

XML-indeksien vaikutus DB2 Express-C:n suorituskykyyn XML-käsittelyssä

Janita Ahlqvist

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

04.12.2013



Tekijä tai tekijät Janita Ahlqvist	Ryhmätunnus tai aloitusvuosi 2007
Raportin nimi XML-indeksien vaikutus DB2 Express-C:n suorituskykyyn XML-käsittelyssä	Sivu- ja liitesivumäärä 38 + 37
Opettajat tai ohjaajat Martti Laiho	
<p>Tämän opinnäytetyön tarkoituksena on tutkia indeksien vaikutusta DB2:n suorituskyky-suunnitelmaan XML -päivitysten käsittelyssä. Ennako-odotuksena indeksit nopeuttavat XML-dokumentin löytymistä suuremmasta dokumenttijoukosta, mutta myös hidastavat löydetyn dokumentin päivittämistä.</p> <p>Opinnäytetyön teoreettisessa osiossa käydään läpi XML ja tietokannat yleisesti, indeksien vaikutusta suorituskyky-suunnitelmaan. Osiossa käydään myös läpi TPoX – kuormitustestin kehitystä.</p> <p>Opinnäytetyössä selvitettiin indeksien vaikutusta 10 minuutin testiajojen perusteella. Testit suoritettiin ilman indeksejä sekä indeksien kanssa. Vaikutusta vertailtiin saatuihin statistiikkoihin sekä suorituskyky-suunnitelmaan.</p> <p>Tulokset osoittavat indeksien nopeuttavan XML-dokumenttien hakua kyselyiden, päivitysten ja poistojen osalta. Mahdollinen XML-indeksin päivitys hidasti marginaalisesti käsittelyä. Validointi ilman indeksejä sekä indeksien kanssa hidasti XML-käsittelyä. Johtopäätöksenä voidaan todeta indeksien vaikuttavan positiivisesti XML-käsittelyiden suorituskykyyn, jonka takia niiden käyttäminen on suositeltavaa.</p>	
Asiasanat XML, suorituskyky, päivitys, relaatiotietokannat, testaus	

Degree Programme in Information Technology

<p>Authors Janita Ahlqvist</p>	<p>Group or year of entry 2007</p>
<p>The title of thesis The impact of XML indexes to DB2 Express-C access plan in processing XML</p>	<p>Number of pages and appendices 38 + 37</p>
<p>Supervisor(s) Martti Laiho</p>	
<p>The purpose of this thesis was to examine the impact of XML indexes to DB2's access plan in processing XML. The assumption was that the indexes will speed up discovering a document from a larger range of documents but they also make it slower to update the document.</p> <p>The theoretical part of the thesis discussed the following concepts: XML and databases on a general level; the effects of indexes in an access plan; and an introduction to the development of the TPoX benchmark.</p> <p>The empirical part of the thesis consisted of 10-minute test runs which were executed with and without indexes. The performance impact was examined using the gathered statistics and the acquired access plans.</p> <p>The thesis indicated that the XML indexes have a remarkable impact on the query performance: indexes enhance the performance of XML document handling with updates and deletions. However, validation, with and without indexes, slows down XML document handling.</p> <p>The thesis concludes that the indexes have a positive performance impact on XML document handling and their use is advisable.</p>	
<p>Key words XML, access plan, update, relational database, testing</p>	

Sanasto

Access Plan	Suoritusuunnitelma
CLOB	Character large object. SQL tietotyyppi, joka voi pitää sisällään jopa 4GB tietoa
DB2	IBM:n SQL + XML hybriditietokanta
DTD	Document Type Definition. Määrittää dokumentin rakenteen
Explain Tables	Aputaulut, joihin suorituskykyuunnitelma tallennetaan
FIXML	Talousovellus
HTML	Hyper Text Markup Language. Kieli, jota käytetään web-sivujen kuvaamiseen
pureXML	natiivi XML, DB2:n XML-toteustus
RDBMS	Relational Database Management System. Relaatiotietokannan hallinjärjestelmä
Schema	Dokumentti, joka määrittää XML:n oikeakielisyyden ja sallittujen elementtien rakenteen
SGML	Itsensä kuvaavan tiedon merkintäkieli
Shredding	Tiedon varastointimenetelmä
SQL/XML	SQL-standardin osa, joka määrittää SQL:n XML-laajennukset
Tablespace	Yhdestä tai useammasta tiedostosta koostuva taulutila
Timeron	Mittausyksikkö, jolla lasketaan kustannusarvio DB2 käytettävistä resursseista
TPoX	Transaction processing over XML
W3C	World Wide Web Consortium
Validi	Oikein muodostettu XML-dokumentti, joka täyttää vaaditut kriteerit
Oracle VM Virtual Box	Isäntäohjelmisto valmiiksi luodulle virtuaalikoneelle
XGRN	XML regions -indeksi
XHTML	eXtensible Hypertext Markup Language. HTML:stä kehitetty web-sivujen kuvaamisen tarkoitettu kieli, joka täyttää XML:n muotovaatimukset

XML	Merkintäkieli, jonka avulla kuvataan tietoa tiedosta
XPTH	XML path -indeksi
XQuery	XML-dokumentin käsittelykieli
XVIL	Looginen indeksi
XVIP	Fyysinen indeksi

Sisällys

1	Johdanto	1
2	XML ja tietokannat	2
2.1	SQL/XML – spesifikaatiot	3
2.2	XML -tiedon hallinta DB2:ssa	5
2.2.1	XML-indeksityypit	5
2.3	XQuery-kieli ja sen toteutus DB2:ssa	8
2.4	XML-käsittelyn kuormittavuudesta ja suoritusnopeuden säädyistä	8
2.4.1	XML-dokumentin validointi	12
3	Suoritus suunnitelmien tutkiminen	16
4	TPoX-kuormitustestin kehityksestä	19
5	TPoX 2: -testi virtuaalikone ympäristössä	22
5.1	TPoX 2:n asennus	22
5.2	Testauksen valmistelu	23
5.2.1	Testattavan tiedon lataaminen	23
5.2.2	Suorituskykytestin valmistelu	23
6	Tutkimus	25
6.1	Tutkimusongelma	25
6.2	Tutkimusmenetelmät	25
6.3	Testaus	27
6.4	Tulokset	28
6.4.1	Query -lauseiden käsittely	28
6.4.2	Update -lauseiden käsittely	31
6.4.3	Delete - lauseiden käsittely	31
6.4.4	Insert – lauseiden käsittely	32
7	Johtopäätökset	33
8	Yhteenveto	34
8.1	Kooste	34
8.2	Suosituksset	34
8.3	Oppimisen analysointi	35
	Lähteet	37

Liitteet.....	1
Liite 1. Testikyselyt.....	1
Liite 2. Mixed.xml – tiedoston sisältö.....	1
Liite 3. Testitulokset.....	1
Liite 4. Indeksien luominen	1
Liite 5. Asennushistoria verrattuna Quick Start Guideen.....	1

1 Johdanto

Nicolan tutkimusryhmän mukaan XML:n tietokantateknologia tukee tehokkaasti XML:n käyttöä. XML-tietokantateknologialla on kasvava kysyntä kaupallisten yritysten kuten rahoitus-, pankki-, teollisuus-, koulu-, hallitus ja terveydenhoidon keskuudessa. Tästä johtuen ajankohtaisena aiheena voisi kasvavan kysynnän vuoksi pitää XML-tietokannan kehittämistä, jossa suorituskyky on painopisteenä. (Nicola, Kogan & Schiefer 2007c, 1.)

Tämän opinnäytetyön tarkoituksena on tutkia indeksien vaikutusta DB2:n suorituskykysuunnitelmaan XML -päivitysten käsittelyssä. Ennako-odotuksena indeksit nopeuttavat XML-dokumentin löytymistä suuremmasta dokumenttijoukosta, mutta myös hidastavat löydetyn dokumentin päivittämistä, lisäyksiä ja poistoja. Myös validointi vaikuttaa suoritustehoon, oletettavasti hidastaen sitä.

Tutkimusmenetelmänä käytetään kvantitatiivista eli määrällistä tutkimusmenetelmää. Määrällisen tutkimuksen tarkoituksena on saada tietoa, joka on yleistettävissä. Tutkimuksessa käytettävä testimateriaali on Nicolan tutkimusryhmän valmiiksi määrittelemä.

Kappaleessa 2 käydään läpi XML ja tietokannat yleisesti. Kappaleessa 3 käydään läpi indeksien vaikutusta suorituskykysuunnitelmaan. TPoX – kuormitustestin kehityksestä kerrotaan kappaleessa 4. Kappaleessa 5 testataan virtuaaliympäristöä. Kappaleessa 6 keskitytään itse tutkimuksen suorittamiseen. Kappaleet 7 ja 8 sisältävät johtopäätökset ja yhteenvedon saaduista tutkimustuloksista.

Opinnäytetyöksi valitsin aiheen, koska halusin syventää koulussa oppimiani DB2- ja XML-tietoja ja taitoja sekä soveltaa opittua käytännössä. Pidän tätä myös tulevaisuudessa hyvin tärkeänä asiana koskien tietokantoja.

2 XML ja tietokannat

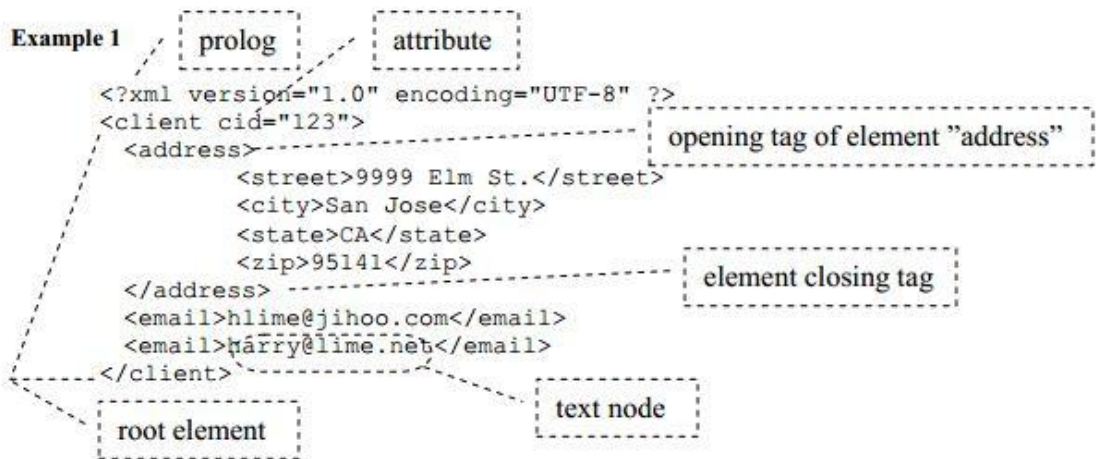
XML (eXtended Markup Language) on luotu hallitsemaan dokumenttien tietojen sisältöä. XML on oikeastaan yksinkertaistettu osajoukko aiemman SGML Markup Language dokumentteja (ISO 8879:1986), jolla on kasvava rooli Web- ja tietojärjestelmissä. Sillä on ollut suuri vaikutus ICT-alalla muun muassa alustasta riippumattomien sanomien rakenteeseen, tiedonsiirtoon sanoman välitys-sovellusten välillä sekä erilaisiin konfigurointi tiedostoihin. Koska XML on laajassa käytössä tietojärjestelmissä, on siitä tullut läheinen partneri RDBMS:n kanssa. Itse asiassa jokaisen ICT-ammattilaisen olisi hyvä omata ainakin joitakin perustietoja XML:stä sekä XML:stä tietokannoissa. (Duffoux, A., Keramopolous, E., Kurki, M., Laiho, M. & Laux F. 2011, 3.)

XML on itsekuvaava merkkikieli, toisin sanoen se sallii laajennettujen, sovelluksesta riippumattomien merkkien käytön sen tunnettujen ”serkkujen” HTML:n ja XHTML:n tavoin. Nämä ovat pääasiassa suuntautuneet määrittelemään dokumentin esittämisen käyttämällä integroituja merkkejä. Vaikka suunnittelijat voivat luoda merkkejä, XML-kieli on rajoittunutta merkkien käytössä. XML-dokumentissa täytyy olla oikein muodostettu (well-formed) puurakenne:

- Elementit määritellään elementti-merkein. Jokaiselle aloitusmerkille (start tag) on oltava myös lopetusmerkki (end tag). Merkki voi myös olla tyhjä (empty tag), jolloin se on muotoa <tag />.
- Uloimmaista eli pääelementtiä (root) voi olla vain yksi.
- Kaikille muille elementeille on olemassa yksi isä-elementti. Elementti on täysin sisällytetty isä-elementin sisältöön.

(Duffoux ym. 2011, 3.)

Elementit voivat sisältää yksinkertaisia tietotyyppisiä (merkkijono, numeerinen) tai monimutkaisia tietotyyppisiä (muita elementtejä). Elementeillä voi myös olla sidottuja attribuutteja, kuten kuviossa 1. on esitetty. (Duffoux ym. 2011, 3.)



Kuvio 1. Puurakenne (Duffoux ym. 2011, 4)

XML-kieli on merkkiriippuvainen, josta johtuen elementin nimi on kirjoitettava aloitus- ja lopetusmerkkeihin täysin samaan formaattiin. Attribuuttien arvot on liitetty mukaan joko lainausmerkeillä (") tai puolilainausmerkeillä ('), toisin kuin tekstit (text node) joita ei ole liitetty mukaan lainausmerkeillä. (Duffoux ym. 2011, 4.)

XML:n ollessa itsekuvaava merkkikieli, tarvitaan erillinen dokumentti määrittämään sallituille elementtityypeille oikeakielisyys sekä rakenne. Oikeakielisyyden, jota kutsutaan Schemaksi, määrittämisen jälkeen voidaan XML-dokumentti validoida sitä vasten. Validoinnilla varmistetaan XML-dokumentissa rakenteen ja käytettyjen elementtien vastaavan niiden määrittelyä Schemassa. Validoinnin onnistuessa kutsutaan dokumenttia validiksi, toisin sanoen se on määritelty XML-dokumentti. (Duffoux ym. 2011, 5.)

Viimeaikaisten kehitysten myötä tietokantateknologiassa on huomattu tarve hallita XML-dokumentteja tietokannoissa. XML-tietokantajärjestelmät ovat vuosien mittaan ilmestyneet markkinoille ja monet relaatiotietokantajärjestelmät on laajennettu tukemaan XML dataa. (Duffoux ym. 2011, 6.)

2.1 SQL/XML – spesifikaatiot

Sen sijaan, että käyttäisi erillisiä XML-dokumentteja tai pureXML-tietokantoja, XML-dokumenttien tallentaminen yhdessä SQL- tiedon kanssa relaatiokantoihin tarjoaa monia etuja. Näitä on esimerkiksi vahvat, skaalautuvat ja liiketoiminnalliset palvelut

RDBMS järjestelmiin. Myös järjestelmän eheys, turvallisuus, välimuisti ja pysyvyys kuuluvat etuihin. SQL/XML:n duality-ominaisuuden ansiosta SQL- ja XML-tietomallien integrointi mahdollistaa tiedon johdonmukaisen hallittavuuden. (Duffoux ym. 2011, 20.)

Termi SQL/XML viittaa tarkoin määrättyihin XML-ominaisuuksiin ja toimintoihin SQL 2003 ja 2006 standardeissa. Alla olevassa taulukossa (Taulukko 1) on esitetty, kuinka SQL/XML määritellään.

Taulukko 1. SQL/XML-määrittely

XML-tietotyyppi, yleensä natiivi SQL-tyyppi	Esimerkiksi INTEGER tai CHAR. SQL/XML määrittää tyypin semantiikan, ei sen tallennusmuotoa
Toiminnot, jotka muuntavat arvot ei-XML-tietotyypistä XML-tyyppiin ja toisinpäin	Esimerkiksi CHAR, VARCHAR, CLOB. Nämä toiminnot ovat XMLSERIALIZE, XMLPARSE ja XMLCAST
Toiminto XML Scheman validoimiseksi ja predikaatti, joka tarkistaa validoinnin tilan XML dokumentille tai kappaleelle	Esimerkiksi XMLVALIDATE, IS VALIDATED
XML julkaisutoiminnot (muodostin-toiminnot), jotka sallivat uuden XML dokumentin tai kappaleen muodostamisen. Syöttötiedot XML muodostamiseen voivat tulla relaatio-, XML-sarakkeista tai molemmista	Esimerkiksi XMLELEMENT, XMLATTRIBUTES, ja XMLAGG
Toiminnot XPathin ja XQueryn sulauttamiseksi SQL lauseisiin	Esimerkiksi XMLQUERY, XMLTABLE ja XMLEXISTS predikaatti

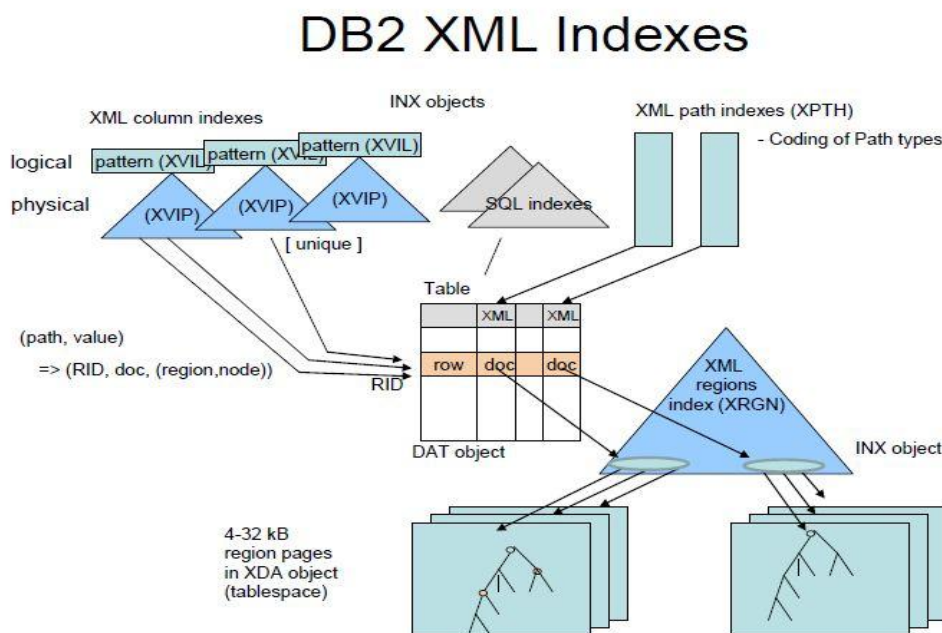
Kaikki nämä SQL/XML toiminnot ovat tuettuja DB2-järjestelmissä. (Kumar-Chattarjee & Nicola 2009, 160.)

2.2 XML -tiedon hallinta DB2:ssa

DB2 pureXML tarjoaa älykkäitä ja monipuolisia ominaisuuksia XML-dokumenttien kanssa työskentelyyn ja niiden säilytykseen. Yksi niistä on XML-sarakkeiden indeksointiominaisuus, jolla tehostetaan XQuery ja SQL/XML-käsittelyä. (Kumar-Chatterjee & Nicola 2009, 174.) Indeksointi nopeuttaa tiedon hakemista ja käyttämistä. Sitä käytetään yleensä kyselyn suorituskyvyn parantamiseen. Tässä luvussa esitetään pureXML indeksin ominaisuuksia, miten XML-hakemistoja käytetään suorituskyvyn parantamiseen. Indeksien luominen on esitetty kokonaisuudessaan Liitteessä 4.

2.2.1 XML-indeksityypit

DB2 9 pureXML Guide – oppaassa (IBM 2007.) on yksityiskohtainen esittely kolmesta XML indeksityypistä: XML regions -indeksi, XML path- indeksi ja XML column -indeksi. Seuraavassa kuviossa (Kuvio 2) on esitetty, miten XML indeksit toimivat DB2:ssa.



Kuvio 2. XML-indeksien rakenne DB2:ssa (Opettajan luentomoniste)

Kuvio 2 havainnollistaa, kuinka XML-indeksit toimivat. Taulukkoa ja XML-sarakkeita ylläpidetään DAT-objektissa. Tämän taulukon XML-sarake ei sisällä varsinaisia XML-dokumentteja vain ainoastaan loogisia viitteitä niihin, koska XML-dokumentti voi olla liian iso mahtuakseen relaatorivin yhdelle sivulle. Seuraavissa kappaleissa kuvataan yksityiskohtaisemmin nämä indeksit.

XML regions – indeksi

XML regions -indeksi tallentaa kunkin XML-dokumentin sijainnin, joka on tallennettu XDA sivulle DB2:ssa. XML regions -indeksi luodaan automaattisesti DB2:ssa kun ensimmäinen XML -sarake luodaan tai lisätään tauluun. Taululle luodaan vain yksi XML regions -indeksi vaikka se sisältäisi useita XML sarakkeita. (IBM 2007, 174–175.)

Jokainen XML regions -indeksi on identifioitu XRGN-arvolla INDEXTYPE-sarakkeessa ja tallennettu SYSCAT.INDEX:iin. (Kumar-Chatterjee & Nicola, 2009, 34). XML regions -indeksi mittaa, kuinka XML-dokumentti on sisäisesti jaettu alueisiin (IBM Infocenter 2012b).

Oletusarvoisesti XML-dokumentit tallennetaan XDA-objektiin. Jos taulukossa on useita XML-sarakkeita, jakavat ne kaikki yhteisen XDA-objektin. Kun asiakirjapuu ei mahdu yhdelle sivulle, DB2 automaattisesti pilkkoo puun useiksi alipuiksi, joita kutsutaan alueiksi (regions). Jokainen alue on tallennettu erilliselle XDA-sivulle niin, että yksittäinen dokumentti voi ylettyä usealle sivulle. Toisaalta dokumenttien ollessa merkittävästi pienempiä kuin sivun koko, voidaan useat alueet (regions) tallentaa yhdelle sivulle ettei tilaa mene hukkaan. Keskeinen näkökulma fyysisen tietokannan suunnitteluun on sivun koko tablespacessa (taulutila). Suorituskyky paranee, mitä pienempi on alueiden lukumäärä yhdessä XML-dokumentissa. Alueiden määrä dokumentissa määrittyy sivun koon mukaan (4KB, 8KB, 16KB, or 32KB). Tablespacen sivun koon ollessa suuri, alueiden määrä dokumentissa on puolestaan pienempi. (Kumar-Chatterjee & Nicola 2010, 34).

Pääsy XML-varastoon tallennettuihin XML-dokumentteihin kulkee aina XML regions -indeksi kautta. XML regions – indeksi mahdollistaa loogisen mappauksen niille alueille

joilta dokumentin data voidaan hakea. Regions – indeksistä haettaessa XML-tiedon avainsanana käytetään dokumentin- ja version ID:tä indeksihauussa. Taulun sisältäessä XML – sarakkeita syntyy sille myös yhden XML regions – indeksi. Näitä indeksejä ei voi poistaa (komennolla drop). (IBM 2007, 175.)

XML path – indeksi

Järjestelmä tuottaa XML path -indeksin kaikille tauluun luoduille tai lisätyille XML-sarakkeille. Se tallennetaan SYSCAT.INDEXES:iin. XML path -indeksi on esitetty XPTH:na SYSCAT.INDEXES.INDEXTYPE:ssä. Jos luodaan taulu kahdella XML sarakkeella, luo DB2 yhden XML regions -indeksin mutta kaksi path -indeksiä. XML column path – indeksi mappaa polusta polun ID:n jokaisessa XML – sarakkeessa. (IBM 2007, 175–176.) TPoX:ssä järjestelmä luo XML path -indeksin itsessään jokaiseen tauluun.

XML column – indeksi

XML regions- indeksi ja XML path -indeksi ovat sisäisiä indeksejä, jotka liittyvät XML-sarakkeeseen. Nämä indeksit eivät tunnusta mitään ohjelmointirajapintaa, joka palauttaa indeksin metatietoa. (IBM Infocenter 2012c.)

XML column -indeksi eroaa näistä kahdesta indeksistä, se on luotu XML-sarakkeelle. Sitä käytetään XQuery:n ja SQL/XML:n suorituskyvyn lisäämiseen. XML-indeksi on luotu kuten B-puu indeksi ja tallennetaan samaan paikkaan, kuin relaatioindeksit tallennetaan. Useita XML-indeksejä voidaan määritellä yhteen XML-sarakkeeseen. Samaan aikaan tulee olla varovainen indeksien luonnissa XML-sarakkeeseen, koska se voi vähentää Insert-, Update- ja Delete -lauseiden suorituskykyä. Lisäksi indeksit vievät tilaa. Tästä johtuen turhien indeksien tekemistä tulee välttää. (IBM 2007, 176.)

Kun XML-sarakkeeseen luodaan indeksi, luodaan oikeastaan kaksi indeksiä, looginen ja fyysinen. Looginen indeksi sisältää XML-kuvion (pattern) tiedot CREATE INDEX -lauseessa. Fyysisessä indeksissä on DB2 luomat avainsarakkeet, jotka tukevat loogista indeksiä ja sisältävät indeksin todellisen arvon. Käyttäjä työskentelee loogisella tasolla

CREATE INDEX ja DROP INDEX -lauseilla. DB2:n fyysisen indeksin käsittely taustalla ei näy käyttäjälle. Loogisen indeksin indeksinimi on määritetty CREATE INDEX -lauseessa, ja on indeksityyppiä XVIL. Fyysisellä indeksillä on järjestelmän määrittämä nimi ja se on indeksityyppiä XVIP. (IBM 2007, 181–182.)

2.3 XQuery-kieli ja sen toteutus DB2:ssa

XQuery on toiminnallinen ohjelmointikieli, jonka suunnitteli W3C. Sen tarkoituksena on vastata XML-datan muokkaamisen ja kyselyiden vaatimuksiin. Toisin kuin relaati-naalinen data, joka on ennustettavissa ja omaa säännöllisen rakenteen, XML-data taas on hyvin vaihtelevaa. Se on usein arvaamatonta, harvaa ja itsensä kuvaavaa. XML-datan ollessa arvaamatonta, eroavat myös esitettävät kyselyt tyypillisistä relaatiokyselyistä. XQuery tarjoaa joustavuutta tämän tyylisten toimintojen suorittamiseen. (IBM Infocenter 2012 a.)

XQuerya tarvitaan muun muassa seuraavien toimintojen suorittamiseen:

- XML-tiedon etsimiseen kohteista, jotka sijaitsevat hierarkian tuntemattomilla tasoilla.
- Tiedon rakenteellisten muutosten suorittamiseen, esimerkiksi hierarkian päinvastoin kääntämiseen.
- Palauttaa tuloksia eri tietotyypeistä.
- Jo olemassa olevan XML-tiedon päivittämiseen. (IBM Infocenter 2012 a.)

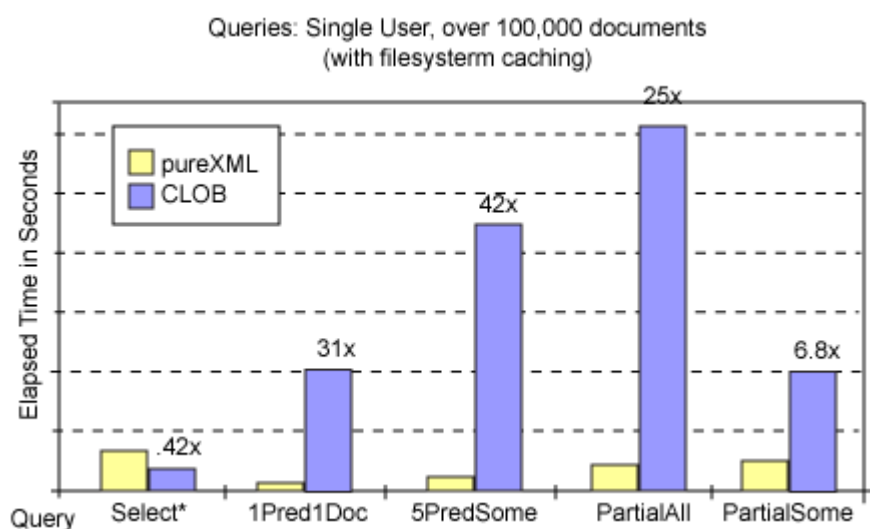
2.4 XML-käsittelyn kuormittavuudesta ja suoritustehon säädöistä

O'Mahonyn (2008) mukaan on perinteisesti kaksi lähestymistapaa XML – datan varastointiin relaatiotietokannassa, CLOB tai Shredding.

CLOB

XML -datan ollessa tallennettu CLOB -sarakeeseen (character large object) on data yleisimmin lisätty ei-jäsenneltynä tekstinä. Näin vältetään XML -jäsennys Insert-lauseen yhteydessä, kuitenkin sen vaatiessa XML-jäsennystä kyselyä suoritettaessa. Tämä johtaa

alhaiseen etsintä- ja purkamissuoritukseen. Seuraavassa kuvassa on esitettyä kummankin suorituskykyä kyselyissä. (O'Mahony 2008.)



Kuvio 3. CLOB ja pureXML suorituskykytestissä (O'Mahony 2008.)

Kuvio 3 esittää tulokset viidelle tehdylle kyselylle, jotka on esitetty Taulukossa 2.

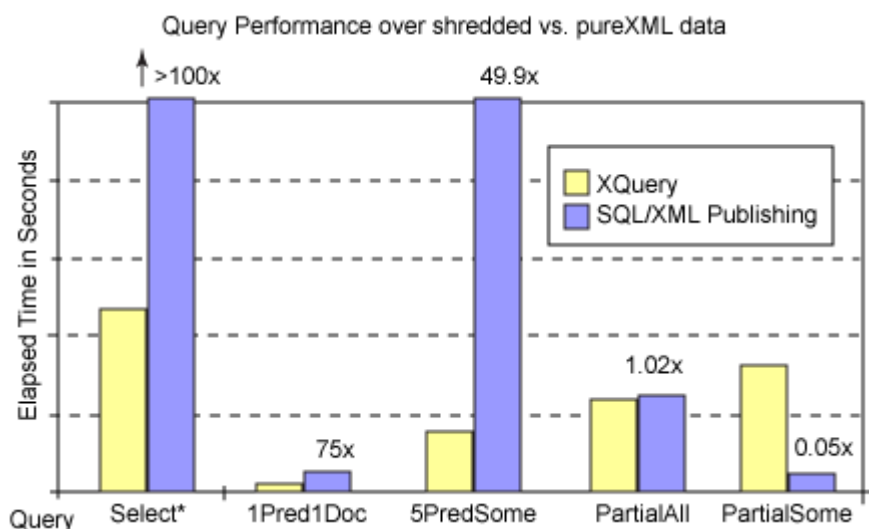
Taulukko 2. CLOB ja pureXML suorituskykytestin kyselyt (O'Mahony 2008.)

Select*	Täysi dokumenttihaku kaikista dokumenteista, ei predikaatista (hakuehto).
1Pred1Doc	Täysi dokumenttihaku yhdestä dokumentista, joka on yhteensopiva predikaatin kanssa
5PredSome	Täysi dokumenttihaku dokumenteista jotka ovat yhteensopivia useamman predikaatin kanssa
PartialAll	Osittainen haku kaikista dokumenteista
PartialSome	Osittainen haku kaikista dokumenteista jotka vastaavat valittuja kriteerejä

Taulukossa 2 on esitetty kyselyt, joiden tulokset löytyvät kuviosta 3. Kuten kuviosta 3 selviää, useimmissa kyselyissä pureXML -varastointi päihittää huomattavasti CLOB -varastoinnin. Vain täysi dokumenttihaku joka jättää huomioimatta XML - rakenteen lukee XML - tietoa nopeasti CLOB - sarakkeista. (O'Mahony 2008.)

Shredding

Käyttämällä Shreddingia on XML – tieto mapattu relaatorakenteeseen, joka on varastoitu relaatiotietokantaan. Seuraavassa kuviossa esitetään, kuinka XQuery (jota käytetään natiivissa XML:ssä) päihittää SQL-kyselyn XML-muuntamisella (jota käytetään shreddingiin) suurimmassa osassa kyselyitä. Kysely, joka hakee osan tiedoista kaikista tietueista, on nopeampi kuin tiedon ollessa Shredded. Relaatiotiedon etsiminen on näin ollen nopeampaa. (O'Mahony 2008.)

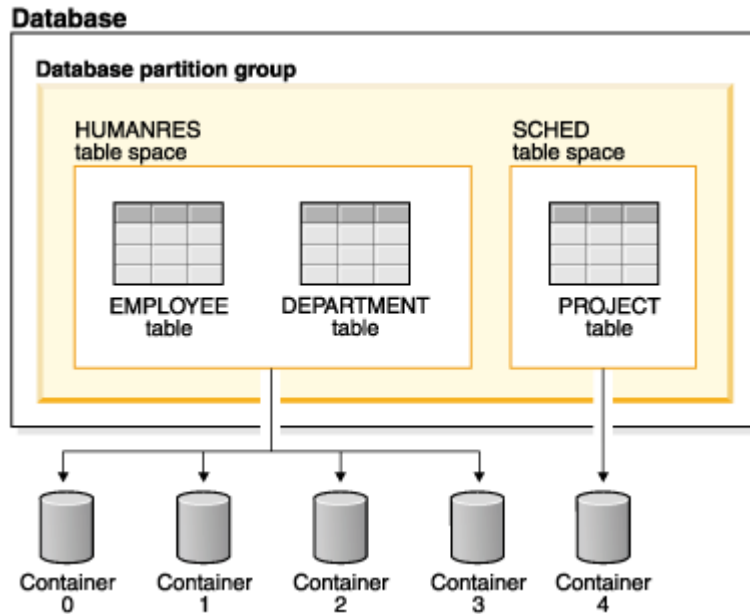


Kuvio 4. Shredded vastaan pureXML –tieto suorituskyky testissä (O'Mahony 2008.)

Kuvioiden 3 ja 4 löydökset esittävät, että useimmissa olosuhteissa, pureXML-varastointi tarjoaa huomattavia suorituskyky etuja, verrattuna CLOB-varastointiin tai Shreddingiin. (O'Mahony 2008.)

Tablespace (taulutila)

Tablespace on varastointirakenne, jonka sisältö koostuu tauluista, indekseistä, large objekteista ja long datasta. Tablespace sijaitsee jossakin tietokannan partitioryhmissä. Se mahdollistaa tietokannan ja taulukon sijainnin määrittämisen suoraan containeriin (säiliö), mahdollistaen paremman suorituskyvyn ja joustavamman kokoonpanon. Container voi olla hakemiston-, laitteen- tai tiedostojoukon nimi. (IBM Infocenter 2013a.) Alla oleva kuvio (Kuvio 5) havainnollistaa taulujen, tablespacen, sekä niihin liittyvien containereiden välistä suhdetta tietokannassa.



Kuvio 5. Taulujen, tablespacen sekä containereiden välinen suhde (IBM Infocenter 2013a.)

Suunniteltaessa Tablespace:ä tulee sivun koko ottaa huomioon. Sivun kokorajoituksia ovat 4K, 8K, 16K tai 32K. Jokaiselle sivun koolle on myös tablespace:n maksimikoko, jota tulee noudattaa. Maksimikoko määräytyy tablespace:n hallintatyyppin perusteella. (SMS, DMS, Automatic storage, regular/large). (IBM Infocenter 2013b.) Alla olevassa kuviossa (Kuvio 6) on esitetty sivun koko rajoitukset erityyppisille Tablespaceille.

Table space type (in gigabytes)	4K page size limit	8K page size limit	16K page size limit	32K page size limit
SMS table spaces	64	128	256	512
DMS table spaces (regular)	64	128	256	512
DMS table spaces (large)	2048	4096	8192	16 384
Automatic storage table spaces (regular)	64	128	256	512
Automatic storage table spaces (large)	2048	4096	8192	16 384
Temporary table spaces	64	128	256	512

Kuvio 6. Tablespaceen kokorajoitukset (IBM Infocenter 2013b.)

Taulukossa 3 on esitettyinä muutama esimerkkikomento SMS-, DMS- ja Automatic storage tablespace:n luonnista. Automatic storage tablespacen luomiseen on kaksi tapaa, jotka molemmat on esitettyinä taulukossa.

Taulukko 3. Esimerkkikomentoja Tablespace:n luomiseen (IBM Infocenter 2012c.)

SMS tablespace	CREATE TABLESPACE NAME MANAGED BY SYSTEM USING ('path')
DMS tablespace	CREATE TABLESPACE NAME MANAGED BY DATABASE USING (FILE 'path' size)
Automatic storage table space	CREATE TABLESPACE NAME MANAGED BY AUTOMATIC STORAGE

2.4.1 XML-dokumentin validointi

XML validointi on prosessi, joka määrittää onko XML-dokumentin rakenne, sisältö ja tietotyypit valideja XML Scheman suhteen. XML validointi myös poistaa ylimääräiset rivit dokumentista. Validointi on valinnaista, mutta suositeltavaa kun kyseessä on datan eheys. Validointi takaa XML-dokumentin noudattavan XML Schemaan määritettyjä sääntöjä sekä olevan oikein muodostettu. DB2 tukee XML-dokumenttien validointia vain XML Schemaa vasten. Dokumenttia ei voi validoida DTD:tä vasten. (IBM Infocenter 2013c.)

XML-dokumentin validoimiseen käytetään XMLVALIDATE -komentoa. XMLVALIDATE – komentoa voi tarkentaa SQL – lauseella, joka lisää tai päivittää XML – dokumentteja DB2 tietokantaan. XML – dokumenttien automaattiseen validointiin annetaan BEFORE triggeri XML – sarakkeessa, joka suorittaa XMLVALIDATE – komennon. (IBM Infocenter 2013c.)

Kaikki dokumentit joista Schema koostuu on rekisteröitävä sisäänrakennettuun XML Schema-varastoon ennen XMLVALIDATE-komennon kutsumista. Itse XML-dokumentin ei tarvitse olla tietokannassa, jotta sen voi validoida XMLVALIDATE-komennolla. (IBM Infocenter 2013c.)

Kuten aiemmin mainitsin, on XML-dokumentin validointi XML Schemaa vasten vapaaehtoista DB2:ssa. Tyypillisin skenaario DB2:ssa on XML-dokumentin validointi kun se on lisätty tai ladattu tietokantaan. Kun varmistaa tietokantaan lisättävän XML:n olevan validia, kyselyt olettavat tiedon olevan validia ja käyttävät sen XML Schemaa. XML Update -lauseiden validointi varmistaa ettei dokumentin korvaaminen tai muokkaaminen vahingoita määriteltyä XML Schemaa. (Nicola 2012).

Seuraavasta esimerkistä (Kuvio 7) selviää, kuinka Insert ja Update-lauseita tulisi validoida XML Schemaa vasten. XML Schema on rekisteröity DB2 XML Schema Repository (XSR) alle SQL-nimellä db2admin.myschema. (Nicola 2012)

```
CREATE TABLE mytable (id INTEGER NOT NULL PRIMARY KEY, doc XML);

INSERT INTO mytable
VALUES(?, XMLVALIDATE(? ACCORDING TO XMLSCHEMA ID db2admin.myschema));

UPDATE mytable
SET doc = XMLVALIDATE(? ACCORDING TO XMLSCHEMA ID db2admin.myschema)
WHERE id = ?;

UPDATE mytable
SET doc = XMLVALIDATE( XMLQUERY('copy $new := $DOC
                                modify do insert <status>delivered</activated>
                                into $new/message/header
                                return $new')
                        ACCORDING TO XMLSCHEMA ID db2admin.myschema)
WHERE id = ?;
```

Kuvio 7. Insert ja Update -lauseiden validointi (Nicola 2012.)

Joskus on tapauksia, jolloin XML halutaan validoida osana kyselyä. Syitä tähän voi olla esimerkiksi

- Tiedosto on lisätty tai päivitetty ilman validointia, jolloin se on validoitava ennen käyttöä.
- XML -tiedosto halutaan validoida eri Schemaa vasten kuin alun perin.
- Puretaan XML -tiedostoon tallennettuja fragmentteja, jotka halutaan validoida tiettyä Schemaa vasten.
- Kysely rakentaa täysin uuden XML – dokumentin ja se halutaan validoida siten että rakennettu XML noudattaa annettua Schemaa.

XML-validointi kyselyssä on yksinkertaista. Se onnistuu käyttämällä XMLVALIDATE-toimintoa Select-lauseessa. XMLVALIDATE-toiminnossa on käytettävissä kaikki samat vaihtoehdot kuin Insert- ja Update-lauseissa. (Nicola 2012.)

```
SELECT XMLVALIDATE(doc ACCORDING TO XMLSCHEMA ID
db2admin.myschema)
FROM mytable
WHERE id = 5;
```

Kuvio 8. XMLVALIDATE-lauseen käyttö Select-lauseessa (Nicola 2012.)

Yllä esitetty kysely (Kuvio 8) lukee määritellyn dokumentin ja suorittaa Scheman validoinnin XML Schemaa vasten joka on rekisteröity db2admin.myschemaan. Jos valittu dokumentti on validi määritellylle Schemalle, palautuu dokumentti. Jos taas valittu dokumentti ei ole validi määritellyyn Schemaan, kysely epäonnistuu ja tuottaa virhekoodin. Virhekoodi osoittaa, miksi Schema on rikkiäinen. (Nicola 2012)

Sen sijaan että XML-sarakkeen nimi olisi doc, voi se olla mikä tahansa sallittu SQL -nimi. XMLVALIDATE – funktion argumenttina voi olla sarakkeessa oleva XML – tiedosto, joka palauttaa XMLQUERY – funktion tuloksena. (Nicola 2012)

Seuraava kysely (Kuvio 9) käyttää XMLQUERY toimintoa poimiakseen pelkän viestin XML-dokumentista ja validoi sen XML Schema db2admin.msgbody.XSD:tä vasten.

```

SELECT XMLVALIDATE( XMLQUERY('${DOC}/message/body')
    ACCORDING TO XMLSCHEMA ID db2admin.msgbodyXSD )
FROM mytable
WHERE id = 5;

```

Kuvio 9. XMLVALIDATE-lauseen käyttö XMLQuery-toiminnolla (Nicola 2012.)

Seuraava kysely (Kuvio 10.) puolestaan rakentaa uuden XML-dokumentin ja validoi sen osana kyselyä.

```

SELECT XMLVALIDATE(
    XMLQUERY('document{
        <newdocument>
        <header>{${DOC}/party/identity}</header>
        <body>
        {${DOC}/party/name}
        {${DOC}/party/details/address}
        </body>
        </newdocument>}')
    ACCORDING TO XMLSCHEMA ID db2admin.newdocXSD)
FROM mytable
WHERE id = 5;

```

Kuvio 10. XMLVALIDATE-lauseen käyttö (Nicola 2012.)

Nämä Nicolan (2012) esimerkit antavat käsityksen XML kyselyn validoinnin mahdollisuuksista XML Schemaa vasten.

3 Suoritus suunnitelmien tutkiminen

DB2 kyselyn optimoija analysoi kyselyn ja laatii sille suoritus suunnitelman. Optimoija laatii useita eri suoritus suunnitelmia kyselylle määrittääkseen, mikä suunnitelmista on tehokkain arvioitaessa resurssien kustannuksia. Kyselyn optimoija arvioi sekä CPU:n että I/O:n kustannuksia laskiessaan arvioitua kustannusta suoritus suunnitelmalle. Suoritus suunnitelma tallennetaan Explain- tauluihin DB2 Explain-työkalulla. (IBM Infocenter 2013d.)

Ajamalla kysely lisäämällä sen eteen EXPLAIN PLAN FOR -määre, ei kyselyä suoritetaan vaan optimoitu suoritus suunnitelma tallentuu Explain- tauluihin.

Suoritus suunnitelmille on erilaisia esitystapoja. Ne voidaan esittää merkkipohjaisena tai graafisena riippuen, mitä Explain-työkalua käytetään. DB2EXFMT tulostaa merkkipohjaisen suoritus suunnitelman. Control Center ja Data Studio Developer käyttävät Visual Explain-työkalua, jolla voidaan tulostaa suoritus suunnitelma graafisena. (Kumar-Chatterjee & Nicola 2009, 401.)

DB2EXFMT-työkalulla voidaan listata Explain- taulut. Komento jolla DB2EXFMT-työkalulla saadaan suoritus suunnitelma näkyviin Explain- taulusta on esitetty alla olevassa taulukossa selityksineen (Taulukko 4).

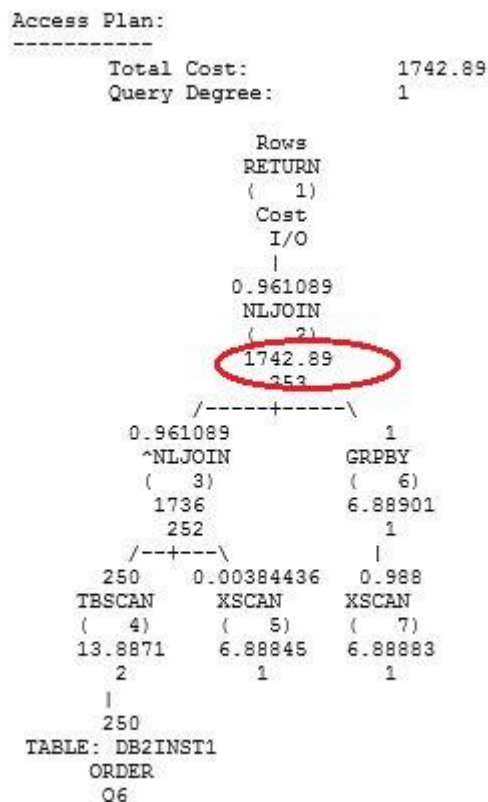
Taulukko 4. DB2EXFMT-työkalu

db2exfmt -d tpoX -1 -o Q1.plan	
-d	Tietokanta
-1	Viimeisin suoritus suunnitelma
-o	Mihin tiedostoon suoritus suunnitelma tulostetaan

DB2 mittayksikkö kustannusten laskemiseen on ”timeron”. Timeron ei ole suoraan rinnastettavissa CPU:n suoritus aikaan. Se antaa suhteellisen mittauksen arvioiduista resursseista, joita vaaditaan kyselyn suorittamiseen valitun suoritus suunnitelman mukaan. Vaikkei arvioidut kustannukset timeron- arvoina ennusta kyselyn suoritus aikaa,

tarjoaa se pohjan kahden kyselyn vertailulle samassa ympäristössä vastaavan kuorman alla. (IBM Infocenter 2013d.)

Alla olevassa kuviossa (Kuvio 12.) on esimerkki erään kyselyn suoritus suunnitelmasta. Suoritus suunnitelmasta näkyy laskettu timeron-arvo jokaiselle vaiheelle sekä kokonaiskustannusarvo.



Kuvio 11. Esimerkki suoritus suunnitelmasta

Kuten kuviosta 12 näkyy, suoritus suunnitelmassa on viisi operaattoria ja kolme numeroa, jotka tulisi ymmärtää. Numero jokaisen operaattorin nimen (Return, Nljoin, Grpby, Tbscan, Xscan) yläpuolella on arvioitu operaattorin tuottama rivimäärä. Seuraavat kaksi rivinumeroa ovat operaation kustannusarvio timeroneina sekä I/O kustannuksista.

4 TPoX-kuormitustestin kehityksestä

Laamasen tutkimuksen mukaan XML-tietokantojen suorituskyvyn mittauksiin tarkoitettuja mittausmenetelmiä on useita. Menetelmiä ovat The Michigan Benchmark, The XOO7 Benchmark, Xbench, XMach-1 ja XMark. (Bressan et al. 2002; Böhme & Rahm 2001; Runapongsa et al. 2003; Schmidt et al. 2001a; Yeo et al. 2004.)

Laamasen mukaan XML -dataa käyttävien B2B- ja verkkosovelluksien suorituskyvyn mittaamista varten on suunniteltu XMach-1. XMach-1:llä voidaan asettaa testiajoon useita yhtäaikaista virtuaaliasiakkaita, jolloin saavutetaan verkkosovelluksille ominainen usean yhtäaikaisen käyttäjän aiheuttama kuormitus. Testidata on määritelty DTD:llä koostuen pienikokoisista, 10 kt:n XML-dokumenteista. Ominaisuuksiltaan datan rakenne on muistuttaa testausjärjestelmän testiraportteja. XMach-1 koostuu yhdestätoista erilaisesta kyselystä, jotka on jaoteltu eri ryhmiin sen mukaan, liittyykö kysely dataan hakuun vai sen manipulointiin. XMach-1:lle on julkaistu referenssitoteutus testiajojen suorittamista varten. (Bressan et al. 2003, s. 488; Böhme & Rahm 2001, s. 9; Böhme & Rahm 2005.)

Nicolan tutkimusryhmän mukaan, että TPoX on sovellustason XML:n tietokantakäsittelyn suorituskykytesti, joka perustuu FIXML-talousovellus skenaarioon. Sitä käytetään pääasiassa arvioimaan XML-tietokantajärjestelmän suorituskykyä. Se keskittyy XQuery:n, SQL/XML:n, XML:n varastointiin, indeksointiin, päivityksiin, kirjaukseen ja samanaikaisuuteen kuten myös XML Scheman validointiin. Nicolan tutkimusryhmä valitsi useiden talousovellusten joukosta online kaupankäynnin ja välityspalkkion, koska se on tärkeä sovellus. Se on sekä suorituskykytestin, että tietokantakäyttäjien helposti ymmärrettävissä. (Kogan, Nicola & Schiefer 2007a.)

FIXML:ssä, Insert-, Update- ja Delete-transaktioita voidaan käyttää muun muassa seuraavissa havainnoissa:

- Asiakkaan tilit on päivitetty vastaamaan kauppoja, muttei välittömästi heti jokaisen tilauksen jälkeen (execute-lause tilaukseen)
- Uusia tilauksia saapuu jatkuvasti, vanhat karsitaan pois järjestelmästä samaan tahtiin (useita Insert-lauseita order:n ja useita Delete-lauseita order:n)

- Vakuuksien hinta päivittyy säännöllisesti työpäivän mittaan. (updates-lauseita)
- Asiakkaan liikevaihto on matala (muutama insert-lause CustAcc ja muutama delete CustAcc)
- Vakuuksien summa pysyy kiinteänä (ei delete- tai insert-lauseita vakuuksiin)

(Kogan, Nicola & Schiefer 2007b.)

TPoX sisältää seuraavat osiot:

- XML Schema kaikille suorituskykytestissä käytetyille tiedostotyypeille, mukaan lukien FIXML Schema.
- Joukosta transaktioita, jotka voidaan ajaa luotuun tietoon.
- Ohjelmiston XML-tiedon tuottamiseen, joka mahdollistaa miljoonien XML-tiedostojen tehokkaan tuottamisen selkeästi määritellyillä arvoilla.
- Suorituskykytestiajuri suorittamaan käyttäjän määrittelemää kuormaa, joka myös kerää ja tulostaa testituloksia. Sitä voidaan muokata asetustiedostojen kautta.
- Dokumentaation kuvaus TPoX:n täytäntöönpanon yksityiskohdista sekä kuinka käyttää dokumenttia.

TPoX suorituskykytesti on tarkoin suunniteltu ja toteutettu. Se saavuttaa tietyt tekniset vaatimukset, kuten siirrettävyyden, yksinkertaisuuden ja skaalautuvaisuuden. (Kogan, Nicola & Schiefer 2007a.)

TPoX suorituskykytesti koostuu kahdesta vaiheesta. Ensimmäinen vaihe suorittaa rinnakkain tietokannan täyttämisen Insert-lauseilla ja ylläpitää kaikki halutut indeksit samaan aikaan. Toinen vaihe kuormittaa täytettyä tietokantaa usean käyttäjän lue/kirjoita -lauseilla, joista 70 % on kyselyitä ja 30 % kirjoita -operaatioita, sisältäen Insert-, Update-, Delete-lauseita yhdisteltyinä. Molemmat vaiheet toteutetaan kuormitusajurilla. (Kogan, Nicola & Schiefer 2007a.)

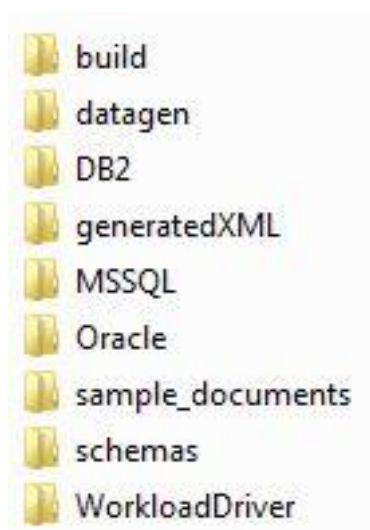
TPoX viitekehys on erittäin venyvä ja sitä voidaan käyttää määrittämään useita erilaisia liiketoimia eri tarkoituksiin. Sekalainen kuormittavuus (mixed workload) koostuu Insert-, Delete-, ja Update-lauseissa sekä kyselyissä. (Kogan, Nicola & Schiefer 2007b.)

TPoX:n testaus suoritetaan käyttäjän määrittelemillä transaktioilla. TPoX tarjoaa tavalliset mallit transaktioille, jotka ovat tallennettu hakemistoon TPoX/WorkloadDriver/DB2. Käyttäjä voi muokata tai lisätä tarpeellisia toteutuksia transaktioille. Jokainen transaktio on määritelty omaan tiedostoonsa. Se voi sisältää yhden tai useamman lauseen. Jokainen lause päättyy merkkiin ”%”. (Kogan, Nicola & Schiefer 2007b.)

5 TPoX 2: -testi virtuaalikone ympäristössä

5.1 TPoX 2:n asennus

Tutkimus on tehty kannettavalla tietokoneella. Ensiksi koneelle asennettiin virtuaalikoneen, Oracle VM Virtual Boxin. Virtuaalikoneessa on DB2 LUW 10.1 ja TPoX 2.1-paketit. Uusimman version TPoX:stä voi ladata sivulta, <https://sourceforge.net/projects/tpox/files/>. TPoX -paketin purkamisen jälkeen kansion rakenne on seuraava.



Kuvio 12. TPoX kansiorakenne

Kansion rakenne on lyhyesti kuvattuna alla olevassa taulukossa (Taulukko 5).

Taulukko 5. TPoX-kansion rakenne lyhyesti kuvattuna

Datagen	Testiaineiston generointi välineet
DB2, MSSQL, Oracle	Tiettyyn tietokantaan liittyvät testi-tiedostot
GeneratedXML	Käytetty generoitujen testi-tiedostojen varastointiin
Schemas	Testissä käytetty schema-tiedosto
WorkloadDriver	Kuormitusajo-kansio, missä TPoX pääohjelma sijaitsee

5.2 Testauksen valmistelu

Seuraavissa kappaleissa kerrotaan testauksen valmistelusta ja sen vaiheista. Testaus on valmisteltu Nicolan työryhmän laatiman Quick Start Guiden -ohjeiden mukaan. (Kogan, Nicola & Schiefer 2007d.)

5.2.1 Testattavan tiedon lataaminen

Testauksessa asensin kannettavalleni virtuaalikonetta varten, Oracle VM Virtual Boxin. Virtuaalikoneessa on asennettuna DB2 Express C 10.1 ja TPoX 2.1-paketit. Testausmateriaali on valmiiksi asennettuna itse pakettiin. Testaustiedostojen puuttuessa, on ne ladattavissa osoitteessa <http://sourceforge.net/>. Tiedostot on generoitu datagenissa XXS kokoluokkaan (extra-extra small eli 1GB). Isompaan testitiedostojen tarpeeseen voi tiedostoja ajaa erikseen datagenissa generoidakseen lisää testitiedostoja omaan tarpeeseen. Tiedostot voi purkaa latauksen jälkeen ja siirtää generatedXML – kansion alle.

TPoX/generatedXML/custacc/batch-[1-2]

TPoX/generatedXML/order/batch-[1-2]

TPoX/generatedXML/security

TPoX/generatedXML/account/batch-1

5.2.2 Suorituskykytestin valmistelu

Tässä kappaleessa kuvataan alkuvalmistelut testausta varten. Ensiksi tulee varmistaa että Classpath pitää sisällään seuraavat Class- tai Jar-paketit:

db2jcc4.jar

TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar

TPoX/WorkloadDriver/classes

TPoX/DB2/classes

Suorituskykytesti tehdään kansion TPoX/WorkloadDriver:in alle. Ennen testin ajamista tulee määrittää oikea java classpath Terminaali-ikkunassa. Seuraavasta kuvasta selviää, kuinka polku määritetään.

```
db2inst1@db2v10:~> cd TPoX
db2inst1@db2v10:~/TPoX> cd WorkloadDriver
db2inst1@db2v10:~/TPoX/WorkloadDriver> export CLASSPATH=/opt/ibm/db2/V10.1/java/
db2jcc4.jar:/home/db2inst1/TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar:/home
/db2inst1/TPoX/WorkloadDriver/classes:/home/db2inst1/TPoX/DB2/classes
```

Kuva 13. Java Classpath -polun määrittäminen

Kuvio 14 ilmaisee polun määrittämisen suorituskykytestiä tehtäessä WorkloadDriver:ssä. Latasin testiaineiston mixed.xml properties-kansioon WorkloadDriver- kansion alle. Valmistelut testausta varten löytyy kokonaisuudessaan Liitteestä 5. Varsinaisen testin toteutus on kuvattu kappaleessa 6.3.

6 Tutkimus

6.1 Tutkimusongelma

Tämän opinnäytetyön tavoitteena on tutkia indeksien vaikutusta DB2 Express-C:n suorituskykyyn XML-päivitysten käsittelyssä.

Suorituskykyyn vaikuttavat tekijät on kuvattuna alla olevassa taulukossa (Taulukko 6).

Taulukko 6. Suorituskykyyn vaikuttavat tekijät

XML:n tallennusmuodot CLOB, binary XML, shredded	CLOB ja shredded – tallennusmuodot voivat vaikuttaa suoritustehoon, mutta tutkimus rajataan vain binary XML:n tutkimiseen
XML-indeksien vaikutus suorituskyky-suunnitelmaan	Ennako-odotuksena indeksit nopeuttavat XML-dokumentin löytymistä suuremmasta dokumenttijoukosta. Oletuksena indeksit myös hidastavat löydetyn dokumentin päivityst
Muutoksen validointi XML Schemaa vasten	Myös validointi vaikuttaa suoritustehoon, oletettavasti hidastaen suorituskykyä

Tutkimuksessa käytetään ohjelmistoa sellaisenaan kuin se on TPoX-paketissa. Työn rajaukseen valitaan muutama esimerkki, joiden avulla tutkitaan ennako-oletuksia. Esimerkkitapauksina käytetään muutamaa Nicolan tutkimusryhmän käyttämiä (Kogan, Nicola & Schiefer 2007a.) lauseita. Kyseiset lauseet löytyvät Liitteestä 1.

6.2 Tutkimusmenetelmät

Tutkimusmenetelmänä käytetään kvantitatiivista eli määrällistä tutkimusmenetelmää. Määrällisen tutkimuksen tarkoituksena on saada tietoa, joka on yleistettävissä. Vaatimus yleistettävyydestä juontuu empiirisen tietoteorian lähtökohtiin. Määrällinen yleistettävissä oleva tieto on kriteereiltään tilastollis-matemaattinen, jossa tietoa käsitellään tilastolli-

sina yksiköinä. Kaikki vähänkään subjektiiviseen tulkintaan viittaavat seikat häivytetään pois monin eri menetelmin. Käsitteet tilastoyksikkö, otos ja näyte ovat määrällisen tutkimuksen perusta. (Virtuaali ammattikorkeakoulu 2010.)

Tavoitteena yleiselle tarkastelulle on kokonaiskuvan antaminen aineistosta, sekä saada esiin siinä mahdollisesti esiintyvät piirteet, luokittelut sekä poikkeamat. Tutkimuksen kannalta tärkeää on tuoda esiin muuttuja-arvojen merkitsevyyden merkitys. Samalla luodaan mahdollisuus tutkimuksen luotettavuuden arviointiin. (Virtuaali ammattikorkeakoulu 2010.) Teoriatausta on kerätty alan kirjallisuudesta sekä julkaistuista artikkeleista.

Tutkimuksessa testausympäristönä toimii virtuaalikone, joka on ladattu DB2 on Campus – nettisivustolta. Virtuaalikoneeseen on asennettu DB2 Express-C 10.1, DataStudio 3.1.1 ja sen Java-kääntäjä sekä TPoX 2.1. Lisäksi Haaga-Heliassa virtuaalikoneeseen on asennettu PDF Reader, joka mahdollistaa PDF-dokumenttien lukemisen virtuaalikoneessa. Käyttöjärjestelmänä on SUSE Linux, jossa on Gnome Desktop. Keskusmuistia virtuaalikoneessa on 2GB. Virtuaalikoneessa käytetään TPoX:n pienintä kokoluokkaa XXS, joka on 1GB. Virtuaalikone käynnistetään kannettavaan asennetun Virtual Boxin avulla.

Alle olevassa taulukossa (Taulukko 7) on esitetty testauksessa käyttämäni tietokoneen tarkemmat spesifikaatiot.

Taulukko 7. Testauksessa käytetyn tietokoneen spesifikaatiot

Tietokone	Acer Aspire 3820T
Proessori	Intel Core i3-350M
Välimuisti	(2.26GHz, 3MB L3 cache)
Muisti	4GB DDR3
Kiintolevy	500 GB HDD
Käyttöjärjestelmä	Windows 7 Home Premium 64-bit

6.3 Testaus

Testiaineistona käytettiin WorkloadDriver/properties – kansioon tallennettua mixed.xml – tiedostoa. Mixed-kuormaan on lisätty kaksi transaktiota, insValidorder ja insNoValidcustacc, jotta nähdään myös validoinnin vaikutus suorituskykyyn. Ennen testiajoja tietokannasta otettiin varmuuskopio, joka palauttamalla saadaan testit ajettua samoilla lähtökohdilla.

Seuraava komento (Kuvio 15) ensin palauttaa tietokannan varmistuksista, jonka jälkeen ajaa mixed-kuorman 50 yhtäaikaisella käyttäjällä 10 minuutin ajan (600 sekuntia). Statiikkaa tulostuu 30 sekunnin) välein stats.txt – tiedostoon. Ensimmäiset 3 minuuttia (180 sekuntia) eivät ole mukaan luettuna keskimääräiseen suorituskykyyn vaan kuormituksen mittaaminen alkaa vasta tämän jälkeen. Tulokset testiajon on kokonaisuudessaan esitetty statistiikkana Liitteessä 3.

```
# Restoring the database
db2stop
db2start
export DBBACKUP=/home/db2inst1/sqllib/backup
export TIMESTAMP=20131115222555
date
time db2 -v RESTORE DATABASE tpox FROM $DBBACKUP TAKEN AT $TIMESTAMP WITHOUT ROLLING FORWARD WITHOUT PROMPTING
date

cd
cd TPox/workloadDriver
# Before running any tests, be sure to adjust and export your CLASSPATH to include the following:
export CLASSPATH=/opt/ibm/db2/v10.1/java/db2jcc4.jar:/home/db2inst1/TPox/workloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPox/workloadDriver/classes:/home/db2inst1/TPox/DB2/classes
#
java net.sf.tpox.workload.core.workloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -u 50
-w properties/mixed.xml -ti 600 -r 180 -pd 30
```

Kuva 14. Testiajo ilman indeksejä

Kuviossa 16 näkyy, kuinka testiajo indeksien kanssa on suoritettu. Ensin palautettiin tietokanta varmistuksista, jotta saadaan täysin sama alkutilanne, kuin ensimmäisessä testiajossa. Tämän jälkeen otettiin yhteys tietokantaan, jonka jälkeen indeksit lisättiin. Indeksien lisäyksen jälkeen ajettiin sama komento kuin ensimmäisessä testissä. Tulokset kuormituksesta indeksien kanssa on esitetty kokonaisuudessa Liitteessä 3.

```

# Restoring the database
db2stop
db2start
export DBBACKUP=/home/db2inst1/sql11b/backup
export TIMESTAMP=20131115222555
date
time db2 -v RESTORE DATABASE tpox FROM $DBBACKUP TAKEN AT $TIMESTAMP WITHOUT ROLLING FORWARD WITHOUT PROMPTING
date

# Create the XML indexes
cd TPox/DB2/dd1
db2 connect to tpox
db2 -td% -f createindexes.sql

cd
cd TPox/workloadDriver
# Before running any tests, be sure to adjust and export your CLASSPATH to include the following:
export CLASSPATH=/opt/ibm/db2/v10.1/java/db2jcc4.jar:/home/db2inst1/TPox/workloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPox/workloadDriver/classes:/home/db2inst1/TPox/DB2/classes
#
java net.sf.tpox.workload.core.workloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -u 50
-w properties/mixed.xml -ti 600 -r 180 -pd 30

```

Kuva 15. Testiajo indeksien kanssa

6.4 Tulokset

Tulokset on esitetty kokonaisuudessaan Liitteessä 3. Tuloksista voidaan pintapuolisella tarkastelulla todeta hypoteesien pitäneen paikkansa. Indeksit nopeuttivat XML-käsittelyä. Insert – käsittelyitä indeksit hidastivat. Validointi myös oletetusti hidasti käsittelyä.

Tarkasteltaessa testien статистиikkaa (Liite 3) on huomattavissa, kuinka ilman indeksejä tehdyssä testiajossa läpikäytyjä rivejä on huomattavasti vähemmän kuin testiajossa indeksien kanssa. Indeksien nopeuttaessa kyselyitä, ehditään testiajossa myös käydä useampia rivejä läpi.

6.4.1 Query -lauseiden käsittely

Alla olevassa taulukossa (Taulukko 8) on esitetty keskimääräiset suoritusajat Query -lauseille ilman indeksejä ja indeksien kanssa. Taulukosta näkee selkeästi indeksien nopeuttavan kyselyiden käsittelyä merkittävästi. Lisäksi läpikäytyjen rivien määrä testiajossa kasvoi indeksien kanssa 7,9 % alkuperäisestä. Indeksit nopeuttivat kyselyiden käsittelyä 97,88 %.

Taulukko 8. Keskimääräiset suoritusajat kyselyille

Tr. #	Tr. Nimi	Ilman indeksejä	XML-indeksien kanssa
1	get_order_sqlxml	0.0820	0.0010
2	get_security_sqlxml	0.0300	0.0009

3	customer_profile_sqlxml	0.0779	0.0018
4	search_securities_sqlxml	0.0623	0.0030
5	account_summary_sqlxml	0.0784	0.0019
6	get_security_price_sqlxml	0.0297	0.0009
7	customer_max_order_sqlxml	0.1690	0.0017

Alla olevassa kuviossa (Kuvio 17) on esitettyä transaktion 1 (get_order) suoritussuunnitelma ilman indeksejä. Kuvioista näkyy suunnitelmalle laskettu timeron-arvo, 1742,89.

```

Access Plan:
-----
Total Cost:          1742.89
Query Degree:        1

      Rows
      RETURN
      ( 1)
      Cost
      I/O
      |
      0.961089
      NLJOIN
      ( 2)
      1742.89
      252
    /-----+-----\
    0.961089          1
    ^NLJOIN          GRPBY
    ( 3)             ( 6)
    1736             6.88901
    252              1
  /-----+-----\
  250  0.00384436  0.988
  TBSCAN  XSCAN  XSCAN
  ( 4)   ( 5)   ( 7)
  13.8871 6.88845 6.88883
  2       1       1
  |
  250
TABLE: DB2INST1
ORDER
Q6

```

Kuva 16. Suoritusuunnitelma ilman indeksejä

Kuviossa 18 on esitettyä suoritusuunnitelma samalle transaktiolle 1 indeksien kanssa. Kuten kuvioista näkyy, indeksien kanssa timeron-arvo on huomattavasti pienempi. Indeksit pienensivät suoritusuunnitelman timeron-arvoa 99 %.

```

Access Plan:
-----
Total Cost:          20.6855
Query Degree:       1

      Rows
      RETURN
      ( 1)
      Cost
      I/p
      |
      0.961089
      NLJOIN
      ( 2)
      20.6855
      3
    /-----\
    0.961089      1
    ^NLJOIN      GRPBY
    ( 3)         ( 9)
    13.7963     6.88901
    2           1
  /-----\
  1 0.961089 0.988
  FEICH XSCAN XSCAN
  ( 4) ( 8) ( 10)
  6.90783 6.88845 6.88883
  1 1 1
/-----\
1 250
RIDSCN TABLE: DB2INST1
( 5) ORDER
0.01933 Q6
|
1
SORT
( 6)
0.019026
|
1
XISCAN
( 7)
0.0183114
|
250
XMLIN: DB2INST1
ORDER_ID
Q6

```

Kuva 17. Suoritussuunnitelma indeksien kanssa

6.4.2 Update -lauseiden käsittely

Alla olevassa taulukossa (Taulukko 9) on esitetty keskimääräiset suoritusajat Update-lauseille ilman indeksejä ja indeksien kanssa. Taulukosta näkee indeksien nopeuttaneen päivitysten käsittelyä merkittävästi. Indeksit nopeuttivat käsittelyä 97,8 %.

Taulukko 9. Keskimääräiset suoritusajat päivityksille

Tr. #	Tr. Nimi	Ilman indeksejä	XML-indeksien kanssa
8	U1CloseAccount	0.1205	0.0009
9	U2OpenAccount	0.0835	0.0060
10	U3SecurityPrice	0.0317	0.0014
11	U4OrderStatus	0.0856	0.0016
12	U5BuySecurity	0.1221	0.0013
13	U6SellSecurity	0.1234	0.0012

6.4.3 Delete - lauseiden käsittely

Alle oleva taulukko (Taulukko 10) esittää keskimääräiset suoritusajat Delete-lauseille ilman indeksejä ja indeksien kanssa. Myös poistoissa indeksit nopeuttivat transaktioiden käsittelyä huomasti. Käsittelyt nopeutuivat keskimäärin 97,81 %.

Taulukko 10. Keskimääräiset suoritusajat poistoille

Tr. #	Tr. Nimi	Ilman indeksejä	XML-indeksien kanssa
14	delcustacc	0.0805	0.0018
15	delorder	0.0857	0.0009

6.4.4 Insert – lauseiden käsittely

Alla olevasta taulukosta (Taulukko 11) näkee keskimääräiset suoritusajat Insert - lauseissa ilman indeksejä ja indeksien kanssa. Taulukosta näkee myös, kuinka validointi vaikuttaa suorituskykyyn. Tuloksien mukaan validointi indeksien kanssa on nopeampaa kuin käsittely ilman validointia. Transaktiot 16 ja 17 validoidaan. Transaktiot 18 ja 19 käsitellään ilman validointia. Indeksit hidastivat Insert-lauseiden käsittelyä 21,02 %. Order-dokumentin Schema on laaja, josta johtuen validointi sitä vasten oli huomattavasti raskaampaa. Taulukosta 11 nähdään transaktion 17 validoinnin kanssa kestäneen yli 16 kertaa enemmän kuin transaktion 19 ilman validointia käsittely.

Taulukko 11. Keskimääräiset suoritusajat lisäyksille

Tr. #	Tr. Nimi	Ilman indeksejä	XML-indeksien kanssa
16	insValidcustacc	0.0282	0.0192
17	insValidorder	0.1350	0.1211
18	insNoValidcustacc	0.0139	0.0176
19	insNoValidorder	0.0065	0.0075

7 Johtopäätökset

IBM:n (2007, 176.) mukaan useiden indeksien luonnissa XML-sarakkeeseen tulee olla varovainen, koska se voi vähentää Insert-, Update- ja Delete-lauseiden suorituskykyä. Lisäksi indeksit vievät tilaa, josta johtuen turhien indeksien tekemistä tulee välttää. Tulokset osoittivat indeksien hidastavan Insert-käsittelyitä. Käsittelyiden hidastuminen tosin on marginaalista. Query-, Update- ja Delete-käsittelyissä indeksit nopeuttivat suorituskykyä huomattavasti dokumentin haun osalta. Update- ja Delete-käsittelyn vaikutus XML-indeksin päivitykseen oli haun nopeutumiseen verrattuna merkityksetöntä. Johtopäätöksenä voi todeta indeksien käytön nopeuttavan kyselyiden ja päivitysten käsittelyä, jonka takia niiden käyttö olisi suotavaa.

XML-validointi tarkistaa XML-dokumentin rakenteen, sisällön ja tietotyyppien olevan valideja XML Scheman suhteen. XML-validointi lisäksi poistaa ylimääräiset rivit dokumentista. Validointi on valinnaista, mutta suositeltavaa kun kyseessä on datan eheys. Validointi takaa XML-dokumentin noudattavan XML Schemaan määritettyjä sääntöjä sekä olevan oikein muodostettu. (IBM Infocenter 2013c.) Verrattaessa transaktioita 16 & 18 sekä 17 & 19, aiheuttaa validointi lisäkuormitusta Insert-lauseiden käsittelyssä.

8 Yhteenveto

8.1 Kooste

Nicolan tutkimusryhmän mukaan XML:n tietokantateknologia tukee tehokkaasti XML:n käyttöä. XML-tietokantateknologialla on kasvava kysyntä kaupallisten yritysten kuten rahoitus-, pankki-, teollisuus-, koulu-, hallitus ja terveydenhoidon keskuudessa. Tästä johtuen ajankohtaisena aiheena voisi kasvavan kysynnän vuoksi pitää XML-tietokannan kehittämistä, jossa suorituskyky on painopisteenä. (Nicola, Kogan & Schiefer 2007c, 1.)

Tämän opinnäytetyön tarkoituksena oli tutkia indeksien vaikutusta DB2:n suorituskyky-suunnitelmaan XML -päivitysten käsittelyssä. Ennako-odotuksena oli indeksien nopeuttavan XML-dokumentin löytymistä suuremmasta dokumenttijoukosta, mutta myös hidastavan löydetyn dokumentin päivittämistä. Myös validoinnin odotettiin vaikuttavan suoritustehoon, oletettavasti hidastaen sitä.

Tulokset osoittivat myös indeksien hidastavan Insert-käsittelyitä. Käsittelyiden hidastuminen on marginaalista. Query-, Update- ja Delete-käsittelyissä indeksit kuitenkin nopeuttivat suorituskykyä huomattavasti. Johtopäätöksenä todettiin indeksien käytön nopeuttavan kyselyiden ja päivitysten käsittelyä, jonka takia niiden käyttö olisi suotavaa.

8.2 Suositukset

Tämän opinnäytetyön perusteella jatkotoimenpiteiksi suositeltavaa olisi laajentaa testausta myös Oracleen sekä My SQL:n. Molemmissa XML-käsittely eroaa DB2:n XML-käsittelyssä. Tarvittavat materiaalit testauksen suorittamiseen myös näillä tietokantaohjelmistoilla löytyvät TPoX-paketista.

TPoX-ohjelmiston voisi soveltaa myös perinteisten SQL-tehtävien kuorman tutkimiseen.

8.3 Oppimisen analysointi

Projektisuunnitelmassa kartoitettiin mahdolliset riskit opinnäytetyöprojektille. Riskeiksi oli arvioitu aikataulun pettäminen, työn häviäminen tai tuhoutuminen sekä tekijän sairastuminen. Riskeistä aikataulun pettäminen sekä sairastuminen tapahtuivat. Projekti ei pysynyt määritetyssä aikataulussa, johon vaikutti omalta osaltaan myös riskiksi arvioitu sairastuminen. Aikataulun pettämiseen suurimmalta osalta kuitenkin vaikutti kokopäiväinen työssäkäynti projektin ohella sekä tekniset ongelmat virtuaalikoneen kanssa. Tämä vaikutti projektiin siten, ettei työhön pystytty käyttämään suunniteltua 34 tuntia viikoittain. Projektille oli varattu aikaa 12 viikkoa, yhteensä 400 työtuntia. Projektiin todellisuudessa kului aikaa 76 viikkoa.

Tämän opinnäytetyöprojektin aikana olen huomannut aikataulutuksen ja siinä pysymisen tärkeyden. Aikataulun tulee olla realistinen. Sitä laatiessa tulee ottaa huomioon myös projektin ulkoiset tekijät, esimerkiksi henkilökohtaiset velvoitteet kuten työssäkäynti. Aikataulun pettäessä tulisi olla huolella laadittu varasuunnitelma, jota noudattaen projektin eteneminen ei viivästyisi.

Projektin riskejä kartoittaessa tulisi myös huomioida projektin ulkoiset tekijät, esimerkiksi mahdolliset tekniset ongelmat. Omassa projektissa riskiksi osoittautuivat myös tekniset ongelmat virtuaalikoneen kanssa, jota ei ollut lainkaan huomioitu riskejä kartoitettaessa. Teknisiä vaikeuksia esiintyi virtuaalikonealustan kanssa. Alun perin oli suunniteltu käytettävän Oraclen VM Virtual Boxia., joka ei soveltunutkaan kannettavani kanssa. Kokeilimme myös käyttää VMware Playeriä, joka ei myöskään soveltunut kannettavalleni. Kummatkin virtuaalikonealustat kadottivat kannettavasta ensin langattoman verkkokortin ajurin, jonka jälkeen katosi myös verkkokortin ajurit. Kannettavaa oli tarkoitus käyttää testauksen lisäksi myös opinnäytetyön kirjoittamiseen. Tästä johtuen pääsy Internetiin kannettavalla oli olennaista. Vastoinkäymisten jälkeen päätimme vaihtaa kannettavaa jolla itse testaus suoritettiin.

Teknisten ongelmien lisäksi ongelmia ilmeni myös Quick Start Guiden ohjeen kanssa. Ohjeessa oli yhteensopimattomia ohjeita SUSE Linux –käyttöjärjestelmälle. Esimerkkinä tästä voisi mainita tiedostojen purkamisen. Ohjeissa neuvottiin käyttämään komen-

toa unzip -a, joka ei ollut yhteensopiva SUSE Linuxin kanssa. Quick Start Guiden yhteensopimattomat ohjeet Linux-käyttöjärjestelmän kanssa veivät projektilta aikaa kuukausia.

Projektin aikana opin ettei aikatauluni ollut realistinen, eikä riskeiltä toipumista oltu suunniteltu riittävästi. Kaikkia riskejä ei myöskään ollut otettu huomioon niitä kartoitettaessa. Tulevissa projekteissa ymmärrän aikataulun realistisen suunnittelemisen tärkeyden. Lisäksi osaan ottaa huomioon myös projektin ulkopuoliset tekijät ja suunnitella riskeistä toipumisen perusteellisesti.

Lähteet

Conor O'Mahony 2008. Native XML versus CLOB and Shredding. Luettavissa:
<http://nativexmldatabase.com/?s=clob>. Luettu: 17.01.2013.

Duffoux, A., Keramopolous, E., Kurki, M., Laiho, M. & Laux F. 2011. XML, SQL/XML and the Big Three. Luettavissa:
http://www.dbtechnet.org/labs/xml_lab/XMLandDatabasesTutorial.pdf. Luettu: 24.5.2012.

IBM 2007. DB2 9 pureXML Guide. Luettavissa:
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247315.pdf>. Luettu: 20.9.2012.

IBM, Infocenter 2012a. pureXML / Querying XML data / Introduction to XQuery. Luettavissa:
<http://pic.dhe.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=%2Fcom.ibm.db2.luw.xml.doc%2Fdoc%2Fqxgrover.html>. Luettu: 11.9.2012.

IBM Infocenter 2012b. pureXML / Indexing XML data / Database objects associated with XML data indexing / Other database objects associated with XML columns. Luettavissa:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.xml.doc/doc/c0024071.html>. Luettu: 14.9.2012.

IBM Infocenter 2012c. Databases / Table spaces / Creating table spaces. Luettavissa:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=%2Fcom.ibm.db2.luw.admin.dboobj.doc%2Fdoc%2Fc0052381.html>. Luettu: 16.9.2012.

IBM Infocenter 2013a. Databases / Table spaces. Luettavissa:
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.admin.dboobj.doc/doc/c0004935.html>. Luettu: 5.1.2013.

IBM Infocenter 2013b. Databases / Table spaces / Designing table spaces / Table space page sizes. Luettavissa:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=%2Fcom.ibm.db2.luw.admin.dboobj.doc%2Fdoc%2Fc0052381.html>. Luettu: 6.2.2013.

IBM Infocenter 2013c. pureXML / Inserting XML data / XML data integrity / XML validation. Luettavissa:

<http://pic.dhe.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=%2Fcom.ibm.db2.luw.xml.doc%2Fdoc%2Fc0023539.html>. Luettu: 17.1.2013.

IBM Infocenter 2013d. Cost estimation in Query Patroller. Luettavissa:

<http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.uodb.doc/opt/c0010441.htm>. Luettu: 11.11.2013.

Kumar-Chatterjee, P. & Nicola, M. 2009. DB2 pureXML Cookbook. Pearson plc. Michigan.

Laamanen, H. 2005. XML-verkkotietokantojen suorituskykyvertailu. Diplomityö. Lappeenrannan Teknillinen Yliopisto. Lappeenranta.

Nicola, M. 2012. How to Validate XML Document in Database Queries. Luettavissa: <http://nativexmldatabase.com/2012/09/13/how-to-validate-xml-documents-in-database-queries>. Luettu: 30.10.2012.

Nicola, M., Kogan, I. & Schiefer, B. 2007a. XML Database Benchmark: Transaction Processing over XML (TPoX). Luettavissa: <http://tpox.sourceforge.net/>. Luettu: 11.9.2012.

Nicola, M., Kogan, I. & Schiefer, B. 2007b. XML Database Benchmark: Transaction Processing over XML (TPoX) / Workload Driver. Luettavissa: <http://tpox.sourceforge.net/>. Luettu: 17.9.2012.

Nicola, M., Kogan, I. & Schiefer, B. 2007c. An XML Transaction Processing Benchmark. Luettavissa: <http://tpox.sourceforge.net/>. Luettu: 20.9.2012.

Nicola, M., Kogan, I. & Schiefer, B. 2007d. XML Database Benchmark: Transaction Processing over XML (TPoX) / Quick Start Guide. Luettavissa: <http://tpox.sourceforge.net/>. Luettu: 15.9.2012.

O'Mahony, C. 2008. Native XML versus CLOB and Shredding. Luettavissa: <http://nativexmldatabase.com/?s=clob>, Luettu: 17.01.2013

Virtuaali ammattikorkeakoulu 2010. Ylemman AMK-tutkinnon metodifoorumi / 3. Ammatillinen tiedonhankinta / Kvantitatiivisen analyysin perusteet. Luettavissa: <http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/0709019/1193463890749/1193464131489/1194289328583/1194289824724.html>. Luettu: 30.10.2013.

Liitteet

Liite 1. Testikyselyt

Kyselyt

Q1: get_order

```
SELECT XMLQUERY
    ('declare namespace o="http://www.fixprotocol.org/FIXML-4-4";
    for $ord in $doc/o:FIXML
    return $ord/o:Order'
    PASSING odoc AS "odoc")
FROM order
WHERE XMLEXISTS
    ('declare namespace o="http://www.fixprotocol.org/FIXML-4-4";
    $doc/o:FIXML/o:Order[@ID=$id]
    'PASSING odoc AS "odoc", cast(? as varchar(10)) as "id")
```

Q2: get_security

```
SELECT XMLQUERY
    ('declare default element namespace "http://tpox-benchmark.com/security";
    for $sec in $doc/Security
    return $sec'
    PASSING sdoc AS "sdoc")
FROM security
WHERE XMLEXISTS
    ('declare default element namespace "http://tpox-benchmark.com/security";
    $sdoc/Security[Symbol=$sym]
    'PASSING sdoc AS "sdoc", cast(? as varchar(10)) as "sym")
```

Q3: customer_profile

SELECT XMLQUERY

```

('declare default element namespace "http://tpox-benchmark.com/custacc";
for $cust in $cadoc/Customer
return
  <Customer_Profile CUSTOMERID="{ $cust/@id }">
    { $cust/Name }
    { $cust/DateOfBirth }
    { $cust/Gender }
    { $cust/CountryOfResidence }
    { $cust/Languages }
    { $cust/Addresses }
    { $cust/EmailAddresses }
  </Customer_Profile>'
PASSING cadoc AS "cadoc")

```

FROM custacc

WHERE XMLEXISTS

```

('declare default element namespace "http://tpox-benchmark.com/custacc";
$cadoc/Customer[@id=$id]
PASSING cadoc AS "cadoc", cast (? as double) as "id" )

```

Q4: search_securities

SELECT XMLQUERY

```

('declare default element namespace "http://tpox-benchmark.com/security";
for $sec in $sdoc/Security
return
  <Security>
    { $sec/Symbol }
    { $sec/Name }
    { $sec/SecurityType }
    { $sec/SecurityInformation//Sector }
    { $sec/PE }
    { $sec/Yield }
  </Security>'
PASSING sdoc AS "sdoc")

```

FROM security

WHERE XMLEXISTS

```

('declare default element namespace "http://tpox-benchmark.com/security";
$sdoc/Security[SecurityInformation/*/Sector=$sector and Yield>$yield]
PASSING      sdoc AS "sdoc",
              cast (? as varchar(25)) as "sector",
              cast (? as double) as "yield" )

```

AND XMLEXISTS

```

('declare default element namespace "http://tpox-benchmark.com/security";
$sdoc/Security/PE[.>=$pe1 and .<$pe2]
PASSING      sdoc AS "sdoc",
              cast (? as double) as "pe1",
              cast (? as double) as "pe2")

```

Q5: account_summary

SELECT XMLQUERY

```

('declare default element namespace "http://tpox-benchmark.com/custacc";
for $cust in $cadoc/Customer

```



```

return
  <Customer> {$cust/@id}
    {$cust/Name}
  <Customer_Securities>
    { for $account in $cust/Accounts/Account
      return
        <Account
          BAL-
ANCE=" {$account/Balance/OnlineActualBal}"
          AC-
COUNT_ID=" {$account/@id}">
          <Securities>
            {Sac-
count/Holdings/Position/Name}
          </Securities>
        </Account>
      }
    </Customer_Securities>
  </Customer>'
PASSING cadoc AS "cadoc")
FROM custacc
WHERE XMLEXISTS
  ('declare default element namespace "http://tpox-benchmark.com/custacc";
  $cadoc/Customer[@id=$id]'
  PASSING cadoc AS "cadoc", cast (? as integer) as "id")

```

Q6: get_security_price

```

SELECT XMLQUERY
  ('declare namespace s="http://tpox-benchmark.com/security";
  for $sec in $sdoc/s:Security
  return
    <print>The open price of the security " {$sec/s:Name/text()} " is
      {$sec/s:Price/s:PriceToday/s:Open/text()} dollars</print>'
  PASSING sdoc AS "sdoc")
FROM security
WHERE XMLEXISTS
  ('declare namespace s="http://tpox-benchmark.com/security";
  $sdoc/s:Security[s:Symbol=$sym]'
  PASSING sdoc AS "sdoc", cast (? as varchar(10)) as "sym")

```

Q7: customer_max_order

```

SELECT DECIMAL(CAST(MAX(price) AS INTEGER), 15, 2) AS maxprice
FROM
  (SELECT
    XMLCAST(XMLQUERY
      ('declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";
      let $orderprice := $odoc/FIXML/Order/OrdQty/@Cash
      return $orderprice'
      PASSING odoc AS "odoc")
    AS DOUBLE) AS price
  FROM custacc, order
  WHERE XMLEXISTS
    ('declare namespace c="http://tpox-benchmark.com/custacc";
    $cadoc/c:Customer[@id=$id]'
    PASSING cadoc AS "cadoc", cast (? as double) as "id")
  AND XMLEXISTS

```

```
(declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";
declare namespace c="http://tpox-benchmark.com/custacc";
$odoc/FIXML/Order[@Acct=$cadoc/c:Customer/c:Accounts/c:Account/@id/fn:string()]'
PASSING cadoc AS "cadoc", odoc AS "odoc")) AS T
```

Update-lauseet

U1: close_account

```
UPDATE custacc
SET cadoc = XMLQUERY(' declare default element namespace "http://tpox-benchmark.com/custacc";
transform
copy $c := $doc
modify
(: do not delete the account if the customer has only one account :)
if (count($c/Customer/Accounts/Account) >= 2)
then do delete
        $c/Customer/Accounts/Account[@id="104138966"]
else ()
return $c'
PASSING cadoc AS "doc")
WHERE XMLEXISTS
('declare default element namespace "http://tpox-benchmark.com/custacc";
$cadoc/Customer/Accounts/Account[@id="104138966"]'
PASSING cadoc AS "cadoc")
```

U2: open_account

```
UPDATE custacc
SET cadoc = XMLVALIDATE(XMLQUERY('declare default element namespace
"http://tpox-benchmark.com/custacc";
transform
copy $c := $doc
modify
(: do not add the account if it exceeds the max of seven accounts :)
if (count($c/Customer/Accounts/Account) < 7)
then do insert
        <Account id="104138966">
                <Category>9</Category>
                <AccountTitle>Dr Vineeta Krablin YEN</AccountTitle>
                <ShortTitle>Krablin YEN</ShortTitle>
                <Mnemonic>KrablinYEN</Mnemonic>
                <Currency>YEN</Currency>
                <CurrencyMarket>5</CurrencyMarket>
                <OpeningDate>1993-01-31</OpeningDate>
                <AccountOfficer>Lorraine Bos</AccountOfficer>
                <LastUpdate>2003-12-10T09:07:20</LastUpdate>
                <Balance>
                        <OnlineActualBal>932797</OnlineActualBal>
                        <OnlineClearedBal>852847</OnlineClearedBal>
                        <WorkingBalance>739379</WorkingBalance>
                </Balance>
                <Passbook>Yes</Passbook>
                <gValueDate>
                        <mValueDate>
```

```

        <ValueDate>2001-08-21</ValueDate>
        <CreditMovement>77452.85</CreditMovement>
        <ValueDatedBal>210573</ValueDatedBal>
    </mValueDate>
</gValueDate>
<ChargeCcy>YEN</ChargeCcy>
<InterestCcy>YEN</InterestCcy>
<AllowNetting>Yes</AllowNetting>
<gInputter>
    <Inputter>Mostefa Kruseman</Inputter>
    <Inputter>Mostefa Kruseman</Inputter>
</gInputter>
<Holdings>
    <Position>
        <Symbol>OIIM</Symbol>
        <Name>Cendant Corporation</Name>
        <Type>Stock</Type>
        <Quantity>2020.072</Quantity>
    </Position>
</Holdings>
</Account>
into $c/Customer/Accounts
else ()
return $c'
PASSING cadoc AS "doc")
according to xmlschema id custacc_schema)
WHERE XMLEXISTS
('declare default element namespace "http://tpox-benchmark.com/custacc";
$cadoc/Customer[@id=1011]'
PASSING cadoc AS "cadoc")

```

U3: price_change

```

UPDATE security
SET sdoc = XMLQUERY('declare default element namespace "http://tpox-benchmark.com/security";
transform
copy $sdoc := $doc
modify
    let $price := $sdoc/Security/Price
    let $newlasttrade := $price/PriceToday/Open*0.95
    return
    (
    do replace value of
        $price/LastTrade with $newlasttrade,
    do replace value of
        $price/Ask with $newlasttrade*1.01,
    do replace value of
        $price/Bid with $newlasttrade*0.99)
return $sdoc'
PASSING sdoc AS "doc")
WHERE XMLEXISTS
('declare default element namespace
"http://tpox-benchmark.com/security";

```

```

$doc/Security[Symbol="OIIM"]
PASSING sdoc AS "sdoc"
)

```

U4: order_status

```

UPDATE order
SET odoc = XMLVALIDATE(XMLQUERY('declare default element namespace
"http://www.fixprotocol.org/FIXML-4-4";
transform
copy $o := $doc
modify (
do replace value of $o/FIXML/Order/@SolFlag with "N",
do replace value of $o/FIXML/Order/Instrmt/@Src with "C")
return $o'
PASSING odoc AS "doc")
according to xmlschema id order_schema)
WHERE XMLEXISTS
('declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";
$doc/FIXML/Order[@ID="103415"]'
PASSING odoc AS "doc")

```

U5: buy_security

```

UPDATE custacc
SET cadoc = XMLQUERY(
'declare default element namespace "http://tpox-benchmark.com/custacc";
declare namespace s = "http://tpox-benchmark.com/security";
transform
copy $c := $doc
modify
let $acct := $c/Customer/Accounts/Account[@id="104138966"]
let $actualbalance := $acct/Balance/OnlineActualBal
let $clearedbalance := $acct/Balance/OnlineClearedBal
let $workingbalance := $acct/Balance/WorkingBalance
let $mvaluedate := $acct/gValueDate/mValueDate[last()]
let $currentdate := fn:current-date()
(: need to get security information :)
let $sec := db2-fn:xmlcolumn("SECURITY.SDOC")/s:Security[
s:Symbol="OIIM"]
return (
(: do not buy if it exceeds the maximum of 10 holdings per acc :)
if (count($acct/Holdings/Position)=10 and
count($acct/Holdings/Position[Symbol="OIIM"])=0)
then ()
else (
(: add new position if no shares of this security exist in this acc :)
if (count($acct/Holdings/Position[Symbol="OIIM"])=0) then
do insert
<Position>
<Sym-
bol> {$sec/s:Symbol/text()}</Symbol>
<Name> {$sec/s:Name/text()}</Name>

```

```

<Type> {$sec/s:SecurityType/text()} </Type>
<Quantity>50</Quantity>
</Position>
into $acct/Holdings
(: if shares of this security existed in this acc, add the new shares:)
else
do replace value of
$acct/Holdings/Position[Symbol="OIIM"][1]/Quantity
with
xs:decimal($acct/Holdings/Position[Symbol="OIIM"][1]/Quantity
+ 50),
(: now update the account balances and timestamp :)
do replace value of $acct/LastUpdate with fn:current-dateTime(),
do replace value of $actualbalance with
xs:decimal($actualbalance+(50*$sec/s:Price/s:Ask)),
do replace value of $clearedbalance with
xs:decimal($clearedbalance+(50*$sec/s:Price/s:Ask)),
do replace value of $workingbalance with
xs:decimal($workingbalance+(50*$sec/s:Price/s:Ask)),
do replace $mvaluedate with
<mValueDate>
<ValueDate> {$currentdate}
</ValueDate>
<CreditMovement> {xs:decimal(156882.77)} </CreditMovement>
<ValueDatedBal> {xs:decimal(45736.85)} </ValueDatedBal>
</mValueDate> ) )
return $c'
PASSING cadoc AS "doc")

WHERE XMLEXISTS (
'declare default element namespace "http://tpox-benchmark.com/custacc";
$cadoc/Customer/Accounts/Account[@id="104138966"]'
PASSING cadoc AS "cadoc")

U6: sell_security
UPDATE custacc
SET cadoc = XMLQUERY (
'declare default element namespace "http://tpox-benchmark.com/custacc";
declare namespace s= "http://tpox-benchmark.com/security";
transform
copy $c := $doc
modify
let $acct := $c/Customer/Accounts/Account[@id="104138966"]
let $actualbalance := $acct/Balance/OnlineActualBal
let $clearedbalance := $acct/Balance/OnlineClearedBal
let $workingbalance := $acct/Balance/WorkingBalance
let $mvaluedate := $acct/gValueDate/mValueDate[last()]
let $currentdate := fn:current-date()
(: need to get security information :)

```

```

let $sec := db2-fn:xmlcolumn("SECURITY.SDOC")/s:Security[
    s:Symbol=$acct/Holdings/Position[1]/Symbol/fn:string()]
return (
    (: do not sell if it depletes the last position in this account :)
    if (count($acct/Holdings/Position) < 2 and
        $acct/Holdings/Position[1]/Quantity <= 50) then ()
    else (
        (: delete the position if the sell redeems all shares held :)
        if ($acct/Holdings/Position[1]/Quantity <= 50) then
            do delete $acct/Holdings/Position[1]
        (: otherwise subtract the sold shares :)
        else
            do replace value of $acct/Holdings/Position[1]/Quantity
            with xs:decimal($acct/Holdings/Position[1]/Quantity - 50),
            (: now update the account balances and timestamp :)
            do replace value of $acct/LastUpdate with fn:current-dateTime(),
            do replace value of $actualbalance with
                xs:decimal($actualbalance - (50*$sec/s:Price/s:Bid)),
            do replace value of $clearedbalance with
                xs:decimal($clearedbalance - (50*$sec/s:Price/s:Bid)),
            do replace value of $workingbalance with
                xs:decimal($workingbalance - (50*$sec/s:Price/s:Bid)),
            do replace $mvaluedate with
                <mValueDate>
                    <ValueDate> {$currentdate} </ValueDate>

            <CreditMovement> {xs:decimal(156882.77)} </CreditMovement>

            <ValueDatedBal> {xs:decimal(45736.85)} </ValueDatedBal>
                </mValueDate> ) )

return $c'
PASSING cadoc AS "doc"
)
WHERE XMLEXISTS
('declare default element namespace "http://tpox-benchmark.com/custacc";
$cadoc/Customer/Accounts/Account[@id="104138966"]'
PASSING cadoc AS "cadoc")

```

Liite 2. Mixed.xml – tiedoston sisältö

```
<?xml version="1.0" encoding="UTF-8"?>
<workload name="mixed" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="workload_properties.xsd">
  <collections>
    <collection name="custacc">
      <ids first="1002" last="51001"/>
      <files dir="../generatedXML/custacc" firstfileid="2" lastfileid="2" docsperfile="50000" fileprefix="batch-" filepostfix="xml.offset"/>
    </collection>
    <collection name="order">
      <ids first="103282" last="603281"/>
      <files dir="../generatedXML/order" firstfileid="2" lastfileid="2" docsperfile="500000" fileprefix="batch-" filepostfix="xml.offset"/>
    </collection>
    <collection name="account">
      <ids first="104130876" last="104228145"/>
      <files dir="../generatedXML/account" firstfileid="1" lastfileid="1" docsperfile="50000" fileprefix="batch-" filepostfix="xml.offset"/>
    </collection>
  </collections>
  <collection name="security"/>
  <transactions>
    <transaction id="1" weight="5">
      <location>../DB2/queries_pm/get_order_sqlxml.xqr</location>
      <parameters>
        <parameter>
          <ids collection="order"/>
        </parameter>
      </parameters>
    </transaction>
    <transaction id="2" weight="5">
      <location>../DB2/queries_pm/get_security_sqlxml.xqr</location>
      <parameters>
        <parameter>
          <file>../WorkloadDriver/input/security_symbols.txt</file>
        </parameter>
      </parameters>
    </transaction>
    <transaction id="3" weight="8">
      <location>../DB2/queries_pm/customer_profile_sqlxml.xqr</location>
      <parameters>
        <parameter>
          <ids collection="custacc"/>
        </parameter>
      </parameters>
    </transaction>
  </transactions>
</workload>
```

```

        </parameter>
    </parameters>
</transaction>

<transaction id="4" weight="10">
    <location>../DB2/queries_pm/search_securities_sqlxml.xqr</location>
    <parameters>
        <parameter>
            <file>../WorkloadDriver/input/security_sectors.txt</file>
        </parameter>
        <parameter>
            <uniformint min="20" max="23"/>
        </parameter>
        <parameter>
            <uniformint min="25" max="27"/>
        </parameter>
        <parameter>
            <uniformint min="4" max="6"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="5" weight="10">
    <location>../DB2/queries_pm/account_summary_sqlxml.xqr</location>
    <parameters>
        <parameter>
            <ids collection="custacc"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="6" weight="10">
    <location>../DB2/queries_pm/get_security_price_sqlxml.xqr</location>
    <parameters>
        <parameter>
            <file>../WorkloadDriver/input/security_symbols.txt</file>
        </parameter>
    </parameters>
</transaction>

<transaction id="7" weight="10">
    <location>../DB2/queries_pm/customer_max_order_sqlxml.xqr</location>
    <parameters>
        <parameter>
            <ids collection="custacc"/>
    </parameters>
</transaction>

```



```

        </parameter>
    </parameters>
</transaction>

<transaction id="8" weight="1">
    <location>../DB2/xqupdates_pm/U1CloseAccount.xqr</location>
    <parameters>
        <parameter>
            <ids collection="account"/>
        </parameter>
        <parameter>
            <same param="1"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="9" weight="1">
    <location>../DB2/xqupdates_pm/U2OpenAccount.xqr</location>
    <parameters>
        <parameter>
            <files collection="account"/>
        </parameter>
        <parameter>
            <ids collection="custacc"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="10" weight="3">
    <location>../DB2/xqupdates_pm/U3SecurityPrice.xqr</location>
    <parameters>
        <parameter>
            <uniformdec min="0.95" max="1.05"/>
        </parameter>
        <parameter>
            <uniformdec min="1.0001" max="1.01"/>
        </parameter>
        <parameter>
            <uniformdec min="0.99" max="0.9999"/>
        </parameter>
        <parameter>
            <file>../WorkloadDriver/input/security_symbols.txt</file>
        </parameter>
    </parameters>
</transaction>

```

```

<transaction id="11" weight="3">
  <location>../DB2/xqupdates_pm/U4OrderStatus.xqr</location>
  <parameters>
    <parameter>
      <file>../WorkloadDriver/input/order_sol_flags.txt</file>
    </parameter>
    <parameter>
      <file>../WorkloadDriver/input/order_src_values.txt</file>
    </parameter>
    <parameter>
      <ids collection="order"/>
    </parameter>
  </parameters>
</transaction>

<transaction id="12" weight="3">
  <location>../DB2/xqupdates_pm/U5BuySecurity.xqr</location>
  <parameters>
    <parameter>
      <ids collection="account"/>
    </parameter>
    <parameter>
      <file>../WorkloadDriver/input/security_symbols.txt
    </parameter>
    <parameter>
      <uniformint min="1" max="5000"/>
    </parameter>
    <parameter>
      <uniformint min="1000" max="100000"/>
    </parameter>
    <parameter>
      <uniformint min="1000" max="100000"/>
    </parameter>
    <parameter>
      <same param="1"/>
    </parameter>
  </parameters>
</transaction>

<transaction id="13" weight="3">
  <location>../DB2/xqupdates_pm/U6SellSecurity.xqr</location>
  <parameters>
    <parameter>

```

```

        </parameter>
        <parameter>
            <ids collection="account"/>
        </parameter>
        <parameter>
            <uniformint min="1" max="5000"/>
        </parameter>
        <parameter>
            <uniformint min="1000" max="100000"/>
        </parameter>
        <parameter>
            <uniformint min="1000" max="100000"/>
        </parameter>
        <parameter>
            <same param="1"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="14" weight="1">
    <location>../DB2/deletes_pm/delcustacc.sql</location>
    <parameters>
        <parameter>
            <ids collection="custacc"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="15" weight="7">
    <location>../DB2/deletes_pm/delorder.sql</location>
    <parameters>
        <parameter>
            <ids collection="order"/>
        </parameter>
    </parameters>
</transaction>

<transaction id="16" weight="5">
    <location>../DB2/inserts_pm/insValidcustacc.sql</location>
    <parameters>
        <parameter>
            <files collection="custacc"/>
        </parameter>
    </parameters>
</transaction>
<transaction id="17" weight="5">
    <location>../DB2/inserts_pm/insValidorder.sql</location>

```

```
<parameters>
  <parameter>
    <files collection="order"/>
  </parameter>
</parameters>
</transaction>
<transaction id="16" weight="5">
  <location>../DB2/inserts_pm/insNoValidcustacc.sql</location>
  <parameters>
    <parameter>
      <files collection="custacc"/>
    </parameter>
  </parameters>
</transaction>
<transaction id="17" weight="5">
  <location>../DB2/inserts_pm/insNoValidorder.sql</location>
  <parameters>
    <parameter>
      <files collection="order"/>
    </parameter>
  </parameters>
</transaction>
</transactions>
</workload>
```

Liite 3. Testitulokset

Ilman indeksejä

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks
1	get_order_sqlxml	Q	429	2	5.37	35.1610	0.0320	0.2120	0.0820	0
2	get_security_sqlxml	Q	399	4	5.00	11.9760	0.0250	0.1690	0.0300	0
3	customer_profile_sqlxml	Q	665	9	8.33	51.8220	0.0300	0.1910	0.0779	0
4	search_securities_sqlxml	Q	769	712	9.63	47.9390	0.0530	0.1480	0.0623	0
5	account_summary_sqlxml	Q	791	17	9.91	62.0130	0.0290	0.1490	0.0784	0
6	get_security_price_sqlxml	Q	774	14	9.70	23.0180	0.0250	0.0670	0.0297	0
7	customer_max_order_sqlxml	Q	778	778	9.75	131.4460	0.0610	0.3370	0.1690	0
8	U1CloseAccount	U	90	2	1.13	10.8440	0.0540	0.1970	0.1205	0
9	U2OpenAccount	U	80	1	1.00	6.6790	0.0370	0.1660	0.0835	0
10	U3SecurityPrice	U	257	2	3.22	8.1400	0.0270	0.0990	0.0317	0
11	U4OrderStatus	U	221	0	2.77	18.9140	0.0340	0.1480	0.0856	0
12	U5BuySecurity	U	238	1	2.98	29.0680	0.0500	0.2290	0.1221	0
13	U6SellSecurity	U	259	0	3.24	31.9650	0.0530	0.2140	0.1234	0
14	delcustacc	D	76	2	0.95	6.1200	0.0320	0.1400	0.0805	0
15	delorder	D	559	1	7.00	47.8810	0.0300	0.2750	0.0857	0
16	insValidcustacc	I	404	404	5.06	11.4120	0.0050	0.2640	0.0282	0
17	insValidorder	I	425	425	5.32	57.3731	0.0030	6.3530	0.1350	0
18	insNoValidcustacc	I	383	383	4.80	5.3280	0.0070	0.2860	0.0139	0
19	insNoValidorder	I	386	386	4.84	2.5190	0.0040	0.1640	0.0065	0

Indeksien kanssa

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks
1	get_order_sqlxml	Q	3211	25	5.07	3.1170	0.0000	0.0520	0.0010	0
2	get_security_sqlxml	Q	3031	38	4.79	2.8230	0.0000	0.0920	0.0009	0
3	customer_profile_sqlxml	Q	5091	302	8.04	9.0410	0.0000	0.0650	0.0018	0
4	search_securities_sqlxml	Q	6419	6179	10.14	19.4211	0.0000	0.0210	0.0030	0
5	account_summary_sqlxml	Q	6363	442	10.05	12.3912	0.0000	0.0660	0.0019	0
6	get_security_price_sqlxml	Q	6363	72	10.05	5.8808	0.0000	0.0630	0.0009	0
7	customer_max_order_sqlxml	Q	6267	6267	9.90	10.8921	0.0000	0.0630	0.0017	0
8	U1CloseAccount	U	637	7	1.01	0.6030	0.0000	0.0320	0.0009	0
9	U2OpenAccount	U	623	33	0.98	3.7220	0.0010	0.3260	0.0060	0
10	U3SecurityPrice	U	1914	19	3.02	2.7000	0.0000	0.0890	0.0014	0
11	U4OrderStatus	U	1903	16	3.00	3.0130	0.0000	0.2900	0.0016	0
12	U5BuySecurity	U	1843	14	2.91	2.4240	0.0000	0.0630	0.0013	0
13	U6SellSecurity	U	1889	10	2.98	2.1740	0.0000	0.0580	0.0012	0
14	delcustacc	D	620	34	0.98	1.1180	0.0000	0.0480	0.0018	0
15	delorder	D	4433	35	7.00	4.1099	0.0000	0.0640	0.0009	0
16	insValidcustacc	I	3162	3162	4.99	60.7788	0.0050	0.5130	0.0192	0
17	insValidorder	I	3085	3085	4.87	373.6547	0.0030	3.7900	0.1211	0
18	insNoValidcustacc	I	3175	3175	5.01	55.9538	0.0070	0.7380	0.0176	0
19	insNoValidorder	I	3299	3299	5.21	24.7669	0.0040	0.3140	0.0075	0

Liite 4. Indeksien luominen

```
-----  
-- (C) Copyright IBM Corp. 2006  
-- This program is made available under the terms of the Common Public  
-- License 1.0 as published by the Open Source Initiative (OSI).  
-- http://www.opensource.org/licenses/cpl1.0.php  
-----  
-- Basic set of indexes for DB2 to support the regular TPoX workload:  
-----  
  
-- create xml indexes for custacc  
create unique index custacc_id on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/@id' as sql double %  
create unique index custacc_accountid on custacc(CADOC) generate key using xmlpattern 'declare namespace c="http://tpox-benchmark.com/custacc";/c:Customer/c:Accounts/c:Account/@id' as sql varchar(15) %  
  
-- create xml indexes for order  
create index order_accountid on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/@Acct' as sql varchar(15)%  
create unique index order_id on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/@ID' as sql varchar(15) %  
  
-- create xml indexes for security  
create unique index secsymbol on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/s:Symbol' as sql varchar(10) %  
create index sec_sector on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/SecurityInformation//Sector' as sql varchar(25) %  
create index sec_PE on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/PE' as sql double %  
create index sec_Yield on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/Yield' as sql double %  
  
-- Alternative (extended) set of indexes for DB2,  
-- to support a wider range of possible queries:  
-----  
-- create xml indexes for custacc  
create unique index custacc_id on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/@id' as sql double %
```

```

create index custacc_lastname on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Name/LastName' as sql varchar(20) %
create index custacc_firstname on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Name/FirstName' as sql varchar(20) %
create index custacc_opening on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Accounts/Account/OpeningDate' as sql date %
create index custacc_currency on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Accounts/Account/Currency' as sql varchar(5) %
create index custacc_state on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Addresses/Address/State' as sql varchar(30) %
create index custacc_type on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Accounts//Position/Type' as sql varchar(15) %
create index custacc_country on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/CountryOfResidence' as sql varchar(30) %
create unique index custacc_accountid on custacc(CADOC) generate key using xmlpattern 'declare namespace c="http://tpox-benchmark.com/custacc";/c:Customer/c:Accounts/c:Account/@id' as sql varchar(15) %
create index custacc_postalcode on custacc(CADOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/custacc";/Customer/Addresses/Address/PostalCode' as sql double %

-- create xml indexes for order
create index order_secsymbol on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/Instrmt/@Sym' as sql varchar(10) %
create index order_accountid on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/@Acct' as sql varchar(15)%
create unique index order_id on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/@ID' as sql varchar(15) %
create index order_snt on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/Hdr/@Snt' as sql date %
create index order_cash on order(ODOC) generate key using xmlpattern 'declare default element namespace "http://www.fixprotocol.org/FIXML-4-4";/FIXML/Order/OrdQty/@Cash' as sql double %

-- create xml indexes for security
create unique index secsymbol on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/s:Symbol' as sql varchar(10) %
create index sec_name on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/Name' as sql varchar(70) %
create index sec_type on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/s:SecurityType' as sql varchar(15) %
create index sec_id on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/@id' as sql double %
create index sec_industry on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/s:SecurityInformation/*/s:Industry' as sql varchar(40) %
create index sec_sector on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/SecurityInformation//Sector' as sql varchar(25) %
create index sec_PE on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/PE' as sql double %
create index sec_Yield on security(SDOC) generate key using xmlpattern 'declare default element namespace "http://tpox-benchmark.com/security";/Security/Yield' as sql double %
create index sec_Price on security(SDOC) generate key using xmlpattern 'declare namespace s="http://tpox-benchmark.com/security";/s:Security/s:Price/s:Price50DayAvg' as sql double %

```


Liite 5. Asennushistoria verrattuna Quick Start Guideen

DBTechNet ML 2013-10-24, updated 2013-12-01

Notes on TPoX 2.1 test on SUSE Linux / DB2 Express-C 10.1 downloaded from DB2university.com
applying the Quick Start Guide August 2011 and using XXS scaled database

DB2 Express-C preinstalled by user db2inst1/password in the directory
/opt/ibm/db2/V10.1

2. Obtaining TPoX

#-----

TPoX 2.1 downloaded from <http://tpox.sourceforge.net>

3. Installing TPoX

#-----

starting from the home directory

cd

note: without the -a option!

unzip Downloads/TPoX.v2.1.zip

...

inflating: TPoX/DB2/queries/README.TXT

inflating: TPoX/DB2/queries/run.ksh

inflating: TPoX/DB2/queries/search_securities.xqr

inflating: TPoX/DB2/queries/search_securities_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/account_summary_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/customer_max_order_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/customer_profile_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/get_order_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/get_security_price_sqlxml.xqr

inflating: TPoX/DB2/queries_pm/get_security_sqlxml.xqr
inflating: TPoX/DB2/queries_pm/search_securities_sqlxml.xqr
inflating: TPoX/DB2/README.TXT
inflating: TPoX/DB2/xqupdates/README.TXT
inflating: TPoX/DB2/xqupdates/U1CloseAccount.xqr
inflating: TPoX/DB2/xqupdates/U1CloseAccount_v2.xqr
inflating: TPoX/DB2/xqupdates/U2OpenAccount.xqr
inflating: TPoX/DB2/xqupdates/U3SecurityPrice.xqr
inflating: TPoX/DB2/xqupdates/U4OrderStatus.xqr
inflating: TPoX/DB2/xqupdates/U5BuySecurity.xqr
inflating: TPoX/DB2/xqupdates/U5BuySecurity_v2.xqr
inflating: TPoX/DB2/xqupdates/U6SellSecurity.xqr
inflating: TPoX/DB2/xqupdates/U6SellSecurity_v2.xqr
inflating: TPoX/DB2/xqupdates_pm/U1CloseAccount.xqr
inflating: TPoX/DB2/xqupdates_pm/U2OpenAccount.xqr
inflating: TPoX/DB2/xqupdates_pm/U3SecurityPrice.xqr
inflating: TPoX/DB2/xqupdates_pm/U4OrderStatus.xqr
inflating: TPoX/DB2/xqupdates_pm/U5BuySecurity.xqr
inflating: TPoX/DB2/xqupdates_pm/U6SellSecurity.xqr
inflating: TPoX/documentation/TPoX_BenchmarkProposal_v2.1.doc
inflating: TPoX/documentation/TPoX_BenchmarkProposal_v2.1.pdf
inflating: TPoX/documentation/TPoX_DataGeneration_v2.1.doc
inflating: TPoX/documentation/TPoX_DataGeneration_v2.1.pdf
inflating: TPoX/documentation/TPoX_Quickstart_v2.1.doc
inflating: TPoX/documentation/TPoX_Quickstart_v2.1.pdf
inflating: TPoX/documentation/WorkloadDriverUsage_v2.1.doc
inflating: TPoX/documentation/WorkloadDriverUsage_v2.1.pdf
inflating: TPoX/fix
inflating: TPoX/fixAll
creating: TPoX/generatedXML/

```
inflating: TPoX/MSSQL/classes/net/sf/tpox/databaseoperations/DatabaseOperations.class
..
inflating: TPoX/MSSQL/README.txt
inflating: TPoX/Oracle/classes/net/sf/tpox/databaseoperations/DatabaseOperations.class
..
inflating: TPoX/Oracle/README.txt
inflating: TPoX/WorkloadDriver/properties/README.TXT
inflating: TPoX/WorkloadDriver/properties/workload_properties.xsd
inflating: TPoX/WorkloadDriver/README.TXT
creating: TPoX/WorkloadDriver/xml/
```

```
chmod a+x fix
chmod a+x fixAll
./fixAll
```

```
Now converting setenv.ksh
Now converting DB2/ddl/backupdb.ksh
Now converting DB2/ddl/reorg.ksh
Now converting DB2/ddl/restartdb2.ksh
Now converting DB2/ddl/restore.ksh
Now converting DB2/ddl/runstats.ksh
```

```
# 4. Obtaining XML data files
```

```
#-----
gunzip account.v2.1.tar.gz
gunzip custacc.v2.1.tar.gz
gunzip order.v2.1.tar.gz
gunzip security.v2.1.tar.gz
```

```
cd TPoX
cd generatedXML/
tar xvf ../security.v2.1.tar
tar xvf ../account.v2.1.tar
tar xvf ../custacc.v2.1.tar
tar xvf ../order.v2.1.tar
order/
order/batch-1.xml
order/batch-1.xml.offset
order/batch-2.xml
order/batch-2.xml.offset
db2inst1@db2v10:~/TPoX/generatedXML>
```

```
db2inst1@db2v10:~/TPoX/generatedXML> java -version
java version "1.7.0"
Java(TM) SE Runtime Environment (build pxi3270-20110827_01)
IBM J9 VM (build 2.6, JRE 1.7.0 Linux x86-32 20110810_88604 (JIT enabled, AOT enabled)
J9VM - R26_Java726_GA_20110810_1208_B88592
JIT - r11_20110810_20466
GC - R26_Java726_GA_20110810_1208_B88592
J9CL - 20110810_88604)
JCL - 20110809_01 based on Oracle 7b147
```

```
db2inst1@db2v10:~/TPoX/generatedXML> javac -version
```

If 'javac' is not a typo you can run the following command to lookup the package that contains the binary:

```
command-not-found javac
bash: javac: command not found
```

```
#as root:
db2v10:/usr/bin # ln -s /opt/IBM/DS3.1.1/jdk/bin/javac javac
#
```

```
db2inst1@db2v10:~/TPoX/generatedXML> javac -version
javac 1.6.0-internal
```

```
# 5. Creating the tpox database
```

```
#-----
```

```
db2
```

```
create db tpox using codeset utf-8 territory us pagesize 16 K
```

```
connect to tpox
```

```
create table custacc (cadoc xml)
```

```
create table security (sdoc xml)
```

```
create table order (odoc xml)
```

```
connect reset
```

```
# 6. Workload property files
```

```
#-----
```

```
# The xml files in TPoX/WorkloadDriver/properties
```

```
# from the TPoX.zip have already been scaled for the XXS database.
```

```
#-----
```

```
# Beyond the Quick Start Guide:
```

```
# Downloading the FIXML schemas
```

```
cd
```

```
cd TPoX/schemas/order
```

```
# download by FireFox
```

```
http://www.fixprotocol.org/documents/352/fixml-schema-4-4-20040109rev1.zip
```

```
# and unzip the files as follows:
```

```
db2inst1@db2v10:~/TPoX/schemas/order> ls
```

```
fixml-schema-4-4-20040109rev1.zip README.TXT
```

```
unzip fixml-schema-4-4-20040109rev1.zip
```

```
Archive: fixml-schema-4-4-20040109rev1.zip
```

```
  creating: Documentation/
```

```
  inflating: Schema/fixml-allocation-base-4-4.xsd
```

```
  inflating: Schema/fixml-allocation-impl-4-4.xsd
```

```
...
```

```
  inflating: ReleaseNotesRev1_20061006.txt
```

```
db2inst1@db2v10:~/TPoX/schemas/order> ls
```

```
Documentation          README.TXT
```

```
fixml-schema-4-4-20040109rev1.zip  ReleaseNotesRev1_20061006.txt
```

```
fixml-schema-4-4-guide-20040109.pdf  Schema
```

```
cd
```

```
cd TPoX/DB2/ddl
```

```
db2 -td% -vf registerschemas.sql
```

```
# 7. Running insert tests
```

```
#-----
```

```
# Before running any tests, be sure to adjust and export your CLASSPATH to include the following:
```

```
export CLASSPATH=/opt/ibm/db2/V10.1/java/db2jcc4.jar:/home/db2inst1/TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPoX/WorkloadDriver/classes:/home/db2inst1/TPoX/DB2/classes
```

```
cd TPoX/WorkloadDriver
```

```
# Insert 250 security documents using 5 users and 50 transactions per user:
```

```
java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -w properties/insNoValidsecurity.xml -u 5 -tr 50
```

db2inst1@db2v10:~/TPoX/WorkloadDriver> java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -w properties/insNoValidsecurity.xml -u 5 -tr 50
The workload description file properties/insNoValidsecurity.xml is valid against the following XML Schema file:
properties/workload_properties.xsd

The WorkloadDriver program is running...

The following arguments are used (user id/password omitted):

-d tpox -u 5 -ht localhost -pt 50001 -w properties/insNoValidsecurity.xml -tr 50

The following workload is going to be executed: insNoValidSecurity

Longest connection time: 1 seconds

Workload execution starting date/time: Thu Oct 24 16:21:28 EEST 2013

Workload is now running...

Workload execution finishing date/time: Thu Oct 24 16:21:29 EEST 2013

Workload execution elapsed time: 1 seconds

STATISTICS OVER THE COMPLETE RUN:

*** SYSTEM WORKLOAD STATISTICS ***

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks
1	insertSecurity	I	250	250	100.00	5.1480	0.0050	0.2300	0.0206	0

*** SYSTEM THROUGHPUT ***

The throughput is 15000 transactions per minute (250.00 per second).

The output/output2013_10_24_1621 directory contains the files output.txt and stats.txt (as well as stats_per_user.txt, if the verbosity level is 1 or 2, and user1.txt, etc., if the verbosity level is 2). Additionally, it contains comment.txt if -c option was used.

The last security document inserted was No. 250 from file ../generatedXML/security/batch-1.xml.offset
db2inst1@db2v10:~/TPoX/WorkloadDriver>

Insert 250 custacc documents using 5 users and 50 transactions per user:

```
java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpoX -id db2inst1 -pw password -pt 50001 -w properties/insNoValidcustacc.xml -u 5 -tr 50
```

db2inst1@db2v10:~/TPoX/WorkloadDriver> # Insert 250 custacc documents using 5 users and 50 transactions per user:

```
db2inst1@db2v10:~/TPoX/WorkloadDriver> java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpoX -id db2inst1 -pw password -pt 50001 -w properties/insNoValidcustacc.xml -u 5 -tr 50
```

The workload description file properties/insNoValidcustacc.xml is valid against the following XML Schema file:

properties/workload_properties.xsd

The WorkloadDriver program is running...

The following arguments are used (user id/password omitted):

```
-d tpoX -u 5 -ht localhost -pt 50001 -w properties/insNoValidcustacc.xml -tr 50
```

The following workload is going to be executed: insNoValidCustacc

Longest connection time: 1 seconds

Workload execution starting date/time: Wed Nov 13 10:59:00 EET 2013

Workload is now running...

Workload execution finishing date/time: Wed Nov 13 10:59:02 EET 2013

Workload execution elapsed time: 2 seconds

STATISTICS OVER THE COMPLETE RUN:

*** SYSTEM WORKLOAD STATISTICS ***

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks
1	insertCustAcc	I	250	250	100.00	10.7300	0.0070	0.3200	0.0429	0

*** SYSTEM THROUGHPUT ***

The throughput is 7500 transactions per minute (125.00 per second).

The output/output2013_11_13_1058 directory contains the files output.txt and stats.txt (as well as stats_per_user.txt, if the verbosity level is 1 or 2, and user1.txt, etc., if the verbosity level is 2). Additionally, it contains comment.txt if -c option was used.

The last custacc document inserted was No. 250 from file ../generatedXML/custacc/batch-1.xml.offset
db2inst1@db2v10:~/TPoX/WorkloadDriver>

Insert 250 order documents using 5 users and 50 transactions per user:

```
java net.sf.tpoX.workload.core.WorkloadDriver -ht localhost -d tpoX -id db2inst1 -pw password -pt 50001 -w properties/insNoValidorder.xml -u 5 -tr 50
```

```
db2inst1@db2v10:~/TPoX/WorkloadDriver> java net.sf.tpoX.workload.core.WorkloadDriver -ht localhost -d tpoX -id db2inst1 -pw password -pt 50001 -w properties/insNoValidorder.xml -u 5 -tr 50
```

The workload description file properties/insNoValidorder.xml is valid against the following XML Schema file:

properties/workload_properties.xsd

The WorkloadDriver program is running..

The following arguments are used (user id/password omitted):

```
-d tpox -u 5 -ht localhost -pt 50001 -w properties/insNoValidorder.xml -tr 50
```

The following workload is going to be executed: insNoValidOrder

Longest connection time: 0 seconds

Workload execution starting date/time: Wed Nov 13 11:01:09 EET 2013

Workload is now running..

Workload execution finishing date/time: Wed Nov 13 11:01:10 EET 2013

Workload execution elapsed time: 1 seconds

STATISTICS OVER THE COMPLETE RUN:

*** SYSTEM WORKLOAD STATISTICS ***

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks
1	insertOrder	I	250	250	100.00	5.1070	0.0030	0.2410	0.0204	0

*** SYSTEM THROUGHPUT ***

The throughput is 15000 transactions per minute (250.00 per second).

The output/output2013_11_13_1101 directory contains the files output.txt and stats.txt (as well as stats_per_user.txt, if the verbosity level is 1 or 2, and user1.txt, etc., if the verbosity level is 2).

Additionally, it contains comment.txt if -c option was used.

The last order document inserted was No. 250 from file ../generatedXML/order/batch-1.xml.offset
db2inst1@db2v10:~/TPoX/WorkloadDriver>

Backup the database for extra test runs

db2stop

db2start

export DB_BACKUP=/home/db2inst1/sqllib/backup

time db2 -v backup database tpox to \$DB_BACKUP

date

#--- running the backup: -----

db2inst1@db2v10:~> db2stop

SQL11064N DB2STOP processing was successful.

db2inst1@db2v10:~> db2start

SQL11063N DB2START processing was successful.

db2inst1@db2v10:~> time db2 -v backup database tpox to \$DB_BACKUP

backup database tpox to /home/db2inst1/sqllib/backup

Backup successful. The timestamp for this backup image is : 20131113111547

real 0m7.359s

user 0m0.012s

sys 0m0.008s

db2inst1@db2v10:~> ls /home/db2inst1/sqllib/backup

TPOX.0.db2inst1.DBPART000.20131113111547.001

db2inst1@db2v10:~> ls /home/db2inst1/sqllib/backup -l

total 164168

```
-rw----- 1 db2inst1 db2grp1 167936000 Nov 13 11:15 TPOX.0.db2inst1.DBPART000.20131113111547.001
```

```
db2inst1@db2v10:~>
```

```
# Test restoring the database
```

```
db2stop
```

```
db2start
```

```
export DBBACKUP=/home/db2inst1/sqllib/backup
```

```
export TIMESTAMP=20131113111547
```

```
date
```

```
time db2 -v RESTORE DATABASE tpox FROM $DBBACKUP TAKEN AT $TIMESTAMP WITHOUT ROLLING FORWARD WITHOUT PROMPTING
```

```
date
```

```
db2inst1@db2v10:~>
```

```
db2inst1@db2v10:~> db2stop
```

```
SQL1025N The database manager was not stopped because databases are still active.
```

```
db2inst1@db2v10:~> db2stop
```

```
SQL1064N DB2STOP processing was successful.
```

```
db2inst1@db2v10:~> db2start
```

```
SQL1063N DB2START processing was successful.
```

```
db2inst1@db2v10:~> time db2 -v backup database tpox to $DB_BACKUP
```

```
backup database tpox to /home/db2inst1/sqllib/backup
```

```
Backup successful. The timestamp for this backup image is : 20131115222555
```

```
real          0m57.134s
```

```
user          0m0.016s
```

```

sys                                0m0.024s
db2inst1@db2v10:~>
db2inst1@db2v10:~> ls /home/db2inst1/sqllib/backup -l
total 384752
-rw----- 1 db2inst1 db2grp1 167936000 Nov 13 11:15 TPOX.0.db2inst1.DBPART000.20131113111547.001
-rw----- 1 db2inst1 db2grp1 225652736 Nov 15 22:26 TPOX.0.db2inst1.DBPART000.20131115222555.001
db2inst1@db2v10:~>

# 8. Creating property files for the query tests
#-----

# copy the sampleQueries.xml from the Quick Start Guide page 6
# to TPoX/WorkloadDriver/properties directory

# 9. Running query tests
#-----

cd
cd TPoX/WorkloadDriver

# Before running the WorkloadDriver always adjust and export the CLASSPATH as follows:
export CLASSPATH=/opt/ibm/db2/V10.1/java/db2jcc4.jar:/home/db2inst1/TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPoX/WorkloadDriver/classes:/home/db2inst1/TPoX/DB2/classes

# A query test using 4 users for 5 minutes (300 seconds):
java net.sf.tpoX.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -w properties/sampleQueries.xml -u 4 -ti 300

#---- Setup and TPoX 2.1 workload tests using the XXS scaled tpox database ----
#
# 10. Loading the scale XXS database

```

```
#-----  
# based on correspondence with Matthias Nicola and Tim Kiefer  
  
# First emptying the database  
  
db2  
connect to tpox  
truncate order immediate  
truncate custacc immediate  
truncate security immediate  
terminate  
  
# Create EXPLAIN PLAN tables:  
cd  
db2 connect to tpox  
db2 -tf sqllib/misc/EXPLAIN.DDL  
  
cd  
cd TPoX/WorkloadDriver  
  
# Before running the WorkloadDriver always adjust and export the CLASSPATH as follows:  
export CLASSPATH=/opt/ibm/db2/V10.1/java/db2jcc4.jar:/home/db2inst1/TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPoX/WorkloadDriver/classes:/home/db2inst1/TPoX/DB2/classes  
  
# Insert 20833 security documents using 83 users and 251 transactions per user:  
java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -w properties/insNoValidsecurity.xml -u 83 -tr 251  
  
# Insert 50000 custacc documents using 50 users and 1000 transactions per user:  
java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -w properties/insNoValidcustacc.xml -u 50 -tr 1000  
  
# Insert 500000 order documents using 50 users and 10000 transactions per user:
```

```
java net.sf.tpx.workload.core.WorkloadDriver -ht localhost -d tpx -id db2inst1 -pw password -pt 50001 -w properties/insNoValidorder.xml -u 50 -tr 10000
```

```
# A. Making offline backup of the database without the XML indexes
```

```
db2 terminate
```

```
db2stop
```

```
db2start
```

```
export DB_BACKUP=/home/db2inst1/sqllib/backup
```

```
time db2 -v backup database tpx to $DB_BACKUP
```

```
# - Make note of the backup timestamp!
```

```
# B. Then create the XML indexes
```

```
cd
```

```
cd TPoX/DB2/ddl
```

```
db2 -td% -f createindexes.sql
```

```
# Making offline backup of the database with the XML indexes
```

```
db2 terminate
```

```
db2stop
```

```
db2start
```

```
export DB_BACKUP=/home/db2inst1/sqllib/backup
```

```
time db2 -v backup database tpx to $DB_BACKUP
```

```
# - Make note of the backup timestamp!
```

```
# -----
```

```
# Template for WorkloadDriver runs
```

```
#
```

```
# Restore the initial XXS database backup, either the backup containing the XML indexes
```

```
# or the backup without the XML indexes
```

```
db2stop
db2start
export DBBACKUP=/home/db2inst1/sqllib/backup
export TIMESTAMP=<replace here the timestamp of the backup>
date
time db2 -v RESTORE DATABASE tpox FROM $DBBACKUP TAKEN AT $TIMESTAMP WITHOUT ROLLING FORWARD WITHOUT PROMPTING
date
```

```
SELECT COUNT(*) AS #Orders FROM order %
-- should be 500 000
SELECT COUNT(*) AS #CustAccs FROM custacc %
-- should be 50 000
SELECT COUNT(*) AS #Securities FROM security %
-- should be 20 833
VALUES 'Customer ids in CustAcc' %
SELECT MIN(id) AS custacc_min_id,MAX(id) AS custacc_max_id FROM (SELECT XMLCAST(XMLQUERY('$CADOX/*:Customer/@id') AS INTEGER) AS id FROM custacc) %
-- should be 1002, 51001
VALUES 'Customer ids in Orders' %
SELECT MIN(id) AS order_min_id,MAX(id) AS order_max_id FROM (SELECT XMLCAST(XMLQUERY('$ODOX/*:FIXML/*:Order/@ID') AS INT) AS id FROM order) %
-- should be 103282, 603281
```

```
# Verifying the proper XXS contents
```

```
cd
db2 connect to tpox
db2 -td% -f XXS_contents.qry
# - see the XXS_contents.qry for proper values!
```

```
# This could be a proper step to investigate some execution plan(s)
```

```
cd
rm Q1.plan.basic
```



```

db2 connect to tpox
db2 -td% -f Q1_get_order_sqlxml.xqr
db2exfmt -d tpox -1 -o Q1.plan.basic
cat Q1.plan | more
#-----
# Workload test run:
cd
cd TPoX/WorkloadDriver
# Before running the WorkloadDriver always adjust and export the CLASSPATH as follows:
export CLASSPATH=/opt/ibm/db2/V10.1/java/db2jcc4.jar:/home/db2inst1/TPoX/WorkloadDriver/plugins/commons-cli-1.0.jar:/home/db2inst1/TPoX/WorkloadDriver/classes:/home/db2inst1/TPoX/DB2/classes
#
java net.sf.tpox.workload.core.WorkloadDriver -ht localhost -d tpox -id db2inst1 -pw password -pt 50001 -u 50 -w properties/mixed.xml -ti 7200 -r 1800 -pd 240 -pc -cl
date
#
#-----
# As example on studying the execution plans the script for get_order_sqlxml.xqr
# has been copied to the home directory of db2inst1
# and EXPLAIN PLAN FOR clause has been inserted in front of the command

# generating the plan into EXPLAIN tables:
cd
db2 connect to tpox
db2 -td% -f Q1_get_order_sqlxml.xqr

# formatting the plan into Q1.plan file
db2exfmt -d tpox -1 -o Q1.plan

# looking at the plan
cat Q1.plan | more
#----- end of the document -----

```

Liite 6. Testi Nicolan työryhmän määrittämällä kuormalla ja XXS-skaalatulla tietokannalla

Testikuorma löytyy TPoX 2.1-paketista. Testissä on käytetty mixed.xml-tiedostoa sellaisenaan.

Statistiikka ilman indeksejä

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks	90% Percentile	95.0% Conf. Inter- val
1	get_order_sqlxml	Q	20	20	8.37	1833.9891	79.4550	128.6350	91.6995	0	105.2910	[66.8459 - 116.5530]
2	get_security_sqlxml	Q	25	25	10.46	91.6220	1.5560	6.7090	3.6649	0	6.2940	[0.0000 - 7.8880]
3	customer_profile_sqlxml	Q	30	30	12.55	368.3870	4.0730	23.5030	12.2796	0	21.3210	[0.0000 - 26.5795]
4	search_securities_sqlxml	Q	16	548	6.69	80.3550	3.1860	9.6570	5.0222	0	8.5370	[0.2841 - 9.7603]
5	account_summary_sqlxml	Q	33	33	13.81	407.9950	4.0850	26.2990	12.3635	0	21.2430	[0.0000 - 26.9197]
6	get_security_price_sqlxml	Q	26	26	10.88	95.7290	1.5470	9.5880	3.6819	0	6.0700	[0.0000 - 8.3408]
7	customer_max_order_sqlxml	Q	17	17	7.11	1938.0361	95.7050	161.5120	114.0021	0	125.1130	[84.2728 - 143.7315]
8	U1CloseAccount	U	2	2	0.84	28.2370	8.7610	19.4760	14.1185	0	19.4760	[3.6178 - 24.6192]
9	U2OpenAccount	U	3	3	1.26	38.8440	4.1830	18.7950	12.9480	0	18.7950	[0.5763 - 25.3197]
10	U3SecurityPrice	U	10	10	4.18	33.6710	1.6080	7.1500	3.3671	0	5.9260	[0.0000 - 7.6510]
11	U4OrderStatus	U	6	6	2.51	573.7610	82.0710	106.9030	95.6268	0	100.3770	[80.2309 - 111.0227]
12	U5BuySecurity	U	5	5	2.09	82.9670	8.3820	24.9060	16.5934	0	24.9060	[2.7561 - 30.4307]
13	U6SellSecurity	U	8	8	3.35	137.0330	7.9380	32.2360	17.1291	0	25.1810	[0.5326 - 33.7257]
14	delcustacc	D	2	2	0.84	27.7980	13.3720	14.4260	13.8990	0	14.4260	[12.8661 - 14.9319]
15	delorder	D	16	16	6.69	1429.3842	79.3720	103.3680	89.3365	0	100.1520	[75.6997 - 102.9733]
16	insValidcustacc	I	4	4	1.67	0.3510	0.0060	0.2730	0.0878	0	0.2730	[0.0000 - 0.3025]
17	insNoValidorder	I	16	16	6.69	0.8330	0.0050	0.5090	0.0521	0	0.0480	[0.0000 - 0.2900]

*** SYSTEM THROUGHPUT ***

The throughput is 1 transactions per minute (0.03 per second).

Statistiikka indeksien kanssa

Tr.#	Name	Type	Count	Affected Rows	%-age	Total Time (s)	Min Time (s)	Max Time (s)	Avg Time (s)	#Deadlocks	90% Percentile	95.0% Conf. Inter- val
1	get_order_sqlxml	Q	21145	20870	10.04	348.6857	0.0000	2.8860	0.0165	0	0.0290	[0.0000 - 0.1073]
2	get_security_sqlxml	Q	20948	20948	9.95	194.9443	0.0000	2.0710	0.0093	0	0.0210	[0.0000 - 0.0637]
3	customer_profile_sqlxml	Q	21389	20962	10.16	335.8724	0.0000	1.4090	0.0157	0	0.0260	[0.0000 - 0.0895]
4	search_securities_sqlxml	Q	21190	890521	10.06	1022.9453	0.0060	1.5930	0.0483	0	0.0800	[0.0000 - 0.1099]
5	account_summary_sqlxml	Q	20981	20555	9.96	355.0197	0.0000	2.5660	0.0169	0	0.0290	[0.0000 - 0.0867]
6	get_security_price_sqlxml	Q	21288	21288	10.11	196.6355	0.0000	1.1800	0.0092	0	0.0200	[0.0000 - 0.0668]
7	customer_max_order_sqlxml	Q	20816	20816	9.88	1975.9993	0.0000	3.4210	0.0949	0	0.1810	[0.0000 - 0.2887]
8	U1CloseAccount	U	2096	2027	1.00	86.6169	0.0000	1.1590	0.0413	0	0.0600	[0.0000 - 0.1844]
9	U2OpenAccount	U	2108	2076	1.00	161.0699	0.0010	1.3800	0.0764	0	0.1180	[0.0000 - 0.2369]
10	U3SecurityPrice	U	6060	6060	2.88	144.3663	0.0040	1.2640	0.0238	0	0.0330	[0.0000 - 0.1472]
11	U4OrderStatus	U	6354	6259	3.02	745.5122	0.0000	2.8070	0.1173	0	0.2670	[0.0000 - 0.4098]
12	U5BuySecurity	U	6269	6021	2.98	332.4692	0.0000	1.6540	0.0530	0	0.0760	[0.0000 - 0.2120]
13	U6SellSecurity	U	6319	6080	3.00	347.3459	0.0000	2.2070	0.0550	0	0.0790	[0.0000 - 0.2334]
14	delcustacc	D	2076	2038	0.99	75.6057	0.0000	2.2240	0.0364	0	0.0550	[0.0000 - 0.2059]
15	delorder	D	14719	14494	6.99	408.7568	0.0000	2.2110	0.0278	0	0.0470	[0.0000 - 0.1269]
16	insValidcustacc	I	2101	2101	1.00	146.8753	0.0040	1.6060	0.0699	0	0.1050	[0.0000 - 0.2697]
17	insNoValidorder	I	14738	14738	7.00	313.3703	0.0040	1.6550	0.0213	0	0.0350	[0.0000 - 0.1357]

The throughput is 1755 transactions per minute (29.25 per second).