

SISÄLLYS

1 JOHDANTO (ST)	1
1.1 OHJELMAN TAUSTA JA KÄYTTÖTARKOITUS (ST)	1
1.2 DOKUMENTISSA KÄYTETYT KÄSITTEET JA LYHENTEET (ST)	1
1.3 DOKUMENTIN KIRJOITTAJAT (KS)	2
2 UUDET OMINAISUUDET (KS)	3
2.1 VAATIMUKSET LYHYESTI (KS)	3
2.1.1 Relatiaoalgebran harjoittelu (KS)	3
2.1.2 Ylläpito-operaatiot (KS)	3
2.1.3 Tehtävänlaadinta (KS)	3
2.2 RAJAUKSET (KS)	4
2.2.1 Relatiaoalgebran harjoittelu (KS)	4
2.2.2 Ylläpito-operaatiot (KS)	4
2.2.3 Tehtävienlaadinta (KS)	4
2.3 SQL-TRAINER VUOKAAVIO (KS)	5
3 OHJELMISTON YMPÄRISTÖN KUVAUS (GE)	7
3.1 NOUDATETTAVAT STANDARDIT (KS)	7
3.2 OHJELMOINTITYÖKALUT (GE)	7
3.3 MUUT KÄYTETYT OHJELMISTOT (KS)	8
4 TIETOKANTAKUVAUS (KS)	9
4.1 UUDEN VERSION TIETOKANTA (KS)	11
4.2 UUDEN VERSION MUUT APUTAULUT (KS)	12
5 LUOKKAKUVAUKSET (JN)	13
5.1 TEHTÄVÄNVALINTALUOKKA (GE)	13
5.1.1 OlistPage (GE)	13
5.2 RELATIOALGEBRAN HARJOITTELUVÄLINE (GE)	14
5.2.1 AlgebraParser (GE)	14
5.2.1 AlgebraParser –luokan metodit (GE)	15
5.3 YLLÄPITO-OPERAATIOIDEN HARJOITTELUVÄLINE (ST)	17
5.3.1 OupdateAnswerForm (ST)	17
5.3.2 OUpdateAnswerForm metodit (ST)	17

5.4 TEHTÄVIENLAADINNAN TYÖKALUJEN LUOKAT (JN).....	19
5.4.1 <i>OlistPage (JN)</i>	19
5.4.2 <i>OtaskDef (JN)</i>	20
5.4.3 <i>OtaskReg (JN)</i>	20
5.4.4 <i>OtaskStore (JN)</i>	21
5.5 TARKISTUSLUOKAN PARANNUKSET (ST).....	22
5.5.1 <i>Suoritettavan SQL:n tarkistus (ST)</i>	22
5.5.2 <i>OAnswerForm (ST)</i>	22
6 KÄYTTÖLIITTYMÄT (KS)	23
6.1 RELAATIOALGEBRAN HARJOITTELU (JN,KS).....	24
6.1.1 <i>Relaatioalgebrakäyttöliittymän toiminta (JN,KS)</i>	25
6.1.2 <i>Käyttöliittymän virheilmoitukset (JN)</i>	26
6.2 YLLÄPITO-OPERAATIOIDEN KÄYTTÖLIITTYMÄ (ST, KS)	28
6.2.1 <i>Ylläpito-operaatioiden toiminnallinen määrittely (ST,KS)</i>	29
6.3 TEHTÄVIENLAADINNAN KÄYTTÖLIITTYMÄN MÄÄRITTELY (JN,KS)	31
6.3.1 <i>Tehtävänlaadinnan käyttöliittymän toiminnallinen määrittely (JN,KS)</i>	33
7 TESTAUS (GE, JN, ST).....	34
7.1 RELAATIOALGEBRAN HARJOITTELUVÄLINEEN TESTAUS (GE)	34
7.1.1 <i>AlgebraParser-luokan testaus (GE)</i>	34
7.1.2 <i>Relaatioalgebran harjoitteluvälineen käyttöliittymän testaus (JN)</i>	35
7.2 YLLÄPITO-OPERAATIOIDEN TESTAUS (ST).....	36
7.3 TEHTÄVÄNANTOVÄLINEIDEN TESTAUSSUUNNITELMA (JN).....	37

LÄHTEET



1 Johdanto (ST)

Tässä dokumentissa määritetään OHTU SQL-Trainer kesä 2000 -ryhmälle toimeksiannettuun SQL-Trainer ohjelmaan lisättävä uusi toiminnallisuus

SQL-trainer on tietokantojen perusteet -kurssilla käytössä oleva ohjelmisto, jolla opiskelijat suorittavat kurssiin liittyviä harjoituksia. Tällä hetkellä ohjelmisto sisältää mahdollisuudet harjoitella vain SQL-kyselyitä.

Ohjelmistoa laajennetaan dokumentissa esitellyin tavoin siten, että sillä voidaan jatkossa tehdä myös relaatioalgebraan ja tietokannan ylläpitoon liittyviä tehtäviä. Lisäksi ohjelmistoon lisätään aputyökalu, jolla voidaan laatia tehtäviä kaikkiin edellä mainittuihin osioihin.

1.1 Ohjelman tausta ja käyttötarkoitus (ST)

SQL-trainer harjoitteluohjelmisto otettiin käyttöön syksyllä 1999 kurssilla Tietokantojen perusteet, jossa opiskelijat käyttävät sitä harjoitustehtävien tekemiseen. SQL-trainer on kurssin pitäjän Harri Laineen käsialaa.

Tällä hetkellä ohjelmisto soveltuu SQL:n harjoitteluun, mutta ohjelmistoa on tarkoitus laajentaa siten, että sillä voitaisiin harjoitella myös SQL:n ylläpito-operaatioita ja relaatioalgebraa, joita kumpaakin opetetaan Tietokantojen perusteet -kurssilla.

Ohjelmistoon sisältyy tehtävienlaadinta työkalu, jota on tarkoitus kehittää siten, että sillä voidaan tehdä kaiken tyyppisiä tehtäviä.

1.2 Dokumentissa käytetyt käsitteet ja lyhenteet (ST)

JDBC (Java Database Connectivity) tarkoittaa rajapintaa Java-ohjelmien ja tietokantojen välillä.

Java-servletti on laitteistoriippumaton ohjelmamoduuli, jota suoritetaan palvelimella. Servletit tarjoavat ohjelmoijalle kaikki Javan sovellusohjelmointirajapinnat (mm. JDBC, ks. alla) sekä HTTP-kutsukirjaston. Tämän ansiosta servletit ovat luonnollinen valinta interaktiivisten Internet-sovellusten toteutuksille.



Servlettimoottori (palvelmamoottori) on palvelimeen asennettava ohjelmisto, joka tekee palvelimesta täysin Java-servletti yhteensopivan. Servlettimoottori toimii Java-ohjelmille virtuaalikoneena ja Javan käyttö CGI-kielenä on historiaa.

1.3 Dokumentin kirjoittajat (KS)

Tässä dokumentissa esiintyneet kirjoittajat on mainittu kappaleiden otsakkeissa nimikirjaimin. Lyhenteet on esitetty taulukossa 1.

KS	Kimmo Sinkko
JN	Jaakko Nurro
ST	Santtu Toivakka
GE	George El-Khoury

Taulukko 1. Nimikirjaimet

2 Uudet ominaisuudet (KS)

Tässä projektissa laajennetaan SQL-trainer -ohjelmistoa siten, että pelkkien SQL -kyselyiden lisäksi sillä voidaan harjoitella myös relaatioalgebraa sekä SQL:n ylläpito-operaatioita. Tehtävien tarkistuksen suorittavaa ohjelman osaa täydennetään muutamalla uudella tarkistusmenetelmällä. Lisäksi laaditaan käyttöliittymä ja menetelmät, kuinka eri tyyppisiä tehtäviä jatkossa tullaan syöttämään järjestelmään.

2.1 Vaatimukset lyhyesti (KS)

Uusien ominaisuuksien määrittely ja tarkemmat vaatimuskuvaukset löytyvät määrittelydokumentista [Maar00].

2.1.1 Relaatioalgebran harjoittelu (KS)

Lähtökohtana relaatioalgebran harjoitteluosuudelle on, että uuden käyttöliittymän avulla voidaan harjoitella alkeisoperaatioita sekä monimutkaisempia kyselyrakenteita alkeisoperaatioita yhdistelemällä. Vaikka toiminnallisuus toteutetaan käyttöliittymässä relaatioalgebran alkeisoperaatioina, ne konvertoidaan palvelimessa SQL-operaatioiksi ja käsitellään kuten muutkin tehtävätyypit. Alkeisoperaatioiden käytöstä syntyneet kyselyt tallennetaan tilapäistauluihin ja niitä voidaan käyttää osina laajempia operaatioita tai myöhemmin tutkimuksen pohjana.

2.1.2 Ylläpito-operaatiot (KS)

Lähtökohtana on laajentaa tunnettujen SQL -operaatioiden valikoimaa INSERT, UPDATE ja DELETE käskyillä. Perusvaatimuksena on ollut, että opiskelijan suorittaman tietoa muuttavat operaatiot eivät saa muuttaa perustehtävätaulujen sisältöä. Tästä seuraa muutamia rajoituksia (ks. 2.2 Rajaukset (KS))

2.1.3 Tehtävänlaadinta (KS)

Tarkoituksena on helpottaa eri tyyppisten tehtävien syöttämistä järjestelmään. Merkittävimpinä muutoksina on tarkistusmenetelmälle syötettävän tarkistussumman laskeminen järjestelmän puolesta.

2.2 Rajaukset (KS)

Projektin aikataulun rajallisuuden huomioon ottaen joudutaan tekemään tiettyjä rajoituksia järjestelmän uudistamiselle. OHTU –ryhmä ei uudelleenkirjoita tai dokumentoi sellaisia jo olemassa olevia järjestelmän luokkia, joita ei muuteta tämän projektin kuluessa. Sellaiset alkupe-
räiset luokat, joita muutetaan tai jotka toimivat uusien luokkien perustana, kuuluvat luonnolli-
sesti projektin piiriin ja niihin kohdistetaan tässä projektissa käytettäviä työ- ja dokumentointi-
menetelmiä.

2.2.1 Relaatiaoalgebran harjoittelu (KS)

Käyttöliittymä edellyttää käyttäjän selaimelta Javascript –tuen. Lisäksi relaatioalgebran lausek-
keet esitetään Symbol –fonttia käyttäen, joten jos käyttäjän ympäristö ei sisällä kyseistä fonttia,
näkyvät relaatioalgebran lausekkeet standardifontilla kirjoitettuna.

2.2.2 Ylläpito-operaatiot (KS)

Ylläpito-operaatiot suoritetaan simuloimalla tietoa muuttavia operaatioita aputaulun avulla. Syy
menettelylle on se, että lähtökohtaisesti on vaadittu, että tehtävätaulujen sisältöä ei saa muuttaa.
Tästä syystä yhden tehtävän ylläpito-operaatiot saavat kohdistua yhteen harjoittelutauluun. Jos
tehtävä laaditaan toisin ja sitä yritetään suorittaa, ei ole mahdollista saada hyväksyttyä vastausta
(koska järjestelmä ei osaa kohdentaa muutettuja rivejä oikein).

Samasta syystä ylläpitotehtävät eivät saa riippua toisistaan (ts. ”ota tehtävässä n saatu tulos
ja...” tai tehtävää ei voida jatkaa toisessa tehtävässä (ts. ”jatka tehtävää n siten että...”)).

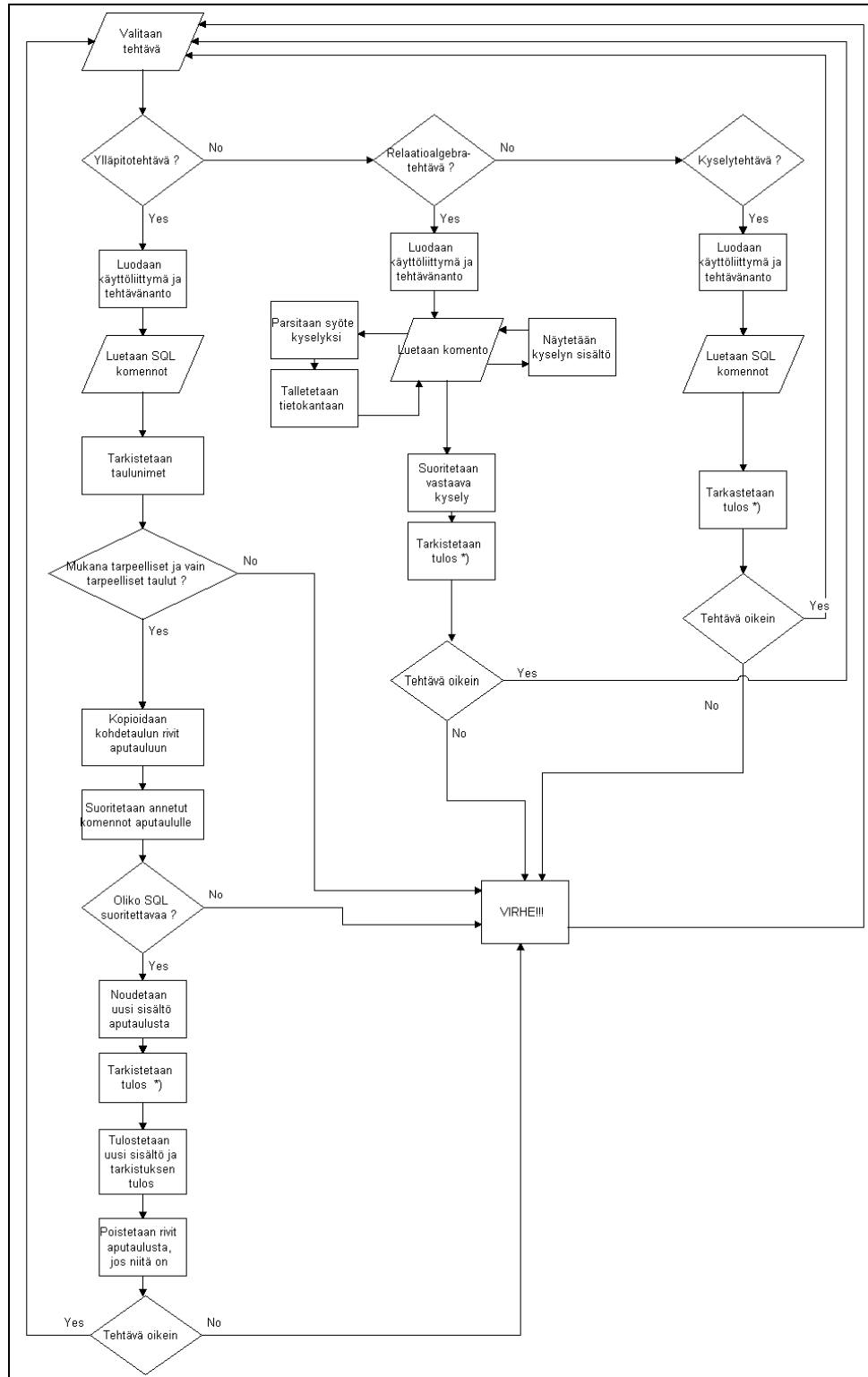
Käyttäjän muokkaama sisältö poistetaan aputaulusta tehtävän tarkistamisen yhteydessä. Näin
ollen mitään muokattua tietosisältöä ei jää järjestelmään kun tehtävä on valmis.

2.2.3 Tehtävienlaadinta (KS)

Tehtävienlaadinnan käyttöliittymään toteutetaan laatijan kannalta tärkeitä lisäyksiä. Tässä pro-
jektissa toteutetaan ainoastaan tarkistussumman laskeminen ja tehtävienlaadinnan muokkaami-
nen sellaiseksi, että sillä voidaan syöttää myös uusien tehtävätyyppien tehtävät samaan tapaan
kuin aiemminkin.

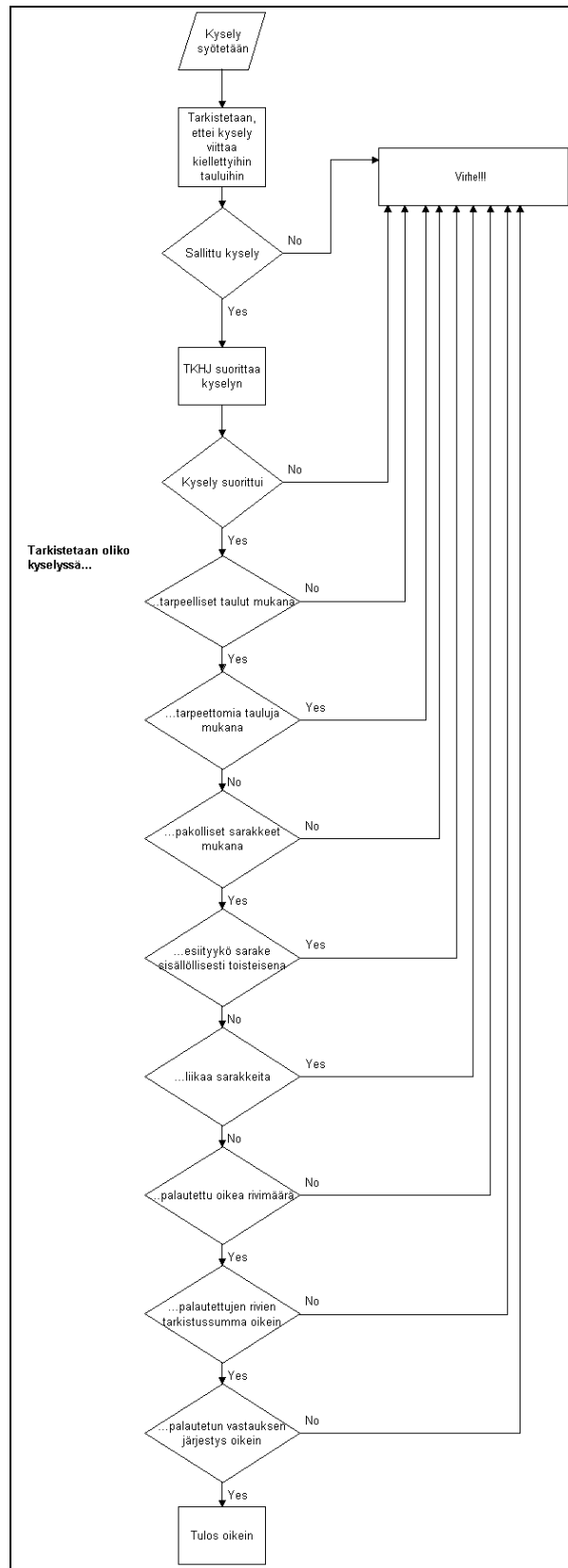
2.3 SQL-Trainer vuokaavio (KS)

Oheisessa kuvassa 1. selvitetään kuinka järjestelmä toimii erityyppisten tehtävien tapauksessa.



Kuva 1. SQL-Trainer vuokaavio

Kuvassa 2 tarkennetaan edellisen kuvan ”Tarkistus” –osaprosessia.



Kuva 2. Tarkistus

3 Ohjelmiston ympäristön kuvaus (GE)

Sqltr toimii asiakas-palvelin periaatteella internetin tai intranetin välityksellä (WWW-selaimen kautta). Järjestelmän kaikki luokat ja tietokanta-taulut sijaitsevat tietojenkäsittelytieteen laitoksen db.cs.helsinki.fi -palvelimella, johon käyttäjien on mahdollista ottaa yhteyttä myös kyseiseen laitoksen verkkoon kuulumattomilla tietokoneilla (palvelin ei ole ns. sisäänpäin suljettu).

Httpd palvelimena toimii Apache palvelin versio 1.3.12 ja Java servlettimoottorina Apache Jserv versio 1.1. Järjestelmän tietokantana käytetään Oraclen 7.3.2.3.0, johon kytkeydytään JDBC -ajureilla.

Järjestelmä pyritään rakentamaan mahdollisuuksien mukaan yllämainitusta ympäristöstä riippumattomaksi, varsinkin tietokannan käsittelyyn liittyvissä asioissa.

3.1 Noudatettavat standardit (KS)

Java-koodi noudattaa SQL-Trainerin edellisessä versiossa käytettyä ohjelmointitapaa sekä soveltuvien osin Sunin Java-tyyliopasta [Sun00]. HTML-koodi on World Wide Web Consortiumin HTML 4.0 standardin mukaista [Html00].

3.2 Ohjelmointityökalut (GE)

Järjestelmä ohjelmoidaan käyttäen Sun Microsystems Java Jdk 1.2.2 (Java Development Kit) – versioita. Koodi kommentoidaan Javadoc:ia hyväksi käyttäen. Tietokanta- ja tietokantayhteydet toteutetaan kuten yllä mainitaan.

SQL-Trainer järjestelmän versiohallintaan käytetään CVS versionhallintaa versio 1.10.6.



3.3 Muut käytetyt ohjelmistot (KS)

Varsinaisten edellä esitettyjen ohjelmointityökalujen lisäksi projektissa on käytetty seuraavia apuvälineitä:

Tekstinkäsittely	Microsoft Word 97
Yleispiirto-ohjelma	Corel Draw 7, xFig
Vuokaavioiden piirto	iGrafx Professional 8
Tietokannan piirto	Sybase Power Designer 7
Projektinhallinta	Microsoft Project 98

Taulukko 2. Käytetyt ohjelmat

4 Tietokantakuvaus (KS)

Taulukkoryhmä 3 esittelee tietokannan perustaulujen sarakkeiden kenttien sisältämän tiedon ja niiden tietotyypit. Merkintä *ve* –tarkoittaa viite-eheyttä, alleviivaus avainta ja * -merkki ”NOT NULL” -määrettä.

pscourse	Kurssin tiedot		
<u>year</u>	Kurssin vuosi	integer	*
<u>term</u>	Kurssin lukukausi	char	*
startdate	Kurssin aloitusajankohta	date	
enddate	Kurssin lopetusajankohta	date	

psterm	Apurelaatiotaulu, jolla liitetään pscourse ja pstopic loogisesti toisiinsa		
<u>year</u>	Kurssin vuosi, <i>ve pscourse</i>	integer	*
<u>term</u>	Kurssin lukukausi, <i>ve pscourse</i>	char	*
<u>tno</u>	Tehtäväryhmän numero, <i>ve pscourse</i>	integer	*

pstopic	Tehtäväryhmätaulu		
<u>tno</u>	Tehtäväryhmän numero	integer	*
title_fi	Tehtäväryhmän otsikko suomeksi	varchar(60)	
title_en	Tehtäväryhmän otsikko englannuksi	varchar(60)	
tasks	Ei käytössä	integer	*
credits_upto	Tehtäväryhmän vanhenemispäivämäärä	date	*

psstudent	Opiskelijatietotaulu		
<u>sid</u>	Opiskelijan tunnus. Järjestelmän sisäinen	varchar(10)	*
lname	Sukunimi	varchar(48)	*
fname	Etunimi	varchar(48)	
last_visit	Viimeisin käyttöajankohta	date	*
HETU	Henkilötunnus	varchar(11)	
SS	Salasana selkokielenä	varchar(12)	
stnumber	Opiskelijanumero. Yliopiston antama	varchar(20)	

psanswer	Vastaustaulu		
<u>sid</u>	Opiskelijan tunnus, <i>ve psstudent</i>	varchar(10)	*
<u>task_id</u>	Kuten taulussa ptask, <i>ve ptask</i>	integer	*
<u>trycnt</u>	Vastauksien lukumäärä tähän tehtävään	integer	*
asnw	Opiskelijan antama vastaus	varchar(1000)	
trydate	Vastausajankohta	date	
response	Tehtävän tarkistuksen paluuarvo	integer	
extra	Tehtävän tarkistuksen mahdollinen lisäviesti	varchar(500)	

psstate		Opiskelijan tehtävästatus -taulu	
<u>sid</u>	Kuten taulussa psstudent, ve <i>psstudent</i>	varchar(10)	*
<u>task_id</u>	Kuten taulussa pstask, ve <i>pstask</i>	integer	*
lasttry	Vastausten lukumäärälaskuri tälle tehtävälle	integer	
state	Vastauksen tulos tälle tehtävälle	integer	

pstask		Tehtävätaulu	
<u>task_id</u>	Tehtävän numero	integer	*
title_fi	Tehtävän otsikko suomeksi	varchar(60)	*
title_en	Tehtävän otsikko englanniksi	varchar(60)	
task_text_fi	Tehtävänanto suomeksi	varchar(500)	
task_text_en	Tehtävänanto englanniksi	varchar(500)	
ok_rows	Oikean vastauksen koko riveinä	integer	
nonunique	Vastauksen koko, jos distinct puuttuu	integer	
ok_cols	Vastauksen sarakkeiden enimmäismäärä	integer	
ok_checkable	Tarkistussummasarakkeen nimi	varchar(32)	
ok_checkvalue	Oikeasta vastauksesta laskettu tarkistussumma	integer	
ordered	Onko järjestettävä (ei ole käytössä)	char	
neededcols	Välttämättömien sarakkeiden luettelo pilkulla eroteltuna	varchar(120)	
eqpairs	Identtiset sarakeparit (sarake1=sarake2) Tällaisia ei ole syytä ottaa mukaan tulokseen	varchar(120)	
tno	Tehtäväryhmän numero, johon tämä tehtävä kuuluu ve <i>pstopic</i>	integer	
included	Kyselyyn sisältyvä merkkijono, jota käytetään järjestysmääreen tarkistukseen. Ennen vertailua kyselystä karsitaan turhat välilyönnit ja tarkentimet.	varchar(120)	
col_max	Vastauksen sarakkeiden määrä (tämä on tarpeeton)	integer	
needed_tabs	Välttämättömien taulujen luettelo pilkulla eroteltuna Ohjelma antaa virheilmoituksen jos jokin näistä puuttuu	varchar(120)	
needless_tabs	Tarpeettomien taulujen luettelo pilkulla eroteltuna Ohjelma antaa virheilmoituksen, jos jokin näistä on mukana	varchar(120)	
task_type	Tehtävätyyppi	char	
task_answer	Tehtävän mallivastaus	varchar(500)	

Taulukkokryhmä 3. Tietomäärittely

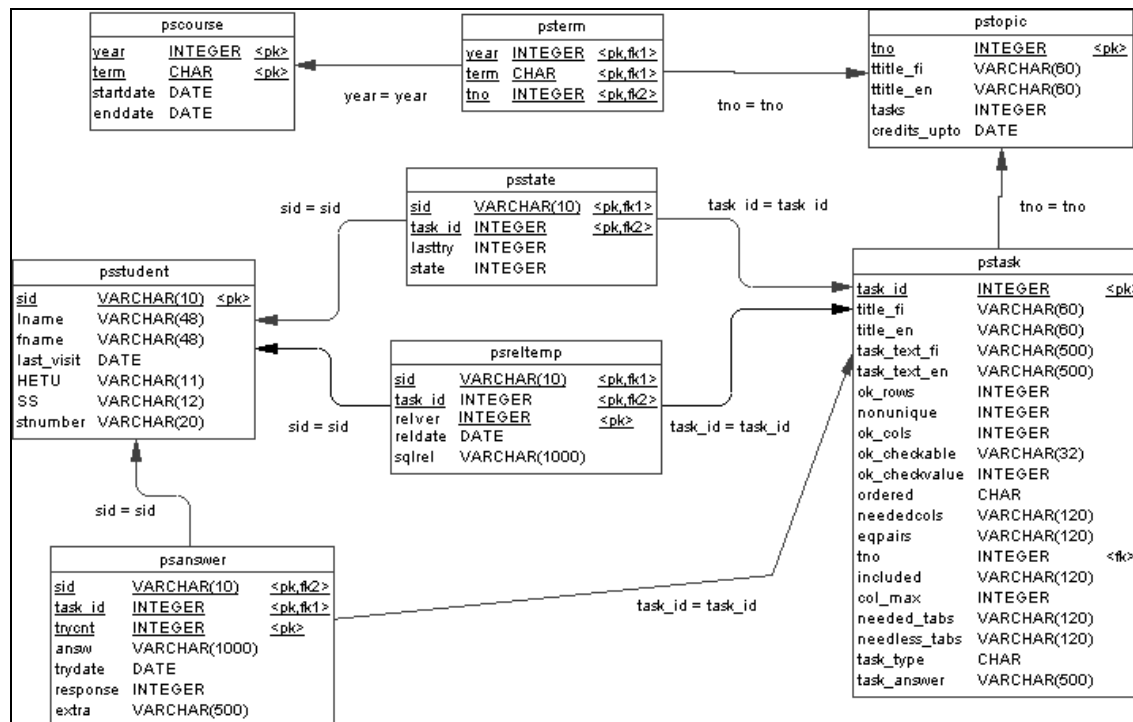
Taulukossa 4 on esitetty relaatioalgebran aputaulun tietomäärittely.

psreltemp		Relaatioalgebran harjoittelun aputaulu	
<u>sid</u>	Kuten taulussa psstudent, ve psstudent	varchar(10)	*
<u>task_id</u>	Kuten taulussa pstask, ve pstask	integer	*
<u>relver</u>	Relaation versiolaskuri	integer	*
reldate	Välituloksen päivämäärä	date	
sqlrel	Välituloksen SQL -kysely	varchar(1000)	

Taulukko 4. Psreltemp taulu

4.1 Uuden version tietokanta (KS)

Kuvassa 3 on esitetty SQL-Trainer perustietokanta määrittelyvaiheen jälkeen tehdyin muutoksin. Kuvassa ei ole esitetty tehtävätauluja, jotka luodaan tehtävien mukaan eikä niiden aputauluja. Nämä taulut käsitellään kohdassa 4.2.



Kuva 3. SQL-Trainer tietokanta

4.2 Uuden version muut aputaulut (KS)

Ylläpito-operaatioiden vaatimukset täyttävä toteutus edellyttää, että jokaista ns. tehtävätaulua kohden luodaan oma aputaulunsa, johon kohdistetaan opiskelijan tekemiä ylläpito-operaatioita. Aputaulut ovat tyhjiä, eli niihin ei tarvitse syöttää tehtävätaulun sisältöä.

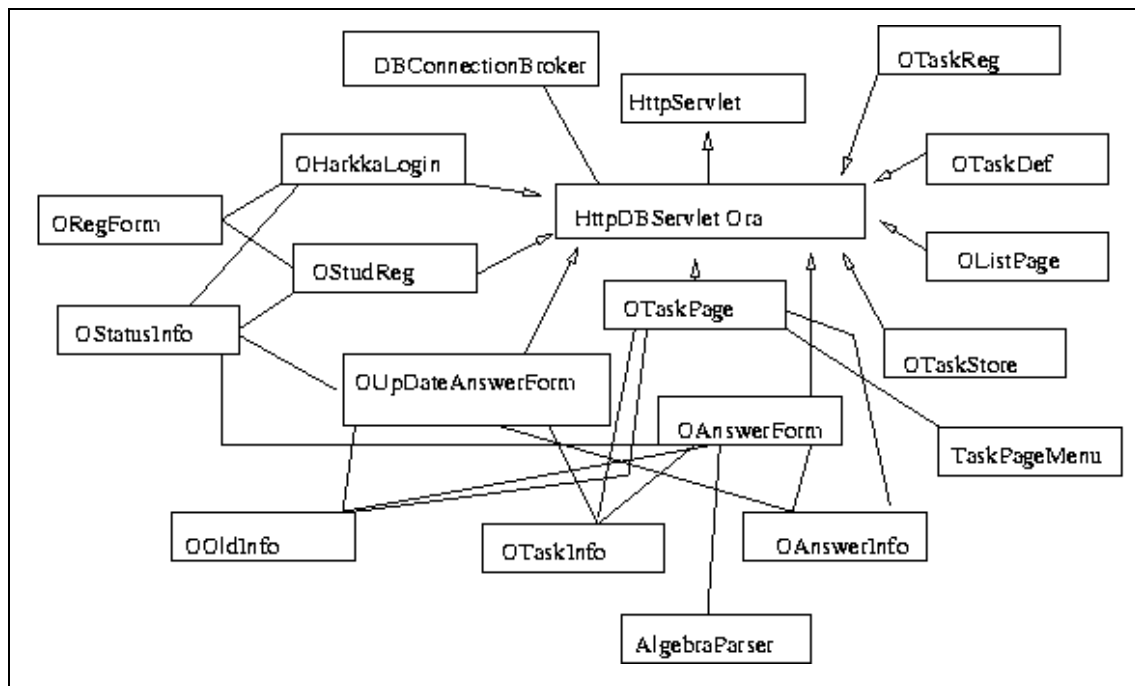
Aputaulut sisältävät kaikki samat sarakkeet, kuin varsinainen tehtävätaulukin, mutta niiden lisäksi kentän sid (student ID), joka viittaa **psstudent** -tauluun. Sid -kentän avulla aputauluun tehtävät muutokset kohdennetaan oikealle käyttäjälle. Sid -sarakkeen tulee olla aina taulun viimeinen sarake.

Aputaulut tulee nimetä alkavaksi "tmp_ " -merkkijonolla ja nimen lopun tulee olla sama kuin tehtävätaulut. Esimerkiksi tehtävätaulun "student" aputaulu olisi nimeltään "tmp_student". Pääkäyttäjän, joka luo tehtävätaulut tehtäväkokonaisuuksia syöttäessään, tulee luoda vähintäänkin niitä tehtävätauluja vastaavat aputaulut, joihin kohdistetaan tietokannan ylläpito-operaatioita sisältäviä tehtäviä.

Jos aputaulua ei löydy järjestelmän olettamalla tavalla, saadaan tulokseksi väärä vastaus vaikka opiskelija olisikin suorittanut tehtävän aivan oikein. TKHJ ilmoittaa tällöin puuttuvasta tai väärästä taulunnimestä.

5 Luokkakuvaukset (JN)

Tässä luvussa esitetään SQL-Trainer järjestelmän luokat luokkakaaviossa lukuunottamatta virheen- ja poikkeusten käsittelyluokkia. Tarkat määrittelyt esitetään niistä luokista ja metodeista, joita uuteen versioon tullaan lisäämään tai alkuperäisistä luokista ja metodeista, joita muutetaan uuden toiminnallisuuden mahdollistamiseksi.



Kuva 4. Luokkakaavio

5.1 Tehtävänvalintaluokka (GE)

Koska nykyiseen järjestelmään liitetään uusia piirteitä, olemassa oleva tehtävänvalintaluokkaa *OListPage* on muutettava tietämään milloin kyseessä on relaatioalgebra-, SQL-, tai SQL -ylläpitotehtävä ja tulostamaan kyseiselle tehtäväryhmälle kuuluvan sivun.

5.1.1 OlistPage (GE)

Tämä luokka suorittaa tehtävisivun tulostamista (lähettämistä) käyttäjälle. Luokassa on periaatteessa kaksi metodia *doPost* ja *doGet*, mutta käytännössä vain yksi *doPost* koska *doGet* ei tee muuta kuin kutsuu *doPost*:ia. Nämä kaksi metodia toteuttavat servletin rajapinnan.

Tietokanta-tauluun *pstask* on lisätty uusi attribuutti *task_type* sisältämään tiedon siitä, onko tehtävä relaatioalgebra, SQL-ylläpito vai SQL-tehtävä. Täten muutetaan luokkaa *OlistPage* luokka siten, että ennen html -sivun tulostamista, kysytään tietokannasta kyseisen tehtävän tyyppi. Tyypin mukaan lähetetään käyttäjälle oikean tehtävätyypin sivu.

5.2 Relaatioalgebran harjoitteluväline (GE)

Tässä luvussa selitetään relaatioalgebran harjoitteluun liittyvät luokat, niiden sisältö ja mahdollisesti muutokset olemassa oleviin luokkiin.

Relaatioalgebran harjoittelu poikkeaa SQL –operaatioiden harjoittelusta siinä, että relaatioalgebran lauseke joudutaan antamaan osissa. Servlet -puolella osat kootaan yhdeksi järkeväksi lausekkeeksi joka kerralla kun käyttäjä lisää uuden osan lausekkeeseen. Kun käyttäjä on tyytyväinen vastaukseensa ja painaa ”Valmis” -painiketta, saatu SQL –lauseke on pystyttävä suorittamaan, jotta vastauksen oikeellisuutta voidaan tarkistaa samoin kuten muissa tehtävätyypeissä.

Relaatioalgebran vastauksen oikeellisuuden tarkistaminen on samanlaista kuin SQL-kyselyn tarkistaminen, mutta relaatioalgebran lauseke on muutettava SQL-kyselyksi ennen tarkistamista. Tästä johtuen rakennetaan *AlgebraParser*-luokka hoitamaan muunnosta lausekkeesta kyselyyn.

5.2.1 *AlgebraParser* (GE)

Tämä luokka sisältää tarvittavat metodit SQL-kyselyn tuottamiseksi relaatioalgebran lausekkeesta. Luokan metodit eivät käytä ulkoisia tietovarastoja, *parseNaturalJoin*-Metodia lukuun ottamatta, joka käyttää tietokantaa hyväksi luonnollisen liitoksen saman nimisten attribuuttien selvittämisessä.

Tässä luokassa relaatioalgebran operaatioiden kuvaamiseksi käytetään seuraavassa taulukossa esitettyjä merkintätapoja. Nämä merkintätavat eivät näy käyttäjälle.

Relaatioalgebran operaatio	Tässä luokassa käytettävä merkintätapa
Projektiio	p(sarakkeet pilkulla erotettuna)(lauseke/taulu)
Valinta	s(ehdot loogisessa muodossa)(lauseke/taulu)
Ristitulo	Taulu/lauseke x taulu/lauseke
Liitos	Taulu/lauseke l(ehdot loogisessa muodossa) taulu/lauseke
Luonnollinen liitos	Taulu/lauseke * taulu/lauseke
Erotus	Taulu/lauseke / taulu/lauseke
Leikkaus	Taulu/lauseke I taulu/lauseke
Yhdiste	Taulu/lauseke U taulu/lauseke

Taulukko 5. Relaatioiden merkinnät

5.2.1 AlgebraParser –luokan metodit (GE)

parse

Koko muuntamisprosessin aloituskohta. Metodi rakennetaan rekursiiviseksi, ja se käyttää luokan kahta muuta metodia hyväkseen. Metodia *split* tämä metodi käyttää erottamaan kaksi saman tason lauseketta toisistaan, ja metodia *parseNaturalJoin* generoimaan luonnollisesta liittoksesta tavallisen liittosta ehtoineen.

Parametrit : Relaatioalgebran lauseke.

Palauttaa : Muunnettu relaatioalgebran lauseke SQL-kyselynä.

split

Erottaa relaatioalgebran lausekkeesta kaksi saman tasoista lauseketta toisistaan, yhdistettynä jollakin operaatiolla, jos lauseke on kahden operaation yhdistelmä. Kahden saman tasoisen lausekkeen välillä voi olla yksi seuraavista operaatioista: yhdiste, leikkaus, ristitulo, liitos, luonnollinen liitos tai erotus.

Parametrit : Relaatioalgebran lauseke

Palauttaa : Merkkijonotaulukko, jossa on joko kolme tai yksi paikkaa. Taulukko on kolmepaikainen, jos parametrin annettu lauseke sisälsi kaksi saman tasoista lauseketta. Tällöin ensimmäisessä taulukon paikassa on ensimmäinen osa lausekkeesta, toisessa on lausekkeiden välinen operaatio ja kolmannessa on toinen lauseke. Jos taulukossa on vain yksi paikka, parametrinä annettu lauseke ei sisältänyt kahta saman tasoista lauseketta.

parseNaturalJoin

Tämä metodi muuttaa luonnollisen liitoksen vastaavaksi ristituloon, valintaan ja projektioon perustuvaksi operaatioiksi ja tuottaa siitä SQL-kyselyn. Metodi käyttää hyväkseen parse -metodia, muuttaakseen luonnollisen liitoksen molemmat osapuolet SQL-kyselyiksi. Tämän jälkeen metodi suorittaa molemmat kyselyt tietokannassa. Tällöin tietokannasta tulevista vastuksista saadaan molempien kyselyiden attribuuttien nimet, joista valitaan sellaiset, jotka sisältyvät molempiin tauluihin. Näiden ja alkuperäisten lausekkeiden avulla rakennetaan vastaava SQL-kysely.

Parametrit : Relaatioalgebran lauseke, joka koostuu kahdesta saman tasoisesta lausekkeesta joita yhdistää luonnollinen liitos -operaatio.

Palauttaa : Parametrinä annetun relaatioalgebran lauseketta vastaavan SQL-kyselyn.

5.3 Ylläpito-operaatioiden harjoitteluväline (ST)

SQL ylläpito-operaatioiden harjoittelemista varten joudutaan muuttamaan OupdateAnswerForm -luokkaa, jota on käytetty SQL -kyselytehtävien toteutuksessa.

5.3.1 OupdateAnswerForm (ST)

Ylläpito-operaatioiden toteuttamista varten tarvitaan ohjelmistoon yksi uusi luokka: OUpdateAnswerForm. Luokka tehdään jo valmiina olevan OAnswerFormin pohjalta lisäten sinne muutamia metodeja, poistaen turhia ja muokaten olemassa olevia.

OUpdateAnswerForm käyttää hyväkseen samoja luokkia kuin OAnswerForm. (ks luku 2.3)

5.3.2 OUpdateAnswerForm metodit (ST)

public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException

Metodi luo käyttöliittymän ja ottaa vastaan siltä välittyvät tapahtumat. Se myös hoitaa SQL-lauseen suorittamisen ja vastaussivun tulostamisen.

public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException

Metodi kutsuu saamillaan parametreilla metodia doPost.

private boolean rowCount(boolean finnish, StringBuffer problems, OTaskInfo tsk, int rcount)

Metodi tarkistaa sen että vastauksen rivien määrä on oikea, jos näin ei ole niin parametrinä annettuun muuttujaan problems laitetaan virheilmoitus ja palautetaan false.

private boolean sumError(boolean finnish, StringBuffer problems, OTaskInfo tsk, int checktotal, String checkable, boolean checkableTypeError)



Metodi vertaa tarkistussummaa ja jos se ei ole oikea niin parametrinä annettuun muuttujaan problems laitetaan virheilmoitus ja palautetaan false.

private boolean validateUpdateStatement(String qry)

Metodi tarkistaa onko kyseessä ylläpito- vai tavallinen sql-lause ja sen että ylläpito kohdistuu vain yhteen tauluun.

private boolean validateSQL(String qry)

Metodi tarkistaa käyttäjän SQL-lauseesta, että hän ei yritä kajota järjestelmän tauluihin.

private String modifyUpdateStatement(String qry)

Metodi muuntaa kyselyn muotoon, jossa se kohdistuu aputauluihin todellisten harjoittelutaulujen sijasta.

private String singleSpaces(String str)

Metodi poistaa merkkijonosta ylimääräiset välilyönnit.

private static String noSpecifiers(String str)

Metodi poistaa merkkijonosta kontrollimerkit.

5.4 Tehtävienlaadinnan työkalujen luokat (JN)

Luokat korvaavat ylliluokkansa metodeista ainoastaan metodit doPost() ja doGet(), jotka toimivat keskenään samalla tavalla. Muuten luokkia käytettäessä käytetään ylliluokan metodeja.

5.4.1 OlistPage (JN)

OlistPage on luokan HttpDBServletOra aliluokka, servlet, joka tulostaa vastaussivulleen listan taulun pstask tehtävistä.

Metodit public void doPost (HttpServletRequest req, HttpServletResponse res) ja public void doGet (HttpServletRequest req, HttpServletResponse res)

Nämä metodit toimivat keskenään identtisesti. Ne hakevat tietokannan taulusta pstopic aihepiirien nimet ja numerot, taulusta pstask jokaisen tehtävän tiedoista tehtävän lyhyen nimen, määräjän ja tunnuksen, ja tulostavat näistä tehtävät aihepiireittäin sivuille. Sivulle tulostetaan jokaista tehtävää kohden painike, jota painamalla voidaan tutustua tehtävänantoon tai muokata sitä. Painikkeen valitseminen lähettää tiedon katsottavasta tehtävästä servletille OTaskDef.

Muutos uuteen versioon: Sivulle tulostetaan myös jokaista tehtävää kohden (lukuun ottamatta ns. uutta tehtävää) painike, jonka painaminen lähettää servletille OTaskDef tiedon, että tehtävän tiedot on poistettava kannasta.

5.4.2 OtaskDef (JN)

OtaskDef on HttpDBServletOra-luokan aliluokka, servlet, joka tulostaa tehtävänantolomakkeen.

Muutokset aiempaan versioon:

Metodit doPost ja doGet:

Jos edelliseltä lomakkeelta tulee tieto, että valittu tehtävä on poistettava, metodi poistaa tehtävän taulusta pstask ja tulostaa operaation onnistumisesta kertovan sivun.

Tulostettava lomake on varsin erilainen kuin aiemmin, sillä uusi järjestelmä laskee lomakkeella annettavasta esimerkkikyselystä valmiiksi ne arvot, jotka ovat laskettavissa.

Aihepiirin valintaa helpotetaan kysymällä pstopic-aulusta aihepiirien nimet ja numerot ja tulostamalla ne valikkoon lomakkeeseen.

Jos käyttäjä editoi jo olemassa olevaa tehtävää, metodi tulostaa myös napin, jota painamalla OtaskRegille annetaan tieto, että valittu tehtävä on poistettava.

Tehtävän Jatka -painike ohjataan OtaskReg-servletille.

5.4.3 OtaskReg (JN)

OtaskReg on täysin uusi luokan HttpDBServletOra aliluokka, se on servlet, joka käsittelee tehtävänannon ja tulostaa tehtävänannonhyväksymislomakkeen.

Metodit doPost ja doGet.

Jos edelliseltä lomakkeelta tulee tieto, että valittu tehtävä on poistettava, metodi poistaa tehtävän taulusta pstask, ja tulostaa operaation onnistumisesta kertovan sivun.

Metodi käyttää tehtäväntyyppin (SQL-kysely, SQL-ylläpito tai relaatioalgebran harjoittelu) mukaista välinettä esimerkkikyselyn suorittamiseen. Jos kyselyssä esiintyy avainsana distinct, me-

todi suorittaa kyselyn kahdesti, kerran ilman distinct-sanaa ja toisen kerran sen kanssa. Metodi laskee kummastakin vastauksesta rivit ja toisesta sarakkeet. Tämän jälkeen se tulostaa hyväksymissivun, jonka alkuun tulee esimerkkikyselyn tulostaulu, josta käyttäjä voi valita, mistä sarakkeesta järjestelmä laskee tarkistussumman, rivi- ja saraketiedot taulusta, ja loppuun edellinen lomake uudestaan. Lomakkeelle tulostuu napit, joista voi joko hyväksyä tehtävän, jolloin se välitetään OTaskStore-servletille, tai uudelleen muokata tehtävää, jolloin se välitetään uudelleen tälle samalle servletille.

5.4.4 OtaskStore (JN)

Tämä on luokan HttpDBServletOra aliluokka, servlet, joka varastoi tietokantaan tehtävänannon tiedot.

Metodit DoPost ja DoGet

Metodi laskee tarkistussumman kutsuvalla lomakkeella valitusta sarakkeesta ja tallentaa tehtävänannon tiedot tauluun pstask.

Tämän jälkeen se tulostaa sivun, joka kertoo suorituksen onnistumisesta, ja jolle tulostuvasta napista pääsee taas tehtävölistaan.

Muutokset aiempaan versioon:

Tauluun on nykyään tallennettava tehtävän tyyppi ja esimerkkikysely.

5.5 Tarkistusluokan parannukset (ST)

Tarkistusluokkaa OAnswerForm parannetaan alla olevan kuvauksen mukaisesti sekä muutetaan järjestelmän kannalta turvallisemmaksi lisäämällä siihen kiellettyjen taulunimien tarkistus.

5.5.1 Suoritettavan SQL:n tarkistus (ST)

Jokaiselle käyttäjältä tulevalle SQL-lauseelle tehdään tarkistus, jossa tutkitaan kohdistuuko lause johonkin systeemitauluun, joita *ovat pscourse, psanswer, psstudent, psstate, pstask, pstopic, pstemp, psterm*, ylläpito-operaatioiden vaatimat aputaulut ja mahdolliset Oraclen tai muun käytettävät tietokantajärjestelmän systeemitaulut.

5.5.2 OAnswerForm (ST)

OAnswerFormiin lisätään metodi, jolla tarkistetaan suoritettava SQL. Tämän lisäksi luokkaan lisätään metodit joilla parannetaan vastauksen oikeellisuuden tarkistamista, siten että tarkistetaan onko vastauksessa kaikki tarpeelliset taulut mukana ja että mitään turhaa taulua ei ole mukana.

private boolean checkNeededTables()

Tarkistaa sen että kaikki tarpeelliset taulut ovat mukana, jos näin ei ole palautetaan false ja annetaan virheilmoitus.

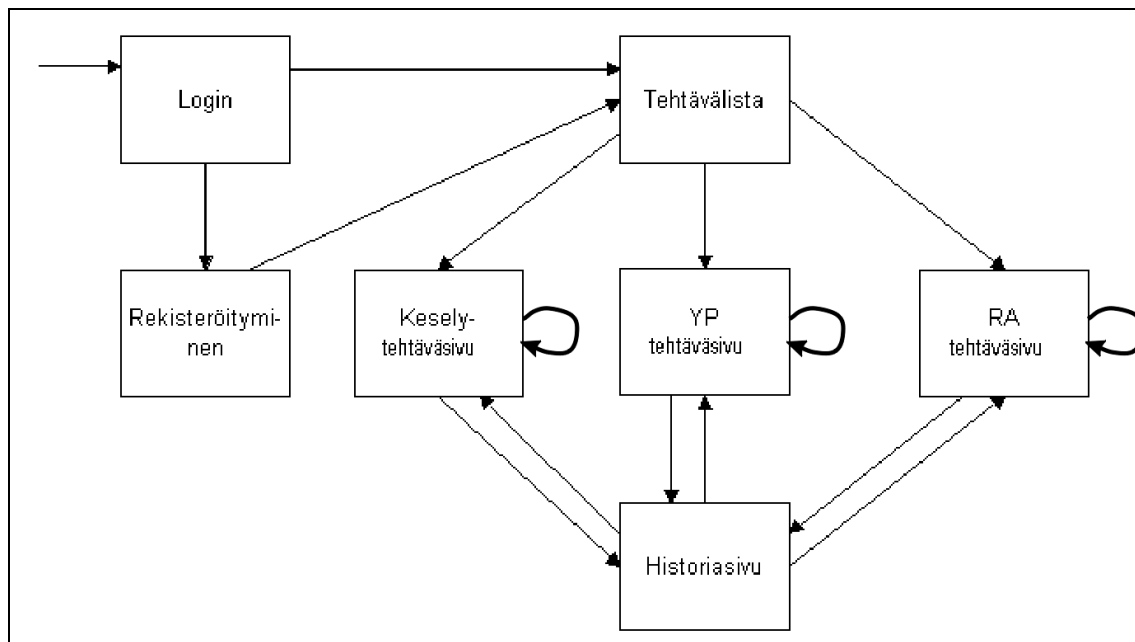
private boolean checkNeedlessTables()

Tarkistaa sen että vastauksessa ei ole tarpeettomia tauluja mukana, jos tarpeettomia on mukana niin palautetaan false ja annetaan virheilmoitus.

6 Käyttöliittymät (KS)

Käyttöliittymäsuunnittelu suoritettiin jo määrittelyvaiheen ohessa. Tässä kappaleessa on esitetty sisällöltään yhteneväinen käyttöliittymäsuunnitelma.

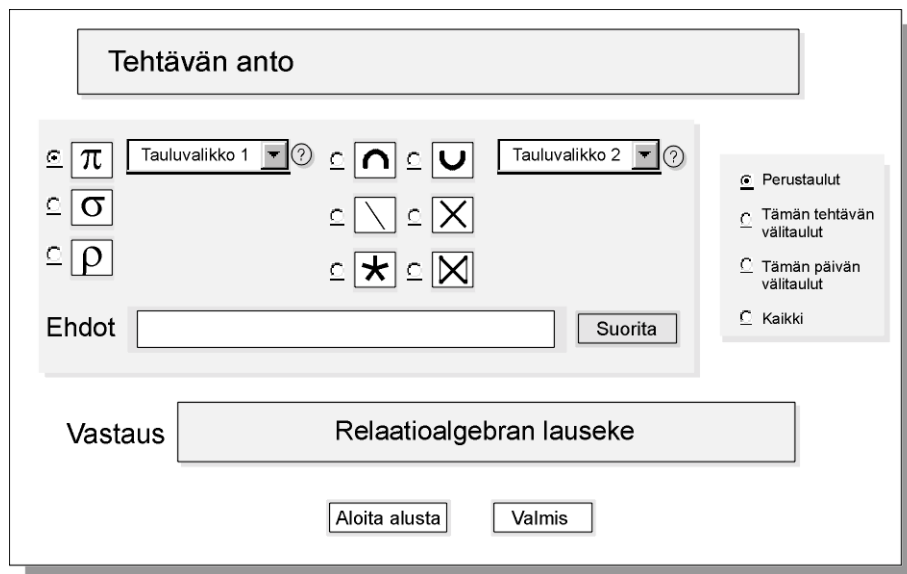
Kuvassa 5 on esitetty laajennetulla toiminnallisuudella varustetun SQL-Trainerin sivusiirtymä-kaavio.



Kuva 5. Käyttöliittymän siirtymäperiaate

6.1 Relaatioalgebran harjoittelu (JN,KS)

Relaatioalgebran harjoittelun sivu on muuten samanlainen kuin SQL-harjoittelun sivu, mutta kysely annetaan tekstikentän sijasta kuvassa 6 näkyvällä käyttöliittymällä.



Kuva 6. Relaatioalgebran harjoittelunäyttö

Käyttö tapahtuu askeleittain siten, että opiskelija valitsee haluamansa operaatioon sekä tarvittavan taulun tai taulut näytöltä vapaassa järjestyksessä. Valintojen jälkeen ”Suorita” –painiketta painamalla kyseinen operaatio suoritetaan. Harjoittelu etenee alkeisoperaatioita yhdistelemällä askel kerrallaan. Kun opiskelija katsoo saavuttaneensa tehtävänannossa pyydetyn tuloksen, hyväksytään näytöllä oleva tulos vastaukseksi ”Valmis” -nappulasta. Tauluvalinnan oikealla puolella olevasta ”?” –nappulasta voidaan tarkastella kyseisestä valintalistasta valitun taulun tai lausekkeen sisältöä. Sisältö avataan omaan ns. Popup –ikkunaan.

Jokainen alkeisoperaation tuottama lauseke tallennetaan tietokantaan SQL –kyselynä. Näin opiskelija voi palata halutessaan minkä tahansa tehtävän välivaiheeseen ja jatkaa tehtävää tai muuttaa saamaansa vastausta. Välivaiheita ei poisteta tietokannasta, vaan ne säilötään. Opiskelija voi valita, mitä välivaiheita tauluvalintalistoiissa näkyy. Välivaiheet tallennetaan harjoittelu- taulukohtaisiin aputauluihin.

6.1.1 Relaatioalgebrakäyttöliittymän toiminta (JN,KS)

Tehtävänanto annetaan suomeksi tai englanniksi samaan tapaan kuin SQL-tehtävissäkin ylläpänä. Operaatiot on jaettu kahteen ryhmään nappeja sen mukaan, koskevatko ne yksittäistä taulua vai ovatko ne kahden taulun välisiä. Täten siis operaatiota ovat näytön vasemmalla puolella projektiio, valinta ja uudelleennimeäminen. Taululistojen välissä olevat operaatiot ovat leikkaus, yhdiste, erotus, ristitulo ja liitos. Operaationapin painallus lähettää tiedon käsittelevälle proseduurille, joka tulostaa samanlaisen sivun, mutta lisää taululistoihin operaation tulosta vastaavan taulun ja muuttaa näkyvää käsiteltävää relaatioalgebran lauseketta tehdyn operaation sisältäväksi. Jos operaatiota ei pystytä suorittamaan, tulostetaan vastaava sivu, jolla kerrotaan myös virheen syystä. Yleensä virhe johtuu siitä, että ehdot-kenttään ei ole laitettu mitään.

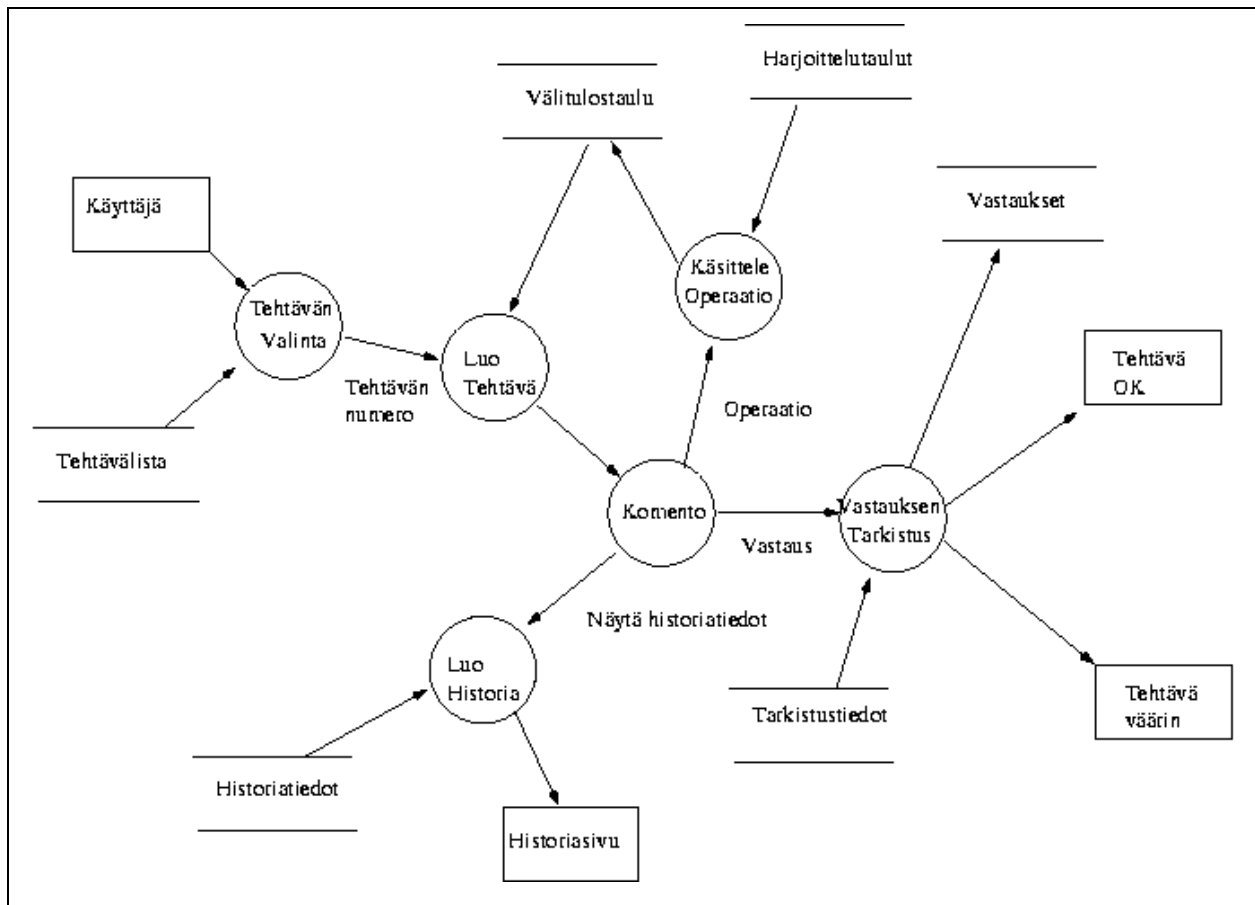
Ehdot-kenttään tulee laittaa operaatioon tarvittava lisätieto. Projektiota tehtäessä listataan haluttu taulun sarakkeet pilkulla erotettuna.

Valintaa tai liitosta tehtäessä annetaan valintakriteeri loogisena lausekkeena, jossa saa esiintyä attribuuttien nimiä, sql:n arvojoukkojen arvoja, vertailuoperaattorit =, <, >, <=, >= ja <> sql:n syntaksin mukaisesti, sekä loogisia operaattoreita not, and ja or. Jos liitoksesta jättää pois ehdon, järjestelmä yrittää luonnollista liitosta.

Kun nimetään sarakkeita uudelleen, uudelleen nimettävät sarakkeet annetaan muodossa sarakkeen nimi = uusi nimi, pilkulla erotettuna.

Kaikki käyttäjän saamat välitulokset ovat valittavissa menulistasta (Tauluvalikot) jatkooperaatioita varten. Näytön oikeassa reunassa olevat valintanapit on tarkoitettu suodattamaan listoja. Niillä voi valita näkykö listoissa vain tehtävien perustaulut, vain tämän tehtävän välitaulut, vain välitaulut, jotka on saatu samana päivänä kuin kysely vai kaikki käyttäjän koskaan tekemät välitaulut. Välitulokset esitetään listassa relaatioalgebrana jos toteutusympäristön merkistö sen sallii.

Kuvassa 7. on esitetty relaatioalgebran käyttöliittymän sisäinen toimintamalli.



Kuva 7. Relaatioalgebran käyttöliittymän tietovuokaavio

6.1.2 Käyttöliittymän virheilmoitukset (JN)

Virheilmoitukset näytetään käyttäjälle selkokieლისenä. Virheilmoituksia ovat:

"Projektiossa ei annettu haluttujen sarakkeiden nimiä. Projektiolla voi valita relaatiosta nähtäväksi halutut sarakkeet. Sarakkeet pitää antaa Ehdot-kohtaan pilkulla erotettuna."

Tämä tulee, jos Ehdot -kohta on jätetty projektiossa tyhjäksi. Jos Ehdot-kohtaan laitetaan jotain muuta kuin järkeviä sarakkeiden nimiä, tämä ilmenee lausetta suoritettaessa tietokantavirheenä.

"Valinnassa ei annettu perusteita rivien valinnalle. Valinta -operaatiolla voi relaatiosta valita tietyn ehdon täyttävät rivit. Ehto pitää antaa Ehdot-kohtaan loogisena lausekkeena, kannattaa tutustua käyttöohjeeseen."



Tämä tulee jos Ehdot-kohta on jätetty tyhjäksi valinnassa. Taaskaan, jos Ehdot-kohtaan laitetaan jotain järjetöntä, tämä ei välttämättä ilmene heti.

"Uudelleennimeämisoperaatiolla voi antaa sarakkeille uudet nimet. Ne pitää antaa muodossa Sarakkeen nimi=uusi nimi, pilkulla erotettuna."

Jos jättää uudelleennimeämisoperaatiosta pois, mitä nimetään miksikin, saa tämän virheilmoituksen.

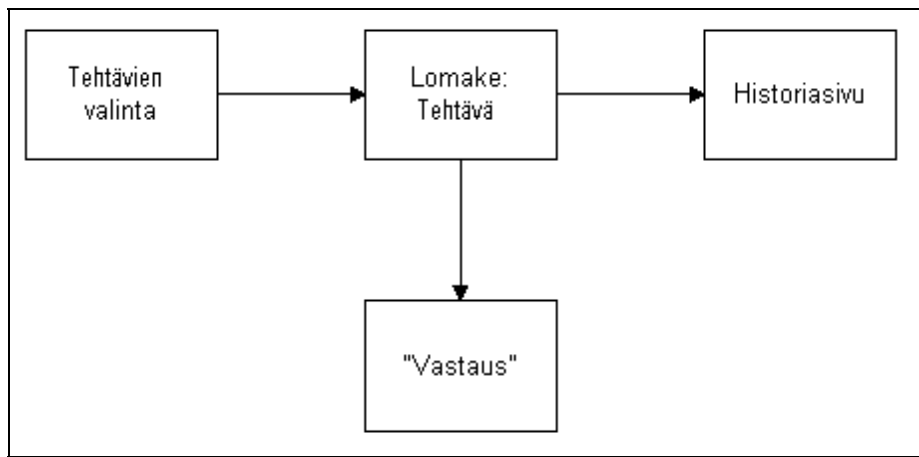
"Kohdetaulu pitää valita. Valittu operaatio tarvitsee taulun, jolle se tehdään. Taulun voi valita vasemmanpuoleisesta listasta."

"Valittu operaatio on kahden taulun välinen operaatio. Kummastakin valikosta on valittava taulu, jotta operaatio voitaisiin suorittaa."

Yllä esitetyistä virheilmoituksista näytetään sopivampi, jos on jätetty jompikumpi tauluista pois operaatio-painiketta painettaessa.

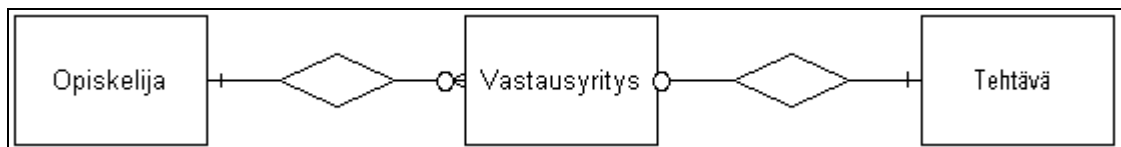
6.2 Ylläpito-operaatioiden käyttöliittymä (ST, KS)

SQL-Trainer järjestelmää laajennetaan kattamaan INSERT, UPDATE ja DELETE SQL-lauseet. Kyseisiä operaatioita harjoitellaan samalla käyttöliittymällä kuin SQL -tehtäviä. Toiminnallisuus toteutetaan olemassa olevia toimintoluokkia laajentamalla sekä uusia lauseita käsittävien toimintojen lisäyksellä. Käyttöliittymän sivujen siirtymäperiaate käy ilmi kuvassa 8.



Kuva 8. Ylläpito-operaatioiden harjoittelun käyttöliittymän siirtymäkaavio

Kuten nykyjärjestelmässä, ylläpito-operaatiotehtävissä voidaan jättää useita vastausyrityksiä, jotka tallennetaan tietokantaan.



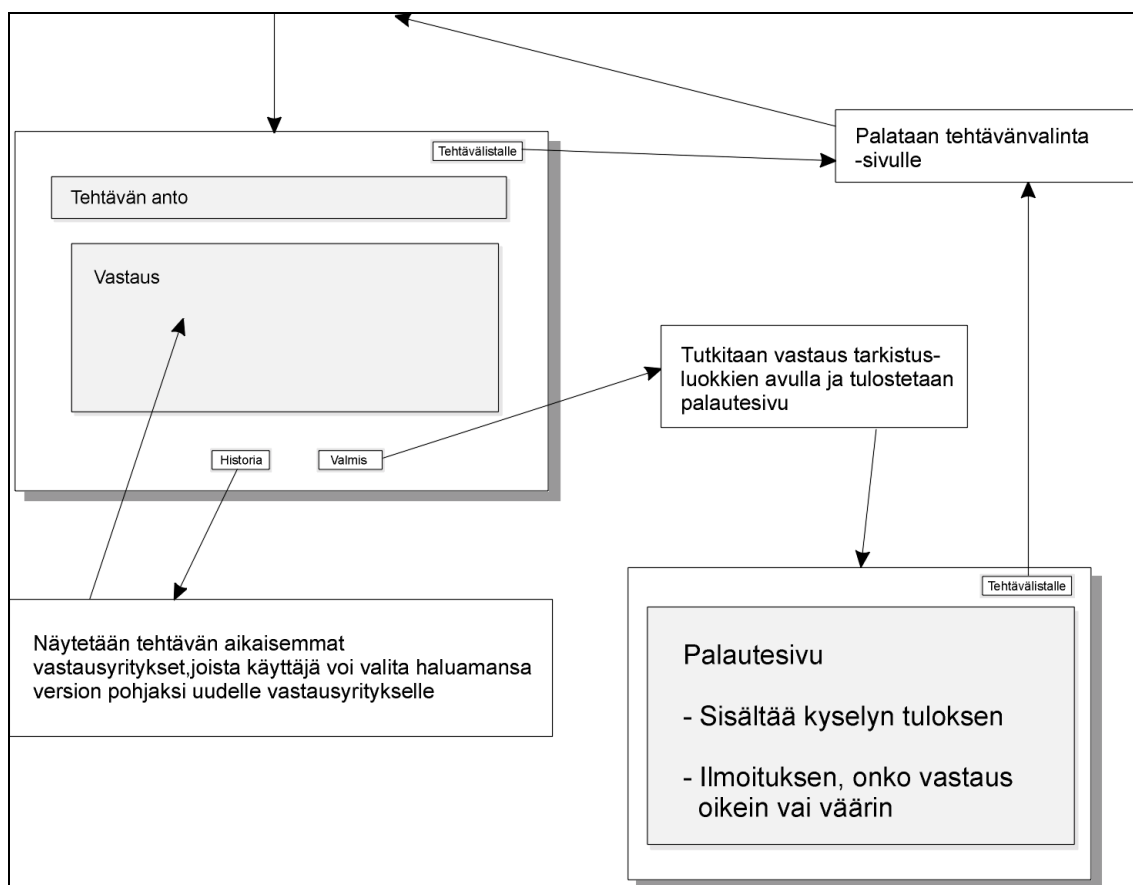
Kuva 9. Ylläpito-operaatioiden harjoittelun vastaus/tehtävä suhdekaavio

Ylläpito-operaatiot toteutetaan siten, että varsinaisiin harjoittelutauluihin ei tehdä muutoksia, vaan muutokset tallennetaan tehtäväkohtaiseen aputauluun (ks. luku 4.2). Aputauluun tallennetaan lisätty, muutettu tai poistettu rivi lisättynä opiskelijan tunnistetiedoilla. Kun opiskelija on antanut vastauksensa hän saa nähtäväkseen alkuperäisen taulun kera aputaulusta löytyvien muutosten. Aputaulussa pidetään muuttunutta tietoa yllä vain siihen asti, että muutos tulostetaan opiskelijan nähtäväksi. Tämän jälkeen taulusta poistetaan opiskelijalle kuuluvat rivit.

Koska mitään muutoksia ei kohdisteta alkuperäisiin tauluihin, opiskelijan tekemät tehtävät kohdistuvat aina tauluihin, jotka on lähtötilanteessa samanlaisia. Tämä mahdollistaa sen, että operaatioita voidaan kohdistaa mihin tahansa järjestelmän tauluun.

6.2.1 Ylläpito-operaatioiden toiminnallinen määrittely (ST,KS)

Ylläpito-operaatioiden harjoittelemista varten oleva sivusto toimii periaatteeltaan samalla tavoin kuin jo olemassa oleva SQL harjoittelun sivusto. Kuvassa 10. on esitetty ylläpito-operaatioiden käyttöliittymän periaate.

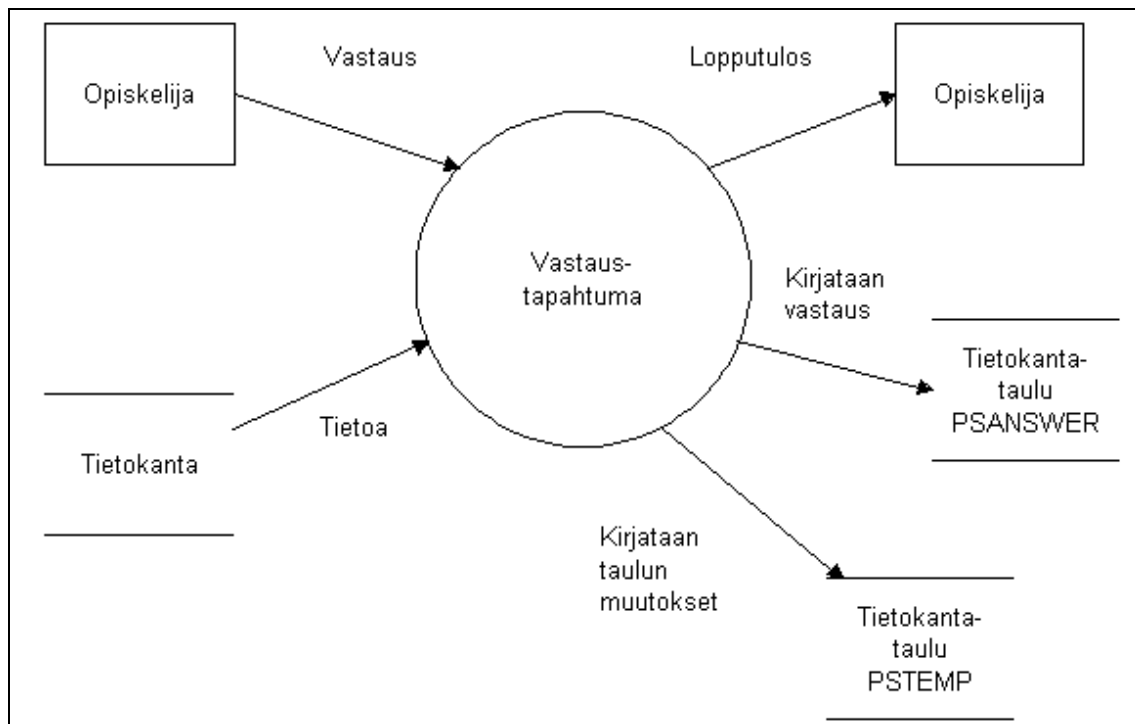


Kuva 10. Ylläpito-operaatioiden harjoittelun käyttöliittymäperiaate

Tehtävä anto -sivun alussa annetaan tehtävä, joko suomeksi tai englanniksi riippuen käyttäjän asetuksista. Tehtävän alla on tekstikenttä, jonne opiskelija kirjoittaa vastaukseksi kyselyn, jonka on tarkoitus tuottaa haluttu ylläpito-operaatio. Valmis -painike lähetetään kysely eteenpäin luokille, jotka tarkistavat sen oikeellisuuden ja sen että tuottiko kysely tehtävässä halutun vastauksen. Lisäksi sivulla on historia -painike, jolla käyttäjä pääsee tarkastelemaan aiempia ratkaisuyrityksiään, ja takaisin tehtävälistaan -painike joka vie käyttäjän takaisin tehtävälistaan.

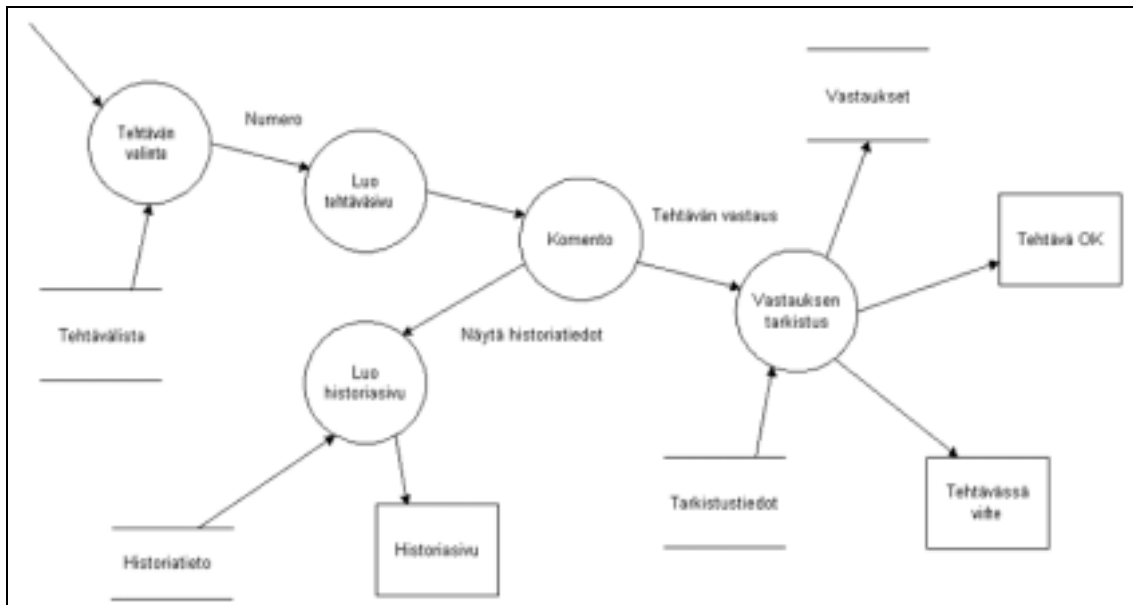
Valmis -painikkeen tuottamalla palautesivulla tulostuu mahdolliset virheilmoitukset, jos sellaisia on. Virheilmoitukset tulevat joko sql-trainerin luokilta tai suoraan tietokannasta. Lisäksi sivulla näkyy taulu ennen ja jälkeen ylläpito-operaation. Jos vastaus oli väärin sivulla on painike, jolla päästään takaisin tehtävän anto - tai historia -sivulle. Sivulla on myös painike, jolla päästään takaisin tehtävälistaan.

Historia -painike luo sivun, jolta voidaan SQL-puolen tapaan valita joku aiemmista vastauksista pohjaksi uuteen yritykseen. Lisäksi sivulla on painikkeet, joilla päästää takaisin tehtävän anto -sivulle ilman, että mitään tehtävää otetaan pohjaksi ja painike jolla päästään takaisin tehtävälistaan. Käyttöliittymän yleinen toimintaperiaate on esitetty tietovuona kuvassa 11.



Kuva 11. Ylläpito-operaatioiden harjoittelun käyttöliittymän tietovuokaavio

Edelleen, vastaustapahtumaprosessia tarkennetaan kuvassa 12.

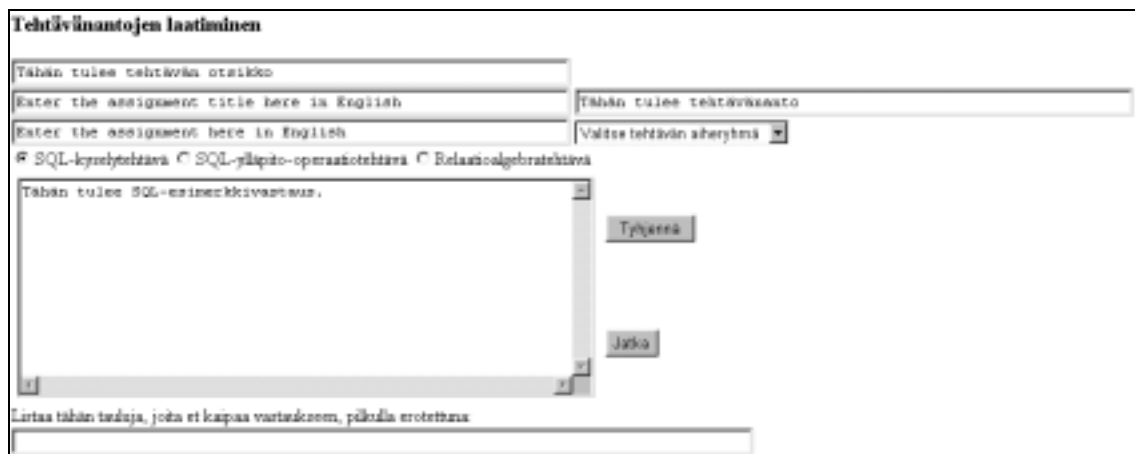


Kuva 12. Tarkennettu ylläpito-operaatioiden harjoittelun käyttöliittymän tietovuokaavio

6.3 Tehtävienlaadinnan käyttöliittymän määrittely (JN,KS)

SQL-Trainer –tehtävien laadintaa helpotetaan lisäämällä järjestelmään käyttöliittymä tehtävänlaadintaan.

Tehtäviä voidaan antaa erityisen salasanalla suojattavan WWW-käyttöliittymän kautta. Tehtävä annetaan kuvassa 13. esitetyllä lomakkeella. Käyttäjältä vaaditaan tehtävän otsikko, kuvaus, aiheryhmä, tehtävätyyppi (joku vaihtoehdoista SQL-kysely, SQL-ylläpitemehtävä tai relaatioalgebra-kysely) valintanapeista ja esimerkkivastaus SQL-kyselynä samaan tapaan kuin tehtäviä tehdessä. Jopa relaatioalgebran tehtävät annetaan SQL-kyselyinä.



Kuva 13. Tehtävänlaadinnan käyttöliittymä

Kun käyttäjä suorittaa esimerkkikyselyn, sen tulostaulu tulostetaan sivun alalaitaan samalla tavalla kuin tehtävien tekosivuilla, mutta jokaista saraketta kohden tulostuu painike, jota painamalla voi hyväksyä tehtävän tehtäväjoukkoon. Tilanne on esitetty kuvassa 14. Valitusta sarakeesta järjestelmä laskee sitten tarkistuskoodin. SQL-Trainerin tietokannan tauluun *pstask* tallennetaan hyväksyttäessä tarvittavat tiedot tulostaulusta, tarkistuskoodi, tehtävän tyyppi, tehtävänannot, otsikot ja annettu esimerkkikysely.

Tehtävänantojen laatiminen

Tähän tulee tehtävän otsikko

Enter the assignment title here in English

Enter the assignment here in English

Tähän tulee tehtävänanto

Valitse tehtävän aiheryhmä

☒ SQL-kyselytehtävä ☐ SQL-ylläpito-operaatiotehtävä ☐ Relatioalgebratehtävä

Tähän tulee SQL-esimerkivastaus.

Tyhjennä

Valmis

Listaa tähän tauluja, joita et kaipaa vastaukseen, pilkulla erotettuna:

Valitse tarkistussummasarake

tabid	tabname	nrows
664	spi_huomio	1
663	sportsinfo_huomio	1
46	sysaggregates	0
40	sysams	2
31	sysattrtypes	1030
100	sysblderrorlog	45
105	sysbldipprovided	2
348	sysbldirequired	0
102	sysbldobjdepends	153
101	sysbldobjects	148

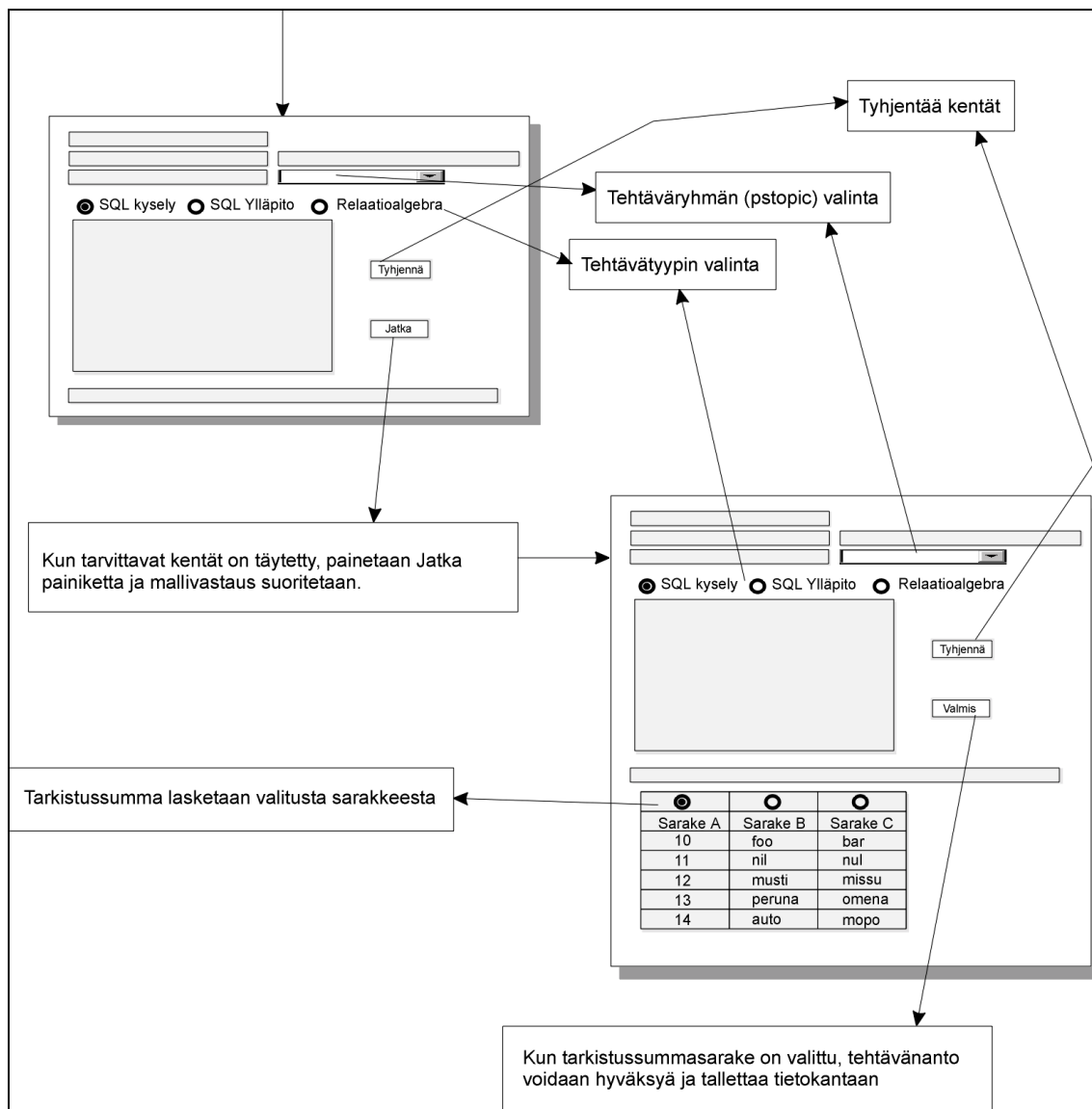
Kuva 14. Tehtävänlaadinnan käyttöliittymä tarkistussummavalinnalla

Jos esimerkkikyselyssä on jotain virheitä, järjestelmä antaa tulostaulun sijaan tietokantajärjestelmän virheilmoituksen. Jos kyselyssä esiintyy tauluja, joita tehtävänannon mukaan ei tulisi esiintyä, tästäkin tulostetaan virheilmoitus.

Tehtävän voi jättää myös hyväksymättä ja lomakkeen saa tyhjennettyä tyhjennä-napista, jonka jälkeen voi syöttää uuden tehtävän.

6.3.1 Tehtävänlaadinnan käyttöliittymän toiminnallinen määrittely (JN,KS)

Kuvissa 13. ja 14. esitellyn käyttöliittymän toiminnallisuus esitellään alla.



Kuva 15. Tehtävänlaadinnan käyttöliittymän toiminnallinen määrittely

7 Testaus (GE, JN, ST)

Testaussuunnitelma on jaettu kolmeen osaan toteutettavien kokonaisuuksien mukaan.

7.1 Relatioalgebran harjoitteluvälineen testaus (GE)

Relatioalgebran harjoitteluvälineen testaus jakautuu kahteen osaan, AlgebraParser-luokan testaukseen ja käyttöliittymän testaukseen. AlgebraParser-luokan testauksella varmistutaan siitä, että luokka generoi relatioalgebran lausekkeesta oikean SQL-kyselyn, joka tuottaa saman vastauksen. Relatioalgebran harjoitteluvälineen käyttöliittymätestauksella varmistutaan siitä, että AlgebraParser-luokkaa käytetään oikein.

7.1.1 AlgebraParser-luokan testaus (GE)

AlgebraParser-luokan testaus jakautuu kolmeen osaan. Koska luokassa oleva metodi parse käyttää hyväkseen luokan kaksi muuta metodia ja koska luokan kaksi muuta metodia ei missään tapauksessa luokan ulkopuolelta kutsuta, testaus perustuu metodin parse testaukseen ja sellaisiin testitapauksiin jotka aiheuttavat metodissa parse metodien split ja splitNaturalJoin kutsumista. Nämä testitapaukset ovat parse-metodin parametrejä. Testitapauksia kirjoitettaessa oletettiin, että tietokannassa on taulut taulu1 ja taulu2, jotka sisältävät kumpikin vain attribuutit eka ja toka.

Testi nro.	Parametrit	Odotettava vastaus
1	Taulu1	taulu1
2	p(eka,toka)(taulu1)	Select eka,toka from taulu1
3	s(eka=10)(taulu1)	Select * from taulu1 where eka=10
4	Taulu1 U taulu2	(Select * from taulu1) Union (select * from taulu2)
5	Taulu1 I taulu2	(select * from taulu1) Intersect (select * from taulu2)
6	Taulu1 \ taulu2	(Select * from taulu1) minus (Select * from taulu2)
7	Taulu1 l(eka=toka) taulu2	Select * from taulu1,taulu2 where eka=toka
8	Taulu1 * taulu2	Select eka, toka from (select eka,toka from taulu1), (Select eka as eka1, toka as toka1 from taulu2) where eka=eka1, toka = toka1
9	P(eka,toka)(s(eka=toka)(taulu1 x taulu2))	Select eka,toka from (select * from taulu1,taulu2 where eka =toka)
10	s(eka=1 and toka=2)(taulu1 l(eka<toka) taulu2)	Select * from (Select * from taulu1,taulu2 where eka<toka) where eka=1 and toka=2

Taulukko 6. Testitapaukset

7.1.2 Relaatioalgebran harjoitteluvälineen käyttöliittymän testaus (JN)

Yksityiskohtaisemmat testitapaukset kuvataan tarkemmin projektin loppudokumentissa.

Unaariset operaatiot

Unaaristen operaatioiden testauksen tarkoituksena on selvittää, että operaatiot toimivat määrittelyssä annetulla tavalla.

Operaation testaaminen: valitaan tauluvalikosta taulu, ja sille tehtävä operaatio, kirjoitetaan ehdot niille tarkoitettuun tekstikenttään ja painetaan välitulospainiketta. Odotettava tulos on se, että operaation tulostaulu ilmestyy tauluvalikkoon, ja sen tuottava relaatioalgebran lauseke vastauspaikalle. Taulun sisällön oikeellisuus on tarkistettavissa katsomalla sen sisältö.

Operaation testaaminen virheellisin lähtöarvoin: tarkoitus on katsoa, että väline toimii myös virheellisillä syötteillä. Tämä testataan muuten samalla menetelmällä kuin yllä, mutta käydään jokaiselle operaatiolle läpi testitapaukset, joissa ehdoissa on jotain vikaa tai ne puuttuvat, tai sitten kohdetaulua ei ole määritetty. Näiden toimenpiteiden tulee johtaa virheilmoitukseen ja virhetilanteeseen. Tähän testijoukkoon kuuluu myös testi, jossa yritetään ottaa välitulos tekemättä mitään valintoja.

Binääriset operaatiot

Binääriset operaatiot vastaavat unaarisia muuten mutta vaativat kaksi taulua toimiakseen. Niille tehdään vastaavanlainen testijoukko.

Taululistojen suodattaminen

Taululistojen suodatuksen testaamisen tarkoituksena on katsoa, että tauluvalikossa näkyvät kerrallaan vain halutut taulut. Tämä testataan valitsemalla jokainen vaihtoehtoista ja tarkistamalla taululistoista mitä niissä sitten on.

Alusta alkaminen

Alusta alkamisnapin painamisen tuloksen pitäisi olla se, että tekeillä oleva lauseke nollautuu pois. Tämä testataan painamalla painiketta kesken jonkin lauseen rakentamisen. Lisäksi testataan mitä käy, kun painetaan aloita alusta - painiketta ilman lauseketta. Tästä ei pitäisi käydä mitään.

Lopullisen vastauksen antaminen

Testien tarkoitus on selvittää toimiiko väline lopulta. Testit tehdään painamalla valmis - painiketta. Jos mitään lauseketta ei ole tekeillä tai valittuna, mitään ei pitäisi tapahtua. Muussa tapauksessa järjestelmän tulisi suorittaa lauseke ja tulostaa tulostaulu mahdollisine virheilmoituksineen.

7.2 Ylläpito-operaatioiden testaus (ST)

Ylläpito-operaatioiden muutokset kohdistuvat yhteen luokkaan OUpdateAnswerForm, joten se on ainoa luokka joka vaatii testausta. Luokkaan tulee muutama uusi metodi ja olemassa oleviin hieman muutoksia.

Koska luokan muuttuminen alkuperäisestä toimivasta luokasta on varsin pientä, niin sille voidaan tehdä sekä peruspolkutestaus, että ehtotestaus.

Toiminnallista testausta luokalle tehdään antamalla testaamalla, että kaikki eri sivut tuottavat painikkeet tekevät halutut sivut ja koittamalla erilaisia SQL-lauseita, joista osan pitäisi toimia ja osan ei.

7.3 Tehtävänantovälineiden testaussuunnitelma (JN)

Tehtävän poisto

Tehtävä voidaan poistaa järjestelmästä kahdesta paikasta, joten kumpaakin on testattava. Napin painalluksen pitäisi poistaa tehtävä kannasta ja tulostaa sivu, jossa kerrotaan näin tapahtuneen. Poisto on varmistettavissa kannasta erikseen katsomalla. Kannattaa huomata, että tehtävän poistumisesta ei kysytä mitään.

Tehtävän lisääminen

Tehtävän lisäämisen testauksen tarkoituksena on selvittää tulevatko tehtävän tiedot kantaan oikein. Testaus suoritetaan antamalla tehtävänantoja ja tarkkailemalla kannasta, että ne myös saatiin oikein sisään. Myös epätäydellisesti täytettyjen lomakkeiden aiheuttamat virheet testataan, tuloksena pitäisi olla virheilmoitus eikä mitään muutoksia kantaan.

Tehtävän muokkaaminen

Tehtävän muokkaamisen testaaminen muistuttaa tehtävän lisäämisen testaamista melkein täydellisesti. On kuitenkin tarkistettava erikseen, että tehtävä todellakin korvautuu uudella. Tehtävä ei korvaudu millään, jos lomakkeet on täytetty epätäydellisesti.



Lähteet

- Maar00 Laine Harri, SQLTR-ryhmä kesä2000, Määrittelydokumentti 27.6.2000
- Sun00 SUNin Java-tyyliopas. <http://java.sun.com/docs/codeconv/> (16.8.2000).
- Html00 World Wide Web -konsortion standardi. <http://www.w3.org/TR/WD-html40/> (16.8.2000).