DATA RESOURCES MANAGEMENT

PROJECT

Donation System

21.01.2010 v.3

APPLICATION DESCRIPTION

There was no integrated system in BUVAK which keeps track of the relationship between a donator and donations deeply. Therefore, the staff was not included in the donation process, thus the responsibility was separated each personnel even if some of them were not related with a specific donation. Previous system records the donator as members of the foundation even if they did not donate, and donation with donator names in different databases. This causes data redundancy and lack of data accuracy. According to this problem, conflicts in donator definition may happen, managerial reports are hard to prepare. For a foundation, relationships with members are the most important issue, thus ease in evaluating a member with its donation was necessary.

According to those problems we mentioned above, we plan to design an integrated system for BUVAK which aims to clarify the relationships of donator, staff and payment, decreasing the waste of time of donation processed and evaluated. The system not only provides us in terms of time management, but also provides us to increase the pleasure of members.

By using the system, each process at the donation can be controlled and monitored. Each staff is aware of which transactions and adjustments are executed by which staff, so any confusion or erroneous recording in process can be controlled faster. In addition, honoring a donator is automated by the system, awarding a donator can be realized in real time, so donation is encouraged. Moreover, the management of donations and ease of reporting will increase the effectiveness of the foundation.

The system is used by the foundation staff. Each staff logs in to the system with his staff id in order to make a relation between a donator and a function. The system is used via computers at each donation points and at the foundation. We design the system with user-friendly interfaces to ease the donation process and to increase the accuracy of data given.

Roles of the user in the system are explained briefly below:

- Staff: Each staff is assigned to different donation points. A staff is responsible with identifying an account to donators according to their donation type. For individual donations inputting donator information may be sufficient, while group donations need to be identified with the members of the group and other specifications. Staff is also responsible with the payment process. When a donator wants to make a payment, staff inputs the payment information and performs the transaction afterwards. In addition, staff is authorized by monitoring and adjusting the data.

BUSINESS RULES

- The users of the system are the staffs working in the foundation with different positions, each are identified by a user id and a password.

- We keep id, password, first name, last name, phone number, location and the position of the staff.

- Each staff assigned to a donation point, which are the foundation locations. A staff works at a single location.

- Donators are defined by their donator id, name, e-mail, telephone number, contact address and city. If a donator is a private person, then his birthdate, occupation, graduated university, graduation year and graduation department are recorded. If a donator is an artificial person, then contact person name is recorded additionally. Donators are separated into these two types. No other type can be created and a donator cannot be included into both.

- Different honor titles are given to donators according to their total amount donated. Total amount is calculated from the payments which may belong to any of the donation types.

- Donations are grouped into two: individual donations, and group donations. A donation cannot include more than one, and there are no other types of donation.

- If a single donator wants to pay the donation poll, it is called individual donation. A donator may donate as individual many times with different donation ids, or may divide its payment into installments within a single individual donation. Individual donations are recorded with an id and creation date.

- If more than one donator identifies themselves as a group, and pays the poll separately within the group name, it is called group donation. A group has an id, name, creation date, total amount paid. Also a group has many members and payment information of each is recorded.

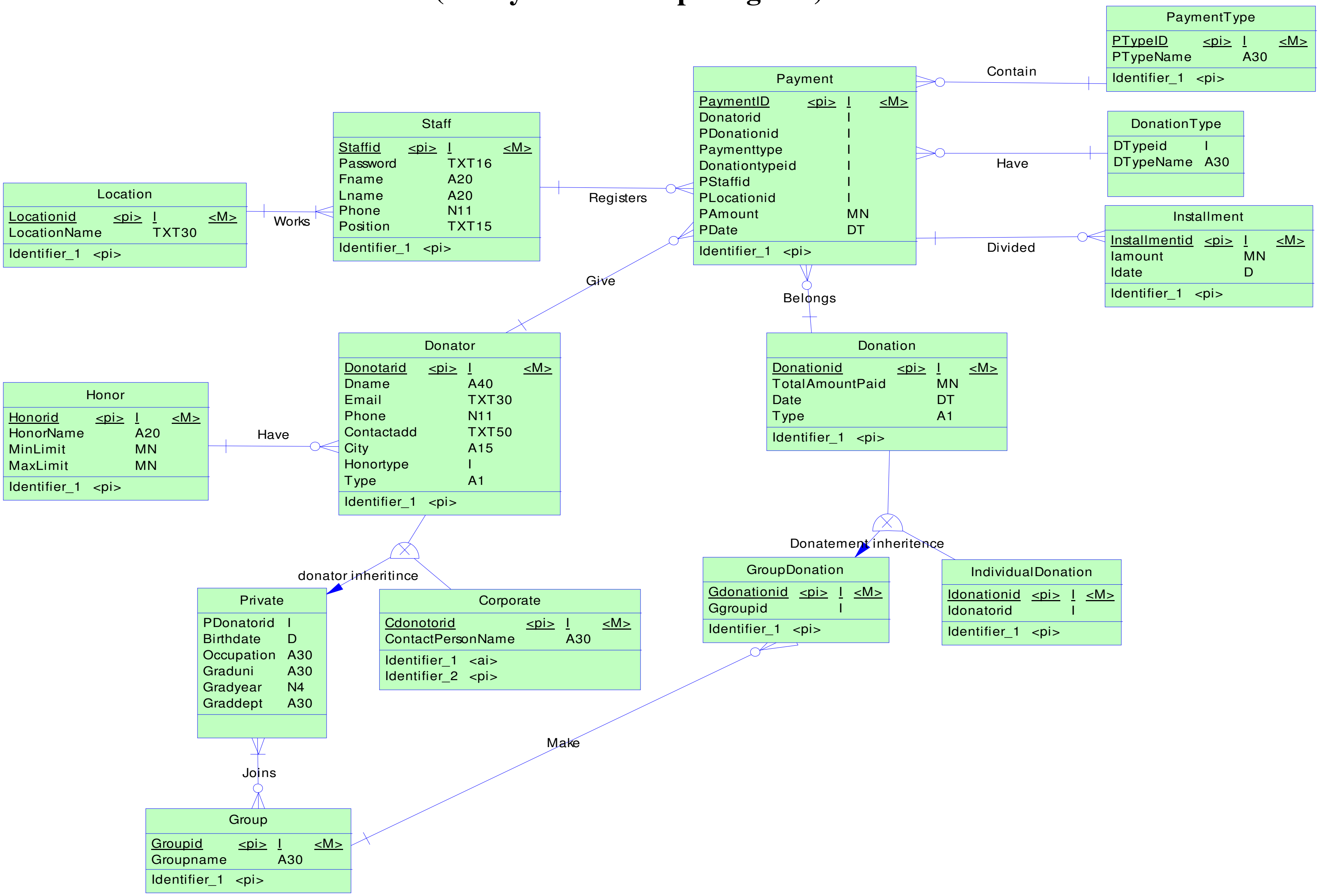
- A donator is related with a donation via payment. A payment is done for a single donation from a single location via a single payment choice. Each payment is performed by a single staff. On the other hand, a donation may have many payments from different locations and via different payment choices. A donator may perform many payments under different donations, or divide a single payment to a single donation into installments. However, each payment belongs to a single donator.

- Installments are donator-defined payment schedules, which may be defined for a single payment. Each installment has a date, amount and remaining amount. Installments are optional for payments.

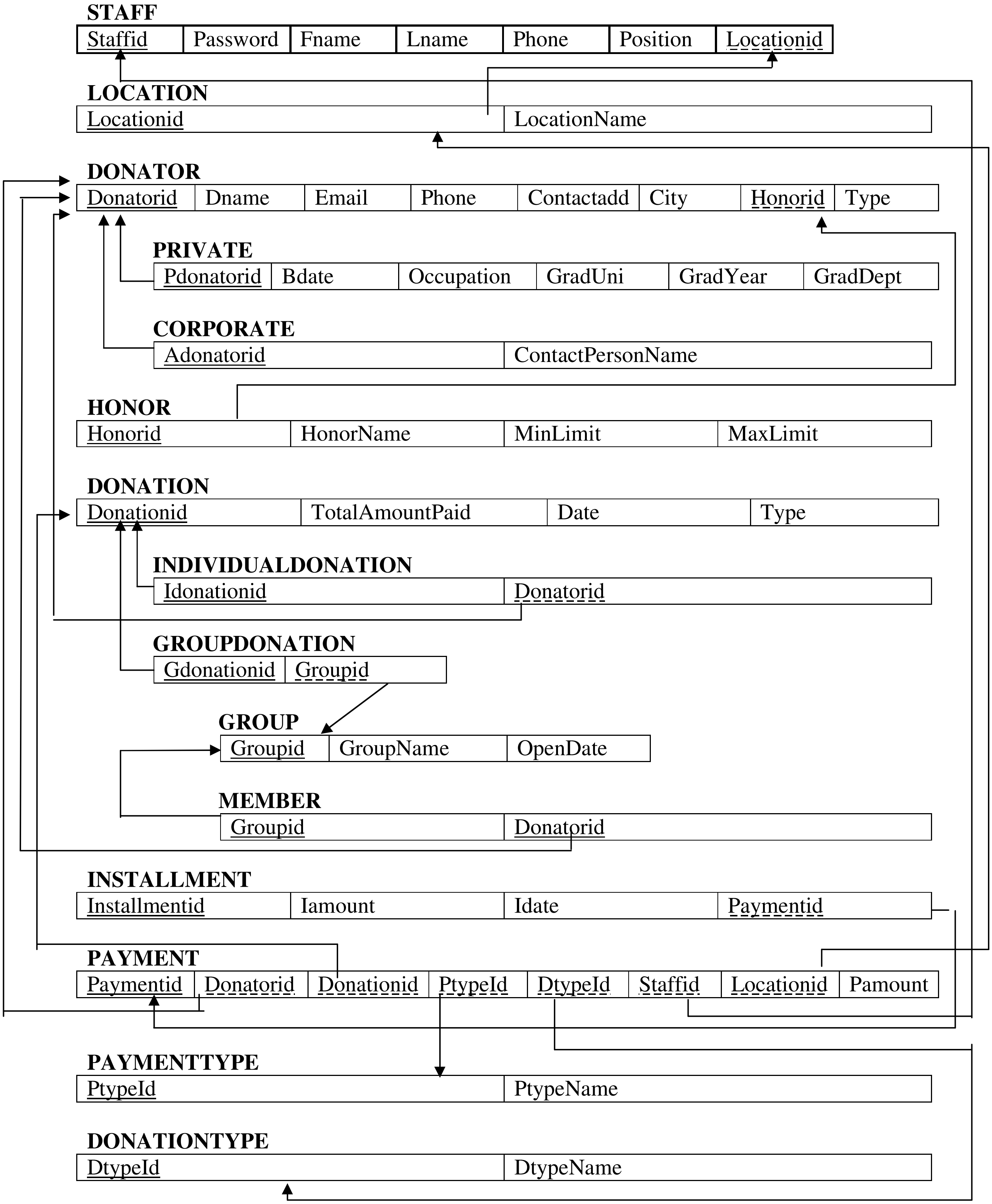
- A donator may realize its payment via cash, credit card, check, etc. which are payment types in the system. A payment can only be paid via one payment type.

- A donation may not be necessarily restricted with money donation option. A donation may include staff donation or income yielding property. However, whether a donator pays cash or else, total estimated cash values are recorded.

CONCEPTUAL DATA MODEL (Entity Relationship Diagram)



LOGICAL DATA MODEL (Normalized Relational Schema)



STAFF

Staffid	Password	Fname	Lname	Phone	Position	Locationid

LOCATION

Locationid	LocationName

DONATOR

Donatorid	Dname	Email	Phone	Contactadd	City	Honorid	Type

PRIVATE

Pdonatorid	Bdate	Occupation	GradUni	GradYear	GradDept

CORPORATE

Adonatorid	ContactPersonName

HONOR

Honorid	HonorName	MinLimit	MaxLimit

DONATION

Donationid	TotalAmountPaid	Date	Type

INDIVIDUALDONATION

Idonationid	Donatorid

GROUPDONATION

Gdonationid	Groupid

GROUP

Groupid	GroupName	OpenDate

MEMBER

Groupid	Donatorid

INSTALLMENT

Installmentid	Iamount	Idate	Paymentid

PAYMENT

Paymentid	Donatorid	Donationid	PtypeId	DtypeId	Staffid	Locationid	Pamount

PAYMENTTYPE

PtypeId	PtypeName

DONATIONTYPE

DtypeId	DtypeName

PHYSICAL DATA MODEL

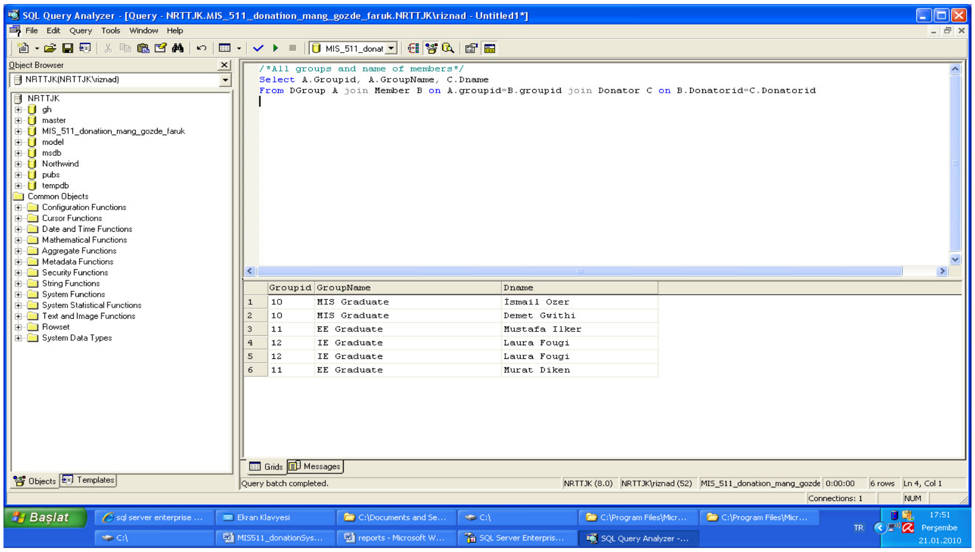
```
CREATE TABLE STAFF (
        Staffid         int          NOT NULL IDENTITY(100,1),
        Password        char(5)      NOT NULL,
        Fname           char(25)     NOT NULL,
        Lname           char(25)     NOT NULL,
        Phone           numeric(11)  NULL,
        Position        char         NULL,
        Locationid      int          NULL,

        CONSTRAINT StaffPK           PRIMARY KEY(Staffid),
        CONSTRAINT StaffLocationFK   FOREIGN KEY(Locationid)
                                              REFERENCES LOCATION (Locationid),
                                              ON UPDATE CASCADE,
                                              ON DELETE NOACTION,
);

CREATE TABLE LOCATION (
        Locationid      int          NOT NULL IDENTITY(200,1),
        LocationName    char(25)     NOT NULL,

        CONSTRAINT LocationPK        PRIMARY KEY(Locationid),
);


CREATE TABLE DONATOR (
        Donatorid       int          NOT NULL IDENTITY(1000,1),
        Dname           char(25)     NOT NULL,
        Email           char(100)    NULL,
        Phone           numeric(11)  NULL,
        ContactAdd      char(100)    NULL,
        City            char(20)     NULL,
        Type            char(2)      NOT NULL,
        Honorid         int          NOT NULL,

        CONSTRAINT DonatorPK         PRIMARY KEY(Donatorid),
        CONSTRAINT DonatorHonorFK    FOREIGN KEY(Honorid)
                                              REFERENCES HONOR (Honorid),
                                              ON UPDATE NO ACTION,
                                              ON DELETE NO ACTION,
        CONSTRAINT TypeValues        CHECK (Type IN (‘P’,’C’)),
);

CREATE TABLE PRIVATE (
        Pdonatorid      int          NOT NULL IDENTITY(1000,1),
        Bdate           datetime     NULL,
        Occupation      char(25)     NULL,
        GradUni         char(50)     NULL,
        GradYear        numeric(4)   NULL,
        GradDept        char(50)     NULL,

        CONSTRAINT PrivatePK         PRIMARY KEY(Pdonatorid),
        CONSTRAINT PrivateDonatorFK FOREIGN KEY(Pdonatorid)
                                              REFERENCES DONATOR (Donatorid),
                                              ON UPDATE CASCADE,
                                              ON DELETE CASCADE,
        CONSTRAINT GradYearValues    CHECK (GradYear LIKE ‘[1-2][0-9][0-9][0-9]’)
);
```

```
CREATE TABLE CORPORATE (
        Cdonatorid              int       NOT NULL IDENTITY (1000,1),
        ContactPersonName       char(50)  NOT NULL,

        CONSTRAINT CorporatePK            PRIMARY KEY (Cdonatorid),
        CONSTRAINT CorporateFK            FOREIGN KEY (Cdonatorid)
                                                  REFERENCES DONATOR (Donatorid),
                                                  ON UPDATE CASCADE,
                                                  ON DELETE CASCADE,
);

CREATE TABLE HONOR (
        Honorid         int       NOT NULL IDENTITY (1,1),
        HonorName       char(50)  NOT NULL,
        MinLimit        numeric   NOT NULL,
        MaxLimit        numeric   NOT NULL,

        CONSTRAINT HonorPK PRIMARY KEY (Honorid),
);

CREATE TABLE DONATION (
        Donationid      int       NOT NULL IDENTITY (10000,1)
        TotalAmount     numeric   NOT NULL,
        Date            datetime  NOT NULL,
        Type            char(2)   NOT NULL,

        CONSTRAINT DonationPK             PRIMARY KEY (Donationid),
        CONSTRAINT TypeValues             CHECK (Type IN (‘I’,’G’)),
);


CREATE TABLE INDIVIDUALDONATION (
        Idonationid     int       NOT NULL,
        Donatorid       int       NOT NULL,

        CONSTRAINT IndividualPK                   PRIMARY KEY (Idonationid),
        CONSTRAINT IndividualFK                   FOREIGN KEY (Donatorid)
                                                          REFERENCES DONATOR (Donatorid),
                                                          ON UPDATE NO ACTION,
                                                          ON DELETE NO ACTION,
        CONSTRAINT IndividualFK                   FOREIGN KEY (Idonationid)
                                                          REFERENCES DONATION (Donationid),
                                                          ON UPDATE CASCADE,
                                                          ON DELETE CASCADE,
);

CREATE TABLE GROUPDONATION (
        Gdonationid     int               NOT NULL IDENTITY(1,1),
        Groupid         int               NOT NULL,

        CONSTRAINT GroupPK                        PRIMARY KEY (Gdonationid),
        CONSTRAINT GroupDonationFK                FOREIGN KEY (Gdonationid)
                                                          REFERENCES DONATION (Donationid),
                                                          ON UPDATE CASCADE,
                                                          ON DELETE CASCADE,
        CONSTRAINT GroupFK                        FOREIGN KEY (Groupid)
                                                          REFERENCES GROUP (Groupid),
                                                          ON UPDATE NO ACTION,
                                                          ON DELETE NO ACTION,
);
```

```
CREATE TABLE GROUP(
    Groupid int                         NOT NULL IDENTITY (10,1),
    GroupName       char(30)            NOT NULL,
    OpenDate        datetime            NOT NULL,
    TotalAmount     numeric             NOT NULL,

    CONSTRAINT GroupPK                          PRIMARY KEY (Groupid),
);

CREATE TABLE MEMBER (
    Groupid         int         NOT NULL,
    Donatorid       int         NOT NULL,

    CONSTRAINT MemberPK                         PRIMARY KEY (Groupid, Donatorid),
    CONSTRAINT MemberGroupFK                    FOREIGN KEY (Groupid)
                                                        REFERENCES GROUP (Groupid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
    CONSTRAINT MemberDonatorFK                  FOREIGN KEY (Donatorid)
                                                        REFERENCES DONATOR (Donatorid),
                                                        ON UPDATE NO ACTION,
                                                         DELETE NO ACTION,
);

CREATE TABLE INSTALLMENT (
    Installmentid   int                 NOT NULL IDENTITY (1,1),
    Iamount         numeric             NOT NULL,
    Idate           datetime            NOT NULL,
    Paymentid       int                 NOT NULL,

    CONSTRAINT InstallmentPK                    PRIMARY KEY(Installmentid),
    CONSTRAINT InstallmentPaymentFK             FOREIGN KEY(Paymentid)
                                                        REFERENCES PAYMENT(Paymentid),
                                                        ON UPDATE CASCADE,
                                                        ON DELETE CASCADE,
);

CREATE TABLE PAYMENT(
    Paymentid       int         NOT NULL IDENTITY(1,1),
    Donatorid       int         NOT NULL,
    Donationid      int         NOT NULL,
    Ptypeid         int         NOT NULL,
    Dtypeid         int         NOT NULL,
    Staffid         int         NULL,
    Locationid      int         NULL,
    Pamount         numeric NOT NULL,
    Pdate           datetimeNULL DEFAULT CURRENT_TIMESTAMP,

    CONSTRAINT PaymentPK                        PRIMARY KEY(Paymentid),
    CONSTRAINT PaymentDonatorFK                 FOREIGN KEY (Dontorid)
                                                        REFERENCES DONATOR(Donatorid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
    CONSTRAINT PaymentDonationFK                FOREIGN KEY (Donationid)
                                                        REFREENCES DONATION(Donationid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
    CONSTRAINT PaymentTypeFK                    FOREIGN KEY (Ptypeid)
                                                        REFERENCES PAYMENTYPE(Ptypeid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
                CONSTRAINT PaymentDTypeFK           FOREIGN KEY (Dtypeid)
                                                        REFERENCES DONATIONTYPE (Dtypeid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
                CONSTRAINT PaymentStaffFK           FOREIGN KEY (Staffid)
                                                        REFERENCES STAFF (Staffid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
                CONSTRAINT PaymentLocationFK        FOREIGN KEY (Locationid)
                                                        REFERENCES LOCATION (Locationid),
                                                        ON UPDATE NO ACTION,
                                                        ON DELETE NO ACTION,
);

CREATE TABLE PAYMENTTYPE(
                PTypeid         int         NOT NULL IDENTITY(1,1),
                PTypeName       char(10) NOT NULL,

                CONSTRAINT PTypePK                  PRIMARY KEY (PTypeid),
                CONSTRAINT PTypeAK                  UNIQUE (PTypeName),
);

CREATE TABLE DONATIONTYPE(
                DTypeid         int         NOT NULL IDENTITY(1,1),
                DTypeName       char(10) NOT NULL,

                CONSTRAINT DTypePK                          PRIMARY KEY(DTypeid),
                CONSTRAINT DTypeNameAK                      UNIQUE(DTypeName),
);
```

OPERATIONS AND REPORTS

```
/*All groups and name of members*/
Select A.Groupid, A.GroupName, C.Dname
From DGroup A join Member B on A.groupid=B.groupid join Donator C on
B.Donatorid=C.Donatorid;
```

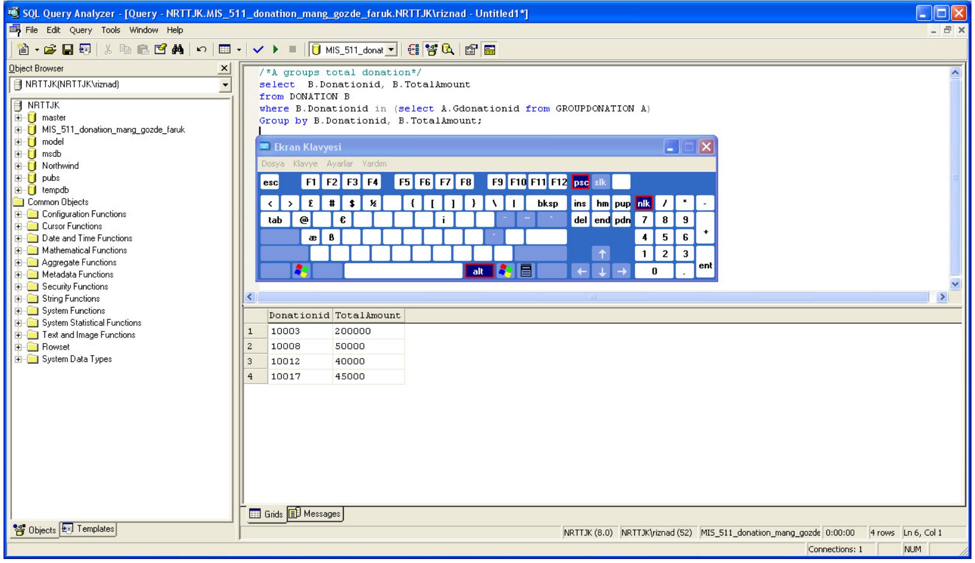


```
/*All groups and name of members*/
Select A.Groupid, A.GroupName, C.Dname
From DGroup A join Member B on A.groupid=B.groupid join Donator C on B.Donatorid=C.Donatorid
```

	Groupid	GroupName	Dname
1	10	MIS Graduate	İsmail Ozer
2	10	MIS Graduate	Demet Gwithi
3	11	EE Graduate	Mustafa Ilker
4	12	IE Graduate	Laura Fougi
5	12	IE Graduate	Laura Fougi
6	11	EE Graduate	Murat Diken

Query batch completed. NRTTJK (8.0) NRTTJK\riznad (52) MIS_511_donation_mang_gozde 0:00:00 6 rows Ln 4, Col 1

```
/*A groups total donation*/
select  B.Donationid, B.TotalAmount
from DONATION B
where B.Donationid in (select A.Gdonationid from GROUPDONATION A)
Group by B.Donationid, B.TotalAmount;
```

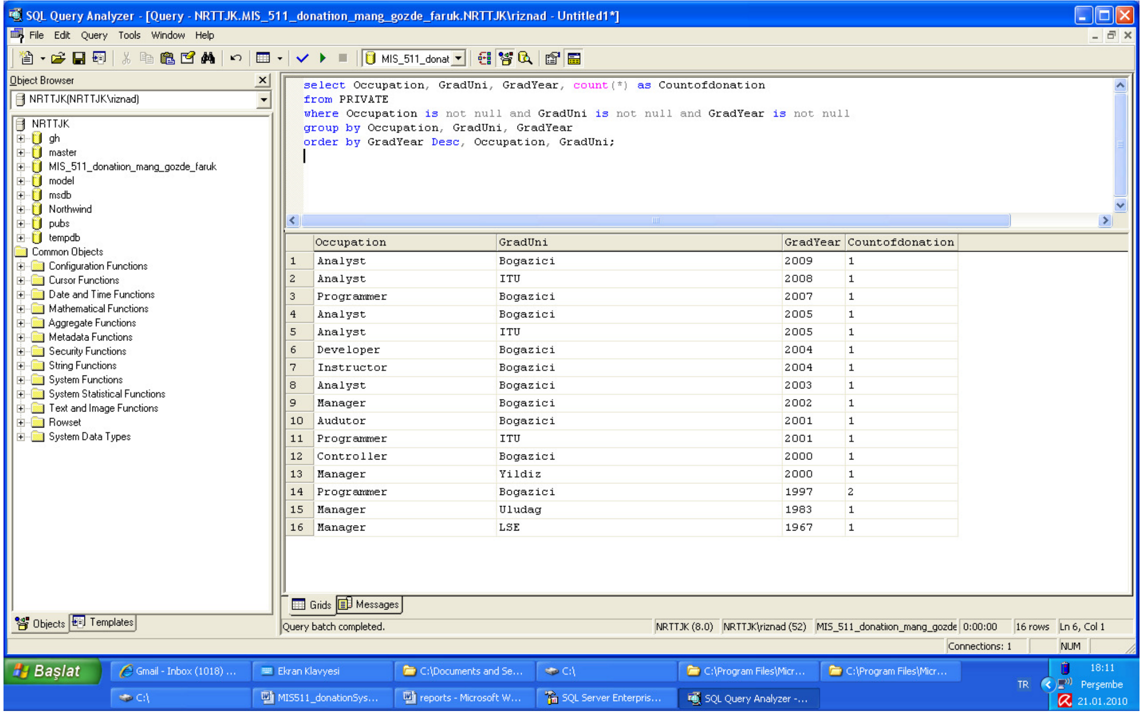


```
/*A groups total donation*/
select  B.Donationid, B.TotalAmount
from DONATION B
where B.Donationid in (select A.Gdonationid from GROUPDONATION A)
Group by B.Donationid, B.TotalAmount;
```

	Donationid	TotalAmount
1	10003	200000
2	10008	50000
3	10012	40000
4	10017	45000

SQL Query Analyzer - [Query - NRTTJK.MIS_511_donation_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
Ekran Klavyesi
Grids
Messages
4 rows
Ln 6, Col 1

```
select Occupation, GradUni, GradYear, count(*) as Countofdonation
from PRIVATE
where Occupation is not null and GradUni is not null and GradYear is not null
group by Occupation, GradUni, GradYear
order by GradYear Desc, Occupation, GradUni;
```



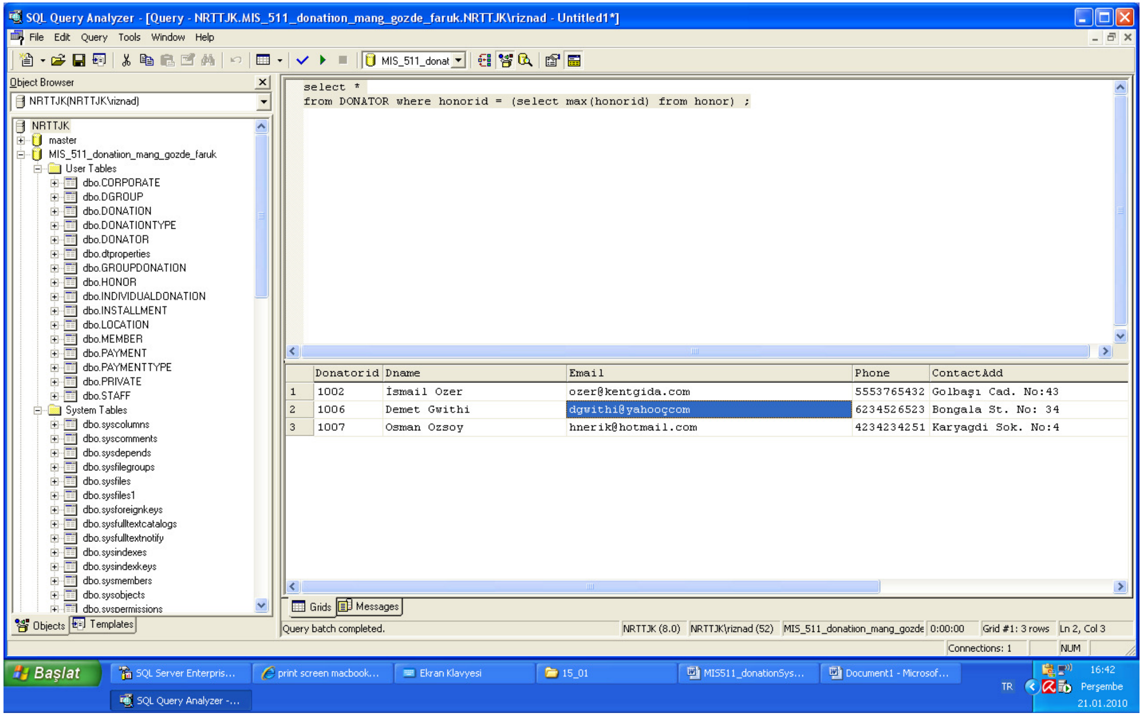
```
select Occupation, GradUni, GradYear, count(*) as Countofdonation
from PRIVATE
where Occupation is not null and GradUni is not null and GradYear is not null
group by Occupation, GradUni, GradYear
order by GradYear Desc, Occupation, GradUni;
```

	Occupation	GradUni	GradYear	Countofdonation
1	Analyst	Bogazici	2009	1
2	Analyst	ITU	2008	1
3	Programmer	Bogazici	2007	1
4	Analyst	Bogazici	2005	1
5	Analyst	ITU	2005	1
6	Developer	Bogazici	2004	1
7	Instructor	Bogazici	2004	1
8	Analyst	Bogazici	2003	1
9	Manager	Bogazici	2002	1
10	Audutor	Bogazici	2001	1
11	Programmer	ITU	2001	1
12	Controller	Bogazici	2000	1
13	Manager	Yildiz	2000	1
14	Programmer	Bogazici	1997	2
15	Manager	Uludag	1983	1
16	Manager	LSE	1967	1

Query batch completed.

SQL Query Analyzer - [Query - NRTTJK.MIS_511_donation_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
Grids
Messages
16 rows
Ln 6, Col 1
Başlat
18:11
Perşembe
21.01.2010

select *
from DONATOR where honorid = (select max(honorid) from honor) ;

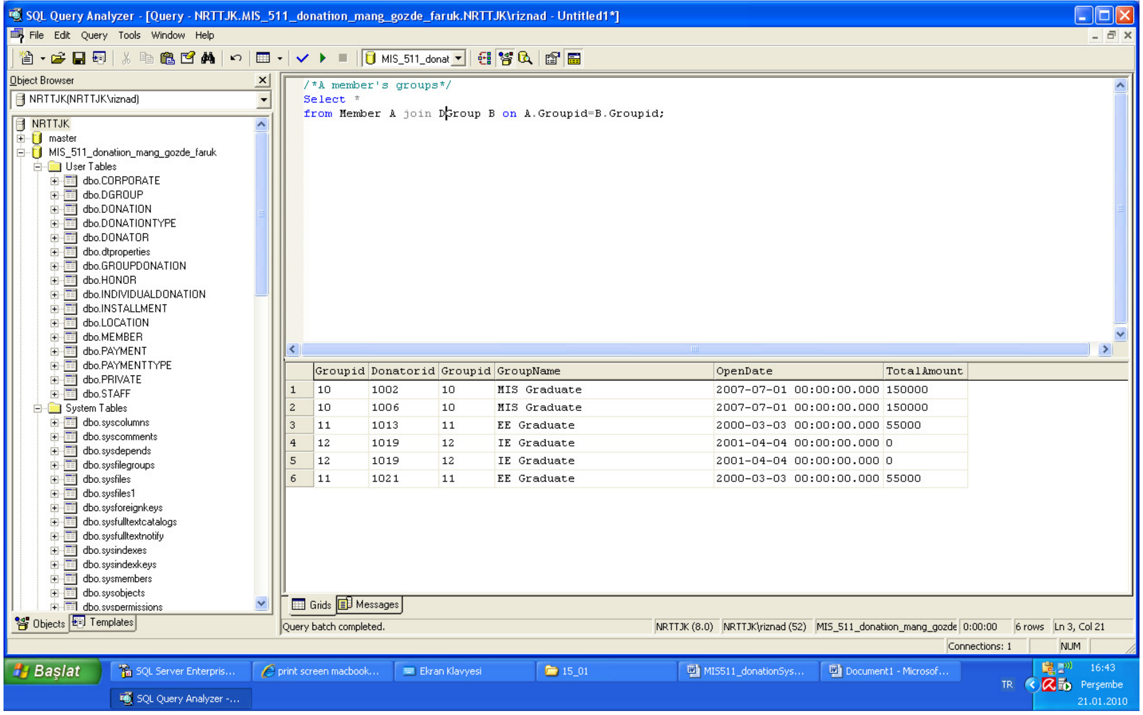

SQL Query Analyzer - [Query - NRTTJK.MIS_511_donatiion_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
File Edit Query Tools Window Help
Object Browser
NRTTJK(NRTTJK\riznad)
Objects
Templates
Grids
Messages
Başlat

```
select *
from DONATOR where honorid = (select max(honorid) from honor) ;
```

	Donatorid	Dname	Email	Phone	ContactAdd
1	1002	İsmail Ozer	ozer@kentgida.com	5553765432	Golbaşı Cad. No:43
2	1006	Demet Gwithi	dgwithi@yahooçcom	6234526523	Bongala St. No: 34
3	1007	Osman Ozsoy	hnerik@hotmail.com	4234234251	Karyagdi Sok. No:4

Query batch completed. NRTTJK (8.0) NRTTJK\riznad (52) MIS_511_donatiion_mang_gozde 0:00:00 Grid #1: 3 rows Ln 2, Col 3

/*A member's groups*/
Select *
from Member A join Group B on A.Groupid=B.Groupid;

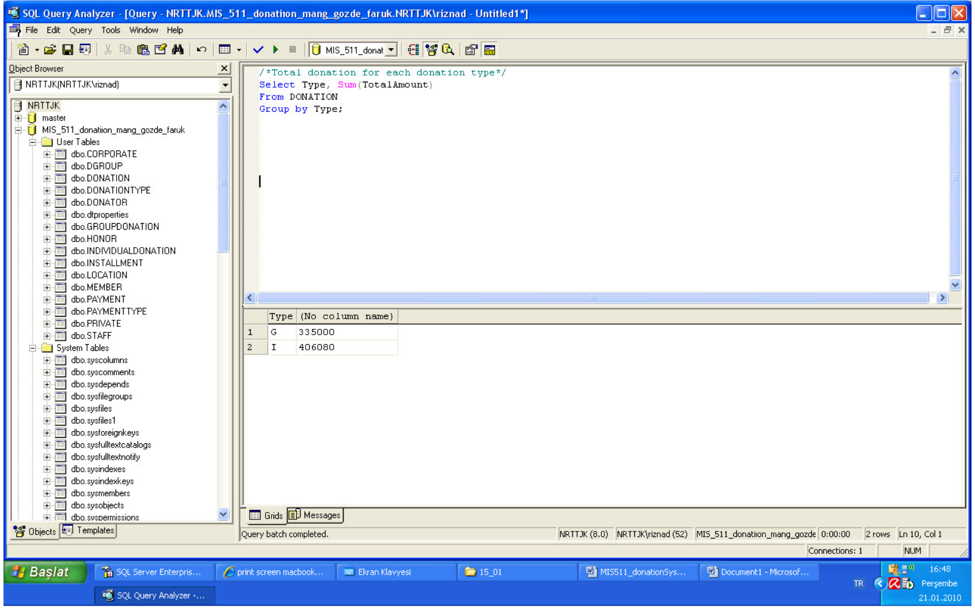

SQL Query Analyzer - [Query - NRTTJK.MIS_511_donatiion_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
File Edit Query Tools Window Help
Object Browser
NRTTJK(NRTTJK\riznad)
Objects
Templates
Grids
Messages
Başlat

```
/*A member's groups*/
Select *
from Member A join DGroup B on A.Groupid=B.Groupid;
```

	Groupid	Donatorid	Groupid	GroupName	OpenDate	TotalAmount
1	10	1002	10	MIS Graduate	2007-07-01 00:00:00.000	150000
2	10	1006	10	MIS Graduate	2007-07-01 00:00:00.000	150000
3	11	1013	11	EE Graduate	2000-03-03 00:00:00.000	55000
4	12	1019	12	IE Graduate	2001-04-04 00:00:00.000	0
5	12	1019	12	IE Graduate	2001-04-04 00:00:00.000	0
6	11	1021	11	EE Graduate	2000-03-03 00:00:00.000	55000

Query batch completed. NRTTJK (8.0) NRTTJK\riznad (52) MIS_511_donatiion_mang_gozde 0:00:00 6 rows Ln 3, Col 21

```
/*Total donation for each donation type*/
Select Type, Sum(TotalAmount)
From Donation
Group by Type;
```



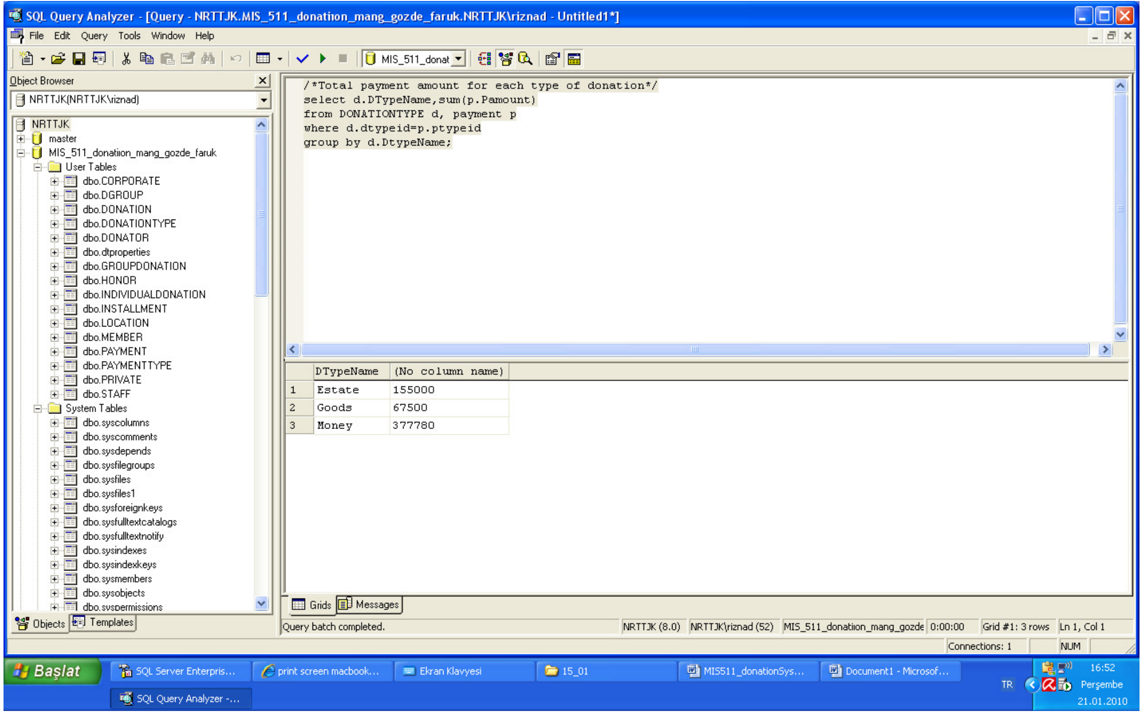
SQL Query Analyzer - [Query - NRTTJK.MIS_511_donatiion_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
File Edit Query Tools Window Help
MIS_511_donat
Object Browser
NRTTJK(NRTTJK\riznad)
Objects
Templates
Grids
Messages

```
/*Total donation for each donation type*/
Select Type, Sum(TotalAmount)
From DONATION
Group by Type;
```

```
   Type  (No column name)
1  G     335000
2  I     406080
```

Query batch completed.

```
/*Total payment amount for each type of donation*/
select d.DTypeName,sum(p.Pamount)
from DONATIONTYPE d, payment p
where d.dtypeid=p.ptypeid
group by d.DtypeName;
```



SQL Query Analyzer - [Query - NRTTJK.MIS_511_donatiion_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]
File Edit Query Tools Window Help
MIS_511_donat
Object Browser
NRTTJK(NRTTJK\riznad)
Objects
Templates
Grids
Messages

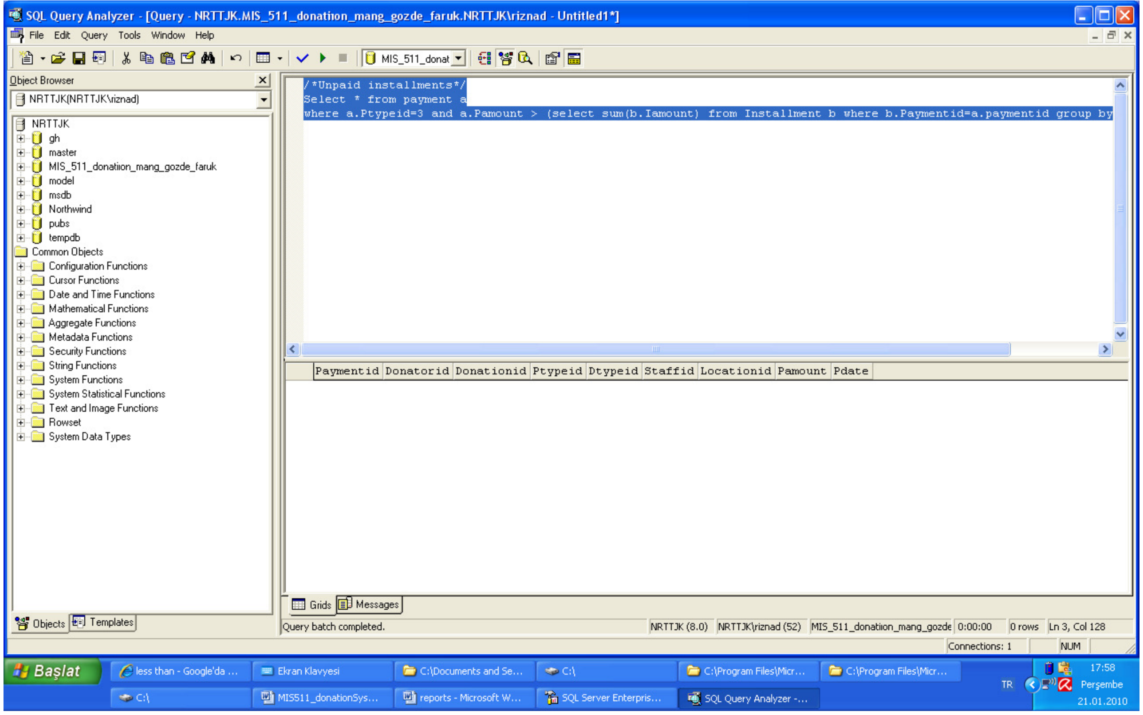
```
/*Total payment amount for each type of donation*/
select d.DTypeName,sum(p.Pamount)
from DONATIONTYPE d, payment p
where d.dtypeid=p.ptypeid
group by d.DtypeName;
```

```
   DTypeName  (No column name)
1  Estate     155000
2  Goods      67500
3  Money      377780
```

Query batch completed.

```
/*SP updates honurid by taking donatorid*/


CREATE PROCEDURE updatehonour  (@p_donatorid int )

 AS

Declare @Totaldonation as Int
Declare @v_honourid as Int


Select @Totaldonation=sum(Pamount) from Payment where donatorid=@p_donatorid

select @v_honourid=Honorid from Honor where  @Totaldonation between Minlimit and
Maxlimit

update donator set honorid=@v_honourid where Donatorid=@p_donatorid

Return
GO
```

```
/*Unpaid installments*/
Select * from payment a
where a.Ptypeid=3 and a.Pamount > (select sum(b.Iamount) from Installment b where
b.Paymentid=a.paymentid group by b.paymentid)
```



SQL Query Analyzer - [Query - NRTTJK.MIS_511_donatiion_mang_gozde_faruk.NRTTJK\riznad - Untitled1*]

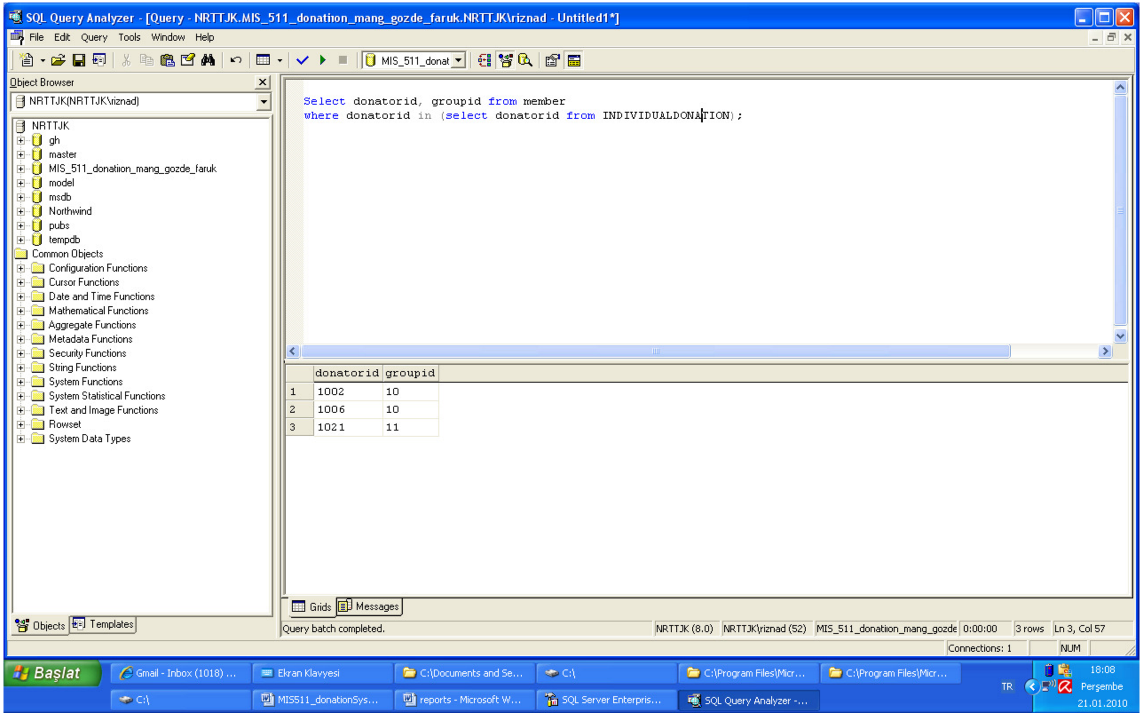
```
/*Unpaid installments*/
Select * from payment a
where a.Ptypeid=3 and a.Pamount > (select sum(b.Iamount) from Installment b where b.Paymentid=a.paymentid group by
```

```
Paymentid Donatorid Donationid Ptypeid Dtypeid Staffid Locationid Pamount Pdate
```

Query batch completed. | NRTTJK (8.0) | NRTTJK\riznad (52) | MIS_511_donatiion_mang_gozde | 0:00:00 | 0 rows | Ln 3, Col 128

Connections: 1 | NUM

/*Donators which make both individual and group payment*/
Select donatorid, groupid from member
where donatorid in (select donatorid from INDIVIDUALDONATION);



```
Select donatorid, groupid from member
where donatorid in (select donatorid from INDIVIDUALDONATION);
```

	donatorid	groupid
1	1002	10
2	1006	10
3	1021	11

Query batch completed.