با استفاده از

Microsoft Sql Server

گزارشکار فاز دو پروژه پایگاه داده ها

اعضای گروه: امیرحسین قهرمانی – محمدحیدری – امیرمحمدعرفانیان

فهرست مطالب

مروری بر فاز یک پروژه: ۳

معرفی موجودیت‌ها..... ۳

موجودیت UserAccount ۳

موجودیت Ad ۳

موجودیت Car ۳

موجودیت Motorcycle..... ۴

موجودیت HeavyVehicle ۴

موجودیت PriceEstimate..... ۴

موجودیت CarFinder ۴

بررسی روابط میان موجودیت‌ها ۴

تحلیل طراحی ۵

مرحله اول : ایجاد پایگاه داده ۶

مرحله دوم : ایجاد جداول ۷

مرحله سوم : ثبت رکورد ها ۹

مرحله چهارم : ایجاد stored procedure ۱۰

پرسمان ۱ : درج آگهی فروش خودرو..... ۱۲

تست پرسمان ۱ (درج آگهی برای کاربر u4042) ۱۷

خروجی:..... ۱۷

پرسمان ۲: مدیریت آگهی‌های کاربران (لیست و حذف) ۱۷

تست پرسمان ۲ (لیست آگهی‌ها و حذف آگهی‌های کاربر دوم) ۲۵

خروجی:..... ۲۶

پرسمان ۳: خودروها بر اساس استان و وضعیت بدنه ۲۶

تست پرسمان ۳ (برند، مدل و کارکرد خودروهای استان البرز با وضعیت‌های بدون رنگ یا یک لکه رنگ) ۲۹

خروجی:..... ۲۹

پرسمان ۴: نمایش مرتب‌شده آگهی‌های پرسمان۳ ..... ۲۹
تست پرسمان ۴: نمایش خودروهای استان البرز (بدون رنگ یا یک لکه رنگ) مرتب شده بر اساس: ۳۱
ارزان‌ترین: ۳۱
خروجی: ۳۱
به‌روزترین ۳۱
جدیدترین سال تولید ۳۱
کم‌کارکردترین: ۳۲
پرسمان ۵: حذف کم‌کارکردترین آگهی و نمایش مجدد پرسمان ۳۲
تست پرسمان ۵ (حذف کم‌کارکردترین آگهی و اجرای مجدد پرسمان ۴) ۳۴
خروجی: ۳۴
پرسمان ۶: موتورسیکلت‌ها بر اساس برند، حجم و قیمت ۳۵
تست پرسمان ۶ (موتورسیکلت‌های تی وی اس یا استلز، حجم < ۱۰۰۰، قیمت > ۳۰۰ میلیون) ۳۶
خروجی: ۳۶
پرسمان ۷: کف و سقف قیمت تخمینی خودرو ۳۶
تست پرسمان ۷ (کف و سقف قیمت بنز، کارکرد ۱۰۰۰۰، وضعیت بدون رنگ) ۳۷
خروجی: ۳۷
پرسمان ۸: برندهای خودروهای ساخت کشور مشخص که فقط در دو شهر خاص آگهی شده‌اند ۳۸
تست پرسمان ۸ (برند خودروهای آلمانی که فقط در تهران یا ساری آگهی شده‌اند) ۳۸
خروجی: ۳۹
پرسمان ۹: مشخصات خودروهایی با رنگ غیر از C، شاسی و سوخت مشخص در استان مشخص ۳۹
تست پرسمان ۹ (مشخصات خودروهای غیر سفید، شاسی‌بلند، هیبریدی، استان اصفهان) – قبل از تغییر رنگ ۴۰
خروجی: ۴۱
پرسمان ۱۰: تغییر رنگ خودروها و اجرای مجدد پرسمان۹ ..... ۴۱
تست پرسمان ۱۰ (تغییر رنگ مشکی به سفید و اجرای مجدد پرسمان ۹) ۴۲
خروجی: ۴۳

به نام خدا

مروری بر فاز یک پروژه:

معرفی موجودیت‌ها

موجودیت UserAccount

این موجودیت اطلاعات مربوط به کاربران سامانه را نگهداری می‌کند. شماره تلفن به عنوان شناسه اصلی کاربر در نظر گرفته شده و هر کاربر از طریق آن در سیستم شناسایی می‌شود. همچنین وضعیت آگهی‌های هر کاربر نیز در این موجودیت ذخیره می‌شود. تمامی آگهی‌های ثبت‌شده در سامانه به یکی از کاربران این بخش تعلق دارند.

موجودیت Ad

موجودیت Ad هسته اصلی پایگاه داده محسوب می‌شود و اطلاعات عمومی هر آگهی در آن ذخیره می‌شود. این اطلاعات شامل شناسه آگهی، شماره تلفن ثبت‌کننده، نوع وسیله نقلیه، قیمت نقدی، امکان مذاکره، شرایط فروش اقساطی، تاریخ تحویل، توضیحات و فایل‌های چندرسانه‌ای است.

با قرار گرفتن اطلاعات عمومی در این موجودیت، از تکرار داده‌ها جلوگیری شده و مدیریت آگهی‌ها ساده‌تر خواهد شد.

موجودیت Car

این موجودیت مشخصات اختصاصی خودروهای سواری را نگهداری می‌کند. اطلاعاتی مانند برند، مدل، تیپ، سال تولید، میزان کارکرد، رنگ بدنه، رنگ داخلی، وضعیت بدنه، نوع گیربکس، کشور سازنده، نوع خودرو و محل استقرار خودرو در این بخش ثبت می‌شود.

جدا کردن اطلاعات تخصصی خودرو از اطلاعات عمومی آگهی، باعث افزایش انسجام و کاهش افزونگی داده‌ها شده است.

موجودیت Motorcycle

این موجودیت برای ذخیره اطلاعات موتورسیکلت‌ها طراحی شده است. ویژگی‌هایی مانند برند، مدل، سال تولید، نوع سوخت، گیربکس، میزان کارکرد، رنگ و محل استقرار در آن نگهداری می‌شود. ساختار این موجودیت مشابه موجودیت خودرو است، اما تنها ویژگی‌های مرتبط با موتورسیکلت را شامل می‌شود.

موجودیت HeavyVehicle

این موجودیت اطلاعات مربوط به وسایل نقلیه سنگین را ذخیره می‌کند. علاوه بر مشخصات عمومی، نوع وسیله و نوع کاربری نیز در آن ثبت می‌شود. وجود این موجودیت امکان مدیریت مستقل خودروهای سنگین را در کنار سایر وسایل نقلیه فراهم می‌کند.

موجودیت PriceEstimate

این موجودیت برای تخمین قیمت خودرو در نظر گرفته شده است. کاربر با وارد کردن اطلاعاتی مانند برند، مدل، سال ساخت، میزان کارکرد و وضعیت بدنه، قیمت تقریبی خودرو را دریافت می‌کند. نتیجه محاسبه نیز در این موجودیت ذخیره می‌شود.

موجودیت CarFinder

این موجودیت مربوط به جست‌وجوهای ذخیره‌شده کاربران است. کاربران می‌توانند فیلترهایی مانند محدوده قیمت، نوع خودرو، نوع گیربکس، کشور سازنده، استان، شهر و سایر شرایط موردنظر خود را ذخیره کرده و در مراجعات بعدی مجدداً از آن‌ها استفاده کنند. همچنین زمان ایجاد و آخرین استفاده از هر جست‌وجو نیز ثبت می‌شود.

بررسی روابط میان موجودیت‌ها

در این مدل، ارتباط میان موجودیت‌ها به گونه‌ای طراحی شده است که از افزونگی اطلاعات جلوگیری شود و یکپارچگی داده‌ها حفظ گردد.

هر کاربر می‌تواند چندین آگهی ثبت کند، اما هر آگهی تنها متعلق به یک کاربر است؛ بنابراین رابطه میان UserAccount و Ad از نوع یک به چند است.

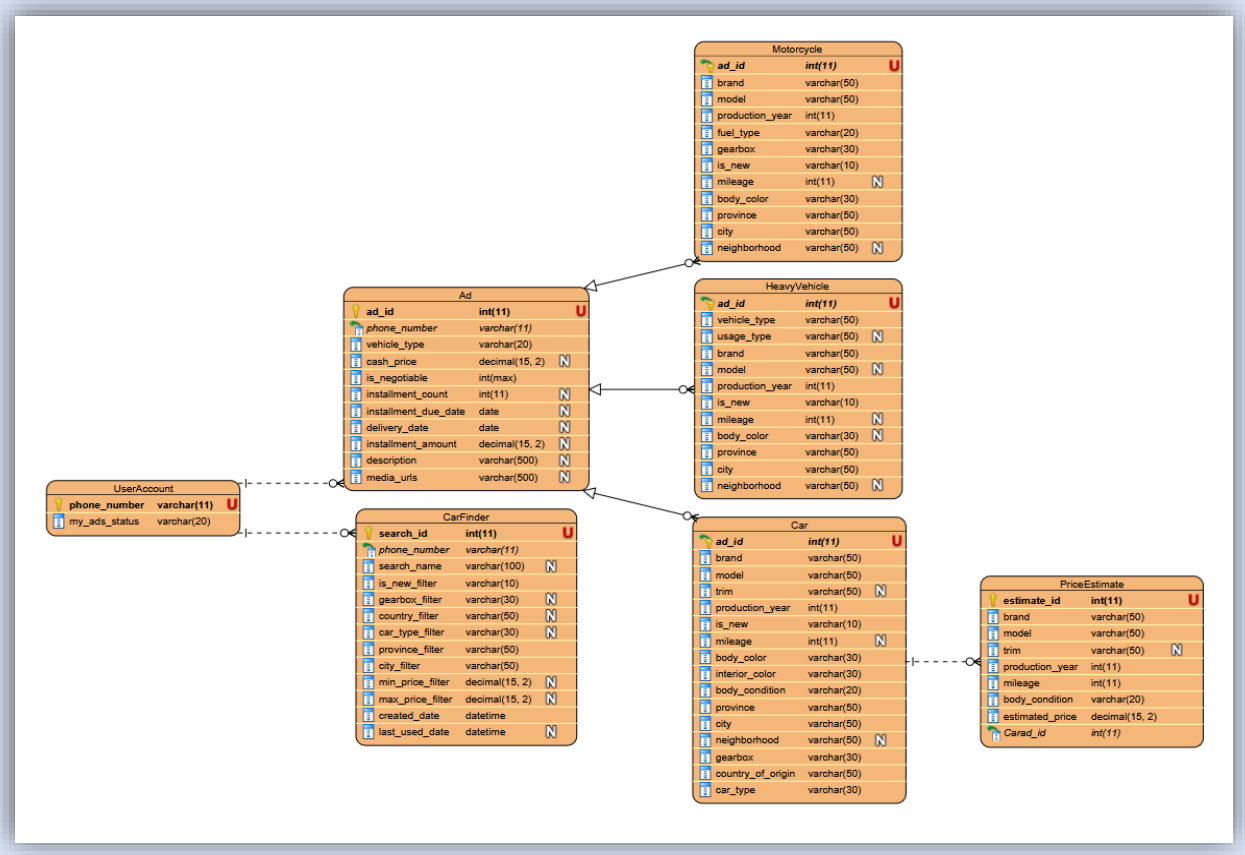
هر آگهی، بسته به نوع وسیله نقلیه، دارای اطلاعات اختصاصی در یکی از موجودیت‌های Car، Motorcycle یا HeavyVehicle خواهد بود. این ساختار باعث می‌شود ویژگی‌های هر نوع وسیله نقلیه به صورت مجزا نگهداری شده و طراحی پایگاه داده انعطاف‌پذیرتر باشد.

همچنین موجودیت PriceEstimate برای ثبت عملیات تخمین قیمت خودرو و موجودیت CarFinder برای ذخیره جست‌وجوهای کاربران با سایر بخش‌های سیستم در ارتباط هستند و امکانات تکمیلی سامانه را فراهم می‌کنند.

تحلیل طراحی

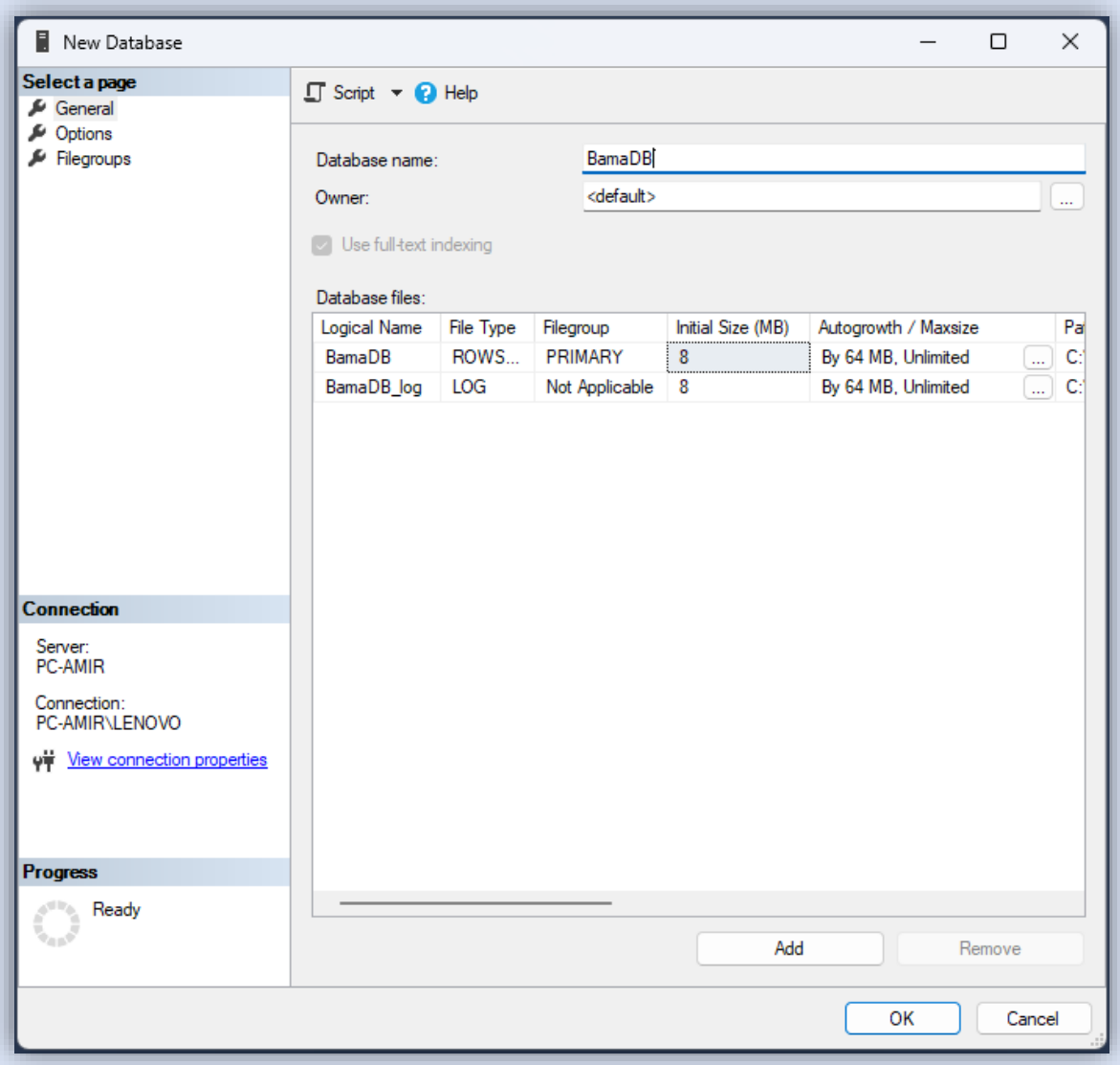
اطلاعات عمومی و اطلاعات تخصصی از یکدیگر تفکیک شده‌اند و هر موجودیت تنها مسئول نگهداری داده‌های مرتبط با خود است. همچنین از وراثت استفاده شده تا از تکرار داده ها جلوگیری شود و از مزایای وراثت استفاده شود. این روش علاوه بر کاهش افزونگی، موجب افزایش خوانایی مدل و سهولت نگهداری پایگاه داده می‌شود.

در انتخاب نوع داده‌ها نیز تناسب لازم رعایت شده است؛ به‌طوری‌که داده‌های متنی با نوع VARCHAR، شناسه‌ها و مقادیر عددی با INT، اطلاعات مالی با DECIMAL و داده‌های زمانی با DATE و DATETIME ذخیره می‌شوند که موجب استفاده بهینه از فضای ذخیره‌سازی می‌شود.



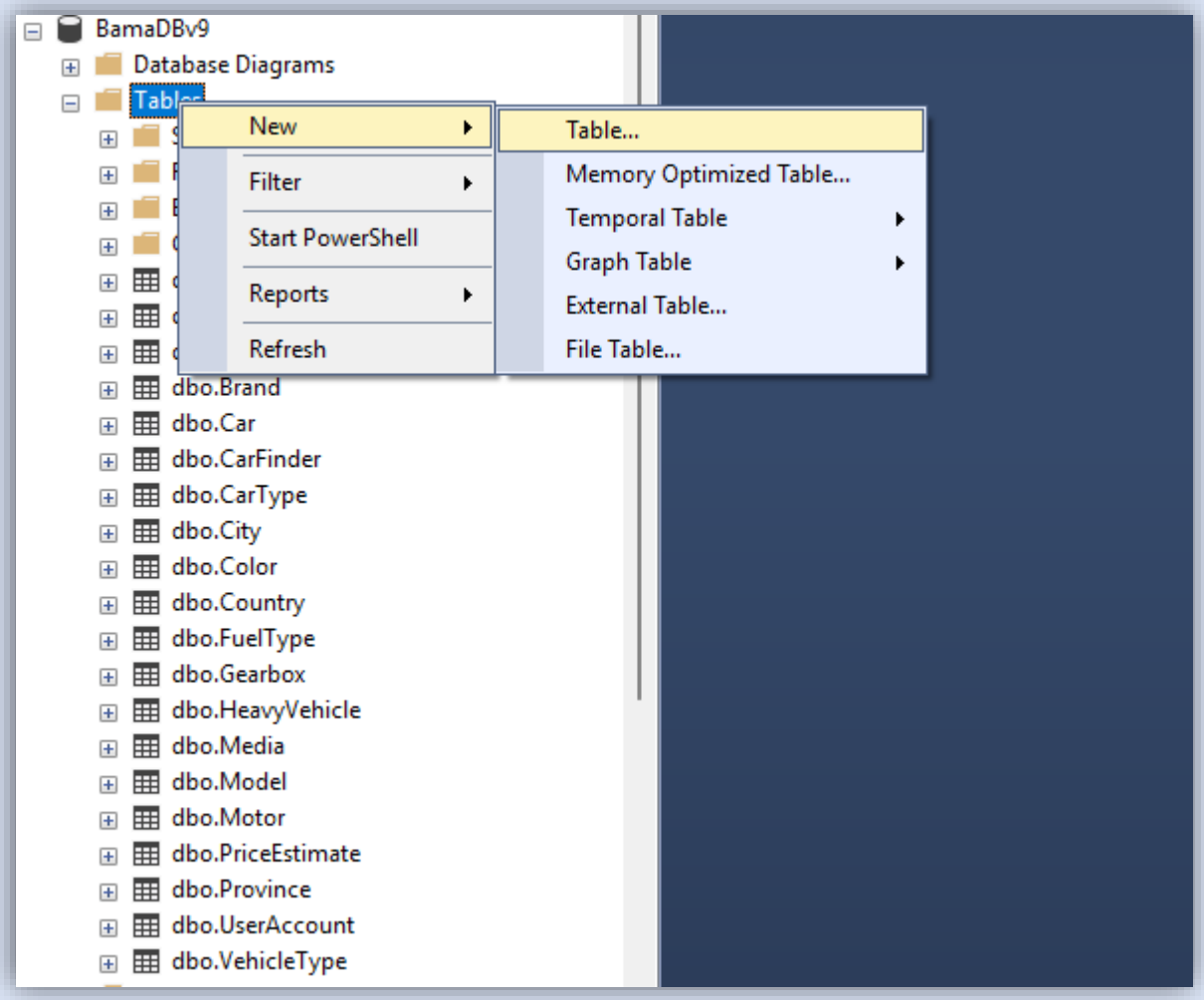
مرحله اول : ایجاد پایگاه داده

ابتدا در SSMS پایگاه داده خود را ایجاد میکنیم:



مرحله دوم : ایجاد جداول

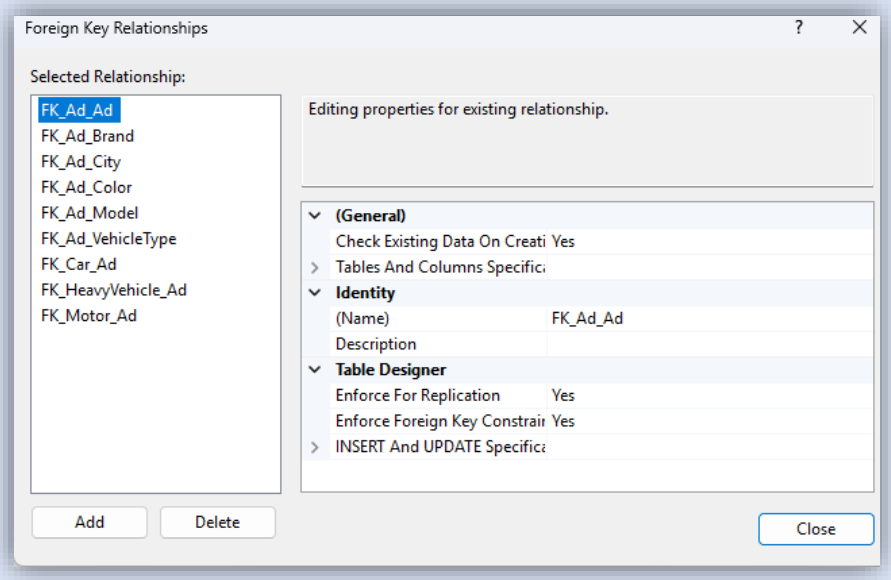
سپس با توجه به ER طراحی شده در فاز یک پروژه و اعمال تغییراتی جهت نرمال سازی تا 3NF به ایجاد و طراحی جداول می پردازیم:



در طراحی هر جدول نام ستون ، نوع داده، نال پذیری و کلید اصلی و روابط با کلید خارجی را تنظیم میکنیم:

Column Name	Data Type	Allow Nulls
ad_id	int	☐
phone_number	nvarchar(11)	☐
cash_price	decimal(15, 2)	☑
is_negitiable	int	☐
installment_count	int	☑
installment_due_date	date	☑
delivery_date	date	☑
installment_amount	decimal(15, 2)	☑
description	nvarchar(500)	☑
media_id	int	☐
production_year	date	☑
is_new	bit	☐
city_id	int	☐
neighborhood	nvarchar(50)	☑
mileage	int	☑
color_id	int	☐
brand_id	int	☐
model_id	int	☑
vehicle_type_id	int	☐
		☐

با استفاده از گزینه Relationships کلید های خارجی را تنظیم میکنیم:



مرحله سوم : ثبت رکورد ها

بعد از طراحی جداول تعدادی رکورد واقع گرایانه با توجه به سایت bama.ir به هر جدول اضافه میکنیم(حواسمون به ترتیب وارد کردن رکورد ها هست که کلید خارجی جداول به مشکل نخوره)

PC-AMIR.BamaDBv9 - dbo.City | PC-AMIR.BamaDBv9 - db

city_id	city_name	province_id
1	تهران	1
2	اصفهان	2
3	شیراز	3
4	مشهد	4
5	اهواز	5
6	ساری	6
7	رشت	7
8	تبریز	8
9	کرمان	9
10	همدان	10
11	کرج	1
12	کاشان	2
13	یزد	3
14	نیشابور	4
15	آبادان	5

با اجرای یک کوئری از ثبت رکورد ها اطمینان حاصل میکنیم:

```
/****** Script for SelectTopNRows
SELECT TOP (1000) [brand_id]
      ,[brand_name]
  FROM [BamaDBv9].[dbo].[Brand]
```

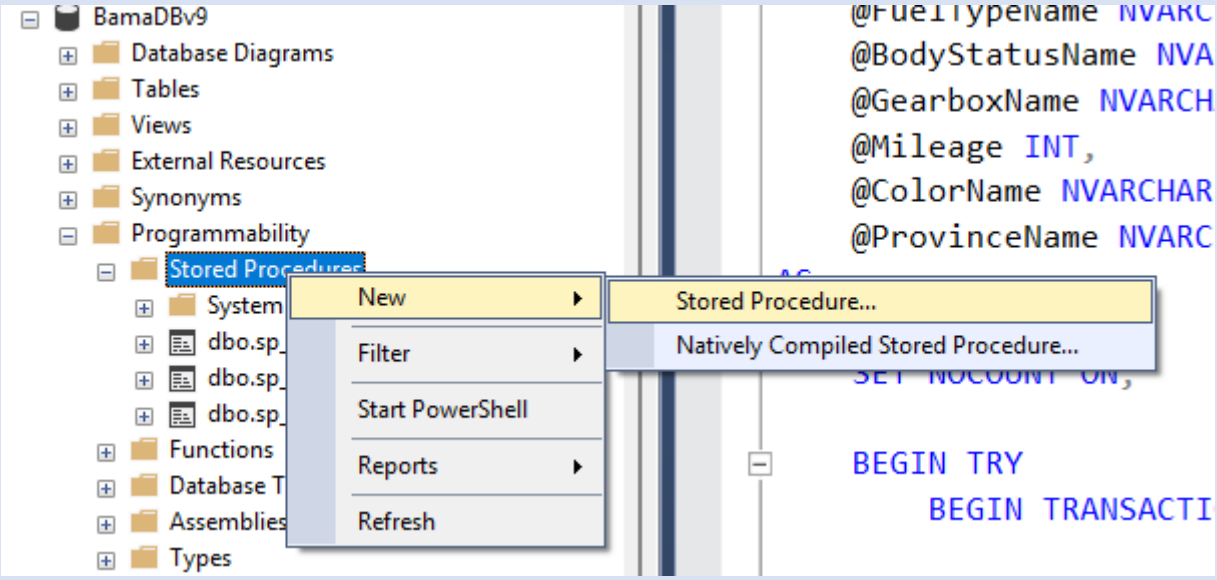
133 %

Results | Messages

	brand_id	brand_name
1	1	ایران خودرو
2	2	سایپا
3	3	بامو
4	4	مرسدس
5	5	تویوتا
6	6	هیوندای
7	7	کیا
8	8	رنو
9	9	پژو
10	10	فولکس واگن
11	11	ام وی ام

مرحله چهارم : ایجاد stored procedure

از مسیر زیر به طراحی استور پروسیجر ها می پردازیم:



```sql
USE [BamaDBv9]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[sp_InsertCarAd]
    @UserName NVARCHAR(50),
    @BrandName NVARCHAR(50),
    @ModelName NVARCHAR(50),
    @Year INT,
    @FuelTypeName NVARCHAR(50),
    @BodyStatusName NVARCHAR(50),
    @GearboxName NVARCHAR(50),
```

۱. درج آگهی فروش خودرو برای کاربر با پارامترهای نام کاربری userName ، برند Brand ، مدل Model، سال Year ، سوخت Fuel ، وضعیت بدنه Body status، گیربکس Gearbox ، کارکرد Function، رنگ Color و استان Province. درج آگهی فروش خودرو برای کاربر u4042 ، برند ام وی ام، مدل ۵۳۰ ، سال ۱۴۰۱ ، سوخت بنزین، وضعیت بدنه بدون رنگ، دندهای، استان فارس.

SQLQuery3.sql - P...AMIR\LENOVO (56))* | SQLQuery2.sql - P...AMIR\LENOV

```sql
USE [BamaDBv9]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[sp_ManageUserAds]
    @UserName1 NVARCHAR(50),
    @UserName2 NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;
```

۲. لیست آگهی های کاربران با نام کاربری userName1وuserName2. حذف تمام آگهیهای کاربر با نام. userName2 لیست آگهیهای کاربران با نام کاربری userName1 و , u4021 :userName2userName1. u4032 :userName2

پرسمان ۱ : درج آگهی فروش خودرو

"sp_InsertCarAd" برای ثبت آگهی فروش خودرو استفاده می‌شود و اطلاعاتی مانند نام کاربر، برند، مدل، سال تولید، نوع سوخت، وضعیت بدنه، نوع گیربکس، کارکرد، رنگ و استان را دریافت می‌کند.

در این رویه ابتدا بررسی می‌شود که آیا شماره تلفن کاربر در جدول "UserAccount" وجود دارد یا خیر. اگر وجود نداشته باشد، حساب کاربری جدید برای او ایجاد می‌شود. سپس شناسه‌های مربوط به برند، مدل، نوع سوخت، وضعیت بدنه، گیربکس، رنگ و استان بازیابی می‌شوند. در صورتی که هر یک از این مقادیر در جدول موجود نباشند، مقدار جدید برای آن‌ها ساخته و در جدول مناسب درج می‌شود.

پس از آن، برای استان مورد نظر یک شهر پیش‌فرض انتخاب یا ایجاد می‌شود. همچنین یک رکورد پیش‌فرض برای جدول "Media" ثبت می‌گردد. سپس آگهی اصلی در جدول "Ad" درج می‌شود و اطلاعات تکمیلی خودرو نیز در جدول‌های "Car" و "Motor" ذخیره می‌شوند.

در پایان، در صورت موفقیت عملیات، پیام موفقیت همراه با اطلاعات آگهی بازگردانده می‌شود و در صورت بروز خطا، تراکنش Rollback شده و پیام خطا نمایش داده می‌شود.

```
CREATE PROCEDURE [dbo].[sp_InsertCarAd]
    @UserName NVARCHAR(50),
    @BrandName NVARCHAR(50),
    @ModelName NVARCHAR(50),
    @Year INT,
    @FuelTypeName NVARCHAR(50),
    @BodyStatusName NVARCHAR(50),
    @GearboxName NVARCHAR(50),
    @Mileage INT,
    @ColorName NVARCHAR(50),
    @ProvinceName NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        DECLARE @PhoneNumber NVARCHAR(11);
```

```
    DECLARE @BrandId INT;
    DECLARE @ModelId INT;
    DECLARE @FuelTypeId INT;
    DECLARE @BodyConditionId INT;
    DECLARE @GearboxId INT;
    DECLARE @ColorId INT;
    DECLARE @ProvinceId INT;
    DECLARE @CityId INT;
    DECLARE @AdId INT;
    DECLARE @MediaId INT;
    DECLARE @VehicleTypeId INT = 1;
    DECLARE @CountryId INT = 1;
    DECLARE @CarTypeId INT = 1;
    DECLARE @ProductionYear DATE;
    DECLARE @IsNew BIT = 0;

    SELECT @PhoneNumber = [phone_number]
    FROM [dbo].[UserAccount]
    WHERE [phone_number] = @UserName;

    IF @PhoneNumber IS NULL
    BEGIN
        SET @PhoneNumber = @UserName;
        INSERT INTO [dbo].[UserAccount] ([phone_number], [my_ads_status_id])
        VALUES (@PhoneNumber, 1);
    END

    SELECT @BrandId = [brand_id]
    FROM [dbo].[Brand]
    WHERE [brand_name] = @BrandName;

    IF @BrandId IS NULL
    BEGIN
        SELECT @BrandId = ISNULL(MAX([brand_id]), 0) + 1 FROM [dbo].[Brand];
        INSERT INTO [dbo].[Brand] ([brand_id], [brand_name])
        VALUES (@BrandId, @BrandName);
    END

    SELECT @ModelId = [model_id]
    FROM [dbo].[Model]
    WHERE [model_name] = @ModelName;
```

```
    IF @ModelId IS NULL
    BEGIN
        SELECT @ModelId = ISNULL(MAX([model_id]), 0) + 1 FROM [dbo].[Model];
        INSERT INTO [dbo].[Model] ([model_id], [model_name])
        VALUES (@ModelId, @ModelName);
    END

    SELECT @FuelTypeId = [fuel_type_id]
    FROM [dbo].[FuelType]
    WHERE [fuel_type_name] = @FuelTypeName;

    IF @FuelTypeId IS NULL
    BEGIN
        INSERT INTO [dbo].[FuelType] ([fuel_type_name])
        VALUES (@FuelTypeName);
        SET @FuelTypeId = SCOPE_IDENTITY();
    END

    SELECT @BodyConditionId = [body_condition_id]
    FROM [dbo].[BodyCondition]
    WHERE [body_condition_name] = @BodyStatusName;

    IF @BodyConditionId IS NULL
    BEGIN
        INSERT INTO [dbo].[BodyCondition] ([body_condition_name])
        VALUES (@BodyStatusName);
        SET @BodyConditionId = SCOPE_IDENTITY();
    END

    SELECT @GearboxId = [gearbox_id]
    FROM [dbo].[Gearbox]
    WHERE [gearbox_name] = @GearboxName;

    IF @GearboxId IS NULL
    BEGIN
        INSERT INTO [dbo].[Gearbox] ([gearbox_name])
        VALUES (@GearboxName);
        SET @GearboxId = SCOPE_IDENTITY();
    END

    SELECT @ColorId = [color_id]
    FROM [dbo].[Color]
```

```
    WHERE [color_name] = @ColorName;

    IF @ColorId IS NULL
    BEGIN
        SELECT @ColorId = ISNULL(MAX([color_id]), 0) + 1 FROM [dbo].[Color];
        INSERT INTO [dbo].[Color] ([color_id], [color_name])
        VALUES (@ColorId, @ColorName);
    END

    SELECT @ProvinceId = [province_id]
    FROM [dbo].[Province]
    WHERE [province_name] = @ProvinceName;

    IF @ProvinceId IS NULL
    BEGIN
        SELECT @ProvinceId = ISNULL(MAX([province_id]), 0) + 1 FROM [dbo].[Province];
        INSERT INTO [dbo].[Province] ([province_id], [province_name])
        VALUES (@ProvinceId, @ProvinceName);
    END

    SELECT TOP 1 @CityId = [city_id]
    FROM [dbo].[City]
    WHERE [province_id] = @ProvinceId;

    IF @CityId IS NULL
    BEGIN
        SELECT @CityId = ISNULL(MAX([city_id]), 0) + 1 FROM [dbo].[City];
        INSERT INTO [dbo].[City] ([city_id], [city_name], [province_id])
        VALUES (@CityId, N'مرکز استان', @ProvinceId);
    END

    SELECT @MediaId = ISNULL(MAX([media_id]), 0) + 1 FROM [dbo].[Media];
    INSERT INTO [dbo].[Media] ([media_id], [media_url], [upload_date])
    VALUES (@MediaId, N'/images/default_car.jpg', GETDATE());

    SELECT @AdId = ISNULL(MAX([ad_id]), 0) + 1 FROM [dbo].[Ad];

    SET @ProductionYear = DATEFROMPARTS(@Year, 1, 1);
    INSERT INTO [dbo].[Ad] (
        [ad_id], [phone_number], [cash_price], [is_negitiable],
        [installment_count], [installment_due_date], [delivery_date],
```

```
        [installment_amount], [description], [media_id],
        [production_year], [is_new], [city_id], [neighborhood],
        [mileage], [color_id], [brand_id], [model_id], [vehicle_type_id]
    ) VALUES (
        @AdId, @PhoneNumber, NULL, 1,
        NULL, NULL, NULL,
        NULL, CONCAT(N'آگهی فروش ', @BrandName, N' ', @ModelName, N' مدل ', @Year),
        @MediaId, @ProductionYear, @IsNew, @CityId,
        NULL, @Mileage, @ColorId, @BrandId, @ModelId, @VehicleTypeId
    );

    INSERT INTO [dbo].[Car] (
        [ad_id], [trim], [interior_color_id],
        [country_id], [body_condition_id], [gearbox_id], [car_type_id]
    ) VALUES (
        @AdId, NULL, NULL, @CountryId,
        @BodyConditionId, @GearboxId, @CarTypeId
    );

    INSERT INTO [dbo].[Motor] (
        [ad_id], [fuel_type_id], [gearbox_id]
    ) VALUES (
        @AdId, @FuelTypeId, @GearboxId
    );

    COMMIT TRANSACTION;

    SELECT
        @AdId AS AdId,
        @PhoneNumber AS PhoneNumber,
        @BrandName AS Brand,
        @ModelName AS Model,
        @Year AS [Year],
        @FuelTypeName AS FuelType,
        @BodyStatusName AS BodyStatus,
        @GearboxName AS Gearbox,
        @Mileage AS Mileage,
        @ColorName AS Color,
        @ProvinceName AS Province,
        N'آگهی با موفقیت درج شد' AS Message;

END TRY
```

```
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        SELECT
            ERROR_NUMBER() AS ErrorNumber,
            ERROR_MESSAGE() AS ErrorMessage,
            N'خطا در درج آگهی' AS Message;
    END CATCH
END
GO
```

تست پرسمان ۱ (درج آگهی برای کاربر u4042)

```
EXEC sp_InsertCarAd N'u4042', N'ام وی ام', N'530', 1401, N'بنزین', N'بدون رنگ',
N'دنده‌ای (دستی)', 0, N'سفید', N;'فارس'
```

خروجی:

Results | Messages

	AdId	PhoneNumber	Brand	Model	Year	FuelType	BodyStatus	Gearbox	Mileage	Color	Province	Message
1	26	u4042	ام وی ام	530	1401	بنزین	بدون رنگ	دنده‌ای (دستی)	0	سفید	فارس	آگهی با موفقیت درج شد

پرسمان ۲: مدیریت آگهی‌های کاربران (لیست و حذف)

"sp_ManageUserAds" برای نمایش و حذف آگهی‌های دو کاربر طراحی شده است. این رویه دو نام کاربری دریافت می‌کند و ابتدا آگهی‌های کاربر اول را نمایش می‌دهد، سپس آگهی‌های کاربر دوم را نشان می‌دهد و در ادامه آگهی‌های کاربر دوم را از جداول وابسته حذف می‌کند.

در این رویه ابتدا اطلاعات هر آگهی شامل شناسه آگهی، نام کاربر، برند، مدل، سال ساخت، نوع خودرو، قیمت، کارکرد، رنگ، شهر، استان، وضعیت و توضیحات از طریق JOIN بین جدول‌های مختلف استخراج می‌شود.

سپس تعداد آگهی‌های کاربر دوم شمارش می‌شود. اگر آگهی برای حذف وجود داشته باشد، ابتدا رکوردهای وابسته از جدول‌های ""PriceEstimate"، "Car"، "Motor"، "HeavyVehicle"" و "CarFinder" حذف می‌شوند و در نهایت رکوردهای مربوط به کاربر دوم از جدول "Ad" پاک می‌گردند.

در پایان، آگهی‌های کاربر اول و دوم پس از حذف نیز مجدداً نمایش داده می‌شوند تا نتیجه عملیات قابل مشاهده باشد. این رویه از تراکنش برای حفظ یکپارچگی داده‌ها استفاده می‌کند.

```
CREATE PROCEDURE [dbo].[sp_ManageUserAds]
    @UserName1 NVARCHAR(50),
    @UserName2 NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        PRINT '==========================================';
        PRINT N'لیست آگهی‌های کاربر اول (قبل از حذف):';
        PRINT '==========================================';

        SELECT
            a.ad_id,
            a.phone_number AS UserName,
            b.brand_name AS Brand,
            ISNULL(m.model_name, N'نامشخص') AS Model,
            YEAR(a.production_year) AS [Year],
            vt.vehicle_type_name AS VehicleType,
            a.cash_price AS Price,
```

```
        a.mileage,
        col.color_name AS Color,
        c.city_name AS City,
        p.province_name AS Province,
        CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
        a.description
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[VehicleType] vt ON a.vehicle_type_id = vt.vehicle_type_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE a.phone_number = @UserName1
    ORDER BY a.ad_id;

    PRINT '==========================================';
    PRINT N'لیست آگهی‌های کاربر دوم (قبل از حذف):';
    PRINT '==========================================';

    SELECT
        a.ad_id,
        a.phone_number AS UserName,
        b.brand_name AS Brand,
        ISNULL(m.model_name, N'نامشخص') AS Model,
        YEAR(a.production_year) AS [Year],
        vt.vehicle_type_name AS VehicleType,
        a.cash_price AS Price,
        a.mileage,
```

```
        col.color_name AS Color,
        c.city_name AS City,
        p.province_name AS Province,
        CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
        a.description
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[VehicleType] vt ON a.vehicle_type_id = vt.vehicle_type_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE a.phone_number = @UserName2
    ORDER BY a.ad_id;

    DECLARE @AdCount INT;

    SELECT @AdCount = COUNT(*)
    FROM [dbo].[Ad]
    WHERE [phone_number] = @UserName2;

    IF @AdCount > 0
    BEGIN
        PRINT '==========================================';
        PRINT N'در حال حذف آگهی‌های کاربر دوم...';
        PRINT N'تعداد آگهی‌های یافت شده: ' + CAST(@AdCount AS NVARCHAR(10));
        PRINT '==========================================';

        DELETE FROM [dbo].[PriceEstimate]
```

```
        WHERE [car_id] IN (
            SELECT [ad_id] FROM [dbo].[Car] WHERE [ad_id] IN (
                SELECT [ad_id] FROM [dbo].[Ad] WHERE [phone_number] = @UserName2
            )
        );

        DELETE FROM [dbo].[Car]
        WHERE [ad_id] IN (
            SELECT [ad_id] FROM [dbo].[Ad] WHERE [phone_number] = @UserName2
        );

        DELETE FROM [dbo].[Motor]
        WHERE [ad_id] IN (
            SELECT [ad_id] FROM [dbo].[Ad] WHERE [phone_number] = @UserName2
        );

        DELETE FROM [dbo].[HeavyVehicle]
        WHERE [ad_id] IN (
            SELECT [ad_id] FROM [dbo].[Ad] WHERE [phone_number] = @UserName2
        );

        DELETE FROM [dbo].[CarFinder]
        WHERE [phone_number] = @UserName2;

        DELETE FROM [dbo].[Ad]
        WHERE [phone_number] = @UserName2;

        PRINT N'آگهی‌های کاربر دوم با موفقیت حذف شدند.';
        PRINT N'تعداد آگهی‌های حذف شده: ' + CAST(@AdCount AS NVARCHAR(10));
```

```
    END
    ELSE
    BEGIN
        PRINT '==========================================';
        PRINT N'هیچ آگهی برای کاربر دوم یافت نشد.';
        PRINT '==========================================';
    END

    PRINT '==========================================';
    PRINT N'لیست آگهی‌های کاربر اول (بعد از حذف):';
    PRINT '==========================================';

    SELECT
        a.ad_id,
        a.phone_number AS UserName,
        b.brand_name AS Brand,
        ISNULL(m.model_name, N'نامشخص') AS Model,
        YEAR(a.production_year) AS [Year],
        vt.vehicle_type_name AS VehicleType,
        a.cash_price AS Price,
        a.mileage,
        col.color_name AS Color,
        c.city_name AS City,
        p.province_name AS Province,
        CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
        a.description
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
```

```
    INNER JOIN [dbo].[VehicleType] vt ON a.vehicle_type_id = vt.vehicle_type_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE a.phone_number = @UserName1
    ORDER BY a.ad_id;

    PRINT '==========================================';
    PRINT N'لیست آگهی‌های کاربر دوم (بعد از حذف):';
    PRINT '==========================================';

    SELECT
        a.ad_id,
        a.phone_number AS UserName,
        b.brand_name AS Brand,
        ISNULL(m.model_name, N'نامشخص') AS Model,
        YEAR(a.production_year) AS [Year],
        vt.vehicle_type_name AS VehicleType,
        a.cash_price AS Price,
        a.mileage,
        col.color_name AS Color,
        c.city_name AS City,
        p.province_name AS Province,
        CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
        a.description
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[VehicleType] vt ON a.vehicle_type_id = vt.vehicle_type_id
```

```
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE a.phone_number = @UserName2
    ORDER BY a.ad_id;

    DECLARE @User1Count INT, @User2Count INT;

    SELECT @User1Count = COUNT(*)
    FROM [dbo].[Ad]
    WHERE [phone_number] = @UserName1;

    SELECT @User2Count = COUNT(*)
    FROM [dbo].[Ad]
    WHERE [phone_number] = @UserName2;

    PRINT '==========================================';
    PRINT N'خلاصه عملیات:';
    PRINT N'کاربر اول (' + @UserName1 + N') تعداد آگهی‌ها: ' + CAST(@User1Count AS NVARCHAR(10));
    PRINT N'کاربر دوم (' + @UserName2 + N') تعداد آگهی‌ها: ' + CAST(@User2Count AS NVARCHAR(10));
    PRINT N'تعداد آگهی‌های حذف شده: ' + CAST(@AdCount AS NVARCHAR(10));
    PRINT '==========================================';

    COMMIT TRANSACTION;

    SELECT
        N'عملیات با موفقیت انجام شد' AS Message,
        @User1Count AS User1_AdCount,
```

```
        @User2Count AS User2_AdCount,
        ISNULL(@AdCount, 0) AS Deleted_AdCount;

    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        SELECT
            ERROR_NUMBER() AS ErrorNumber,
            ERROR_MESSAGE() AS ErrorMessage,
            N'خطا در انجام عملیات' AS Message;

        PRINT '==========================================';
        PRINT N'خطا در انجام عملیات:';
        PRINT N'شماره خطا: ' + CAST(ERROR_NUMBER() AS NVARCHAR(10));
        PRINT N'پیام خطا: ' + ERROR_MESSAGE();
        PRINT '==========================================';
    END CATCH
END
GO
```

تست پرسمان ۲ (لیست آگهی‌ها و حذف آگهی‌های کاربر دوم)

```
EXEC sp_ManageUserAds N'u4021', N'u4032;'
```

خروجی:

Results | Messages

	ad_id	UserName	Brand	Model	Year	VehicleType	Price	mileage	Color	City	Province	Status	description
1	30	u4021	ایران خودرو	پژو ۲۰۶	2022	Car	150000000.00	10000	سفید	تهران	تهران	کارکرده	آگهی کاربر u4021

	ad_id	UserName	Brand	Model	Year	VehicleType	Price	mileage	Color	City	Province	Status	description
1	31	u4032	سایپا	تیبا	2023	Car	200000000.00	20000	مشکی	اصفهان	اصفهان	کارکرده	آگهی کاربر u4032

	ad_id	UserName	Brand	Model	Year	VehicleType	Price	mileage	Color	City	Province	Status	description
1	30	u4021	ایران خودرو	پژو ۲۰۶	2022	Car	150000000.00	10000	سفید	تهران	تهران	کارکرده	آگهی کاربر u4021

	ad_id	UserName	Brand	Model	Year	VehicleType	Price	mileage	Color	City	Province	Status	description

	Message	User1_AdCount	User2_AdCount	Deleted_AdCount
1	عملیات با موفقیت انجام شد	1	0	1

پرسمان ۳: خودروها بر اساس استان و وضعیت بدنه

"sp_GetCarsByProvinceAndBodyStatus" برای فیلتر کردن خودروهای یک استان مشخص بر اساس وضعیت بدنه طراحی شده است. این رویه سه ورودی دریافت می‌کند: نام استان، یک وضعیت بدنه اصلی و در صورت نیاز یک وضعیت بدنه دوم.

در این رویه، فقط آگهی‌های مربوط به خودروهای سواری بررسی می‌شوند و از جدول‌های ""Ad"، "Car"، "City", "Color", "Model", "Brand", "BodyCondition"" و "Province" اطلاعات لازم گرفته می‌شود.

خروجی شامل برند، مدل، کارکرد، شماره تماس کاربر، شناسه آگهی، قیمت، سال تولید، رنگ، شهر، وضعیت صفر یا کارکرده، وضعیت بدنه و توضیحات است.

در پایان، تعداد کل خودروهای یافت‌شده نیز محاسبه و نمایش داده می‌شود. این رویه برای جست‌وجوی هدفمند خودروها در یک استان خاص کاربرد زیادی دارد.

```
CREATE PROCEDURE [dbo].[sp_GetCarsByProvinceAndBodyStatus]
    @ProvinceName NVARCHAR(50),
```

```
    @BodyStatus1 NVARCHAR(50),
    @BodyStatus2 NVARCHAR(50) = NULL
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        PRINT '==========================================';
        PRINT N'لیست خودروهای استان ' + @ProvinceName + N' با وضعیت بدنه ' + @BodyStatus1;
        IF @BodyStatus2 IS NOT NULL
            PRINT N'و ' + @BodyStatus2;
        PRINT '==========================================';

        SELECT
            ROW_NUMBER() OVER (ORDER BY b.brand_name, ISNULL(m.model_name, N'')) AS RowNumber,
            b.brand_name AS Brand,
            ISNULL(m.model_name, N'نامشخص') AS Model,
            a.mileage AS Mileage,
            a.phone_number AS UserPhone,
            a.ad_id AS AdId,
            a.cash_price AS Price,
            YEAR(a.production_year) AS [Year],
            col.color_name AS Color,
            c.city_name AS City,
            CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
            bc.body_condition_name AS BodyCondition,
            a.description
        FROM [dbo].[Ad] a
        INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
        LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
        INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
        INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
        INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
        INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
        INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
        WHERE p.province_name = @ProvinceName
            AND a.vehicle_type_id = 1
            AND (
                bc.body_condition_name = @BodyStatus1
                OR (@BodyStatus2 IS NOT NULL AND bc.body_condition_name = @BodyStatus2)
            )
        ORDER BY b.brand_name, ISNULL(m.model_name, N'');

        DECLARE @TotalCount INT;

        SELECT @TotalCount = COUNT(*)
```

```
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
    INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE p.province_name = @ProvinceName
      AND a.vehicle_type_id = 1
      AND (
        bc.body_condition_name = @BodyStatus1
        OR (@BodyStatus2 IS NOT NULL AND bc.body_condition_name = @BodyStatus2)
      );

    PRINT '==========================================';
    PRINT N'خلاصه آماری:';
    PRINT N'تعداد کل خودروهای یافت شده: ' + CAST(ISNULL(@TotalCount, 0) AS NVARCHAR(10));
    PRINT '==========================================';

    SELECT
      @ProvinceName AS Province,
      @BodyStatus1 AS BodyStatus1,
      ISNULL(@BodyStatus2, N'') AS BodyStatus2,
      ISNULL(@TotalCount, 0) AS TotalCount;

  END TRY
  BEGIN CATCH
    SELECT
      ERROR_NUMBER() AS ErrorNumber,
      ERROR_MESSAGE() AS ErrorMessage,
      N'خطا در دریافت اطلاعات' AS Message;

    PRINT '==========================================';
    PRINT N'خطا در دریافت اطلاعات:';
    PRINT N'شماره خطا: ' + CAST(ERROR_NUMBER() AS NVARCHAR(10));
    PRINT N'پیام خطا: ' + ERROR_MESSAGE();
    PRINT '==========================================';
  END CATCH
END

GO
```

تست پرسمان ۳ (برند، مدل و کارکرد خودروهای استان البرز با وضعیت‌های بدون رنگ یا یک لکه رنگ)

```
EXEC sp_GetCarsByProvinceAndBodyStatus N ,'البرز'N ,'بدون رنگ'N;'یک لکه رنگ'
```

خروجی:

Results | Messages

	RowNumber	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	1	ایران خودرو	پژو ۲۰۶	50000	09121111111	16	250000000.00	2020	سفید	کرج	کارکرده	بدون رنگ	آگهی تست البرز - بدون رنگ
2	2	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

پرسمان ۴: نمایش مرتب‌شده آگهی‌های پرسمان۳

"sp_GetOrderedAdsByProvinceAndBodyStatus" نسخه توسعه‌یافته‌تری از رویه قبلی است. این رویه علاوه بر فیلتر کردن بر اساس استان و وضعیت بدنه، امکان مرتب‌سازی نتایج بر اساس چهار معیار مختلف را نیز فراهم می‌کند:

ارزان‌ترین قیمت

جدیدترین آگهی

جدیدترین سال تولید

کمترین کارکرد.

ورودی "@OrderType" تعیین می‌کند که نتایج بر چه اساسی مرتب شوند.

در این رویه، با استفاده از "ORDER BY" شرطی، خروجی بر اساس نوع مرتب‌سازی انتخاب‌شده سازمان‌دهی می‌شود. این قابلیت باعث می‌شود کاربر بتواند آگهی‌های مورد نظر را با توجه به اولویت خود مشاهده کند.

```
CREATE PROCEDURE [dbo].[sp_GetOrderedAdsByProvinceAndBodyStatus]
    @ProvinceName NVARCHAR(50),
    @BodyStatus1 NVARCHAR(50),
    @BodyStatus2 NVARCHAR(50) = NULL,
    @OrderType NVARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;
    SELECT
        b.brand_name AS Brand,
        ISNULL(m.model_name, N'نامشخص') AS Model,
        a.mileage AS Mileage,
        a.phone_number AS UserPhone,
        a.ad_id AS AdId,
        a.cash_price AS Price,
        YEAR(a.production_year) AS [Year],
        col.color_name AS Color,
        c.city_name AS City,
        CASE WHEN a.is_new = 1 THEN N'صفر' ELSE N'کارکرده' END AS [Status],
        bc.body_condition_name AS BodyCondition,
        a.description
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
    INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    WHERE p.province_name = @ProvinceName
        AND a.vehicle_type_id = 1
        AND (bc.body_condition_name = @BodyStatus1 OR (@BodyStatus2 IS NOT NULL AND
bc.body_condition_name = @BodyStatus2))
    ORDER BY
        CASE WHEN @OrderType = 'Cheapest' THEN a.cash_price END,
        CASE WHEN @OrderType = 'MostRecent' THEN a.ad_id END DESC,
        CASE WHEN @OrderType = 'NewestYear' THEN a.production_year END DESC,
        CASE WHEN @OrderType = 'LowestMileage' THEN a.mileage END;
END
GO
```

تست پرسمان ۴: نمایش خودروهای استان البرز (بدون رنگ یا یک لکه رنگ) مرتب شده بر اساس:

ارزان‌ترین:

```
EXEC sp_GetOrderedAdsByProvinceAndBodyStatus N ,'البرز'N, 'بدون رنگ'N, 'یک لکه
' ,'رنگCheapest;'
```

خروجی:

Results | Messages

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	ایران خودرو	پژو ۲۰۶	50000	09121111111	16	250000000.00	2020	سفید	کرج	کارکرده	بدون رنگ	آگهی تست البرز - بدون رنگ
2	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

به‌روزترین:

```
EXEC sp_GetOrderedAdsByProvinceAndBodyStatus N ,'البرز'N, 'بدون رنگ'N, 'یک لکه
' ,'رنگMostRecent;'
```

خروجی:

Results | Messages

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ
2	ایران خودرو	پژو ۲۰۶	50000	09121111111	16	250000000.00	2020	سفید	کرج	کارکرده	بدون رنگ	آگهی تست البرز - بدون رنگ

جدیدترین سال تولید:

```
EXEC sp_GetOrderedAdsByProvinceAndBodyStatus N ,'البرز'N, 'بدون رنگ'N, 'یک لکه
' ,'رنگNewestYear;'
```

خروجی:

Results | Messages

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ
2	ایران خودرو	پژو ۲۰۶	50000	09121111111	16	250000000.00	2020	سفید	کرج	کارکرده	بدون رنگ	آگهی تست البرز - بدون رنگ

کم‌کارکردترین:

```
EXEC sp_GetOrderedAdsByProvinceAndBodyStatus N ,'البرز'N, 'بدون رنگ'N'یک لکه
' ,'رنگLowestMileage;'
```

خروجی:

Results | Messages

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	ایران خودرو	پژو ۲۰۶	50000	09121111111	16	250000000.00	2020	سفید	کرج	کارکرده	بدون رنگ	آگهی تست البرز - بدون رنگ
2	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

پرسمان ۵: حذف کم‌کارکردترین آگهی و نمایش مجدد پرسمان

"sp_DeleteLowestMileageAdAndShow" برای حذف آگهی با کمترین کارکرد در یک استان و با وضعیت بدنه مشخص طراحی شده است. این رویه پس از حذف، دوباره نتایج جست‌وجو را نمایش می‌دهد تا مشخص شود کدام آگهی حذف شده است.

در این رویه، ابتدا شناسه آگهی با کمترین مقدار "Mileage" پیدا می‌شود. اگر چنین آگهی‌ای وجود داشته باشد، رکوردهای وابسته آن از جدول‌های ""PriceEstimate"، "Car"، "Motor"، "HeavyVehicle"" و سپس خود جدول "Ad" حذف می‌شوند.

سپس همان جست‌وجو از طریق "sp_GetOrderedAdsByProvinceAndBodyStatus" اجرا می‌شود تا نتایج جدید به‌صورت به‌روز نمایش داده شوند.

این رویه به‌خوبی نشان می‌دهد که چگونه می‌توان حذف یک رکورد را با یک جست‌وجوی مجدد ترکیب کرد تا کاربر نتیجه نهایی را بلافاصله مشاهده کند.

```
CREATE PROCEDURE [dbo].[sp_DeleteLowestMileageAdAndShow]
   @ProvinceName NVARCHAR(50),
   @BodyStatus1 NVARCHAR(50),
   @BodyStatus2 NVARCHAR(50) = NULL
AS
BEGIN
   SET NOCOUNT ON;
   BEGIN TRY
      BEGIN TRANSACTION;
      DECLARE @AdIdToDelete INT;
      SELECT TOP 1 @AdIdToDelete = a.ad_id
      FROM [dbo].[Ad] a
      INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
      INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
      INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
      INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
      WHERE p.province_name = @ProvinceName
         AND a.vehicle_type_id = 1
         AND (bc.body_condition_name = @BodyStatus1 OR (@BodyStatus2 IS NOT NULL AND
bc.body_condition_name = @BodyStatus2))
      ORDER BY a.mileage ASC;

      IF @AdIdToDelete IS NOT NULL
      BEGIN
         DELETE FROM [dbo].[PriceEstimate] WHERE [car_id] = @AdIdToDelete;
         DELETE FROM [dbo].[Car] WHERE [ad_id] = @AdIdToDelete;
         DELETE FROM [dbo].[Motor] WHERE [ad_id] = @AdIdToDelete;
         DELETE FROM [dbo].[HeavyVehicle] WHERE [ad_id] = @AdIdToDelete;
         DELETE FROM [dbo].[Ad] WHERE [ad_id] = @AdIdToDelete;
      END

      COMMIT TRANSACTION;

      EXEC sp_GetOrderedAdsByProvinceAndBodyStatus @ProvinceName, @BodyStatus1, @BodyStatus2,
'Cheapest';
      EXEC sp_GetOrderedAdsByProvinceAndBodyStatus @ProvinceName, @BodyStatus1, @BodyStatus2,
'MostRecent';
```

```
        EXEC sp_GetOrderedAdsByProvinceAndBodyStatus @ProvinceName, @BodyStatus1, @BodyStatus2,
'NewestYear';
        EXEC sp_GetOrderedAdsByProvinceAndBodyStatus @ProvinceName, @BodyStatus1, @BodyStatus2,
'LowestMileage';
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
        THROW;
    END CATCH
END
GO
```

تست پرسمان ۵ (حذف کم‌کارکردترین آگهی و اجرای مجدد پرسمان ۴)

```
EXEC sp_DeleteLowestMileageAdAndShow N ,'البرز'N ,'بدون رنگ'N;'یک لکه رنگ'
```

خروجی:

Results | Messages

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

	Brand	Model	Mileage	UserPhone	AdId	Price	Year	Color	City	Status	BodyCondition	description
1	سایپا	تیبا	60000	09122222222	17	300000000.00	2021	مشکی	کرج	کارکرده	یک لکه رنگ	آگهی تست البرز - یک لکه رنگ

پرسمان ۶: موتورسیکلت‌ها بر اساس برند، حجم و قیمت

"sp_GetMotorcyclesByBrandsAndCCAndPrice" برای جست‌وجوی موتورسیکلت‌ها طراحی شده است. این رویه دو برند، حداکثر حجم موتور و حداقل قیمت را به عنوان ورودی دریافت می‌کند.

در این جست‌وجو، فقط آگهی‌های مربوط به نوع وسیله نقلیه موتور سیکلت بررسی می‌شوند. اطلاعاتی مانند شناسه آگهی، شماره تماس، قیمت، امکان خرید اقساطی، تعداد اقساط، تاریخ سررسید، تاریخ تحویل، مبلغ هر قسط، توضیحات، سال تولید، وضعیت نو یا کارکرده، کارکرد، برند، مدل، رنگ، شهر، استان، نوع وسیله، نوع سوخت، گیربکس و ظرفیت موتور استخراج می‌شوند.

در پایان، فقط موتورهایی نمایش داده می‌شوند که برند آن‌ها یکی از دو برند مشخص‌شده باشد، حجم موتور آن‌ها کمتر از حد تعیین‌شده باشد و قیمت آن‌ها بالاتر از مقدار مشخص‌شده باشد. این رویه برای فیلترهای دقیق در حوزه موتورسیکلت کاربردی است.

```
CREATE PROCEDURE [dbo].[sp_GetMotorcyclesByBrandsAndCCAndPrice]
    @Brand1 NVARCHAR(50),
    @Brand2 NVARCHAR(50),
    @MaxCC INT,
    @MinPrice DECIMAL(15,2)
AS
BEGIN
    SET NOCOUNT ON;
    SELECT
        a.ad_id, a.phone_number, a.cash_price, a.is_negitiable, a.installment_count,
        a.installment_due_date, a.delivery_date, a.installment_amount, a.description,
        a.production_year, a.is_new, a.mileage,
        b.brand_name, m.model_name, col.color_name, c.city_name, p.province_name,
        vt.vehicle_type_name, ft.fuel_type_name, g.gearbox_name,
        mot.engine_capacity
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[Motor] mot ON a.ad_id = mot.ad_id
    INNER JOIN [dbo].[FuelType] ft ON mot.fuel_type_id = ft.fuel_type_id
    INNER JOIN [dbo].[Gearbox] g ON mot.gearbox_id = g.gearbox_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    INNER JOIN [dbo].[VehicleType] vt ON a.vehicle_type_id = vt.vehicle_type_id
```

```
    WHERE a.vehicle_type_id = 2
        AND (b.brand_name = @Brand1 OR b.brand_name = @Brand2)
        AND mot.engine_capacity < @MaxCC
        AND a.cash_price > @MinPrice
    ORDER BY b.brand_name, a.cash_price;
END
GO
```

تست پرسمان ۶ (موتورسیکلت‌های تی وی اس یا استلز، حجم < ۱۰۰۰، قیمت > ۳۰۰ میلیون)

```
EXEC sp_GetMotorcyclesByBrandsAndCCAndPrice N 'استلز', ۱۰۰۰, N'تی وی اس', N
;۳۰۰۰۰۰۰۰۰
```

خروجی:

Results | Messages

	ad_id	phone_number	cash_price	is_negitiable	installment_count	installment_due_date	delivery_date	installment_amount	description	production_year	is_new	mileage	brand_name	model_name	color_name	city_na
1	13	09122222222	400000000.00	1	NULL	NULL	NULL	NULL	استلز 900 مدل 2024	2024-01-01	0	3000	استلز	استلز 900	مشکی	صفهان
2	12	09121111111	350000000.00	1	NULL	NULL	NULL	NULL	تی وی اس 800 مدل 2023	2023-01-01	0	5000	تی وی اس	تی وی اس 800	سفید	تهران

Results | Messages

	delivery_date	installment_amount	description	production_year	is_new	mileage	brand_name	model_name	color_name	city_name	province_name	vehicle_type_name	fuel_type_name	gearbox_name	engine_capacity
1	NULL	NULL	استلز 900 مدل 2024	2024-01-01	0	3000	استلز	استلز 900	مشکی	اصفهان	اصفهان	Motorcycle	Petrol	Manual	900
2	NULL	NULL	تی وی اس 800 مدل 2023	2023-01-01	0	5000	تی وی اس	تی وی اس 800	سفید	تهران	تهران	Motorcycle	Petrol	Manual	800

پرسمان ۷: کف و سقف قیمت تخمینی خودرو

"sp_GetPriceRangeByBrandMileageBody" برای محاسبه بازه قیمت تخمینی خودروها استفاده می‌شود. این رویه نام برند، کارکرد و وضعیت بدنه را دریافت می‌کند و سپس بر اساس جدول "PriceEstimate" کمترین و بیشترین قیمت تخمینی را محاسبه می‌نماید.

در این رویه، جدول‌های "Ad"، "BodyCondition", "Car", "Brand"" و "PriceEstimate" با یکدیگر JOIN شده‌اند. خروجی شامل:

حداقل قیمت، حداکثر قیمت، تعداد خودروهای منطبق با شرایط ورودی است.

این رویه برای تحلیل بازار و بررسی حدود قیمت خودروهای هم‌رده مفید است.

```
CREATE PROCEDURE [dbo].[sp_GetPriceRangeByBrandMileageBody]
   @BrandName NVARCHAR(50),
   @Mileage INT,
   @BodyStatusName NVARCHAR(50)
AS
BEGIN
   SET NOCOUNT ON;
   SELECT
      MIN(pe.estimate_price) AS MinPrice,
      MAX(pe.estimate_price) AS MaxPrice,
      COUNT(*) AS CarCount
   FROM [dbo].[Ad] a
   INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
   INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
   INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
   INNER JOIN [dbo].[PriceEstimate] pe ON car.ad_id = pe.car_id
   WHERE b.brand_name = @BrandName
      AND a.mileage = @Mileage
      AND bc.body_condition_name = @BodyStatusName
      AND a.vehicle_type_id = 1;
END
GO
```

تست پرسمان ۷ (کف و سقف قیمت بنز، کارکرد ۱۰۰۰۰، وضعیت بدون رنگ)

```
EXEC sp_GetPriceRangeByBrandMileageBody N ,۱۰۰۰۰ ,'بنز'N;'بدون رنگ'
```

خروجی:

Results | Messages

	MinPrice	MaxPrice	CarCount
1	3400000000.00	3600000000.00	2

پرسمان ۸: برندهای خودروهای ساخت کشور مشخص که فقط در دو شهر خاص آگهی شده‌اند

"sp_GetBrandsByCountryAndCities" برای استخراج برندهایی طراحی شده است که خودروهای آن‌ها از یک کشور مشخص هستند و فقط در دو شهر خاص آگهی شده‌اند.

این رویه نام کشور و دو شهر را به عنوان ورودی دریافت می‌کند و سپس برندهایی را برمی‌گرداند که خودروهای مربوط به آن‌ها در آن کشور ثبت شده‌اند و در شهرهایی غیر از دو شهر ورودی، آگهی نداشته باشند.

در این بخش از "GROUP BY" و "HAVING" استفاده شده تا شرط‌های تجمیعی به‌درستی اعمال شوند.

این نوع جست‌وجو برای تحلیل توزیع جغرافیایی آگهی‌ها و یافتن الگوهای مکانی کاربردی است.

```
CREATE PROCEDURE [dbo].[sp_GetBrandsByCountryAndCities]
   @CountryName NVARCHAR(50),
   @City1 NVARCHAR(50),
   @City2 NVARCHAR(50)
AS
BEGIN
   SET NOCOUNT ON;
   SELECT b.brand_id, b.brand_name
   FROM [dbo].[Brand] b
   INNER JOIN [dbo].[Ad] a ON b.brand_id = a.brand_id
   INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
   INNER JOIN [dbo].[Country] cnt ON car.country_id = cnt.country_id
   INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
   WHERE cnt.country_name = @CountryName
   GROUP BY b.brand_id, b.brand_name
   HAVING COUNT(DISTINCT CASE WHEN c.city_name NOT IN (@City1, @City2) THEN c.city_id END) = 0
      AND COUNT(DISTINCT c.city_name) > 0;
END
GO
```

تست پرسمان ۸ (برند خودروهای آلمانی که فقط در تهران یا ساری آگهی شده‌اند)

```
EXEC sp_GetBrandsByCountryAndCities N ,'آلمان'N ,'تهران'N;'ساری'
```

خروجی:

Results | Messages

	brand_id	brand_name
1	4	مرسدس
2	14	بنز
3	15	آئودی

پرسمان ۹: مشخصات خودروهایی با رنگ غیر از C، شاسی و سوخت مشخص در استان مشخص

"sp_GetCarsByExcludedColorChassisFuelProvince" برای نمایش خودروهایی استفاده می‌شود که رنگ آن‌ها با یک رنگ خاص متفاوت است و هم‌زمان دارای شاسی، نوع سوخت و استان مشخصی هستند.

در این رویه، اطلاعات کامل خودرو از جدول‌های "Ad"، "Brand", "Model", "Car", "CarType", "BodyCondition", "Color", "City", "Province", "Country", "Motor", "FuelType" و "Gearbox" استخراج می‌شود.

شرایط جست‌وجو شامل موارد زیر است:

نوع خودرو باید سواری باشد،

رنگ خودرو نباید برابر با رنگ حذف‌شده باشد،

نوع شاسی باید با مقدار ورودی منطبق باشد،

نوع سوخت باید با مقدار ورودی برابر باشد،

استان باید با مقدار مشخص‌شده یکی باشد.

این رویه برای فیلترهای ترکیبی و دقیق مفید است و امکان جست‌وجوی پیشرفته را فراهم می‌کند.

```
CREATE PROCEDURE [dbo].[sp_GetCarsByExcludedColorChassisFuelProvince]
```

```
    @ExcludedColorName NVARCHAR(50),
    @ChassisTypeName NVARCHAR(50),
    @FuelTypeName NVARCHAR(50),
    @ProvinceName NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    SELECT
        a.ad_id, a.phone_number, a.cash_price, a.mileage, a.production_year, a.is_new,
        b.brand_name, m.model_name, col.color_name, c.city_name, p.province_name,
        ct.car_type_name, ft.fuel_type_name, g.gearbox_name, bc.body_condition_name,
        car.trim, car.interior_color_id, cnt.country_name
    FROM [dbo].[Ad] a
    INNER JOIN [dbo].[Brand] b ON a.brand_id = b.brand_id
    LEFT JOIN [dbo].[Model] m ON a.model_id = m.model_id
    INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
    INNER JOIN [dbo].[CarType] ct ON car.car_type_id = ct.car_type_id
    INNER JOIN [dbo].[BodyCondition] bc ON car.body_condition_id = bc.body_condition_id
    INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
    INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
    INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
    INNER JOIN [dbo].[Country] cnt ON car.country_id = cnt.country_id
    INNER JOIN [dbo].[Motor] mot ON a.ad_id = mot.ad_id
    INNER JOIN [dbo].[FuelType] ft ON mot.fuel_type_id = ft.fuel_type_id
    LEFT JOIN [dbo].[Gearbox] g ON car.gearbox_id = g.gearbox_id
    WHERE a.vehicle_type_id = 1
        AND col.color_name != @ExcludedColorName
        AND ct.car_type_name = @ChassisTypeName
        AND ft.fuel_type_name = @FuelTypeName
        AND p.province_name = @ProvinceName
    ORDER BY a.ad_id;
END
GO
```

تست پرسمان ۹ (مشخصات خودروهای غیر سفید، شاسی‌بلند، هیبریدی، استان اصفهان) – قبل از تغییر رنگ

```
EXEC sp_GetCarsByExcludedColorChassisFuelProvince N ,'سفید'N ,'شاسی‌بلند'
N ,'هیبرید'N;'اصفهان'
```

خروجی:

Results | Messages

	ad_id	phone_number	cash_price	mileage	production_year	is_new	brand_name	model_name	color_name	city_name	province_name	car_type_name	fuel_type_name
1	21	09125555555	1200000000.00	15000	2022-01-01	0	تویوتا	تویوتا کرولا	قرمز	اصفهان	اصفهان	شاسی‌بلند	هیبرید
2	22	09126666666	1300000000.00	10000	2023-01-01	0	تویوتا	تویوتا کرولا	آبی	اصفهان	اصفهان	شاسی‌بلند	هیبرید
3	23	09127777777	1100000000.00	25000	2021-01-01	0	تویوتا	تویوتا کرولا	مشکی	اصفهان	اصفهان	شاسی‌بلند	هیبرید
4	24	09128888888	1400000000.00	5000	2024-01-01	0	تویوتا	تویوتا کرولا	سبز	اصفهان	اصفهان	شاسی‌بلند	هیبرید

Results | Messages

	er	cash_price	mileage	production_year	is_new	brand_name	model_name	color_name	city_name	province_name	car_type_name	fuel_type_name	gearbox_name	body_condition_name	trim	interior_color_id	country_name
1	5	1200000000.00	15000	2022-01-01	0	تویوتا	تویوتا کرولا	قرمز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن
2	6	1300000000.00	10000	2023-01-01	0	تویوتا	تویوتا کرولا	آبی	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن
3	7	1100000000.00	25000	2021-01-01	0	تویوتا	تویوتا کرولا	مشکی	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Good	NULL	NULL	ژاپن
4	8	1400000000.00	5000	2024-01-01	0	تویوتا	تویوتا کرولا	سبز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن

پرسمان ۱۰: تغییر رنگ خودروها و اجرای مجدد پرسمان۹

"sp_UpdateCarColorAndRerunQuery9" برای تغییر رنگ خودروها و سپس اجرای دوباره پرسمان نهم طراحی شده است. این رویه یک رنگ قدیمی، یک رنگ جدید، نوع شاسی، نوع سوخت و نام استان را دریافت می‌کند.

در ابتدا، رنگ خودروهایی که با رنگ قدیمی منطبق هستند به رنگ جدید تغییر داده می‌شود. این کار در قالب یک تراکنش انجام می‌گیرد تا در صورت بروز خطا، داده‌ها دچار ناسازگاری نشوند.

پس از ثبت تغییرات، رویه "sp_GetCarsByExcludedColorChassisFuelProvince" دوباره اجرا می‌شود تا نتایج جدید نمایش داده شوند.

این ساختار نشان می‌دهد که چگونه می‌توان یک عملیات به‌روزرسانی را با یک جست‌وجوی تحلیلی ترکیب کرد تا اثر تغییرات بلافاصله مشاهده شود.

```
CREATE PROCEDURE [dbo].[sp_UpdateCarColorAndRerunQuery9]
    @OldColorName NVARCHAR(50),
    @NewColorName NVARCHAR(50),
    @ChassisTypeName NVARCHAR(50),
    @FuelTypeName NVARCHAR(50),
    @ProvinceName NVARCHAR(50)
AS
```

```
BEGIN
   SET NOCOUNT ON;
   BEGIN TRANSACTION;
   BEGIN TRY
      UPDATE a
      SET a.color_id = (SELECT color_id FROM [dbo].[Color] WHERE color_name = @NewColorName)
      FROM [dbo].[Ad] a
      INNER JOIN [dbo].[Car] car ON a.ad_id = car.ad_id
      INNER JOIN [dbo].[CarType] ct ON car.car_type_id = ct.car_type_id
      INNER JOIN [dbo].[Motor] mot ON a.ad_id = mot.ad_id
      INNER JOIN [dbo].[FuelType] ft ON mot.fuel_type_id = ft.fuel_type_id
      INNER JOIN [dbo].[City] c ON a.city_id = c.city_id
      INNER JOIN [dbo].[Province] p ON c.province_id = p.province_id
      INNER JOIN [dbo].[Color] col ON a.color_id = col.color_id
      WHERE a.vehicle_type_id = 1
         AND col.color_name = @OldColorName
         AND ct.car_type_name = @ChassisTypeName
         AND ft.fuel_type_name = @FuelTypeName
         AND p.province_name = @ProvinceName;

      COMMIT TRANSACTION;
      EXEC sp_GetCarsByExcludedColorChassisFuelProvince @NewColorName, @ChassisTypeName,
@FuelTypeName, @ProvinceName;
   END TRY
   BEGIN CATCH
      IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;
      THROW;
   END CATCH
END
GO
```

تست پرسمان ۱۰ (تغییر رنگ مشکی به سفید و اجرای مجدد پرسمان ۹)

```
EXEC sp_UpdateCarColorAndRerunQuery9 N ,'مشکی'N ,'سفید'N ,'شاسی‌بلند'N ,'هیبرید'
N;'اصفهان'
```

خروجی:

Results | Messages

	ad_id	phone_number	cash_price	mileage	production_year	is_new	brand_name	model_name	color_name	city_name	province_name	car_type_name	fuel_type_name	gearbox_name
1	21	09125555555	1200000000.00	15000	2022-01-01	0	تویوتا	تویوتا کرولا	قرمز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic
2	22	09126666666	1300000000.00	10000	2023-01-01	0	تویوتا	تویوتا کرولا	آبی	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic
3	24	09128888888	1400000000.00	5000	2024-01-01	0	تویوتا	تویوتا کرولا	سبز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic

Results | Messages

	er	cash_price	mileage	production_year	is_new	brand_name	model_name	color_name	city_name	province_name	car_type_name	fuel_type_name	gearbox_name	body_condition_name	trim	interior_color_id	country_name
1	5	1200000000.00	15000	2022-01-01	0	تویوتا	تویوتا کرولا	قرمز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن
2	6	1300000000.00	10000	2023-01-01	0	تویوتا	تویوتا کرولا	آبی	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن
3	8	1400000000.00	5000	2024-01-01	0	تویوتا	تویوتا کرولا	سبز	اصفهان	اصفهان	شاسی‌بلند	هیبرید	Automatic	Excellent	NULL	NULL	ژاپن

با تشکر از توجه شما