Gestiunea universitatilor

Proiect realizat in SAP ABAP de Scarlat Simona Ileana

Facultate: Cibernetica, Statistica si Informatica economica

Specializare: Informatica economica

Seria B, Informatica economica

Grupa 1034

Cerinte proiect

1. Minim 3 tabele ( campurile folosite sa fie de tipul data element creat de voi ) - 1 p;
2. Tabelele create trebuie relationate - 1p ;
3. Structuri si table type pentru tabelele create – 1p;
4. Toate obiectele create in data dictionary sa fie traduse – 1p ;
5. Cel putin 3 dintre domeniile folosite pentru data element sa aiba value range - 1p;
6. Tabelele sa aiba create maintenance view -1p ;
7. Programul creat sa contina minim 4 parametrii, 1 radiobutton si 1 check-box - 1p;
8. Sa fie definit cel putin un tip intern - 1p;
9. Sa fie folosite instructiuni repetitive( do, loop, while) – 1p.
10. Programul structurat pe include-uri - 1p;
11. Codul sa fie lizibil si aliniat folosind Pretty Printer – 1p .
12. Interogari pe baza de date( SELECT) – 1p.
13. Modificari pe baza de date(INSERT, UPDATE, MODIFY, DELETE ) -1p.
14. Operatii pe tabele interne ( APPEND, COLLECT, INSERT, DELETE ) – 1p.
15. Afisarea datelor folosind ALV – 1p ;
16. De folosit cel putin 2 function modules standard SAP – 1p.
17. Folosirea subroutinellor I or. – 1p.
18. Folosirea naming-convention. – 1p
19. Folosirea evenimentelor in program( INITIALIZATION, AT SELECTION-SCREEN, START\_OF\_SELECTION, etc) . 1p

Tehnologia utilizata

SAP ABAP

Sap reprezinta Sistemul Aplicatiilor si Produselor realizat de firma germana SAP AG. SAP-ul este la ora actuala cel mai important program de tip ERP (Planificarea resurselor intreprinderii) din lume.

Programul SAP este foarte complex, fiind de mare ajutor in Servicii de Afaceri, Solutii Tehnologice, in arii precum Contabilitate, Managementul Relatiilor cu Clientii, Lanturi de furnizare si Aprovizionare, Bancar, Asigurari, Infrastructura IT, Servicii Software.

SAP este liderul recunoscut in solutii software colaborative pentru afaceri pentru toate industriile si toate pietele semnificative, SAP este cea mai mare companie de software pentru afaceri si al treilea producator independent de software la nivel mondial datorita inovarii si dezvoltarii. SAP Romania se afla pe primul loc in clasamentul furnizorilor de aplicatii de afaceri, cu o cota de piata de 25,2% in anul 2006, in Romania, arata cel mai recent studiu, pentru piata EAS - Enterprise Aplications Software.

Cu ajutorul platformei tehnologice de aplicatii si integrare, aplicatiile software SAP permit companiilordin intreaga lume sa isi imbunatateasca relatiile cu clientii si partenerii, sa isi eficientizeze activitatea si sa obtina avantaje semnificative in cadrul intregii organizatii.

Interfata programului SAP



Meniul SAP Easy Acces este format din urmatoarele componente:

* Logistica
* Contabilitate si costuri
* Resurse umane
* Planificare a productiei

Domeniile de competenta ale echipei sunt focusate pe urmatoarele domenii de afaceri:

* Distributie
* Retail
* Productie
* Constructii
* Administrare Publica
* Banking

Tranzactii:



Descrierea proiectului

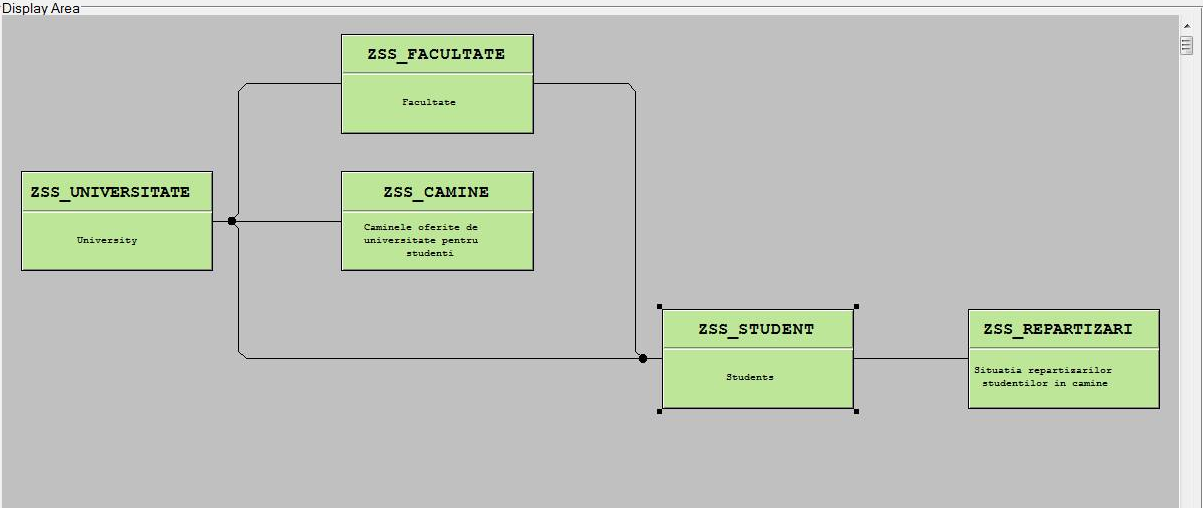
1. Prezentarea proiectului

Tema acestui proiect este gestiunea unei baze de date care contine evidenta universitatilor, atat informatii generale despre acestea, cat si toate facultatile unei universitati, caminele oferite studentilor pentru cazare, studentii inregistrati la fiecare facultate si repartizarea acestora in camine.

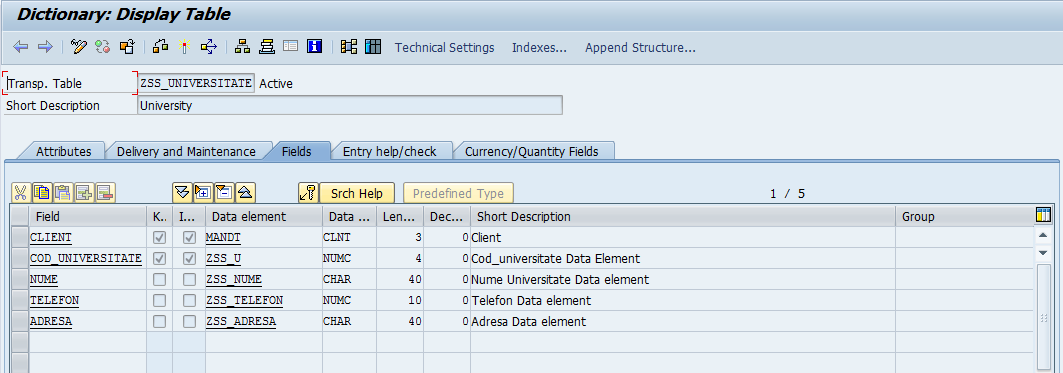
Reguli:

* o universitate are mai multe facultati;
* o facultate apartine unei singure universitati
* fiecare universitate are un cod unic
* fiecare facultate are un cod unic
* fiecare student apartine unei singure facultati din cadrul unei universitati
* fiecare student e repartizat la un singur camin oferit de universitatea la care apartine
* fiecare student are un cod unic
* fiecare universitate ofera studentilor pentru cazare un numar de camine
* fiecare camin are un cod unic si un singur administrator de camin

1. Schema UML



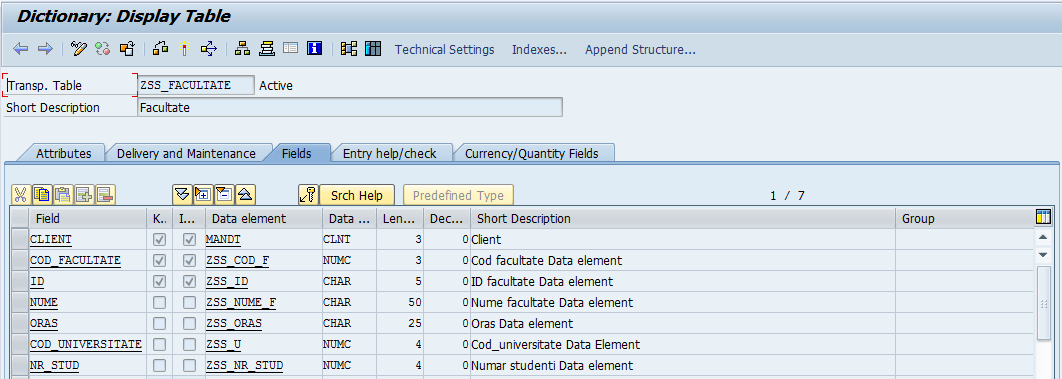
1. Descrierea entitatilor, relatiilor si atributelor
2. Entitatea independenta ZSS\_UNIVERSITATE - orice universitate din tara. Cheie primara este COD\_UNIVERSITATE, iar fiecare obiect este tradus in limba germana.



Atribute:

* COD\_UNIVERSITATE, avand ca Data element ZSS\_U cu domeniul de tipul NUMC de lungime 4
* NUME, avand ca Data element ZSS\_NUME cu domeniul de tipul CHAR de lungime 40
* TELEFON, avand ca Data element ZSS\_TELEFON cu domeniul de tipul NUMC de lungime 10
* ADRESA, avand ca Data element ZSS\_ADRESA cu domeniul de tipul CHAR de lungime 40

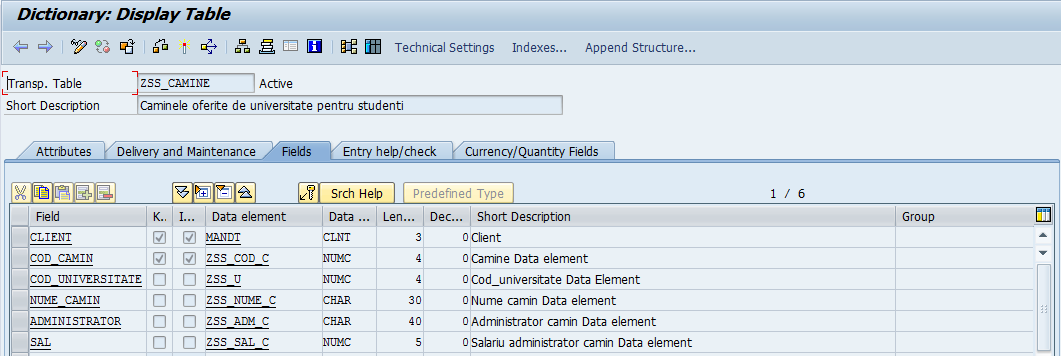
1. Entitatea dependenta ZSS\_FACULTATE - contine informatii despre facultatile universitatilor. Cheia primara este COD\_FACULTATE, iar cheia externa, care face legatura cu tabela ZSS\_UNIVERSITATE este COD\_UNIVERSITATE. Toate atributele sunt traduse in limba germana.



Atribute:

* COD\_FACULTATE, avand ca Data element ZSS\_COD\_F cu domeniul de tipul NUMC de lungime 3
* ID, avand ca Data element ZSS\_ID cu domeniul de tipul CHAR de lungime 5. Acest domeniu are si un value range : AMP, CSIE, COM, MRK, MAN, ETA, FABBV, REI, CIG, EAM
* NUME, avand ca Data element ZSS\_NUME\_F cu domeniul de tipul CHAR de lungime 50
* ORAS, avand ca Data element ZSS\_ORAS cu domeniul de tipul CHAR de lungime 25 si value range BUCURESTI, PITESTI, BUZAU, GALATI, SLATINA, CLUJ
* COD\_UNIVERSITATE, ca si cheie externa, pentru a face legatura cu tabela ZSS\_UNIVERSITATE
* NR\_STUD, numarul de studenti al fiecarei facultati, avand ca Data element ZSS\_NR\_STUD cu domeniul de tipul NUMC de lungime 4

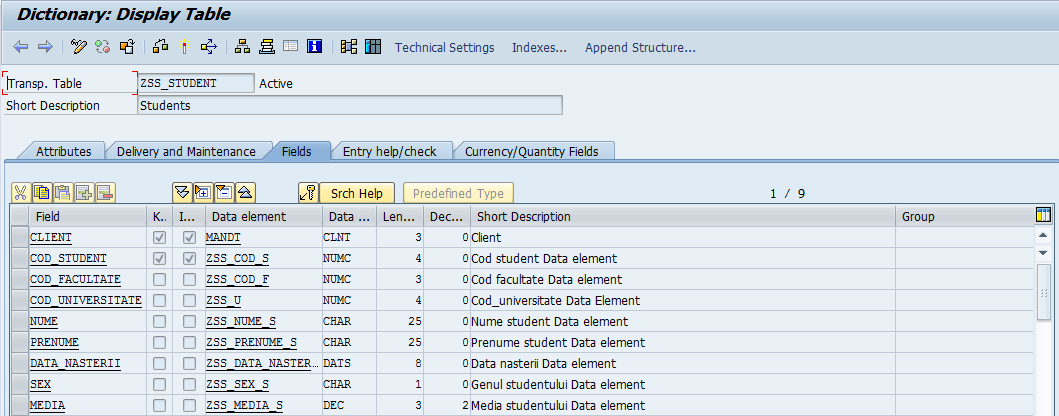
1. Entitatea dependenta ZSS\_CAMINE - contine informatii despre caminele oferite de fiecare universitate. Cheia primara este COD\_CAMIN, iar cheia externa este COD\_UNIVERSITATE si face legatura cu tabela ZSS\_UNIVERSITATE. Toate atributele sunt traduse in limba germana.



Atribute:

* COD\_CAMIN, avand ca Data element ZSS\_COD\_C cu domeniul de tipul NUMC de lungime 4
* COD\_UNIVERSITATE ca si cheie externa, pentru a face legatura cu tabela ZSS\_UNIVERSITATE
* NUME\_CAMIN, avand ca Data element ZSS\_NUME\_C cu domeniul de tipul CHAR de lungime 30, acest domeniu are value range: BELV A1, BELV A2, BELV A3, BELV A4, BELV A6, BELV A7, BELV A8, MOXA D, AGRON C1, AGRON C2, TEI C1.
* ADMINISTRATOR - administratorul fiecarui camin, avand ca Data element ZSS\_ADM\_C, cu domeniul de tipul CHAR de lungime 40
* SAL- salariul fiecarui administrator pentru fiecare camin, avand ca Data element ZSS\_SAL\_C, cu domeniul de tipul NUMC de lungime 5.

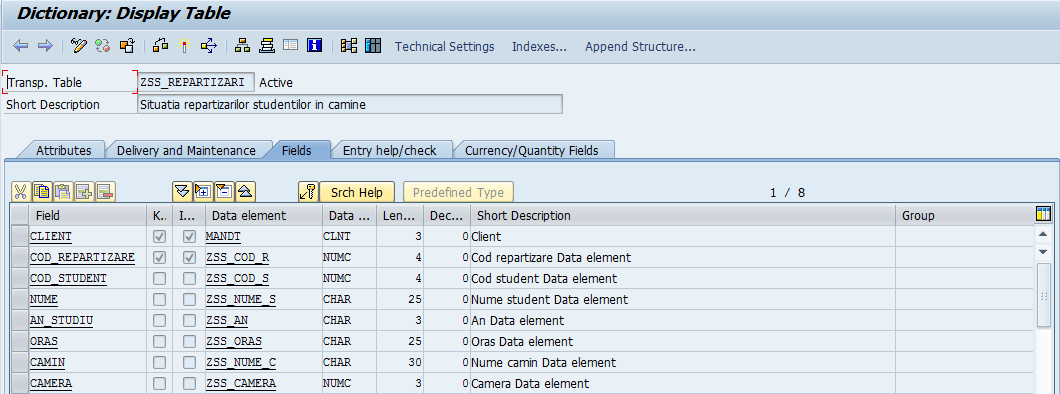
1. Entitatea dependenta ZSS\_STUDENT - contine informatii despre studentii fiecarei facultati a unei universitati. Cheia primara este COD\_STUDENT, iar cheile externe sunt COD\_UNIVERSITATE, pentru a face legatura cu tabela ZSS\_UNIVERSITATE si COD\_FACULTATE, pentru a face legatura cu tabela ZSS\_FACULTATE. Toate atributele sunt traduse in limba germana.



Atribute:

* COD\_STUDENT, avand ca Data element ZSS\_COD\_S cu domeniul de tipul NUMC de lungime 4
* COD\_FACULTATE - cheie externa, pentru a face legatura cu tabela ZSS\_FACULTATE
* COD\_UNIVERSITATE - cheie externa, pentru a face legatura cu tabela ZSS\_UNIVERSITATE
* NUME, avand ca Data element ZSS\_NUME\_S cu domeniul de tipul CHAR de lungime 25
* PRENUME, avand ca Data element ZSS\_PRENUME\_S cu domeniul de tipul CHAR de lungime 25
* DATA\_NASTERII, avand ca Data element ZSS\_DATA\_NASTERII cu domeniul de tipul DATS de lungime 8
* SEX - genul fiecarui student, avand ca Data element ZSS\_SEX\_S cu domeniul de tipul CHAR de lungime 1, acest domeniu contine value range: M pentru masculin si F pentru feminin
* MEDIA, avand ca Data element ZSS\_MEDIA\_S, cu domeniul de tipul DEC de lungime 3 (cu doua zecimale)

1. Entitatea dependenta ZSS\_REPARTIZARI - contine informatii despre repartizarea studentilor in caminele studentesti. Cheia primara este COD\_REPARTIZARE, iar cheia externa este COD\_STUDENT pentru a face legatura cu tabela ZSS\_STUDENT. Toate atributele sunt traduse in limba germana. Aceasta tabela contine Maintenance view.

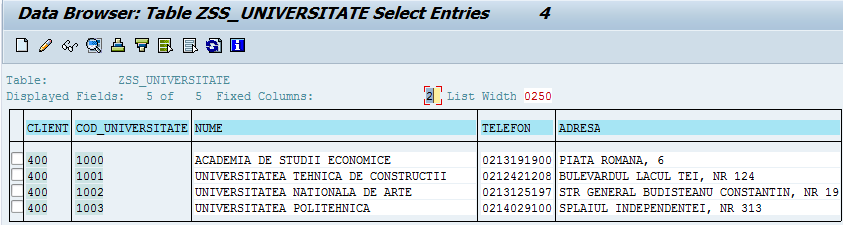


Atribute:

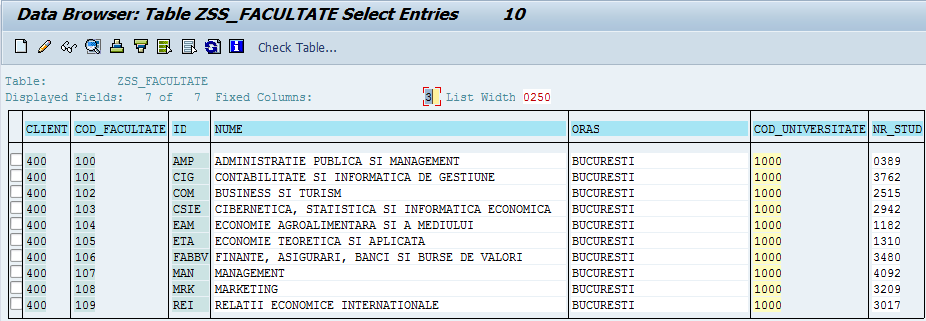
* COD\_REPARTIZARE, avand ca Data element ZSS\_COD\_R, cu domeniul de tipul NUMC de lungime 4
* COD\_STUDENT - cheie externa, pentru a face legatura cu tabela ZSS\_STUDENT
* NUME, avand ca Data element ZSS\_NUME\_S, cu domeniul de tipul CHAR de lungime 25
* AN\_STUDIU, avand ca Data element ZSS\_AN, cu domeniul de tipul CHAR de lungime 3, acest domeniu are value range: I, II , III.
* ORAS, avand ca Data element ZSS\_ORAS, cu domeniul de tipul CHAR de lungime 25
* CAMIN, avand ca Data element ZSS\_NUME\_C, cu domeniul de tipul CHAR de lungime 30
* CAMERA, avand ca Data element ZSS\_CAMERA, cu domeniul de tipul NUMC de lungime 3

1. Tabelele asociate

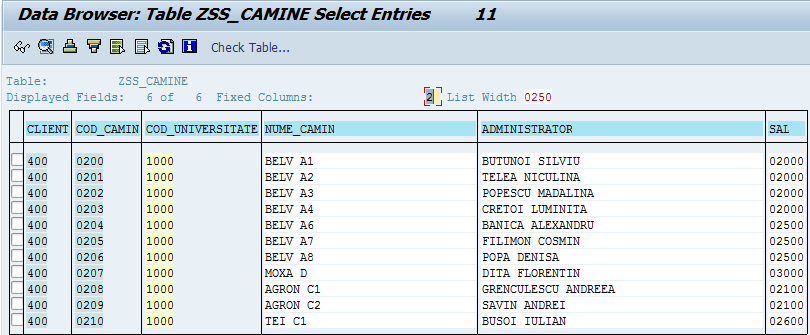
* ZSS\_UNIVERSITATE



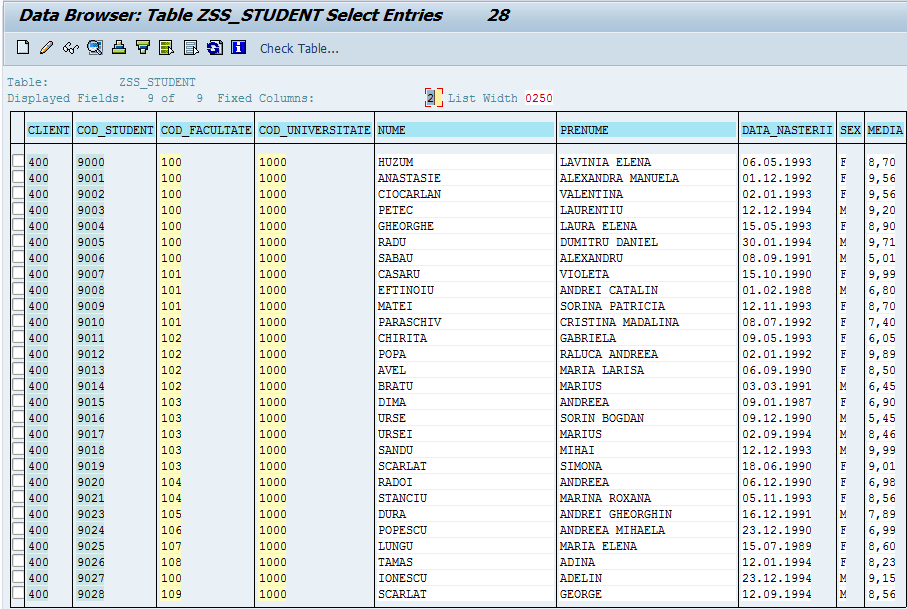
* ZSS\_FACULTATE



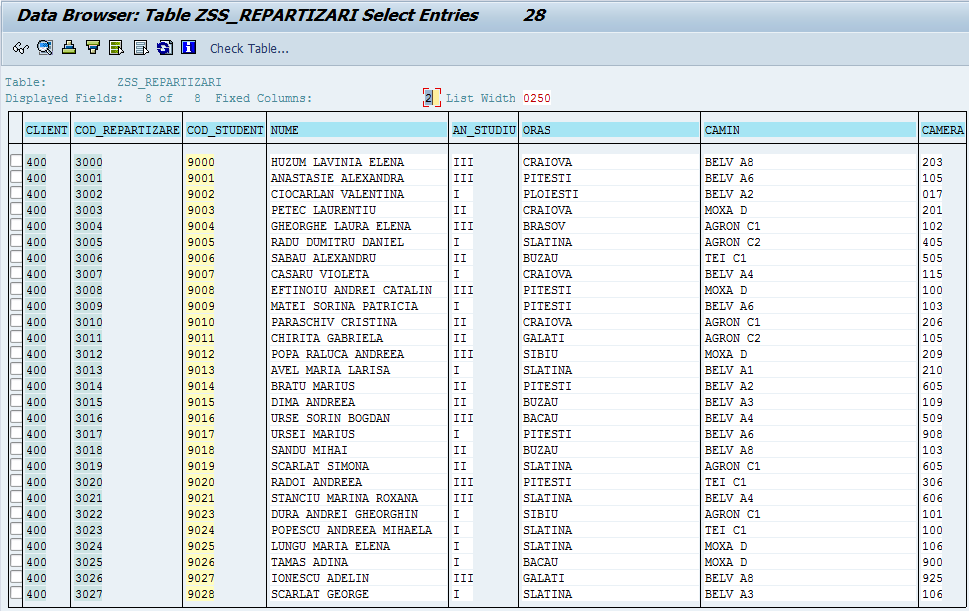
* ZSS\_CAMINE



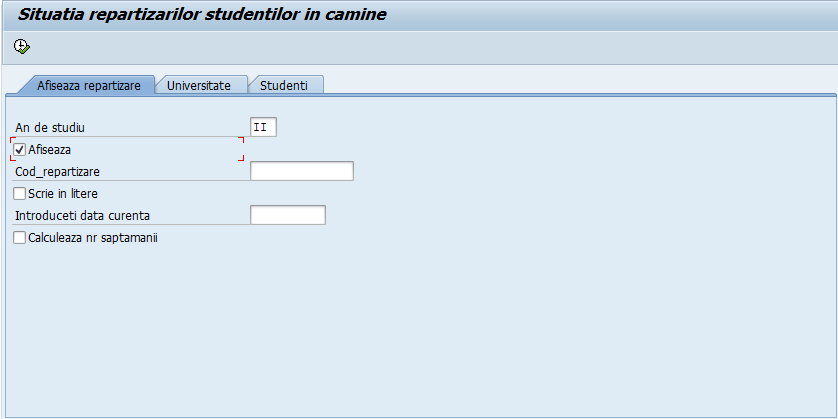
* ZSS\_STUDENT



* ZSS\_REPARTIZARE



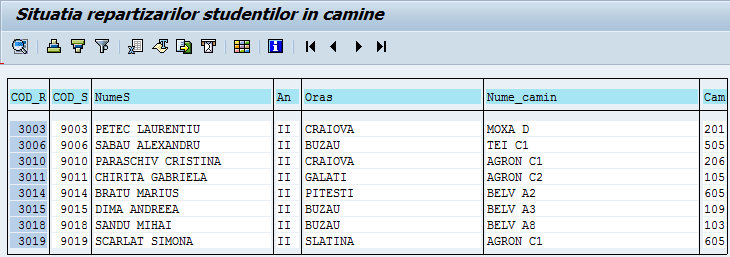
1. Aplicatia ZSS\_PROIECT\_FINAL3

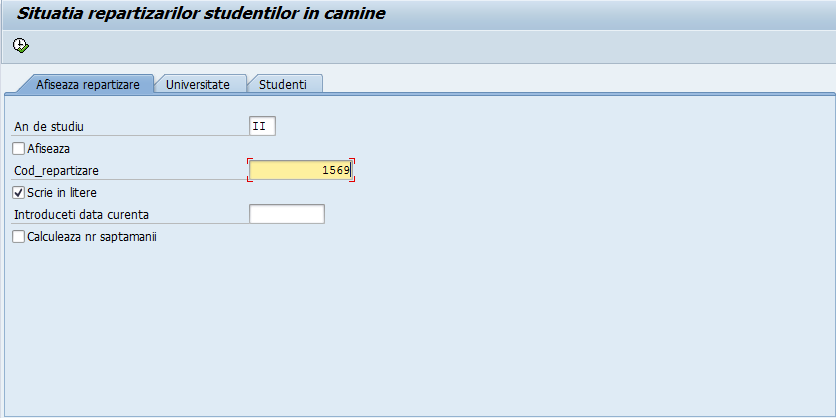


Se bifeaza checkbox-ul Afiseaza pentru a se executa comanda

Se specifica anul de studiu pentru care se doreste afisarea repartizarii studentilor in camine (I, II sau III)

Dupa bifarea checkbox-ului Afiseaza, si specificarea Anului de studiu, in acest caz, II, se va afisa urmatorul ecran (se foloseste ALV):



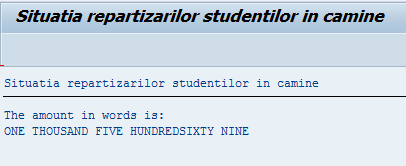


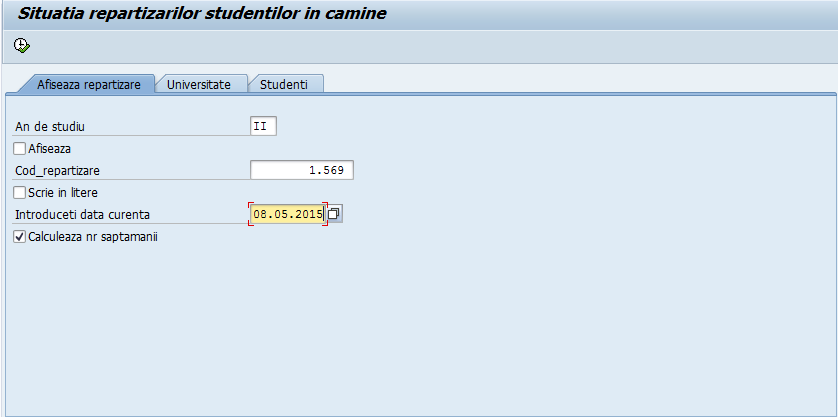
Se bifeaza checkbox-ul Scrie in litere, care va scrie numarul introdus in litere

Se introduce un cod de repartizare, indiferent daca acesta exista sau nu in baza de date

Dupa bifarea checkbox-ului Scrie in litere, se va afisa urmatorul ecran:

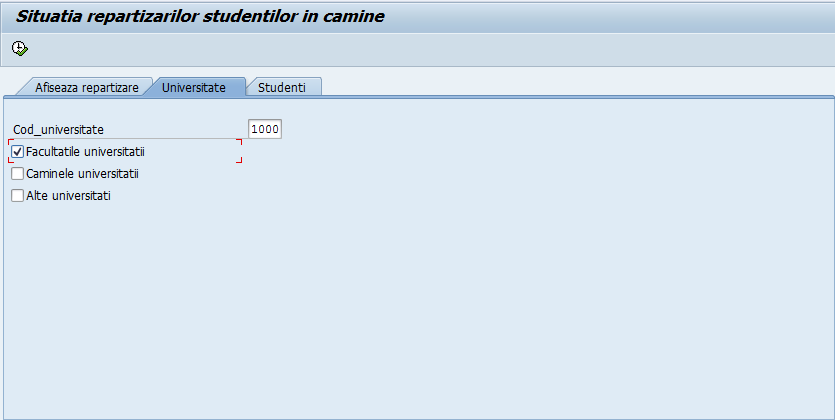
(s-a folosind sell\_amount -function module)





Se bifeaza checkbox-ul Calculeaza nr saptamanii, care va afisa pe ecran anul curent si calculeaza in a cata saptamana din an suntem. Se va afisa 201519

Se introduce data curenta



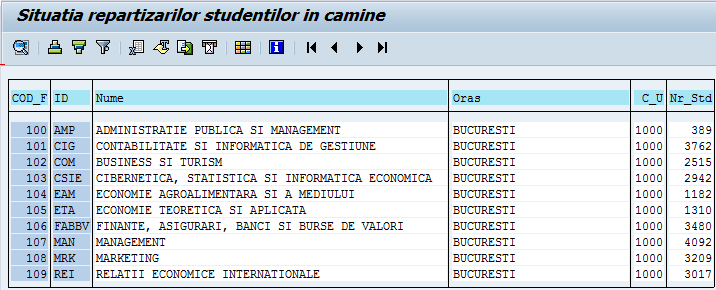
Cand se bifeaza checkbox-ul Caminele universitatii, vor fi afisate toate caminele oferite de universitatea al carei cod l-am introdus.

Se bifeaza checkbox-ul Facultatile universitatii care va afisa toate facultatile care apartin universitatii al carei cod l-am introdus

Se introduce codul universitatii inregistrat in tabela ZSS\_UNIVERSITATE

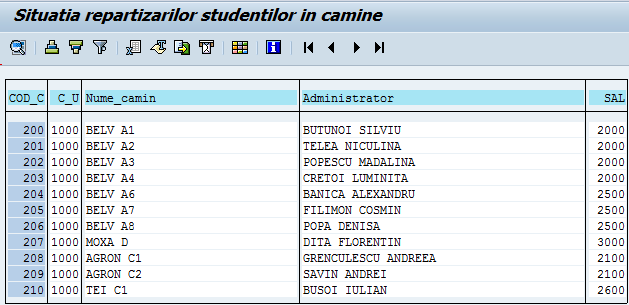
Dupa bifarea checkbox-ului Facultatile universitatii, se va afisa urmatorul ecran :

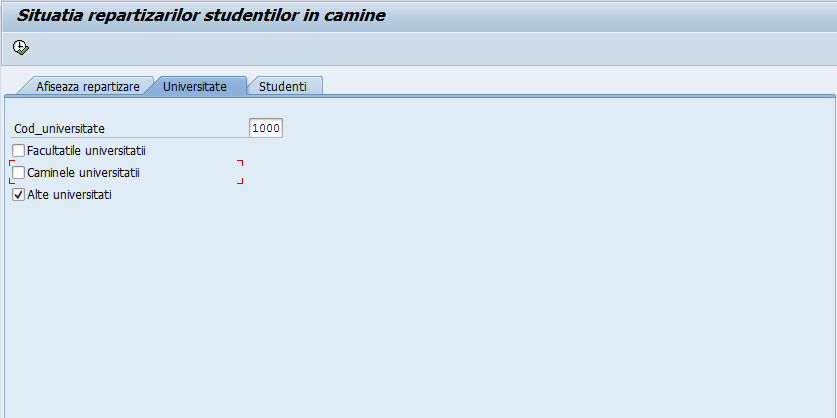
(s-a folosit ALV)



Dupa bifarea checkbox-ului Caminele universitatii, se va afisa urmatorul ecran:

(s-a folosit ALV)

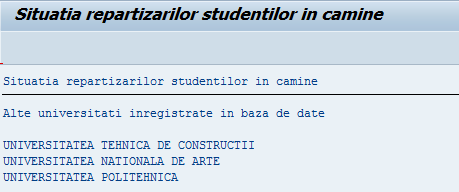


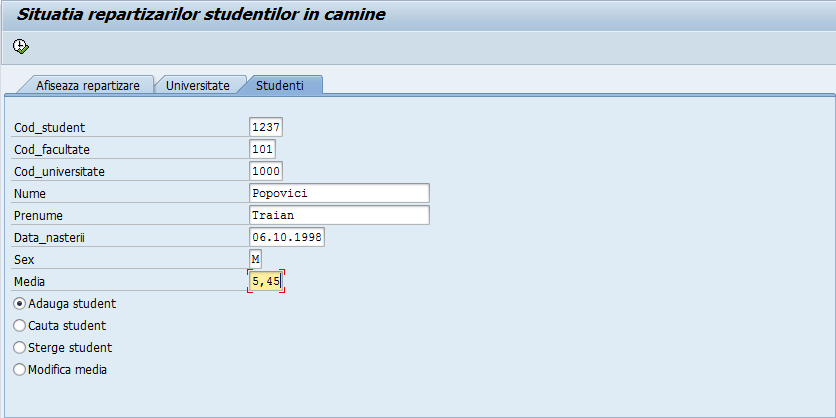


Cand se bifeaza checkbox-ul Alte universitati, se va afisa pe ecran titlul "Alte universitati inregistrate in baza de date" si celelalte universitati inregistrate in baza de date.

Se introduce codul universitatii inregistrat in tabela ZSS\_UNIVERSITATE

Dupa bifarea checkbox-ului Alte universitati se va afisa urmatorul ecran:

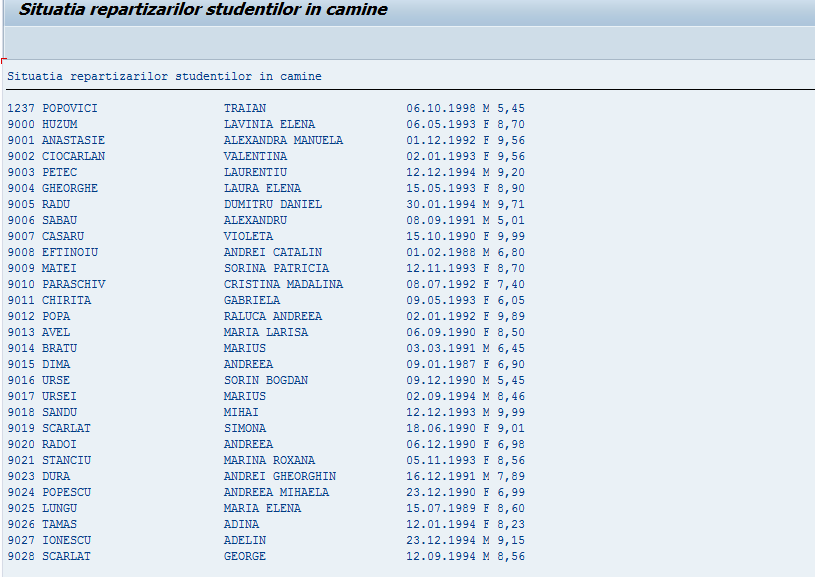




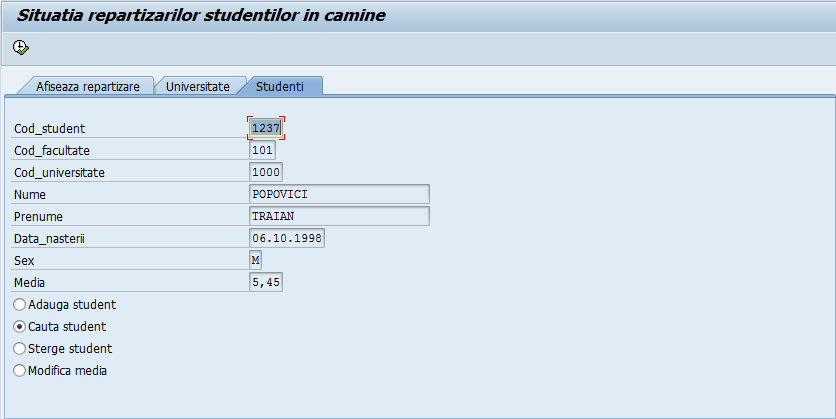
Cand este bifat radiobutton-ul Adauga student, se va adauga un student nou in baza de date, dupa care se va afisa pe ecran toate inregistrarile existente acum in baza de date, inclusiv studentul adaugat.

Se introduce un cod care nu exista deja in tabela ZSS\_STUDENT

Dupa bifarea radiobutton-ului Adauga student, se va afisa urmatorul ecran:



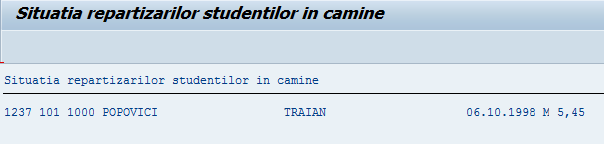
studentul pe care l-am adaugat in baza de date

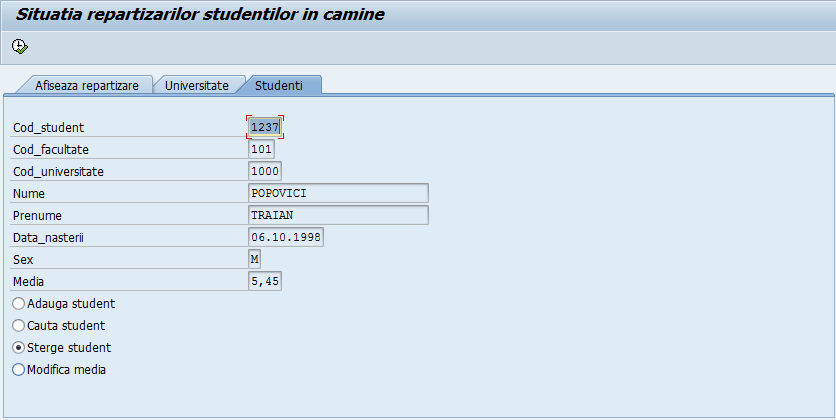


Se introduce codul studentului

Cand este bifat radiobutton-ul Cauta student, va fi activ doar parametrul pentru Cod\_student, deoarece se va cauta studentul in baza de date doar folosindu-ne de codul acestuia

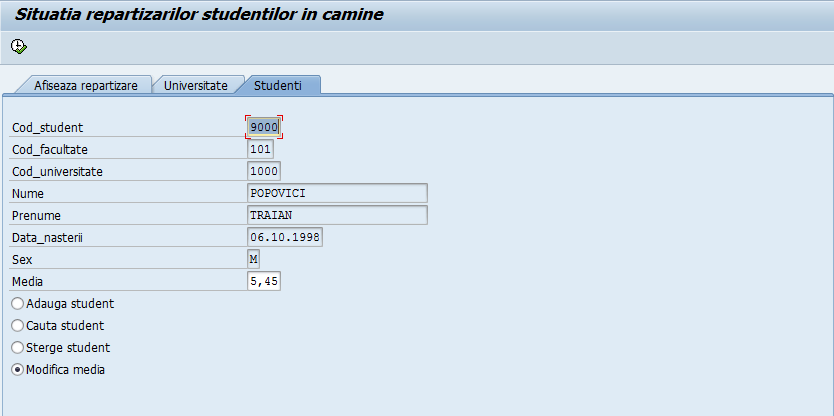
Dupa bifarea radiobutton-ului Cauta student, se va afisa urmatorul ecran:





Cand va fi bifat radiobutton-ul Sterge student, studentul va fi sters din baza de date, nu se va afisa nimic pe ecran.

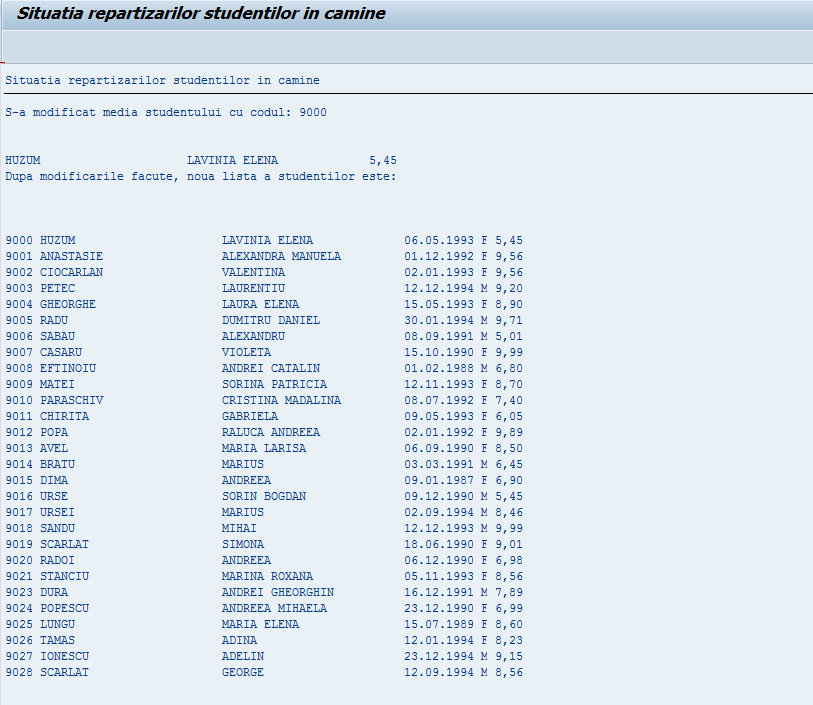
Se introduce codul studentului pe care vrem sa-l stergem din baza de date



Cand este bifat radiobutton-ul Modifica media se va modifica din baza de date media studentului al carui cod l-am introdus, iar apoi se va afisa pe ecran inregistrarea corespunzatoare studentului care are codul respectiv si toate inregistrarile existente acum in baza de date, dupa modificarile facute

Se introduce codul studentului caruia vrem sa-i modificam media

Dupa bifarea radiobutton-ului Modifica media, se va afisa urmatorul ecran:



\*&---------------------------------------------------------------------\*

\*& Include ZSS\_PROIECT\_FINAL3\_TOP

\*&---------------------------------------------------------------------\*

\*--------------------------------------------------------------------\*

\* Types declaration \*

\*--------------------------------------------------------------------\*

TYPES: ty\_repartizare TYPE zss\_repartizari\_struct,

tt\_repartizare TYPE zss\_repartizari\_tt,

ty\_facultate TYPE zss\_facultate\_struct,

ty\_student TYPE zss\_student\_struct,

ty\_camine TYPE zss\_camine\_struct,

tt\_student TYPE zss\_student\_tt,

ty\_universitate TYPE zss\_universitate\_struct.

\*definire tip intern

TYPES: BEGIN OF ty\_repartizare\_add,

cod\_repartizare TYPE zss\_repartizari\_struct-cod\_repartizare,

id TYPE zss\_facultate\_struct-id,

nume TYPE zss\_repartizari\_struct-nume,

camin TYPE zss\_repartizari\_struct-camin,

camera TYPE zss\_repartizari\_struct-camera,

END OF ty\_repartizare\_add.

TYPES: tt\_repartizari TYPE ty\_repartizare\_add.

\*---------------------------------------------------------------------------\*

\* Data declaration \*

\*----------------------------------------------------------------------------\*

DATA: ls\_repartizare TYPE zss\_repartizari,

\* lt\_repartizare TYPE tt\_repartizare,

\* ls\_repartizare\_add TYPE ty\_repartizare\_add,

result LIKE spell,

week LIKE scal-week,

\* gs\_facultate TYPE ty\_facultate,

\* gs\_camine TYPE ty\_camine,

gs\_student TYPE ty\_student,

\* gs\_repartizare TYPE ty\_repartizare\_add,

ls\_student TYPE ty\_student,

lt\_student TYPE tt\_student,

ls\_universitate TYPE ty\_universitate,

lt\_universitate TYPE STANDARD TABLE OF zss\_universitate,

wa\_student TYPE zss\_student\_struct,

gf\_student TYPE zss\_student\_struct,

gt\_student TYPE zss\_student\_tt,

it\_view TYPE TABLE OF zss\_repartizari,

it\_facultate TYPE TABLE OF zss\_facultate,

it\_camine TYPE TABLE OF zss\_camine,

text TYPE string VALUE 'Alte-universitati-inregistrate-in-baza-de-date'.

\*--------------------------------------------------------------------------------\*

\* Selection screen \* \*

\*--------------------------------------------------------------------------------\*

SELECTION-SCREEN BEGIN OF SCREEN 100 AS SUBSCREEN.

PARAMETERS:

p\_an TYPE zss\_repartizari\_struct-an\_studiu, "Introduceti anul de studiu pentru care sa se faca afisarea

p\_afis AS CHECKBOX, "buton pentru afisarea repartizarii studentilor in camine

mynum TYPE i, "parametru pentru a introduce o valoare

p\_number AS CHECKBOX, "buton care trebuie bifat pentru a se afisa in litere codul introdus

mydata TYPE zss\_student\_struct-data\_nasterii, "parametru pt a introduce data curenta

cal AS CHECKBOX. "Calculeaza nr saptamanii

SELECTION-SCREEN END OF SCREEN 100.

SELECTION-SCREEN BEGIN OF SCREEN 200 AS SUBSCREEN.

PARAMETERS: p\_cod\_s TYPE zss\_student\_struct-cod\_student, "Cod\_student

p\_cod\_f TYPE zss\_student\_struct-cod\_facultate, "Cod facultate

p\_cod\_u TYPE zss\_student\_struct-cod\_universitate, "Cod universitate

p\_nume TYPE zss\_student\_struct-nume, "Nume student

p\_prenum TYPE zss\_student\_struct-prenume, "Prenume student

p\_datan TYPE zss\_student\_struct-data\_nasterii, "Data nasterii a studentului

p\_sex TYPE zss\_student\_struct-sex, "Sexul studentului

p\_media TYPE zss\_student\_struct-media, "media studentului

p\_insert RADIOBUTTON GROUP r1 USER-COMMAND md1, "buton care atunci cand e bifat se insereaza in baza de date un nou student si in tabela interna

p\_cauta RADIOBUTTON GROUP r1, "buton care atunci cand e bifat cauta un student in baza de date si il afiseaza pe ecran

p\_delete RADIOBUTTON GROUP r1, "buton care atunci cand e bifat imi sterge un student din baza de date

p\_modif RADIOBUTTON GROUP r1. "buton care atunci cand e bifat imi modifica media pentru studentul al carui cod l-am introdus in parametru

SELECTION-SCREEN END OF SCREEN 200.

SELECTION-SCREEN BEGIN OF SCREEN 300 AS SUBSCREEN.

PARAMETERS: p\_u TYPE zss\_universitate\_struct-cod\_universitate, "Introducem codul universitatii pt care vrem sa se faca urmatoarele afisari

p\_afisf AS CHECKBOX, "Afiseaza facultatile universitatii

p\_afisc AS CHECKBOX, "Afiseaza caminele universitatii

p\_c AS CHECKBOX. "Afiseaza celelalte universitati

SELECTION-SCREEN END OF SCREEN 300.

SELECTION-SCREEN: BEGIN OF TABBED BLOCK mytab FOR 13 LINES,

TAB (20) button1 USER-COMMAND push1, "Situatia repartizarilor

TAB (20) button2 USER-COMMAND push2, "Universitate

TAB (20) button3 USER-COMMAND push3, "Studenti

END OF BLOCK mytab.

\*&---------------------------------------------------------------------\*

\*& Include ZSS\_PROIECT\_FINAL3\_E01

\*&---------------------------------------------------------------------\*

\*------------------------------------------------------------------------------------\*

\* Initialization \*

\*-------------------------------------------------------------------------------------\*

INITIALIZATION.

perform initialization.

\*--------------------------------------------------------------------------------------------\*

\* At Selection screen \*

\*---------------------------------------------------------------------------------------------\*

AT SELECTION-SCREEN.

CASE sy-dynnr.

WHEN 1000.

CASE sy-ucomm.

WHEN 'PUSH1'.

mytab-dynnr = 100.

WHEN 'PUSH2'.

mytab-dynnr = 300.

WHEN 'PUSH3'.

mytab-dynnr = 200.

ENDCASE.

ENDCASE.

"sa-mi blocheze anumiti parametri atunci cand este bifat un anumit buton

"cand sunt bifate butoanele p\_delete sau p\_cauta, sa-mi blocheze tot in afara de cod\_student

AT SELECTION-SCREEN OUTPUT."PBO

LOOP AT SCREEN.

CASE screen-name.

WHEN 'P\_COD\_F' OR 'P\_COD\_U' OR 'P\_NUME' OR 'P\_PRENUM' OR 'P\_DATAN' OR 'P\_SEX' OR 'P\_MEDIA'.

IF p\_delete = 'X' OR p\_cauta = 'X'.

screen-input = 0.

MODIFY SCREEN.

ENDIF.

ENDCASE.

ENDLOOP.

"cand este bifat butonul p\_modif, sa-mi blocheze tot mai putin cod\_student si p\_media

LOOP AT SCREEN.

CASE screen-name.

WHEN 'P\_COD\_U' OR 'P\_COD\_F' OR 'P\_NUME' OR 'P\_PRENUM' OR 'P\_DATAN' OR 'P\_SEX'.

IF p\_modif = 'X'.

screen-input = 0.

MODIFY SCREEN.

ENDIF.

ENDCASE.

ENDLOOP.

\* --------------------------------------------------------------------------\*

\* Start of processing event \*

\* ---------------------------------------------------------------------------\*

START-OF-SELECTION.

"Afiseaza facultatile cand apasam butonul p\_afisf cu alv

IF p\_afisf = 'X'.

SELECT \* FROM zss\_facultate INTO TABLE it\_facultate WHERE cod\_universitate = p\_u.

CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'

EXPORTING

\* I\_INTERFACE\_CHECK = ' '

\* I\_BYPASSING\_BUFFER =

\* I\_BUFFER\_ACTIVE = ' '

\* I\_CALLBACK\_PROGRAM = ' '

\* I\_CALLBACK\_PF\_STATUS\_SET = ' '

\* I\_CALLBACK\_USER\_COMMAND = ' '

i\_structure\_name = 'zss\_facultate'

\* IS\_LAYOUT =

\* IT\_FIELDCAT =

\* IT\_EXCLUDING =

\* IT\_SPECIAL\_GROUPS =

\* IT\_SORT =

\* IT\_FILTER =

\* IS\_SEL\_HIDE =

\* I\_DEFAULT = 'X'

\* I\_SAVE = ' '

\* IS\_VARIANT =

\* IT\_EVENTS =

\* IT\_EVENT\_EXIT =

\* IS\_PRINT =

\* IS\_REPREP\_ID =

\* I\_SCREEN\_START\_COLUMN = 0

\* I\_SCREEN\_START\_LINE = 0

\* I\_SCREEN\_END\_COLUMN = 0

\* I\_SCREEN\_END\_LINE = 0

\* IR\_SALV\_LIST\_ADAPTER =

\* IT\_EXCEPT\_QINFO =

\* I\_SUPPRESS\_EMPTY\_DATA = ABAP\_FALSE

\* IMPORTING

\* E\_EXIT\_CAUSED\_BY\_CALLER =

\* ES\_EXIT\_CAUSED\_BY\_USER =

TABLES

t\_outtab = it\_facultate

\* EXCEPTIONS

\* PROGRAM\_ERROR = 1

\* OTHERS = 2

.

IF sy-subrc <> 0.

\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO

\* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

ENDIF.

"Alta modalitate de a afisa datele fara alv

\* WRITE: / 'Universitatea al carui cod l-ai introdus are urmatoarele facultati:'.

\* SKIP.

\* WRITE: sy-uline(80).

\* WRITE: / '|', AT 5 'Cod facultate', AT 26 '|','Nume facultate', AT 80 '|'.

\* SELECT cod\_facultate nume

\* FROM zss\_facultate

\* INTO CORRESPONDING FIELDS OF gs\_facultate

\* WHERE cod\_universitate = p\_u.

\* WRITE: / sy-uline(80).

\* WRITE: / '|',gs\_facultate-cod\_facultate UNDER 'Cod facultate', AT 26 '|',gs\_facultate-nume UNDER 'Nume facultate', AT 80'|'.

\* ENDSELECT.

\* WRITE: / sy-uline(80).

ENDIF.

"Afiseaza caminele cand apasam butonul p\_afisc folosind alv

IF p\_afisc = 'X'.

SELECT \* FROM zss\_camine INTO TABLE it\_camine WHERE cod\_universitate = p\_u.

CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'

EXPORTING

\* I\_INTERFACE\_CHECK = ' '

\* I\_BYPASSING\_BUFFER =

\* I\_BUFFER\_ACTIVE = ' '

\* I\_CALLBACK\_PROGRAM = ' '

\* I\_CALLBACK\_PF\_STATUS\_SET = ' '

\* I\_CALLBACK\_USER\_COMMAND = ' '

i\_structure\_name = 'ZSS\_camine'

\* IS\_LAYOUT =

\* IT\_FIELDCAT =

\* IT\_EXCLUDING =

\* IT\_SPECIAL\_GROUPS =

\* IT\_SORT =

\* IT\_FILTER =

\* IS\_SEL\_HIDE =

\* I\_DEFAULT = 'X'

\* I\_SAVE = ' '

\* IS\_VARIANT =

\* IT\_EVENTS =

\* IT\_EVENT\_EXIT =

\* IS\_PRINT =

\* IS\_REPREP\_ID =

\* I\_SCREEN\_START\_COLUMN = 0

\* I\_SCREEN\_START\_LINE = 0

\* I\_SCREEN\_END\_COLUMN = 0

\* I\_SCREEN\_END\_LINE = 0

\* IR\_SALV\_LIST\_ADAPTER =

\* IT\_EXCEPT\_QINFO =

\* I\_SUPPRESS\_EMPTY\_DATA = ABAP\_FALSE

\* IMPORTING

\* E\_EXIT\_CAUSED\_BY\_CALLER =

\* ES\_EXIT\_CAUSED\_BY\_USER =

TABLES

t\_outtab = it\_camine

\* EXCEPTIONS

\* PROGRAM\_ERROR = 1

\* OTHERS = 2

.

IF sy-subrc <> 0.

\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO

\* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

ENDIF.

"Afisarea catorva campuri din tabela ZSS\_camine

\* WRITE: / 'Universitatea al carui cod l-ai introdus ofera studentilor urmatoarele camine: '.

\* SKIP.

\* WRITE: sy-uline(30).

\* WRITE: / '|', 'Cod camin', AT 15 '|','Nume camin', AT 30 '|'.

\* SELECT cod\_camin nume\_camin FROM zss\_camine INTO CORRESPONDING FIELDS OF gs\_camine

\* WHERE cod\_universitate = p\_u.

\* WRITE: / sy-uline(30).

\* WRITE: / '|',gs\_camine-cod\_camin UNDER 'Cod\_camin', AT 15 '|',gs\_camine-nume\_camin UNDER 'Nume camin', AT 30'|'.

\* ENDSELECT.

\* WRITE: / sy-uline(30).

ENDIF.

"cauta student si il afiseaza intr-o tabela interna atunci cand este apasat butonul p\_cauta

IF p\_cauta = 'X'.

SELECT SINGLE \*

FROM zss\_student INTO gs\_student

WHERE cod\_student = p\_cod\_s.

WRITE: gs\_student-cod\_student,

gs\_student-cod\_facultate,

gs\_student-cod\_universitate,

gs\_student-nume,

gs\_student-prenume,

gs\_student-data\_nasterii,

gs\_student-sex,

gs\_student-media.

ENDIF.

"afiseaza situatia repartizarilor studentilor in camine folosind alv, in functie de anul de studiu introdus in parametru atunci cand este apasat butonul p\_afis

IF p\_afis = 'X'.

SELECT \* FROM zss\_repartizari INTO TABLE it\_view WHERE an\_studiu = p\_an.

CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'

EXPORTING

\* I\_INTERFACE\_CHECK = ' '

\* I\_BYPASSING\_BUFFER =

\* I\_BUFFER\_ACTIVE = ' '

\* I\_CALLBACK\_PROGRAM = ' '

\* I\_CALLBACK\_PF\_STATUS\_SET = ' '

\* I\_CALLBACK\_USER\_COMMAND = ' '

i\_structure\_name = 'zss\_repartizari'

\* IS\_LAYOUT =

\* IT\_FIELDCAT =

\* IT\_EXCLUDING =

\* IT\_SPECIAL\_GROUPS =

\* IT\_SORT =

\* IT\_FILTER =

\* IS\_SEL\_HIDE =

\* I\_DEFAULT = 'X'

\* I\_SAVE = ' '

\* IS\_VARIANT =

\* IT\_EVENTS =

\* IT\_EVENT\_EXIT =

\* IS\_PRINT =

\* IS\_REPREP\_ID =

\* I\_SCREEN\_START\_COLUMN = 0

\* I\_SCREEN\_START\_LINE = 0

\* I\_SCREEN\_END\_COLUMN = 0

\* I\_SCREEN\_END\_LINE = 0

\* IR\_SALV\_LIST\_ADAPTER =

\* IT\_EXCEPT\_QINFO =

\* I\_SUPPRESS\_EMPTY\_DATA = ABAP\_FALSE

\* IMPORTING

\* E\_EXIT\_CAUSED\_BY\_CALLER =

\* ES\_EXIT\_CAUSED\_BY\_USER =

TABLES

t\_outtab = it\_view

\* EXCEPTIONS

\* PROGRAM\_ERROR = 1

\* OTHERS = 2

.

IF sy-subrc <> 0.

\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO

\* WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.

ENDIF.

ENDIF.

"Insereaza un student nou in baza de date, apoi foloseste o tabela interna pentru a afisa tot ce se afla acum in baza de date folosind append pt inserarea

"tuturor randurilor existente acum in baza de date

IF p\_insert = 'X'.

wa\_student-cod\_student = p\_cod\_s.

wa\_student-cod\_facultate = p\_cod\_f.

wa\_student-cod\_universitate = p\_cod\_u.

wa\_student-nume = p\_nume.

wa\_student-prenume = p\_prenum.

wa\_student-data\_nasterii = p\_datan.

wa\_student-sex = p\_sex.

wa\_student-media = p\_media.

INSERT zss\_student FROM wa\_student.

SELECT cod\_student nume prenume data\_nasterii sex media FROM zss\_student INTO CORRESPONDING FIELDS OF

gf\_student.

APPEND gf\_student TO gt\_student.

WRITE:/ gf\_student-cod\_student, gf\_student-nume, gf\_student-prenume, gf\_student-data\_nasterii,

gf\_student-sex, gf\_student-media.

ENDSELECT.

ENDIF.

"Sterge studentul al carui cod este introdus, din baza de date

IF p\_delete = 'X'.

DELETE FROM zss\_student WHERE cod\_student = p\_cod\_s.

ENDIF.

"Modifica media studentului al carui cod este introdus de la tastatura atunci cand este apasat butonul p\_modif, dupa care face o selectie din baza de date

"intr-o tabela interna folosind insert pe tabela interna a tututor randurilor din tabela

IF p\_modif = 'X'.

UPDATE zss\_student SET media = p\_media WHERE cod\_student = p\_cod\_s.

WRITE: 'S-a modificat media studentului cu codul:',p\_cod\_s.

SKIP.

SKIP.

SELECT SINGLE \* FROM zss\_student INTO CORRESPONDING FIELDS OF ls\_student WHERE cod\_student = p\_cod\_s.

WRITE: / ls\_student-nume, ls\_student-prenume, ls\_student-media.

WRITE: / 'Dupa modificarile facute, noua lista a studentilor este:'.

SKIP.

SKIP.

SKIP.

SELECT \* FROM zss\_student INTO CORRESPONDING FIELDS OF ls\_student.

INSERT ls\_student INTO TABLE lt\_student.

WRITE: / ls\_student-cod\_student,

ls\_student-nume,

ls\_student-prenume,

ls\_student-data\_nasterii,

ls\_student-sex,

ls\_student-media.

ENDSELECT.

ENDIF.

"atunci cand este apasat butonul p\_c, imi ia textul care are in loc de spatiu '-'din data space si imi inlocuieste fiecare - cu un spatiu, apoi il afiseaza

"pe ecran

"imi insereaza in tabela interna inregistrari duplicate din tabela zss\_universitate, dupa care imi sterge din tabela interna inregistrarea care are cod universitate

"introdus, si imi sterge inregistrarile duplicate si afiseaza pe ecran.

IF p\_c = 'X'.

WHILE sy-subrc = 0.

REPLACE '-' WITH space INTO text.

ENDWHILE.

WRITE: / text.

SKIP.

DO 2 TIMES.

SELECT nume cod\_universitate FROM zss\_universitate INTO CORRESPONDING FIELDS OF TABLE lt\_universitate.

ENDDO.

DELETE lt\_universitate WHERE cod\_universitate = p\_u.

DELETE ADJACENT DUPLICATES FROM lt\_universitate COMPARING cod\_universitate.

LOOP AT lt\_universitate INTO ls\_universitate.

WRITE: / ls\_universitate-nume.

ENDLOOP.

ENDIF.

IF p\_number = 'X'.

CALL FUNCTION 'SPELL\_AMOUNT'

EXPORTING

amount = mynum

\* CURRENCY = ' '

\* FILLER = ' '

language = sy-langu

IMPORTING

in\_words = result

\* EXCEPTIONS

\* NOT\_FOUND = 1

\* TOO\_LARGE = 2

\* OTHERS = 3

.

IF sy-subrc <> 0.

WRITE: / sy-subrc.

ELSE.

WRITE: 'The amount in words is: ',result-word.

ENDIF.

ENDIF.

IF cal = 'X'.

CALL FUNCTION 'DATE\_GET\_WEEK'

EXPORTING

date = mydata

IMPORTING

week = week

\* EXCEPTIONS

\* DATE\_INVALID = 1

\* OTHERS = 2

.

IF sy-subrc <> 0.

WRITE sy-subrc.

ELSE.

WRITE: 'Anul si numarul saptamnii din an in format YYYYMM:', week.

ENDIF.

ENDIF.

\* -----------------------------------------------------------------------------------------\*

\* End of processing event

\* -----------------------------------------------------------------------------------------\*

END-OF-SELECTION.

\*&---------------------------------------------------------------------\*

\*& Include ZSS\_PROIECT\_FINAL3\_F01

\*&---------------------------------------------------------------------\*

\*&---------------------------------------------------------------------\*

\*& Form INITIALIZATION

\*&---------------------------------------------------------------------\*

\* text

\*----------------------------------------------------------------------\*

\* --> p1 text

\* <-- p2 text

\*----------------------------------------------------------------------\*

form INITIALIZATION .

button1 = 'Afiseaza repartizare'.

button2 = 'Universitate'.

button3 = 'Studenti'.

mytab-prog = sy-repid.

mytab-dynnr = 100.

mytab-activetab = 'PUSH1'.

endform. " INITIALIZATION