SỬA KIỂU DỮ LIỆU AMOUNT VÀ OVERDUE\_FEE THÀNH DECIMAL(18)

SỬA KIỂU DỮ LIỆU BOOKING\_DATE VÀ RETURN\_DATE THÀNH DATETIME

LÀM CÁC FORM CUSTOMER

CHỈNH SỬA CODE UPDATE AMOUNT KHI QUÁ HẠN..

1. TẠO TRIGGER CUSTOMER

CREATE TRIGGER [Trigger\_Validate\_Input\_Customer]

ON [dbo].[Customers]

FOR INSERT, UPDATE

AS

DECLARE @pwd AS NVARCHAR(30), @usm AS NVARCHAR(30)

SELECT @pwd = i.password, @usm = i.username

FROM INSERTED i

IF (LEN(@pwd) <= 6 OR LEN(@usm) <= 6)

BEGIN

SET NOCOUNT ON

RAISERROR ('Your input data is wrong by default structure' ,16,1)

PRINT('Username or Password of Customer must be more than 6 characters')

ROLLBACK TRANSACTION;

END

1. TẠO FUNCTION CUSTOMER

CREATE FUNCTION Function\_LOGIN

(

@username NVARCHAR(30),

@password NVARCHAR(30)

)

RETURNS VARCHAR(20)

AS

BEGIN

DECLARE @match\_count INTEGER, @result VARCHAR(20)

SET @match\_count = (SELECT COUNT(\*)

FROM Customers

WHERE Customers.username = @username

AND Customers.password = @password)

IF @match\_count = 0

BEGIN

SET @result = 'Wrong username or password';

END

ELSE

BEGIN

IF @match\_count = 1

BEGIN

SET @result = 'Login successfull';

END

ELSE

BEGIN

SET @result = 'Too many matches, this should never happen before!';

END

END;

return @result

END;

1. TẠO PROCEDURE

CREATE PROCEDURE [dbo].[addNewCustomer]

@cus\_id INT,

@fname NVARCHAR(20),

@lname NVARCHAR(20),

@photo IMAGE,

@phone NVARCHAR(50),

@username NVARCHAR(30),

@passwd NVARCHAR(30)

AS

BEGIN

SET NOCOUNT ON

INSERT INTO dbo.Customers

(

firstname,

lastname,

photo,

phone,

username,

password

)

VALUES

(

@fname,

@lname,

@photo,

@phone,

@username,

@passwd

)

END

1. FUNCTION count customer

CREATE FUNCTION Function\_countCustomers()

RETURNS INTEGER

AS

BEGIN

DECLARE @count INTEGER

SET @count = (Select Count(\*) From Customers)

RETURN @count

END;.

1. FUNCTION get customer by id

CREATE FUNCTION [dbo].[Function\_getCustomerbyID]

(

@id int

)

RETURNS @returnTable TABLE

(

customer\_id INT,

firstname NVARCHAR(20),

lastname NVARCHAR(20),

photo IMAGE,

phone NVARCHAR(50),

username NVARCHAR(30),

password NVARCHAR(30)

)

AS

BEGIN

INSERT @returnTable

SELECT \* FROM dbo.Customers WHERE dbo.Customers.customer\_id = @id

RETURN

END

1. FUNCTION kiểm tra username đã tồn tại

CREATE FUNCTION [dbo].[Function\_checkUsernameExisted]

(

@username NVARCHAR(30)

)

RETURNS INTEGER

AS

BEGIN

DECLARE @result INTEGER

SET @result = (SELECT COUNT(\*) FROM Customers WHERE dbo.Customers.username = @username)

RETURN @result

END

1. FUNCTION getAllRooms

CREATE FUNCTION [dbo].[Function\_getAllRooms] ()

RETURNS @returnTable TABLE

(

room\_id INT,

type\_name NVARCHAR(30),

room\_capacity INT,

room\_bed INT,

room\_price DECIMAL(18),

status\_name NVARCHAR(20)

)

AS

BEGIN

INSERT @returnTable

SELECT Rooms.room\_id, RoomTypes.type\_name, RoomTypes.room\_capacity, RoomTypes.room\_bed, RoomTypes.room\_price

, RoomStatus.status\_name

FROM Rooms, RoomTypes, RoomStatus

WHERE Rooms.room\_type = RoomTypes.room\_type\_id

AND Rooms.room\_status = RoomStatus.status\_id

RETURN

END

1. FUNCTION getRoomType

CREATE FUNCTION [dbo].[Function\_getRoomType] ()

RETURNS @returnTable TABLE

(

type\_name NVARCHAR(30),

room\_type\_id INT

)

AS

BEGIN

INSERT @returnTable

SELECT RoomTypes.type\_name, RoomTypes.room\_type\_id

FROM RoomTypes

RETURN

END

1. Function\_getAllRooms\_byTypes

CREATE FUNCTION [dbo].[Function\_getAllRooms\_byTypes]

(

@room\_type INT

)

RETURNS @returnTable TABLE

(

room\_id INT,

type\_name NVARCHAR(30),

room\_capacity INT,

room\_bed INT,

room\_price DECIMAL(18),

status\_name NVARCHAR(20)

)

AS

BEGIN

INSERT @returnTable

SELECT Rooms.room\_id, RoomTypes.type\_name, RoomTypes.room\_capacity, RoomTypes.room\_bed, RoomTypes.room\_price

, RoomStatus.status\_name

FROM Rooms, RoomTypes, RoomStatus

WHERE Rooms.room\_type = RoomTypes.room\_type\_id

AND Rooms.room\_status = RoomStatus.status\_id

AND Rooms.room\_type = @room\_type

RETURN

END

1. Function\_countBooking

CREATE FUNCTION Function\_countBooking()

RETURNS INTEGER

AS

BEGIN

DECLARE @count INTEGER

SET @count = (Select Count(\*) From Booking)

RETURN @count

END;

CREATE PROCEDURE [dbo].[createBooking]

@customer\_id INT,

@room\_id INT,

@booking\_date DATE,

@return\_date DATE,

@amount DECIMAL(18),

@overdue\_fee DECIMAL(18)

AS

BEGIN

SET NOCOUNT ON

INSERT INTO dbo.Booking

(

customer\_id,

room\_id,

booking\_date,

return\_date,

amount,

overdue\_fee

)

VALUES

(

@customer\_id,

@room\_id,

@booking\_date,

@return\_date,

@amount,

@overdue\_fee

)

END

CREATE FUNCTION Function\_getPriceRoom\_byTypeId(

@room\_type\_id INT

)

RETURNS DECIMAL(18)

AS

BEGIN

DECLARE @price DECIMAL(18)

SET @price = (SELECT RoomTypes.room\_price FROM RoomTypes

WHERE RoomTypes.room\_type\_id = @room\_type\_id)

RETURN @price

END;

CREATE FUNCTION Function\_getBooking

(

@customer\_id INT

)

RETURNS @returnTable TABLE

(

booking\_id INT,

room\_id INT,

booking\_date DATE,

return\_date DATE,

checkin DATETIME,

checkout DATETIME

)

AS

BEGIN

INSERT @returnTable

SELECT booking\_id, room\_id, booking\_date, return\_date, checkin, checkout

FROM Booking

WHERE customer\_id = @customer\_id

AND checkout IS NULL

RETURN

END

CREATE PROCEDURE [dbo].[Procedure\_updateStatusofRoom]

@room\_id INT,

@room\_status BIT

AS

BEGIN

UPDATE dbo.Rooms

SET room\_status = @room\_status

WHERE room\_id = @room\_id

END

CREATE FUNCTION [dbo].[Function\_getAllHistoryBooking\_byID]

(

@customer\_id int

)

RETURNS @returnTable TABLE

(

booking\_id INT,

room\_id INT,

booking\_date DATE,

return\_date DATE,

checkin DATETIME,

checkout DATETIME,

room\_price DECIMAL(18)

)

AS

BEGIN

INSERT @returnTable

SELECT Booking.booking\_id, Booking.room\_id, Booking.booking\_date, Booking.return\_date, Booking.checkin,

Booking.checkout, RoomTypes.room\_price

FROM dbo.Booking, dbo.Rooms, dbo.RoomTypes

WHERE dbo.Booking.room\_id = Rooms.room\_id

AND dbo.Rooms.room\_type = RoomTypes.room\_type\_id

AND dbo.Booking.customer\_id = @customer\_id

RETURN

END

CREATE PROCEDURE [dbo].[Procedure\_updateOverdue\_fee]

@booking\_id INT,

@overdue\_fee DECIMAL(18)

AS

BEGIN

UPDATE dbo.Booking

SET overdue\_fee = @overdue\_fee

WHERE booking\_id = @booking\_id

END

CREATE FUNCTION Function\_getRoomtype\_byRoomID

(

@room\_id INT

)

RETURNS DECIMAL(18)

AS

BEGIN

DECLARE @result INT

SET @result = (SELECT room\_type FROM Rooms

WHERE room\_id = @room\_id)

RETURN @result

END;

CREATE PROCEDURE [dbo].[Procedure\_updateAmount]

@booking\_id INT,

@overdue\_fee DECIMAL(18)

AS

BEGIN

UPDATE dbo.Booking

SET amount = amount + @overdue\_fee

WHERE booking\_id = @booking\_id

END

CREATE FUNCTION [dbo].[Function\_getInfoCustomerByRoomID]

(

@room\_id int

)

RETURNS @returnTable TABLE

(

customer\_id INT,

firstname NVARCHAR(20),

lastname NVARCHAR(20),

photo IMAGE,

phone NVARCHAR(50),

username NVARCHAR(30),

password NVARCHAR(30),

booking\_date DATETIME,

return\_date DATETIME

)

AS

BEGIN

INSERT @returnTable

SELECT Customers.customer\_id, Customers.firstname, Customers.lastname, Customers.photo, Customers.phone,

Customers.username, Customers.password, booking\_date, return\_date

FROM Customers, (

SELECT DISTINCT customer\_id, booking\_date, return\_date

FROM dbo.Booking

WHERE room\_id = @room\_id

AND checkout IS NULL

) H

WHERE H.customer\_id = Customers.customer\_id

RETURN

END