CAL 3 - DATABASES

2014 - 2015

CARLOS JIMÉNEZ SÁNCHEZ – CRISTIAN KNELLEKEN BARROSO

ÍNDICE

TRIGGERS

We create a series of triggers to provide functionality to the fundamentals and use it to efficiently manage the business logic of the company Mediamarkt.

/* Triggers work correctly*/

```
--Bill
create or replace function my_bill() returns trigger as $bill$
        begin
        if  new.number_bill is null then
                raise exception 'Not insert number of bill';
        end if;
        if new.shopping_number is null then
                raise exception '% Not show number of shopping',new.shopping_number;
        end if;

        return  new;
        end;
$bill$ language plpgsql;

create trigger my_bill before insert on bill
for each row --para cada linea
execute procedure my_bill();

                        /*Mostramos la raise exception*/
insert into bill (number_bill,shopping_number)
values (2000,null);
```

```
--Packs
create or replace function my_pack() returns trigger as $pack$

        begin

        if  new.id is null then

                raise exception 'Not insert ID';

        end if;

        if new.price is null then

                raise exception '% Don´t insert price',new.price;

        end if;


        return  new;

        end;

$pack$ language plpgsql;


create trigger pack before insert or update on packs

for each row --para cada linea

execute procedure my_pack();

                                /*Con este codigo podemos insertar y actualizar packs, ademas
usando null provocamos excepciones */

insert into packs (id,price)

values(100011,2000);


update packs set

id =100011,

price = 2000

where id = 100023
```

```
select * from packs



--Offer Tiene problemas
create function my_offer() returns trigger as $offer$
declare
        x integer;
        y integer;
begin

if new.id_offer != old.id_offer then
        select count(*) into x
        from offer
        where new.id_offer = offer.id_offer;
        if (x = 0) then
                raise exception 'Cant show';
        end if;
 end if;
return old;
end;
&offer&
language 'plpgsql';


Create trigger my_offer after insert or update or delete on offer
for each row
execute procedure my_offer();
```

```
--Services

create or replace function my_service() returns trigger as $service$
        begin
        if  new.code is null then
                raise exception 'Not insert code';
        end if;
        if new.name is null then
                raise exception '% Don´t insert name',new.price;
        end if;
        if new.description is null then
                raise exception ' Not insert description',new.price;
        end if;
        /*if new.additional_cost_with_VAT is null then
                raise exception '% Don´t insert additional_cost_with_VAT',new.price;
        end if;  */

        return  new;
        end;
$service$ language plpgsql;

create trigger service2 before insert or update or delete on services
for each row --para cada linea
execute procedure my_service();
                                /*Con este codigo podemos insertar y actualizar services, ademas usando null provocamos excepciones */
insert into services (code,name,description)
values(null,'ServiceVIP2','New Service two');

delete from "services"
where new."code" = 15000;
```

```
update services set
code = null,
description = 'ServiceVIP2'
where code = 15000


select * from services

--Generated code product

CREATE function generated_customer() RETURNS trigger AS $customer$
DECLARE
nifCustomer INTEGER;
BEGIN
SELECT count(*) INTO nifCustomer
FROM product;
UPDATE customer SET nif = nifCustomer WHERE new.nif = customer.nif;
RAISE NOTICE 'Code customer generated successfully';
RETURN new;
END;
$customer$
LANGUAGE plpgsql;
CREATE TRIGGER generated_customer AFTER INSERT ON customer
FOR EACH ROW
execute procedure generated_customer();

--Generated code offer

CREATE function generated_offer() RETURNS trigger AS
DECLARE
```

```
idOffer INTEGER;

BEGIN

SELECT count(*) INTO idOffer FROM offer;


UPDATE offer SET id = idOffer WHERE new.id = offer.id;

RAISE NOTICE 'Code offer generated successfully';

RETURN new;

END;

LANGUAGE plpgsql;

CREATE TRIGGER generated_offer AFTER INSERT ON offer

FOR EACH ROW

execute procedure generated_offer();
```

CONNECTING THE DATABASE WITH JAVA

In this section we proceed to the creation of a JAVA application to connect to the database and allows us to perform queries described in PECL2. The source of this connection is as follows:

```
package CAL3;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;



public class CAL3 extends javax.swing.JFrame {
```

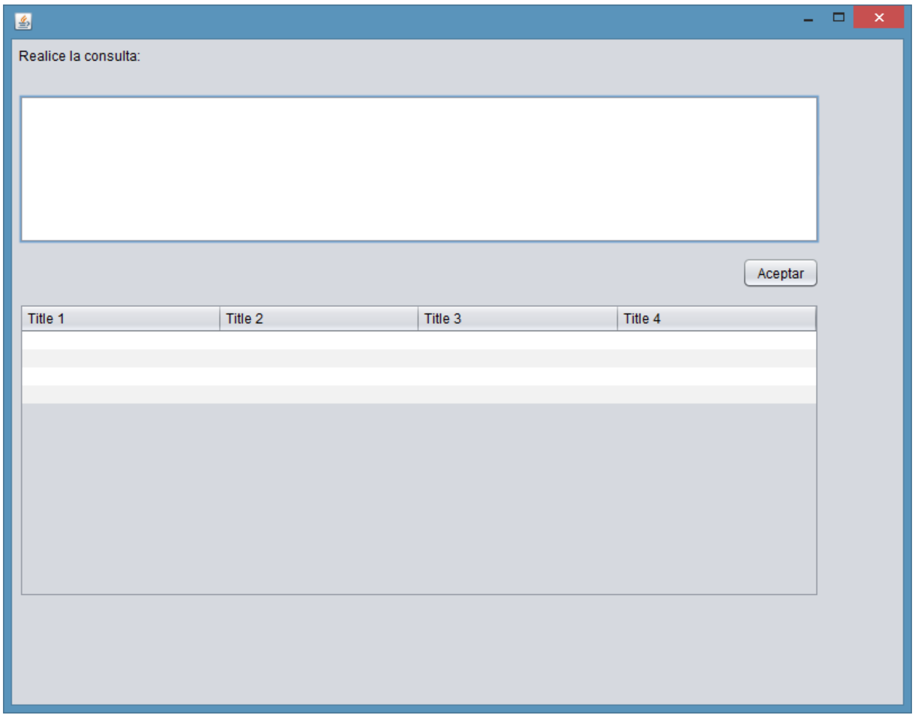
```
public CAL3() {
    initComponents();
}
/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea_SQL = new javax.swing.JTextArea();
    jButton_Aceptar = new javax.swing.JButton();
    jScrollPane2 = new javax.swing.JScrollPane();
    jTable_Datos = new javax.swing.JTable();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jTextArea_SQL.setColumns(20);
    jTextArea_SQL.setRows(5);
    jScrollPane1.setViewportView(jTextArea_SQL);

    jButton_Aceptar.setText("Aceptar");
    jButton_Aceptar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton_AceptarActionPerformed(evt);
        }
    });
    jTable_Datos.setModel(new javax.swing.table.DefaultTableModel(
      new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
      },
      new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
      }
    ));
    jScrollPane2.setViewportView(jTable_Datos);

    jLabel1.setText("Realice la consulta:");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jButton_Aceptar)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jScrollPane1)
            .addComponent(jLabel1)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 711,
Short.MAX_VALUE)))
        .addContainerGap(74, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(27, 27, 27)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 132, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jButton_Aceptar)
                .addGap(13, 13, 13)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 261, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(168, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton_AceptarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
            try {
                Class.forName("org.postgresql.Driver");
            } catch (ClassNotFoundException cnfe) {
                JOptionPane.showMessageDialog(this, "driver no disponible", "Mensaje", JOptionPane.INFORMATION_MESSAGE);
            }
            Connection c = null;
            int numero=0;
            ResultSet rs = null;
            Statement s = null;
            BufferedReader entrada=new BufferedReader(new InputStreamReader(System.in));
```

```
try{
  c = DriverManager.getConnection("jdbc:postgresql://localhost:5432/PECL2", "postgres", "oqk&2435");
} catch (SQLException se) {
  JOptionPane.showMessageDialog(this, "No se pudo realizar la conexión: " + se.toString(), "Mensaje", JOptionPane.INFORMATION_MESSAGE);
}
try {
  s = c.createStatement();
} catch (SQLException se) {
  JOptionPane.showMessageDialog(this, "problema al crear la consulta", "Mensaje", JOptionPane.INFORMATION_MESSAGE);
}
try {
  String variableSQL = this.jTextArea_SQL.getText().trim() + ";";
  rs =s.executeQuery(variableSQL) ;
} catch (SQLException se) {
  JOptionPane.showMessageDialog(this, "Excepción al ejecutar consulta: error de sintaxis en el SQL", "Mensaje", JOptionPane.INFORMATION_MESSAGE);
}
DefaultTableModel modelo = new DefaultTableModel();
this.jTable_Datos.setModel(modelo);
ResultSetMetaData rsmd = rs.getMetaData();
int numberOfColumns = rsmd.getColumnCount();
Object[] etiquetas = new Object[numberOfColumns];
Object[] BD = new Object[numberOfColumns];
try {
  for (int i = 0; i < numberOfColumns; i++)
  {
    // Nuevamente, para ResultSetMetaData la primera columna es la 1.
    etiquetas[i] = rsmd.getColumnLabel(i + 1);
```

```
                }

            } catch (SQLException se) {

                JOptionPane.showMessageDialog(this, "Error grave al recoger los resultados3",
"Mensaje", JOptionPane.INFORMATION_MESSAGE);

            }

            modelo.setColumnIdentifiers(etiquetas);

            int aux = 0;

            try {

                while (rs.next()) {

                    for (int i = 0; i < numberOfColumns; i++)

                    {

                        //                    Nuevamente, para ResultSetMetaData la primera columna es la 1.

                        BD[i] = rs.getString(i + 1);

                    }

                    modelo.addRow(BD);

                    aux++;

                }

            } catch (SQLException se) {

                JOptionPane.showMessageDialog(this, "Error grave al recoger los resultados",
"Mensaje", JOptionPane.INFORMATION_MESSAGE);

            }

            rs.close();

            s.close();

            c.close();

        } catch (SQLException ex) {

            JOptionPane.showMessageDialog(this, "Error grave1", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

        }catch(Exception e){

        }

    }

    /**

     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
        new CAL3().setVisible(true);
      }
    });
  }
  // Variables declaration - do not modify
  private javax.swing.JButton jButton_Aceptar;
  private javax.swing.JLabel jLabel1;
  private javax.swing.JScrollPane jScrollPane1;
  private javax.swing.JScrollPane jScrollPane2;
  private javax.swing.JTable jTable_Datos;
  private javax.swing.JTextArea jTextArea_SQL;
  // End of variables declaration
}
```

In this we differ by color code: First yellow see the package name; On the other hand gray we appreciate the comments of the program itself and comments already created directly by the program; Red observe all the code that we have inserted for connecting the database; Finally in blue, are the interface buttons and deprived of them created by Neatbeans variables.

When it comes to represent the graphical interface of the program, as shown below, we perform a simple graphical interface and will be easier to interact with it more easily. The appearance is as follows:



The first blank is reserved to write the query to run, where we introduce each of the consultations PECL2, and pressing the OK button will appear in the space below. Moreover, we can see how the code is divided into four distinct segments.

1- loaded the driver to interact with the database.
2- We establish connection to the database (name of database, user name and password)
3- Consultation, if correct will be shown on screen

Then we will make some consultations called for in the PECL2 to verify that makes what is required for the program
*** We will have to make changes to adapt to the new model queries ***

VIEWS

In this part of the practice we conducted views, where we can see the tables in the database by simply clicking on each of the views

/*CONSULTA VIEW1:*/

create view consultaview1 as (select sum("amount_with_VAT")

from "shopping")

/*CONSULTA VIEW2:*/

SELECT COUNT(*), "categorie"

FROM "product"

```
GROUP BY "product"."categorie";

/*CONSULTA VIEW3:*/
select *
from public. "product"
where "categorie" = 'Phone';

/*CONSULTA VIEW4:*/
SELECT "name"
FROM "product"
WHERE "name" not in (
SELECT "name"
FROM "product","is_offered"
WHERE "product"."code_product" = "is_offered"."id");

/*CONSULTA VIEW5:*/
SELECT "customer"."name_customer", "shopping"."nif", SUM("amount_with_VAT")
FROM "shopping","customer"
WHERE "shopping"."nif"="customer"."nif"
GROUP BY "customer"."name_customer","shopping"."nif";

/*CONSULTA VIEW6:*/
SELECT "Name", "product"."code_product"
FROM "product", "is_offered"
WHERE "product"."code_product"="is_offered"."code_product";

/*CONSULTA VIEW7:*/
SELECT "name", AVG("score")
FROM "product", "reviews"
WHERE "product"."code_product"="reviews"."code_product"
```

```
GROUP BY "name";

/*CONSULTA VIEW8:*/
SELECT COUNT(*) AS time, "product"."name"
FROM "product","shopping"
WHERE "product"."code_product"="shopping"."code_product"
GROUP BY "name"
ORDER BY time desc
limit 1;

/*CONSULTA VIEW9:*/
select SUM("shopping"."amount_with_VAT" - "shopping"."amount_without_VAT")
from "shopping","offer"
where "offer"."id_offer"= "shopping"."id_offer";

/*CONSULTA VIEW10:*/
SELECT COUNT(*), "name"
FROM "product", "refund"
WHERE "product"."code_product"="refund"."codeP"
GROUP BY "name";

/*CONSULTA VIEW11:*/
select sum("amount_with_VAT")
from "customer_online","customer", "shopping"
where "customer"."nif"="customer_online"."nif" and "customer"."nif"="customer"."nif";

select sum("amount_with_VAT")
from "customer_physical","customer", "shopping"
where "customer"."nif"="customer_physical"."nif" and "customer"."nif"="shopping"."nif";
```

```
/*CONSULTA VIEW12:*/

select "customer"."name_customer","shopping"."amount_with_VAT"

from "customer","shopping","bill"

where "customer"."nif" = "shopping"."nif" and "bill"."shopping_number" = "shopping"."shopping_number";
```

```
/*CONSULTA VIEW13:*/

select  avg("date_return"), "categorie"

from (select ("refund"."date_return"-"shopping"."date") from "refund","shopping" )as "date_return","product";
```

```
/*CONSULTA VIEW14:*/

SELECT "services"."name", count(*) as time

FROM "services","shopping"

where "services"."code"="shopping"."code"

GROUP BY "services"."name"

ORDER BY time desc;
```

```
/*CONSULTA VIEW15:*/

SELECT sum("services"."additional_cost_with_VAT")

from "services","shopping"

where "services"."code"="shopping"."code";
```

```
/*CONSULTA VIEW16:*/

select count(*)

from "refund"

where "imperfection" = true ;
```

```
/*CONSULTA 17:*/
```

```
/*CONSULTA VIEW18:*/
```

```
SELECT "code_product", "product"."name"

FROM "product", "offer"

WHERE "product"."code_product"<>"offer"."id_offer";

--WHERE "Code_product" NOT IN (SELECT "Code_product" FROM "have");


/*CONSULTA VIEW19:*/

SELECT "date","amount_with_VAT", "offer"."name"

FROM "shopping", "offer"

WHERE "shopping"."date" BETWEEN "offer"."start_date" AND "offer"."end_date";


/*CONSULTA VIEW20:*/

select "categorie",count(*) as time

from "product"

group by "categorie"

order by "time" desc limit 10;
```

CREATING USERS

In a database it is important to consider the different profiles or roles of users who will access it, and clearly identify how can interact with the database, or what is the same, specify what will be permissions
Management considered three types of profiles:

1- Administrators: have all kinds of permissions to the database
2- Managers: You will manage the database (insert, update, delete and query data), the downside of this is that users cannot create new tables and elements that affect the structure of the database.
3- Commercial: Only may consult the product information

The code for creating these users is:

```
/* Rol de los Administradores */

CREATE ROLE "administrator" SUPERUSER INHERIT CREATEDB CREATEROLE;

GRANT ALL ON associated,bill,customer,customer_online,customer_physical,have,offer,packs,product,refund,reviews,services,shopping,is_offered TO "administrator";

CREATE USER "admin" WITH PASSWORD 'admin' CREATEROLE CREATEUSER INHERIT;
```

```
GRANT "administrator" TO "admin";

/* Rol de los Gestores*/

CREATE ROLE "manager" NOSUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE;

GRANT SELECT, INSERT, UPDATE, DELETE ON
associated,bill,customer,customer_online,customer_physical,have,offer,packs,product,refund,
reviews,services,shopping,is_offered TO "manager";

CREATE USER "manger" WITH PASSWORD 'manger';

GRANT "manager" TO "manger";


/* Rol de Comerciales*/

CREATE ROLE "cashier" NOSUPERUSER NOINHERIT NOCREATEDB NOCREATEROLE;

GRANT SELECT ON
associated,bill,customer,customer_online,customer_physical,have,offer,packs,product,refund,
reviews,services,shopping,is_offered TO "cashier";

CREATE USER "cash" WITH PASSWORD 'cash';

GRANT "cashier" TO "cash";

set role "administrator"

REVOKE CREATE ON SCHEMA public FROM "manager"

REVOKE CREATE ON TABLESPACE pg_default FROM "manager"

REVOKE CREATE ON SCHEMA public FROM "cashier"

REVOKE CREATE ON TABLESPACE pg_default FROM "cashier"
```

The operations that are requested for each user are shown:

```
SET ROLE "administrator"

INSERT INTO product VALUES (181, 'photography', 'Camara reflex' ,'Canon EOS 1200D',349,
'Canon',8);

UPDATE product SET Stock = 7 WHERE product.Stock = 8;

SELECT * FROM product WHERE Stock = 5;
```

```
DELETE FROM product WHERE Stock = 7;

ALTER TABLE product ADD COLUMN Example text

ALTER TABLE product DROP COLUMN Example

CREATE TABLE Example (example1 text)

DROP TABLE Example

select * from product;


SET ROLE "manager"

INSERT INTO product VALUES (150, 'kitchen', 'Coffee capsules' ,'coffee nespressoX470',519, 'Nespresso',13);

UPDATE product SET Stock = 11 WHERE product.Stock = 13;

SELECT * FROM product WHERE Stock = 1;

DELETE FROM product WHERE Stock = 11;

ALTER TABLE product ADD COLUMN Example text

ALTER TABLE product DROP COLUMN Example

CREATE TABLE Example (example1 text)

DROP TABLE Example

select * from product;


SET ROLE "cashier"

INSERT INTO product VALUES (281, 'photography', 'Camara reflex' ,'Canon EOS 1200D',249, 'Canon',58);

UPDATE product SET Stock = 10 WHERE product.Stock = 58;

DELETE FROM product WHERE Stock = 10;

/*Aqui vemos como solo puede consultar*/

Select * from product;
```

CONNECTION TO THE DATABASE FROM EXTERNAL PROGRAMS AND SECURITY

In this last part we modify the JAVA program that we previously created to select one of the three users created in the database. Modify it depending on the privileges held by each user.

We have created two classes to login and then show you the code:

Login:

```
package CAL3;

import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

  public Login() {

    initComponents();

  }


  /**

   * This method is called from within the constructor to initialize the form.

   * WARNING: Do NOT modify this code. The content of this method is always

   * regenerated by the Form Editor.

   */

  @SuppressWarnings("unchecked")

  // <editor-fold defaultstate="collapsed" desc="Generated Code">

  private void initComponents() {


    jPasswordField = new javax.swing.JPasswordField();

    jButtonInicioSesion = new javax.swing.JButton();

    jLabel2 = new javax.swing.JLabel();

    jLabel1 = new javax.swing.JLabel();

    jComboBox_Usuario = new javax.swing.JComboBox();


    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
        jButtonInicioSesion.setText("Iniciar Sesion");
        jButtonInicioSesion.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonInicioSesionActionPerformed(evt);
            }
        });

        jLabel2.setText("Password");

        jLabel1.setText("Escoja el usuario");

        jComboBox_Usuario.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "admin", "manger", "cash" }));
        jComboBox_Usuario.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jComboBox_UsuarioActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(57, 57, 57)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jLabel2)
                    .addComponent(jLabel1))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(jPasswordField, javax.swing.GroupLayout.Alignment.LEADING)
```

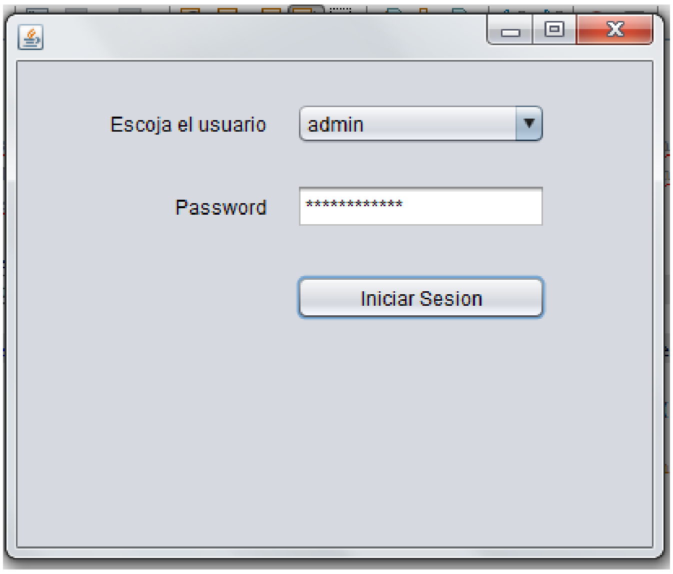
```
          .addComponent(jButtonInicioSesion, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
          .addComponent(jComboBox_Usuario, javax.swing.GroupLayout.Alignment.LEADING, 0, 154,
Short.MAX_VALUE))
        .addContainerGap(65, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(25, 25, 25)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel1)
          .addComponent(jComboBox_Usuario, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(24, 24, 24)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel2)
          .addComponent(jPasswordField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(28, 28, 28)
        .addComponent(jButtonInicioSesion)
        .addContainerGap(139, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void jButtonInicioSesionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (this.jComboBox_Usuario.getSelectedItem().toString().compareTo("admin") == 0) {

      if (this.jPasswordField.getText().compareTo("admin") == 0) {
        new CAL3("admin").setVisible(true);
      } else {

        JOptionPane.showMessageDialog(this, "Password incorrecta", "Error de inicio de
sesión", JOptionPane.ERROR_MESSAGE);

      }
    }
    if (this.jComboBox_Usuario.getSelectedItem().toString().compareTo("manager") == 0) {

      if (this.jPasswordField.getText().compareTo("manger") == 0) {
        new CAL3("manger").setVisible(true);
      } else {

        JOptionPane.showMessageDialog(this, "Password incorrecta", "Error de inicio de
sesión", JOptionPane.ERROR_MESSAGE);

      }
    }
    if (this.jComboBox_Usuario.getSelectedItem().toString().compareTo("cash") == 0) {

      if (this.jPasswordField.getText().compareTo("cash") == 0) {
        new CAL3("cash").setVisible(true);
      } else {

        JOptionPane.showMessageDialog(this, "Password incorrecta", "Error de inicio de
sesión", JOptionPane.ERROR_MESSAGE);

      }
    }
  }

  private void jComboBox_UsuarioActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
        new Login().setVisible(true);
      }
    });
  }
  // Variables declaration - do not modify
  private javax.swing.JButton jButtonInicioSesion;
  private javax.swing.JComboBox jComboBox_Usuario;
  private javax.swing.JLabel jLabel1;
  private javax.swing.JLabel jLabel2;
  private javax.swing.JPasswordField jPasswordField;
  // End of variables declaration
}
```

As we have previously done in this we differ by color code: First yellow see the package name; On the other hand gray we appreciate the comments of the program itself and comments already created directly by the program; Red observe all the code that we have inserted for connecting the database; Finally in blue, are the interface buttons and deprived of them created by Neatbeans variables.

Pec3:

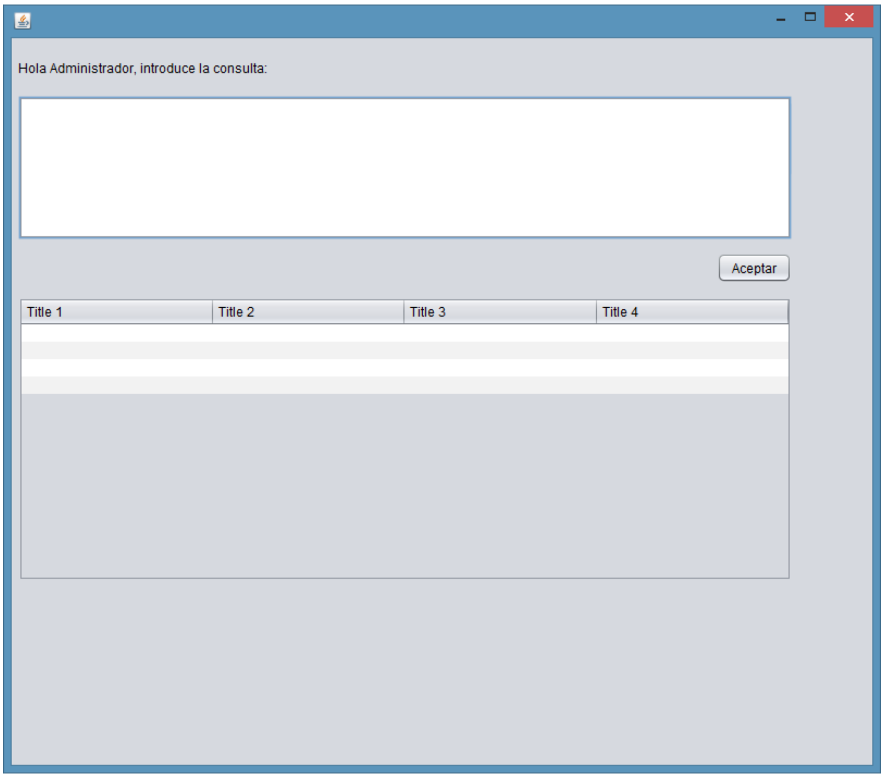
And then we'll see how it turned out we interface this class:
First we see the sign, which can enter one of the three users



Escoja el usuario: admin

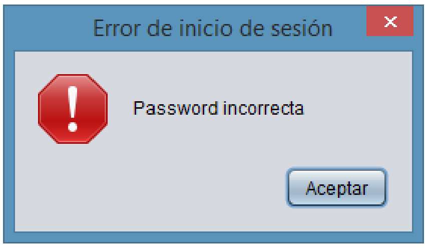
Password: *************

In the next picture we can enter the query to perform and changes the welcome message as the user has entered



Hola Administrador, introduce la consulta:

If you enter the wrong password we will jump a message like the following:



Error de inicio de sesión

Password incorrecta

Finally we see that the query works correctly. We can check with the following code:

```
package CAL3;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class CAL3 extends javax.swing.JFrame {

  public static String users;

  public CAL3(String users) {

    initComponents();

    this.users = users;

    jLabelSaludo.setText("Hola "+users+", introduce la consulta: ");

  }


  /**

   * This method is called from within the constructor to initialize the form.

   * WARNING: Do NOT modify this code. The content of this method is always

   * regenerated by the Form Editor.

   */
```

```
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

  jButton_Aceptar = new javax.swing.JButton();
  jScrollPane1 = new javax.swing.JScrollPane();
  jTextArea_SQL = new javax.swing.JTextArea();
  jScrollPane2 = new javax.swing.JScrollPane();
  jTable_Datos = new javax.swing.JTable();
  jLabelSaludo = new javax.swing.JLabel();

  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

  jButton_Aceptar.setText("Aceptar");
  jButton_Aceptar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
      jButton_AceptarActionPerformed(evt);
    }
  });

  jTextArea_SQL.setColumns(20);
  jTextArea_SQL.setRows(5);
  jScrollPane1.setViewportView(jTextArea_SQL);

  jTable_Datos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
      {null, null, null, null},
      {null, null, null, null},
      {null, null, null, null},
      {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
      }
    ));
    jScrollPane2.setViewportView(jTable_Datos);

    jLabelSaludo.setText("jLabel1");

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jButton_Aceptar)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
              .addComponent(jScrollPane1)
              .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 711,
Short.MAX_VALUE)))
          .addComponent(jLabelSaludo))
        .addContainerGap(74, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
      layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabelSaludo)
        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton_Aceptar)
        .addGap(13, 13, 13)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 261,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(169, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>
  private void jButton_AceptarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{
      try {
        Class.forName("org.postgresql.Driver");
      } catch (ClassNotFoundException cnfe) {
        JOptionPane.showMessageDialog(this, "driver no disponible", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
      }

      Connection c = null;

      int numero=0;
      ResultSet rs = null;
      Statement s = null;

      BufferedReader entrada=new BufferedReader(new InputStreamReader(System.in));

      try{

        if(users.compareTo("admin") == 0){
          c = DriverManager.getConnection("jdbc:postgresql://localhost:5432/PECL2",
"admin", "admin");
        }
```

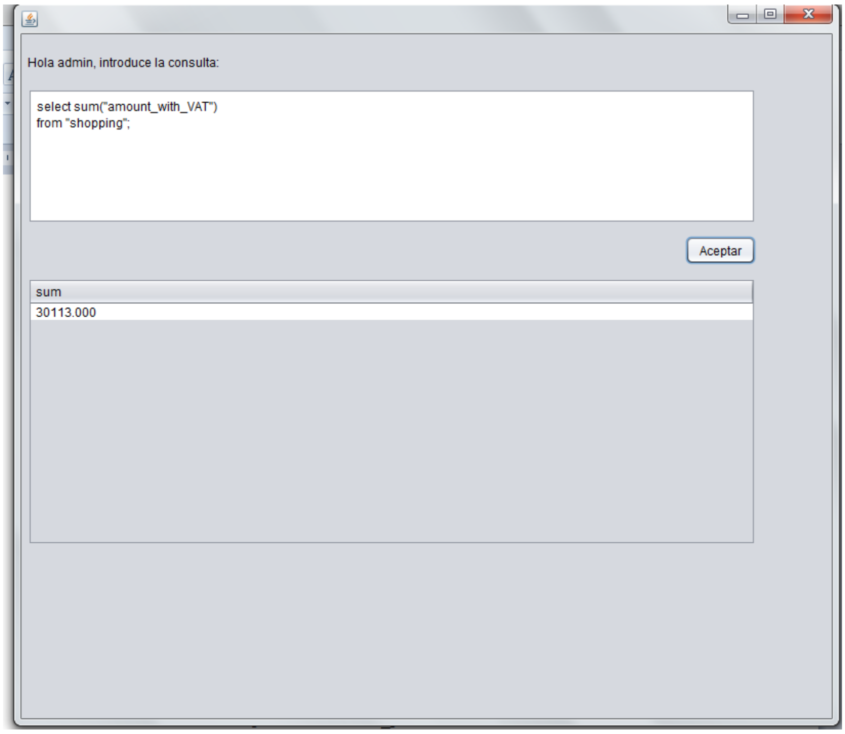
```
            if(users.compareTo("manger") == 0){
               c = DriverManager.getConnection("jdbc:postgresql://localhost:5432/PECL2","manger", "manger");
            }
            if(users.compareTo("cash") == 0){
               c = DriverManager.getConnection("jdbc:postgresql://localhost:5432/PECL2","cash", "cash");
            }


         } catch (SQLException se) {
            JOptionPane.showMessageDialog(this, "No se pudo realizar la conexión: " + se.toString(), "Mensaje", JOptionPane.INFORMATION_MESSAGE);


         }


         try {
            s = c.createStatement();
         } catch (SQLException se) {
            JOptionPane.showMessageDialog(this, "problema al crear la consulta", "Mensaje", JOptionPane.INFORMATION_MESSAGE);


         }


         try {


            String variableSQL = this.jTextArea_SQL.getText().trim() + ";";
            rs =s.executeQuery(variableSQL) ;
         } catch (SQLException se) {
            JOptionPane.showMessageDialog(this, "Excepción al ejecutar consulta: error de sintaxis en el SQL", "Mensaje", JOptionPane.INFORMATION_MESSAGE);


         }
```

```
DefaultTableModel modelo = new DefaultTableModel();
this.jTable_Datos.setModel(modelo);

ResultSetMetaData rsmd = rs.getMetaData();
int numberOfColumns = rsmd.getColumnCount();
Object[] etiquetas = new Object[numberOfColumns];
Object[] BD = new Object[numberOfColumns];

try {

  for (int i = 0; i < numberOfColumns; i++)
  {
    // Nuevamente, para ResultSetMetaData la primera columna es la 1.
    etiquetas[i] = rsmd.getColumnLabel(i + 1);
  }

} catch (SQLException se) {
  JOptionPane.showMessageDialog(this, "Error grave al recoger los resultados3", "Mensaje", JOptionPane.INFORMATION_MESSAGE);

}
modelo.setColumnIdentifiers(etiquetas);
int aux = 0;
try {
  while (rs.next()) {
    for (int i = 0; i < numberOfColumns; i++)
    {
      //                    Nuevamente, para ResultSetMetaData la primera columna es la 1.
      BD[i] = rs.getString(i + 1);
```

```
                }
                modelo.addRow(BD);
                aux++;
            }

        } catch (SQLException se) {
            JOptionPane.showMessageDialog(this, "Error grave al recoger los resultados",
"Mensaje", JOptionPane.INFORMATION_MESSAGE);

        }

        rs.close();
        s.close();
        c.close();

      } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Error grave1", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
      }catch(Exception e){
      }
   }

   /**
    * @param args the command line arguments
    */
   public static void main(String args[]) {
      /* Set the Nimbus look and feel */
      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
       * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */
```

```
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(CAL3.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>


        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CAL3(users).setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton_Aceptar;
    private javax.swing.JLabel jLabelSaludo;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTable jTable_Datos;
    private javax.swing.JTextArea jTextArea_SQL;
    // End of variables declaration
}
```

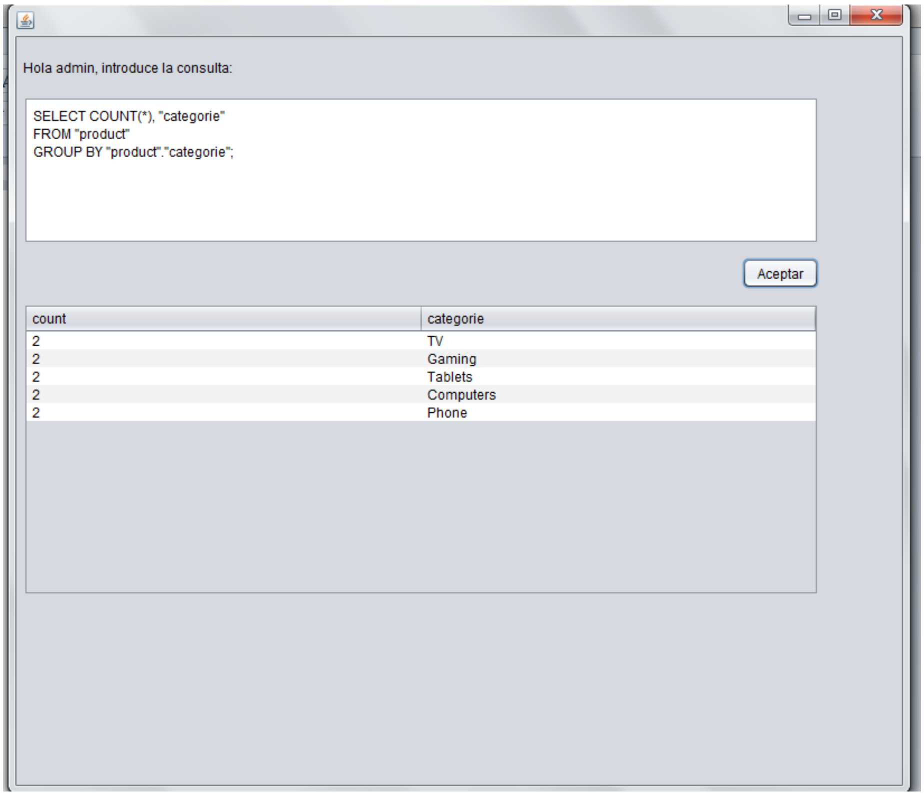
As we have previously done in this we differ by color code: First yellow see the package name; On the other hand gray we appreciate the comments of the program itself and comments already created directly by the program; Red observe all the code that we have inserted for connecting the database; Finally in blue, are the interface buttons and deprived of them created by Neatbeans variables.



Hola admin, introduce la consulta:

```
select sum("amount_with_VAT")
from "shopping";
```

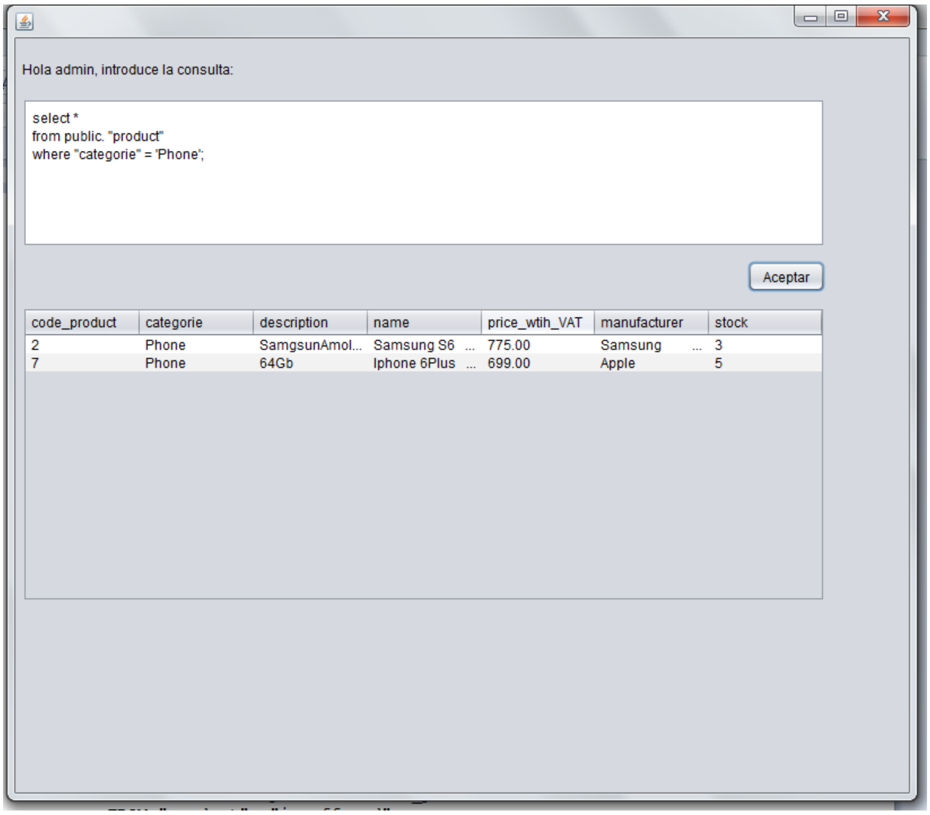
sum
30113.000



Hola admin, introduce la consulta:

```
SELECT COUNT(*), "categorie"
FROM "product"
GROUP BY "product"."categorie";
```

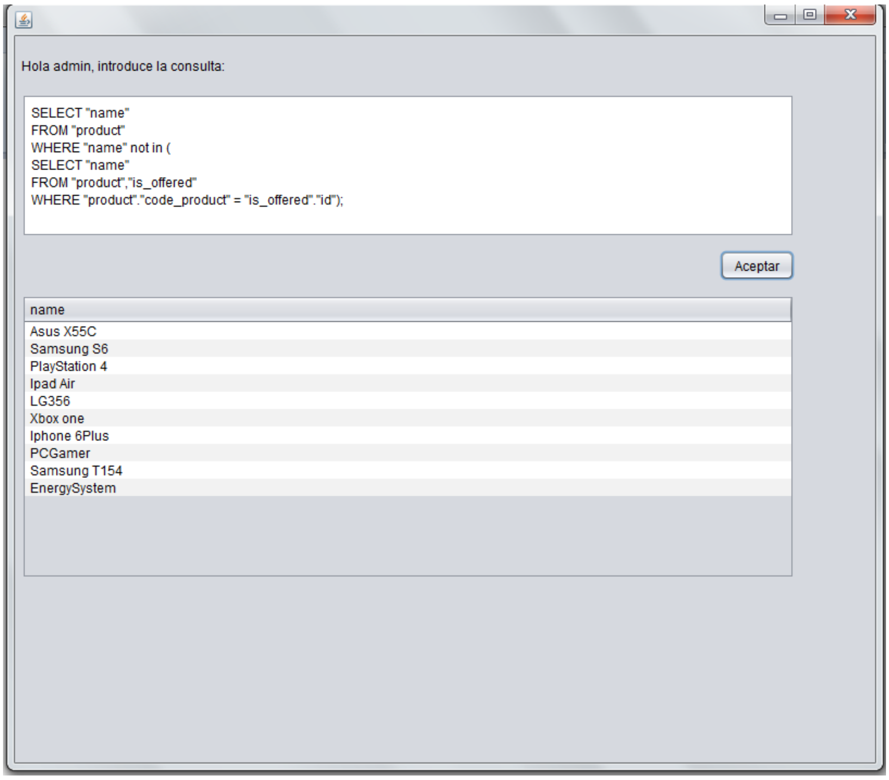
count	categorie
2	TV
2	Gaming
2	Tablets
2	Computers
2	Phone



Hola admin, introduce la consulta:

```
select *
from public. "product"
where "categorie" = 'Phone';
```

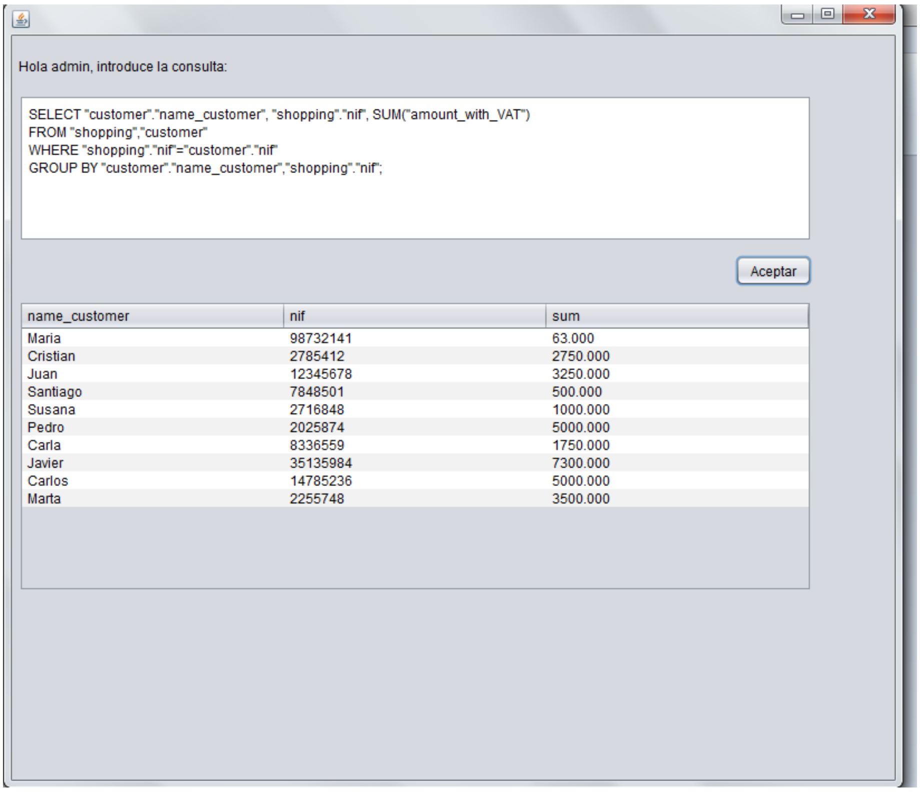
code_product	categorie	description	name	price_wtih_VAT	manufacturer	stock
2	Phone	SamgsunAmol...	Samsung S6 ...	775.00	Samsung ...	3
7	Phone	64Gb	Iphone 6Plus ...	699.00	Apple	5



Hola admin, introduce la consulta:

```
SELECT "name"
FROM "product"
WHERE "name" not in (
SELECT "name"
FROM "product","is_offered"
WHERE "product"."code_product" = "is_offered"."id");
```

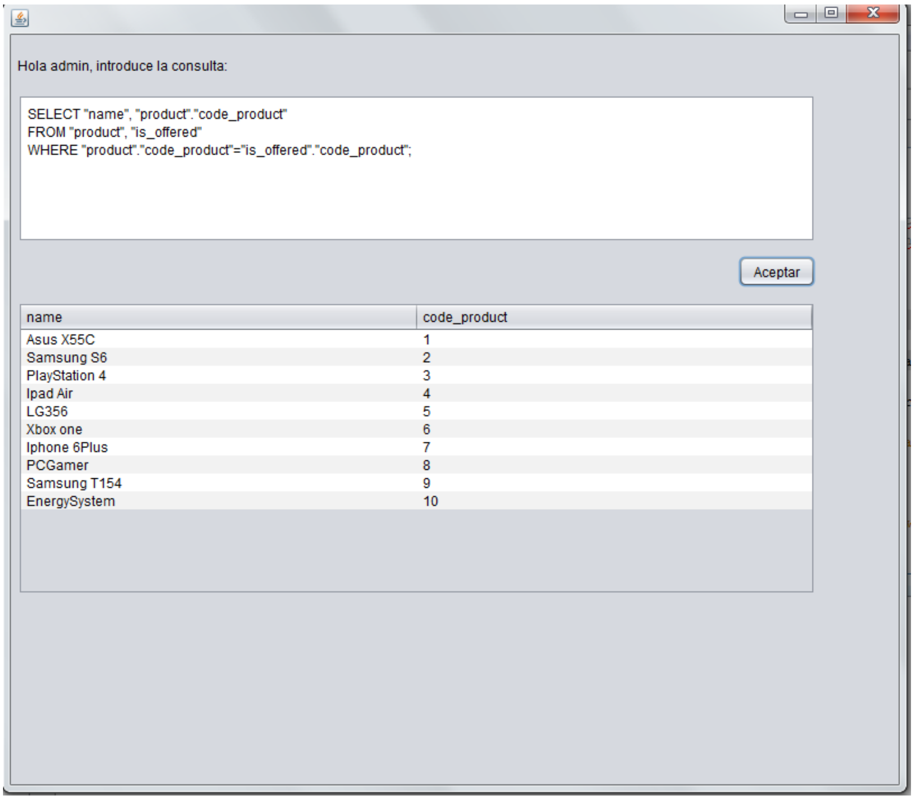
name
Asus X55C
Samsung S6
PlayStation 4
Ipad Air
LG356
Xbox one
Iphone 6Plus
PCGamer
Samsung T154
EnergySystem



Hola admin, introduce la consulta:

```
SELECT "customer"."name_customer", "shopping"."nif", SUM("amount_with_VAT")
FROM "shopping","customer"
WHERE "shopping"."nif"="customer"."nif"
GROUP BY "customer"."name_customer","shopping"."nif";
```

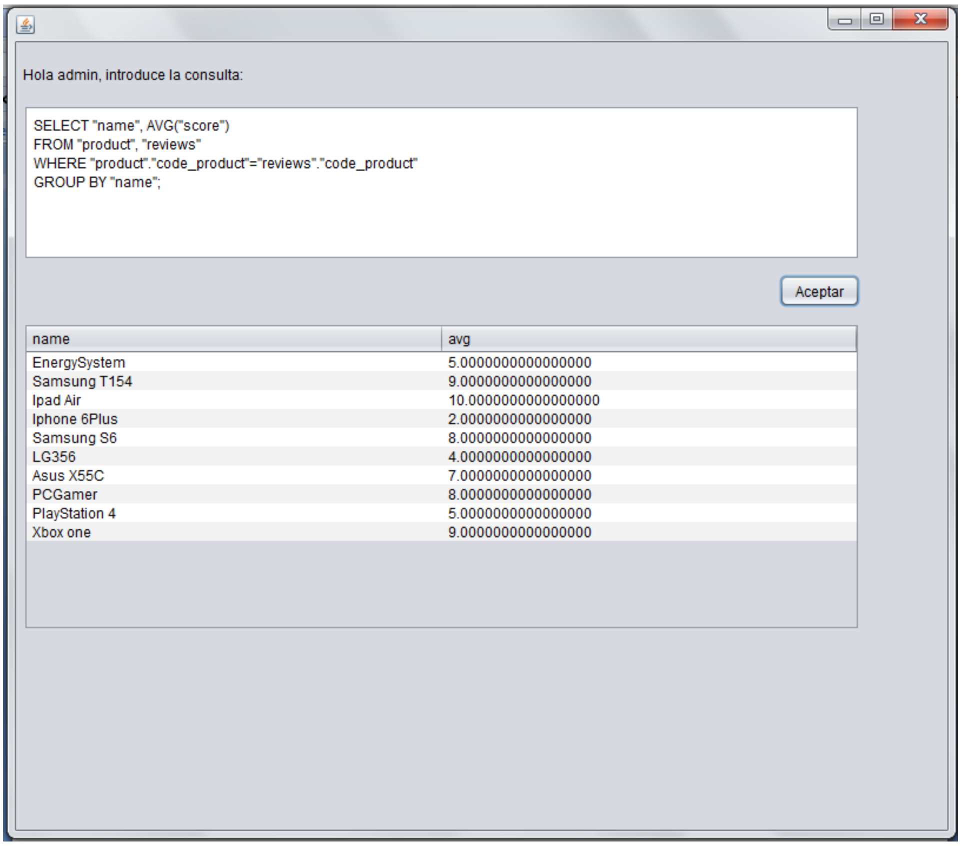
name_customer	nif	sum
Maria	98732141	63.000
Cristian	2785412	2750.000
Juan	12345678	3250.000
Santiago	7848501	500.000
Susana	2716848	1000.000
Pedro	2025874	5000.000
Carla	8336559	1750.000
Javier	35135984	7300.000
Carlos	14785236	5000.000
Marta	2255748	3500.000



Hola admin, introduce la consulta:

```
SELECT "name", "product"."code_product"
FROM "product", "is_offered"
WHERE "product"."code_product"="is_offered"."code_product";
```

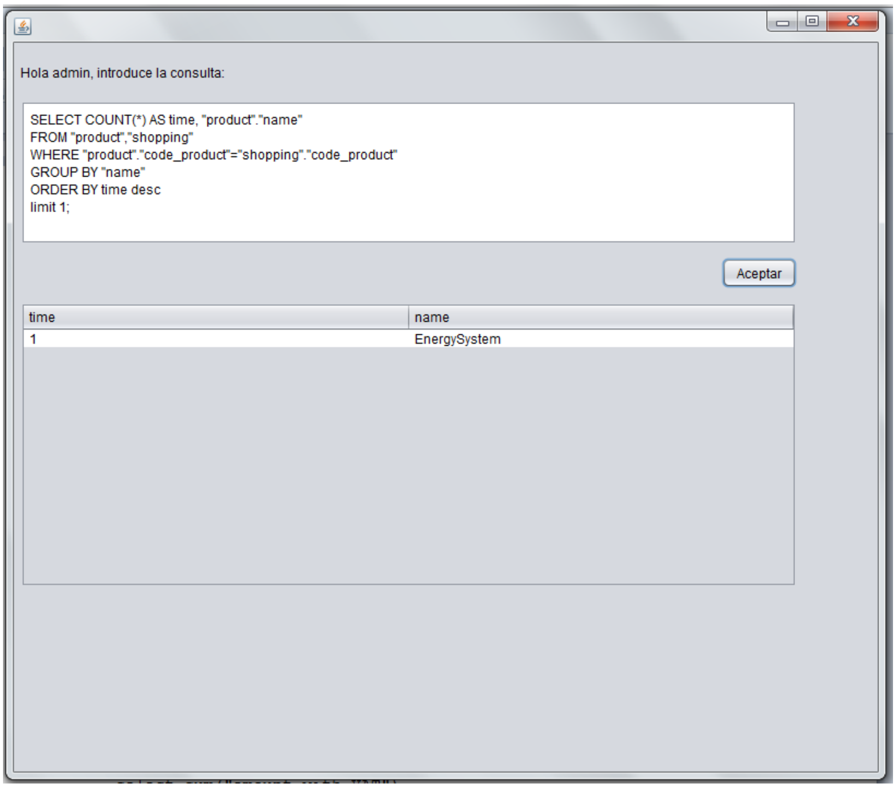
name	code_product
Asus X55C	1
Samsung S6	2
PlayStation 4	3
Ipad Air	4
LG356	5
Xbox one	6
Iphone 6Plus	7
PCGamer	8
Samsung T154	9
EnergySystem	10



Hola admin, introduce la consulta:

```
SELECT "name", AVG("score")
FROM "product", "reviews"
WHERE "product"."code_product"="reviews"."code_product"
GROUP BY "name";
```

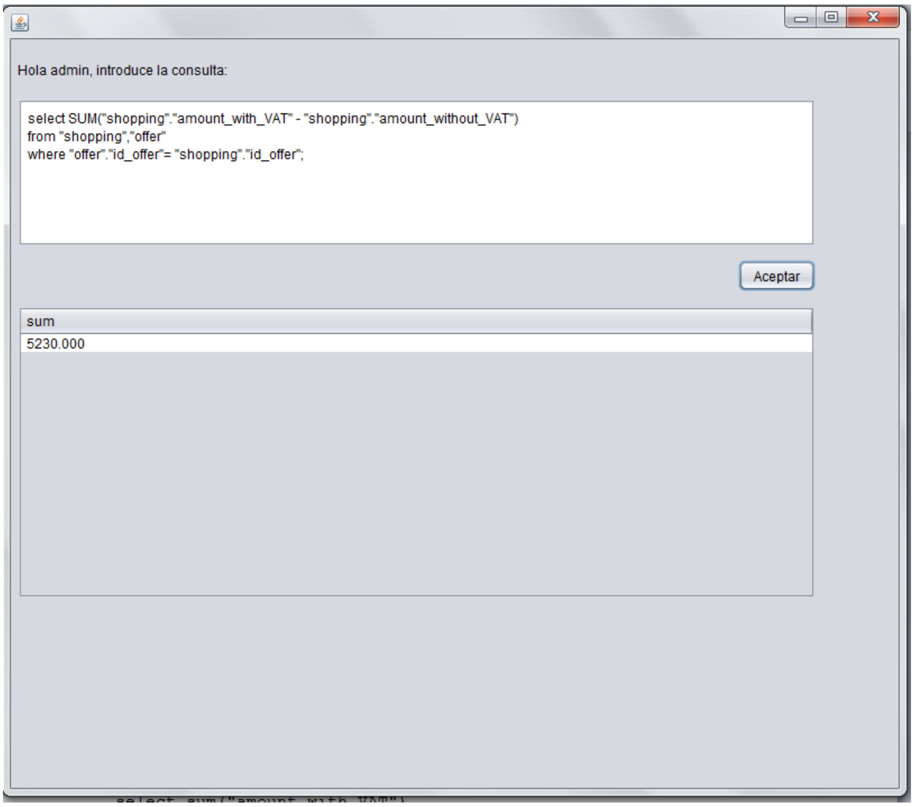
name	avg
EnergySystem	5.0000000000000000
Samsung T154	9.0000000000000000
Ipad Air	10.0000000000000000
Iphone 6Plus	2.0000000000000000
Samsung S6	8.0000000000000000
LG356	4.0000000000000000
Asus X55C	7.0000000000000000
PCGamer	8.0000000000000000
PlayStation 4	5.0000000000000000
Xbox one	9.0000000000000000



Hola admin, introduce la consulta:

```
SELECT COUNT(*) AS time, "product"."name"
FROM "product","shopping"
WHERE "product"."code_product"="shopping"."code_product"
GROUP BY "name"
ORDER BY time desc
limit 1;
```

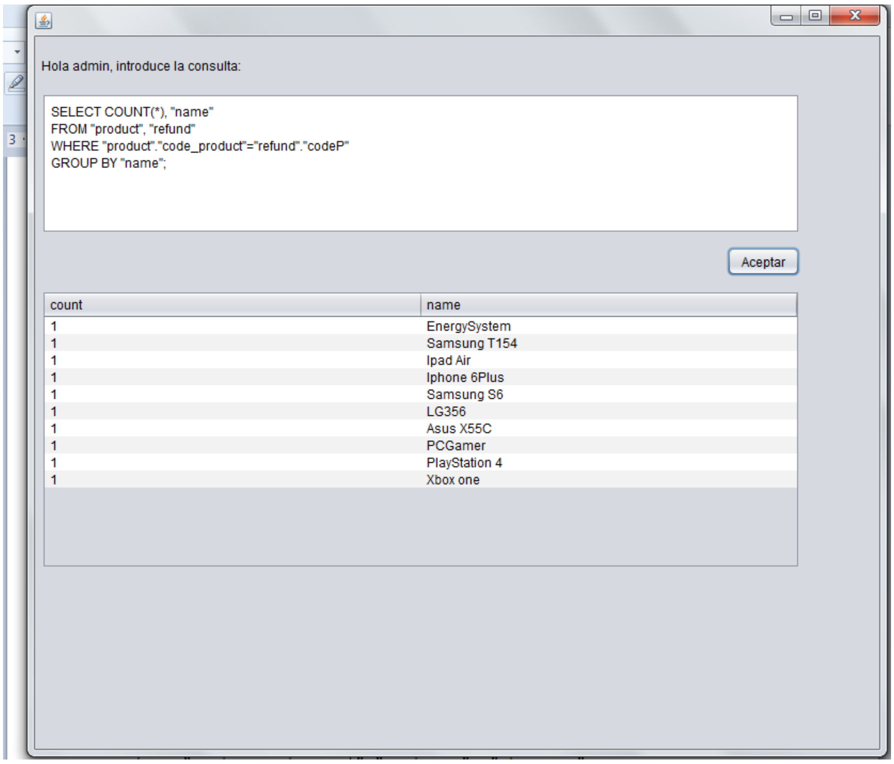
time	name
1	EnergySystem



Hola admin, introduce la consulta:

```
select SUM("shopping"."amount_with_VAT" - "shopping"."amount_without_VAT")
from "shopping","offer"
where "offer"."id_offer"= "shopping"."id_offer";
```

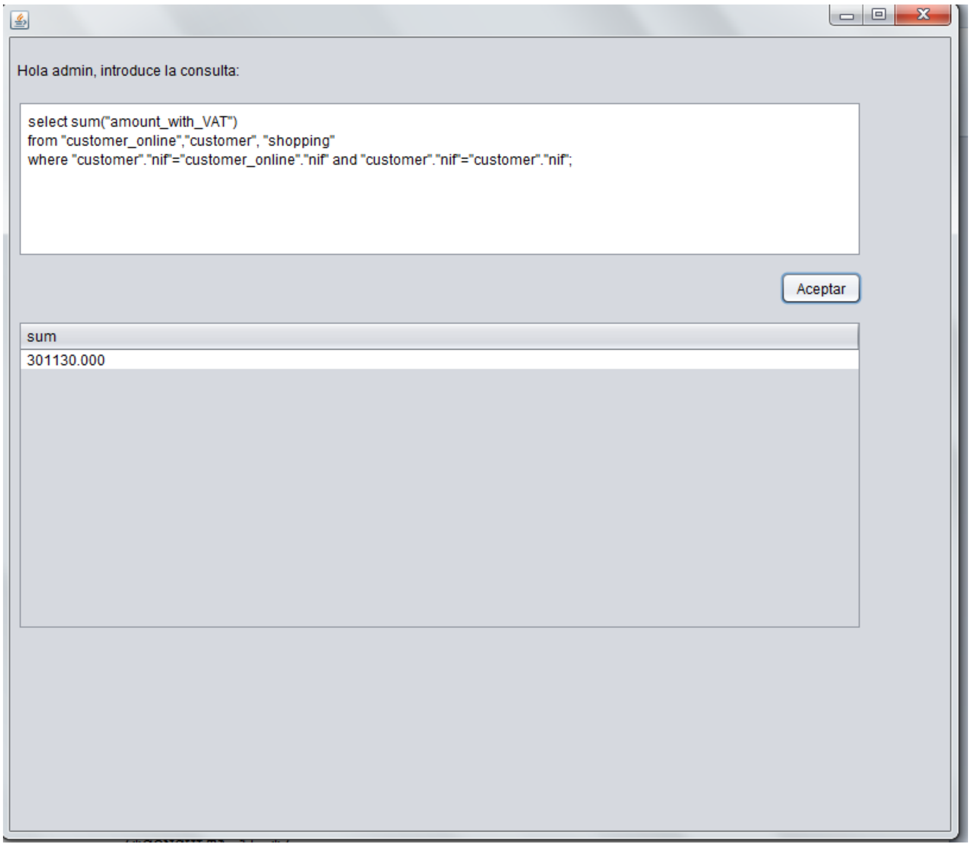
sum
5230.000



Hola admin, introduce la consulta:

```
SELECT COUNT(*), "name"
FROM "product", "refund"
WHERE "product"."code_product"="refund"."codeP"
GROUP BY "name";
```

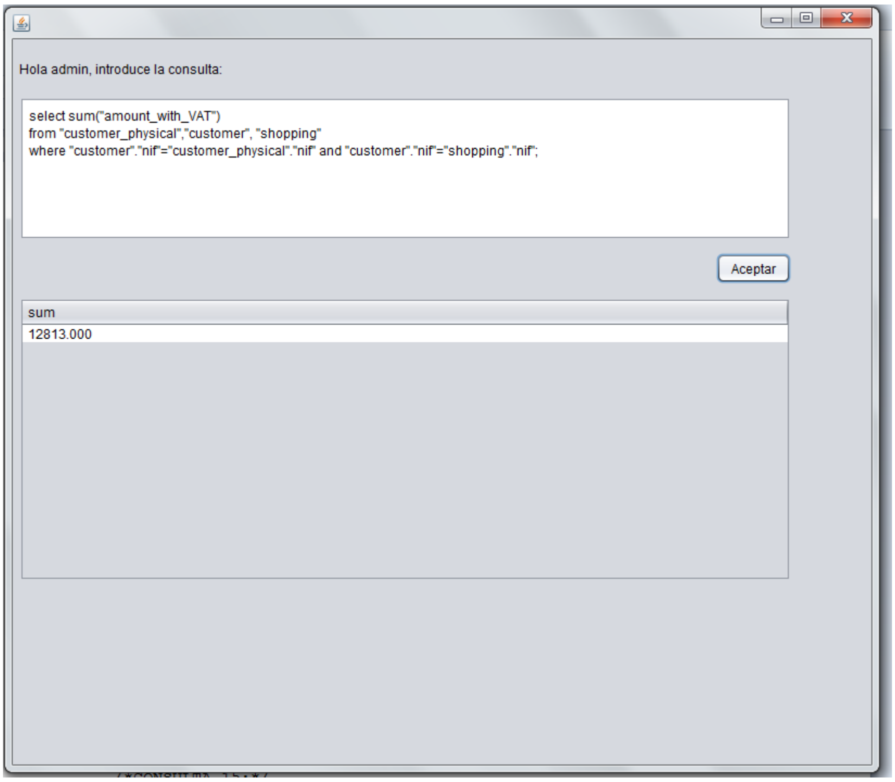
count	name
1	EnergySystem
1	Samsung T154
1	Ipad Air
1	Iphone 6Plus
1	Samsung S6
1	LG356
1	Asus X55C
1	PCGamer
1	PlayStation 4
1	Xbox one



Hola admin, introduce la consulta:

```
select sum("amount_with_VAT")
from "customer_online","customer", "shopping"
where "customer"."nif"="customer_online"."nif" and "customer"."nif"="customer"."nif";
```

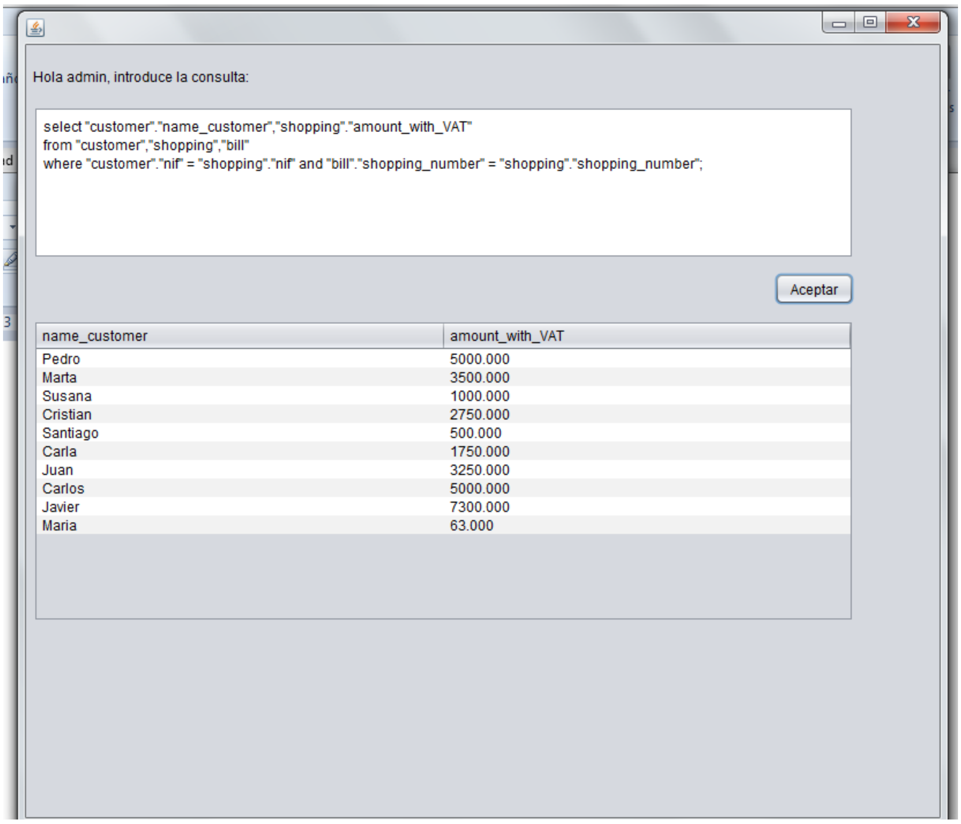
sum
301130.000



Hola admin, introduce la consulta:

```
select sum("amount_with_VAT")
from "customer_physical","customer", "shopping"
where "customer"."nif"="customer_physical"."nif" and "customer"."nif"="shopping"."nif";
```

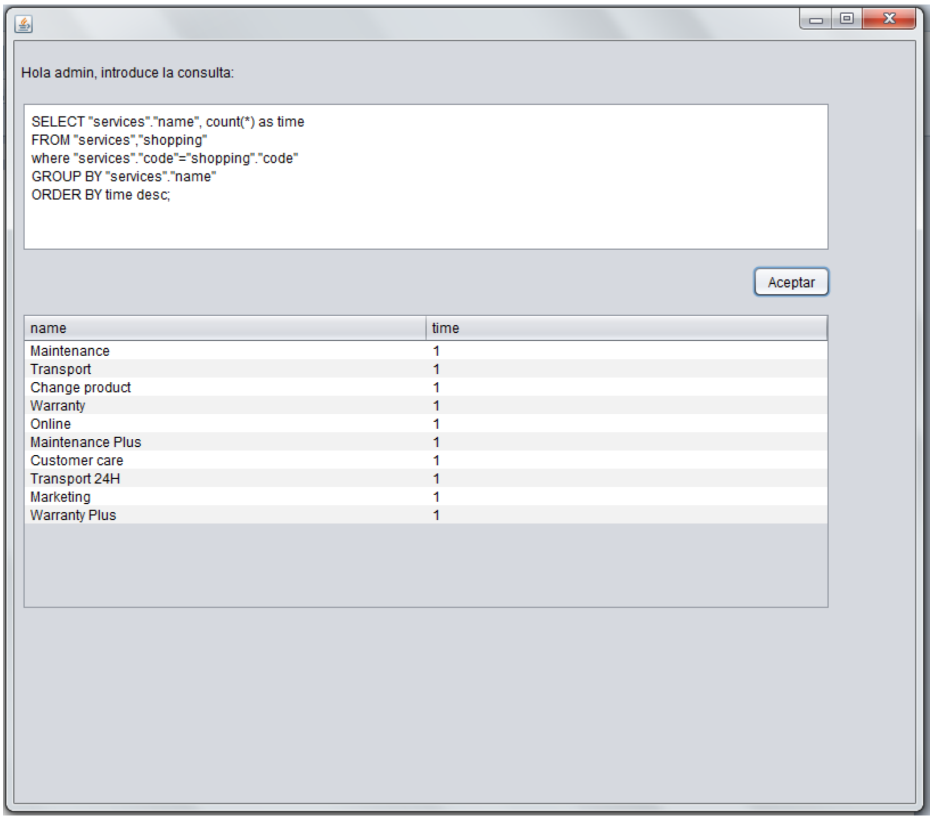
sum
12813.000



Hola admin, introduce la consulta:

```
select "customer"."name_customer","shopping"."amount_with_VAT"
from "customer","shopping","bill"
where "customer"."nif" = "shopping"."nif" and "bill"."shopping_number" = "shopping"."shopping_number";
```

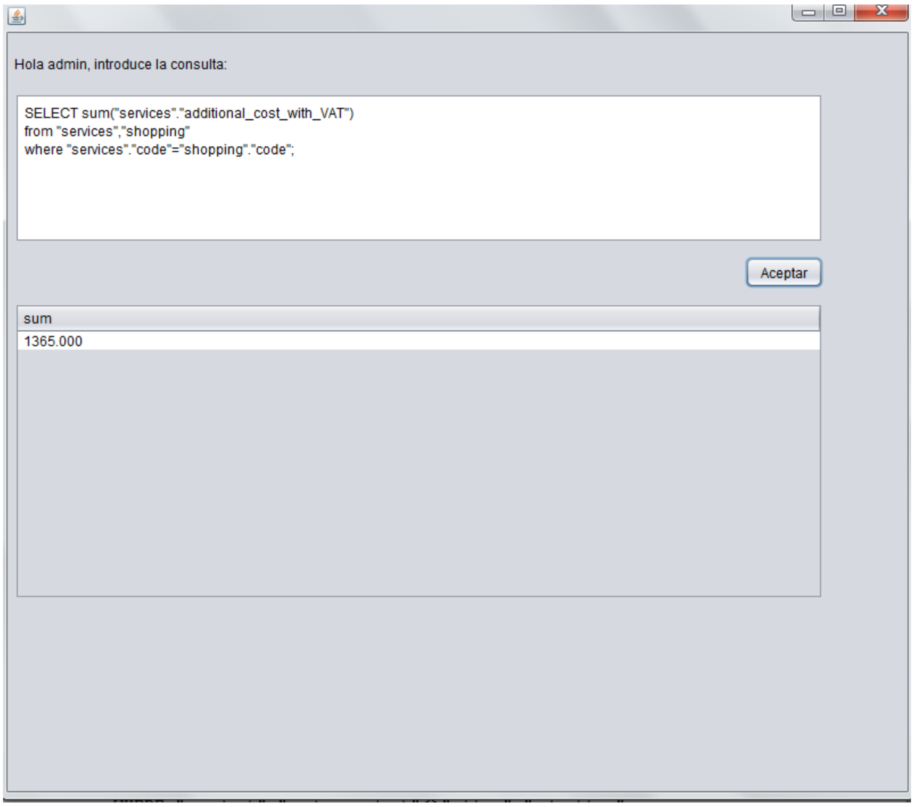
name_customer	amount_with_VAT
Pedro	5000.000
Marta	3500.000
Susana	1000.000
Cristian	2750.000
Santiago	500.000
Carla	1750.000
Juan	3250.000
Carlos	5000.000
Javier	7300.000
Maria	63.000



Hola admin, introduce la consulta:

```
SELECT "services"."name", count(*) as time
FROM "services","shopping"
where "services"."code"="shopping"."code"
GROUP BY "services"."name"
ORDER BY time desc;
```

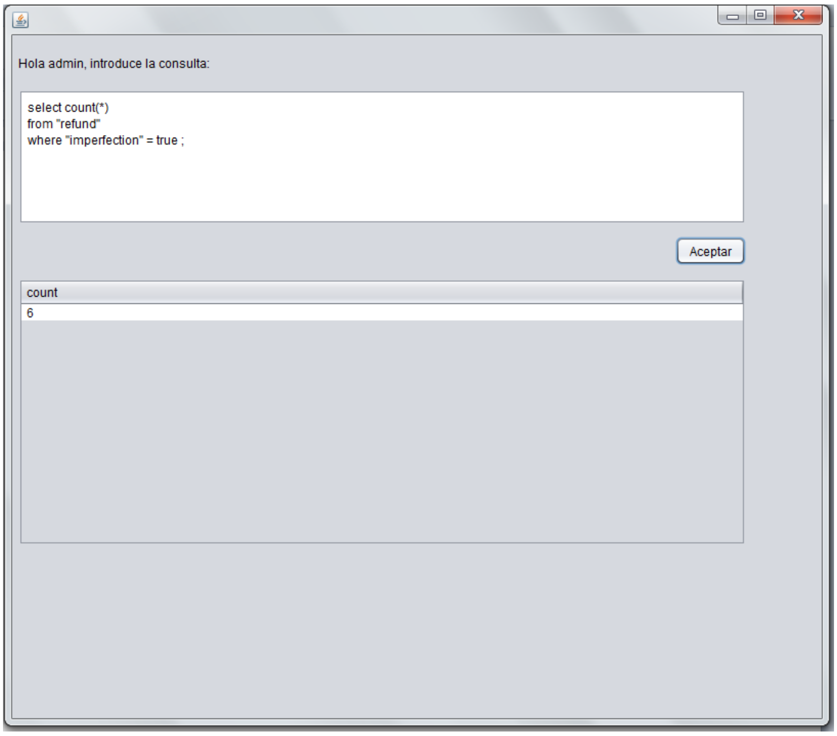
name	time
Maintenance	1
Transport	1
Change product	1
Warranty	1
Online	1
Maintenance Plus	1
Customer care	1
Transport 24H	1
Marketing	1
Warranty Plus	1



Hola admin, introduce la consulta:

```
SELECT sum("services"."additional_cost_with_VAT")
from "services","shopping"
where "services"."code"="shopping"."code";
```

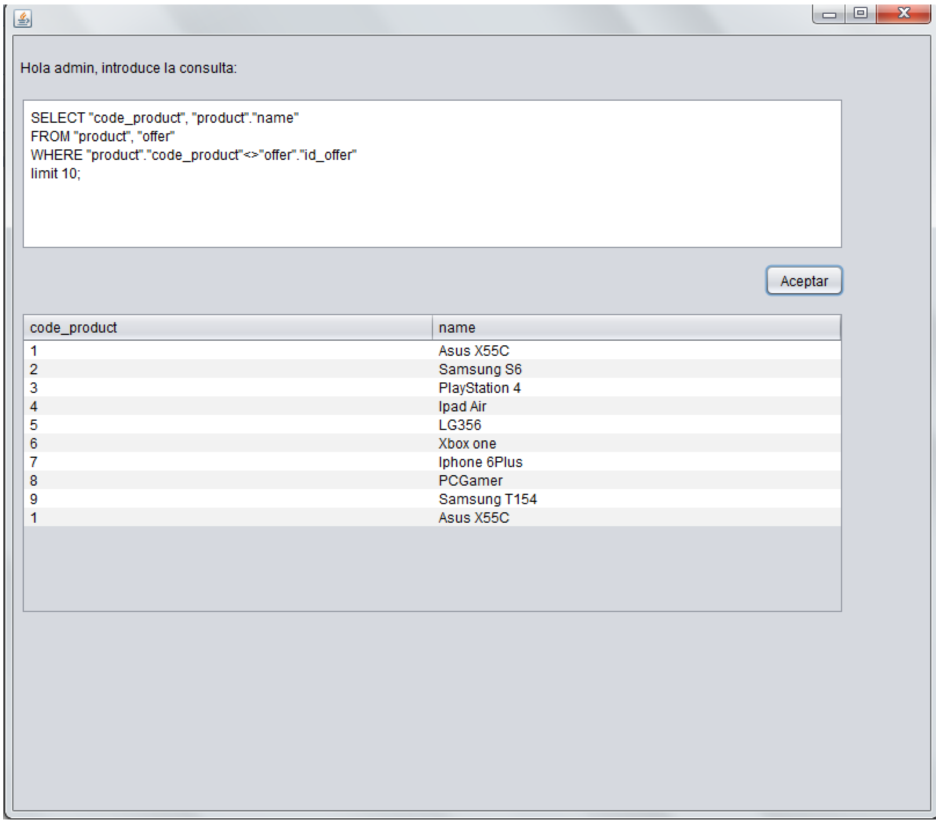
sum
1365.000



Hola admin, introduce la consulta:

```
select count(*)
from "refund"
where "imperfection" = true ;
```

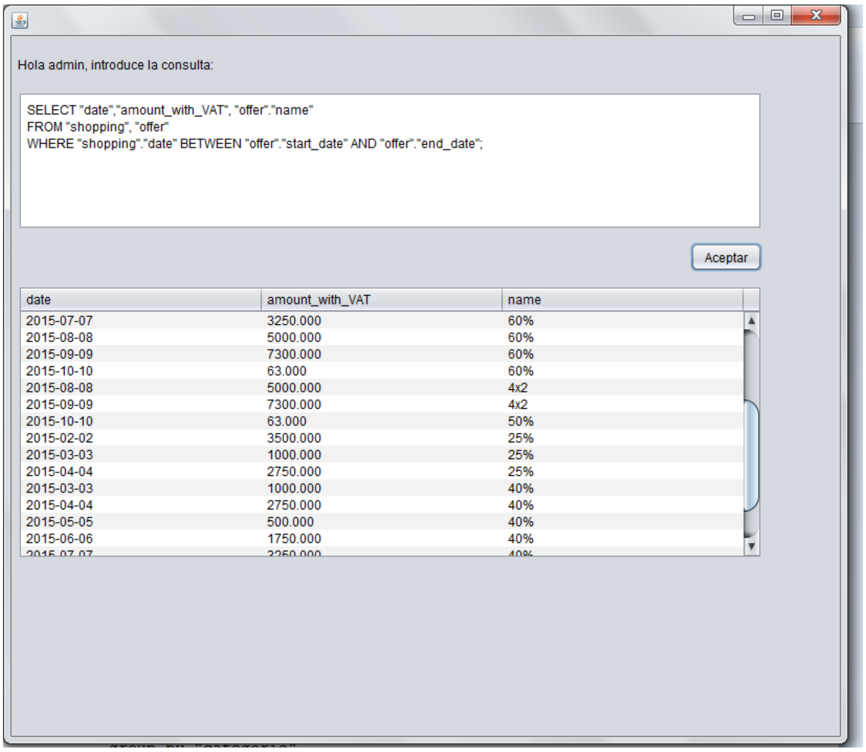
count
6



Hola admin, introduce la consulta:

```
SELECT "code_product", "product"."name"
FROM "product", "offer"
WHERE "product"."code_product"<>"offer"."id_offer"
limit 10;
```

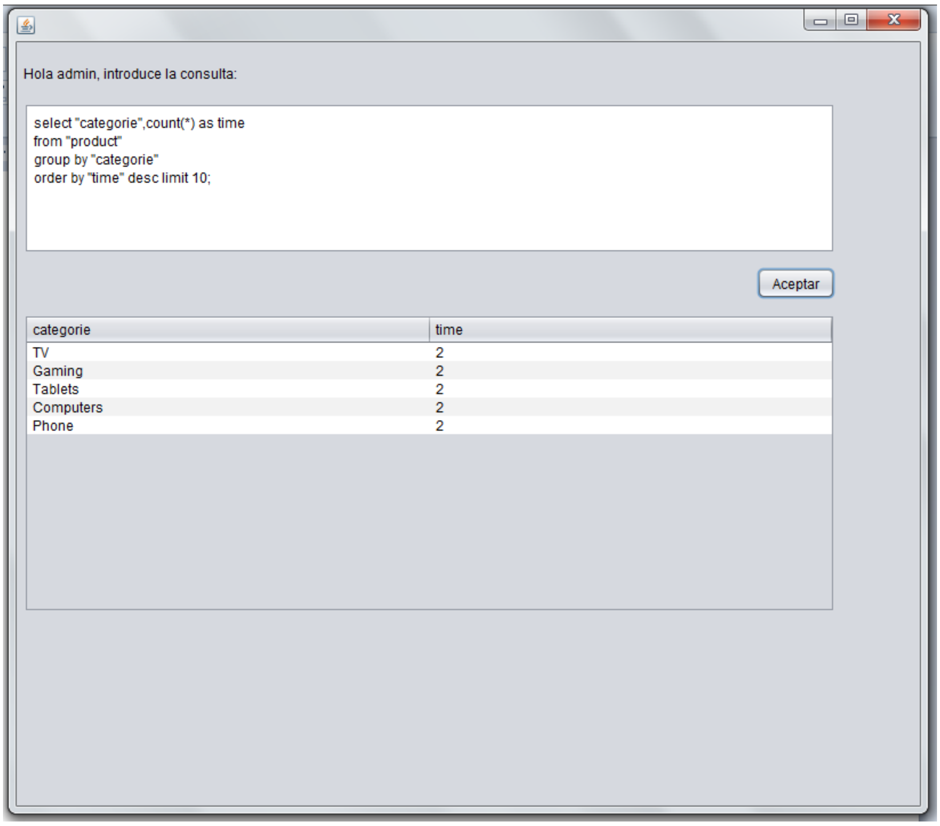
code_product	name
1	Asus X55C
2	Samsung S6
3	PlayStation 4
4	Ipad Air
5	LG356
6	Xbox one
7	Iphone 6Plus
8	PCGamer
9	Samsung T154
1	Asus X55C



Hola admin, introduce la consulta:

```
SELECT "date","amount_with_VAT", "offer"."name"
FROM "shopping", "offer"
WHERE "shopping"."date" BETWEEN "offer"."start_date" AND "offer"."end_date";
```

date	amount_with_VAT	name
2015-07-07	3250.000	60%
2015-08-08	5000.000	60%
2015-09-09	7300.000	60%
2015-10-10	63.000	60%
2015-08-08	5000.000	4x2
2015-09-09	7300.000	4x2
2015-10-10	63.000	50%
2015-02-02	3500.000	25%
2015-03-03	1000.000	25%
2015-04-04	2750.000	25%
2015-03-03	1000.000	40%
2015-04-04	2750.000	40%
2015-05-05	500.000	40%
2015-06-06	1750.000	40%
2015-07-07	3250.000	40%

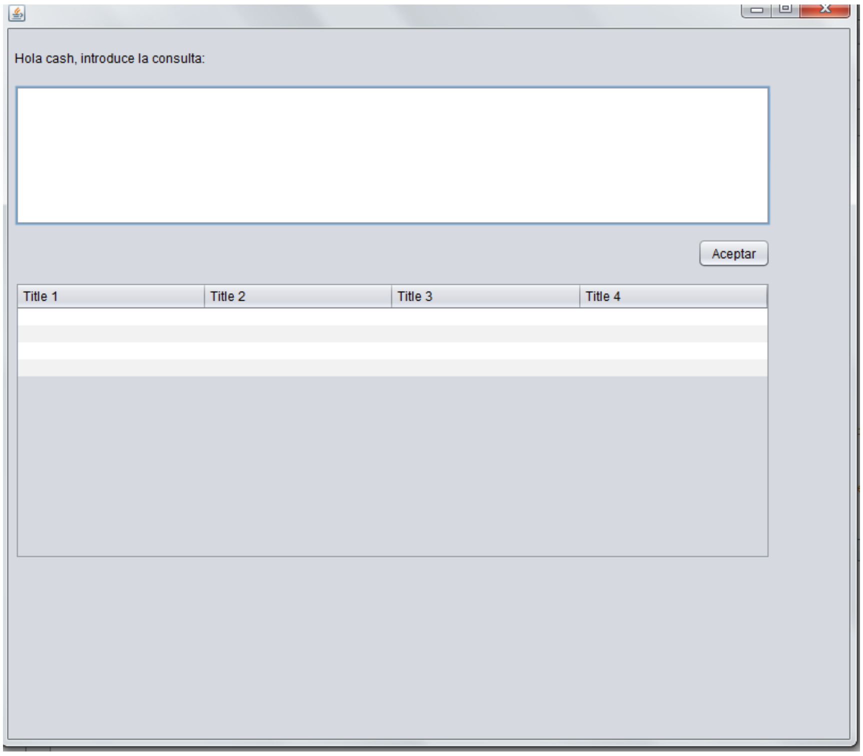


Hola admin, introduce la consulta:

```
select "categorie",count(*) as time
from "product"
group by "categorie"
order by "time" desc limit 10;
```

categorie	time
TV	2
Gaming	2
Tablets	2
Computers	2
Phone	2

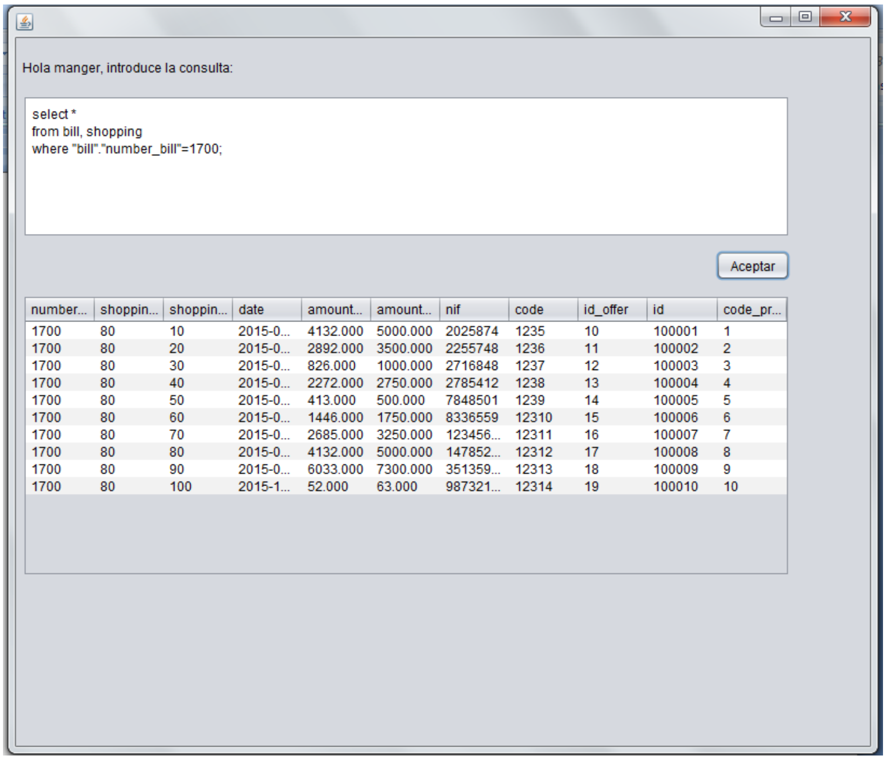
We can go as cash:



Hola cash, introduce la consulta:

Title 1	Title 2	Title 3	Title 4

We can go as manager:



Hola manger, introduce la consulta:

```
select *
from bill, shopping
where "bill"."number_bill"=1700;
```

number...	shoppin...	shoppin...	date	amount...	amount...	nif	code	id_offer	id	code_pr...
1700	80	10	2015-0...	4132.000	5000.000	2025874	1235	10	100001	1
1700	80	20	2015-0...	2892.000	3500.000	2255748	1236	11	100002	2
1700	80	30	2015-0...	826.000	1000.000	2716848	1237	12	100003	3
1700	80	40	2015-0...	2272.000	2750.000	2785412	1238	13	100004	4
1700	80	50	2015-0...	413.000	500.000	7848501	1239	14	100005	5
1700	80	60	2015-0...	1446.000	1750.000	8336559	12310	15	100006	6
1700	80	70	2015-0...	2685.000	3250.000	123456...	12311	16	100007	7
1700	80	80	2015-0...	4132.000	5000.000	147852...	12312	17	100008	8
1700	80	90	2015-0...	6033.000	7300.000	351359...	12313	18	100009	9
1700	80	100	2015-1...	52.000	63.000	987321...	12314	19	100010	10