## **Computer Center**

## الدوال و الإجراءات اللازمة للمشروع

إجراء بقوم بإضافة كمبات جديدة من منتجات معبنة 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\_ld || ' updated successfully.'); END; -----أمر الإستدعاء-----أمر begin addProductQuantity;(\,\\\) 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 (Orderld, Productld, Customerld, 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 invoiceld 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 (Invoiceld, Customerld, Orderld)
 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-----أمر
begin
addNewInvoice(9,'5-5-2024',1,1);
end;
إجراء يقوم بحساب فاتورة طلبات زبون محدد
```

```
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-----أمر
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 الاستدعاء-----
insert into Order details values(2,2,2,20,200);
منتجات الشركة الموردة/ VIEW
CREATE OR REPLACE FORCE VIEW ProductOfSupplier
As
select product.name,category,description,price from product,supplier
where ProductId=SupplierId;
-----أمر الإستدعاء-----
select * from ProductOfSupplier;
فاتورة الزبون/ VIEW
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
رواتب الموظفين في كل قسم/ VIEW
CREATE OR REPLACE FORCE VIEW EmpSec
As
select Sections.name, Section, job title, Salary from Sections, Employee
where Sections.Id=Employee.section;
-----أمر الاستدعاء-----أمر
select * from EmpSec
إجراء يقوم بحساب وطباعة 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 الاستدعاء-----
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------أمر
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_line(' Orderld:' || x(1) || '~~~'|| ' Productld:' || x(2) || '~~~'|| ' Quantity:'
|| x(3) ||'~~~'|| ' Total_price:' || x(4) );
End;
-----أمر الإستدعاء-----أمر
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_line(c.PRODUCTID||''||c.name||''||c.DESCRIPTION||''||c.PRICE||'
'||c.QUANTITYINSTOCK||''||c.CATEGORYID||''||c.SUPPLIER);
end;
-----أمر الاستدعاء-----
begin
getdata(2);
&&&
            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;
```

عمل الطلاب: صالح خالوصي صالح العيروط علي رضا الرشيد عمران الأنطكلي

عبد الجبار الصالح محمد خير عيسى