

TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Infotehnoloogia eriala

Henri Trees

**Enamlevinud andmebaasisüsteemide veebiteenuste
võrdlus**

Bakalaureusetöö (6 EAP)

Juhendaja: Vambola Leping

Autor: “.....“ mai 2012

Juhendaja: “.....“ mai 2012

Lubada kaitsmisele

Professor: “.....“ mai 2012

TARTU 2012



Sisukord

Sissejuhatus	5
1. Ülevaade veebiteenustest	7
1.1 Veebiteenused (Web services)	7
1.2 SOAP.....	7
1.3 WSDL	9
2. Võrdlus ja testrakendus	10
2.1 Võrdluses osalevad andmebaasisüsteemid.....	10
2.2 Testrakendus	10
2.3 Võrdlemise meetodika	11
3. Sybase SQL AnyWhere 11	13
3.1 Ülevaade.....	13
3.2 Andmetüübid ja vorming	14
3.3 Turvalisus	14
3.4 Testrakenduse loomine.....	15
3.5 Võrdlus	16
4. Oracle Database 11g	19
4.1 Ülevaade.....	19
4.2 WSDL-dokument ja andmetüübid	20
4.3 Turvalisus	20
4.4 Testrakenduse loomine.....	21
4.5 Võrdlus	22
5. Microsoft SQL Server 2008	24
5.1 Ülevaade.....	24
5.2 WSDL genereerimine.....	25
5.3 Turvalisus	25
5.4 Testrakenduse loomine.....	27
5.5 Võrdlus	28
6. Võrdluse tulemused	30
Kokkuvõte	31
Comparison of Native Web Services of the Most Common Relational Database Management Systems	32
Viited	33
Lisad	36

Sissejuhatus

Veebiteenused on XML-keelel (*Extensible Markup Language* – laiendatav märgistuskeel) põhinevad tehnoloogiad, tänu millele on võimalik erinevates programmeerimiskeeltes implementeeritud tarkvaral hajusalt paigutatud arvutisüsteemides suhelda. Selliseid teenuseid kasutatakse erinevate ressursside kättesaadavaks tegemisel ning tihti peale on teenustega seotud andmed, mida hoitakse andmebaasis. Tavapäraselt implementeeritakse veebiteenuseid rakendusserveril, mis on andmebaasiga ühenduses [1]. Käesolev töö käsitleb veebiteenuste loomist, kasutades selleks andmebaasisüsteemidesse sisseehitatud võimalusi, mis võimaldavad andmebaasis olevat infot veebis kättesaadavaks teha.

Käesoleva bakalaureusetöö eesmärgiks on pakkuda ülevaade kolme andmebaasisüsteemi Sybase SQL Anywhere 11, Oracle Database 11g ja Microsoft SQL Server 2008 poolt pakutavatest, SOAP-protokollil (*Simple Object Access Protocol* – lihtne objektipöördus-protokoll) põhinevate, veebiteenuste loomise võimalustest. Iga vastava süsteemi puhul tuuakse välja üldine kirjeldus veebiteenuste loomise kohta ning võrreldakse konkreetset süsteemi teistega, vastavalt väljatoodud võrdluse metoodikale. Võrdluse toetamiseks realiseeritakse iga andmebaasisüsteemiga sama funktsionaalsusega testrakendus kolme erinevat tüüpi veebiteenusega. Teenuste korrektses töös veendutakse andmebaasile SOAP-päringuid saates. Viimaks tuuakse välja kokkuvõtte võrdluses antud hinnangute kohta ning selgitatakse parim andmebaasisüsteem vastavate veebiteenuste loomiseks. Kuna käsitletav meetod veebiteenuste loomiseks on alternatiiv tavapärasele viisile ning ei ole laialdaselt levinud, kasutatakse võrdluses eelkõige autori poolt omistatud hinnanguid.

Töö koosneb viiest peatükist. Esimeses peatükis on ülevaatlilikult kirjeldatud veebiteenuste olemusest ning seotud mõistetest. Teises peatükis antakse ülevaade süsteemide testimiseks loodavatest veebiteenustest ning võrdlemiseks kasutatavast metoodikast. Kolmas, neljas ja viies peatükk keskenduvad kolmele valitud andmebaasisüsteemile, kirjeldades nende võimalusi veebiteenuseid luues, andes ülevaate testrakenduse loomisest ning hinnates neid vastavalt kriteeriumitele. Kuuendas peatükis analüüsitakse võrdlustulemusi.

Lisana on esitatud andmebaasisüsteemidega loodud veebiteenuste vastavad WSDL-dokumendid (*Web Service Description Language*) ning teenuse käivitamiseks saadetud SOAP-päringud ja süsteemi poolt vastuseks saadud SOAP-sõnumid.

1. Ülevaade veebiteenustest

1.1 Veebiteenused (Web services)

Veebiteenused on Rahvusvahelise Interneti ja veebiga tegelevate firmade konsortsiumi (W3C – *World Wide Web Consortium*) poolt defineeritud kui tarkvara süsteem, mille eesmärk on toetada masinatevahelist suhtlust üle arvutivõrgu. Veebiteenusele juurdepääsuks on formuleeritud masinloetavas vormingus liides, ehk WSDL-dokument. Suhtlus toimub veebiteenustega SOAP-sõnumeid kasutades, mis tavaliselt edastatakse üle HTTP (*HyperText Transfer Protocol* - hüpertexti edastusprotokoll) [2].

Veebiteenuste põhieesmärk on pakkuda standardiseeritud viise onlain rakenduste kirjeldamiseks, leidmiseks ning rakendusevahelisteks suhtluseks. Lisaks eelnevalt nimetatud suhtlusele (SOAP) ja kirjeldamisele (WSDL) kasutatakse teenuste leidmiseks UDDI (*Universal Description, Discovery, and Integration*) katalooge, mis sisaldavad erinevate veebiteenuste kirjeldusi. Tänu veebiteenuste raamistiku modulaarsusele on võimalik kasutatavaid rakendusi täiendada samm-sammult [3].

Agent on veebiteenuste mõistes osa tark- või riistvarast, mille ülesandeks on sõnumeid saata ja vastu võtta. Teenus, mida agent võrgus kättesaadavaks teeb, on teenuse poolt pakutava funktsionaalsuse põhjal iseloomustatav ressurss. Sellise piiritluse tõttu võime eeldada, et erinevates programmeerimiskeeles implementeeritud agendid suudavad pakkuda sama teenust, muutest veebiteenused platvormist ja programmeerimiskeelest sõltumatuks. Inimest või organisatsiooni, kes või mis teeb läbi veebiteenuse kättesaadavaks kindla funktsionaalsuse, kasutades selleks sobilikku agenti, nimetatakse pakkujaks (*provider*). Pakutavat funktsionaalsust ära kasutavat inimest või organisatsiooni defineeritakse kui teenuse taotleja (*requester*), kes kasutab taotleja agenti, et pakkuja agendiga sõnumeid vahetada [2].

1.2 SOAP

Veebiteenuste sõlmed kasutavad omavaheliseks suhtluseks, mis koosneb struktureeritud ja tüübitud informatsiooni saatmisest, XML-keelel baseeruvat SOAP-protokolli. Laialt kasutatav ning platvormist sõltumatu XML-keel on vägagi sobiv veebiteenuste suhtluse

alustalaks. SOAP-sõnumite transpordiks kasutatakse olemasolevaid HTTP ja SMTP (*Simple Mail Transfer Protocol* – lihtne meiliedastusprotokoll) rakenduskihi protokolle. Tänu eelnevalt loetletud omadustele on SOAP platvormist ja programmeerimiskeelest sõltumatu [3].

SOAP-sõnumite, mida nimetatakse SOAP-ümbrieks (`env:Envelope`), struktuur koosneb päise (`env:Header`) ja keha (`env:Body`) XML-elementidest. Nende elementide sisu ei ole SOAP-protokolli poolt spetsifitseeritud, vaid veebiteenust implementeerivate rakenduste poolt määratletud. Keha ja päise sees kasutatakse konkreetseid nimeruumikirjeldusi. Päise element, mis koosneb päise blokkidest, ei ole kohustuslik, kuid võimaldab lisaks sõnumi sisule edasi anda täiendavat informatsiooni kogu sõnumi või rakenduse kohta. Näiteks teavet, kuidas töödelda sõnumi sisu osa. Keha elemendis asub informatsioon, mida veebiteenuse sõlmed edastavad ning see on kohustuslik osa SOAP-ümbrikust.

SOAP sõnumivahetus on oma olemuselt ühesuunaline ja olekuta, kuid võimaldab kasutatavate suhtlusprotokollide ja rakenduste abil implementeerida hulga keerulisemaid suhtlusmudeleid. Tavapärast toimub suhtlus kahe SOAP-sõlme vahel, kelleks on saatja ja vastuvõtja. Suhtluseks võib olla pelgalt saatja poolt edastatud sõnum ning sellele vastuseks saadetud kinnitus, kuid edasiarendusena on võimalik luua ka kahe sõlme vahel toimuvat pidevat sõnumivahetust. Sõnumid võivad lisaks läbida SOAP-vahendajaid, mille eesmärkideks võivad näiteks olla turvalisuse tagamine, sõnumite auditeerimine või logi pidamine [4].

Kasutades SOAP-protokolli lihtsaks sõnumiedastuseks on veebiteenuse kliendil vaja teada teenuse lõpp-punkti aadressi, kuhu sõnum saata. Olles samuti teadlik meetodist, mille poole pöörduda, ning viisist, kuidas sõnumi sisu kodeerida, saab klientrakendus veebiteenusele SOAP-sõnumeid saata.

SOAP-protokolli abil kaugprotseduurikutsete tegemiseks on veebiteenuse kliendil vaja teada SOAP-sõlme aadressi ja väljakutsutava protseduuri või meetodi nime. Täiendavalt peab olema informatsioon konkreetse operatsiooni sisend- ja väljundargumentide kohta, sõnumivahetuse meetodi ning võimalusel ka andmete kohta, mida päises edastatakse. Luues selliseid teenuseid on tähtis määratleda viis, kuidas rakendus-spetsiifised andmetüübid SOAP-sõnumite kaudu edasi antakse ning meetodid, mille poole pöörduda koos vastavate parameetritega. SOAP-sõnumites sisalduva XML-dokumendi struktuuri

ning XML-struktuuride andmetüüpide defineerimiseks kasutatakse enamjaolt W3C poolt defineeritud XML-skeemi (*XML Schema*). Andmetüüpide edastamiseks eeldab SOAP, et kasutatakse XML-skeemil põhinevat andmetüüpide süsteemi ning määratleb vastavatel reeglitel põhineva kodeeringu XML-dokumendis. Antud kodeerimismeetodit kasutades on võimalik luua igat tüüpi struktureeritud andmetele vastavat XML-kodeeringut. Tehes kaugprotseduurikutset SOAP-protokolli abil, paigutatakse kodeeringu info XML-elementi, mis sisaldab meetodile vajalikke argumente. Elementidele, mis sisaldavad argumente, lisatakse andmetüüp atribuudiga `type` [3].

1.3 WSDL

Veebiteenuste kirjeldamiseks kasutatakse WSDL-dokumente. WSDL (*Web Service Description Language*) on XML-dokumendi formaat, välja töötatud IBM ja Microsoft'i poolt, mis teeb võimalikuks veebiteenuste liideste kirjeldamise ning tagab teenuse klientidele kindla pidepunkti informatsiooni hankimiseks. Teenuse kirjeldamiseks kasutatav täieliku funktsionaalsusega WSDL-dokument koosneb kahest osast. Esiteks kirjeldatakse teenust, mida veebiteenus pakub (teisisõnu abstraktne kirjeldus), teisalt aga protokollispetsiifilisi andmeid, mida teades saavad kliendirakendused veebiteenust kasutada.

Teenuse abstraktsel kirjeldamisel määratletakse, millist meetodit kasutatakse andmetüüpide kirjeldamiseks, millised sõnumid võivad suhtlusel esineda ning suhtlusmeetodi. Andmetüüpide kirjeldamisel kasutatakse enamasti väliseid andmetüüpide süsteeme, milleks on tavapäraselt samuti XML-skeemi definitsioon. WSDL-dokumendi element `message` defineerib sõnumid, eelpool mainitud andmetüüpide süsteemi kasutades, mis suhtluses esinevad, ehk mida teenus vastu võtab ja välja saadab. Element `operation` seob sõnumid kokku suhtlusmustriks, mida vastav lõpp-punkt toetab. Selleks on tavaliselt sisend-, väljund- ja veasõnumid. `PortType` on kogumik operatsioonidest [3].

2. Võrdlus ja testrakendus

2.1 Võrdluses osalevad andmebaasisüsteemid

Käesolevas töös on läbi viidud SOAP-tüüpi veebiteenuste loomise võrdlus kolme enamlevinud andmebaasisüsteemiga, milleks on: Sybase SQL Anywhere 11 , Oracle Database 11g ning Microsoft SQL Server 2008. Võrreldud teenused on loodud otse andmebaasi-süsteemist ning ei kasutata eraldi rakendusserveri abi.

Kuigi SQL Server 2008 R2 versioonist alates on veebiteenuste (SOAP lõpp-punktide) loomine mittesoovitav ning edasiarendus on katkestatud, kasutatakse võrdluses antud süsteemi, kuna nimetatud andmebaasisüsteemi versioonid 2005, 2008 ja 2008 R2 on laialdaselt kasutusel. Lisaks on alternatiivina soovitatava WCF-raamistiku (Windows Communication Framework) [5] abil veebiteenuste loomiseks vaja eraldi rakendusserverit ning sellist meetodit antud töös ei käsitleta.

2.2 Testrakendus

Iga võrreldava andmebaasisüsteemiga luuakse sama funktsionaalsusega rakendus. Kõigile paigaldatakse täpselt sama andmebaas, mida kasutades veebiteenuseid luuakse. Andmebaas on loodud Tartu Ülikoolis, aine Andmebaasid raames, ning sisaldab andmeid maleturniiride toimumise, tulemuste ja osavõtjate kohta. Antud töös kasutatakse isikute ja klubide tabeleid.

Loodud veebiteenused on oma olemuselt lihtsakoelised. Teenuste eesmärk on eelkõige näidata, mis funktsionaalsust on võimalik antud andmebaasisüsteemidega kättesaadavaks teha. Implementeeritud veebiteenuseid on kolme erinevat tüüpi, milleks on (illustreeriva näitena on toodud SQL Anywhere andmebaasisüsteemis sisestatud käsud protseduuride loomiseks):

1. teenus, mis lubab andmebaasile saata SQL-käsu läbi SOAP-päringu. Käsk ise kujutab endast lihtsat SQL-keele päringut, mis väljastab *isik* tabelist kõik inimesed, kes kuuluvad klubisse , mille ID on 54;

```
SELECT id, eesnimi, perenimi FROM isik WHERE klubi = 54;
```

2. teenus, millele vastavaid SOAP-päringuid saates kutsutakse välja protseduur `sp_listKlubid`, millel ei ole sisendargumente, ning mis tagastab kõikide *klubi* tabelis olevate klubide nimed ning aadressid;

```
ALTER PROCEDURE "sp_listKlubid" ()
RESULT (
  Nimi      CHAR(30),
  aadress   CHAR(20))
BEGIN
  SELECT Nimi, aadress FROM klubi;
END;
```

3. teenus, mis kutsub välja klubi kustutamise protseduuri `sp_kustuta_klubi`, ning saab sisendparameetriks klubinime, mida kustutada.

```
ALTER PROCEDURE "sp_kustuta_klubi" (IN klubinimi
  VARCHAR(100))
BEGIN
  DELETE FROM klubi WHERE nimi = klubinimi;
END
```

Loodud veebiteenustele SOAP-sõnumite saatmiseks ja vastuvõtmiseks kasutati vabavaralist tarkvara soap UI, mille koduleht asub aadressil <http://www.soapui.org/>.

Esimese teenuse väljakutsumiseks paigutatakse SQL-käsk vastavasse märgendisse ning hoitakse CDATA XML-sektsiooni sees, et vastavate süsteemide parserid käsku ei parsiks. Kolmanda teenuse väljakutsumise järel kontrolliti klubi kustutamist andmebaasi tabeli ülevaatamisega.

2.3 Võrdlemise metoodika

Andmebaasisüsteemide poolt võimaldatavate SOAP-veebiteenuste loomise võrdlemiseks hinnatakse iga võrreldavat süsteemi järgnevat alampunktides:

- Dokumentatsioon – hinnatakse dokumentatsiooni kättesaadavust, struktureeritust ning üldist ülesehitust. Koodinäidete olemasolule ning võimalustele, kuidas vajaduse korral abi on võimalik saada, pööratakse samuti tähelepanu
- Õppimise kurv – antud töö autori hinnang sellele, kui arusaadav on kogu süsteemi ülesehitus ning talitus, ehk kui palju kulub aega, et aru saada, kuidas veebiteenuseid luua ning seejärel produktiivselt funktsionaalsust kasutada.

- Käskude arusaadavus – autori poolt antud hinnang kasutatud käskude arusaadavusele, mida veebiteenuste loomiseks on vaja käivitada. Lisaks võetakse arvesse käskude sisukust, ehk millist funktsionaalsust on võimalik käskudega seadistada.
- Tagatud funktsionaalsus – hinnang võimalikule funktsionaalsusele, mida läbi veebiteenuste saab teenuse klientidele pakkuda. Hinnatakse samuti kontrolli nii WSDL-dokumentide kui ka SOAP-sõnumite struktuuri ning kohandamise üle.
- Veebiteenuste turvalisus – kuna andmebaasisüsteemi poolt pakutavad veebiteenused on otseses ühenduses andmebaasiga, hinnatakse vastavaid meetmeid, mida on võimalik turvalisuse tagamiseks rakendada.

Iga punkti on hinnatud skaalal [0, 1, ..., 5], kus 0 – väga halb ja 5 – väga hea, ning kõigi võrreldavate andmebaasisüsteemide puhul on arvatud kogutulemus (maksimum 25 punkti, miinimum 0 punkti).

3. Sybase SQL AnyWhere 11

3.1 Ülevaade

Veebiteenuste loomine on toetatud otseselt Sybase SQL Anywhere 11 andmebaasisüsteemi poolt, ilma lisamoodulite abita. Süsteemi sisseehitatud HTTP-server suudab vastu võtta klientide poolt üle HTTP-protokolli saadetud SOAP-päringuid ning tagastada SOAP-formaadis vastuseid. Samuti on võimalik andmebaasisüsteemi kasutada SOAP-veebiteenuste klientrakendusena. Lisaks eelnevale funktsionaalsusele toimib SQL Anywhere ka HTTP-päringute vastuvõtjana ning tagastab klientidele vormindatud HTML-lehekülgi, ehk kasutajad saavad andmebaasiga suhelda ainuüksi veebibrauserit kasutades. Päringute tagastustüüpina kasutatakse lisaks eelnevale nii standardseid XML-dokumente, kui töötlemata (*raw*) andmeid. Viimase puhul on soovitatav päringule tagastatavad andmed töödelda, kasutades talletatud protseduure või kasutaja defineeritud funktsioone. Käesolevas töös keskendutakse ainult SOAP tüüpi veebiteenuste loomisele [6].

SOAP-teenused võimaldavad klientsõlmedele kättesaadavaks teha andmebaasisüsteemis salvestatud funktsioone (*Stored Functions*) ja protseduure (*Stored Procedures*). Veebiteenuste loomiseks on käsk `CREATE SERVICE`, mis antud juhul võimaldab täpsustada veebiteenuse ja vastussõnumi tüübi, nime ning turvalisust puudutavad valikud. Luues SOAP-teenuseid, on vajalik defineerida samuti `DISH` (*Determine SOAP Handler*) tüüpi veebiteenus, mis grupeerib eraldiseisvad SOAP-teenused ning loob neile vastavad WSDL-dokumendid.

`CREATE SERVICE` käsuga luuakse teenus, mis teeb kättesaadavaks kas SQL-käsu, funktsiooni või protseduuri. `DISH` tüüpi teenused eeldavad et SOAP-veebiteenused on nimetatud ühesuguse prefiksiga, mille lõppu lisatakse kaldkriips, näiteks „soap/teenusenimi“. Määratledes `DISH`-teenuse `GROUP` klauselis eeltoodud prefiksi („soap“), ühendatakse antud liitega SOAP-teenused `DISH` proksisse ning luuakse neid teenuseid grupeeriv WSDL-dokument. `DISH` veebiteenus, millele pole lisatud gruppi, teeb kättesaadavaks kõik SOAP tüüpi veebiteenused andmebaasis [7].

3.2 Andmetüübid ja vorming

Päringu vastussõnumi ja WSDL-dokumendi vormingu määrab `CREATE SERVICE` käsu `FORMAT` klausel. Valikuteks on `DNET`, `CONCRETE` või `XML`. `DNET` on vaikeväärtus ning kasutatakse `.NET` raamistikus implemeteeritud klientide puhul, kuna sellised rakendused kasutavad `.NET DataSet` vormingut. `CONCRETE` väärtuse puhul eeldatakse, et `JAX-WS` või `.NET` klientrakendused genereerivad automaatselt vajalikud liidesed vastavalt WSDL-dokumendile, kus elemendid on `SimpleDataset` vormingus, sisaldades ridu, mis omakorda koosnevad veergudest. `XML` vormingut kasutades saadetakse kogu `SOAP`-vastus ühe sõnena ning klientrakendused peavad ise vastuse sisus oleva `XML`-dokumendi sõeluma (*parse*), et vajalikud elemendid ja väärtused omandada [8].

`SOAP`-teenuseid on võimalik defineerida ilma vorminguta, mis võimaldab luua seadistuse, mis koosneb formaadita `SOAP`-teenustest, ning `DISH`-teenustest, millele on omistatud erinevad vormingud. Sellisel moel mitmesse `DISH` proksisse lisatud identseid `SOAP`-teenuseid saab erinevatele klientrakendustele kättesaadavaks teha erinevate vormingutega.

Vaikimisi ei toeta `SQL Anywhere` andmebaasis loodud `SOAP` veebiteenused andmetüüpide teisendusi. `DNET` vormingus `SOAP`-teenused kasutavad WSDL-dokumendis ja `SOAP`-sõnumite sisuosas vaikimisi `XML`-skeemi sõne andmetüüpi. `CONCRETE` ja `XML` vormingutes ei kasutata andmetüüpe. Sisend- ja väljundparameetrite andmetüüpide informatsiooni kontrollitakse `DATATYPE` klauselis. Parameetrite andmetüübid defineeritakse WSDL-dokumendis [9].

3.3 Turvalisus

`SQL Anywhere` andmebaasisüsteemi sisseehitatud `HTTP`-server toetab turvalist `HTTPS`-protokolli ning vaikimisi saab veebiteenusele päringuid saata nii üle tavalise `HTTP` kui ka `HTTPS`-protokolli. Käsu `CREATE SERVICE` atribuudi `SECURE` väärtuse määramisel on võimalik lubada ainult turvalised `HTTPS` ühendused. Eeldatakse, et andmebaasi serverit luues on nendele protokollide vastavad pordid juurdepääsuks seadistatud [7]. Andmebaasisserver vaikimisi veebiteenuste päringuid vastu ei võta [10].

Veebiteenuse turvalisuse tagamiseks on CREATE SERVICE käsul kaks atribuuti: AUTHORIZATION (autoriseerimine) ja USER (kasutaja). Autoriseerimine peab olema sisse lülitatud, kui veebiteenus ei käivita kindlat protseduuri, funktsiooni või SQL-käsku. Vastupidisel juhul peab olema määratletud kasutaja, kelle õigustes päringud käivitatakse. Olles autoriseerimise sisse lülitatud, saab kasutajana määrata kindla andmebaasisüsteemi kasutajakonto, -grupi, või väärtuse NULL. Viimasel juhul on võimalik kõikidel kasutajatel päringuid teostada [7]. Autentimiseks kasutatakse kooloniga eraldatud kasutajanime ja parooli lisades selle veebiteenuse URL-aadressi hosti ette [11].

Süsteemile saadetavaid SOAP-päringuid on võimalik detailsemalt läbi töödada salvestatud protseduuridega. NEXT_SOAP_HEADER ja SOAP_HEADER funktsioone kasutades saab otsida SOAP-päringute päisest näiteks turvalisusele suunatud informatsiooni [12].

3.4 Testrakenduse loomine

Olles loonud vastava andmebaasi ning testrakenduse tööks vajalikud salvestatud protseduurid toimub sisseehitatud serveri käivitamine Windows operatsioonisüsteemi käsureal käsuga dbeng12 -xs http(port=8082) [andmebaasi_asukoht], kus andmebaasi asukohaks määratakse andmebaasifaili asukoht arvutis. Käsuga määratakse ühtlasi kuulatavaks pordiks 8082. Interactive SQL'i käivitame käsuga dbisql -c "dbf=[andmebaasi_asukoht];uid=[kasutajanimi];pwd=[parool]" Järgnevalt avaneb liides, kuhu sisestatakse veebiteenuse loomise káske.

SQL Anywhere SOAP-veebiteenust ei ole võimalik määratleda ilma täidetava lauseta. Seetõttu puudub teenuste klientrakendustel võimalus saata andmebaasile SQL-päringuid SOAP-päringutega ning testrakenduse esimest veebiteenust ei saa implementeerida.

Salvestatud protseduuride sp_listklubid ja sp_kustuta_klubi kättesaadavaks tegemiseks on loodud kaks SQL Anywhere andmebaasisüsteemi kontekstis SOAP-teenust järgnevate käskudega:

```
CREATE SERVICE
"Test/ListKlubid"
TYPE 'SOAP'
AUTHORIZATION OFF
SECURE OFF
```

```
CREATE SERVICE "Test/KustutaKlubi"
TYPE 'SOAP'
AUTHORIZATION OFF
SECURE OFF
USER DBA
```

```
USER DBA                                DATATYPE OFF
DATATYPE OUT                            AS CALL sp_kustuta_klubi( :klubi );
AS CALL sp_listklubid();
```

Veebiteenuste nimede prefiksina kasutatakse sõna „Test“, et need grupeerida DISH tüüpi veebiteenusega, mis loob vastava WSDL-dokumendi, koosnedes kahest veebimeetodist ListKlubid ja KustutaKlubi. DISH-teenuse loomise käsk on järgnev:

```
CREATE SERVICE "Test_Teenus"
TYPE 'DISH'
GROUP "Test"
FORMAT 'DNET'
AUTHORIZATION ON
SECURE OFF
USER DBA;
```

Vorminguna on määratud DNET, mis kasutab .NET raamistiku DataSet vormingut. Autentimine on sisse lülitatud ning ainult kasutajal DBA on võimalik veebiteenust kasutada. Loodud veebiteenustele vastav WSDL-dokument asub aadressil `http://[host]:8082/[andmebaasi_nimi]/Test_Teenus`. Dokument on samuti toodud lisas 1.

SOAP-vastuste sisuna edastatakse andmetele lisaks ka käivitatud lause täitmise kohta käiv informatsioon. Näiteks ilma vigadeta lõpetatud lause täitmisel väljastatakse kood 0. Lisades 6 ja 8 on välja toodud vastavad päringud teenustele. Neile vastavad SOAP-sõnumid asuvad lisades 7 ja 9.

3.5 Võrdlus

Dokumentatsioon

Dokumentatsioon on küllaltki hästi struktureeritud. Esilehel on toodud hüperlingid alamosadele. Sissejuhatus annab väga hea ülevaate SQL Anywhere süsteemiga loodavatest veebiteenustest. Kohati on paigutatud eraldi veebiteenuste tüüpidele pühendatud peatükid segamini. Dokumentatsioonis esineb palju erinevaid koodinäiteid koos seletustega (kuni viis näidet SOAP-tüüpi veebiteenuse loomise ning klientrakenduse implementeerimise kohta). Kasutusjuhendit saab vaadelda *DocCommentXChange* liideses, kus on sisse logides võimalik kirjutada vastavate temade kohta kommentaare. Hinnang dokumentatsioonile – 4 punkti.

Õppimise kurv

Autori hinnangul on süsteemi ülesehitus arusaadav ning pärast esmase informatsiooni läbitöötamist on järgnevaid teenuseid lihtne implementeerida. Heaks ning samuti mõningal viisil ebageeldivaks omaduseks võib pidada DISH tüüpi teenuseid, mis võimaldavad SOAP-teenuseid grupeerida ning teha homogeenseid veebiteenuseid, kus erinevatele klientrakendustele pakutakse erinevate vormingutega WSDL-dokumente ja SOAP-vastuseid. See aga tähendab, et iga teenuse loomise jaoks on vaja määratleda kaks teenust. Hinnang õppimise kurvile – 4 punkti.

Käskude arusaadavus

Veebiteenuste loomiseks kasutatav käsk `CREATE SERVICE` on lihtsa struktuuriga ning ei sisalda üleliia palju valikuid. Antud andmebaasisüsteem on võrdluses ainuke, millel on võimalik teenuse loomise käsuga spetsifitseerida käivitav lause SQL-käsuna. Võrreldavate süsteemidega määratletakse protseduur või funktsioon, mida välja kutsuda, aga SQL Anywhere käivitab etteantud käsku. Selline funktsionaalsus on hea lihtsakoelise veebiteenuse loomiseks, mis näiteks väljastab andmeid. Hinnang käskude arusaadavusele – 5 punkti.

Tagatud funktsionaalsus

Teenuste klientidele on võimalik kättesaadavaks teha salvestatud protseduure ja kasutaja defineeritud funktsioone. Erinevalt kahest teisest võrdluses osalevast andmebaasisüsteemist ei ole võimalik teha sellist teenust, millele oleks võimalik SOAP-päringuna saata SQL-keele päringuid. WSDL-dokumentidele ja SOAP-sõnumitele on võimalik määrata erinevaid vorminguid, mis soodustavad Microsoft .NET ja Java programmeerimiskeeles implementeeritud klientprogrammide loomist, kuid kasutada saab samuti lihtsamat vormingut. SOAP-sõnumite andmetüüpe saab eraldi kontrollida sisenevate ja väljasaadetavate sõnumite puhul. Hinnang tagatud funktsionaalsusele – 4 punkti

Veebiteenuste turvalisus

Veebiteenuseid saab luua ainult kasutaja, kellele on määratud DBA roll, seetõttu ei saa tavalised kasutajad andmebaasi informatsiooni kättesaadavaks teha. Ainult vastava käsuga loodud teenused on klientidele avatud. Kuigi saab luua olukorra, kus teenindatakse ainult kliente, kes kasutavad turvalist HTTPS-protokolli, on ikkagi võimalik teenuseid defineerida ilma autentimiseta, mis võib turvalisusele ohtu kujutada. Hinnang turvalisusele – 3 punkti.

4. Oracle Database 11g

4.1 Ülevaade

Oracle Database 11g ning hilisemad versioonid pakuvad andmebaasisüsteemi SOAP-veebiteenuste loomise võimalust Oracle XML DB mooduli kaudu. Klientrakendustele saab kättesaadavaks teha SQL ja XQuery päringuid, PL/SQL funktsioone ja protseduure ning nende pakette. Kuigi teenuste loomine on lihtne ning neid on võimalik välja kutsuda klientide ja rakendusserveri poolt, on soovitatav parema skaleeritavuse, turvalisuse ja usaldusväärsuse tagamiseks implementeerida veebiteenused eraldiseisvas Oracle Fusion Middleware rakendusserveris [13]. SOAP-päringuid kuulab, võtab vastu ja vastab päringutele Oracle XML DB HTTP server. Päringute saatmiseks kasutatakse HTTP POST meetodit ning SOAP-protokolli toetatud versiooniks on 1.1.

Veebiteenuseid teenindab `orawsv` servlet, mille konfiguratsioon on määratletud Oracle XML DB konfiguratsioonifailis `xdbconfig.xml`. Servleti nimi tuleb lisada URL-aadressile hosti ja pordi järele, ning on kohustuslik osa kõikide andmebaasis asuvate veebiteenuste aadressites. SQL ja Xquery päringud saadetaksegi `http://host:port/orawsv` aadressile, kus port on HTTP või HTTPS-protokollile vastav pordinumber. Sarnaselt võrreldavate süsteemidega on võimalik vastava veebiteenuse WSDL-dokument kätte saada lisades URL-rajale argumendi `?wsdl`. Saadetav käsk ning käsu formaat peab asuma SOAP-päringu sisuosa `query_text` märgendis. Lisavalikud annavad võimaluse kaasata käsuga erinevaid muutujaid, vastuse vormingu eelistusi ja muud.

Oracle Database veebiteenuste, mis teevad kättesaadavaks PL/SQL protseduurid või funktsioonid, loomine erineb tunduvalt teistest võrdluses esinevatest andmebaasisüsteemidest, kuna veebiteenuse loomiseks ei ole vajalik käivitada eraldi käsku. Olles veebiteenused sisse lülitatud, saavad klientrakendused pöörduda kõigi andmebaasisüsteemis asuvate protseduuride ja funktsioonide poole. Seda tehakse saates SOAP-päringuid URL-aadressile `http://host:port/orawsv/skeem/pakett/fun_prots`, kus `skeem` tähistab andmebaasiskeemi, `fun_prots` funktsiooni või protseduuri nime ning `pakett` paketti, milles see asub. Juhul kui eelnimetatud operatsioonid ei asu paketi sees tuleb see URL-aadressi osa ära jätta. SOAP-päring kindlale veebiteenusele peab sisaldama vastavaid sisendväärtusi,

kui teenus eksponeerib funktsiooni või protseduuri, milles kasutatakse kohustuslikke sisendparameetreid. Veateadete edastamiseks kasutatakse SOAP-raamistiku veatöötlust [14].

4.2 WSDL-dokument ja andmetüübid

Iga PL/SQL protseduur ja funktsioon on seostatud erinevate dünaamiliste veebiteenustega, millele omane WSDL-dokument asub veebiteenuse aadressil, millele on lisatud parameeter *?wsdl*. Sarnaselt SQL Anywhere veebiteenuste loomisele, kus kasutati SOAP-teenuste grupeerimiseks DISH-teenuseid, on võimalik Oracle andmebaasisüsteemis sisuda erinevad eraldiseisvad SOAP teenused ühe proksiga (puhvriga). Selleks tuleb luua teenused ühise paketi alla ning seejärel pöörduda aadressile `http://host:port/orawsv/skeem/pakett?wsdl`, kus *pakett* on eelnimetatud pakett. Vastavas WSDL-dokumendis on kirjeldatud kõik veebiteenused, mis asuvad selles pakettis. Andmebaasisüsteemi poolt kasutatavad SQL andmetüübid teisendatakse SOAP-sõnumites alati XML-skeemi andmetüüpideks [14].

4.3 Turvalisus

Veebiteenuse kliendi autentimiseks kasutatakse HTTP-protokolli *basic* (tavaline) autentimismeetodit, mille puhul saadetakse kasutajanimi ning parool serverile HTTP-päises *base64* kodeeringus. Ühtegi täiendavat võimalust veebiteenuse kasutatavaid sõlmi turvalisemalt autentida pole süsteemi poolt toetatud.

Veebiteenuste funktsionaalsuse implementeerimiseks on õigused ainult andmebaasisüsteemi kasutajal SYS. Ülejäänud kasutajatele SOAP-teenuste kasutamise õiguste tagamiseks tuleb konkreetsele kasutajakontole omistada roll XDB_WEBSERVICES. Vaikimisi tagatakse teenustele juurdepääs ainult üle turvalise HTTPS-protokolli. Tavalise HTTP-protokolli kasutamiseks on kasutajal vaja ka XDB_WEBSERVICES_OVER_HTTP rolli. Kasutajal on ligipääs kõikidele objektidele, millele tal õigused on, välja arvatud avalikud (*public*) objektid. Rolli XDB_WEBSERVICES_WITH_PUBLIC omistades, garanteeritakse ka ligipääs avalikele objektidele [14].

4.4 Testrakenduse loomine

Sisesehitatud veebiteenused ei ole konfigureeritud ning selle jaoks tuleb kasutajana SYS luua vastav servlet järgneva käsuga:

```
DECLARE
  SERVLET_NAME VARCHAR2(32) := 'orawsv';
BEGIN
  DBMS_XDB.deleteServletMapping(SERVLET_NAME);
  DBMS_XDB.deleteServlet(SERVLET_NAME);
  DBMS_XDB.addServlet(NAME      => SERVLET_NAME,
                     LANGUAGE => 'C',
                     DISPNAME => 'Oracle Query Web Service',
                     DESCRIPT  => 'Servlet for issuing
queries as a Web Service',
                     SCHEMA   => 'XDB');
  DBMS_XDB.addServletSecRole(SERVNAME => SERVLET_NAME,
                             ROLENAME => 'XDB_WEBSERVICES',
                             ROLELINK => 'XDB_WEBSERVICES');
  DBMS_XDB.addServletMapping(PATTERN => '/orawsv/*',
                             NAME     => SERVLET_NAME);
END;
/
```

Kuna kasutatav andmebaas on loodud kasutaja *test* poolt, omistatakse kasutajale rollid XDB_WEBSERVICES, mis on kohustuslik veebiteenuste kasutada saamiseks, ning XDB_WEBSERVICES_OVER_HTTP, et garanteerida teenusele ligipääs HTTP-protokolli kasutades.

Käsuga EXEC dbms_xdb.setHttpPort(8081) seame serveri kuulama pordil 8081. Koheselt on võimalik pöörduda veebiteenuse poole, läbi mille on võimalik saata SQL-päringuid. Vastava teenuse WSDL-dokument asub loodud testrakenduse puhul aadressil `http://127.0.0.1:8081/orawsv?wsdl` ning lisas 2. Teenusele saadetud SOAP-päring ning vastu saadetud SOAP-vastus asuvad lisades 10 ja 11.

Teine veebiteenus, mis kutsub välja salvestatud protseduuri, on Oracle andmebaasisüsteemis implementeeritud teistega võrreldes erinevalt. Nimelt tagastab teenus ainult klubi „Laudnikud“ ning on vastavalt nimega `sp_listKlubi`, kuid ei muuda testitud teenuse olemust. Protseduurid `sp_kustuta_klubi` ning `sp_listKlubi` saavad koheselt pärast loomist teenuse lõpp-punkti vastavalt (tähelepanu suurtähtedele protseduuri ja skeemi nimes) `http://127.0.0.1:8081/orawsv/TEST/SP_KUSTUTA_KLUBI` (toodud lisana

4) ja http://127.0.0.1:8081/orawsv/TEST/SP_LISTKLUBI (toodud lisana 3) aadressil. Antud veebiteenuste agentidele saadetud päringud on kujutatud lisades 12 ja 14. SOAP-vastused asuvad lisades 13 ja 15.

4.5 Võrdlus

Dokumentatsioon

Dokumentatsioon on üldjoontes lihtne ja otsekohene. Sissejuhatuses mainitakse põgusalt võimalustest ning süsteemi struktuurist. Klientrakenduste loomise kohta pole näitekoodi välja toodud, nagu näiteks kahel teisel võrreldaval süsteemil. Samas on näited veebiteenuste käivitamise kohta ammendavad. Erinevalt SQL Anywhere'st on dokumentatsiooni lehel ka viited sellele, kuidas Oracle klienditoega ühendust saada. Veebiversioonis on samuti võimalik sarnaselt teiste võrreldevate süsteemidega kommentaare lisada. Kokkuvõttes jätab dokumentatsioon autorile keskmise mulje, paremaks muudaks detailsem süsteemi kirjeldus. Hinnang dokumentatsioonile – 3 punkti.

Õppimise kurv

Võrreldavatest süsteemidest veebiteenuste loomise vaatepunktist kõige lihtsam. Töötab põhimõttel, kus pärast veebiteenuste rakendamist tehakse klientidele kättesaadavaks kõik protseduurid ja funktsioonid, mis andmebaasis asuvad. Mõningat segadust tekitab protokolliserveri tööleseadmine ja konfigureerimine, mis on eraldatud veebiteenuste loomise juhendmaterjalist. Üheks parimaks küljeks on Oracle Database puhul järgnevate veebiteenuste implementeerimise kiirus. Kõik loodavad protseduurid ja funktsioonid saavad hetkega kättesaadavaks veebiteenuste klientidele. Võrreldavatel süsteemidel peab sisestama vähemalt ühe teenuse loomise või muutmise käsu. Hinnang õppimise kurvile – 4 punkti.

Käskude arusaadavus

Oracle XML DB abil veebiteenuste loomiseks otsest käsku ei eksisteeri. Veebiteenuste rakendamiseks on vajalik sisestada teenuste servleti loomiseks etteantud käsk kasutusjuhendist, millega ei ole põhimõtteliselt võimalik midagi seadistada. Serveri pordi määramise käsk on arusaadav. Hinnang käskude arusaadavusele – 5 punkti.

Tagatud funktsionaalsus

Läbi veebiteenuste on ligipääs kõikidele PL/SQL protseduuridele ja funktsioonidele. Samuti on võimalik saata SQL ja XQuery päringuid läbi esmase servleti. Viimane meetod on võrdluses üks paremini konfigureeritavaid, lõpp-punkt lubab SOAP-päringuga kaasata rohkelt valikuid vastussõnumi kohta. Funktsionaalsuse poolest on Oracle Database võrdluse parim, SQL Server'iga ei ole võimalik vastu võtta XQuery päringuid ja SQL Anywhere'iga SQL ega XQuery päringuid. Miinusena ei ole võimalik kontrollida WSDL-dokumentide ja SOAP-sõnumite vorminguid, samuti on andmetüüpide teisendamine alati sisse lülitatud. Mõneti piiravaks osutub ka vajadus kasutada teenuste URL-aadressis servleti nime *orawsv*. Hinnang tagatud funktsionaalsusele – 4 punkti.

Veebiteenuste turvalisus

Veebiteenuste servleti loomise õigust omab kasutaja SYS, samas kui vastav servlet on loodud, tehakse kõik protseduurid ja funktsioonid kasutajatele kättesaadavaks. Veebiteenustele juurdepääsuks on muidugi vaja kindlale kasutajale omistada vastavad rollid ning esialgu ongi kasutajal võimalus veebiteenust kasutada ainult üle turvalise HTTPS-protokolli, aga kuna süsteem kasutab ainukese sisseehitatud autentimismeetodina HTTP *basic* autentimist, on turvalisus nõrk. Hinnang veebiteenuste turvalisusele – 4 punkti.

5. Microsoft SQL Server 2008

5.1 Ülevaade

Microsoft SQL Server andmebaasisüsteemi võimaldab alates 2005. aasta versioonist luua veebiteenuseid kasutades sisseehitatud XML veebiteenuseid (*Native XML Web Services*) [15]. Veebiteenused muutuvad klientidele kättesaadavaks HTTP lõpp-punktide (*endpoints*) abil, mis tähistavad transpordikihi üht otspunkti. Kuna veebiteenused kasutavad suhtlemiseks HTTP transport-kihti, ongi selline lõpp-punkt vajalik. Läbi HTTP lõpp-punkti saavad veebiteenuse kliendid SOAP-päringuid andmebaasisüsteemile edastada. Klientidena on toetatud SOAP 1.1 ja 1.2 versioone implementeerivaid rakendusi [16].

HTTP ühenduse tagamiseks klientrakenduse ja SQL Server andmebaasisüsteemi vahel kasutatakse *http.sys* draiverit. Veebiteenuse loomisel või muutmisel spetsifitseeritud URL-aadressile lisatakse serveri nimi ning port ja reserveeritakse kokkuliidetud nimeruum *http.sys* draiveris. Seejärel eelpool mainitud lõpp-punktile HTTP-päringuid saates suunatakse need SQL Server instantsile [17].

Veebiteenuste abil on võimalik kättesaadavaks teha SQL Server andmebaasisüsteemi salvestatud protseduure (*Stored Procedures*) ja kasutaja defineeritud funktsioone (*User Defined Functions*). Pärast eelnimetatud protseduuride ja funktsioonide sidumist mõne konkreetse lõpp-punktiga nimetatakse neid veebimeetoditeks (*Web Methods*) [16]. SOAP-päringus on samuti võimalik andmebaasisüsteemile saata *ad hoc* TSQL, ehk Microsoft'i poolt edasiarendatud SQL-keele, päringuid. Antud juhul tuleb SOAP-päringus kutsuda välja *sqlbatch* veebimeetod. Päring ise tuleb paigutada `<BatchCommands>` märgendisse [18].

Lõpp-punkte luuakse käsuga `CREATE ENDPOINT`. See käsk koosneb kahest osast: `AS` ja `FOR` klausel. Käsu alguses omistatakse lõpp-punktile nimi, `AS` klauselis defineeritakse suhtlusprotokoll, mida kasutatakse, ning samuti täiendavad valikud, nagu näiteks port ja turvameetod. `FOR` klauselis täpsustatakse edastamise meetod, salvestatud protseduur või kasutaja defineeritud funktsioon, mida konkreetne lõpp-punkt pakkuma hakkab, WSDL-dokumendi genereerimise meetod, andmebaasi kontekst ja SOAP-vastuse spetsiifilised suvandid [19].

5.2 WSDL genereerimine

SQL Server andmebaasisüsteemi HTTP lõpp-punktid võimaldavad klientidele pakkuda kolme tüüpi WSDL-dokumente: lihtne, vaikimisi genereeritud ja kasutaja defineeritud. Olles spetsifitseerinud veebiteenuste loomise `CREATE ENDPOINT` käsu `FOR SOAP` klauselis `WSDL = default`, pakub SQL Server klientidele lihtsustatud ja laiendatud formaadis WSDL-dokumente. Need on konkreetsete lõpp-punktidega seostatud kaugprotseduurikutsete põhjal dünaamiliselt genereeritud XML-dokumendid. Juhul kui automaatselt genereeritud WSDL-dokumendid ei ole kindlatele klientidele sobivad, on võimalik kasutaja poolt defineerida spetsiaalne WSDL-dokument, kasutades salvestatud protseduuri.

Laiendatud WSDL-dokument tagastatakse, kui klient lisab veebiteenuse URL-aadressi lõppu `?wsdl` argumendi. Sarnaselt tuleb lihtsustatud WSDL-dokumendi jaoks kasutada liidet `?wsdlsimple` [20]. Laiendatud WSDL seab XML-skeemist tuletatud andmetüübid vastavusse keerukate SQL-andmetüüpidega, mis on kasutuses andmebaasisüsteemis [21]. Lihtsas WSDL-dokumendis on kõik andmetüübid kirjeldatud primitiivseid XML-skeemi andmetüüpe kasutades. Kuigi lihtsustatud versiooni kasutades ei ole võimalik implementeerida kõiki SQL Server'i andmetüüpe, tagatakse võimalus genereerida veebiteenuste klientrakendust luues puhver (*proxy*), läbi mille on võimalik hakata saatma SOAP-päringuid. Eelkõige on see mõeldud klientrakendustele, mis ei ole võimelised aru saama või kasutama laiendatud WSDL-dokumendis kirjeldatud andmetüüpe [22]. Laiendatud WSDL-dokumendi genereerimise võimalusega on peetud silmas pigem Microsoft Visual Studio abil C# või Visual Basic programmeerimiskeeltes veebiteenuse klientprogramme implementeerivaid kasutajaid, kuna keerukamad andmestruktuurid on täielikult toetatud ning nendega töötamine on tehtud võimalikult lihtsaks.

5.3 Turvalisus

Veebiteenuste turvalisus on tagatud mitmel tasemel. SQL Server'i veebiteenustele ei ole võimalik tagada anonüümset juurdepääsu, mis välistab tundmatute klientide poolt saadetud SOAP-päringute käivitamist andmebaasisüsteemis [16]. Lõpp-punktide autentimine on sügavamalt analüüsitud järgmises lõigus. Kasutajal on samuti vaja vastavaid privileege, et käivitada käsked `CREATE ENDPOINT`, `ALTER ENDPOINT` ja `DROP ENDPOINT`.

Mainitud käskudele saab käivitamise lube omistada käsuga GRANT <privileeg> ON ENDPOINT <lõpp-punkti nimi> TO <kasutaja> [23].

Lõpp-punkti autentimine spetsifitseeritakse AUTHENTICATION klauselis CREATE ENDPOINT või ALTER ENDPOINT käsus. Turvalisuse tagamiseks on võimalik kasutada ainult Windows'i kasutajaid või kasutajagruppe ning SQL Server'i kasutajaid. Kasutades SQL Server'i kasutajakontosid, peab klientrakendus spetsifitseerima *WS-Security* päised autentimise jaoks. *WS-Security (Web Services Security)* on OASIS-konsortsiumi (*Organization for the Advancement of Structured Information Standards*) poolt publifitseeritud avalik standard, mis spetsifitseerib mehhanismi, mille alusel on võimalik SOAP-sõnumitega seostada kindlat kasutajat, kes sõnumi saatis [24]. Samuti on antud päiste abil võimalik muuta SQL Server'i kasutaja salasõna. *WS-Security* päisega SOAP-päringu saatmisel SQL Server'ile ei arvestata HTTP-protokolli autentimismeetoditega. Autentitud kasutajatel on võimalik luua ühendus lõpp-punktiga ning sellele HTTP-päringuid saata. Autentimistüüpideks on võimalik kasutada BASIC (tavaline), DIGEST, NTLM, Kerberos või Integrated (integreeritud) autentimist.

Tavaline autentimine põhineb *base64* kodeeringus saadetud krüpteerimata kasutajanimel ja paroolil, mis on eraldatud semikooloniga, ning edastatakse SQL Serverile autentimispäises. Kuna seda kodeeringut on võimalik ilma vaevata dekodeerida, ei ole soovitatav tavalist autentimist kasutada, kuid vajadusel kasutada turvasoklite kihti sõnumite edastamiseks (SSL). DIGEST autentimisel räsitakse parool ja kasutajanimi MD5 ühesuunalise räsifunktsiooniga ning saadetakse serverile. Sellist meetodit kasutades on kliendil võimalik tõestada autentsus, ilma parooli serverile saatmata. Kasutajal peab olema valideeritud Windowsi domeeni kasutajakonto.

NTLM-autentimist kasutavad Windows 95/98/NT 4.0. Autentimise tüüp on turvalisem, kui eelnevad kaks, kuid võrguga ühenduses olevate süsteemidega soovitatakse kasutada Kerberos autentimist. Kerberos on üks levinumaid autentimise meetodeid ning seda toetavad Windows 2000 ja uuemad versioonid. Integreeritud autentimist kasutades toetab server NTLM ja Kerberos autentimist ning server üritab klienti autentida klienti soovitud meetodiga. Kui klienti ei suudeta ühe meetodiga autentida, siis teist ei rakendata, ning server sulgeb ühenduse [25].

5.4 Testrakenduse loomine

Andmebaasi ning protseduuride loomise järel on testrakenduse jaoks defineeritud veebiteenuste loomiseks vaja sisestada üksainus käsk, mis on järgnev:

```
CREATE ENDPOINT test_teenus
STATE = STARTED
AS HTTP (
    PATH = '/ope/test_ws',
    AUTHENTICATION = (INTEGRATED ),
    PORTS = ( CLEAR ),
    CLEAR_PORT = 8082
)
FOR SOAP (
    WEBMETHOD 'ope'. 'listKlubid'
        (name='ope.dbo.sp_listKlubid',
        SCHEMA=STANDARD ),
    WEBMETHOD 'ope'. 'kustutaKlubi'
        (name='ope.dbo.sp_kustutaKlubi',
        SCHEMA=STANDARD ),
    BATCHES = ENABLED,
    WSDL = DEFAULT,
    SCHEMA = STANDARD,
    DATABASE = 'ope',
    NAMESPACE = 'http://localhost:80/ope/test_ws'
);
```

AS-klauselis on määratud integreeritud autentimismeetod ning lõpp-punkti asukoht ja kasutatav port. FOR-klauselis on kaks veebimeetodi, ehk loodud protseduurid, ning BATCHES = ENABLED konfigureerib kolmanda veebimeetodi, millele on võimalik saata esimeses testteenuses olevat päringut. Antud SOAP-sõlme WSDL-dokument asub URL-aadressil `http://localhost:8082/ope/test_ws?wsdl`, mis on samuti kättesaadav lisas 5. Testitud teenustele saadetud päringud asuvad lisades 16, 18 ja 20 ning neile vastavad SOAP-vastused lisades 17, 19 ja 21.

5.5 Võrdlus

Dokumentatsioon

Kindlasti võrreldavate süsteemide parim dokumentatsioon. On olemas kogu funktsionaalsuse põhjalik kirjeldus, sageli selgitavate tabelite ja joonistega. Selgelt on välja toodud tegevuste järjestus, millele toetudes veebiteenuseid luua. Palju suunavaid teemasid, nagu näiteks juhised, tavad ja piirangud teenuseid implementeerides. Iga teema juures on välja toodud seostuvad- ja ülemteemad. Koodinäiteid on piisavalt, välja on toodud mitme erineva veebiteenuse loomise näited koos klientrakenduste loomisega. Lehtedel on võimalik samuti kommenteerida või abi küsida teemaga seoses. Hinnang dokumentatsioonile – 5 punkti.

Õppimise kurv

Kuna dokumentatsioon on väga palju aspekte hõlmav ning suures osas selgitav, on süsteemi ülesehitusest lihtne aru saada. Veebiteenuste loomiseks piisab ainult teenuste lõpp-punktide määratlemisest. Lõpp-punktide konfigureerimine on samas teistest süsteemidest keerulisem ning nõuab pikemat süvenemist, et omandada täpset arusaamist, kuidas loodav teenuse lõpp-sõlm tööle hakkab. Hinnang õppimise kurvile – 4 punkti.

Käskude arusaadavus

Veebiteenuste loomiseks, muutmiseks ja kustutamiseks olevad käsud on võrreldes teiste süsteemidega pigem keerulisemad hoomata, kuid pakuvad rohkem võimalusi teenustega manipuleerimiseks. Hinnang käskude arusaadavusele – 4 punkti.

Tagatud funktsionaalsus

Lisaks tavapärasele talletatud protseduuridele ja kasutaja defineeritud funktsioonidele, on võimalik läbi veebiteenuse süsteemile saata SQL-päringuid. WSDL-dokumente pakutakse iga lõpp-punkti kohta kolmes erinevas vormingus: tavapärane, lihtne või kasutaja defineeritud. Lõpp-punktide funktsionaalsus on võrreldavatest süsteemidest tunduvalt parem. Lisaks ühistele funktsioonidele, pakub SQL Server täiendavaid valikuid näiteks SOAP-sõnumite skeemi, vormingu, märgistiku ja nimeruumi kohta. Toetatud on SOAP-sessioonid ja vastuste kompressimine. Hinnang tagatud funktsionaalsusele – 5 punkti.

Veebiteenuste turvalisus

Sarnaselt funktsionaalsusele on SQL Server andmebaasisüsteemi turvalisus võrdluse parim. Toetatud on viite tüüpi autentimist, samas on võimalik kasutada ka nende kombinatsioone. Kasutajatena aktsepteeritakse lisaks süsteemi kasutajatele ka Windows domeeni kasutajakontosid. Anonüümne juurdepääs ei ole võimalik. Hinnang veebiteenuste turvalisusele – 5 punkti.

6. Võrdluse tulemused

Andmebaasisüsteemide veebiteenuste võrdluse koondtulemused on välja toodud järgnevas tabelis 1.

	Sybase SQL Anywhere	Oracle Database 11g	MicroSoft SQL Server 2008
Dokumentatsioon	4	3	5
Üldine ülesehitus	4	4	4
Käskude arusaadavus	5	5	4
Tagatud funktsionaalsus	4	4	5
Veebiteenuste turvalisus	3	4	5
Tulemus kokku (maksimum 25)	20	20	23

Tabel 1. Võrdluse koondtabel.

Parima tulemuse saavutas Microsoft SQL Server 2008 andmebaasisüsteem, suuresti tänu lisafunktsionaalsusele, mida teised süsteemid ei suutnud pakkuda. Lisafunktsionaalsuse alla võib liigitada rohkete autentimismeetodite olemasolu ning lõpp-punktide paremat funktsionaalsust võrreldes teistega. Põhjalikule dokumentatsioonile toetudes võib järeldada, et SQL Server andmebaasisüsteemi puhul on veebiteenuste loomise funktsionaalsuse realiseerimiseks kulunud teistest tunduvalt rohkem jõupingutusi: kasutusjuhendis ei keskenduta ainult vajalike käskude lahtiseletamisele, vaid antakse kasutajale edasi konseptsioone, tavasid ning viise, kuidas veebiteenuseid luua ning kasutada.

Oracle Database ja SQL Anywhere tulemused on võrdsed, kuid lahendused erinevad teineteisest ning mõlemad omavad tugevaid ja nõrku külgi. SQL Anywhere dokumentatsioon on rohkem seletav, kui Oracle oma, kuid funktsionaalsus ja turvalisus on pigem kesised. Oracle Database paistab silma oma lihtsusega ning tänu sellele on võimalik pärast esmast konfigureerimist järgnevaid veebiteenuseid rakendada ilma lisavaevata. Pärast protseduuri või funktsiooni lisamist tekib kohe vastav teenuse sõlm, kuhu on võimalik pöörduda. Samas pole võimalik Oracle andmebaasiga loodud veebiteenuseid seadistada.

Kokkuvõte

Töös kirjeldati veebiteenuste olemust, milliseid tehnoloogiaid kasutatakse ning kuidas neid rakendatakse. Samuti anti ülevaade kolme populaarse andmebaasisüsteemi (Sybase SQL Anywhere 11, Oracle Database 11g, MicroSoft SQL Server 2008) abil veebiteenuste loomise võimalustest, funktsionaalsusest ning turvalisusest.

Teises peatükis kirjeldatud meetodikat kasutades võrreldi kõiki kolme süsteemi, mida töös käsitleti. Veebiteenuste realiseerimisele ning autori arvamusele põhinedes anti hinnang erinevatele aspektidele. Võrdluse tulemusena leiti, et parim võrdluses osalenud andmebaasisüsteem veebiteenuste loomiseks on Microsoft SQL Server tänu suurepärasele dokumentatsioonile, silmapaistvale funktsionaalsusele ja turvalisusele. Teised kaks süsteemi said võrdse hinnangu, kuid SQL Anywhere puudusteks on võrreldavatega halb turvalisus ning Oracle Database 11g ei võimalda veebiteenuste üle piisavat kontrolli.

Testrakendust luues implementeeriti erinevat tüüpi veebiteenused kõigil kolmel andmebaasisüsteemil ning toodi töös välja käsud, millega teenuseid loodi. Töö lisana on toodud kõikide teenuste agentide WSDL-dokumendid ning funktsionaalsuse testimiseks lõpp-punktidele saadetud SOAP-päringud ja vastusena saadud SOAP-vastused.

Autori hinnangul on antud töös käsitletud veebiteenuste loomise meetod väga kasutajasõbralik ning ei nõua palju valdkonnaspetsiifilisi teadmisi. Looja peab omama põhilisi teadmisi andmebaaside loomisest ning käskudest, millega luuakse protseduure ja funktsioone, mida tahetakse läbi veebiteenuse kättesaadavaks teha. Kindlasti on tähtis keskenduda turvalisuse tagamisele, sest teenuste lõpp-punktid võivad endaga kaasa tuua turvaauke.

Comparison of Native Web Services of the Most Common Relational Database Management Systems

Bachelor thesis

Henri Trees

Abstract

Web services are a set of XML based technologies, which provide communication, discovery and description of implemented services for distributed resources. As opposed to the traditional way of developing web services on an application server, this paper focuses on web services that can be created using a relational database management system.

The paper gives a short introduction to web services and compares three management systems, which are Sybase SQL Anywhere 11, Oracle Database 11g and Microsoft SQL Server 2008. In addition to giving a brief overview of what functionality each compared system is providing in terms of web services deployment, the author evaluates each one according to the presented criteria. In order to support the evaluation and comparison, a test application which consists of the three different web services is implemented with each comparable system. The services are for sending ad hoc SQL statements, running a stored procedure and running a procedure with an argument.

The results of the comparison suggest that the best tool for creating native database web services is Microsoft SQL Server, thanks to its helpful and well-written documentation, best functionality of the comparison and security measurements. Oracle Database and SQL Anywhere received the same amount of points, but as pointed out, Oracle is the easiest to work with, but does not have a detailed documentation. Also the web services created with Oracle Database cannot be configured in a desired way. The documentation for SQL Anywhere web services is meaningful, with well described examples, but service endpoints lack in security measurements and are in some ways more time-consuming to create.

In conclusion database web services provide an easy way to deploy database functionality on the web, although the creators of such services must realise the security risks that may occur.

Viited

- [1] Eric Newcomer, Greg Lomow, „Understanding SOA with Web Services,“ Pearson Education, Inc., 2005.
- [2] D. Booth, H. Haas, F. McCabe, E. Newcomer, C. Ferris, Orchard, „Web Services Architecture,“ *W3C Working Group Note*, 2004.
- [3] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, S. Weerawarana, „Unraveling the Web Services Web: An Introduction to SOAP,“ *IEEE Internet Computing*, 2002.
- [4] Nilo Mitra, Yves Lafon, „SOAP Version 1.2 Part 0: Primer,“ 2007. [Võrgumaterjal]. Kättesaadav: <http://www.w3.org/TR/soap12-part0/>. [Kasutatud 02.05.2012].
- [5] Microsoft Corporation, „What Is Windows Communication Foundation,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms731082.aspx>. [Kasutatud 12.05.2012].
- [6] iAnywhere Solutions, Inc., „Introduction to web services,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/http-s-5500129.html. [Kasutatud 28.04.2012].
- [7] iAnywhere Solutions, Inc., „Creating web services,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/create-service-http.html. [Kasutatud 29.04.2012].
- [8] iAnywhere Solutions, Inc., „Creating SOAP and DISH web services,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/create-soap-http.html. [Kasutatud 03.05.2012].
- [9] iAnywhere Solutions, Inc., „Working with data types,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/datatypes-http.html. [Kasutatud 04.05.2012].
- [10] iAnywhere Solutions, Inc., „Starting a database server that listens for web requests,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/start-server-http.html. [Kasutatud 02.05.2012].

- [11] iAnywhere Solutions, Inc., „Understanding how URLs are interpreted,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/urls-http.html. [Kasutatud 02.05.2012].
- [12] iAnywhere Solutions, Inc., „Working with SOAP headers,“ 2009. [Võrgumaterjal]. Kättesaadav: http://dcx.sybase.com/1101/en/dbprogramming_en11/working-soap.html. [Kasutatud 13.05.2012].
- [13] Oracle Corporation, „Application Architecture,“ *Oracle Database Concepts, 11g Release 1*, 2011, pp. 10-4 - 10-5.
- [14] Oracle Corporation, „Using Native Oracle XML DB Web Services,“ %1 *Oracle® XML DB Developer's Guide 11g Release 1*, 2008.
- [15] Microsoft Corporation, „Native XML Web Services Concepts,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms191274%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [16] Microsoft, „Native XML Web Services Overview,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms188266%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [17] Microsoft Corporation, „Reserving URL Namespaces by Using Http.sys,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms188682%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [18] Microsoft Corporation, „SOAP Request Message Structure,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms190796%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [19] Microsoft Corporation, „CREATE ENDPOINT (Transact-SQL),“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms181591%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [20] Microsoft Corporation, „Using WSDL,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms175476%28v=sql.105%29.aspx>. [Kasutatud 24.04.2012].
- [21] Microsoft Corporation, „Default WSDL,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms190983%28v=sql.105%29.aspx>. [Kasutatud 25.04.2012].

- [22] Microsoft Corporation, „Simple WSDL,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms190968%28v=sql.105%29.aspx>. [Kasutatud 25.04.2012].
- [23] Microsoft Corporation, „GRANT Endpoint Permissions (Transact-SQL),“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms187811%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].
- [24] International Business Machines Corporation, Microsoft Corporation, VeriSign Inc., „Web Services Security (WS-Security),“ 05.04.2002. [Võrgumaterjal]. Kättesaadav: <http://www.cgisecurity.com/ws/ws-secure.pdf>. [Kasutatud 23.04.2012].
- [25] Microsoft Corporation, „Endpoint Authentication Types,“ [Võrgumaterjal]. Kättesaadav: <http://msdn.microsoft.com/en-us/library/ms191264%28v=sql.105%29.aspx>. [Kasutatud 23.04.2012].

Lisad

Kõik lisad on samuti esitatud failidena tööga kaasneval CD-plaadil.

Lisa 1

Sybase SQL Anywhere kahe viimase veebiteenuse WSDL-dokument.

(Esitatud failina *Sybase.wSDL*)

```
<definitions name="Test_Teenus"
targetNamespace="http://localhost:8082/ope/TEST"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:s0="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types"
xmlns:s1="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlMessage"
xmlns:s2="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlResultStream" xmlns:s3="http://localhost:8082/ope/TEST"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://localhost:8082/ope/TEST">
      <s:import
namespace="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types"/>
      <s:import
namespace="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlMessage"/>
      <s:import
namespace="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlResultStream"/>
      <s:element name="KustutaKlubi">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="klubi"
nillable="true" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="KustutaKlubiResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1"
name="KustutaKlubiResult" type="s0:SqlRowSet"/>
            <s:element name="sqlcode" type="s:int"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ListKlubid">
        <s:complexType>
          <s:sequence/>
        </s:complexType>
      </s:element>
    </s:schema>
  </types>

```

```

        </s:element>
        <s:element name="ListKlubidResponse">
            <s:complexType>
                <s:sequence>
                    <s:element minOccurs="1" maxOccurs="1"
name="ListKlubidResult" type="s0:SqlRowSet"/>
                    <s:element name="sqlcode" type="s:int"/>
                </s:sequence>
            </s:complexType>
        </s:element>
        <s:element name="error" type="s:string"/>
    </s:schema>
    <s:schema attributeFormDefault="qualified"
elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types
s">
        <s:import namespace="http://www.w3.org/2001/XMLSchema"/>
        <s:complexType name="SqlRowSet">
            <s:sequence>
                <s:element ref="s:schema"/>
                <s:any/>
            </s:sequence>
        </s:complexType>
    </s:schema>
</types>
<message name="KustutaKlubiIn">
    <part name="parameters" element="s3:KustutaKlubi"/>
</message>
<message name="KustutaKlubiOut">
    <part name="parameters" element="s3:KustutaKlubiResponse"/>
</message>
<message name="ListKlubidIn">
    <part name="parameters" element="s3:ListKlubid"/>
</message>
<message name="ListKlubidOut">
    <part name="parameters" element="s3:ListKlubidResponse"/>
</message>
<message name="FaultMessage">
    <part name="error" element="s3:error"/>
</message>
<portType name="Test_TeenusSoapPort">
    <operation name="KustutaKlubi">
        <input message="s3:KustutaKlubiIn"/>
        <output message="s3:KustutaKlubiOut"/>
        <fault message="s3:FaultMessage" name="error"/>
    </operation>
    <operation name="ListKlubid">
        <input message="s3:ListKlubidIn"/>
        <output message="s3:ListKlubidOut"/>
        <fault message="s3:FaultMessage" name="error"/>
    </operation>
</portType>
<binding name="Test_TeenusSoapBinding" type="s3:Test_TeenusSoapPort">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="KustutaKlubi">
        <soap:operation
soapAction="http://localhost:8082/ope/Test/KustutaKlubi"
style="document"/>
        <input>
            <soap:body use="literal"/>

```

```

        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="error">
            <soap:fault name="error" use="literal"/>
        </fault>
    </operation>
    <operation name="ListKlubid">
        <soap:operation
soapAction="http://localhost:8082/ope/Test/ListKlubid" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
        <fault name="error">
            <soap:fault name="error" use="literal"/>
        </fault>
    </operation>
</binding>
<service name="Test_Teenus">
    <port name="Test_TeenusSoap" binding="s3:Test_TeenusSoapBinding">
        <soap:address location="http://localhost:8082/ope/Test_Teenus"/>
    </port>
</service>
</definitions>

```

Lisa 2

Oracle Database 11g esimese veebiteenuse WSDL-dokument.

(Esitatud failina *Oracle_1.wsdl*)

```

<definitions name="orawsv"
targetNamespace="http://xmlns.oracle.com/orawsv"
xsi:schemaLocation="http://schemas.xmlsoap.org/wsdl/
http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://xmlns.oracle.com/orawsv"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <types>
        <xsd:schema targetNamespace="http://xmlns.oracle.com/orawsv"
elementFormDefault="qualified">
            <xsd:element name="query">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="DDL_text" type="xsd:string"
minOccurs="0" maxOccurs="unbounded"/>
                        <xsd:element name="query_text">
                            <xsd:complexType>
                                <xsd:simpleContent>
                                    <xsd:extension base="xsd:string">
                                        <xsd:attribute name="type">
                                            <xsd:simpleType>
                                                <xsd:restriction base="xsd:NMTOKEN">
                                                    <xsd:enumeration value="SQL"/>
                                                    <xsd:enumeration value="XQUERY"/>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="bind">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                    <xsd:attribute name="name" type="xsd:string"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="bindXML">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:any/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:choice>
<xsd:element name="null_handling" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:NMTOKEN">
            <xsd:enumeration value="DROP_NULLS"/>
            <xsd:enumeration value="NULL_ATTR"/>
            <xsd:enumeration value="EMPTY_TAG"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="max_rows" type="xsd:positiveInteger"
minOccurs="0"/>
    <xsd:element name="skip_rows" type="xsd:positiveInteger"
minOccurs="0"/>
    <xsd:element name="pretty_print" type="xsd:boolean"
minOccurs="0"/>
    <xsd:element name="indentation_width"
type="xsd:positiveInteger" minOccurs="0"/>
    <xsd:element name="rowset_tag" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="row_tag" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="item_tags_for_coll" type="xsd:boolean"
minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="queryOut">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:any/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
</xsd:schema>
</types>
<message name="QueryInput">

```

```

    <part name="body" element="tns:query"/>
</message>
<message name="XMLOutput">
    <part name="body" element="tns:queryOut"/>
</message>
<portType name="ORAWSVPortType">
    <operation name="XMLFromQuery">
        <input message="tns:QueryInput"/>
        <output message="tns:XMLOutput"/>
    </operation>
</portType>
<binding name="ORAWSVBinding" type="tns:ORAWSVPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="XMLFromQuery">
        <soap:operation soapAction="http://127.0.0.1:8081/orawsv"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
        </output>
    </operation>
</binding>
<service name="ORAWSVService">
    <documentation>Oracle Web Service</documentation>
    <port name="ORAWSVPort" binding="tns:ORAWSVBinding">
        <soap:address location="http://127.0.0.1:8081/orawsv"/>
    </port>
</service>
</definitions>

```

Lisa 3

Oracle Database 11g teise veebiteenuse WSDL-dokument. (Esitatud failina *Oracle_2.wsdl*)

```

<definitions name="SP_LISTKLUBI"
targetNamespace="http://xmlns.oracle.com/orawsv/TEST/SP_LISTKLUBI"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://xmlns.oracle.com/orawsv/TEST/SP_LISTKLUBI"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types>
        <xsd:schema
targetNamespace="http://xmlns.oracle.com/orawsv/TEST/SP_LISTKLUBI"
elementFormDefault="qualified">
            <xsd:element name="SP_LISTKLUBIInput">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="P_KLUBI-VARCHAR2-OUT">
                            <xsd:complexType/>
                        </xsd:element>
                        <xsd:element name="P_ID-NUMBER-OUT">
                            <xsd:complexType/>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="SP_LISTKLUBIOutput">
                <xsd:complexType>
                    <xsd:sequence>

```



```

        <xsd:element name="P_KLUBI" type="xsd:string"/>
        <xsd:element name="P_ID" type="xsd:double"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</types>
<message name="SP_LISTKLUBIInputMessage">
    <part name="parameters" element="tns:SP_LISTKLUBIInput"/>
</message>
<message name="SP_LISTKLUBIOutputMessage">
    <part name="parameters" element="tns:SP_LISTKLUBIOutput"/>
</message>
<portType name="SP_LISTKLUBIPortType">
    <operation name="SP_LISTKLUBI">
        <input message="tns:SP_LISTKLUBIInputMessage"/>
        <output message="tns:SP_LISTKLUBIOutputMessage"/>
    </operation>
</portType>
<binding name="SP_LISTKLUBIBinding" type="tns:SP_LISTKLUBIPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="SP_LISTKLUBI">
        <soap:operation soapAction="SP_LISTKLUBI"/>
        <input>
            <soap:body parts="parameters" use="literal"/>
        </input>
        <output>
            <soap:body parts="parameters" use="literal"/>
        </output>
    </operation>
</binding>
<service name="SP_LISTKLUBIService">
    <documentation>Oracle Web Service</documentation>
    <port name="SP_LISTKLUBIPort" binding="tns:SP_LISTKLUBIBinding">
        <soap:address
location="http://127.0.0.1:8081/orawsv/TEST/SP_LISTKLUBI"/>
    </port>
</service>
</definitions>

```

Lisa 4

Oracle Database 11g kolmanda veebiteenuse WSDL-dokument.

(Esitatud failina *Oracle_3.wSDL*)

```

<definitions name="SP_KUSTUTA_KLUBI"
targetNamespace="http://xmlns.oracle.com/orawsv/TEST/SP_KUSTUTA_KLUBI"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://xmlns.oracle.com/orawsv/TEST/SP_KUSTUTA_KLUBI"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <types>
        <xsd:schema
targetNamespace="http://xmlns.oracle.com/orawsv/TEST/SP_KUSTUTA_KLUBI"
elementFormDefault="qualified">
            <xsd:element name="SP_KUSTUTA_KLUBIInput">
                <xsd:complexType>
                    <xsd:sequence>

```

```

        <xsd:element name="IN_KLUBI-VARCHAR2-IN"
type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="SP_KUSTUTA_KLUBIOutput">
    <xsd:complexType>
        <xsd:sequence/>
    </xsd:complexType>
</xsd:element>
</xsd:schema>
</types>
<message name="SP_KUSTUTA_KLUBIInputMessage">
    <part name="parameters" element="tns:SP_KUSTUTA_KLUBIInput"/>
</message>
<message name="SP_KUSTUTA_KLUBIOutputMessage">
    <part name="parameters" element="tns:SP_KUSTUTA_KLUBIOutput"/>
</message>
<portType name="SP_KUSTUTA_KLUBIPortType">
    <operation name="SP_KUSTUTA_KLUBI">
        <input message="tns:SP_KUSTUTA_KLUBIInputMessage"/>
        <output message="tns:SP_KUSTUTA_KLUBIOutputMessage"/>
    </operation>
</portType>
<binding name="SP_KUSTUTA_KLUBIBinding"
type="tns:SP_KUSTUTA_KLUBIPortType">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="SP_KUSTUTA_KLUBI">
        <soap:operation soapAction="SP_KUSTUTA_KLUBI"/>
        <input>
            <soap:body parts="parameters" use="literal"/>
        </input>
        <output>
            <soap:body parts="parameters" use="literal"/>
        </output>
    </operation>
</binding>
<service name="SP_KUSTUTA_KLUBIService">
    <documentation>Oracle Web Service</documentation>
    <port name="SP_KUSTUTA_KLUBIPort"
binding="tns:SP_KUSTUTA_KLUBIBinding">
        <soap:address
location="http://127.0.0.1:8081/orawsv/TEST/SP_KUSTUTA_KLUBI"/>
    </port>
</service>
</definitions>

```

Lisa 5

Microsoft SQL Server andmebaasisüsteemiga loodud veebiteenuse WSDL-dokument. Suur osa dokumendi tüübikirjeldustest eemaldataud, terviklik dokument esitatud failina *Microsoft.wSDL*)

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions targetNamespace="http://localhost:80/ope/test_ws"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:sqloptions="http://schemas.microsoft.com/sqlserver/2004/SOAP/Option

```

```

s" xmlns:s0="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:s1="ope" xmlns:tns="http://localhost:80/ope/test_ws">
  <wsdl:types>
    <!-- Eemaldatud dokumendi tarbeks -->
  </wsdl:types>
  <wsdl:message name="sqlbatchSoapIn">
    <wsdl:part name="parameters" element="s0:sqlbatch"/>
  </wsdl:message>
  <wsdl:message name="sqlbatchSoapOut">
    <wsdl:part name="parameters" element="s0:sqlbatchResponse"/>
  </wsdl:message>
  <wsdl:message name="s1MsgkustutaKlubiSoapIn">
    <wsdl:part name="parameters" element="s1:kustutaKlubi"/>
  </wsdl:message>
  <wsdl:message name="s1MsgkustutaKlubiSoapOut">
    <wsdl:part name="parameters" element="s1:kustutaKlubiResponse"/>
  </wsdl:message>
  <wsdl:message name="s1MsglistKlubidSoapIn">
    <wsdl:part name="parameters" element="s1:listKlubid"/>
  </wsdl:message>
  <wsdl:message name="s1MsglistKlubidSoapOut">
    <wsdl:part name="parameters" element="s1:listKlubidResponse"/>
  </wsdl:message>
  <wsdl:portType name="test_teenusSoap">
    <wsdl:operation name="sqlbatch">
      <wsdl:input message="tns:sqlbatchSoapIn"/>
      <wsdl:output message="tns:sqlbatchSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="kustutaKlubi">
      <wsdl:input name="s1MsgkustutaKlubiSoapIn"
message="tns:s1MsgkustutaKlubiSoapIn"/>
      <wsdl:output name="s1MsgkustutaKlubiSoapOut"
message="tns:s1MsgkustutaKlubiSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="listKlubid">
      <wsdl:input name="s1MsglistKlubidSoapIn"
message="tns:s1MsglistKlubidSoapIn"/>
      <wsdl:output name="s1MsglistKlubidSoapOut"
message="tns:s1MsglistKlubidSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="test_teenusSoap" type="tns:test_teenusSoap">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
    <wsdl:operation name="sqlbatch">
      <soap:operation
soapAction="http://schemas.microsoft.com/sqlserver/2004/SOAPsqlbatch"
style="document"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="kustutaKlubi">
      <soap:operation soapAction="opekustutaKlubi" style="document"/>
      <wsdl:input name="s1MsgkustutaKlubiSoapIn">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="s1MsgkustutaKlubiSoapOut">

```

```

        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="listKlubid">
    <soap:operation soapAction="opelistKlubid" style="document"/>
    <wsdl:input name="s1MsglistKlubidSoapIn">
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="s1MsglistKlubidSoapOut">
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="test_teenus">
    <wsdl:port name="test_teenus" binding="tns:test_teenusSoap">
        <soap:address location="http://localhost:8082/ope/test_ws"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Lisa 6

SOAP-päring SQL Anywhere 11 teisele veebiteenusele.

(Esitatud failina *Sybase_2_päring.xml*)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:test="http://localhost:8082/ope/TEST">
    <soapenv:Header/>
    <soapenv:Body>
        <test:ListKlubid/>
    </soapenv:Body>
</soapenv:Envelope>

```

Lisa 7

SQL Anywhere 11 süsteemi teise veebiteenuse SOAP-vastus.

(Esitatud failina *Sybase_2_vastus.xml*)

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:sqltypes='http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types'
'
xmlns:sqlmessage='http://schemas.microsoft.com/SQLServer/2001/12/SOAP/typ
es/SqlMessage'
xmlns:sqlresultstream='http://schemas.microsoft.com/SQLServer/2001/12/SOA
P/types/SqlResultStream' xmlns:tns='http://localhost:8082/ope/TEST'>
    <SOAP-ENV:Body>
        <tns:ListKlubidResponse>
            <tns:ListKlubidResult xsi:type='sqlresultstream:SqlRowSet'>
                <xsd:schema id='Schema2'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:msdata='urn:schemas-
microsoft.com:xml-msdata'>
                    <xsd:element name='rowset' msdata:IsDataSet='true'>
                        <xsd:complexType>
                            <xsd:sequence>

```

```

        <xsd:element name='row' minOccurs='0'
maxOccurs='unbounded'>
        <xsd:complexType>
        <xsd:sequence>
        <xsd:element name='Nimi' minOccurs='0'
type='xsd:string' />
        <xsd:element name='aadress' minOccurs='0'
type='xsd:string' />
        </xsd:sequence>
        </xsd:complexType>
        </xsd:element>
        </xsd:sequence>
        </xsd:complexType>
        </xsd:element>
    </xsd:schema>
    <diffgr:diffgram xmlns:msdata='urn:schemas-microsoft-com:xml-
msdata' xmlns:diffgr='urn:schemas-microsoft-com:xml-diffgram-v1'>
        <rowset>
            <row>
                <Nimi>Musta kivi kummardajad</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Võitmatu Valge</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Ruudu Liine</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Laudnikud</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Ajurebend</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Valge Mask</Nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <Nimi>Valga Valge</Nimi>
                <aadress>Valga</aadress>
            </row>
        </rowset>
    </diffgr:diffgram>
</tns:ListKlubidResult>
<tns:sqlcode>0</tns:sqlcode>
</tns:ListKlubidResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Lisa 8

SOAP-päring SQL Anywhere 11 kolmandale veebiteenusele.

(Esitatud failina *Sybase_3_päring.xml*)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:test="http://localhost:8082/ope/TEST">
  <soapenv:Header/>
  <soapenv:Body>
    <test:KustutaKlubi>
      <test:klubi>Tartu Klubi</test:klubi>
    </test:KustutaKlubi>
  </soapenv:Body>
</soapenv:Envelope>
```

Lisa 9

SQL Anywhere 11 süsteemi kolmanda veebiteenuse SOAP-vastus.

(Esitatud failina *Sybase_3_vastus.xml*)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:sqltypes='http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types'
xmlns:sqlmessage='http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlMessage'
xmlns:sqlresultstream='http://schemas.microsoft.com/SQLServer/2001/12/SOAP/types/SqlResultStream' xmlns:tns='http://localhost:8082/ope/TEST'>
  <SOAP-ENV:Body>
    <tns:KustutaKlubiResponse>
      <tns:KustutaKlubiResult xsi:type='sqlresultstream:SqlRowSet'>
        <xsd:schema id='Schema2'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:msdata='urn:schemas-
microsoft.com:xml-msdata'>
          <xsd:element name='rowset' msdata:IsDataSet='true'>
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name='row' minOccurs='0'
maxOccurs='unbounded'>
                  <xsd:complexType>
                    <xsd:sequence>
                      </xsd:sequence>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
              </xsd:complexType>
            </xsd:element>
          </xsd:schema>
          <diffgr:diffgram xmlns:msdata='urn:schemas-microsoft-com:xml-
msdata' xmlns:diffgr='urn:schemas-microsoft-com:xml-diffgram-v1'>
            <rowset>
              </rowset>
          </diffgr:diffgram>
        </tns:KustutaKlubiResult>
        <tns:sqlcode>0</tns:sqlcode>
      </tns:KustutaKlubiResponse>
```

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Lisa 10

SOAP-päring Oracle Database esimesele veebiteenusele.

(Esitatud failina *Oracle_1_päring.xml*)

```
<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-envelope ">
  <env:Body>
    <query xmlns="http://xmlns.oracle.com/orawsv">
      <query_text type="SQL">
        <![CDATA[SELECT "Id", "Eesnimi", "Perenimi" FROM test.isik
WHERE "Klubi" = 54 ]]>
      </query_text>
      <pretty_print>>false</pretty_print>
    </query>
  </env:Body>
</env:Envelope>
```

Lisa 11

Oracle Database süsteemi esimese veebiteenuse SOAP-vastus.

(Esitatud failina *Oracle_1_vastus.xml*)

```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://www.w3.org/2002/06/soap-envelope">
  <soap:Body>
    <queryOut xmlns="http://xmlns.oracle.com/orawsv">
      <ROWSET>
        <ROW>
          <Id>72</Id>
          <Eesnimi>Maari</Eesnimi>
          <Perenimi>Mustikas</Perenimi>
        </ROW>
        <ROW>
          <Id>82</Id>
          <Eesnimi>Maria</Eesnimi>
          <Perenimi>Murakas</Perenimi>
        </ROW>
        <ROW>
          <Id>85</Id>
          <Eesnimi>Liis</Eesnimi>
          <Perenimi>Metsonen</Perenimi>
        </ROW>
        <ROW>
          <Id>86</Id>
          <Eesnimi>Toomas</Eesnimi>
          <Perenimi>Remmelgas</Perenimi>
        </ROW>
      </ROWSET>
    </queryOut>
  </soap:Body>
</soap:Envelope>
```

Lisa 12

SOAP-päring Oracle Database teisele veebiteenusele. (Esitatud failina *Oracle_2_päring.xml*)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sp="http://xmlns.oracle.com/orawsv/TEST/SP_LISTKLUBI">
  <soapenv:Header/>
  <soapenv:Body>
    <sp:SP_LISTKLUBIInput>
      <sp:P_KLUBI-VARCHAR2-OUT/>
      <sp:P_ID-NUMBER-OUT/>
    </sp:SP_LISTKLUBIInput>
  </soapenv:Body>
</soapenv:Envelope>
```

Lisa 13

Oracle Database süsteemi teise veebiteenuse SOAP-vastus.

(Esitatud failina *Oracle_2_vastus.xml*)

```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SP_LISTKLUBIOutput
xmlns="http://xmlns.oracle.com/orawsv/TEST/SP_LISTKLUBI">
      <P_ID>54</P_ID>
      <P_KLUBI>Ajurebend</P_KLUBI>
    </SP_LISTKLUBIOutput>
  </soap:Body>
</soap:Envelope>
```

Lisa 14

SOAP-päring Oracle Database kolmandale veebiteenusele.

(Esitatud failina *Oracle_3_päring.xml*)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sp="http://xmlns.oracle.com/orawsv/TEST/SP_KUSTUTA_KLUBI">
  <soapenv:Header/>
  <soapenv:Body>
    <sp:SP_KUSTUTA_KLUBIInput>
      <sp:IN_KLUBI-VARCHAR2-IN>Tartu Klubi</sp:IN_KLUBI-VARCHAR2-IN>
    </sp:SP_KUSTUTA_KLUBIInput>
  </soapenv:Body>
</soapenv:Envelope>
```

Lisa 15

Oracle Database süsteemi kolmanda veebiteenuse SOAP-vastus.

(Esitatud failina *Oracle_3_vastus.xml*)

```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```



```

    <soap:Body>
      <SP_KUSTUTA_KLUBIOutput
xmlns="http://xmlns.oracle.com/orawsv/TEST/SP_KUSTUTA_KLUBI">
        </SP_KUSTUTA_KLUBIOutput>
      </soap:Body>
    </soap:Envelope>

```

Lisa 16

SOAP-päring SQL Server esimesele veebiteenusele.

(Esitatud failina *Microsoft_1_päring.xml*)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:soap="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/SqlPara
meter">
  <soapenv:Header/>
  <soapenv:Body>
    <soap:sqlbatch>
      <soap:BatchCommands><![CDATA[SELECT isik.Id, isik.eesnimi,
isik.perenimi FROM ope.dbo.isik WHERE Klubi = 54;]]></soap:BatchCommands>
    </soap:sqlbatch>
  </soapenv:Body>
</soapenv:Envelope>

```

Lisa 17

SQL Server süsteemi esimese veebiteenuse SOAP-vastus.

(Esitatud failina *Microsoft_1_vastus.xml*)

```

<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/type
s"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/2004/SOAP/types
/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/
SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/2004/SOAP/t
ypes/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/2004/SOAP/ty
pes/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <SOAP-ENV:Body>
    <sql:sqlbatchResponse>
      <sql:sqlbatchResult>
        <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true" msdata:UDTColumnValueWrapped="true">
          <xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
            <xsd:simpleType name="int">
              <xsd:restriction base="xsd:int"/>
            </xsd:simpleType>

```

```

        <xsd:simpleType name="varchar">
          <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
      </xsd:schema>
      <xsd:schema targetNamespace="urn:schemas-microsoft-
com:sql:SqlRowSet1" elementFormDefault="qualified">
        <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes"/>
        <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="Id" type="sqltypes:int"/>
                    <xsd:element name="eesnimi">
                      <xsd:simpleType>
                        <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1061" sqltypes:sqlCompareOptions="IgnoreCase
IgnoreKanaType IgnoreWidth">
                          <xsd:maxLength value="50"/>
                        </xsd:restriction>
                      </xsd:simpleType>
                    </xsd:element>
                    <xsd:element name="perenimi">
                      <xsd:simpleType>
                        <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1061" sqltypes:sqlCompareOptions="IgnoreCase
IgnoreKanaType IgnoreWidth">
                          <xsd:maxLength value="50"/>
                        </xsd:restriction>
                      </xsd:simpleType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:schema>
      <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-
diffgram-v1">
        <SqlRowSet1 xmlns="urn:schemas-microsoft-
com:sql:SqlRowSet1">
          <row>
            <Id>72</Id>
            <eesnimi>Maari</eesnimi>
            <perenimi>Mustikas</perenimi>
          </row>
          <row>
            <Id>82</Id>
            <eesnimi>Maria</eesnimi>
            <perenimi>Murakas</perenimi>
          </row>
          <row>
            <Id>85</Id>
            <eesnimi>Liis</eesnimi>
            <perenimi>Metsonen</perenimi>
          </row>
        </SqlRowSet1>
      </diffgram>

```

```

        </row>
        <row>
            <Id>86</Id>
            <eesnimi>Toomas</eesnimi>
            <perenimi>Rommelgas</perenimi>
        </row>
    </SqlResultSet1>
</diffgr:diffgram>
</sqlresultstream:SqlResultSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
    <sqlrowcount:Count>4</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
</sql:sqlbatchResult>
</sql:sqlbatchResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Lisa 18

SOAP-päring SQL Server teisele veebiteenusele. (Esitatud failina *Microsoft_2_päring.xml*)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ope="ope">
    <soapenv:Header/>
    <soapenv:Body>
        <ope:listKlubid/>
    </soapenv:Body>
</soapenv:Envelope>

```

Lisa 19

SQL Server süsteemi teise veebiteenuse SOAP-vastus.

(Esitatud failina *Microsoft_2_vastus.xml*)

```

<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/types"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/2004/SOAP/types
/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/
SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/2004/SOAP/t
ypes/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/2004/SOAP/t
ypes/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:method="ope">
    <SOAP-ENV:Body>
        <method:listKlubidResponse>
            <method:listKlubidResult>

```

```

    <sqlresultstream:SqlRowSet xsi:type="sqlsoaptypes:SqlRowSet"
msdata:UseDataSetSchemaOnly="true" msdata:UDTColumnValueWrapped="true">
    <xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes">
        <xsd:simpleType name="varchar">
            <xsd:restriction base="xsd:string"/>
        </xsd:simpleType>
    </xsd:schema>
    <xsd:schema targetNamespace="urn:schemas-microsoft-
com:sql:SqlRowSet1" elementFormDefault="qualified">
        <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/2004/sqltypes"/>
        <xsd:element name="SqlRowSet1" msdata:IsDataSet="true"
msdata:DataSetNamespace="urn:schemas-microsoft-com:sql:SqlDataSet"
msdata:DataSetName="SqlDataSet">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="row" minOccurs="0"
maxOccurs="unbounded">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="nimi">
                                    <xsd:simpleType>
                                        <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1061" sqltypes:sqlCompareOptions="IgnoreCase
IgnoreKanaType IgnoreWidth">
                                            <xsd:maxLength value="100"/>
                                        </xsd:restriction>
                                    </xsd:simpleType>
                                </xsd:element>
                                <xsd:element name="aadress">
                                    <xsd:simpleType>
                                        <xsd:restriction base="sqltypes:varchar"
sqltypes:localeId="1061" sqltypes:sqlCompareOptions="IgnoreCase
IgnoreKanaType IgnoreWidth">
                                            <xsd:maxLength value="50"/>
                                        </xsd:restriction>
                                    </xsd:simpleType>
                                </xsd:element>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-
diffgram-v1">
        <SqlRowSet1 xmlns="urn:schemas-microsoft-
com:sql:SqlRowSet1">
            <row>
                <nimi>Laudnikud</nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <nimi>Ajurebend</nimi>
                <aadress>Tartu</aadress>
            </row>
            <row>
                <nimi>Ruudu Liine</nimi>
                <aadress>Tartu</aadress>
            </row>
        </SqlRowSet1>
    </diffgram>

```

```

        </row>
        <row>
            <nimi>Võitmatu Valge</nimi>
            <address>Tartu</address>
        </row>
        <row>
            <nimi>Valge Mask</nimi>
            <address>Tartu</address>
        </row>
        <row>
            <nimi>Musta kivi kummardajad</nimi>
            <address>Tartu</address>
        </row>
        <row>
            <nimi>Valga Valge</nimi>
            <address>Valga</address>
        </row>
        <row>
            <nimi>Tartu Klubi</nimi>
            <address>Tartu</address>
        </row>
    </SqlRowSet1>
</diffgr:diffgram>
</sqlresultstream:SqlRowSet>
<sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
    <sqlrowcount:Count>8</sqlrowcount:Count>
</sqlresultstream:SqlRowCount>
<sqlresultstream:SqlResultCode
xsi:type="sqlsoaptypes:SqlResultCode">0</sqlresultstream:SqlResultCode>
</method:listKlubidResult>
</method:listKlubidResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Lisa 20

SOAP-päring SQL Server kolmandale veebiteenusele.

(Esitatud failina *Microsoft_3_päring.xml*)

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ope="ope"
xmlns:test="http://localhost:80/ope/test_ws">
    <soapenv:Header/>
    <soapenv:Body>
        <ope:kustutaKlubi>
            <ope:klubi>Tartu Klubi</ope:klubi>
        </ope:kustutaKlubi>
    </soapenv:Body>
</soapenv:Envelope>

```

Lisa 21

SQL Server süsteemi kolmanda veebiteenuse SOAP-vastus.

(Esitatud failina *Microsoft_3_vastus.xml*)

```

<SOAP-ENV:Envelope xml:space="preserve"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:sql="http://schemas.microsoft.com/sqlserver/2004/SOAP"
xmlns:sqlsoaptypes="http://schemas.microsoft.com/sqlserver/2004/SOAP/type
s"
xmlns:sqlrowcount="http://schemas.microsoft.com/sqlserver/2004/SOAP/types
/SqlRowCount"
xmlns:sqlmessage="http://schemas.microsoft.com/sqlserver/2004/SOAP/types/
SqlMessage"
xmlns:sqlresultstream="http://schemas.microsoft.com/sqlserver/2004/SOAP/t
ypes/SqlResultStream"
xmlns:sqltransaction="http://schemas.microsoft.com/sqlserver/2004/SOAP/ty
pes/SqlTransaction"
xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:method="ope">
  <SOAP-ENV:Body>
    <method:kustutaKlubiResponse>
      <method:kustutaKlubiResult>
        <sqlresultstream:SqlRowCount
xsi:type="sqlrowcount:SqlRowCount">
          <sqlrowcount:Count>1</sqlrowcount:Count>
        </sqlresultstream:SqlRowCount>
        <sqlresultstream:SqlResultCode
xsi:type="sqlsoaptypes:SqlResultCode">0</sqlresultstream:SqlResultCode>
      </method:kustutaKlubiResult>
    </method:kustutaKlubiResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```