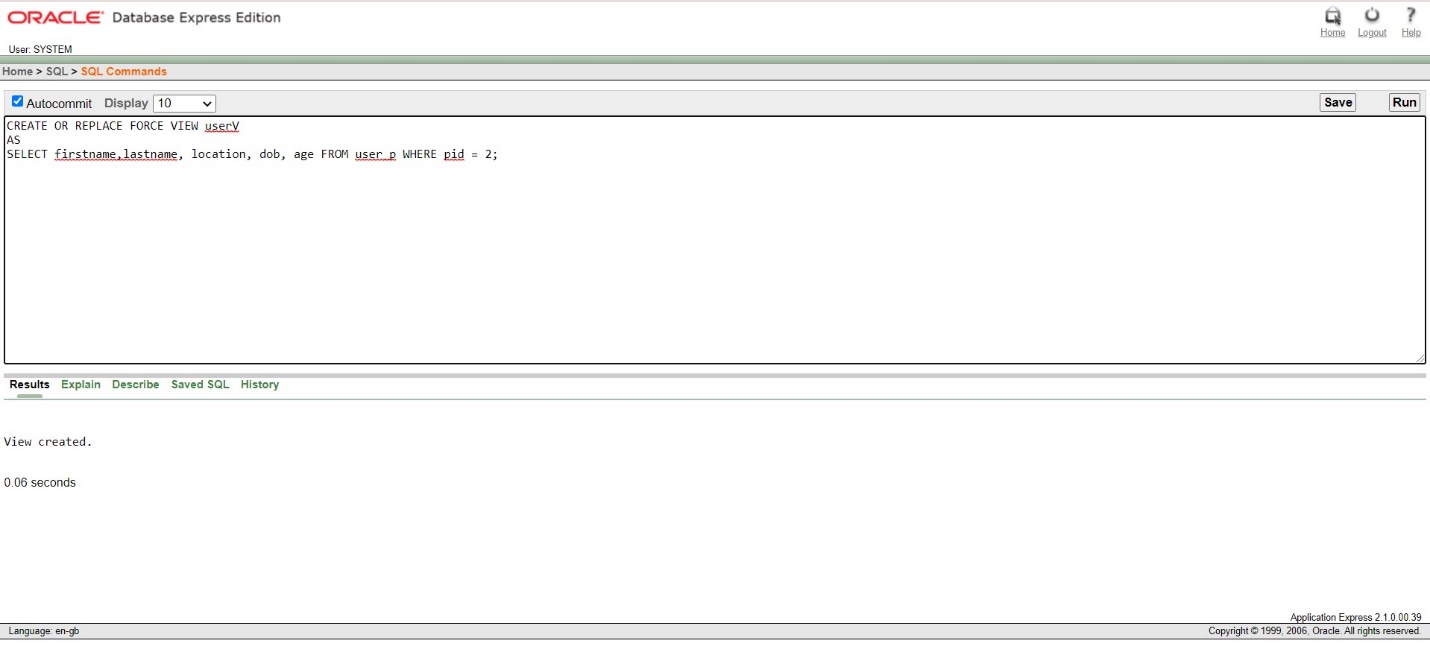
6. View

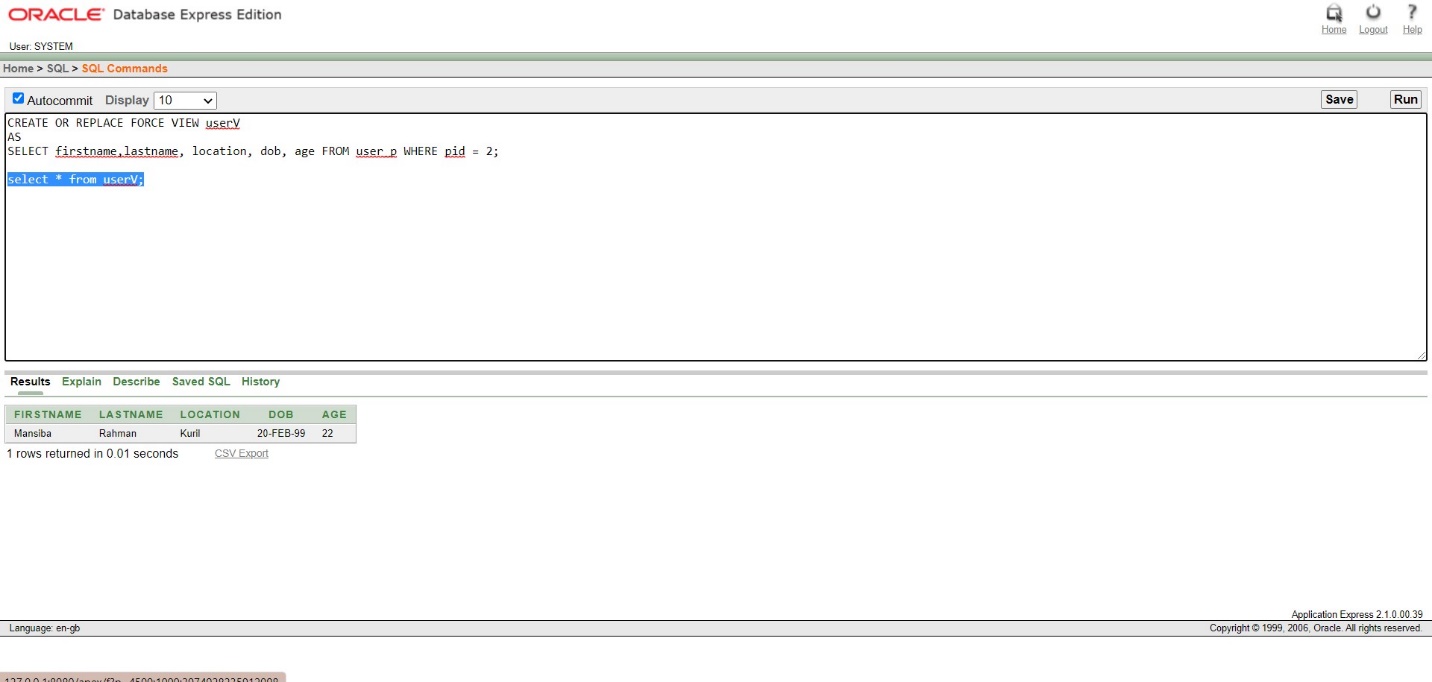
**1.**

CREATE OR REPLACE FORCE VIEW userV

AS

SELECT firstname,lastname, location, dob, age FROM user\_p WHERE pid = 2;





**2.**

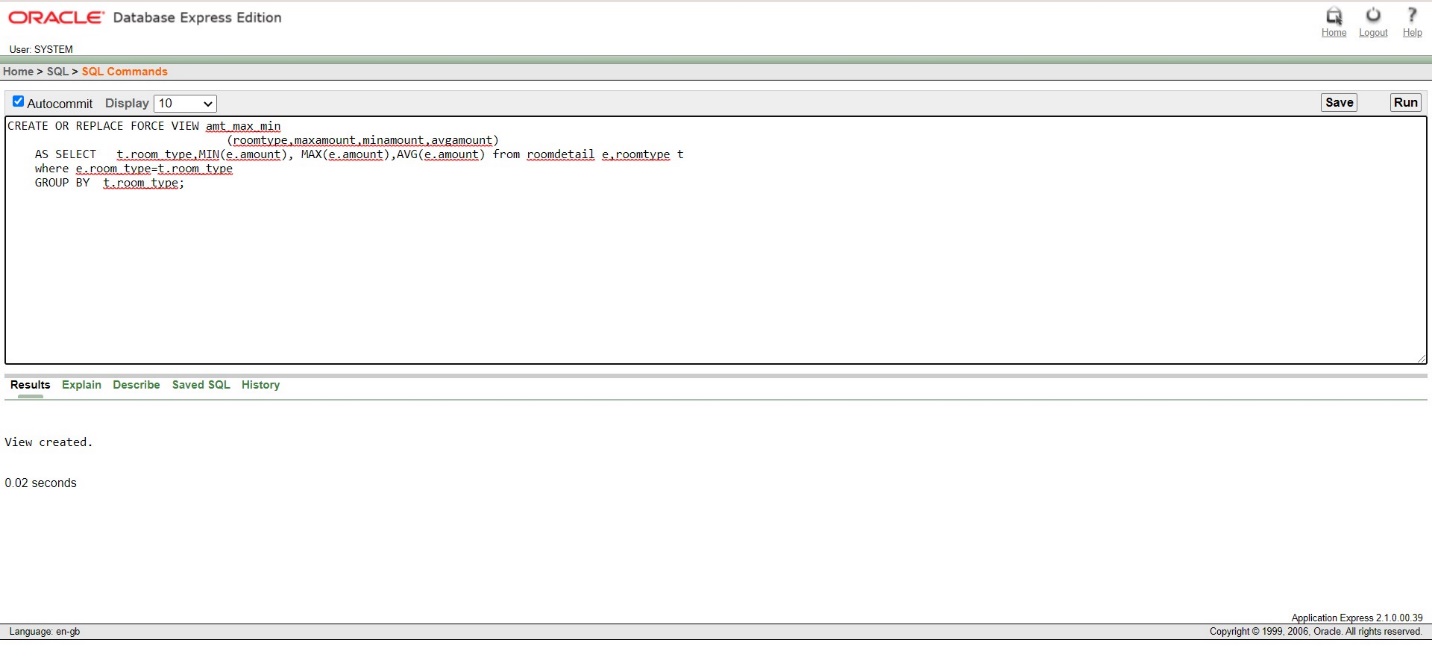
CREATE OR REPLACE FORCE VIEW amt\_max\_min

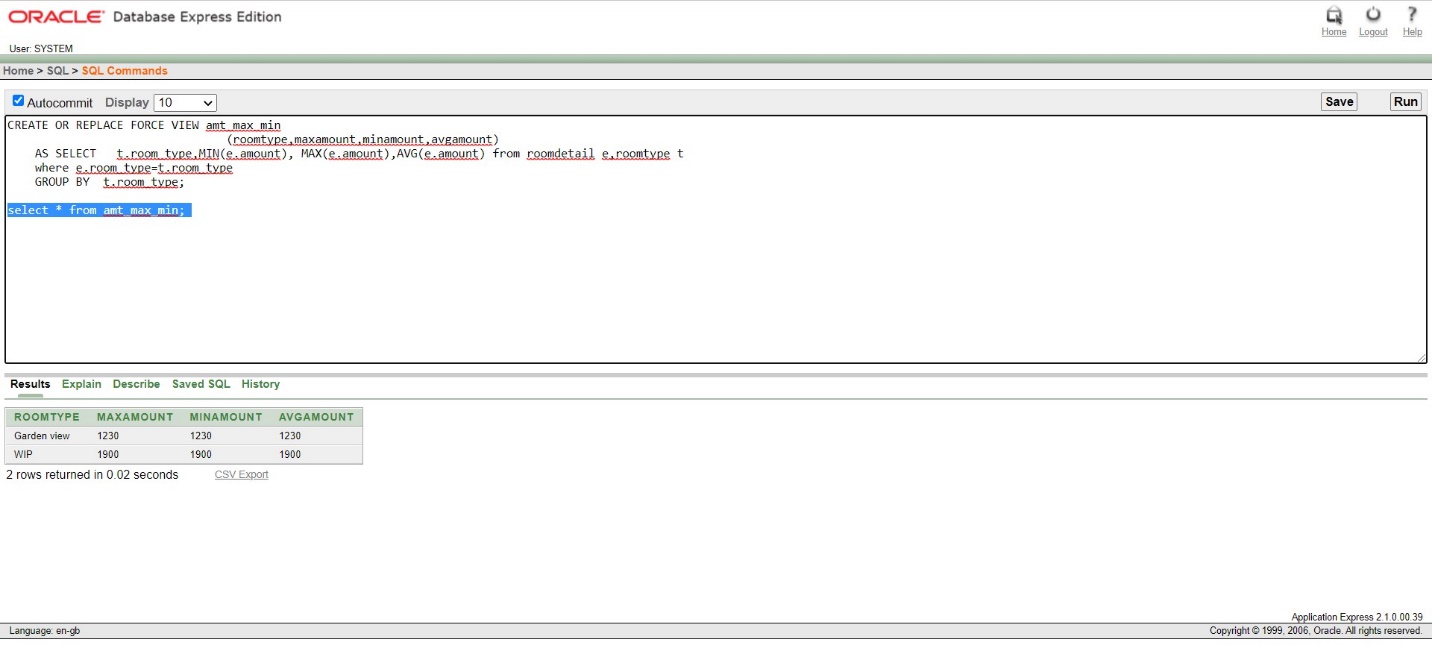
(roomtype,maxamount,minamount,avgamount)

AS SELECT t.room\_type,MIN(e.amount), MAX(e.amount),AVG(e.amount) from roomdetail e,roomtype t

where e.room\_type=t.room\_type

GROUP BY t.room\_type;





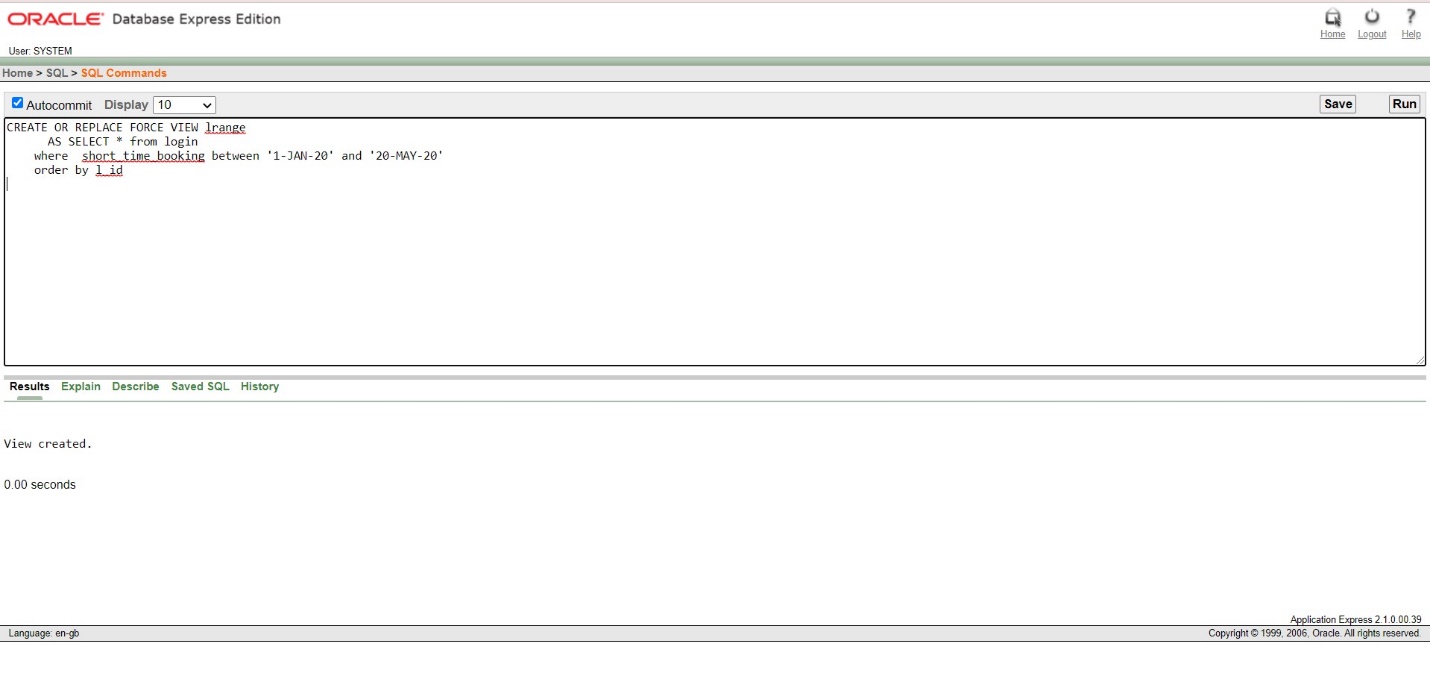
**3.**

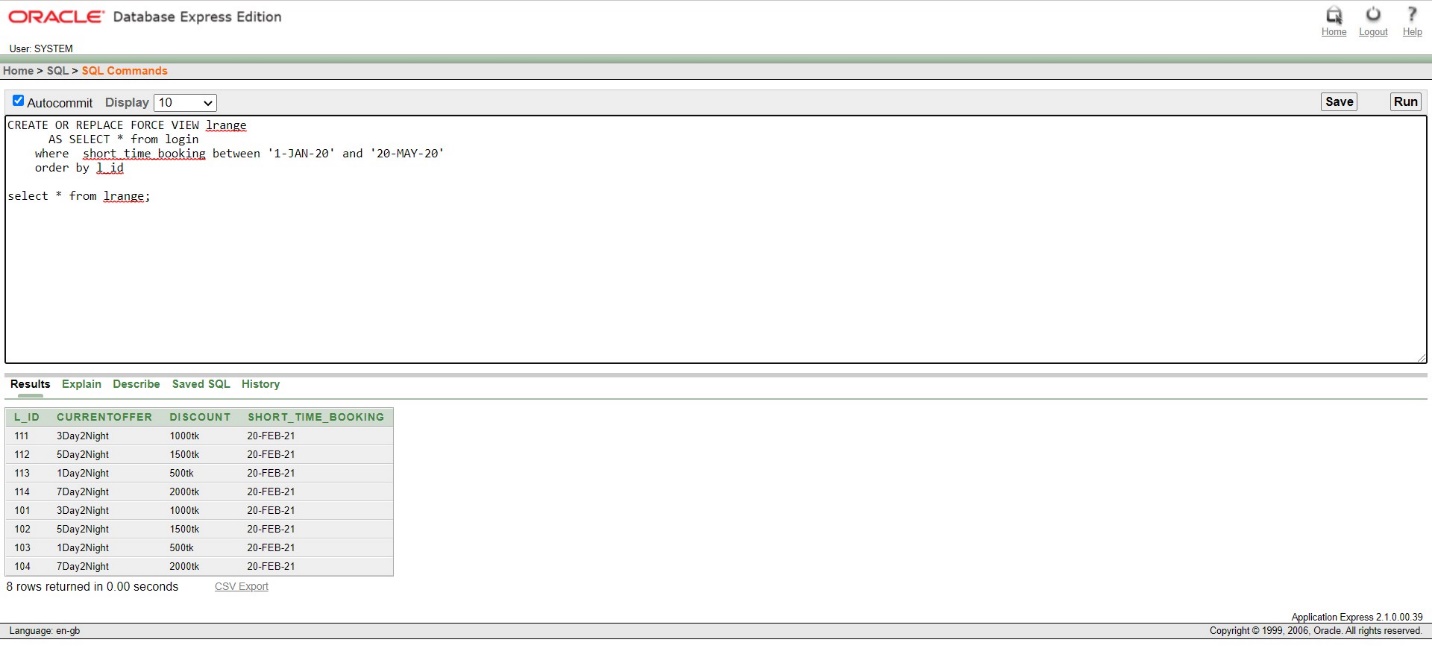
CREATE OR REPLACE FORCE VIEW lrange

AS SELECT \* from login

where short\_time\_booking between '1-JAN-20' and '20-MAY-20'

order by l\_id





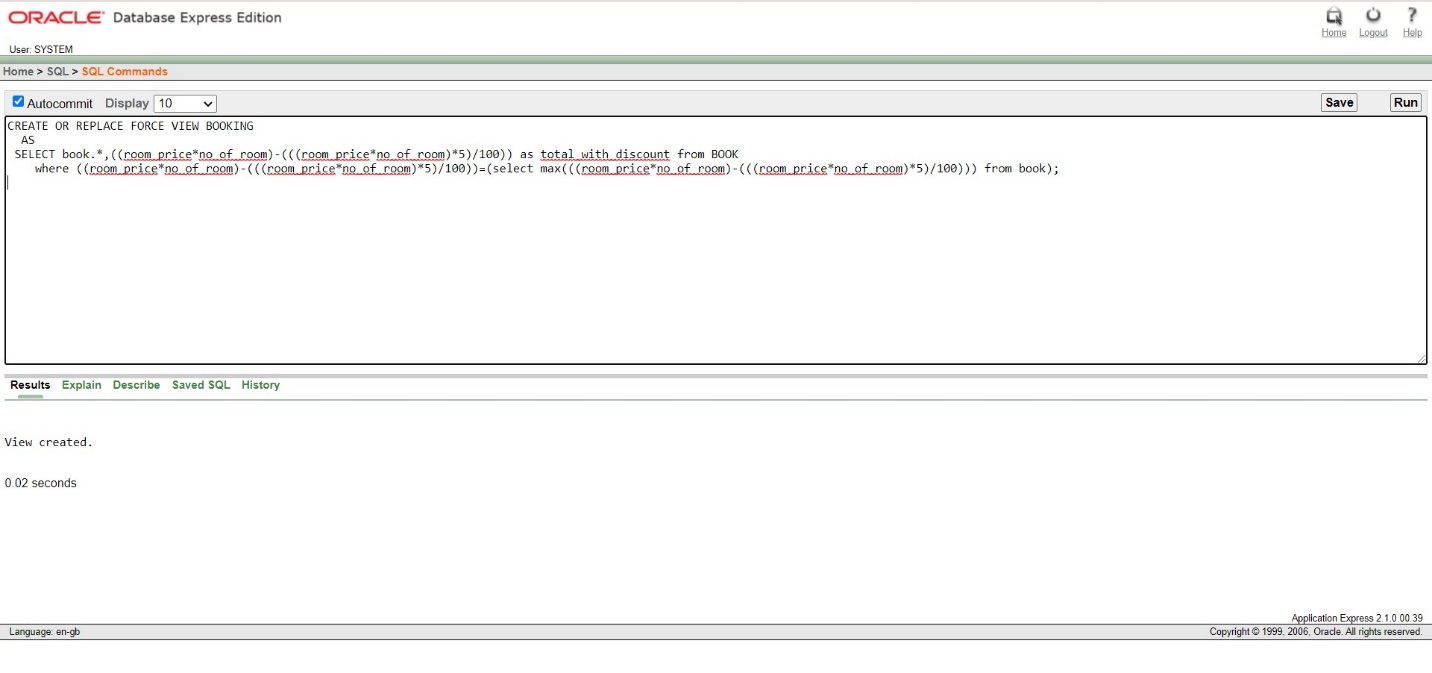
**4.**

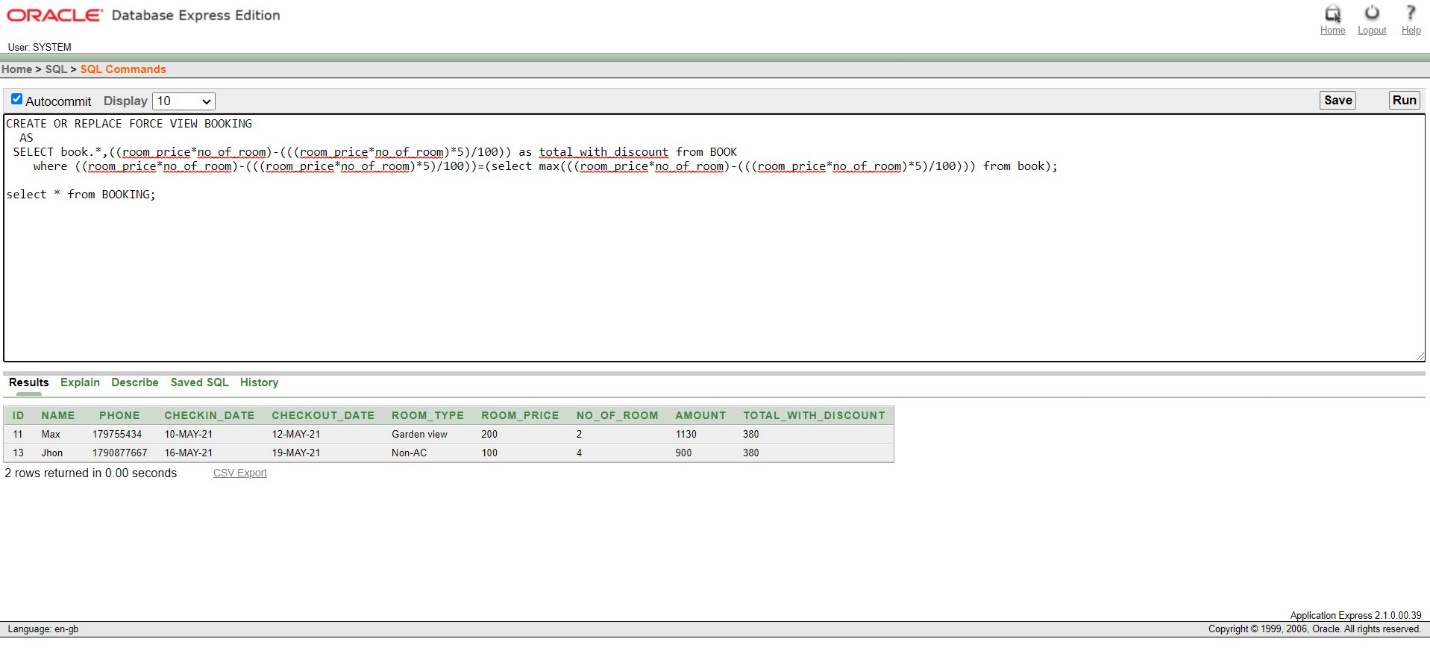
CREATE OR REPLACE FORCE VIEW BOOKING

AS

SELECT book.\*,((room\_price\*no\_of\_room)-(((room\_price\*no\_of\_room)\*5)/100)) as total\_with\_discount from BOOK

where ((room\_price\*no\_of\_room)-(((room\_price\*no\_of\_room)\*5)/100))=(select max(((room\_price\*no\_of\_room)-(((room\_price\*no\_of\_room)\*5)/100))) from book);





7. **Procedures and Function:**

**1.**

create or replace procedure Users\_details(U\_no IN user\_p.u\_id%type,u\_fn OUT user\_p.firstname%type,u\_ln OUT user\_p.lastname%type,u\_loc OUT user\_p.location%type,u\_dob OUT user\_p.dob%type,u\_age OUT user\_p.age%type)

is

Begin

select firstname,lastname,location,dob,age into u\_fn,u\_ln,u\_loc,u\_dob,u\_age from user\_p where u\_id=U\_no;

END;

Declare

Uno user\_p.u\_id%type:=:User\_number;

Ufn user\_p.firstname%type;

Uln user\_p.lastname%type;

Uloc user\_p.location%type;

Udob user\_p.dob%type;

Uage user\_p.age%type;

Begin

Users\_details(Uno,Ufn,Uln,Uloc,Udob,Uage);

dbms\_output.put\_Line('FirstName:'||Ufn);

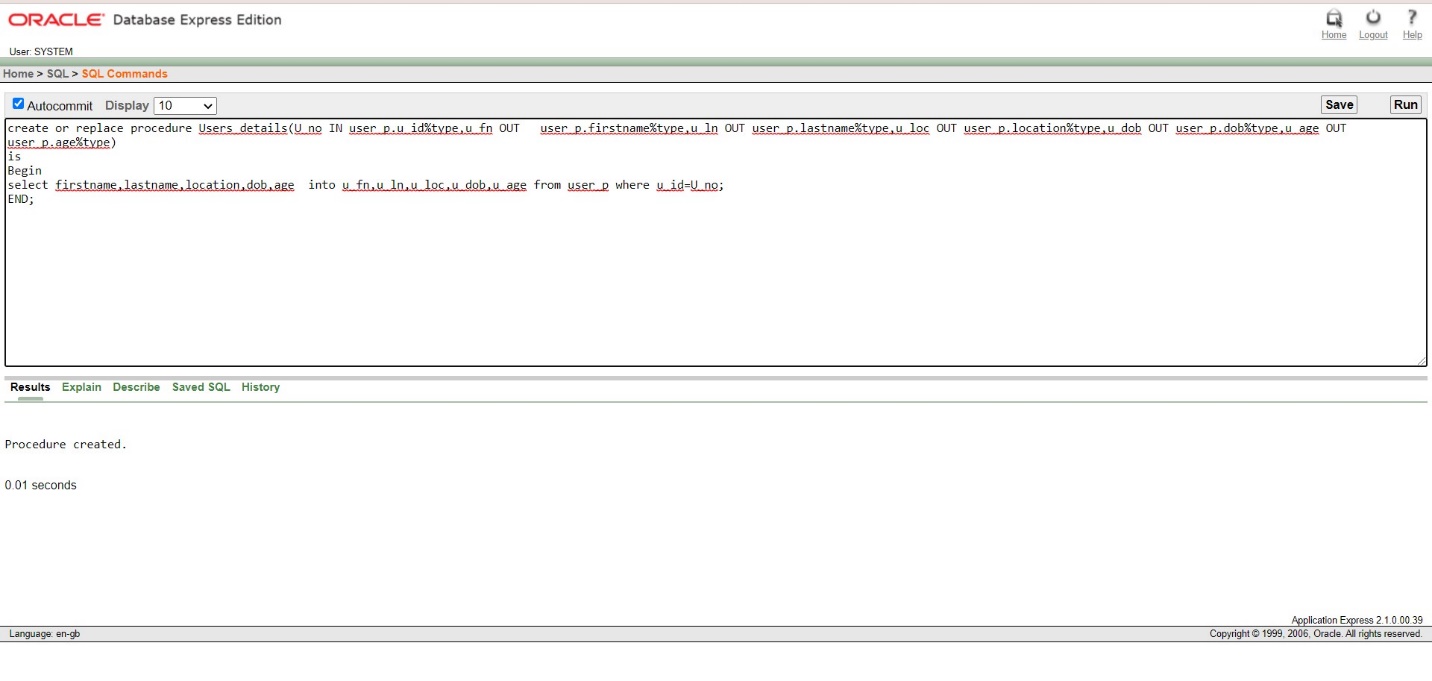
dbms\_output.put\_Line('LastName:'||Uln);

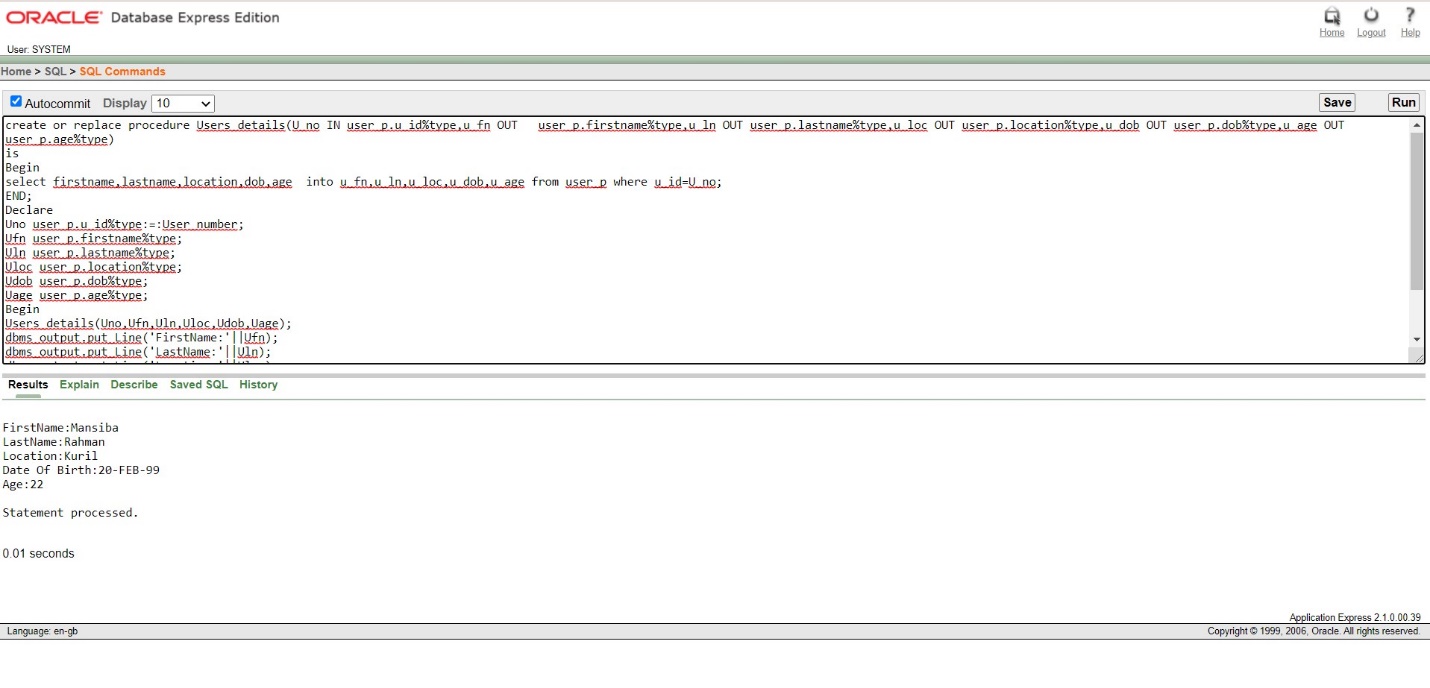
dbms\_output.put\_Line('Location:'||Uloc);

dbms\_output.put\_Line('Date Of Birth:'||Udob);

dbms\_output.put\_Line('Age:'||Uage);

END;





**2.**

create or replace procedure customer\_count

is

u\_id project.id%type;

Begin

select count(id) into u\_id from project;

dbms\_output.put\_line('Total: '||u\_id||'USER');

END;

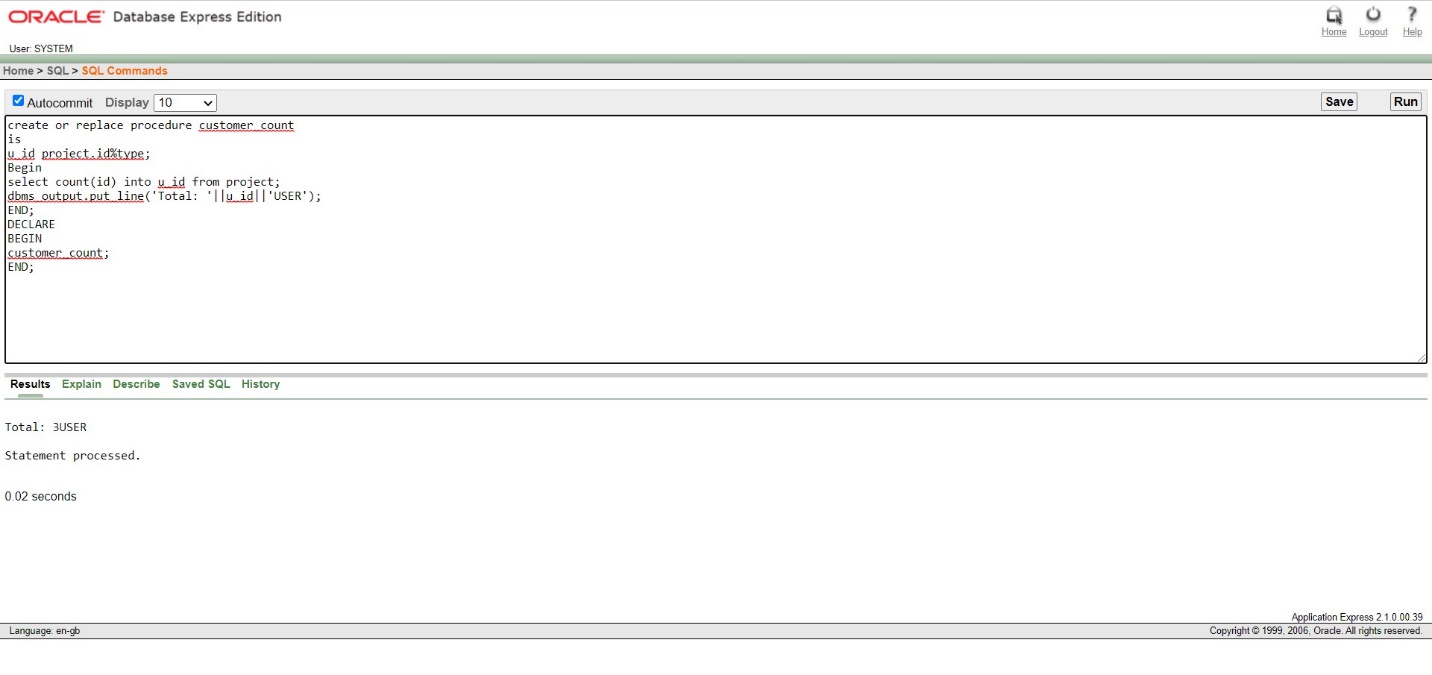
DECLARE

BEGIN

customer\_count;

END;





**3.**

CREATE OR REPLACE PROCEDURE Booking\_list

IS

v\_id NUMBER(4);

v\_name VARCHAR2(10);

v\_phone NUMBER(15);

v\_checkin\_date date;

v\_checkout\_date date;

v\_room\_type varchar2(50);

v\_room\_price varchar2(10);

CURSOR book\_cur IS

SELECT id, name,phone,checkin\_date, checkout\_date ,room\_type,room\_price FROM book ORDER BY id;

BEGIN

OPEN book\_cur;

DBMS\_OUTPUT.PUT\_LINE('Booking ID NAME Phone no CHECKIN DATE CHECKOUT DATE ROOM TYPE ROOM PRICE');

DBMS\_OUTPUT.PUT\_LINE('---------- ------ ------- ------------- ------------- --------- ----------');

LOOP

FETCH book\_cur INTO v\_id, v\_name,v\_phone,v\_checkin\_date,v\_checkout\_date,v\_room\_type, v\_room\_price;

EXIT WHEN book\_cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_id ||' '|| v\_name ||' '||v\_phone||' '||v\_checkin\_date||' '||v\_checkout\_date||' '||v\_room\_type||' '|| v\_room\_price);

END LOOP;

CLOSE book\_cur;

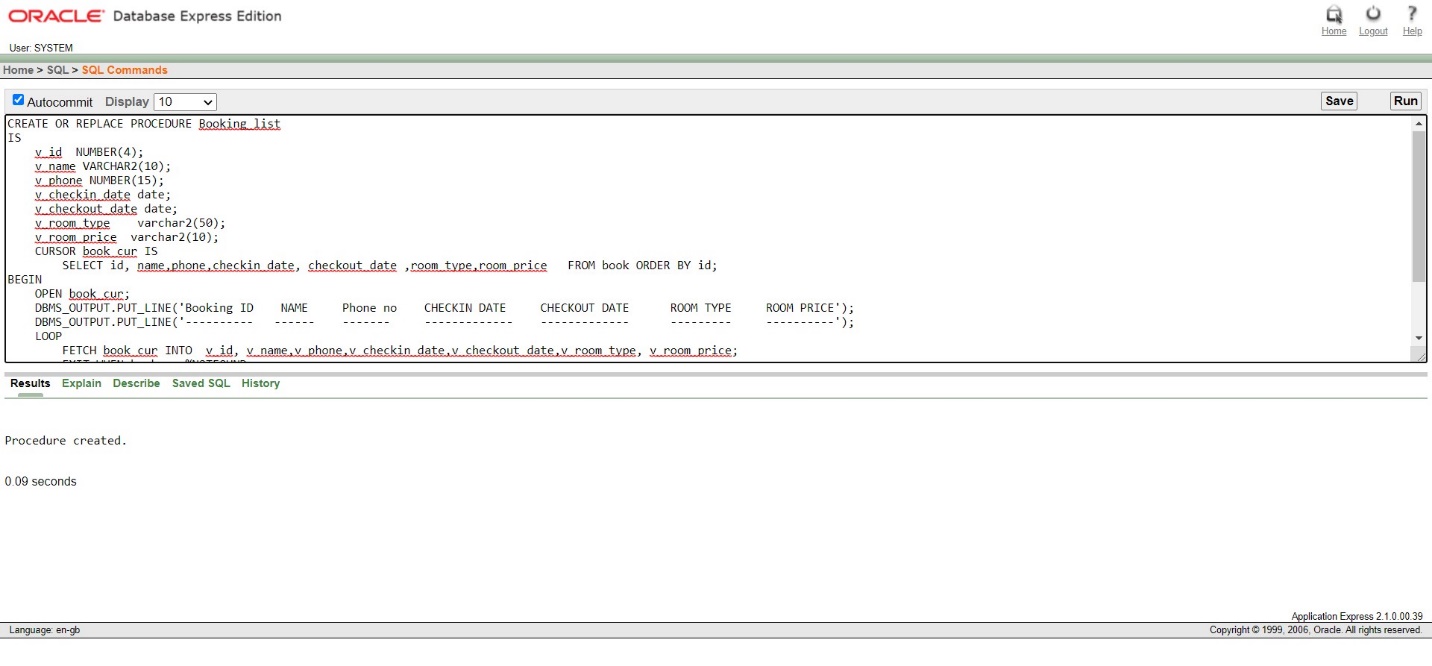
END;

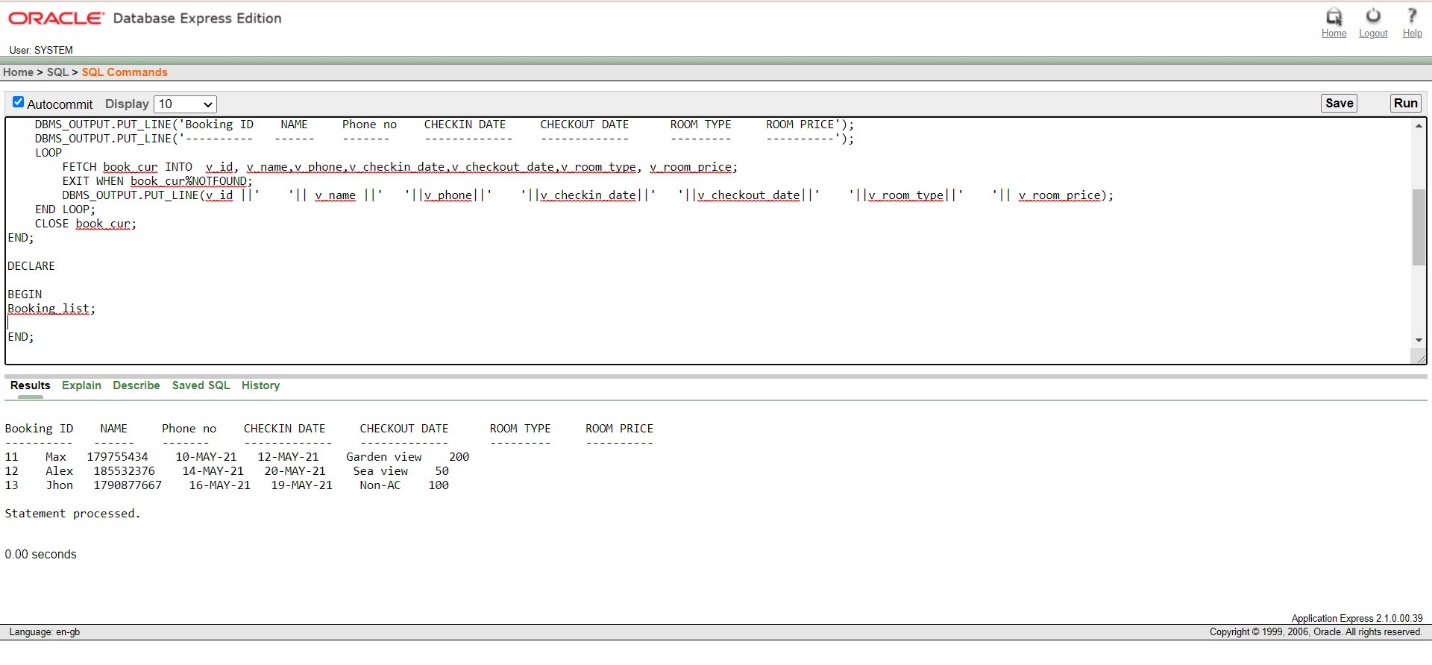
DECLARE

BEGIN

Booking\_list;

END;





**4.**

CREATE OR REPLACE PROCEDURE update\_roomtype(up\_roomtype IN roomtype.room\_type%TYPE)

IS

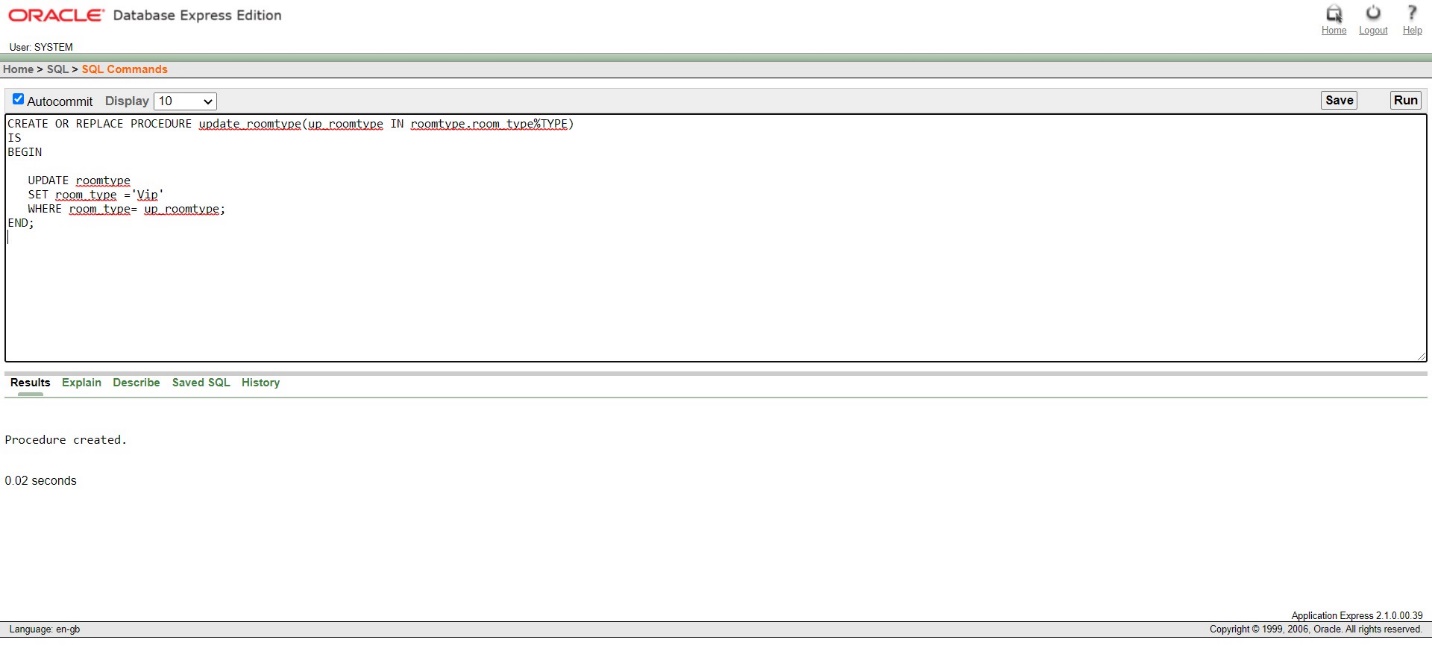
BEGIN

UPDATE roomtype

SET room\_type ='Vip'

WHERE room\_type= up\_roomtype;

END;



**5.**

create or replace function r\_percentage(uno roomdetail.id%type)

return roomdetail.amount%type

is

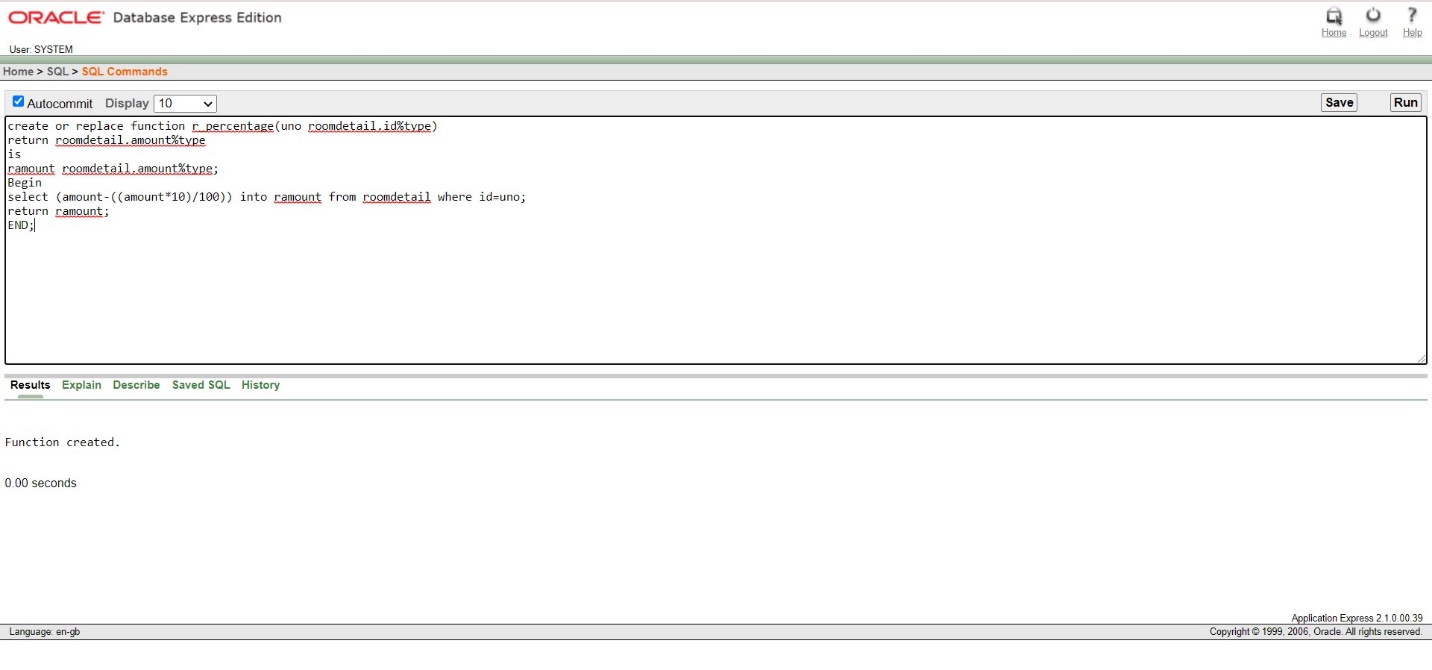
ramount roomdetail.amount%type;

Begin

select (amount-((amount\*10)/100)) into ramount from roomdetail where id=uno;

return ramount;

END;



**6.**

create or replace function transportdetails(t\_no transport.b\_id%type)

return transport.b\_name%type

is

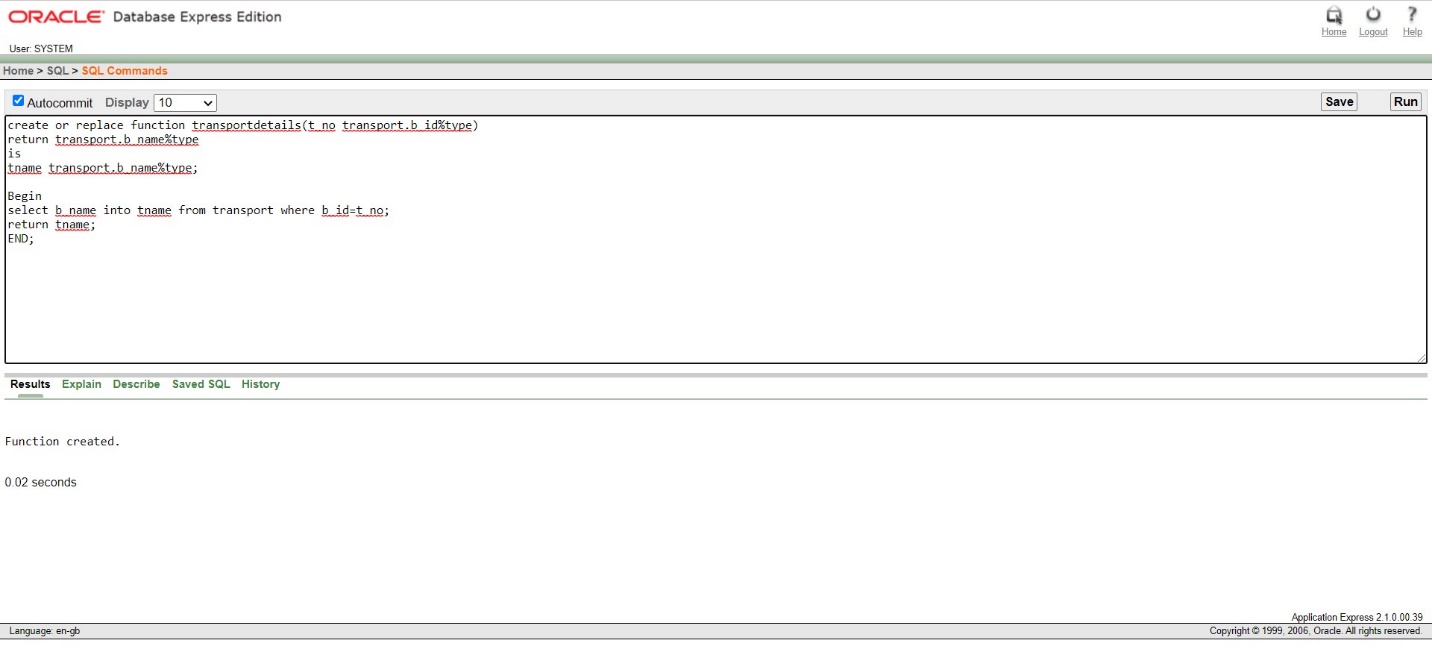
tname transport.b\_name%type;

Begin

select b\_name into tname from transport where b\_id=t\_no;

return tname;

END;



8. **Trigger:**

**1.**

Table:

CREATE TABLE sys\_evnt\_audit

(

event\_type VARCHAR2(30),

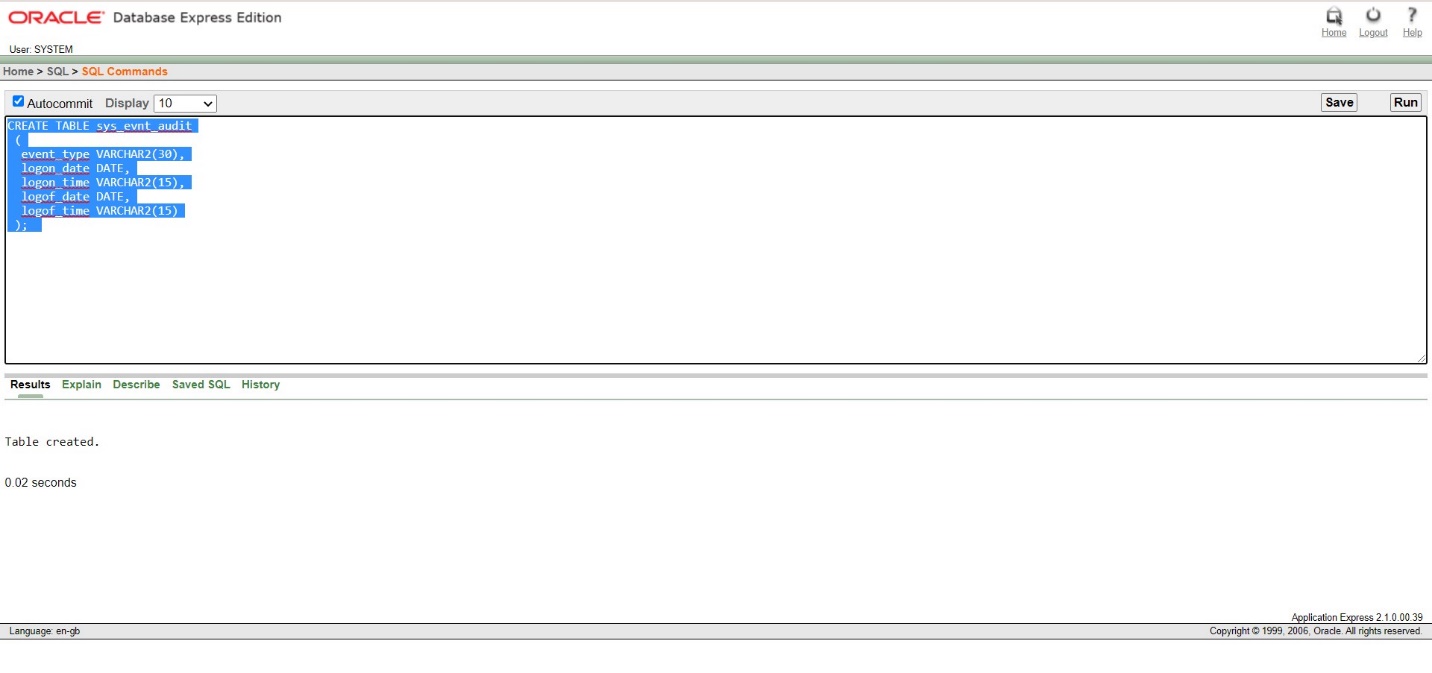
logon\_date DATE,

logon\_time VARCHAR2(15),

logof\_date DATE,

logof\_time VARCHAR2(15)

);



-----------------------------------------trigger--------------------

CREATE OR REPLACE TRIGGER sys\_lgon\_audit

AFTER LOGON ON SCHEMA

BEGIN

INSERT INTO sys\_evnt\_audit VALUES(

ora\_sysevent,

sysdate,

TO\_CHAR(sysdate, 'hh24:mi:ss'),

NULL,

NULL

);

COMMIT;

END;



**2.**

Table:

CREATE TABLE sys\_evnt\_logof

(

event\_type VARCHAR2(30),

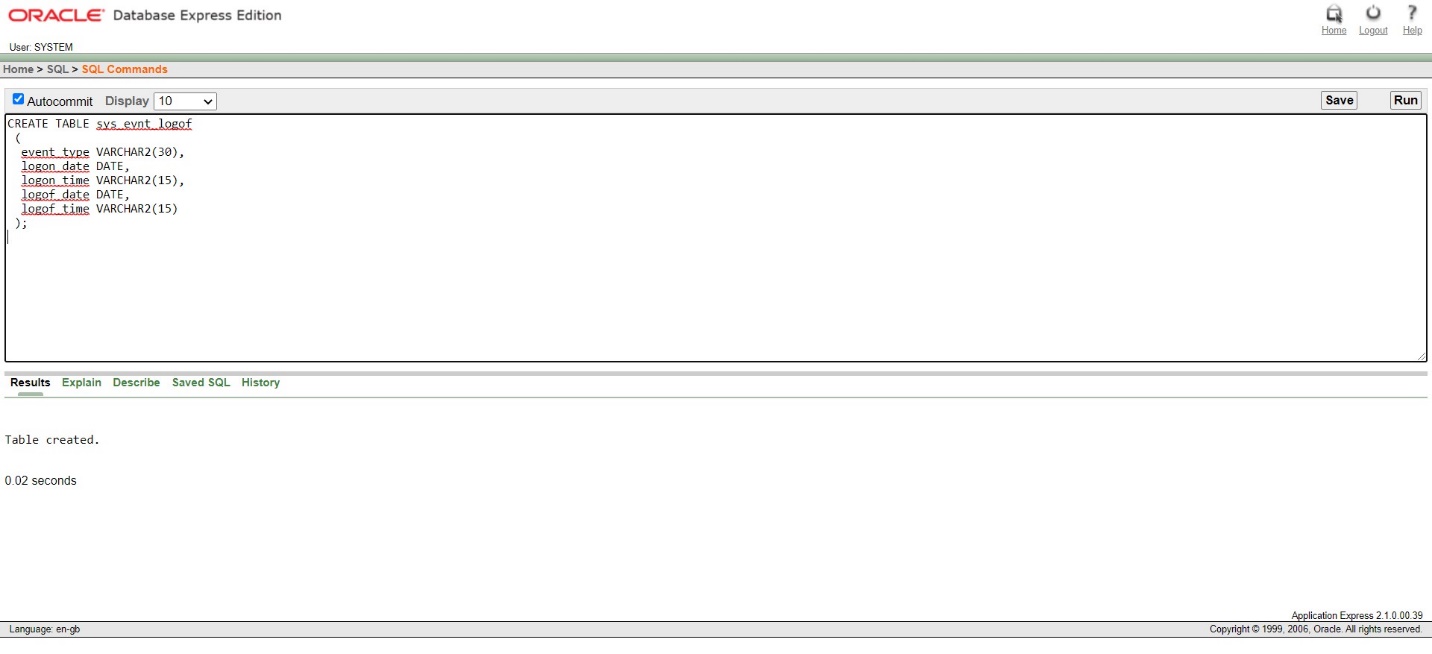
logon\_date DATE,

logon\_time VARCHAR2(15),

logof\_date DATE,

logof\_time VARCHAR2(15)

);



-------------------Trigger----------------------

CREATE OR REPLACE TRIGGER log\_off\_audit

BEFORE LOGOFF ON SCHEMA

BEGIN

INSERT INTO sys\_evnt\_logof VALUES

(

ora\_sysevent,

NULL,

NULL,

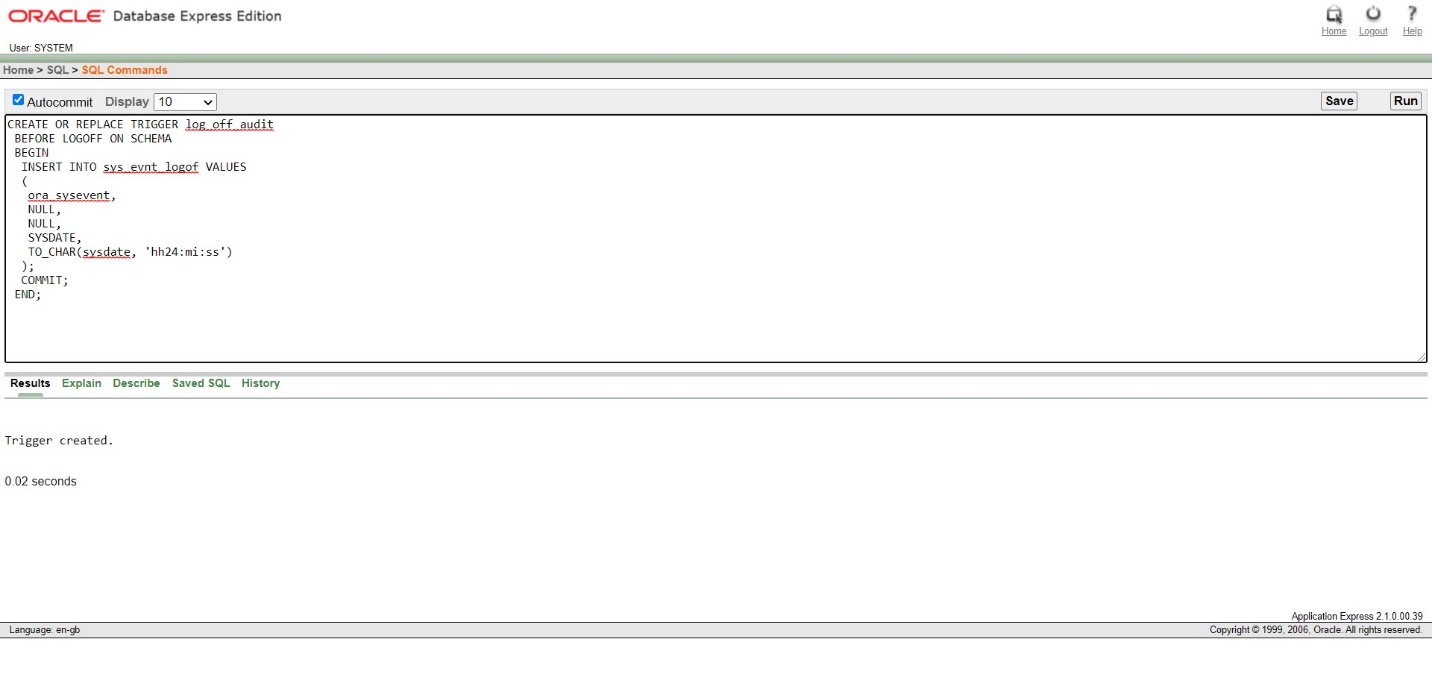
SYSDATE,

TO\_CHAR(sysdate, 'hh24:mi:ss')

);

COMMIT;

END;



3.

CREATE TABLE project\_audit (

new\_name varchar2(30),

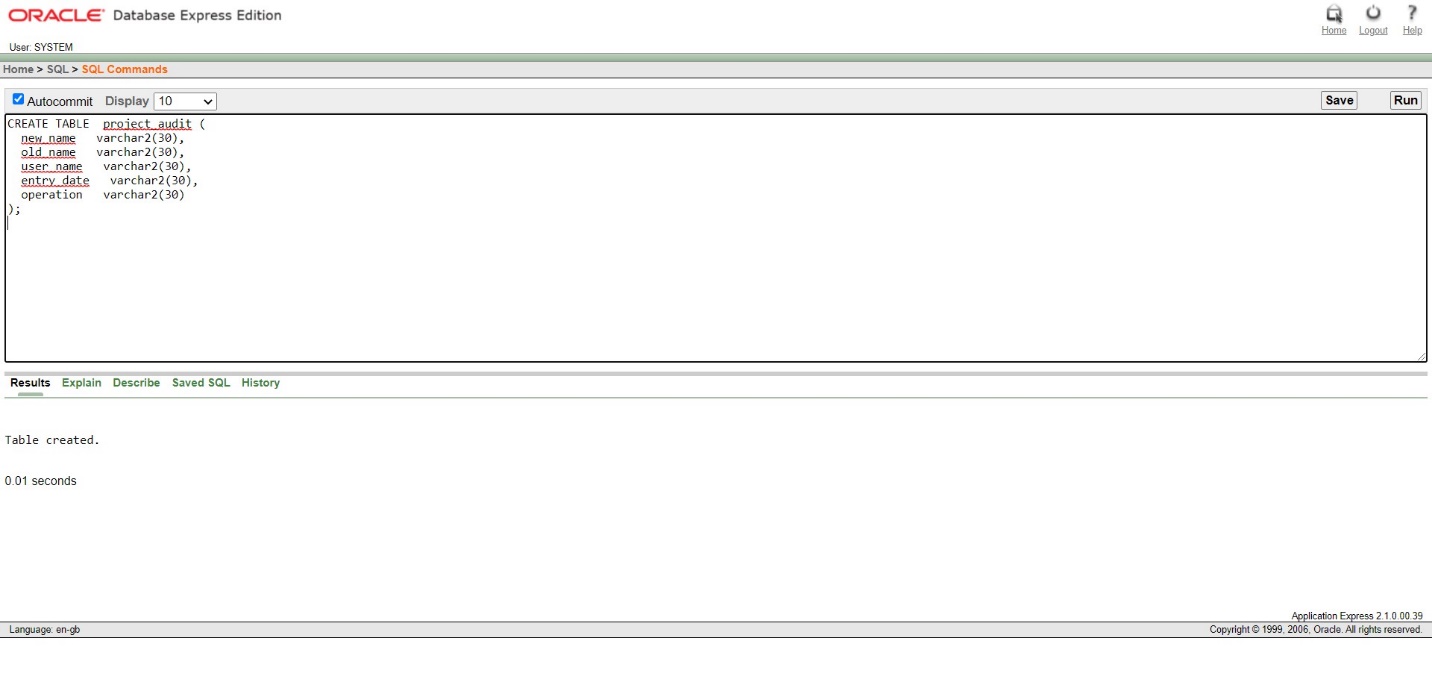
old\_name varchar2(30),

user\_name varchar2(30),

entry\_date varchar2(30),

operation varchar2(30)

);



--------------------------trigger-----------------------

CREATE OR REPLACE TRIGGER reg\_audit

BEFORE INSERT ON project

FOR EACH ROW

DECLARE

v\_user varchar2 (30);

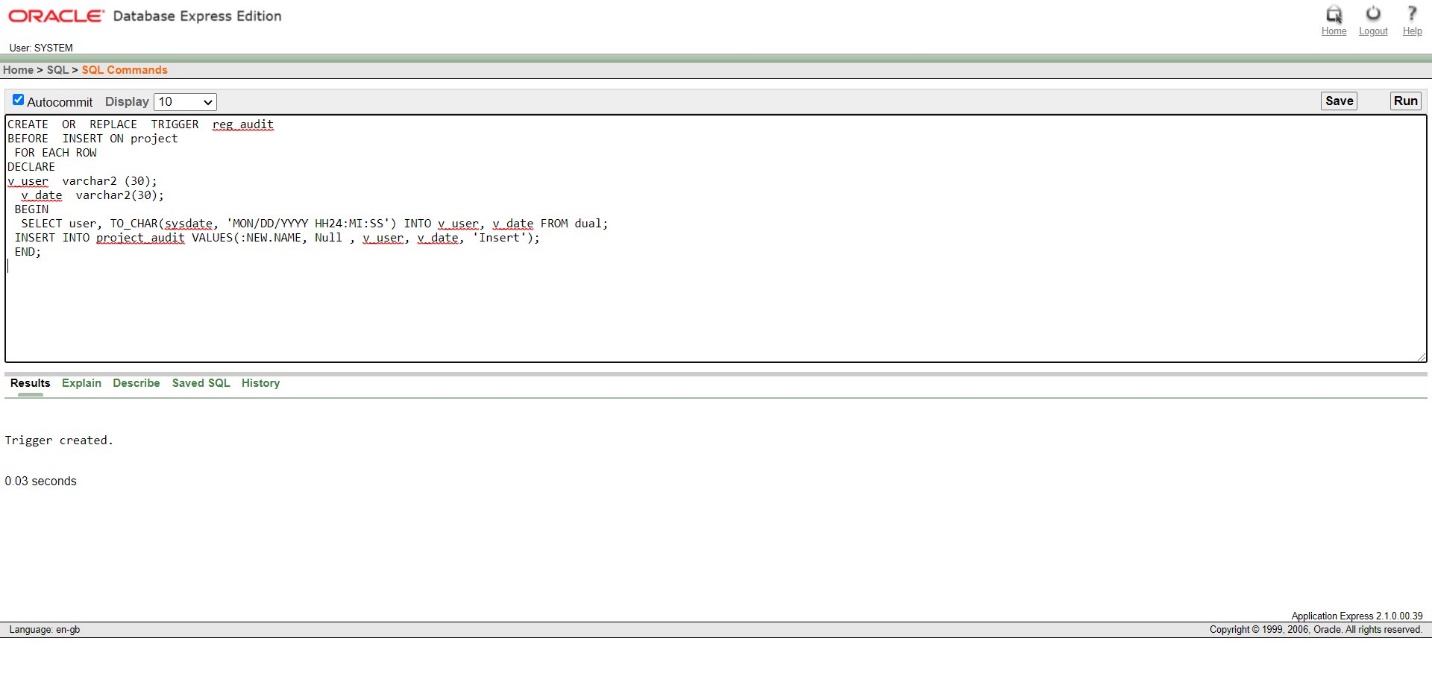
v\_date varchar2(30);

BEGIN

SELECT user, TO\_CHAR(sysdate, 'MON/DD/YYYY HH24:MI:SS') INTO v\_user, v\_date FROM dual;

INSERT INTO project\_audit VALUES(:NEW.NAME, Null , v\_user, v\_date, 'Insert');

END;



4.

--------------------trigger------------

CREATE SEQUENCE project\_seq;

CREATE OR REPLACE TRIGGER project\_id

BEFORE INSERT ON project

FOR EACH ROW

WHEN (new.id IS NULL)

BEGIN

SELECT project\_seq.NEXTVAL

INTO :new.id

FROM dual;

END;

