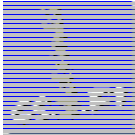


SISÄLLYS

1 YLEISTÄ (KS)	1
1.1 PROJEKTIN HENKILÖT	1
1.2 PROJEKTIMENETELMÄ JA KÄYTETTÄVÄT TYÖKALUT	1
2 SQL-TRAINER - KUVAAUS NYKYISESTÄ JÄRJESTELMÄSTÄ (HL,KS)	2
2.1 OHJELMAN TOIMINTA	2
2.2 OHJELMAN TEKNIikka	3
2.3 OHJELMAN KAAVIOITA	5
3 JATKOKEHITYKSEN VAATIMUKSET (HL,KS)	7
3.1 RELAATIOALGEBRAN HARJOITTELU	7
3.2 TIETOKANNAN YLLÄPITO-OPERAATIOT	8
3.3 TEHTÄVIENLAADINNAN JA TARKASTAMISEN APUVÄLINEET	8
4 PROJEKTIN TEHTÄVÄT (KS,JN,ST,GE)	9
4.1 RYHMÄJAKO	10
4.2 VASTUUT	10
5 AIKATAULU JA TARKISTUSPISTEET (KS)	11
6 DOKUMENTIN KIRJOITTAJAT (KS)	12

LIITTEET	SQL-Trainer tietokannan luonti-SQL
	Projektikaavio



1 Yleistä (KS)

Projekti on tietojenkäsittelytieteen Cum laude approbatur -opiskelukokonaisuuteen kuuluva pakollinen kurssi. Projekti toteutetaan kesän 2000 aikana ja sen on tarkoitus jatkokehittää SQL-trainer ohjelmistoa.

Tietokantojen perusteet -kurssilla on ollut syksystä -99 käytössä laskuharjoitukset korvaava harjoitteluohjelmisto (<http://www.cs.helsinki.fi/u/laine/tikape/ftrainer.html>). Harjoitteluohjelmisto antaa käyttäjille joukon ongelmia, jotka heidän on ratkaistava SQL-kyselyinä. Ohjelmisto kirjaa käyttäjien vastaukset, analysoi ne ja antaa palautetta vastauksen oikeellisuudesta. Oikein ratkaistut tehtävät tuottavat suorituspisteitä. Harjoitteluohjelmiston tietokantana oli syksyllä 99 Solid-kanta ja keväällä 2000 Oracle-kanta. Ohjelmisto on toteutettu Java-servlet tekniikalla.

1.1 Projektin henkilöt

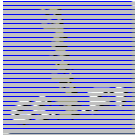
Projektiin osallistuvat George El-Khouri, Kimmo Sinkko, Jaakko Nurro ja Santtu Toivakka. Projektin vetäjänä toimii Indrek Peri ja työn tilaajana Harri Laine.

Projektiryhmien esimiehenä toimivat Anni Rytönen ja Juha Taina.

1.2 Projektimenetelmä ja käytettävät työkalut

Projektissa käytetään lineaarista vesiputous -projektimallia. Ohjelmointi tehdään Java -kielellä ja lähdekoodit talletetaan CVS -versiohallintajärjestelmään. Dokumentoinnissa käytetään Microsoft Word 97 tekstinkäsittelyohjelmaa ja tarvittavia graafisia apuvälineitä, kuten vuokaaviopiirto-ohjelmia.

Tämän projektin puitteissa tehtävä Java -koodi kommentoidaan Javadoc -järjestelmällä, mutta Harri Laineen aiemmin laatimaa koodia ei kommentoida, ellei siihen tehdä muutoksia.



2 SQL-trainer - Kuvaus nykyisestä järjestelmästä (HL,KS)

Tietokantojen perusteet -kurssilla on ollut syksystä -99 käytössä laskuharjoitukset korvaava harjoitteluohjelmisto (<http://www.cs.helsinki.fi/u/laine/tikape/ftrainer.html>). Harjoitteluohjelmisto antaa käyttäjille joukon ongelmia, jotka heidän on ratkaistava SQL-kyselyinä. Ohjelmisto kirjaa käyttäjien vastaukset, analysoi ne ja antaa palautetta vastauksen oikeellisuudesta. Oikein ratkaistut tehtävät tuottavat suorituspisteitä. Harjoitteluohjelmiston tietokantana oli syksyllä 99 Solid-kanta ja keväällä 2000 Oracle-kanta. Ohjelmisto on toteutettu Java-servlet tekniikalla.

2.1 Ohjelman toiminta

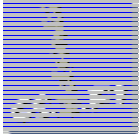
Ohjelmalla voi harjoitella SQL-kyselyjä. Ohjelma tarjoaa käyttöliittymän, jonka kautta opiskelija voi antaa kyselyn ja saa tuloksena kyselyn vastauksen sekä tietoa kyselyyn liittyvistä ongelmista. Ohjelman käyttöliittymä on sekä suomen- että englanninkielinen. Opiskelijalla on käytettävissään joukko tauluja (englanninkielisiä). Taulujen rakenne ja sisältö kuvataan avustustiedostossa. Järjestelmä voi käyttää mitä tahansa käyttäjälle *harkka* näkyviä tauluja.

Kirjoittautuessaan järjestelmään opiskelija valitsee itselleen käyttäjätunnuksen ja salasanan. Opiskelija voi halutessaan kirjata henkilötietonsa. Jos hän ei kirjaa henkilötunnustaan ei tuloksia voi kytkeä kurssikirjanpitoon. Seuraavilla käyttökerroilla opiskelija pääsee antamalla käyttäjätunnuksensa ja salasanasensa siihen tilanteeseen, johon hän edellisellä kerralla päätyi.

Järjestelmä esittää käyttäjälle listan tehtävistä. Listassa näkyvät yrityskertojen lukumäärä ja onko käyttäjä onnistunut jo ratkaisemaan tehtävän. Listalta pääsee vastaamaan tehtävään. Tehtäviin voi vastata missä järjestyksessä tahansa. Samaa tehtävää voi yrittää ratkaista useita kertoja. Käyttäjän tehtävään antama vastaus tarkastetaan ja kirjataan tietokantaan. Annetun vastauksen voi valita pohjaksi uuteen vastausyritykseen joko samalle tai eri tehtävälle.

Tarkistuksessa

1. Suoritetaan kysely.
2. Tutkitaan onko kysely syntaktisesti oikein (tkhj tarkastaa)
3. Tutkitaan ovatko kaikki pakolliset sarakkeet mukana
4. Tarkastetaan esiintyykö sarake sisällöllisesti toisteisena
5. Tarkastetaan onko vastauksessa liikaa sarakkeita
6. Tarkastetaan onko vastauksen rivimäärä oikea
7. Tarkastetaan onko tarkistussumma oikein



8. Tarkastetaan onko vastauksen järjestys oikein (tarkastetaan kyselyä analysoimalla)
9. Tarkastetaan ovatko kaikki tarpeelliset taulut mukana (ei vielä toteutettu)
10. Tarkastetaan onko kyselyssä tarpeettomia tauluja (ei vielä toteutettu)

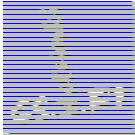
Käyttäjälle annetaan syntaksivirheestä palautteena tkhj:ltä saatu virheilmoitus. Muista virheistä annetaan virhekohtaiset omat ilmoitukset. Jos kysely tuotti tulostaulun se esitetään käyttäjälle.

2.2 Ohjelman tekniikka

Järjestelmä käyttää seuraavia systeemitauluja:

- pscourse; (kurssitiedot, ei käytössä)
- psanswer; (kirjatut vastaukset)
- psstudent; (opiskelijoiden tiedot)
- psstate; (vastaus laskurit)
- pstask; (tehtävätiedot)
- pstopic; (tehtävähyhmätiedot)

Järjestelmän käytön kannalta keskeinen on taulu **pstask**. Siinä annetaan tehtävä ja kuvaillaan oikea vastaus. Tehtävä annetaan sekä suomeksi että englanniksi.



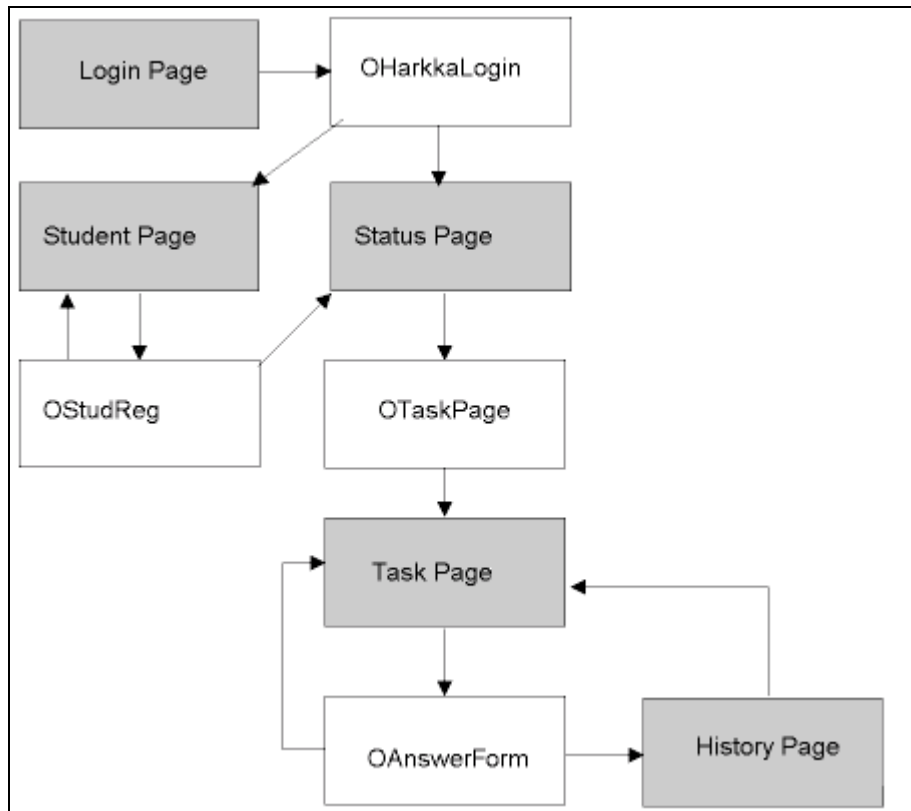
Vastauksen kuvailuun liittyvät seuraavat tiedot:

ok_rows integer,	Oikean vastauksen koko riveinä
Nonunique integer,	Vastauksen koko, jos välttämätön unique määre on jäänyt pois
ok_cols integer,	Vastauksen sarakkeiden enimmäismäärä
ok_checkable varchar(32),	Tarkistussummasarakkeen nimi
ok_checkvalue integer,	Oikeasta vastauksesta laskettu tarkistussumma
Ordered char,	Onko järjestettävä (ei ole käytössä)
Neededcols varchar(120),	Välttämättömien sarakkeiden luettelo, Sarakenimet annetaan pilkulla eroteltuina
Eqpairs varchar(120),	Identtiset sarakeparit (sarake1=sarake2). Tällaisia ei ole syytä ottaa mukaan tulokseen.
Included varchar(120),	Kyselyyn sisältyvä merkkijono jota käytetään järjestysmääreen tarkistukseen. Ennen vertailua kyselystä karsitaan turhat välilyönnit ja tarkentimet.
col_max integer,	Vastauksen sarakkeiden määrä (tämä on tarpeeton)
Needed_tabs varchar(120),	Välttämättömien taulujen luettelo, taulunimet pilkulla eroteltuna. Tuottaa virhe ilmoituksen jos jokin näistä puuttuu
Needless_tabs varchar(120),	Tarpeettomien taulujen luettelo. taulunimet pilkulla eroteltuina. Tuottaa virheilmoituksen, jos jokin näistä on mukana

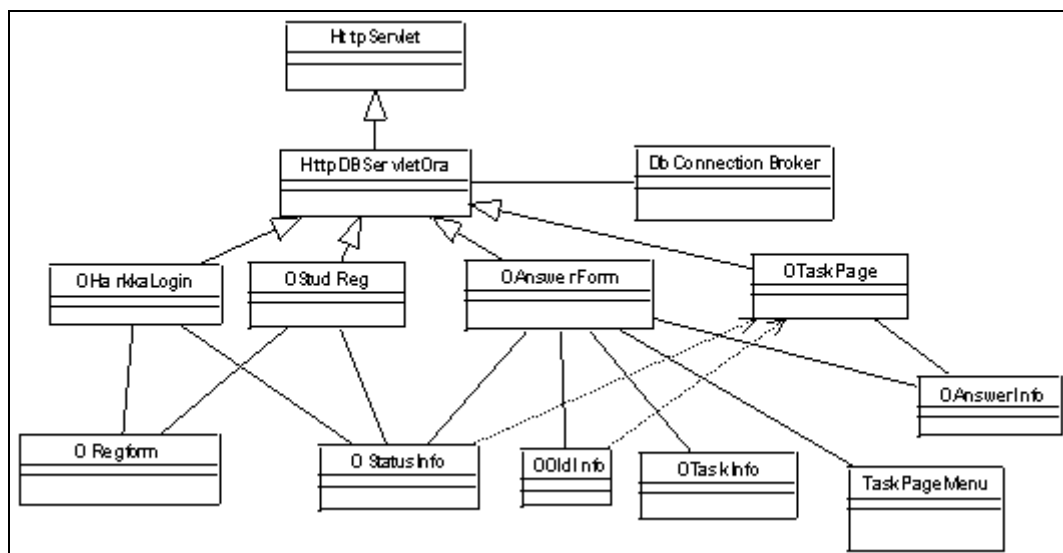
Kaavio 1. Taulun pstask sarakkeet

Tehtävät näytetään tehtävänumerojärjestyksessä.

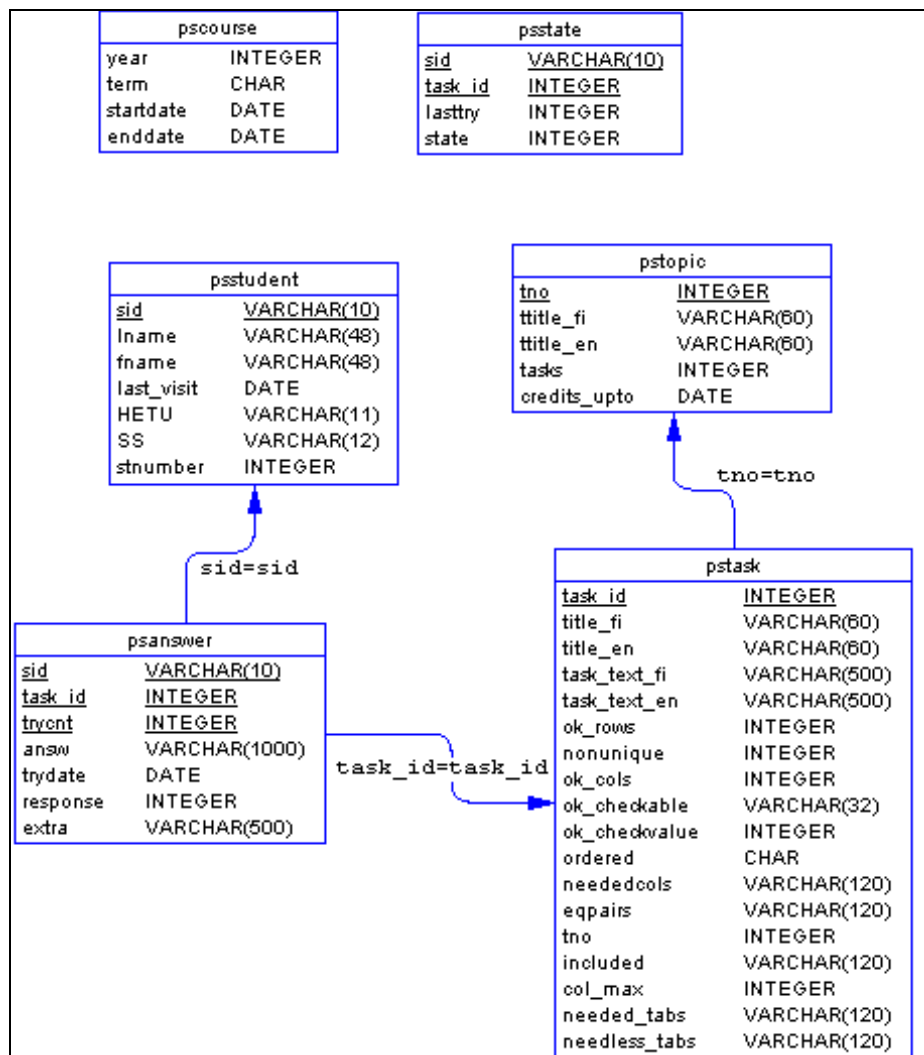
2.3 Ohjelman kaavioita



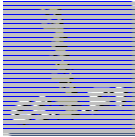
Kaavio 2. Nykyohjelman sivurakenne



Kaavio 3. Nykyohjelman luokkarakenne (osittain)



Kaavio 4. Nykyohjelman tietokantarakenne



3 Jatkokehityksen vaatimukset (HL,KS)

Kesän 2000 ohjelmistotuotantoprojektissa laajennetaan SQL-trainer -ohjelmistoa siten, että pelkkien kyselyiden lisäksi sillä voidaan harjoitella myös relaatioalgebraa ja SQL:n ylläpito-operaatioita. Relaatioalgebran harjoitteluosuus toteutetaan alkeisoperaatioina, jotka konvertoitetaan SQL-operaatioiksi. Alkeisoperaatioiden tulokset kirjataan tilapäistauluihin ja niitä voidaan käyttää osina laajempia operaatioita.

3.1 Relaatioalgebran harjoittelu

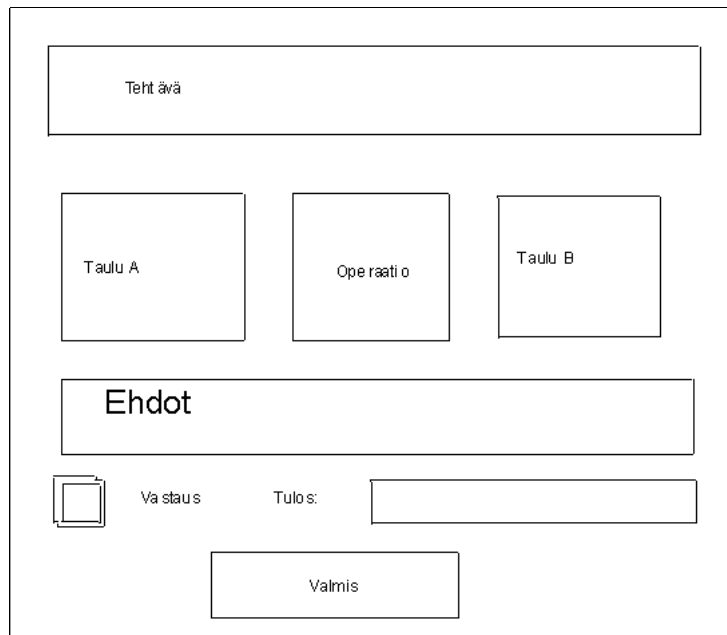
Relaatioalgebraa harjoitellaan käyttämällä samoja tauluja kuin SQL-harjoittelussa. Harjoittelu voisi perustua alkeisoperaatioihin = relaatioalgebran perusoperaatiot. Alkeisoperaatio tuottaa tulostaulun.

Kyselyn tulostaulu määritellään näkymänä, jonka nimen osana on käyttäjän tunnus. Käyttäjä rakentaa kyselynsä osina alkeisoperaatioiden ja niiden tulostaulujen avulla.

Esim.:

- käytössä taulut: t_1, t_2, t_3
- käyttäjä määrittelee aputaulun $t_4 = t_1 * t_2$
- käytössä taulut t_1, t_2, t_3, t_4
- käyttäjä määrittelee aputaulun $t_5 = \text{Projektio}(A, B)(t_4)$
- Käyttäjä voi tulostaa minkä tahansa taulun.
- Käyttäjä määrittelee jonkin taulun, vaikka t_5 , tehtävän vastaukseksi
- Vastaus analysoidaan samoin kuin SQL-kyselyjen vastaukset

Käyttäjä voisi valita operaation osapuolet valintalistosta samoin kuin operaation. Tulostiedoston nimi ja operaatioon liitettävä ehto annetaan tekstikentissä. Muista kuin perustauluista pitäisi saada näkyviin paitsi nimi myös relaatioalgebranlauseke.



Kaavio 5. Hahmottelua käyttöliittymästä

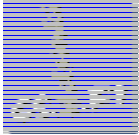
3.2 Tietokannan ylläpito-operaatiot

SQL-trainerissa ei nykymuodossa voi harjoitella ylläpito-operaatioita. Jos muutoksia kohdistetaan kyselyissä käytettäviin tauluihin eivät kyselyihin liittyvät vastausten kuvailutiedot enää täsmää. Ylläpito pitäisi siis kohdistaa muihin tauluihin.

Ylläpidon pitäisi kohdistua aina samanlaiseen lähtötauluun, jolloin jollakin standardikyselyllä voitaisiin varmistua siitä onko se muuttunut oikein. On vielä avoin ongelma miten tämän voisi tehdä.

3.3 Tehtävienlaadinnan ja tarkastamisen apuvälineet

Nykyjärjestelmässä joudutaan tehtäviä laadittaessa suorittamaan manuaalista tarkistussummien laskemista. Lisäksi tehtävien tarkistamisessa käytettävä algoritmi saattaa antaa oikean vastauksen tehtävälle, vaikka sen tulos on väärä. Siksi tarvitaan työkaluja, joilla käsin tehtävästä työstä päästään eroon ja tarkistusmenetelmä on saatava tarkemmaksi.



4 Projektin tehtävät (KS,JN,ST,GE)

Tehtävät jakautuvat neljään selvään kokonaisuuteen:

- **Relaatioalgebra**

Relaatioalbegan opetusosassa toteutetaan uusi käyttöliittymän osa, jolla kaikkia relaatioalgebran perusoperaatiota voidaan harjoitella. Käyttöliittymää varten toteutetaan toimintoluokat, jolla toiminnallisuus toteutetaan. Käyttöliittymässä esitetään käyttäjän valitsema operaatio selkokielisenä relaatioalbegan lauseena.

- **Tietokantaluokat**

Olemassa olevia tietokantaluokkia hyväksikäyttäen toteutetaan tarpeellinen määrä laajennuksia toteuttamaan uusien toimintojen tarvitsemat tietokantaoperaatiot. Ajan salliessa korjataan nykyversiossa esiintyviä ongelmia mm. avoimiksi jäävien kursorien osalta.

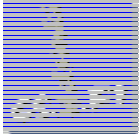
- **Ylläpito-operaatiot**

Laajennetaan nykyistä SQL -operaatioiden valikoimaa insert, update ja delete -käskyillä. Käskyjen implementoinnissa käytetään hyväksi olemassa olevia menetelmiä niitä laajentaen. Operaatiot kohdistetaan yhteen järjestelmän uuteen tietokantatauluun, joka luodaan ja jota varten suunnitellaan tarvittavat käsittelyluokat.

Lisätään järjestelmään luokka, joka hyväksyy ainoastaan sallitut SQL -operaatiot ja hylkää järjestelmän kannalta epäilyttävät operaatiot kokonaan.

- **Tarkistustyökalut**

Suunnitellaan ja toteutetaan parempi tehtävän vastauksen tarkistamismenetelmä, sekä tehdään apuvälineitä tehtävien laadintaan. Tarkoituksena on vähentää käsin tehtävän työn osuutta mallivastausten ja tarkistustoimenpiteiden valmistelun yhteydessä. Lähtökohtana on mallivastauksen syöttö järjestelmään, josta talletetaan tunnistetiedot (rivit, sarakkeet ja checksum). Mallivastusta ei talleteta.



4.1 Ryhmäjako

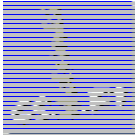
Edellä mainittuja kokonaisuuksia työtetään seuraavan ryhmäjaon pohjalta:

Relaatioalgebra, käyttöliittymä	Jaakko Nurro, George El-Khoury
Ylläpito-operaatiot, tarkistustyökalut	Kimmo Sinkko, Santtu Toivakka
Tietokantaluokat	Kaikki

4.2 Vastuut

Ryhmissä tehtävien töiden lisäksi projektissa on erikoisvastuita seuraavilla henkilöillä:

Projektipäällikkö	Santtu Toivakka
CVS versionhallinta	George El-Khoury
Dokumentointi HTML-muodossa	Kimmo Sinkko
Dokumentoinnin konvertointi tulosteiksi	Santtu Toivakka



5 Aikataulu ja tarkistuspisteet (KS)

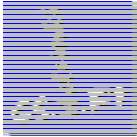
Projektin elinkaari alkaa OHTU -ryhmän järjestäytymisen jälkeen ja päättyy 30.8.2000 mennessä. Käytännössä projekti on jo käynnistetty ja valmistumisajankohdaksi on asetettu 15.8.2000. Tämän ajankohdan jälkeen on varattu kahden viikon liukuma-aika, jos tarvetta projektin pitkitämiselle ilmenee. Toinen mahdollinen liukuma-aika on ohjelmoinnin lopettamisen ja testausten aloittamisen välissä siten, että testaukselle varattua aikaa voidaan lyhentää ja komponentti-testausta voidaan tehdä päällekkäin ohjelmoinnin kanssa.

Tärkeimmät tarkistuspisteet ja työsuoritteiden palautusajankohdat on esitetty taulukossa 1. Työsuoritteet palautetaan Indrek Perille, joka koordinoi niiden tarkistamisen. Työvaiheesta riippuen muita tarkistajia ovat Anni Rytönen, Harri Laine ja Juha Taina.

Katselmukset asiakkaan ja tarkistajien kanssa järjestetään kulloisenkin tarkistuspisteen yhteydessä tarpeen mukaan tai kun ryhmän työsuoritteet on tarkistettu.

Projektisuunnitelma valmis	12.6.2000
Määrittely valmis	27.6.2000
Suunnitteludokumentti ja testaussuunnitelma valmiita	12.7.2000
Ohjelmointi	1.7.-1.8.2000
Testaus	1.8.-15.8.2000
Käyttöohje ja ylläpitodokumentti valmiita	7.8.2000
Projekti päättyy	15..30.8.2000

Taulukko 1. Tarkistuspisteet ja tärkeät ajankohdat

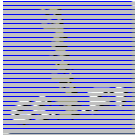


6 Dokumentin kirjoittajat (KS)

Tässä dokumentissa esiityneet kirjoittajat on mainittu kappaleiden otsakkeissa nimikirjaimin. Kappaleet 2 ja 3 perustuvat Harri Laineen SQL-Trainer HTML –dokumentaatioon, jota on osin tarkennettu.

Käytetyt lyhenteet ja kirjoittajat:

KS	Kimmo Sinkko
JN	Jaakko Nurro
ST	Santhu Toivakka
GE	George El-Khoury
HL	Harri Laine



Liite 1: SQL-trainer tietokannan luonti-SQL Oracle8 tietokantaan

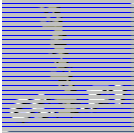
```
drop table pscourse;
drop table psanswer;
drop table psstudent;
drop table pstask;
drop table psstate;
drop table pstopic;

create table psstudent(
  sid varchar(10) not null,
  lname varchar(48) not null,
  fname varchar(48),
  last_visit date not null,
  HETU VARCHAR(11),
  SS VARCHAR(12),
  stnumber integer,
  primary key (sid)
);

create table pscourse(
  year integer,
  term char,
  startdate date,
  enddate date
);

create table psstate (
  sid varchar(10),
  task_id integer,
  lasttry integer,
  state integer,
  primary key (sid,task_id));
rem psstate täytyy initialisoida tehtävien lisäämisen jälkeen
rem viemällä laskureille alkuarvot 0,0

create table pstopic (
  tno integer not null,
  ttitle_fi varchar(60),
  ttitle_en varchar(60),
  tasks integer not null,
  credits_upto date not null,
  primary key (tno)
);
```



```
create table ptask (
  task_id integer not null,
  title_fi varchar(60) not null,
  title_en varchar(60),
  task_text_fi varchar(500),
  task_text_en varchar(500),
  ok_rows integer,
  nonunique integer,
  ok_cols integer,
  ok_checkable varchar(32),
  ok_checkvalue integer,
  ordered char,
  neededcols varchar(120),
  eqpairs varchar(120),
  tno integer,
  included varchar(120),
  col_max integer,
  needed_tabs varchar(120),
  needless_tabs varchar(120),
  primary key (task_id),
  foreign key (tno) references pstopic(tno)
);

create table psanswer (
  sid varchar(10) not null,
  task_id integer not null,
  trycnt integer not null,
  answ varchar(1000),
  trydate date,
  response integer,
  extra varchar(500),
  primary key (sid, task_id, trycnt),
  foreign key (task_id) references ptask(task_id),
  foreign key (sid) references psstudent(sid)
)
storage (initial 5M
  next 3M
  pctincrease 5);
```