**Computer Center**

الدوال و الإجراءات اللازمة للمشروع

إجراء يقوم بإضافة كميات جديدة من منتجات معينة الى جدول المنتجات

CREATE OR REPLACE PROCEDURE addProductQuantity (p\_Id INT,NewQuantity INT )

AS OldQuantity INT;

BEGIN

SELECT QuantityInStock INTO OldQuantity FROM Product WHERE ProductId = p\_Id;

UPDATE Product

SET QuantityInStock = OldQuantity + NewQuantity

WHERE ProductId = p\_Id;

DBMS\_OUTPUT.PUT\_LINE('Quantity of product ' || p\_Id || ' updated successfully.');

END;

--------------أمر الإستدعاء----------------

begin

addProductQuantity(1,10);

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

إجراء يقوم بإضافة طلب جديد في جدول الطلبات و لإضافة بياناته في جدول تفاصيل الطلب

مع شرط التحقق من الكمية المتوفر قبل تنفيذ الطلب

CREATE OR REPLACE PROCEDURE addNewOrder (

in\_orderId IN INT,

in\_orderDate IN DATE,

in\_productId IN INT,

in\_customerId IN INT,

in\_quantity IN INT,

in\_totalPrice IN INT

)

AS

availableQuantity INT;

BEGIN

SELECT QuantityInStock INTO availableQuantity

FROM Product

WHERE ProductId = in\_productId;

IF availableQuantity < in\_quantity THEN

DBMS\_OUTPUT.PUT\_LINE('The quantity required is insufficient');

ELSE

INSERT INTO Orders (OrderId, OrderDate)

VALUES (in\_orderId, in\_orderDate);

INSERT INTO Order\_Details (OrderId, ProductId, CustomerId, Quantity)

VALUES (in\_orderId, in\_productId, in\_customerId, in\_quantity);

DBMS\_OUTPUT.PUT\_LINE('Order added successfully.');

END IF;

END;

--------------أمر الإستدعاء----------------

BEGIN

addNewOrder (15,'7-2-2024',1,1,10,200);

END;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

اجراء يقوم بإضافة فاتورة و تفاصيل الفاتورة

CREATE OR REPLACE PROCEDURE addNewInvoice (

in\_invoiceId IN INT,

in\_invoiceDate IN DATE,

in\_customerId IN INT,

in\_orderId IN INT

)

AS

TotalAmount INT := 0;

BEGIN

INSERT INTO Invoice (Id, IDate)

VALUES (in\_invoiceId, in\_invoiceDate);

INSERT INTO Invoice\_Customer (InvoiceId, CustomerId, OrderId)

VALUES (in\_invoiceId, in\_customerId, in\_orderId);

SELECT SUM(TOTAL\_PRICE) INTO TotalAmount

FROM Order\_Details

WHERE OrderId = in\_orderId;

DBMS\_OUTPUT.PUT\_LINE('Invoice added successfully.');

DBMS\_OUTPUT.PUT\_LINE(TotalAmount );

END;

--------------أمر الإستدعاء----------------

begin

addNewInvoice(9,'5-5-2024',1,1);

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

إجراء يقوم بحساب فاتورة طلبات زبون محدد

CREATE OR REPLACE PROCEDURE CalcInvoice (in\_customerId IN INT)

AS

TotalAmount INT := 0;

BEGIN

SELECT SUM(TOTAL\_PRICE) INTO TotalAmount

FROM Order\_Details

WHERE CUSTOMERID = in\_customerId;

DBMS\_OUTPUT.PUT\_LINE('TotalAmount: ' ||TotalAmount );

END;

--------------أمر الإستدعاء----------------

begin

CalcInvoice(1);

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

يتم انقاص الكمية المتوفرة في كل عملية شراء

CREATE OR REPLACE TRIGGER trig3

AFTER INSERT ON Order\_Details

FOR EACH ROW

BEGIN

UPDATE Product

SET QuantityInStock = QuantityInStock - :NEW.Quantity

WHERE ProductId = :NEW.ProductId;

END;

/

--------------أمر الإستدعاء----------------

insert into Order\_details values(2,2,2,20,200);

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

VIEW /منتجات الشركة الموّردة

CREATE OR REPLACE FORCE VIEW ProductOfSupplier

As

select product.name,category,description,price from product,supplier

where ProductId=SupplierId;

--------------أمر الإستدعاء----------------

select \* from ProductOfSupplier;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

VIEW /فاتورة الزبون

CREATE OR REPLACE FORCE VIEW CustomerOrder

As

select name,ProductId,quantity,total\_price from Customer,Order\_details

where Customer.CustomerId=OrderId;

--------------أمر الإستدعاء----------------

select \* from CustomerOrder

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

VIEW /رواتب الموظفين في كل قسم

CREATE OR REPLACE FORCE VIEW EmpSec

As

select Sections.name,Section,job\_title,Salary from Sections,Employee

where Sections.Id=Employee.section;

--------------أمر الإستدعاء----------------

select \* from EmpSec

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

إجراء يقوم بحساب و طباعة إجمالي المبيعات

CREATE OR REPLACE PROCEDURE GetTotalSalesOverall

AS

x int;

BEGIN

select sum(total\_price) into x from order\_details;

dbms\_output.put\_line('total sales are ' || x);

END;

--------------أمر الإستدعاء----------------

begin

GetTotalSalesOverall;

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

اجمالي المبيعات بين تاريخين

CREATE OR REPLACE PROCEDURE GetTotalSalesBetween (y date , z date)

AS

a date;

x int;

BEGIN

for i in 1..10 loop

select OrderDate into a from Orders where OrderId=i ;

end loop;

if a between y and z then

select sum(total\_price) into x from order\_details ;

dbms\_output.put\_line(' sales are ' || x);

end if;

End;

--------------أمر الإستدعاء----------------

begin

GetTotalSalesBetween('1/1/1999' ,'1/1/2026');

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

اعادة تفاصيل طلب زبون من خلال رقمه

CREATE OR REPLACE PROCEDURE getCustomerOrder (Cid int)

AS

Type arr is table of int index by pls\_integer;

x arr;

begin

select OrderId,ProductId,Quantity,Total\_price into x(1),x(2),x(3),x(4) from Order\_Details where CustomerId=Cid;

dbms\_output.put\_line(' OrderId:' || x(1) ||'~~~'|| ' ProductId:' || x(2) ||'~~~'|| ' Quantity:' || x(3) ||'~~~'|| ' Total\_price:' || x(4) );

End;

--------------أمر الإستدعاء----------------

begin

getCustomerOrder (1);

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

اجراء ديناميكي لاظهار بيانات المنتجات الي يحويها القسم عن طريق رقم القسم

create or replace procedure getdata(se int )

is

y int;

c product%rowtype;

begin

select PRODUCTS into y from sections where id= se;

select \* into c from product where PRODUCTID=y;

dbms\_output.put\_line(c.PRODUCTID||' '||c.name||' '||c.DESCRIPTION||' '||c.PRICE||' '||c.QUANTITYINSTOCK||' '||c.CATEGORYID||' '||c.SUPPLIER);

end;

--------------أمر الإستدعاء----------------

begin

getdata(2);

end**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

اجراء ديناميكي يقوم بالاستعلام عن بيانات الزبون عن طريق رقم الفاتورة

create or replace procedure datacustomer(inv int )

is

y int;

c customer%rowtype;

begin

select CUSTOMERID into y from Invoice\_Customer where InvoiceId=inv;

select \* into c from customer where CUSTOMERID =y;

dbms\_output.put\_line(c.CUSTOMERID||' '||c.name||' '||c.EMAIL||' '||c.ADDRESS);

end;

--------------أمر الإستدعاء----------------

begin

datacustomer(2);

end;

**&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&**

تم بعون الله

دكتور المادة:د.عبدالجبار عباس

**عمل الطلاب:**

صالح خالوصي

صالح العيروط

علي رضا الرشيد

عمران الأنطكلي

عبد الجبار الصالح

محمد خير عيسى