# Tietokantojen perusteet, viikko 3

## Harjoituksissa tehtävät tehtävät

Kolmansissa laskuharjoituksissa tehtävät harjoitustehtävät.

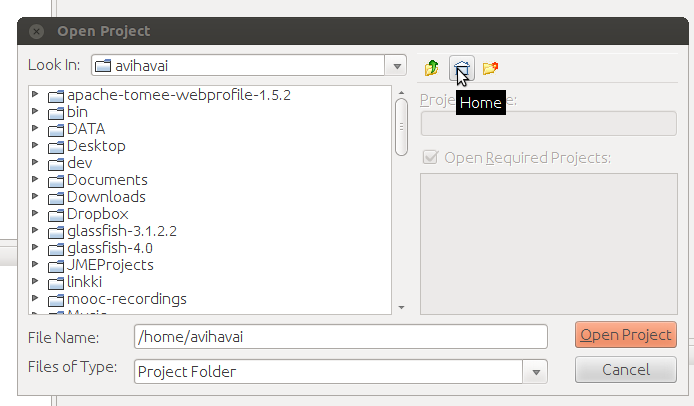
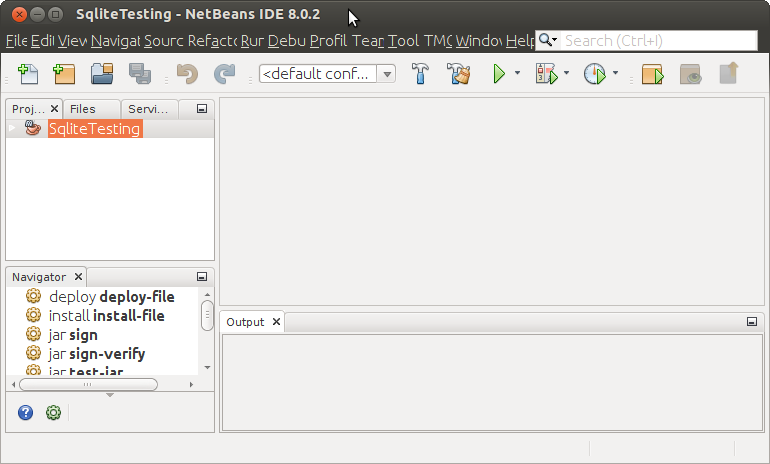
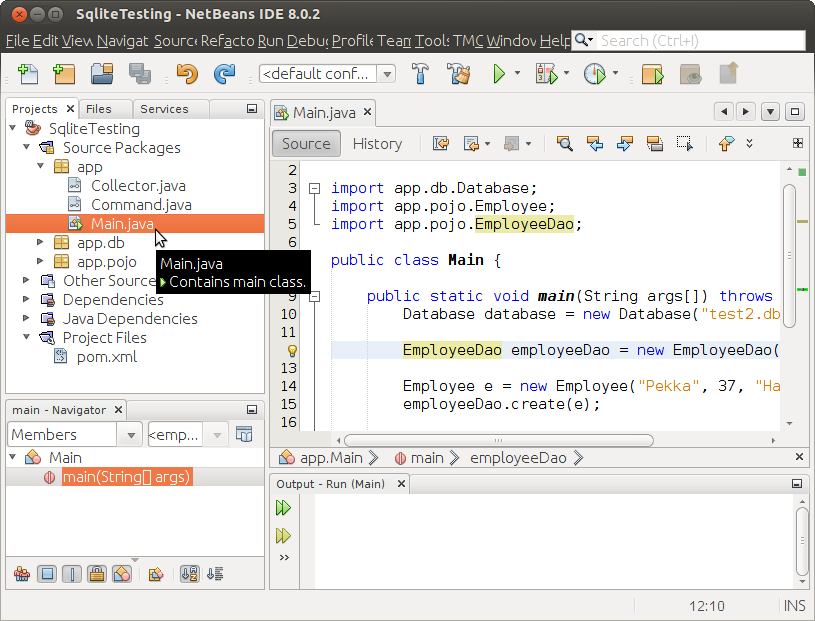
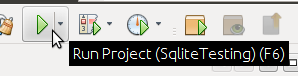
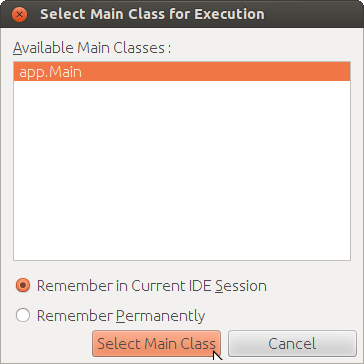
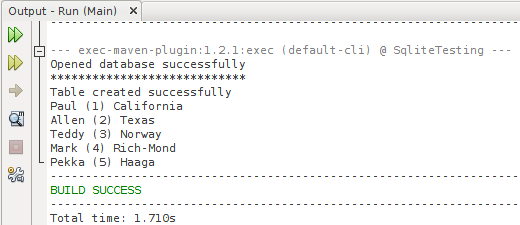
Tehtävät tehdään projektiryhmissä tai pareittain. Läsnä tehtävissä tehtävissä kerrataan tietokannan ohjelmallista käyttöä laitoksen koneilla.

**Osa 1. Projektin noutaminen github-palvelusta omalle koneelle**

Työstettävä projekti noudetaan ohjelmistoprojektien versioiden hallintaan näppärästä github-palvelusta omalle koneelle, jotta sen parissa voi työskennellä paikallisella koneella. Jos komennot eivät ole vielä tuttuja, ne tulevat myös muilla kursseilla hiljalleen tutuksi.

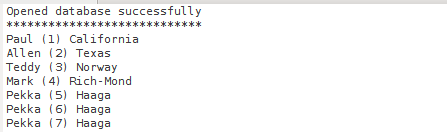
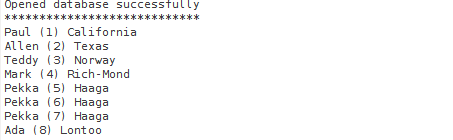
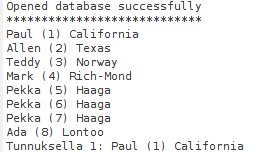
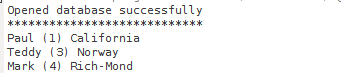
1. Vaihtakaa Windowsista Linuxiin (Ubuntu) käynnistämällä kone uudestaan.
2. Avatkaa terminaali (Vasemmassa ylälaidassa oleva Ubuntun käynnistysvalikko, etsikää "Terminal")
3. Kun terminaali avautuu, luokaa kansio "tikape" komennolla "mkdir tikape"
4. Menkää juuri luomaanne kansioon komennolla "cd tikape"
5. Avatkaa selain (Vasemmassa ylälaidassa oleva Ubuntun käynnistysvalikko, etsikää "Chrome" tai “Firefox”)
6. Avatkaa selaimella github-versionhallintapalvelussa olevan tietokantaa käyttävän sovelluksen osoite https://github.com/tietokantojen-perusteet/SqliteJava ja kopioikaa projektiin liittyvä "**HTTPS clone URL**"-osoite (https://github.com/tietokantojen-perusteet/SqliteJava.git), eli osoite, jonka avulla projektin voi noutaa omalle koneelle
7. Noutakaa projekti omalle juuri luotuun kansioon koneelle kirjoittamalla "git clone <https://github.com/tietokantojen-perusteet/SqliteJava.git>"
8. Menkää juuri noudettuun projektikansioon komennolla "cd SqliteJava". Nyt terminaalissa olevan alimman rivin tekstin pitäisi olla muotoa "omatunnus@koneentunnus:~/tikape/SqliteJava$"
9. Pyytäkää Mavenia (projektinhallintatyöväline) hakemaan projektiin liittyvät apukirjastot komennolla "mvn dependency:resolve". Kun apukirjastot on noudettu, näette tekstin "BUILD SUCCESS"

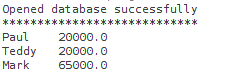
**Osa 2. Projektin avaaminen Netbeansissa**

1. Avatkaa Vasemmassa ylälaidassa olevan Ubuntun aloitusvalikon TmcBeans-nimistä ohjelmointiympäristöä.
2. Valitkaa ohjelmointiympäristössä File -> Open Project. Näette ikkunan, jonka kautta voitte avata projekteja. Hakekaa ensin kotikansio klikkaamalla “Home”.
3. Etsikää kansio “tikape” ja tuplaklikatkaa sitä, jolloin se aukeaa. Tuplaklikatkaa tämän jälkeen kansiossa olevaa SqliteJava-nimistä projektia, joka avaa sen ohjelmointiympäristöön. Projektin nimi ympäristössä on SqliteTesting.  
     
   
4. Etsikää projektiin liittyvä Main.java -tiedosto, ja tuplaklikatkaa sitä, jolloin se aukeaa muokattavaksi.  
   
5. Painakaa projektiin liittyvää play-nappia, joka käynnistää ohjelman.  
     
     
   Joudutte mahdollisesti vielä valitsemaan tiedoston, johon liittyvä ohjelma käynnistetään.  
     
     
     
   Output-ikkunaan ilmestyvän tulostuksen pitäisi kutakuinkin seuraavanlainen, luonnollisesti päivämäärät ja projektin sijainti riippuu ohjelman suorituspaikasta ja hetkestä.   
   
6. Noniin. Käytössä on tietokantaa käyttävä projekti, ja tietokantakyselyjen tekeminen tuntuu onnistuvan.

**Osa 3. Projektiin tutustuminen ja sen muokkaaminen**

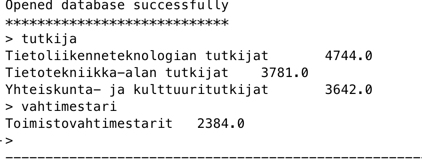
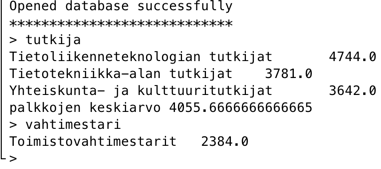
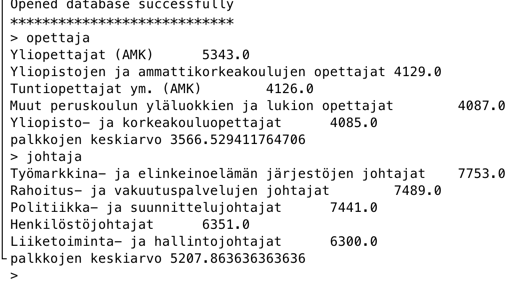
Tutustutaan seuraavaksi projektiin, ja tehdään siihen muutoksia.

1. Käynnistäkää sovellus vielä muutamaan otteeseen. Näyttää siltä, että Pekkojen määrä lisääntyy.  
     
   Muokatkaa ohjelmaa Main.java siten, että uutta Pekkaa ei lisätä tietokantaan aina sovelluksen käynnistyksen yhteydessä.
2. Muokatkaa ohjelmaa Main.java siten, että ohjelman suorituksen yhteydessä tietokantaan lisätään yksi uusi työntekijä, jonka nimi on “Ada”. Ada asuu Lontoossa, loput yksityiskohdat saatte keksiä itse.  
   
3. Sovelluksen käyttämä EmployeeDao tarjoaa toiminnallisuuksia työntekijöiden hakemiseen. Etsikää sieltä metodi, jonka avulla voi hakea työntekijää tunnuksen perusteella. Lisätkää Main.java-tiedostoon ohjelmakoodi, jolla haetaan työntekijä, jonka tunnus on 1.  
     
   
4. SQL-lauseen WHERE-hakuehdon avulla voi tehdä rajattavia haettaviin tietoihin. Tehkää findAll-metodista versio, joka hakee kaikki työntekijät, joiden palkka on vähintään annettu summa. Antakaa metodin nimeksi findAllWithSalaryAtLeast -- se saa parametrinaan double-tyyppisen muuttujan.  
     
   Nyt koodi  
     
   *for (Employee e: employeeDao.findAllWithSalaryAtLeast(20000)) {  
    System.out.println(e);  
   }*  
     
   tulostaa kaikki työntekijät, joiden palkka on vähintään 20000.  
   
5. Muokatkaa tulostusta siten, että tulostus tulostaa nimen ja palkan. Nimen ja palkan väliin tulee asettaa tab-lyönti, jonka saa aikaan merkillä “\t”.



**Osa 4. Aitoa dataa**

Tutustutaan hieman yksityisen sektorin mediaanipalkkoihin. Palkkatilastot on noudettu tilastokeskuksen sivuilta osoitteesta http://www.stat.fi/til/yskp/2014/yskp\_2014\_2015-08-20\_tie\_001\_fi.html .

1. Vaihtakaa käytettävää tietokantaa **test2.db**:stä **palkkatilastot.db**:ksi -- tiedosto palkkatilastot.db on tullut projektin mukana. Nyt käytössä on tilastokeskuksen palkkatilasto.
2. Laajenna tietokantaa käsittelevää ohjelmistoa siten, että ohjelma kysyy käyttäjältä syötettä, ja hakee tietokannasta syötettä vastaavat ammattiryhmät ja tulostaa niiden keskipalkat. Kun käyttäjän antama syöte (alla > merkin eteen käyttäjän kirjoittama merkkijono) on tyhjä, ohjelman suoritus loppuu:  
     
   
3. Jos ohjelma palauttaa yhdessä kyselyssä *useamman kuin yhden* ammattiryhmän, laske ja tulosta ryhmän *palkkojen keskiarvo.*   
     
   
4. Laajenna vielä ohjelmaa siten, että kyselyn vastauksena olevat ammattiryhmät tulostetaan palkkojen mukaan järjestettynä:  
     
   
5. Viimeisenä askeleena hiotaan ohjelmaa vielä siten, että jos kyselyn vastauksessa on enemmän kuin 5 ammattiryhmää, näytetään niistä ainoastaan 5 eniten ansaitsevaa:  
     
     
     
   Parin edellisen tehtävän tapaan myös nyt voidaan näytettävien vastausrivien määrä rajata viiteen joko ohjelmakoodissa, tai voidaan tehdä SQL-kysely, joka rajoittaa vastauksen viiteen, parhaan palkan omaavaan ammattiryhmään.

### Jos tekeminen loppuu kesken:

Harjoitellaan web-ohjelmointia, jota tarvitaan ryhmätyössä. Tarvitset käyttöösi TMC:n (käytössä esim. ohjelmoinnin perusteet -kurssilla). Tehtävissä hyödynnettävä materiaali löytyy osoitteesta <http://web-palvelinohjelmointi.github.io/s2015-wepa-mooc.html>  
  
TMC:ssä palvelimen osoitteeksi tulee valita <https://tmc.mooc.fi/org/hy/>-- kurssin nimeksi hy-tikape-kesa2017. Jos ryhmässäsi on henkilöitä, jotka eivät ole vielä tehneet alkupään tehtäviä

* W1-06.HelloModel
* W1-07.HelloForm
* W1-08.HelloList
* W1-11.HelloObjects
* W1-12.HelloPathVariables
* W1-13.HelloIndividualPages

Tehkää ne nyt.

Jatkakaa tehtävillä:

* W2-24.HelloDatabase
* W2-25.HelloInsert
* W2-26.HelloDao
* W2-27.ReadingList
* W2-28.HelloOneToMany
* W2-29.HelloCategories

Tehtävien tekoa saa ja kannattaa jatkaa myös myöhemminkin.

Jos ylläolevat tehtävät on jo tehty, tutustukaa yhdessä seuraavan viikon materiaaliin :)