FUNCTII TABELARE

--CREATE TABLE NUME\_TABEL(LISTA\_PARAMETRI)

--RETURNS TABLE

--AS

--RETURN(SELECT...)

---------------------------------

--AFISARE ANG CU SALARIU >90000

CREATE FUNCTION PERS\_ANG()

RETURNS TABLE

AS

RETURN(SELECT A.\*,AA.FUNCTIE\_ID,AA.DEPARTAMENT\_ID,AA.SALARIU,AA.BONUS\_SALARIU,AA.PENALITATI\_SALARIU FROM ANGAJAT A INNER JOIN ANGAJARE AA ON A.

ANGAJAT\_ID=AA.ANGAJAT\_ID)

--APELAREA FUNCTIEI TABELARE

SELECT \* FROM DBO.PERS\_ANG() WHERE SALARIU>90000

---------------------------------

--SA SE DETERMINE PRIN UTILIZAFREA FUNCTIILOR TABELARE PRODUSUL CU CEL

--MAI MARE PRET

--SA SE DET PENTRU FIECARE PRODUCATOR NR DE PIESE VANDUTE

CREATE FUNCTION PRODUS\_FUNCTII()

RETURNS TABLE

AS

RETURN(SELECT PC.MODEL\_ID,P.TIP,PC.VITEZA,PC.RAM,PC.HDD,PC.CD,P.PRODUCATOR,PC.PRET FROM PC INNER JOIN PRODUSE P ON P.MODEL\_ID=PC.MODEL\_ID

UNION

SELECT L.MODEL\_ID,P.TIP,L.VITEZA,L.RAM,L.HDD,'',P.PRODUCATOR,L.PRET FROM LAPTOP L INNER JOIN PRODUSE P ON P.MODEL\_ID=L.MODEL\_ID

UNION

SELECT I.MODEL\_ID,P.TIP,0,0,0,'',P.PRODUCATOR,I.PRET FROM IMPRIMANTA I INNER JOIN PRODUSE P ON P.MODEL\_ID=I.MODEL\_ID)

SELECT \* FROM DBO.PRODUS\_FUNCTII()

WHERE PRET=(SELECT MAX(PRET) FROM DBO.PRODUS\_FUNCTII())

SELECT DISTINCT PRODUCATOR,COUNT(DISTINCT(MODEL\_ID)) AS NR\_UNICALE,COUNT(\*) AS CANTITATEA FROM DBO.PRODUS\_FUNCTII()

GROUP BY PRODUCATOR

SELECT DISTINCT PRODUCATOR,

(SELECT SUM(PRET) FROM DBO.PRODUS\_FUNCTII() PR

WHERE PR.PRODUCATOR=P.PRODUCATOR)AS SUMA\_PRET

FROM PRODUSE P

--SA SE GASEASCA ANGAJAT CU CEL MAI MARE SALARIU

CREATE FUNCTION ANGAJATII()

RETURNS TABLE

AS

RETURN(SELECT CONCAT(A.NUME,' ',A.PRENUME)AS NUME\_PRENUME,SUM(ANG.SALARIU)AS SALARIU\_TOTAL FROM ANGAJAT A INNER JOIN ANGAJARE ANG

ON A.ANGAJAT\_ID=ANG.ANGAJAT\_ID

GROUP BY A.NUME,A.PRENUME )

SELECT \*

FROM DBO.ANGAJATII()

WHERE SALARIU\_TOTAL=(SELECT MAX(SALARIU\_TOTAL)FROM DBO.ANGAJATII() )

---PROCEDURI---

--PROCEDURA NU SE CHEAMA DIN INTERIORUL UNUI CAREVA SELECT

--SINTAXA:

--CREATE PROCEDURE NUME\_PROCEDURA LISTA\_PARAMETRII[IN/OUT]

--AS

--INSTRUCTIUNI

--APELUL PROCEDURII

---EXEC///EXECUTE NUME\_PROCEDURA ATRIBUIREA\_PARAMETRI

--SA SE SCRIE PROCEDURA CE VA AFISA O INF LA ECRAN PE BAZA UNOR PARAMETRI

ALTER PROCEDURE AFIS\_PC @PRET1 FLOAT=0, @PRET2 FLOAT

AS

SELECT \*FROM PRODUSE P INNER JOIN PC ON P.MODEL\_ID=PC.MODEL\_ID

WHERE PRET BETWEEN @PRET1 AND @PRET2

EXEC AFIS\_PC 700,5000

EXEC AFIS\_PC @PRET2=5000

--UTILIZREA TABELELOR VIRTUALE CU PROCEDURI

CREATE PROCEDURE TEST

AS

SELECT MODEL\_ID,PRET FROM PC

UNION

SELECT MODEL\_ID,PRET FROM LAPTOP

UNION

SELECT MODEL\_ID,PRET FROM IMPRIMANTA

--INSCRIEM REZULTATELE PROCEDURII IN TABELUL VIRTUAL

CREATE PROCEDURE XXXX

AS

CREATE TABLE #MYTABLE(NR INT PRIMARY KEY IDENTITY(1,1),MODEL\_ID INT,PRET FLOAT)

INSERT INTO #MYTABLE(MODEL\_ID,PRET) EXEC TEST

SELECT \* FROM #MYTABLE

WHERE PRET=(SELECT MIN(PRET) FROM #MYTABLE)

EXEC XXXX

--Pentru fiecare angajat sa se calculeze

--1. Numarul de angajari active

--2. numarul de angajari inactive

--3. si suma salariului activ

ALTER FUNCTION ANGG (@ANG INT,@TIP INT)

RETURNS INT

AS

BEGIN

DECLARE @REZ INT

IF(@TIP =1)

SET @REZ=(SELECT COUNT(\*) FROM ANGAJARE WHERE ANGAJARE\_ID=@ANG AND DATA\_ELIBERARII IS NULL);

ELSE

SET @REZ=(SELECT COUNT(\*) FROM ANGAJARE WHERE ANGAJARE\_ID=@ANG AND DATA\_ELIBERARII IS NOT NULL);

RETURN @REZ;

END

CREATE PROCEDURE AGG

AS

SELECT NUME,PRENUME,DBO.ANGG(ANGAJAT\_ID,1) ANGAJARI\_ACTIVE,DBO.ANGG(ANGAJAT\_ID,2) ANGAJARI\_INACTIVE

FROM ANGAJAT

EXEC AGG

--PENTRU FIECRE FUNCTIE SA SE DET NR.DE ANGAJATI ACTIVI SI SUMA SAL ACHIAT

CREATE FUNCTION ANG\_ACTIV (@FUNC INT,@TIP INT)

RETURNS INT

AS

BEGIN

RETURN (SELECT IIF(@TIP=1,COUNT(\*),SUM(SALARIU)) FROM ANGAJARE WHERE DATA\_ELIBERARII IS NULL

AND FUNCTIE\_ID=@FUNC )

END

CREATE PROCEDURE FUNCTIE\_ANG

AS

SELECT DENUMIRE\_FUNCTIE,DBO.ANG\_ACTIV(FUNCTIE\_ID,1)AS NR\_ANG,DBO.ANG\_ACTIV(FUNCTIE\_ID,2)AS SUMA\_SAL FROM FUNCTIE

EXEC FUNCTIE\_ANG

--sa se scrie procedura care va

--afisa pentru o anumita pers numarul de ang active

--si suma sal achitat

CREATE FUNCTION PP (@PERS INT,@TIP INT)

RETURNS INT

AS

BEGIN

RETURN(SELECT IIF(@TIP=0,COUNT(\*),SUM(SALARIU))

FROM ANGAJARE

WHERE DATA\_ELIBERARII IS NULL

AND ANGAJAT\_ID=@PERS)

END;

CREATE PROCEDURE PERS @COD\_PERS INT=0

AS

SELECT CONCAT(NUME,' ',PRENUME)AS NUME\_PRENUME, DBO.PP(ANGAJAT\_ID,0)AS NR\_ANGAJARI,

DBO.PP(ANGAJAT\_ID,1)AS SUM\_SALARIU

FROM ANGAJAT WHERE ANGAJAT\_ID=@COD\_PERS

EXEC PERS @COD\_PERS=3

--SA SE AFISEZE PENTRU UN ANUMIT DEPARTAMENT

--CARE SUNT CEI MAI BINE PLATITI

--ANGAJATI PE BAZA DE GEN

CREATE FUNCTION GEN\_SAL(@DEP INT)

RETURNS VARCHAR(200)

AS

BEGIN

DECLARE @SAL\_M DECIMAL(11,2), @SAL\_F DECIMAL(11,2), @REZ VARCHAR(200);

SET @SAL\_M=(SELECT SUM(SALARIU) FROM ANGAJARE A INNER JOIN ANGAJAT AA

ON A.ANGAJAT\_ID=AA.ANGAJAT\_ID WHERE GEN=1 AND DEPARTAMENT\_ID=@DEP AND DATA\_ELIBERARII IS NULL );

SET @SAL\_F=(SELECT SUM(SALARIU) FROM ANGAJARE A INNER JOIN ANGAJAT AA

ON A.ANGAJAT\_ID=AA.ANGAJAT\_ID WHERE GEN=0 AND DEPARTAMENT\_ID=@DEP AND DATA\_ELIBERARII IS NULL );

IF(@SAL\_M=@SAL\_F) BEGIN

SET @REZ='SALARIUL LA GENUL MASCULUIN('+CAST(@SAL\_M AS VARCHAR(20))+' LEI) = SALARIUL LA GENUL FEMININ('+CAST(@SAL\_F AS VARCHAR(20))

END;

ELSE IF(@SAL\_M>@SAL\_F) BEGIN

SET @REZ='SALARIUL LA GENUL MASCULUIN('+CAST(@SAL\_M AS VARCHAR(20))+' LEI) > SALARIUL LA GENUL FEMININ('+CAST(@SAL\_F AS VARCHAR(20))

END;

ELSE

BEGIN

SET @REZ='SALARIUL LA GENUL MASCULUIN('+CAST(@SAL\_M AS VARCHAR(20))+' LEI) < SALARIUL LA GENUL FEMININ('+CAST(@SAL\_F AS VARCHAR(20))

END;

RETURN @REZ;

END

CREATE PROCEDURE ANALIZA\_SAL\_GEN @COD\_DEPARTAMENT INT

AS

SELECT DENUMIRE AS DEPARTAMENT,DBO.GEN\_SAL(DEPARTAMENT\_ID)AS ANALIZA FROM DEPARTAMENT

WHERE DEPARTAMENT\_ID=@COD\_DEPARTAMENT

EXEC ANALIZA\_SAL\_GEN @COD\_DEPARTAMENT=1