create database 易驾驾校 collate Chinese\_PRC\_CI\_AS  
go  
grant connect on database :: 易驾驾校 to dbo  
go

**一、考试记录表ExamRecords**

考试（考试编号、开始日期、考试类型、考试成绩、未通过原因、学员编号、驾校编号）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **字段描述** | **允许为空** | **备注** |
| **ExamID** | int | 考试编号 | N | 主键，唯一 |
| **ExamDate** | date | 考试日期 | N |  |
| **ExamType** | tinyint | 考试类型：实践/理论 | N | 0：理论；1：实践 |
| **Grade** | int | 考试成绩 | Y | 这里当做整数处理 |
| **FailureReason** | varchar(255) | 未通过考试原因 | Y |  |
| **StuID** | int | 学员编号 | N | 外键 |
| **SchoolID** | int | 驾校编号 | N | 外键 |

-- auto-generated definition  
create table ExamRecord  
(  
ExamID int identity  
constraint ExamRecord\_pk  
primary key nonclustered,  
ExamDate date not null,  
ExamType tinyint default 0 not null,  
Grade int,  
FailureReason varchar(255),  
StuID int not null  
constraint StuID\_fk  
references Students,  
SchoolID int not null  
constraint ExamRecord\_\_fk  
references Schools  
)  
go

create unique index ExamRecord\_ExamID\_uindex  
on ExamRecord (ExamID)  
go

**二、车辆表Cars**

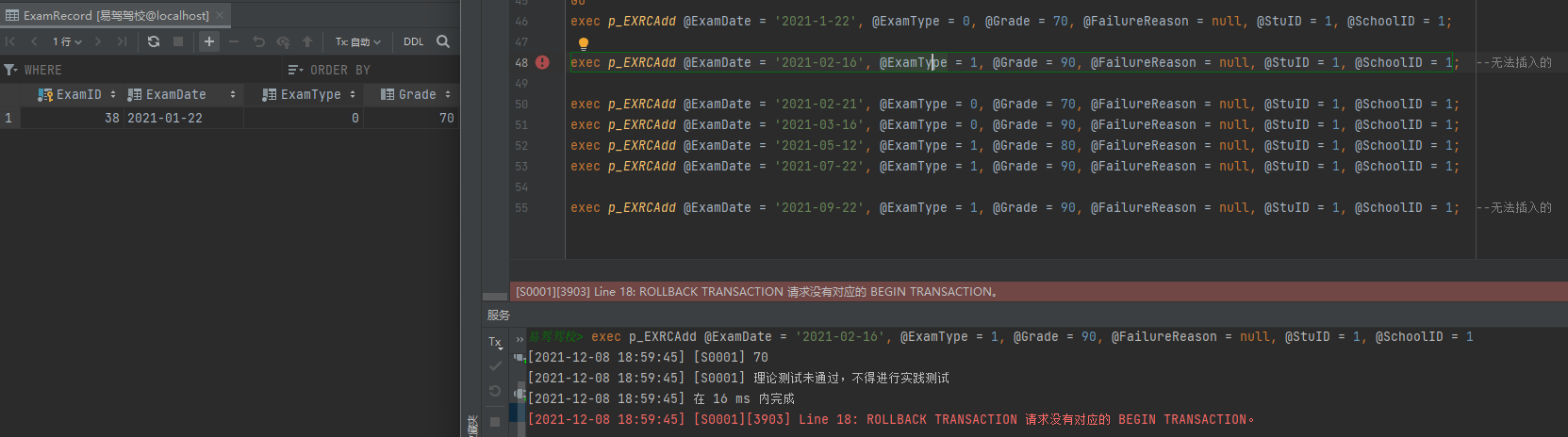
车辆（车牌号、驾校编号、教练编号）

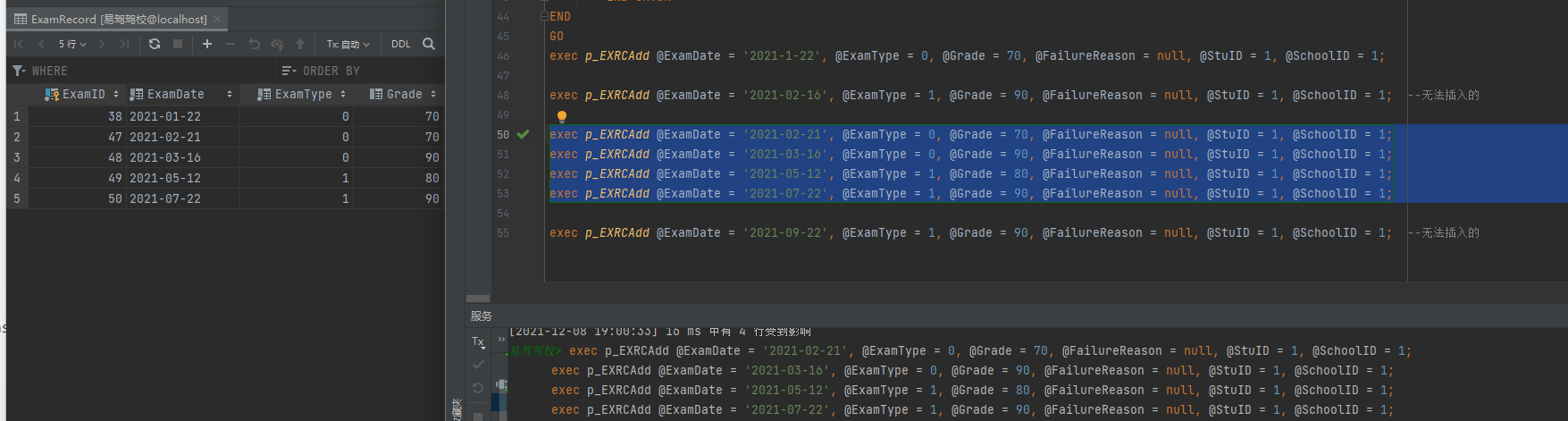
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名称** | **数据类型** | **字段描述** | **允许为空** | **备注** |
| **CarID** | char(8) | 车牌号 | N | 主键，唯一 |
| **SchoolID** | int | 驾校编号 | N | 外键 |
| **CoachID** | int | 教练编号 | N | 外键 |

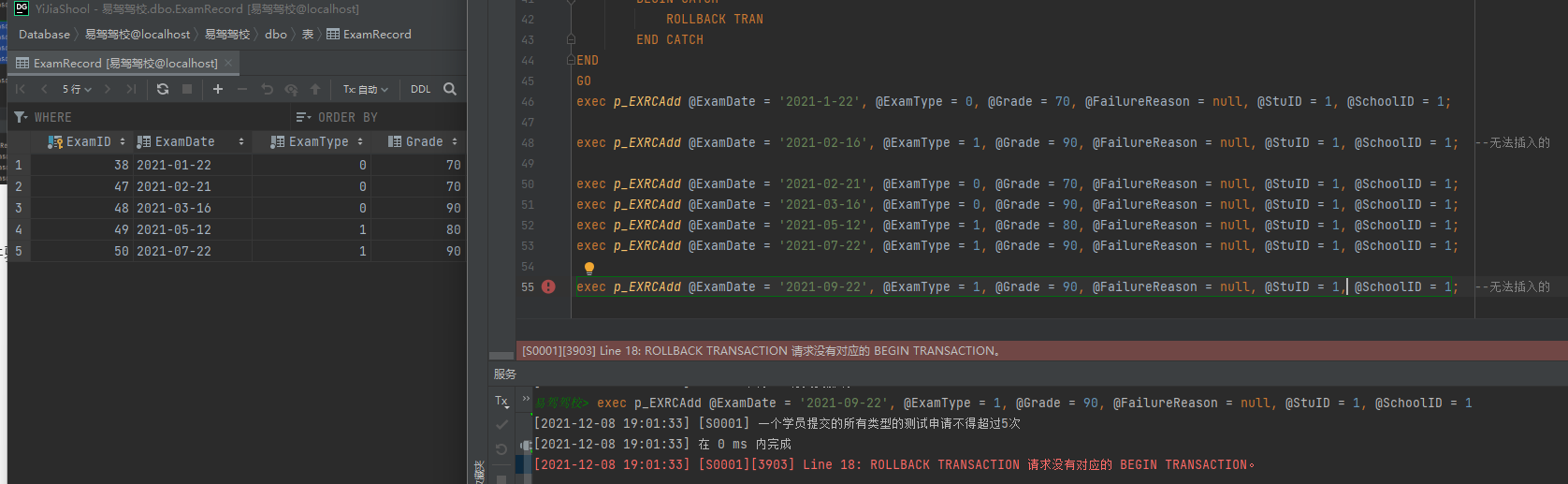
-- auto-generated definition  
create table Cars  
(  
CarID char(8) not null  
constraint Cars\_pk  
primary key nonclustered,  
SchoolID int not null  
constraint SchoolID\_fk  
references Schools,  
CoachID int not null  
constraint CoachID\_fk  
references Employees  
)  
go  
  
create unique index Cars\_CarID\_uindex  
on Cars (CarID)  
go

**三、触发器**

6学员提交实践测试申请时，要求其理论测试已经通过，时间的间隔要超过一个月；采用触发器实现该约束，并验证该约束的有效性。7、要求一个学员提交的所有类型的测试申请不得超过5次；采用触发器实现该约束，并验证该约束的有效性。

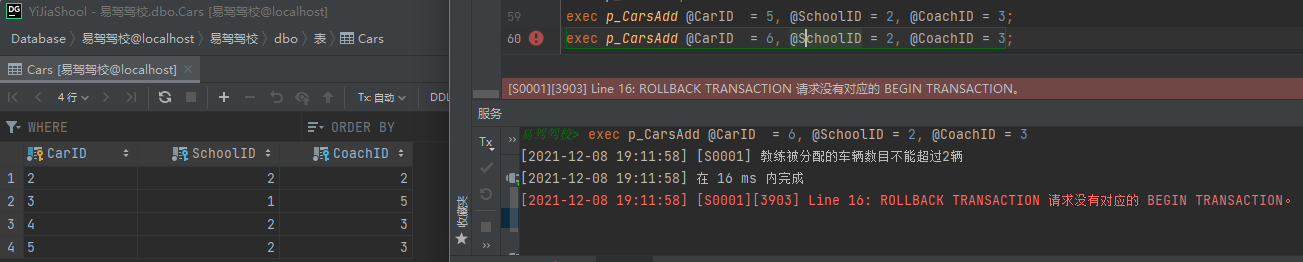
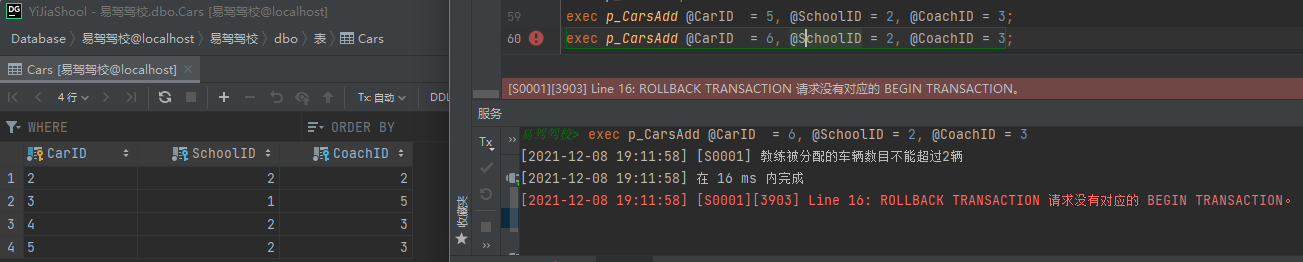
可以合并在这一个触发器中，插入后检测是否满足条件，不满足则回退。  
---创建insert触发器 ---  
create trigger trg\_examRecordBFInsert  
on ExamRecord  
after insert  
as  
begin  
declare @sid int;  
set @sid = (select StuID from inserted)  
if (select COUNT(\*) from ExamRecord a where a.StuID = @sid GROUP BY a.StuID) > 5  
begin  
print '一个学员提交的所有类型的测试申请不得超过5次'  
rollback transaction  
end  
  
declare @d1 date;  
declare @d2 date;  
declare @grade1 int;  
  
set @d1 = (select max(ExamDate) from ExamRecord a where a.StuID = @sid and a.ExamType != 1 GROUP BY a.StuID)  
set @d2 = (select ExamDate from inserted)  
set @grade1 = (select max(Grade) from ExamRecord a where a.StuID = @sid and a.ExamType != 1 GROUP BY a.StuID)  
if ( (select ExamType from inserted) = 1)  
begin  
if( @grade1 < 90)  
begin  
print '理论测试未通过，不得进行实践测试'  
rollback transaction  
end  
if((select datediff(month , @d1, @d2)) < 1)  
begin  
print '上次测试时间间隔未超过一个月，不得进行实践测试'  
rollback transaction  
end  
end  
end  
  






1.8要求教练被分配的车辆数目不能超过2辆；采用触发器实现该约束，并验证该约束的有效性。

创建触发器：

---创建insert触发器 ---  
create trigger trg\_CarsInsert  
on Cars  
after insert  
as  
if (select COUNT(\*) from Cars a where a.CoachID = (select CoachID from inserted) GROUP BY a.CoachID) > 2  
begin  
print '教练被分配的车辆数目不能超过2辆'  
rollback transaction  
end  


**四、存储过程**

**CarProcedure.sql**

# --添加  
CREATE PROCEDURE *p\_CarsAdd*@CarID char(8),  
@SchoolID int,  
@CoachID int  
AS  
BEGIN  
BEGIN TRAN  
BEGIN TRY  
INSERT INTO dbo.Cars  
( CarID, SchoolID, CoachID )  
VALUES ( @CarID, @SchoolID, @CoachID )  
COMMIT TRAN  
END TRY  
  
BEGIN CATCH  
ROLLBACK TRAN  
END CATCH  
END  
GO  
  
--删除  
CREATE PROCEDURE *p\_CarsDel*@CarID int  
AS  
BEGIN  
DELETE FROM dbo.Cars WHERE CarID = @CarID  
END  
GO  
  
--查询  
CREATE PROCEDURE *p\_CarsFindByID*@CarID int  
AS  
BEGIN  
select \* FROM dbo.Cars WHERE CarID = @CarID  
END  
GO  
  
--显示全部  
CREATE PROCEDURE *p\_CarsAll*AS  
BEGIN  
select \* FROM dbo.Cars  
END  
GO  
  
--------测试-------  
exec *p\_CarsAdd* @CarID = 1, @SchoolID = 1, @CoachID = 1;  
exec *p\_CarsDel* @CarID = 1;  
  
exec *p\_CarsAdd* @CarID = 2, @SchoolID = 2, @CoachID = 2;  
exec *p\_CarsFindByID* @CarID = 2;  
  
exec *p\_CarsAdd* @CarID = 3, @SchoolID = 1, @CoachID = 5;  
exec *p\_CarsAdd* @CarID = 4, @SchoolID = 2, @CoachID = 3;  
exec *p\_CarsAll*;  
exec *p\_CarsAdd* @CarID = 5, @SchoolID = 2, @CoachID = 3;  
exec *p\_CarsAdd* @CarID = 6, @SchoolID = 2, @CoachID = 3;

**ExamProcedure.sql**

use 易驾驾校  
create table ExamRecord  
(  
ExamID int identity  
constraint ExamRecord\_pk  
primary key nonclustered,  
ExamDate date not null,  
ExamType tinyint default 0 not null,  
Grade int,  
FailureReason varchar(255),  
StuID int not null  
constraint StuID\_fk  
references Students,  
SchoolID int not null  
constraint ExamRecord\_\_fk  
references Schools  
)  
go  
create unique index ExamRecord\_ExamID\_uindex  
on ExamRecord (ExamID)  
go  
  
  
--添加  
CREATE PROCEDURE *p\_EXRCAdd*@ExamDate date,  
@ExamType tinyint,  
@Grade int,  
@FailureReason varchar(255),  
@StuID int,  
@SchoolID int  
AS  
BEGIN  
BEGIN TRAN  
BEGIN TRY  
INSERT INTO ExamRecord  
( ExamDate, ExamType, Grade, FailureReason, StuID, SchoolID )  
VALUES ( @ExamDate, @ExamType, @Grade, @FailureReason, @StuID, @SchoolID )  
COMMIT TRAN  
END TRY  
BEGIN CATCH  
ROLLBACK TRAN  
END CATCH  
END  
GO  
exec *p\_EXRCAdd* @ExamDate = '2021-1-22', @ExamType = 0, @Grade = 70, @FailureReason = null, @StuID = 1, @SchoolID = 1;  
  
exec *p\_EXRCAdd* @ExamDate = '2021-02-16', @ExamType = 1, @Grade = 90, @FailureReason = null, @StuID = 1, @SchoolID = 1; --无法插入的  
  
exec *p\_EXRCAdd* @ExamDate = '2021-02-21', @ExamType = 0, @Grade = 70, @FailureReason = null, @StuID = 1, @SchoolID = 1;  
exec *p\_EXRCAdd* @ExamDate = '2021-03-16', @ExamType = 0, @Grade = 90, @FailureReason = null, @StuID = 1, @SchoolID = 1;  
exec *p\_EXRCAdd* @ExamDate = '2021-05-12', @ExamType = 1, @Grade = 80, @FailureReason = null, @StuID = 1, @SchoolID = 1;  
exec *p\_EXRCAdd* @ExamDate = '2021-07-22', @ExamType = 1, @Grade = 90, @FailureReason = null, @StuID = 1, @SchoolID = 1;  
  
exec *p\_EXRCAdd* @ExamDate = '2021-09-22', @ExamType = 1, @Grade = 90, @FailureReason = null, @StuID = 1, @SchoolID = 1; --无法插入的