**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**View**

یک View یک جدول مجازی است که نتیجه حاصل از عبارات و دستورات SQL می باشد.

یک view شامل سطر و ستونهایی دقیقا شبیه یک جدول واقعی است. فیلدها در یک View فیلدهایی هستند که از یک یا چند جدول واقعی در پایگاه داده گرفته شده اند.

شما می توانید توابع موجود در SQL، عبارت WHERE و JOIN را به یک View اضافه کنید و داده های نتیجه را به همان صورت که داده ها در یک جدول آورده می شوند، نشان دهید.

* View دارای پارامتر ورودی نیست.

|  |
| --- |
| CREATE VIEW GetList  AS  SELECT Categories.CatName, Forums.ID, Forums.ForumsName  FROM Categories INNER JOIN Forums ON Categories.ID = Forums.CatID |
| CREATE VIEW GetList  AS  SELECT \* FROM Forums WHERE CatID = 1 |

**طریقه استفاده از GetList**

|  |
| --- |
| SELECT \* FROM GetList |

**حذف کردن GetList**

|  |
| --- |
| Drop VIEW GetList |

**قفل کردن GetList**

|  |  |
| --- | --- |
|  | CREATE VIEW GetList  WITH ENCRYPTION  AS  SELECT \* FROM Forums  WHERE CatID = 1 |

* در صورت قفل کردن GetList دیگر نمی توان به کد ها دسترسی پیدا کرد.
* جهت ویرایش View از دستور Alter به جای CREATE استفاده کنید.

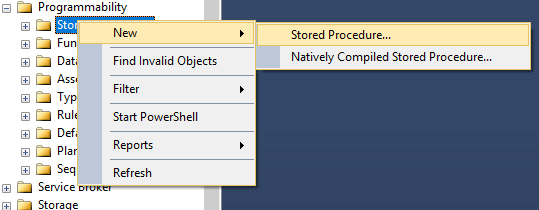
**Stored Procedures چیست ؟**

مجموعه ای از دستورات (Query) مربوط به SQL میباشد که به عنوان یک واحد با یکدیگر Compile می شوند .

Stored Procedure می تواند شامل چندین دستور باشد, مانند دستورات Select, INSERT, UPDATE, DELETE

**فواید استفاده از Stored Procedure :**

* استفاده از Store Procedure زمان Compile را در Overhead کاهش میدهد, همچنین در جلوگیری از دستورات مخرب   
  ( SQL Injection) ما را یاری می کند .
* یکبار آن را ایجاد کرده و به دفعات متعدد می توانیم از آن استفاده کنیم
* کاهش ترافیک (حجم دستورات)
* می تواند قسمتی ویژه از دستورات را به کاربری خاص واگزار کند .
* Stored Procedure را می توان به راحتی در قسمتی معین تغییر داد بدون تغییر هیچ دستوری در نرم افزار
* براحتی در Stored Procedure می توانیم از Transaction استفاده کنیم .
* تعریف یک Stored Procedure به صورت زیر است



**مثال**

|  |  |
| --- | --- |
| CREATE PROC GetList  AS  SELECT \* FROM Forums | CREATE Procedure GetList  AS  SELECT \* FROM Forums |
| CREATE Procedure GetList  (  @ID INT=0,  @CatName NVARCHAR(50)  )  AS  SELECT Categories.CatName, Forums.ForumsName, Forums.Subject  FROM Categories INNER JOIN Forums ON Categories.ID = Forums.CatID  WHERE (Categories.CatName = @CatName) | |

**طریقه استفاده از GetList**

|  |  |  |
| --- | --- | --- |
| EXEC GetList 0,N'SoftWare' | EXEC GetList | EXECUTE GetList |
| EXEC ('Select \* from Categories) | | |

**حذف کردن GetList**

|  |
| --- |
| Drop Procedure GetList |

**قفل کردن GetList**

|  |  |
| --- | --- |
|  | CREATE Procedure GetList  WITH ENCRYPTION  AS  SELECT \* FROM Forums |

* در صورت قفل کردن GetList دیگر نمی توان به کد ها دسترسی پیدا کرد.

**مثال Stored Procedures**

|  |
| --- |
| DECLARE @D NVARCHAR(max)='';  SET @D = 'SELECT Categories.CatName, Forums.ForumsName, Forums.Subject FROM Categories INNER JOIN Forums ON Categories.ID = Forums.CatID'  exec sp\_executesql @D |
| CREATE Procedure GetList(@ID INT)  AS  DECLARE @T NVARCHAR(50)='';  SELECT @T=CatName FROM Categories WHERE ID=@ID  SELECT @T |
| CREATE Procedure GetList(@ID INT)  AS  IF @ID >0  SELECT CatName FROM Categories WHERE ID=@ID  ELSE  SELECT ID FROM Categories |
| CREATE Procedure GetList(@ID INT)  AS  SELECT CASE @ID WHEN 1 THEN '1-1' WHEN 2 THEN '2-2' ELSE 'Not' END |
| CREATE Procedure GetList(@ID INT,@Text NVARCHAR(40)='')  AS  SELECT @ID,@Text  **دستور EXEC GetList 3** چون مقدار پیشفرض گزاشته ایم نیازی به مقدار دهی نیست. |

* جهت ویرایش Procedure از دستور Alter به جای CREATE استفاده کنید.

|  |
| --- |
| Alter Procedure GetList(@ID INT,@Text NVARCHAR(40)='')  AS  SELECT @ID,@Text |

**کاربرد:**گاهی ما نیاز داریم که چند تا کار رو همیشه انجام بدیم. مثلا برای ثبت سفارش اول تست کنیم ببینیم شرایط خاصی وجود داره یا نه، بعد جدولی رو آپدیت کنیم، بعد سلکت کنیم. خوبی SPاینه که در اون می تونیم شرط - حلقه - تراکنش - فراخوانی یک SP دیگه - ساخت گزارش وابسته به پارامتر و ... داشته باشیم.

**Natively Compiled Stored Procedures**

SP های معمولی برای کار با جداول معمولی هست ولی Native compiled SP ها برای کار با جداول InMemory هستند و برای آن کار کاملا بهینه شده اند و سرعت پردازش بالایی دارند .