Tietokannan dokumentaatio

iizo3030

Tuomas Kyttä H3408

Mikko Leppänen H3543

Eetu Manninen H3667

Huhtikuu 2015

Ohjelmistotekniikan koulutusohjelma

Tekniikan ja liikenteen ala

# Mitä tehtiin

Tarkoituksena oli luoda tietokanta blogi sivustolle, joka tehtiin osana JAMKin palvelinohjelmointi kurssia. Kantaan pystyy tallentamaan tietoa käyttäjistä, postauksista, kommenteista, oikeuksista ja tageistä.

# Yhteenveto ongelmista ja niiden ratkaisuista

## Leppänen

Ensimmäinen ongelma oli keksiä tauluja ja niiden sisältöä. Ryhmällä miettimällä keksimme joitain tauluja ja päätimme jakaa taulut ja niiden sisällön rakentamisen keskenämme. Tehtäväkseni jäi käyttäjä-taulun ja käyttäjien oikeuksien teko.

Usein tällaisissa palveluissa oikeudet ovat vain yksi arvo käyttäjän tiedoissa, mutta meillä oikeudet on oma taulunsa, lähinnä että tietokanta olisi tarpeeksi monimutkainen. Taulujen välillä on monen suhde moneen-yhteys(Välitaulun nimi on Rooli). Tämä mahdollistaa sen, että käyttäjän oikeuksia voi määritellä tarkemmin esim. käyttäjällä voi olla oikeus luoda uusia blogipostauksia, mutta ei kommentoida niitä.

Toisena ongelmana voisi mainita käyttäjätietojen tulostuksen. Tietoihin haluttiin lukumäärät käyttäjän tekemistä kommenteista ja blogipostauksista. Kolmen taulun yhdistämisestä seurasi kuitenkin se, että luvut olivat vääriä, joten nopeana ratkaisuna tein ensin näkymän(Kayttajatiedot\_valitaulu) kahden taulun(Kayttaja- ja Postaus-taulut) yhdisteestä ja toisen näkymän(Kayttajatiedot) ensimmäisen näkymän ja kolmannen taulun yhdisteestä(Kayttajatiedot\_valitaulu ja Kommentti-taulu).

## Kyttä

Seuraavaksi ongelmaksi muodostuivat erilaiset poistot, esim käyttäjän poisto jolla on kommentteja tai postauskia. Ongelmaan oli helppo ratkaisu ja vyörytys operaatioiden pitäisi toimia kutakuinkin seuraavasti: kun käyttäjä poistetaan hänen postauksensa ja kommenttinsa jäävät, kun postaus poistetaan sen ja tagi taulun välissä oleva esiintymä poistetaan, roolit poistetaan käyttäjältä poistettaessa.

Taulujen luominen tehtiin MySQL Workbenchillä joten tietokanan rakentaminen oli helposti tehty forward engineering operaatiolla ja testidataa lisättiin jälkikäteen insert operaatioilla.

Siistit lomakkeet tietokannan muokkaamiseen ovat JAMKIN palvelinohjelmointi kurssin harjoitustyömme, jota tekivät samat henkilöt kuin tässä työssä.

## Manninen

Tein Kommentti taulun ja Postaus taulun. Tein niihin aluksi paljon enemmän sarakkeita, mutta totesimme niiden olevan sql komennoilla kohtuullisen helposti muutenkin saatavilla. Kommentti taulussa kummittelee vieläkin tila, jota ei nykyisessä toteutuksessa käytetä lainkaan. Sen alkuperäinen tarkoitus oli näyttää onko kommentti poistettu tai piilotettu eli kommentin tilalle tulisi teksti ”poistettu” tms. Tätä emme kuitenkaan ehtineet toteuttamaan.

Tietokannan suunnittelussa olin myös aktiivisesti mukana kun pohdimme eri taulujen yhteyksiä ja mitä niille tapahtuu kun niitä poistetaan.

Php puolelle kirjoitin myös melko suuren määrän sql kyselyitä. Näiden kanssa tuli välillä taisteltua hetki, kun piti liittää tauluja.

# Arvosana ehdotukset perusteluineen

* Kyttä: 4. Tietokanta on helppolukuinen, toistoja ei ole muuten kuin postaus ja kommentti taulujen identtisen oloiset taulut. Taulujen väliset suhteet on tehty ja suunniteltu toimestani, silloin tällöin muiden ryhmäläisten palautteesta
* Leppänen: 4. Alueeni tietokannoista olivat käyttäjät ja niiden oikeudet. Tietokannasta tein 2 view:iä ja niitä käytetään järkevän oloisesti. Lisäksi osallistuin yhdessä muiden kanssa tietokannan erinäisiin korjaamisiin, ja loin php:llä luokan, joka suorittaa kaikki sql kyselyt.
* Manninen: 4. Olin osallisena tietokannan suunnittelussa ja tekemisessä sekä korjaamisessa kun se ei toiminut niin kuin oli tarkoitus. Kirjoitin myös melko suuren määrän SQL kyselyitä Tietokanta luokkaamme.

Liitteet

vaatimusMaarittelyTietokanta

TietokannanKasitemalli