

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse Instituut
Informaatika eriala

Sander Ott
Veebiteenuste kasutamine SQL Anywhere andmebaasis
Bakalaureusetöö (6 EAP)

Juhendaja: M. Sc. Informaatika Vambola Leping

Autor:““ juuni 2011

Juhendaja:““ juuni 2011

Lubada kaitsmisele

Professor:““ juuni 2011

TARTU 2011

Sisukord

Sissejuhatus	3
1 Sybase ja SQL Anywhere ülevaade	4
1.1 Sybase tänapäeval ja tema ajalugu.....	4
1.2 SQL Anywhere tutvustus	6
2 Veebiserver, -teenus ja nende kasutamine	12
2.1 Sisseehitatud veebiserver	12
2.1.1 Serveri käivitamine.....	12
2.1.2 Serveri sulgemine	14
2.1.3 Serveriga ühendumine	15
2.2 Veebiteenused	19
2.2.1 Veebiteenuste kirjeldus.....	19
2.2.2 Veebiteenuste vaatamine	21
2.2.3 Veebiteenuste loomine	23
3 Andmebaasimudelid ja ülesanded	26
3.1 Andmebaaside kursuses kasutatavad andmebaasi mudelid	26
3.2 Ülesanded tudengi andmebaasidele	27
4 Tudengi andmebaaside automaatne kontrollimine.....	34
Kokkuvõte	36
Web services in SQL Anywhere database.....	37
Viited	38
LISA 1	42
LISA 2	44

Sissejuhatus

Tartu Ülikoolis õpetatavate andmebaasi kursuste raames õpetatakse tudengitele põhjalikult andmebaase puudutavat teooriat. Praktikates demonstreeritakse kuidas käib reaalse andmebaasi loomine ja haldamine, kuid puuduvad näited andmebaasi kasutamisest. Antud bakalaureusetöö on loodud eesmärgiga anda tudengitele esmane kokkupuude andmebaasi kasutamisest läbi veebilehitseja.

Töös on kasutatud relatsiooniliste andmebaaside haldamissüsteemi SQL Anywhere, mida arendab firma Sybase tütarettevõtte iAnywhere. Hetkel (30.05.2011) on SQL Anywhere'ist väljas seitse versiooni: Developer, Educational, Web, OEM, Workgroup, Standard ja Advanced, millest kolm esimest on tasuta versioonid. Antud töös on kasutatud Web Editionit, kuna teda on lubatud tasuta kasutada veebipõhiste teenuste loomisel. Samuti on SQL Anywhere kasutusel kursusel Andmebaasid MTAT.03.105, kuhu antud bakalaureuse töös luuakse õppematerjalid õpetamaks tudengitele andmebaasi ja veebibrauseri vahelist suhtlust. Töös toodud näited sobivad üks ühele Andmebaaside kursusel kasutatava SQL Anywhere versiooni korral.

Bakalaureusetöö on jagatud nelja põhilisse ossa: Sybase ja SQL Anywhere tutvustus, veebiserveri, -teenuse loomine ja nende esilekutsumine, tudengitele õppematerjalide ja näiteülesannete koostamine ning õppejõududele tudengi andmebaasi automaatse kontrollimise võimaluse loomine.

Kursusel on kasutusel kaks relatsioonilist andmebaaside mudelit. Üks nendest kajastab male turniire ja teine on lihtsustatud versioon õppeinfosüsteemist. Andmaks tudengitele reaalsemat pilti andmebaaside kasutamisest tuleks hetkel eksisteerivaid õppematerjale täiendada. Andmebaas ei ole mõeldud kasutamiseks mõne üksiku indiviidi poolt. Andmebaasi eesmärk on olla masskasutuses, seega on oluline õpetada tudengitele andmebaasi ja veebibrauseri vahelist suhtlust. Kasutades SQL Anywhere'i sisseehitatud võrguserverit saavad tudengi loodud andmebaasi ühenduda kõik samas võrgus olevad kasutajad, kaasaarvatud õppejõud. Sellega tekib õppejõududel mugav võimalus, oma arvuti tagant lahkumata kontrollida tudengi loodud andmebaasi vastavust kursusel püstitatud eesmärkidele.

1 Sybase ja SQL Anywhere ülevaade

1.1 Sybase tänapäeval ja tema ajalugu

Sybase, SAP'i (Systems, Applications and Products in Data Processing) tütarettevõtte, on andmebaasi lahendusi pakkuv firma. SAP on üks suurimaid tarkvaraarenduse firmasid, tal on üle 100 tütarettevõtte, üle 53000 töötaja ja üle 109 000 kliendi ning 2010. aasta käive oli 12.4 miljardit eurot [1]. Sybase on üks SAP'i suurimaid sõsarfirmasid, kellel on üle 4000 töötaja 60 erinevas riigis, sealhulgas ka Eestis. Kliente on Sybasel rohkem kui 40000, kusjuures 91 Fortune 100 firmast kasutab Sybase'i teenuseid. Fortune 100 on ajakirja Fortune poolt koostatav edetabel parimatest USA ettevõtetest. Patente on Sybase saanud 148 ja 185 patenti ootab kinnitamist [2]. Sybase on jaganud enda poolt pakutavad ja arendatavad tooted kaheksasse kategooriasse, milleks on:

1. andmebaasi haldamine (Database management),
2. ettevõtte järjepidevus (Business Continuity),
3. andmete ladustamine (Data Warehousing),
4. modelleerimine ja arendus (Modeling & Development),
5. mobiilsed teenused (Mobile Services),
6. mobiilsed ettevõtted (Mobile Enterprise),
7. äri strateegia ja analüüs (Business Intelligence & Analytics),
8. finantsteenused (Financial Service Solutions).

Nendesse kategooriatesse kuulub 36 toodet, mis omakorda jagunevad veel erinevateks osadeks [3]. Sybase on loonud partnersidemed mitmete juhtivate tehnoloogiafirmadega parandamaks toodete kvaliteeti ja konkurentsivõimet. Koostööd tehakse näiteks selliste firmadega nagu Bradmark, Hewlett Pacard, IBM, Informatica, Intel ja Red Hat [4].

Sybase alguseks võib lugeda aastat 1984, mil Mark Hoffman, Bob Epstein, Jane Doughty ja Tom Haggin asutasid firma nimega Systemware. Eesmärk oli luua relatsiooniliste andmebaaside haldamise süsteem (RDBMS), mis korrastaks informatsiooni ja muudaks selle kättesaadavaks paljudele võrgus olevatele arvutitele. 1986. aasta viimasel veerandil valmisid esimesed testprogrammid ja järgmise aasta mais avaldati ametlikult Sybase System, esimene suure jõudlusega (*high-performance*) RDBMS mis oli mõeldud onlain rakendustele. Sybase oli esimene, kes tuli turule klient-server relatsioonilise andmebaasiga. Samal ajal sõlmiti koostööleping Microsoftiga, et too pordiks Sybase SQL Serveri

Windows ja OS/2 operatsioonisüsteemile. Firmadevaheline koostöö sujus ja kiirelt hakati välja andma uuendusi. Kuni SQL Server 4.9'ni olid Sybase ja Microsoft SQL Serverid peaaegu identsed. Veel samal aastal tekkisid aga firmade vahel erimeelsused kasumi jagamisel ja otsustati koostöö lõpetada. 1989 valmisid Sybase'il uued tooted, näiteks Sybase Open Client/Server Interface, mis pakkus loomulikku suhtlust klient- ja serveriarvuti vahel, lubades rohkematel võrgu arvutitel andmebaasi ühenduda. Sellel aastal saavutas Sybase 56 miljoni dollari suuruse käive, seda suuresti tänu uute toodete edule. Kaks aastat hiljem, 1991 märgiti Sybase'i aktsiad esmakordselt avalikul börsil. Järgmise aasta juunis teatas Sybase, et hakkab looma raamistikku, mis lubaks firmadel üle minna vanadelt keskse serveriga süsteemilt klient-server süsteemile. 1993 aasta aprillis tutvustas Sybase esimest selle seeria uuendust, programmi nimega OmniSQL Gateway. Programm ühendas omavahel arvutite võrgu erinevad osad, mis lubas igale kasutajatele edastada ükskõik millises võrgu osas asuva teise kasuaja poolt süsteemi tehtud muudatusi. Sama aasta lõpuks olid kõik uuendused valmis, nende seas SQL Server 10, Back-up Server, Open Client/Server API, SQL Monitor ja SA Companion. 1995 ostis Sybases ära Powersoft'i [5], mis oli juhtiv arendusvahendite valmistaja klient-server süsteemidele, kellele kuulus ligi 40% sellest turust. Selle tehinguga sai Sybase endale Powersoft'i peamise toote, PowerBuilder'i. Tehing pani aluse ka Sybase'i siirdumise mobiilsete andmebaaside turule, kuna omandati õigused ka Watcom SQL'ile, mille uueks nimeks sai SQL Anywhere. SQL Anywhere'ist sai Sybase'i relatsiooniliste andmebaaside haldamise süsteemide lipulaev. 2000 aasta mais eraldati mobiilsete andmebaasidega tegelev osakond iAnywhere Sybase'st ja temast saab Sybase'i tütarfirma. iAnywhere'i saadab edu ja ta võidab mitmeid auhindu, näiteks 2002 mais võideti Mobility Award parima andmebaasi kategoorias. Veebruaris 2003 ostab Sybase ära AvantGo tugevdades veelgi oma positsiooni mobiilse vahevara turul. 2005 aastal ostab Sybase ära veel kolm firmat ISDD, Avaki ja Extended Systems'i suurendades sellega veel oma edu mobiilsete rakenduste pakkumises. 2006 ostis Sybase üles Mobile 365'e, mille nimi muudeti Sybase 365'ks [6], selle käiguga sai Sybase suurimaks mobiilsete sõnumite edastajaks. 2006 aastal edastas Sybase 365 rekordilised 25 miljardit sõnumit. Samal ajal on Sybase suutnud hoida tugevat positsiooni ka andmete haldamises. Sybase'i teenuseid kasutavad paljud finants-, telekommunikatsiooni-, tehnoloogia- ja valitsuse ettevõtted. Sybase teeb koostööd mitmete oma ala liidritega infrastruktuuri, andmete hoiustamise ja esitluse vallas, optimeerimaks oma tooteid. Tegeletakse ka virtuaalsete pilvedega, mis pakuvad kasutajatele andmete paremat kättesaamist ja paindlikkust. 2007 aastal kasvas Sybase'i väärtus üle miljardi

dollari. Järgmise aasta mais tegi Sybase IQ analüütiline server uue Guinness maailma rekordi, käideldes maailma suurimat andmeladu. Samal aastal tõi Sybase turule analüütilise platvormi RAP, mis oli mõeldud Wall Street'ile, lisaks anti välja veel Sybase Unwired Platform, mis oli mõeldud heterogeenses keskkonnas kasutatavate mobiilsete rakenduste arendamiseks. Sama aasta septembris laienes Sybase 365'le ka multimeedia sõnumite edastus. 21 jaanuar 2009 ostis Sybase ära Paybox Solutions AG ning nüüd suudab Sybase pakkuda kõiki mobiilseid maksevõimalusi [7]. Märtsis sõlmivad Sybase ja SAP koostöölepingu arendamiseks uut SAP Business Suit tarkvara iOS'le, Windows Mobile'le, Android'le ja veel mõnele mobiilsele operatsioonisüsteemile. Septembris sõlmitakse koostööleping Verizon'iga, et hallata ettevõtete mobiilseid lahendusi rahvusvaheliselt, selleks hakati arendama Verizon Mobility Solutions'it, mis kasutab Sybase'i platvormi. Mais 2010 teatas SAP, et ostab Sybase'i 5,8 miljardi dollari eest [8]. Sama aasta juunis sai Sybase'st SAP'i iseseisev tütarettevõtte [9].

1.2 SQL Anywhere tutvustus

SQL Anywhere'i algseks nimeks võib pidada Watcom SQL'i, mis loodi 1992 Watcom'i poolt. Nimi muutus SQL Anywhere Studio'ks kui Watcom ühines Powersoft'iga 1993. Sellest koostööst lisandusid näiteks trigerid ja salvestatud protseduurid. 1995 ostis Sybase Powersoft'i, samal aastal anti välja ka uus versioon, mille tähtsamad uuendused olid graafiline administreerimise võimalus ja andmete replikatsioon läbi SQL Remote'i. 1998 aasta juunis anti välja kuues versioon, millega muutus uuesti ka nimi, milleks nüüd sai Adaptive Server Anywhere [10]. Sellele anti välja veel kaks parandusversiooni 6.0.2, mis ilmus märtsis 1999 [11] ja 6.0.3, mis ilmus juunis 1999 [12]. Adaptive Server Anywhere'iga lisandusid näiteks Java objektid, MobLink andmete sünkronisatsioon ja UltraLite mobiilsed andmebaasid Palm ja Windows CE operatsioonisüsteemidele. (Täpsemat infot uuenduste kohta saab lisast 1.) 2000 aasta märtsis anti välja seitsmes versioon [13], seitsmendale versioonile tehti veel 3 parandust - 7.0.2, mis ilmus 2000 aasta novembris [14], 7.0.3, mis ilmus aprill 2001 [15] ja 7.0.4, mis ilmus märtsis 2002 [16]. Seitsmes seeria lisas näiteks dünaamilise vahemälu, käskude ajastamise (*task scheduling*), sündmuse halduse (*event handling*) ja mitme platvormi vahelise administreerimise. Kaheksas versioon lasti välja 2001 aasta novembris [17]. See versioon sai ka ühe paranduse järgmise aasta oktoobris, kui avalikustati versioon 8.02 [18]. Nende uuendustega lisandus päringu optimeerimine ja krüpteeritud andmete hoiustamine ja transport ning veel mõningaid uuendusi nagu ajatsoonide täiustamine, et lihtsustada

rahvusvaheliste andmebaaside haldamist. Versioon üheksa tuli välja 2003 ja sellele versioonile tehti kaks uuendust 9.0.2 ja 9.0.3 nende uuendustega esitleti sisseehitatud HTTP serverit, XML tuge ja palju muud [19]. 2006 aastal tuli välja kümnes versioon [20] ja sellega muudeti taas nime, milleks nüüd sai SQL Anywhere, lisandusid näiteks kõrge kättesaadavus (*high availability*), mitme tuumalise protsessori tugi ja palju muud [21]. 2007 märtsis tehti ka parandus versioon 10.0.1. Kaks aastat hiljem esitleti üheteistkümnendat versiooni, samal aastal anti välja ka versioon 11.0.1. Nendega lisandus näiteks tervikteksti otsing (*full text search*) ja BlackBerry tugi [22]. 2010 avalikustati versioon 12.0.0 ja aasta hiljem 12.0.1, mis on 30.5.2011 seisuga viimane avalikustatud versioon. Kaheteistkümnenda versiooniga lisandus näiteks *read-only scale-out* tugi [23].

Tänapäevaks on SQL Anywhere'ist saanud terviklik pakett-lahendus, mis pakub andmehalduse, sünkroniseerimise ja andmete vahetamise tehnoloogiaid. Võimaldades andmebaasi rakenduste kiiret arendamist ja rakendamist isegi isoleeritud või mobiilsust nõudvates keskkondades [24]. Lisaks SQL Anywhere'ile kuulub SQL Anywhere paketti ka MobLink, Qanywhere, UltraLite ja Relay Server. SQL Anywhere omab veebirakenduste loomisel paljude programmeerimiskeelte tuge nagu näiteks PHP, Python, Ruby on Rails, AJAX, Java, ASP.NET, Perl, Flex ja Silverlight. Trigereid ja protseduure saab kirjutada nii SQL'is kui ka PHP's, Java's, .NET'is ja Perl'is. SQL Anywhere toetab ka XML'i, veebiteenuseid ja tervikteksti (*full-text*) otsingut [25]. Seda arendab Sybase iAnywhere, mis on Sybase'i tütarettevõtte. SQL Anywhere töötab väga paljudel erinevatel platvormidel [26].

1. Windows x86, x64 ja Itanium
2. Windows mobile
3. Novell NetWare
4. Linux x86, x64 ja Itanium
5. Sun Solaris SPARC ja x86
6. MAC OS X on Intel
7. IBM AIX
8. HP-UX PA-RISC ja Itanium

SQL Anywhere'ist on hetkel (04.05.2011) väljas 7 erinevat versiooni:

1. Developer
2. Educational

3. Web
4. OEM
5. Workgroup
6. Standard
7. Advanced

Nendest esimesed kolm on tasuta versioonid. Üleminek ühelt versioonilt teisele on tehtud väga lihtsaks, selleks tuleb tellida vaid uue versiooni registreerimisvõti ja see sisestada olemasolevasse versiooni ning sellega ongi versiooniuuendus läbiviidud. Iga versiooniga tuleb kaasa kogu SQL Anywhere'i funktsionaalsus, välja arvatud erinevused, mis on välja toodud allolevas tabelis.

Tabel 1. Erinevused erinevate SQL Anywhere versioonide vahel.

	Developer	Educational	Web	OEM	Workgroup	Standard	Advanced
Üldised omadused							
Maksimaalne protsessorisrite arv	Piiranguteta	Piiranguteta	Piiranguteta	Piiranguteta	2	4	Piiranguteta
Toetatavad operatsiooni süsteemid	Windows Linux Mac OS x Solaris HP-UX AIX	Windows Linux Mac OS x	Windows Linux Mac OS x	Windows Linux Mac OS x Solaris HP-UX AIX	Windows Linux Mac OS x	Windows Linux Mac OS x Solaris HP-UX AIX	Windows Linux Mac OS x Solaris HP-UX AIX
Kasutus piirangud	Ainult arendamiseks ja testimiseks	Ainult akadeemiliseks kasutamiseks	Ainult veebi- brauseri põhiste rakenduste loomiseks	Ainult spetsiifilise tarkvaraga kasutamiseks	Piiranguteta	Piiranguteta	Piiranguteta
Omadused							
High Availability	Olemas	Olemas	Juurde ostetav	Juurde ostetav	Olemas	Olemas	Olemas
Monitoring Tool (Develop-	Olemas	Olemas	Olemas	Olemas	Olemas	Olemas	Olemas

ment And testing)							
Monitoring Tool (Deploy- ment)	Puudub	Puudub	Puudub	Juurde ostetav	Puudub	Olemas	Olemas
Read-only scale-out	Olemas	Olemas	Olemas	Juurde ostetav	Puudub	Puudub	Olemas
In-Memory Runtime	Olemas	Olemas	Puudub	Juurde ostetav	Puudub	Puudub	Olemas
RSA Security Option	Olemas	Olemas	Olemas	Olemas	Olemas	Olemas	Olemas
FIPS Security Option	Puudub	Puudub	Juurde ostetav	Juurde ostetav	Juurde ostetav	Juurde ostetav	Olemas
ECC Security Option	Puudub	Puudub	Juurde ostetav	Juurde ostetav	Juurde ostetav	Juurde ostetav	Juurde ostetav

Omaduste seletused

High Availability on süsteem, mis üritab minimeerida andmebaasi maas oleku aega, selleks kasutatakse kahte või kolme andmebaasi serverit, mis töötavad erinevatel arvutitel ning teevad koostööd hoidmaks andmebaasi ja muutuste (*transaction*) logide koopiaid. Süsteemi komponentideks on peamine server, peegeldatud server ja eemalseisev server. Peamine ja peegeldatud server hoiavad koopiaid andmebaasist ja muutuste logist, eemalseisvat serverit kasutatakse otsustamisel, kumb serveritest on peamine. Süsteem toimib järgnevalt: kliendid on ühendatud peamisesse serverisse, kõik muutused andmebaasis salvestatakse peamises serveris ja kajastatakse muutuste logis, mis saadetakse peegeldatud serverisse, logide järgi viiakse samad muutused läbi ka peegeldatud andmebaasis. Kui peamine server peaks minema maha kas siis tark- või riistvara vea tõttu katkeb ühendus ka klientidega. Seejärel saadab peegeldatud server kõrvalseisvale serverile signaali, millega teatab soovist hakata peamiseks serveriks. Kõrvalseisev server kontrollib kas peegeldatud serveri andmebaas on peamise serveriga sünkroniseeritud, kui on, saab peegeldatud serverist peamine server. Kliendid, kelle ühendus andmebaasiga eelnevalt

katkestati, peavad vaid proovima uuesti andmebaasi ühendada ja nende jaoks ei ole midagi muutunud. Maas oleva serveri töökorda saamisel võtab ta endale peegeldatud serveri staatuse [27].

Monitoring Tool on brauseri põhine administreerimise vahend, mis pakub informatsiooni andmebaasi seisundi ja kättesaadavuse kohta. Andmeid kogutakse pidevalt, isegi siis, kui keegi ei ole Monitor'i sisse loginud. Vahend võimaldab saata e-maile, kui eelnevalt määratud normid on ületatud. Kuna Monitor on jälgitav läbi veebibrauseri, saab seda jälgida igalt poolt, kus on internetiühendus. Andmete monitooring nõuab väga vähe arvutusressurssi. Monitor'i kasutamiseks on vaja SQL Anywhere 11.0.1 või uuemat versiooni, veebibrauseris peab olema lubatud JavaScript ja Adobe Flash Player [28].

Read-only scale-out on konfiguratsioon, mis võimaldab mahalaadida (*Offload*) operatsioone, mis vajavad ainult lugemist (*read-only*) juurdepääsu andmebaasile. See süsteem töötab puu kujuliselt, mille tipp on ainuke andmebaasiserver, mis lubab nii lugemise kui kirjutamise käsked, kõik teised puu tipud on *read-only* seisundis. Kui juhttipule ühendada külge mõni andmebaasi server, siis juhtserver saadab talle oma muutuste logi, kui alamserver saab logi kätte, viib ta need muutused läbi enda andmebaasis ja saadab logi edasi oma alamtippudele, kui tal on neid. Tipp saab aru kui mõni tema alamtipp muutub kättesaamatuks ja peatab sinna tippu logide saatmise. Alamtipu taas kättesaadavaks muutumisel saadab ta oma ülemusele teate, milles teatab soovist taas saada muutuste logisid. Juhttipu mahaminemisel jäävad alamtipud endiselt kättesaadavaks, aga nad ei saa enam muutuste logisid. Kui juhttipp tuleb taas üles, siis alamtipud taastavad temaga ühenduse ja kõik toimib edasi nagu varem. Read-only scale-out'i on soovitatav kasutada koos High Availability süsteemiga, et parandada andmebaasi kättesaadavust [29].

In-Memory Runtime on süsteem, mis võimaldab anda juurde lisa jõudlust, kuid väheneb vastupidavust. See funktsioon on kasulik rakenduste puhul, kus kiirus on oluline ja süsteemil on palju vaba mälu. Kõik andmebaasi failid võiksid ära mahtuda vahemällu. Valida saab kahe erineva In-Memory tüübi vahel, kas *never-write* või *checkpoint-only*. *Never-write* tüübi puhul ei kirjutata muutusi andmebaasi faili. Kui kaob ühendus andmebaasiga, siis kaovad ka kõik muutused, mis selle sessiooni ajal tehti. *Checkpoint-only* tüübi puhul ei koostata muutuste logisid, kuid koostatakse *checkpoint* logisi. Sellisel viisil ei saa andmebaasi taastada kõige viimase muutuse koha pealt aga andmebaasi

seisund on võimaik taastada viimase *checkpoint*'i koha pealt. Järgnevas tabelis on erinevused *never-write*, *checkpoint only* ja klassikalise tüübi vahel [30].

Tabel 2. Erinevused In-Memory tüüpide ja klassikalise tüübi vahel.

	Never-write	Checkpoint-only	klassikaline
Muutused	Jah	Jah	Jah
.db faili uuendused	Ei	Jah	Jah
Muutuste logi	Ei	Ei	Jah
Ajutised failid	Ei	Ei	Jah
Checkpoint logi	Ei	Jah	Jah
Checkpoints	Ei	Jah	Jah
Dirty page flushes	Ei	Jah(1)	Jah
Piiramatu kasv	Ei	Jah(2)	Jah
Taastatav	Ei	Jah(3)	jah

(1)- *Dirty page flush* tehakse ainult *checkpoint*'i tehes.

(2)- Ilma ajutiste failideta on *checkpoint* ainuke, mis piirab vahemälu piiride ületamist

(3)- taastatakse viimasest *checkpointist*, mitte viimasest muutusest [31].

RSA Security Option on USA firma RSA Data Security poolt 1977. aastal välja töötatud avaliku võtmega krüpteerimismeetod, mis võimaldab nii andmete krüpteerimist kui autentimist. RSA algoritm põhineb faktil, et pole olemas efektiivset meetodit väga suurte arvude algteguriteks lahutamiseks. Seetõttu nõuab RSA võtme lahtimuukimine tohutult arvutusvõimsust ja aega [32].

FIPS Security Option – Federal Information Processing Standard loodi, et kooskõlastada krüptograafiliste meetodite nõudmised ja standardid, seda nii riist- kui tarkvara komponentide puhul. SQL Anywhere võimaldab kasutada FIPS sertifikaadiga turvalisuse algoritme, et krüpteerida andmebaasi faile ning kliendi ja serveri vahelist liiklust [33].

ECC Security Option on elliptiliste kõverate krüptosüsteem, alternatiiv RSA krüpteerimissüsteemile, sest elliptiliste funktsioonide diskreetsete logaritmid arvutamine on keerulisem kui suurte arvude algteguriteks lahutamine [34].

2 Veebiserver, -teenus ja nende kasutamine

2.1 Sisseehitatud veebiserver

2.1.1 Serveri käivitamine

Soovides läbi veebilehitseja esile kutsuda andmebaasis asuvaid veebiteenuseid peab see andmebaas töötama veebiserveris. Vastasel juhul ei tea server millisesse porti edastada http päringuid. SQL Anywhere'i on sisse ehitatud http server ja see käivitub automaatselt kui andmebaasi serveri käivitamisel käsurealt lisada juurde `-xs` käsk. Selline serveri käivitamine võimaldab konfigureerida võrgu protokollide seadeid nagu serveri port, logimine, *time-out* tingimused, maksimaalne päringu suurus jpm. Näiteks Windowsi operatsioonisüsteemis SQL Anywhere'iga kaasas olevat demo andmebaasi saab käivitada järgneva käsuga: `dbeng12 -xs http(port=8082) "%SQLANYXSAMP12%\demo.db"`. Selles käsus on kasutatud vaid ühte seadistamise valikut „`http(port=8082)`“, mis määrab, et http päringuid kuulatakse portis 8082. süntaks veebiserveri alustamiseks on [35]:

```
{ dbeng12 | dbsrv12 } -xs { protocol[,...] } ...
protocol : {
    NONE
    | HTTP [ ( option=value;... ) ]
    | HTTPS [ ( option=value;... ) ]
```

Dbeng12 määrab, et tegemist on personaalse serveriga ja sellele serverile saab ligi ainult sellest samast arvutist. Kasutades `dbsrv12` käsku luuakse võrguserver, ehk sellesse serverisse saavad end ühendada kõik võrgus olevad arvutid. järgnevas tabelis on SQL Anywhere'i poolt toetatavate protokollide seadistamise võimalused. [36]

Tabel 3. Võimalikud protokollide suvandite (*option*) valikud.

TCP/IP	HTTP	HTTPS	TSL
Broadcast (BCAST) protocol option	DatabaseName (DBN) protocol option	certificate_company protocol option	certificate_company protocol option
BroadcastListener (BLISTENER) protocol option	KeepaliveTimeout (KTO) protocol option	certificate_name protocol option	certificate_name protocol option
ClientPort (CPORT) protocol option	LocalOnly (LOCAL) protocol option	certificate_unit protocol option	certificate_unit protocol option

DoBroadcast (DOBROAD) protocol option	LogFile (LOG) protocol option	DatabaseName (DBN) protocol option	FIPS protocol option
Host (IP) protocol option	LogFormat protocol (LF) option	Identity protocol option	LocalOnly (LOCAL) protocol option
LDAP protocol option	LogMaxSize (LSIZE) protocol option	Identity_Password protocol option	trusted_certificates protocol option
LocalOnly (LOCAL) protocol option	LogOptions (LOPT) protocol option	FIPS protocol option	
MyIP (ME) protocol option	MaxConnections (MAXCONN) protocol option	LocalOnly (LOCAL) protocol option	
ReceiveBufferSize (RCVBUFSZ) protocol option	MaxRequestSize (MAXSIZE) protocol option	LogFile (LOG) protocol option	
SendBufferSize (SNDBUFSZ) protocol option	MyIP (ME) protocol option	LogFormat protocol (LF) option	
ServerPort (PORT) protocol option	ServerPort (PORT) protocol option	LogMaxSize (LSIZE) protocol option	
TDS protocol option	Timeout (TO) protocol option	LogOptions (LOPT) protocol option	
Timeout (TO) protocol option		MaxConnections (MAXCONN) protocol option	
VerifyServerName (VERIFY) protocol option		MaxRequestSize (MAXSIZE) protocol option	
		MyIP (ME) protocol option	
		ServerPort (PORT) protocol option	
		Timeout (TO) protocol option	

		trusted_certificates protocol option	
--	--	---	--

Tõeväärtus tüüpi suvandeid saab sisse lülitada kasutades väärtusi YES, Y, ON, TRUE, T või 1 ja suvandi väljalülitamiseks saab kasutada NO, N, OFF, FALSE, F või 0, kusjuures tõusutundlikus on oluline [37]. Soovides konfigurereida oma serverit enama kui ühe parameetriga, peaksid parameetrid olema eraldatud semikooloniga.

```
(-xs "HTTP(OPTION1=value1;OPTION2=value2;...)" )
```

näiteks:

```
dbsrv12 -xs https(PORT=544;FIPS=YES;
  Identity=certificate.id;Identity_Password=password) your-database-
  name.db
```

Selle käsuga käivitatakse võrguserver, mis kasutab HTTPS protokollit, kuulab porti 544, lubab kasutada FIPS turvalisusega algoritme, määrab faili (certificate.id), kus asuvad avalik sertifikaat ja selle salajane võti ning kontrollib salajase võtme vastavust salasõnale *password*. [38]

SQL Anywhere lubab alustada ka rohkem kui ühte HTTP veebiserverit. Selleks tuleb anda serveritele erinevad pordid. Kahte või enam serverit saab käivitada ka samaaegselt, selleks tuleb käsurealt `-xs` käsuga kaasa anda mõlema serveri konfiguratsioon, näiteks:

```
dbsrv12 -xs
  http(port=80;dbn=your-first-database)
  http(port=8800;dbn=your-second-database)
  your-first-database.db your-second-database.db
```

sellise käsuga käivitatakse 2 serverit, esimene server kuulab porti 80 ja kannab nime *your-first-database*, ning see server kasutab andmebaasi nimega *your-first-database.db*. Teine server kuulab porti 8800, kannab nime *your-second-database* ja kasutab andmebaasi nimega *your-second-database.db*. [39]

2.1.2 Serveri sulgemine

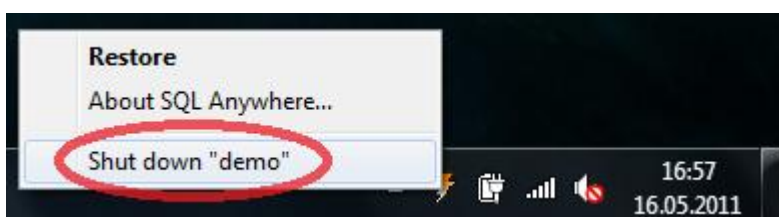
Serverit saab sulgeda nii graafilises keskkonnas kui ka käsurealt. Käsurealt andmebaasi serveri sulgemiseks tuleb sisestada `dbstop` käsk. Käsu täpne süntaks on:

`dbstop [options] [server-name]` kus võimalikud suvandid on esitatud järgnevas tabelis:

Tabel 4. Käsu `dbstop` suvandite loend ja nende kirjeldus.

Suvand	Kirjeldus
@data	Loeb sisse suvandid konfiguratsiooni failist või keskkonna muutujast.
-c „võtmesõna= väärtus;..“	Määrab ühenduse sõne, võrguserveri sulgemisel peab ühenduse sõne sisaldama sellise kasutaja ID'd, kellel on õigus andmebaasi sulgeda. Privaatserverit saavad sulgeda kõik kasutajad, kui seda ei ole muudetud <code>-gk</code> suvandiga.
-d	Ei peata serverit vaid peatab andmebaasi, mis oli määratud ühenduse sõnes.
-o <i>filename</i>	Kirjutab väljumise sõnumi määratud faili.
-q	Käivitab vaikse seade ehk sõnumeid ei kuvata.
-x	Ei peata serverit, kui sellel on veel aktiivseid ühendusi
-y	Peatab serveri isegi siis kui sellel on aktiivseid kasutajaid
Server-name	Määrab serveri nime. Andmebaas peab olema käivitatud nii, et selle sulgemiseks ei ole mingeid piiranguid.

Näiteks, kui töötab server nimega `myserver`, kuhu on ühendatud `demo.db` andmebaas, siis selle serveri peatamiseks võib sisestada näiteks järgneva käsu: `dbstop -c "UID=DBA;PWD=sql;Server=myserver" [40]`. `Dbstop` käsku saab kasutada vaid Windowsi operatsioonisüsteemi kasutades. Graafilises keskkonnas saab serverit sulgeda, vajutades hiire parema klahviga tegumireal asuval serveri ikoonil, ning seejärel valida *shut down „serveri nimi“*



Joonis 1. veebiserveri sulgemine Windowsi tegumirealt.

2.1.3 Serveriga ühendumine

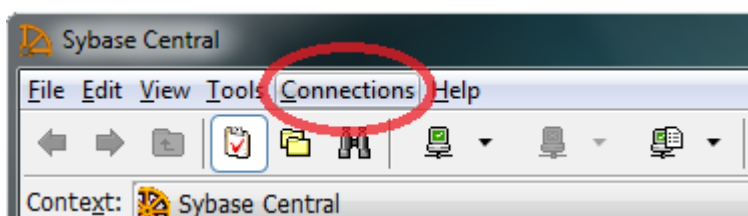
Olles loonud http toega andmebaasi serveri tuleb sellesse ühenduda, selleks saab kasutada käsuriida, Sybase Centralit või mõnda kolmanda tootja poolt valmistatud programmi. Käsurealt interaktiivse `sql`'i käivitamiseks tuleb sisestada `dbisql` käsk. Käsu süntaks on järgnev: `dbisql -c "connection-string" [options] [dbisql-command | command-file]`. Options seade võimalikud sisendid on toodud allolevas tabelis.

Tabel 5. Käsu `dbisql` suvandite loend ja nende kirjeldus.

Suvand	Kirjeldus
@data	Loeb suvandeid mõnest muutujast või konfiguratsiooni failist
-c „võtmesõna=väärtus;...”	Määrab ühenduse parameetrid. Kui nende parameetrite alusel ei ole võimalik ühendust luua, siis kuvatakse aken kus saab uuesti parameetreid sisestada. Kui sisestamata jäävad mõlemad nii kasutajanimi kui ka salasõna, siis eeldatakse, et need on vastavalt DBA ja sql.
-d <i>eraldaja</i>	Määrab käsu eraldaja.
-dl	Kuvab kõik kasutaja poolt sisestatud käsud. Seda suvandit on võimalik kasutada vaid siis, kui interaktiivne SQL jookseb käsureal.
-datasource <i>DSN-name</i>	Määrab ODBC andmefaili, kuhu ühenduda
-f <i>filename</i>	Avab interaktiivses SQL'is faili nimega <i>filename</i> aga ei käivita seda.
-host <i>hostname</i>	Määrab andmebaasi käitleva serverarvutiniime või -ip aadressi.
-nogui	Käivitab interaktiivse sql'i kui käsurea rakenduse, ilma eraldi aknata, see on kasulik <i>batch</i> 'ide loomisel
-onerror{ continue exit}	Määrab, mis tehakse kui sql käsu sisestamisel on tehtud viga.
-port <i>port-number</i>	Määrab pordi numbrit mida SQL Anywhere andmebaas kuulab.
-q	vaigistab väljund sõnumid, väljaarvatud error sõnumid
-ul	Määrab, standardseks andmebaasiks UltraLite andmebaasi
-version	Kuvab interaktiivse SQL'i versiooni numbrit
-x	Vaatab käske aga ei sisesta neid, see on kasulik süntaksi vigade otsimisel
Dbisql-command command-file	

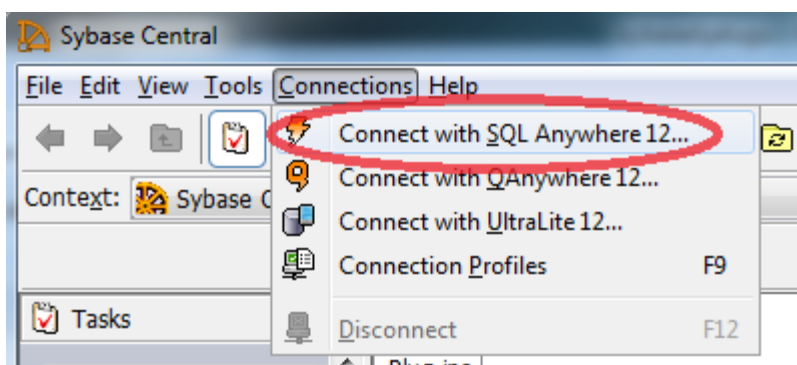
SQL Anywhere'iga kaasasoleva demo andmebaasi interaktiivset sql'i saab avada käsurealt näiteks käsuga:

`dbisql -c "dbf=%SQLANYSAMP12%\demo.db;uid=DBA;pwd=sql"` [41]. Kasutades Sybase Central'it saab andmebaasi ühenduda, kui Sybase Centrali menüüst valida *Connections*.



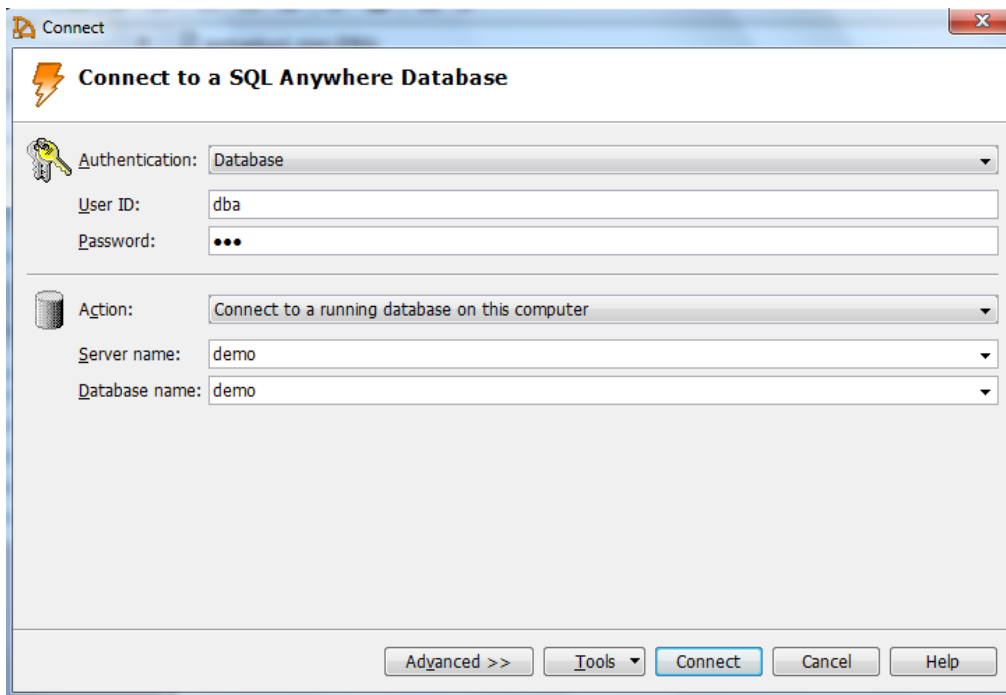
Joonis 2. Sybase Centralis veebiserveriga ühenduse loomine.

Sealt *connect with SQL Anywhere 12*.



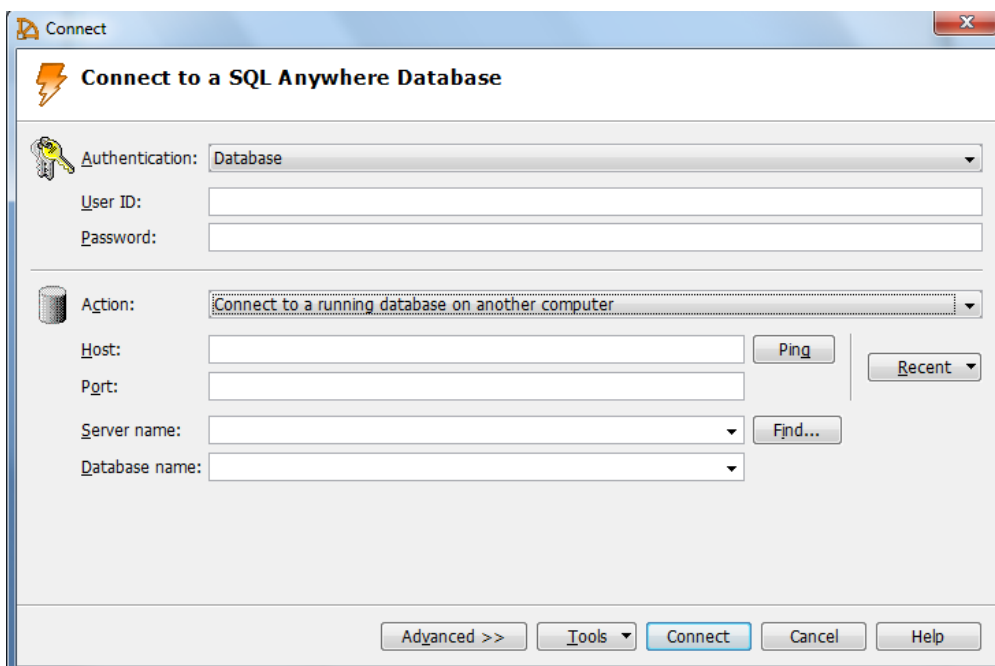
Joonis 3. Sybase Centralis veebiserveriga ühenduse loomine.

Seejärel ilmub andmebaasiga ühildumise aken. Sinna tuleb sisestada kasutajanimi (*User ID*) ja parool (*password*), Demo andmebaasi puhul on kasutajanimi: DBA ja parool: sql. Kui eelnevalt on juba käivitatud andmebaasi server näiteks server Demo, mille käivitamise käsk oli (`dbeng12 -xs http(port=8082) "%SQLANYSAMP12%\demo.db"`), siis tuleb valida *Action* rippmenüüst *connect to a running database on this computer* ja seejärel sisestada serveri (*Server name*) ja andmebaasi (*Database name*) nimed, Demo andmebaasi puhul on need mõlemad demo.



Joonis 4. Veebiserveri andmete sisestamine.

Peale kasutaja, ühenduse ja andmebaasi andmete sisestamist tuleb vajutada nupule *Connect*, ühendamise katkestamiseks tuleb vajutada nupule *Cancel*. Kui andmebaasi serverit ei ole eelnevalt käivitatud, tuleb valida *Action* rippmenüüst *Start and connect to a database on this computer*. Kui serverit käideldakse mõnes teises võrgu arvutis, tuleb valida *Connect to a running database on another computer* ja seejärel sisestada serveri andmed või kasutada *find* funktsiooni.



Joonis 5. Võrgus asuvasse veebiserverisse ühendumine.

Olles saanud ligi andmebaasile saab alustada veebiteenuste loomisega. Seda saab teha nii Sybase Central'iga kui ka interaktiivses sql'is.

2.2 Veebiteenused

2.2.1 Veebiteenuste kirjeldus

Veebiteenuste SQL süntaks on järgnev:

```
CREATE SERVICE service-name
TYPE { 'RAW' | 'HTML' | 'JSON' | 'XML' }
[ URL [PATH] { ON | OFF | ELEMENTS } ]
[ common-attributes ]
[ AS { statement | NULL } ]
common-attributes:
[ AUTHORIZATION { ON | OFF } ]
[ ENABLE | DISABLE ]
[ METHODS 'method, ...' ]
[ SECURE { ON | OFF } ]
[ USER { user-name | NULL } ]
```

Veebiteenuse nimi võib olla ükskõik milline jada tähestiku tähtedest, numbritest, kaldkriipsudest (/), sidekriipsudest (-), alakriipsudest (_), punktidest (.), hüüumärkidest (!), tildedest (~), tärnidest (*), ülakomadest ('), vasakpoolsetest ja parempoolsetest ümarsulgudest (()), aga ta ei tohi alata kaldkriipsuga (/) ega sisaldada kahte või enamat kõrvuti asetsevat kaldkriipsu (/). Kui panna veebiteenuse nimeks *root*, siis ta saab automaatselt erilised tunnused. Kutsudes välja mingi veebiteenus, mida andmebaasis ei ole, käsitletakse seda kui *root* teenust, kui *root* teenus on defineeritud. *Root* teenusega on lihtne käsitleda valesti sisestatud teenuseid või teenuseid, mida andmebaasis ei leidu.

TYPE tingimus määrab millist tüüpi veebiteenus välja näeb, valida saab kas RAW, HTML, JSON, XML, sellel tingimusel vaikeväärtus puudub.

- RAW - andmed saadetakse brauserile ilma igasuguse formaatimiseta.
- HTML - andmed saadetakse HTML esitusena tabelist või vaatest.
- JSON - andmed saadetakse JavaScript objektidena.
- XML - andmed saadetakse XML kujul.

URL tingimus määrab, kas URL teed (*path*) on lubatud ja kui on, siis kuidas neid töödeldakse.

- OFF määrab, et teenuse nimele URL päringus ei tohi järgneda tee. See on vaikeväärtus.
- ON määrab, et teenuse nimele URL päringus võib järgneda tee, vaid üks tee on lubatud.
- ELEMENTS määrab, et teenuse nimele URL päringus võib järgneda tee, 10 teed on lubatud.

AUTHORIZATION tingimus määrab, kas kasutaja peab teenusele juurdepääsemiseks ennast tuvastama kasutajanime ja parooliga. Vaikeväärtus on ON, mis määrab, et kasutaja peab ennast tuvastama. Kui AUTHORIZATION on OFF, siis tuleb USER tingimusega kindlaks määrata, mis kasutaja õigustes see teenus välja kutsutakse.

ENABLE/DISABLE tingimus määrab, kas teenus on kättesaadav või mitte. Vaikeväärtus on ENABLE, mis määrab, et teenus on kättesaadav. Hooldustöödeks on hea kasutada DISABLE suvandit, nii ei pea teenust tervikuna kustutama, et seda kasutajatele kättesaamatuks muuta.

METHODS tingimus määrab, millised HTTP meetodeid teenus toetab. Valida saab DEFAULT, POST, GET, HEAD, PUT, DELETE, või NONE. Seejuures RAW, HTML ja XML tüübi puhul on POST, GET, HEAD vaikeväärtused.

SECURE tingimus määrab, kas teenusele saab ligi läbi turvalise või ebaturvalise pordi. ON määrab, et ainult HTTPS ühendused on lubatud ja kui esineb HTTP ühendus, siis see suunatakse HTTPS porti. OFF määrab, et mõlemad - nii HTTP kui HTTPS ühendused on lubatud, OFF on ka vaikeväärtus.

USER tingimus määrab, millisel andmebaasi kasutajal või kasutajagrupil on õigused teenust välja kutsuda. Kui AUTHORIZATION tingimus on ON, aga USER tingimus on määramata, siis lubatakse juurdepääs kõigile kasutajatele.

AS tingimus lubab veebiteenusel välja kutsuda andmebaasi käske. Veebiteenust saab muuta kasutades käsku `ALTER SERVICE teenuse nimi` ning lisades sellele muudetava tingimuse nime ja väärtuse. Veebiteenuseid saab kustutada käsuga `DROP SERVICE`

teenuse nimi. Teenuste loomiseks, muutmiseks ja kustutamiseks peavad olema andmebaasi administraatori õigused [42].

2.2.2 Veebiteenuste vaatamine

Esimese veebiteenusena võiks luua klassikalise HelloWorld'i näite. Selleks tuleks minna interaktiivsesse sql'i ja sisestada käsk:

```
CREATE SERVICE TereMaailm
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS SELECT 'Hello world!';
```

Loodud veebiteenust saab vaadata kasutades suvalist veebibrauserit. Eeldades, et server on loodud eelnevalt toodud käsuga (`dbeng12 -xs http(port=8082) "%SQLANYXSAMP12%\demo.db"`), saab loodud veebiteenust vaadata kui brauserisse sisestada järgnev aadress: `http://localhost:8082/demo/TereMaailm`. Selline aadress tuleb sisestada sellepärast, et meie poolt loodud server kasutab http protokollit, see oli privaatne server, mis kuulab http päringuid pordis 8082, andmebaasi nimi oli demo ja veebiteenuse nimi oli TereMaailm. Veebiteenuste aadressi määrab lisaks serveri andmetele ka veebiteenuse nimi ja kuidas teenus on disainitud. On võimalik ka defineerida *root* veebiteenus, mis käsitleb kõiki neid http päringuid, millele ei vasta ükski veebiteenus. Lähemalt *root* teenustest saab informatsiooni peatükis veebiteenuste loomine. Täpne süntaks millega veebiteenuseid vaadata on

```
{http|https}://host-name[:port-number][/dbn]/service-name[/path-name?url-query]
```

Parameetrite kirjeldus:

- *Host-name* ja *port-number* määravad veebiserveri asukoha ja pordi numbri, kuhu päring saadetakse. *Host-name* võib olla veebiserverit käitleva arvuti IP aadress või selle arvuti nimi. *Port-number* peab olema sama, mis oli defineeritud serveri käivitamisel või kui server kuulab mitut porti, siis üks nendest. Kui server käivitati standardportidega, http puhul port 80 ja https puhul port 443, siis ei pea pordi numbrit lisama.
- *Dbn* tähendab andmebaasi nime, andmebaasi peab käitlema samas veebiserveris ja temas peavad olema defineeritud veebiteenused. *Dbn*'i ei pea sisestama, kui

veebiserveris käideldakse ainult ühte andmebaasi või kui serveri käivitamisel oli määratud andmebaasi nimi.

- `Service-name` määrab väljakutsutava veebiteenuse nime. Sellise nimega veebiteenus peab vastavas andmebaasis olemas olema. Kui `service-name` jääb sisestamata ja `root` teenust ei ole defineeritud siis kuvatakse *404 Not Found* error, kui `root` teenus on defineeritud, siis edastatakse päring `root` teenusele, kui ka `root` ei saa päringust aru, siis kuvatakse sammuti *404 Not Found* error.
- `Path-name` - kui on kindlaks tehtud veebiteenuse nimi, siis ülejäänud osa päringust edastatakse `path-name`'le. Kui teenus loodi URL ON parameetriga, siis on kasutades URL http muutujat kättesaadav terve tee. Kui veebiteenuse loomisel kasutati parameetrit URL ELEMENTS, siis on kättesaadav iga tee element eraldi kasutades http muutujaid URL1 kuni URL10. Muutujaid saab kasutada kui protseduuri sees kasutada HTTP_VARIABLE funktsiooni. Näiteks kui on loodud veebiteenus, milles URL parameeter on ELEMENTS.

```
CREATE SERVICE TestWebService
TYPE 'HTML'
URL ELEMENTS
AUTHORIZATION OFF
USER DBA
AS CALL TestProcedure(:url1,:url2);
```

Kui see veebiteenus välja kutsuda näiteks järgneva URL päringuga: `http://localhost/demo/TestWebService/Assignment1/Assignment2/Assignment3` siis kutsutakse välja veebiteenus nimega `TestWebService`, mis omakorda kutsub välja protseduuri nimega `TestProcedure` ja määrab `url1` väärtuseks `Assignment1` ja `url2` väärtuseks määratakse `Assignment2` ja protseduuri sees on võimalik kätte saada ka kolmas parameeter kasutades `HTTP_VARIABLE` funktsiooni, näiteks `HTTP_VARIABLE('url3')` annab väärtuseks `Assignment3`

- `url-query` - lisaks `path-name`'le saab muutujaid edasi anda ka `url-query`'na sel juhul peavad olema muutuja nimed defineeritud. Kasutada saab neid samamoodi nagu ka `path-name`'is ette antud muutujaid. Oletame, et meil on järgmine veebiteenus

```

CREATE SERVICE ShowSalesOrderDetail
TYPE 'HTML'
URL OFF
AUTHORIZARION OFF
USER DBA
AS CALL ShowSalesOrderDetail(:costomer_id, :product_id);

```

siis saab veebiteenust välja kutsuda näiteks järgneva URL aadressina
http://localhost/demo/ShowSalesOrderDetail?costomer_id=101&product_id=300

sellisel juhul kutsutakse välja veebiteenus nimega `ShowSalesOrderDetail`, mis omakorda kutsub välja protseduuri nimega `ShowSalesOrderDetail` ja annab protseduurile ette muutujad `costomer_id`, mille väärtus on 101 ja `product_id`, mille väärtus on 300. Protseduuri tulemus kuvatakse seejärel ekraanile [43].

Muutujaid saab kasutada järgnevates olukordades

- `SELECT`, `INSERT`, `UPDATA` või `DELETE` lausetes ükskõik mis kohal, kus on lubatud number või sõne väärtus.
- `INTO` tingimuses `SELECT` või `FETCH` lauses
- muutujaid on võimalik kasutada ka tingimuse, kursori või parameetri nimena.
- `CONNECT`, `DISCONNECT` ja `SET CONNECT` lausetes saab muutujat kasutada `ID`, parooli, ühenduse-, serveri-, või andmebaasi nimena.
- `SET OPTION` ja `GET OPTION` käskudes saab muutujaid kasutada suvandi väärtusena.

Muutujaid ei saa kasutada

- Tabeli või tulba nimena,
- Pakkfailides (*batch*),
- `SET` lause alampäringutes [44].

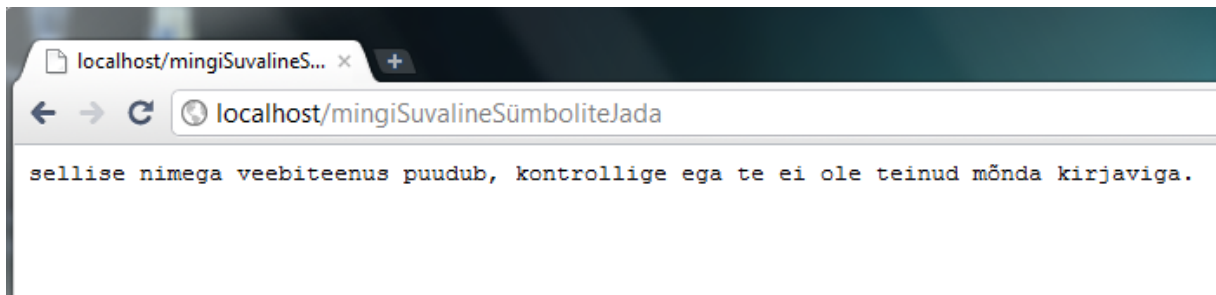
2.2.3 Veebiteenuste loomine

Praeguseks on käsitletud `http` veebiserveri ja veebiteenuse loomist ning nende väljakutsumist veebibrauseris. Järgmine loodav veebiteenus võiks olla *root* teenus, millest oli enne ka põgusalt juttu. *Root* teenus kutsutakse välja, kui läbi veebilehitseja pöördatakse kas *root* teenuse poole või mõne sellise teenuse poole, mis andmebaasis puudub [45]. *Root* teenuse loomiseks tuleb kõigepealt luua andmebaasi server, kui seda ei ole juba tehtud,

juhendid serveri loomiseks asuvad peatükis 2.1.1 serveri käivitamine. Seejärel tuleb avada interaktiivne sql ja luua veebiteenus, veebiteenuse loomisest oli juttu peatükis 2.2.1 veebiteenuste kirjeldus. *Root* teenuse loomiseks tuleb teenuse nimeks panna *root*. Lihtsa näitena võib luua näiteks sellise veebiteenuse:

```
CREATE SERVICE root
  TYPE 'RAW'
  AUTHORIZATION OFF
  SECURE OFF
  URL OFF
  USER DBA
  AS SELECT 'Sellise nimega veebiteenust andmebaasis ei
leidu.';
```

Kontrollimaks, kas antud teenus töötab, tuleb avada suvaline veebibrauser, näiteks Mozilla Firefox, Internet Explorer või mõni muu. Eeldusel, et on loodud privaatserver standardses http pordis, tuleb sisestada aadressiribale `http://localhost/*`, kus tärn (*) võib olla ükskõik milline sümbolite jada, väljaarvatud mõne juba andmebaasis eksisteeriva veebiteenuse nimi. Tulemuseks peab olema sarnane järgnevale.



Joonis 6. Root veebiteenuse väljund veebilehitsejas.

Saadessellise tulemuse olemise veendunud, et *root* veebiteenus töötab. Järgmise näitena võiks luua veebiteenuse, mis väljastab mõne andmebaasis asuva tabeli sisu. Selleks on kasulik veebiteenuse tüübina kasutada `html`, kuna `html` formaat tagastab tulemuse automaatselt tabeli kujul. Loodav veebiteenus võiks välja näha analoogne järgnevaga:

```
CREATE SERVICE veebiteenuse_nimi
  TYPE 'HTML'
  AUTHORIZATION OFF
  SECURE OFF
  URL OFF
  USER DBA
  AS SELECT * FROM tabeli_nimi;
```

Kus `tabeli_nimi` on mõne andmebaasis asuva tabeli nimi. Teenuse väljundit näeme, kui läheme aadressile `http://localhost/veebiteenuse_nimi`.

Loomaks muutujatega veebiteenusid tuleb luua järgnevaga analoogne veebiteenus

```
CREATE SERVICE veebiteenuse_nimi
  TYPE 'HTML'
  AUTHORIZATION OFF
  SECURE OFF
  URL ELEMENTS
  USER DBA
  AS SELECT protseduuri_nimi (:url1, :url2, :muutuja);
```

Kasutamaks seda veebiteenust peame looma ka protseduuri nimega protseduuri_nimi.

Selleks tuleb minna interaktiivsesse sql'i ja luua näiteks järgnev protseduur

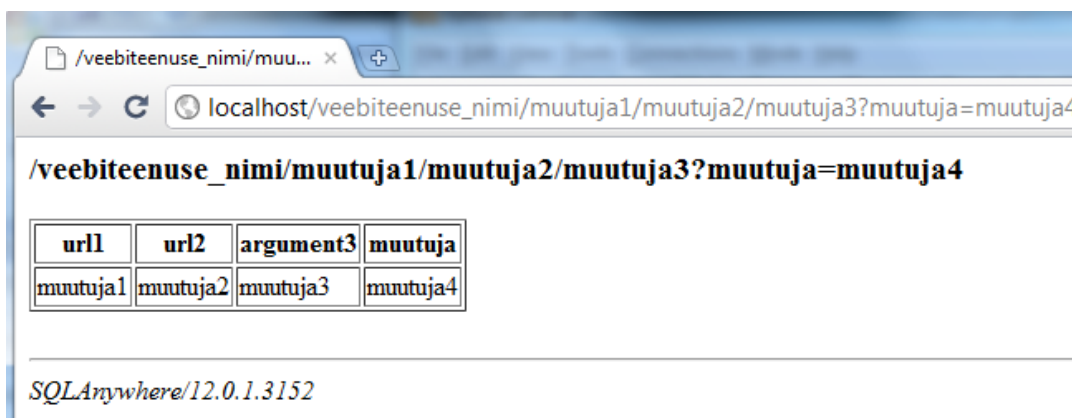
```
CREATE PROCEDURE protseduuri_nimi(url1 long varchar, url2 long
varchar, muutuja long varchar)

BEGIN
  DECLARE argument3 LONG VARCHAR;
  SET argument3 = HTTP_VARIABLE('url3');
  SELECT url1, url2, argument3, muutuja;
END
```

Sellise veebiteenuse saab välja kutsuda kui brauseri aadressribale kirjutada:

`http://localhost/veebiteenuse_nimi/muutuja1/muutuja2/muutuja3?muutuja=muutuja4`

sellisel juhul omistatakse url1 väärtuseks muutuja1, url2 väärtuseks muutuja2, muutuja väärtuseks muutuja4 ja kuigi veebiteenuses spetsifitseeriti vaid 3 muutujat saab ette anda neid rohkem ning protseduuri sees HTTP_VARIABLE käsuga saab muutuja väärtuse kätte. Antud juhul deklareeritakse muutuja argument3, mis on long varchar tüüpi ning omistatakse talle väärtus, mis on peale kolmandat kaldkriipsu alates veebiteenuse nimest. Muutujate määramisel tuleb alati kaldkriipsuga eraldatud muutujad enne sisestada kui nimega defineeritud muutujad. Kui nimega muutujaid on rohkem kui üks, siis tuleb nad eraldada ampersandiga (&). Sellise veebiteenuse väljund peaks olema analoogne järgnevaga:



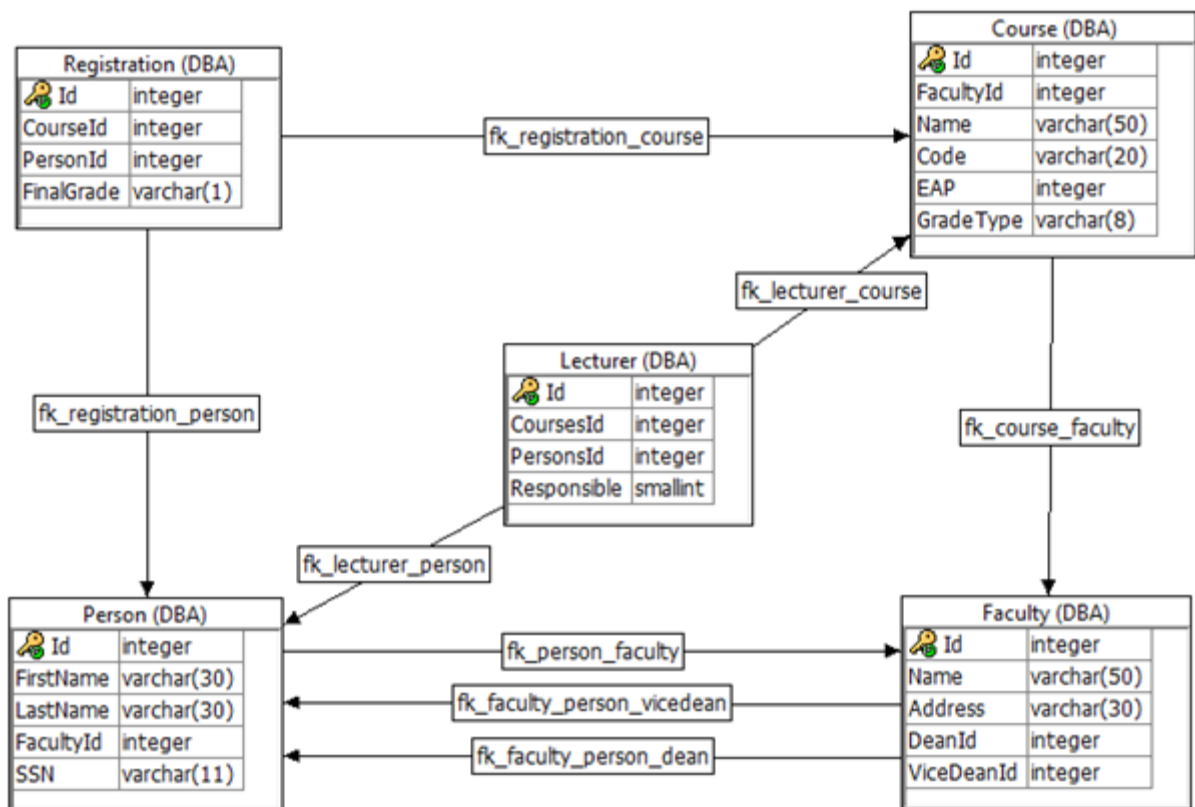
Joonis 7. Muutujate kasutamise väljund veebibrauseris.

3 Andmebaasimudelid ja ülesanded

3.1 Andmebaaside kursuses kasutatavad andmebaasi mudelid

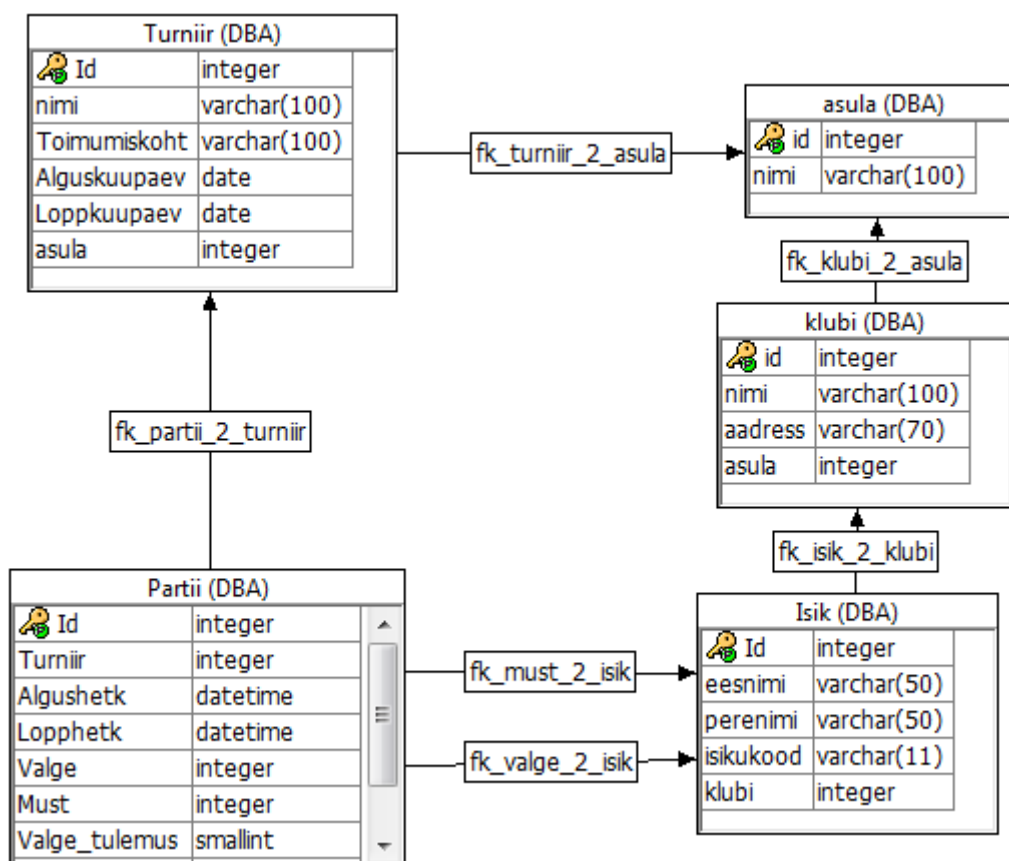
Andmebaaside kursuses kasutatakse kahte andmebaasi mudelit. Üks nendest kajastab male turniire, koos mängijate, klubide ja asulatega. Teine andmebaasi mudel on lihtsustatud versioon õppeinfosüsteemist. Mudelite ER diagrammid on järgnevad:

Õppeinfosüsteemi mudel



Joonis 8. Lihtsustatud õppeinfosüsteemi kujutava andmebaasi ER-diagramm.

Maleturniiride mudel



Joonis 9. Maleturniire kujutava andmebaasi ER-diagramm.

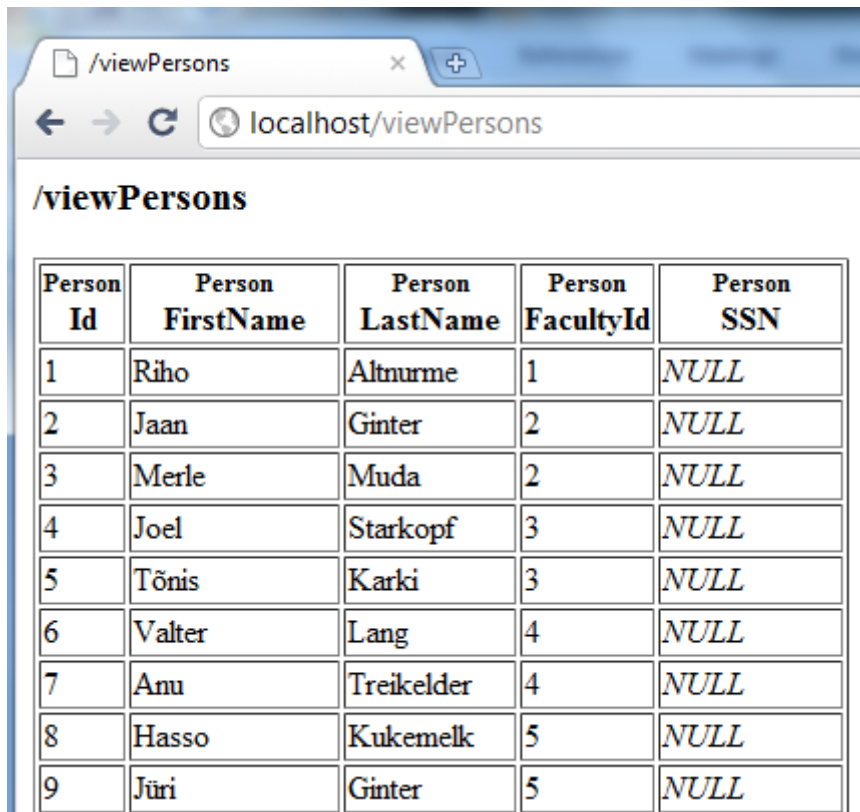
3.2 Ülesanded tudengi andmebaasidele

Nendele andmebaasidele peaksid Andmebaaside kursusel õppivad tudengid looma veebiteenuseid, mida oleks võimalik veebilehitsejas välja kutsuda. Seda on võimalik realiseerida kirjutades komplekti veebiteenuseid ja funktsioone. Kõige esimese ülesandena võiksid tudengid luua andmebaasi tabelite vaatamise võimaluse läbi veebibrauseri. Järgnevalt demonstreeritakse, kuidas see käib õppeinfosüsteemi kujutava andmebaasi tabeli Person näitel. Tudeng peaks tegema iga andmebaasis esineva tabeli kohta analoogse veebiteenuse:

```
CREATE SERVICE viewPersons
  TYPE 'HTML'
  AUTHORIZATION OFF
  USER DBA
  AS SELECT * FROM Person;
```

Tüübiks võiks valida html, kuna siis väljastatakse andmed just meile vajalikul tabeli kujul. Kuna tegemist on õppeotstarbeliste veebiteenustega ja antud veebiteenus lubab vaid

andmeid vaadata, siis ei ole autori hinnangul kasutaja autentimine vajalik, kuid kindlasti võiks funktsiooniga tutvumise mõttes luua ühe analoogse veebiteenuse, mille vaatamiseks peab kasutaja ennast autentima, seda saab teha kui `AUTHORIZATION OFF` asemel kirjutada `AUTHORIZATION ON`. Kuna kursuse raames ei ole loodud kasutajaid, kellel oleks tabelite vaatamise õigus, siis kasutajaks tuleks märkida DBA ehk andmebaasi administraator. Seda veebiteenust saab välja kutsuda, kui veebilehitseja aadressribale kirjutada: `http://localhost/viewPersons` ja väljundi algus peaks olema analoogne järgnevaga:



Person Id	Person FirstName	Person LastName	Person FacultyId	Person SSN
1	Riho	Altnurme	1	NULL
2	Jaan	Ginter	2	NULL
3	Merle	Muda	2	NULL
4	Joel	Starkopf	3	NULL
5	Tõnis	Karki	3	NULL
6	Valter	Lang	4	NULL
7	Anu	Treikelder	4	NULL
8	Hasso	Kukemelk	5	NULL
9	Jüri	Ginter	5	NULL

Joonis 10. Veebiteenuse viewPerson väljundi algus veebibrauseris.

Kontrollimaks, kas antud väljund on õige tuleks interaktiivses sql'is sisestada käsk `SELECT * FROM Person;` ja kontrollida kas need väljundid on identsed. Järgmise ülesandega võiks realiseerida tabelist kirje(te) otsimise etteantud parameetri alusel ehk kasutada tuleb muutujat. Näitena on kirjutatud veebiteenus ja protseduur, mis väljastavad Person tabeli kõik need kirjed, kus eesnimeks on märgitud kasutaja poolt sisestatav väärtus

```
CREATE SERVICE searchPersons
  TYPE 'HTML'
  AUTHORIZATION OFF
  USER DBA
  AS CALL p_searchPersons(:eesnimi);
```

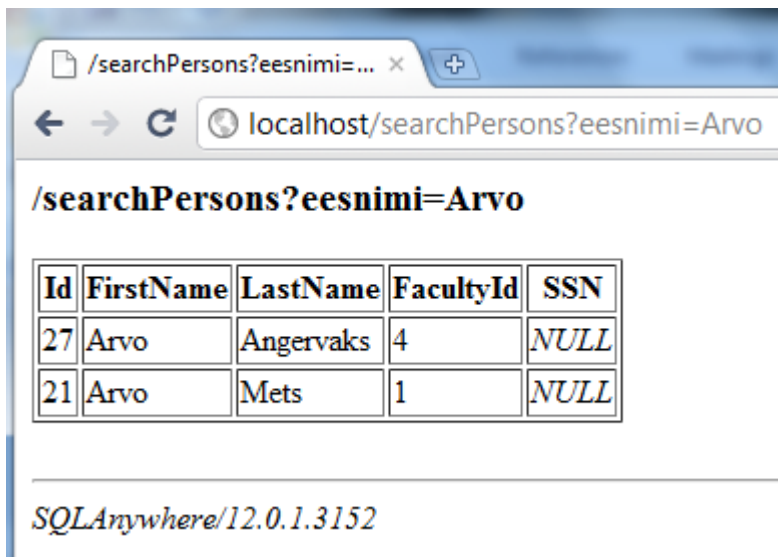
```

CREATE PROCEDURE p_searchPersons(eesnimi varchar(30))
BEGIN
    SELECT * FROM Person
    where FirstName = eesnimi
END

```

Eeldusel, et on loodud http veebiserver standardses pordis saab veebiteenust kasutada kui brauseris aadressribale kirjutada:

http://localhost/searchPersons?eesnimi=A kus A on mingi suvaline väärtus näiteks Arvo, sellisel juhul peaks veebibrauseris kuvatav väljund olema sarnane järgnevale:



Joonis 11. Veebiteenuse searchPerson väljund veebilehitsejas, kui muutujaks määrata: Arvo.

Järgmise sammuna võiks üritada sisestada uusi kirjeid mõnda tabelisse. Näiteks võib teha järgneva veebiteenuse ja funktsiooni, millega saab lisada kirjeid tabelisse Person

```

CREATE SERVICE addPerson
TYPE 'RAW'
AUTHORIZATION OFF
USER DBA
AS SELECT
f_addPerson(:isikuId,:eesnimi,:perenimi,:teaduskond,:isikukood);

CREATE FUNCTION f_addPerson( isikuId integer, eesnimi
varchar(30), perenimi varchar(30), teaduskond integer, isikukood
varchar(11))
RETURNS LONG VARCHAR
BEGIN
INSERT INTO Person
(Id,FirstName,LastName, FacultyId,SSN)
VALUES (isikuId,eesnimi,perenimi,teaduskond,isikukood)
END

```

Antud veebiteenusega saab lisada näiteks isiku nimega Sander Ott, kes õpib matemaatika-informaatikateaduskonnas ja isikukoodiga 38902131234 tabelisse Person, kui brauserisse kirjutada järgnev aadress:

```
http://localhost/addPerson?isikuId=301&eesnimi=Sander&perenimi=Ott&teaduskond=9&isikukood=38902131234
```

Selline isikute sisestamise viis on aga äärmiselt kohmakas. Andmete inimsõbralikuks sisestamiseks on mõistlik kasutada mõnda programmeerimiskeelt, näiteks PHP või JavaScript, aga andmete sisestamist saab parandada ka kasutades SQL Anywhere'i funktsioone kuhu on sisse kirjutatud html lauseid. Näiteks isikute lisamiseks tabelisse Person võib kirjutada juurde ühe funktsiooni ja muuta natuke veebiteenust. Veebiteenuses tuleb muuta vaid väljakutsutava funktsiooni nime. Uus veebiteenus võiks välja näha järgnev:

```
ALTER SERVICE addPerson
  TYPE 'RAW'
  AUTHORIZATION OFF
  USER DBA
  AS SELECT
  f_Person(:isikuId,:eesnimi,:perenimi,:teaduskond,:isikukood);
```

lisatav funktsioon võiks olla analoogne järgnevaga:

```
CREATE FUNCTION f_Person(isikuId integer, eesnimi varchar(30), perenimi
varchar(30), teaduskond integer, isikukood varchar(11))
returns LONG VARCHAR
begin
  call sa_set_http_header('Content-type', 'text/html');
  RETURN '<html><head><title>Isikud</title></head><body><h2>sellel lehel
saab lisada isikuid tabelisse Person</h2>
<form action="addPerson" method="get">
sisesta ID:
<input size="4" type="integer" name="isikuId" value=""
|| isikuId
|| "/>
<br />
sisesta eesnimi:
<input size="30" type="text" name="eesnimi" value=""
|| eesnimi
|| "/>
<br />
sisesta perenimi:
<input size="30" type="text" name="perenimi" value=""
|| perenimi
|| "/>
<br />
vali teaduskonna Id nime järgi:<br>
<select name="teaduskond">
```

```

<option value=1 > Usuteaduskond</option>
<option value=2 > Õigusteaduskond</option>
<option value=3 > Arstiteaduskond</option>
<option value=4 > Filosoofiateaduskond</option>
<option value=5 > Haridusteaduskond</option>
<option value=6 > Kehakultuuriteaduskond</option>
<option value=7 > Loodus- ja tehnoloogiateaduskond</option>
<option value=8 > Majandusteaduskond</option>
<option value=9 > Matemaatika-informaatikateaduskond</option>
<option value=10 > Sotsiaalteaduskond</option>
<option value=15 > Kotsukeste abikool</option>
<option value=16 > Kotsukeste põhikool</option>
</select>
<br />
sisesta isikukood:
<input size="11" type="text" name="isikukood" value=""
|| isikukood
|| '"/>
<br />
<input type="submit" value="Sisesta" />
</form><hr/>'
|| if (length (isikuId) > 0 and length (eesnimi) > 0 and length
(perenimi) > 0
and length (teaduskond) > 0) then '<code>'
||'isik lisatud andmebaasi'
|| f_addPerson(isikuId, eesnimi, perenimi, teaduskond, isikukood) else ''
end IF
|| '</body></html>';
end

```

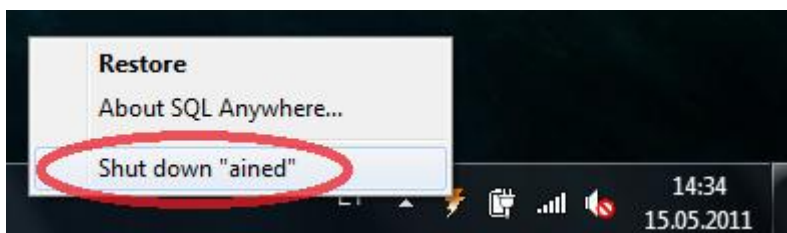
Selles funktsioonis on kasutatud sisseehitatud protseduuri (`sa_set_http_header`), mis lubab veebiteenusel määrata http väljundi päise (response header). Antud juhul on protseduuri `sa_set_http_header('Content-type', 'text/html')` väärtused 'Content-type' ja 'text/html' mis määravad, et protseduuri sisu on html. Selle sisseehitatud protseduuri süntaks on järgnev: `sa_set_http_header(fldname, val)`, kus `fldname` on CHAR(128) tüüpi parameeter, mis määrab http päise välja ja `val` on LONG VARCHAR tüüpi parameeter, mis määrab `fldname` poolt määratus http päise väärtuse. Määrates `val` väärtuseks NULL kustutatakse antud päis.

Veebilehitsejas saab seda välja kutsuda samamoodi nagu eelmist veebiteenust, sest teenus ise ei muutunud, juurde tuli vaid üks funktsioon. Ainuke erinevus on selles, et enam ei pea ette andma muutujaid mitte aadressribalt vaid andmete sisestamiseks saab kasutada lahtreid. Veebiteenus peaks välja nägema sarnane järgnevaga:



Joonis 12. Veebiteenuse addPerson väljund veebilehitsejas.

Seniste juhendite järgi on loodud privaatne veebiserver, mis on hea serveri ja andmebaasi arendamiseks. Oletades, et nüüdseks loonud andmebaasi funktsionaalsus on lõplik ja oleks aeg hakata jagama oma teenuseid ka teistele meiega samas võrgus olevatele arvutitele. Selleks tuleb sulgeda hetkel töötav privaatserver. Seda saab teha nii tegumirealt kui ka käsurealt. Käsureal tuleb sisestada järgnev käsk: `dbstop -y serveri_nimi`. Tegumirealt saab serverit sulgeda, kui serveri ikoonil teha hiirega parem klõps ja valida *Shut down „serveri_nimi“*



Joonis 13. Veebiserveri sulgemine Windowsi tegumirealt.

Kui privaatserver on suletud saab luua võrguserveri. Selle loomine käib samamoodi nagu privaatserveri loomine, ainult dbeng tuleb asendada dbsrv'iga. Näiteks saab käivitada serverit käsuga: `dbsrv12 -xs http(port=8082) "A"`, kus A on andmebaasi faili asukoht arvutis. Olles loonud võrguserveri tuleks paluda, et mõni kaastudeng üritaks selles serveris asuvaid veebiteenuseid oma arvutist välja kutsuda. Eeldades, et server on alustatud ülaltoodud käsuga tuleb veebibrauseri aadressribale kirjutada `http://A:8082/B`, kus A on veebiserverit käitleva arvuti ip aadress ja B on veebiteenuse nimi. Näiteks kui ip

aadress on 172.17.1.251 ja veebiteenuse nimi on addPerson, siis näeks aadress välja selline: <http://172.17.1.251:8082/addPerson>. Kui serverisse ühendumine õnnestus siis peab veebilehitsejas kuvatama eelnevalt loodud veebiteenus. Katsetada kas selles veebiserveris on loodud kõik veebiteenused ja kas nende väljund on õige.

4 Tudengi andmebaaside automaatne kontrollimine

Tudengite loodud veebiteenuseid saab õppejõud kontrollida samamoodi nagu seda saab teha teine tudeng, neid veebilehitsejas väljakutsudes. Kuna veebiteenuse sisu tuleb täielikult andmebaasist, saab veebiteenuste pealt vaadata, kas andmebaas on sellises seisus nagu kursuse tingimustes ette antud. Selle kontrollimiseks tuleb luua komplekt funktsioonidest ja veebiteenuselt, mis käitub interaktiivse sql'ina. Selleks saab kasutada näiteks järgnevat veebiteenuse ja funktsioonide komplekti

```
CREATE service isql
authorization off
user dba
type 'raw'
url elements
AS SELECT f_isql(:paring);
```

```
CREATE FUNCTION f_isql (paring long varchar)
returns long varchar
begin
  call sa_set_http_header('Content-type', 'text/html');
  RETURN '<html><head><title>Web iSQL</title></head><body><h2>Web
iSQL</h2>
<form action="ws_isql" method="get">
<TEXTAREA NAME="paring", ROWS=10, COLS=100></TEXTAREA>
<br>
<input type="submit" value="Sisesta" />
</form><hr/>'
  || IF length (paring) > 0 then '<code>'
  || paring
  || '</code><br/><br/>'
  || dynamic_sql_to_html_table(paring) else '' end IF
  || '</body></html>';
End
```

```
CREATE FUNCTION dynamic_sql_to_html_table(dynamic_sql_query long varchar)
returns long varchar
begin
  declare html_table_header      long varchar;
  declare xml_query              long varchar;
  declare temp_table_definition  long varchar;
  declare temp_table_column_list long varchar;
  declare res                    long varchar;
  SELECT
    XMLELEMENT(name tr, XMLATTRIBUTES('header' AS class), XMLAGG
(XMLELEMENT(name th, XMLCONCAT(XMLELEMENT(name small,
"base_table_name"), XMLELEMENT(name br), 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 sa_foo
```

```

(sa_maintain_order_of_derived_table int default autoincrement, ' ||
list('"' || "name" || "' ' || "domain_name_with_size", ',') || ')',
LIST('"' || "name" || "' ', ',')
INTO html_table_header, xml_query, temp_table_definition,
temp_table_column_list
FROM sa_describe_query(dynamic_sql_query);
execute immediate temp_table_definition;
execute immediate 'insert into sa_foo(' || temp_table_column_list ||
') ' || dynamic_sql_query;
execute immediate
'select '<table border=1>'
|| html_table_header
|| "' ' || '
|| xml_query
|| ' || '</table>' into res from sa_foo';
RETURN res;
exception
when others then
RETURN '<b>Error!</b> ' || ERRORMSG(SQLCODE);
end;

```

See veebiteenuse ja funktsioonide komplekt on saadud Sybase blogist [46]. Väikeseid muudatusi on tehtud funktsioonile `f_isql`, seal on asendatud sisestamisväli (*input field*) tekstialaga (*textarea*), et päringute sisestamine oleks loomulikum. Antud veebiteenuse võiks tudeng oma andmebaasi ise luua. Kui teenus on edukalt andmebaasis ja võrguserver on käivitatud, saab õppejõud läbi brauseri juurdepääsu tudengi andmebaasile. Kui eelnevalt on õppejõul olemas komplekt funktsioone, saab ta need lihtsa vaevaga sisestada tudengi andmebaasi. Antud funktsioone saab seejärel brauseris esile kutsuda ja vaadata kas nende väljund on ootuspärane. Sellisel viisil saab õppejõud mugavalt oma arvuti tagant lahkumata kontrollida tudengi andmebaasi õigsust.

Kokkuvõte

Töös selgus, et lihtsamaid veebiteenusi on väga kerge luua. Selleks on vaja teha vaid mõnerealine veebiteenus, mida on võimalik esile kutsuda suvalises veebilehitsejas. Samas ei paku veebiteenused meile kõiki soovitavaid võimalusi, seda just muutuja piirangute pärast. Näiteks ei saa muutujatega edasi anda tabeli või veeru nime. Osutub võimatuks ilma teiste programmeerimiskeelte abita luua veebiteenusi uute tabelite koostamiseks. Selleks on SQL Anywhere'i sisse ehitatud paljude programmeerimiskeelte tugi. Andmebaaside kursus on aga esimese aasta sügissemestri aine ja seega ei saa tudengitelt eeldada piisavaid teadmisi, et antud kursuse raames sisse tuua veel mõni programmeerimiskeel. Tudengitel on võimalus teiste keelte abil andmebaasi manipuleerida hilisemates kursustes, näiteks Tarkvaratehnika MTAT.03.094.

Autori hinnangul pakuvad SQL Anywhere'i veebiteenused siiski selle kursuse jaoks piisavat funktsionaalsust andmaks tudengitele esmase kokkupuute andmebaaside reaalsest kasutamisest, mis on oluline tekitamiseks edasist huvi andmebaaside vastu. Teadmised andmebaasidest on aga tudengite edasist elu arvestades ülimalt olulised. Raske on leida head äritarkvara, mis ei kasuta andmebaase. Edaspidises elus võib tudengil kokkupuuteid tulla ka Sybase toodanguga, loodan, et antud bakalaureusetöö sisendab lugejale kindlust kasutamaks Sybase poolt pakutavaid tooteid. Sybase on pika ajalooga tarkvara tootja, mille juured ulatuvad aastasse 1984, mis selles valdkonnas on küllaltki pikk aeg. Lisaks ajaloole on Sybasel ette näidata ka hulganisti saavutusi, millest osa on välja toodud lisas 2. Sybase'i nimel on ka 148 patenti ning 185 patenti ootab kinnitamist.

Kasutades SQL Anywhere'i sisseehitatud veebiserverit koos veebiteenusiga, mis muudab brauseri interaktiivseks sql'iks, tekib õppejõududele mugav võimalus kontrollida tudengite andmebaase. Selleks tuleb läbi veebilehitseja sisestada andmebaasi funktsioone, mis kontrollivad andmebaasi vastavust kursuse jooksul püstitatud eesmärkidele. Sisestatud funktsioone saab välja kutsuda interaktiivses keskkonnas, kuid on võimalik teha ka vastav veebiteenus, mis antud funktsioone välja kutsub. Antud meetodit kasutades peaks õppejõul kuluma vähem aega tudengite kontrollimisele ja seega jääb tal rohkem aega õpetamisele.

Web services in SQL Anywhere database

Bachelor Thesis

Sander Ott

Summary

The aim of this graduation work is to create learning material about making web services in database that allow accessing database by a web browser. The work is divided into four main chapters:

- summary of Sybase and SQL Anywhere,
- starting a http web server, creating web service and accessing them in web browser,
- making example exercises,
- student database evaluation.

In the first chapter I give a brief history of Sybase and SQL Anywhere and also show where they are positioned today. The aim of this chapter is to show that Sybase is a big and trustworthy enterprise with long history on its field.

The second chapter explains how to start and stop the SQL Anywhere inbuilt http server. Gives and explains the syntax of web service and also gives some example web services. The chapter also shows how to access the web service via web browser.

In the third chapter I introduce database models that are used in database course. Then I show some database specific exercises like writing a web service that shows some of the tables when called in web browser or creating a web service for inserting new lines into database table.

In the final chapter a convenient way for evaluating student database is shown. For that a set of functions and web service that turn browser into interactive sql have to be inserted into database. Then lecturers can insert already existing control functions into student database with their own computer.

In conclusion I would say that making web applications only with sql is not the best options. But for learning purposes it is a easy way to make some kind of user interface to your database, which in my opinion is essential to rise students interest of databases.

Viited

1. SAP'i andmed: <http://www.sap.com/corporate-en/our-company/index.epx>, 30.05.2011
2. Sybase andmed: http://www.sybase.com/about_sybase, 30.05.2011
3. Sybase tooted ja nende grupid: <http://www.sybase.com/products>, 30.05.2011
4. Sybase partnerid: <http://www.sybase.com/partner/alliance>, 30.05.2011
5. Sybase ostab powersofti:
http://www.lannigan.org/powersoft_powerbuilder_history.htm, 30.05.2011
6. Sybase ostab mobile 365'e: <http://www.sybase.com/detail?id=1047934>, 30.05.2011
7. Sybase ostab paypoxi: <http://www.sybase.com/detail?id=1061874>, 30.05.2011
8. SAP ostab Sybase: <http://www.sap.com/corporate-en/press/newsroom/press.epx?pressid=13202>, 30.05.2011
9. Sybase ajalugu: http://www.sybase.com/about_sybase/history, 30.05.2011
10. Adaptive Server Anywhere 6.0 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0600e/dbupen6/@Generic__BookView, 30.05.2011
11. Adaptive Server Anywhere 6.0.2 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0602e/dbfgen6/@Generic__BookView, 30.05.2011
12. Adaptive Server Anywhere 6.0.3 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0603e/dbfgen6/@Generic__BookView, 30.05.2011
13. Adaptive Server Anywhere 7.0.0 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0700e/dbbgen7/@Generic__BookView, 30.05.2011
14. Adaptive Server Anywhere 7.0.2 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0702e/dbbgen7/@Generic__BookView, 30.05.2011
15. Adaptive Server Anywhere 7.0.3 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0702e/dbbgen7/@Generic__BookView, 30.05.2011
16. Adaptive Server Anywhere 7.0.4:
<http://download.sybase.com/pdfdocs/awg0704e/dbbgen7.pdf>, 30.05.2011

17. Adaptive Server Anywhere 8.0 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sas/awg0800e/dbbgen8/@Generic__BookView, 30.05.2011
18. Adaptive Server Anywhere 8.0.2 arhiiv:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbbgen8/@Generic__BookView, 30.05.2011
19. Adaptive Server Anywhere 9.0.0 uuendused:
http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1001/en/html/dbwnen10/wn-new-new900.html, 30.05.2011
Adaptive Server Anywhere 9.0.1 uuendused:
http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1001/en/html/dbwnen10/wn-new-new901.html, 30.05.2011
Adaptive Server Anywhere 9.0.2 uuendused:
http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1001/en/html/dbwnen10/wn-new902.html, 30.05.2011
20. SQL Anywhere 10.0 <http://www.sybase.com/detail?id=1042264>, 30.05.2011
21. Sql anywhere 10 uuendused:
http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1001/en/html/dbwnen10/wn-newjasper.html, 30.05.2011
SQL Anywhere 10.0.1 uuendused:
http://www.ianywhere.com/developer/product_manuals/sqlanywhere/1001/en/html/dbwnen10/wn-new1001.html, 30.05.2011
22. Sql anywhere 11.0.0 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/newpanorama.html>,
30.05.2011
SQL Anywhere 11.0.1 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/new1101.html>, 30.05.2011
23. SQL Anywhere 12.0.0 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/newwinnsbruck.html>,
30.05.2011
sql anywhere 12.0.1uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/new1201.html>, 30.05.2011
24. SQL Anywhere kirjeldus:
<http://www.sybase.com/products/databasemanagement/sqlanywhere>, 30.05.2011

25. SQL anywhere arendus keeled:
<http://www.sybase.com/products/databasemanagement/sqlanywhere/webapplications>, 30.05.2011
26. SQL Anywhere'i toetatavate platvormide nimekiri -
<http://www.sybase.com/products/databasemanagement/sqlanywhere>, 30.05.2011
27. High Availability kirjeldus:
http://dcx.sybase.com/index.html#1101/en/dbadmin_en11/database-mirroring-overview.html, 30.05.2011 ja demonstratsioon sellest
<http://sqlanywhere.blogspot.com/2009/04/demonstrating-high-availability.html>, 30.05.2011
28. Monitor Tools kirjeldus: <http://dcx.sybase.com/index.html#1201/en/dbadmin/da-server-monitoring.html>, 30.05.2011
29. Read-only scale-out kirjeldus:
<http://dcx.sybase.com/index.html#1201/en/dbadmin/da-copy-nodes.html>, 30.05.2011
30. In-Memory kirjeldus: <http://dcx.sybase.com/index.html#1201/en/dbusage/ug-perform-s-3273136.html>, 30.05.2011
31. In-memory tüüpide võrdlev tabel: <http://sqlanywhere.blogspot.com/2008/03/who-wants-in-memory-mode.html>, 30.05.2011
32. RSA: <http://vallaste.ee/> otsingusõnaga RSA, 30.05.2011
33. FIPS kirjeldus: <http://dcx.sybase.com/index.html#1201/en/dbadmin/dbserver-s-5299616.html>, 30.05.2011
34. ECC: <http://vallaste.ee/> otsingusõnaga ECC, 30.05.2011
35. Veebiserveri käivitamine: <http://dcx.sybase.com/index.html#1201/en/dbadmin/xs-database-dbengine.html>, 30.05.2011
36. -xs dbeng12/serv12 server options:
<http://dcx.sybase.com/index.html#1201/en/dbadmin/xs-database-dbengine.html>, 30.05.2011
37. Network protocol options: <http://dcx.sybase.com/index.html#1201/en/dbadmin/da-protocol-options.html>, 30.05.2011
38. Mitme parameetriga serveri alustamine:
<http://dcx.sybase.com/index.html#1201/en/dbprogramming/sidebar-configuring-network-protocol-options.html>, 30.05.2011

39. Mitme serveri samaaegne alustamine:
<http://dcx.sybase.com/index.html#1201/en/dbprogramming/pg-https-server-s-4992010.html>, 30.05.2011
40. Dpstop käsu andmed:
<http://dcx.sybase.com/index.html#1201/en/dbadmin/dbstop.html>, 30.05.2011
41. dbisql käsu andmed: <http://dcx.sybase.com/index.html#1201/en/dbadmin/dbisql-interactive-dbutilities.html>, 30.05.2011
42. CREATE SERVICE käsu kirjeldus:
<http://dcx.sybase.com/index.html#1201/en/dbreference/create-service-statement.html>, 30.05.2011
43. Veebiteenuste avamine brauseris:
<http://dcx.sybase.com/index.html#1201/en/dbprogramming/urls-http.html>,
30.05.2011
44. Muutujate võimalikud kasutused:
<http://dcx.sybase.com/index.html#1201/en/dbprogramming/hovusag.html>,
30.05.2011
45. Root teenused: <http://dcx.sybase.com/index.html#1201/en/dbprogramming/pg-https-server-rootservices.html>, 30.05.2011
46. Interaktiivne sql brauseris: <http://iablog.sybase.com/efarrar/2009/09/creating-a-simple-browser-based-isql/#more-442>, 30.05.2011

LISA 1

1. Adaptive server anywhere 6.0.1 uuendused:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0603e/dbupen6/@Generic__BookView
2. Adaptive server anywhere 6.0.2 uuendused:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0603e/dbupen6/@Generic__BookView
3. Adaptive server anywhere 6.0.3 uuendused:
http://manuals.sybase.com/onlinebooks/group-sasarc/awg0603e/dbupen6/@Generic__BookView
4. Adaptive server anywhere 7.0.1 uuendused:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbwnen8/@Generic__BookView
5. Adaptive server anywhere 7.0.2 uuendused:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbwnen8/@Generic__BookView
6. Adaptive server anywhere 7.0.3 uuendused:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbwnen8/@Generic__BookView
7. Adaptive server anywhere 8.0 uuendused:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbwnen8/@Generic__BookView
8. Adaptive server anywhere 8.02 uuendused:
http://manuals.sybase.com/onlinebooks/group-sas/awg0802e/dbwnen8/@Generic__BookView
9. Sql Anywhere 10.0.0 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/newjasper.html>
10. Sql anywhere 10.0.1 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/new1001.html>
11. Sql anywhere 11.0.0 uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/newpanorama.html>
12. Sql anywhere 11.0.1uuendused:
<http://dcx.sybase.com/index.html#1201/en/sachanges/new1101.html>

13. Sql anywhere 12.0.0 uuendused:

<http://dcx.sybase.com/index.html#1201/en/sachanges/newinnsbruck.html>

14. Sql anywhere 12.0.1 uuendused:

<http://dcx.sybase.com/index.html#1201/en/sachanges/new1201.html>

LISA 2

1988 – Sybase on esimene kes tuleb turule klient-server relatsioonilise andmebaasiga

1994 – Sybase kuulutatakse vahevara tootmises turu liidriks.

2001 - Sybase iAnywhere juhib mobiilsete andmebaaside turgu viiendat aastat järjest

2002 Mai - Sybase iAnywhere võidab Mobility Award parima andmebaasi kategoorias.

2002 Juuli - Sybase saavutab rekordilise tulemuse ECperf benchmark testil.

2003 August - Sybase võidab Hiinas "Best Customer Service Award" kolmandat aastat järjest.

2006 Mai - Sybase Adaptive Server Enterprise (ASE) 15 ja AvantGo võidavad vastavalt "Best Database Management Solution" ja "Best Mobile Solution" Codie auhinna, mis antakse kasutajate poolt silmapaistvate saavutuste eest.

2006 Juuni - Sybase märgitakse IT Week poolt koostatavas edetabelis "Top 100 Vendors of 2006".

2006 detsember Sybase 365 on aastaga edastanud rekordilised 25 miljardit sõnumit.

2007 oktoober - IDC annab Sybase iAnywhere'ile kuuendat aastat järjest esikoha mobiilsete seadmete haldamistarkvara loomise eest.

2008 mai - Sybase IQ analüütiline server teeb uue Guinness maailma rekordi, kui ta käitleb maailma suurimat andmeladu Sun SPARC® Enterprise M9000 serveril.

2009 mai - Forrester märgib Sybase'i mobiilsete seadmete haldamisel turu liidriks.

2009 detsember - Sybase 365 saab auhinna "mCommerce Innovator of The Year" [8]