andmebaaside juhtimissüsteemide ülevaade

2.1 MySQL 5.6

MySQL on populaarne avatud lähtekoodiga relatsiooniliste SQL andmebaaside juhtimissüsteem, mida arendab, jagab ning millele pakub tuge Oracle Corporation. MySQL-i populaarsuse põhjusteks on selle kiirus, töökindlus, skaleeruvus ning lihtne kasutamine. MySQL Server võib töötada laua- või sülearvutis ilma kasutajat ja teisi töötavaid programme segamata või privaatserveris, kus seda on võimalik seadistada kasutama kogu mälu ning protsessori võimsust. Hea kiiruse, turvalisuse ning ühilduvuse tõttu sobib MySQL väga hästi olukordadesse, kus on vaja andmebaasile interneti kaudu ligi pääseda. [3]

MySQL 5.6 toetab nelja erinevat andmebaaside juhtimissüsteemi.

- Enterprise Edition on kõige skaleeravam, turvalisem ja töökindlam MySQL andmebaaside juhtimissüsteem. [4]
- Standard Edition on optimeeritud transaktsioonide jaoks. [5]
- Classic Editionit kasutatakse manussüsteemides, kus on vaja andmeid palju ja tihti lugeda. Kasutab ainult MyISAM salvestusmootorit. [6]
- Community Edition on vabavaraline avatud lähtekoodiga andmebaaside juhtimissüsteem. [7]

2.1.1 Füüsiline struktuur

MySQL 5.6 füüsiline struktuur koosneb meta-andmete failidest, logifailidest, andmefailidest ning kui andmebaaside administraator on sisse lülitanud binaarse logi salvestamise, siis ka binaarsetest logifailidest.

Iga tabeli meta-andmeid, nagu tabelite definitsioonid, salvestatakse eraldi .frm. laiendiga failidesse. Kui MySQL kustutab tabeli, siis kustutatakse ka vastav .frm fail või vastavad .frm failid. Tabeleid ei saa liigutada ühest andmebaasist teise lihtsalt .frm faile liigutades. [8]

InnoDB tabelite andmed, indeksid, tagasivõttelogi (*undo log*) ja muutuste puhver kirjutatakse ibdata failidesse. Andmebaaside administraator saab määrata ühe ibdata faili maksimaalse mahu, mille täitumisel luuakse uus andmefail ning suurendatakse loendurit (näiteks ibdata1, ibdata2 jne). Laiendit ibdata failidel ei ole. Kui tabeli loomisel on sisse lülitatud säte `innodb_file_per_table`, ei salvestata tabelite andmeid ibdata faili vaid iga tabeli andmed ning indeksid salvestatakse eraldi tabeliruumifaili laiendiga .ibd. [9]

Kõik päringud, mis üritavad teha muutusi tabelitele, salvestatakse `ib_logfile` faili ning binaarsesse logifaili. Nagu `ibdata` failide korralgi, saab andmebaaside administraator määrata `ib_logfile` faili maksimaalse mahu, mille täitumisel luuakse uus logifail ning suurendatakse loendurit (näiteks `ib_logfile0`, `ib_logfile1` jne). Kui transaktsiooni täitmise ajal tekib tõrge, mille tulemusena andmebaasi server peatub ning transaktsiooni ei saa lõpuni täita, viiakse transaktsioon serveri käivitamisel `ib_logfile`'ide abil lõpuni. Binaarsed logifailid võimaldavad diferentsiaalsete varukoopiate tegemise. [9]

2.2 Oracle Database 11g

Oracle pakub nelja erinevat andmebaaside juhtimissüsteemi: Express Edition, Standard Edition One, Standard Edition ning Enterprise Edition.

Oracle Database 11g Express Edition on tasuta saadaval andmebaaside juhtimissüsteem, mis on piiratud salvestama kõige rohkem 11 GB-i andmeid, kasutama kõige rohkem 1 GB-i mälu ja kasutama ainult ühte protsessorit. Express Edition on mõeldud eelkõige:

- arendajatele, kes kasutavad PHP-d, Javat, .NET-i, XML-i või töötavad mõne avatud lähtekoodiga rakenduse kallal;
- andmebaasi administraatoritele, kellel on mingil põhjusel, näiteks koolituse eesmärgil, vaja kasutada tasuta andmebaaside juhtimissüsteemi;
- haridusasutustel ning õpilastel, kellel on vaja seda kasutada andmebaaside tundma õppimiseks. [10]

Oracle Database 11g Standard Edition One on optimeeritud väikeettevõtetele ning Standard Edition keskmise suurusega ettevõtetele. Mõlemaid on lihtne installeerida ning seadistada, mõlemad sobivad kõikide andmetüüpide hoidmiseks, kõikide rakenduste jaoks ning võimaldavad vajadusel lihtsasti installeerida Oracle Database 11g andmebaaside juhtimissüsteemi, mis on mõeldud suurematele ettevõtetele. Erinevuseks on see, et Standard Edition One töötab ainult ühes masinas, aga Standard Edition võib töötada ka klastris. [11]
[12]

Oracle Database 11g Enterprise Edition on suurettevõtetele optimeeritud suure jõudlusega, skaleeruv, turvaline ja töökindel andmebaaside juhtimissüsteem. Enterprise Edition kaitseb nii serveripoolsete vigade kui ka võimalike inimese poolt põhjustatud vigade eest ning lühendab hooldusaega, mil andmebaasi server ei tööta.

2.2.1 Füüsiline struktuur

Oracle andmebaasi füüsiline struktuur koosneb andmefailidest (*data file*), ajutistest andmefailidest (*temp file*), kontrollfailidest (*control file*) ning taastelogi failidest (*redo log file*). [13]

Andmefail on Oracle andmebaasi poolt loodud kettal asuv füüsiline fail, mis hoiab informatsiooni andmestruktuuride (näiteks tabelid, indeksid) kohta. Ajutine andmefail hoiab, nagu nimigi viitab, ajutisi andmeid ning kuulub ajutisse tabeliruumi. [13]

Kontrollfailis hoiab Oracle andmebaas infot füüsilise struktuuri kohta. Lisaks füüsiliste failide nimedele ja asukohtadele salvestatakse siia ka andmebaasi loomise kuupäev, logi järjekorra number ning taastelogist taastamiseks vajalik info. [13]

Taastelogi failidesse kirjutatakse kõik andme- ja kontrollfailides toimunud muutused. Pärast muudatust salvestab Oracle andmebaas taastekirje puhvrissse ning spetsiaalne programm LGWR kirjutab muutused puhvrists taastelogi failidesse. [13]

2.3 Sybase

Sybase'i relatsiooniliste SQL andmebaaside juhtimissüsteemid soodustavad andmebaasi-põhiste rakenduste arendamist andmete haldamiseks ja ettevõtetele andmevahetuseks mõeldud vahenditega. Need töötavad nii suure jõudlusega andmebaaside serverites kui ka mobiilsetes seadmetes ning võimaldavad turvaliste andmevahetustehnoloogiate kasutamist mitteturvaliste juhtmeta või juhtmega ühenduste kaudu. [14] Sybase'il on kolm erinevat andmebaaside juhtimissüsteemi.

- SAP Sybase SQL Anywhere sobib kõige paremini mobiilsete seadmete jaoks, mis peavad andmebaasiga juhtmevabalt suhtlema. Kasutajad pääsevad lihtsalt ja turvaliselt ligi vajalikele andmetele ning kui seadmed pole võrku ühendatud, saab transaktsioone panna järjekorda, mis täidetakse pärast seadme võrku ühendamist. [15]
- SAP Sybase Adaptive Server Enterprise on suure jõudlusega relatsiooniliste andmebaaside juhtimissüsteem, mis on mõeldud ülesande-kriitilistesse (*mission-critical*) ja suurte andmemahutudega töötavatesse keskkondadesse. [16]
- SAP Sybase Advantage Database Server on klient-server mudeli põhjal töötav relatsiooniliste andmebaaside juhtimissüsteem, mis võimaldab andmete kätte saamiseks kasutada ISAM meetodit. [17]

2.3.1 SQL Anywhere'i füüsiline struktuur

Kõik andmed salvestatakse tavaliselt ühte .db laiendiga andmefaili, mida saab kopeerida ühest arvutist teise. Väga suurte andmebaaside korral on ka võimalus, et andmete salvestamiseks tehakse mitu faili. Nendes failides hoitakse nii tabelite andmeid ja indekseid kui ka muud infot nagu info andmete jagunemise kohta andmebaasis. [18]

Lisaks andmebaasi failidele, kasutab SQL Anywhere transaktsioonide logi faili ning ühte ajutist faili. Transaktsioonide logi failis on kirjas kõik operatsioonid, mis andmebaasi peal tehtud on. Selletõttu on võimalik taastada seda faili kasutades andmebaas või sünkroniseerida andmebaas teiste andmebaasidega SQL Remote'i või MobiLinki abil. Logifail aitab tõsta ka andmebaasi jõudlust, kuna kirjutades informatsiooni transaktsioonide logisse, saab server turvaliselt päringuid töödelda ning ei pea nii sagedasti andmebaasi faili kirjutama. Tavaliselt on logi failil sama nimi, mis andmefailil, kuid .db laiendi asemel kasutatakse .log laiendit. [18]

Ajutine fail luuakse, kui andmebaasi server käivitub ning kustutatakse, kui selle töö lõpetatakse. Ajutises failis ei hoita informatsiooni, mis peaks alles jääma pärast serveri taaskäivitamist, vaid ainult infot, mida on vaja serveri töötamise ajal. [18]

3. Varundamine ja taaste

3.1 MySQL 5.6

Üheks võimaluseks InnoDB tabelitega andmebaasi varundamiseks, on teha koopia kõikidest failidest, mida salvestusmootor kasutab. Selle jaoks tuleb esmalt MySQL server aeglaselt peatada (*slow shutdown*), mille käigus kirjutatakse kõik vajalikud andmed mälust andmefailidesse. Vaikimisi kirjutab server seiskudes vahemällu salvestatud andmed andmefailidesse alles järgmisel käivitamisel. Koopia tuleb teha tabeliruumi failidest, milles asuvad tabelid ja indeksid (ibdata failid), tabeliruumi failidest, mis sisaldavad ühte tabelit ja selle indeksid (.ibd laiendiga failid), failidest, mis hoiavad tabelite meta-andmeid (.frm laiendiga failid), logifailidest (ib_logfile failid) ja MySQL serveri seadete failist (my.cnf või my.ini). [19]

Kui muutuja `innodb_file_per_table` väärtus on 0, salvestatakse kõikide andmebaaside tabelite andmed kokku ühte või mitmesse faili, mille tõttu on ainukeseks võimaluseks teha täielik varukoopia kõikide andmebaaside kõikidest tabelitest. Kui muutuja väärtus on 1, siis on võimalik varundada ainult üks tabel või andmebaas. Tabelite varundamise korral on neid võimalik taastada ainult juhul, kui pärast varundamist ei ole neid tabelleid andmebaasist ära kustutatud (ei ole kasutatud `DROP TABLE`, `TRUNCATE TABLE` või `DROP DATABASE` päringuid). [20]

MySQL pakub võimalust teha andmebaasi tõmmis, kasutades varundamise programmi `mysqldump`. Programmi väljundiks on SQL päringud, CSV formaadis andmed või XML formaadis andmed, mille abil on võimalik taastada kas andmebaasi objektid, tabelite andmed või mõlemad. Selleks, et tõmmis oleks vastavuses andmetega, mis olid andmebaasis tõmmise tegemise hetkel, tuleb enne tõmmise tegemist keelata tabelitesse andmete lisamine ja tabelis andmete muutmine. Selle jaoks võib kasutada käsku `LOCK TABLES` või käivitada programmi parameetriga `--single-transaction` või `--lock-tables`. [21]

Tõmmise tegemise eeliseks on selle arusaadavus kasutajatele. Enne taastamist on võimalik fail avada mõne tekstiredaktoriga ning see üle vaadata või teha muudatusi. Nõrgaks küljeks on selle kiirus. Kui tegemist on suure andmehulgaga, võtab tõmmise tegemine palju rohkem aega, kui andmefailide kopeerimine. Andmebaasi taastamine tõmmisest on kõikide päringute täitmise ning indeksite uuesti loomise pärast veelgi aega nõudvam kui selle tegemine. [21]

Varundamise programmi mysqldump loodud tõmmise taastamiseks tuleb see lihtsalt mysql-ile sisendiks anda. Kui tõmmise tegemisel kasutati parameetreid `--all-databases` või `--databases`, on tõmmises juba kirjas andmebaasi loomis- (`CREATE DATABASE`) ja valimispäring (`USE`), mille tõttu ei ole vaja isegi täpsustada andmebaasi, millesse andmed panna:

```
shell> mysql < dump.sql
või
mysql> source dump.sql.
```

Kui tõmmis ei sisalda andmebaasi loomis- ja valimispäringut, tuleb andmebaas enne luua ning valida.

MySQL toetab ka diferentsiaalset varundamist. Selle jaoks tuleb käivitada server parameetriga `--log-bin` või lisada konfiguratsiooni faili mysqld muutujate alla `log-bin=mysql-bin`, mille tulemusena hakatakse salvestama binaarseid logisid, kus hoitakse igat andmebaasile tehtud muutust. Enne binaarsetest logidest koopia tegemist, tuleb veenduda, et ka kõik andmed, mis vahemälus on, logisse kirjutataks. Selle jaoks tuleb kasutada käsku `FLUSH LOGS`. [22]

Binaarsetest logidest taastamiseks saab kasutada programmi `mysqlbinlog`, mis teisendab logi failid binaarsest formaadist tekstiks nii, et logi saab vaadata, muuta ja taastada. Programm võimaldab taastada kõiki logis olevaid sündmusi, mingil ajavahemikul tehtud sündmusi või valida taastatavad sündmused logis asumise positsiooni järgi. Mitme logi faili korral peaks taastamise ebaõnnestumise vältimiseks kasutama faili, kuhu on kõik logi failid järjestikku kokku kirjutatud. [23]

Taastamine mitut binaarset logifaili kasutades:

```
shell> mysqlbinlog binlog.000001 > /tmp/statements.sql
shell> mysqlbinlog binlog.000002 >> /tmp/statements.sql
shell> mysql -u root -p -e "source /tmp/statements.sql"
```

Andmete varundamiseks on võimalik kasutada ülem-alluv-süsteemi, kus kõik ülem-serveris olevad andmed peegeldatakse alluv-serverisse. Serverit, mis käitub alluvana, on võimalik välja lülitada või panna selle töö pausile nii, et ülem-serveri kasutajad midagi aru ei saaks. Seega on võimalik kasutajaid segamata teha varukoopiaid, mis muidu vajaks ülem-serveri välja lülitamist. [24]

Üheks võimaluseks alluv-serveri andmete varundamiseks on kasutada programmi mysqldump. Selle jaoks tuleb esmalt peegeldamine lõpetada:

```
shell> mysqladmin stop-slave
```

või peatada alluv-serveri SQL päringute töötlemine:

```
shell> mysql -e 'STOP SLAVE SQL_THREAD;'
```

SQL päringute töötlemise lõpetamise tulemusena saab alluv ülema binaarsest logist küll kätte muutused, mis andmetele tehtud on ning kirjutab need vahelogisse (*relay log*), kuid ei uuenda enda andmete seis. Andmed uuendatakse alles pärast päringute töötlemise lubamist. Juhul kui andmeid muudetakse tihti, on alluv-serveri andmete uuendamine pärast päringute lubamist kiirem, kui pärast peegeldamise keelamist ning seejärel lubamist. [25]

Teiseks võimaluseks on alluv-server välja lülitada ning teha füüsiline varukoopia. Kui on vajalik, et varukoopiat kasutades saaks lisaks andmetele taastada ka alluv-serveri seaded ning muud sellega seotud failid, tuleb varundada ka vahelogi failid ning master.info ja relay-log.info fail. [26]

MySQL Enterprise Editioni kasutajad saavad kasutada MySQL Enterprise Backup'i, et teha varukoopiaid kogu süsteemist või valitud andmebaasidest ja tabelitest. MySQL Enterprise Backup võimaldab nii täielike kui ka diferentsiaalsete varukoopiate tegemise kui ka nende kokkupakkimise. Kõikidest InnoDB tabelitest on võimalik teha „kuum varukoopia“ ehk varukoopia tegemine toimub andmebaasi serveri töötamise ajal. [27]

3.2 Oracle Database 11g

Põhiline programm Oracle'i andmebaaside varundamiseks on Recovery Manager (RMAN). Selle ülesandeks on teha varukoopiaid ning taastada andmebaase, hoida varundatud andmebaaside andmeid RMAN repositooriumis, kontrollida, et varundamiseks oleks piisavalt kettaruumi ning et varundatud failid oleksid korrektsed. [28]

RMAN salvestab andmebaaside varundusi kas täpse kujutisena (*image copy*), mis on bait-baidilt sama, mis originaal, või varunduse komplektina (*backup set*), mis on RMAN-i spetsiifiline formaat. [29]

Täpse kujutise tegemine on peaaegu sama, mis operatsioonisüsteemi tasemel failide kopeerimine, aga kuna RMAN saab faile taastada ainult siis, kui andmed varukoopia kohta on salvestatud RMAN repositooriumis, võiks varukoopiate salvestamine toimub RMAN-i kaudu. Kui varukoopia tegemiseks ei kasutata RMAN-i on administraatoril võimalik

andmed varukoopia kohta käsitsi sisestada. Täieliku varukoopia puhul peab olema tehtud varukoopia kõikidest andmefailidest, kontroll-failidest (*control file*), taastelogist (*redo log*) ja serveri parameetrite failist. [29]

Varunduste komplekt koosneb füüsilistest failidest, mida kutsutakse varunduse tükkiideks (*backup piece*). Iga varunduse tükk koosneb ühest või mitmest andmebaasi failist. Varunduse komplekti eeliseks on tema kettaruumi säästev pakkimismeetod. RMAN salvestab komplekti ainult need andmefailide plokid, mida reaalselt kasutatakse andmete hoidmiseks. Lisaks on võimalik kasutada binaarset pakkimist, mis küll suurendab protsessori koormust, kuid vähendab varukoopiate suurust veelgi. [29]

RMAN lubab varunduste komplekti kolmel erineval meetodil krüpteerida:

- kasutades Oracle'i rahakotti (*Oracle wallet*), mis on parooliga kaitstud privaatvõtmeid, sertifikaate ja usaldatavaid sertifikaate sisaldav konteiner;
- kasutades parooli;
- kasutades mõlemat meetodit korraga, nii et dekrüpteerimiseks on vaja kas parooli või Oracle'i rahakotti.

Krüpteeritud varukoopiad dekrüpteeritakse taastamisel automaatselt, kui vajalikud võtmed on saadaval. [30]

Diferentsiaalne varundamine teeb varukoopia ainult nendest andmefaili plokkidest, mis on muutunud pärast viimast varundamist. Diferentsiaalsel varundamisel on kaks taset: 0 ja 1. Tase 0 teeb koopia kõikidest andmefaili plokkidest ning on täpselt sama kui täielik varukoopia, kuid seda loetakse diferentsiaalse varunduse osaks. Tase 1 võib olla kumulatiivne ehk teeb varukoopia kõikidest plokkidest, mis on muutunud pärast viimast tasemel 0 tehtud varukoopiat või diferentsiaalne ehk teeb varukoopia plokkidest, mis on muutunud pärast viimast tasemel 1 või tasemel 0 tehtud varukoopiat. [29]

Andmebaasi failide taastamiseks on RMAN-is käsud `RESTORE`, mis valib õige varukoopia, ja `RECOVER`, mis teeb vajalikud uuendused taastelogidest ja diferentsiaalsetest varundustest, et andmefail oleks vastavuses soovitud ajahetkel eksisteerinud andmefailiga. Terve andmebaasi taastamiseks on käsud `RESTORE DATABASE` ja `RECOVER DATABASE`, tabeliruumi taastamiseks käsud `RESTORE TABLESPACE` ja `RECOVER TABLESPACE`, vigaste plokkide taastamiseks käsk `RECOVER CORRUPTION LIST` ja üksiku ploki taastamiseks käsk `RECOVER DATAFILE [andmefaili number] BLOCK [ploki number]`. [30]

Igal Oracle Database'i eksemplaril (*instance*) on oma taastelogi. Iga taastelogi koosneb kahest või enamast failist, mis salvestavad kõik muutused, mida andmebaasis tehakse. Taastelogi failid täidetakse taastekirjetega (*redo record* või *redo entry*), mis moodustatakse mitmest muutuse vektorist (*change vector*). Iga muutuse vektor kirjeldab ühele andmebaasi plokile tehtud muutust. Kui taastada andmebaas taastelogi abil, vaatab andmebaaside juhtimissüsteem taastekirjetes olevaid vektoreid ning teeb vastavad muutused vajalikele plokkidele, mille tulemusena taastatakse andmebaas. [31]

Oracle Flashback Technology on erinevaid võimalusi pakkuvatest vahenditest koosnev tehnoloogia, mis lubab ilma varukoopiat kasutamata vaadata andmete seis, nii nagu see oli minevikus, või taastada andmeid. Peale Flashback Technology vahendi Oracle Flashback Drop, kasutavad kõik teised vahendid andmete taastamise logi, kus on kirjas kõik andmete tehtud muutused ning välja väärtused enne muutust. Oracle Flashback Technology koosneb kuuest vahendist.

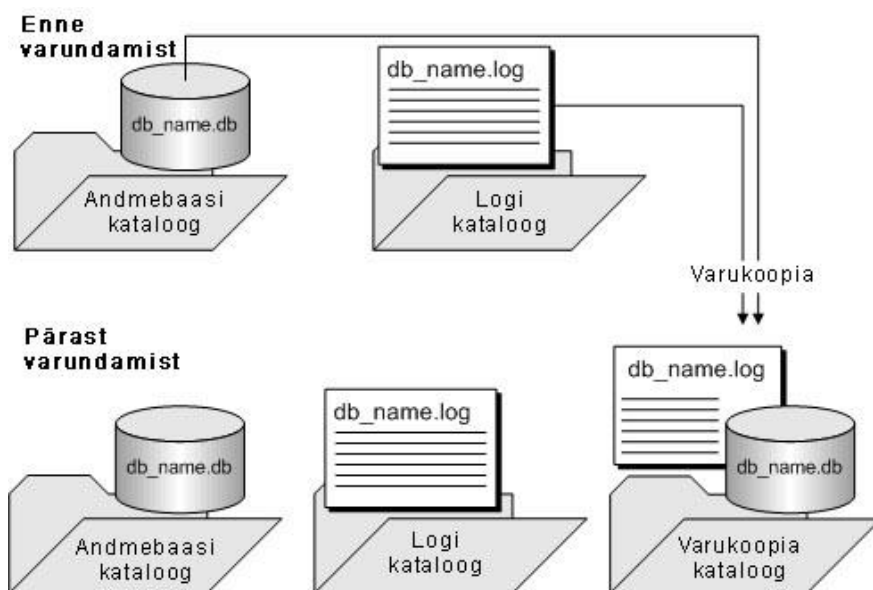
- Oracle Flashback Query võimaldab määrata aja minevikus ning kuvab vastuseid päringutele vastavalt sellele, millised olid andmed valitud ajal. Vajadusel on võimalik need andmed taastada.
- Oracle Flashback Version Query võib kõikidest ridadest kuvada kõik mingisuguses ajavahemikus eksisteerinud versioonid ning iga versiooni meta-andmed nagu alguse ja lõpu aeg ning muutuse põhjustanud transaktsiooni ID. Vajadusel on võimalik soovitud versioon taastada.
- Oracle Flashback Transaction Query näitab ühe või mitme transaktsiooni mingisugusel ajaperioodil põhjustatud muutusi.
- Oracle Flashback Transaction lubab võtta tagasi mingisuguse transaktsiooni põhjustatud muutused. Oracle Database teeb kindlaks kõik, mida see transaktsioon mõjutab ning genereerib transaktsiooni, mis muudab andmebaasi seis selliseks nagu transaktsiooni ning sellest sõltuvaid transaktsioone poleks toimunudki.
- Oracle Flashback Table lubab viia ühe või mitme tabeli andmed sellisesse seis nagu need olid kindlal hetkel minevikus, säilitades indeksid, trigerid ning piirangud, ilma et peaks andmebaasi serveri välja lülitama.
- Oracle Flashback Drop lubab tagasi võtta tabeli kustutamise päringu. [30]

RMAN-i abil saab kontrollida, et andmefailide taastamiseks vajalikud varukoopiad on olemas ning et nendest varukoopiatest taastamine ka toimiks. Valides varunduse komplekti

või varundatud täpse kujutise, on võimalik lasta RMAN-il kontrollida, kas kõik taastamiseks vajalikud failid on olemas ning kas nende failide koostamisel on tekkinud vigu või mitte. Selline kontroll ei tuvasta probleemi juhul kui andmebaasi enamike tabeliruumide jaoks on varukoopia olemas, kuid mõne jaoks mitte. Probleemi lahendab võimalus testida, kas andmebaasi jaoks on olemas vastavad varukoopia ning kas neid on võimalik ka taastada. [29]

3.3 SQL Anywhere 12

Kõige lihtsam viis SQL Anywhere andmebaasi varundamiseks on teha andmebaasi kujutisest varukoopia (*image backup*), mis koosneb andmebaasi faili varukoopiast ja/või transaktsioonide logist (Joonis 1). SQL Anywhere pakub andmebaasist kujutise loomiseks kahte võimalust: Sybase Centrali varundamise viisardi kasutamine või päringu `BACKUP DATABASE` kasutamine. Olenemata, milline võimalus varukoopia tegemiseks valitakse, peab andmebaasi server varundamise ajal töötama. Enne varukoopia tegemist kirjutatakse vahemälu olevad andmed failidesse ning keelatakse andmebaasi muutmine. [32]



Joonis 1. Andmebaasist kujutise tegemine

Varundamise automatiseerimiseks võib kasutada sündmuste (*events*) ning planeerija (*scheduler*) võimalusi. Sündmusi kasutades on võimalik defineerida mingi tegevus (näiteks diferentsiaalne varundamine või täieliku varukoopia tegemine) ning planeerija abil saab määrata, kui tihti ja mis ajal seda sündmust käivitatakse. Kuna vaikimisi salvestatakse varukoopia sama nimega, mis originaal, siis tuleb failide ülekirjutamise vältimiseks salvestada kõik failid erineva nimega või liigutada varukoopia enne uue tegemist teise

kausta. Planeerijad ning sündmused salvestatakse andmebaasi, mille kohta nad käivad, nii et pärast andmebaasi varundamist ning taastamist pole automaatse varundamise jätkamiseks kasutajapoolset pingutust vaja. Kui kõik andmebaasis tehtud muutused peegeldatakse koheselt ka mingisse teise serverisse, siis selles serveris, kuhu andmed peegeldatakse, peamises serveris defineeritud sündmust käima ei panda. [33]

Näide automaatsest diferentsiaalsest varundamisest, mis toimub iga päev kell 1:00:

```
CREATE EVENT IncrementalBackup
SCHEDULE
  START TIME '1:00 AM' EVERY 24 HOURS
HANDLER
BEGIN
  BACKUP DATABASE DIRECTORY 'c:\\backup'
  TRANSACTION LOG ONLY
  TRANSACTION LOG RENAME MATCH
END;. [33]
```

Varundamise automatiseerimiseks võib kasutada ka hooldusplaani (*maintenance plan*). Hooldusplaani täitmine käivitatakse administraatori poolt määratud ajal ning selle käigus täidetakse vähemalt üks järgmistest ülesannetest:

- kontrollitakse, et andmebaasis ei oleks vigu;
- varundatakse andmebaas;
- genereeritakse raport plaani täitmise kohta.

Hooldusplaani loomiseks saab kasutada Sybase Centrali hooldusplaani loomise viisardit. Genereeritud raport salvestatakse andmebaasi ning administraator saab seda vaadata Sybase Centralist või lasta selle automaatselt oma emailile saata. Hooldusplaani on võimalik kirjutada SQL päringuid, mida täidetakse enne või pärast varukoopia tegemist. [34]

Diferentsiaalse varundamise korral tehakse serveri käimise ajal varukoopia ainult logist. Sellist varukoopiat kasutades on võimalik andmebaas taastada ainult juhul, kui varem on samast andmebaasist tehtud täielik varukoopia. [35] Diferentsiaalseks varundamiseks on mitmeid võimalusi.

- Sybase Centrali andmebaasist kujutiste tegemise või hooldusplaani loomise viisardi kasutamine.

- SQL päringu kasutamine:

```
BACKUP DATABASE
DIRECTORY backup-directory
[ TRANSACTION LOG ONLY ]
TRANSACTION LOG TRUNCATE; .
```

- Utiliidi dbbackup käivitamine käsurealt parameetriga `-t`, mis tähendab, et varukoopia tehakse ainult transaktsioonide logist, ning parameetriga `-x`, mis tähendab, et transaktsioonide logi originaal kustutatakse ning alustatakse uue kirjutamist. [36]

Andmebaasi taastamiseks selle kujutisest tuleb andmebaasi failid kopeerida nende algsesse asukohta. [37] Diferentsiaalsete varukoopiate lisamise juures on tähtis, et lisamine toimuks varukoopiate loomise järjekorras. Diferentsiaalsetest varukoopiatest taastamiseks on mitmeid võimalusi.

- Kasutada andmebaasi serveri käivitamisel parameetreid nagu `-a`, mille abil on võimalik varukoopiaid ükshaaval lisada, `-ad`, mille abil on võimalik anda ette tehingute logifailide kataloog ning server lisab need ise õiges järjekorras ja `-ar`, mis paneb serveri logifailide asukohta andmebaasist järele vaatama ning lisab need ise õiges järjekorras.
- Teisendada transaktsioonide logifailid Sybase Centrali või utiliidi dbtran abil .sql failideks ning lisada muutused andmebaasile neid .sql faile rakendades. Utiliit dbtran võimaldab mitme transaktsiooni logifaili ühendamist üheks .sql failiks, mis lihtsustab muutuste rakendamist. [38]

Pakitud varukoopia koosneb ühest või mitmest failist, mida on vaja andmebaasi taastamiseks. Kuna pakitud varukoopiat koostades jäetakse vahele tühjad leheküljed, on selle maht väiksem ning varukoopia tegemise kiirus suurem kui täieliku koopia korral, välja arvatud juhul, kui varukoopiat tehakse transaktsioonide logist. Transaktsioonide logis ei ole tühju lehekülgi. Pakitud varukoopia tegemiseks ning taastamiseks tuleb kasutada päringut `BACKUP DATABASE` või Sybase Centrali varukoopiate tegemise viisardit. [39]

Pideva varundamise (*live backup*) korral uuendatakse pärast igit muutust lisaks kohalikus masinas olevale transaktsioonide logile ka teises masinas olevat transaktsioonide logi. Kui kohalik masin läheb katki, on tänu teises masinas olevale täielikule andmebaasi faili varukoopiale ja logile võimalik taastada andmebaasi viimane seis. Pideva varundamise

korral peab arvestama sellega, et kui peamine andmebaasi server on suure koormuse all, võib juhtuda, et kõiki transaktsioone ei jõuta teises masinas asuvasse transaktsioonide logisse kirja panna ehk kui peamine masin lõpetab töötamise, ei pruugi teisest masinast taastatud andmebaasi andmed olla täpselt samad, mis originaal andmebaasis. [40]

4. Võrdlus

Töös käsitletavate andmebaaside juhtimissüsteemide võrdlemiseks andmete varundamise ning taastamise vaatenurgast on kasutatud järgmisi dimensioone.

- Dokumentatsioon. Hinnatakse dokumentatsiooni arusaadavust, ülevaatlikkust ning seda, kui lihtne on leida otsitavat informatsiooni.
- Funktsionaalsus. Hinnatakse milliseid võimalusi pakub andmebaaside juhtimissüsteem andmete varundamiseks ning taastamiseks ja nende ülesannete täitmise mugavdamiseks.
- Õppimise kurv. Hinnatakse kui lihtne on omandada teadmised erinevatest võimalustest, mida andmebaaside juhtimissüsteem andmete varundamiseks ning taastamiseks pakub.

Iga dimensiooni hinnatakse skaalal [1, 2, 3, 4, 5], kus 1 on väga halb ja 5 väga hea. Võrreldavas dimensioonis parim andmebaaside juhtimissüsteem saab hindeks 5. Lõpuks liidetakse hinded kokku nii, et tekib paremusjärjestus.

4.1 Dokumentatsioon

4.1.1 MySQL 5.6

Dokumentatsioonis on kergesti arusaadavalt kirjeldatud varundamise ja taaste vajalikkust, erinevaid varundamise tüüpe, varundamise ja taastamise meetodeid ning nende kasutamist. See on sobilik nii algajale andmebaaside administraatorile, kes vajab üldisi teadmisi andmete varundamise ning taastamise kohta kui ka professionaalile, kes tahab teada erinevaid võimalusi, mida MySQL 5.6 pakub, ning nende kasutamise süntaksit. Info on dokumentatsioonist, tänu loogilisele struktuurile, tähenduslikele pealkirjadele ja rohketele linkidele, lihtsasti leitav. Erinevate varundamise ning taastamise võimaluste kirjelduses on toodud mitmeid koodinäiteid.

Hinnang: 5 punkti.

4.1.2 Oracle Database 11g

Oracle on varundamises ja taastamises teistest võrreldavatest andmebaaside juhtimissüsteemidest kõige rohkemate võimalustega ning selletõttu ka kõige mahukama dokumentatsiooniga. Kirjeldatud on erinevaid varundamise tüüpe, varundamise ja taastamise meetodeid ning nende kasutamist. Dokumentatsioon on küll arusaadav ka algajale, kuid nõuab rohkem eelteadmisi kui näiteks MySQL-i dokumentatsioon. Info on

tänu tähenduslikele pealkirjadele ja loogilisele struktuurile üsna lihtsasti leitav, kuid võiks olla rohkem tekstiseseid linke (Näiteks diferentsiaalset varundamist üldiselt kirjeldades võiks olla lingitud ka konkreetsemad juhendid.). Ühe lehe peal olev infohulk on liiga suur. Erinevate varundamise ning taastamise võimaluste kirjelduses on toodud mitmeid koodinäiteid.

Hinnang: 4 punkti.

4.1.3 Sybase SQL Anywhere 12

Vaadeldava Sybase'i andmebaaside juhtimissüsteemi dokumentatsioon on võrreldes teistega kõige väiksema mahuga. Lühidalt on seletatud mida ja kuidas teha saab, kuid puudub pikem mõisteid ja teguviise seletav osa. Varundamise ja taastamise kohta üldise informatsiooni saamiseks on algajale andmebaaside administraatorile paremad Oracle Database 11g ja MySQL 5.6 dokumentatsioonid. Kui algteadmised kõrvale jätta, on info leidmine tänu loogilisele struktuurile, tähenduslikele pealkirjadele ja rohketele linkidele lihtsasti leitav. Dokumentatsioonist arusaamist lihtsustavad rohked koodinäited ning joonised.

Hinnang: 4 punkti.

4.2 Funktsionaalsus

4.2.1 MySQL 5.6

MySQL võimaldab teha tõmmiseid, andmebaasi kujutisi ja diferentsiaalseid varukoopiaid ühest või mitmest andmebaasist või tabelist, kuid ei paku võimalusi nende tegevuste automatiseerimiseks.

Hinnang: 2 punkti.

4.2.2 Oracle Database 11g

Oracle võimaldab teha tõmmiseid, andmebaasi kujutisi ning pakitud ja diferentsiaalseid varukoopiaid. Kõiki neid tegevusi on võimalik automatiseerida. Lisaks teeb taastamist lihtsamaks Oracle Flashback Technology.

Oracle andmebaaside varundamiseks ja taastamiseks mõeldud programm RMAN salvestab meta-andmed varukoopiate kohta. Selle abil on võimalik vaadata iga andmebaasi kohta, millised varukoopiad ja kuna sellest tehtud on.

Kasutaja saab määrata ala kõvakettal, kuhu salvestatakse kõik varukoopiad. Seda ala jälgitakse ning hallatakse automaatselt. Näiteks saab määrata tingimused, mille täitmisel kustutatakse varukoopia kõvakettalt.

Hinnang: 5 punkti.

4.2.3 Sybase SQL Anywhere 12

Võimaldab teha tõmmiseid, andmebaasi kujutisi, pakitud ning diferentsiaalseid varukoopiaid. Andmebaasi kujutisi ning pakitud ja diferentsiaalseid varukoopiaid saab teha ainult andmebaasidest, kuid tõmmiseid ka valitud tabelitest. Kõiki neid tegevusi on võimalik automatiseerida.

Hinnang: 2 punkti.

4.3 Õppimise kurv

4.3.1 MySQL 5.6

Füüsilise varukoopia ning tõmmise tegemine ei ole väga keeruline, kuna dokumentatsioonis on täpselt ja lihtsasti kirjas, mida teha tuleb. Samuti on lihtsasti õpitav nende varukoopiate kasutamine andmebaaside taastamiseks. Diferentsiaalse varukoopia tegemise ning selle taastamiseks kasutamise juures on mitmeid nüansse, mistõttu võtab nende õppimine rohkem aega.

Hinnang: 4 punkti.

4.3.2 Oracle Database 11g

Suure funktsionaalsuse tõttu võib Oracle andmebaaside varundamise ning taastamise õppimine võtta väga palju aega, eriti kui puuduvad varasemad kogemused. Tuleb tutvuda võimalustega nagu Flashback Technology või Fast Recovery Area, mida teised võrreldavad andmebaaside juhtimissüsteemid ei kasutagi.

Hinnang: 3 punkti.

4.3.3 Sybase SQL Anywhere 12

Võrreldes teiste töös vaadeldavate andmebaaside juhtimissüsteemidega on SQL Anywhere 12 andmebaaside varundamine ning taastamine kõige lihtsamini õpitav. Iga andmebaasi kohta on enamasti ainult kaks faili: andme- ja logifail. Andmebaaside administraator valib, kas ta teeb nendest failidest lihtsalt koopia või pakib need enne kokku. Diferentsiaalse

varukoopia korral tehakse koopia logifailist, kustutatakse see ning alustatakse logi uuesti kirjutamist.

Hinnang: 5

4.4 Võrdluse tulemused

Võrdluse tulemused on väljatoodud järgnevas tabelis 1.

	MySQL 5.6	Oracle Database 11g	Sybase SQL Anywhere 12
Dokumentatsioon	5	4	4
Funktsionaalsus	2	5	2
Õppimise kurv	4	3	5
KOKKU:	11	12	11

Tabel 1. Võrdluse tulemuste tabel.

Igal andmebaaside juhtimissüsteemil on omad eelised, kuid parimaks osutus, tänu oma suurele funktsionaalsusele, Oracle Database 11g. Parima dokumentatsiooniga MySQL 5.6 ning kõige lihtsamini õpitav Sybase SQL Anywhere 12 said võrdse hinnangu.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli anda ülevaade kolme erineva andmebaaside juhtimissüsteemi Sybase SQL Anywhere 12, Oracle 11g ning MySQL 5.6 poolt andmebaaside varundamiseks ning taastamiseks pakutavatest võimalustest, võrrelda neid ja luua õppematerjale, mida oleks võimalik kasutada andmebaaside kursuse praktikumis.

Esimeses peatükis kirjeldati andmebaaside varundamise ning taastamise vajalikkust, toodi välja erinevused varukoopiate tüüpide vahel ning anti soovitusi, milliseid varukoopiaid erinevates olukordades teha.

Teine, kolmas ja neljas peatükk andsid ülevaate vastavalt MySQL 5.6, Oracle 11g ja Sybase SQL Anywhere 12 andmebaaside juhtimissüsteemide andmete varundamise ja taastamise võimalustest.

Viiendas peatükis võrreldi töös käsitletavaid andmebaaside juhtimissüsteeme andmete varundamise ja taastamise vaatenurgast ning erinevaid aspekte arvesse võttes pandi kokku paremusjärjestus. Igal andmebaaside juhtimissüsteemil oli oma tugev külg, MySQL 5.6-l dokumentatsioon, Oracle Database 11g-l funktsionaalsus ning Sybase SQL Anywhere 12-l õppimise kurv, kuid parima summaarse hinnangu saavutas Oracle Database 11g.

Töö käigus valmisid ka õppematerjalid andmebaaside kursuse praktikumide jaoks, mis on lisatud tööga kaasas olevale CD-le.

Backing up and restoring MySQL, Sybase and Oracle databases

Bachelor Thesis

Roland Roose

Summary

Backing up and restoring databases has an important part in ensuring data availability and integrity. Without having a well-planned approach to database backup and recovery it is difficult or even impossible for a database administrator to restore access to right information in case of database failure or service disruption. For building a database backup and recovery plan it is essential to have the knowledge about supported backup and restore features of a database management system.

This paper gives a brief overview of backing up and restoring databases, short description of three different database management systems – MySQL 5.6, Oracle Database 11g, Sybase SQL Anywhere 12 and their features for backing up and restoring, and comparison of these features. There are several different versions of each database management system, therefore version name is only mentioned if some feature is not supported by other versions. It is considered that MySQL 5.6 uses its default storage engine - InnoDB.

The comparison suggests that the best database management system for backing up and restoring data is Oracle Database 11g due its large functionality. MySQL 5.6 and Sybase SQL Anywhere 12 received the same amount of points, but each database management system has its strong side. MySQL 5.6 has a well-written documentation which makes finding and understanding information easier. Because of its simple physical structure and Querys, Sybase SQL Anywhere 11g is the most easy-to-learn database management system and, as pointed out, Oracle Database 11g has the largest number of features.

As an output of this thesis study materials that could be used in the databases course practicum were created. These are included on the CD.

Viited

- [1] C. S. Mullins, Database Administration: The Complete Guide to DBA Practices and Procedures, 2nd Edition toim., 2012.
- [2] „Backup and Recovery Types,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/backup-types.html>. [Kasutatud 18.04.2013].
- [3] „What is MySQL?,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html>. [Kasutatud 04.03.2013].
- [4] „MySQL Enterprise Edition,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.mysql.com/products/enterprise/>. [Kasutatud 15.04.2013].
- [5] „MySQL Standard Edition,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.mysql.com/products/standard/>. [Kasutatud 15.04.2013].
- [6] „MySQL Classic Edition,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.mysql.com/products/classic/>. [Kasutatud 15.04.2013].
- [7] „MySQL Community Edition,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.mysql.com/products/community/>. [Kasutatud 15.04.2013].
- [8] „InnoDB Table and Index Structures,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/innodb-table-and-index.html>. [Kasutatud 15.04.2013].
- [9] „MySQL Glossary,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://dev.mysql.com/doc/refman/5.6/en/glossary.html#glos_ibd_file. [Kasutatud 15.04.2013].
- [10] „Oracle Database 11g Express Edition,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.oracle.com/technetwork/products/express-edition/overview/index.html>. [Kasutatud 15.04.2013].
- [11] „Oracle Database 11g Standard Edition One,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.oracle.com/us/products/database/standard-edition-one/overview/index.html>. [Kasutatud 15.04.2013].

- [12] „Oracle Database 11g Standard Edition,“ [Võrgumaterjal]. Kättesaadav: <http://www.oracle.com/us/products/database/standard-edition/overview/index.html>. [Kasutatud 15.04.2013].
- [13] „Physical Storage Structures,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/server.112/e25789/physical.htm#g21431. [Kasutatud 15.04.2013].
- [14] „SQL Anywhere 12 overview,“ iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://dcx.sybase.com/index.html#1201/en/saintro/fg-intro-to-sqla.html>. [Kasutatud 25.03.2013].
- [15] „SQL Anywhere in frontline environments,“ iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://dcx.sybase.com/index.html#1201/en/saintro/uses-aboutasa.html>. [Kasutatud 25.03.2013].
- [16] „SAP Sybase Adaptive Server Enterprise,“ Sybase Inc., [Võrgumaterjal]. Kättesaadav: <http://www.sybase.com/products/databasemanagement/adaptiveserverenterprise>. [Kasutatud 18.04.2013].
- [17] „Advantage Database Server,“ Sybase Inc., [Võrgumaterjal]. Kättesaadav: <http://www.sybase.com/products/databasemanagement/advantagedatabaseserver>. [Kasutatud 18.04.2013].
- [18] „Inside the SQL Anywhere database server,“ iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://dcx.sybase.com/index.html#1201/en/saintro/server-inside-asa-architecture.html>. [Kasutatud 26.03.2013].
- [19] „Backing Up and Recovering an InnoDB Database,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/mysql-backup-excerpt/5.6/en/innodb-backup.html>. [Kasutatud 28.03.2013].
- [20] „Backing Up and Restoring a Single .ibd File,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/mysql-enterprise-backup/3.8/en/partial.restoring.single.html>. [Kasutatud 02.05.2013].

- [21] „mysqldump — A Database Backup Program,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/mysqldump.html>. [Kasutatud 28.03.2013].
- [22] „Database Backup Methods,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/mysql-backup-excerpt/5.6/en/backup-methods.html>. [Kasutatud 28.03.2013].
- [23] „Point-in-Time (Incremental) Recovery Using the Binary Log,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/point-in-time-recovery.html>. [Kasutatud 20.04.2013]
- [24] „Using Replication for Backups,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/replication-solutions-backups.html>. [Kasutatud 20.04.2013].
- [25] „Backing Up a Slave Using mysqldump,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/replication-solutions-backups-mysqldump.html>. [Kasutatud 20.04.2013].
- [26] „Backing Up Raw Data from a Slave,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/replication-solutions-backups-rawdata.html>. [Kasutatud 20.04.2013].
- [27] „Database Backup Methods,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/backup-methods.html>. [Kasutatud 20.04.2013].
- [28] „Glossary,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/backup.112/e10642/glossary.htm#i433290. [Kasutatud 28.03.2013].
- [29] „Performing Backup and Recovery,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/server.112/e10897/backrest.htm#CHDBCGGB. [Kasutatud 28.03.2013].
- [30] „Backup and Recovery User's Guide,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/backup.112/e10642.pdf. [Kasutatud 28.03.2013].

- [31] „What Is the Redo Log?“, Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/server.112/e25494/onlineredo001.htm#i1006163. [Kasutatud 28.03.2013].
- [32] „Image backups“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/dbadmin/da-backup-dbs-5402662.html>. [Kasutatud 01.04.2013].
- [33] „Schedules“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/dbadmin/scheduling-events.html>. [Kasutatud 22.04.2013].
- [34] „Maintenance plan creation“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/dbadmin/help-maintplan-task.html>. [Kasutatud 22.04.2013].
- [35] „Incremental backups“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/dbadmin/da-backup-dbs-5483205.html>. [Kasutatud 01.04.2013].
- [36] „Backing up and deleting the original transaction log“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://dcx.sybase.com/index.html#1201/en/dbadmin/da-backup-dbs-5772240.html>. [Kasutatud 22.04.2013].
- [37] „Restoring from an image backup“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: http://infocenter.sybase.com/help/index.jsp?docset=/com.sybase.help.sqlanywhere.12.0.1/sqlanywhere_en12/help_top_index.htm&docSetID=1744. [Kasutatud 01.04.2013].
- [38] „Database recovery with multiple transaction logs“, iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: http://infocenter.sybase.com/help/index.jsp?docset=/com.sybase.help.sqlanywhere.12.0.1/sqlanywhere_en12/help_top_index.htm&docSetID=1744. [Kasutatud 01.04.2013].

- [39] „Archive backups,“ iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.help.sqlanywhere.12.0.1/dbadmin/da-backup-dbs-5402662.html>. [Kasutatud 01.04.2013].
- [40] „Live backups,“ iAnywhere Solutions, Inc., [Võrgumaterjal]. Kättesaadav: http://infocenter.sybase.com/help/index.jsp?docset=/com.sybase.help.sqlanywhere.12.0.1/sqlanywhere_en12/help_top_index.htm&docSetID=1744. [Kasutatud 01.04.2013].
- [41] „File Space Management,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://dev.mysql.com/doc/refman/5.6/en/innodb-file-space.html>. [Kasutatud 26.03.2013].
- [42] Schwartz, Baron; Zaitsev, Peter; Tkachenko, Vadim; Zawodny, Jeremy; Lentz, Arjen; Balling, Derek J., High Performance MySQL: Optimization, Backups, Replication, and More, Kindle Edition toim., O'Reilly Media, 2008.
- [43] „Tables and Table Clusters,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E11882_01/server.112/e25789/tablecls.htm. [Kasutatud 26.03.2013].
- [44] „Creating Tablespaces,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://docs.oracle.com/cd/E14072_01/server.112/e10595/tspaces002.htm. [Kasutatud 26.03.2013].
- [45] „MySQL Glossary,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: http://dev.mysql.com/doc/refman/5.6/en/glossary.html#glos_system_tablespace. [Kasutatud 26.03.2013].
- [46] „MySQL Cluster CGE,“ Oracle Corporation, [Võrgumaterjal]. Kättesaadav: <http://www.mysql.com/products/cluster/>. [Kasutatud 15.04.2013].

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Roland Roose

(*autori nimi*)

(sünnikuupäev: 28.01.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
MySQL, Sybase ja Oracle andmebaaside varundamine ja taaste

(*lõputöö pealkiri*)

mille juhendaja on Vambola Leping,

(*juhendaja nimi*)

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **07.05.2013**