

CREATION OF TABLES AND INSERTING VALUES INTO IT

Creation and Insertion of Values into TheatreDetails

```
Create table TheatreDetails(  
    TheatreId int IDENTITY(1,1) NOT NULL,  
    TheatreName varchar(50) NOT NULL,  
    Location varchar(50) NOT NULL  
)  
  
INSERT INTO TheatreDetails(TheatreName,Location) Values('PVR','Pune'),('Inox','Delhi');  
  
Select * from TheatreDetails  
  
Alter TABLE TheatreDetails ADD CONSTRAINT pk_TheatreDetails_TheatreId Primary  
KEY(TheatreId)
```

Creation and Insertion of Values into Users

```
Create table Users(  
    UserId VARCHAR(50) constraint pk_Users_UsersId Primary key,  
    UserName VARCHAR(50) NOT NULL,  
    Password VARCHAR(50) NOT NULL,  
    Age INT NOT NULL,  
    Gender CHAR(1) not null constraint chk_Users_Gender CHECK(Gender IN ('M','F')),  
    EmailId varchar(50) NOT NULL UNIQUE,  
    PhoneNumber NUMERIC(10) NOT NULL  
        constraint unq_Users_PhoneNumber unique,  
        constraint chk_Users_PhoneNumber CHECK(PhoneNumber LIKE '[6-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
)  
  
INSERT INTO Users(UserId,UserName,Password,Age,Gender,EmailId,PhoneNumber)  
Values('mary_potter','Mary Potter','Mary@123',25,'F','mary_p@gmail.com',9876543210),  
('jack_sparrow','Jack Sparrow','Spar78!jack',28,'M','jack_spa@yahoo.com',7876543210);
```

Creation and Insertion of Values into ShowDetails

```
CREATE TABLE ShowDetails(  
    ShowId int IDENTITY(1001,1) Primary Key,  
    TheatreId INT constraint fk_TheatreDetails_TheatreId_TheatreId foreign Key  
references TheatreDetails(TheatreId),  
    ShowDate Date NOT NULL,  
    ShowTime Time NOT NULL ,  
    MovieName VARCHAR(50) NOT NULL,
```

```

        TicketCost Decimal(6,2) NOT NULL,
        TicketAvailable INT NOT NULL
    )

INSERT INTO ShowDetails(TheatreId,ShowDate,ShowTime,MovieName,TicketCost,TicketAvailable)
VALUES(2,'2018-05-28','14:30','Avengers',250.00,100),
(2,'2018-05-30','17:30','Hit Man',200.00,150);

```

Creation and Insertion of Values into BookingDetails

```

Create table BookingDetails(
    BookingId VARCHAR(5) PRIMARY KEY,
    UserId VARCHAR(50) NOT NULL constraint fk_Users_UserId_UserId FOREIGN KEY
REFERENCES Users(UserId),
    ShowID INT NOT NULL constraint fk_ShowDetails_ShowId_ShowId FOREIGN KEY REFERENCES
ShowDetails(ShowId),
    NoOfTickets INT NOT NULL,
    TotalAmt DECIMAL(6,2) NOT NULL
)

INSERT INTO BookingDetails(BookingId,UserId,ShowID,NoOfTickets,TotalAmt)
VALUES('B1001','jack_sparrow',1001,2,500.00),
('B1002','mary_potter',1002,5,1000.00)

```

---Database Before Execution of Stored Procedure and Function

```

select * from TheatreDetails
Select * from ShowDetails
Select * from Users
Select * from BookingDetails

```

	TheatreId	TheatreName	Location
1	1	PVR	Pune
2	2	Inox	Delhi

	ShowId	TheatreId	ShowDate	ShowTime	MovieName	TicketCost	TicketAvailable
1	1001	2	2018-05-28	14:30:00.0000000	Avengers	250.00	95
2	1002	2	2018-05-30	17:30:00.0000000	Hit Man	200.00	140

	UserId	UserName	Password	Age	Gender	EmailId	PhoneNumber
1	jack_sparrow	Jack Sparrow	Spar78!jack	28	M	jack_spa@yahoo.com	7876543210
2	mary_potter	Mary Potter	Mary@123	25	F	mary_p@gmail.com	9876543210

	BookingId	UserId	ShowID	NoOfTickets	TotalAmt
1	B1001	jack_sparrow	1001	2	500.00
2	B1002	mary_potter	1002	5	1000.00
3	B1003	mary_potter	1001	5	1250.00
4	B1004	mary_potter	1002	10	2000.00

---Creating the Stored Procedure

```
GO
Alter PROC usp_BookTheTicket(@UserId VARCHAR(50),@ShowID INT,@NoOfTickets INT)
as
if(not exists(SELECT 'a' from Users WHERE UserId = @UserId ))
begin
return -1
end
else if(not exists(SELECT 'a' from ShowDetails WHERE ShowId = @ShowID ))
begin
return -2
end
else if(@NoOfTickets < 0)
begin
return -3
end
else if(@NoOfTickets > (SELECT TicketAvailable from ShowDetails WHERE ShowID = @ShowID ))
begin
return -4
end
else
begin
DECLARE @PreFix VARCHAR(1) = 'B';
DECLARE @NewId VARCHAR(5);
```

```

        DECLARE @Name VARCHAR(50);
        DECLARE @Id INT;
        DECLARE @Total Decimal(6,2) ,@AvailableTicket INT;
        Select @Total= TicketCost, @AvailableTicket=TicketAvailable from ShowDetails
where ShowId=@ShowID
        SELECT @Name = UserName FROM Users where UserId= @UserId
        SELECT @Id = COUNT(*)+1001 FROM BookingDetails
        SELECT @NewId = @PreFix + CAST(@Id AS VARCHAR(4))
        Set @AvailableTicket = @AvailableTicket - @NoOfTickets
        Set @Total = @Total * @NoOfTickets
        INSERT INTO BookingDetails(BookingId,UserId,ShowID,NoOfTickets,TotalAmt)
Values(@NewId,@UserId,@ShowID,@NoOfTickets,@Total)
        UPDATE ShowDetails Set TicketAvailable = @AvailableTicket where ShowId= @ShowID
        return 1
    end

```

---Executing the stored Procedure

```

Declare @Ret int
Execute @Ret = usp_BookTheTicket 'mary_potter',1002,10
Print @Ret

```

---Database after Execution of Stored Procedure

	TheatreId	TheatreName	Location
1	1	PVR	Pune
2	2	Inox	Delhi

	ShowId	TheatreId	ShowDate	ShowTime	MovieName	TicketCost	TicketAvailable
1	1001	2	2018-05-28	14:30:00.0000000	Avengers	250.00	95
2	1002	2	2018-05-30	17:30:00.0000000	Hit Man	200.00	130

	UserId	UserName	Password	Age	Gender	EmailId	PhoneNumber
1	jack_sparrow	Jack Sparrow	Spar78	28	M	jack_spa@yahoo.com	7876543210
2	mary_potter	Mary Potter	Mary@123	25	F	mary_p@gmail.com	9876543210

	BookingId	UserId	ShowID	NoOfTickets	TotalAmt
1	B1001	jack_sparrow	1001	2	500.00
2	B1002	mary_potter	1002	5	1000.00
3	B1003	mary_potter	1001	5	1250.00
4	B1004	mary_potter	1002	10	2000.00
5	B1005	mary_potter	1002	10	2000.00

--- Creation of Function

```

go
Alter Function ufn_GetMovieShowtimes(@MovieName varchar(50), @Location varchar(50))
returns table
as
return
Select s.MovieName ,s.ShowDate,s.ShowTime,s.TicketCost,t.Location from ShowDetails s JOIN
TheatreDetails t ON s.TheatreId = t.TheatreId where t.Location=@Location and s.MovieName
=@MovieName
go

```

--Executing the Function

```

Select * from ufn_GetMovieShowtimes('Avengers','Delhi');

```

--Output for the execution of the function

	MovieName	ShowDate	ShowTime	TicketCost	Location
1	Avengers	2018-05-28	14:30:00.0000000	250.00	Delhi