MySQL Assignment – 4

1. Create a stored procedure in the Northwind database that will calculate the average value of Freight for a specified customer.Then, a business rule will be added that will be triggered before every Update and Insert command in the Orders controller,and will use the stored procedure to verify that the Freight does not exceed the average freight. If it does, a message will be displayed and the command will be cancelled.

Insert Trigger -

DROP TRIGGER IF EXISTS `northwind`.`check\_freight\_before\_insert`;

DELIMITER $$

CREATE TRIGGER `check\_freight\_before\_insert` BEFORE INSERT ON `orders` FOR EACH ROW BEGIN

DECLARE avg\_freight double;

select avg(Freight) into avg\_freight from orders where CustomerID=new.CustomerID;

if new.Freight > avg\_freight then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "Error Inserting Record, Your Average Freight Value is Greater than Customer Freight Value";

end if;

END$$

DELIMITER ;

INSERT INTO `northwind`.`orders` (`OrderID`, `CustomerID`, `EmployeeID`, `OrderDate`, `RequiredDate`, `ShipVia`, `Freight`, `ShipName`, `ShipAddress`, `ShipCity`, `ShipPostalCode`, `ShipCountry`) VALUES ('11078', 'VINET', '4', '1998-05-06 00:00:00', '1998-05-06 00:00:00', '2', '100', 'Rattlesnake Canyon Grocery', '12, rue des Bouchers', 'Marseille', '380007', 'India');



INSERT INTO `northwind`.`orders` (`OrderID`, `CustomerID`, `EmployeeID`, `OrderDate`, `RequiredDate`, `ShipVia`, `Freight`, `ShipName`, `ShipAddress`, `ShipCity`, `ShipPostalCode`, `ShipCountry`) VALUES ('11078', 'VINET', '4', '1998-05-06 00:00:00', '1998-05-06 00:00:00', '2', '5.5', 'Rattlesnake Canyon Grocery', '12, rue des Bouchers', 'Marseille', '380007', 'India');



Update Trigger

DELIMITER $$

CREATE TRIGGER check\_freight\_before\_update

BEFORE UPDATE

ON orders FOR EACH ROW

BEGIN

DECLARE avg\_freight double;

select avg(Freight) into avg\_freight from orders where CustomerID=new.CustomerID;

if new.Freight > avg\_freight then

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "Error Updating Record, Your Average Freight Value is Greater than Customer Freight Value";

end if;

END$$

UPDATE `northwind`.`orders` SET `Freight` = '1000' WHERE (`OrderID` = '10248');



UPDATE `northwind`.`orders` SET `Freight` = '1.5' WHERE (`OrderID` = '10248');



2. write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country

drop procedure if exists `employee\_sales\_by\_country`;

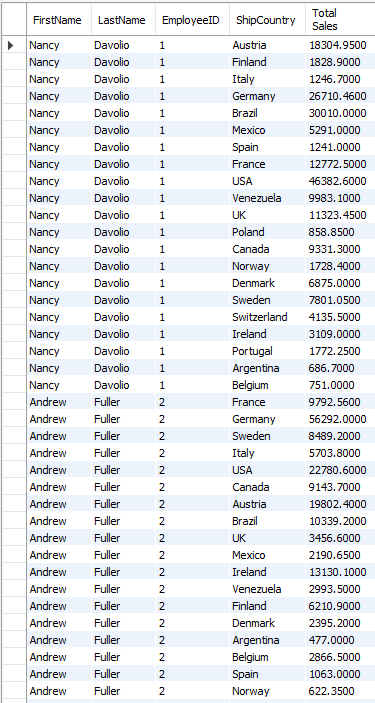
create procedure `employee\_sales\_by\_country` ()

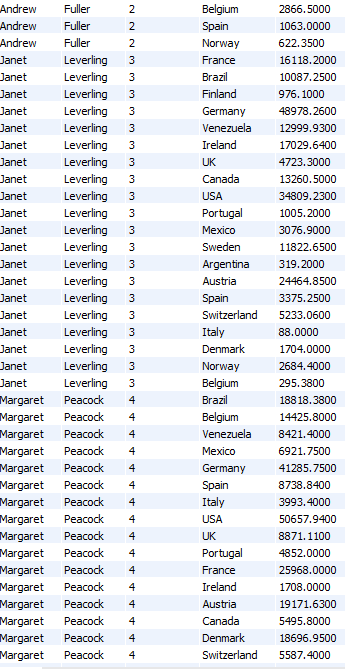
begin

select e.FirstName, e.LastName, e.EmployeeID, k.ShipCountry, k.`Total Sales` from employees e inner join (select o.EmployeeID, o.ShipCountry, sum(od.UnitPrice\*od.Quantity) as "Total Sales" from orders o inner join `order details` od on o.OrderID=od.OrderID group by o.EmployeeID, o.ShipCountry) k on e.EmployeeID=k.EmployeeID;

end $$

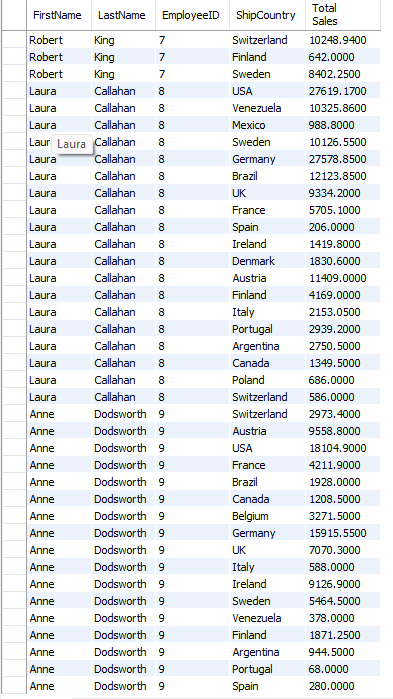
call employee\_sales\_by\_country();





…

…



3. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year

DELIMITER $$

drop procedure if exists `sales\_by\_year`;

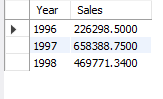
create procedure `sales\_by\_year` ()

begin

select year(o.OrderDate) 'Year', sum(od.UnitPrice \* od.Quantity) Sales from `order details` od inner join orders o on od.OrderID=o.OrderID group by year(o.OrderDate);

end $$

call sales\_by\_year();



4. write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category

DELIMITER $$

drop procedure if exists `sales\_by\_category`;

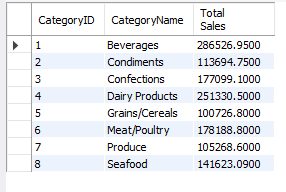
create procedure `sales\_by\_category` ()

begin

select p.CategoryID, c.CategoryName, sum(od.UnitPrice \* od.Quantity) 'Total Sales' from `order details` od inner join products p on p.ProductID=od.ProductID inner join categories c on p.CategoryID=c.CategoryID group by p.CategoryID;

end $$

call sales\_by\_category();



5. write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products

drop procedure if exists `most\_expensive\_product`;

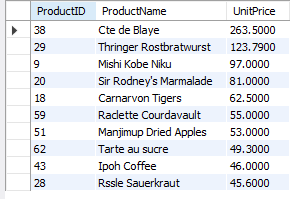
create procedure `most\_expensive\_product` (in max\_limit int)

begin

select ProductID, ProductName, UnitPrice from products order by UnitPrice desc limit max\_limit;

end $$

call most\_expensive\_product(10);



6. write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details

DELIMITER $$

drop procedure if exists `insert\_customer\_order`;

create procedure `insert\_customer\_order` (in CustomerID varchar(5), in EmployeeID int, in OrderDate datetime, in RequiredDate datetime, in ShipVia int, in Freight double, in ShipName varchar(40), in ShipAddress varchar(60), in ShipCity varchar(15),in ShipRegion varchar(15), in ShipPostalCode varchar(10), in ShipCountry varchar(15), in ProductID int, in UnitPrice double, in Quantity int, in Discount int, in orderID int, in addingProduct bool)

begin

DECLARE newOID int;

if addingProduct then

INSERT INTO `order details` (`OrderID`, `ProductID`, `UnitPrice`, `Quantity`, `Discount`) VALUES (orderID, ProductID, UnitPrice, Quantity, Discount);

else

INSERT INTO `orders` (`CustomerID`, `EmployeeID`, `OrderDate`, `RequiredDate`, `ShipVia`, `Freight`, `ShipName`, `ShipAddress`, `ShipCity`, `ShipRegion`, `ShipPostalCode`, `ShipCountry`)

VALUES (CustomerID, EmployeeID, OrderDate, RequiredDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry);

select o.oID into newOID from (select max(k.OrderID) oID from orders k) o;

select newOID as 'New Order ID';

INSERT INTO `order details` (`OrderID`, `ProductID`, `UnitPrice`, `Quantity`, `Discount`) VALUES (newOID, ProductID, UnitPrice, Quantity, Discount);

end if;

end $$

DELIMITER ;

call insert\_customer\_order('RICSU', 4, '1998-05-06 00:00:00', '1998-05-06 00:00:00', 2, 5.1, 'Rattlesnake Canyon Grocery', '2817 Milton Dr.', 'Ahmedabad', 'North', '380007', 'India', 42, 25.5, 2, 0, null, false);



Record Inserted:

SELECT \* FROM northwind.orders where OrderID=11091;

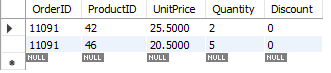


Inserting Records Only in “Order Details”

call insert\_customer\_order(null, null, null, null, null, null, null, null, null, null, null, null, 46, 20.5, 5, 0, 11091, true);

Record Inserted:

SELECT \* FROM northwind.`order details` where OrderId=11091;



7. write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details

DELIMITER $$

drop procedure if exists `update\_customer\_order`;

create procedure `update\_customer\_order` (in iOrderID int, in nCustomerID varchar(5), in nEmployeeID int, in nOrderDate datetime, in nRequiredDate datetime, in nShipVia int, in nFreight double, in nShipName varchar(40), in nShipAddress varchar(60), in nShipCity varchar(15),in nShipRegion varchar(15), in nShipPostalCode varchar(10), in nShipCountry varchar(15))

begin

update orders set

`CustomerID`=nCustomerID,`EmployeeID`=nEmployeeID, `OrderDate`=nOrderDate, `RequiredDate`=nRequiredDate, `ShipVia`=nShipVia,

`Freight`=nFreight, `ShipName`=nShipName, `ShipAddress`=nShipAddress, `ShipCity`=nShipCity, `ShipRegion`=nShipRegion,

`ShipPostalCode`=nShipPostalCode, `ShipCountry`=nShipCountry where OrderID=iOrderID;

end $$

DELIMITER ;

call update\_customer\_order(11091, 'VINET', 2, '1998-05-06 00:00:00', '1998-05-06 00:00:00', 2, 1.1, 'Grocery Store', 'Paldi', 'Ahmedabad', 'West', '380007', 'India');



SELECT \* FROM northwind.orders where OrderID=11091;

