

Tietokannat II -kurssin harjoitustyö

Jyri Lehtonen (72039), jkoleh@utu.fi
Azad Hajipour (72187), azhaji@utu.fi

10.6.2007

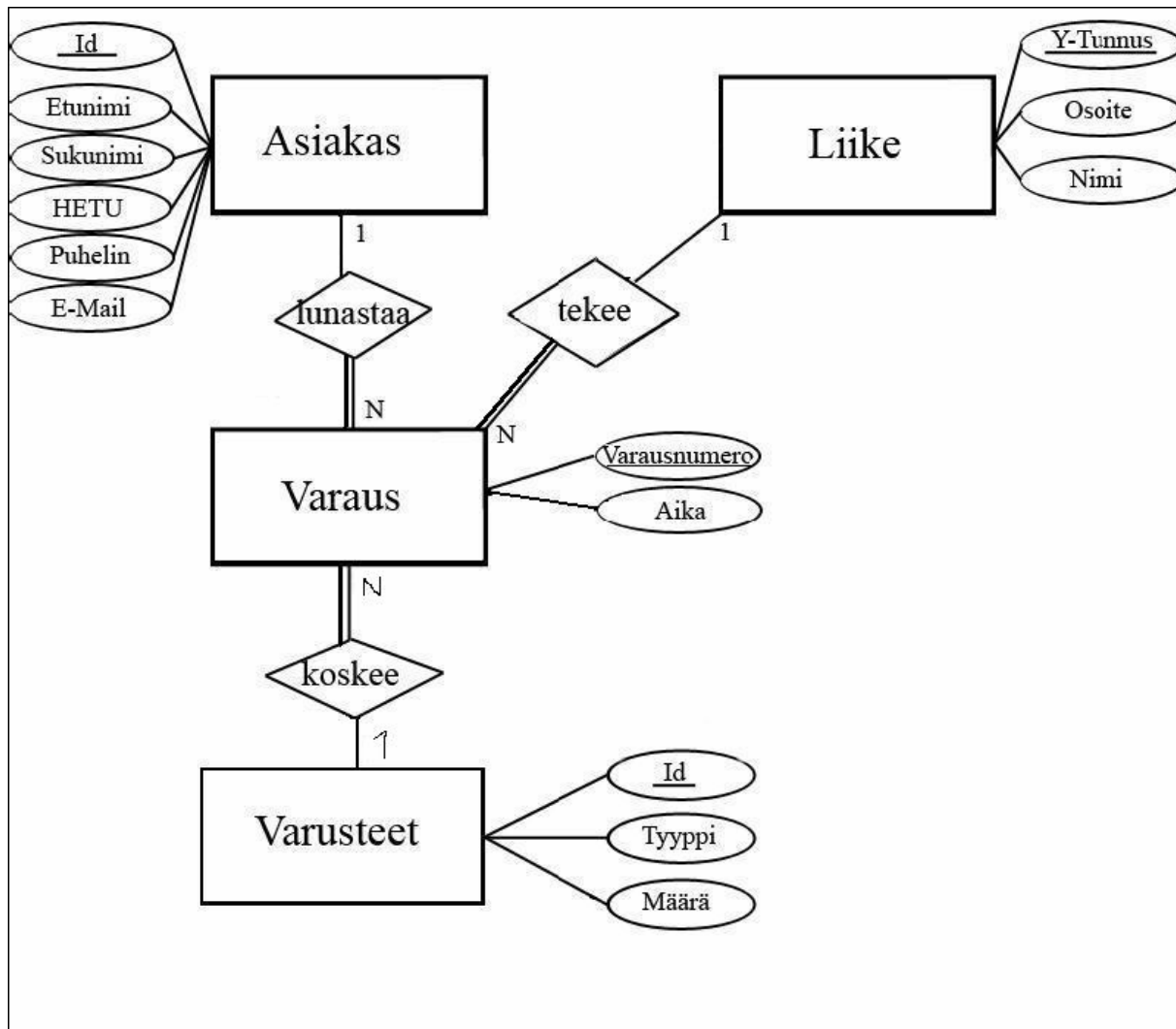
Sisältö

1. Tietokannan kuvaus	1
1.1 Tietokannan rakenne	1
1.2 Relaatiokaava ja viiteavaimet	2
1.3 Relaatioiden kuvaus	2
2. Luokkien kuvaus	3
2.1 Asiakas	3
2.2 Varaus	5
2.3 Liike	6
2.4 Varusteet	6
2.5 Tietokanta.....	7
3. Testaus	9
 Liitteet:	
A Tietokannan luontikomennot	12
B Tietokannan sisältö.....	13
C Ohjelmointilistaukset	14

1. Tietokannan kuvaus

1.1 Tietokannan rakenne

Aikomuksenamme on toteuttaa pienen erikoismyymälän tietokanta. Tavoitteena on mallintaa Mustelma Oy:n toimintaa. Kyseisen firman toimenkuvaan kuuluu paintball varusteiden ja aseiden vuokraus kolmessa toimipisteessä, Turussa, Helsingissä ja Tampereella. Otamme tässä kuvauksessa huomioon vain ”arkipäiväiset” tapahtumat, jolloin laajemmat tapahtumat, kuten maahantuojiin liittyvät toimenpiteet ja varusteiden/tilojen huoltoa koskevat toimenpiteet jäävät kuvauksen ulkopuolelle. Tietokannan ER-kaavio on esitetty alla kuvassa 1.



Kuva 1: Tietokannan ER-kaavio.

1.2 Relaatiokaava ja viiteavaimet

Tavallisia muunnossääntöjä käyttäen kuvassa 1 esitetty ER-kaavio muunnettiin seuraaviksi relaatioiksi:

Asiakas (Id, Etunimi, Sukunimi, SOTU, Puhelin, EMail)

Varaus (Varausnumero, Aika, Varaaja, Varattava, Varauspaikka)

Liike (YTunnus, Osoite, Nimi)

Varusteet (Id, Tyyppi, Määrä)

ER-kaavion perusteella määräytyivät myös tietokannan viiteavaimet, jotka on kohteineen lueteltu alla. Kuvassa 2 on hahmoteltu graafisesti ER-to-Relational muunnos.

Varaaja -> asiakas.Id

Varattava -> varusteet.Id

Varauspaikka -> liike.YTunnus



Kuva 2: ER-to-Relational

1.3 Relaatioiden kuvaus

- **Asiakas**: Vastaa yrityksessä asioivaa asiakashenkilöä. Relaatiolla on pääavain Id, joka vastaa asiakasnumeroa ja joukko attribuutteja, jotka kuvastavat tarkempaa tietoa henkilöstä.
- **Varaus**: Relaatio liittää toisiinsa muut tietokannan relaatiot. Relaatiolla on pääavaimena Varausnumero ja attribuuttina Aika. Viiteavaimena ovat varaaja, varattava ja varauspaikka.

- **Liike:** Vastaa yrityksen omistamia liikkeitä, joita on tällä hetkellä kolme kappaletta. Tähän sisältyy päävain YTunnus, joka vastaa yritystunnusta ja attribuutteina Osoite ja Nimi.
- **Varusteet:** Relaatio sisältää liikkeiden omistamat varusteet, jotka voidaan vuokrata asiakkaille. Päävaimena on Id joka vastaa tuotenumeroa ja attribuutteina on varusteen Tyyppi ja Määrä.

2. Luokkien kuvaus

2.1 Asiakas

```
//Asiakasta mallintava luokka.
public class Asiakas {

    /**
     * LUOKKAMUUTTUJAT
     */

    //Asiakkaan Id tietokannassa.
    private int id;

    //Asiakkaan etunimi tietokannassa.
    private String etunimi;

    //Asiakkaan sukunimi tietokannassa.
    private String sukunimi;

    //Asiakkaan puhelinnumero tietokannassa.
    private String puhelin;

    //Asiakkaan sosiaaliturvatunnus tietokannassa.
    private String SOTU;

    //Asiakkaan sähköpostiosoite tietokannassa.
    private String Email;
```

```

/****
**** KONSTRUKTORI
****/
public Asiakas(int id, String etunimi, String sukunimi,
                String puhelin, String SOTU, String Email) {}

/****
**** FUNKTIOT
****/

//Palauttaa asiakkaan Id:n tietokannasta.
public int annaId() {}

//Palauttaa asiakkaan etunimen tietokannasta.
public String annaEtunimi() {}

//Palauttaa asiakkaan sukunimen tietokannasta.
public String annaSukunimi() {}

//Palauttaa asiakkaan puhelinnumeron tietokannasta.
public String annaPuhelin() {}

//Palauttaa asiakkaan sosiaaliturvatunnuksen tietokannasta.
public String annaSOTU() {}

//palauttaa asiakkaan sähköpostiosoitteen tietokannasta.
public String annaEmail() {}

/**
 * Palauttaa asiakkaan tilaamat tuotteet.
 * @.pre: Tietokantayhteys on alustettu.
 */
public ArrayList<Varusteet> annaVarusteet() {}

// Palauttaa asiakkaan merkkijonoesityksen.
public String toString() {}

/****
**** PROSEDUURIT
****/

/**
 * Lisää asiakkaan tilauksiin tuotteen.
 * @.pre: (Tietokantayhteys on alustettu) & (tuote != null)
 */
public void lisääVaruste(Varusteet varuste, Liike liike) {}

}

```

2.2 Varaus

```
//Yrityksen varausta mallintava luokka.
public class Varaus {

    /****
     **** LUOKKAMUUTTUJAT
     ****/

    //Varauksen tunnus tietokannassa.
    private int varausnumero;

    //Varauksen tekijä tietokannassa.
    private int varaaaja;

    //Varauksen kohde tietokannassa.
    private int varattava;

    //Varauksen teko paikka tietokannassa.
    private int varauspaikka;

    //Aika jolloin varaus on voimassa.
    private String aika;

    /****
     **** KONSTRUKTORI
     ****/
    public Varaus(int varausnumero, int varaaaja, int varattava,
                  int varauspaikka, String aika) {}

    /****
     **** FUNKTIOT
     ****/

    //Palauttaa varauksen tunnuksen tietokannasta.
    public int annaVarausnumero() {}

    //Palauttaa varauksen tekijän tunnuksen tietokannasta.
    public int annaVaraaaja() {}

    //Palauttaa varauksen kohteen tunnuksen tietokannasta.
    public int annaVarattava() {}

    //Palauttaa varauksen tekopaikan tunnuksen tietokannasta.
    public int annaVarauspaikka() {}

    //Palauttaa varauksen ajankohdan tietokannasta.
    public String annaAika() {}

}
```

2.3 Liike

```
//Yrityksen liikettä mallintava luokka.
public class Liike {

    /**
     * LUOKKAMUUTTUJAT
     */

    //Yrityksen liikkeen tunnus tietokannassa
    private int YTunnus;

    //Yrityksen liikkeen osoite tietokannassa.
    private String Osoite;

    //Yrityksen liikkeen nimi tietokannassa.
    private String Nimi;

    /**
     * KONSTRUKTORI
     */
    public Liike(int YTunnus, String Osoite, String Nimi) {}

    /**
     * FUNKTIOT
     */

    //Palauttaa liikkeen yritystunnuksen tietokannasta.
    public int annaYTunnus() {}

    //Palauttaa liikkeen osoitteen tietokannasta.
    public String annaOsoite() {}

    //Palauttaa liikkeen nimen tietokannasta.
    public String annaNimi() {}

    //Palauttaa liikkeen merkkijonona.
    public String toString() {}

}
```

2.4 Varusteet

```
//Yrityksen varusteita mallintava luokka.
public class Varusteet {

    /**
     * LUOKKAMUUTTUJAT
     */

    //Varusteen tunnus tietokannassa.
    private int Id;
```



```

//Varusteen tiedot tietokannassa.
private String Tyyppi;

//Varusteen lukumäärä tietokannassa.
private int Maara;

/****
**** KONSTRUKTORI
****/
public Varusteet(int Id, String Tyyppi, int Maara) {}

/****
**** FUNKTIOT
****/

//Palauttaa varusteen tunnuksen tietokannasta.
public int annaId() {}

//Palauttaa varusteen tyyppin tietokannasta.
public String annaTyyppi() {}

//Palauttaa varusteen lukumäärän tietokannasta.
public int annaMaara() {}

//Palauttaa varusteet merkkijonona.
public String toString() {}
}

```

2.5 Tietokanta

```

// Liitetään SQL toiminnot luokkaan.
import java.sql.*;
import java.io.*;
import java.util.*;

//Tietokantaa mallintava luokka.
public class Tietokanta {

    //Luokkamuuttujat
    private static Connection mustelma;
    private static int varausnumero=2000;

    //Tietokannan toimintaa testauttava metodi.
    public static void main(String... args) {}
}

```

```

/*****
**** YHTEYDEN MUODOSTUS
**** TIETOKANTAAN
****/

    public static void alusta() {}

    public static void sulje() {}

/*****
**** TIETOKANTA TOIMINNOT
**** (FUNKTIOT)
****/

//Palauttaa tietyn asiakkaan tekemän varauksen varusteet.
public static ArrayList<Varusteet> annaTilaukset(Asiakas asiakas) {}

//Palauttaa koko yrityksen varusteet järjestettynä lukumäärällä.
    public static ArrayList<Varusteet> annaVarusteet() {}

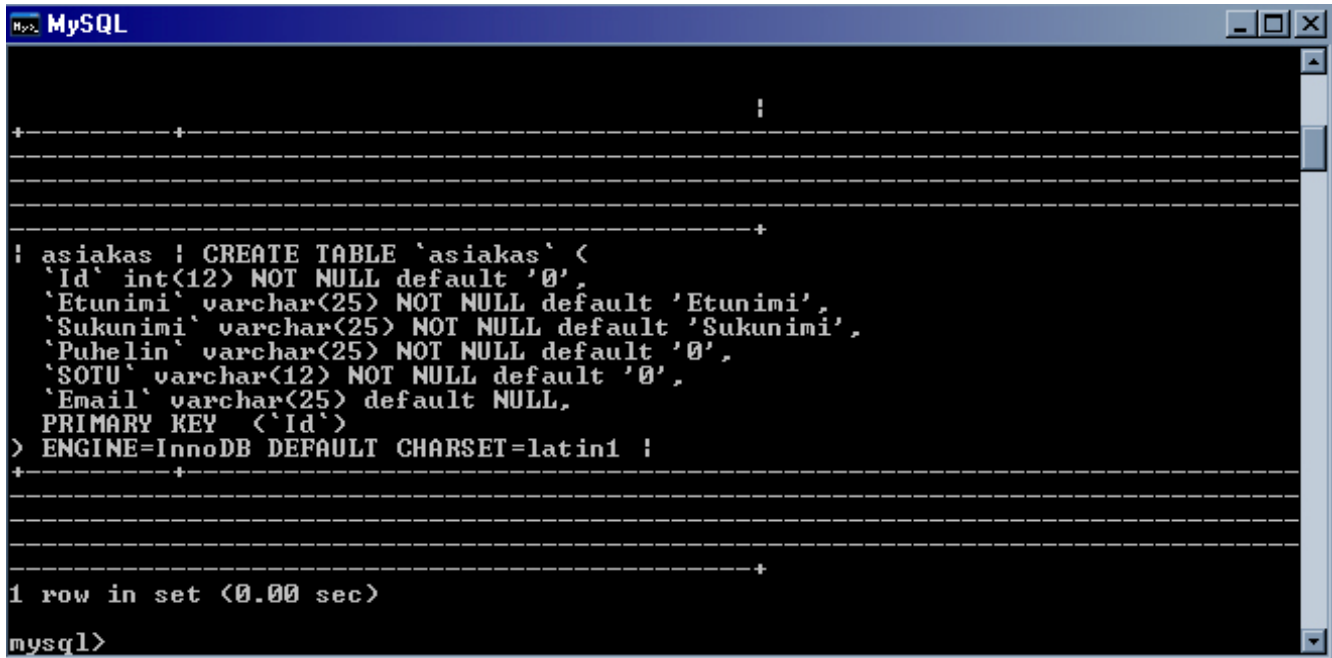
//Lisää uuden varauksen tietokantaan.
public static void lisaaVaraus(Asiakas asiakas,
Varusteet varusteet, Liike liike) {}

}

```

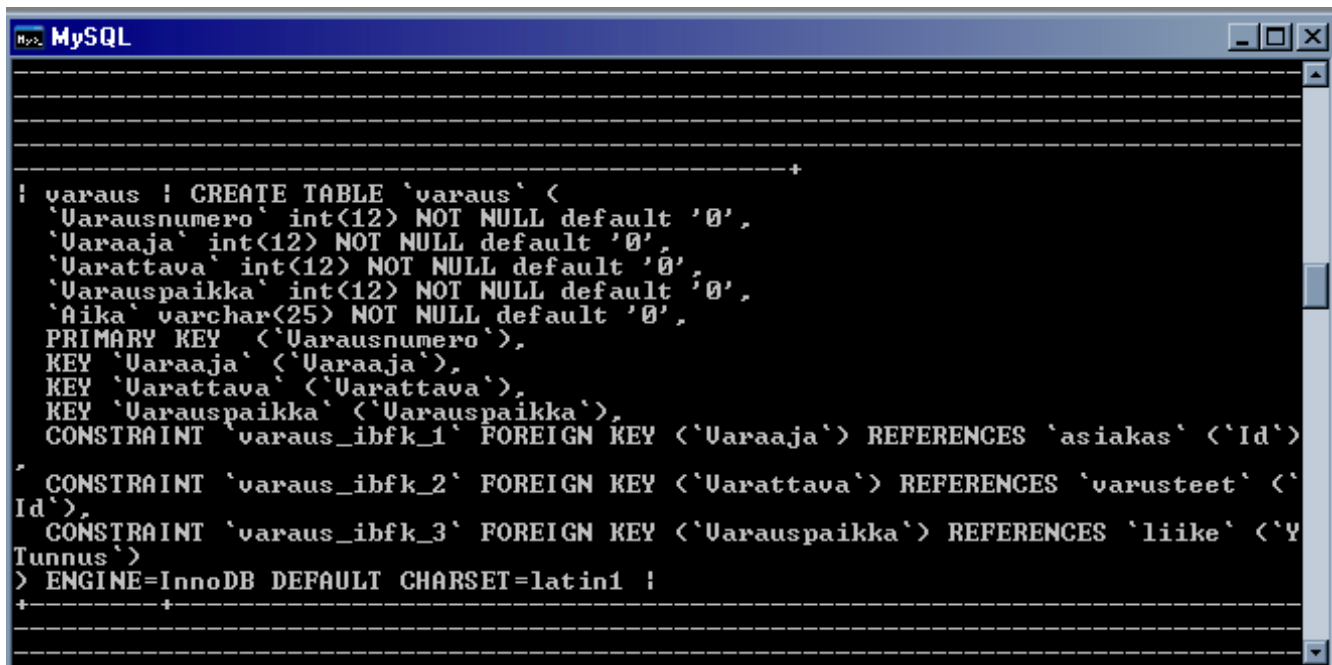

A Tietokannan luontikomennot

Alla olevat kuvat on otettu ulos tietokannasta komennolla `show create table`, joka mallintaa samaa kuin komennot jolla tietokannan taulukko on luotu.



```
MySQL
+-----+
|
+-----+
| asiakas | CREATE TABLE `asiakas` (  
| `Id` int(12) NOT NULL default '0',  
| `Etunimi` varchar(25) NOT NULL default 'Etunimi',  
| `Sukunimi` varchar(25) NOT NULL default 'Sukunimi',  
| `Puhelin` varchar(25) NOT NULL default '0',  
| `SOTU` varchar(12) NOT NULL default '0',  
| `Email` varchar(25) default NULL,  
| PRIMARY KEY (`Id`)  
> ENGINE=InnoDB DEFAULT CHARSET=latin1 ;  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

Kuva 4: Asiakas taulun luontikomento



```
MySQL
+-----+
|
+-----+
| varaus | CREATE TABLE `varaus` (  
| `Varausnumero` int(12) NOT NULL default '0',  
| `Uaraaja` int(12) NOT NULL default '0',  
| `Varattava` int(12) NOT NULL default '0',  
| `Varauspaikka` int(12) NOT NULL default '0',  
| `Aika` varchar(25) NOT NULL default '0',  
| PRIMARY KEY (`Varausnumero`),  
| KEY `Uaraaja` (`Uaraaja`),  
| KEY `Varattava` (`Varattava`),  
| KEY `Varauspaikka` (`Varauspaikka`),  
| CONSTRAINT `varaus_ibfk_1` FOREIGN KEY (`Uaraaja`) REFERENCES `asiakas` (`Id`)  
|  
| CONSTRAINT `varaus_ibfk_2` FOREIGN KEY (`Varattava`) REFERENCES `varusteet` (`  
Id`),  
| CONSTRAINT `varaus_ibfk_3` FOREIGN KEY (`Varauspaikka`) REFERENCES `liike` (`Y  
Tunnus`)  
> ENGINE=InnoDB DEFAULT CHARSET=latin1 ;  
+-----+

```

Kuva 5: Varaus taulun luontikomento

```

+-----+
|
+-----+
| liike | CREATE TABLE `liike` (
| `Ytunnus` int(12) NOT NULL default '0',
| `Osoite` varchar(25) NOT NULL default 'Osoite',
| `Nimi` varchar(25) NOT NULL default 'Nimi',
| PRIMARY KEY (`Ytunnus`)
| > ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
+-----+
|
+-----+
| 1 row in set (0.00 sec)
|
mysql>

```

Kuva 6: Liike taulun luontikomento

```

+-----+
|
+-----+
| varusteet | CREATE TABLE `varusteet` (
| `Id` int(12) NOT NULL default '0',
| `Tyyppi` varchar(25) NOT NULL default 'Tyyppi',
| `Maara` int(12) NOT NULL default '0',
| PRIMARY KEY (`Id`)
| > ENGINE=InnoDB DEFAULT CHARSET=latin1 ;
+-----+
|
+-----+
| 1 row in set (0.00 sec)
|
mysql>

```

Kuva 7: Varusteet taulun luontikomento.

B Tietokannan sisältö

Alla on otettu otteet tietokannasta, jokaisen taulun sisältämistä arvoista.

Testausjärjestelyjen lisää varaus metodin vaikutus näkyy varaus taulukossa (varausnumero 2001).

```

1 row in set (0.00 sec)
mysql> select *
-> from asiakas;
+----+-----+-----+-----+-----+-----+
| Id | Etunimi | Sukunimi | Puhelin | SOTU | Email |
+----+-----+-----+-----+-----+-----+
| 1 | Pertti | Uirtanen | 040-4732821 | 010370-312P | perde@gmail.com |
| 2 | Simo | Ehtamo | 040-1241513 | 140282-002S | simbe@hotmail.com |
| 3 | Esa | Mustajoki | 050-4982710 | 301279-122E | musta@turku.fi |
| 4 | Leena | Punkka | 040-9938355 | 051083-209L | punkka@utu.fi |
| 5 | Jirka | Salo | 040-6651211 | 100182-310J | jirkasalo@turku.fi |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
mysql>

```

Kuva 8: Asiakas taulun sisältö

```
mysql> select * from varaus;
+-----+-----+-----+-----+-----+
| Varausnumero | Varaaja | Varattava | Varauspaikka | Aika |
+-----+-----+-----+-----+-----+
| 1000 | 1 | 1 | 100201 | 10032007 |
| 1001 | 1 | 5 | 100201 | 10032007 |
| 2001 | 1 | 5 | 100201 | 20082007 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Kuva 9: Varaus taulun sisältö

```
mysql> select *
-> from liike;
+-----+-----+-----+
| YTunnus | Osoite | Nimi |
+-----+-----+-----+
| 100201 | Aurankatu 13, Turku | Mustelma |
| 100202 | Piispankatu 6, Tampere | No Pain No Gain |
| 100203 | Kasarmintie 4, Helsinki | Urban Army |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Kuva 10: Liike taulun sisältö

```
mysql> select * from varusteet;
+-----+-----+-----+
| Id | Tyyppi | Maara |
+-----+-----+-----+
| 1 | Suojat | 13 |
| 2 | Pistoolit | 10 |
| 3 | Kiväärit | 6 |
| 4 | Haulikot | 4 |
| 5 | Automaatit | 11 |
| 6 | Savukranaatit | 30 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Kuva 11: Varusteet taulun sisältö

C Ohjelmointilistaukset

Lähdekoodien listaus alkaa seuraavalta sivulta.

```

import java.util.*;

//Asiakasta mallintava luokka.
public class Asiakas {

    /***
     *** LUOKKAMUUTTUJAT
     ***/

    //Asiakkaan Id tietokannassa.
    private int id;

    //Asiakkaan etunimi tietokannassa.
    private String etunimi;

    //Asiakkaan sukunimi tietokannassa.
    private String sukunimi;

    //Asiakkaan puhelinnumero tietokannassa.
    private String puhelin;

    //Asiakkaan sosiaaliturvatunnus tietokannassa.
    private String SOTU;

    //Asiakkaan sähköpostiosoite tietokannassa.
    private String Email;

    /***
     *** KONSTRUKTORI
     ***/
    public Asiakas(int id, String etunimi, String sukunimi,
                   String puhelin, String SOTU, String Email) {
        this.id = id;
        this.etunimi = etunimi;
        this.sukunimi = sukunimi;
        this.puhelin = puhelin;
        this.SOTU = SOTU;
        this.Email = Email;
    }

    /***
     *** FUNKTIOT
     ***/

    //Palauttaa asiakkaan Id:n tietokannasta.
    public int annaId() {
        return id;
    }

    //Palauttaa asiakkaan etunimen tietokannasta.
    public String annaEtunimi() {
        return etunimi;
    }
}

```

```

//Palauttaa asiakkaan sukunimen tietokannasta.
public String annaSukunimi() {
    return sukunimi;
}

//Palauttaa asiakkaan puhelinnumeron tietokannasta.
public String annaPuhelin() {
    return puhelin;
}

//Palauttaa asiakkaan sosiaaliturvatunnuksen tietokannasta.
public String annaSOTU() {
    return SOTU;
}

//palauttaa asiakkaan sähköpostiosoitteen tietokannasta.
public String annaEmail() {
    return Email;
}

/**
 * Palauttaa asiakkaan tilaamat tuotteet.
 * @.pre: Tietokantayhteys on alustettu.
 */
public ArrayList<Varusteet> annaVarusteet() {
    return Tietokanta.annaTilaukset(this);
}

//Palauttaa asiakkaan merkkinojona.
public String toString() {
    return (annaId() +", "+ annaEtunimi() +", "+ annaSukunimi() +", "+
        annaPuhelin() +", "+ annaSOTU() +", "+ annaEmail());
}

/****
**** PROSEDUURIT
****/

/**
 * Lisää asiakkaan tilauksiin tuotteen.
 * @.pre: (Tietokantayhteys on alustettu) & (tuote != null)
 */
public void lisaaVaruste(Varusteet varuste, Liike liike) {
    Tietokanta.lisaaVaraus(this, varuste, liike);
}
}

```

```

//Yrityksen varausta mallintava luokka.
public class Varaus {

    /****
     **** LUOKKAMUUTTUJAT
     ****/

        //Varauksen tunnus tietokannassa.
        private int varausnumero;

        //Varauksen tekijä tietokannassa.
        private int varaaaja;

        //Varauksen kohde tietokannassa.
        private int varattava;

        //Varauksen teko paikka tietokannassa.
        private int varauspaikka;

        //Aika jolloin varaus on voimassa.
        private String aika;

    /****
     **** KONSTRUKTORI
     ****/
    public Varaus(int varausnumero, int varaaaja, int varattava,
                  int varauspaikka, String aika) {

        this.varausnumero=varausnumero;
        this.varaaaja=varaaaja;
        this.varattava=varattava;
        this.varauspaikka=varauspaikka;
        this.aika=aika;
    }

    /****
     **** FUNKTIOT
     ****/

        //Palauttaa varauksen tunnuksen tietokannasta.
        public int annaVarausnumero() {
            return varausnumero;
        }

        //Palauttaa varauksen tekijän tunnuksen tietokannasta.
        public int annaVaraaaja() {
            return varaaaja;
        }

        //Palauttaa varauksen kohteen tunnuksen tietokannasta.
        public int annaVarattava() {
            return varattava;
        }
}

```

```

//Palauttaa varauksen tekopaikan tunnuksen tietokannasta.
public int annaVarauspaikka() {
    return varauspaikka;
}

//Palauttaa varauksen ajankohdan tietokannasta.
public String annaAika() {
    return aika;
}
}

```

```

//Yrityksen liikettä mallintava luokka.
public class Liike {
    /****
     **** LUOKKAMUUTTUUJAT
     ****/

    //Yrityksen liikkeen tunnus tietokannassa
    private int YTunnus;

    //Yrityksen liikkeen osoite tietokannassa.
    private String Osoite;

    //Yrityksen liikkeen nimi tietokannassa.
    private String Nimi;

    /****
     **** KONSTRUKTORI
     ****/
    public Liike(int YTunnus, String Osoite, String Nimi) {
        this.YTunnus=YTunnus;
        this.Osoite=Osoite;
        this.Nimi=Nimi;
    }

    /****
     **** FUNKTIOT
     ****/

    //Palauttaa liikkeen yritystunnuksen tietokannasta.
    public int annaYTunnus() {
        return YTunnus;
    }

    //Palauttaa liikkeen osoitteen tietokannasta.
    public String annaOsoite() {
        return Osoite;
    }

    //Palauttaa liikkeen nimen tietokannasta.
    public String annaNimi() {
        return Nimi;
    }
}

```

```

//Palauttaa liikkeen merkkijonona.
public String toString() {
    return (annaYTunnus() +", "+ annaOsoite() +", "+ annaNimi());
}
}

```

```

//Yrityksen varusteita mallintava luokka.
public class Varusteet {
    /****
    **** LUOKKAMUUTTUJAT
    ****/

    //Varusteen tunnus tietokannassa.
    private int Id;

    //Varusteen tiedot tietokannassa.
    private String Tyyppi;

    //Varusteen lukumäärä tietokannassa.
    private int Maara;

    /****
    **** KONSTRUKTORI
    ****/
    public Varusteet(int Id, String Tyyppi, int Maara) {
        this.Id = Id;
        this.Tyyppi = Tyyppi;
        this.Maara = Maara;
    }

    /****
    **** FUNKTIOT
    ****/

    //Palauttaa varusteen tunnuksen tietokannasta.
    public int annaId() {
        return Id;
    }

    //Palauttaa varusteen tyyppin tietokannasta.
    public String annaTyyppi() {
        return Tyyppi;
    }

    //Palauttaa varusteen lukumäärän tietokannasta.
    public int annaMaara() {
        return Maara;
    }

    //Palauttaa varusteet merkkijonona.
    public String toString() {
        return (annaId() +", "+ annaTyyppi() +", "+ annaMaara());
    }
}

```

```

// Liitetään SQL toiminnot luokkaan.
import java.sql.*;
import java.io.*;
import java.util.*;

//Tietokantaa mallintava luokka.
public class Tietokanta {

    //Luokkamuuttujat
    private static Connection mustelma;
    private static int varausnumero=2000;

    //Tietokannan toimintaa testauttava metodi.
    public static void main(String... args) {

        Tietokanta.alusta();

        System.out.println();
        ArrayList<Varusteet> varusteet = Tietokanta.annaVarusteet();
        System.out.println("Liikkeissä olevat varusteet suuruus
            järjestyksessä: ");
        System.out.println(varusteet);
        System.out.println();

        System.out.println("****-****-****-****-****-****-****-****-****-
            ****-****-****-****");

        System.out.println();
        Asiakas asiakas = new Asiakas(1, "Pertti", "Virtanen",
            "040-4732821", "010370-312P", "perde@gmail.com");
        System.out.println("Asiakkaan: " + asiakas);
        System.out.println("Tilaamat tuotteet: " +
            asiakas.annaVarusteet());
        System.out.println();

        System.out.println("****-****-****-****-****-****-****-****-****-
            ****-****-****-****");

        System.out.println();

        Liike liike = new Liike(1002001, "Aurankatu 13, Turku",
            "Mustelma");
        Varaus varaus = new Varaus(varausnumero, 1,5,100201,
            "10-12-2007");
        System.out.println("Varauksen lisäys: ");
        varausnumero=varausnumero+1;
        asiakas.lisaaVaruste(varusteet.get(2), liike);

        System.out.println();

        Tietokanta.sulje();

    }
}

```

```

/*****
**** YHTEYDEN MUODOSTUS
**** TIETOKANTAAN
****/

public static void alusta() {
    try {

        Class.forName("com.mysql.jdbc.Driver").newInstance();
        mustelma = DriverManager.getConnection("
jdbc:mysql://localhost:3306/mustelma","root", "SilverOne");

        System.out.println("*****");
        System.out.println("*** Yhdistetty Tietokantaan ***");
        System.out.println("*****");
    }
    catch (Exception e) {
        System.out.println("*** Virhe yhdistyksessä *** ");
        System.out.println(e);
    }
}

public static void sulje() {
    try {
        mustelma.close();

        System.out.println("*****");
        System.out.println("*** Yhteys Tietokantaan Terminoitu ***");
        System.out.println("*****");
    }
    catch (SQLException e) {
        System.out.println("*** Virhe tietokannan sulkemisessa ***");
        System.out.println(e);
    }
}

/*****
**** TIETOKANTA TOIMINNOT
**** (FUNKTIOT)
****/

//Palauttaa tietyn asiakkaan tekemän varauksen varusteet.
public static ArrayList<Varusteet> annaTilaukset(Asiakas asiakas) {
    ArrayList<Varusteet> varusteet = new ArrayList<Varusteet>();
    try {
        Statement lause = mustelma.createStatement();
        String kysely = "SELECT DISTINCT varusteet.Id,
                        varusteet.Tyyppi, varusteet.Maara"
            + " FROM varaus, varusteet, asiakas "
            + " WHERE varusteet.id = varaus.varattava "
            + " AND varaus.varaaja = " + asiakas.annaId();

        ResultSet tulos = lause.executeQuery(kysely);
    }
}

```

```

while (tulos.next()) {
    varusteet.add(new Varusteet(
        tulos.getInt("Id"),
        tulos.getString("Tyyppi"),
        tulos.getInt("Maara")));
}

tulos.close();
lause.close();
}
catch (Exception e) {
    System.out.println("*** VIRHE TIETOKANNASSA ***");
    System.out.println(e);
}
return varusteet;
}

```

```

//Palauttaa koko yrityksen varusteet järjestettynä lukumäärällä.
public static ArrayList<Varusteet> annaVarusteet() {
    ArrayList<Varusteet> varusteet = new ArrayList<Varusteet>();
    try {
        Statement lause = mustelma.createStatement();
        String kysely = "SELECT DISTINCT *"
            + " FROM varusteet"
            + " ORDER BY Maara"
            + " DESC" ;

        ResultSet tulos = lause.executeQuery(kysely);
        while (tulos.next()) {
            varusteet.add(new Varusteet(
                tulos.getInt("Id"),
                tulos.getString("Tyyppi"),
                tulos.getInt("Maara")));
        }
        tulos.close();
        lause.close();
    }
    catch (Exception e) {
        System.out.println("*** VIRHE TIETOKANNASSA ***");
        System.out.println(e);
    }
    return varusteet;
}

```

```

//Lisää uuden varauksen tietokantaan.
public static void lisaaVaraus(Asiakas asiakas, Varusteet varusteet,
                               Liike liike) {
    try{
        Statement lause = mustelma.createStatement();
        String kysely = "INSERT INTO varaus VALUES ("
            + varausnumero + (" , ")
            + asiakas.annaId() + (" , ")
            + varusteet.annaId() + (" , ")
            + 100201 + (" , ")
            + 20082007 + (" ");
        lause.executeUpdate(kysely);
        lause.close();

        System.out.println("Varaus lisätty tietokantaan.");
        System.out.println(" Varauksen tiedot: "+
            asiakas.annaEtunimi() + " " +
            asiakas.annaSukunimi());
        System.out.println(" Varausnumero: " + varausnumero);
    }

    catch (Exception e) {
        System.out.println("*** VIRHE TIETOKANNASSA *** ");
        System.out.println(e);
    }
}
}

```