

```

--Να δημιουργηθεί μια function που θα έχει σαν παράμετρο τον κωδικό ενός χρήστη
--και θα υπολογίζει το σύνολο των ωρών που εργάστηκε σε όλα το project
create function EmpHours (@id char(8)) returns decimal(3,1) as
begin
    declare @hr decimal(3,1)
    set @hr = (select isnull(sum(hours), 0) from works_on where emp_id = @id )
    return @hr
end
-----
declare @h decimal(3,1)
begin
    exec @h=emphours 'G276548'
print convert(char(10), @h)
end

```

--Να γράψετε ένα μπλοκ κώδικα που θα εμφανίζει όλους τους εργαζόμενους (κωδικό, πλήρες όνομα).
 -- Για κάθε εργαζόμενο θα χρησιμοποιεί την function EmpHours για να υπολογίσει και εμφανίσει τις ώρες που εργάζεται
 -- στα projects

```

declare c1 cursor for
    select emp_id, lastname + ' ' + firstname
    from employee
declare @id char(8), @fullname char(30), @hours decimal(3,1)

begin
    open c1
    fetch c1 into @id, @fullname
    while @@fetch_status = 0
    begin
        exec @hours = EmpHours @id
        print @id + ' ' + @fullname + ' ' + convert(char(10), @hours)
        fetch c1 into @id, @fullname
    end
    close c1
    deallocate c1
end

```

--Να δημιουργηθεί μια procedure ShowDept με παράμετρο τον κωδικό ενός τμήματος (dnumber)
 --και θα εμφανίζει το όνομα τμήματος, το πλήρες όνομα του μανατζερ και τα
 --ονόματα των project στα οποία συμμετέχει.

```

create procedure ShowDept @dn int as
declare @dnm varchar(15), @mname varchar(30)
declare pr cursor for
    select pname from project where dnumber=@dn
declare @pn varchar(30)

begin
    select @dnm=d.dname, @mname=e.lastname+' '+e.firstname
    from department d, employee e
    where d.dnumber=e.dnumber
    and d.dnumber = @dn
    print @dnm + ' ' + @mname
    print '-----'
    open pr
    fetch pr into @pn
    while @@fetch_status = 0
    begin
        print @pn
        fetch pr into @pn
    end
    close pr
    deallocate pr
end

exec ShowDept 6

```

```

--procedure EmpSalary
--Να υπολογιστεί το πλήθος των projects που συμμετέχει ο κάθε εργαζόμενος.
--Αν είναι πάνω από 2 τότε να αυξάνεται ο μισθός του κατά 2%
create procedure EmpSalary as
set nocount on
declare e1 cursor for
    select emp_id, count(*)
    from works_on
    group by emp_id
declare @id char(8), @p1 int
begin

    open e1
    FETCH e1 INTO @id, @p1
    WHILE @@FETCH_STATUS = 0 --έλεγχος για το τέλος του cursor
    Begin
        print @id + convert(varchar(10),@p1)
        if @p1>2
        begin
            update employee
            set salary=1.02*salary
            where emp_id=@id
            print '-----, έγινε αύξηση στον' + @id
        end
        FETCH e1 INTO @id, @p1
    End
    CLOSE e1
    DEALLOCATE e1
end

--exec EmpSalary

```

```

--Να γραφεί τμήμα κώδικα που θα εμφανίζει τον κωδικό και το ονομ/μο του supervisor
--καθώς και το πλήθος των υπαλλήλων που προϊστάται
declare e1 cursor for
    select e1.emp_id, e1.lastname+' ' +e1.firstname supname, count(*)
    from employee e1, employee e2
    where e1.emp_id = e2.supervisor_id
    group by e1.emp_id, e1.lastname+' ' +e1.firstname
declare @id char(8), @sn char(20), @pl int
begin
    open e1
    FETCH e1 INTO @id, @sn, @pl
    WHILE @@FETCH_STATUS = 0 --έλεγχος για το τέλος του cursor
    Begin
        print @id + ' ' + @sn + ' ' + convert(varchar(10), @pl)
        FETCH e1 INTO @id, @sn, @pl
    end
    close e1
    deallocate e1
end

```

Υπολογίζοντας τον μέσο μισθό όλων των εργαζομένων και όσο είναι κάτω από 900 να κάνετε συνολική αύξηση μισθών κατά την τιμή της μεταβλητής @extra_money. Η ενέργεια θα επαναλαμβάνεται όσο ο μέγιστος μισθός είναι πάνω από 3000.

```

Declare @extra_money money, @max_salary float
set @extra_money=100
WHILE (SELECT AVG(Salary) FROM employee) < 900
BEGIN
    UPDATE employee
    SET Salary = Salary + @extra_money
    set @max_salary = (SELECT MAX(salary) FROM employee)

```

```
IF (@max_salary) > 3000  
    BREAK  
ELSE  
    CONTINUE  
END
```