MARMARA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

CSE3055 PROJECT REPORT

CONSTRUCTION COMPANY DATABASE

PREPARED BY:

Ahmet Can Bağırgan 150119510

Barış Hazar 150118019

Berkay Ağar 150119804

Hasancan Özen 150118882

Description

In our project, we designed the database of a construction company. As a result of our discussions with the company, we have decided which data we will keep. After completing the conceptual and logical designs, we transferred these designs to SQL Server. Afterwardsi we worked on the effective processing and use of these data.

Data and Requirement Analysis For Database

We need to design a database for a construction company which should have the data for all kinds of employees (White collar, Office worker, technician, worker). We need to also store the data about building sites (their buyers, their suppliers, workers that work in those building sites etc.). Also we need data about hired services that the company hires for offices.

Scope of the Project

The scope of this Project includes processes about Employees, BuildingSites and Buyers. But doesn't include processes related with financial transactions of the company with banks and land purchase data for building sites.

Business Process

There are 4 types of employees in the company. These are white collars, office workers, technicians and workers. White collars and office workers work in companies offices. Workers work building sites. Technicians work both offices and building sites. Manager-employee, supervisor-employee relationships abound among employees.(Shown in the EER diagram) White collars, office workers and technicians have salaries. However, workers are paid hourly. However, workers are paid hourly.

There are other companies that the company receives short-term or long-term services.(Advocacy, accounting…) There are also supplier companies from which the company buys materials. The company buys materials from these suppliers and uses these materials on their building sites.

The company also stores the data of customers who have bought apartments before.

Tables

- BUILDING | buildingID, buildingSiteCode, numberOfApartments
- Stores buildings informations that built in building sites
- int, nvarchar(50), tinyint
- buildingID(PK), buildingSiteCode(FK)

- BUILDINGSITE | buildingsiteCode, location, startDate, endDate, managerID, technicianID
- Stores building sites informations
- nvarchar(100), date, date, int, int
- buildingSiteCode(PK), managerID(FK), technicianID(FK)

- BUYER | TCKN, fName, lName, phoneNumber, email, city, Street
- Stores buyers informations
- Varchar(11), nvarchar(50), nvarchar(50), varchar(11), nvarchar(100), nvarchar(30), nvarchar(50)
- TCKN(PK)
- City column of buyer table has the default value 'İstanbul'. phoneNumber column of the table city is unique.

- COMPANY | companyName, location
- Stores supplier companies informations
- Nvarchar(100), nvarchar(100)
- companyName(PK)

- CONTRACT | contractID, buildingID, TCKN, apartmentNo, buyPrice, date
- Stores contracts signs between our company and buyers
- int, int, varchar(11), tinyint, decimal(20,2), date
- contractID(PK), buildingID(FK), TCKN(FK)

- DELAY | delayID, buildingSiteCode, delayReason, delayTime
- Stores delay informations that occured in building sites
- int, nvarchar(50), nvarchar(100), int
- delayID(PK), buildingSiteCode(PK, FK)

- EMPLOYEE | empID, fName, lName, birthDate, age, gender, startDate, email, employeeType, managerID
- Stores datas of all employees
- int, nvarchar(50), nvarchar(50), date, int, char(1), date, nvarchar(100), varchar(50), int
- empID(PK), managerID(FK)
- age is a Computed value. Computed from birthDate. Also, gender has a check constraint that checks whether the given gender is 'M' or 'F'.
- When a new employee inserts or deletes to the table, our trigger fires and add this process to log table as a log message
- We created an index named 'employee_index' which uses fName, lName, empId.

- HIRED SERVICES | companyName, givenService, startDate, rentalType
- Stores informations of hired service companies
- Nvarchar(100), nvarchar(50), date, varchar(20)
- companyName(PK)

- INVOICE | invoiceID, buildingSiteCode, companyName, materialName, materialAmount, invoiceUnit, invoiceDate, invoicePrice
- Stores invoices issued by supplier companies
- int, nvarchar(50), nvarchar(100), nvarchar(50), decimal(7,2), varchar(15), datetime, decimal(12,2)
- invoiceID(PK), buildingSiteCode(FK), companyName(FK)

- LOGTABLE | LogID, LogMessage
- Stores log information of system
- int, nvarchar(50)
- LogID(PK)

- LONGTERM | companyName, monthlyWage
- Stores informations of long-term hired companies
- nvarchar(100), decimal(10,2)
- companyName(PK, FK)

- LONGTERM_OFFICE | companyName, officeID
- Stores data of long-term hired companies working in our offices
- nvarchar(100), tinyint
- companyName(PK, FK), officeID(PK, FK)

- OFFICE | officeID, location, capacity
- Stores informations of our offices
- tinyint, nvarchar(100), smallint
- officeID(PK)

- OFFICEWORKER | empID, salary, title
- Stores datas of office workers
- int, decimal(9,2), nvarchar(30)
- empID(PK, FK)

- OFFICEWORKER_OFFICE | officeID, officeWokerID
- Stores informations about office workers who work on particular offices
- tinyint, int
- officeID(PK, FK), officeWorkerID(PK, FK)
- SHORTTERM | companyName, leasingTerm
- Stores datas of short-term hired companies
- nvarchar(100), smallint
- companyName(PK, FK)

- SHORTTERM_OFFICE | companyName, officeID, price
- Stores informations of short-term hired companies that work in our offices
- nvarchar(100), tinyint, decimal(10,2)

- TECHNICIAN | empID, degree, yearsOfExperince, technicianType, salary
- Stores data about employee type technician
- int, nvarchar(100), tinyint, nvarchar(30), decimal(9,2)
- empID(PK, FK)

- WHITECOLLAR | empID, title, yearsOfExperience, salary, supervisorID
- Stores datas of white collars
- int, nvarchar(30), tinyint, decimal(9,2), int
- empID(PK, FK), supervisorID(FK)

- WHITECOLLAR_OFFICE | officeID, empID
- Stores datas of white collars who work on particular office
- tinyint, int
- empID(PK, FK), officeID(PK, FK)

- WHITECOLLARDEGREE | empID, degree
- Stores white collars degrees
- int, nvarchar(100)
- empID(PK, FK), degree(FK)

- WORKER | empID, workerType, workerExperince, supervisorID, departmentID
- Stores datas of workers
- int, nvarchar(30), tinyint, int, tinyint
- empID(PK, FK), supervisorID(FK), departmentID(FK)

- WORKER_BUILDINGSITE | workerID, buildingSiteCode, workingHour, hourlyFee
- Stores informations about workers who work on particular building site
- int, nvarchar(50), smallint, decimal(6,2)
- workerID(PK, FK), buildingSiteCode(PK, FK)

- WORKERDEPARTMENT | departmentID, departmentName, departmentType, managerID
- Stores datas about worker departments
- tinyint, nvarchar(50), nvarchar(30), int
- departmentID(PK), managerID(FK)

Triggers

Before

```
--If an employee data is added or removed, it writes the transaction type,
--empID and employee name in the log message of the log table
Create Trigger trg_Log
  on Employee
  after insert
As
Begin
    IF exists (Select * From inserted)
    Begin
    Insert Into LogTable(LogMessage)
    Select('New employee hired : ' + CONVERT(nvarchar(50), insertedEmp.empID) + ' ' + insertedEmp.EmpName)
    From  (Select e.empID, e.fName + ' ' + e.lName EmpName
           From Employee e, inserted i
           Where e.empID = i.empID) insertedEmp
    End

    IF exists (Select * From deleted)
    Begin
    Insert Into LogTable(LogMessage)
    Select('An employee fired : ' + CONVERT(nvarchar(50), deletedEmp.empID) + ' ' + deletedEmp.EmpName)
    From  (Select e.empID, e.fName + ' ' + e.lName EmpName
           From Employee e, deleted d
           Where e.empID = d.empID) deletedEmp
    End
End

Select * From LogTable
```

121 %

Results | Messages

LogID	LogMessage

Query executed successfully. DESKTOP-

After

```sql
--If an employee data is added or removed, it writes the transaction type,
--empID and employee name in the log message of the log table
Create Trigger trg_Log
  on Employee
  after insert
As
Begin
    IF exists (Select * From inserted)
    Begin
    Insert Into LogTable(LogMessage)
    Select('New employee hired : ' + CONVERT(nvarchar(50), insertedEmp.empID) + ' ' + insertedEmp.EmpName)
    From  (Select e.empID, e.fName + ' ' + e.lName EmpName
           From Employee e, inserted i
           Where e.empID = i.empID) insertedEmp
    End

    IF exists (Select * From deleted)
    Begin
    Insert Into LogTable(LogMessage)
    Select('An employee fired : ' + CONVERT(nvarchar(50), deletedEmp.empID) + ' ' + deletedEmp.EmpName)
    From  (Select e.empID, e.fName + ' ' + e.lName EmpName
           From Employee e, deleted d
           Where e.empID = d.empID) deletedEmp
    End
End

Insert Into Employee(empID, fName, lName, birthDate, gender, startDate, employeeType, managerID)
Values(192, 'Mustafa', 'Ağaoğlu', '1985-07-19' , 'M', '2020-12-21', 'WhiteCollar', 101)
Select * From LogTable
-------------------------------------------------------------
```

110 %

Results | Messages

	LogID	LogMessage
1	4	New employee hired : 192 Mustafa Ağaoğlu

Query executed successfully.

Stored Procedures

1)

```sql
--Stored Procedures

--1) Retrieves total material shopping fee with particular company up to now, groups by materials
Create Procedure sp_MaterialPrice
@companyName nvarchar(50)
As
Begin

    Select i.companyName, i.materialName, sum(i.invoicePrice) TotalPaid
    From Invoice i
    Where i.companyName=@companyName
    Group By i.companyName, i.materialName

End
--------------------------------------
exec sp_MaterialPrice 'Yurdagül Hırdavat'

Create Procedure sp_depInfo
@departmentName nvarchar(50),
@buildingSiteCode nvarchar(50)
As
```

146 %

Results | Messages

	companyName	materialName	TotalPaid
1	Yurdagül Hırdavat	Cement	1785000.00
2	Yurdagül Hırdavat	Combi Boiler	1785000.00
3	Yurdagül Hırdavat	Electrical Materials	89250.00
4	Yurdagül Hırdavat	Paint	71400.00
5	Yurdagül Hırdavat	Plaster	2677500.00

2)

```sql
--2) Takes departmentName and buildingSiteCode as parameters and retrieves
--employee's departmentID, employeeID, number of workers who works that building site,
--manager of that building site, and the total payment of that departments workers on that building site
Create Procedure sp_depInfo
@departmentName nvarchar(50),
@buildingSiteCode nvarchar(50)
As
Begin

Select wd.departmentID, wd.departmentName, wbs.buildingSiteCode, count(w.empID) NoOfWorker, e.fName + ' ' + e.lName ManagerName, e.empID, sum(workersalary.WorkerSalary)TotalPay
From Worker w, WorkerDepartment wd, Employee e, Worker_BuildingSite wbs, (Select sum(wbs.hourlyFee*workingHour*26)WorkerSalary
                                                                          From Worker_BuildingSite wbs
                                                                                 inner join Worker w on wbs.workerID=w.empID
                                                                                 inner join WorkerDepartment wd on wd.departmentID=w.departmentID
                                                                          Where wd.departmentName=@departmentName and wbs.buildingSiteCode=@buildingSiteCode) workersalary
Where w.departmentID=wd.departmentID and wd.managerID=e.empID and wd.departmentName=@departmentName and wbs.workerID=w.empID and wbs.buildingSiteCode=@buildingSiteCode
Group By wd.departmentID, wd.departmentName, e.fName + ' ' + e.lName, e.empID, wbs.buildingSiteCode

End

exec sp_depInfo 'Blacksmith', 'IST1002'
----------------------------------------
```

133 %

Results | Messages

	departmentID	departmentName	buildingSiteCode	NoOfWorker	ManagerName	empID	TotalPay
1	1	Blacksmith	IST1002	1	Ali Bayram	401	5850.00

3)

```sql
----------------------------------------
--3) Retrieves the addresses of buyers who buy apartments under a certain price
Create Procedure sp_BuyerAddress
@price decimal(20,2)
As
Begin
    Select b.city + ' ' + b.street Address
    From Contract c inner join Buyer b on c.TCKN=b.TCKN
    Where c.buyPrice <= @price
    Order By b.city asc
End
exec sp_BuyerAddress 500000
----------------------------------------
```

133 %

Results | Messages

	Address
1	Adana Eren Sk.
2	Adana Ali Fuat Başgil Cd.
3	Adana Çetin Sk.
4	Adıyaman Arslan Sk.
5	Adıyaman Arslan Sk.
6	Adıyaman Vural Sk.
7	Ankara Akdeniz Caddesi
8	Ankara Küçükyali Cd.
9	Ankara Küçükyali Cd.
10	Ankara Yamanevler Cd.
11	Ankara Zümrütevler Cd.
12	Ankara Çerkez Cd.
13	Ankara Atatürk Cd.
14	Ankara Sevgi Cd.
15	Ankara Pazar Caddesi
16	Ankara Zümrütevler Cd.
17	Ankara Kuru Sk.
18	Ankara Yancar Sk.
19	Ankara Sevgi Cd.
20	Bolu Selamsız Cd.

4)

Before

```sql
Create Procedure sp_UpdateWorkerHourlyFee
@delayReason nvarchar(100)
As
Begin

    If @delayReason='Walkout'
    Begin
        Update Worker_BuildingSite
        Set hourlyFee=hourlyFee*1.10
        From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode
        Where d.delayReason=@delayReason
    End

    If @delayReason='Salary Issue'
    Begin
        Update Worker_BuildingSite
        Set hourlyFee=hourlyFee*1.05
        From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode
        Where d.delayReason=@delayReason
    End

End
Select wbs.workerID, wbs.hourlyFee From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode Where d.delayReason='Walkout'
```

133 %

Results | Messages

	workerID	hourlyFee
1	401	25.00
2	417	25.00
3	419	25.00
4	422	25.00
5	425	13.00
6	433	8.00
7	440	13.00
8	445	13.00
9	448	25.00
10	453	25.00
11	463	13.00

After

```sql
Create Procedure sp_UpdateWorkerHourlyFee
@delayReason nvarchar(100)
As
Begin

    If @delayReason='Walkout'
    Begin
        Update Worker_BuildingSite
        Set hourlyFee=hourlyFee*1.10
        From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode
        Where d.delayReason=@delayReason
    End

    If @delayReason='Salary Issue'
    Begin
        Update Worker_BuildingSite
        Set hourlyFee=hourlyFee*1.05
        From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode
        Where d.delayReason=@delayReason
    End

End
exec sp_UpdateWorkerHourlyFee 'Walkout'
Select wbs.workerID, wbs.hourlyFee From Worker_BuildingSite wbs inner join Delay d on wbs.buildingSiteCode=d.buildingSiteCode Where d.delayReason='Walkout'
```

133 %

Results | Messages

	workerID	hourlyFee
1	401	27.50
2	417	27.50
3	419	27.50
4	422	27.50
5	425	14.30
6	433	8.80
7	440	14.30
8	445	14.30
9	448	27.50
10	453	27.50

Query executed successfully. DESKTOP-B6NUO9M (15.0 RTM) DESKTOP-B6NUO9M\canba ... Construction_

5)

Before

```sql
-----------------------------------------
--5)In the selected department, it increases the salary of the employees within a certain age range by the desired amount.
Create Procedure sp_UpdateSalaryByAge
  @empType varchar(50),
  @age1 int,
  @age2 int,
  @raisePercent float
 As
 Begin

    If(@empType='OfficeWorker')
    Begin
    Update OfficeWorker
    Set Salary += ow.salary * @raisePercent / 100
    From OfficeWorker ow inner join Employee e on ow.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='Technician')
    Begin
    Update Technician
    Set Salary += t.salary * @raisePercent / 100
    From Technician t inner join Employee e on t.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='WhiteCollar')
    Begin
    Update WhiteCollar
    Set Salary += wc.salary * @raisePercent / 100
    From WhiteCollar wc inner join Employee e on wc.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='Worker')
    Begin
    Update Worker_BuildingSite
    Set hourlyFee += wbs.hourlyFee * @raisePercent / 100
    From Worker_BuildingSite wbs inner join Employee e on wbs.workerID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End
 End
 Select ow.empID, ow.salary From OfficeWorker ow inner join Employee e on ow.empID=e.empID Where e.age>=36 and e.age<=41
 ---------------------------------------------------
Create Procedure sp_ExtraPaymentDueToDelay
```

83 %

Results | Messages

	empID	salary
1	301	5000.00
2	303	4700.00
3	304	4500.00

After

```sql
--5)In the selected department, it increases the salary of the employees within a certain age range by the desired amount.
Create Procedure sp_UpdateSalaryByAge
  @empType varchar(50),
  @age1 int,
  @age2 int,
  @raisePercent float
 As
 Begin

    If(@empType='OfficeWorker')
    Begin
    Update OfficeWorker
    Set Salary += ow.salary * @raisePercent / 100
    From OfficeWorker ow inner join Employee e on ow.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='Technician')
    Begin
    Update Technician
    Set Salary += t.salary * @raisePercent / 100
    From Technician t inner join Employee e on t.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='WhiteCollar')
    Begin
    Update WhiteCollar
    Set Salary += wc.salary * @raisePercent / 100
    From WhiteCollar wc inner join Employee e on wc.empID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End

    If(@empType='Worker')
    Begin
    Update Worker_BuildingSite
    Set hourlyFee += wbs.hourlyFee * @raisePercent / 100
    From Worker_BuildingSite wbs inner join Employee e on wbs.workerID=e.empID
    Where e.age >= @age1 and e.age <= @age2
    End
 End
exec sp_UpdateSalaryByAge 'OfficeWorker', 36, 41, 3
Select ow.empID, ow.salary From OfficeWorker ow inner join Employee e on ow.empID=e.empID Where e.age>=36 and e.age<=41
```

83 %

Results | Messages

	empID	salary
1	301	5150.00
2	303	4841.00
3	304	4635.00

6)

```sql
--------------------------------------------------
--6) Calculates the total overpaid salary due to the extension of the building site duration
-- at the building sites with delay.
Create Procedure sp_ExtraPaymentDueToDelay
  @buildingSiteCode nvarchar(50)
 As
 Begin
    Select sum(wbs.hourlyFee*wbs.workingHour*d.delayTime*26) as ExtraPayment
    From BuildingSite bs inner join Delay d on bs.buildingSiteCode=d.buildingSiteCode
         inner join Worker_BuildingSite wbs on bs.buildingSiteCode=wbs.buildingSiteCode
    Where bs.buildingSiteCode=@buildingSiteCode
 End
 exec sp_ExtraPaymentDueToDelay 'IST1003'
--------------------------------------------------
```

146 %

Results | Messages

	ExtraPayment
1	32500.00

Query executed successfully.

7)

```sql
--7)Takes a location as parameter and returns the number of
--apartments sold which resides in that location
Create Procedure sp_getNumOfSoldApsbyLocation
(@location nvarchar(100))
as
begin
    Select c.buildingID, bs.location, COUNT(*) #ofSoldApartments
    From Contract c inner join Buyer b on c.TCKN = b.TCKN
    inner join Building bl on bl.buildingID = c.buildingID
    inner join BuildingSite bs on bs.buildingSiteCode = bl.buildingSiteCode
    Where bs.location like '%' + @location + '%'
    Group By c.buildingID, bs.location
end
exec sp_getNumOfSoldApsbyLocation 'ankara'
--Queries
```

161 %

Results | Messages

	buildingID	location	#ofSoldApartments
1	4	Ankara/Mamak	5
2	9	Ankara/Çankaya	1
3	26	Ankara/Keçiören	16

8)

```sql
-----------------------------------------------------------------
--8)Retrieves Employee Id, name, surname and email of the workers
--which works on building sites that has less than the delay passed
--as the argument
Create Procedure sp_getWorkersBuildingSiteByDelay
(@delay int)
As
Begin
    Select distinct w.empID, e.fName + ' ' +  e.lName FullName, e.email
    From Worker w
    inner join Worker_BuildingSite wb on w.empID = wb.workerID
    inner join BuildingSite bs on wb.buildingSiteCode = bs.buildingSiteCode
    inner join Employee e on w.empID = e.empID
    Where bs.buildingSiteCode in (Select bs.buildingSiteCode
                                  From BuildingSite bs
                                  inner join Delay d on bs.buildingSiteCode = d.buildingSiteCode
                                  Group by bs.buildingSiteCode
                                  Having SUM(d.delayTime) < @delay)
End
exec sp_getWorkersBuildingSiteByDelay 3
```

121 %

Results | Messages

	empID	FullName	email
1	402	Nurettin Dönmez	donmeznuri@gmail.com
2	405	Serhat Akıncı	akinciserhat@gmail.com
3	408	Ramazan Demir	NULL
4	410	Ramazan Bayram	ramazanbayrami@gmail.com
5	416	Ertuğrul Öztenekeci	oztenekecierto@gmail.com
6	418	Mehmet Ali Yılmaz	mali@gmail.com
7	420	Emin Genç	NULL
8	422	Yunus Emre Gök	yunus_emre_gok@gmail.com
9	423	Yüksel Bayrak	yukselbayrak@hotmail.com
10	426	İbrahim Eskimez	eskimez@gmaik.com
11	430	Mehmet Demirtaş	mehmetdemirtas@hotmail.com
12	431	Egemen Kutay	kutay_egemen@gmail.com
13	434	Baran Delil	delil_baran@gmail.com
14	439	Mert Yılmaz	yilmazmert@gmail.com

Query executed successfully. DESKTOP-B6NUO9M (15.0 RTM) DESKTOP-B6NUO9M\canba ... C

9)

```sql
--------------------------------------------------------------------------------
--9)Retrieve full name of buyers that live in the same location with building site location
Create Proc sp_BuyersThatLivesInSameLoc
(@location varchar(50))
As
Begin

Select b.fName + ' ' + b.lName as FullName, bs.buildingSiteCode, bs.location
From Buyer b
inner join Contract c ON b.TCKN = c.TCKN
Inner join Building bu ON c.buildingID = bu.buildingID
Inner join BuildingSite bs ON bu.buildingSiteCode = bs.buildingSiteCode
Where bs.location like  '%' + @location + '%'

End

exec sp_BuyersThatLivesInSameLoc 'ankara'
--------------------------------------------------------------------------------

--Queries
```

121 %

Results | Messages

	FullName	buildingSiteCode	location
1	Sedef Kilaç	ANK0001	Ankara/Mamak
2	Mehmet Özgür Yadigaroğlu	ANK0001	Ankara/Mamak
3	Yusuf Köker	ANK0001	Ankara/Mamak
4	Ismail Evren Yiğit Kuplay	ANK0001	Ankara/Mamak
5	Volkan Nazli	ANK0001	Ankara/Mamak
6	Zümrüt Ela Kumral	ANK0002	Ankara/Çankaya
7	Kutlu Mardin	ANK0004	Ankara/Keçiören
8	Burcu Parlatan	ANK0004	Ankara/Keçiören
9	Ilknur Tuğcugil	ANK0004	Ankara/Keçiören
10	Yener Oral	ANK0004	Ankara/Keçiören
11	Mehri Yarar	ANK0004	Ankara/Keçiören
12	Hakan Sönmez	ANK0004	Ankara/Keçiören
13	Göker Görgülü	ANK0004	Ankara/Keçiören
14	Uğur Egemen	ANK0004	Ankara/Keçiören

Query executed successfully. DESKTOP-B6NUO9M (15.0 RTM) | DESKTOP-B6NUO9M\canba ...

10)

Before

```sql
--10) Update employee
Create Proc sp_UpdateEmployee(

    @empID int,
    @fName nvarchar(50),
    @lName nvarchar(50),
    @birthDate date,
    @gender char(1),
    @startDate date,
    @email nvarchar(100),
    @employeeType varchar(50),
    @managerID int
)
As
Begin

    Update Employee
    Set fName = @fName, lName = @lName, birthDate = @birthDate, gender = @gender,
    startDate = @startDate, email = @email, employeeType = @employeeType
    Where empID = @empID and managerID = @managerID

End

exec sp_UpdateEmployee 433, 'Murat', 'Çelik', '1990-10-19', 'M', '2017-12-19', 'celik_murat@outlook.com', 'Worker',101
Select * From Employee e Where e.empID=433
```

121 %

Results | Messages

	empID	fName	lName	birthDate	age	gender	startDate	email	employeeType	managerID
1	433	Murat	Çelik	1990-10-19	31	M	2017-10-19	celik_murat@outlook.com	Worker	101

After

```sql
--------------------------------------------------------------------------
--10) Update employee
Create Proc sp_UpdateEmployee(

    @empID int,
    @fName nvarchar(50),
    @lName nvarchar(50),
    @birthDate date,
    @gender char(1),
    @startDate date,
    @email nvarchar(100),
    @employeeType varchar(50),
    @managerID int
)
As
Begin

    Update Employee
    Set fName = @fName, lName = @lName, birthDate = @birthDate, gender = @gender,
    startDate = @startDate, email = @email, employeeType = @employeeType
    Where empID = @empID and managerID = @managerID

End

exec sp_UpdateEmployee 433, 'Murat', 'Çelik', '1990-10-19', 'F', '2017-12-19', 'mcelik@outlook.com', 'Worker',101
Select * From Employee e Where e.empID=433
```

121 %

Results | Messages

	empID	fName	lName	birthDate	age	gender	startDate	email	employeeType	managerID
1	433	Murat	Çelik	1990-10-19	31	F	2017-12-19	mcelik@outlook.com	Worker	101

Queries

1)

```sql
--1) Retrieves the empID, first name, last name, salary, and the number of sells of manager who are the top 3 best seller
Select Top 3 e.empID ,e.fName, e.lName, wc.salary, count(bs.managerID)topSeller
From BuildingSite bs inner join Employee e on bs.managerID = e.empID
     inner join WhiteCollar wc on e.empID = wc.empID
Group by e.empID, e.fName, e.lName, wc.salary
Order By topSeller desc
---------------------------------------------
Select e.officeID, (e.shortTermPrice + e.longtermprice + e.whitecollarprice + e.officeworkerprice) totalMonthlyPrice
```

146 %

Results | Messages

	empID	fName	lName	salary	topSeller
1	105	Recep	Erdoğan	15650.00	10
2	103	Barış	Hazmaz	8400.00	9
3	111	Nevin	Mutlu	4836.89	4

2)

```sql
--3) Retrieves building site code, total earned money, building site cost, and percentage of profit of building sites that completed
Select f.buildingSiteCode ,f.earnedMoney, f.buildingSiteCost ,(f.earnedMoney - f.buildingSiteCost) / f.buildingSiteCost * 100 '%Profit'
From(Select bs.buildingSiteCode, SUM(i.invoicePrice) buildingSiteCost, SUM(c.buyPrice) earnedMoney
From BuildingSite bs inner join Invoice i on bs.buildingSiteCode = i.buildingSiteCode
inner join Building b on bs.buildingSiteCode = b.buildingSiteCode
inner join Contract c on b.buildingID = c.buildingID
Group By bs.buildingSiteCode) f
---------------------------------------------
Select wd.departmentName, avg(e.age * 1.0) AvgAgeOfWorkers
From Worker w inner join Employee e On w.empID = e.empID
```

146 %

Results | Messages

	buildingSiteCode	earnedMoney	buildingSiteCost	%Profit
1	IST1008	113734000.00	34302400.00	231.562800
2	MAL0001	11400000.00	194900.00	5749.153400
3	CRM0001	182210000.00	62368000.00	192.153000
4	KUT0001	7505000.00	194900.00	3750.692600
5	IST2003	158574000.00	81858000.00	93.718300
6	ANK0004	156028000.00	49854400.00	212.716400
7	IST1007	74860000.00	10914400.00	585.882800
8	ANK0002	7600000.00	1556200.00	387.429400
9	IST2001	48260000.00	11694000.00	312.690200
10	ANK0001	45904000.00	11694000.00	292.543100
11	IST1004	9500000.00	1949000.00	387.429400
12	IST1002	65645000.00	14032800.00	367.796800
13	IST1001	75810000.00	31184000.00	143.105400
14	IST1003	63270000.00	28065600.00	125.436100
15	IST1006	87580000.00	37420800.00	134.067600
16	IST1011	134140000.00	56131200.00	138.975800
17	IST2002	70490000.00	18710400.00	276.742300
18	KUT0002	152760000.00	105246000.00	45.145600
19	IST1010	59660000.00	11694000.00	410.176100

Query executed successfully.

DESKTOP-B6NUO9M (15.0 RTM) DESKTOP-B6NUO9M\canba ...

3)

```sql
--------------------------------------------------
--4) Retrieves department name and average age of workers for every worker department
Select wd.departmentName, avg(e.age * 1.0) AvgAgeOfWorkers
From Worker w inner join Employee e On w.empID = e.empID
inner join WorkerDepartment wd On w.departmentID = wd.departmentID
Group By wd.departmentName
--------------------------------------------------
```

146 %

Results | Messages

	departmentName	AvgAgeOfWorkers
1	Blacksmith	40.571428
2	Ceramist	31.428571
3	Electrician	53.000000
4	Furnisher	32.909090
5	Glazier	33.600000
6	Moulder	40.750000
7	Painter	36.800000
8	Plasterer	37.800000
9	Plumber	41.000000
10	Sheathing	34.714285

4)

```sql
--5) Retrieves building site code, construction time and construction cost of every building site that completed
Select distinct bs.buildingSiteCode, Concat(Datediff(Day, bs.startDate, bs.endDate) , ' Day') as ConstructionTime, Sum(i.invoicePrice) ConstructionCost
From BuildingSite bs inner join Invoice i ON bs.buildingSiteCode = i.buildingSiteCode
Group by bs.buildingSiteCode, bs.startDate, bs.endDate
Having bs.endDate is not null
Order by ConstructionTime Desc
--------------------------------------------------
```

146 %

Results | Messages

	buildingSiteCode	ConstructionTime	ConstructionCost
1	IST1002	939 Day	2338800.00
2	ANK0001	775 Day	2338800.00
3	IST1007	713 Day	1559200.00
4	ANK0003	686 Day	5847000.00
5	IST1008	543 Day	3118400.00
6	CRM0001	460 Day	3898000.00
7	IST2001	440 Day	2923500.00
8	ANK0002	424 Day	1559200.00
9	MAL0001	414 Day	194900.00
10	IST2002	404 Day	3118400.00
11	KUT0001	394 Day	194900.00
12	IST1001	394 Day	3898000.00
13	IST1003	328 Day	4677600.00
14	KUT0002	322 Day	7016400.00
15	IST1004	307 Day	1949000.00
16	SVS0001	302 Day	2338800.00
17	IST1011	291 Day	4677600.00
18	IST1005	274 Day	1559200.00

5)

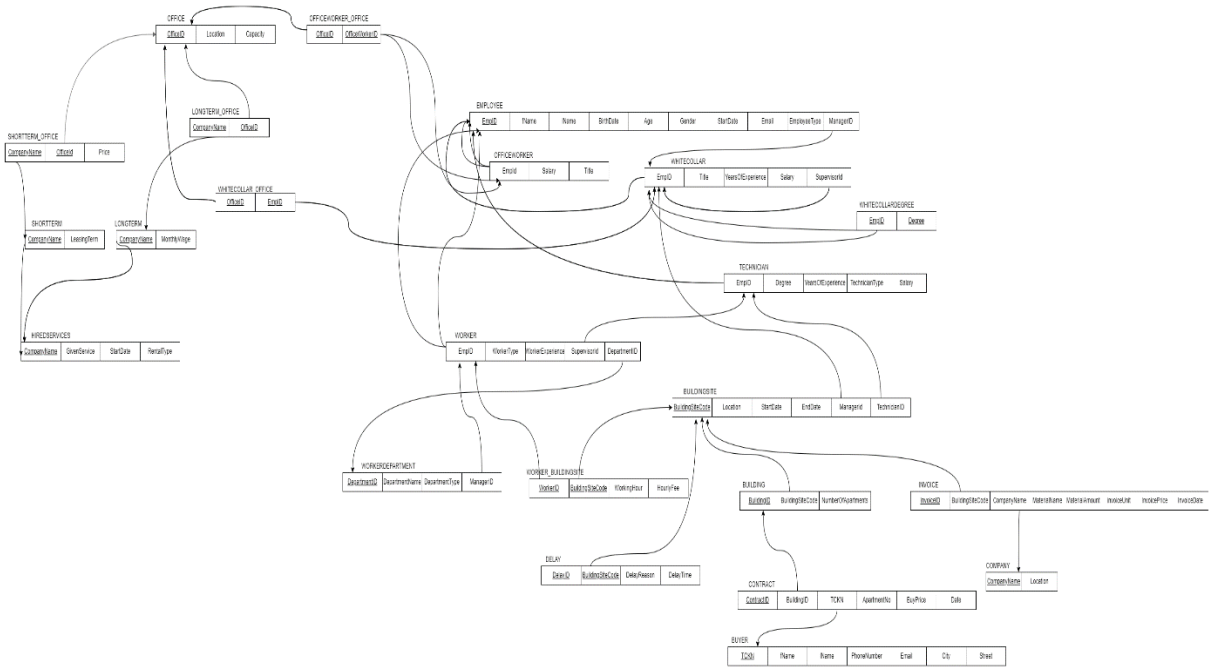
```sql
--6) Retrieve name, surname, salary, numberof degrees and supervisor name of all white collar employees which have at least 2 degrees
--by ordering according to degrees increasing and salaries decreasing
Select q.fullName, q.#ofDegrees, q.salary, m.fName + ' ' + m.lName SupervisorName
From (
    Select e.fName + ' ' + e.lName fullName,COUNT(*) #ofDegrees, w.salary,  w.supervisorID
    From WhiteCollar w
    inner join WhiteCollarDegree wd on w.empID = wd.empID
    inner join Employee e on e.empID = wd.empID
    Group By w.empID, e.fName, e.lName, w.supervisorID, w.salary
    Having COUNT(*) >= 2) q
inner join Employee m on q.supervisorID = m.empID
Order By q.#ofDegrees desc, q.salary desc
```

146 %

Results | Messages

	fullName	#ofDegrees	salary	SupervisorName
1	Barış Hazmaz	3	8400.00	Nuran Yılmaz
2	Mustafa Kol	3	7600.00	Nuran Yılmaz
3	Binali Albayrak	3	6500.00	Nuran Yılmaz
4	Recep Erdoğan	2	15650.00	Nuran Yılmaz
5	Sergen Akbaba	2	11858.41	Binali Albayrak

Mapping Diagram



Note: We have this mapping file as pdf.