

CT60A4301 Tietokannat

Harjoitustyö 1 – Tietokannan suunnittelu

Lappeenrannan teknillinen yliopisto
Innovation and Software (IS), LUT LBM

CT60A4302 Tietokannat
Kevät 2017

Simo Viljakainen
simo.viljakainen@student.lut.fi

SISÄLLYSLUETTELO

SISÄLLYSLUETTELO	1
1 MÄÄRITYS.....	2
2 KÄSITEMALLI.....	4
3 TIETOKANTATOTEUTUS	5
4 KESKUSTELU.....	6

1 MÄÄRITYS

Roodle on asiakkaalle tuotettu tietokanta ja käyttöliittymä paketti, joka on suunniteltu palvelemaan erilaisten koulujen tarpeita. Roodle-ohjelmisto tarjoaa oppilaille ja opettajille yhteisen alustan, jossa voi ilmoittautua kursseille ja arvostella suorituksia. Ohjelmisto tarjoaa käyttäjilleen kolme erilaista roolia: opiskelija, opettaja ja ylläpitäjä. Roolien mukaan määritetään käyttäjän omaavat oikeudet järjestelmään. Opiskelijat voivat selata kursseja, ilmoittautua kursseille, seurata omia ilmoittautumisia ja suorituksia. Opettajat voivat taas selata kursseja, seurata heidän vastuullaan olevien kurssien opiskelijoiden ilmoittautumisia ja arvostella heidän suorituksia. Kolmas käyttäjä ryhmä, ylläpito, on eristetty muista käyttäjäryhmistä. Ylläpitäjät pystyvät poistamaan oppilaita järjestelmästä, poistaa kursseja ja lisätä järjestelmään opettajia. Jokainen käyttäjäryhmä voi omien tehtävien lisäksi myös vaihtaa salasansa halutessaan.

Ohjelmisto tulee tarjota mahdollisuuden rekisteröityä ja kirjautua järjestelmään. ”Vierailijat” eli käyttäjät, jotka eivät ole kirjautuneet voivat nähdä vain etu-, rekisteröitymis- ja kirjautumissivun. Rekisteröitymisen yhteydessä tarkistetaan tietokannasta, että käyttäjän antama sähköposti on uniikki. Salasanan vahvuus ja annettujen salasanojen yhtenevyys tarkistetaan ohjelmistopohjaisesti. Kirjautumisen yhteydessä, käyttäjän antamasta salasanasta luodaan tiiviste ja tietokannasta tarkistetaan sen oikeellisuus. Ohjelmistoon voi ainoastaan rekisteröityä opiskelijaksi. Järjestelmän ylläpitäjä voi luoda opiskelijoista opettajia ja ylläpitäjiä voidaan vain luoda suoraan tietokantaan. Kirjautunut käyttäjä voi vaihtaa halutessaan salasansa. Salasanan vaihtamisen yhteydessä tarkistetaan käyttäjän antama ”vanha” salasana tietokannasta, samalla tavalla kuin kirjautumisen yhteydessä.

Jos kirjautunut käyttäjä omaa opiskelijan roolin, voi hän ilmoittautua kursseille siirtymällä sivulle, jossa kurssit on listattu. Ilmoittautuminen tapahtuu kurssin nimeä painamalla. Ainoastaan opiskelijat voivat ilmoittautua kursseille. Opiskelijat voivat seurata suorituksiaan ja ilmoittautumisiaan niille osoitetuilla sivuilla. Näille sivuille (myös kurssilista) tiedot haetaan tietokannasta ja listataan sivulle selkeällä formaatilla. Kun opiskelijan kurssi on opettajan arvioima, siirtyy se suoritukset-sivulle.

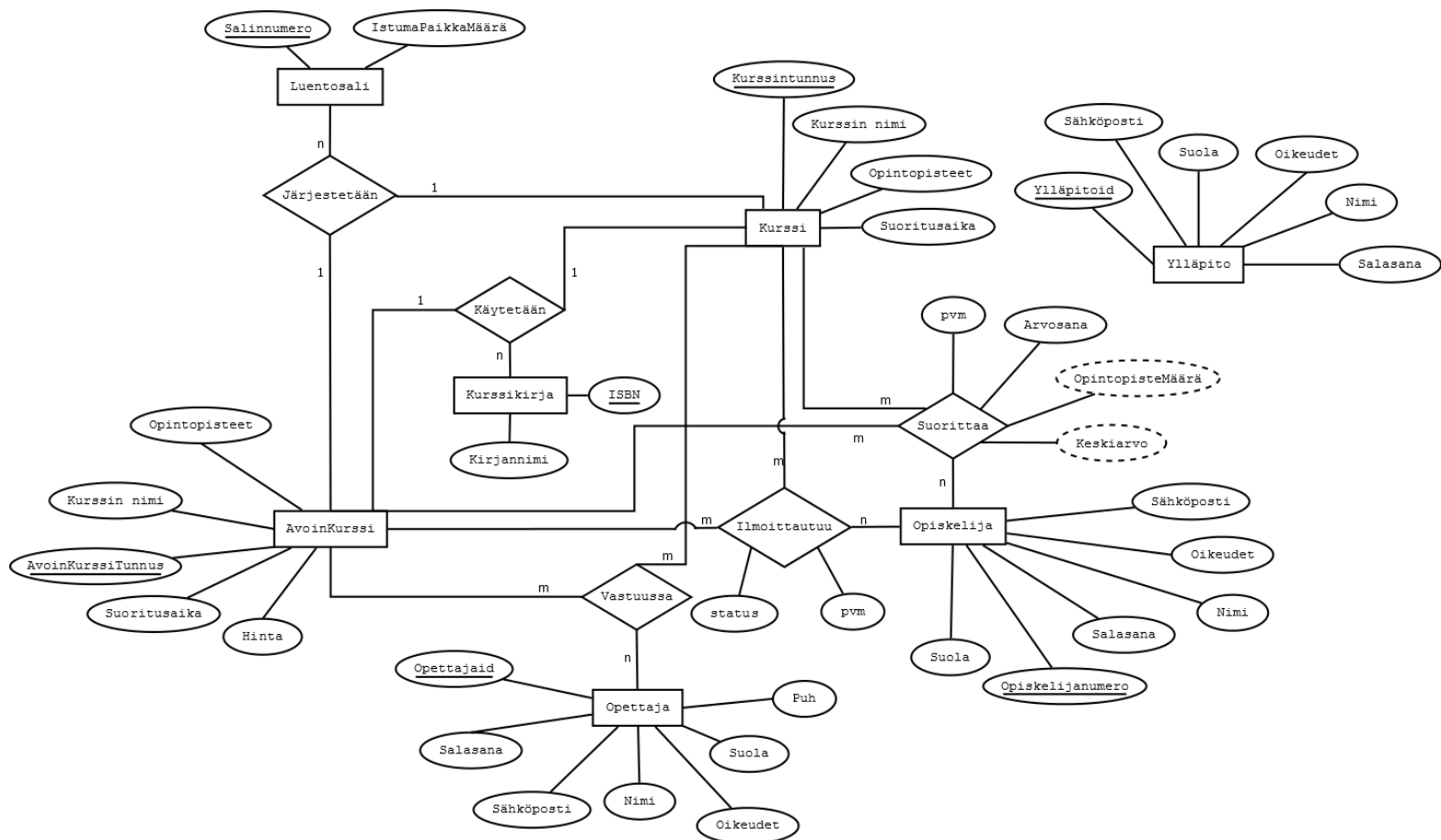
Kirjautunut opettaja voi halutessaan seurata omille kurseillensa ilmoittautuneita. Samalta sivulta, voi opettaja halutessaan arvostella opiskelijoiden suorituksia. Arvioinnin jälkeen ilmoittautuminen poistuu opettajan näkymästä ja siirtyy suoritukset-sivulle.

Ylläpitäjä voi luoda opiskelijoista opettajia. Tämä tapahtuu ylläpidolle osoitetulla sivulla. Opettajan luonnin yhteydessä, siirretään muokattavan opiskelijan tiedot opettaja tauluun ja hänen sähköpostinsa korvataan opettaja sähköpostilla (@koulu.com). Lisäksi opettajalle annetaan puhelinnumero, joka myös tallennetaan tietokantaan. Ylläpitäjät voivat myös poistaa opiskelijoita. Tämä tapahtuu samalta sivulta kuin opettajan luonti. Poistamisessa annetaan tietokannalle etsittävä termi (sähköposti), jonka mukaan tietokannasta poistetaan kyseinen rivi.

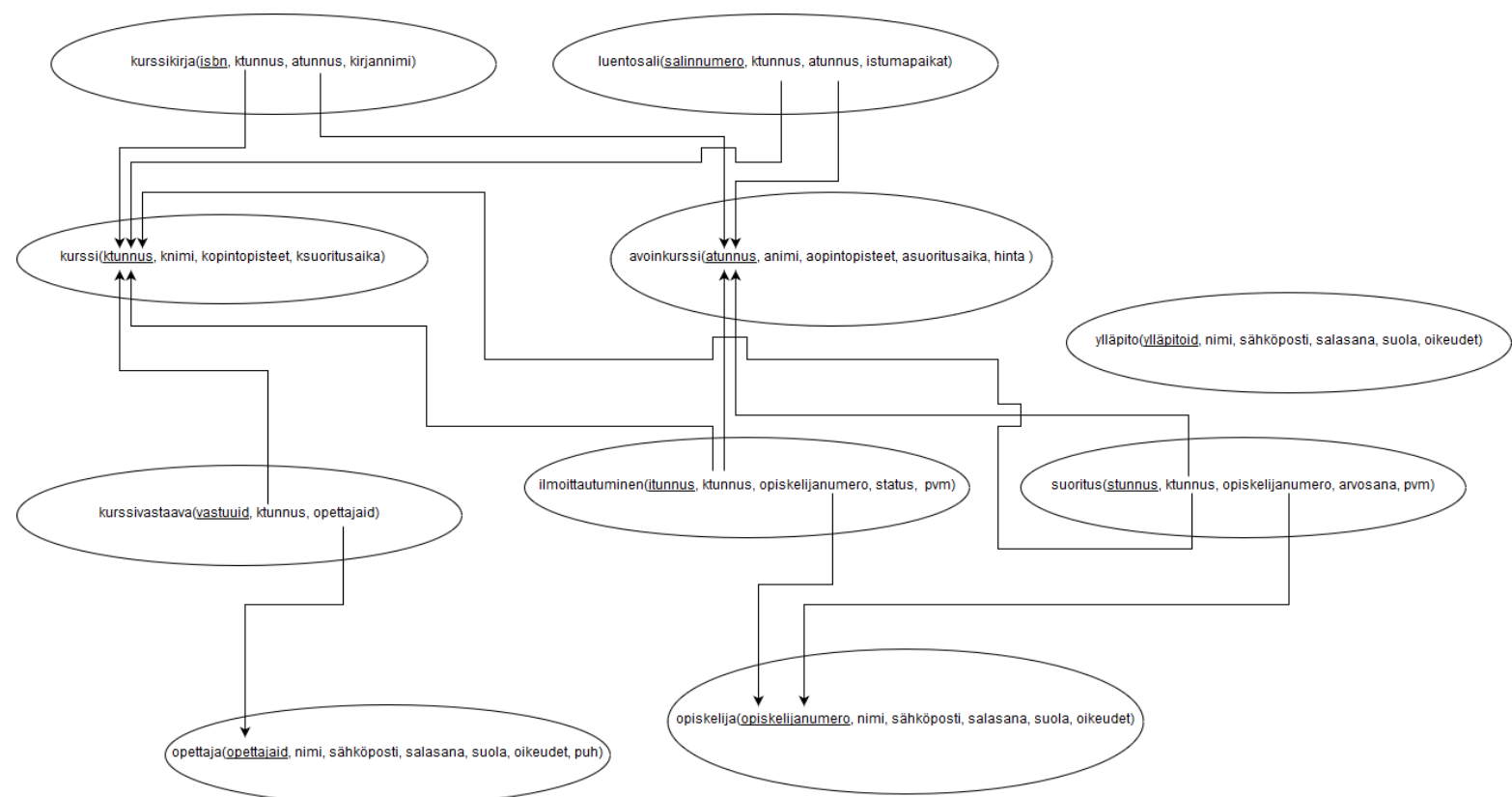
Tietokannan luonnissa on tärkeää, että roolit on eristetty toisistaan, sillä ne omaavat eri ominaisuuksia. Oikeudet on myös hyvä varmistaa erillisellä ominaisuudella.

Seuraavat tietokantakyselyt on vähintään toteutettava: (1) Listaa kaikki kurssit: kurssintunnus, kurssinimi, aikataulu, opintopistemäärä (2) Listaa tietyn opiskelijan ilmoittautumiset: kurssintunnus, kurssinimi, aika ja status. Listaa tietyn opiskelijan suoritukset: kurssintunnus, kurssinimi, aika ja arvosana. (3) Tarkista onko annettu sähköposti käytössä. (4) Tarkista onko annettu salasana oikeellinen. (5) Tarkista, onko kurssi X olemassa. (6) Laske tietyn opiskelijan opintopistemäärä ja opintopisteiden keskiarvo. (7) Listaa kaikki tietyn opettajan vastuulla olevat kurssit. (8) Listaa kaikki tietyn opettajan arvostelemaat kurssisuoritukset.

2 KÄSITEMALLI



Kuva 1 ER-kaavio



Kuva 2 Relaatiomalli

Relaatiomallissa (kuva 2) on merkitty pääavaimet alleviivauksilla ja vierasavaimet on merkitty nuolilla (vierasavain->pääavain) pääavaimiin johon ne viittaavat.

3 TIETOKANTATOTEUTUS

Yleistä: Käsitelmän toteutuksessa relaatiomalliksi ei tullut suurempia ongelmia. Toteutusvaiheessa kuitenkin ilmeni kurssi-aulun ja avoinkurssin-aulun kanssa ongelmia. Kyselyissä jouduttiin hakemaan data molemmista tauluista ja erotella se jälkeen päin. Tietokanta toteutuksessa lisättiin jokaiselle JOIN-aululle omat yksilöivät tunnisteet.

Toteutustapa:

Käytin toteutukseen www sovellukset -kurssilla luomaani käyttöliittymää (forum) pohjana. Käyttöliittymä on selain pohjainen HTML+Javascript-toteutus, jonka näkymää ja tiedonsiirtoa hallitsee PHP-palvelin koodi. Tietokantana käytin MySQL-tietokantaa.

Testatessa ja Koodissa huomioitavaa:

Toteutuksessa ja relaatiomallissa tuli nimettyä pari attribuuttia eri nimillä, mutta kaikki attribuutit ovat samoja kuin er- ja relaatiomallissa.

Ennen toteutuksen testaamista, tulee alustaa pari vakiota.

Alustettavat vakiot:

Tiedostoista: datahandler.php, databasehandler.php, logout.php

- Alusta kyseisten .php tiedostojen (sivun alussa, define(vakion nimi, vakion arvo)) vakiot käyttämiesi vakio arvojen mukaan.

WEBSITE -vakio on sivuston url , muista lisätä nimen alkuu "http://" ja loppuun "/"(jos et ole määritellyt muokattua url -osoitetta missä sivusto on, aseta arvoksi 'http://localhost/').

Vaihda tietokantavakiot DATABASE, HOST, USERNAME, PASSWORD tietokantasi mukaan.

Tietokannan nimeksi on määritelty vakiona ”opisto”, jos ei halua arvoa muuttaa voi vain luoda samannimisen tietokannan ja suorittaa schema kyselyt siihen.

Kaikki tietokantaan liittyvä toteutus löytyy databaseHandler.php tiedostosta. Käyttäjältä pyydettyä tai käyttäjän kysyessä dataa menee syöte/pyyntö Ajaxin/POST/GET:ien kautta datahandler.php (model controller) sivulle, joka tekee pyyntöjä käskyjä databaseHandler.php (model) sivulle. DatabaseHandler.php palauttaa halutun datan datahandler.php sivulle, joka taas kontroloi käyttäjän näkymää. Datahandler.php tarkistaa ensisijaisesti annetun datan, ennen kuin se menee databaseHandlerille.

MySQL-tietokannat ei tue CHECK-rajoitteita, joten suurin osa check-rajoitteista on toteutettu palvelinpuolella. Kuten arvosanan muoto (1-5, is_numeric). Oletusarvot ja mahdolliset tyhjät kentät on varmistettu tietokanta puolella. Myös tietenkin kenttien pituus ja tietotyypit on rajoitettu tietokantapuolella.

4 KESKUSTELU

Toteutin työssäni kaksi laajennusta:

- Itse toteutettu ohjelmallinen käyttöliittymä (tekstipohjainen, selain, työpöytäympäristö.)
- SQLiten korvaaminen tietokantapalvelimella, jolloin koodipohja muutetaan asiakasohjelma-käyttöliittymäksi

Muita huomioita:

Sivusto on kehitetty käyttämällä Windows käyttöjärjestelmälle tehtyä Xampp PHP kehitysympäristöä.

- Käyttöjärjestelmä : Windows 10
- Xampp 5.6.24
- Pavelin ohjelmisto: Apache 2.4.23
- PHP versio: PHP 5.6.28 (compiler: VC11 X86 32bit)
- Tietokanta: MariaDB 10.1.19

opisto.sql on koko tietokannan skeema (SQL-komennot.txt on sama tiedosto). Se on valmiiksi määritelty luomaan uusi tietokanta nimellä ”opisto”. Jos haluaa tehdä toisen nimisen tietokannan, voi tätä muuttaa.

Käytetyt INSERT, UPDATE ja DELETE käskyt ovat myös SELECT.txt tiedostossa.