

# TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduste instituut

Infotehnoloogia eriala

**Sander Sepp**

## **SAP Sybase SQL Anywhere veebiteenused**

**Bakalaureusetöö**

Juhendaja: Vambola Leping

Autor: ..... “.....” ..... 2013

Juhendaja: ..... “.....” ..... 2013

Lubada kaitsmisele

Professor: ..... “.....” ..... 2013

TARTU 2013

# Sisukord

<b>Sissejuhatus .....</b>	<b>4</b>
<b>1 SQL Anywhere veebiteenused .....</b>	<b>5</b>
1.1 SAP .....	5
1.2 Sybase .....	5
1.3 SQL Anywhere .....	6
1.4 Veebiteenused .....	6
1.5 SQL Anywhere veebiteenused .....	6
1.6 SQL Anywhere serveri seadistamine .....	7
1.7 HTTP veebiteenuse loomine .....	8
1.8 HTTP veebiteenuse näide .....	10
1.9 SOAP ja WSDL .....	12
1.10 SOAP teenus SQL Anywhere andmebaasis .....	12
1.11 SOAP ja DISH teenuse loomine .....	12
1.12 SOAP veebiteenuse näide .....	15
1.13 Veebiteenuste arendamise lihtsustamiseks .....	16
<b>2 Veebiteenused ja dünaamiliselt koostatud SQL päringud .....</b>	<b>19</b>
2.1 Dünaamiliselt koostatud SQL päringud .....	19
2.2 SQL Anywhere veebiteenused ja EXECUTE IMMEDIATE .....	20
2.3 Süsteemne protseduur sa_describe_query .....	22
2.4 HTML veebiteenuse vastuse käsitsi vormindamine .....	22
2.5 XML Veebiteenuse vastuse käsitsi vormindamine .....	26
2.6 SOAP veebiteenuse vastuse käsitsi vormindamine .....	29
<b>3 SQL Anywhere veebiteenuste õppematerjal .....</b>	<b>30</b>
3.1 Andmebaasid (MTAT.03.264) .....	30
3.2 Andmemudel .....	30
3.3 Veebiteenused aines Andmebaasid .....	31
3.4 PmWiki .....	31
3.5 Veebiteenuste õppematerjal .....	31
3.6 Ülesehitus .....	32
3.7 Õppematerjali kasutamine .....	33
<b>Kokkuvõte .....</b>	<b>34</b>
<b>SAP Sybase SQL Anywhere web services .....</b>	<b>35</b>

<b>Kirjandus.....</b>	<b>36</b>
<b>Lisad.....</b>	<b>40</b>
Lisa 1 Andmebaasid kursuse õppematerjal .....	40
Lisa 2 Veebiteenuse DISH_veebiteenus WSDL sisu .....	41
Lisa 3 Veebiteenuse SOAP_veebiteenus päring ja vastus.....	44

# Sissejuhatus

Töö eesmärk on uurida SAP Sybase SQL Anywhere andmebaasi veebiteenuseid ja luua ainele Andmebaasid (MTAT.03.264) vastavasisulised õppematerjalid. SAP (Saksa keeles *Systemanalyse und Programmentwicklung*) on 1972. aastal Saksamaal loodud tarkvara arendamise ettevõtte [1]. Sybase on SAP tütarettevõtte, mille SAP omandas aastal 2010 [2]. SQL Anywhere on Sybase poolt väljaarendatud relatsiooniline andmebaas koos juhtimissüsteemiga.

Töö koosneb kolmest põhilisest osast - SQL Anywhere veebiteenused, veebiteenuste ja dünaamiliste SQL päringute koostöö probleem ja aine Andmebaasid õppematerjali tutvustus.

Esimeses peatükis tutvustatakse SQL Anywhere veebiteenuste toimimist andmebaasis. Eraldi tuuakse välja andmebaasi serveri käivitamine, näidatakse kuidas lubada veebiteenustele võrgusuhtlust. Vaadeldakse tavapärase HTTP veebiteenuse ja SOAP veebiteenuse loomist, mis on illustreeritud lahtiseletatud näidetega. Lisaks on tutvustatud SQL Anywhere funktsionaalsusi, mis lihtsustavad veebiteenuste arendamist.

Töö teises osas tuuakse välja dünaamiliste SQL päringute ja veebiteenuste probleemne koostöö, mis seletatakse lahti koos näitega. Probleemi lahendamiseks antud töös uuritakse SQL päringut lahtiseletavat andmebaasi protseduuri. Näitena on toodud eelmainitud protseduuri kasutavad parsijad, mis vormindavad dünaamilise SQL päringu vastuse HTML või XML vormindusse. Näidatakse kuidas parsijaid kasutada koos veebiteenustega.

Kolmas peatükk tutvustab aine Andmebaasid (MTAT.03.264) õppematerjali, mille koostas autor antud bakalaureusetöö raames. Seletatakse, mis vahenditega õppematerjal on koostatud, kuidas ülesehitatud ja kuidas on seda võimalik kasutada.

Käesoleva töö väljunditeks on SQL Anywhere veebiteenuste temaatiline kirjalik töö ja sama temaatiline wiki formaadis õppematerjalid ainele Andmebaasid (MTAT.03.264). Töös on kasutatud SQL Anywhere Developer Edition tarkvarapaketi versiooni 12.0.1. Kõik SQL laused on koostatud ja käivitatud tarkvarapaketi kaasasoleva rakendusega Interactiv SQL (iSQL).

# Peatükk 1

## 1 SQL Anywhere veebiteenused

### 1.1 SAP

SAP on ettevõtte tarkvarale keskenduv rahvusvaheline kontsern, mille peakontor asub Saksamaal. Firma tulu arvestades on SAP ettevõtte tarkvara ja tarkvaraga seonduval alal maailmas esikohal. Turu hõivamise seisukohalt on SAP maailmas kolmas iseseisev tarkvara arendamise ettevõtte. SAP omab üle 232 000 kliendi 188 riigis. Ettevõtte tegutseb rohkem kui 130 riigis ja seisuga 31.12.12 oli firmal 64 422 töötajat. SAP 2012. aasta käive oli üle 16,22 miljardi euro. [3] [4] [5]

### 1.2 Sybase

Sybase on alatest aastast 2010 SAP tütar-ettevõtte [1], mis on töötanud välja mitmeid tarkvaralisi tooteid. Ettevõtte pakkus 20.04.13 seisuga 37 toodet, mis on jagatud järgmisteks kategooriateks [6]:

- Andmebaaside haldamine
- Ettevõtte järjepidavus
- Andmelaod
- Modelleerimine ja arendamine
- Mobiilsed teenused
- Mobiilne ettevõtte
- Ettevõtte strateegia ja analüüs
- Finantsiliste teenuste lahendused

## **1.3 SQL Anywhere**

SQL Anywhere on SAP Sybase poolt pakutav relatsioonilise andmebaasi haldamise tarkvarapakett ehk andmebaasi juhtimissüsteem [7]. SAP andis 16.04.13 välja kõige uuema SQL Anywhere versiooni, SQL Anywhere 16.0 [8], mis on peale versiooni 12.0.1 esimene ametliku väljalaskega SQL Anywhere tarkvarapakett.

Antud töös on kasutatud SQL Anywhere 12.0.1 versiooni Developer Edition paketti. SQL käskluste rakendamiseks on kasutatud tarkvarapaketiga kaasas olevat Interactive SQL rakendust.

## **1.4 Veebiteenused**

W3C [9] kohaselt on veebiteenus tarkvaraline lahendus, mis on välja arendatud, et toetada koostööd eri platvormide ja masinate vahel. Veebiteenus töötab üle võrgu. Veebiteenuse informatsioon on teenuse tarbija jaoks kirjeldatud masinloetavas formaadis, näiteks WSDL faili kujul. Teised süsteemid suhtlevad veebiteenusega vastavalt ette kirjutatud reeglitele, kasutades selle jaoks SOAP teatiseid. Tüüpiliselt edastatakse SOAP ümbrik HTTP kaudu, kasutades XML serialiseerimist koos teiste veebiga seotud standarditega [10].

## **1.5 SQL Anywhere veebiteenused**

SQL Anywhere andmebaas sisaldab endas funktsionaalsust luua veebiteenuseid. Need veebiteenused on loodavad ja hallatavad otse andmebaasis. Selle funktsionaalsuse kasutamiseks on vaja andmebaasi server käivitada veebiteenustele võrgusuhtlust lubaval viisil. SQL Anywhere veebiteenused toetavad tavalisi HTTP päringuid kui ka SOAP päringuid, mis on saadetud veebibrauserist või mingist teisest klient-rakendusest. Selline tehnoloogia pakub andmebaasis alternatiivi üldlevinud liidestele JDBC (Java DataBase Connection) ja ODBC (Oracle Database Connection). Veebiteenused on andmebaasis kergesti kasutatavad ja lisa komponentide järele vajadust ei ole. Lisaks veebiteenuste pakkumisele on võimalik ka teenuseid tarbida üle HTTP. Sellisel juhul on SQL Anywhere võimeline töötama ka klient-rakendusena, mis tarbib teisi standardseid veebiteenuseid. [11]

## 1.6 SQL Anywhere serveri seadistamine

Veebiteenuste kasutamiseks on vaja käsurealt käivitada SQL Anywhere andmebaasi server veebiteenustele võrgusuhtlust lubava klausliga `-xs` [12]. Alljärgnevalt on toodud käskluse süntaks.

```
{ dbeng12 | dbsrv12 } -xs { võrguprotokoll[,...] } ...
```

*võrguprotokoll*:

```
NONE  
| HTTP [ ( valik=väärtus;... ) ]  
| HTTPS [ ( valik=väärtus;... ) ]
```

Andmebaasi serveri käivitamiseks on kaks eraldiseisvat käsku [13]:

- **dbeng12** käsklus käivitab SQL Anywhere andmebaasi lokaalse serveri. Server on kättesaadav ainult lokaalsest arvutist. Lokaalne arvuti on masin, kus server käivitatakse.
- **dbsrv12** käsklus käivitab SQL Anywhere andmebaasi võrgu serveri. Server on kättesaadav üle võrgu.

Koos `-xs` klausliga tuleb määrata ka võrguprotokoll, mille kaudu veebiteenused suhtlema hakkavad. Võrguprotokolliks võib olla HTTP või HTTPS. Koos võrguprotokolliga saab määrata erinevaid protokollide parameetreid. Näiteks port, päringu *timeout* e. ajalõpp, HTTPS puhul sertifikaadiga seotud valikud jne. [14]

### Lokaalne HTTP veebiserver

```
dbeng12 -xs http(port=8082) "andmebaasi_asukoht\andmebaas.db"
```

Antud käsklus käivitab andmebaasi `"andmebaasi_asukoht\andmebaas.db"` serveri lokaalses režiimis. Klausliga `-xs http(port=8082)` lubatakse veebiteenustele võrgusuhtlust, kasutades HTTP võrguprotokollide. Server kuulab HTTP päringuid pordilt 8082.

### Lokaalne HTTPS veebiserver

```
dbeng12 -xs https(port=8082; identity=sertifikaadi.fail;  
identity_password=parool) "andmebaasi_asukoht\andmebaas.db"
```

Antud käsklus käivitab andmebaasi `"andmebaasi_asukoht\andmebaas.db"` serveri lokaalses režiimis. Klausliga `-xs https(port=8082; identity=sertifikaadi.fail; identity_password=parool)` lubatakse veebiteenustele võrgusuhtlust, kasutades HTTPS

võrguprotokolli. Server kuulab HTTPS päringuid pordilt 8082. HTTPS kasutamiseks antakse serverile ette sertifikaat ja selle parool.

### Võrgust kättesaadav HTTP veebiserver

```
dbsrv12 -xs http(port=80) "andmebaasi_asukoht\andmebaas.db"
```

Antud käsklus käivitab andmebaasi "*andmebaasi\_asukoht\andmebaas.db*" serveri võrgu režiimis. Klausliga *-xs http(port=80)* lubatakse veebiteenustele võrgusuhtlus, kasutades HTTP võrguprotokolli. Server kuulab HTTP päringuid pordilt 80.

## 1.7 HTTP veebiteenuse loomine

Andmebaasis tavapärase HTTP veebiteenuse loomiseks [15] tuleb kävitada CREATE SERVICE käsklus. Alljärgnevalt on toodud käskluse süntaks.

```
CREATE SERVICE teenuse_nimi  
  TYPE { 'RAW' | 'HTML' | 'JSON' | 'XML' }  
  [ URL [PATH] { ON | OFF | ELEMENTS } ]  
  [ atribuudid ]  
  [ AS { sql | NULL } ]
```

*atribuudid:*

```
[ AUTHORIZATION { ON | OFF } ]  
[ ENABLE | DISABLE ]  
[ METHODS 'meetod,...' ]  
[ SECURE { ON | OFF } ]  
[ USER { kasutaja_nimi | NULL } ]
```

*meetod:*

```
DEFAULT  
| POST  
| GET  
| HEAD  
| PUT  
| DELETE  
| NONE  
| *
```

*teenuse\_nimi* on nimi, mida teenus andmebaasis omandab. Nimi võib olla ükskõik mis tähtede ja numbrite jada ja võib sisaldada ka järgmisi kirjamärke: /, -, \_, ., !, ~, \*, ', (, ). Ainuke tingimus on, et nimi ei tohi alata kaldkriipsuga ja ei tohi sisaldada kahte järjestikust



kalkkriipsu. Kui teenuse nimeks on *root*, siis see määratakse vaikimisi teenuseks. Päring suunatakse vaikimisi veebiteenusele, kui päritavat teenust andmebaasist ei leitud.

**TYPE** klausel määrab ära, kuidas teenuse väljundit vormindatakse. Tavapärase HTTP teenuse puhul on neli erinevat vormindamise tüüpi: RAW, HTML, JSON ja XML. Iga variandi korral vormindatakse väljund vastavasse tüüpi.

- RAW - Teenuse väljund jäetakse vormindamata, kuvatakse väljund andmete tekstilise jadana.
- HTML - Väljund vormindatakse HTML formaadis tabeliks.
- JSON - Väljund vormindatakse JSON formaati.
- XML -Väljund vormindatakse XML dokumendiks, kus ülelement on *root* ja väljundi iga rea element on *row*. Veerud vormindatakse elemendi *row* parameetriteks.

**URL** klausel määrab ära teenuse aadressis kalkkriipsuga eraldatud parameetrite kasutamise. Võimalikud väärtused on OFF, ON, ELEMENTS.

- OFF - Selle piiranguga keelatakse teenuse URL järel kasutatavad parameetrid. See tähendab, et teenuse nime järel ei tohi tulla kalkkriipsuga eraldatud parameetreid.
- ON - Selle piiranguga lubatakse teenuse URL järel kasutatavad parameetrid. Teenuse nime järel võib kalkkriipsuga eraldatult kirjutada ühe parameetri, mis on koodis kättesaadav funktsiooniga HTTP\_VARIABLE('URL'). Näiteks *teenuse\_nimi/parameeter/parameeter* puhul on HTTP\_VARIABLE('URL') käsuga võimalik saada kätte parameeter väärtusega *parameeter/parameeter*.
- ELEMENTS - Selle piiranguga lubatakse teenuse URL järel kasutatavad parameetrid. Lubatud on kuni kümme eraldi tuvastatavat parameetrit. Koodist on parameetrid kättesaadavad funktsiooniga HTTP\_VARIABLE('URL<X>'), kus <X> on parameetri järjekorra arv. Näiteks *teenuse\_nimi/parameeter1/parameeter2* puhul saab funktsiooniga HTTP\_VARIABLE('URL') kätte *parameeter1/parameeter2*, HTTP\_VARIABLE('URL1') funktsiooniga *parameeter1* ja HTTP\_VARIABLE('URL2') abil *parameeter2*.

**AUTHORIZATION** klausel määrab ära, kas teenus nõuab väljakutsudes autoriseerimist. Autoriseerimise jaoks kasutatakse RFC 2617 autentimist, mis varjab kasutajanime ja parooli, kasutades base64 kodeeringut [16].

**ENABLE / DISABLE** määrab, kas veebiteenus on väljakutsutav. Vaikimisi on ENABLE ehk veebiteenus on aktiivne ja väljakutsutav.

**METHODS** klausliga saab määrata, mis HTTP meetodid on loodavas teenuses kasutatavad. Valikus on DEFAULT, POST, GET, HEAD, PUT, DELETE, NONE ja \*.

**SECURE** määrab ära, kas veebiteenus on kasutatav HTTP kaudu või on teenust lubatud kasutada ainult üle HTTPS võrguprotokolliga.

**USER** klausel määrab ära andmebaasi kasutaja või kasutajate grupi, kellele on veebiteenuse kasutamine lubatud. Autoriseerimise käigus lubatakse selle klausliga määratud kasutajatel veebiteenust tarbida.

## 1.8 HTTP veebiteenuse näide

Järgnevalt on toodud näide tavalisest HTTP kaudu kättesaadavast veebiteenusest.

```
CREATE SERVICE HTML_veebiteenus
  TYPE 'HTML'
  AUTHORIZATION OFF
  USER DBA
  AS SELECT 'Hello world!';
```

Selle koodijupiga luuakse veebiteenus nimega *HTML\_veebiteenus*. Teenus on HTML tüüpi, lisaks ei nõua see autoriseerimist, lubatud kasutaja on DBA ja teenus väljastab SQL päringu *SELECT 'Hello world!'* tulemuse.

Kui eelnevalt on SQL Anywhere lokaalne server käivitatud koos *-xs* atribuudiga, HTTP protokolliga ja pordiga 8082, siis on teenus kätte saadav aadressilt localhost:8082/HTML\_veebiteenus.

Järgnevalt on toodud päringu tulemused, mis on tehtud veebibrauserist aadressile [http://localhost:8082/HTML\\_veebiteenus](http://localhost:8082/HTML_veebiteenus).

### Päringu päis

```
GET /HTML_veebiteenus HTTP/1.1
Host: localhost:8082
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101
Firefox/15.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: et,et-ee;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

### Vastuse päis

```
HTTP/1.1 200 OK
Date: Sun, 21 Apr 2013 12:02:53 GMT
Transfer-Encoding: chunked
Connection: Keep-Alive
Expires: Sun, 21 Apr 2013 12:02:53 GMT
Content-Type: text/html; charset=windows-1257
Server: SQLAnywhere/12.0.1.3457
```

### Vastus

```
<html>
  <head>
    <title>/HTML_veebiteenus</title>
  </head>
  <body>
    <h3>/HTML_veebiteenus</h3>
    <table border=1>
      <tr class="header">
        <th>
          <b>
            'Hello world!'
          </b>
        </th>
      </tr>
      <tr>
        <td>
          Hello world!
        </td>
      </tr>
    </table>
    <br>
    <hr>
    <address>
```

```
        SQLAnywhere/12.0.1.3457
    </address>
</body>
</html>
```

Veebilehitseja tegi HTTP GET päringu veebiteenuseni HTML\_veebiteenus, mis asub aadressil `http://localhost:8082/HTML_veebiteenus`, ja sai sealt vastuseks HTML vormingus andmed, mille sees on päringu *SELECT 'Hello world!'* vastus HTML tabeli vormindusega.

## 1.9 SOAP ja WSDL

SOAP [17] on lihtsa ülesehitusega protokoll, mille ülesandeks on vahetada struktureeritud informatsiooni erinevate keskkondade vahel. SOAP kasutab teatiste raamistiku defineerimiseks XML tehnoloogiat. Protokoll võimaldab info vahetamist üle mitme erineva võrguprotokolliga. Raamistik on kavandatud olema programmeerimiskeeltest sõltumatu. WSDL [18] defineerib, kasutades XML tehnoloogiaid, SOAP teenuste ülesehitust. Iga SOAP teenuse päringu ja vastuse struktuur kirjeldatakse WSDL failis, et teenuse tarbijal oleks standardne võimalus teenust kasutada. WSDL määratleb iga teenuse kohta tema sihtpunkti (*endpoint*) olles ise kõigi kirjeldatud teenuste ülemsihtpunkt.

### 1.10 SOAP teenus SQL Anywhere andmebaasis

SOAP teenustega seostuvad SQL Anywhere andmebaasis kahte tüüpi veebiteenused: SOAP ja DISH. SOAP tüüpi teenusega saab teha eraldi SOAP teenuseid, mis on väljakutsutatavad vaid päringutega, mis on SOAP vormingus. DISH tüüpi teenusega saab SOAP teenuseid grupeerida. DISH teenus töötab SOAP teenuste jaoks nagu sihtpunkt (*endpoint*) ja pakub enda grupeeringus olevate SOAP teenuste kohta ka WSDL faili.

### 1.11 SOAP ja DISH teenuse loomine

SOAP veebiteenuse loomiseks [19] andmebaasis tuleb kävitada CREATE SERVICE käsklus. Alljärgnevalt on toodud käskluse süntaks.

```

CREATE SERVICE teenuse_nimi
  TYPE 'SOAP'
  [ DATATYPE { ON | OFF | IN | OUT } ]
  [ FORMAT {'DNET' | 'CONCRETE' [EXPLICIT {ON|OFF}] | 'XML' | NULL} ]
  [ atribuudid ]
  AS sql

```

*atribuudid:*

```

[ AUTHORIZATION { ON | OFF } ]
[ ENABLE | DISABLE ]
[ METHODS 'meetod,...' ]
[ SECURE { ON | OFF } ]
[ USER { kasutaja_nimi | NULL } ]

```

*meetod :*

```

DEFAULT
| POST
| HEAD
| NONE

```

DISH teenuse loomiseks andmebaasis tuleb k ivitada CREATE SERVICE k asklus. Allj argnevalt on toodud k askluse s yntaks.

```

CREATE SERVICE teenuse_nimi
  TYPE 'DISH'
  [ GROUP { grupi_nimi | NULL } ]
  [ FORMAT {'DNET' | 'CONCRETE' [EXPLICIT {ON|OFF}] | 'XML' | NULL} ]
  [ atribuudid ]

```

*atribuudid:*

```

[ AUTHORIZATION { ON | OFF } ]
[ ENABLE | DISABLE ]
[ METHODS 'meetod,...' ]
[ SECURE { ON | OFF } ]
[ USER { kasutaja_nimi | NULL } ]

```

*meetod:*

```

DEFAULT
| POST
| GET
| HEAD
| NONE
| *

```

**TYPE** klausel määrab ära, kas on tegu SOAP või DISH tüüpi teenusega.

- SOAP puhul parsitakse väljund XML dokumendiks, mis on SOAP ümbriku vormindusega. Teenuse väljundit saab täpsustada FORMAT klausli abil. Päring SOAP teenusele peab olema SOAP vormindusega.
- DISH on SOAP teenuste sihtpunkt (*endpoint*), mis viitab igale SOAP teenusele tema grupis. DISH teenus genereerib kõiki viidatud teenuseid kirjeldava WSDL dokumendi.

**GROUP** klausliga saab DISH veebiteenuses määrata ära SOAP teenuste grupi, millele DISH teenus viitab. Gruppi saab teenuses määrata teenuse nimega. Kui teenuse nimeks on *grupp/grupp/teenuse\_nimi* siis see tähendab, et teenuse enda nimi on *teenuse\_nimi* ja tema grupiks *grupp/grupp*. Kui DISH teenuses jääb grupp määramata, siis viitab teenus kõikidele SOAP teenustele andmebaasis.

**DATATYPE** klausliga saab andmetüüpide markeerimist muuta SOAP teenuse sisendis ja väljundis. Võimalikud väärtused saavad olla ON, OFF, IN, OUT.

- ON - Määratakse ära sisendi ja väljundi andmetüübid.
- OFF - Kõik sisendi ja väljundi andmetüübid on XMLSchema [20] sõne tüüpi.
- IN - Määratakse ainult sisendi andmetüübid. Väljundi andmetüübid on XMLSchema sõne tüüpi.
- OUT - Määratakse ainult väljundi andmetüübid. Sisendi andmetüübid on XMLSchema sõne tüüpi.

**FORMAT** klausliga saab määrata SOAP teenuse väljundi vormingut. Võimalikud väärtused on DNET, CONCRETE, XML ja tühi väärtus.

- DNET - Väljund on kasutatav .NET klient-aplikatsioonides ja on kooskõlas Microsoft'i poolt kasutatava System.Data.DataSet andmetüübi markeeringuga. DNET on vaikimis väärtus.
- CONCRETE - Selline väljund sobib hästi Java ja .NET klient-aplikatsioonidele.
  - CONCRETE EXPLICIT ON - WSDL dokumendis määratakse ära väljundi andmetüübid.

- CONCRETE EXPLICIT OFF -WSDL dokumendis määratakse väljundi jaoks SimpleDataset, mis ei anna kogu vajalikku informatsiooni nagu ridade arv, veerud jne.
- XML - Väljund vormindatakse XMLSchema sõne vormingusse. Selline väljund sobib SOAP-klientidele, mis ei suuda automaatselt luua WSDL põhjal liidesobjekte.
- Tühi väärtus - Sunnib teenust kasutama vaikimisi valikut. Kui teenust muutes FORMAT klausel vahele jätta, siis muudetakse teenuse väljundi vormingu tüüp vaikimisi väärtuseks.

**METHODS** klausliga saab määrata, mis HTTP meetodid on kasutatavad selle teenusega. DISH veebiteenuse puhul on valikus DEFAULT, POST, GET, HEAD, NONE ja \*. SOAP teenuse puhul on valikus DEFAULT, POST, HEAD ja NONE.

**AUTHORIZATION, ENABLE/DISABLE, SECURE, USER** klauslid ja *teenuse\_nimi* on analoogsed SQL Anywhere HTTP veebiteenustega.

## 1.12 SOAP veebiteenuse näide

Järgnevalt on toodud näide lihtsast SOAP veebiteenusest SQL Anywhere andmebaasis.

```
CREATE SERVICE SOAP_veebiteenus
  TYPE 'SOAP'
  FORMAT 'DNET'
  AUTHORIZATION OFF
  USER DBA
  AS SELECT 'Hello world!';
```

Selle koodijupiga luuakse SOAP tüüpi veebiteenus nimega *SOAP\_veebiteenus*. Teenuse väljund vormindatakse DNET tüüpi, teenus ei nõua autoriseerimist, lubatud kasutaja on DBA ja teenus väljastab SQL päringu *SELECT 'Hello world!'* tulemuse.

```
CREATE SERVICE DISH_veebiteenus
  TYPE 'DISH'
  FORMAT 'DNET'
  AUTHORIZATION OFF
  USER DBA;
```

Eelneva koodijupiga luuakse DISH tüüpi veebiteenus nimega *DISH\_veebiteenus*. Teenusega on seotud kõik andmebaasis olevad SOAP veebiteenused. *DISH\_veebiteenus* kaudu väljakutsutud SOAP teenused vormindatakse DNET tüüpi. Teenuse kasutamisel ei nõuta autoriseerimist, lubatud kasutaja on DBA ja teenus koostab kõigist andmebaasi SOAP teenustest WSDL dokumendi.

Kui eelnevalt on SQL Anywhere lokaalne server käivitatud koos -xs atribuudiga, HTTP protokolliga ja pordiga 8082, siis on DISH teenuse poolt genereeritud WSDL dokument kättesaadav aadressil localhost:8082/DISH\_veebiteenus. WSDL dokumendi sisu on väljatoodud lisa 2.

SOAP veebiteenuse testimiseks on kasutatud rakendust SoapUI [21]. Päring koostatakse DISH teenuse poolt pakutud WSDL dokumendist. Päring saadetakse DISH teenuse aadressile [http://localhost:8082/DISH\\_veebiteenus](http://localhost:8082/DISH_veebiteenus) ja saadakse päringu *SELECT 'Hello world!'* vastus, mis on DNET tüüpi vormindatud SOAP ümbrikus. Testimise käigus saadud SOAP päring ja vastus on väljatoodud lisa 3.

## 1.13 Veebiteenuste arendamise lihtsustamiseks

SQL Anywhere sisaldab mitmeid protseduure, funktsioone ja süsteemseid parameetreid, mis abistavad veebiteenuste arendamist. Need vahendid lihtsustavad veebiteenuste vastuste genereerimist kui ka päringute lugemist ja käsitlemist. Tähtsamad funktsioonid ja protseduurid, mida ka selles töös kasutatakse, on järgnevalt väljatoodud.

### HTTP\_VARIABLE

Funktsioon HTTP\_VARIABLE [22] tagastab sissetuleva HTTP päringu muutuja väärtuse. Süntaks on järgmine:

```
HTTP_VARIABLE( muutuja_nimi [ , aste [ , tunnus ] ] )
```

- *muutuja\_nimi* - HTTP muutuja nimi
- *aste* - Kui samasugust nime omavad rohkem kui üks muutujat, siis astmega saadakse kätte vastavad muutujad. Tühi väärtus tagastab esimese.
- *tunnus* - Mitme osalise päringu puhul määratleb tunnuse atribuut päise välja nime, mille väärtus tagastatakse. Näiteks *Content-Type* või *@BINARY*.



Näiteks veebiteenuste puhul saab funktsiooni HTTP\_VARIABLE abil kätte teenuse parameetrite väärtusi. Veebiteenuses, millel on määratletud URL ON piirang, saab HTTP\_VARIABLE('URL') käsuga parameetri väärtuse kätte.

### **sa\_set\_http\_header**

Protseduuriga sa\_set\_http\_header [23] saab määrata andmebaasis veebiteenuse HTTP vastuse päisele väärtusi. Süntaks on järgmine:

```
sa_set_http_header(  
    välja_nimi,  
    väärtus  
)
```

- *välja\_nimi* - HTTP päise välja nimi.
- *väärtus* - HTTP päise väljale vastav väärtus.

Näiteks *CALL sa\_set\_http\_header('Content-type', 'text/html');* määrab HTTP päises Content-type väärtuseks text/html.

### **XMLEMENT**

Funktsioon XMLEMENT [24] koostab SQL päringu sees XML struktuuriga elemendi. Süntaks on järgmine:

```
XMLEMENT( {NAME elemendi_nimi} | sõne  
    [, XMLATTRIBUTES(tunnuse_väärtus[AS tunnuse_nimi],... ) ]  
    [, elemendi_sisu,... ]  
)
```

- *elemendi\_nimi* - Andmeid ümbritseva XML elemendi nimi.
- *sõne* - Andmeid ümbritseva XML elemendi nimi sõnena.
- *XMLATTRIBUTES ( tunnuse\_väärtus [ AS tunnuse\_nimi ],... )* - Koostab XML elemendile atribuudi. Näiteks käsklus XMLATTRIBUTES('class' AS "element") koostab XML elemendi atribuudi class="element".
- *elemendi\_sisu* - XML elemendi sisu.

Näiteks päring *SELECT XMLELEMENT(NAME element, XMLATTRIBUTES('class' AS "element"), 'väärtus');* koostab XML struktuuriga sõne *<element element="class">väärtus</element>*.

## **XMLAGG**

Funktsioon XMLAGG [25] koostab järjestatava XML elementide struktuuri. Süntaks on järgmine:

*XMLAGG( xml\_väärtus [ ORDER BY järjestus\_väljend ] )*

- *xml\_väärtus* - XML väärtused, millest koostada XML puud.
- *järjestus\_väljend* - Määrab, mille järgi järjestada XML väärtusi.

Näiteks päring *SELECT XMLAGG(XMLELEMENT(NAME id, id)) FROM Tabel;* koostab tabeli iga *id* kohta elemendi *<id>id</id>* ja tagastab need ühes sõnes *<id>1</id><id>2</id>...<id>X</id>*.

## **XMLCONCAT**

Funktsioon XMLCONCAT [26] koostab XML elementide struktuuri. Süntaks on järgmine:

*XMLCONCAT( xml\_väärtus [, ... ] )*

- *xml\_väärtus* - XML väärtused, millest koostada XML puud.

Näiteks päring *SELECT XMLCONCAT(XMLELEMENT(NAME id, id), XMLELEMENT(NAME nimi, nimi)) FROM Tabel;* koostab iga tabeli rea kohta elemendid *<id>id</id><nimi>nimi</nimi>* ja tagastab need ühes sõnes *<id>1</id><nimi>nimi1</nimi><id>2</id><nimi>nimi2</nimi>...<id>X</id><nimi>nimiX</nimi>*

## Peatükk 2

# 2 Veebiteenused ja dünaamiliselt koostatud SQL päringud

### 2.1 Dünaamiliselt koostatud SQL päringud

EXECUTE IMMEDIATE [27] käsklus võimaldab käivitada koodi sees dünaamiliselt koostatud SQL lauseid. Näiteks saab protseduuri sees vastavalt sisend argumentidele koostada SQL päringu, mida siis EXECUTE IMMEDIATE käsklusega käivitada. Käskluse süntaks on järgmine:

```
EXECUTE IMMEDIATE [ käivitamis_valikud ] sql_string
```

*käivitamis\_valikud* :

```
WITH QUOTES [ ON | OFF ]  
| WITH ESCAPES { ON | OFF }  
| WITH BATCH { ON | OFF }  
| WITH RESULT SET { ON | OFF }
```

- WITH QUOTES klausel määrab, kas *sql\_string* sees topelt jutumärkidega eraldatud sõne on identifikaator või mitte. OFF puhul vaadatakse, mis on globaalse muutuja *quoted\_identifier* väärtus. Globaalne muutuja on vaikimis ON.
- WITH ESCAPES klausel määrab, kas arvestatakse kaldkriipsuga sisestatud käsklusi. Näiteks reavahetust `\n`.
- WITH BATCH lubab või ei luba BATCH SQL lauseid. BATCH SQL on rohkem kui ühe SQL lause töötlemine ühe käivitusega.
- WITH RESULT SET määrab ära, kas *sql\_lause* tagastab midagi või ei tagasta. Näiteks, kui RESULT SET ON ja *sql\_lause* on tavaline SQL päring, siis tagastatakse selle päringu vastus, vastasel juhul kutsutakse päring välja, kuid vastust ei tagastata.

Järgnevalt on väljatoodud EXECUTE IMMEDIATE käskluse kasutamise näide.

```
CREATE PROCEDURE execute_immediate()  
BEGIN  
    EXECUTE IMMEDIATE WITH RESULT SET ON 'SELECT 'Hello World!''';  
END;
```

Antud koodiga luuakse andmebaasi protseduur nimega *execute\_immediate*, mis käivitab EXECUTE IMMEDIATE käsklusega tekstikujulist SQL lauset *SELECT 'Hello World!'*. Protseduuri väljakutsumiseks peab käivitama järgmise käskluse:

```
CALL execute_immediate();
```

Käsklus tagastab protseduuri vastuse:

```
Hello World!
```

## 2.2 SQL Anywhere veebiteenused ja EXECUTE IMMEDIATE

Kui kasutada EXECUTE IMMEDIATE käsklust koos SQL Anywhere veebiteenustega, siis tekib probleem. Tavapäraselt analüüsib SQL Anywhere andmebaas eelnevalt teenuse väljundi kujunemist, kuid EXECUTE IMMEDIATE käskluse puhul võib väljund olla päringuti erinev. Kui kasutada EXECUTE IMMEDIATE käsklust iSQL rakendusega, siis analüüsib rakendus väljundi kujunemist ja suudab seda korrektselt kuvada. Probleem tekib, kui EXECUTE IMMEDIATE käsklust kasutada koos veebiteenustega. SQL Anywhere andmebaas ei ole võimeline automaatselt analüüsima EXECUTE IMMEDIATE väljakutsutatavat päringut ja seetõttu väljundit ei osata korrektselt koostada. Veebiteenus, mis kasutab EXECUTE IMMEDIATE päringut, tagastab ainult päringu esimeses veerus olevaid tulemusi.

Ülal toodud probleemi näitamiseks koostame järgmise protseduuri:

```
CREATE PROCEDURE execute_sql(p_sql LONG VARCHAR)  
BEGIN  
    EXECUTE IMMEDIATE WITH RESULT SET ON p_sql;  
END;
```

Kood loob protseduuri nimega *execute\_sql*, mis võtab sisendiks SQL päringu sõne *p\_sql* ja käivitab selle, kasutades EXECUTE IMMEDIATE käsklust.

Kutsudes välja protseduuri *execute\_sql*, kasutades iSQL rakendust, kuvatakse vastus probleemideta:

```
CALL execute_sql('SELECT 'Hello' as veerg_1, 'World' as veerg_2,
                ''!' as veerg_3');
```

Antud käsklus tagastab iSQL keskkonda vastuse:

veerg_1	veerg_2	veerg_3
Hello	World	!

**Tabel 1** Protseduuri *execute\_sql* vastus.

Nüüd koostame veebiteenus, mis kutsub välja protseduuri *execute\_sql* sama sisendiga, mis eelnevalt iSQL keskkonnas.

```
CREATE SERVICE execute_sql_veebiteenus
  TYPE 'HTML'
  AUTHORIZATION OFF
  USER DBA
AS CALL execute_sql('SELECT 'Hello' as veerg_1, 'World' as veerg_2,
                    ''!' as veerg_3');
```

Kui eelnevalt on SQL Anywhere lokaalne server käivitatud koos *-xs* atribuudiga, HTTP protokolliga ja pordiga 8082, siis on veebiteenus kättesaadav aadressil `localhost:8082/execute_sql_veebiteenus`. Selle teenuse väljakutsumisel tagastatakse järgmine vastus:

```
<html>
  <head>
    <title>
      /execute_sql_veebiteenus
    </title>
  </head>
  <body>
    <h3>
      /execute_sql_veebiteenus
    </h3>
    <table border=1>
      <tr class="header">
        <th>
          <b>
            Expression
          </b>
        </th>
      </tr>
    </table>
```

```

        <td>
            Hello
        </td>
    </tr>
</table>
<br>
<hr>
<address>
    SQLAnywhere/12.0.1.3457
</address>
</body>
</html>

```

HTML koodist on näha, et tagastati ainult esimese veeru sisu ilma veeru nimeta.

## 2.3 Süsteemne protseduur `sa_describe_query`

Veebiteenustega seoses võib ette tulla olukordi, kus oleks hea kasutada EXECUTE IMMEDIATE funktsionaalsust. Vältimaks väljundi kuvamise probleemi, siis peab päringu sisu ise analüüsima. SQL Anywhere pakub selle jaoks protseduuri `sa_describe_query` [28]. Alljärgnevalt on toodud käskluse süntaks.

```

sa_describe_query(
    päring
    [, lisa_võtmed ]
)

```

Sisend parameeter *päring* on SQL päring, mida funktsioon hakkab lahti seletama. *lisa\_võtmed* funktsionaalsust käesolevas töös ei käsitleta. Protseduuri väljundiks on päringu vastuse struktuuri lahti seletus. Tagastatakse päringu vastuse iga veeru täpne kirjeldus (näiteks veeru nimi, andmetüüp, andmetüübi suurus jne). Käesolevas töös kasutatavad protseduuri tulemused on *name* ja *domain\_name\_with\_size*.

- *name* - Uuritava SQL päringu vastuse veeru nimi.
- *domain\_name\_with\_size* - Uuritava SQL päringu vastuse veeru andmetüübi nimi, andmetüübi suurus ja täpsus.

## 2.4 HTML veebiteenuse vastuse käsitsi vormindamine

Kasutades protseduuri `sa_describe_query` võib SQL päringule vastavalt koostada HTML tabeli vormingus vastuse nagu seda teeb tavaline HTML tüüpi veebiteenus.

```

CREATE OR REPLACE FUNCTION sql_html_tabeliks(sql_lause LONG VARCHAR)
RETURNS LONG VARCHAR
BEGIN
    DECLARE html_table_pais          LONG VARCHAR;
    DECLARE xml_paring              LONG VARCHAR;
    DECLARE temp_tabeli_loomine     LONG VARCHAR;
    DECLARE temp_tabeli_veerud     LONG VARCHAR;
    DECLARE vastus                  LONG VARCHAR;

    SELECT
        XMLELEMENT(name tr, XMLATTRIBUTES('header' AS class),
            XMLAGG (XMLELEMENT(name th, XMLELEMENT(
                name b, "name"))),
            'XMLAGG(XMLELEMENT(name tr, XMLCONCAT(' ||
                list('XMLELEMENT(name td, "" || "name" || ''') ||
                ')) order by sa_maintain_order_of_derived_table)',
            'declare local temporary table temp_tabel
                (sa_maintain_order_of_derived_table int default
                autoincrement, ' || list('"' || "name" || "' '
                || "domain_name_with_size", ',') || ')',
            LIST('"' || "name" || "'", ',')
    INTO
        html_table_pais,
        xml_paring,
        temp_tabeli_loomine,
        temp_tabeli_veerud
    FROM
        sa_describe_query(sql_lause);

    EXECUTE IMMEDIATE temp_tabeli_loomine;
    EXECUTE IMMEDIATE 'insert into temp_tabel(' ||
        temp_tabeli_veerud || ') ' || sql_lause;
    EXECUTE IMMEDIATE 'select ''<table border=1>' ||
        html_table_pais || '' || ' || xml_paring || ' ||
        ''</table>'' into vastus from temp_tabel';

    RETURN vastus;
EXCEPTION
    WHEN OTHERS THEN
        RETURN '<b>Esines viga!</b> ' || ERRORMSG(SQLCODE);
END;

```

Selle näite koostamisel on aluseks võetud ettevõttes SAP Sybase töötava Eric Farreri [29] loodud kood, mis näitab EXECUTE IMMEDIATE käskluse kasutamist koos veebiteenustega [30].

Näites loodud funktsioon *sql\_html\_tabeliks* võtab sisendiks sõne SQL päringust ja tagastab HTML tabeli struktuuriga SQL päringu vastuse.

1. Esmalt täidetakse *sa\_describe\_query* funktsioonis 4 muutujat:
  - a. *html\_table\_pais* – Tabeli päise sõne. Tehakse HTML `<tr>` element, mis kirjeldab sissetuleva päringu vastuse veergude nimesid.
  - b. *xml\_paring* – Sissetuleva päringu vastuse ridade tarvis tehakse XML päring, mis on võimeline tekitama vastuse iga rea kohta HTML `<tr>` elemendi, kasutades XML abimeetodeid.
  - c. *temp\_tabeli\_loomine* – TEMPORARY TABLE [31] loomise sõne defineerimine. Loodud sõne abil saab luua ajutise tabeli, mis koosneb samadest veergudest, mis sissetuleva päringu vastuski.
  - d. *temp\_tabeli\_veerud* – Sissetuleva päringu ja ajutise tabeli veergude nimed.
2. Käivitatakse *temp\_table\_definition* ehk luuakse ajutine tabel.
3. Täidetakse ajutine tabel andmetega sissetulevast päringust.
4. Käivitatakse päring, mis teeb HTML `<table>` elemendi päisega *html\_table\_pais* ja andmetega, mis asuvad vastloodud ajutises tabelis.

### **sql\_html\_tabeliks kasutamine**

Funktsiooni *sql\_html\_tabeliks* töö demonstreerimiseks loome ühe abistava funktsiooni ja veebiteenuse.

```
CREATE OR REPLACE FUNCTION raw_html(sql_lause LONG VARCHAR)
RETURNS LONG VARCHAR
BEGIN
    CALL sa_set_http_header('Content-Type',
        'text/html; charset=UTF-8');
    RETURN '<html><body>' || sql_html_tabeliks(sql_lause) ||
        '</body></html>';
END;
```

```
CREATE SERVICE raw_html_veebiteenus
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS SELECT raw_html(:sql_lause);
```

Luuakse funktsioon *raw\_html*, mis sätib HTTP päise konteksti tüübiks *text/html*. See tagab veebibrauseri käitumise, et vastust käsitletakse kui HTML dokumenti. Funktsioon kutsub



välja funktsiooni *sql\_html\_tabeliks*, et sissetulevast päringust *sql\_lause* teha HTML tekst ja tagastada seda.

Luuakse RAW tüüpi veebiteenus nimega *raw\_html\_veebiteenus*, mis kutsub välja funktsiooni *raw\_html*. Funktsiooni sisendiks on *:sql\_lause* parameeter. Veebiteenuse väljund vormindatakse ise, sellepärast on teenuse tüübiks valitud RAW.

Kui eelnevalt on SQL Anywhere lokaalne server käivitatud koos *-xs* atribuudiga, HTTP protokolliga ja pordiga 8082, siis on veebiteenus kättesaadav aadressil *localhost:8082/raw\_html\_veebiteenus*.

Veebiteenuse *raw\_html\_veebiteenus* testimiseks kutsume selle välja koos parameetriga *sql\_lause=SELECT 'Hello' as veerg\_1, 'World' as veerg\_2, '!' as veerg\_3*. Väljakutsutav päring on järgmine:

```
http://localhost:8082/edu/raw_html_veebiteenus?sql_lause=SELECT%20'Hello'%20as%20veerg_1,%20'World'%20as%20veerg_2,%20'!'%20as%20veerg_3
```

### Päringu vastus

```
<html>
  <body>
    <table border=1>
      <tr class="header">
        <th>
          <b>
            veerg_1
          </b>
        </th>
        <th>
          <b>
            veerg_2
          </b>
        </th>
        <th>
          <b>
            veerg_3
          </b>
        </th>
      </tr>
      <tr>
        <td>
```

```

        Hello
    </td>
    <td>
        World
    </td>
    <td>
        !
    </td>
</tr>
</table>
</body>
</html>

```

HTML koodist on näha, et tagastati kõik päritavad veerud koos veeru nimedega.

## 2.5 XML Veebiteenuse vastuse käsitsi vormindamine

Analoogselt funktsioonile *sql\_html\_tabeliks*, mis vormindab sissetuleva päringu vastusest HTML tabeli, on võimalik luua funktsioon, mis vormindab sissetuleva päringu vastusest XML struktuuriga dokumendi. Selle jaoks loome *sql\_html\_tabeliks* funktsioonile sarnase funktsiooni *sql\_xml\_kehaks*.

```

CREATE OR REPLACE FUNCTION sql_xml_kehaks(sql_lause LONG VARCHAR)
RETURNS LONG VARCHAR
BEGIN
    DECLARE xml_paring          LONG VARCHAR;
    DECLARE temp_tabeli_loomine LONG VARCHAR;
    DECLARE temp_tabeli_veerud LONG VARCHAR;
    DECLARE vastus              LONG VARCHAR;

    SELECT
        'XMLAGG(XMLELEMENT(name rida, XMLCONCAT(' ||
            list('XMLELEMENT(name ' || "name" || ', '' || "name"
                || ''')) || ')) order by
            sa_maintain_order_of_derived_table)',
        'declare local temporary table temp_tabel
            (sa_maintain_order_of_derived_table int default
            autoincrement, ' || list('' || "name" || '' '
                || "domain_name_with_size", ',') || ')',
        LIST('' || "name" || '', ',')
    INTO
        xml_paring,
        temp_tabeli_loomine,
        temp_tabeli_veerud
    FROM
        sa_describe_query(sql_lause);

```

```

EXECUTE IMMEDIATE temp_tabeli_loomine;
EXECUTE IMMEDIATE 'insert into temp_tabel(' || temp_tabeli_veerud
|| ') ' || sql_lause;
EXECUTE IMMEDIATE 'select '<keha>' || ' || xml_paring || ' ||
''</keha>' into vastus from temp_tabel';

RETURN vastus;
EXCEPTION
WHEN OTHERS THEN
RETURN '<viga>Esines viga!</viga> ' || ERRORMSG(SQLCODE);
END;

```

Antud koodist on eemaldatud HTML päise loomine ja muutuja *xml\_paring* sisu on muudetud. Muutuja *xml\_paring* on nüüd võimeline XML abimeetodeid kasutades koostama iga rea kohta XML struktuuriga andmed.

### sql\_html\_tabeliks kasutamine

Funktsiooni *sql\_xml\_kehaks* töö demonstreerimiseks loome ühe abistava funktsiooni ja veebiteenus.

```

CREATE OR REPLACE FUNCTION raw_xml(sql_lause LONG VARCHAR)
RETURNS LONG VARCHAR
BEGIN
CALL sa_set_http_header('Content-Type', 'text/xml;
charset=UTF-8');
RETURN '<?xml version="1.0" encoding="UTF-8"?>' ||
sql_xml_kehaks(sql_lause);
END;

CREATE SERVICE raw_xml_veebiteenus
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS SELECT raw_xml(:sql_lause);

```

Luuakse funktsioon *raw\_xml*, mis sätib HTTP päise konteksti tüübiks *text/xml*. See tagab klient-rakenduse käitumise, et vastust käsitletakse kui XML dokumenti. Funktsioon kutsub välja funktsiooni *sql\_xml\_kehaks*, et sissetulevast päringust *sql\_lause* teha XML struktuuriga tekst ja tagastada seda.

Luuakse RAW tüüpi veebiteenus nimega *raw\_html\_veebiteenus*, mis kutsub välja funktsiooni *raw\_html*. Funktsiooni sisendiks on *:sql\_lause* parameeter. Veebiteenuse väljund vormindatakse ise, sellepärast on teenuse tüübiks valitud RAW.

Kui eelnevalt on SQL Anywhere lokaalne server käivitatud koos *-xs* atribuudiga, HTTP protokolliga ja pordiga 8082, siis on veebiteenus kättesaadav aadressil [http://localhost:8082/raw\\_xml\\_veebiteenus](http://localhost:8082/raw_xml_veebiteenus).

Veebiteenuse *raw\_xml\_veebiteenus* testimiseks kutsume selle välja koos parameetriga *sql\_lause=SELECT 'Hello' as veerg\_1, 'World' as veerg\_2, '!' as veerg\_3*. Väljakutsutav päring on järgmine:

[http://localhost:8082/edu/raw\\_xml\\_veebiteenus?sql\\_lause=SELECT%20%27Hello%27%20as%20veerg\\_1,%20%27World%27%20as%20veerg\\_2,%20%27!%27%20as%20veerg\\_3](http://localhost:8082/edu/raw_xml_veebiteenus?sql_lause=SELECT%20%27Hello%27%20as%20veerg_1,%20%27World%27%20as%20veerg_2,%20%27!%27%20as%20veerg_3)

### **Päringu vastus**

```
<?xml version="1.0" encoding="UTF-8"?>
<keha>
  <rida>
    <veerg_1>
      Hello
    </veerg_1>
    <veerg_2>
      World
    </veerg_2>
    <veerg_3>
      !
    </veerg_3>
  </rida>
</keha>
```

XML koodist on näha, et tagastati kõik päritavad veerud koos veeru nimedega.

## **2.6 SOAP veebiteenuse vastuse käsitsi vormindamine**

Analoogselt ülal toodud viisile on võimalik luua SOAP teenuste jaoks väljundi vormindamine. Sellisel juhul kaasneb teenuse tüübi muutmine RAW peale ja siis ei saa enam kasutada SQL Anywhere enda abistvaid funktsionaalsusi. Näiteks tuleb ise kontrollida teenuse päringu sisendit, teostada ise andmetüüpide kontrolle jne. SOAP teenuste puhul ei ole see lähenemine soovitatav kuna SOAP veebiteenuse puhul võiks olla väljund paika pandud, et klient-rakendus saaks sellega arvestada.

# Peatükk 3

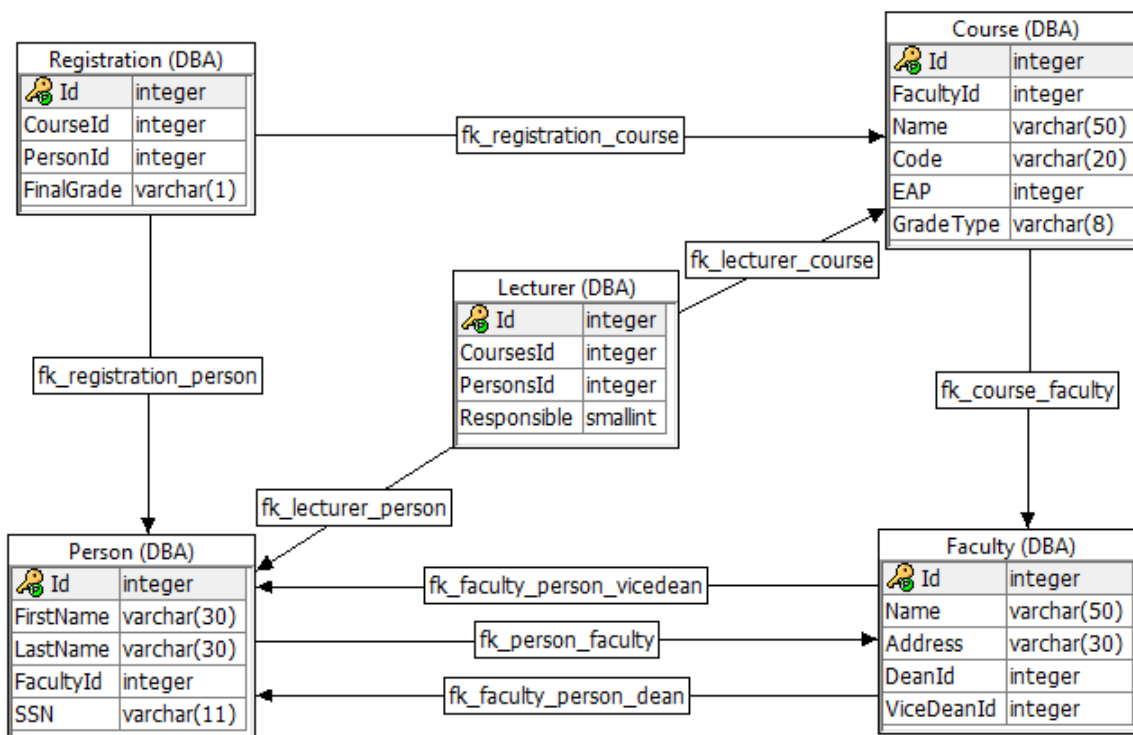
## 3 SQL Anywhere veebiteenuste õppematerjal

### 3.1 Andmebaasid (MTAT.03.264)

Tartu Ülikooli matemaatika-informaatikateaduskonna aine Andmebaasid (MTAT.03.264) on mõeldud informaatika esimese kursuse tudengitele. Aine käigus tutvustatakse tudengile SQL keele kasutamist andmebaasis. Õpetatakse päringute, vaadete, funktsioonide, protseduuride ja muude andmebaasi elementide koostamist ja kasutamist. Andmebaasi aines kasutatakse õpetamiseks SQL Anywhere tarkvarapaketti ja andmebaasi.

### 3.2 Andmemudel

Andmebaaside aines on õpetamise eesmärgil kasutusel õppeinfosüsteemi lihtsustatud andmemudel, mis koosneb viiest tabelist ja nende vahelistest ühendustest.



Joonis 1 Andmebaasid (MTAT.03.264) õppeinfosüsteemi lihtsustatud andmemudel.

### 3.3 Veebiteenused aines Andmebaasid

Andmebaaside aine käigus tutvustatakse ka SQL Anywhere andmebaasi veebiteenuseid. Veebiteenuste õpetamise eesmärk on tutvustada tudengitele mittestandardseid vahendeid andmebaasis ja andmebaaside kasutamist laiemas perspektiivis. Õppematerjal SQL Anywhere veebiteenustest on puudulik ja väiksemahuline, materjal ei anna piisavat ülevaadet SQL Anywhere veebiteenustest.

### 3.4 PmWiki

PmWiki on Patrick R. Michaudi poolt PHP keeles kirjutatud wiki põhine süsteem, mille eesmärk on koostöös veebilehtede loomine ja hooldamine. PmWiki veebilehti saab muuta, kasutades määratletuid muutmise reegleid. Wikiga töötamiseks ei ole vaja osata HTML ja CSS keelt. Lehtede muutmise võib olla avalik või piiratud kasutajaõigustega. Kasutajaõiguste haldamiseks on võimalik kasutada .htaccess, LDAP serverit või MySQL andmebaasi. PmWiki on kergesti laiendatav ja välimust saab lihtsasti muuta, kasutades erinevaid stiile. Kõige uuem stabiilne versioon PmWiki mootorist on 2.2.50. [32]

Tartu Ülikooli arvutiteaduste instituut kasutab oma kursuste lehel *courses.cs.ut.ee* wiki mootorina samuti PmWiki wiki mootorit.

### 3.5 Veebiteenuste õppematerjal

Käesoleva töö käigus valmistas autor Andmebaaside kursuse jaoks SQL Anywhere andmebaasi veebiteenuste temaatilise õppematerjal, mis on väljatoodud lisa 1. Materjal on koostatud wiki formaadis. Tagamaks õppematerjali kooskõlastuse Tartu Ülikooli arvutiteaduste instituudi kursuste wiki veebilehega on õppematerjal loodud, kasutades PmWiki määratletud reeglistikku.

## 3.6 Ülesehitus

Õppematerjali ülesehitus on õpetamiseks sobiv, tuues välja teoreetilise info, hulgaliselt näiteid ja teemakohaseid ülesandeid koos lahendustega. Materjali alguses tutvustatakse SQL Anywhere veebiteenuseid ja andmebaaside kursuse andmemudelit. Edasi seletatakse lahti veebiteenuste loomine ning teooria kinnitamiseks on koostatud kõikvõimalikke näiteid veebiteenustest. Lisaks on õppematerjalis väljatoodud veebiteenuste ja EXECUTE IMMEDIATE käskluse kasutamise probleem koos näidetega. Materjali lõpus on tudengile suunatud ülesanded ja iga ülesande kohta üks võimalik lahendus.

### **Teooria**

Õppematerjalis on näidatud, kuidas käivitada andmebaasi serverit veebiteenustele võrgusuhtlust lubava klausliga. Tuuakse välja veebiteenuste loomise süntaks ja see seletatakse lahti täpsemalt punkt-punkt haaval.

### **Näited**

Kõikide võimalike veebiteenuse tüüpide kohta on tehtud eraldi selle kasutamise näide. Iga näite juures on toodud välja antud näite loomise SQL kood, koodi seletus ja pilt veebiteenuse kasutamisest.

### **Ülesanded**

Kõik ülesanded on koostatud lihtsustatud õppeinfosüsteemi andmemudeli põhjal. Ülesanded on koostatud mitmel erineval õppematerjalis käsitletud teemal, et tudeng saaks veebiteenuste kasutamist katsetada erinevatel viisidel. Ülesandeid on 13 ning nad on erinevate raskus astmetega.

### **Lahendused**

Iga ülesande jaoks on toodud välja üks võimalik lahendus.



### **3.7 Õppematerjali kasutamine**

Andmebaaside aines õppematerjali kasutamiseks on kaks varianti.

- Valida terve wiki dokument ja seade selle ülesse [courses.cs.ut.ee](https://courses.cs.ut.ee) veebilehele.
- Õppematerjalist on võimalik valida välja vastavalt vajadusele mingid kindlad osad ja nendest koostada õppematerjal, näiteks slaidide kujul või lühema wiki lehena.

# Kokkuvõte

Töö eesmärk oli uurida SAP Sybase SQL Anywhere andmebaasi veebiteenuseid ja luua ainele Andmebaasid (MTAT.03.264) vastavasisulised õppematerjalid. Püstitatud eesmärgid lõputööga saavutati.

Töö sisaldab veebiteenuste tutvustust ja näiteid. Lisaks veebiteenuste uurimisele käsitleb lõputöö ka SQL Anywhere veebiteenuste üht suuremat probleemi, veebiteenuste probleemne koostöö dünaamiliste SQL päringutega. Probleem seletati täpsemalt lahti ning toodi välja üks võimalik lahenduse viis koos näidetega. Lõputöö käigus valmistas autor sisuka õppematerjali, mis on koostatud wiki formaadis. Materjal sisaldab teooriat, hulgaliselt näiteid, veebiteenuste ja dünaamilise SQL päringute probleemi tutvustust koos ühe võimaliku lahendusega, veebiteenustega seotud ülesandeid ja lahendusi. Õppematerjal on koostatud viisil, et õppejõul oleks võimalus kasutada seda vastavalt vajadusele.

Käesoleva töös käsitleti kõige tähtsamaid teemasid, et tutvustada veebiteenuste olemust SQL Anywhere andmebaasis. SQL Anywhere andmebaasi veebiteenustega seotuvad mitmed erinevad andmebaasi funktsionaalsused. Töö käigus ilmsid mitmed teemad, mida veebiteenustega edasi tegeledes uurida:

- XML elementide lugemine ja vormindamine
- HTTP sessioonid
- SOAP teenuste sisendi ja väljundi vormindamine
- Keerulisemate elementide koostamine SOAP teenustele
- DISH teenuse WSDL genereerimine
- Jne

Antud teemasid käsoleva töös ei käsitletud, kuid neid tasuks edaspidi uurida, kui tegeleda SQL anywhere andmebaasi veebiteenustega.

Autori arvates on SQL Anywhere andmebaasi veebiteenuste loomise funktsionaalsus hea alternatiiv traditsioonilistele liidestele nagu JDBC ja ODBC. Andmebaasi veebiteenustega saab lihtsasti luua SOAP veebiteenuseid ja lihtsamaid veebiliideseid. Olukorras, kus andmebaasil on vaja pakkuda või tarbida veebiteenuseid, on andmebaasisiseste veebiteenuste kasutamine küllaltki õigustatud - andmetele pääsetakse otse ligi ja ei ole vaja andmebaasi serverile lisada muid väliseid komponente.

# **SAP Sybase SQL Anywhere web services**

Bachelor thesis

Sander Sepp

Summary

The goal of this bachelor thesis is to research SAP Sybase SQL Anywhere web services and to put together study materials for Database course in University of Tartu. SQL Anywhere is a relational database management system with a built in database. SQL Anywhere has the functionality to create and use web services inside the database without any external help from other applications.

The thesis introduces and explains how SQL Anywhere web services work and how they are created. Creation syntax is explained thoroughly and examples are used to show how web services work.

SQL Anywhere web services have a problem working alongside with dynamic SQL query executions. This problem is shown and explained in the thesis. One possible solution to this problem is introduced. The solution is explained and shown to work with examples.

During the research the author compiled study materials for the Database course. The material is written in wiki format using PmWiki wiki engine. This ensures the compatibility with Tartu University's computer science institutions courses webpage, which also uses PmWiki for it's wiki engine. The study material consists of theory, several examples, web service and dynamic SQL problem's explanation and solution, exercises for students and solutions for the exercises. The material can be used according to need. The course lecturer could use all the material or select only some parts depending on how much the students should know.

# Kirjandus

- [1] RFP Connect News. *SAP to Acquire Sybase*, 30.05.2010  
<http://www.rfpconnect.com/news/2010/5/13/sap-to-acquire-sybase>  
viimati vaadatud 10.05.2013
- [2] SAP. *1972-1981: the early years*, 2013  
<http://www.sap.com/corporate-en/our-company/history/1972-1981.epx>  
viimati vaadatud 10.05.2013
- [3] SAP. *Fact Sheet – SAP AT A GLANCE Capital Market Information*, 04.2013  
<http://www.sap.com/corporate-en/investors/pdf/SAP-Fact-Sheet-EN.pdf>  
viimati vaadatud 10.05.2013
- [4] SAP. *Integrated Report 2012 – Key Facts*, 2012  
<http://www.sapintegratedreport.com/2012/en/key-facts/key-facts.html>  
viimati vaadatud 10.05.2013
- [5] SAP. *SAP at a glance: company information*, 2013  
<http://www.sap.com/corporate-en/our-company/index.epx>  
viimati vaadatud 10.05.2013
- [6] Sybase. *Products*, 2013  
<http://www.sybase.com/products>  
viimati vaadatud 10.05.2013
- [7] Sybase. *SAP Sybase SQL Anywhere*, 2013  
<http://www.sybase.com/products/databasemanagement/sqlanywhere>  
viimati vaadatud 10.05.2013
- [8] SAP. *SAP Gives Customers Remote Access to Data-Driven Applications to Extend the Power of Real-Time Data*, 16.04.2013  
<http://www.sap.com/news-reader/index.epx?articleID=20742>  
viimati vaadatud 10.05.2013
- [9] The World Wide Web Consortium (W3C), 2013  
<http://www.w3.org/>  
viimati vaadatud 10.05.2013
- [10] W3C. *Web Services Glossary*, 11.02.2004  
<http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>  
viimati vaadatud 10.05.2013

- [11] SAP Sybase DocCommentXchange. *SQL Anywhere as an HTTP web server*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbprogramming/pg-httpserver.html>  
viimati vaadatud 10.05.2013
- [12] SAP Sybase DocCommentXchange. *-xs dbeng12/dbsrv12 server option*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbadmin/xs-database-dbengine.html>  
viimati vaadatud 10.05.2013
- [13] SAP Sybase DocCommentXchange. *SQL Anywhere database server syntax*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbadmin/da-dbserver.html>  
viimati vaadatud 10.05.2013
- [14] SAP Sybase DocCommentXchange. *Network protocol options*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbadmin/da-protocol-options.html>  
viimati vaadatud 10.05.2013
- [15] SAP Sybase DocCommentXchange. *CREATE SERVICE statement [HTTP web service]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/create-service-statement.html>  
viimati vaadatud 10.05.2013
- [16] The Internet Society. *HTTP Authentication: Basic and Digest Access Authentication*, 06.1999  
<https://tools.ietf.org/html/rfc2617>  
viimati vaadatud 10.05.2013
- [17] W3C. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, 27.04.2007  
<http://www.w3.org/TR/soap12-part1/>  
viimati vaadatud 10.05.2013
- [18] W3C. *Web Services Description Language (WSDL) 1.1*, 15.03.2001  
<http://www.w3.org/TR/wsdl>  
viimati vaadatud 10.05.2013
- [19] SAP Sybase DocCommentXchange. *CREATE SERVICE statement [SOAP web service]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/create-service-soap-statement.html>  
viimati vaadatud 10.05.2013

- [20] W3C. *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures*, 05.04.2012  
<http://www.w3.org/TR/xmlschema11-1/>  
viimati vaadatud 10.05.2013
- [21] SoapUI  
<http://www.soapui.org/>  
viimati vaadatud 10.05.2013
- [22] SAP Sybase DocCommentXchange. *HTTP\_VARIABLE function [Web service]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/http-variable-function.html>  
viimati vaadatud 10.05.2013
- [23] SAP Sybase DocCommentXchange. *sa\_set\_http\_header system procedure*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/sa-set-http-header-system-procedure.html>  
viimati vaadatud 10.05.2013
- [24] SAP Sybase DocCommentXchange. *XMLELEMENT function [String]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/xmlelement-function.html>  
viimati vaadatud 10.05.2013
- [25] SAP Sybase DocCommentXchange. *XMLAGG function [Aggregate]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/xmlagg-function.html>  
viimati vaadatud 10.05.2013
- [26] SAP Sybase DocCommentXchange. *XMLCONCAT function [String]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/xmlconcat-function.html>  
viimati vaadatud 10.05.2013
- [27] SAP Sybase DocCommentXchange. *EXECUTE IMMEDIATE statement [SP]*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/execute-immediate-statement.html>  
viimati vaadatud 10.05.2013
- [28] SAP Sybase DocCommentXchange. *sa\_describe\_query system procedure*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/sa-describe-query-system-procedure.html>  
viimati vaadatud 10.05.2013

- [29] SAP Community Network. Eric Farrar  
<https://scn.sap.com/people/eric.farrar/>  
viimati vaadatud 10.05.2013
- [30] SAP Community Network. *Creating a Simple Browser-Based iSQL*, 08.09.2009  
<https://scn.sap.com/community/sybase-sql-anywhere/blog/2009/09/08/creating-a-simple-browser-based-isql/>  
viimati vaadatud 10.05.2013
- [31] SAP Sybase DocCommentXchange. *DECLARE LOCAL TEMPORARY TABLE statement*, 2012  
<http://dcx.sybase.com/index.html#1201/en/dbreference/declare-local-temporary-table-statement.html>  
viimati vaadatud 10.05.2013
- [32] PmWiki  
<http://www.pmwiki.org/wiki/PmWiki/PmWiki>  
viimati vaadatud 10.05.2013

# Lisad

## Lisa 1 Andmebaasid kursuse õppematerjal

Antud tööga on kaasas CD, millel on järgnev sisu:

1. WIKI\_readme.pdf – Õppematerjali kasutamise juhend.
2. Veebiteenused\_wiki.txt – Õppematerjali wiki formaadis tekst.
3. images/\*.png – Images kaust sisaldab PNG formaadis pilte, mida on wikis kasutatud.



## Lisa 2 Veebiteenuse DISH\_veebiteenus WSDL sisu

```
<?xml version="1.0" ?>
<definitions
  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" name="DISH_veebiteenus"
  targetNamespace="http://localhost:8082"
  xmlns:s3="http://localhost:8082"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
  <s:schema attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://localhost:8082">
    <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="SOAP_veebiteenus">
      <s:complexType>
        <s:sequence>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SOAP_veebiteenusResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1"

```

```

        name="SOAP_veebiteenusResult" 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: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="SOAP_veebiteenusIn">
    <part name="parameters" element="s3:SOAP_veebiteenus" />
</message>
<message name="SOAP_veebiteenusOut">
    <part name="parameters" element="s3:SOAP_veebiteenusResponse" />
</message>
<message name="FaultMessage">
    <part name="error" element="s3:error" />
</message>
<portType name="DISH_veebiteenusSoapPort" >
    <operation name="SOAP_veebiteenus">
        <input message="s3:SOAP_veebiteenusIn" />
        <output message="s3:SOAP_veebiteenusOut" />
        <fault message="s3:FaultMessage" name="error" />
    </operation>
</portType>
<binding name="DISH_veebiteenusSoapBinding"
    type="s3:DISH_veebiteenusSoapPort">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="SOAP_veebiteenus">
        <soap:operation soapAction="http://localhost:8082/SOAP_veebiteenus"
            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="DISH_veebiteenus">
    <port name="DISH_veebiteenusSoap"
        binding="s3:DISH_veebiteenusSoapBinding">
        <soap:address location="http://localhost:8082/DISH_veebiteenus" />
    </port>
</service>
</definitions>
```

## Lisa 3 Veebiteenuse SOAP\_veebiteenus päring ja vastus

### Päring

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:loc="http://localhost:8082">
  <soapenv:Header/>
  <soapenv:Body>
    <loc:SOAP_veebiteenus/>
  </soapenv:Body>
</soapenv:Envelope>
```

### Vastus

```
<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">
  <SOAP-ENV:Body>
    <tns:SOAP_veebiteenusResponse>
      <tns:SOAP_veebiteenusResult xsi:type="sqlresultstream:SqlRowSet">
        <xsd:schema id="Schema2"
          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="_x0027_Hello_x0020_world_x0021_x0027_"
        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>
            <_x0027_Hello_x0020_world_x0021_x0027_>
                Hello world!
            </_x0027_Hello_x0020_world_x0021_x0027_>
        </row>
    </rowset>
</diffgr:diffgram>
</tns:SOAP_veebiteenusResult>
<tns:sqlcode>
    0
</tns:sqlcode>
</tns:SOAP_veebiteenusResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina \_\_\_\_\_ Sander Sepp \_\_\_\_\_

(*autori nimi*)

(sünnikuupäev: \_\_\_\_\_ 25.01.1990 \_\_\_\_\_)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose  
\_\_\_\_\_ SAP Sybase SQL Anywhere veebiteenused \_\_\_\_\_,  
(*lõputöö pealkiri*)

mille juhendaja on \_\_\_\_\_ Vambola Leping \_\_\_\_\_,  
(*juhendaja nimi*)

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

Tartus, **11.05.2013**