**TV & MOVIES DATA BASE**

**(TMDb)**

**SUBJECT: DATABASE MANAGEMENT SYSTEM**

**DHARAMSINH DESAI UNIVERSITY**

**FACULTY OF TECHNOLOGY, NADIAD**



**PREPARED BY:** **GUIDED BY:**

CHANGRANI YASH (IT-12) PROF.SUNIL K VITHLANI DESAI JANAM (IT-20)

INDEX

1.Certificate………………………………..

2. Acknowledgement………………………

3. System Overview………………………..

4. E.R.Diagram…………………………….

5. Database Schema/Table Definition……..

6. Implementation………………………….

7. Output…………………………………..

8. Bibliography………..…………………..

**DHARAMSINH DESAI UNIVERSITY**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that **CHANGRANI YASH** and **DESAI JANAM** student(s) of B.Tech. Semester V (Information Technology) have completed their semester project work titled “**(TMDb)**” in subject Database Management System satisfactorily in partial fulfillment of requirement during academic year 2017-2018.

**PROF. SUNIL K VITHLANI PROF.R.S.Chhajed**

**(Assistant Proffesor.) (Head of dept.)**

**Acknowledgements**

On completion of project we would like to express our sincere thanks to all those who have guided, advised, inspired and supported during our project work.

Every work that one completes successfully stands on the constants encouragement, good will and support of the people around. We, hereby, avail this opportunity to express our heartfelt gratitude to a number of people who extended their valuable time, full support and cooperation in developing this project.

We are heartily thankful to the qualified staff of the centre and especially to our lab faculty **Prof. SUNIL K VITHLANI**. We believe that her computer expertise and valuable guidance have made it possible to present such nice project report.

Thanking you.

**Yours Sincerely,**

**CHANGRANI YASH**(IT - 12)

**DESAI JANAM** (IT - 20)

SYSTEM OVERVIEW

This is a project based on pl/sql language. In this, a user can shop books online on website. Every website maintains its own database system which provides all the facilities. Customer can retrieve data of his choice easily. Then the total amount that has to be paid is calculated automatically.

It also contains many procedures to manage database. All updates done on data items is reflected on database. It contains Triggers for auto incrementing primary keys and also for managing quantities of products in warehouse.

Advantges:

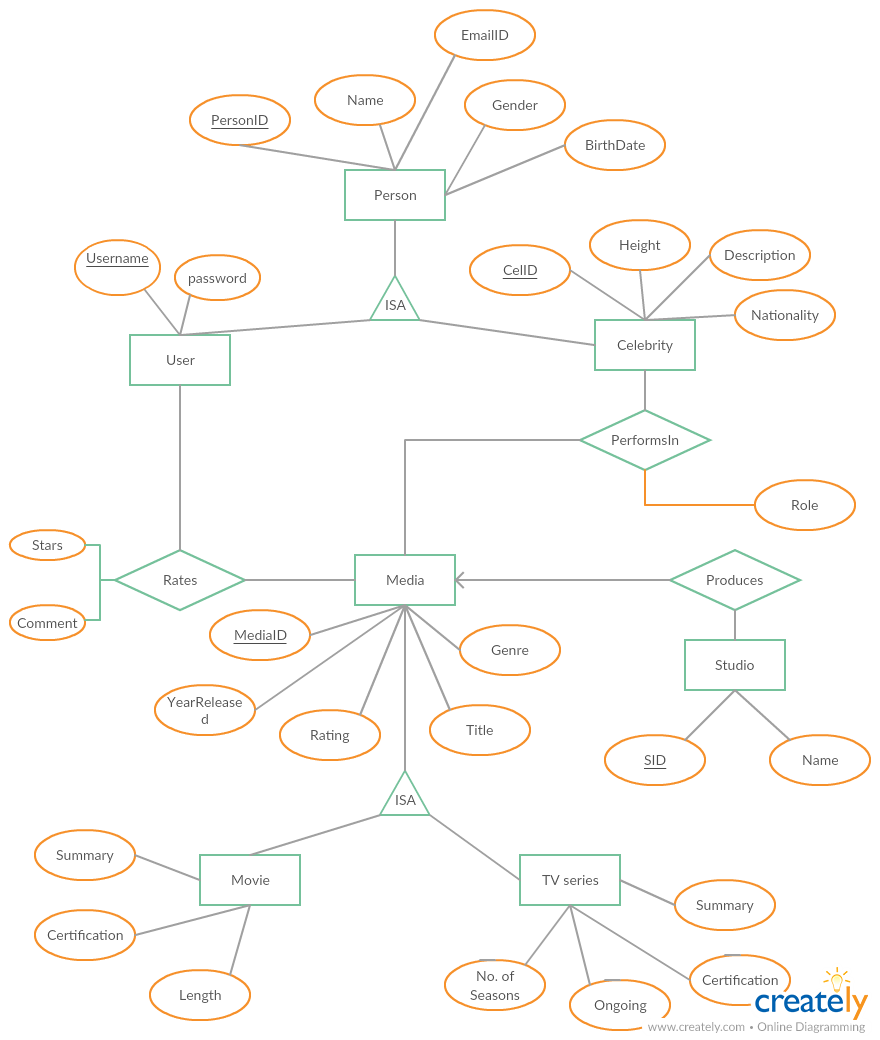
The main advantage of this system is that customer can buy products at low rate compared to the retailer shops. Customers can buy products sitting at their home easily.

Transactions are secured. Customers can do payment at delivery time and can replace or cancel their order in case any fault is present.

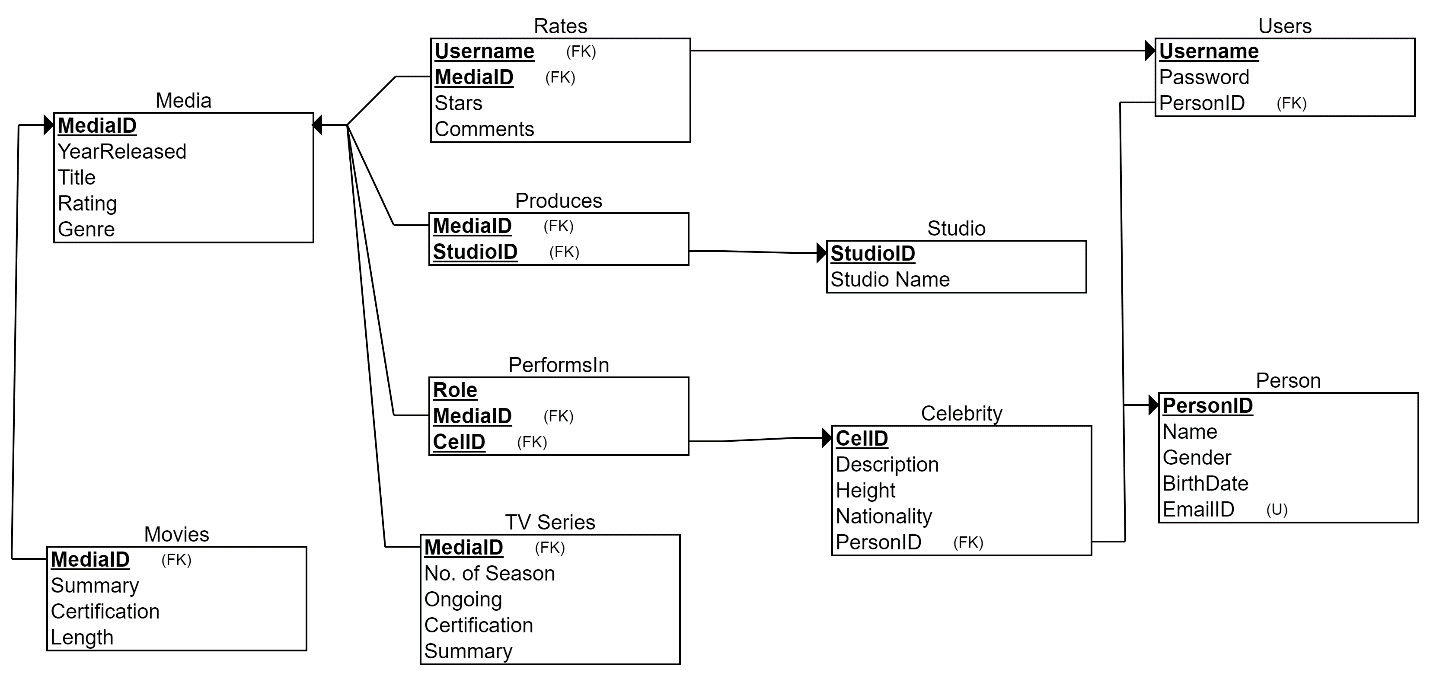
Customers can view all books of all categories and select appropriate one according to their price and offers present on them.

Thus, the system is very flexible for purchasing books of our choice online.

ER DIAGRAM



**SCHEMA DIAGRAM**



**IMPLEMENTATION**

**TABLE CREATION**

**CREATE STUDIO TABLE:-**

CREATE TABLE Studio

(

StudioID INT NOT NULL,

Studio\_Name VARCHAR(30) NOT NULL,

PRIMARY KEY (StudioID)

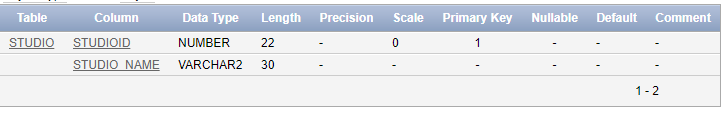
);

CREATE SEQUENCE Studio\_sequence

START WITH 1

INCREMENT BY 1;

DESC STUDIO;



**CREATE MEDIA TABLE:-**

CREATE TABLE Media

(

MediaID INT NOT NULL,

YearReleased INT NOT NULL,

Title VARCHAR(50) NOT NULL,

Rating INT NOT NULL,

Genre VARCHAR(20) NOT NULL,

PRIMARY KEY (MediaID)

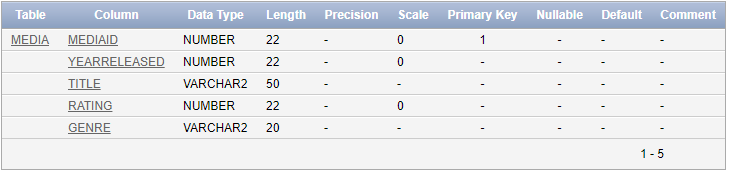
);

CREATE SEQUENCE Media\_sequence

START WITH 1

INCREMENT BY 1;

DESC MEDIA;



**CREATE PERSON TABLE:-**

CREATE TABLE Person

(

PersonID INT NOT NULL,

Name VARCHAR(20) NOT NULL,

Gender VARCHAR(1) NOT NULL,

BirthDate DATE NOT NULL,

EmailID VARCHAR(30) NOT NULL,

PRIMARY KEY (PersonID),

UNIQUE (EmailID)

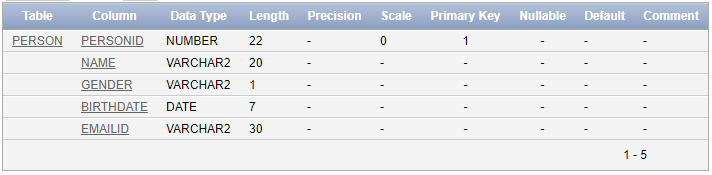
);

CREATE SEQUENCE Person\_sequence

START WITH 1

INCREMENT BY 1;

DESC PERSON;



**CREATE CELEBRITY TABLE:-**

CREATE TABLE Celebrity

(

CelID INT NOT NULL,

Description VARCHAR(300) NOT NULL,

Height INT NOT NULL,

Nationality VARCHAR(15) NOT NULL,

PersonID INT NOT NULL,

PRIMARY KEY (CelID),

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

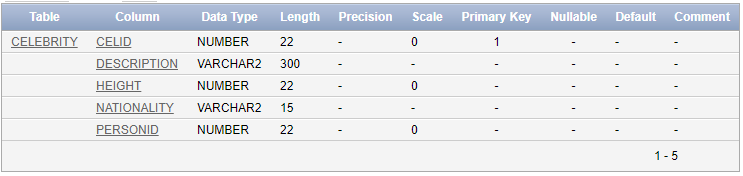
);

CREATE SEQUENCE Cel\_sequence

START WITH 1

INCREMENT BY 1;

DESC CELEBRITY;



**CREATE PERFORMSIN TABLE:-**

CREATE TABLE PerformsIn

(

Role VARCHAR(20) NOT NULL,

MediaID INT NOT NULL,

CelID INT NOT NULL,

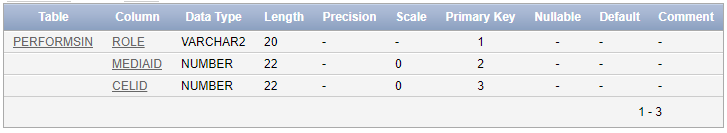
PRIMARY KEY (Role, MediaID, CelID),

FOREIGN KEY (MediaID) REFERENCES Media(MediaID),

FOREIGN KEY (CelID) REFERENCES Celebrity(CelID)

);

DESC PERFORMSIN;



**CREATE MOVIES TABLE:-**

CREATE TABLE Movies

(

Summary VARCHAR(300) NOT NULL,

Certification VARCHAR(3) NOT NULL,

Length INT NOT NULL,

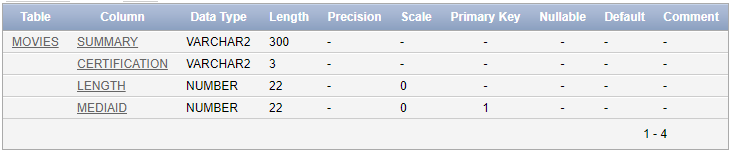
MediaID INT NOT NULL,

PRIMARY KEY (MediaID),

FOREIGN KEY (MediaID) REFERENCES Media(MediaID)

);

DESC MOVIES;



**CREATE TV\_SERIESTABLE:-**

CREATE TABLE TV\_Series

(

No\_of\_Season INT NOT NULL,

Ongoing INT NOT NULL,

Certification VARCHAR(2) NOT NULL,

Summary VARCHAR(300) NOT NULL,

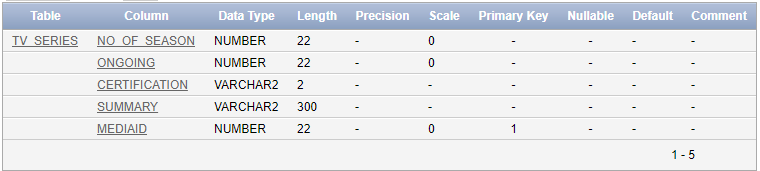
MediaID INT NOT NULL,

PRIMARY KEY (MediaID),

FOREIGN KEY (MediaID) REFERENCES Media(MediaID)

);

DESC TV\_SERIES;



**CREATE PRODUCES TABLE:-**

CREATE TABLE Produces

(

MediaID INT NOT NULL,

StudioID INT NOT NULL,

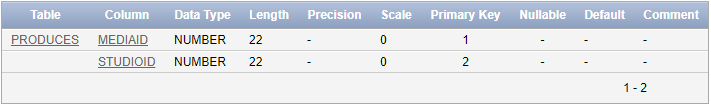
PRIMARY KEY (MediaID, StudioID),

FOREIGN KEY (MediaID) REFERENCES Media(MediaID),

FOREIGN KEY (StudioID) REFERENCES Studio(StudioID)

);

DESC PRODUCES;



**CREATE USERS TABLE:-**

CREATE TABLE Users

(

Username VARCHAR(20) NOT NULL,

Password VARCHAR(15) NOT NULL,

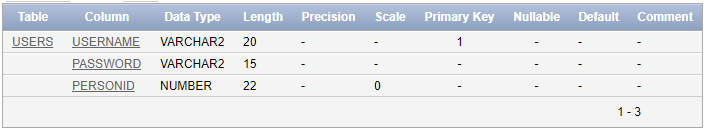
PersonID INT NOT NULL,

PRIMARY KEY (Username),

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

DESC USERS;



**CREATE RATES TABLE:-**

CREATE TABLE Rates

(

Stars INT NOT NULL,

Comments VARCHAR(100) NOT NULL,

Username VARCHAR(20) NOT NULL,

MediaID INT NOT NULL,

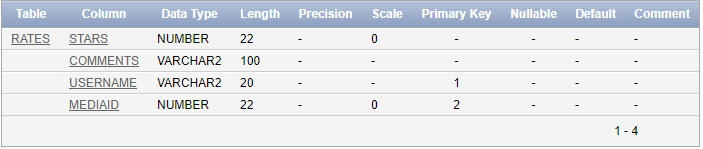
PRIMARY KEY (Username, MediaID),

FOREIGN KEY (Username) REFERENCES Users(Username),

FOREIGN KEY (MediaID) REFERENCES Media(MediaID)

);

DESC RATES;



**INSERTING VALUES**

**INSERT VALUES IN STUDIO TABLE:-**

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Paramount Pictures/20th Century Fox');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Warner Bros/DC comics');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'20th Century Fox');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'New line cinema');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval ,'Genre Films/Scott Free Productions');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Warner Bros');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'salman khan films');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Vinod Chopra Films; Rajkumar Hirani Films.');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Red Chillies Entertainment');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Friday Filmworks');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Film kraft');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Dark Horse Comics');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'AMC');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Warner Bros. Television');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Hartswood Films');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Bright/Kauffman/Crane Productions');

insert into Studio (StudioID,Studio\_Name) values (Studio\_sequence.nextval,'Netflix');



**INSERT VALUES IN CATEGORIES TABLE:-**

(1). insert into categories

(category\_id,rating,type)

values('C0001',4,'Inspiration');

(2). insert into categories

(category\_id,rating,type)

values('C0002',3,'Spiritual');

(3). insert into categories

(category\_id,rating,type)

values('C0003',3,'Autobiography');

(4). insert into categories

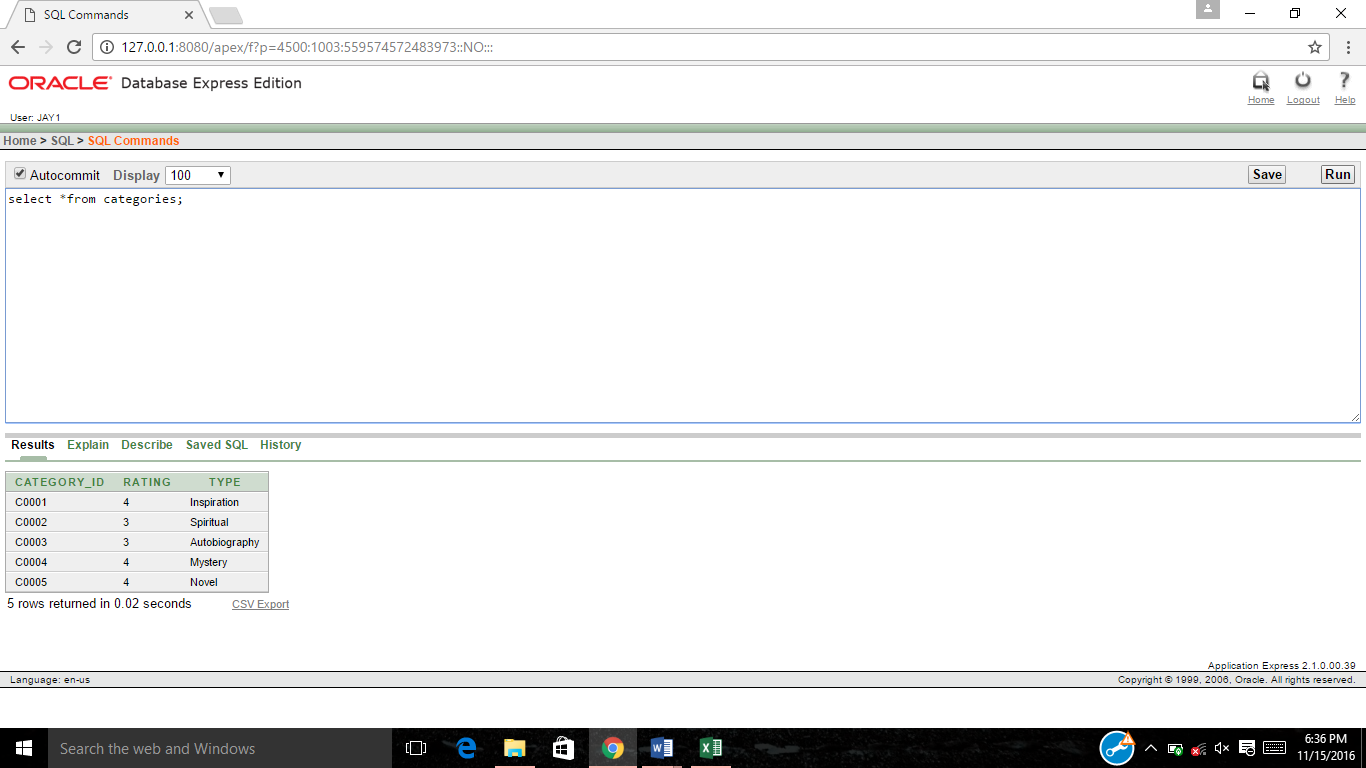
(category\_id,rating,type)

values('C0004',4,'Mystery');

(5). insert into categories

(category\_id,rating,type)

values('C0005',4,'Novel');



**INSERT VALUES IN WAREHOUSE TABLE:-**

(1). insert into warehouse

(w\_id,book\_id,quantity)

values(1,’B0001’,5);

(2). insert into warehouse

(w\_id,book\_id,quantity)

values(2,’B0002’,5);

(3). insert into warehouse

(w\_id,book\_id,quantity)

values(3,’B0003’,5);

(4). insert into warehouse

(w\_id,book\_id,quantity)

values(4,’B0004’,5);

(5). insert into warehouse

(w\_id,book\_id,quantity)

values(5,’B0005’,5);

(6). insert into warehouse

(w\_id,book\_id,quantity)

values(6,’B0006’,5);

(7). insert into warehouse

(w\_id,book\_id,quantity)

values(7,’B0007’,5);

(8). insert into warehouse

(w\_id,book\_id,quantity)

values(8,’B0008’,5);

(9). insert into warehouse

(w\_id,book\_id,quantity)

values(9,’B0009’,5);

(10). insert into warehouse

(w\_id,book\_id,quantity)

values(10,’B0010’,5);

(11). insert into warehouse

(w\_id,book\_id,quantity)

values(11,’B0011’,5);

(12). insert into warehouse

(w\_id,book\_id,quantity)

values(12,’B0012’,5);

(13). insert into warehouse

(w\_id,book\_id,quantity)

values(13,’B0013’,5);

(14). insert into warehouse

(w\_id,book\_id,quantity)

values(14,’B0014’,5);

(15). insert into warehouse

(w\_id,book\_id,quantity)

values(15,’B0015’,5);

(16). insert into warehouse

(w\_id,book\_id,quantity)

values(16,’B0016’,5);

(17). insert into warehouse

(w\_id,book\_id,quantity)

values(17,’B0017’,5);

(18). insert into warehouse

(w\_id,book\_id,quantity)

values(18,’B0018’,5);

(19). insert into warehouse

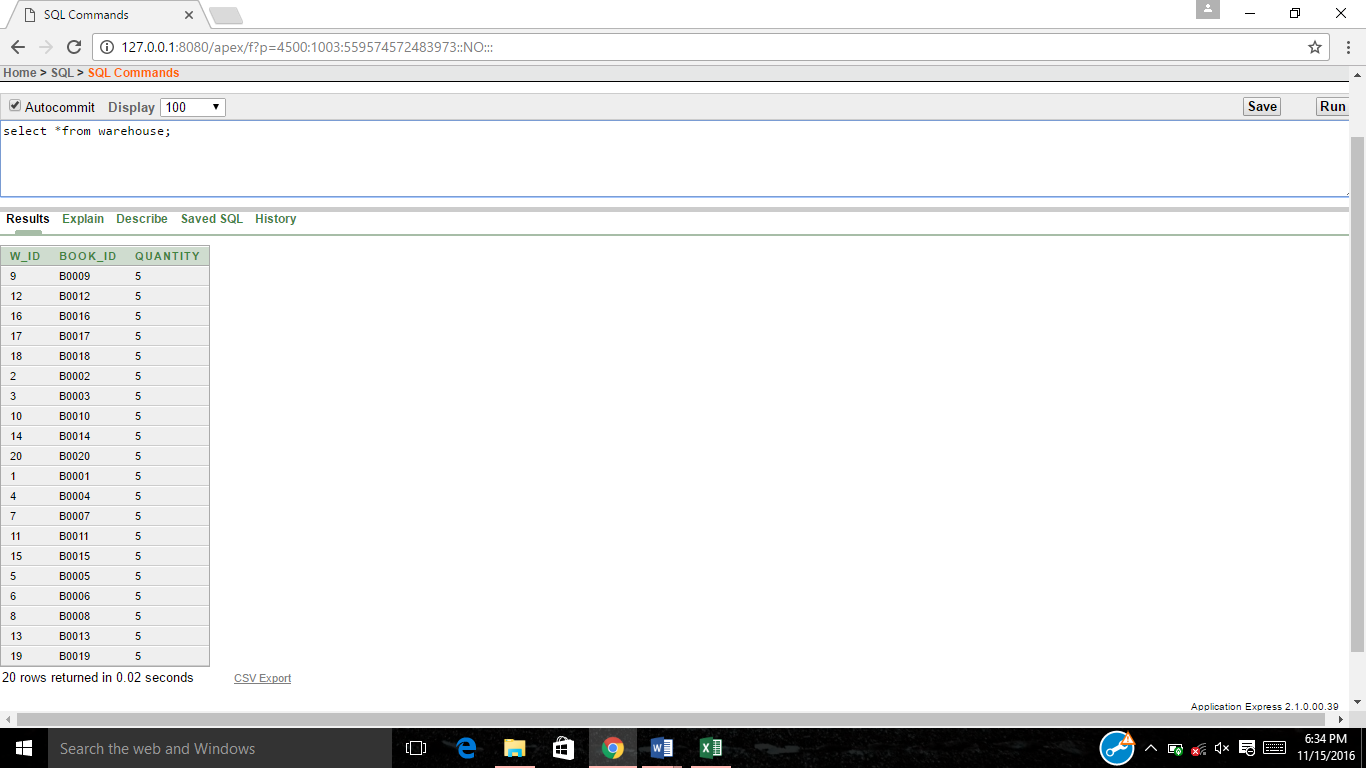
(w\_id,book\_id,quantity)

values(19,’B0019’,5);

(20). insert into warehouse

(w\_id,book\_id,quantity)

values(20,’B0020’,5);



**INSERT VALUES IN FEEDBACK TABLE:-**

(1). insert into feedback

(book\_id,cid,rating,f\_id,comments)

values(‘B0001’,’A0001’,5,’F001’,’AWESOME’);

(2). insert into feedback

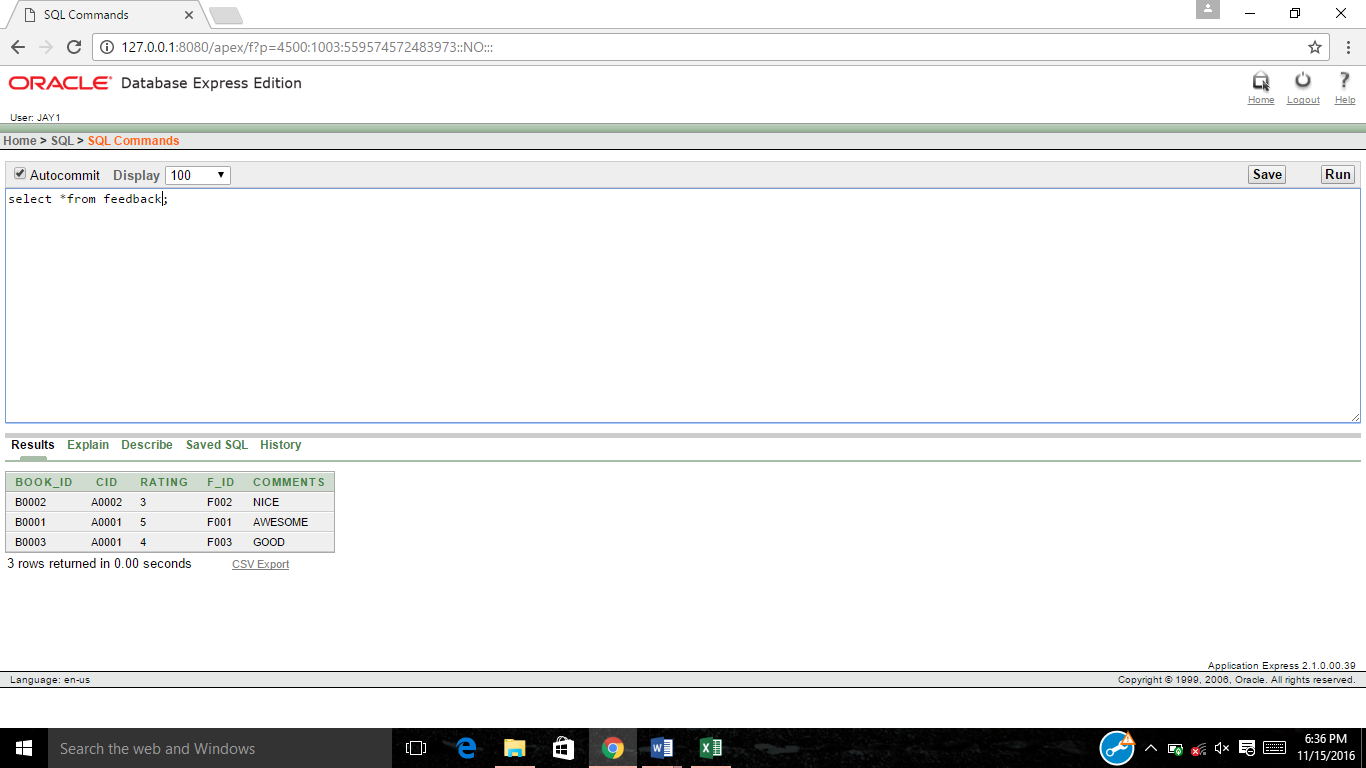
(book\_id,cid,rating,f\_id,comments)

values(‘B0002’,’A0002’,3,’F002’,’NICE);

(3). insert into feedback

(book\_id,cid,rating,f\_id,comments)

values(‘B0003’,’A0001’,4,’F003’,’GOOD’)



**INSERT VALUES IN OFFERS TABLE:-**

(1). insert into offers

(of\_id, book\_id,discount,valid)

values(1, 'B0001', 15,’09-NOV-2016’);

(2). insert into offers

(of\_id, book\_id,discount,valid)

values(2, 'B0002', 10,’12-DEC-2016’);

(3). insert into offers

(of\_id, book\_id,discount,valid)

values(3, 'B0003', 14,’08-NOV-2016’);

(4). insert into offers

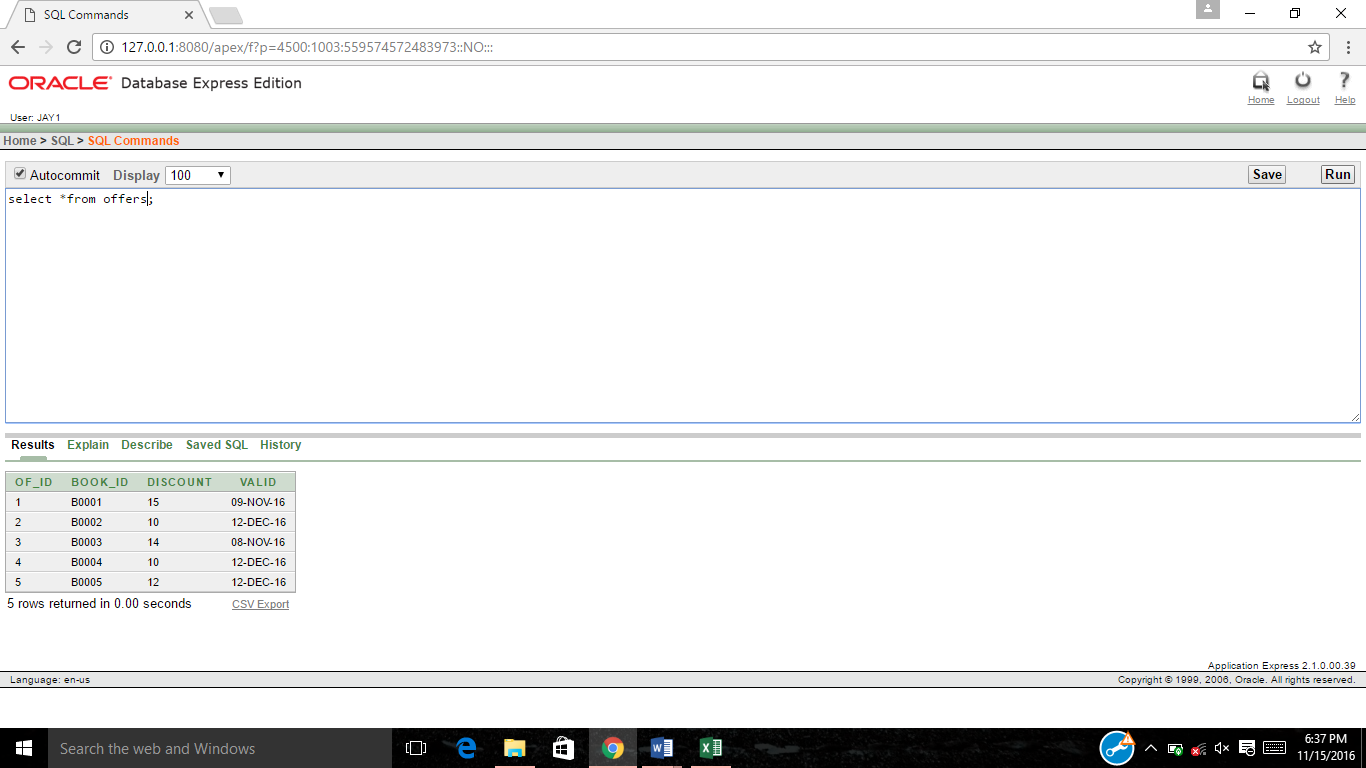
(of\_id, book\_id,discount,valid)

values(4, 'B0004', 10,’12-DEC-2016’);

(5). insert into offers

(of\_id, book\_id,discount,valid)

values(5, 'B0005', 12,’12-DEC-2016’);



**INSERT VALUES IN AUTHOR TABLE:-**

(1). insert into author

(author\_id,name,bookname)

values('D0001',’Abhinav’,’Star of life’);

(2). insert into author

(author\_id,name,bookname)

values('D0002',’Abhilash’,’ Mistry of little girl’);

(3). insert into author

(author\_id,name,bookname)

values('D0003',’Kalpan’,’ Polyana’);

(4). insert into author

(author\_id,name,bookname)

values('D0004',’Smit’,’ All about attitude’);

(5). insert into author

(author\_id,name,bookname)

values('D0005',’Parth’,’ A Day’);

(6). insert into author

(author\_id,name,bookname)

values('D0006',’Bob’,’ 9 clouds’);

(7). insert into author

(author\_id,name,bookname)

values('D0007',’Jasmin’,’ life of pie’);

(8). insert into author

(author\_id,name,bookname)

values('D0008',’Jennifer’,’ New york’);

(9). insert into author

(author\_id,name,bookname)

values('D0009',’Robert’,’ Harry Potter’);

(10). insert into author

(author\_id,name,bookname)

values('D0010',’Atul’,’ Think and grow rich’);

(11). insert into author

(author\_id,name,bookname)

values('D0011',’Gopal’,’Abdul kalam’);

(12). insert into author

(author\_id,name,bookname)

values('D0012',’Sanjay’,’ Poor dad Rich dad’);

(13). insert into author

(author\_id,name,bookname)

values('D0013',’Sanjiv’,’ The secret of shadow’);

(14). insert into author

(author\_id,name,bookname)

values('D0014',’Vats’,’Solar system’);

(15). insert into author

(author\_id,name,bookname)

values('D0015',’Chahna’,’All about business’);

(16). insert into author

(author\_id,name,bookname)

values('D0016',’Nidhi’,’The life after death’);

(17). insert into author

(author\_id,name,bookname)

values('D0017',’Jingal’,’National tressure’);

(18). insert into author

(author\_id,name,bookname)

values('D0018',’Govind’,’Indiana johnes’);

(19). insert into author

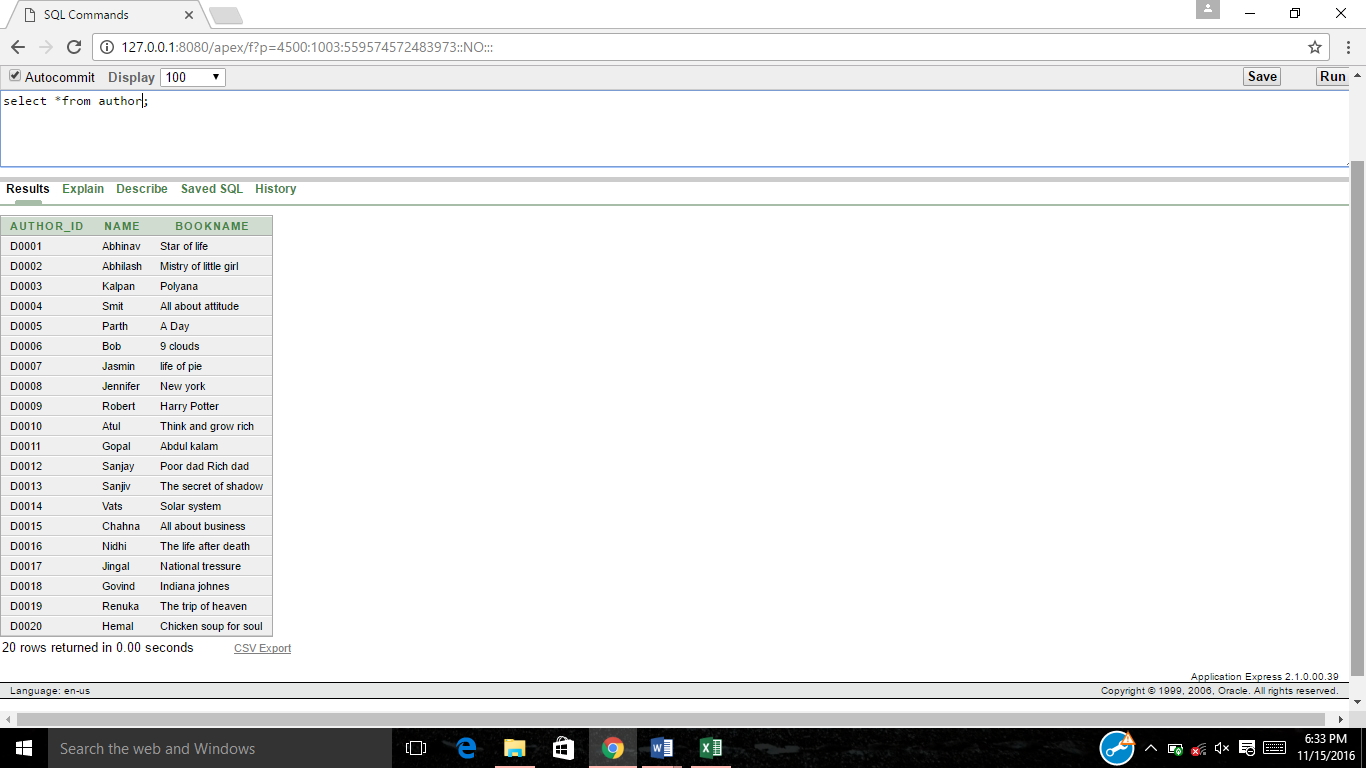
(author\_id,name,bookname)

values('D0019',’Renuka’,’The trip of heaven’);

(20). insert into author

(author\_id,name,bookname)

values('D0020',’Hemal’,’Chicken soup for soul’);



**PL/SQL IMPLEMENTATION:-**

**(1). PROCEDURE FOR PLACING THE ORDER:-**

set serveroutput on;

declare

product\_id varchar2(7);

qty1 number(7);

qty11 number(7);

customer\_id varchar2(7);

amt number(8);

delcharges number(7);

dis number(7);

temp number(7);

temp1 number(7);

cartno VARCHAR2(7);

wareid VARCHAR2(7);

cartid varchar2(7);

payid number(7);

orderid varchar2(7);

outorder exception;

begin

customer\_id:=:customer\_id;

product\_id:=:product\_id;

qty1:=:qty1;

orderid:=:orderid;

cartid:=:cartid;

payid:=:payid;

insert into cart

(cart\_id,book\_id, cid, quantity)

values(cartid,product\_id,customer\_id,qty1);

select cart\_id into cartno from cart where book\_id=product\_id and cid=customer\_id;

select price into amt from books where book\_id=product\_id;

select w\_id into wareid from warehouse where book\_id=product\_id;

amt:=amt\*qty1 ;

select discount into dis from offers where book\_id=product\_id;

temp:=(dis/100)\*amt;

temp1:=(amt-temp);

delcharges:=50;

temp1:=temp1+50;

insert into payments

(pay\_id,cart\_id,amount,deleivery\_charge,total\_amount)

values(payid,cartno, amt,delcharges,temp1);

select quantity into qty11 from warehouse where book\_id=product\_id;

if qty1>qty11 then raise outorder;

end if;

insert into orders

(oid,cid, book\_id, o\_date,quantity,o\_amt)

values(orderid,customer\_id,product\_id,SYSDATE,qty1,amt);

update warehouse set quantity=quantity-qty1 where book\_id=product\_id;

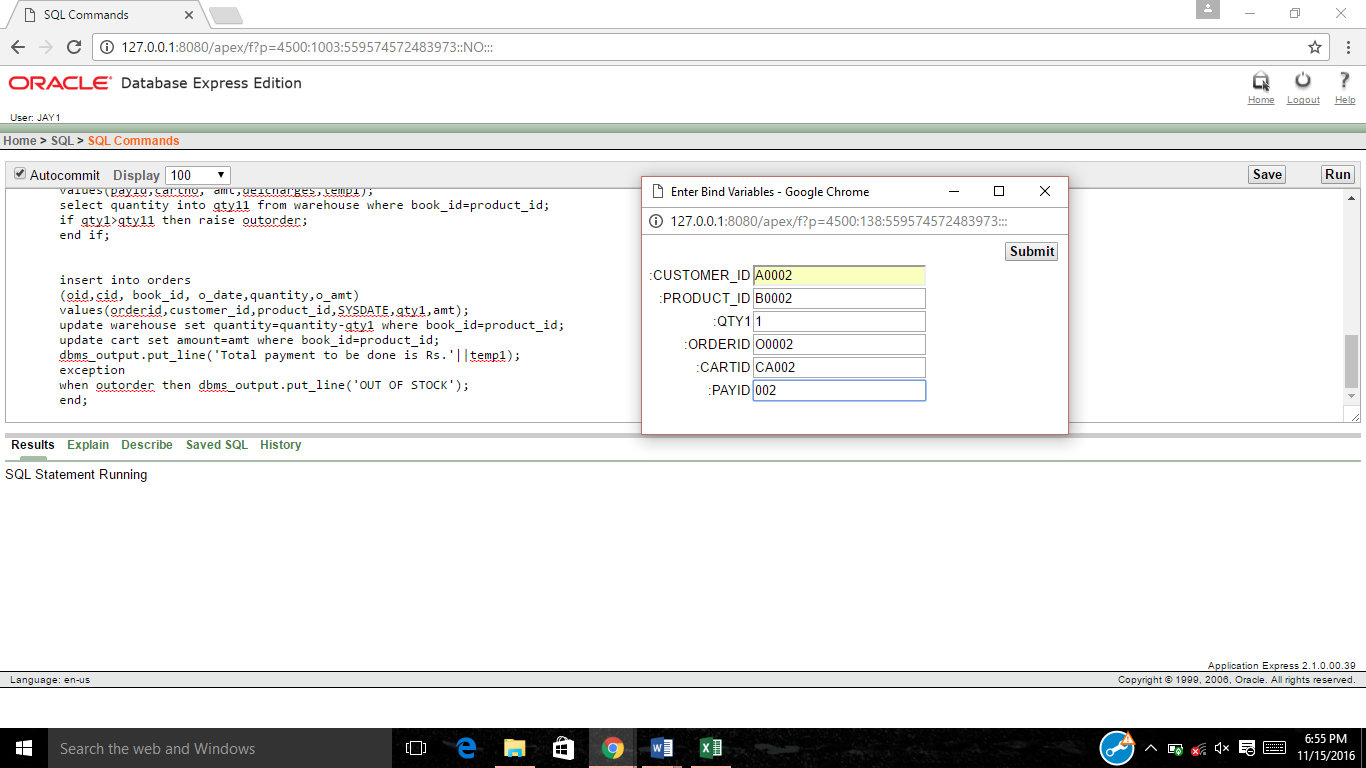
update cart set amount=amt where book\_id=product\_id;

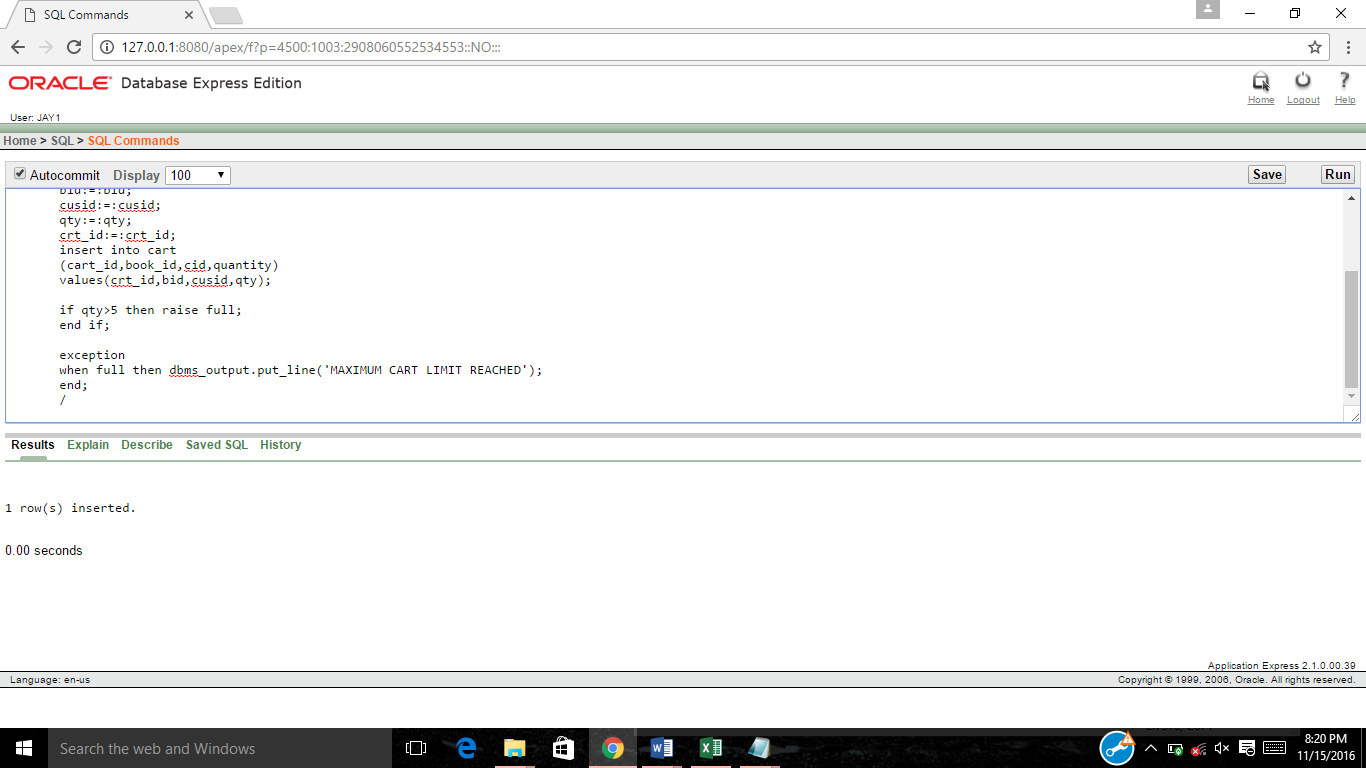
dbms\_output.put\_line('Total payment to be done is Rs.'||temp1);

exception

when outorder then dbms\_output.put\_line('OUT OF STOCK');

end;





**(2). UPDATE WAREHOUSE TRIGGER:-**

create or replace trigger wareupdate

after insert

on orders

for each row

declare

qt number(5);

qt11 number(5);

begin

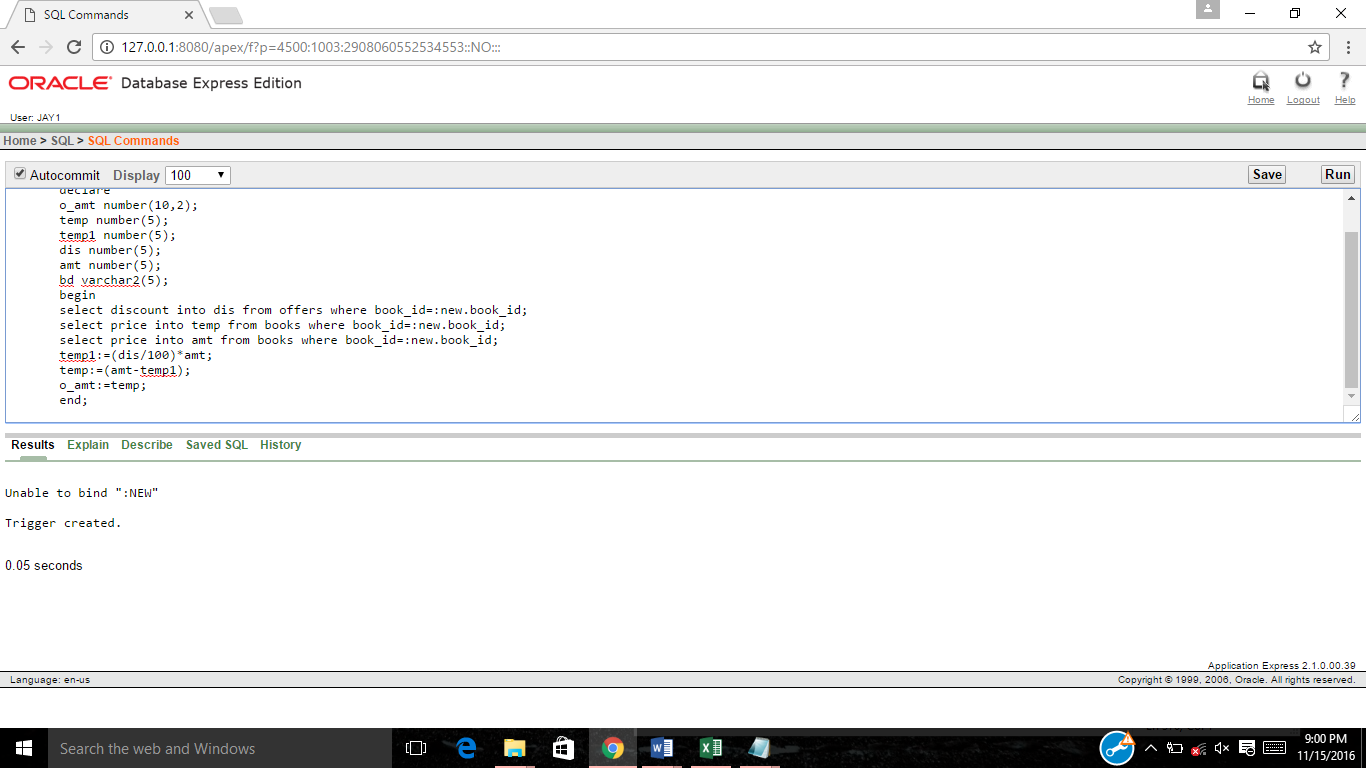
select quantity into qt from cart where book\_id=:new.book\_id;

select quantity into qt11 from warehouse where book\_id=:new.book\_id;

update warehouse set quantity=qt11-qt where book\_id=:new.book\_id;

end;

/



**(3). EXCEPTION FOR OUT OF STOCK:-**

set serveroutput on;

declare

full exception;

crt\_id varchar2(5);

bid varchar2(5);

cusid varchar2(5);

qty number(5);

qty1 number(5);

begin

bid:=:bid;

cusid:=:cusid;

qty:=:qty;

crt\_id:=:crt\_id;

select quantity into qty1 from warehouse where book\_id=bid;

insert into cart

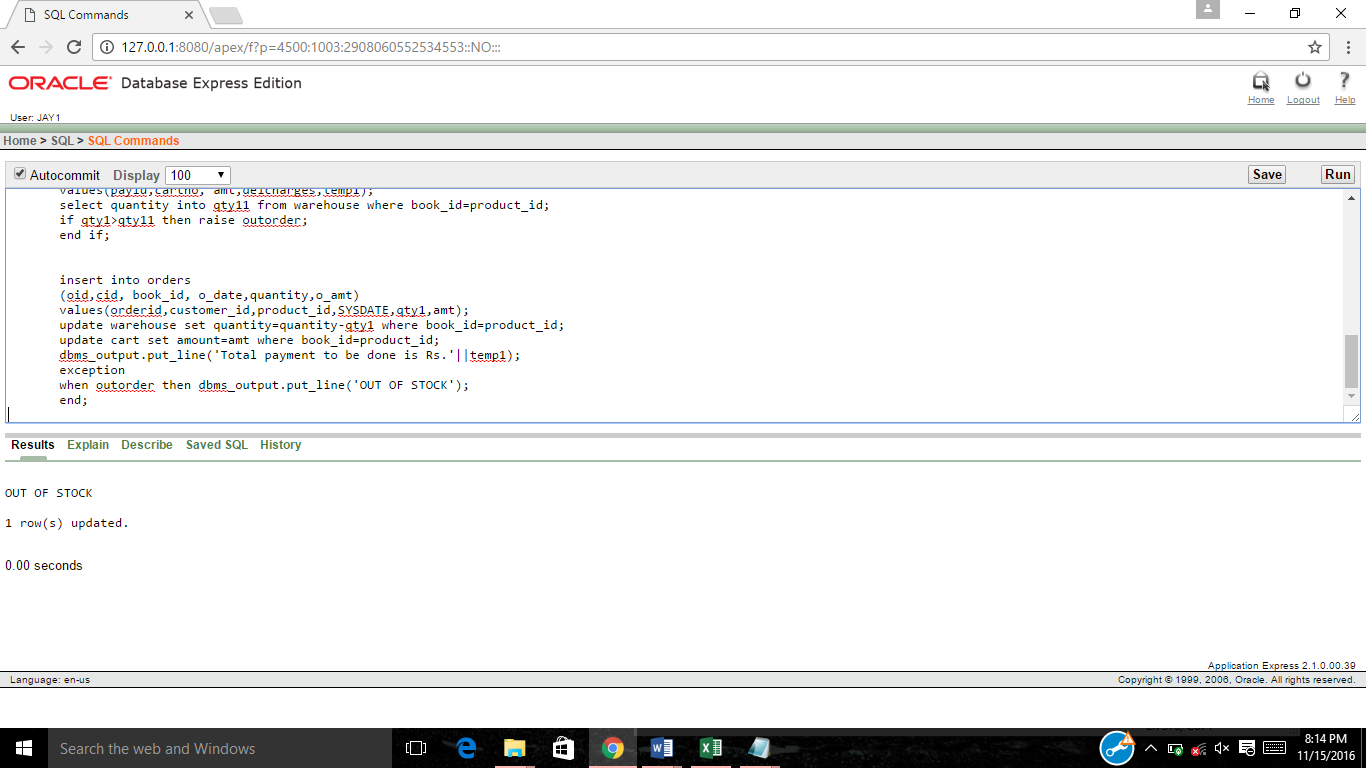
(cart\_id,book\_id,cid,quantity)

values(crt\_id,bid,cusid,qty);

if qty>qty1 then raise full;

end if;

exception when full then dbms\_output.put\_line('OUT OF STOCK');end;



**(4). FILL WAREHOUSE:-**

Declare

bd varchar2(5);

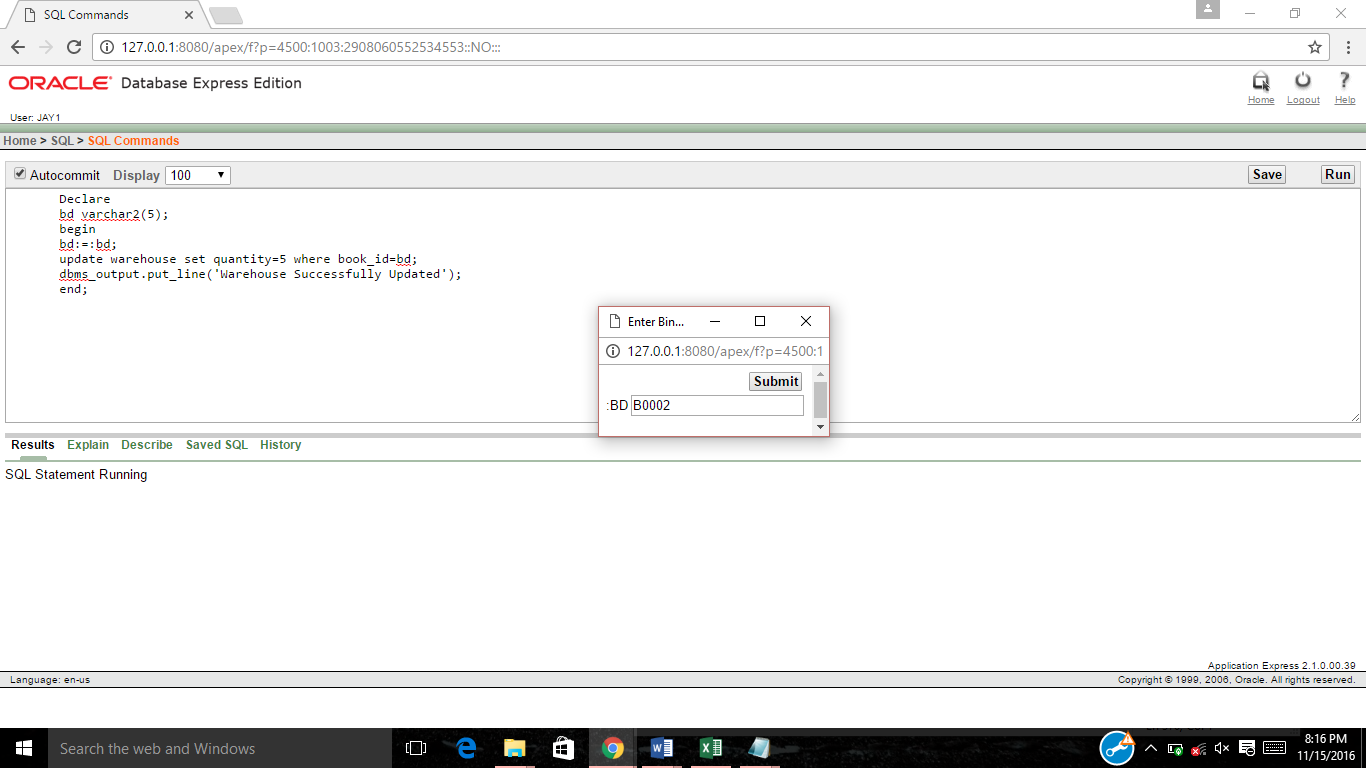
begin

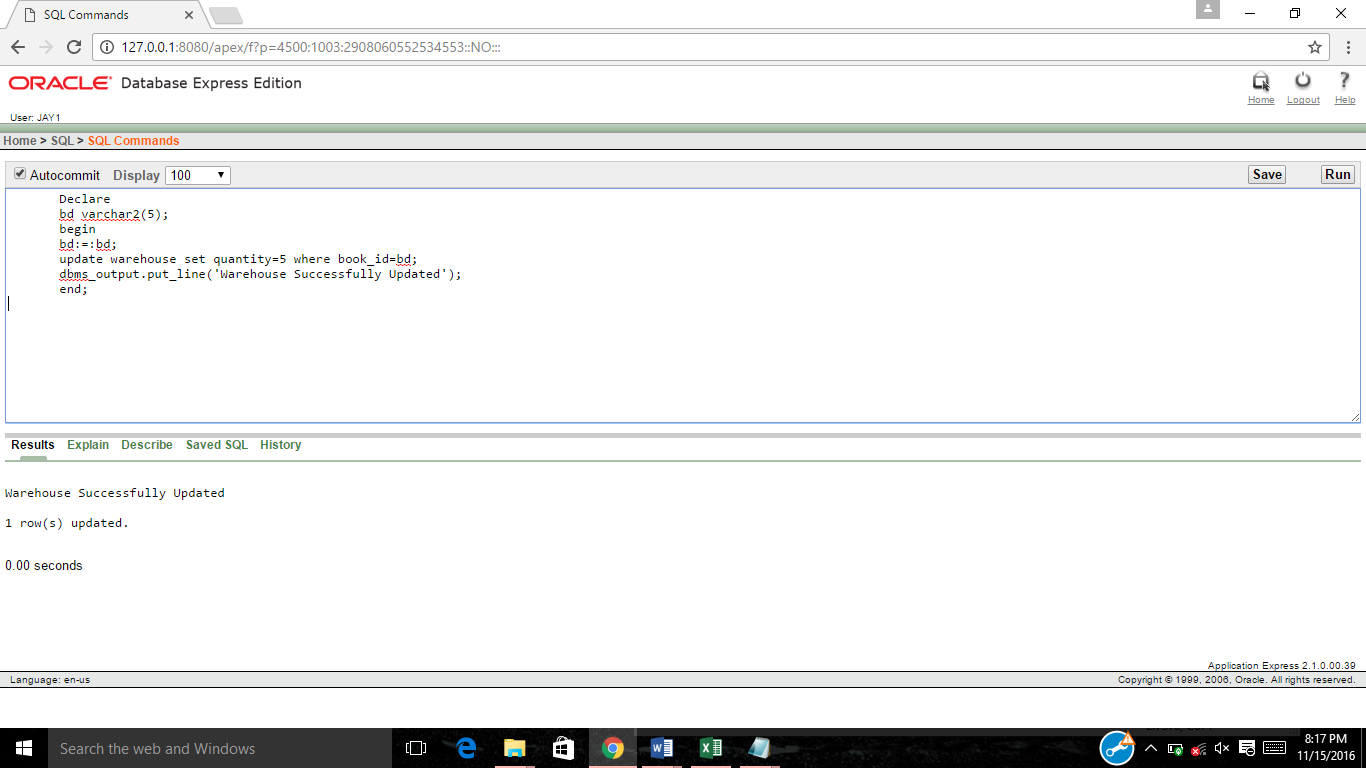
bd:=:bd;

update warehouse set quantity=5 where book\_id=bd;

dbms\_output.put\_line('Warehouse Successfully Updated');

end;





**(5). TRIGGER FOR SUBTRACTING DISCOUNT AMOUNT:-**

create or replace trigger discalc

before insert on orders for each row

declare

temp number(5);

temp1 number(5);

dis number(5);

amt number(5);

bd varchar2(5);

begin

select discount into dis from offers where book\_id=:new.book\_id;

select price into temp from books where book\_id=:new.book\_id;

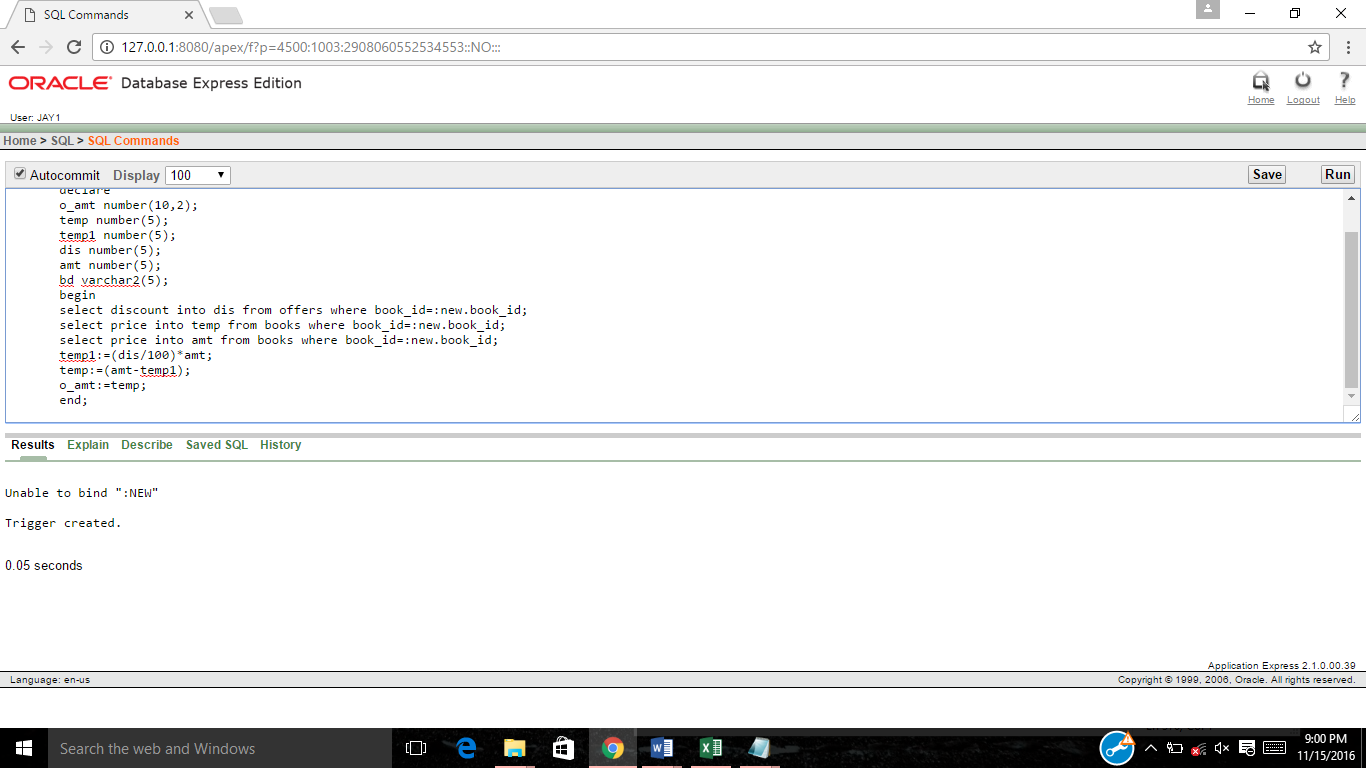
select price into amt from books where book\_id=:new.book\_id;

temp1:=(dis/100)\*amt;

temp:=(amt-temp1);

new.o\_amt:=temp;

end;



**(6). EXCEPTION FOR MAXCART LIMIT:-**

declare

full exception;

crt\_id varchar2(5);

bid varchar2(5);

cusid varchar2(5);

qty number(5);

begin

bid:=:bid;

cusid:=:cusid;

qty:=:qty;

crt\_id:=:crt\_id;

insert into cart

(cart\_id,book\_id,cid,quantity)

values(crt\_id,bid,cusid,qty);

if qty>5 then raise full;

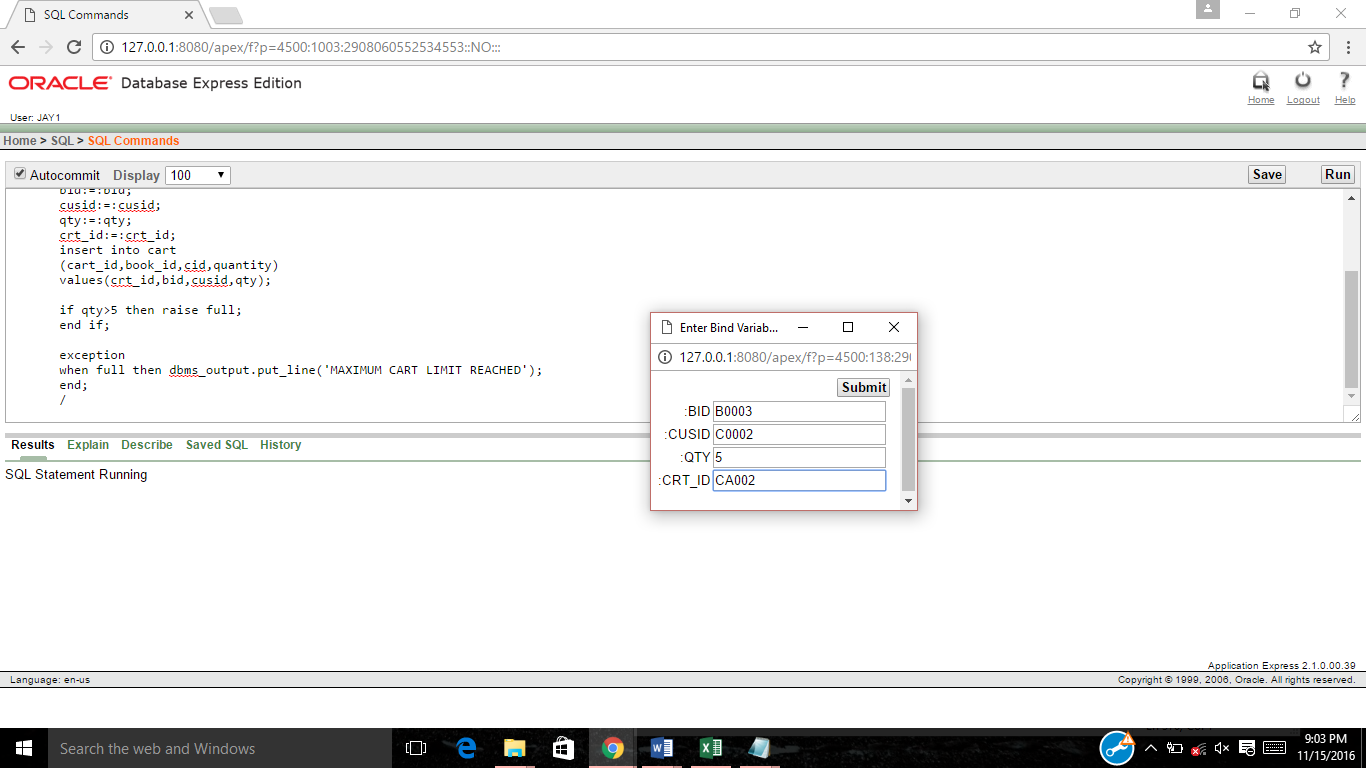
end if;

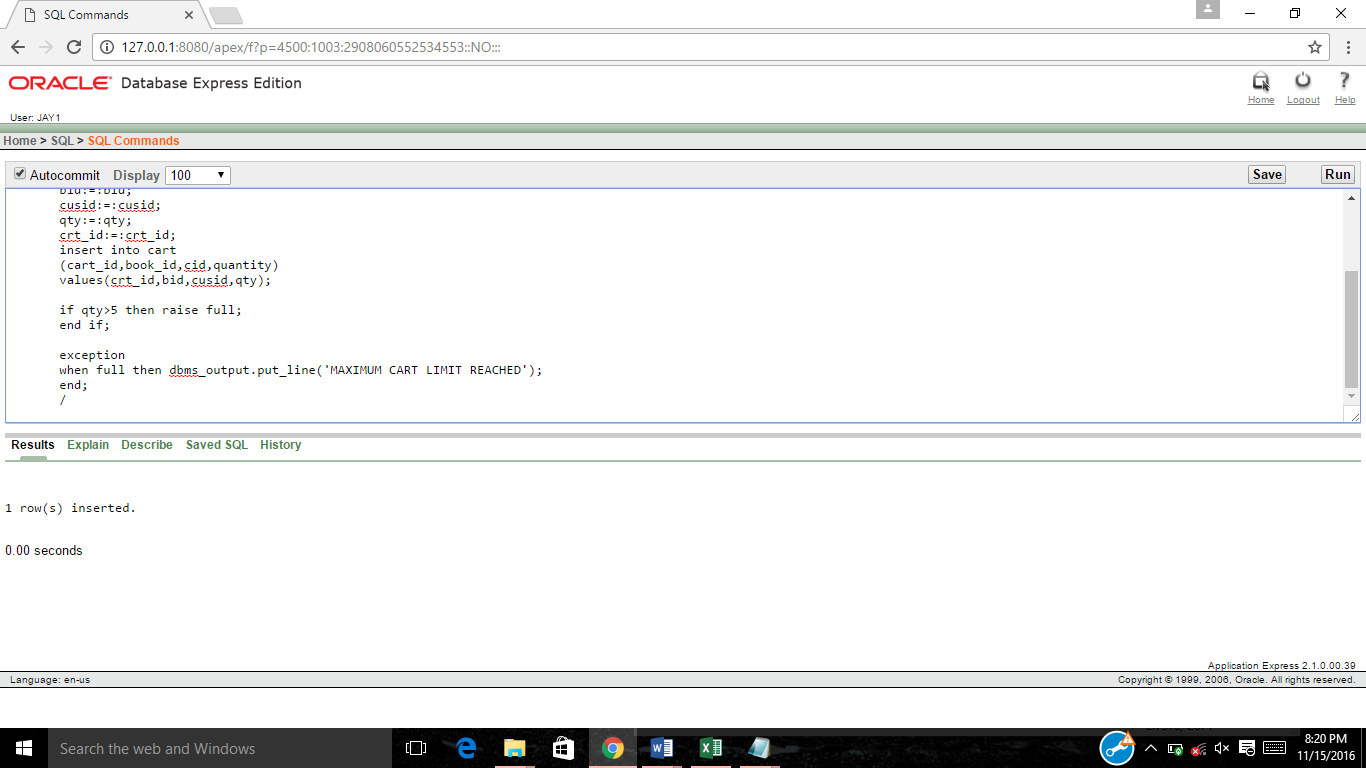
exception

when full then dbms\_output.put\_line('MAXIMUM CART LIMIT REACHED');

end;

/





**(7). TO INCREMENT PRIMARY KEY OF PAYMENT TABLE:-**

CREATE SEQUENCE pay\_seq

START WITH 1

INCREMENT BY 1;

create or replace trigger payidgen

before insert on payments for each row

declare

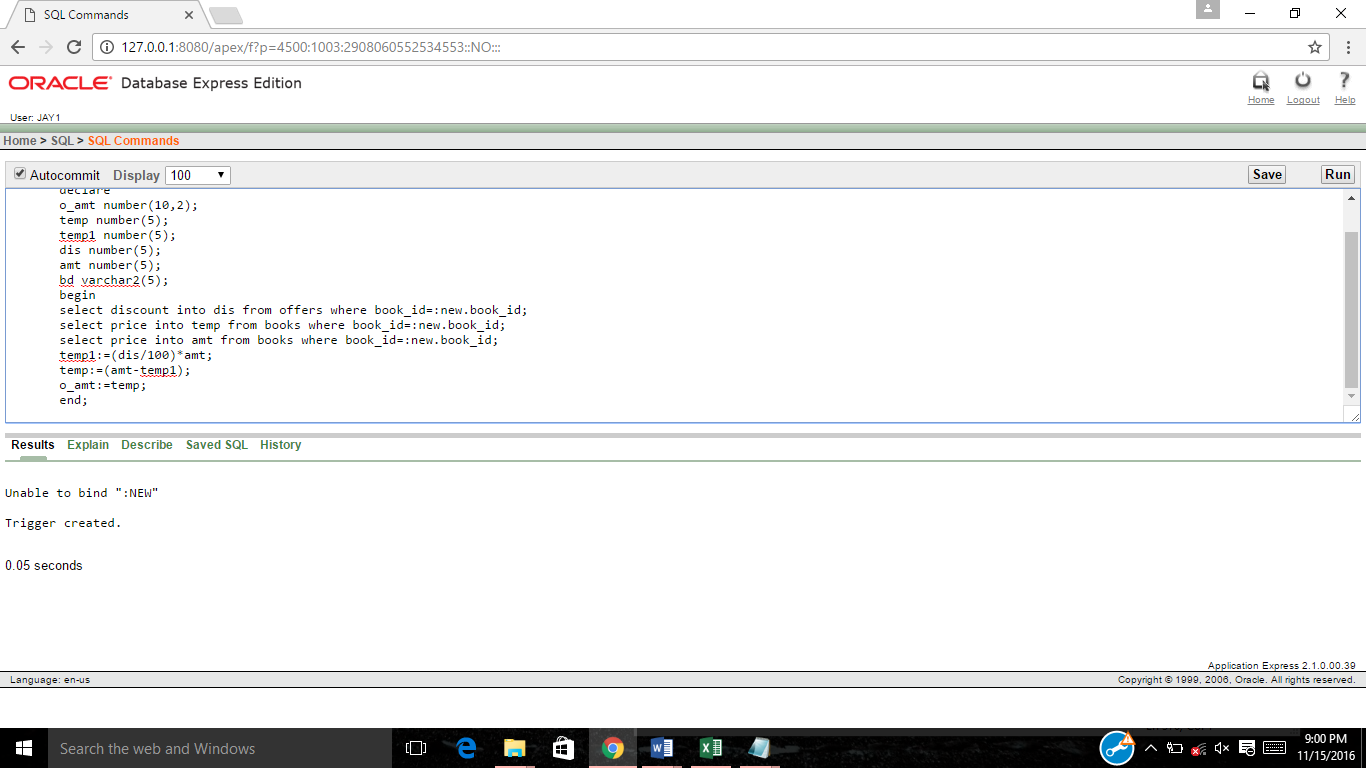
temp number(4);

begin

select pay\_seq.nextval into temp from dual;

:new.pay\_id:=temp;

end;



**(8). TO INCREMENT PRIMARY KEY OF ORDERS TABLE:-**

CREATE SEQUENCE or\_seq

START WITH 1

INCREMENT BY 1;

create or replace trigger oidgen

before insert on orders for each row

declare

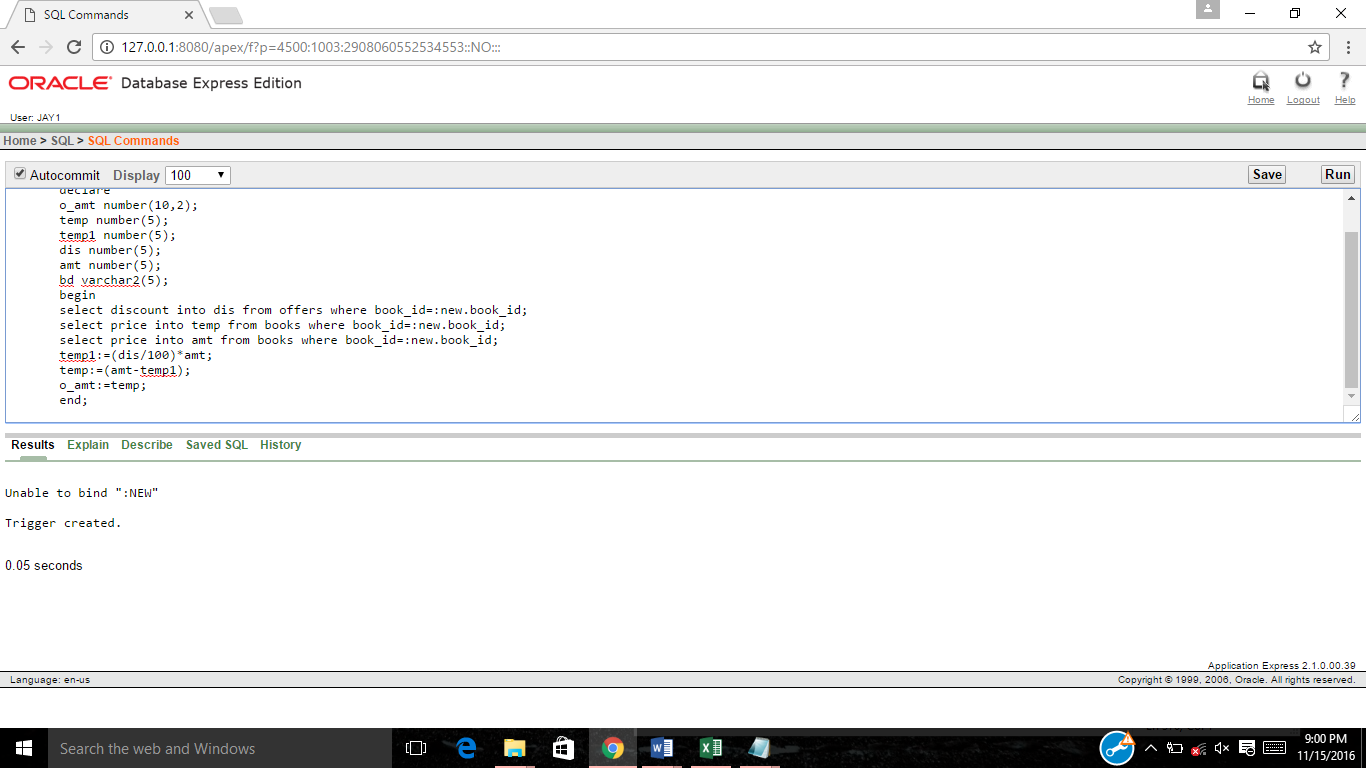
temp number(4);

begin

select or\_seq.nextval into temp from dual;

:new.oid:=temp;

end;



BIBLIOGRAPHY

Reference books:

· Data Base System Concepts

- Henry F.Korth and A.Silberschatz. 2nd Ed. McGraw-Hill 1991.

· SQL, PL/SQL The programming language of Oracle

- Ivan Bayross, BPB Publications

Reference Link:

· <http://en.wikipedia.org/wiki/Database>

· [www.w3schools.com/sql](http://www.w3schools.com/sql)