HOMEPAGE.FMX

FAMILY HISTORY

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\FAMILY HISTORY\FAMILY\_HISTORY.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

FACULTY REGISTRATION

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\FACULTY REGISTRATION\FACULTY REGISTRATION.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

STUDENT REGISTRATION

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\STUDENT REGISTRATION\STUDENT REGISTRATION.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

DIAGNOSIS

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\DIAGNOSIS\DIAGNOSIS.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

PHYSICAL EXAM

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\PHYSICAL EXAMINATION\PHYSICALEXAMINATION.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

PERSONAL CONSULTATION

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\PERSONAL CONSULTATION\PERSONAL CONSULTATION.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

DRUG

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\INSERTION\DRUG\DRUG.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

REPORT GENERATION TO EXCEL

declare

application ole2.obj\_type;

workbooks ole2.obj\_type;

workbook ole2.obj\_type;

worksheets ole2.obj\_type;

worksheet ole2.obj\_type;

cell ole2.obj\_type;

arglist ole2.list\_type;

row\_num number;

col\_num number;

row\_width number;

fontObj ole2.obj\_type;

cursor DRUGSUM is SELECT D\_NO, NAME, COST, QUANTITY, (COST \* QUANTITY) AS PRICE FROM DRUG;

cursor DIAGSUM is SELECT FIRST\_NAME, LAST\_NAME, D\_NO, D\_DATE FROM DIAGSUM;

cursor PESUM IS SELECT FIRST\_NAME, LAST\_NAME, PE\_NO, PE\_DATE FROM PESUM;

CURSOR PCSUM IS SELECT FIRST\_NAME, LAST\_NAME, PC\_NO, PC\_DATE FROM PCSUM;

procedure SetCellValue(rowid number,colid number,cellValue varchar) is

begin

arglist := ole2.create\_arglist;

ole2.add\_arg(arglist,rowid);

ole2.add\_arg(arglist,colid);

cell:= ole2.get\_obj\_property(worksheet,'Cells',arglist);

--row\_width := ole2.get\_obj\_property(worksheet,'WIDTH',arglist);

fontObj := ole2.get\_obj\_property(cell,'Font');

ole2.destroy\_arglist(arglist);

ole2.set\_property(cell,'value',cellValue);

ole2.set\_property(fontObj,'Size',7);

ole2.set\_property(fontObj,'',1); --<- ADD STYLER

ole2.set\_property(fontObj,'ColorIndex',1); --SET 1 FOR BLACK

ole2.release\_obj(cell);

end SetCellValue;

procedure app\_init is

begin

application := ole2.create\_obj('Excel.Application');

ole2.set\_property(application,'Visible',true);

workbooks := ole2.get\_obj\_property(application,'workbooks');

workbook := ole2.invoke\_obj(workbooks,'add');

worksheets := ole2.get\_obj\_property(application,'worksheets');

worksheet := ole2.invoke\_obj(worksheets,'add');

ole2.set\_property(worksheet,'Name','DRUG SHEET');

end app\_init;

procedure save\_excel(path varchar,filename varchar) is

begin

OLE2.Release\_Obj(worksheet);

OLE2.Release\_Obj(worksheets);

-- Save the Excel file created

If path is not null then

Arglist := OLE2.Create\_Arglist;

OLE2.Add\_Arg(Arglist,path||'\'||file\_name||'.xls');

OLE2.Invoke(workbook, 'SaveAs', Arglist);

OLE2.Destroy\_Arglist(Arglist);

end if;

end save\_excel;

BEGIN

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

IF :DB\_HOMEPAGE.LST\_GENERATE= 'A' THEN

MESSAGE ('Physical Exam Summary');

app\_init;

row\_num:=1;

col\_num:=1;

SetCellValue(row\_num,col\_num,'GIVEN NAME');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'FAMILY NAME');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'PE\_NO');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'DATE');

for i in PESUM loop

row\_num:=row\_num + 1;

col\_num:=1;

SetCellValue(row\_num,col\_num,i.first\_name);

col\_num:=2;

SetCellValue(row\_num,col\_num,i.last\_name);

col\_num:=3;

SetCellValue(row\_num,col\_num,i.PE\_NO);

col\_num:=4;

SetCellValue(row\_num,col\_num,i.PE\_DATE);

end loop;

save\_excel('d:\excel\_export','emp\_data');

OLE2.Release\_Obj(workbook);

OLE2.Release\_Obj(workbooks);

OLE2.Release\_Obj(application);

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

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

ELSIF :DB\_HOMEPAGE.LST\_GENERATE= 'B' THEN

MESSAGE ('Personal Consultation Summary');

app\_init;

row\_num:=1;

col\_num:=1;

SetCellValue(row\_num,col\_num,'FIRST');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'LAST');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'PC\_NO');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'DATE');

for i in PCSUM loop

row\_num:=row\_num + 1;

col\_num:=1;

SetCellValue(row\_num,col\_num,i.first\_name);

col\_num:=2;

SetCellValue(row\_num,col\_num,i.last\_name);

col\_num:=3;

SetCellValue(row\_num,col\_num,i.PC\_NO);

col\_num:=4;

SetCellValue(row\_num,col\_num,i.PC\_DATE);

end loop;

save\_excel('d:\excel\_export','emp\_data');

OLE2.Release\_Obj(workbook);

OLE2.Release\_Obj(workbooks);

OLE2.Release\_Obj(application);

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

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

ELSIF :DB\_HOMEPAGE.LST\_GENERATE= 'C' THEN

MESSAGE ('Diagnosis Summary');

app\_init;

row\_num:=1;

col\_num:=1;

SetCellValue(row\_num,col\_num,'GIVEN NAME');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'FAMILY NAME');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'D NO');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'DATE');

for i in DIAGSUM loop

row\_num:=row\_num + 1;

col\_num:=1;

SetCellValue(row\_num,col\_num,i.first\_name);

col\_num:=2;

SetCellValue(row\_num,col\_num,i.last\_name);

col\_num:=3;

SetCellValue(row\_num,col\_num,i.D\_NO);

col\_num:=4;

SetCellValue(row\_num,col\_num,i.D\_DATE);

end loop;

save\_excel('d:\excel\_export','emp\_data');

OLE2.Release\_Obj(workbook);

OLE2.Release\_Obj(workbooks);

OLE2.Release\_Obj(application);

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

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

ELSIF :DB\_HOMEPAGE.LST\_GENERATE= 'D' THEN

MESSAGE ('Drug Use Summary');

app\_init;

row\_num:=1;

col\_num:=1;

SetCellValue(row\_num,col\_num,'DRUG NO.');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'NAME');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'COST');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'QUANTITY');

col\_num:=col\_num + 1;

SetCellValue(row\_num,col\_num,'PRICE');

for i in DRUGSUM loop

row\_num:=row\_num + 1;

col\_num:=1;

SetCellValue(row\_num,col\_num,i.d\_no);

col\_num:=2;

SetCellValue(row\_num,col\_num,i.name);

col\_num:=3;

SetCellValue(row\_num,col\_num,i.cost);

col\_num:=4;

SetCellValue(row\_num,col\_num,i.quantity);

col\_num:=5;

SetCellValue(row\_num,col\_num,i.quantity \* i.cost);

end loop;

save\_excel('d:\excel\_export','emp\_data');

OLE2.Release\_Obj(workbook);

OLE2.Release\_Obj(workbooks);

OLE2.Release\_Obj(application);

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

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

ELSE

MESSAGE('PLEASE SELECT A REPORT THAT YOU WANT TO GENERATE');

END IF;

END;

VIEWING OF SPECIFIC DATA

BEGIN

IF :DB\_HOMEPAGE.LST\_PATIENT= 'A' THEN

MESSAGE ('Physical Examination');

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\VIEW UPDATE DELETE\PHYSICAL EXAMINATION\PHYSICALEXAMINATION.fmx');

EXIT\_FORM('NO VALIDATE');

ELSIF :DB\_HOMEPAGE.LST\_PATIENT= 'B' THEN

MESSAGE ('Diagnosis');

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\VIEW UPDATE DELETE\DIAGNOSIS\DND.fmx');

EXIT\_FORM('NO VALIDATE');

ELSIF :DB\_HOMEPAGE.LST\_PATIENT= 'C' THEN

MESSAGE ('Family History');

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\VIEW UPDATE DELETE\FAMILY HISTORY\FAMILY HISTORY.fmx');

EXIT\_FORM('NO VALIDATE');

ELSIF :DB\_HOMEPAGE.LST\_PATIENT= 'D' THEN

MESSAGE ('Personal Consultation');

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\VIEW UPDATE DELETE\PERSONAL CONSULTATION\CONSULTATION.fmx');

EXIT\_FORM('NO VALIDATE');

ELSIF :DB\_HOMEPAGE.LST\_PATIENT= 'E' THEN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\VIEW UPDATE DELETE\PERSON INFORMATION\GENERAL INFORMATION.fmx');

EXIT\_FORM('NO VALIDATE');

ELSE

MESSAGE('PLEASE SELECT A REPORT THAT YOU WANT TO VIEW');

END IF;

END;

LOGOUT

BEGIN

NEW\_FORM ('C:\Users\Person\Desktop\ORACLE PROJECT\LOGIN.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

LOGIN.FMX

LOGIN

DECLARE

V VARCHAR(90) ;

BEGIN

SELECT 'X' INTO V FROM ACCOUNT

WHERE

LOWER (USERNAME) = LOWER(:DB\_LOGIN.TB\_UN)

AND

PASSWORD = :DB\_LOGIN.TB\_PW;

IF SQL%FOUND THEN

BEGIN

NEW\_FORM ('C:\Users\Person\Desktop\ORACLE PROJECT\HOMEPAGE.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

END IF;

EXCEPTION

WHEN

NO\_DATA\_FOUND THEN

MESSAGE ('USERNAME OR PASSWORD IS INCORRECT!');

END;

REGISTER

BEGIN

NEW\_FORM ('C:\Users\Person\Desktop\ORACLE PROJECT\USERS.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

USERS.FMX

CANCEL

BEGIN

NEW\_FORM ('C:\Users\Person\Desktop\ORACLE PROJECT\USERS.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;

SUBMIT

BEGIN

IF :DB\_USERS.TB\_PW = :DB\_USERS.TB\_CPW THEN

INSERT INTO PERSONINCHARGE VALUES (:DB\_USERS.TB\_PICID,:DB\_USERS.TB\_GN,:DB\_USERS.TB\_FN,:DB\_USERS.TB\_POS);

INSERT INTO ACCOUNT VALUES (:DB\_USERS.TB\_PICID,:DB\_USERS.TB\_UN,:DB\_USERS.TB\_PW);

:system.message\_level := '5';

commit;

:system.message\_level := '0';

MESSAGE('ACCOUNT FOR '||UPPER(:DB\_USERS.TB\_GN)||' IS NOW CREATED');

:DB\_USERS.TB\_PICID:='';

:DB\_USERS.TB\_FN:='';

:DB\_USERS.TB\_GN:='';

:DB\_USERS.TB\_UN:='';

:DB\_USERS.TB\_PW:='';

:DB\_USERS.TB\_CPW:='';

ELSE

MESSAGE('THE PASSWORD AND CONFIRM PASSWORD SHOULD BE THE SAME');

END IF;

END;

DIAGNOSIS.FMX

SUBMIT

BEGIN

INSERT INTO DIAGNOSIS VALUES

(

:DB\_DIAGNOSIS.TB\_PID,

:DB\_DIAGNOSIS.TB\_DESCRIPTION,

:DB\_DIAGNOSIS.TB\_DNO

);

INSERT INTO DIAGNOSISHISTORY VALUES

(

:DB\_DIAGNOSIS.TB\_DNO,

SYSDATE,

:DB\_DIAGNOSIS.TB\_PICID

);

:SYSTEM.MESSAGE\_LEVEL:= '5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL:='0';

MESSAGE('DIAGNOSIS RECORD SUBMITTED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

DRUG.FMX

SUBMIT

BEGIN

INSERT INTO DRUG VALUES

(

:DB\_DRUG.TB\_DNO,

:DB\_DRUG.TB\_NAME,

:DB\_DRUG.TB\_CST,

:DB\_DRUG.TB\_QTY

);

:SYSTEM.MESSAGE\_LEVEL:='5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL:='0';

MESSAGE('DRUG PRESCRIBED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

FACULTY REGISTRATION.FMX

SUBMIT

BEGIN

INSERT INTO CLINIC VALUES

(

:DB\_REG.TB\_PID,

'2',

:DB\_REG.TB\_PFN,

:DB\_REG.TB\_PLN

);

INSERT INTO INFORMATION VALUES

(

:DB\_REG.TB\_PID,

:DB\_REG.TB\_GID,

:DB\_REG.TB\_PMOBILE,

:DB\_REG.TB\_GENDER,

:DB\_REG.TB\_NATIONALITY,

--TO\_CHAR(TO\_DATE(:DB\_REG.TB\_BIRTHDATE,'DD-MON-YYYY'),'DD/MM/YYYY'),

TO\_DATE(:DB\_REG.TB\_BIRTHDATE,'DD/MM/YYYY'),

:DB\_REG.TB\_CITY,

:DB\_REG.TB\_CAMPUS

);

INSERT INTO GUARDIAN VALUES

(

:DB\_REG.TB\_GID,

:DB\_REG.TB\_GFN,

:DB\_REG.TB\_GLN,

:DB\_REG.TB\_GMOBILE,

:DB\_REG.TB\_RELATION

);

--add a type I.D in here.

--[urgent update, remove the TB\_TYPE IN ORDER TO SOLVE THE PROBLEM]

INSERT INTO FACULTY VALUES

(

:DB\_REG.TB\_PID,

'2',

:DB\_REG.TB\_CID

);

:SYSTEM.MESSAGE\_LEVEL:='5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL:='0';

MESSAGE('FACULTY INFORMATION CREATED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

FAMILY HISTORY.FMX

SUBMIT

DECLARE

VAR\_RELATION VARCHAR(99);

CURSOR CUR\_PRACTICAL IS SELECT RELATION FROM FAMILYHISTORY

WHERE PERSON\_ID = 33433;

VAR\_CF BOOLEAN:=FALSE;

VAR\_CM BOOLEAN:=FALSE;

BEGIN

OPEN CUR\_PRACTICAL;

LOOP

FETCH CUR\_PRACTICAL INTO VAR\_RELATION;

IF VAR\_RELATION = 'FATHER' THEN

VAR\_CF := TRUE;

ELSIF VAR\_RELATION = 'MOTHER' THEN

VAR\_CM:= TRUE;

END IF;

EXIT WHEN CUR\_PRACTICAL%NOTFOUND;

END LOOP;

CLOSE CUR\_PRACTICAL;

IF

:DB\_FAMILYHISTORY.TB\_RELATION='FATHER' AND VAR\_CF = TRUE

THEN

MESSAGE('ONLY 1 FATHER CAN BE INSERTED');

ELSIF

:DB\_FAMILYHISTORY.TB\_RELATION='MOTHER' AND VAR\_CM = TRUE

THEN

MESSAGE('ONLY 1 MOTHER CAN BE INSERTED');

ELSE

INSERT INTO FAMILYHISTORY VALUES (:DB\_FAMILYHISTORY.TB\_RELATION,:DB\_FAMILYHISTORY.TB\_PID,:DB\_FAMILYHISTORY.TB\_LN,:DB\_FAMILYHISTORY.TB\_FN,:DB\_FAMILYHISTORY.TB\_ILLNESS);

:SYSTEM.MESSAGE\_LEVEL:='5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL:='0';

MESSAGE('FAMILY HISTORY INSERTED');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

PERSONAL CONSULTATION.FMX

SUBMIT

BEGIN

INSERT INTO PERSONALCONSULTATION VALUES

(

:DB\_CONSULTATION.TB\_PID,

:DB\_CONSULTATION.TB\_PCNO,

:DB\_CONSULTATION.TB\_COMPLAINTS,

:DB\_CONSULTATION.TB\_FINDINGS,

:DB\_CONSULTATION.TB\_ASSESSMENT,

:DB\_CONSULTATION.TB\_PLANS

);

INSERT INTO PERSONALCONSULTATIONHISTORY VALUES

(

:DB\_CONSULTATION.TB\_PCNO,

SYSDATE,

:DB\_CONSULTATION.TB\_PICID

);

:SYSTEM.MESSAGE\_LEVEL := '5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL :='0';

MESSAGE('SUCCESSFULLY RECORDED!');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

PHYSICALEXAMINATION.FMX

SUBMIT

BEGIN

INSERT INTO PHYSICALEXAM VALUES (:DB\_PHYSICALEXAMINATION.TB\_PID, :DB\_PHYSICALEXAMINATION.TB\_PENO, :DB\_PHYSICALEXAMINATION.TB\_FT, :DB\_PHYSICALEXAMINATION.TB\_IN, :DB\_PHYSICALEXAMINATION.TB\_RATE, :DB\_PHYSICALEXAMINATION.TB\_SYS, :DB\_PHYSICALEXAMINATION.TB\_DIA, :DB\_PHYSICALEXAMINATION.TB\_IRREG, :DB\_PHYSICALEXAMINATION.TB\_DIAG);

INSERT INTO PHYSICALEXAMHISTORY

VALUES

(

:DB\_PHYSICALEXAMINATION.TB\_PENO,

SYSDATE,

:DB\_PHYSICALEXAMINATION.TB\_PICID

);

:system.message\_level := '5';

-- to suppress all messages with severity below level 5.

commit;

:system.message\_level := '0';

MESSAGE ('SUCCESSFULLY INSERTED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN MESSAGE('NO DATA FOUND TO SOME FIELDS');

WHEN TOO\_MANY\_ROWS THEN MESSAGE('TOO MANY ROWS ARE BEING RETURNED!');

WHEN OTHERS THEN MESSAGE ('COULD NOT IDENTIFY THE VALUE OR NO VALUE INSERTED DURING REGISTRATION PHASE!');

END;

--INSERT INTO PHYSICALEXAMHISTORY VALUES (:DB\_PHYSICALEXAMINATION.TB\_PENO,TO\_DATE('2003/07/09', 'mm/dd/yyyy'), :DB\_PHYSICALEXAMINATION.TB\_PICID);

BEGIN

INSERT INTO CLINIC VALUES

(

:DB\_REG.TB\_PID,

'1',

:DB\_REG.TB\_PFN,

:DB\_REG.TB\_PLN

);

INSERT INTO INFORMATION VALUES

(

:DB\_REG.TB\_PID,

:DB\_REG.TB\_GID,

:DB\_REG.TB\_PMOBILE,

:DB\_REG.TB\_GENDER,

:DB\_REG.TB\_NATIONALITY,

--TO\_CHAR(TO\_DATE(:DB\_REG.TB\_BIRTHDATE,'DD-MON-YYYY'),'DD/MM/YYYY'),

TO\_DATE(:DB\_REG.TB\_BIRTHDATE,'DD/MM/YYYY'),

:DB\_REG.TB\_CITY,

:DB\_REG.TB\_CAMPUS

);

INSERT INTO GUARDIAN VALUES

(

:DB\_REG.TB\_GID,

:DB\_REG.TB\_GFN,

:DB\_REG.TB\_GLN,

:DB\_REG.TB\_GMOBILE,

:DB\_REG.TB\_RELATION

);

--add a type I.D in here.

--[urgent update, remove the TB\_TYPE IN ORDER TO SOLVE THE PROBLEM]

INSERT INTO STUDENT VALUES

(

:DB\_REG.TB\_PID,

:DB\_REG.TB\_CID,

'1',

:DB\_REG.TB\_YR

);

:SYSTEM.MESSAGE\_LEVEL:='5';

COMMIT;

:SYSTEM.MESSAGE\_LEVEL:='0';

MESSAGE('STUDENT INFORMATION CREATED');

EXCEPTION WHEN OTHERS THEN MESSAGE('PLEASE INPUT A VALID VALUE');

END;

CANCEL

BEGIN

NEW\_FORM('C:\Users\Person\Desktop\ORACLE PROJECT\HOMEPAGE.fmx');

EXIT\_FORM(NO\_VALIDATE);

END;