 **INTERNSHIP BATCH 2022 (ORACLE APPS)**

**BATCH-2**

**HOSPITAL MANAGEMENT SYSTEM PROJECT**

MENTOR: WILLIAM LAWRENCE

TEAM MEMBERS

1. Kuruba Akhil
2. Shadab Salim Mulla
3. Koyena Chakraborty
4. Prajwal
5. Bapusaheb Patil
6. Sv Sai Subhash
7. Selvan senthil
8. Devi VaraPrasad
9. Roopasri Pokuri

**CONTENTS :**

* AIM
* OBJECTIVE
* REPORT

**AIM**

* This software helps in handling the patients admissions with great ease.
* Allows hospitals to add their bills, bed charges, arrange the admission dates and discharge patients without any fuss.

**OBJECTIVE**

* Provide MIS (Management Information System) report on demand to management for better decision making.
* Design a system for better patient care. Reduce hospital operating costs.
* This is indeed the best dedicated software for improving the functioning for Clinics, Hospitals and Doctors

**ASSUMPTION:**

* This design assumes that the following statements are true: •
* List of Patients given by the customer will not change very frequently as this contains pre-created list of patients •

* Penalties are calculated on Indian Rupees as the software is assumed to be used only in India

**SOFTWARE REQUIREMENT**

1. Oracle Forms 10g
2. Oracle Reports 10g.
3. Oracle Database Server 11g

CREATE TABLE PATIENT\_MASTER\_TEAM2(PATIENT\_ID NUMBER(10) PRIMARY KEY,PATIENT\_NAME VARCHAR2(30),BLOOD\_GROUP VARCHAR2(3),DATE\_OF\_FIRST\_VISIT DATE,AGE NUMBER(3),GENDER CHAR(1),SENIOR\_CITIZEN CHAR(1),CREATION\_DATE DATE,CREATED\_BY NUMBER(15),LAST\_UPDATE\_DATE DATE,LAST\_UPDATED\_BY NUMBER(15),LAST\_UPDATE\_LOGIN NUMBER(15));

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

CREATE TABLE CONTACT\_DETAILS\_TEAM2(PATIENT\_ID NUMBER(10), MOBILE VARCHAR2(15), HOME VARCHAR2(15), ADDRESS VARCHAR2(50), CITY\_PIN VARCHAR2(20),FOREIGN KEY(PATIENT\_ID) REFERENCES PATIENT\_MASTER\_TEAM2(PATIENT\_ID) );

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

CREATE TABLE PATIENT\_HISTORY\_TEAM2(BILL\_NUMBER NUMBER(10) PRIMARY KEY,PATIENT\_ID NUMBER(10),STATUS\_CODE VARCHAR2(1),ATTENDER\_NAME VARCHAR2(30),VISIT\_NUMBER NUMBER(4),SYMPTOM\_ISSUE VARCHAR2(50),CONSULTED\_DR VARCHAR2(30),DIAGNOSIS VARCHAR2(50),ADMITTED\_DATE DATE,DISCHARGED\_DATE DATE,BILL\_AMOUNT NUMBER(9,2),CREATION\_DATE DATE,CREATED\_BY NUMBER(15),LAST\_UPDATE\_DATE DATE,LAST\_UPDATED\_BY NUMBER(15),LAST\_UPDATE\_LOGIN NUMBER(15), FOREIGN KEY(PATIENT\_ID) REFERENCES PATIENT\_MASTER\_TEAM2(PATIENT\_ID) );

==========================================================

GRANT ALL ON PATIENT\_MASTER\_TEAM2 TO apps;

GRANT ALL ON CONTACT\_DETAILS\_TEAM2 TO apps;

GRANT ALL ON PATIENT\_HISTORY\_TEAM2 TO apps;

=========================================================

CREATE SYNONYM PATIENT\_HISTORY\_TEAM2 FOR hr.PATIENT\_MASTER\_TEAM2;

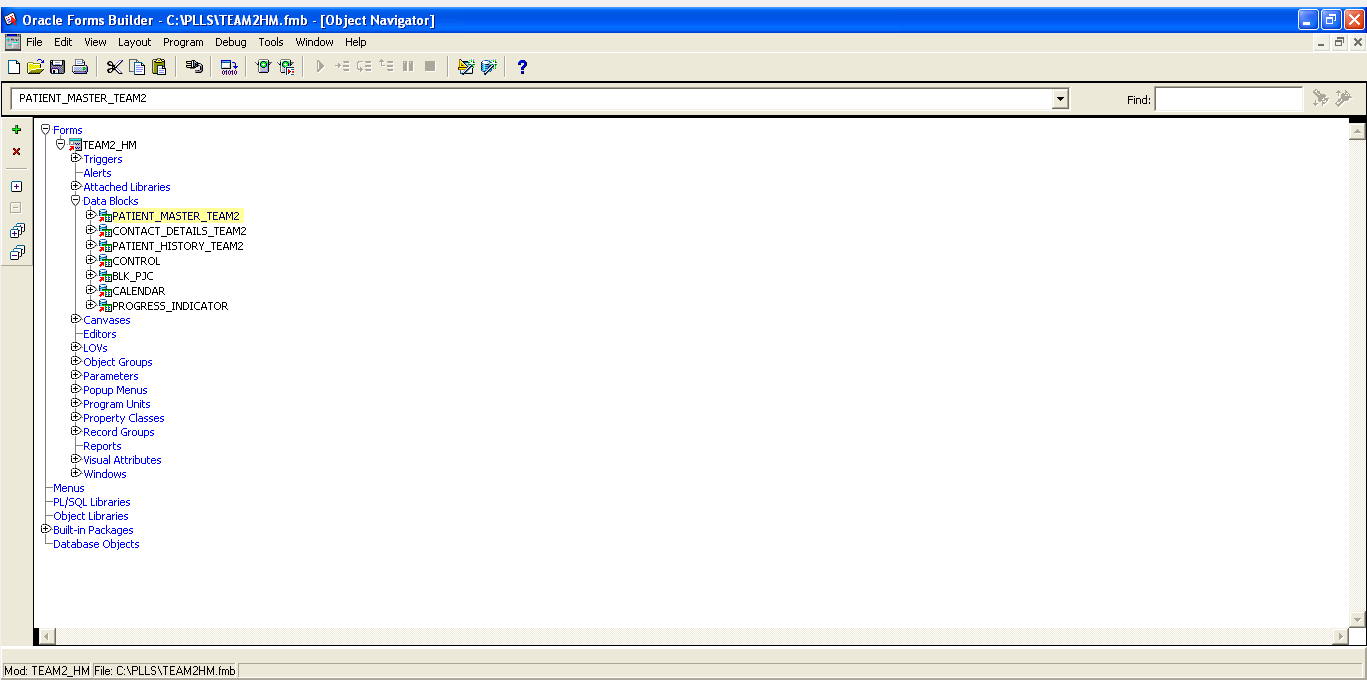
CREATE SYNONYMPATIENT\_MASTER\_TEAM2 FOR hr.PATIENT\_MASTER\_TEAM2;

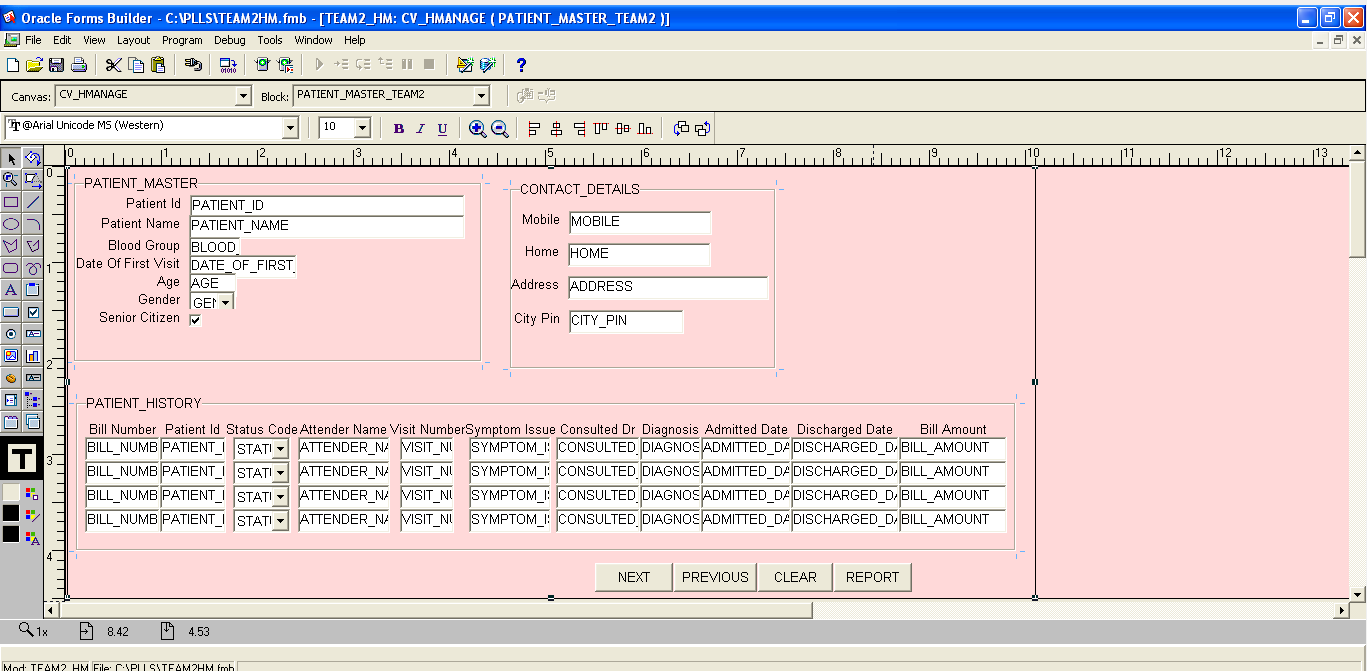
CREATE SYNONYM CONTACT\_DETAILS\_TEAM2 FOR hr.CONTACT\_DETAILS\_TEAM2;

==================================================================================================

**FORM:**

**------------------**

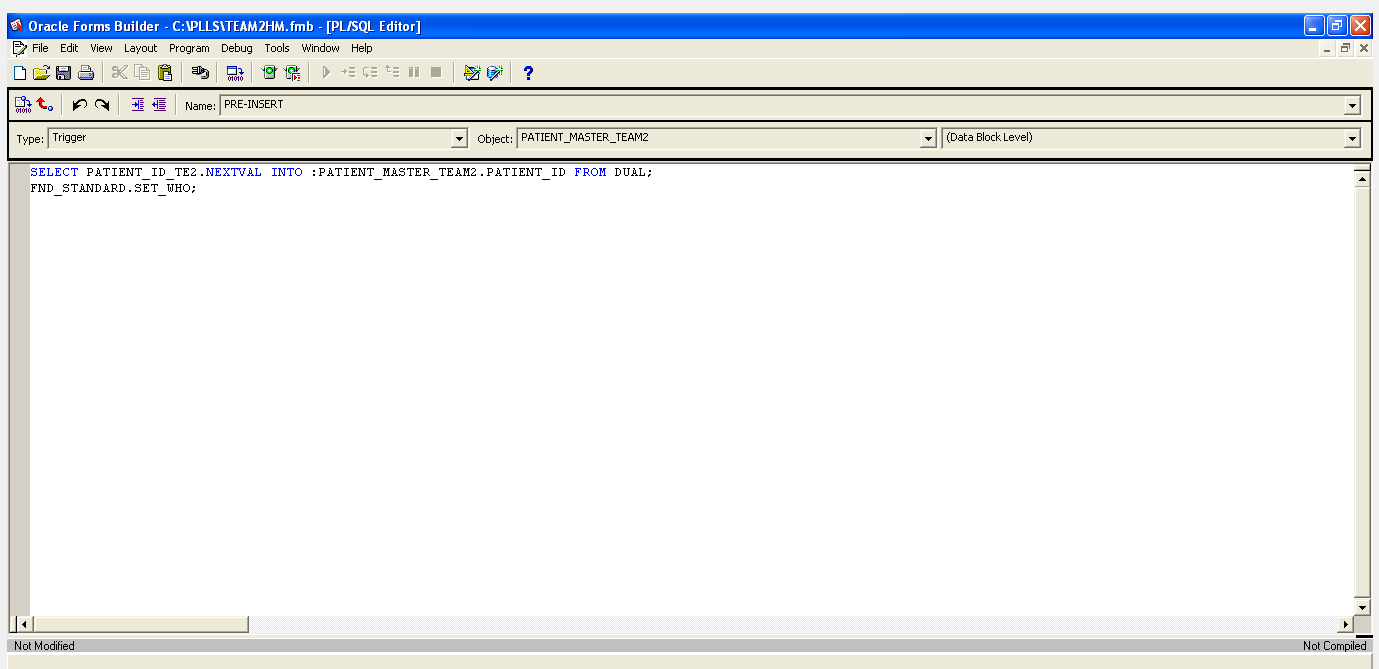




SEQUENCE:

PATIENT ID:

SELECT PATIENT\_ID\_TE2.NEXTVAL INTO :PATIENT\_MASTER\_TEAM2.PATIENT\_ID FROM DUAL;



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

AGE:

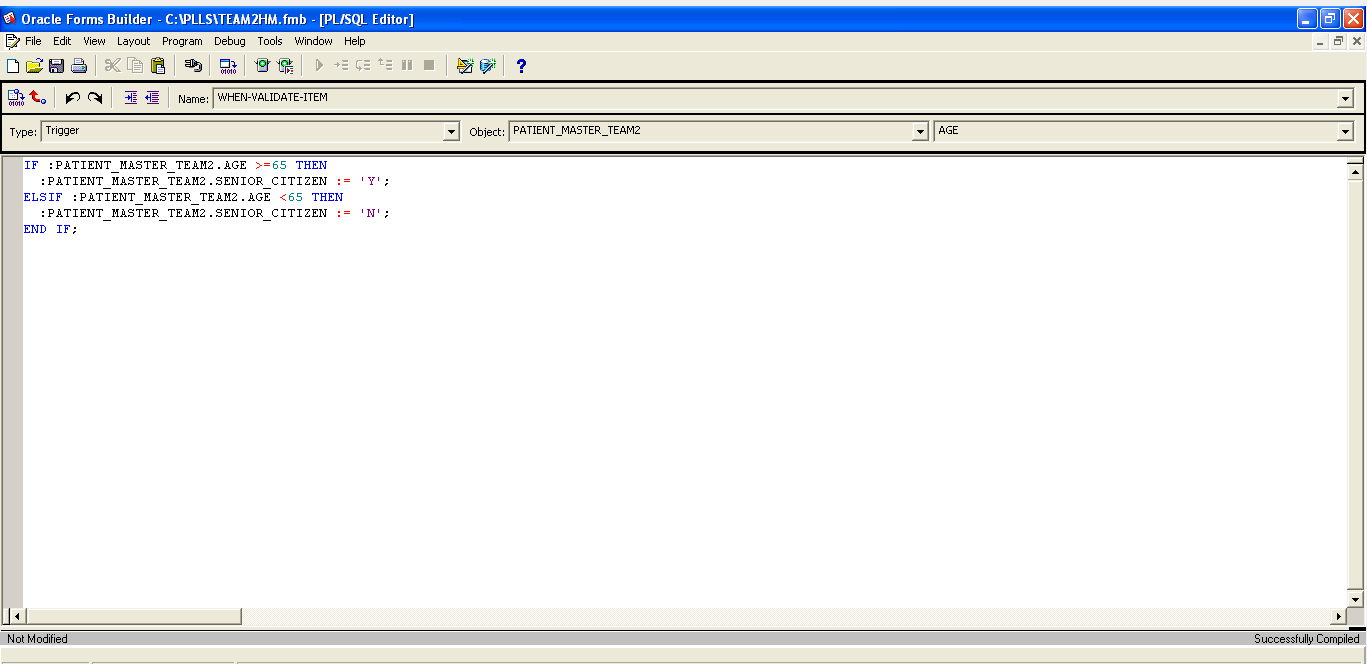
IF :PATIENT\_MASTER\_TEAM2\_.AGE >=65 THEN

:PATIENT\_MASTER\_TEAM2\_.SENIOR\_CITIZEN := 'Y';

ELSIF :PATIENT\_MASTER\_TEAM2.AGE <65 THEN

:PATIENT\_MASTER\_TEAM2.SENIOR\_CITIZEN := 'N';

END IF;



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

STATUS CODE:

IF :PATIENT\_HISTORY\_TEAM2.STATUS\_CODE='O' THEN

:PATIENT\_HISTORY\_TEAM2\_.ADMITTED\_DATE :=NULL;

:PATIENT\_HISTORY\_TEAM2\_.DISCHARGED\_DATE :=NULL;

SET\_ITEM\_PROPERTY('PATIENT\_HISTORY\_TEAM2.ADMITTED\_DATE',ENABLED,PROPERTY\_FALSE);

SET\_ITEM\_PROPERTY('PATIENT\_HISTORY\_TEAM2.DISCHARGED\_DATE',ENABLED,PROPERTY\_FALSE);

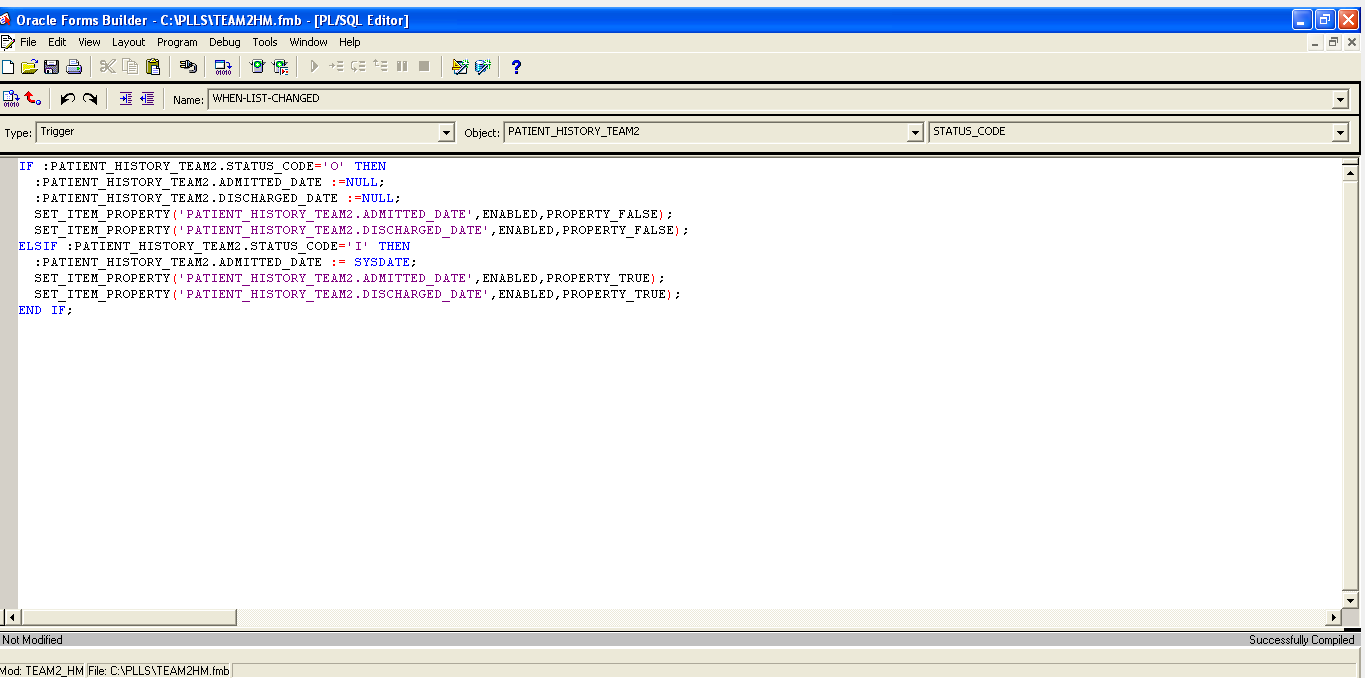
ELSIF :PATIENT\_HISTORY\_TEAM2.STATUS\_CODE='I' THEN

:PATIENT\_HISTORY\_TEAM2.ADMITTED\_DATE := SYSDATE;

SET\_ITEM\_PROPERTY('PATIENT\_HISTORY\_TEAM2.ADMITTED\_DATE',ENABLED,PROPERTY\_TRUE);

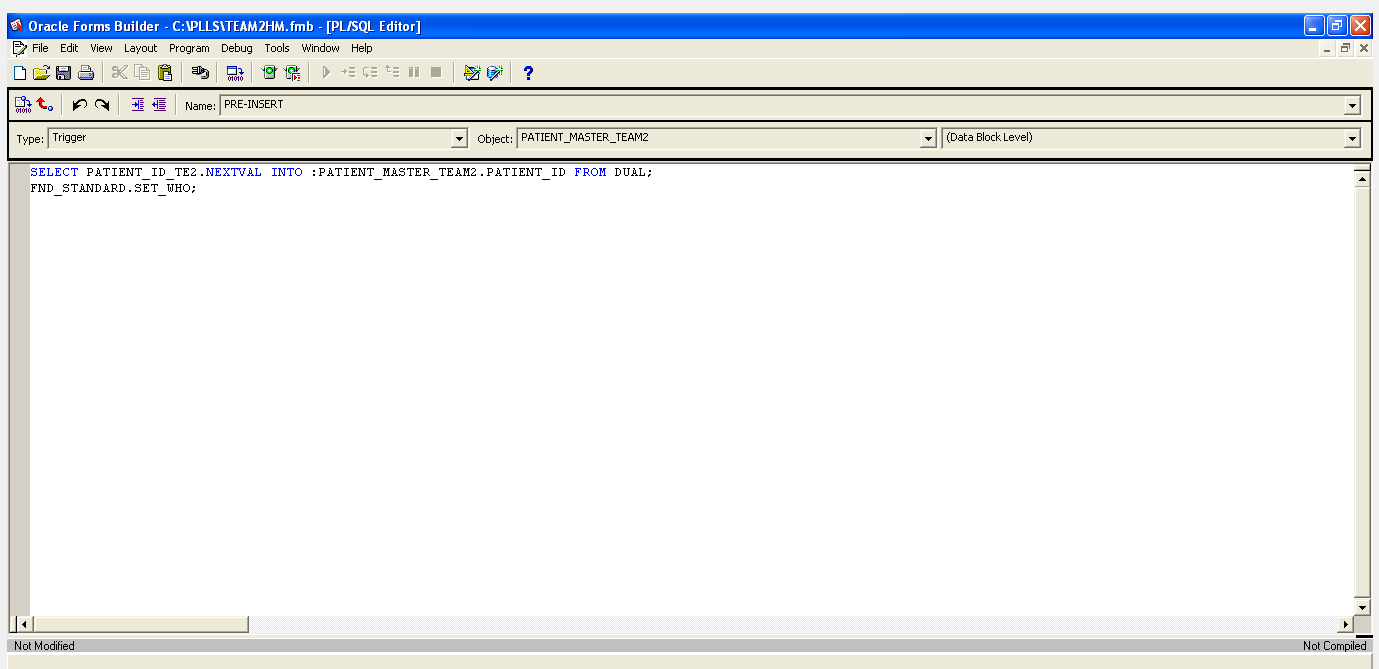
SET\_ITEM\_PROPERTY('PATIENT\_HISTORY\_TEAM2.DISCHARGED\_DATE',ENABLED,PROPERTY\_TRUE);

END IF;

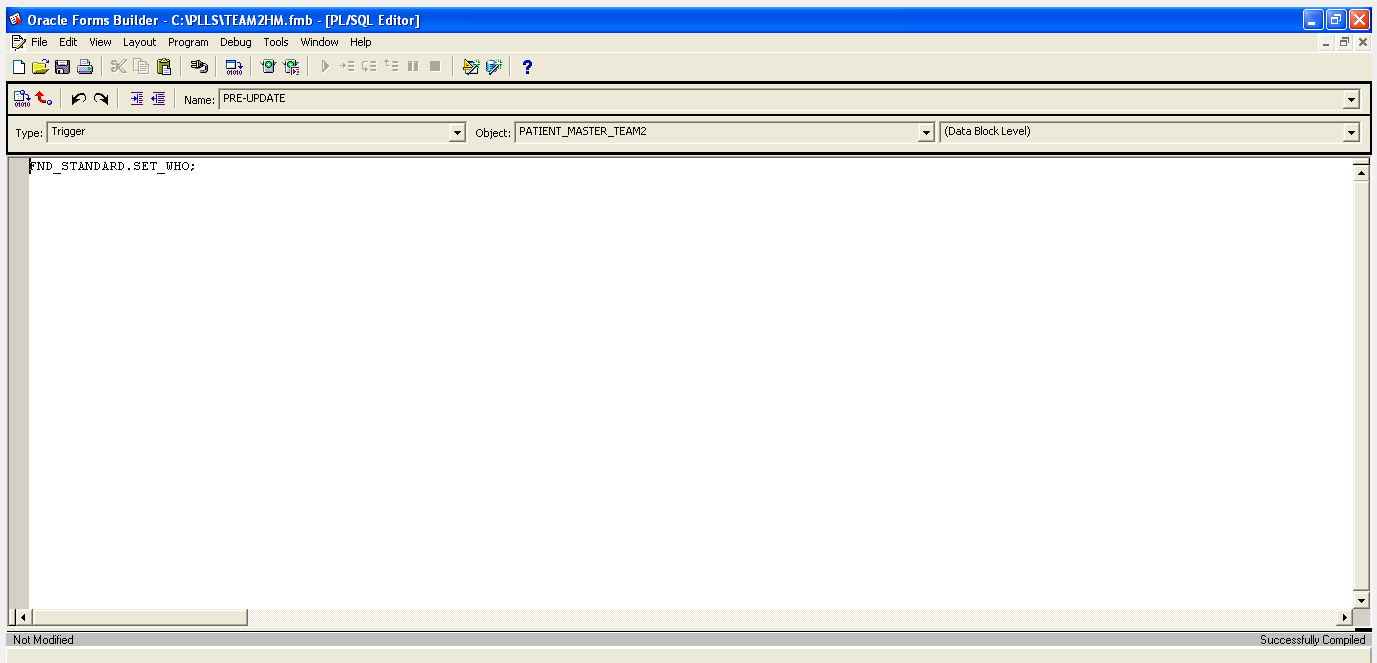


**FOR GETTING WHO COLUMNS:**

PRE-INSERT:FND\_STANDARD.SET\_WHO;



PRE-UPDATE:FND\_STANDARD.SET\_WHO;



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

**FORMS UPLOADING:**

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

pwd

ls

. APPSPROD\_ebs.env

cd $AU\_TOp/forms/US

pwd

UPLOAD THE FORM USING LOCATION

ls TEAM2HM.fmb

NOW WE WANT TO GENERATE A .fmx FILE FOR OUR FORM.

frmcmp\_batch userid=apps/apps MODULE\_TYPE=form MODULE=TEAM2HM.fmb

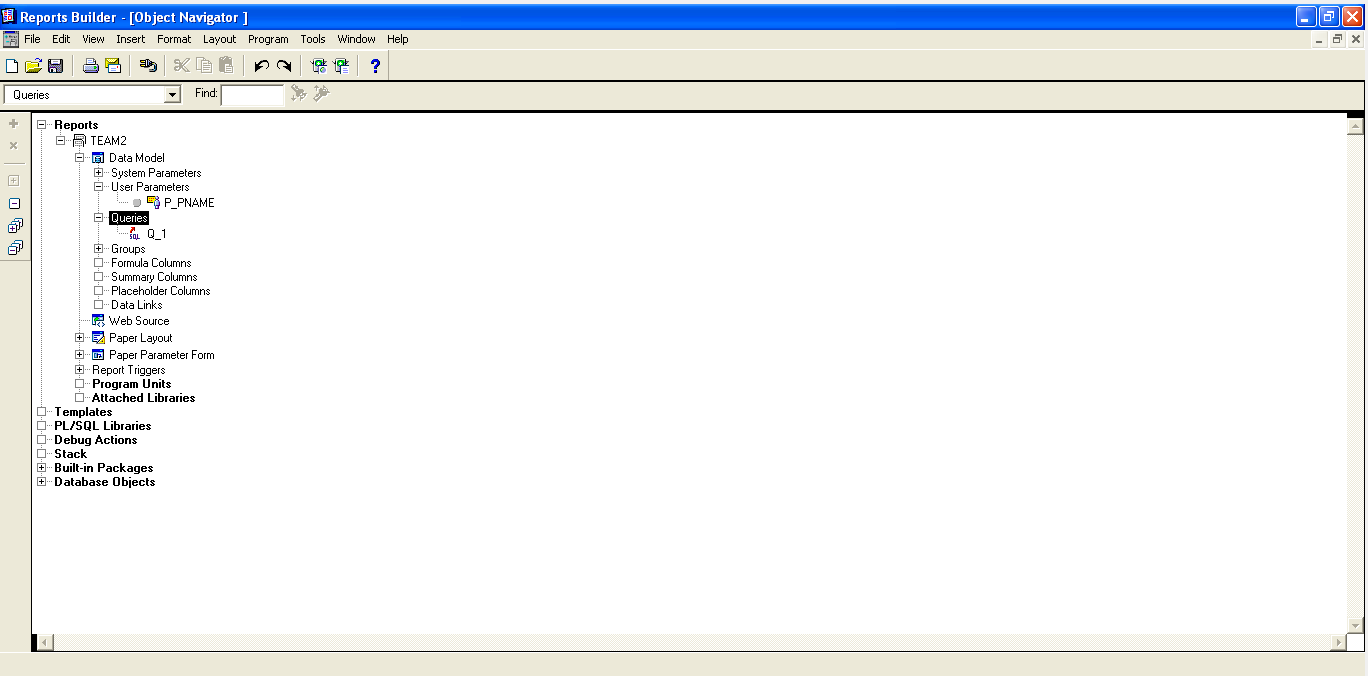
.fmx FILE WILL BE GENERATED

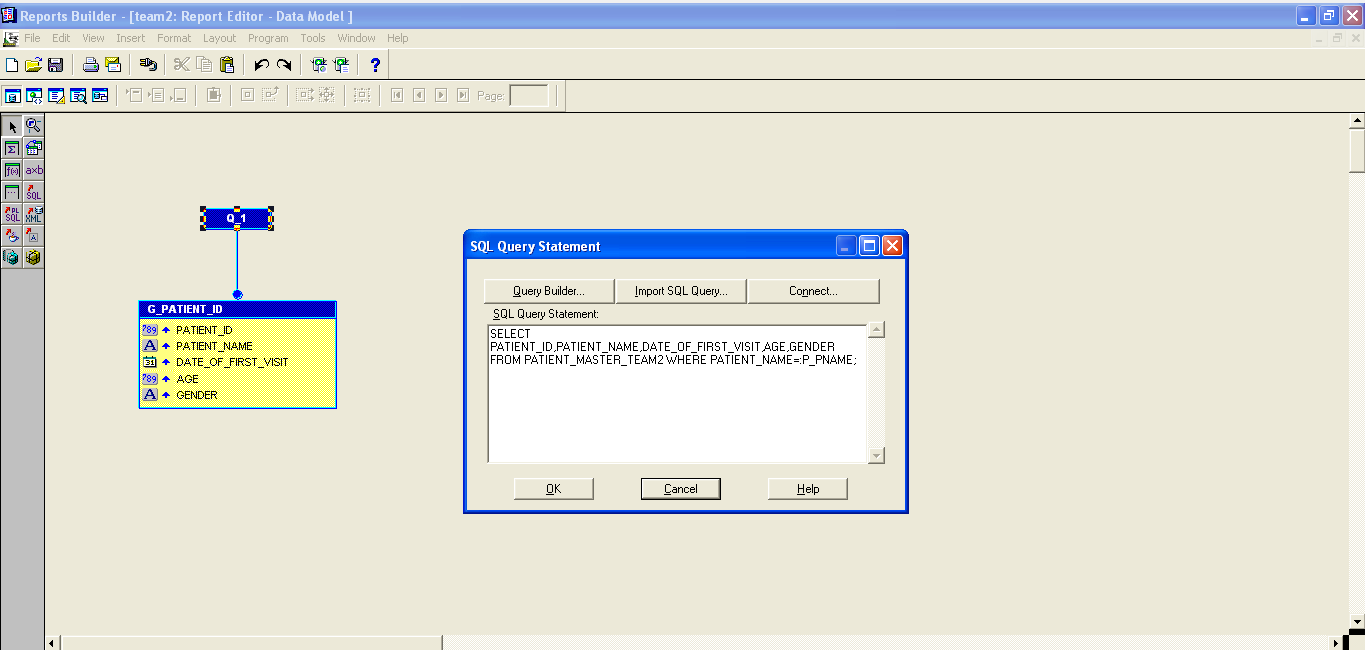
mv TEAM2HM.fmx $PER\_TOP/forms/US

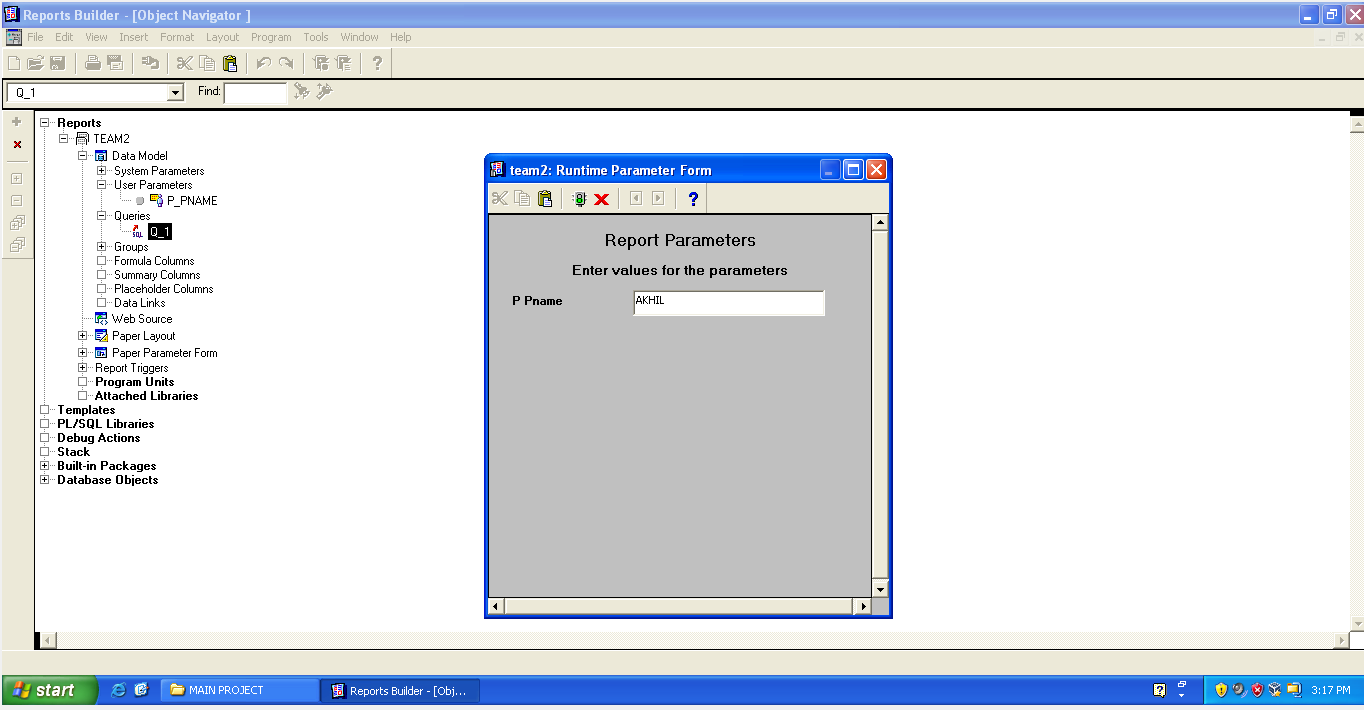
ls TEAM2HM.fmx

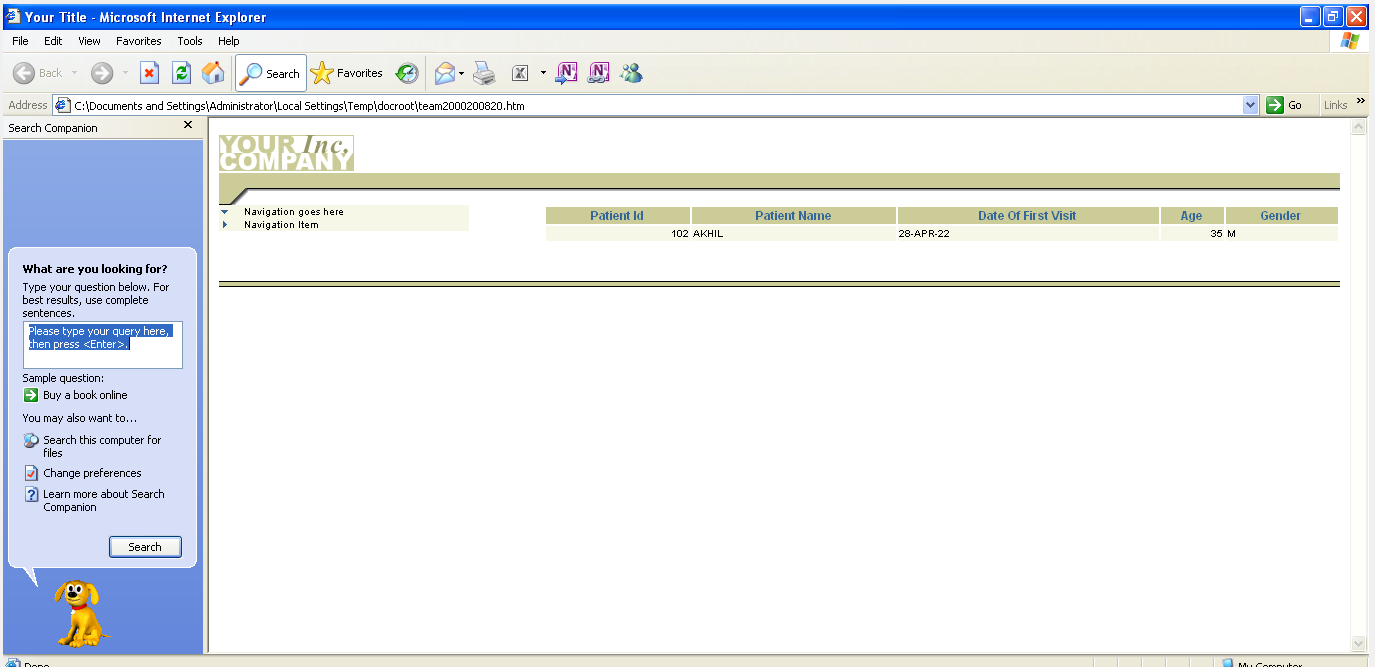
---------------------------------------------------------------------------

**REPORTS:**









**------------------------------------------------**

**ORACLE APPS DETAILS LOGIN DETAILS:**

Application User: INTN2022FEB21TEAM2

Password: 654321

=================================================

Form Name: TEAM2HM

Function Name: TEAM2\_HM\_F1

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

SUB MENU: SUB\_MENU\_TEAM2

PARENT MENU: PARENT\_MENU\_TEAM2

RESPONSIBILITY NAME: RESPONSIBILITY\_TEAM2

Request Group: RG\_TEAM2

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

Registering Form: R \_TEAM\_2

Concurrent Program For Procedure: CP\_TEAM2\_PRO

Value Set: ACTION\_T2

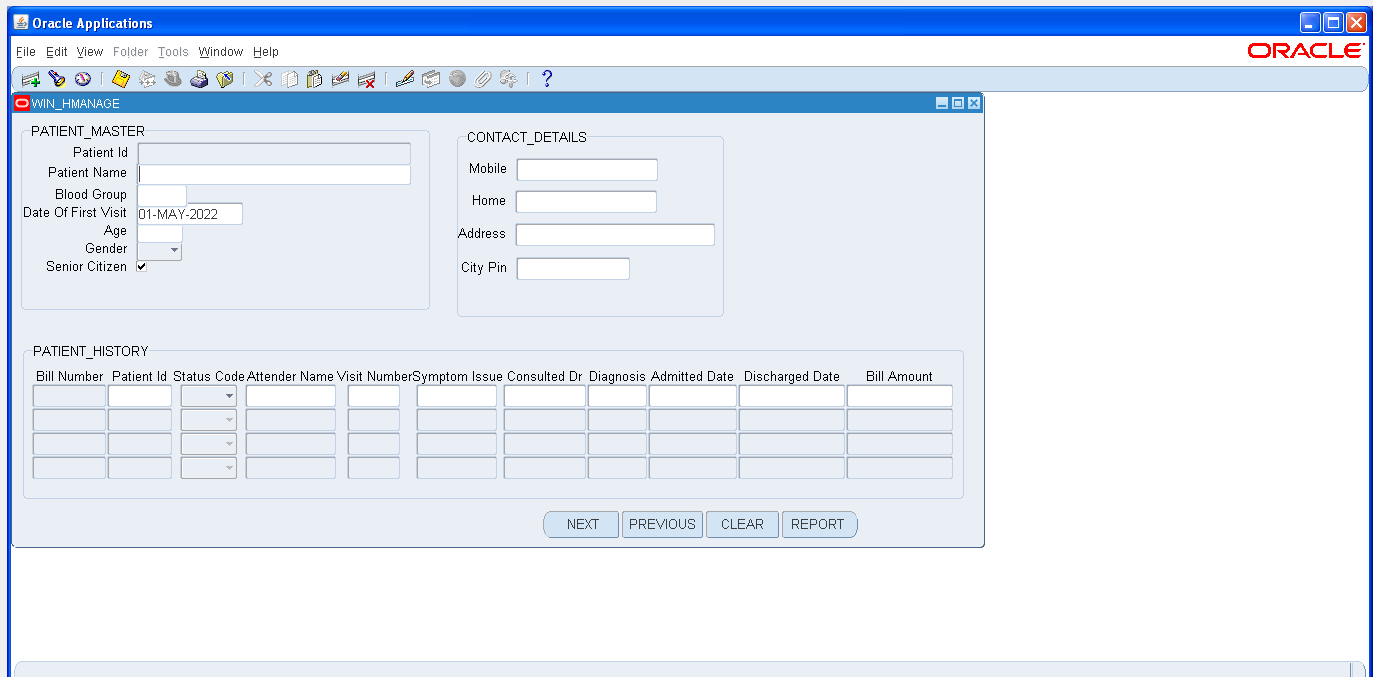
-----------------------------

Registering Report: REG\_TEAM2

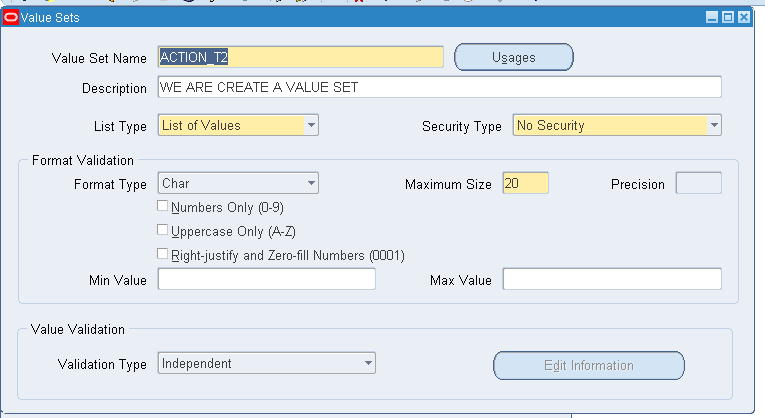
Concurrent Program For Report: CP\_TEAM2

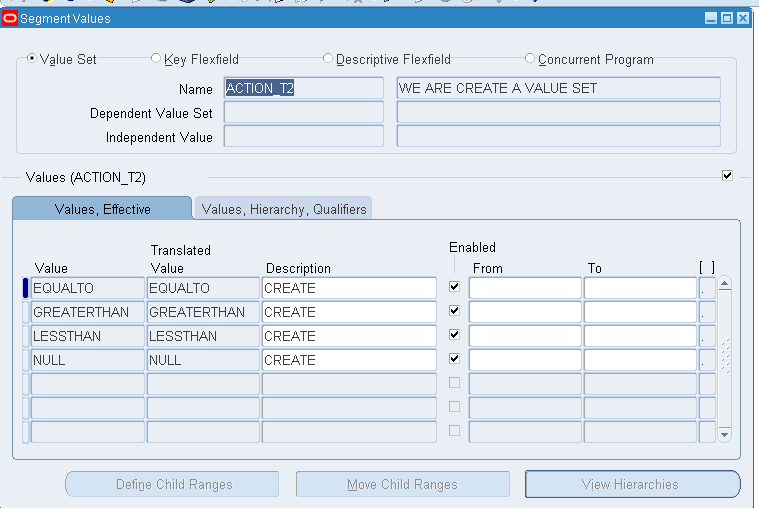
------------------------------------------------------------------------------------

**ORACLE FORM:**



**VALUE SETS:**

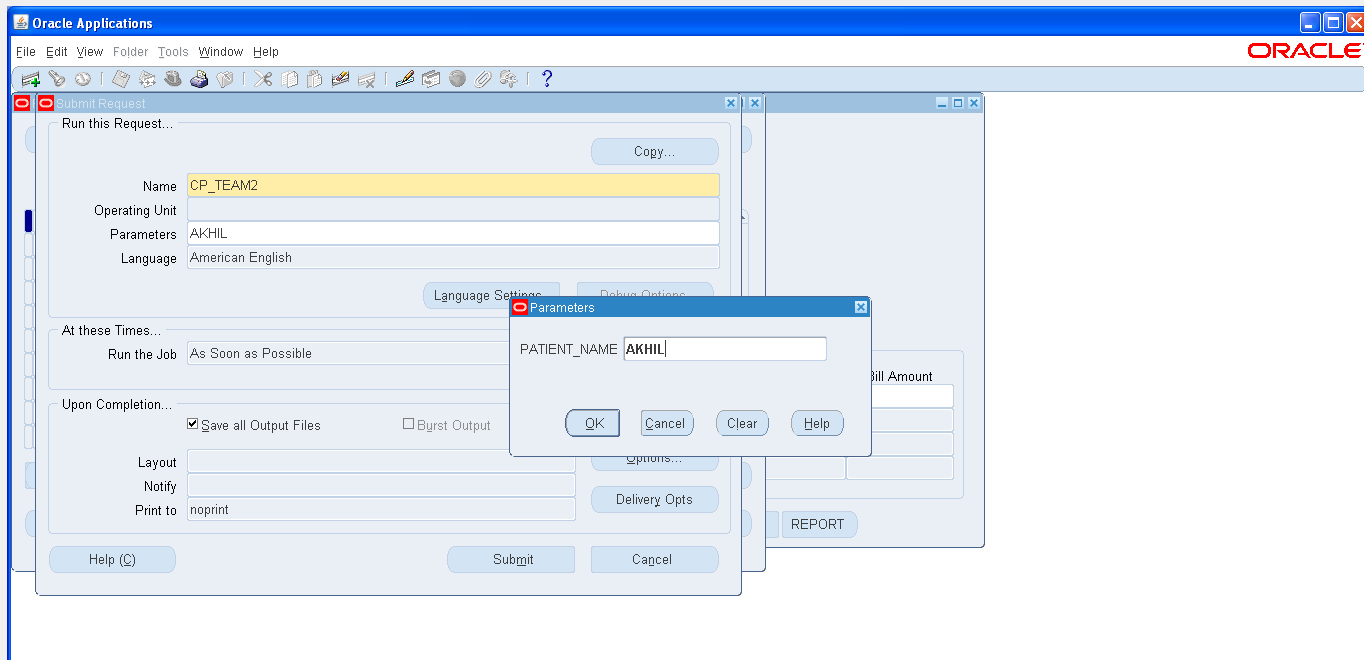


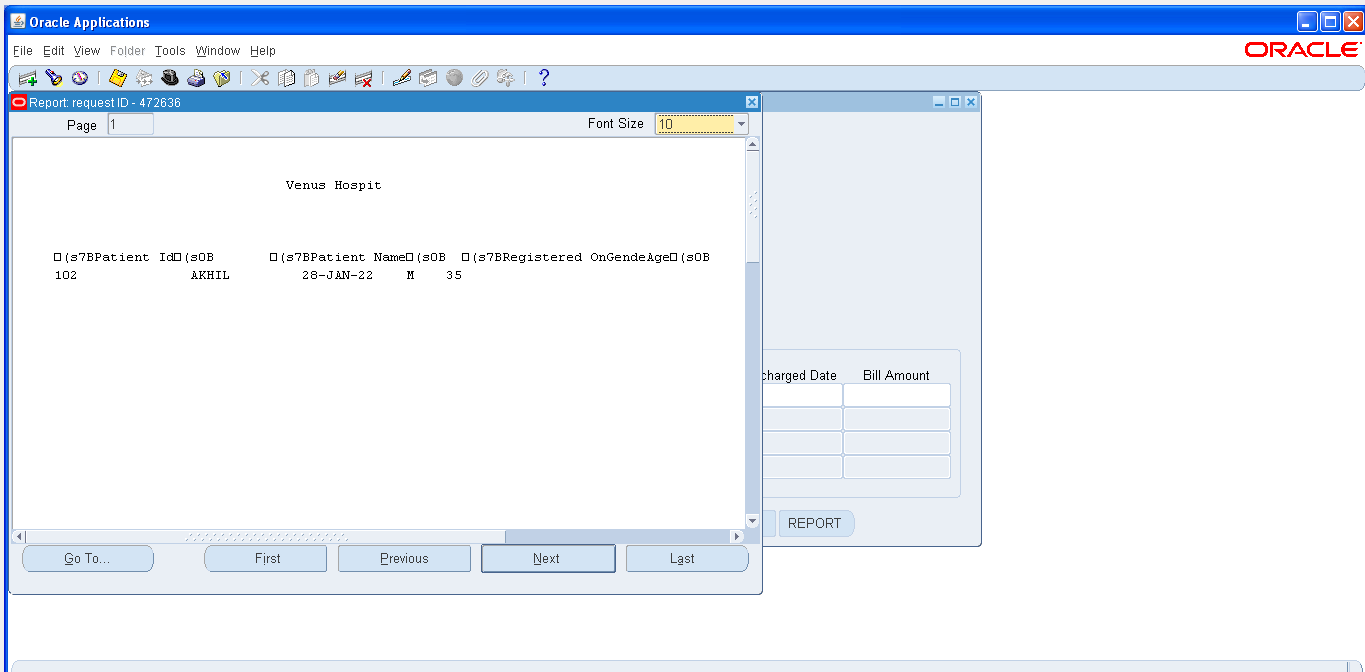


**------------------------------------------------**

**CONCURRENT PROGRAM FOR**

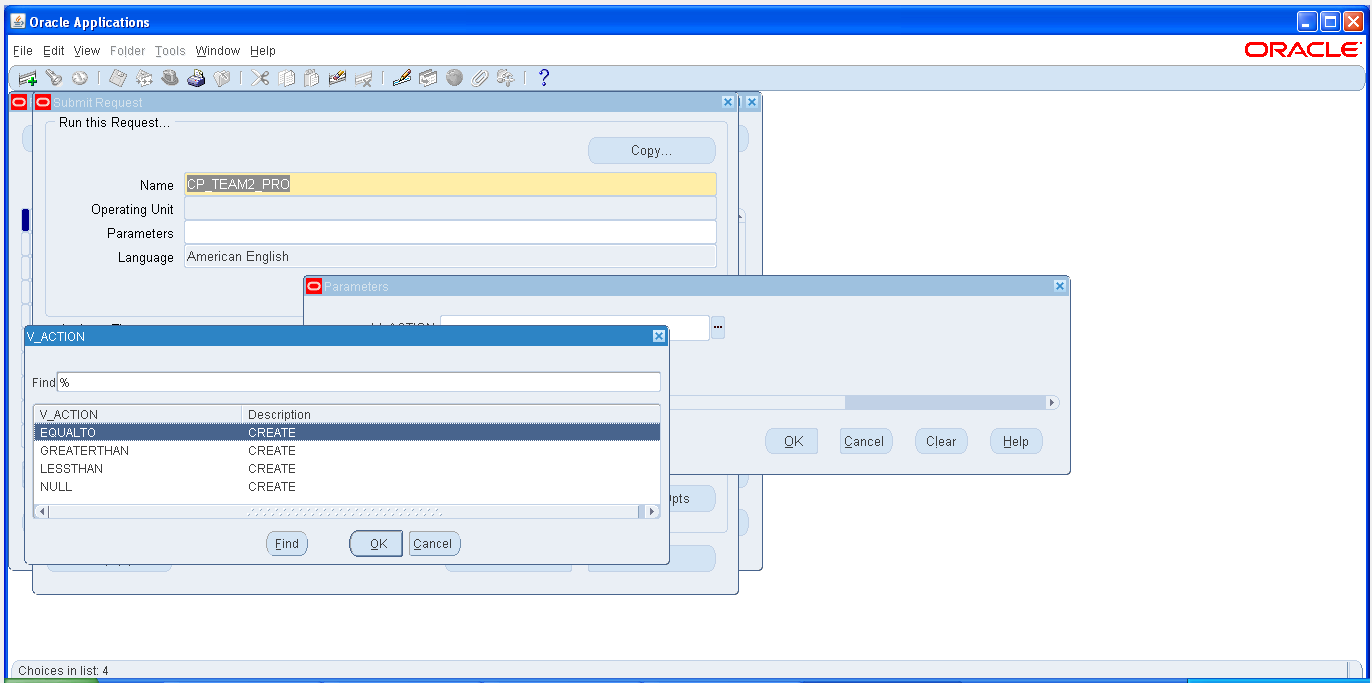
**REPORT:**

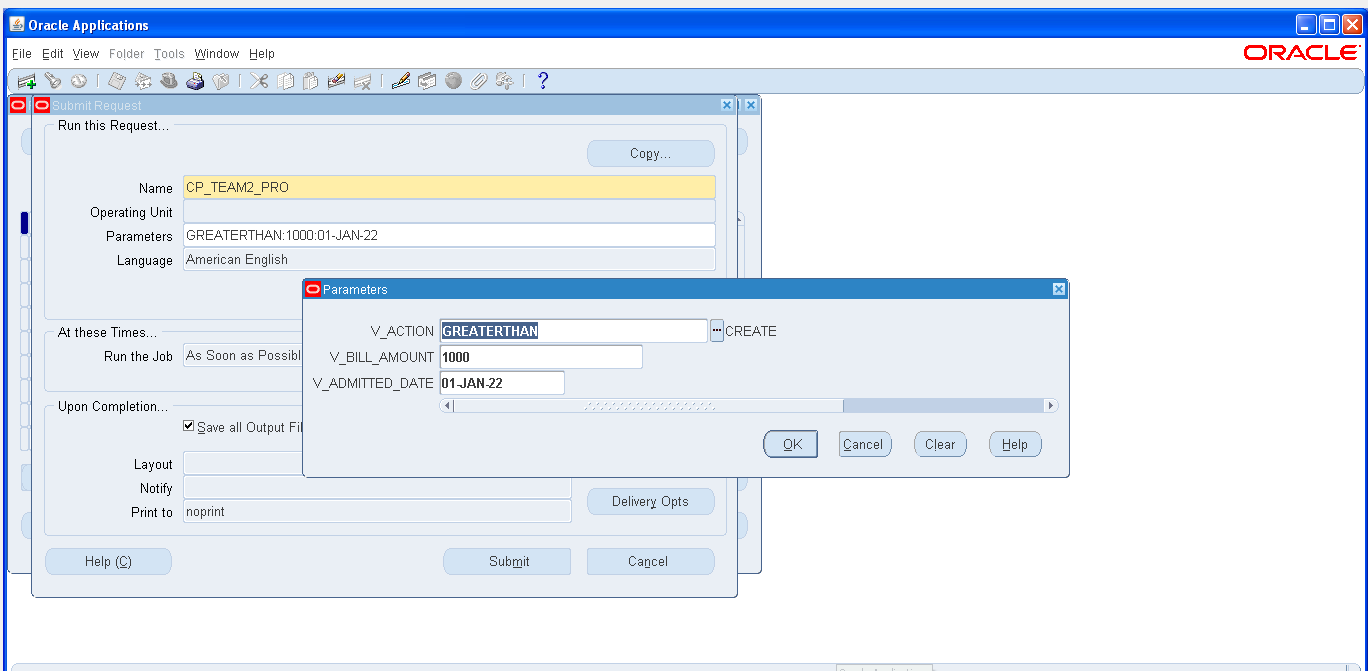




------------------------------

**CONCURRENT PROGRAM FOR PROCEDURE:**





CREATE OR REPLACE PROCEDURE PROG\_TEAM2(V\_ERRCODE OUT NUMBER,V\_ERRMSG OUT VARCHAR2,V\_ACTION VARCHAR2,V\_BILL\_AMOUNT PATIENT\_HISTORY\_TEAM2.BILL\_AMOUNT%TYPE,V\_ADMITTED\_DATE PATIENT\_HISTORY\_TEAM2.ADMITTED\_DATE%TYPE)

IS

V\_PID NUMBER(10);

V\_PNAME VARCHAR2(30);

V\_SYMPTOM VARCHAR2(50);

V\_DOCTOR VARCHAR2(30);

V\_DIA VARCHAR2(50);

V\_BAMOUNT NUMBER(9,2);

CURSOR V\_C1 IS SELECT \* FROM PATIENT\_HISTORY\_TEAM2 FOR UPDATE NOWAIT;

BEGIN

FOR V\_REC IN V\_C1 LOOP

IF V\_ACTION='GREATERTHAN' THEN

IF V\_REC.BILL\_AMOUNT>V\_BILL\_AMOUNT AND V\_REC.ADMITTED\_DATE>V\_ADMITTED\_DATE THEN

SELECT PATIENT\_NAME INTO V\_PNAME FROM PATIENT\_MASTER\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

SELECT V\_REC.PATIENT\_ID,V\_REC.SYMPTOM\_ISSUE,V\_REC.CONSULTED\_DR,V\_REC.DIAGNOSIS,V\_REC.BILL\_AMOUNT INTO V\_PID,V\_SYMPTOM,V\_DOCTOR,V\_DIA,V\_BAMOUNT FROM PATIENT\_HISTORY\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

DBMS\_OUTPUT.PUT\_LINE('RECORD FIND IN GIVEN OBJECT GREATERTHAN');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,'RECORD IN THE TABLES');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PID);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PNAME);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_SYMPTOM);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DOCTOR);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DIA);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_BAMOUNT);

END IF;

ELSIF V\_ACTION='LESSTHAN' THEN

IF V\_REC.BILL\_AMOUNT<V\_BILL\_AMOUNT AND V\_REC.ADMITTED\_DATE<V\_ADMITTED\_DATE THEN

SELECT PATIENT\_NAME INTO V\_PNAME FROM PATIENT\_MASTER\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

SELECT V\_REC.PATIENT\_ID,V\_REC.SYMPTOM\_ISSUE,V\_REC.CONSULTED\_DR,V\_REC.DIAGNOSIS,V\_REC.BILL\_AMOUNT INTO V\_PID,V\_SYMPTOM,V\_DOCTOR,V\_DIA,V\_BAMOUNT FROM PATIENT\_HISTORY\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

DBMS\_OUTPUT.PUT\_LINE('RECORD FIND IN GIVEN OBJECT LESSTHAN');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,'RECORD IN THE TABLES');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PID);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PNAME);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_SYMPTOM);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DOCTOR);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DIA);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_BAMOUNT);

END IF;

ELSIF V\_ACTION='EQUALTO' THEN

IF V\_REC.BILL\_AMOUNT=V\_BILL\_AMOUNT AND V\_REC.ADMITTED\_DATE=V\_ADMITTED\_DATE THEN

SELECT PATIENT\_NAME INTO V\_PNAME FROM PATIENT\_MASTER\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

SELECT V\_REC.PATIENT\_ID,V\_REC.SYMPTOM\_ISSUE,V\_REC.CONSULTED\_DR,V\_REC.DIAGNOSIS,V\_REC.BILL\_AMOUNT INTO V\_PID,V\_SYMPTOM,V\_DOCTOR,V\_DIA,V\_BAMOUNT FROM PATIENT\_HISTORY\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

DBMS\_OUTPUT.PUT\_LINE('RECORD FIND IN GIVEN OBJECT EQUAL');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,'RECORD IN THE TABLES');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PID);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PNAME);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_SYMPTOM);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DOCTOR);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DIA);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_BAMOUNT);

END IF;

ELSIF V\_ACTION='NULL' THEN

SELECT PATIENT\_NAME INTO V\_PNAME FROM PATIENT\_MASTER\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

SELECT V\_REC.PATIENT\_ID,V\_REC.SYMPTOM\_ISSUE,V\_REC.CONSULTED\_DR,V\_REC.DIAGNOSIS,V\_REC.BILL\_AMOUNT INTO V\_PID,V\_SYMPTOM,V\_DOCTOR,V\_DIA,V\_BAMOUNT FROM PATIENT\_HISTORY\_TEAM2 WHERE PATIENT\_ID=V\_REC.PATIENT\_ID;

DBMS\_OUTPUT.PUT\_LINE('RECORD FIND IN GIVEN OBJECT NULL');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,'RECORD IN THE TABLES');

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PID);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_PNAME);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_SYMPTOM);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DOCTOR);

FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_DIA);

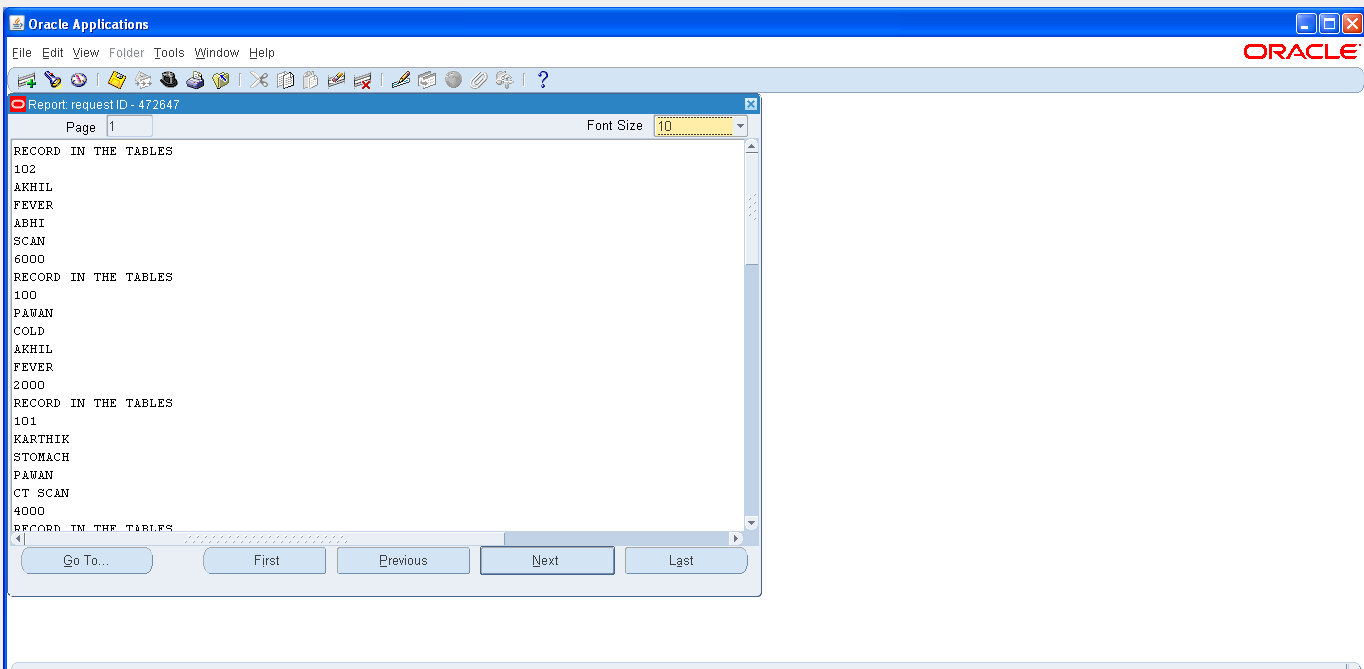
FND\_FILE.PUT\_LINE(FND\_FILE.OUTPUT,V\_BAMOUNT);

END IF;

END LOOP;

COMMIT;

END;



**THANK YOU**