# Tietokantojen perusteet: Keskustelufoorumi

Tietokantojen perusteiden ryhmätehtävänä on suunnitella ja toteuttaa web-pohjainen [keskustelupalsta](https://fi.wikipedia.org/wiki/Keskustelupalsta). Keskustelupalstan tulee tarjota ominaisuudet keskustelualueiden luomiseen, keskustelunavausten lähettämiseen keskustelualueelle, sekä tietysti viesteihin vastaamiseen. Viestien lähettäjät tulee pystyä tunnistamaan esimerkiksi viestiä kirjoitettaessa annettavan nimimerkin avulla.

Ryhmätyötä tehdään kurssin edetessä sitä mukaa kun opimme uusia asioita. Ryhmätyön aikataulu löytyy kurssisivulta.

Palautuksessa tulee olla mukana linkki toimivaan web-sovellukseen, korkeintaan 8 sivun mittainen dokumentaatio sovelluksesta sekä erillinen kirjanpitodokumentaatio työhön käytetystä ajasta. Tämän lisäksi jokainen ryhmän jäsen palauttaa itse- ja vertaisarvion (lomake julkaistaan myöhemmin).

Kurssin lopussa lopulliset raportit (ja niiden kautta valmiiden web-foorumien ja lähdekoodin osoitteet) jaetaan muiden opiskelijoiden nähtäväksi, joten foorumin viesteihin ei kannata kirjoittaa mitään mitä ei halua julkisesti nähtäväksi.

Huom! Työn osana tulee palauttaa myös tuntikirjanpito, josta tulee ilmi jokaisen osallistujan työhön laittamat tunnit. Kirjanpidossa tulee olla eriteltynä minkälaisiin tehtäviin tunnit on laitettu.

## **Kuvaus**

Kun käyttäjä avaa keskustelupalstan, tulee hänen nähdä yleistietoa keskustelualueille kirjoitetuista viesteistä, esimerkiksi seuraavasti:

***Alue Viestejä yhteensä Viimeisin viesti***

Lemmikit 121 2016-02-11 13:30

Lentokoneet 8 2015-08-03 08:00  
Ohjelmointi 4421 2016-02-15 13:30

Kun alue valitaan, näytetään alueeseen liittyvällä sivulla 10 viimeisintä keskustelunavausta esimerkiksi seuraavasti:

***Alue: Ohjelmointi Viestejä Viimeisin viesti***

Java on jees! 8 *pvm*

Python on jeesimpi! 6 *pvm*

LISP on parempi kuin… 2 *pvm*

Ohjelmointikielet on turhia.. 1 *pvm*

...

Yksittäiseen keskusteluun pääsee taas käsiksi valitsemalla keskustelunavauksen -- esimerkiksi ylläolevassa listassa voisi valita keskustelun “Java on jees!”, jolloin keskustelun viestit näkyisivät:

***Alue: Ohjelmointi → Java on jees***

1. Mun mielestä Java on just hyvä kieli. t. Arto
2. No eipäs, Ruby on parempi. t. Matti
3. Ada on selkeästi parempi kuin kumpikin noista. t. Ada
4. ...

Lisää viesti:

[*käyttäjä voisi kirjoittaa viestin..*]

-----

Ryhmätyö koostuu useammasta vaiheesta, joista ensimmäinen on alustavan tietokantakaavion suunnittelu sekä oleellisimpien kyselyiden hahmottelu.

## **Askel 1**

* Etsi laskuharjoituksissa itsellesi ryhmä -- laskarien vetäjä auttaa tässä. Jos missasit laskarit tai et löytänyt ryhmää, ota ohjaajiin yhteyttä sähköpostitse
* Sopikaa milloin pohditte tehtävää, sekä jakakaa yhteystiedot toisillenne. Sopikaa myös työväline (esim. Google docs) ryhmätyön tulosten kirjaamiseen.
* Hahmotelkaa keskustelupalstaan sopivaa tietokantakaaviota. Tässä muutamia ohjaavia kysymyksiä:  
  + Kuka lähettää viestin?
  + Minne viesti lähetetään?
  + Liittyykö viesti johonkin toiseen viestiin?
  + Milloin viesti on lähtenyt?
  + Minkälaiset toiminnallisuudet ovat oleellisia keskustelupalstan käytön kannalta?
  + …
* Niissä määrin kuin kurssin välineillä tällä hetkellä on mahdollista, aloittakaa myös oleellisten tietokantakyselyiden hahmottelu.
* Viilatkaa foorumin tietokantakaavio sopivaksi, ja luokaa sen pohjalta SQL:n create table lauseet.
* Luokaa tietokanta SQL:n create table lauseilla SQLiteen -- määritelkää myös pää- ja viiteavaimet.
* Aloittakaa ensimmäisen raportin kirjoittaminen tähän asti tekemänne työn perusteella. Raportin ohjeet ovat tämän dokumentin lopussa.

## **Askel 2**

Askel 2:n pohjana kannattaa käyttää [tikape-runkoa](https://github.com/tietokantojen-perusteet/tikape-runko), jonka käyttöön ottamiseen löytyy lisäohjeita ‘Ohjeita ryhmätyön organisointiin’ -dokumentista.

1. Lisätkää tietokantaan testidataa (aihealueita, aihealueeseen liittyviä keskustelunavauksia, keskustelunavauksiin liittyviä viestejä), jolla voitte testata sovelluksen toimintaa.
2. Toteuttakaa tietokanta-abstraktio (Database-luokka, Tietokantatauluja kuvaavat luokat, oleellisia kyselyitä helpottavat Dao-oliot). Huom! Tässä ei saa käyttää valmiita Dao-kirjastoja, vaan kyselytoiminnallisuus tulee toteuttaa itse.
3. Luokaa jonkinlainen tekstikäyttöliittymä SQL-kyselyiden ja tietokanta-abstraktioiden testaamiseen. Tekstikäyttöliittymää saattaa kannattaa jatkossa myös päivittää vastaamaan tästä eteenpäin kuvattavan web-sovelluksen toiminnallisuutta.
4. Luokaa “paikallisella koneella” toimiva web-sovellus foorumin ympärille, joka näyttää kaikki foorumin aihealueet kun käyttäjä tekee pyynnön “juuriosoitteeseen” (“/”):

***Alue Viestejä yhteensä Viimeisin viesti***

Lemmikit 121 2016-02-11 13:30

Lentokoneet 8 2015-08-03 08:00  
 Ohjelmointi 4421 2016-02-15 13:30

1. Lisätkää sovellukseen mahdollisuus lisätä uusia alueita.
2. Lisätkää foorumin alueisiin linkit, jonka perusteella yksittäisen alueen keskustelunavauksia voi listata. Jos esimerkiksi alueen Ohjelmointi id on 1, voi alueen viestit näyttää esimerkiksi polun “/alue/1” takaa. Alueen tunnuksen saa selville Sparkin queryParams-metodin avulla. Yksittäisen alueen viestit tulee listata esimerkiksi seuraavalla tavalla -- nyt viestien lukumäärä voi kaikissa olla 1.   
     
   ***Alue: Ohjelmointi Viestejä Viimeisin viesti***

Java on jees! 8 *pvm*

Python on jeesimpi! 6 *pvm*

LISP on parempi kuin… 2 *pvm*

Ohjelmointikielet on turhia.. 1 *pvm*

…

1. Lisätkää foorumin alueisiin mahdollisuus lisätä uusia keskustelunavauksia. Keskustelunavauksessa tulee olla otsikko sekä viesti -- jatketaan seuraavalla kierroksella viestien lisäämisellä ja lukemisella.

## **Askel 3**

1. Seuratkaa kurssin oppimateriaalissa olevia ohjeita ja siirtäkää sovellus verkossa toimivaan Heroku-palveluun. Liittäkää sovellukseen Postgres-tietokanta ja tarvittaessa myös muokatkaa kyselyitänne toimimaan sen kanssa. Huomatkaa että jos tietokannan rakenteeseen tulee tämän jälkeen muutoksia, vanhat tietokantataulut on poistettava Herokusta erikseen.
2. Lisätkää sovellukseen mahdollisuus lukea keskustelunavaukseen liittyvät viestit.

***Alue: Ohjelmointi → Java on jees***

1. Mun mielestä Java on just hyvä kieli. t. Arto
2. No eipäs, Ruby on parempi. t. Matti
3. Ada on selkeästi parempi kuin kumpikin noista. t. Ada
4. ...
5. Lisätkää mahdollisuus lisätä uusi viesti, joka liittyy keskustelunavaukseen. Esimerkiksi:

***Alue: Ohjelmointi → Java on jees***

1. Mun mielestä Java on just hyvä kieli. t. Arto
2. No eipäs, Ruby on parempi. t. Matti
3. Ada on selkeästi parempi kuin kumpikin noista. t. Ada
4. ...

Lisää viesti:

[ Tähän tulee laaja tekstikenttä, johon voi kirjoittaa viestin. Tekstikentän   
 saa luotua HTML:n textarea-elementillä ]

Lähettäjä:

[ Tähän tulee normaali tekstikenttä, johon kirjoitetaan lähettäjän nimi. Tekstikentän  
 saa luotua HTML:n input-elementillä. ]

1. Päivittäkää Askel 2:n kohdassa 4 käytettävää kyselyä, ja varmistakaa, että jokaisen keskustelunavauksen kohdalla näytetään oikea viestien lukumäärä sekä viimeisimmän alueelle tulleen viestin päivämäärä. Keskustelunavaukset tulee järjestää uusimman viestin perusteella.
2. Päivittäkää Askel 2:n kohdassa 6 käytettävää kyselyä, ja varmistakaa, että jokaisen alueen kohdalla näytetään oikea viestien lukumäärä sekä viimeisimmän alueelle tulleen viestin päivämäärä. Alueet tulee järjestää alueen nimen perusteella.
3. Voitte olettaa, että alueita ei tule kovin montaa. Keskustelunavauksien määrä voi kuitenkin kasvaa huomattavasti, joten aluekohtaisille sivulle tarvitaan jonkinlainen tapa näytettävien viestien lukumäärän rajoittamiseen. Rajoittakaa keskustelunavausten määrää niin, että aihealuekohtaisella listalla näytetään aina kymmenen uusinta keskustelunavausta.
4. Yksittäinen keskustelunavaus voi aiheuttaa vilkasta debattia, jolloin keskustelunavaukseen voi tulla merkittävä määrä viestejä. Lisätkää sovellukseen mahdollisuus rajata kerrallaan näytettävien viestien määrää. Huom! On mahdollista lisätä osoitteisiin esimerkiksi parametri “sivu=1”, jolloin kerrotaan, että tulee näyttää ensimmäinen sivu. Näytettävät viestit saa selville parametrin arvon, limit-komennon, sekä kertolaskun avulla.
5. Kun sovelluksen toiminnallisuus on kutakuinkin kasassa, tarkistakaa vielä kerran, ettei sovelluksessa ole mahdollisuuksia SQL-injektioiden tekemiseen. Tämän lisäksi on hyvä rajoittaa sivun toiminnallisuutta siten, että käyttäjä ei voi syöttää HTML-koodia tietokantaan. [Miksi tämä kannattaa tehdä?](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
6. Jos ryhmätyö edistyy hyvin ja ylimääräistä aikaa on riittävästi, web-sovelluksen ulkoasua voi vaikkapa hioa hyödyntämään [Bootstrapia](http://www.w3schools.com/bootstrap/), [w3.css](http://www.w3schools.com/w3css/default.asp)-tyylitiedostoa tai [Skeleton](http://getskeleton.com/)-tyylitiedostoa. Ulkoasun muuttamisen ei tule vaikuttaa ohjelman toimintaan, vaan kaikki muutokset voi tehdä näkymässä.

**...**

## **Selventäviä kysymyksiä ja vastauksia**

Ongelman selventäminen, asiakkaalta on kysytty (Q) tähän mennessä seuraavaa, joihin alla on myös annettu vastauksia (A):

Q: “Pitäisikö käyttäjän pystyä tunnistautumaan myös järjestelmään”

A: “Käyttäjän ei tarvitse tunnistautua mitenkään erityisesti. Kun käyttäjä kirjoittaa viestin, kirjoittaa hän viestin yhteydessä myös oman nimimerkkinsä esimerkiksi erilliseen kenttään.”

Q: “Millä tavalla viestejä tulee ryhmitellä? Tuleeko vastaukset liittää aina siihen viestiin mihin vastataan siten, että yhteen viestiin voi liittyä sisäkkäinen keskustelu?”

A: “Voidaan hyvin ajatella niin, että kaikki vastaukset viestiin nähdään vastauksena alkuperäiseen viestiin. Alkuperäinen viesti on siis hieman kuin keskustelunavaus, ja jokainen voi sitten osallistua keskusteluun kirjoittamalla viestin -- vastauksen -- keskustelunavaukseen.”

Q: “Miten saan päivämäärät toimimaan SQLite-tietokannassa?”

A: “Päivämäärän ja ajan käsittelemiseen on monia mahdollisia tapoja, asiaan voi perehtyä vaikka jonkun hakukoneen avulla.”

Q: “Okei, mitä meidän nyt pitäisi siis tehdä?”

A: “On hyvä yrittää selventää tarkemmin mitä oikeastaan tässä pitäisi tehdä. Kurssilla oleva materiaali tukee toimintaa viikoittain.   
  
Nyt kun materiaalissa on tietoa tietokantakyselyiden tekemisestä ohjelmallisesti, kannattanee sitä jo harjoitella.   
  
Ryhmässä kannattaa miettiä myös työhön liittyviä tehtäviä sekä niiden jakamista ryhmän kesken, sekä tietysti sitä, että miten ohjelmakoodi saadaan jaettua kaikkien ryhmäläisten kesken niin, että jokainen pääsee antamaan panoksensa siihen. Muutamia hyviä työvälineitä ovat esimerkiksi trello ja github. Varsinkin githubin käytön opettelusta on paljon hyötyä myös tulevaisuudessa.”

Q: “Ei vaan oikeasti, mitä tässä koko projektissa pitäisi tehdä?”

A: “Projektin tavoitteena on luoda web-pohjainen foorumisovellus, joka käyttää tietokantaa. Tämä kannattanee tehdä esimerkiksi niin, että ensin pohditaan tietokantaa, sen rakennetta sekä sitä, miltä kyselyt näyttävät tietokannassa. Tämän jälkeen voi olla hyvä kokeilla ensin yksittäisten kyselyiden tekemistä ohjelmallisesti, jonka jälkeen mietitään miten foorumi toimisi ohjelmana sekä luodaan ehkäpä jonkinlainen tekstikäyttöliittymä foorumille. Kun tekstikäyttöliittymä on valmis, voi web-käyttöliittymän rakentamista alkaa miettimään. Pienissä paloissa siis. Lisäksi projektin suorituksesta tulee myös raportoida.”

Q: “Miksi yhteenvetokyselyt eivät toimi Postgresin kanssa?”

A: “Postgres noudattaa vanhempaa SQL-standardia kuin SQLite jota olemme käyttäneet kurssin harjoituksissa. Tässä vanhemmassa standardissa yhteenvetokyselyiden ja GROUP BY -komennon käsittely on hieman erilaista. Ongelmallista kyselyä voi muokata esimerkiksi käyttämään sopivasti alikyselyitä ongelman ratkaisemiseksi.”

Q: “Miten tämä ryhmätyö arvostellaan?”

A: “Ryhmätyössä arvostellaan suunnitellun tietokannan soveltuvuus tarkoitukseen, tietokantakyselyiden hyvyys, sovelluksen rakenne, sekä sovelluksen käytettävyys. Tämän lisäksi käyttöohje ja dokumentaatio vaikuttaa loppuarvosanaan. Jokainen ryhmäläinen myös arvioi sekä omaa että muiden ryhmäläisten työpanosta, mikä myös luonnollisesti vaikuttaa arvosteluun.”

Q: “Miten tällaista ryhmätyötä kannattaa tehdä?”

A: “ Kannattaa välttää tilannetta, missä omii kaiken työn, sekä tilannetta, missä välttää työtä -- kannattaa erityisesti pyrkiä aikatauluttamaan projekti, sekä kysellä lisätietoa tarvittaessa.  
Ryhmän kanssa kannattaa pyrkiä työskentelemään yhdessä jos vain mahdollista. Sovelluksen toteutusta edistävien työtehtävien yhdessä miettiminen sekä jakaminen -- esimerkiksi työpareille -- melko aikaisessa on hyödyksi, kuten myös päätetyt aikataulut sekä aikatauluihin sitoutuminen. Sovelluksen toiminnallisuutta voi toteuttaa myös pariohjelmointina, mikä vähentää riskiä, joka liittyy siihen, että vain yksittäinen ryhmäläinen tietää jonkun ohjelma-alueen toiminnallisuudesta.“

Q: “Saammeko käyttää jotain valmista ORM-kirjastoa?”

A: “Tässä työssä ei. ORM-kirjastot, vaikka ne ovatkin huikeimpia asioita sitten valmiiksi viipaloidun leivän, peittävät aika paljon harjoiteltavasta ydinaineksesta, eli kyselyiden tekemisestä.“

## **Raportointi**

Ryhmätehtävästä palautetaan kaksi PDF-muotoista kirjallista raporttia ja tuntikirjanpidot. Ensimmäisen raportin tulee sisältää:

* Otsikkosivu (Aiheen ja ryhmän nimi)
* Käsitekaavio tai muu esitys, josta selviävät oleelliset käsitteet sekä niiden riippuvuudet.
* Käsitteiden attribuuttien arvoalueet -- esimerkiksi nimet esitetään merkkijonona.
* Tietokantakaavio kuvana.
* SQL:n create table -lauseet kolmelle tärkeimmälle taululle.
* Muutamia keskeisiä [käyttötapauksia](https://en.wikipedia.org/wiki/Use_case#Fowler_style) sekä selvitys siitä miten ne pystytään toteuttamaan tietokannan avulla: mitä tauluja tarvitaan sekä minkälaisia kyselyitä näiden toteuttamiseen tarvitaan (esimerkiksi etusivulla näkyvä aihealueiden listaus, missä näkyy aihealuekohtaiset viestit sekä uusimman viestin päivämäärä)

Ensimmäinen raportti tulee palauttaa kurssialueelle Moodleen. Raportin tiedostonimeen kannattaa lisäksi sisällyttää myös ryhmän nimi jotta se ei huku muiden raporttien joukkoon.

Tämän lisäksi kurssialueelle palautetaan pdf-muodossa ryhmätyön tuntikirjanpito, josta tulee käydä ilmi ryhmän nimi, ryhmäläisten nimet ja opiskelijanumerot, sekä tehtävittäin eriteltynä kunkin ryhmäläisen työhön käyttämä tuntimäärä.

Lopullisen raportin tulee sisältää:

* Otsikkosivu (Aiheen ja ryhmän nimi)
* Web-sovelluksen osoite sekä mahdolliset ohjeet sovelluksen käyttöön
* Linkki sovelluksen koodeihin (esim. githubissa)
* Käsitekaavio tai muu esitys, josta selviävät oleelliset käsitteet sekä niiden riippuvuudet.
* Käsitteiden attribuuttien arvoalueet -- esimerkiksi nimet esitetään merkkijonona.
* Tietokantakaavio kuvana.
* SQL:n create table -lauseet kolmelle tärkeimmälle taululle.
* Muutamia keskeisiä käyttötapauksia sekä selvitys siitä miten ne pystytään toteuttamaan tietokannan avulla: mitä tauluja tarvitaan sekä minkälaisia kyselyitä näiden toteuttamiseen tarvitaan (esimerkiksi etusivulla näkyvä aihealueiden listaus, missä näkyy aihealuekohtaiset viestit sekä uusimman viestin päivämäärä)
* Ratkaisussa havaitut ongelmat, mikäli sellaisia löydettiin.
* Miten sovellusta kannattaisi jatkokehittää.

Kuten on havaittavissa, lopullisen raportin on tarkoitus sisältää myös ensimmäisen raportin tiedot. Jos ensimmäiseen raporttiin kirjatut tiedot ovat projektin edetessä muuttuneet, niin lopullisessa raportissa tulee näistä olla päivitetty versio, ei alkuperäinen.

Lopullisen raportin maksimipituus on 8 sivua, ja se palautetaan samalla tavalla kuin ensimmäinen raportti. Raportin tiedostonimeen kannattaa edelleen sisällyttää myös ryhmän nimi jotta se ei huku muiden raporttien joukkoon.

Tämän lisäksi kurssialueelle palautetaan pdf-muodossa taas ryhmätyön tuntikirjanpito, josta tulee käydä ilmi ryhmän nimi, ryhmäläisten nimet ja opiskelijanumerot, sekä tehtävittäin eriteltynä kunkin ryhmäläisen työhön käyttämä tuntimäärä.

## **Vertaisarviointi**

Ryhmätyön lopuksi ryhmätyön ja projektin onnistumista arvioidaan itse- ja vertaisarviontilomakkeen avulla.

## **Lopuksi**

Sovellusta saa päivittää myös raportin palautuksen jälkeen, arvostelu kuitenkin perustuu palautusajan tilanteeseen. Erityisesti jos luomassanne sovelluksessa paljastuu tietoturvapuutteita, niin sitä kannattaa päivittää näiden korjaamiseksi.

Kun ryhmän raportti on palautettu, jos ryhmä on kirjoittanut tuntikirjanpitoa Google Docsissa anonyymisti, niin poistakaa anonyymi lukuoikeus.