ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΣΤΗΝ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗ ΑΝΑΛΥΤΙΚΗ

Μάθημα: Data Management and Business Intelligence

Διδάσκων: Δ. ΧΑΤΖΗΑΝΤΩΝΙΟΥ

Εργασία: 1η

Όνομα: Ζούρου Μυρσίνη , Βογιατζής Γιώργος

Αριθμός Μητρώου: p2821828, p2821827

Ημερομηνία Παράδοσης: Τετάρτη 14 Νοεμβρίου 2018

Περιεχόμενα

1. Σκοπός της εργασίας

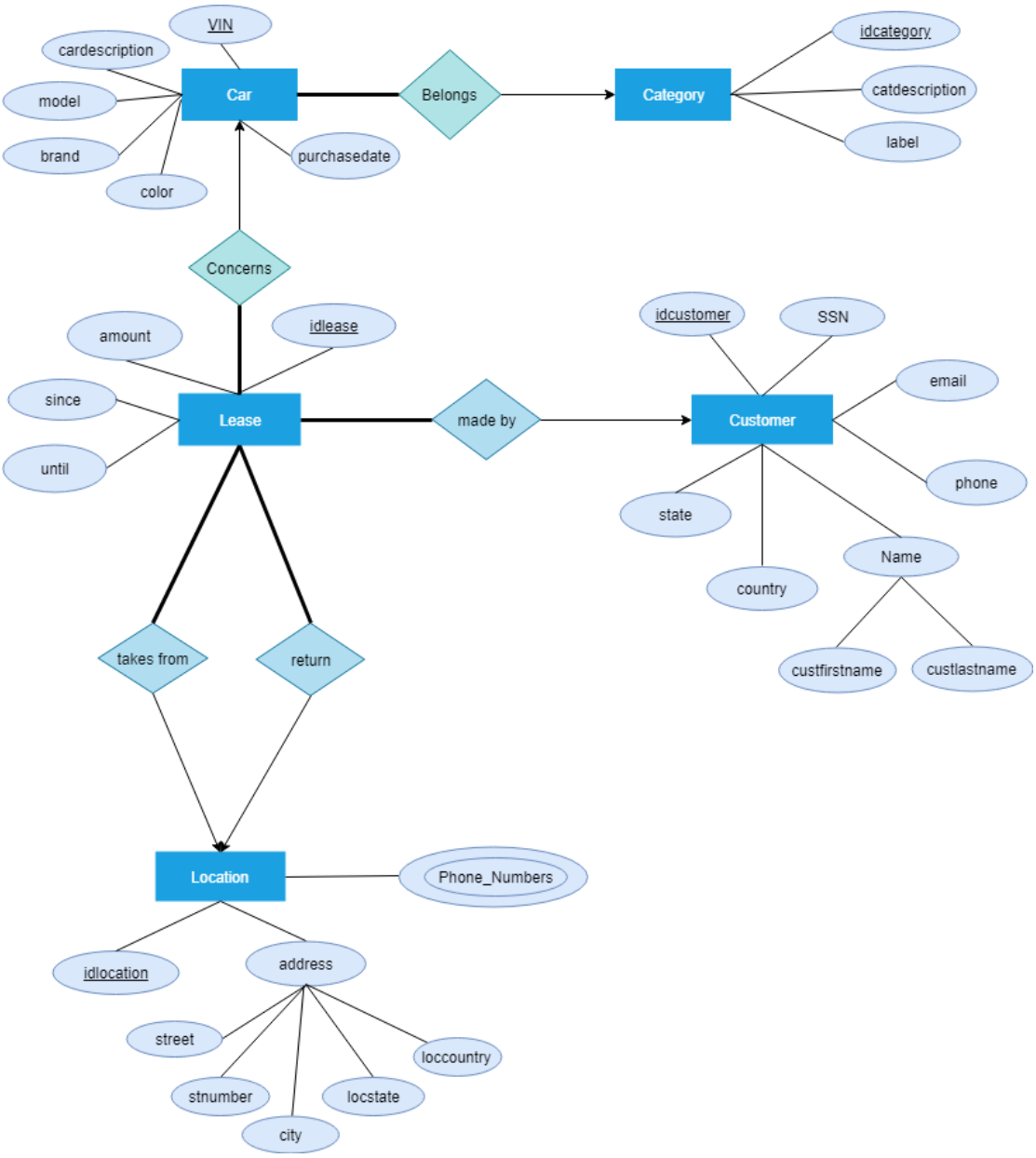
Ο σκοπός αυτής της εργασίας είναι να δημιουργήσουμε μια σχεσιακή βάση δεδομένων για μια εταιρεία ενοικίασης αυτοκινήτων που θα την ονομάσουμε CRC. Η οποία θα περιέχει πληροφορίες για:

- Τους πελάτες της εταιρείας.
- Τις ενοικιάσεις που πραγματοποιούνται.
- Τον στόλο των αυτοκινήτων που κατέχει η εταιρεία.
- Τις τοποθεσείες όπου γίνονται οι ενοικιάσεις.

Μια σχεσιακή βάση δεδομένων περιέχει πίνακες με τις πληροφορίες οπου κρίνονται απαραίτητες για την εταιρεία ή το σκοπό που εξυπηρετεί. Συγκεκριμένα η βάση CRC θα περιλαμβάνει πέντε αντικείμενα ενδιαφέροντος που ονομάζονται οντότητες. Αυτές είναι το αυτοκίνητο, η κατηγορία στην οποία αυτό εντάσσεται, οι πελάτες και η τοποθεσία παραλαβής-παράδοσης ενός αυτοκινήτου. Κάθε οντότητα έχει διάφορα στοιχεία που την προσδιορίζουν τα οποία ονομάζονται χαρακτηριστικά ή ιδιότητες της κάθε οντότητας.

Η οντότητα του αυτοκινήτου(Car) θα περιλαμβάνει τα ακόλουθα χαρακτηριστηκά: ένα μοναδικό κωδικό(VIN), την εταιρεία κατασκευής (brand), το μοντέλο (model), το χρώμα (color), την ημερομηνία αγοράς (purchasedate) και μια περιγραφή του αυτοκινήτου (cardescription). Η κατηγορία(category) που ανήκει το αυτοκίνητο θα αποτελέιται από ένα μοναδικό κωδικό (idcategory), έναν χαρακτηρισμό (label) και μία περιγραφή (catdescription). Αντίστοιχα η οντότητα που αφορά τον πελάτη θα έχει τα εξής χαρακτηριστικά ένα μοναδικό κωδικό(idcustomer), ΑΜΚΑ(SSN), όνομα(custfirstname), επίθετο(custlastname), περιοχή(state), email, χώρα διαμονής(country), τηλέφωνο (phone). Επίσης η οντότητα της τοποθεσίας(location) αποτελείται από έναν μοναδικό κωδικό(idlocation), διεύθυνση(address) και αριθμό τηλέφωνου (Phone_Numbers). Τέλος η ενοικίαση(lease) θα περιγράφεται από έναν μοναδικό κωδικό (idlease), ένα ποσό(amount), ημ/νία έναρξης(since) και ημ/νία λήξης (until).

Επιπρόσθετα είναι λογικό οι οντότητες να συνδέονται μεταξύ τους δηλαδή μια ενοικίαση να αφορά έναν πελάτη, το αυτοκίνητο να ανήκει σε μια κατηγορία. Αυτή η σύνδεση μεταξύ των οντοτήτων ονομάζεται συσχέτιση. Όλα τα παραπάνω μπορούν να αποτυπωθούν με την βοήθεια του διαγράμματος ER στο οποίο οι οντότητες απεικονίζονται με τερταγωνα, τα χαρακτηριστηκά/ιδιότητες με κύκλους και τέλος οι συσχετισεις με ρόμβους. Στην **Εικόνα 1** δίνεται το ER διάγραμμα της βάσης CRC.



Εικόνα 1: ER Διάγραμμα της βάσης CRC

2.Δημιουργία της βάσης στην MySQL

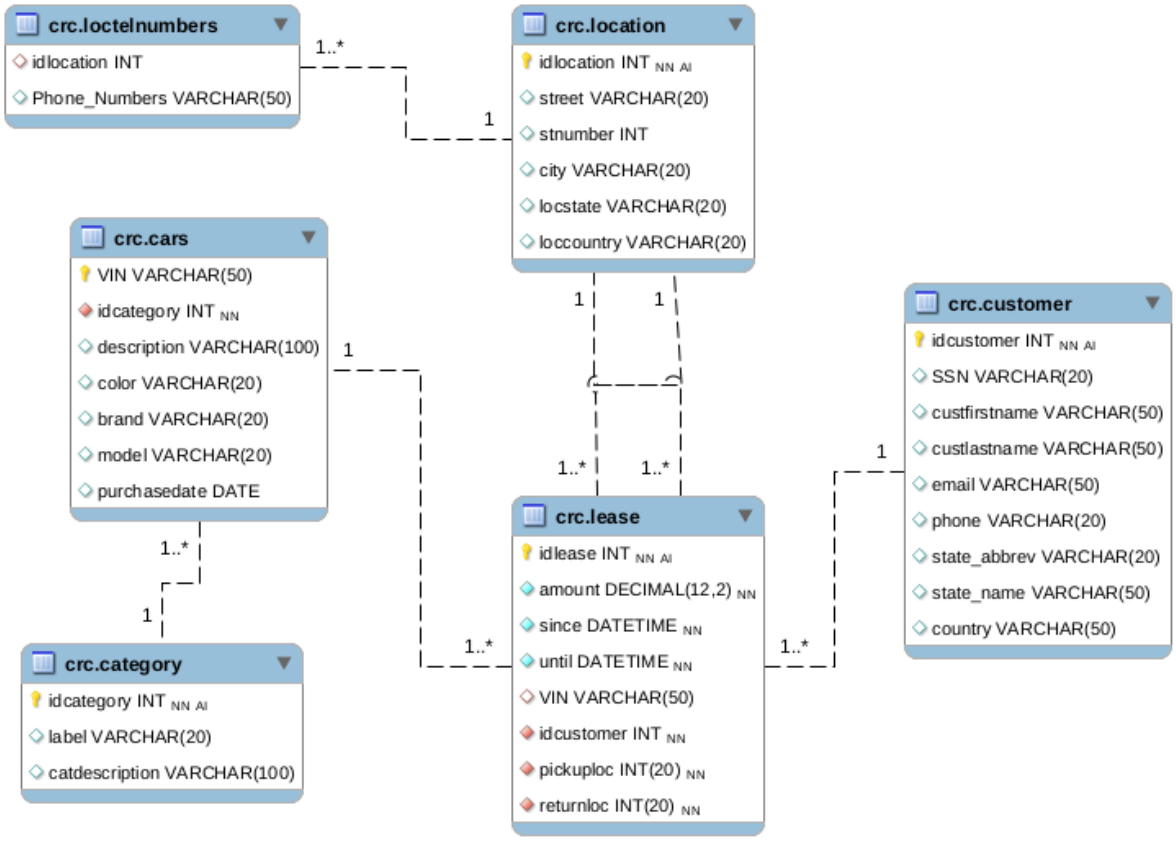
Στο προηγούμενο κεφάλαιο αναφερθηκαμε στο τι πληροφοριες πρέπει να περιέχει η βάση δεδομένων μας. Στο παρόν κεφάλαιο θα σχεδιάσουμε την βάση μας στην MySQL. Για να το καταφέρουμε αυτό γράψαμε και τρέχαμε το παρακάτω κώδικα:

```
-- -----------------------------------------------------
-- Schema CRC
-- -----------------------------------------------------

DROP SCHEMA IF EXISTS crc;
CREATE SCHEMA IF NOT EXISTS crc DEFAULT CHARACTER SET utf8 ;
USE crc ;

-- -----------------------------------------------------
-- Table CRC.category
-- -----------------------------------------------------

DROP TABLE IF EXISTS category;
CREATE TABLE IF NOT EXISTS category(
idcategory INT NOT NULL auto_increment,
label VARCHAR(20),
catdescription VARCHAR(100),
PRIMARY KEY(idcategory));

-- -----------------------------------------------------
-- Table CRC.cars
-- -----------------------------------------------------

DROP TABLE IF EXISTS cars;
CREATE TABLE IF NOT EXISTS cars(
VIN VARCHAR(50),
idcategory INT NOT NULL,
description VARCHAR(100),
color VARCHAR(20),
brand VARCHAR(20),
model VARCHAR(20),
purchasedate DATE,
PRIMARY KEY(VIN),
FOREIGN KEY(idcategory)
REFERENCES category(idcategory)  ON DELETE CASCADE);
```

```
-- -----------------------------------------------------
-- Table CRC.customer
-- -----------------------------------------------------

 DROP TABLE IF EXISTS customer;
 CREATE TABLE IF NOT EXISTS customer(
 idcustomer INT NOT NULL auto_increment,
 SSN VARCHAR(20),
 custfirstname VARCHAR(50),
 custlastname VARCHAR(50),
 email VARCHAR(50),
 phone VARCHAR(20),
 state_abbrev VARCHAR(20),
 state_name VARCHAR(50),
 country VARCHAR(50),
 PRIMARY KEY (idcustomer));
 ALTER TABLE customer AUTO_INCREMENT=10000;

-- -----------------------------------------------------
-- Table CRC.location
-- -----------------------------------------------------
 DROP TABLE IF EXISTS location;
 CREATE TABLE IF NOT EXISTS location(
 idlocation INT NOT NULL auto_increment,
 street VARCHAR(20),
 stnumber INT,
 city VARCHAR(20),
 locstate VARCHAR(20),
 loccountry VARCHAR(20),
 PRIMARY KEY(idlocation));

-- -----------------------------------------------------
-- Table CRC.loctelnumbers
-- -----------------------------------------------------

 DROP TABLE IF EXISTS loctelnumbers;
 CREATE TABLE IF NOT EXISTS loctelnumbers(
 idlocation INT,
 Phone_Numbers VARCHAR(50),
 FOREIGN KEY(idlocation)
 REFERENCES location(idlocation)   ON DELETE CASCADE);

-- -----------------------------------------------------
-- Table CRC.lease
-- -----------------------------------------------------

DROP TABLE IF EXISTS lease;
CREATE TABLE IF NOT EXISTS lease(
idlease  INT NOT NULL auto_increment,
amount DECIMAL(12,2) NOT NULL,
```

since DATETIME NOT NULL,
until DATETIME NOT NULL,
PRIMARY KEY(idlease),
VIN VARCHAR(50),
idcustomer INT NOT NULL,
pickuploc INT(20) NOT NULL,
returnloc INT(20) NOT NULL,
CHECK(amount>0),
FOREIGN KEY(idcustomer) REFERENCES customer(idcustomer) ON DELETE CASCADE,
FOREIGN KEY(pickuploc) REFERENCES location(idlocation) ON DELETE CASCADE,
FOREIGN KEY(returnloc) REFERENCES location(idlocation) ON DELETE CASCADE,
FOREIGN KEY(VIN) REFERENCES cars(VIN) ON DELETE CASCADE);

Στην **Εικόνα 2** δίνεται ένα διάγραμμα της βάσης με τον τύπο των μεταβητών και τις συσχετίσεις που υπάρχουν κατά αντιστοιχία με το ER διάγραμμα το οποίο δημιουργεί η MySQL.



Εικόνα 2: Διάγραμμα της βάσης που εξάγεται από την MySQL

3.Εξαγωγή δεδομένων από την MySQL

Στο παρόν κεφάλαιο σκοπός μας είναι να εξάγουμε συγκεκριμένες πληροφορίες από την βάση μας. Τα ερωτήματα στα οποία πρέπει να απαντήσουμε παρουσιάζονται παρακάτω όπου δίνεται και ο αντίστοιχος κωδικας της MySQL.

- Εξαγωγή όλων των κωδικών ενοικίασης και των κωδικών τοποθεσίας που έγιναν στης 20/05/2015.
  - ```
    select idlease,pickuploc
    from crc.lease
    where DATE(since)='20150520'
    ```

- Εξαγωγή του ονόματος, του επιθέτου και του αριθμού τηλεφώνου των ατόμων που έχουν ενοικιάσει αυτοκίνητο που ανήκει στην κατηγορία «luxury».

  - ```
    select distinct custfirstname, custlastname, phone
    from lease
    inner join customer using(idcustomer)
    inner join cars using(VIN)
    inner join category using(idcategory)
    where category.label='luxury';
    ```

- Εξαγωγή του συνολικού ποσού ενοικιάσεων ανά κωδικό τοποθεσίας.
  - ```
    select pickloc as'Location Id', sum(amount) as 'Total Amount'
    from lease
    group by pickuploc;
    ```

- Εξαγωγή του συνολικού ποσού ανά κατηγορία αυτοκινήτου και ανά μήνα.
  - ```
    select idcategory,monthname(since) as month, sum(amount) as 'Total Amount'
    from lease
    inner join cars using(VIN)
    inner join category using(idcategory)
    group by idcategory, monthname(since);
    ```

- Για κάθε περιοχή ενοικίασης να βρούμε την προτιμότερη κατηγορία αυτοκινητου. Όταν υπάρχει ισοψηφία παρουσιάζονται όλες οι κατηγορίες που έχουν εμφανιστεί τις περισσότερες φορές για την συγκεκριμένη περιοχή.
    - create view clabels (occurences, label, state) as
      select count(*), label, locstate
      from category, cars ,lease, location
      where category.idcategory = cars.idcategory and lease.VIN= cars.VIN and
      lease.pickuploc = location.idlocation
      group by locstate,label;

      select label, locstate
      from clabels
      where occurences=(select max(occurences)
      from clabels as mlabels
      where mlabels.state=clabels.state);

- Εξαγωγή του αριθμού ενοικοίασεων για τον Μάιο του 2015 στην 'NY', 'NJ', 'CA'.
    - select (sum(if(lease.pickuploc=location.idlocation and location.locstate='NY' and month(since)='05'and year(since)='2015',1,0))) as NY,
      (sum(if(lease.pickuploc=location.idlocation and location.locstate='NJ' and month(since)='05' and year(since)='2015',1,0))) as NJ,
      (sum(if(lease.pickuploc=location.idlocation and location.locstate='CA' and month(since)='05' and year(since)='2015',1,0))) as CA
      from lease, location;

- Για κάθε μήνα του έτους 2015 υπολόγισε τον αρθμό των ενοικίασεων που είχαν ποσό μεγαλύτερο από τον μέσο ποσό του μήνα αυτού.
    - create view monthaverage (monthavg, month) as
      select avg(amount), monthname(since)
      from lease
      where year(since)=2015
      group by monthname(since);

```
select count(idlease) as countId, month
from lease, monthaverage
where amount>monthavg and year(since)=2015 and month=monthname(since)
group by month;
```

- Για κάθε μήνα του έτους 2015 υπολόγισε την ποσοστιαία μεταβολή του συνολικού ποσού ενοικιάσεων προς το συνολικό ποσό ενοικιάσεων του αντίστοιχου μήνα του έτους 2014.
  - ```
    select lease2015.month, 100.0 * (lease2015.sum2015-
    lease2014.sum2014)/lease2014.sum2014 as pct_change
    from (select sum(amount) as sum2014, monthname(since) as month
        from lease
        where YEAR(since) = 2014
        group by  monthname(since)) as lease2014
       inner join
    (select sum(amount) as sum2015, monthname(since) as month
       from lease
       where year(since) = 2015
       group by  monthname(since)) as lease2015
       using(month);
    ```

- Για κάθε μήνα του 2015 εμφάνισε σε τρείς στήλες το συνολικό ποσό των ενοικίασεων των προηγούμενων μηνών, αυτού του μήνα και των επόμενων μηνών.
  - ```
    create view monthlySums (month, amount) as
    select month(since), sum(amount)
    from lease
    where year(since)=2015
    group by month(since);

    select t2.month, (sum(if(t1.month<t2.month, t1.amount, 0))) previous_months,
    (sum(if(t1.month=t2.month, t1.amount, 0))) current_month,
    (sum(if(t1.month>t2.month, t1.amount, 0))) following_months
    from monthlySums as t1, monthlySums as t2
    group by t2.month;
    ```

4.Σύνδεση της MySQL με την Java

Σε αυτό το κεφάλαιο δίνεται ο κώδικας της προγραμματιστικής γλώσσας Java με τον οποίο στόχος μας ήταν να εισάγουμε στον πίνακα των πελατών της βάσης νέες εγγραφές.

```
package importCsv;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

public class ImportCsv {

        private static final String url =
"jdbc:mysql://localhost:3306/crc?autoReconnect=true&useSSL=false";

private static final String user = "root";

private static final String password = "6978147098";

        public static void main(String args[])

  {

        loadCsvIntoDb();

  }

        private static void loadCsvIntoDb()

{

  try (Connection connection = DriverManager.getConnection(url, user, password))

{

 String loadQuery = "LOAD DATA LOCAL INFILE '" +
"/home/gvogias/Desktop/Assignment_1_Customers.csv" +

"' INTO TABLE crc.customer FIELDS TERMINATED BY ',' + " LINES TERMINATED BY '\n'
(idcustomer,custfirstname,custlastname,email,SSN,phone,state_abbrev,state_name,country
) ";

        System.out.println(loadQuery);

         Statement stmt = connection.createStatement();

         stmt.execute(loadQuery);

            connection.close();

              }
```

```
catch (Exception e)
        {   e.printStackTrace(); }    } }
```

4.1 Εξαγωγή δεδομένων της βάση μέσω της Java

Σε αυτή την υποενότητα δίνεται ο κώδικας σε Java με τον οποίο εξάγουμε δεδομένα από τον πίνακα των ενοικιάσεων της βάσης και μετέπειτα να υπολογίσουμε μέσω αυτής το για κάθε μήνα του 2015 το συνολικό ποσό των ενοικιάσεων των προηγούμενων μηνών, αυτού του μήνα και των επόμενων μηνών

```
package groupBy;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.DriverManager;

import java.sql.Statement;

import java.text.DecimalFormat;

public class groupBy {

        private static final String url =
"jdbc:mysql://localhost:3306/crc?autoReconnect=true&useSSL=false";

        private static final String user = "root";

        private static final String password = "6978147098";

        private static Double[] monthAmounts = new Double[12];

        private static DecimalFormat df = new DecimalFormat("0.00");

        public static void main(String args[])

  {

                aggregateData();

  }


        private static void aggregateData()


  {

                        for(int i=0; i<12; i++)

                {
```

```
                    monthAmounts[i]= 0.0;
                }

    try (Connection connection = DriverManager.getConnection(url, user, password))
    {
      String query = "select amount, month(since) as month from crc.lease where
year(since)=2015";
      Statement stmt = connection.createStatement();
      ResultSet rs = null;
      rs = stmt.executeQuery(query);

      while(rs.next())
      {
        int month = rs.getInt("month");
        double amount = rs.getDouble("amount");

        if(month==1)
                monthAmounts[month-1]+=amount;
        else if(month==2)
                monthAmounts[month-1]+=amount;
        else if(month==3)
                monthAmounts[month-1]+=amount;
        else if(month==4)
                monthAmounts[month-1]+=amount;
        else if(month==5)
                monthAmounts[month-1]+=amount;
        else if(month==6)
                monthAmounts[month-1]+=amount;
        else if(month==7)
                monthAmounts[month-1]+=amount;
        else if(month==8)
```

```
            monthAmounts[month-1]+=amount;
  else if(month==9)
            monthAmounts[month-1]+=amount;
  else if(month==10)
            monthAmounts[month-1]+=amount;
  else if(month==11)
            monthAmounts[month-1]+=amount;
  else if(month==12)
            monthAmounts[month-1]+=amount;
}
rs.close();
stmt.close();
connection.close();
 System.out.println("Month\tPrevious months\tCurrent month\tFollowing months");
for (int month = 0; month < 12; month++)
{
  System.out.print(month+1+"\t");
  double previousMonthsAmount = 0;
  for (int p = 0; p < month; p++)
  {
            previousMonthsAmount += monthAmounts[p];
  }
  System.out.printf("%10s",df.format(previousMonthsAmount));
  System.out.print("\t");
  System.out.printf("%10s",df.format(monthAmounts[month]));
  System.out.print("\t");
  double nextMonthsAmount = 0;
  for (int n = month + 1; n < 12; n++)
  {
            nextMonthsAmount+= monthAmounts[n];
  }
```

```
            System.out.println("\t"+df.format(nextMonthsAmount));
          }
        }
        catch (Exception e)
        {
              e.printStackTrace();
        }
    }
}
```