**CSE3055 – DATABASE SYSTEM**

Buse Batman – 150117011

Fatmanur Özdemir – 150116046

**CEMETARY SALES COMPANY**

Subject of the Project:

* In this project we are designing a cemetary sales company in Istanbul. People can buy graves before or after death. And they can buy personal or family graves. There are certain number of graves in a cemetary
* The company provides also funeral transport services for the dead customer.
* With additional cost, company also offers grave care service (cleaning, watering and flower planting at desired periods).
* The database contains information about graveyards, graves (family or singular), person [employee (office worker, driver or cleaner), customer (dead or alive)], grave care service, material (cleaning or planting), supplier and cars (staff car or hearse).

Data and Requirement Analysis:

* PERSON\_T has 4 attributes:

PersonID int, FirstName varchar(20), LastName varchar(20) and gender varchar(1). [f for female, m for male]. And the personid is the PRIMARY KEY.

* PHONE\_NUMBER\_T has 2 attributes:

PersonId (primary key of phone\_number\_t, foreign key from person\_t), and PhoneNumber varchar(11) . Phone Numbers are unique. And a person may have more than one phone number.

* EMPLOYEE\_T has 6 attributes:

EmployeeID (primary key of employee\_t, foreign key from person\_t), City varchar(10), District varchar(10), PostalCode int, BloodType varchar(3), SocialSecurityNumber varchar(11).

* OFFICE\_WORKER\_T has 4 attributes:

OfficeWorkerID int (primary key of office\_worker\_t, foreign key from person\_t), MonthlySalary int, Department varchar(20), Email varchar(25).

* CLEANER\_T has 3 attributes:

CleanerID int (primary key of cleaner\_t, foreign key from person\_t), Earnings int, CleanedGraveID int. So the clenaer earns for every service she/he gives to the grave.

* DRIVER\_T has 3 attributes:

DriverID int (primary key of driver\_t, foreign key from person\_t), Earnings int default 100, DrivedCustomer int. So the driver earns for every funeral drive.

* GRAVEYARD\_T has 6 attributes:

GraveyardID int (primary key), GraveyardName varchar(30), Town varchar(30), District varchar(30) , FamilyGraveCapacity int, SingularGraveCapacity int.

* GRAVE\_CARE\_SERVICE\_T has 3 attributes:

ServiceID int (primary key), Period int, Price int. These services consist all of the cleaning, watering and planting services (no distinction). And the period is the repetation of the service per given period month. If the period is 3, that means the service is given to the grave for every 3 months.

* GRAVE\_T has 4 attributes:

GraveID int (primary key), GraveyardID int (foreign key from graveyard\_t), ServiceID int(foreign key from grave\_care\_service\_t), Price int.

* SINGULAR\_ GRAVE\_T has 1 attribute:

SingularGraveID int (primary key of grave\_t, foreign key from singular\_grave\_t)

* FAMILY\_ GRAVE\_T has 1 attribute:

FamilyGraveID int (primary key of grave\_t, foreign key from family\_grave\_t)

* CUSTOMER\_T has 3 attributes:

CustomerID int (primary key of customer\_t, foreign key from person\_t), GraveID int (foreign key from grave\_t), BirthDate date.

After creating CUSTOMER\_T, we added 3 attributes realted to address: City varchar(10), District varchar(10), PostalCode int. Then we updated the values of customers according to that change.

* DEAD\_CUSTOMER\_T has 2 attributes:

DeadCustomerID int (primary key of dead\_customer\_t, foreign key from customer\_t), DeathDate (date).

* ALIVE\_CUSTOMER\_T has 2 attributes:

AliveCustomerID int (primary key of alive\_customer\_t.

* CAR\_T has 6 attributes:

CarID int (primary key of dead\_customer\_t, foreign key from customer\_t), LicensePlate varchar(10), Model varchar(5), Brand varchar(10), Color varchar(10), InspectionDate date

* STAFF\_CAR\_T has 2 attributes:

StaffCarID int (primary key of staff\_car\_t, foreign key from car\_t), OwnerID int.

* HEARSE\_T has 3 attributes:

HearseID int (primary key of heares\_t, foreign key from car\_t), DriverID int, CarriedCustomerID int.

* MATERIAL\_T has 1 attribute:

MaterialID int (primary key of material\_t), MaterialName varchar(20).

* CLEANING\_MATERIAL\_T has 1 attribute:

CleaningMaterialID int (primary key of cleaning\_material\_t, foreign key from material\_t).

* PLANTING\_MATERIAL\_T has 1 attribute:

PlantingMaterialID int (primary key of planting\_material\_t, foreign key from material\_t).

* SUPPLIER\_T has 4 attributes:

SupplierID int (primary key of Supplier\_t), SupplierName varchar(20), CompanyName varchar(30), ContactNumber varchar(11),

* SUPPLY\_RELATION\_T has 5 attributes:

MaterialID int (primary key of supply\_relation\_t, foreign key of material\_t), SupplierID int (primary key of supply\_relation\_t, foreign key of supply\_t), SupplyDate date, SupplyPrice int, SupplyAmount int.

[got\_rise\_workers] view shows that the workers which got rise after working more than 2 years at the company.

* After creating office\_worker\_t, we increased monthly salaries %15, if office workers has worked in the company more than 2 years. We showed that by using [got\_rise\_worker] view.
* After creating cleaner\_t, we created a trigger as increase\_earning\_cleaners which incerases earnings of cleaner everytime he cleaned a new grave.
* After creating driver\_t, we created a trigger as increase\_earning\_drivers which incerases earnings of driver everytime he drived another dead customer.
* We created a view as [owned\_cars], which shows staff cars that have owners.
* We created procedures as cheap\_grave. It detects graves which are cheaper than the value in exec function.
* We created a view as [owned\_expens\_grave], which shows people who bought grave with higher than 2500 tl price.
* We created index as car\_sort which sorts car first based on brand then based on licenseplate.
* We used group by to show dead customers who had died 2 years ago and earlier.