**Khoa Công Nghệ Thông Tin**

--o0o--



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**CƠ SỞ DỮ LIỆU**

Sinh viên thực hiện : Ngô Văn Luân

MSSV : 119160

GVHD : Nguyễn Đình Anh

Hà Nội, 2020

[**Phần 1: Mô tả bài toán 3**](#_Toc535958742)

[**Phần 2: Sơ đồ cơ sở dữ liệu 4**](#_Toc535958743)

[**Phần 3: Mô tả cơ sở dữ liệu 5**](#_Toc535958744)

[**Phần 4: Thiết kế cơ sở dữ liệu bằng SQL Server. 9**](#_Toc535958745)

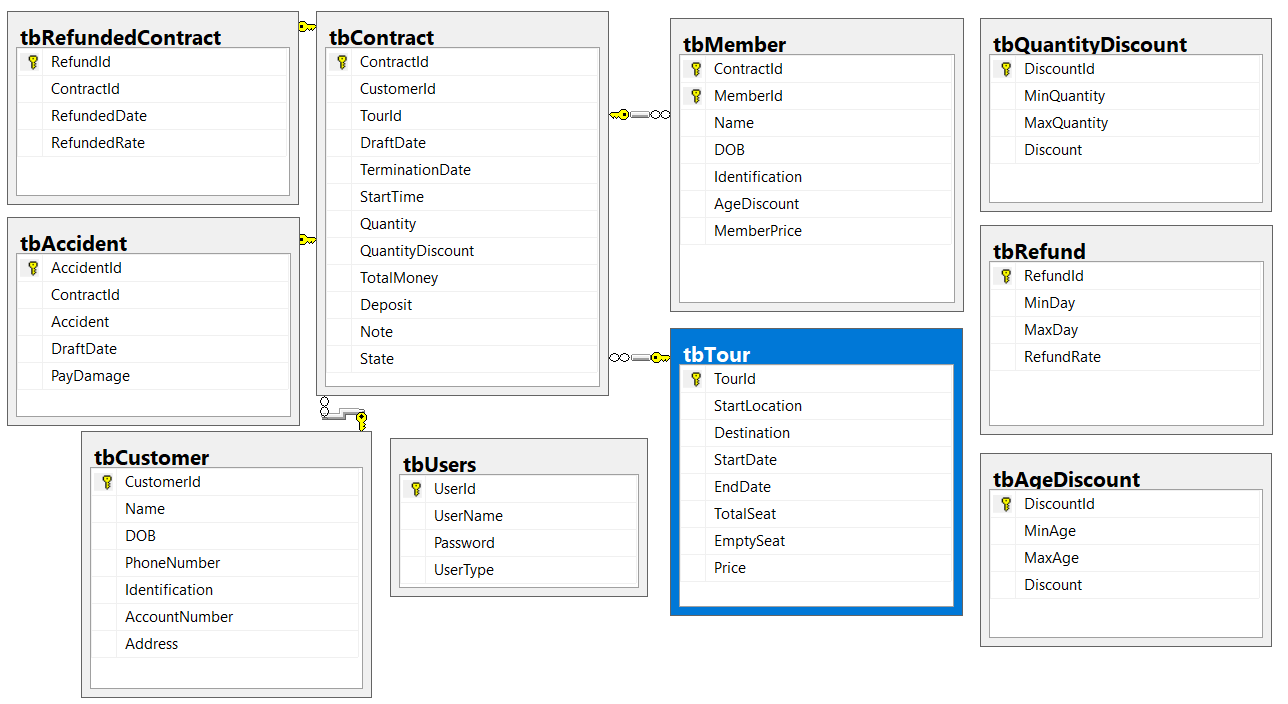
[**Phần 5: Các truy vấn giải quyết vấn đề nghiệp vụ 13**](#_Toc535958742)

1. **Mô tả bài toán.**

Hiện nay du lịch ngày càng phát triển, là một hình thức giải trí rất được quan tâm. Vì vậy, bài toán được đưa ra nhằm hỗ trợ việc đăng ký tour dễ dàng, chính xác hơn. Mô tả bài toán như sau:

* Hỗ trợ việc quản lý thông tin tour, thông tin chiết khấu theo số lượng, thông tin chiết khấu theo độ tuổi, thông tin hoàn cọc bao gồm thêm, sửa, xoá. Việc quản lý sẽ được phân quyền chỉ cho nhân viên quản lý được sử dụng, còn nhân viên thông thường sẽ không được sử dụng.
* Một tour gồm các thông tin: mã tour, địa điểm đi, địa điểm đến, thời gian bắt đầu, thời gian kết thúc, giá tour/người, tổng số lượng, số chỗ còn trống.
* Chiết khấu theo số lượng bao gồm mã chiết khấu, số lượng đầu, số lượng cuối, số chiết khấu.
* Chiết khấu theo đô tuổi bao gồm mã chiết khấu, số tuối đầu, số tuổi cuối, số chiết khấu.
* Thông tin hoàn cọc bao gồm mã hoàn cọc, số ngày bắt đầu, số ngày kết thúc, số phần trăm hoàn cọc.
* Khi khách hàng đến đăng ký tour, nhân viên sẽ giúp tìm kiếm tour khách hàng yêu cầu. Nếu tìm được sẽ tiến đến lập hợp đồng, ngược lại nhân viên có thể gợi ý khách hàng nhưng tour tương tự. Trước khi lập hợp đồng, phải lưu trữ thông tin khách hàng bao gồm: mã khách hàng, tên khách hàng, ngày sinh, số chứng minh thư, số điện thoại, số tài khoản, địa chỉ. Một hợp đồng bao gồm mã hợp đồng, mã khách hàng, mã tour, ngày lập, ngày thanh lý, thời gian đón đón khách, số lượng, chiếu khấu theo số lượng, tổng giá trị hợp đồng, đặt cọc, ghi chú với điều kiện số lượng phải không lớn hơn số lượng trống của tour. Để xác định tổng giá trị hợp đồng phải dựa trên một danh sách thành viên của hợp đồng và chiết khấu theo hợp đồng. Mỗi thành viên sẽ bao gồm các thông tin: mã thành viên, tên thành viên, ngày tháng năm sinh, số chứng minh thư, chiết khấu theo tuổi, giá thành viên. Tổng giá trị hợp đồng sẽ bằng tổng giá trị của các thành viên trừ đi lượng được chiết khấu. Khi đăng ký xong số lượng còn trống của tour sẽ trừ đi số lượng trong hợp đồng.
* Khi khách hàng muốn thay đổi hoặc huỷ hợp, nếu huỷ hợp đồng trong thời gian quy định sẽ được hoàn trả số tiền đặt cọc, số chỗ trong hợp đồng sẽ hoàn lại số chỗ còn trống của tour. Nếu chỉnh sửa hợp đồng như tăng giảm số lượng thì số chỗ còn trống trong tour cũng phải thay đổi, đồng thời danh sách thành viên cũng sẽ thêm hoặc bớt thành viên theo số lượng trong hợp đồng, nếu thay đổi thông tin thành viên thì sẽ tự động cập nhật lại giá thành viên và tổng giá trị hợp đồng.
* Khi kết thúc tour, khách hàng cần đến để thanh lý hợp đồng và thanh toán số tiền còn lại. Nếu trong khi đi tour có xảy ra sự cố gì, công ty sẽ xem xét sự cố thuộc lỗi của bên nào. Nếu lỗi của công ty sẽ lưu trữ lại: mã sự cố, tên sự cố, ngày xảy ra sự cố, ngày lập biên bản và bồi thường cho khách hàng.

1. **Sơ đồ cơ sở dữ liệu.**



1. **Mô tả cơ sở dữ liệu.**

* Bảng tbTour:

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| TourId | Khoá chính, AUTONUMBER |
| StartLocation | NOT NULL |
| Destination | NOT NULL |
| StartDate | NOT NULL, > ngày hiện tại |
| EndDate | NOT NULL, > StartDate |
| TotalSeat | NOT NULL, > 0 |
| EmptySeat | NOT NULL, <= TotalSeat |
| Price | NOT NULL |

* Bảng tbCustomer

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| CustomerId | Khoá chính, AUTONUMBER |
| Name | NOT NULL |
| DOB | NOT NULL, tuổi >=15 |
| PhoneNumber | NOT NULL |
| Identification | NOT NULL |
| AccountNumber | NOT NULL |
| Address |  |

* Bảng tbContract

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| ContractId | Khoá chính, AUTONUMBER |
| CustomerId | Khoá ngoại |
| TourId | Khoá ngoại |
| DraftDate | NOT NULL, DEFAULT = ngày hiện tại |
| TerminationDate | NOT NULL, > DraftDate |
| StartTime | NOT NULL, > thời gian hiện tại |
| Quantity | NOT NULL, >= 0 |
| QuantityDiscount | NOT NULL, >= 0 |
| TotalMoney | DEFAULT = 0, >= 0 |
| Deposit | DEFAULT = 0 |
| Note |  |
| State | DEFAULT = 1 |

* Bảng tbMenber

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| ContractId | Khóa ngoại |
| MemberId | Khoá chính |
| Name | NOT NULL |
| DOB | NOT NULL, < ngày hiện tại |
| Identification |  |
| AgeDiscount | DEFAULT = 0, >= 0 |
| MemberPrice | DEFAULT = 0, >= 0 |

* tbAccident

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| AccdentId | Khoá chính, AUTONUMBER |
| ContracId | Khoá ngoại |
| Accident | NOT NULL |
| DraftDate | NOT NULL, DEFAULT ngày hiện tại |
| PayDamage | DEFAULT = 0, >= 0 |

* tbRefundContract

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| RefundId | Khoá chính, AUTONUMBER |
| ContractId | Khoá ngoại |
| RefundedDate | NOT NULL, ngày hiện tại |
| RefundedRate |  |

* tbQuantityDiscount

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| DiscountId | Khoá chính, AUTONUMBER |
| MinQuantity | NOT NULL, >= 0 |
| MaxQuantity | NOT NULL, >= MinQuantity |
| Discount | NOT NULL, <=100 |

* tbAgeDiscount

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| DiscountId | Khoá chính, AUTONUMBER |
| MinAge | NOT NULL, >= 0 |
| MaxAge | NOT NULL, >= MinAge |
| Discount | NOT NULL, <=100 |

* tbRefund

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| RefundId | Khoá chính, AUTONUMBER |
| MinDay | NOT NULL, >= 0 |
| MaxDay | NOT NULL, >= MinDay |
| RefundRate | NOT NULL, <=100 |

* tbUsers

|  |  |
| --- | --- |
| **Thuộc tính** | **Ràng buộc** |
| UserId | Khoá chính, AUTONUMBER |
| UserName | NOT NULL, UNIQUE |
| Password | NOT NULL |
| UserType |  |

1. **Thiết kế cơ sở dữ liệu bằng SQL Server.**

* tbTour

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| TourId | int |
| StartLocation | nvarchar(30) |
| Destination | nvarchar(30) |
| StartDate | date |
| EndDate | date |
| TotalSeat | int |
| EmptySeat | int |
| Price | int |

* tbCustomer

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| CustomerId | int |
| Name | nvarchar(50) |
| DOB | date |
| PhoneNumber | nvarchar(10) |
| Identification | nvarchar(20) |
| AccountNumber | nvarchar(20) |
| Address | nvarchar(50) |

* tbContract

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| ContractId | int |
| CustomerId | int |
| TourId | int |
| DraftDate | date |
| TerminationDate | date |
| StartTime | date |
| Quantity | int |
| QuantityDiscount | int |
| TotalMoney | int |
| Deposit | int |
| Note | nvarchar(1000) |

* tbMember

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| ContractId | int |
| MemberId | int |
| Name | nvarchar(50) |
| DOB | date |
| Identification | nvarchar(20) |
| AgeDiscount | int |
| MemberPrice | int |

* tbAccident

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| AccidentId | int |
| ContractId | int |
| Accident | nvarchar(100) |
| DraftDate | date |
| PayDamage | int |

* tbRefundedContract

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| RefundId | int |
| ContractId | int |
| RefundedDate | date |
| RefundedRate | int |

* tbQuantityDiscount

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| DiscountId | int |
| MinQuantity | int |
| MaxQuantity | int |
| Discount | int |

* tbAgeDisCount

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| DiscountId | int |
| MinAge | int |
| MaxAge | int |
| Discount | int |

* tbRefund

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| RefundId | int |
| MinDay | int |
| MaxDay | int |
| RefundRate | int |

* tbUsers

|  |  |
| --- | --- |
| **Tên cột** | **Kiểu dữ liệu** |
| UserId | int |
| UserName | nvarchar(50) |
| Password | nvarchar(50) |
| UserType | int |

1. **Các truy vấn giải quyết vấn đề nghiệp vụ.**

Ngoài sử dụng những câu truy vấn thông thường như SELECT, INSERT, UPDATE, DELETE. Em nghiên cứu và áp dụng các kỹ thuật TRIGGER, STORE PROCEDURE để giải quyết nghiệp vụ bài toán dễ dàng hơn.

* Các store procedure sử dụng

|  |  |
| --- | --- |
| **Chức năng** | **Code** |
| Tính chiết khấu hợp đồng: Đầu vào là số lượng thành viên của hợp đồng sẽ tính ra được chiết khấu theo số lượng của hợp đồng. | CREATE PROCEDURE spGetQuantityDiscout  @quantity INT =NULL  AS  DECLARE @quantityDiscount INT  IF @quantity IS NULL  SET @quantityDiscount = 0  ELSE  BEGIN  SELECT @quantityDiscount = Discount  FROM dbo.tbQuantityDiscount  WHERE @quantity >= MinQuantity AND @quantity <= MaxQuantity  IF @quantityDiscount IS NULL  SET @quantityDiscount = 0  END  RETURN @quantityDiscount |
| Tính tuổi: Tính tuối theo một ngày cụ thể truyền vào. | CREATE PROCEDURE spCalculateAge  @dob DATE  AS  DECLARE @age INT, @year INT, @month INT, @day INT, @currentYear INT, @currentMonth INT, @currentDay INT  SET @year = YEAR(@dob)  SET @month = MONTH(@dob)  SET @day = DAY(@dob)  SET @currentYear = YEAR(GETDATE())  SET @currentMonth = MONTH(GETDATE())  SET @currentDay = DAY(GETDATE())  IF @currentYear = @year  SET @age = 0  ELSE  BEGIN  SET @age = @currentYear - @year - 1  IF (@currentMonth > @month) OR (@currentMonth = @month AND @currentDay >= @day)  BEGIN  SET @age = @age + 1  END  END  RETURN @age |
| Tính chiết khấu theo tuổi: Tính chiếu khấu theo tuổi cho từng thành viên của hợp đồng với tuổi cho trước. | CREATE PROCEDURE spGetAgeDiscount  @DOB DATE = NULL  AS  DECLARE @ageDiscount INT  IF @DOB IS NULL  SET @ageDiscount = 0  ELSE  BEGIN  DECLARE @age INT  EXEC @age = dbo.spCalculateAge @dob = @DOB -- date  SELECT @ageDiscount = Discount  FROM dbo.tbAgeDiscount  WHERE MinAge <= @age AND @age <= MaxAge  IF @ageDiscount IS NULL  SET @ageDiscount = 0  END  RETURN @ageDiscount |
| Tính tị lệ hoàn cọc: Khi khách hàng huỷ hợp đồng, dựa vào số ngày từ ngày huỷ đến ngày hợp đồng có hiệu lực sẽ tính được tỉ lệ hoàn cọc cho khách hàng. | CREATE PROCEDURE spGetRefundRate  @contractId INT  AS  DECLARE @dayNumbers INT, @startDate DATE, @refundRate INT  SELECT @startDate = StartDate  FROM dbo.tbContract INNER JOIN dbo.tbTour  ON tbTour.TourId = tbContract.TourId  WHERE ContractId = @contractId  SET @dayNumbers = DATEDIFF(DAY, GETDATE(), @startDate)  SELECT @refundRate = RefundRate  FROM dbo.tbRefund  WHERE MinDay <= @dayNumbers AND MaxDay >= @dayNumbers  IF @refundRate IS NULL  SET @refundRate = 0  RETURN @refundRate |

* Các trigger dùng để kiểm tra khi thêm hoặc thay đổi chính sách của công ty.

|  |  |
| --- | --- |
| **Vấn đề** | **Code** |
| Khi thêm mới hoặc thay đổi thông tin chiết khấu về số lượng thì thông tin mới không được trùng với các thông tin đã có. | CREATE TRIGGER trgAddUpdateQuantityDiscount  ON dbo.tbQuantityDiscount  FOR INSERT, UPDATE  AS  BEGIN  DECLARE @discountId INT, @minQuantity INT, @maxQuantity INT, @duplicatedMin INT, @duplicatedMax INT  SELECT @discountId = Inserted.DiscountId, @minQuantity = Inserted.MinQuantity, @maxQuantity = Inserted.MaxQuantity  FROM Inserted  SELECT @duplicatedMin = COUNT(\*)  FROM dbo.tbQuantityDiscount  WHERE @minQuantity >= MinQuantity AND @minQuantity <= MaxQuantity AND @discountId != DiscountId  SELECT @duplicatedMax = COUNT(\*)  FROM dbo.tbQuantityDiscount  WHERE @maxQuantity >= MinQuantity AND @maxQuantity <= MaxQuantity AND @discountId != DiscountId  IF @duplicatedMin > 0 OR @duplicatedMax > 0  ROLLBACK TRAN  END |
| Khi thêm mới hoặc thay đổi thông tin chiết khấu về độ tuổi thì thông tin mới không được trùng với các thông tin đã có. | CREATE TRIGGER trgAddUpdateAgeDiscount  ON dbo.tbAgeDiscount  FOR INSERT, UPDATE  AS  BEGIN  DECLARE @discountId INT, @minAge INT, @maxAge INT, @duplicatedMin INT, @duplicatedMax INT  SELECT @discountId = Inserted.DiscountId, @minAge = Inserted.MinAge, @maxAge = Inserted.MaxAge  FROM Inserted  SELECT @duplicatedMin = COUNT(\*)  FROM dbo.tbAgeDiscount  WHERE @minAge >= MinAge AND @minAge <= MaxAge AND @discountId != DiscountId  SELECT @duplicatedMax = COUNT(\*)  FROM dbo.tbAgeDiscount  WHERE @maxAge >= MinAge AND @maxAge <= MaxAge AND @discountId != DiscountId  IF @duplicatedMin > 0 OR @duplicatedMax > 0  ROLLBACK TRAN  END |
| Khi thêm mới hoặc thay đổi thông tin về tỉ lệ hoàn cọc thì thông tin mới không được trùng với các thông tin đã có. | CREATE TRIGGER trgAddUpdateRefund  ON dbo.tbRefund  FOR INSERT, UPDATE  AS  BEGIN  DECLARE @refundId INT, @minDay INT, @maxDay INT, @duplicatedMin INT, @duplicatedMax INT  SELECT @refundId = Inserted.RefundId, @minDay = Inserted.MinDay, @maxDay = Inserted.MaxDay  FROM Inserted  SELECT @duplicatedMin = COUNT(\*)  FROM dbo.tbRefund  WHERE @minDay >= MinDay AND @minDay <= MaxDay AND @refundId != RefundId  SELECT @duplicatedMax = COUNT(\*)  FROM dbo.tbRefund  WHERE @maxDay >= MinDay AND @maxDay <= MaxDay AND @refundId != RefundId  IF @duplicatedMin > 0 OR @duplicatedMax > 0  ROLLBACK TRAN  END |
|  |  |

* Quản lý tour.

|  |  |
| --- | --- |
| **Vấn đề** | **Kỹ thuật sử dụng** |
| Thêm mới tour: khi thêm tour mới thì tổng số chỗ phải bằng số lượng chỗ trống. Nếu khác thì không cho INSERT vào bảng. | * Sử dụng trigger, tự động gọi khi thêm mới một tour. |
| * Code:   CREATE TRIGGER trgAddTour  ON dbo.tbTour  FOR INSERT  AS  BEGIN  DECLARE @totalSeat INT, @emptySeat INT  SELECT @totalSeat = Inserted.TotalSeat, @emptySeat = Inserted.EmptySeat  FROM inserted  IF @totalSeat != @emptySeat  ROLLBACK TRAN  END | |
| Xoá tour: Khi xoá một tour bất kỳ thì sẽ xoá toàn bộ hợp đồng cũng nhưng những thông tin liên quan đến hợp đồng mà có đăng ký tour đó. | * Sử dụng trigger, tự động gọi khi xoá tour. |
| * Code:   CREATE TRIGGER trgDeleteTour  ON dbo.tbTour  INSTEAD OF DELETE  AS  BEGIN  DECLARE @tourId int  SELECT @tourId = Deleted.TourId FROM Deleted  DELETE FROM dbo.tbContract WHERE TourId = @tourId  DELETE FROM dbo.tbTour WHERE TourId = @tourId  END | |

* Quản lý khách hàng

|  |  |
| --- | --- |
| **Vấn đề** | **Kỹ thuật sử dụng** |
| Xoá khách hàng: Khi xoá một khách hàng bất kỳ thì sẽ xoá toàn bộ hợp đồng cũng nhưng những thông tin liên quan đến hợp đồng mà có đăng ký bởi khách hàng đó đó. | * Sử dụng trigger, tự động gọi khi xoá khách hàng. |
| * Code:   CREATE TRIGGER trgDeleteCustomer  ON dbo.tbCustomer  INSTEAD OF DELETE  AS  BEGIN  DECLARE @customerId INT  SELECT @customerId = Deleted.CustomerId FROM Deleted  DELETE dbo.tbContract WHERE CustomerId = @customerId  DELETE dbo.tbCustomer WHERE CustomerId = @customerId  END | |

* Xử lý nghiệp vụ hợp đồng.

|  |  |
| --- | --- |
| **Vấn đề** | **Kỹ thuật sử dụng** |
| Thêm hợp đồng: Nếu số lượng thành viên nhiều hơn số chỗ trống của tour thì hợp đồng lập không thành công. Nếu thành công thì sẽ tính chiết khấu theo số lượng, xác định ngày thanh lý của hợp đồng vào cập nhật lại số chỗ chỗ trống của tour. | * Sử dụng trigger, tự động gọi khi thêm hợp đồng và store procedure để tính chiết khấu theo số lượng. |
| * Code:   CREATE TRIGGER trgAddContract  ON dbo.tbContract  FOR INSERT  AS  BEGIN  DECLARE @quantity INT, @tourID INT, @contractID INT, @endDate DATE, @startDate DATE  SELECT @tourID = Inserted.TourID, @quantity = Inserted.Quantity, @contractID = Inserted.ContractID, @endDate = dbo.tbTour.EndDate, @startDate = dbo.tbTour.StartDate  FROM Inserted INNER JOIN dbo.tbTour ON tbTour.TourID = Inserted.TourID  DECLARE @emptySeat INT  SELECT @emptySeat = dbo.tbTour.EmptySeat  FROM dbo.tbTour  WHERE TourID = @tourID  IF @emptySeat < @quantity  ROLLBACK TRAN  ELSE  BEGIN  UPDATE dbo.tbTour  SET EmptySeat = EmptySeat - @quantity  WHERE TourID = @tourID  DECLARE @quantityDiscount INT  EXEC @quantityDiscount = dbo.spGetQuantityDiscount @quantity = @quantity -- int  UPDATE dbo.tbContract SET QuantityDiscount = @quantityDiscount,  TerminationDate = DATEADD(DAY, 2, @endDate),  DraftDate = GETDATE(),  StartTime = @startDate  WHERE ContractID = @contractID  END  END | |
| - Khi thay đổi về số lượng thành viên của hợp đồng, nếu số lượng không hợp lệ thì không thay đổi được, ngược lại thì sẽ tự động thay đổi về chiết khấu theo số lượng, cập nhật lại số chỗ trống của tour.  - Khi huỷ hợp đồng sẽ đánh dấu hợp đồng đã bị huỷ, hoàn lại số chỗ trống của tour và lưu lại thông tin hoàn cọc. | * Sử dụng trigger, tự động gọi khi thay đổi thông tin hợp đồng. * Sử dụng store procedure để tính chiết khấu theo số lượng khi sửa và tính tỉ lệ hoàn cọc khi huỷ hợp đồng. |
| * Code:   CREATE TRIGGER trgUpdateContract  ON dbo.tbContract  FOR UPDATE  AS  BEGIN  DECLARE @oldQuantity INT, @tourID INT, @contractID INT, @newQuantity INT, @emptySeat INT, @state INT  SELECT @contractID = Inserted.ContractID, @newQuantity = Inserted.Quantity, @oldQuantity = Deleted.Quantity,  @emptySeat = dbo.tbTour.EmptySeat, @tourID = Inserted.TourID, @state = Inserted.State  FROM (Inserted INNER JOIN Deleted ON Deleted.ContractID = Inserted.ContractID) INNER JOIN dbo.tbTour ON tbTour.TourID = Inserted.TourID  IF @sate = 2  BEGIN  UPDATE dbo.tbTour  SET EmptySeat = @emptySeat + @oldQuantity  WHERE TourID = @tourID  DECLARE @refundedRate INT  EXEC @refundedRate = dbo.spGetRefundRate @contractID = @contractID -- int  INSERT INTO dbo.tbRefundContract  ( ContractID ,  RefundedDate ,  RefundedRate  )  VALUES ( @contractID , -- ContractID - int  GETDATE() , -- RefundedDate - date  @refundedRate -- RefundedRate - int  )  END  IF @state = 0  BEGIN  IF (@emptySeat + @oldQuantity - @newQuantity) < 0 OR @newQuantity = 0  ROLLBACK TRAN  ELSE  IF @oldQuantity != @newQuantity  BEGIN  DECLARE @quantityDiscount INT  EXEC @quantityDiscount = dbo.spQuantityDiscount @quantity = @newQuantity -- int  UPDATE dbo.tbContract SET QuantityDiscount = @quantityDiscount  WHERE ContractID = @contractID  UPDATE dbo.tbTour SET EmptySeat = @emptySeat + @oldQuantity - @newQuantity  WHERE TourID = @tourID  END  DELETE FROM dbo.tbMember WHERE dbo.tbMember.ContractID = @contractID  END  END | |

|  |  |
| --- | --- |
| Khi xoá 1 hợp đồng sẽ xoá các thông tin liên quan đế hợp đồng đó. | * Sử dụng trigger, tự động gọi khi xoá hợp đồng |
| * Code:   CREATE TRIGGER trgDeleteContract  ON dbo.tbContract  INSTEAD OF DELETE  AS  BEGIN  DECLARE @contractId INT  SELECT @contractId = Deleted.ContractId FROM Deleted  DELETE FROM dbo.tbMember WHERE ContractId = @contractId  DELETE FROM dbo.tbRefundedContract WHERE ContractId = @contractId  DELETE FROM dbo.tbAccident WHERE ContractId = @contractId  DELETE FROM dbo.tbContract WHERE ContractId = @contractId  END | |