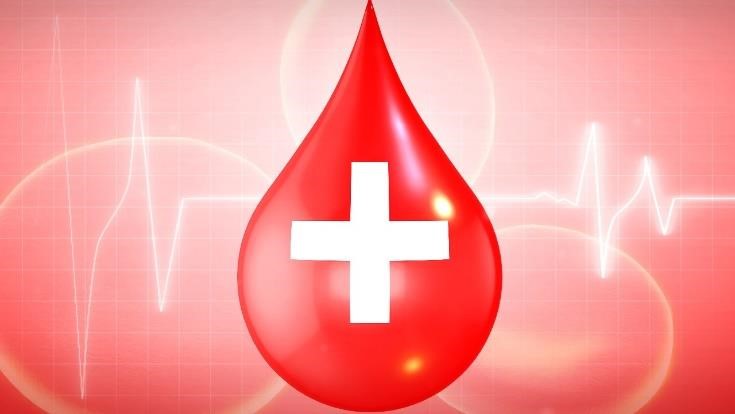
**Blood-Bank-Management-System**



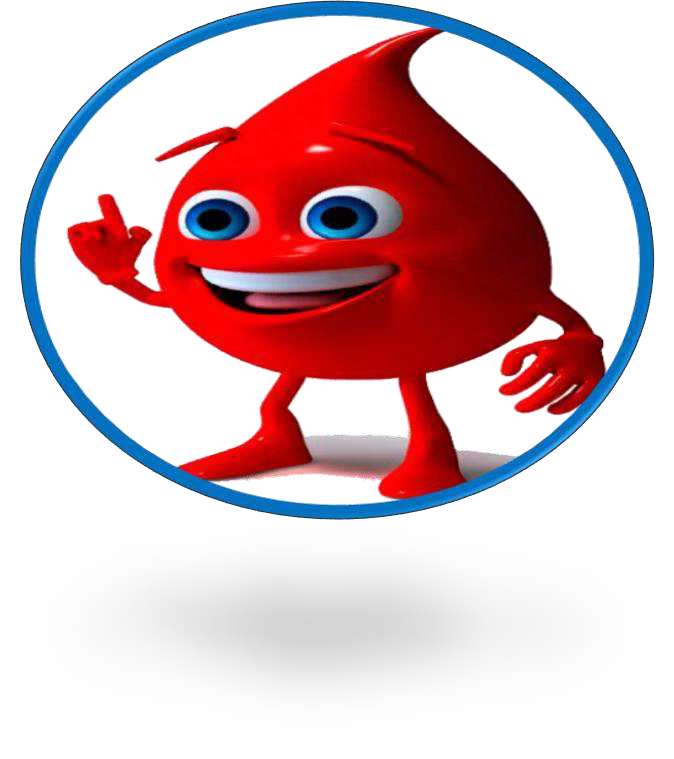
**Student: Luis Rodriguez**

**Group 1068**

**Series G, 2nd Year**

**Romania**

**Bucharest, 2022**



### 1. Description of the chosen theme:

Blood bank is a place where blood bag that is collected from blood donation events is stored in one place. The term “blood bank” refers to a division of a hospital laboratory where the storage of blood product occurs and where proper testing is performed to reduce the risk of transfusion related events.

Blood Bank Management System is a web-based system that can assists the information of blood bag during its handling in the blood bank. With this system, the user of this system can key in the result of blood test that has been conducted to each of the blood bag received by the blood bank. The result of test will indicate whether the blood bag can be delivered to patient or not.

The process of managing the blood bag that is received from the blood donation events needs a proper and systematic management. The blood bag must be handled with care and treated thoroughly as it is related to someone’s life. The development of Web-based Blood Bank Management System is proposed to provide a management functional to the blood bank in order to handle the blood bag.

Consisting data of different blood groups along with the medical history by bringing donors and recipients of blood on a single platform and suggesting nearest blood transfusion center according to the location of user, all the data is located in Romania.

Our database is composed from 9 Tables: (1) HOSPITAL,(2)INVENTORY,(3)PATIENT,(4)TRANS,

(5)REQUEST\_TYPE,(6)USER\_TYPE,(7)USER\_CREDS,(8)PATIENT\_TYPE, (9)BILLING

1.**HOSPITAL**: is a table that store Hospitals name,Adress and phone numbers

2.**INVENTORY:**is a table that store H\_ID ,PT\_BTYPE,TOTAL\_BLOOD\_UNIT,INVENTORY\_ID.

3.**PATIENT:** is a table that store

PT\_ID,USER\_ID,PT\_FIRST\_NAME,PT\_LAST\_NAME,PT\_ADRESS,PT\_TYPE\_ID,PT\_GENDER,PT\_AGE,PT\_BTYP E,PT\_STD,PATIENT\_PHONE,PT\_WEIGHT.

4.**TRANS:**is a table that store

TRANS\_ID,PT\_ID,T\_DATE,REQUEST\_TYPE\_ID,PT\_BTYPE,BLOOD\_UNIT,H\_ID,INVENTORY\_ID.

5.**REQUEST\_TYPE:**is a table that store REQUEST\_TYPE\_ID and REQUEST\_TYPE\_NAME.

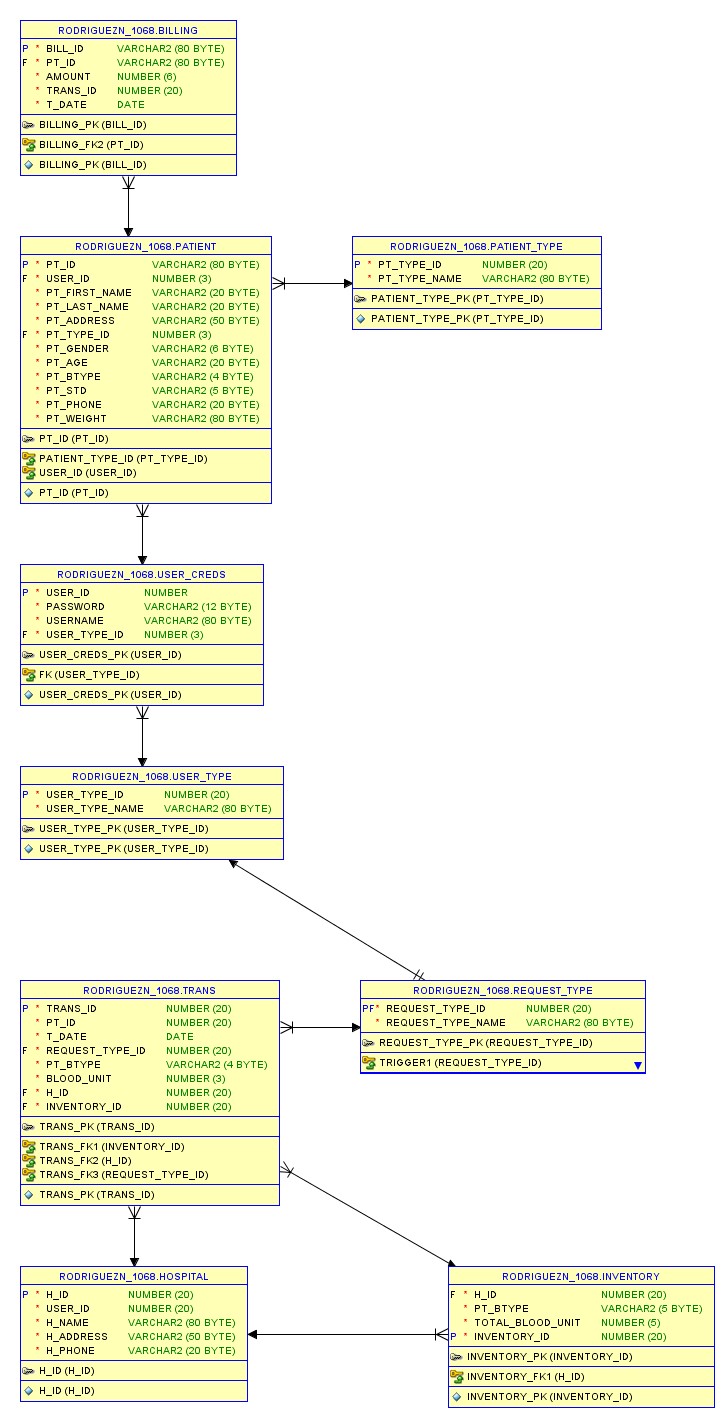
6.**USER\_TYPE:**is a table that store USER\_TYPE\_ID and USER\_TYPE\_NAME.

7**.USER\_CREDS:**is a table that store USER\_ID,PASSWORD,USERNAME,USER\_TYPE\_ID.

8.**PATIENT\_TYPE:**is a table that store PT\_TYPE\_ID and PT\_TYPE\_NAME.

9.**BILLING:**is a table that store BILL\_ID,PT\_ID,AMOUNT,TRANS\_ID,T\_DATE

## 2. Conceptual scheme for modeling the chosen theme:



The attributes have the following constraints:

1. **Table HOSPITAL:**

H\_ID NOT NULL ,

USER\_ID NOT NULL ,

H\_NAME NOT NULL ,

H\_ADDRESS NOT NULL ,

H\_PHONE NOT NULL ,

CONSTRAINT H\_ID PRIMARY KEY (H\_ID)

1. **Table INVENTORY:**

H\_ID NOT NULL , PT\_BTYPE NOT NULL ,

TOTAL\_BLOOD\_UNIT NOT NULL ,

INVENTORY\_ID NOT NULL ,

CONSTRAINT INVENTORY\_PK PRIMARY KEY (INVENTORY\_ID),

CONSTRAINT INVENTORY\_FK1 FOREIGN KEY (H\_ID),

REFERENCES HOSPITAL (H\_ID)

1. **Table PATIENT:**

PT\_ID NOT NULL , USER\_ID NOT NULL ,

PT\_FIRST\_NAME NOT NULL ,

PT\_LAST\_NAME ,

PT\_ADDRESS NOT NULL ,

PT\_TYPE\_ID NOT NULL ,

PT\_GENDER NOT NULL ,

PT\_AGE NOT NULL ,

PT\_BTYPE NOT NULL , PT\_STD NOT NULL ,

PT\_PHONE NOT NULL ,

CONSTRAINT PT\_ID PRIMARY KEY (PT\_ID),

CONSTRAINT USER\_ID FOREIGN KEY (USER\_ID)

REFERENCES USER\_CREDS (USER\_ID) ,

CONSTRAINT PATIENT\_TYPE\_ID FOREIGN KEY (PT\_TYPE\_ID)

REFERENCES PATIENT\_TYPE (PT\_TYPE\_ID)

4. **Table TRANS:**

TRANS\_ID NOT NULL , PT\_ID NOT NULL ,

T\_DATE DATE NOT NULL ,

REQUEST\_TYPE\_ID NOT NULL ,

PT\_BTYPE NOT NULL ,

BLOOD\_UNIT NOT NULL ,

H\_ID NOT NULL ,

INVENTORY\_ID NOT NULL ,

CONSTRAINT TRANS\_PK PRIMARY KEY (TRANS\_ID),

CONSTRAINT TRANS\_FK1 FOREIGN KEY (INVENTORY\_ID)

REFERENCES INVENTORY (INVENTORY\_ID) ,

CONSTRAINT TRANS\_FK2 FOREIGN KEY (H\_ID)

REFERENCES HOSPITAL (H\_ID) ,

1. **Table REQUEST\_TYPE:**

REQUEST\_TYPE\_ID NOT NULL,

REQUEST\_TYPE\_NAME NOT NULL ,

CONSTRAINT REQUEST\_TYPE\_PK

PRIMARY KEY ("REQUEST\_TYPE\_ID")

1. **Table USER\_TYPE:**

USER\_TYPE\_ID NOT NULL ENABLE, USER\_TYPE\_NAME NOT NULL ENABLE,

CONSTRAINT USER\_TYPE\_PK PRIMARY KEY ("USER\_TYPE\_ID")

1. **Table USER\_CREDS:**

USER\_ID NOT NULL , PASSWORD NOT NULL ,

USERNAME NOT NULL ,

USER\_TYPE\_ID NOT NULL ,

CONSTRAINT USER\_CREDS\_PK PRIMARY KEY (USER\_ID),

CONSTRAINT FK FOREIGN KEY (USER\_TYPE\_ID)

REFERENCES USER\_TYPE (USER\_TYPE\_ID)

1. **PATIENT\_TYPE:**

PT\_TYPE\_ID NOT NULL ENABLE, PT\_TYPE\_NAME NOT NULL ENABLE,

CONSTRAINT PATIENT\_TYPE\_PK PRIMARY KEY ("PT\_TYPE\_ID")

1. **BILLING:**

BILL\_ID NOT NULL ENABLE, PT\_ID NOT NULL ,

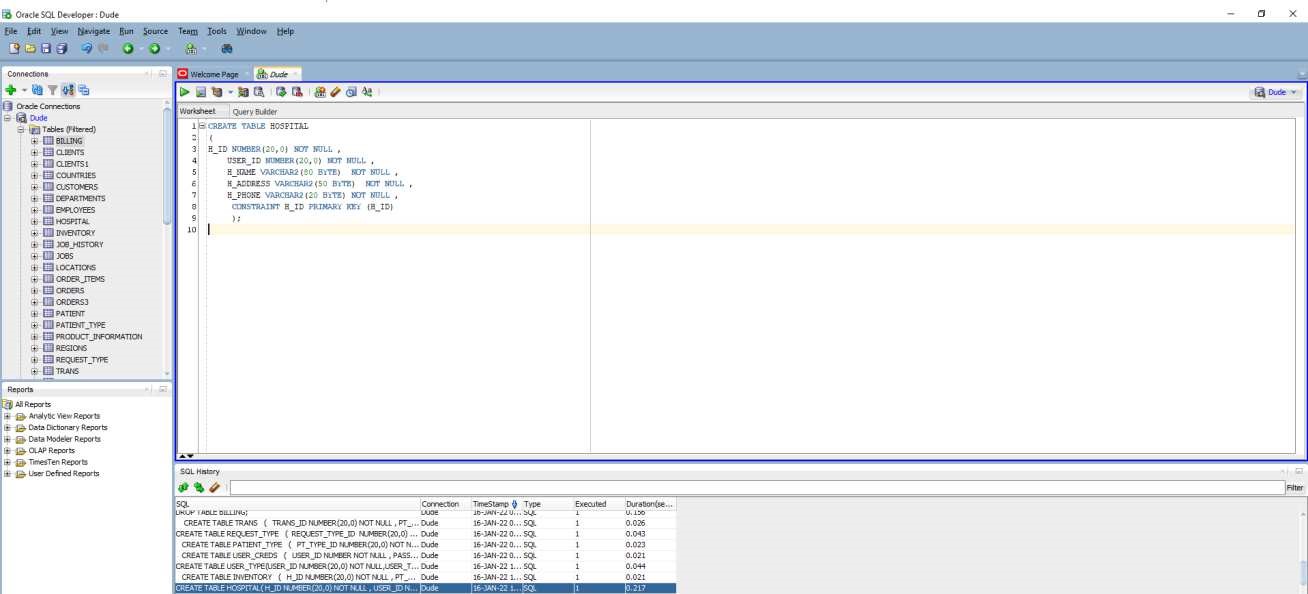
AMOUNT NOT NULL,

TRANS\_ID NOT NULL ,

T\_DATE" DATE NOT NULL ,

CONSTRAINT BILLING\_PK PRIMARY KEY ("BILL\_ID")

## Constructing the database: tables and constraints:



**CREATE TABLE HOSPITAL**

**(**

**H\_ID NUMBER(20,0) NOT NULL ,**

**USER\_ID NUMBER(20,0) NOT NULL ,**

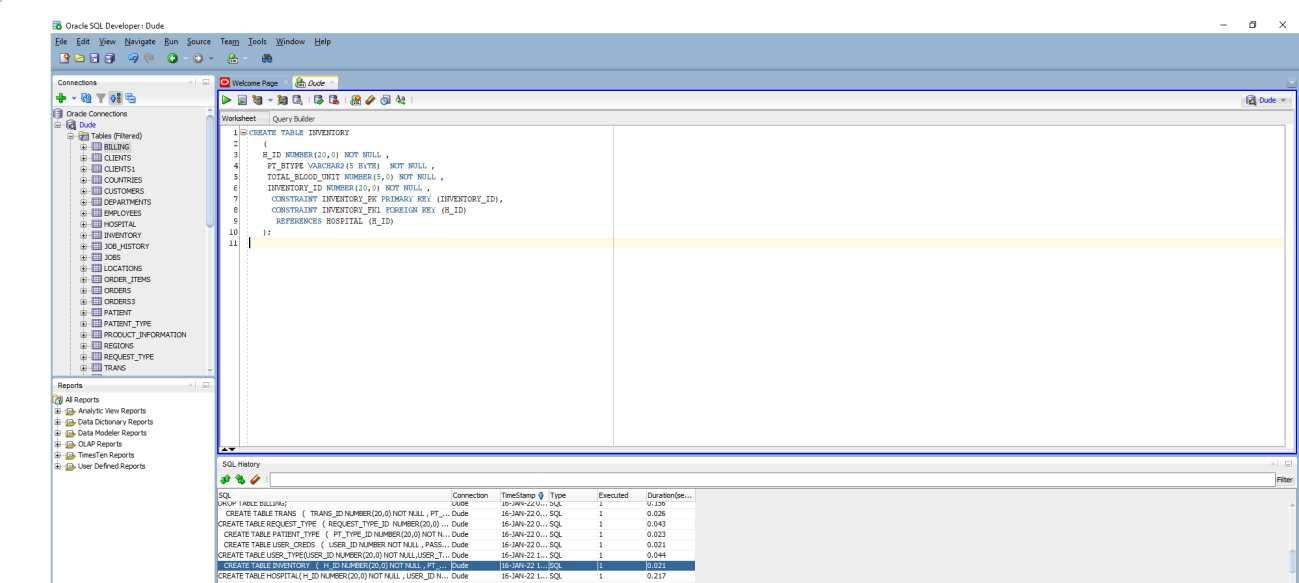
**H\_NAME VARCHAR2(80 BYTE) NOT NULL ,**

**H\_ADDRESS VARCHAR2(50 BYTE) NOT NULL ,**

**H\_PHONE VARCHAR2(20 BYTE) NOT NULL ,**

**CONSTRAINT H\_ID PRIMARY KEY (H\_ID)**

**);**



**CREATE TABLE INVENTORY**

**(**

**H\_ID NUMBER(20,0) NOT NULL ,**

**PT\_BTYPE VARCHAR2(5 BYTE) NOT NULL ,**

**TOTAL\_BLOOD\_UNIT NUMBER(5,0) NOT NULL ,**

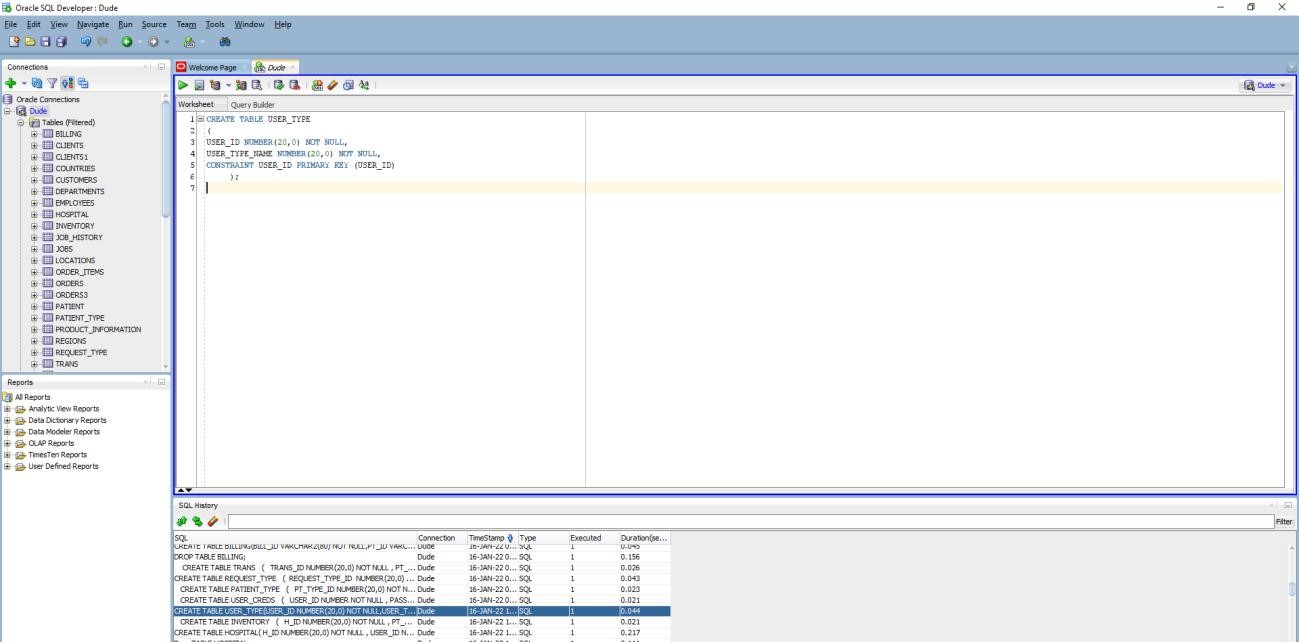
**INVENTORY\_ID NUMBER(20,0) NOT NULL ,**

**CONSTRAINT INVENTORY\_PK PRIMARY KEY (INVENTORY\_ID),**

**CONSTRAINT INVENTORY\_FK1 FOREIGN KEY (H\_ID)**

**REFERENCES HOSPITAL (H\_ID)**

**);**



**CREATE TABLE USER\_TYPE**

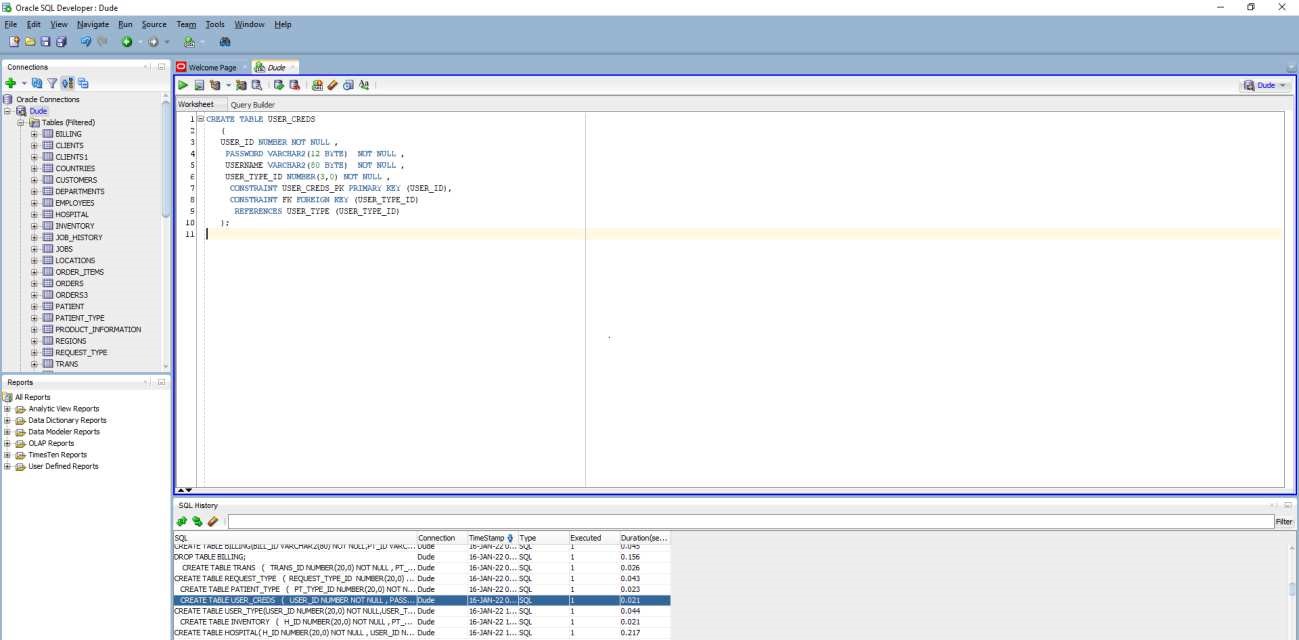
**(**

**USER\_ID NUMBER(20,0) NOT NULL,**

**USER\_TYPE\_NAME NUMBER(20,0) NOT NULL,**

**CONSTRAINT USER\_ID PRIMARY KEY (USER\_ID)**

**);**



**CREATE TABLE USER\_CREDS**

**(**

**USER\_ID NUMBER NOT NULL ,**

**PASSWORD VARCHAR2(12 BYTE) NOT NULL ,**

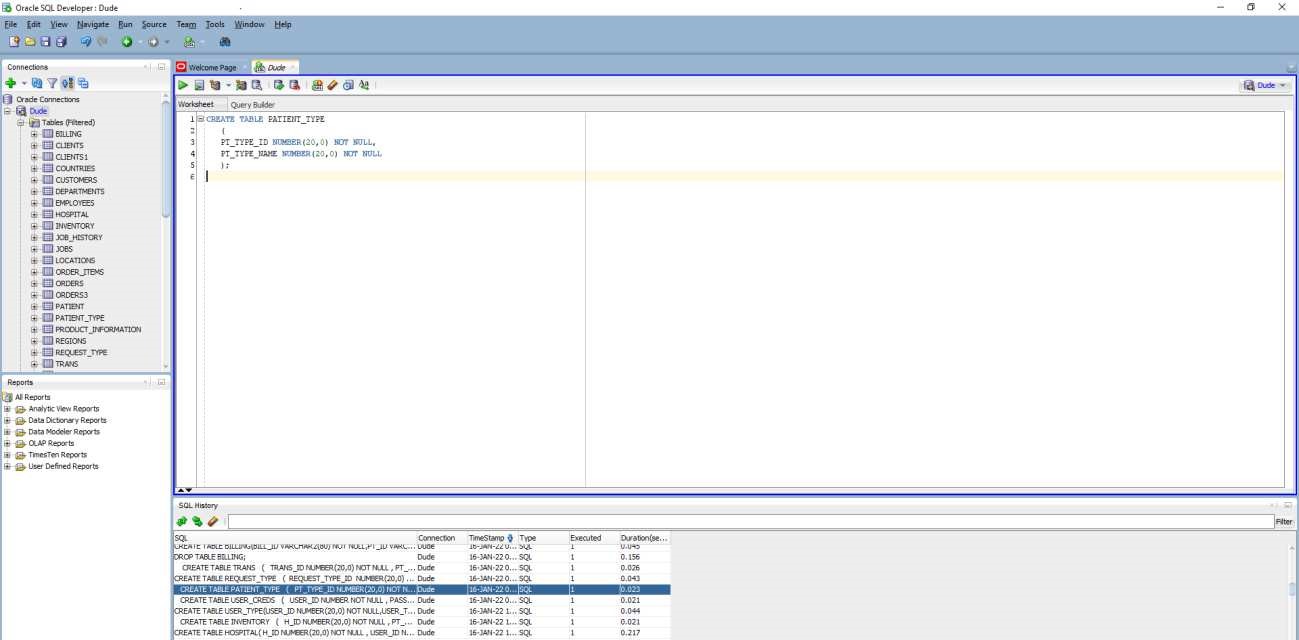
**USERNAME VARCHAR2(80 BYTE) NOT NULL ,**

**USER\_TYPE\_ID NUMBER(3,0) NOT NULL ,**

**CONSTRAINT USER\_CREDS\_PK PRIMARY KEY (USER\_ID),**

**CONSTRAINT FK FOREIGN KEY (USER\_TYPE\_ID) REFERENCES USER\_TYPE (USER\_TYPE\_ID)**

**);**



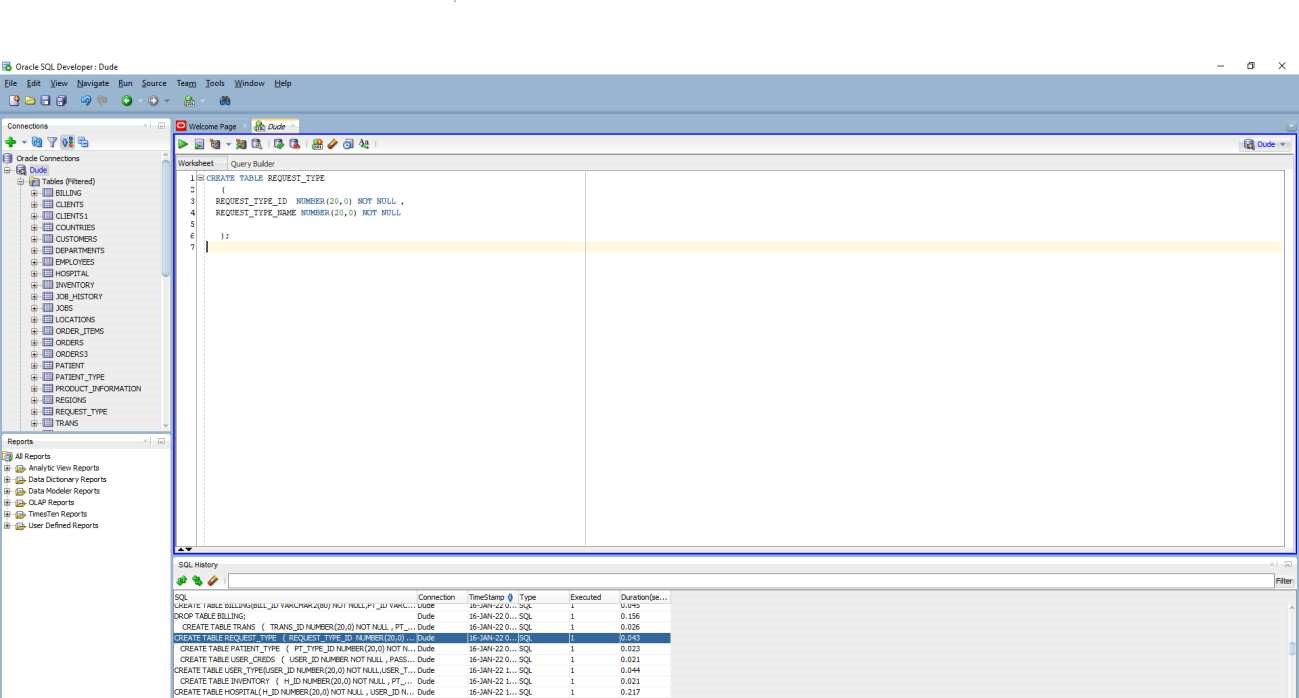
**CREATE TABLE PATIENT\_TYPE**

**(**

**PT\_TYPE\_ID NUMBER(20,0) NOT NULL,**

**PT\_TYPE\_NAME NUMBER(20,0) NOT NULL**

**);**



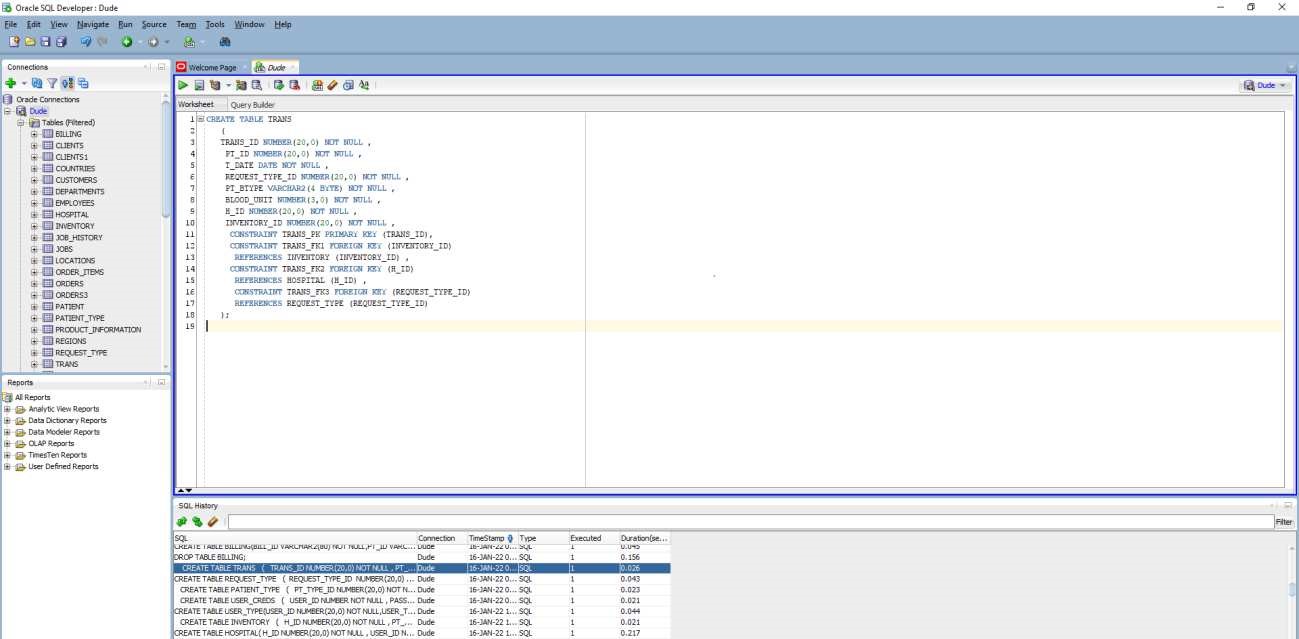
**CREATE TABLE REQUEST\_TYPE**

**(**

**REQUEST\_TYPE\_ID NUMBER(20,0) NOT NULL ,**

**REQUEST\_TYPE\_NAME NUMBER(20,0) NOT NULL**

**);**



**CREATE TABLE TRANS**

**(**

**TRANS\_ID NUMBER(20,0) NOT NULL ,**

**PT\_ID NUMBER(20,0) NOT NULL ,**

**T\_DATE DATE NOT NULL ,**

**REQUEST\_TYPE\_ID NUMBER(20,0) NOT NULL ,**

**PT\_BTYPE VARCHAR2(4 BYTE) NOT NULL ,**

**BLOOD\_UNIT NUMBER(3,0) NOT NULL ,**

**H\_ID NUMBER(20,0) NOT NULL ,**

**INVENTORY\_ID NUMBER(20,0) NOT NULL ,**

**CONSTRAINT TRANS\_PK PRIMARY KEY (TRANS\_ID),**

**CONSTRAINT TRANS\_FK1 FOREIGN KEY (INVENTORY\_ID)**

**REFERENCES INVENTORY (INVENTORY\_ID) ,**

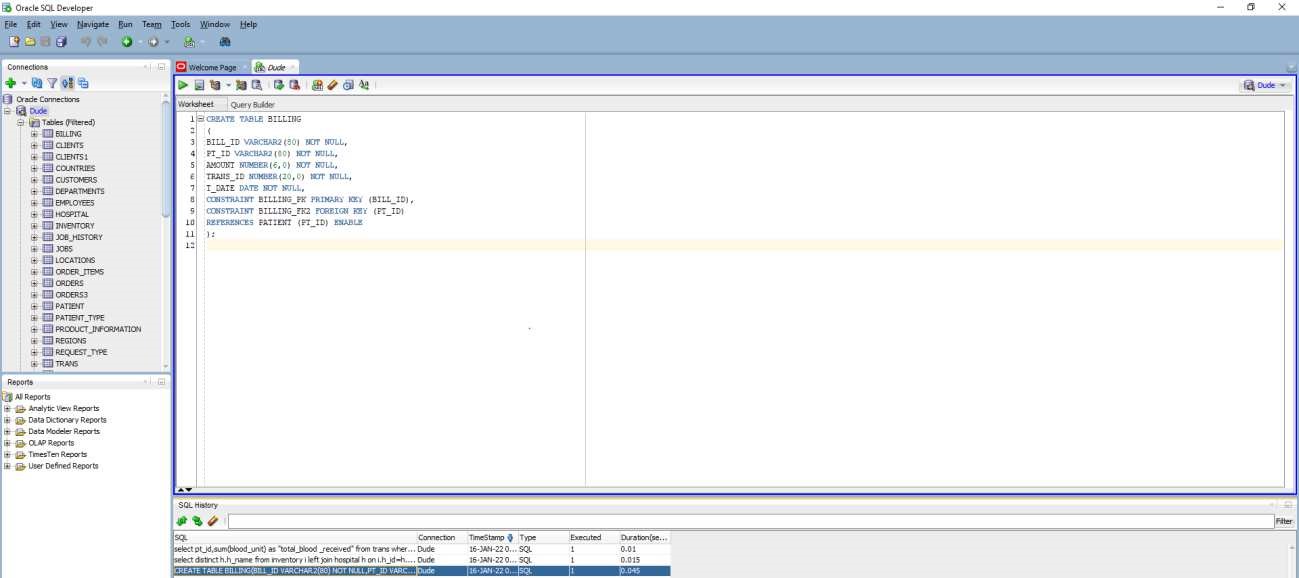
**CONSTRAINT TRANS\_FK2 FOREIGN KEY (H\_ID)**

**REFERENCES HOSPITAL (H\_ID) ,**

**CONSTRAINT TRANS\_FK3 FOREIGN KEY (REQUEST\_TYPE\_ID)**

**REFERENCES REQUEST\_TYPE (REQUEST\_TYPE\_ID)**

**);**



**CREATE TABLE BILLING**

**(**

**BILL\_ID VARCHAR2(80) NOT NULL,**

**PT\_ID VARCHAR2(80) NOT NULL,**

**AMOUNT NUMBER(6,0) NOT NULL,**

**TRANS\_ID NUMBER(20,0) NOT NULL,**

**T\_DATE DATE NOT NULL,**

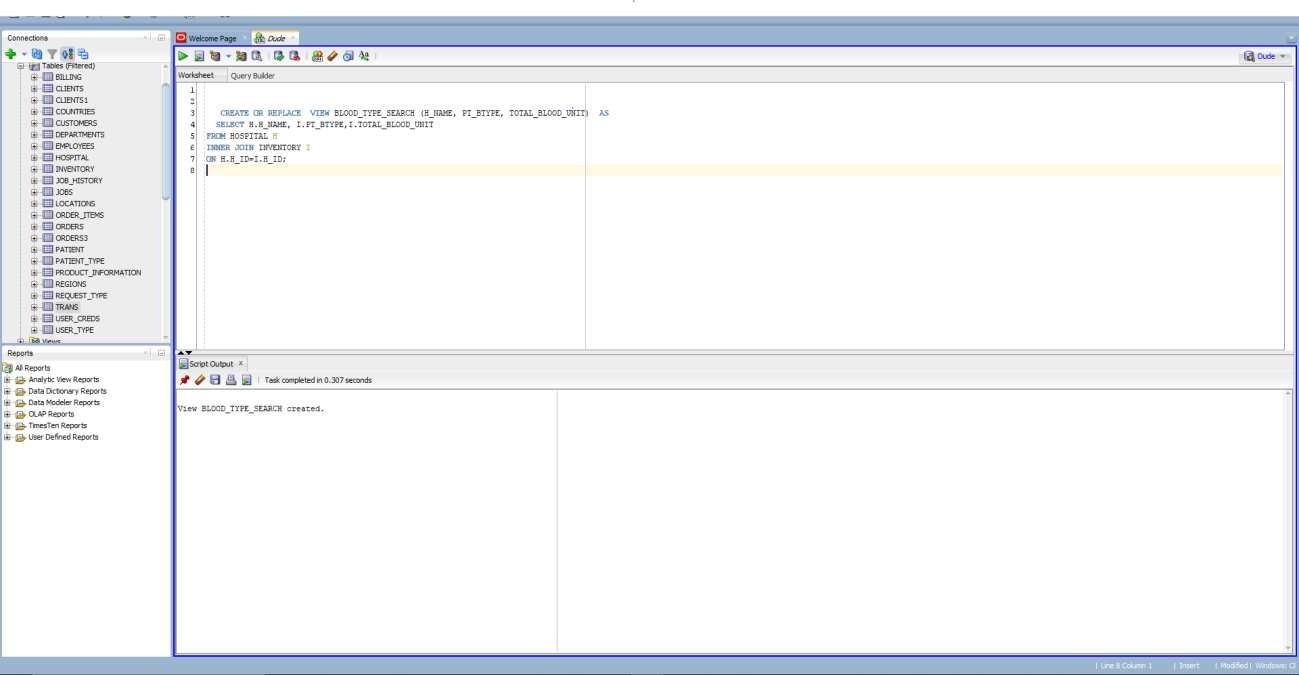
**CONSTRAINT BILLING\_PK PRIMARY KEY (BILL\_ID),**

**CONSTRAINT BILLING\_FK2 FOREIGN KEY (PT\_ID)**

**REFERENCES PATIENT (PT\_ID) ENABLE**

**);**

# DML statements:



Blood type search by location

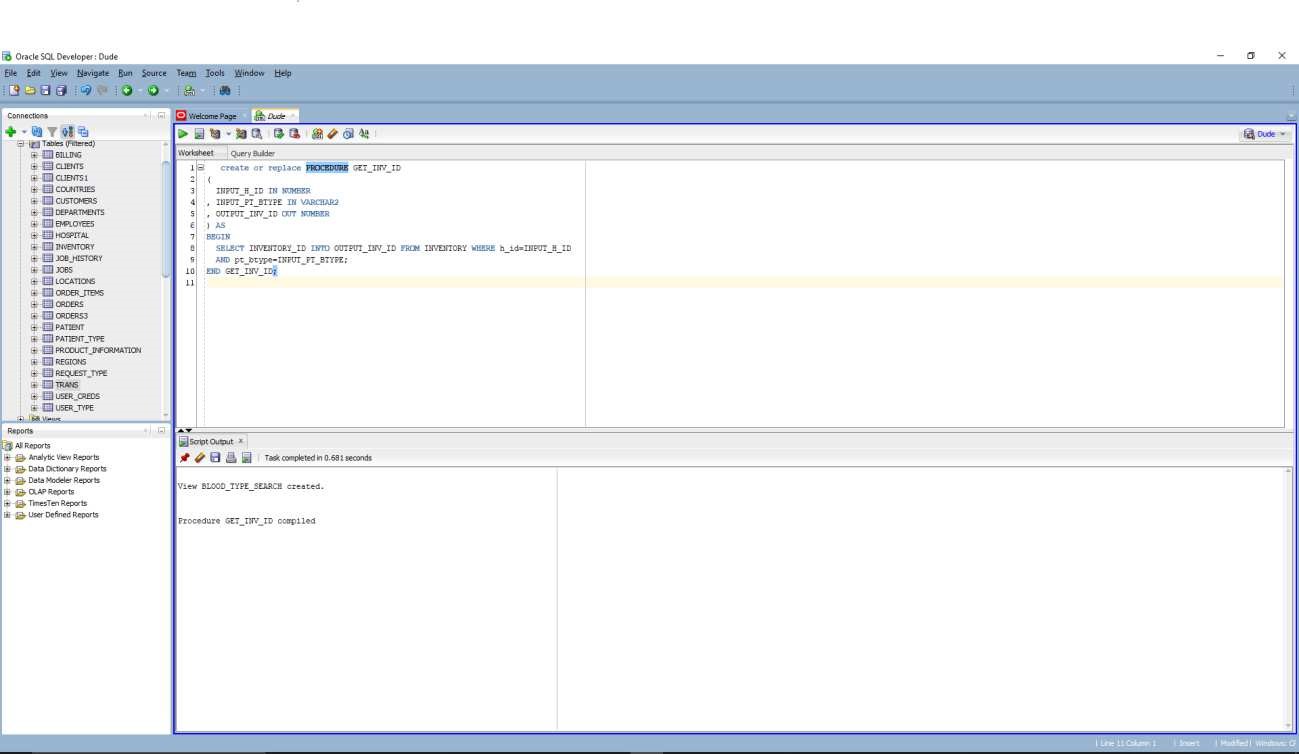
**CREATE OR REPLACE VIEW BLOOD\_TYPE\_SEARCH (H\_NAME, PT\_BTYPE, TOTAL\_BLOOD\_UNIT) AS**

**SELECT H.H\_NAME, I.PT\_BTYPE,I.TOTAL\_BLOOD\_UNIT**

**FROM HOSPITAL H**

**INNER JOIN INVENTORY I**

**ON H.H\_ID=I.H\_ID;**



Stored Procedure ( Determining the inventory\_ID for coresponding H\_id and PT\_btype) create or replace PROCEDURE GET\_INV\_ID

(

INPUT\_H\_ID IN NUMBER

, INPUT\_PT\_BTYPE IN VARCHAR2

, OUTPUT\_INV\_ID OUT NUMBER

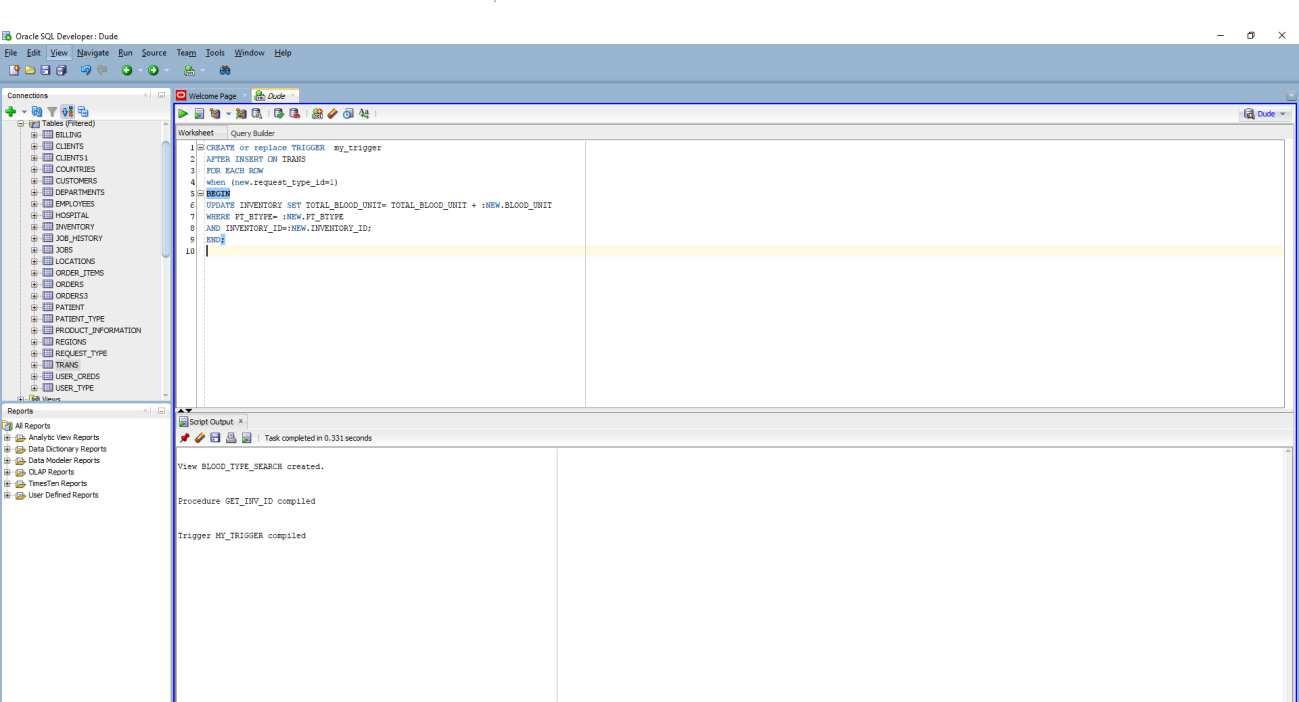
) AS

BEGIN

SELECT INVENTORY\_ID INTO OUTPUT\_INV\_ID FROM INVENTORY WHERE h\_id=INPUT\_H\_ID

AND pt\_btype=INPUT\_PT\_BTYPE;

END GET\_INV\_ID;



CREATE or replace TRIGGER my\_trigger

AFTER INSERT ON TRANS FOR EACH ROW when (new.request\_type\_id=1)

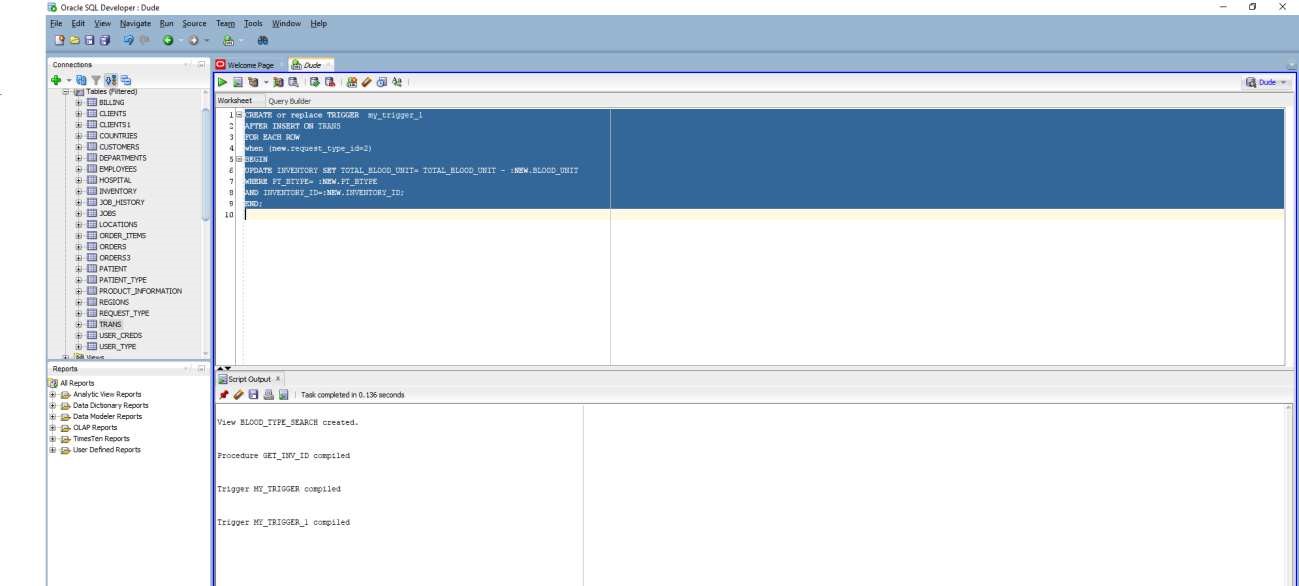
BEGIN

UPDATE INVENTORY SET TOTAL\_BLOOD\_UNIT= TOTAL\_BLOOD\_UNIT + :NEW.BLOOD\_UNIT

WHERE PT\_BTYPE= :NEW.PT\_BTYPE

AND INVENTORY\_ID=:NEW.INVENTORY\_ID;

END;



**CREATE or replace TRIGGER my\_trigger\_1**

**AFTER INSERT ON TRANS**

**FOR EACH ROW when (new.request\_type\_id=2)**

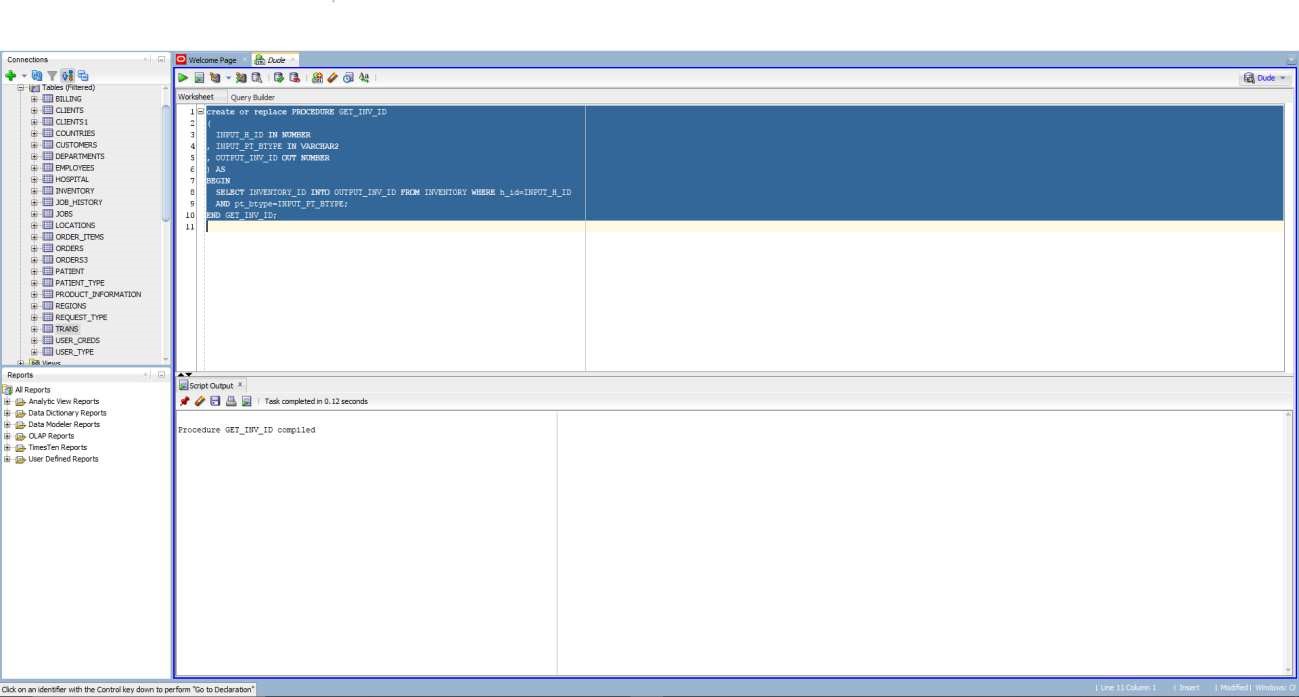
**BEGIN**

**UPDATE INVENTORY SET TOTAL\_BLOOD\_UNIT= TOTAL\_BLOOD\_UNIT - :NEW.BLOOD\_UNIT**

**WHERE PT\_BTYPE= :NEW.PT\_BTYPE**

**AND INVENTORY\_ID=:NEW.INVENTORY\_ID;**

**END;**



create or replace PROCEDURE GET\_INV\_ID

(

INPUT\_H\_ID IN NUMBER

, INPUT\_PT\_BTYPE IN VARCHAR2

, OUTPUT\_INV\_ID OUT NUMBER

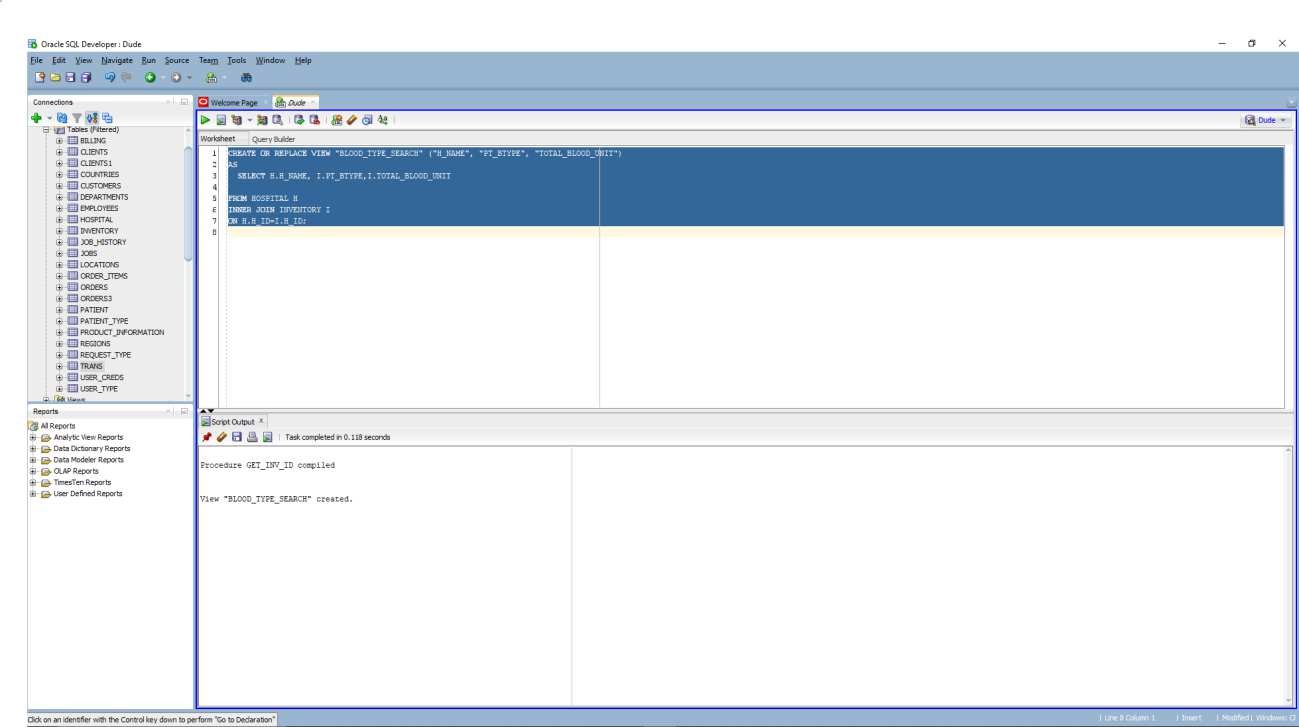
) AS

BEGIN

SELECT INVENTORY\_ID INTO OUTPUT\_INV\_ID FROM INVENTORY WHERE h\_id=INPUT\_H\_ID

AND pt\_btype=INPUT\_PT\_BTYPE;

END GET\_INV\_ID;



**CREATE OR REPLACE VIEW "BLOOD\_TYPE\_SEARCH" ("H\_NAME", "PT\_BTYPE", "TOTAL\_BLOOD\_UNIT")**

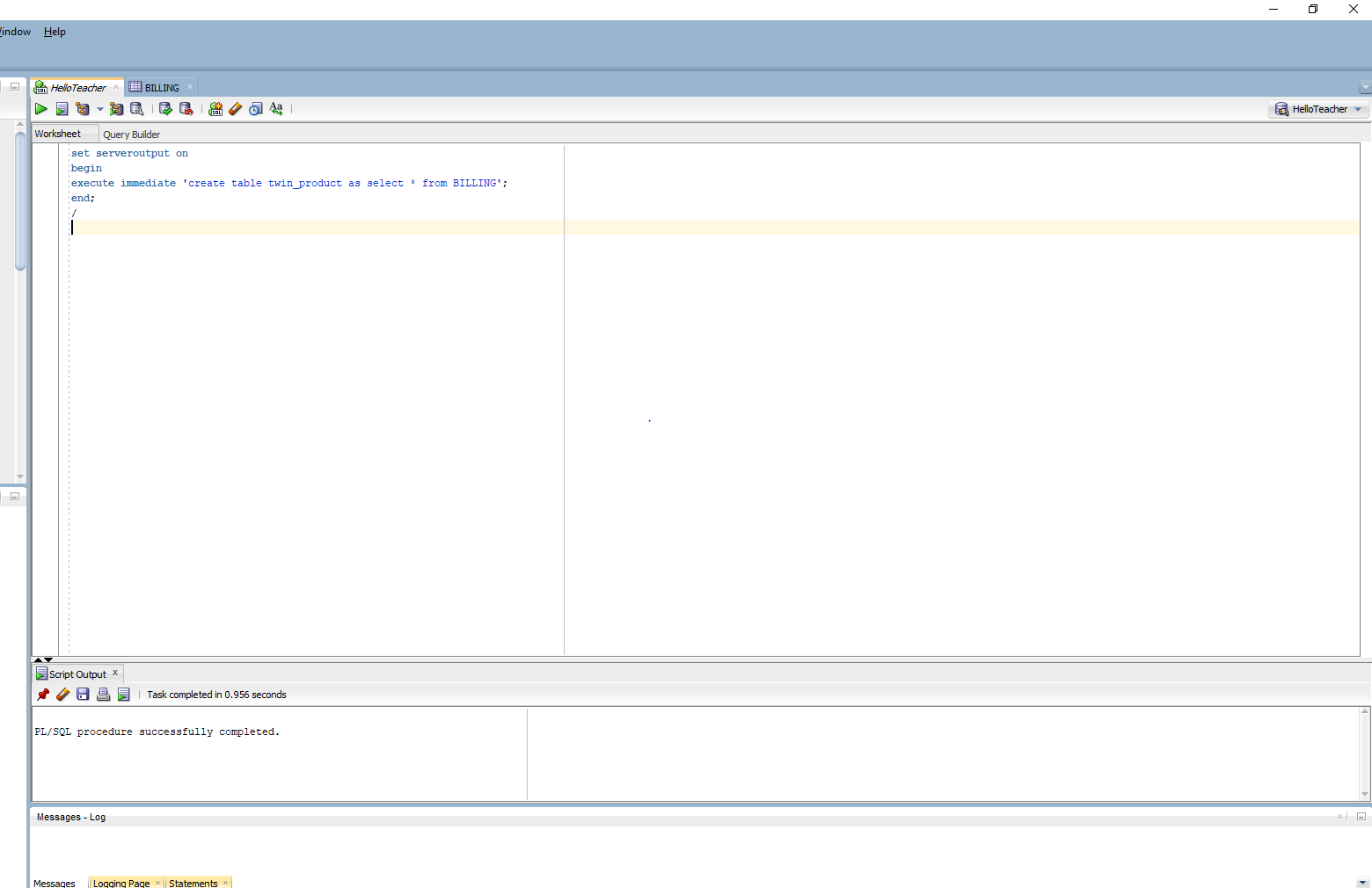
**AS**

**SELECT H.H\_NAME, I.PT\_BTYPE,I.TOTAL\_BLOOD\_UNIT**

**FROM HOSPITAL H**

**INNER JOIN INVENTORY I**

**ON H.H\_ID=I.H\_ID;**



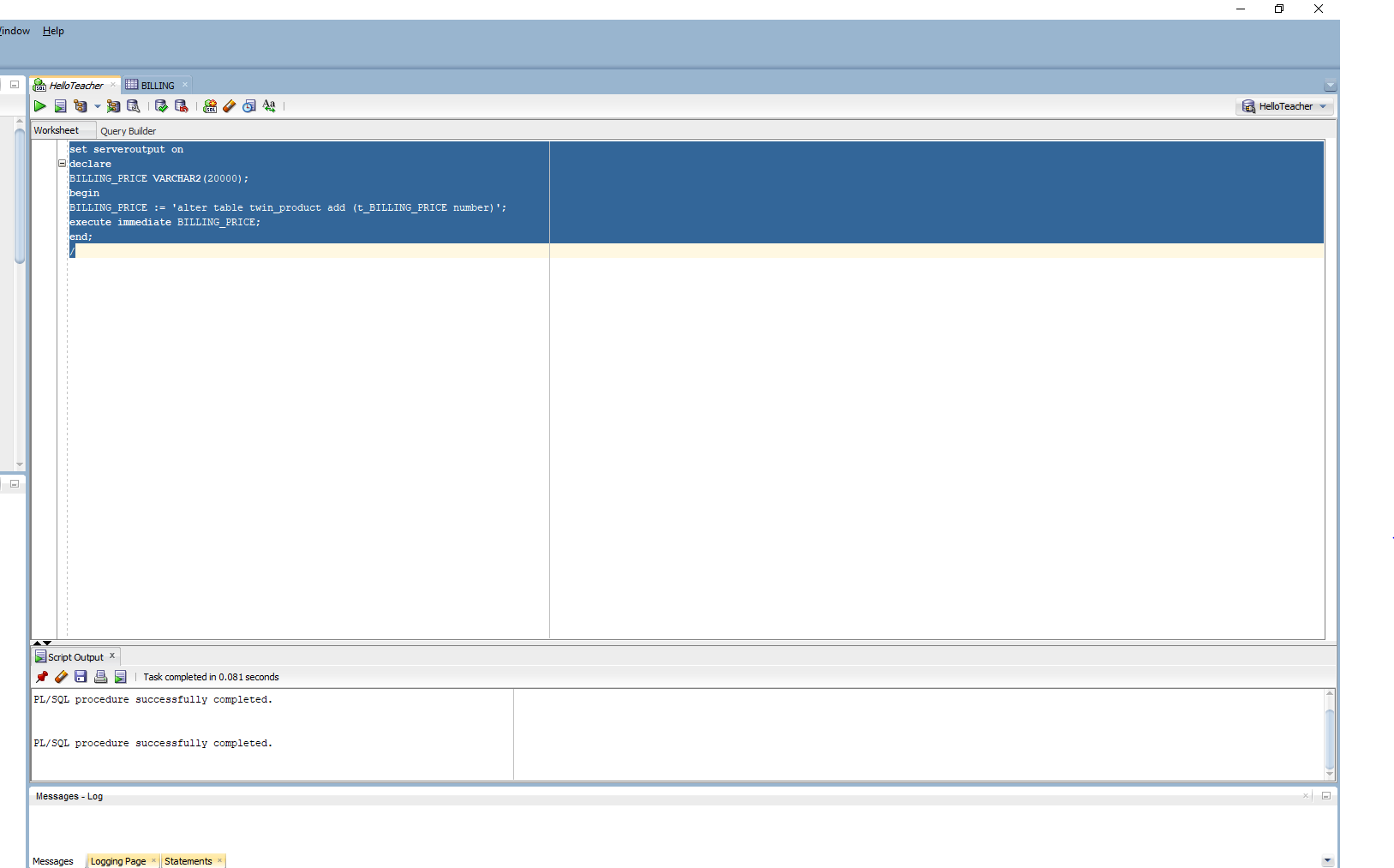
set serveroutput on

begin

execute immediate 'create table twin\_product as select \* from BILLING';

end;

/



set serveroutput on

declare

BILLING\_PRICE VARCHAR2(20000);

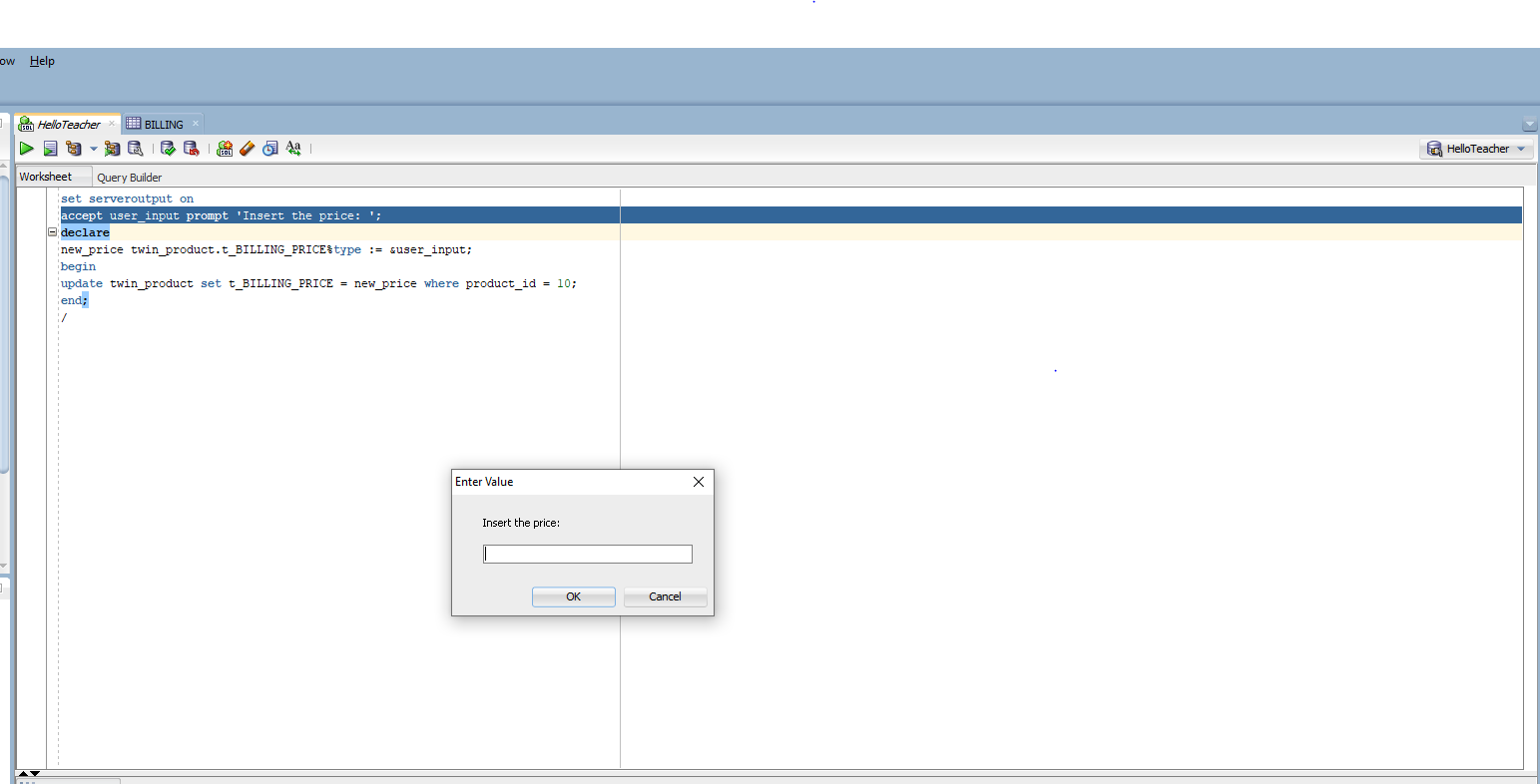
begin

BILLING\_PRICE := 'alter table twin\_product add (t\_BILLING\_PRICE number)';

execute immediate BILLING\_PRICE;

end;

/



set serveroutput on

accept user\_input prompt 'Insert the price: ';

declare

new\_price twin\_product.t\_BILLING\_PRICE%type := &user\_input;

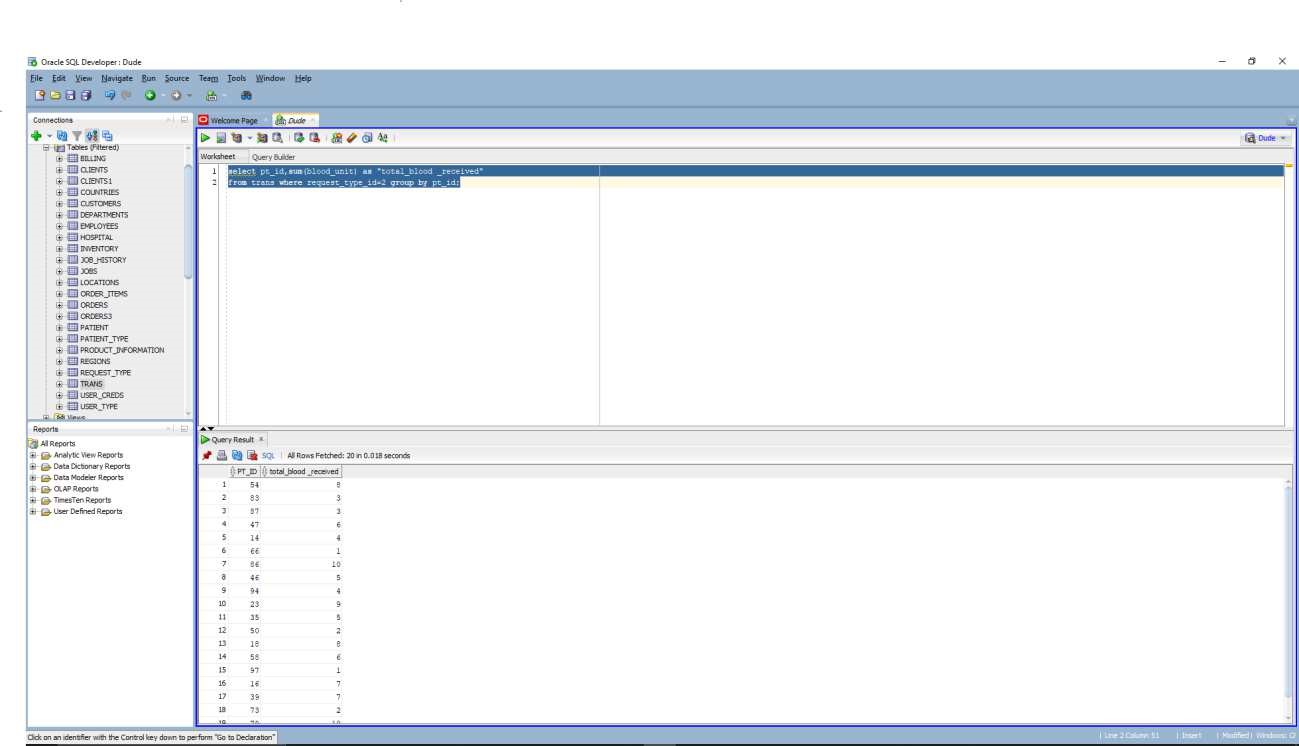
begin

update twin\_product set t\_BILLING\_PRICE = new\_price where product\_id = 10;

end;

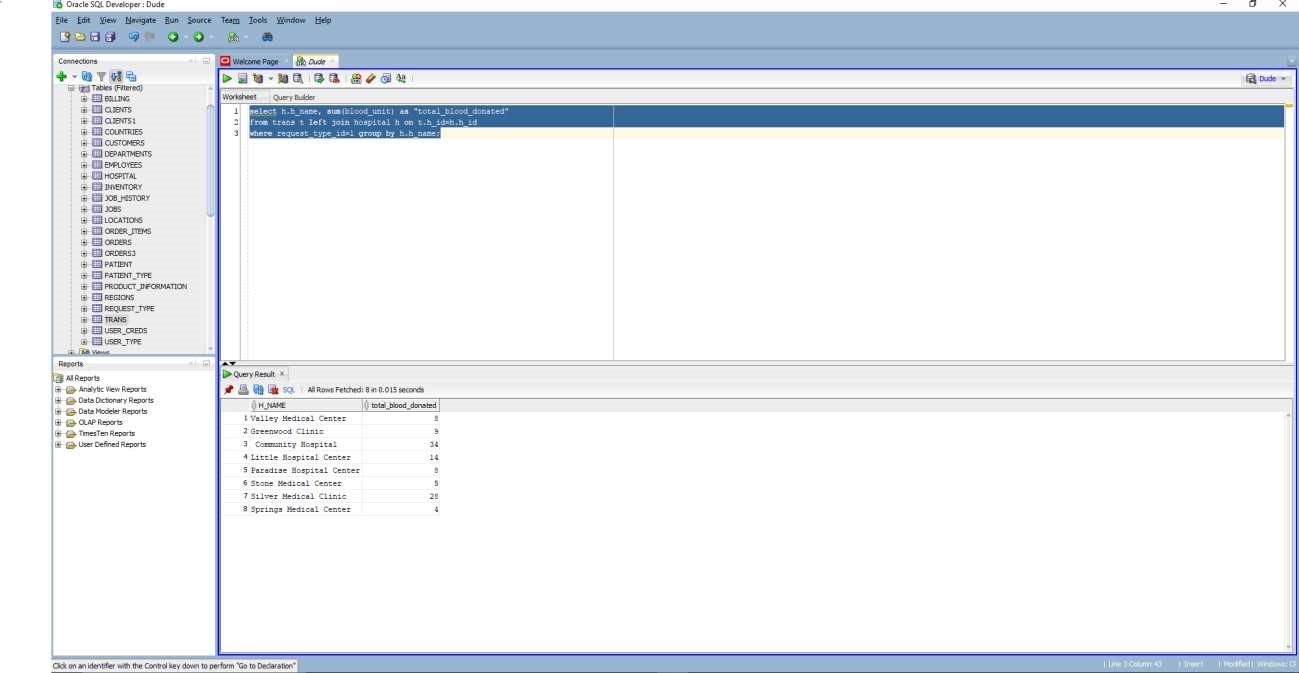
/

# Examples of varied queries:



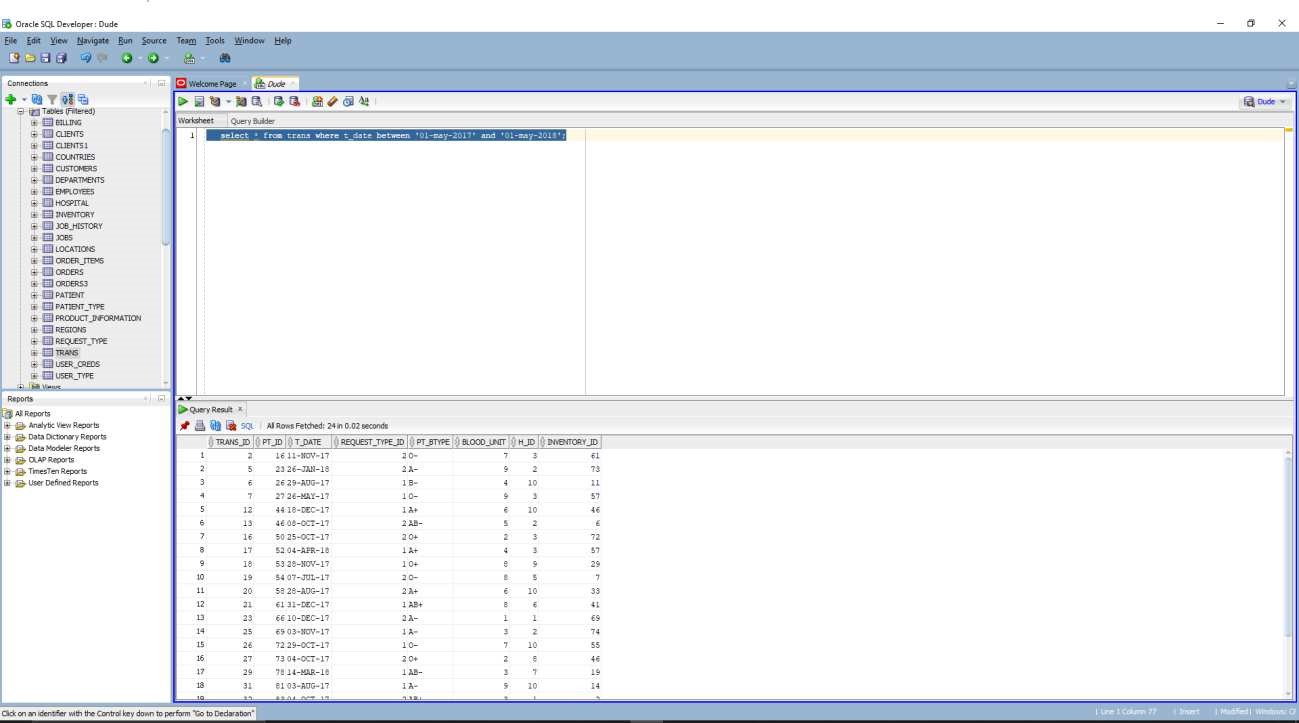
list of total blood units received by a patient

**select pt\_id,sum(blood\_unit) as "total\_blood \_received" from trans where request\_type\_id=2 group by pt\_id;**



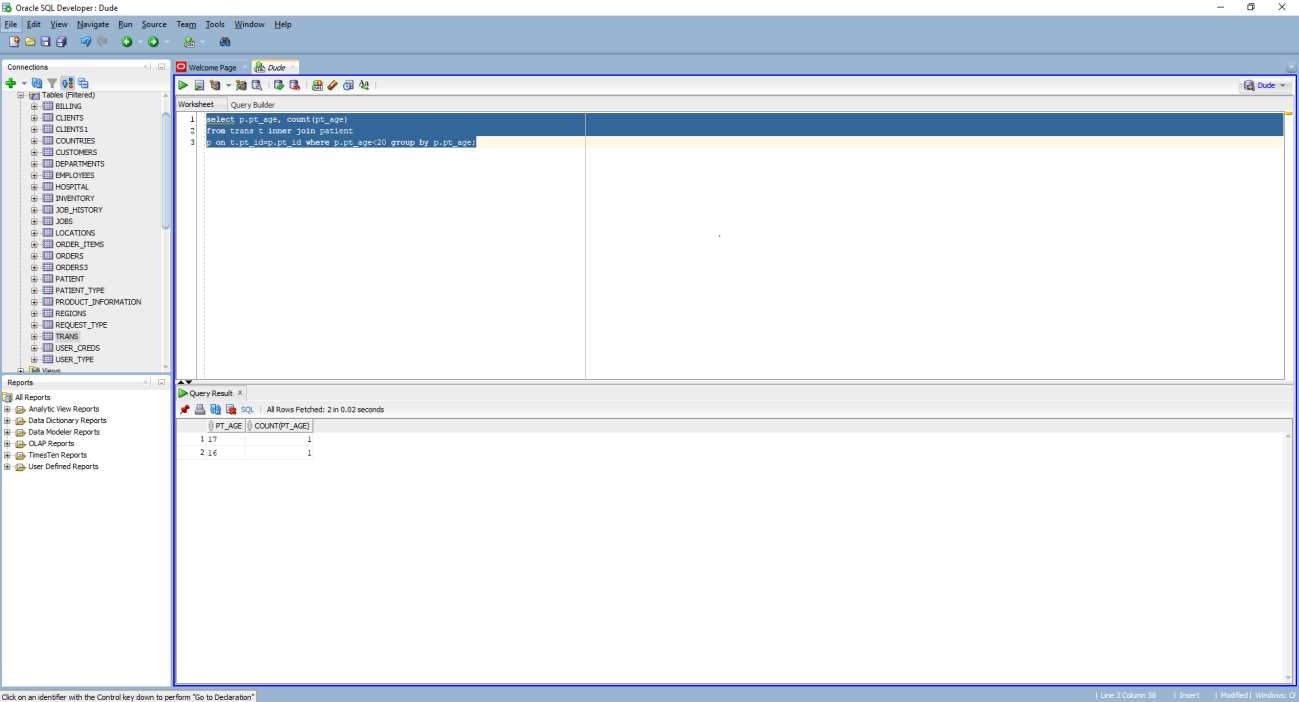
list of patients donating or receiving blood from respective hospitals **select h.h\_name, sum(blood\_unit) as "total\_blood\_donated" from trans t left join hospital h on t.h\_id=h.h\_id**

**where request\_type\_id=1 group by h.h\_name;**

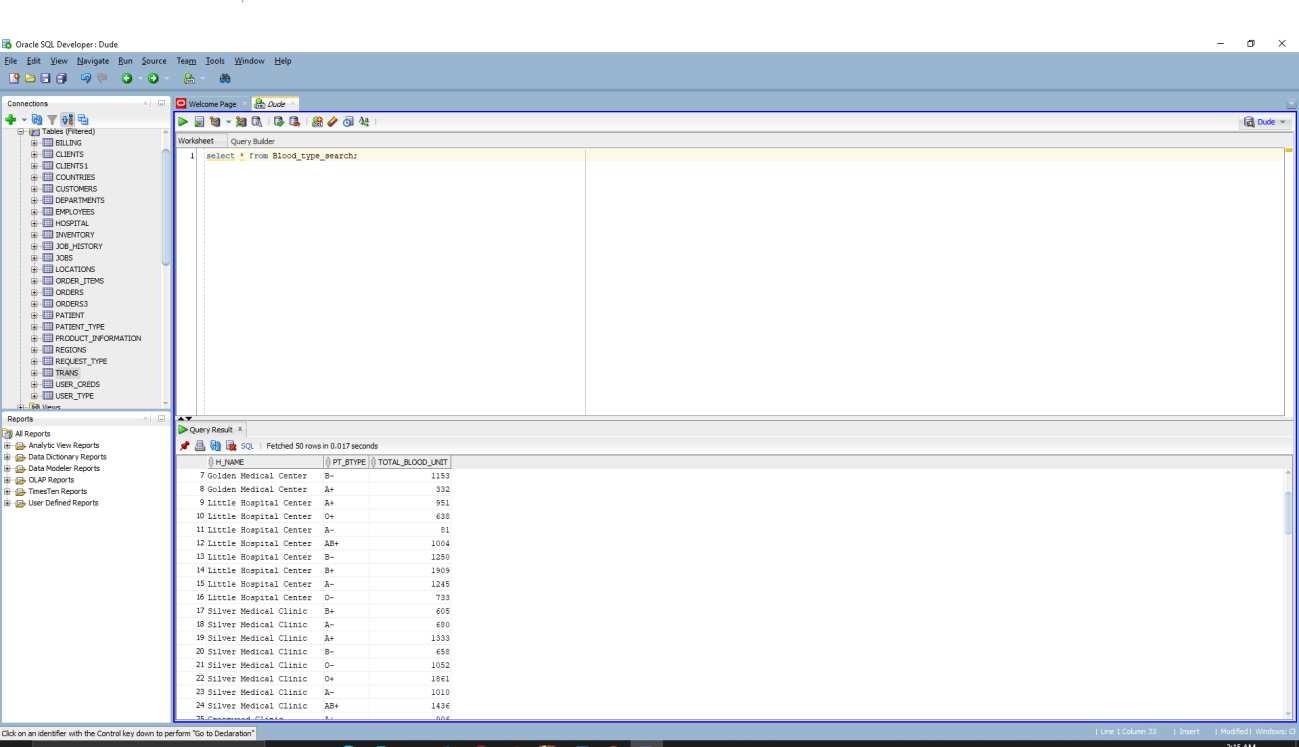


transactions between 2 dates

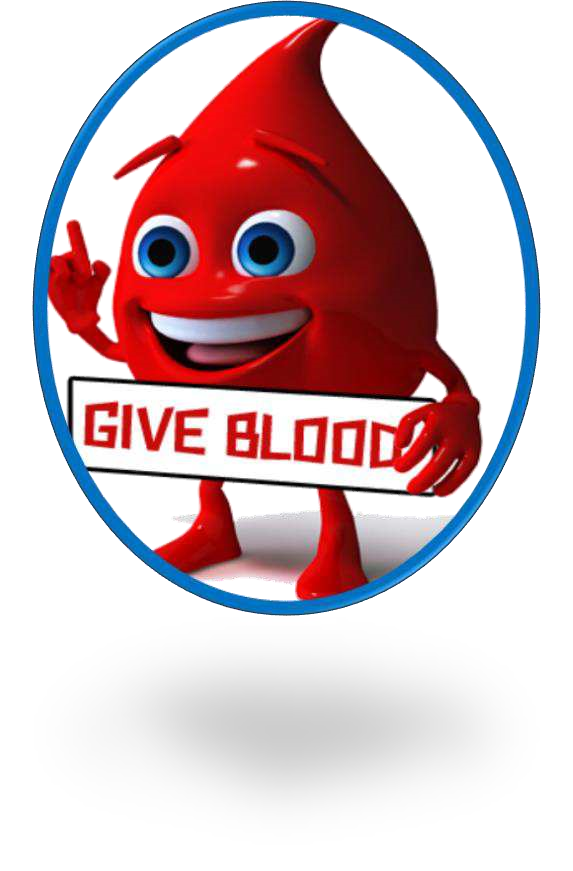
**select \* from trans where t\_date between '01-may-2017' and '01-may-2018';**



total transactions done by population with age <20; **select p.pt\_age, count(pt\_age) from trans t inner join patient p on t.pt\_id=p.pt\_id where p.pt\_age<20 group by p.pt\_age;**



**select \* from Blood\_type\_search;**

**THANKS!**