Charotar University of Science and Technology

Devang Patel Institute of Advance Technology and Research

Department of Computer Engineering

Student ID	**:**	**18DCE115**	**Student Name**	**:**	**KASHYAP SHAH**
Subject Code	**:**	**CE246**	**Subject Name**	**:**	**DATABASE MANAGEMENT SYSTEM**
Date of exam	**:**	**29-05-2020**	**Semester**	**:**	**4th Semester**

Definition:

There is a warehouse of AMart. Create a table to manage the items available in AMart warehouse
Create a table: Warehouse_data

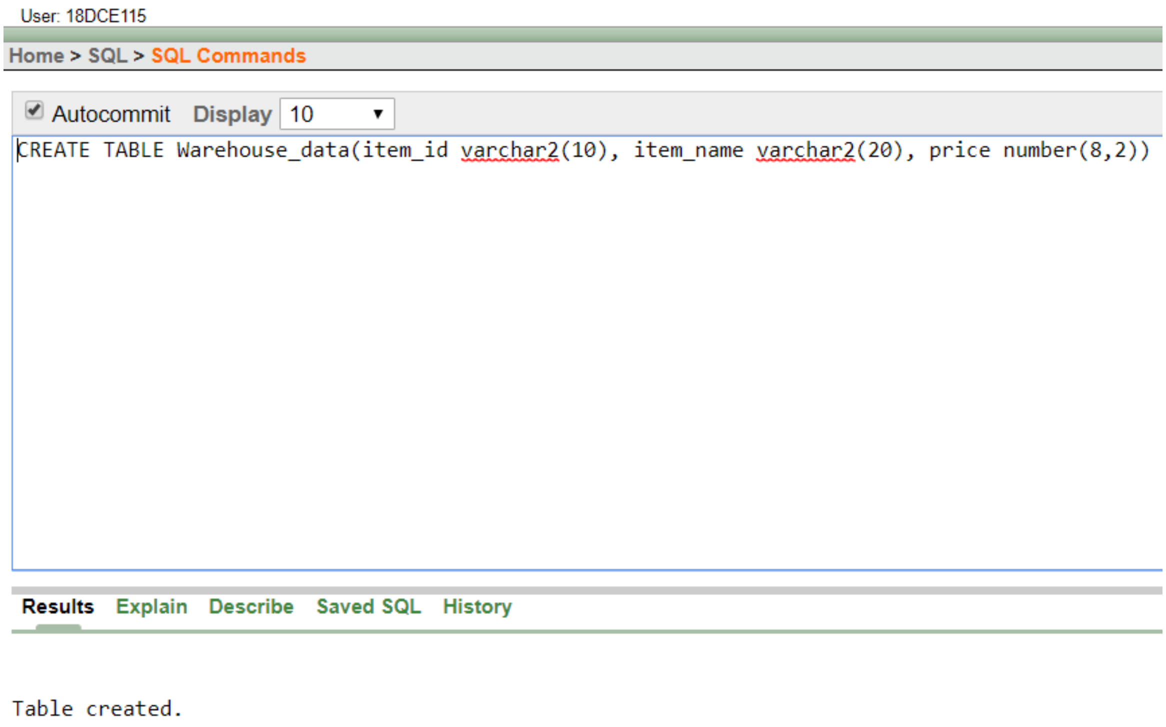
Table: warehouse_data
Field Name Data type
Item_id Varchar2(10)
Item_name Varchar2(20)
Price Number(8,2)

Insert minimum 5 records.
Create a field called Quantity. Take the data of quantity from user and store it into the table.
When a new item imported, check whether the item is available in the given table or not. If it's
available, then update the quantity and if the item is not available then insert a new record into
warehouse_data table.

At last, create a function to display the total price.

Solution (code & screenshot):

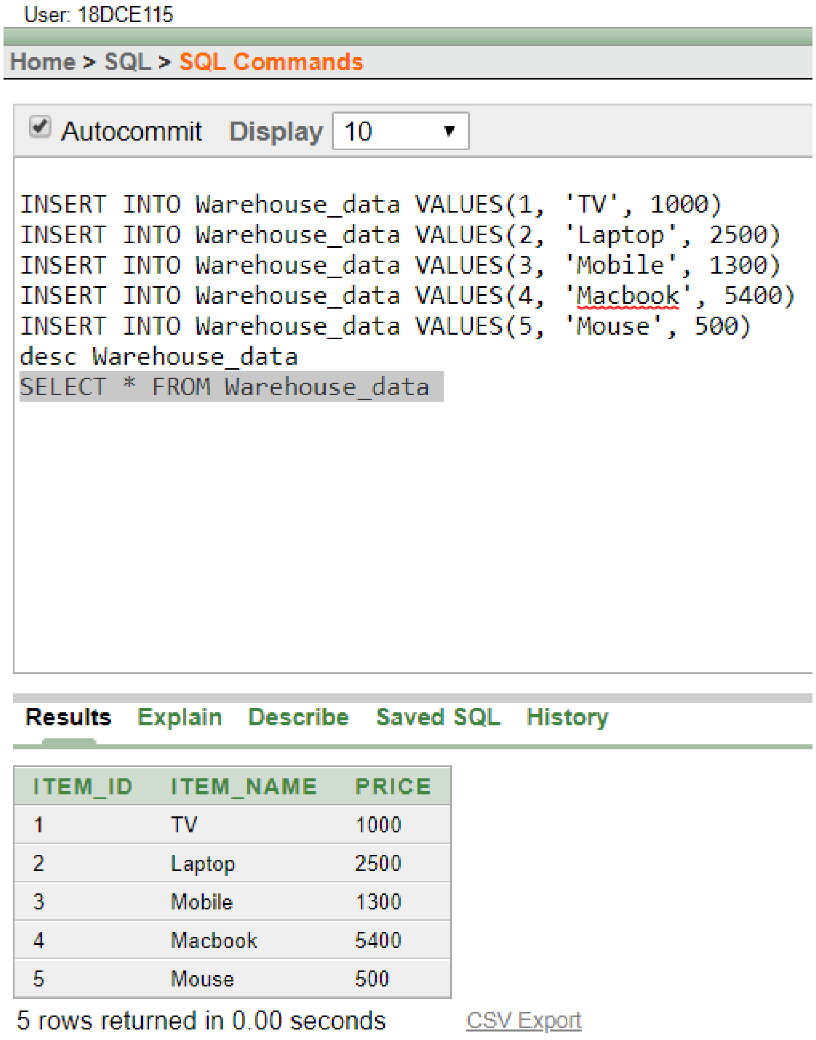
```
CREATE TABLE Warehouse_data(item_id varchar2(10), item_name varchar2(20), price
number(8,2))
```



```
CREATE TABLE Warehouse_data(item_id varchar2(10), item_name varchar2(20), price number(8,2))
```

```
Table created.
```

```
INSERT INTO Warehouse_data VALUES(1, 'TV', 1000)
INSERT INTO Warehouse_data VALUES(2, 'Laptop', 2500)
INSERT INTO Warehouse_data VALUES(3, 'Mobile', 1300)
INSERT INTO Warehouse_data VALUES(4, 'Macbook', 5400)
INSERT INTO Warehouse_data VALUES(5, 'Mouse', 500)
desc Warehouse_data
SELECT * FROM Warehouse_data
```



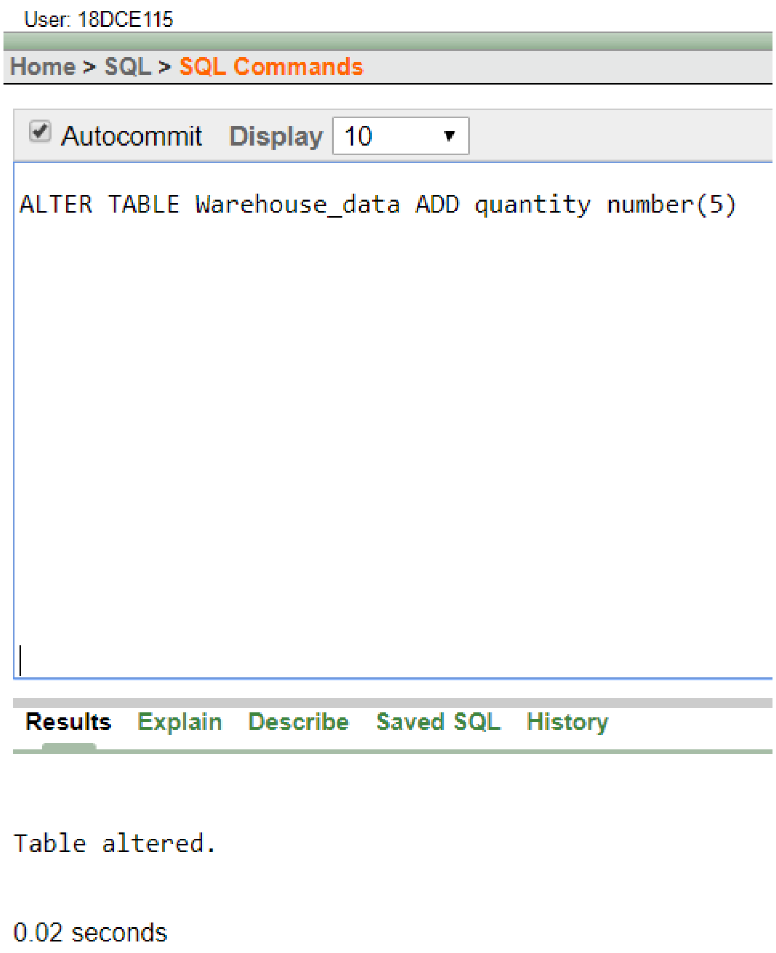
☑ Autocommit Display 10

```
INSERT INTO Warehouse_data VALUES(1, 'TV', 1000)
INSERT INTO Warehouse_data VALUES(2, 'Laptop', 2500)
INSERT INTO Warehouse_data VALUES(3, 'Mobile', 1300)
INSERT INTO Warehouse_data VALUES(4, 'Macbook', 5400)
INSERT INTO Warehouse_data VALUES(5, 'Mouse', 500)
desc Warehouse_data
SELECT * FROM Warehouse_data
```

ITEM_ID	ITEM_NAME	PRICE
1	TV	1000
2	Laptop	2500
3	Mobile	1300
4	Macbook	5400
5	Mouse	500

5 rows returned in 0.00 seconds

ALTER TABLE Warehouse_data ADD quantity number(5)



```
ALTER TABLE Warehouse_data ADD quantity number(5)
```

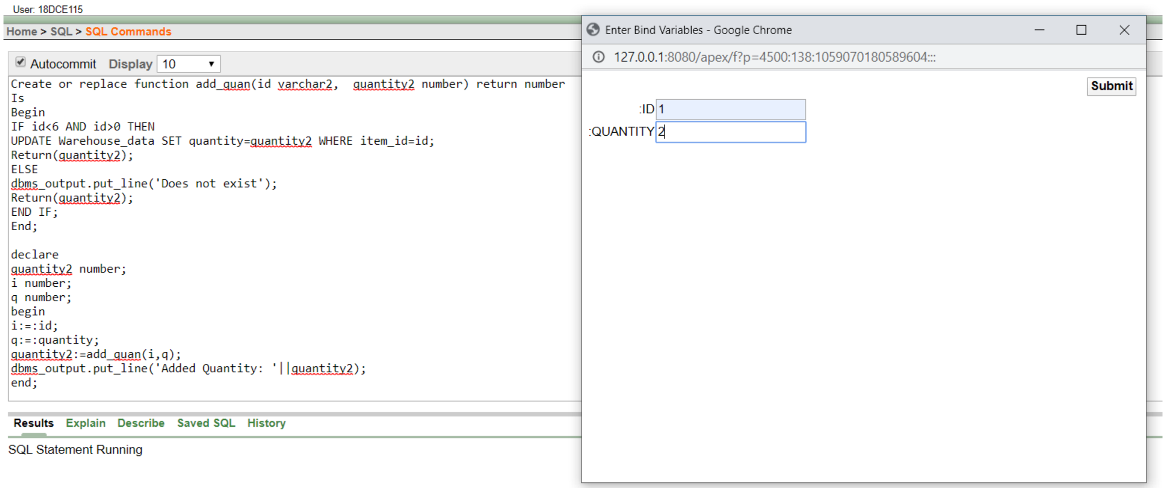
```
Table altered.
```

0.02 seconds

```
Create or replace function add_quan(id varchar2,  quantity2 number) return number
Is
Begin
IF id<6 AND id>0 THEN
UPDATE Warehouse_data SET quantity=quantity2 WHERE item_id=id;
Return(quantity2);
ELSE
dbms_output.put_line('Does not exist');
Return(quantity2);
END IF;
End;

declare
quantity2 number;
i number;
q number;
```

```
begin
i:=:id;
q:=:quantity;
quantity2:=add_quan(i,q);
dbms_output.put_line('Added Quantity: '||quantity2);
end;
```



User: 18DCE115
Home > SQL > SQL Commands
☑ Autocommit Display 10 ▾
Results Explain Describe Saved SQL History
Enter Bind Variables - Google Chrome
Submit

```
Create or replace function add_quan(id varchar2,  quantity2 number) return number
Is
Begin
IF id<6 AND id>0 THEN
UPDATE Warehouse_data SET quantity=quantity2 WHERE item_id=id;
Return(quantity2);
ELSE
dbms_output.put_line('Does not exist');
Return(quantity2);
END IF;
End;

declare
quantity2 number;
i number;
q number;
begin
i:=:id;
q:=:quantity;
quantity2:=add_quan(i,q);
dbms_output.put_line('Added Quantity: '||quantity2);
end;
```

SQL Statement Running

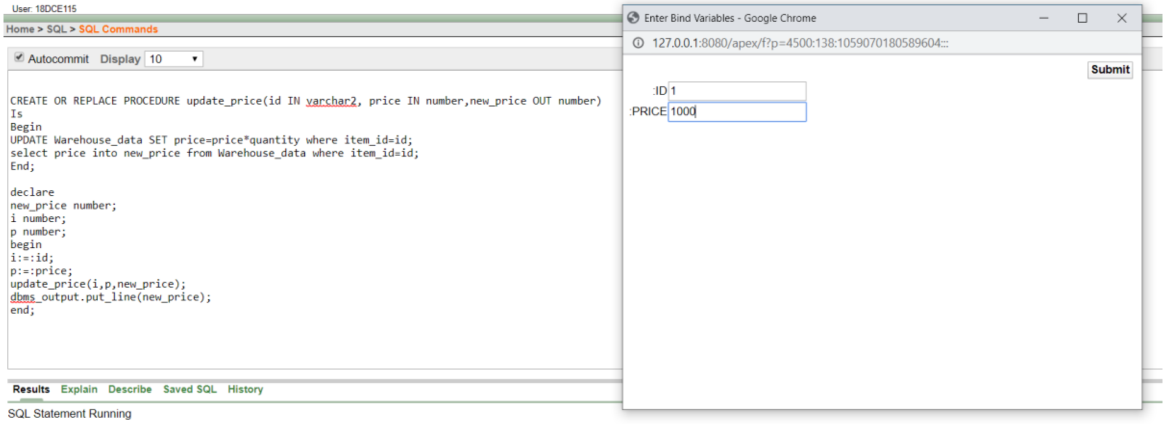
127.0.0.1:8080/apex/f?p=4500:138:1059070180589604:::

:ID 1

:QUANTITY 2

```
CREATE OR REPLACE PROCEDURE update_price(id IN varchar2, price IN
number,new_price OUT number)
Is
Begin
UPDATE Warehouse_data SET price=price*quantity where item_id=id;
select price into new_price from Warehouse_data where item_id=id;
End;

declare
new_price number;
i number;
p number;
begin
i:=:id;
p:=:price;
update_price(i,p,new_price);
dbms_output.put_line(new_price);
end;
```



User: 18DCE115
Home > SQL > SQL Commands
Results
Explain
Describe
Saved SQL
History
Enter Bind Variables - Google Chrome
127.0.0.1:8080/apex/f?p=4500:138:1059070180589604:::
Submit

☑ Autocommit Display 10

```
CREATE OR REPLACE PROCEDURE update_price(id IN varchar2, price IN number,new_price OUT number)
Is
Begin
UPDATE Warehouse_data SET price=price*quantity where item_id=id;
select price into new_price from Warehouse_data where item_id=id;
End;

declare
new_price number;
i number;
p number;
begin
i:=:id;
p:=:price;
update_price(i,p,new_price);
dbms_output.put_line(new_price);
end;
```

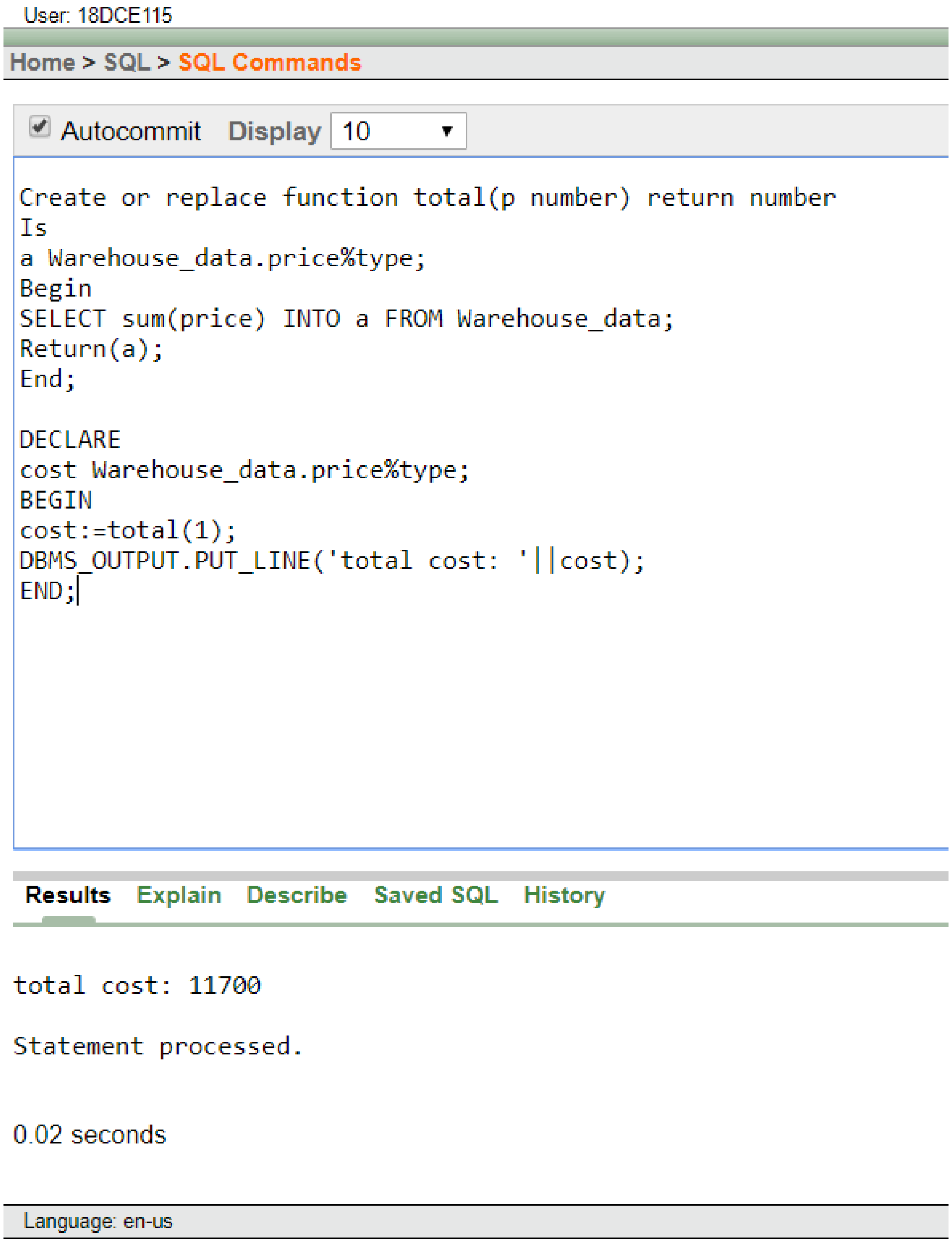
SQL Statement Running

:ID 1

:PRICE 1000

```
Create or replace function total(p number) return number
Is
a Warehouse_data.price%type;
Begin
SELECT sum(price) INTO a FROM Warehouse_data;
Return(a);
End;

DECLARE
cost Warehouse_data.price%type;
BEGIN
cost:=total(1);
DBMS_OUTPUT.PUT_LINE('total cost: '||cost);
END;
```



User: 18DCE115

☑ Autocommit Display 10

```
Create or replace function total(p number) return number
Is
a Warehouse_data.price%type;
Begin
SELECT sum(price) INTO a FROM Warehouse_data;
Return(a);
End;

DECLARE
cost Warehouse_data.price%type;
BEGIN
cost:=total(1);
DBMS_OUTPUT.PUT_LINE('total cost: '||cost);
END;
```

```
total cost: 11700

Statement processed.
```

0.02 seconds

Language: en-us