

Määrittelydokumentti
SQL trainer - tietokantaharjoitteluohjelmiston laajennus
Versio 0.1

Versiohistoria

VERSIO	PÄIVÄYS	TEKIJÄ	SELITE (muutokset jne.)
0.1	1.2.2000	Mika Norismaa (MN)	1. iteraatio

Sisällys

VERSIONHISTORIA	2
SISÄLLYS	3
1 JOHDANTO (MN).....	4
1.1 TARKOITUS JA KATTAVUUS	4
1.2 TUOTE JA YMPÄRISTÖ	4
1.3 VIITTEET	4
2 NYKYJÄRJESTELMÄN KUVAUS	4
2.1 YMPÄRISTÖ	5
2.2 TOIMINTA.....	5
2.3 KÄYTTÄJÄT.....	5
3 PROJEKTIN TAVOITE JA TARKOITUS	5
3.1 TOTEUTETTAVAT OMINAISUUDET	5
3.2 LISÄOMINAISUUDET (TULEVAT ITERAATIOT).....	6
4 TIETOKUVAUS	6
4.1 SIDOSRYHMÄT	6
4.2 TIETOSISÄLTÖ (TULEVAT ITERAATIOT).....	7
4.2.1 Yksilötyypit ja ilmenemismuodot (Tulevat iteraatiot).....	7
4.2.2 Opiskelijan yhteystyypit (Tulevat iteraatiot).....	7
4.2.3 Opettajan yhteystyypit (Tulevat iteraatiot).....	7
4.2.4 Ylläpitäjän yhteystyypit (Tulevat iteraatiot).....	7
4.3 KÄYTTÖINTENSITEETTI JA KAPASITEETTIVAATIMUKSET (TULEVAT ITERAATIOT).....	7
5 TOIMINTOKUVAUS.....	8
5.1 JÄRJESTELMÄN YLEISARKKITEHTUURI (TULEVAT ITERAATIOT)	8
5.2 KOMPONENTIT (TULEVAT ITERAATIOT)	8
6 JÄRJESTELMÄN ULKOISET LIITYMÄT.....	8
6.1 KÄYTTÖLIITYMÄ.....	8
6.2 MUUT LIITYMÄT.....	8
7 MUUT OMINAISUUDET (TULEVAT ITERAATIOT)	9
7.1 SUORITUSKYKY (TULEVAT ITERAATIOT)	9
7.2 VIRHEISTÄ TOIPUMINEN (TULEVAT ITERAATIOT).....	9
7.3 YLLÄPITO (TULEVAT ITERAATIOT)	9
8 TESTAUS	9
9 RAJOITTEET SUUNNITTELULLE JA TOTEUTUKSELLE	9
9.1 NOUDATETTAVAT STANDARDIT	9
9.2 LAITTEISTORAJOITTEET	9
9.3 OHJELMISTORAJOITTEET	9
10 JATKOKEHITYSAJATUKSIA (TULEVAT ITERAATIOT).....	10
10.1 MÄÄRITELTÄVIÄ LISÄPIIRTEITÄ (TULEVAT ITERAATIOT).....	10

1 Johdanto (MN)

Tämän dokumentti rakennetaan siten, että projektin jokaisessa iteraatiossa syntyy versio, jota täydennetään seuraavassa versiossa. Tämä käytäntö johtuu siitä, että se paitsi kuvaa projektin etenemistä myös palvelee dokumentoinnin tarkoitusta parhaalla mahdollisella tavalla. Yleiset osat ovat jokaisen iteraation dokumenteissa samat mutta muita osia toteutetaan sitä mukaa kun siirrytään iteraatiosta seuraavaan. Ne osat, jotka mainitaan tässä dokumentoinnissa mutta kuuluvat seuraavien iteraatioiden toteutukseen on merkitty tähän dokumenttiin (Seuraavat iteraatiot) -kommentilla.

1.1 Tarkoitus ja kattavuus

Työn tavoitteena on tehdä laajennuksia tietokantakyselyiden harjoitteluun tarkoitettuun SQL trainer ohjelmistoon. Ohjelmiston kohderyhmänä on korkeakouluopiskelijat ja opettajat, viiteryhmänä on Helsingin yliopiston tietojenkäsittelytieteen laitos. Projekti on jatkoa kesällä 2000 toimineelle projektityhmälle, joka laajensi lehtori Harri Laineen suunnittelemaa ja toteuttamaa alkuperäistä versiota SQL trainerista.

1.2 Tuote ja ympäristö

SQL trainer on WWW-ympäristössä toimiva SQL-kyselyiden ja relaatioalgebran harjoitteluväline. Sen avulla opiskelijalle esitetään joukko esimerkkietokantaan perustuvia tehtäviä. Opiskelija laatii ratkaisun (joka siis on SQL-kysely) helppokäyttöisellä WWW-lomakkeella ja lähettää sen tarkastettavaksi. Tarkistus hoidetaan automaattisesti. Harjoittelutyökalun tavoitteena on mm. mahdollistaa paikkariippumaton etäharjoittelu ilman käyttäjätunnuksia, lisätä käytännön harjoittelun määrää, antaa opiskelijoille henkilökohtaista palautetta ratkaisuista myös englanniksi ja siten helpottaa korkeakoulutasoista tietokantakyselyiden opiskelua.

1.3 Viitteet

KS00a	Sinkko Kimmo, SQL trainer, Projektisuunnitelma, 2000
KS00b	Sinkko Kimmo, SQL trainer, Määrittelydokumentti, 2000
KS00c	Sinkko Kimmo, SQL trainer, Suunnitteludokumentti, 2000
KS00d	Sinkko Kimmo, SQL trainer, Loppudokumentti, 2000
HL00a	Laine Harri, SQL trainer, 2000
WWW00	World Wide Web –konsortion kotisivu. Http://www.w3.org
SUN00	SUNin Java-tyyliopas

2 Nykyjärjestelmän kuvaus

Nykyjärjestelmällä tarkoitetaan kesän 2000 projektin tekemää laajennusta alkuperäiseen SQL traineriin. Tämän laajennuksen sisältö ilmenee kesän 2000 projektin dokumentoinnista. Nykyjärjestelmän vaatimusten määrittely on kuvattu määrittelydokumentissa [KS00b].

Vaatimusmäärittelyä on tarkennettu suunnitteludokumentissa [KS00c], jossa on kuvattu myös järjestelmän arkkitehtuuri. Loppudokumentista [KS00d] ilmenee kuinka vaatimusmäärittely muuttui ja toteutui projektin toteutumisvaiheessa.

2.1 Ympäristö

Järjestelmä toimii asiakas-palvelin -periaatteella Internetin tai intranetin kautta. Toteutusympäristönä on Linux-palvelinympäristö. Palvelimena on Jserv 1.0-servlettimoottorilla varustettu Apache WWW-palvelin. Tietokantapalvelimena käytetään tietojenkäsittelytieteen laitokselle asennettua Java DataBase Connection (JDBC) -ajurilla varustettua Oracle 8 -tietokantaa.

Järjestelmä on pyritty toteuttamaan mahdollisimman laitteistoriippumattomaksi. Tämän vuoksi käyttäjät voivat olla siihen yhteydessä käyttämällä mitä tahansa käyttöjärjestelmää, johon kuuluu selain, kunhan luvussa 9 esitetyt laitteisto- ja ohjelmistorajotteet otetaan huomioon.

2.2 Toiminta

SQL trainer tarjoaa mahdollisuuden SQL -kyselyiden ja relaatioalgebran harjoitteluun Internetin välityksellä. Järjestelmään sisältyy välineet joilla opiskelija voi suorittaa erilaisia tehtäviä kohdistuen esimerkkietokantaan. SQL trainer tarkastaa opiskelijan tekemät tehtävät ja antaa palautetta virheilmoitusten, tulostaulujen ja ohjeiden muodossa.

2.3 Käyttäjät

Järjestelmää voidaan käyttää SQL-kyselyiden ja relaatioalgebran harjoitteluun mistä tahansa Internetin välityksellä. Käyttäjät kuuluvat kolmeen käyttäjäryhmään. Nämä ovat opiskelija, opettaja ja ylläpitäjä. Käyttäjän käyttöoikeudet vastaavat käyttäjäryhmää. Käyttäjäryhmien ja järjestelmän välisiä suhteita on kuvattu tarkemmin luvussa 4. Tietokuvaus.

3 Projektin tavoite ja tarkoitus

Projektin tavoitteena on laajentaa ja kehittää kesällä 2000 toteutettuja järjestelmän ominaisuuksia. Lisäksi on tarkoitus toteuttaa seuraavassa luvussa esitellyt piirteet ja tämän määrittelyn yhteydessä tulleet ideat.

Projektin asiakas Harri Laine on esittänyt ryhmälle listan tavoiteluontoisista ohjelmiston piirteistä, jotka on jaettu tässä yhteydessä kahteen tasoon:

1. ne piirteet ja korjaukset, jotka aiotaan toteuttaa kevään 2001 kuluessa
2. ne piirteet ja korjaukset, jotka toteutetaan jos ehditään.

Tässä luvussa esitellään erityisesti piirteet, jotka toteutetaan kevään 2001 kuluessa. Lisäksi esitellään siis piirteitä, jotka toteutetaan jos ehditään.

3.1 Toteutettavat ominaisuudet

Tässä esitellyt ominaisuudet toteutetaan kevään 2001 aikana.
(Ensimmäisen iteraation tavoite)

SQL trainerin nykyversion puutteiden korjaus. Puutteiden tai virheellisen toiminnan korjaus suoritetaan projektin ensimmäisen iteraatiokierroksen aikana. Puutteista korjataan olennaisimmat ensin ja loput jos ehditään. Joka tapauksessa puutteiden korjaaminen

keskeytetään iteraatiokierrokselle varatun ajan umpeutuessa, jotta päästään siirtymään aikataulun mukaisesti projektissa eteenpäin.

(Tulevat iteraatiot)

Ohjelmiston yleiskäyttöisen osan ja tapauskohtaisen osan eristäminen. Tämän osan tavoitteena on luoda mahdollisuus määritellä erilaisia tehtävätyyppejä esimerkiksi sql-kyselyt, relaatioalgebra, monivalintakysymykset jne. Joilla olisi kullakin tyyppikohtainen kysymys- ja vastausnäyttönsä ja analysaattorinsa. Uuden tehtävätyypin määrittelyn yhteydessä annettaisiin tieto vastausnäytön generoijasta ja analysaattorista. Tämä aiheuttaa muutoksia sekä traineriin että tehtävien määrittelytyökaluun. Luonnollisesti SQL trainerin nykytoiminnot muunnetaan uuteen arkkitehtuuriin sopiviksi.

(Tulevat Iteraatiot)

Uuden esimerkkietokanta ja tehtäväjoukko. Nykyinen esimerkkietokanta on liian suppea, jotta ohjelmistoa kyettäisiin edes testaamaan kattavasti. Tästä syystä sekä esimerkkietokanta että siihen liittyvät tehtävät on laadittava uudelleen.

(Tulevat Iteraatiot)

Yleiskäyttöinen työkalu suoritustietojen imurointiin. Suorituspisteet tulee saada muodossa, josta voidaan tuottaa erilaisia latausmuotoja eri tarkoituksiin. Esimerkiksi sql:n update lauseet tktl:n kurssikirjanpitoon tai avoimelle yliopistolle. Tietyn opiskelijan tilanne tulee myös pystyä tarkastamaan samoin hänen vastauksiaan tulee pystyä selailemaan.

3.2 Lisäominaisuudet (Tulevat iteraatiot)

Seuraavassa on lueteltu piirteitä, jotka toteutetaan kevään 2001 kuluessa jos ehditään.

Piirteet ovat lähinnä puutteita ohjelmiston nykyversiossa.

Käyttöliittymän parantaminen.

SQL-ilmoitusten hiominen.

4 Tietokuvaus

Järjestelmän tietosuunnittelussa määritellään sidosryhmäkaavio, tietosisältö ja tietokantamalli. Tietokannan mallinnuksessa käytetään ER-mallia.

4.1 Sidoryhmät

Järjestelmän käyttäjäkuntaan sisältyvät opiskelijat, opettajat sekä ylläpitäjät. Opiskelijat, opettajat ja ylläpitäjät muodostavat vastaavat käyttöoikeusryhmät.

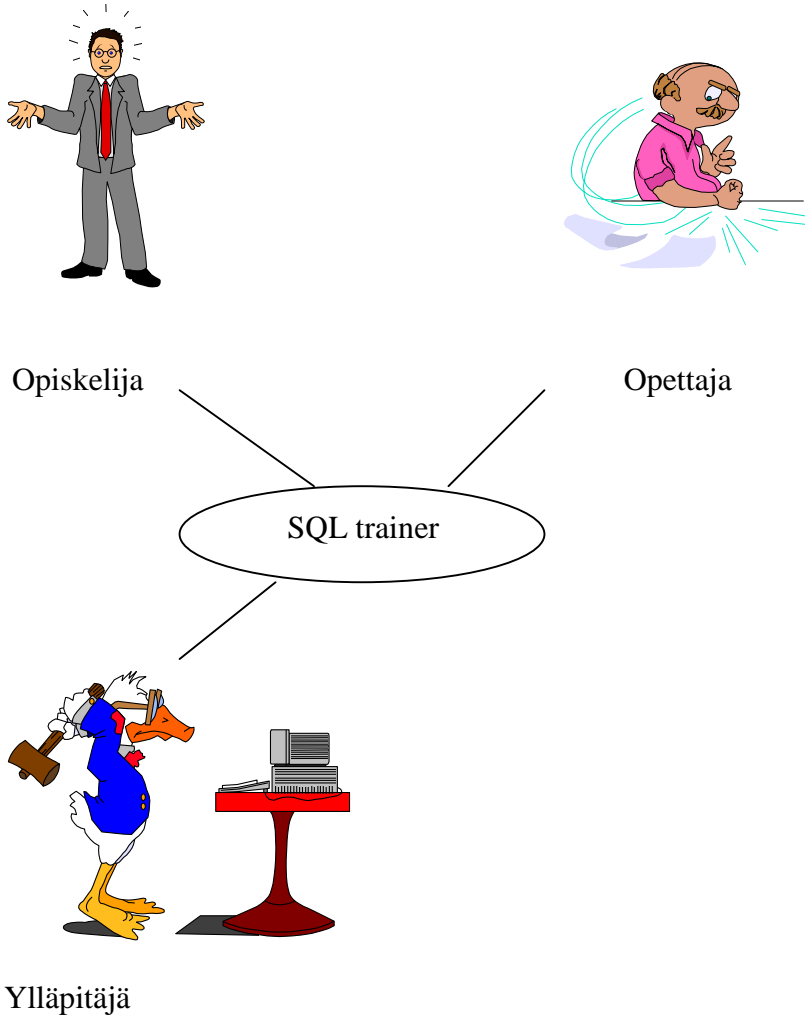
Opiskelijalla tarkoitetaan ketä tahansa internetin kautta ohjelmistoa käyttävää henkilöä.

Opettaja tarkoitetaan henkilöä, joka opettaa tietojenkäsittelytiedettä tai vastaavaa SQL traineria käyttäen.

Ylläpitäjällä tarkoitetaan henkilöä, jonka vastuulla on järjestelmän käyttäjien valtuushallinta.

Valtuushallinta sisältää käyttäjätietojen muuttamisen, selailun ja poiston. Lisäksi ylläpitäjä pitää huolta järjestelmän teknisistä asioista.

Seuraava kuva havainnollistaa järjestelmän ja sidoryhmien välisiä suhteita.



Kuva 1. Järjestelmä ja sen sidosryhmät.

4.2 Tietosisältö (Tulevat iteraatiot)

4.2.1 Yksilötyypit ja ilmenemismuodot (Tulevat iteraatiot)

4.2.2 Opiskelijan yhteystyypit (Tulevat iteraatiot)

4.2.3 Opettajan yhteystyypit (Tulevat iteraatiot)

4.2.4 Ylläpitäjän yhteystyypit (Tulevat iteraatiot)

4.3 Käyttöintensiteetti ja kapasiteettivaatimukset (Tulevat iteraatiot)

5 Toimintokuvaus

Järjestelmän arkkitehtuurisuunnittelu esittää toteutettavan ohjelmiston modulaarisena kokonaisuutena, josta ilmenee korkean tason elementtien väliset suhteet. Arkkitehtuurin mallinnuksessa noudatetaan UML-standardia. Tässä luvussa esitellään vain järjestelmän toiminnan yleisperiaatteet, yksityiskohtaisempi kuvaus esitetään suunnitteludokumentissa.

5.1 Järjestelmän yleisarkkitehtuuri (*Tulevat iteraatiot*)

5.2 Komponentit (*Tulevat iteraatiot*)

6 Järjestelmän ulkoiset liittymät

Järjestelmän liittymäsuunnitelma kuvaa, miten ohjelmiston ulkoiset liittymät toimivat. Liittymäsuunnittelu esittelee toteutettavan ohjelmiston käyttöliittymän sekä laitteisto-, ohjelmisto- ja tietoliikenneliittymät.

6.1 Käyttöliittymä

Tässä kappaleessa kuvataan vain hyvin lyhyesti ohjelmiston käyttöliittymää. Tarkempi kuvaus on esitetty kesän 2000 projektin määrittely- ja suunnitteludokumenteissa [KS00b, KS00c].

SQL trainer ohjelmiston käyttöliittymä on muodostettu HTML-elementeistä ja se perustuu yhteen näyttöön jossa pyritään esittämään kaikki tapahtumat. Opiskelija laatii ratkaisunsa tehtäviin helppokäyttöisellä HTML-lomakkeella ja vastaus sekä ohjeet tulostetaan automaattisesti samaan näyttöön.

6.2 Muut liittymät

Ohjelmisto toimii asiakas-palvelin -periaatteella Internetin tai intranetin kautta. Ohjelmisto ja tietokanta sijaitsevat palvelimella, johon käyttäjät ovat yhteydessä WWW-selaimen avulla.

Asiakaskoneessa eli työasemassa istunto käynnistetään syöttämällä WWW-palvelimen osoite selaimelle.

Palvelimessa sijaitsee järjestelmän palvelinsovellus, joka toimii samankaltaisella periaatteella kuin yleisesti käytössä olevat CGI-skriptit. CGI eli Common Gateway Interface on yksi tapa liittää ulkopuolisia ohjelmia web-palvelimeen. Tämä saadaan aikaan myös JSP-tekniikkaan (Java Server Pages) perustuvilla palvelinsovelluksilla. Palvelinsovellus käyttää Oracle 8 tietokantaa JDBC-ajurin kautta, joka huolehtii tietokantayhteyksistä.

Työaseman ja palvelimen väliseen tietoliikenteen hoitamiseen käytetään laitteita, joilla saadaan Internet-yhteys aikaan. Näiden välineiden yksityiskohtainen selostus kuuluu modeemin, tietokoneen, näytön jne. Asennusohjeisiin. Niitä ei kuvata tarkemmin tässä dokumentissa.

Vaikka työasema ja palvelin sijaitsevat yleensä eri koneissa, voidaan ohjelmistoa käyttää myös palvelinkoneelta normaalisti selaimen avulla.

7 Muut ominaisuudet (Tulevat iteraatiot)

7.1 Suorituskyky (Tulevat iteraatiot)

7.2 Virheistä toipuminen (Tulevat iteraatiot)

7.3 Ylläpito (Tulevat iteraatiot)

8 Testaus

Järjestelmä on testattava huolellisesti. Tässä projektissa erityisesti jo ensimmäisen iteraation aikana testataan ohjelmiston nykyversiota siitä syystä, että iteraatiossa on tarkoitus korjata nykyisen version puutteet ja tutustua ohjelmistoon. Tässä vaiheessa testausta suoritetaan em. syistä johtuen vain tietyiltä osin nykyversioon perustuen. Täysin kattava testaus suoritetaan myöhempien iteraatiokierrosten aikana.

9 Rajoitteet suunnittelulle ja toteutukselle

9.1 Noudatettavat standardit

Arkkitehtuurin mallinnuksessa noudatetaan UML-standardia ja tietokannan mallinnuksessa ER-mallia

Java-koodi kirjoitetaan Java-tyylioppaan [SUN00] mukaisena. HTML-koodi noudattaa HTML Specification v4.0 Transitional -standardia [WWW00]. Ohjelmakoodin muuttujien nimeämisessä ja kommentoinnissa sekä dokumentoinnissa käytetään suomen kieltä.

9.2 Laitteistorajoitteet

Järjestelmä toimii asiakas-palvelin -periaatteella Internetin tai intranetin kautta. Työaseman ja palvelimen välisessä tietoliikenteen hoitamisessa käytetään laitteita, joilla Internet-yhteys saadaan aikaan.

Asiakaskoneessa eli työasemassa Pentium tasoinen kone on riittävä tuotteen käyttöön.

Työaseman monitorin resoluution tulee olla vähintään 800x600 pikseliä.

Palvelimena tulee olla keskitasoinen palvelinkone, jossa on vähintään 64 megatavua keskusmuistia ja normaalikokoinen kiintolevy (~15Gt). Arvio on ylimalkainen ja voi tarkentua jatkossa.

9.3 Ohjelmistorajoitteet

Ohjelmisto toteutetaan Linux-ympäristössä, mutta Java-teknologian ansiosta ohjelmistoa voi käyttää myös muissa Javaa tukevilla käyttöjärjestelmissä. Työaseman ja palvelimen

käyttöjärjestelmä voi olla Windows 95/98, Windows ME, Windows 2000, Windows NT tai Linux.

Tuote toimii Internet Explorer ja Netscape Navigator -selaimissa alkaen niiden nelosversioista.

10 Jatkokehitysajatuksia (Tulevat iteraatiot)

10.1 Määriteltäviä lisäpiirteitä (Tulevat iteraatiot)