

Esimerkki:

Tehtävä 1. Tietojen lisääminen, poistaminen, päivittäminen ja tulostaminen

Luo tietokanta Koulu. Tietokantaan lisätään 3 taulua.

Kurssit-taulu:

kurssiID not null primary key INTEGER
aineID not null primary key VARCHAR(20)
kurssinNimi VARCHAR(30)
oppPisteet INTEGER

Oppilaat-taulu:

oppilasID not null primary key INTEGER
etuNimi VARCHAR(15)
sukuNimi VARCHAR(20)

Arvioinnit:

oppilasID not null primary key INTEGER
kurssiID not null primary key INTEGER
suoritusPvm DATE
arvosana INTEGER

Lisää sen jälkeen kolme kurssia, kolme oppilasta ja anna oppilaille arvosanat, kursseista jotka he ovat suorittaneet.

Luo tämän jälkeen uusi projekti ja avaa tietokantayhteys pääohjelmasta Koulu-tietokantaan.

Lisää pääohjelmasta seuraavat oppilaat Oppilas-tauluun:

4, Inka, Helle
5, Maija, Meikäläinen
6, Samu, Sirkka

Poista oppilasID:llä 2 oleva oppilas Oppilaat-taulusta pääohjelman kautta.

Anna uusille oppilaille arvosanat Arvioinnit-tauluun. Kokeile lähettää kaikki tiedot tietokannan taulukkoon kerralla `statement.addBatch()` ja `statement.executeBatch()`-metodeja käyttämällä.

Päivitä oppilasID:llä 4 olevan henkilön kurssiID:llä 2 olevan kurssin arvosana viitokseksi (5).

Tulosta kaikkien kolmen taulun tiedot komentoriville.

EXTRA:

Tee taulujen tulostaminen metodissa, jota kutsut pääohjelmasta.

```
public static void main(String[] args) throws ClassNotFoundException, SQLException {
    // TODO code application logic here
    String url = "jdbc:derby://localhost:1527/Koulu";
    Class.forName("org.apache.derby.jdbc.ClientDriver");
    System.out.println("Driver ladattu");

    Connection connection = DriverManager.getConnection(url,"root", "root");
    System.out.println("Yhteys luotu");

    Statement statement = connection.createStatement();

    //statement.executeUpdate("INSERT INTO ROOT.OPPILAAT VALUES
(4,'Inka','Helle'");
    //statement.executeUpdate("INSERT INTO ROOT.OPPILAAT VALUES
(5,'Maija','Meikäläinen'");
    //statement.executeUpdate("INSERT INTO ROOT.OPPILAAT VALUES
(6,'Samu','Sirkka'");

    //statement.executeUpdate("DELETE FROM ROOT.OPPILAAT" +
// " WHERE oppilasid=2");

    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (4,1,'2016-01-03',
4)");
    // statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (4,2,'2016-01-03',
2)");
    // statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (4,3,'2016-01-03',
5)");
    // statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (5,1,'2016-01-03',
1)");
    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (5,2,'2016-01-03',
3)");
    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (5,3,'2016-01-03',
2)");
    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (6,1,'2016-01-03',
4)");
    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (6,2,'2016-01-03',
5)");
    //statement.addBatch("INSERT INTO ROOT.ARVIINNIT VALUES (6,3,'2016-01-03',
3)");
    //statement.executeBatch();

    //statement.executeUpdate("UPDATE ROOT.ARVIINNIT SET arvosana=5 WHERE
oppilasID=4 AND kurssiID=2");

    //Suoritetaan kysely

    tulostaTiedot(statement, "ARVIINNIT");
    tulostaTiedot(statement, "KURSSIT");
    tulostaTiedot(statement, "OPPILAAT");

    /** ResultSet resultSet = statement.executeQuery("select * from
ROOT.ARVIINNIT\n" + "");

    //Tulostetaan kyselyn sisältö komentoriville
```

```

        while(resultSet.next()){
            System.out.println(resultSet.getString("oppilasID")+ " "+
resultSet.getString("kurssiID")+ " "+ resultSet.getString("suorituspvm") + " "
                + resultSet.getString("arvosana")
            );
        }
        resultSet = statement.executeQuery("select * from ROOT.OPPILAAT\n" + "");

        //Tulostetaan kyselyn sisältö komentoriville
        while(resultSet.next()){
            System.out.println(resultSet.getString("oppilasID")+ " "+
resultSet.getString("etunimi")+ " "+ resultSet.getString("sukunimi")
            );
        }

        //Tulostetaan kyselyn sisältö komentoriville
        while(resultSet.next()){
            System.out.println(resultSet.getString("kurssiID")+ " "+
resultSet.getString("aine")+ " "+ resultSet.getString("kurssinimi") + " "
                + resultSet.getString("oppisteet")
            );
        }
        /*
        connection.close();

    }
    //muuttujan lisääminen kyselyyn!!
    public static void tulostaTiedot(Statement statement, String tableName) throws
SQLException{
        ResultSet resultColumns = statement.executeQuery("SELECT * FROM
ROOT."+tableName+"");
        ResultSetMetaData rsmd = resultColumns.getMetaData();
        int columnNumber = rsmd.getColumnCount();

        System.out.println(columnNumber);

        while (resultColumns.next()){
            for(int i = 1; i <= columnNumber; i++){
                System.out.print(resultColumns.getString(i) + " ");
            }
            System.out.println(" ");
        }
    }
}

```

Tehtävä 2. Taulun luominen ja päivittäminen

Tee tehtävät käyttämällä Koulu-tietokantaa.

Luo yhteys Koulu-tietokantaan pääohjelmasta ja suorita seuraavat toiminnot pääohjelmasta.

Lisää tietokantaan taulu Opettajat.

Opettajat-taulu:

opettajaID INTERGER not null primary key

etuNimi VARCHAR(20)

sukuNimi VARCHAR(30)

opetettavaAine VARCHAR(30)

Lisää tauluun kolme opettajaa.

Lisää Kurssit-tiluun sarake opettajaID:lle. OpettajaID on kokonaisluku.

Lisää kurssille opettajat.

Tulosta Kurssit-tilu ja tarkasta, että tiedot ovat päivityneet oikein.

```
public static void main(String[] args) throws ClassNotFoundException, SQLException {
    // TODO code application logic here
    String url = "jdbc:derby://localhost:1527/Koulu";
    Class.forName("org.apache.derby.jdbc.ClientDriver");
    System.out.println("Driver ladattu");

    Connection connection = DriverManager.getConnection(url, "root", "root");
    System.out.println("Yhteys luotu");

    Statement statement = connection.createStatement();

    //statement.execute("CREATE TABLE ROOT.OPETTAJAT (opettajaID INTEGER not null
primary key, etuNimi VARCHAR(20), sukuNimi VARCHAR(30), opetettavaAine VARCHAR(30))");

    // statement.executeUpdate("INSERT INTO ROOT.OPETTAJAT VALUES (1, 'Silja', 'Sade',
'Matematiikka'");
    //statement.executeUpdate("INSERT INTO ROOT.OPETTAJAT VALUES (2, 'Aku', 'Ankka',
'Tietotekniikka ja äidinkieli'");

    // statement.execute("ALTER TABLE ROOT.KURSSIT ADD opettajaID INTEGER");

    statement.executeUpdate("UPDATE ROOT.KURSSIT" +
" SET opettajaID=1" +
" WHERE aine='Matematiikka'");
    statement.executeUpdate("UPDATE ROOT.KURSSIT" +
" SET opettajaID=2" +
" WHERE aine='Tietotekniikka'");
    statement.executeUpdate("UPDATE ROOT.KURSSIT" +
" SET opettajaID=3" +
" WHERE aine='Äidinkieli'");

    connection.close();
}
```

Tehtävä 3. Tietokantayhteys graafisella käyttöliittymällä

Toteuta harjoitusten 53 ja 54 pohjalta jokin tehtävien tietokantatoiminto graafisella käyttöliittymällä.

Liitä palautukseen mukaan myös kuva graafisesta käyttöliittymästäsi, jos et palauta tehtävää zipattuna projektikansiona!

Tehtävä 4. Kirjautuminen

Luo uusi tietokanta, jonne voit luoda valmiiksi yhden taulun nimeltään Kayttajat.

Kayttajat-tilu:

kayttajaID INTERGER not null primary key

kayttajaTunnus VARCHAR(30)

salasana VARCHAR(10)

Luo uusi projekti, jossa luot tietokantayhteyden luomaasi tietokantaan.

Lisää käyttäjiä Käyttäjät-tauluun, jotta voit kokeilla kirjautua heidän tiedoillaan kirjautumisikkunassa.

Tee haku salasanoista Kayttajat-taulusta, jossa kayttajatunnus vastaa kirjautumisikkunaan annettua kayttajatunnusta.

Vertaa sen jälkeen annettua salasanaa ja tietokannasta haettua salasanaa. Jos salasanat ovat samat (mieti hyväksytykö isot ja pienet kirjaimet samana ”merkinä” vai ovatko esimerkiksi ”Salasana” ja ”salasana” eri sanoja), ilmoita käyttäjälle ikkunassa ”Kirjautuminen onnistui!”. Vastaavasti, jos salasana ei ole sama ilmoita käyttäjälle ”Kirjautuminen epäonnistui!”

Käyttäjälle voi ilmoittaa myös, jos käyttäjätunnusta ei löydy tietokannasta.

EXTRA: Lisää toiminto, jossa voit lisätä graafisen käyttöliittymän kautta käyttäjiä Kayttajat-tauluun.

```
private void kirjauduNappiActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String ktunnus = tunnusKentta.getText();  
    String salasana = salasanaKentta.getText();  
  
    String url = "jdbc:derby://localhost:1527/Koulu";  
    try {  
        Class.forName("org.apache.derby.jdbc.ClientDriver");  
  
        System.out.println("Driver ladattu");  
  
        Connection connection = DriverManager.getConnection(url,"root", "root");  
  
        System.out.println("Yhteys luotu");  
  
        Statement statement = connection.createStatement();  
  
        JFrame frame = new JFrame("Huomautus");  
  
        System.out.println(salasana);  
        System.out.println(ktunnus);  
        //String sql = ("SELECT SALASANA FROM ROOT.KAYTTAJAT WHERE nimi = " +  
ktunnus);  
        String sql = ("SELECT SALASANA FROM ROOT.KAYTTAJAT WHERE nimi =  
''+ktunnus+'''");  
        System.out.println(sql);  
  
        ResultSet salasanaHaku = statement.executeQuery(sql);  
        String annettuSalasana = "";  
        System.out.println(salasanaHaku.toString());  
        while(salasanaHaku.next()){  
            annettuSalasana = salasanaHaku.getString("salasana");  
  
        }  
    }  
}
```

```
if(annettuSalasana.equals(salasana)){
    JOptionPane.showMessageDialog(frame, "Kirjautuminen onnistui!");
}else{
    JOptionPane.showMessageDialog(frame, "Kirjautuminen epäonnistui!");
}

connection.close();

} catch (ClassNotFoundException ex) {
    } catch (SQLException ex) {
        }
    }
}
```