**Subject: DataBase**

**Project: Courier Management System**

Logo, company name

Description automatically generated

**Submitted by:**

**FA20-BSE-107(Ameer Moavia)**

**Submitted to:**

**Sir Abid Sohail Bhutta**

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

**Procedures**

/\*Procedure to show specific customer from a branch\*/

CREATE PROCEDURE branchDetails

@BRANCH varchar(200),

@Customer int

AS

BEGIN

SELECT b.BillingNumber,b.BranchCode,b.CustomerID,b.ParcelNumber,bc.Price, e.EmployeeID,vs.VehicleNumber,b.Date

FROM Billing b

INNER JOIN Branch br ON b.BranchCode=br.BranchCode

INNER JOIN BillingCategory bc ON bc.CategoryID=b.CategoryID

INNER JOIN Employee e ON br.BranchCode=e.BranchCode

INNER JOIN VehicleShift vs ON e.EmployeeID=vs.EmployeeID

INNER JOIN CourrierVehicle cv ON cv.VehicleNumber=vs.VehicleNumber

Where b.BranchCode = @BRANCH And b.CustomerID = @Customer

END

EXEC branchDetails 'HM01','116'

EXEC branchDetails 'HM34','284'

EXEC branchDetails 'HM34','348'

Table

Description automatically generated

/\*Procedure to SHOW COMPLAINT ON A DATE\*/

CREATE PROCEDURE complaintData

@date DATE

AS

BEGIN

SELECT c.CustomerID, c.ComplaintNumber, c.Description, c.ComplaintDate, s.status

FROM Complaint c

INNER JOIN Status s ON s.status\_ID= c.Status

WHERE c.ComplaintDate= @date

END

EXEC complaintData '2020-08-14'

Graphical user interface, text, application, email

Description automatically generated

/\*Procedure to SHOW VEHICLE DETAILS of a EMPLOYEE\*/

CREATE PROCEDURE vehicleShiftDetails

@employeeID INT

AS

BEGIN

SELECT e.EmployeeID,e.BranchCode,e.Name,e.PhoneNumber,vs.VehicleNumber,vs.WorkingHours FROM Employee e

INNER JOIN VehicleShift vs ON e.EmployeeID=vs.EmployeeID

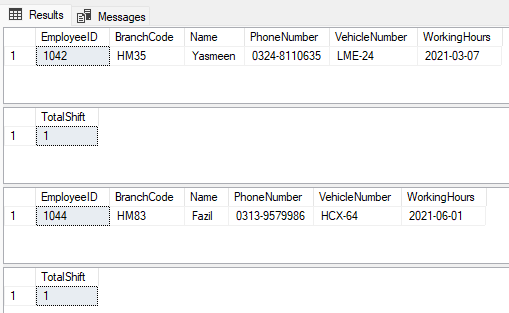
WHERE e.EmployeeID=@employeeID

SELECT @@ROWCOUNT AS TotalShift

END

EXEC vehicleShiftDetails '1042'

EXEC vehicleShiftDetails '1044'



/\*Procedure to search Parcel by deparure city and arrival city\*/

CREATE PROCEDURE searchbyCity

@departure varchar(200),

@arrival varchar(200)

AS

BEGIN

SELECT c.CustomerID,c.Name,c.PhoneNumber,c.City AS DepartureCity,d.City AS ArrivalCity,b.BillingNumber,b.ParcelNumber,bil.Price,s.status

FROM Customer c

INNER JOIN DeliveryInfo d ON c.CustomerID=d.CustomerID

INNER JOIN Status s ON d.DeliveryStatus=s.status\_ID

INNER JOIN Billing b ON c.CustomerID=b.CustomerID

INNER JOIN BillingCategory bil ON b.CategoryID=bil.CategoryID

WHERE @departure= c.City AND @arrival= d.City

END

EXEC searchbyCity @departure='Gojra' , @arrival ='kasur'

EXEC searchbyCity @departure='Narowal' , @arrival ='Timargara'

EXEC searchbyCity @departure='Timargara' , @arrival ='Bagh'

Table

Description automatically generated

/\*Procedure to calculate revenue and count parcel by branch code\*/

CREATE PROCEDURE branchrevenue

@branchCode varchar(100)

AS

BEGIN

SELECT br.BranchCode,SUM(bil.Price) AS Revenue ,COUNT(b.CategoryID) As TotalParcel FROM Branch br

INNER JOIN Billing b ON br.BranchCode=b.BranchCode

INNER JOIN Parcel p ON p.ParcelNumber=b.ParcelNumber

INNER JOIN BillingCategory bil ON bil.CategoryID=b.CategoryID

GROUP BY br.BranchCode

Having @branchCode = br.branchcode

END

EXEC branchrevenue 'HM01'

EXEC branchrevenue 'HM02'

Table

Description automatically generated

/\*Procedure to check nature of parcel\*/

CREATE PROCEDURE parcelNature

@nature VARCHAR(200)

AS

BEGIN

SELECT p.CustomerID,p.BranchCode,p.ParcelNumber,p.Weight\_Kg,pc.Nature FROM ParcelCategory pc

INNER JOIN Parcel p ON pc.CategoryID=p.CategoryID

WHERE @nature=pc.Nature

END

EXEC parcelNature 'shopper'

EXEC parcelNature 'package'

EXEC parcelNature 'box'

Table

Description automatically generated

/\*Procedure Calculate profit or loss of any branch\*/

CREATE PROCEDURE RevenueofBranch(

@branchn varchar(100)

)

AS

BEGIN

DECLARE @parcelsum INT,

@employee inT;

SET @parcelsum = '0';

SET @employee = '0';

SELECT

@parcelsum =

(SELECT SUM(bil.Price) FROM Branch br

INNER JOIN Billing b ON br.BranchCode=b.BranchCode

INNER JOIN Parcel p ON p.ParcelNumber=b.ParcelNumber

INNER JOIN BillingCategory bil ON bil.CategoryID=b.CategoryID

GROUP BY br.BranchCode

HAVING br.BranchCode=@branchn),

@employee=

(SELECT SUM(d.Salary) FROM Designation d

INNER JOIN Employee e ON d.DesignationRole=e.DesignationRole

INNER JOIN Branch br ON br.BranchCode=e.BranchCode

GROUP BY br.BranchCode

HAVING br.BranchCode=@branchn)

FROM

Branch

WHERE

@branchn=branch.branchCode

PRINT CONCAT('Employee Expenses:: ', @employee);

PRINT CONCAT('Parcel Revenue:: ',@parcelsum);

Print CONCAT('Profit:: ', @parcelsum-@employee);

END;

exec RevenueofBranch'HM90'

exec RevenueofBranch'HM01'

exec RevenueofBranch'HM80'

exec RevenueofBranch'HM90'

exec RevenueofBranch'HM12'

exec RevenueofBranch'HM08'

exec RevenueofBranch'HM23'

Graphical user interface, text, application

Description automatically generated

/\*Procedure to Update Status of Parcel Delivery\*/

Alter PROCEDURE UpdateOrderStatus

-- Add the parameters for the stored procedure here

@deliveryID INT,

@deliveryStatus INT

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

UPDATE DeliveryInfo SET DeliveryStatus=@deliveryStatus WHERE DeliveryID=@deliveryID

END

DECLARE @return\_value int

EXEC @return\_value = [dbo].[UpdateOrderStatus]

@deliveryID = 1122,

@deliveryStatus = 2

SELECT 'Return Value' = @return\_value

GO

Graphical user interface, text, application

Description automatically generated

/\*Procedure to show all customers by branch code\*/

CREATE PROCEDURE customerBYbranch

@branchCode varchar(100)

AS

BEGIN

SELECT c.CustomerID,c.Name,c.CNIC,c.Country,c.PhoneNumber,b.BranchCode FROM Branch b

INNER JOIN Parcel p ON b.BranchCode=p.BranchCode

INNER JOIN Customer c ON p.CustomerID=c.CustomerID

where @branchCode=b.BranchCode

END

EXEC customerBYbranch 'HM01'

Graphical user interface, table, Excel

Description automatically generated