DBMS Project - Inventory Management System

Course Code	**:**	18CSC303J	**Course Title**	**:**	Database Management Systems
Reg. No.	**:**	RA1811003010580 RA1811003010570 RA1811003010571	**Name**	**:**	HARSHVARDHAN ARVIND SINGH NIRAV AGARWAL YASH S NARANG
Semester	**:**	VI Semester	**Year**	**:**	III Year
Name of the faculty in-charge :			Ms.D.Hemavathi		

AIM - To create an efficient table structure for the DBMS Project on the topic Inventory Management System.

Theory - An inventory management system (or inventory system) is the process by which you track your goods throughout your entire supply chain, from purchasing to production to end sales. It governs how you approach inventory management for your business. It helps processes and procedures that oversee the monitoring and maintenance of stocked products, whether those products are company assets, raw materials and supplies, or finished products ready to be sent to vendors or end consumers.

Tables -

- ➔ Brand
- ➔ Inventory
- ➔ Category
- ➔ Orders
- ➔ Customer_Orders
- ➔ Company
- ➔ Users
- ➔ Transaction
- ➔ Customers
- ➔ Roles
- ➔ Coupon
- ➔ Employee

Table Structure -

- ➔ Brand:
  - ◆ Name
  - ◆ Id (pk auto increment)

- ➔ Category:
  - ◆ Id
  - ◆ Name

- ➔ Company:
  - ◆ Id
  - ◆ Name
  - ◆ GST number
  - ◆ Address
  - ◆ Registration Date
  - ◆ Registration Id

- ➔ Roles:
  - ◆ Id
  - ◆ Name

- ➔ Users:
  - ◆ Role_Id
  - ◆ Id
  - ◆ First Name
  - ◆ Last Name
  - ◆ Email
  - ◆ UserName
  - ◆ passwordHash
  - ◆ Mob No
  - ◆ DOB

- ➔ Customer:
  - ◆ User_Id
  - ◆ Billing address
  - ◆ Shipping address

- Employee:
  - Users_ID
  - Company_ID
  - Employee_ID
  - Salary
  - DOJ

- Inventory
  - ID
  - Name
  - company_ID
  - category_ID
  - BrandId
  - SKU code
  - HS Code
  - Stock
  - MRP
  - costPrice
  - SalePrice
  - Description

- Coupon
  - Code
  - Discount type(percent/fixed price)
  - Discount value
  - Minimum order price

- Orders
  - Id
  - company
  - Items
  - Coupon
  - Total Price
  - transaction_Id

➔ Transactions:

- Id
- CustomerUserId
- paymentMethod
- Amount
- Discount
- dateTime
- reference:(order_id)

ER Diagram -

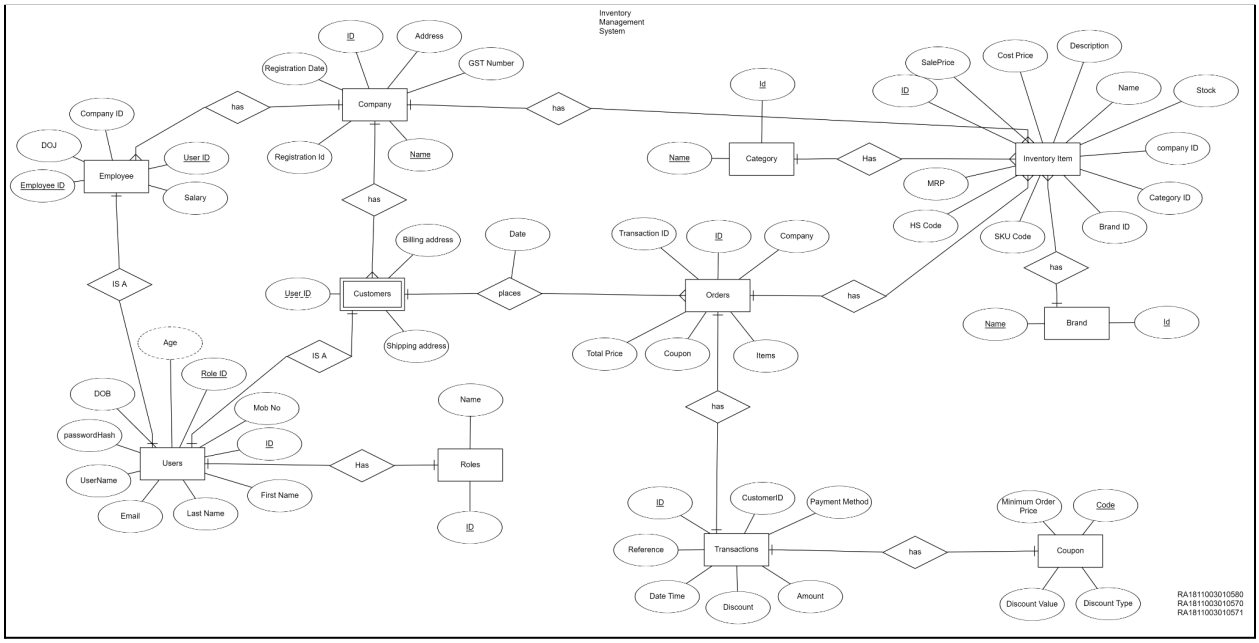


Table Creation and Value Insertion -
BRAND:

CREATE TABLE BRAND(name varchar(50),id int primary key);

```
SQL> CREATE TABLE BRAND(name varchar(50),id int primary key);

Table created.
```

```
Insert into BRAND VALUES('Nike',1);
Insert into BRAND VALUES('Puma',2);
Insert into BRAND VALUES('Jockey',3);
Insert into BRAND VALUES('Gucci',4);
Insert into BRAND VALUES('Reebok',5);
Insert into BRAND VALUES('Cadbury',6);
Insert into BRAND VALUES('Tata',7);
Insert into BRAND VALUES('ITC',8);
```

```
SQL> Insert into BRAND VALUES('Nike',1);

1 row created.

SQL> Insert into BRAND VALUES('Puma',2);

1 row created.

SQL> Insert into BRAND VALUES('Jockey',3);

1 row created.

SQL> Insert into BRAND VALUES('Gucci',4);

1 row created.

SQL> Insert into BRAND VALUES('Reebok',5);

1 row created.

SQL> Insert into BRAND VALUES('Cadbury',6);

1 row created.

SQL> Insert into BRAND VALUES('Tata',7);

1 row created.

SQL> Insert into BRAND VALUES('ITC',8);

1 row created.
```

Category:

CREATE TABLE Category(name varchar(50),id int primary key);

```
SQL> CREATE TABLE Category(name varchar(50),id int primary key);

Table created.
```

Insert into Category VALUES('Shoes',1);
Insert into Category VALUES('Accessories',2);
Insert into Category VALUES('Purse',3);
Insert into Category VALUES('Belts',4);
Insert into Category VALUES('Chocolate',5);
Insert into Category VALUES('Cadbury',6);
Insert into Category VALUES('Shades',7);
Insert into Category VALUES('Notebooks',8);

```
SQL> Insert into Category VALUES('Shoes',1);

1 row created.

SQL> Insert into Category VALUES('Accessories',2);

1 row created.

SQL> Insert into Category VALUES('Purse',3);

1 row created.

SQL> Insert into Category VALUES('Belts',4);

1 row created.

SQL> Insert into Category VALUES('Chocolate',5);
```

```
SQL> select * from category;

NAME                                                        ID
---------------------------------------------------- ----------
Shoes                                                        1
Accessories                                                  2
Purse                                                        3
Belts                                                        4
Chocolate                                                    5
Cadbury                                                      6
Shades                                                       7
Notebooks                                                    8

8 rows selected.
```

Customer_Orders:

CREATE TABLE customer_orders(customer_id int, order_id int, order_date DATE, FOREIGN KEY(customer_id) REFERENCES users(id),FOREIGN KEY(order_id) REFERENCES Orders(id));

```
SQL> Insert into customer_orders values(3,1,'23-Jul-2020');

1 row created.

SQL> Insert into customer_orders values(3,2,'25-Jul-2020');

1 row created.

SQL> Insert into customer_orders values(4,3,'27-Jul-2020');

1 row created.

SQL> Insert into customer_orders values(4,4,'30-Jul-2020');

1 row created.
```

Insert into customer_orders values(3,1,'23-Jul-2020');
Insert into customer_orders values(3,2,'25-Jul-2020');
Insert into customer_orders values(4,3,'27-Jul-2020');
Insert into customer_orders values(4,4,'30-Jul-2020');

```
SQL> CREATE TABLE customer_orders(customer_id int, order_id int, order_date DATE, FOREIGN KEY(customer_id) REFERENCES users(id),FOREIGN KEY(order_id) REFERENCES Orders(id));

Table created.
```

Company:

CREATE TABLE COMPANY(id int primary key,
name varchar(20),
gst_number varchar(15),
address varchar(40),
registration_date Date,
registration_id varchar(20));

Insert into company values(1, 'Laxmi Exports', '10AABCU9603R1Z2', 'KTR-Tamil Nadu', '16-Dec-2020', '26278192');

Insert into company values(2, 'Maharaja Pvt Ltd', '06AABCU9603R1ZR', 'RMP-Tamil Nadu', '20-May-2020', '12279023');

```
SQL> Insert into company values(1, 'Laxmi Exports', '10AABCU9603R1Z2', 'KTR-Tamil Nadu', '16-Dec-2020', '26278192');

1 row created.

SQL>
SQL> Insert into company values(2, 'Maharaja Pvt Ltd', '06AABCU9603R1ZR', 'RMP-Tamil Nadu', '20-May-2020', '12279023');

1 row created.
```

```
SQL> CREATE TABLE COMPANY(id int,
  2  name varchar(20),
  3  gst_number varchar(15),
  4  address varchar(40),
  5  registration_date Date,
  6  registration_id varchar(20));

Table created.
```

```
SQL> alter table company add primary key (id);

Table altered.
```

Inventory:

```
Create TABLE Inventory(id int,
name varchar(30),
company_id int,
category_id int,
brand_id int,
sku_code varchar(12),
hs_code varchar(12),
stock int,
mrp numeric(7),
cost_price numeric(7),
sale_price numeric(7),
Description varchar(50),
FOREIGN KEY (company_id) REFERENCES Company(id),
FOREIGN KEY (company_id) REFERENCES Brand(id),
FOREIGN KEY (company_id) REFERENCES Category(id));
```

```
Insert into inventory values(1, 'MasterBlaster', 1,1, 4, 'TSH-MED-WHI', '85030010',
5000, 4500, 4999, 5500,'This is Shoes');
Insert into inventory values(2, 'PurseTropper', 1, 3, 3, 'LSD-MED-WHI', '64030010',
3000, 2500, 2999, 3500,'This is Purse');
Insert into inventory values(3, 'IIMBelt', 1, 4, 1, 'PSD-MED-WHI', '85030053', 50000,
45000, 49999, 55000,'This is Belt');
Insert into inventory values(4, 'BlastersFire', 2,1, 1, 'QOW-MED-WHI', '85740010',
10000, 9500, 9999, 15000,'This is Shoes');
Insert into inventory values(5, 'WalletKing', 2, 3, 3, 'SMF-MED-WHI', '24030010', 1000,
900, 999,1500,'This is Purse');
Insert into inventory values(6, 'LPABelt', 2, 4, 5, 'LPA-MED-WHI', '85032210', 12000,
11800, 11999,1300, 'This is Belt');
```

```
SQL> Create TABLE Inventory(id int,
  2  name varchar(30),
  3  company_id int,
  4  category_id int,
  5  brand_id int,
  6  sku_code varchar(12),
  7  hs_code varchar(12),
  8  stock int,
  9  mrp numeric(7),
 10  cost_price numeric(7),
 11  sale_price numeric(7),
 12  Description varchar(50),
 13  FOREIGN KEY (company_id) REFERENCES Company(id),
 14  FOREIGN KEY (company_id) REFERENCES Brand(id),
 15  FOREIGN KEY (company_id) REFERENCES Category(id));

Table created.
```

```
SQL> Insert into inventory values(1, 'MasterBlaster', 1,1, 4, 'TSH-MED-WHI', '85030010', 5000, 4500, 4999, 5500,'This is Shoes');

1 row created.

SQL> Insert into inventory values(2, 'PurseTropper', 1, 3, 3, 'LSD-MED-WHI', '64030010', 3000, 2500, 2999, 3500,'This is Purse');

1 row created.

SQL> Insert into inventory values(3, 'IIMBelt', 1, 4, 1, 'PSD-MED-WHI', '85030053', 50000, 45000, 49999, 55000,'This is Belt');

1 row created.

SQL> Insert into inventory values(4, 'BlastersFire', 2,1, 1, 'QOW-MED-WHI', '85740010', 10000, 9500, 9999, 15000,'This is Shoes');

1 row created.

SQL> Insert into inventory values(5, 'WalletKing', 2, 3, 3, 'SMF-MED-WHI', '24030010', 1000, 900, 999,1500,'This is Purse');

1 row created.

SQL> Insert into inventory values(6, 'LPABelt', 2, 4, 5, 'LPA-MED-WHI', '85032210', 12000, 11800, 11999,1300, 'This is Belt');

1 row created.
```

Orders -

CREATE TABLE orders(id int primary key,
transaction_id int,
company_id int,
total_price int,
coupon_code varchar(20),
item varchar(1000),
FOREIGN KEY(transaction_id) REFERENCES Transactions(id), FOREIGN KEY(company_id) REFERENCES company(id), FOREIGN KEY(coupon_code) REFERENCES Coupon(coupon_code));

```
SQL> CREATE TABLE orders(id int primary key,
  2  transaction_id int,
  3  company_id int,
  4  total_price int,
  5  coupon_code varchar(20),
  6  item varchar(1000),
  7  FOREIGN KEY(transaction_id) REFERENCES Transactions(id), FOREIGN KEY(company_id) REFERENCES company(id), FOREIGN KEY(coupon_code) REFERENCES Coupon(coupon_code));

Table created.
```

Insert into orders values(1,1,1,5500,'OFFER','1');
Insert into orders values(2,2,2,3500,'OFFER','2');
Insert into orders values(3,3,1,1500,'OFFER','3');
Insert into orders values(4,4,2,1600,'OFFER','4');

```
SQL> Insert into orders values(1,1,1,5500,'OFFER','1');

1 row created.

SQL> Insert into orders values(2,2,2,3500,'OFFER','2');

1 row created.

SQL> Insert into orders values(3,3,1,1500,'OFFER','3');

1 row created.

SQL> Insert into orders values(4,4,2,1600,'OFFER','4');

1 row created.
```

Transactions -

CREATE TABLE Transactions(id int primary key, customer_id int, payment_method varchar(6), reference int, trans_date DATE, discount int, amount int, coupon_code varchar(10), FOREIGN KEY(customer_id) REFERENCES users(id), FOREIGN KEY(coupon_code) REFERENCES Coupon(coupon_code));

```
SQL> CREATE TABLE Transactions(id int primary key, customer_id int, payment_method varchar(6), reference int, trans_date DATE, discount int, amount int, coupon_code varchar(10), FOREIGN KEY(customer_id) REFERENC
ES users(id), FOREIGN KEY(coupon_code) REFERENCES Coupon(coupon_code));
```

Insert into Transactions values(1,3,'Cash',1, '11-DEC-2020', 1100,5500,'DEAL20');
Insert into Transactions values(2,3,'Card',1, '15-Nov-2020', 875,3500,'NEWYEAR');
Insert into Transactions values(3,4,'Cash',1, '16-Oct-2020', 10,1500,'DEAL10PER');
Insert into Transactions values(4,4,'Card',1, '20-Sep-2020', 15,1600,'OFFER');

```
SQL> Insert into Transactions values(1,3,'Cash',1, '11-DEC-2020', 1100,5500,'DEAL20');

1 row created.
```

```
SQL> Insert into Transactions values(2,3,'Card',1, '15-Nov-2020', 875,3500,'NEWYEAR');

1 row created.

SQL> Insert into Transactions values(3,4,'Cash',1, '16-Oct-2020', 10,1500,'DEAL10PER');

1 row created.
```

```
SQL> Insert into Transactions values(4,4,'Card',1, '20-Sep-2020', 15,1600,'OFFER');

1 row created.
```

Coupon -

CREATE TABLE Coupon(coupon_code varchar(10) primary key, discount_value int, discount_type varchar(10), min_order_price int);

Insert into Coupon VALUES(‘DEAL20’,20,’percent’,200);
Insert into Coupon VALUES(‘OFFER’,15,’fixed’,150);
Insert into Coupon VALUES(‘NEWYEAR’,25,’percent’,0);
Insert into Coupon VALUES(‘DEAL10PER’,10,’fixed,250);

```
SQL> CREATE TABLE Coupon(coupon_code varchar(10) primary key, discount_value int, discount_type varchar(10), min_order_price int);

Table created.
```

```
SQL> Insert into Coupon VALUES('NEWYEAR',25,'percent',0);

1 row created.

SQL> Insert into Coupon VALUES('OFFER',15,'fixed',150);

1 row created.

SQL> Insert into Coupon VALUES('DEAL10PER',10,'fixed',250);

1 row created.
```

```
SQL> Insert into Coupon VALUES('DEAL20',20,'percent',200);

1 row created.
```

Users:

Create table users
(id number primary key,
first_name varchar(50),
last_name varchar(50),
email varchar(50),
username varchar(50),
password_hash varchar(50),
mob_no number(10),
dob date,
role_id number ,
foreign key (role_id) references roles(Id));

```
SQL> Create table users
  2  (id number primary key,
  3  first_name varchar(50),
  4  last_name varchar(50),
  5  email varchar(50),
  6  username varchar(50),
  7  password_hash varchar(50),
  8  mob_no number(10),
  9  dob date,
 10  role_id number ,
 11  foreign key (role_id) references roles(Id));

Table created.
```

Insert into Users VALUES (1,'Raj','Malhotra','raj.malhotra@gmail.com','raju23','sampleHash',8748798777,'04-May-2001',2);

Insert into Users VALUES(2,'Rema','Malhotra','reema.malhotra@gmail.com','rema31','sampleHash1231',8748798666,'01-Mar-2003',2);

Insert into Users VALUES (3,'Jack','Mawia','jack_mawia@hotmail.com','jack71','sampleHash32134',8748798274,'11-OCT-1998'',1);

Insert into Users VALUES(4,'Jim','TimberLake','jim_tim@gmail.com','jimmy6398','sampleHash23231',9548318777,'04-NOV-1988',1);

```
SQL> Insert into Users VALUES (1,'Raj','Malhotra','raj.malhotra@gmail.com','raju23','sampleHash',8748798777,'04-May-2001',2);

1 row created.

SQL>
SQL> Insert into Users VALUES(2,'Rema','Malhotra','reema.malhotra@gmail.com','rema31','sampleHash1231',8748798666,'01-Mar-2003',2);

1 row created.

SQL>
SQL> Insert into Users VALUES (3,'Jack','Mawia','jack_mawia@hotmail.com','jack71','sampleHash32134',8748798274,'11-OCT-1998'',1);
ERROR:
ORA-01756: quoted string not properly terminated


SQL>
SQL> Insert into Users VALUES(4,'Jim','TimberLake','jim_tim@gmail.com','jimmy6398','sampleHash23231',9548318777,'04-NOV-1988',1);

1 row created.

SQL> Insert into Users VALUES (3,'Jack','Mawia','jack_mawia@hotmail.com','jack71','sampleHash32134',8748798274,'11-OCT-1998',1);

1 row created.
```

Roles:

Create table roles(id number primary key,name varchar(50));

```
SQL> Create table roles(id number primary key,name varchar(50));

Table created.
```

Insert into roles VALUES(1,'customer');
Insert into roles VALUES(2,'employee');

```
SQL> Insert into roles values(1,'customer');

1 row created.

SQL> Insert into roles values(2,'employee');

1 row created.
```

Customer:

Create table customer(billing_address varchar(50),shipping_address varchar(50),user_id number, foreign key(user_id) references users(id));

```
SQL> Create table customer( billing_address varchar(50),shipping_address varchar(50),user_id number, foreign key(user_id ) references users(id));

Table created.
```

Insert into customer VALUES('7th main road, langur nagar muradabad India','7th main road, langur nagar muradabad India',3);
Insert into customer VALUES('13th main road, shivaji nagar,nagpur India','13th main road, shivaji nagar,nagpur India',4);

```
SQL> Insert into customer values ('7th main road, langur nagar muradabad India','7th main road, langur nagar muradabad India',3);

1 row created.

SQL> Insert into customer values ('13th main road, shivaji nagar,nagpur India','13th main road, shivaji nagar,nagpur India',4);

1 row created.
```

Employee:

Create table employee(employee_id number primary key,company_id int,user_id number,salary number,doj date,foreign key(company_id) references company(id),foreign key(user_id) references users(id));

```
SQL> Create table employee(employee_id number primary key,company_id int,user_id number,salary number,doj date,foreign key(company_id) references  company(id),foreign key(user_id) references users(id));

Table created.
```

Insert into employee VALUES(2328,1,1,20000,’11-May-2018’);
Insert into employee VALUES(2329,2,2,18000,’15-APR-2020’);

```
SQL> Insert into employee VALUES(2328,1,1,20000,’11-May-2018’);

1 row created.

SQL> Insert into employee VALUES(2329,2,2,18000,’15-APR-2020’);

1 row created.
```

Operations -

-> Get All order of a specific customer

select * from orders where id in (select order_id from customer_orders where customer_id=3);

```
SQL> select * from orders where id in (select order_id from customer_orders where customer_id=3);

        ID TRANSACTION_ID COMPANY_ID TOTAL_PRICE COUPON_CODE
---------- -------------- ---------- ----------- --------------------
ITEM
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------
         1              1          1        5500 DEAL20
1

         2              2          2        3500 OFFER
2
```

->Get the total amount spent by a specific customer

select sum(total_price) as Amount from orders where id in (select order_id from customer_orders where customer_id=3);

```
SQL> select sum(total_price) as Amount from orders where id in (select order_id from customer_orders where customer_id=3);

    AMOUNT
----------
      9000
```

->Get All orders of a company

select * from orders where company_id=1;

```
SQL> select * from orders where company_id=1;

        ID TRANSACTION_ID COMPANY_ID TOTAL_PRICE COUPON_CODE
---------- -------------- ---------- ----------- --------------------
ITEM
--------------------------------------------------------------------------------
         1              1          1        5500 OFFER
1

         3              3          1        1500 OFFER
3
```

->Get the total order price grouped by company
Select company_id ,sum(total_price) as TotalSales from orders group by(company_id);

```
SQL> Select company_id ,sum(total_price) as TotalSales from orders group by(company_id);

COMPANY_ID TOTALSALES
---------- ----------
         1       7000
         2       5100
```

->Get total discount given to customers grouped by company
select orders.company_id,sum(transactions.discount) from orders full outer join transactions on orders.transaction_id=transactions.id group by(company_id);

```
SQL> select orders.company_id,sum(transactions.discount) from orders full outer join transactions on orders.transaction_id=transactions.id group by(company_id);

COMPANY_ID SUM(TRANSACTIONS.DISCOUNT)
---------- --------------------------
         1                      1110
         2                       890
```

->Get total amount spent by all customers grouped by all customer ID
select 'customer with id '||customer_id || ' spent ' || sum(amount) as CustomerSpending from transactions group by(customer_id);

```
SQL> select 'customer with id '||customer_id || ' spent ' || sum(amount) as Customer_Spending from transactions group by(customer_id);

CUSTOMER_SPENDING
--------------------------------------------------------------------------------------------------------
customer with id 4 spent 3100
customer with id 3 spent 9000
```

-> Get Pre Discounted price
Select id,amount+discount as CostPrice from transactions;

```
SQL> Select id,amount+discount as CostPrice from transactions;

        ID  COSTPRICE
---------- ----------
         1       6600
         2       4375
         3       1510
         4       1615
```

->View to print products with low stock

create or replace view low_stock as select 'The stock for the product ' ||name|| ' is ' || stock as ProductStock from inventory where stock<5000;

```
SQL> create or replace view low_stock as select 'The stock for the product ' ||name|| ' is ' || stock as ProductStock from inventory where stock<5000;

View created.
```

```
SQL> select * from low_stock;

PRODUCTSTOCK
--------------------------------------------------------------------------------
The stock for the product MasterBlaster is 4999
The stock for the product PurseTropper is 3000
The stock for the product WalletKing is 1000
```

JOINS

->RIGHT OUTER JOIN

select users.username,users.mob_no,customer.user_id,users.id from customer right outer join users on customer.user_id=users.id;

```
USERNAME                                            MOB_NO    USER_ID         ID
-------------------------------------------------- ---------- ---------- ----------
raju23                                           8748798777                     1
rema31                                           8748798666                     2
jack71                                           8748798274          3          3
jimmy6398                                        9548318777          4          4
```

->LEFT OUTER JOIN

select users.username,users.mob_no,customer.user_id,users.id from users left outer join customer on customer.user_id=users.id;

```
SQL> select users.username,users.mob_no,customer.user_id,users.id from users left outer join customer  on customer.user_id=users.id;

USERNAME                                            MOB_NO    USER_ID         ID
-------------------------------------------------- ---------- ---------- ----------
raju23                                           8748798777                     1
rema31                                           8748798666                     2
jack71                                           8748798274          3          3
jimmy6398                                        9548318777          4          4
```

->INNER JOIN

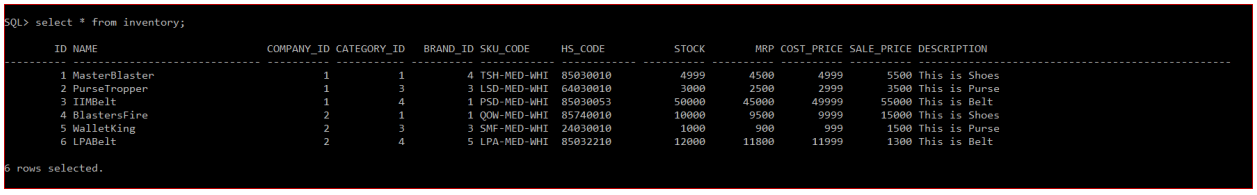
select orders.company_id,sum(transactions.discount) from orders inner join transactions on orders.transaction_id=transactions.id group by(company_id);

```
SQL> select orders.company_id,sum(transactions.discount) from orders inner join transactions on orders.transaction_id=transactions.id group by(company_id);

COMPANY_ID SUM(TRANSACTIONS.DISCOUNT)
---------- --------------------------
         1                      1110
         2                       890
```

Trigger for updating inventory -

```
CREATE OR REPLACE TRIGGER inventory_update
   before
   INSERT ON orders
referencing old as old new as new
FOR EACH ROW
declare
   order_num number;
   temp_value varchar(20);
begin
   temp_value := :new.item;
   order_num := TO_NUMBER(temp_value);
        update inventory set stock = stock - 1 where id = order_num;
        dbms_output.put_line ( 'Inventory Updated for Inventory ID ' || order_num);


end;
/
Insert into orders values(5,4,2,2600,'OFFER','4');
```

BEFORE:



```
SQL> select * from inventory;

        ID NAME                           COMPANY_ID CATEGORY_ID   BRAND_ID SKU_CODE     HS_CODE          STOCK        MRP COST_PRICE SALE_PRICE DESCRIPTION
---------- ------------------------------ ---------- ----------- ---------- ------------ ---------- ---------- ---------- ---------- ----------------------------------------
         1 MasterBlaster                           1           1          4 TSH-MED-WHI  85030010           4999       4500       4999       5500 This is Shoes
         2 PurseTropper                            1           3          3 LSD-MED-WHI  64030010           3000       2500       2999       3500 This is Purse
         3 TIMBelt                                 1           4          1 PSD-MED-WHI  85030053          50000      45000      49999      55000 This is Belt
         4 BlastersFire                            2           1          1 QOW-MED-WHI  85740010          10000       9500       9999      15000 This is Shoes
         5 WalletKing                              2           3          3 SMF-MED-WHI  24030010           1000        900        999       1500 This is Purse
         6 LPABelt                                 2           4          5 LPA-MED-WHI  85032210          12000      11800      11999       1300 This is Belt

6 rows selected.
```

TRIGGER:

```
SQL> CREATE OR REPLACE TRIGGER inventory_update
  2      before
  3      INSERT ON orders
  4  referencing old as old new as new
  5  FOR EACH ROW
  6  declare
  7      order_num number;
  8      temp_value varchar(20);
  9  begin
 10      temp_value := :new.item;
 11      order_num := TO_NUMBER(temp_value);
 12  update inventory set stock = stock - 1 where id = order_num;
 13  dbms_output.put_line ( 'Inventory Updated for Inventory ID ' || order_num);
 14
 15
 16  end;
 17  /

Trigger created.

SQL> Insert into orders values(5,4,2,2600,'OFFER','4');

1 row created.
```

AFTER:

```
SQL> select id,stock from inventory;

        ID      STOCK
---------- ----------
         1       4999
         2       3000
         3      50000
         4       9999
         5       1000
         6      12000

6 rows selected.
```

Function to Display stock for a product -

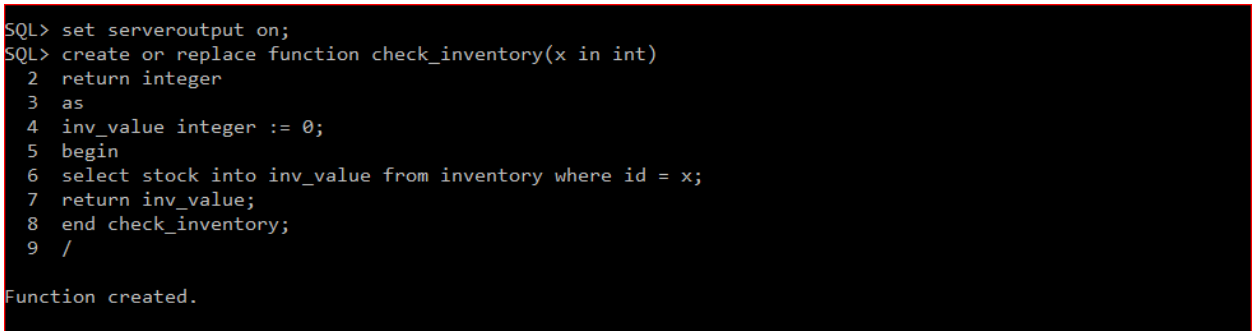
set serveroutput on;

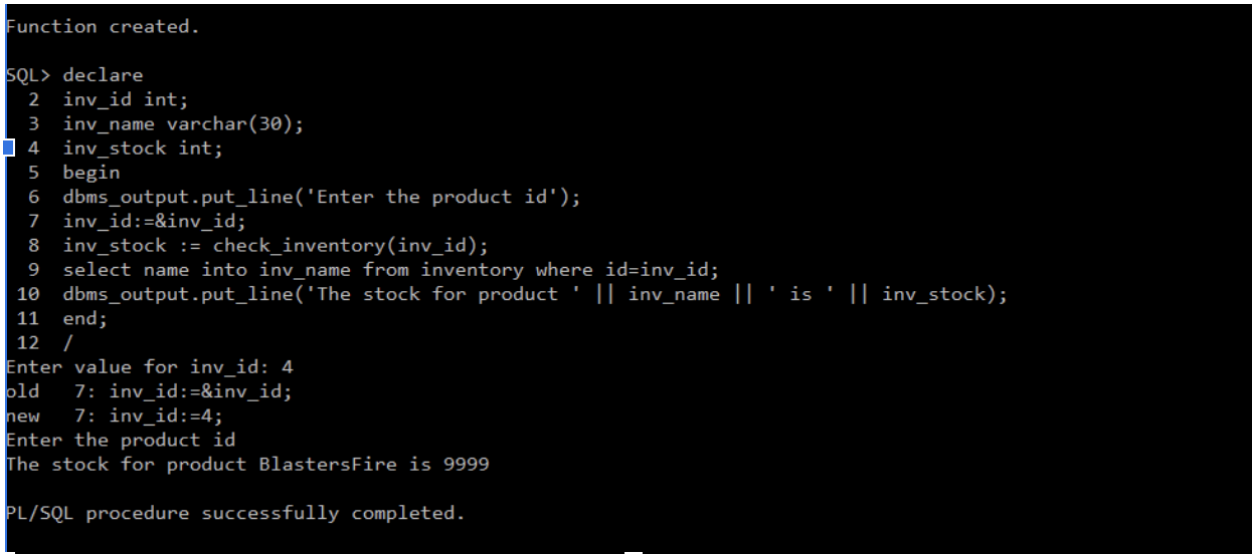
create or replace function check_inventory(x in int)

return integer

as

```
inv_value integer := 0;
begin
select stock into inv_value from inventory where id = x;
return inv_value;
end check_inventory;
/

declare
inv_id int;
inv_name varchar(30);
inv_stock int;
begin
dbms_output.put_line('Enter the product id');
inv_id:=&inv_id;
inv_stock := check_inventory(inv_id);
select name into inv_name from inventory where id=inv_id;
dbms_output.put_line('The stock for product ' || inv_name || ' is ' || inv_stock);
end;
/
```





```
Function created.

SQL> declare
  2  inv_id int;
  3  inv_name varchar(30);
  4  inv_stock int;
  5  begin
  6  dbms_output.put_line('Enter the product id');
  7  inv_id:=&inv_id;
  8  inv_stock := check_inventory(inv_id);
  9  select name into inv_name from inventory where id=inv_id;
 10  dbms_output.put_line('The stock for product ' || inv_name || ' is ' || inv_stock);
 11  end;
 12  /
Enter value for inv_id: 4
old   7: inv_id:=&inv_id;
new   7: inv_id:=4;
Enter the product id
The stock for product BlastersFire is 9999

PL/SQL procedure successfully completed.
```

Result - An efficient table structure for the DBMS Project on the topic Inventory Management System was successfully created.
