آموزش SQL Server (قسمت اول)

مفاهیم ابتدائی

مقدمه :

کاربرد روز افزون بانک اطلاعاتی SQL مرا بر آن داشت تا مطالبی هر چند کوتاه جهت خوانندگان محترم سایت تهیه نماییم. قبلا از هر چیز لازم به ذکر است که مطالب ذیل در حد آشنایی بوده و دوستا برای دستیابی به تکنیکهای بیشتر می بایست از کتابهای مرجع و Book online خود SQL Server استفاده نمایند. در مطالب زیر که سلسله وار مباحث SQL Server را مرور خواهیم کرد ، سعی شده تا ابتدا مطالب مقدماتی جهت آشنایی آورده شود و سپس اگر عمری باقی بود به مطالب پیشرفته آن بپردازیم. همچنین برای یادآوری خدمت دوستان ابتدا مرور سریعی بر چند دستور SQL که کاربرد بیشتری دارند خواهیم پرداخت و سپس به SQL Server و مطالب آن خواهیم پرداخت . مطالب زیر اکثرا از کتاب Implementation Training Microsoft SQL Server 7.0 Database انتخاب گردیده است . این کتاب به همراه CDآموزش آن به عنوان یک مرجع برای امتحانات مایکروسافت استفاده می‌شود.

جداول بکار رفته نیز همگی در SQL Server 7.0 در Database Northwind موجود هستند.

دستور Select

این دستور که دستوری مستقل نیست و حتما باید با اجزایی بکار رود جهت ساخت پرس و جو بر روی بانک اطلاعاتی بکار می‌رود و رکوردهایی که با شرایط این دستور همخوان باشد به عنوان نتیجه پرس و جو برمی‌گرداند . چهار کلمه کلیدی وجود دارند که بخشهای ارزشمند این دستور را تشکیل می‌دهند :

1- select

2- from

3- where

4- order by

شکل کلی دستور :

Select [*|distinct column1, column2,...]

From table[,table2,...]

Where شرط

Order by نام فیلد یا شماره فیلد

مثال :

Select * from customers

این دستور تمام رکوردهای جدول customers را برمی‌گرداند.

که نتیجه 91 سطر از اطلاعات این جدول خواهد بود

حال اگر شرط Country ='uk' اضافه کنیم ، فقط اطلاعات مشتریان انگلیس جواب خواهند بود که به 7 سطر تقلیل می‌یابد.

select * from customers

where Country ='uk'

حال

select City,Country from customers

order by city

فقط ستونهای نام شهر (city) و نام کشور (Country) را بر گردانده و بر اساس نام شهر مرتب میکند.
دستور بالا با دستور پایین هردو یک جواب را میدهند :

select City,Country from customers

order by 1

که 91 سطر بازگردانده خواهد شد . در نتیجه پرس و جو تعدادی سطر تکراری وجود دارد مانند شهر

London که اگر از کلمه Distinct در Select استفاده کنیم این سطرهای تکراری حذف خواهد شد .

select distinct City,Country from customers

1 order by

و جواب 69 سطر خواهد بود.

استفاده از توابع در Select

1- Count : تعداد سطرهای بازگردانده شده توسط select را میشمارد.

Count(*) from Customers Select

'where Country ='uk

در اصل تعداد مشتریانی را میشمارد که در کشور انگلیس هستند. که عدد 7 جواب است.

2- Sum : مجموع یک فیلد عددی را برمی‌گرداند.

[Details Select sum(Quantity) from [Order

where productid = 11

مجموع فیلد Quantity را برای فیلدهایی که شماره محصول آنها (Productid) برابر 11 است را محاسبه میکند

نکته 1 : در دستور select می‌توان از اسم مستعار استفاده کرد ، یعنی نام جدیدی را برای یک ستون در نظر گرفت به عنوان مثال select قبل را به شکل زیر بکار برد :

Select sum(Quantity) as Sum_QTY

[from [Order Details

where productid = 11

که Sum_QTY یک اسم مستعار برای مجموع است. استفاده از کلمه کلیدیas اختیاری است.

نکته 2 : در دستور select هرگاه اسم فیلدی اسم خاص باشد و یا فاصله بین اسم باشد مثل Order Details که فاصله بین اسم جدول است حتماً از علامت براکت [] میبایست استفاده کرد.

نکته 3 : استفاده از group by :

هنگامی که از توابع count و Sum به همراه یک فیلد دیگر در دستور select استفاده می‌شود از group by استفاده می‌کنیم .

به عنوان مثال دستور زیر جمع مقادیر فیلد Quantity را برای هر شماره محصول محاسبه میکند .

```
Select productid, sum(Quantity) as sum_qty
[from [Order Details]
group by productid
```

که نتیجه مانند زیر خواهد بود :

```
productid sum_qty
---------- ----------
603 61
328 3
297 32
301 6
981 41
740 64
95 9
344 12
```

در صورتیکه دستور ordr by 1 بعد از group by استفاده کنیم نتیجه بر اساس کد محصول مرتب

خواهد شد.

نکته 4 : دستور where می تواند خود شامل یک دستور select باشد :

```
select * from Products

where ProductID in

(70< details] where Quantity select distinct ProductID from [order )

order by ProductID
```

تنها نکته ای که می بایست توجه کرد این است که نام فیلدی که در شرط آورده می شود حتما در دستور select آورده شود، به عبارت دیگر select درون شرط تنها یک ستون را می بایست برگرداند .

تمرین : با فرض اینکه دو جدول Products و order details دارای ستون (فیلد) یکسان ProductID هستند ، یک دستور Select بنویسید که تمام فیلدهایی از Products را نشان دهد که فیلد ProductID آن با ProductID جدول order details یکی باشد.؟

حل :

```
Select pr.×

details] as od From Products as pr , [order

Where pr. ProductID = od. ProductID
```

قابل به ذکر است که بیش از 90٪ از کارهایی که ما برروی جداول انجام می دهیم با select و ترکیبات آن انجام می شود. لذا بدست آوردن تبحر در نوشتن select ها می تواند شما را در تهیه برنامه ها یاری کند.

3- Min,max : بیشترین و کمترین مقدار فیلد را در بانک اطلاعاتی بدست می دهد.

```
(Select min (Quantity

[Order Details] from
```

4- Top n : تعداد n سطر اول بانک اطلاعاتی را برمی گرداند.

Select top 5 ×

from [Order Details]

5 سطر اول بانک را برمی گرداند.

نکته 3 : در حالت بالا اگر مقدار سطر 5 و 6 یکی باشد فقط سطر 5 جواب خواهد بود برای گریز از این حالت از شکل زیر در این دستور استفاده میکنیم :

Select top n with ties ×

From table

5- Into

Select * from table1 into table2

اطلاعات table1 را به table2 کپی میکند. البته table2 باید از قبل وجود نداشته باشد.

این دستور خود table2 را میسازد.

دستور select قویترین و کاربردی ترین دستور در sql است که خود ماهها نیازمند تمرین و آموزش است . برای اطلاعات بیشتر به online books خود Sql Server مراجع کنید.

دستور Delete

برای حذف اطلاعات از یک بانک اطلاعاتی استفاده میشود.

شکل کلی دستور :

Delete table1

Where شرط

مثال :

فرض کنید جدول authors موجود باشد و فیلد کلید آن au_id باشد. برای حذف 10 سطر اول این جدول از دستور زیر استفاده می کنیم :

```
DELETE authors
FROM (SELECT TOP 10 * FROM authors) AS t1
WHERE authors.au_id = t1.au_id
```

دستور insert

برای اضافه کردن اطلاعات به یک جدول از این دستور استفاده میشود.

```
Insert into table1 (f1,f2,...)
Values (v1,v2,...)
```

که f1,f2 نام فیلدها و v1,v2 مقادیر آنها میباشد.

البته میتوانید مقادیر را نتیجه یک select قرار داد.

مثال :

```
Insert into table1
Select top 10
table2 From
```

مقدار 10 سطر اول را از table2 را در table1 درج میکند. البته باید تعداد فیلدها یکی باشد. در غیر اینصورت از values استفاده کنید.

آموزش SQL Server (قسمت دوم)

آموزش Enterprise Manager

میتوان گفت قلب Sql Server است. در Manager Enterprise شما میتوانید یک اتصال به سرور Sql برقرار کنید . table بسازید. User تعریف کنید و.....

علامت فلش سبز نمایانگر فعال بودن سرور است.
سرور میتواند local باشد مانند بالا و یا یک Sqlserver باشد برروی یک سرور . برای ایجاد یک سرور جدید یا به عبارت دیگر وصل شدن client (ویندوز 98) به یک سرور دیگر بر روی یکی Microsoft Sql Server یا SQL Server Group و یا برروی سرور موجود کلیک سمت راست کرده و گزینه New Sql Server Registratin را انتخاب کنید. سپس کلید next را انتخاب کنید. سپس در منوی بعدی در available Server نام سرور خود را تایپ کنید . (نام سرور SQL خود را که برروی ویندوز 2000 خود نصب کرده اید) و بعد کلید add را فشار دهید.و گزینه next را انتخاب کنید.

در پتجره بعد از شما سوال میکند که آیا از username ویندوز استفاده کند و یا اینکه از یک username مخصوص خود Sql Server استفاده کنید. گزینه دوم را انتخاب کرده و سپس Login Name و Password را وارد کنید. (در حالت پیش فرض sa بعنوان login name و فضای خالی بجای Password وارد کنید.) پس از چند بار فشار کلید next شروع به وصل شدن به Server میکند. در صورت موفقیت آمیز بودن با پیامی این کار را اطلاع میدهد.
از دیگر گزینه ها شما بیشترین استفاده را از Databases خواهید کرد.
به عبارت دیگر هر کار و پروژه ما بعنوان یک Database در سرور sql قرار میگرد. همه جداول و دستورات مربوط به آنها در این محل نگهداری می شود.
با کلیک سمت راست برروی Databases و انتخاب New Database... میتوانید یک Database جدید برای خودتان بسازید.

پس از انتخاب نام آن را تایید کنید.
هر Database شامل موارد زیر است :
1- Diagram : ارتباط جداول را نشان میدهد.
2- Tables : جداول پروژه را نشان میدهد.
3- Views : دیدهای پروژه را نشان میدهد.
4- Stored Procedure : کدهای sql مربوط به عملیاتهای روی جداول را نگهداری میکند.
5- Users : کاربران تعریف شده بر روی این database را نشان میدهد.
6- Roles: قوانین دسترسی به جداول و پروسیجرها را نشان میدهد.
7- Rules : قوانین مربوط به داده ها را در جداول نشان میدهد.

گزینه های 7 به بعد کاربرد آنچنانی برای کارهای ابتدایی ندارند

ایجاد یک جدول جدید :
برای ایجاد یک جدول جدید بروی tables کلیک سمت راست کرده و گزینه New Table را انتخاب کنید. سپس در کادر بعدی نام جدول را انتخاب کنید. حال فیلدها و نوع آنها را مشخص کنید .
بعد از مشخص کردن نوع و احتمالاً طول فیلد ، باید مشخص کنید که آیا فیلد همیشه می بایست مقدار داشته باشد و یا می تواند null باشد. Allow Nulls اگر تیک داشته باشد یعنی اینکه این فیلد می تواند هیچ مقداری به خود اختصاص ندهد.

تذکر : مقدار null را با فضای خالی اشتباه نگیرید.

در قسمت Value Default مقدار اولیه برای فیلد وارد کنید. تا در صورتیکه هیچ مقداری درج نشد آن مقدار درج شود. (در دستور insert)
اگر Identity را تیک بزنید این فیلد بشکل خود افزا خواهد شد که اولا باید نوع فیلد عددی و ثانیا مقدار گامها در Identity increment مشخص شود. مقدار اولیه آن را هم می توانید در Identity Seed قرار دهید. بدین شکل با این مقدار شروع و با گامهای مشخص شده حرکت خواهد کرد.

تذکر : هیچ مقداری در این فیلد نمی توانید درج کنید ، چراکه خود سیستم این مقدار را تولید می کند .

برای مشخص کردن فیلد کلید (یا فیلدهای کلیدی) فیلد(ها) را انتخاب و بر روی علامت کلید بر روی Toolbar کلیک کنید .

تذکر : فیلد کلیدی نمی تواند Nulls Allow باشد.
تذکر : برای تعریف index کلیک سمت راست کرده و index/keys را انتخاب کنید .
(در مورد index بطور مفصل صحبت خواهد شد)

ایجاد Procedure Stored :
مانند هر زبان دیگر رویه ها در sqlServer نیز موجود می باشند. و بکار میروند. سه رویه , insert Updaet و Delete را می توانید براحتی با ویزاردهای خود SqlServer بسازید. از اینجا به بعد از واژه SP بجای رویه استفاده خواهیم کرد.
ابتدا با ساختار کلی SP آشنا شده و سپس به ویزارد موجود خواهیم پرداخت.
برای ایجاد یک SP جدید ابتدا برروی گزینه Stored Procedure کلیک سمت راست کرده و گزینه new Stored Procedure... را انتخاب کنید. در پنجره بعدی شما می توانید متن SP را وارد کنید.

1- نام sp : ابتدا بجای [PROCEDURE NAME] یک نام برای SP خود در نظر بگیرید. من خود از ساختار زیر بدین منظور استفاده میکنیم : عملیات_نام جدول

عنوان مثال اگر نام جدول Sale و عملیات مورد نظر یک عملیات insert باشد نام SP را Sale_INSERT میگذاریم. بهتر است نام عملیات با حروف بزرگ تایپ شود . البته بعضی از دوستان از سه حرف عملیات استفاده می کنند. برای مثال بالا خواهیم داشت :Sale_INS .

2- تعریف پارامترها : برای تعریف پارامترهای ورودی SP قبل از کلمه کلیدی As آنها را داخل پرانتز مشخص کنید. بدین شکل که ابتدا علامت @ سپس نام پارامتر بعد فاصله و نوع پارامتر.
تذکر : تمامی متغییرها در SP از ساختار نام متغییر@ پیروی میکنند.
بعنوان مثال فرض کنید یک SP دارای دو پارامتر با نامهای Id از نوع int و Name از نوع Varchar(20) باشد ، داریم :(@name@ varchar(20 id int,@name))
حال بعد از As دستورات مورد نظر را تایپ میکنیم :

مثال 1 : SP بنویسید که چهار حرف اول فیلد LastName و فیلد FirstName را از جدول Employees انتخاب کند به شرطی که فیلد LastName با حرف A شروع شود ؟
ابتدا برروی procedure Stored کلیک سمت راست کرده و گزینه New stored procedure را انتخاب کرده و خطوط زیر را تایپ کنید. بعضی از دستورات پایین در کادر باز شده موجود هستند که نیازی به تایپ مجدد آنها نیست .

```
create Stored Procedure Employees_BROWSE
As
Select substring(Lastname,1,4) as LastNmae,FirstName
From Employees
'Where LastName Like '%A
```

حال بر روی دکمه ok کلیک نمایید. حال SP با نام Employees_BROWSE در لیست SP اضافه شده است .
مثال 2 : تمام فیلدهای Employees را انتخاب کنید که فیلد BirthDate در یک بازه تاریخی که به شکل پارامتر وارد میشود قرار گیرد ؟

```
create Stored Procedure Employees_BROWSE2
(@StartDate char(10) , @EndDate char(10))
As
× Select
Employees From
Where BirthDate between @StartDate and @EndDate
```

حال برای اجرای SP ها در Query Analyzer کافی است بنویسیم : نام Exec SP
بعنوان مثال :Exec Employees_browse
اگر پارامتر داشت مقدار پارامترها را هم می آوریم :

```
'Exec Employees_Browse2 '01/01/1940' , '15/06/1955
```

3- تعریف متغیرها : برای تعریف متغیرها می بایست از کلمه کلیدی Declare استفاده کنیم. بعنوان مثال : Declare @myname varchar(50)
متغییر @myname از نوع کارکتری پویا تعریف میکند.

4- برگرداندن کد خطا : بدین منظور از دستور return @@Error
گرچه هرمقداری را که بخواهیم می توانیم با دستور return برگردانیم.

5- تعریف پارامترها که مقداری را برمی گردانند : برای این منظور هنگامی که پارامتر را در ابتدای پروسیجر تعریف میکنیم بعد از نوع آن از کلمه کلیدی output می کنیم بعنوان مثال :
create Stored Procedure Employees_BROWSE2
(@StartDate char(10) ، @pp varchar(10) output , @EndDate char(10))
As
6- مقدار دهی به متغییرها : به دو روش می توانید این کار را بکنید یکی با دستور Set و دیگری با دستور Select .
7- دستورات شرطی :مانند دیگر زبانها شما می توانید در Sql دستورات شرطی را بکار ببرید. ساختار آن به شکل زیر است :
If شرط then دستور
If شرط then
Begin
دستور1
دستور2
...
end
مثال :

If @Sdate < @Sdate = @newDate set Edate

8- دستور Set NOCOUNT on : این دستور از نوشت تعداد سطرهای برگردانده شده توسط دستورات جلوگیری میکند. کاربرد مهم آن زمانی است که شما چند کار را پشت سرهم در SP انجام میدهید. مثلا یک جدوی موقت میسازید و سپس از آن یک تعدادی از فیلدها را با دستور Select انتخاب می کنید. اگر این دستور را در ابتدای SP استفاده نکتید ، هنگامی که میخواهید از آن SP در یک زبان برنامه نویسی استفاده کنید با خطای SP هیچ dataset ی برنمی گرداند مواجه خواهید شد.

تذکر1 : شما می توانید یک SP را در یک SP دیگر فراخوانی کنید. برای این منظور همانطور که قبلا گفته شد از دستور exec استفاده نمایید.
تذکر 2 : با دستور exec شما می توانید یک دستور sql را نیز اجرا کنید. این کار زمانی بکار می آید که دستور مورد نظر پویا و متغیر باشد.

مثال :

```
Exec (" select * From @Shart+"Where Employees)
```

این شرط میتواند بر اساس فیلدهای بانک تولید گردد.

مثال : یک SP بنویسید که اختلاف تعداد سفارشاتی که فیلد ShipCountry آنها France یا German باشد. را برگرداند ؟

```
create Stored Procedure Order_France_German ( @Outp int output)
as
declare @Count_France int ، @Count_German int

select @Count_France = Count(*)
from orders
where ShipCountry = 'France'

select @Count_German = Count(*)
from orders
where ShipCountry = 'German'

if @Count_France is null set @Count_France = 0
if @Count_German is null set @Count_German = 0

set @outp = @Count_France - @Count_German
```

آموزش SQL Server (قسمت سوم)

در این قسمت در ادامه مطالب قبلی مبحث SP خواهیم پرداخت و آن را کامل خواهیم کرد. در قسمت بعدی مطلب با View ها آشنا خواهیم شد.

ایجاد Stored Procedure با استفاده از ویزارد ها:

برای ایجاد SP های استاندارد جهت عملیات درج ، حذف و ویرایش شما می توانید از ویزاردهای خود SQL استفاده نمایید. با این ابزار شما قادرید طی چند دقیقه تعداد زیادی SP جهت عملیاتهای گفته شده بر روی جداول خود بسازید. برای این منظور در Enterprise Manager بر روی کلید ویزارد کلیک نمایید مانند شکل زیر :

بعد در پنجره Select Wizard بر روی Database کلیک و گزینه Create Stored Procedure Wizard را انتخاب نمایید . مانند شکل زیر :

در پنجره بعدی به شما خوش آمد گویی مینمایید . برروی کلید Next کلیک نمایید.در پنجره بعدی نام Database ی را که قرار است بر روی جداول آن کار شود انتخاب نمایید و بروی Next کلیک نمایید. در پنجره بعدی جداول و عملیات مورد نظری که می خواهید انجام دهید انتخاب کنید. در پنجره بعدی نام SP هایی که سیستم ساخته شما خواهید دید. برای ویرایش نام و یا کد هرکدام از آنها میتوانید آن Sp را انتخاب و کلید Edit را فشار دهید. در اینصورت شما پنجره ای به شکل زیر خواهید داشت :

حال شما در قسمت Name می توانید نام Sp را عوض نمایید.

در قسمت Include in Set Clause شما فیلدهایی از جدول مربوطه که می خواهید مقادیر آن به شکل پارامتر برای Sp ارسال شود انتخاب نمایید. اگر Sp شما Update و یا Delete باشد در قسمت Include in Where Clause شما می توانید فیلدهایی که قرار است در شرط (دستور Where) قرار می گیرند انتخاب نمایید.

تذکر 1 : اگر فیلدی از نوع Identity دارید در حالت Insertحتما از قسمت Include in Set Clause خارج کنید. در غیر اینصورت در زمان اجرا با خطا مواجه خواهید شد.

تذکر 2 : SQL فیلد کلیدی جدول را در دستور Update در دستور Where خواهد آورد . همچنین شما

این فیلد را از Include in Set Clause خارج کنید.

در نهایت شما با فشار کلید Finish همزمان این Sp را خواهید ساخت.
تا این قسمت شما با ساخت Sp آشنا شدید. حال برای ویرایش آن نیز کافی است بر روی Sp کلید نموده و در پنجره باز شده کد آن را ویرایش نمایید.

آموزش SQL Server (قسمت چهارم)

در ادامه مطالب آموزش SQL حال به بحث دیدها (view) می پردازیم. دید در اصل یک جدول مجازی است که محتوی آن توسط یک پرس و جو تعریف می گردد. همانند جدول دید هم دارای سطر و ستونهایی می باشد. می توان به موارد زیر به عنوان مزایای دید اشاره کرد :

دیدها به کاربران اجازه میدهند تا بر روی داده هایی که نیاز دارند متمرکز شوند. بنابر این داده های غیر ضروری میتوان از دید خارج کرد. دیدها امنیت داده ها را نیز افزایش میدهند چراکه کاربر فقط داده هایی را می بیند که در دید وجود دارند.

دیدها به کاربران اجازه میدهند تا داده ها را به روشهای متفاوت مشاهد نمایند. دیدها میتوانند برای مبادله داده ها با سایر برنامه های کاربردی بکار روند.

و ....

ایجاد دید :

وقتی دیدی را ایجاد میکنید نام آن می بایست در بین نام جداول و دیگر دیدهایی که کاربر مورد نظر آنها را ساخته ، یکتا باشد.در SQL Server 7.0 شما امکان ایجاد شاخص بر روی دیدها را نداشتید ، لیکن این امکان در SQL 2000 اضافه شده است.

مراحل ایجاد یک دید :

1- Manager erprise را باز کرده و بر روی Databases کلیک کرده و پایگاه داده ای را که می خواهید دید در آن ایجاد کنید . باز کنید.

2- روی Views کلیک راست کرده و سپس گزینه New View... را انتخاب کنید.

3- در پنجره بعدی کلیک سمت راست کرده و گزینه Add Table... را انتخاب کنید.

4- دکمه های Table و یا Views جدول و یا دید های مورد نظر را انتخاب نمایید. و بر روی دکمه Add کلیک نمایید. این کار را برای تمام جداول و یا دیدهای مورد نظر تکرار کنید و سپس بر روی دکمه Close کلیلک نمایید.

در قسمت Column از پانل مشبک ، ستونهایی را که میخواهید در دید به آنها ارجاع نمایید انتخاب کنید. اگر میخواهید ستونی در مجموعه نتیجه دیده شود گزینه Output متناظر با آن می بایستی حتما

تیک داشته باشد.

در ستون Criteria شرط را بنویسید. چند شرط را میتوانید در ستونهای OR تکرار کنید. برای گروه بندی برروی ستون Criteria کلیک سمت راست کرده و گزینه Group By را انتخاب نمایید. در اینصورت شما قادر خواهید بود از توابعی همچون Sum استفاده کنید.

تذکر : اگر Group by را انتخاب کرده باشید. تمام شرایط که در ستون Criteria بنویسید به عنوان شرایط Having در نظر گرفته میشوند. برای اینکه این محدودیتها به شرط Where اضافه گردند ، بر روی پانل مشبک متناظر کلیک کرده و از لیست مورد نظر گزینه where را بجای Group by انتخاب کنید.

تذکر 2 : در ستون Alias شما میتوانید یک اسم مستعار برای این ستون در نظر بگیرید.

برای دیدن نتایج دید بر روی علامت (!) کلیک نمایید.

توجه داشته باشید که ارتباط ها در صورتی برقرار میشود که کلیدهای خارجی بر روی جداول وجود داشته باشد. گرچه شما می توانید با انتخاب فیلد مورد نظر و حرکت ماوس در حالیکه کلیک سمت چپ را فشار داده اید مابین دو جدول ارتباط را برقرار کنید.

آموزش SQLserver (بخش پنجم)

عبارت SELECT

پایه و اساس همه داده های رابطه ای در محیط SQL Server یک حالت Transact-SQL تنها است که عبارت Select می باشد. در این درس ما به مهمترین ترکیبات حالت Select و راههای استفاده کردن از Query Designer برای ساختن خودکار عبارت می پردازیم. با استفاده از Query Designer شما می توانید یک عبارت Select را مستقیماً در قاب SQL وارد سازید یا با داشتن Query Designer آن را به وسیله قابهای Grid و Diagram بسازید. گزینه ها متقابلاً منحصر به فرد نمی باشند. شما می توانید یک Query را به وسیله اضافه کردن جداول برای قاب Diagram ، تغییر نام ستون با استفاده از قاب Grid شروع کنید و دستورات را به طوری که سطرها برگردانده می شود به وسیله وارد کردن شرط ORDER BY به صورت مستقیم در قاب SQL مشخص کنید.

تمرینهای درس یک گونه ای از تکنیکها را به شما نشان خواهد داد. موقعی که خودتان کار می کنید شما می توانید یکی را که در زمان ساده تر به نظر می رسد را انتخاب کنید.

شناختن عبارت SELECT

ساختار دستور عبارت Select خیلی پیچیده می باشد که دارای چندین شرط و اپراتور می باشد اما ساختار اصلی کاملاً ساده می باشد.

```
Column-List [ [Select [ Top n [PERCENT
From Source-List
[ Where Search-Condition ]
[ Expression ORDER By ]
```

فقط اولین و دومین شرط از عبارت Select مورد نیاز می باشد. اولین شرط، Select Column-List مشخص می کند ستونهایی که به وسیله Query بازگردانده خواهد شد. لیست Column می تواند شامل ستونهای اصلی از جداول و نماهایی باشد که بر روی Query پایه ریزی شده است یا اینکه می تواند شامل ستونهای محاسباتی اقتباس شده از ستونهای اصلی باشد. دومین شرط، From Source-LIST ، نماها و جداول را که در Query پایه ریزی شده مشخص می کند.

انتخاب کردن تمام ستونها

ساده ترین شکل از عبارت Select آن انتخابی است که همه ستونها از یک جدول تکی را انتخاب می کند. با بیشترین نسخه های زبان SQL، Transact-SQL اجازه می دهد به شما به استفاده از (×) به عنوان یک Shorthand برای مشخص کردن تمام ستونها، بنابراین این فرم ساده از عبارت این چنین است:

```
× SELECT
FROM Table-Name
```

انتخاب همه ستونها

1- Query Designer را برای Properties Table به وسیله کلیک راست کردن نام آن در قاب

Detail Enterprise Manager باز می کنیم، روی زیر منوی Open Table رفته و Return All Rows را انتخاب می کنیم. SQL Server، Query Designer را برای جدول باز می کند.

2- قاب SQL را به وسیله کلیک کردن دکمه قاب SQL روی نوار ابزار Designer Query راه می اندازیم. Query Designer قاب SQL را نشان می دهد.

3- عبارت SQL را برای نشان دادن همه ستونها از جدول Oils تغییر می دهیم.

4- دکمه Run را روی نوار ابزار Query Designer برای به اجرا در آوردن Query کلیک می کنیم. Designer Query همه رکوردها در جدول Oils را نشان می دهد.

راهنمایی: شما می توانید بیشتر سطرها را در قاب Results به وسیله درج کردن درایور قاب بین قابها نشان دهید.

انتخاب زیر مجموعه از ستونها

اگر چه ساختار دستور Select × آسان و سریع می باشد، شما اغلب بیشتر می خواهید که Query تان فقط به ستونهای انتخاب شده باز گردد. این با مشخص کردن ستونها در Column-List از شرط Select به انجام می رسد.

انتخاب ستونها با استفاده از قاب SQL

1- علامت × را در عبارت Select با تایپ کردن نام ستون Oil Name جایگزین می کنیم.

2- دکمه Run را روی نوار ابزار Query Designer برای به کار بستن Query کلیک می کنیم. Query فقط ستون Name Oil را نشان می دهد.

انتخاب ستونها با استفاده از قاب Diagram

1- قاب SQL را پنهان کرده و قاب Diagram را به وسیله کلیک کردن دکمه ها در نوار ابزار Query Designer نشان می دهیم.

2- فیلد LatinName را در قاب Diagram کلیک کرده Query Designer قاب Results به جهت اینکه زیاد معتبر نیست کم نور نشان می دهد.

3- دکمه Run را برای به کارگیری Query کلیک کرده Query Designer هر دو ستون Latin Name و Oil Name را در قاب Results نشان می دهد.

ایجاد کردن نام مستعار ستون

به طور پیش فرض، یک ستون در یک Query نام یکسان دارد که در جدول یا نمای مورد نظر می باشد. اگر چه اغلب تغییر دادن نام آن مفید می باشد. نامتناسب به نظر می رسد که فیلد به کاربر بدین صورت ("My Long Column Name With No Spaces") یا اینکه به طور خلاصه این چنین 32713 PK-Varchar-50-Col نشان داده شود. عبارت Select به شما اجازه می دهد که یک ستون را در Query به وسیله ایجاد یک alias تغییر نام دهید. نام مستعار، نام ستون را در Query تغییر می دهد نه در جدول.

ایجاد یک نام مستعار برای ستون با استفاده از قاب Grid

1- قاب Diagram را پنهان کرده و قاب Grid را به وسیله کلیک کردن دکمه ها روی نوار ابزار Designer Query نشان می دهیم.

2- یک نام مستعار برای ستون OilName به وسیله تایپ کردن OilName در فیلد alias ایجاد می کنیم. Query Designer به طور خودکار در اطراف alias پرانتز اضافه کرده زیرا alias شامل یک فضا می باشد.

راهنمایی: پرانتزهای چهارگوش در خروجی Query نشان داده نخواهند شد. آنها به نحوه آسان به Server SQL می گویند که با عبارت "Oil Name" به عنوان یک نام مجزا برخورد کنیم. پرانتزها فقط موقعی مورد نیاز می باشند که نام مستعار شامل یک جای خالی باشد، اما آنها می تواند برای نام هر ستون استفاده شود.

3- دکمه Run را روی نوار ابزار Query Designer برای مجدد به کار بستن Query کلیک می کنیم. SQL Server نام را در سر ستون با جای خالی اضافه شده بین دو کلمه نشان می دهد.

ایجاد نام مستعار ستون با استفاده از قاب SQL

1- قاب Grid را پنهان کرده و قاب SQL را به وسیله کلیک کردن دکمه ها در نوار ابزار Query Designer نشان می دهیم.

2- [Name Latin] را به عنوان نام مستعار برای دومین ستون اضافه می کنیم.

3- دکمه Run را روی نوار ابزار Query Designer برای به کارگیری Query کلیک می کنیم. Query Designer نام را در سر ستون با یک فضای خالی بین دو کلمه نشان می دهد.

ایجاد کردن ستونهای محاسباتی

علاوه بر اینکه ستونها به طرز ساده اطلاعات را در جداول Underlying و نماها نشان می دهد، همچنین Query تان می تواند شامل ستونهایی باشد که محاسبه شده اند بر اساس داده های Underlying ،

توابع Server SQL یا هر ترکیب دوتایی. ستون محاسباتی به وسیله مشخص کردن یک عبارت به عنوان ستون ایجاد می گردد.
ما به عبارات Transact-SQL در Detail در درس 21 "Transact-SQL Language The" می پردازیم. بنابراین در این تمرین ما فقط یک جفت از عبارات ساده که بر اساس اپراتور الحاق رشته Transact-SQL که دو رشته و تابع GETDATE را که داده ها و زمان سیستم جاری را باز می گرداند اضافه می کنیم.

ایجاد کردن یک ستون محاسباتی با استفاده از قاب Grid
1- قاب SQL را پنهان کرده و قابGrid را به وسیله کلیک کردن دکمه روی نوار ابزار Query Designer نشان می دهیم.

2- در هر سل ستون خالی در قاب Grid کلیک کرده و Oil Name +' – '+ Latin Name را تایپ می کنیم.

راهنمایی: شما می توانید سل ها را در قاب Grid به وسیله درج کردن خطوط تقسیم بین سر ستونها عریض تر سازید.
3- کلید Tab را فشار می دهیم. SQL Server، Expr 1 را به عنوان نام مستعار ستون پیشنهاد می دهد.

4- نام مستعار را برای Extended Name تغییر می دهیم.

5- دکمه Run را برای به کارگیری مجدد Query کلیک می کنیم. Query Designer ستون جدید را در قاب Results نشان می دهد.

ایجاد کردن یک ستون محاسباتی با استفاده از قاب SQL
1- قاب Grid را پنهان کرده و قاب SQL را به وسیله کلیک کردن دکمه ها روی نوار ابزار Query Designer نشان می دهیم.

2- GETDATE را به عنوان [Today's Data] برای لیست ستون از شروط Select اضافه می کنیم.

راهنمایی: کاما را قبل از GETDATE فراموش نکنید.
3- دکمه Run را در نوار ابزار Query Designer برای به کارگیری مجدد Query کلیک می کنیم.
SQL Server تاریخ جاری در هر سطر را نشان می دهد.

استفاده کردن از شرط Top n

زمانی که شما دستور Return Top را از منوی متن جدول انتخاب می کنید. SQL Server شرط Top n را در پایین پوششها برای ایجاد صفحه نمایش در Query Designer استفاده می کند. علاوه بر اینکه یک شماره مشخصی از سطرها را مشخص می کنید شما می توانید همچنین یک درصد از سطرها را به وسیله استفاده کردن از شرط Top n Percent نشان دهید. همان طوری که شما ممکن است انتظار داشته باشید درصدی از سطرهای مشخص شده را باز می گردانیم.

نشان دادن سطرهای Top s

1- Top 5 را قبل از اولین کلمه در Column-List از شروط Select در قاب SQL اضافه می کنیم.

2- دکمه Run را در نوار ابزار Query Designer برای به کارگیری مجدد Query کلیک می کنیم. SQL Server فقط 5 سطر اول را نشان می دهد.

نشان دادن Top 5 درصد از سطرها

1- کلمه Percent را بعد از Top 5 در قاب SQL اضافه می کنیم.

2- دکمه Run را در نوار ابزار Query Designer برای به کارگیری Query کلیک می کنیم. SQL Server فقط 5 درصد اول از سطرها را در SQL Server نشان می دهد.

آموزش SQLserver (بخش ششم)

شناخت عبارت INSERT

ساختار دستور عبارت Insert شبیه به عبارت Select می باشد، بیشترین شکل مبنای آن عبارت است از:

```
[(Column-List)] INSERT [INTO] Table-or-View
(VALUES (Value-List
```

هر عبارت Insert می تواند یک جدول یا نما منفرد را به هنگام سازد. زمانی که شما از عبارت Insert برای به هنگام سازی یک نما استفاده می کنید شما باید آگاه باشید از محدودیتهای زیر:

View نباید شامل یک تابع به هم پیوسته مانند COUNT یا AVG باشد.

View نباید شامل Top، GROUP BY ، UNION یا DISTINCT باشد.

View نباید شامل یک ستون محاسبه شده باشد.

View باید یک جدول را در شرط From باز گرداند.

عبارت Insert ستونهای یک جدول منفرد را فقط به هنگام می سازد.

لیست ستون در عبارت Insert اختیاری می باشد. اگر آن فراهم نگردد عبارت Insert باید شامل مقادیری برای همه ستونها در جدول یا نما باشد و آنها باید به همان ترتیب به عنوان ستونهایی در جدول یا نما فراهم شوند. اگر چه شما می توانید از کلیدهای ویژه DEFAULT برای مشخص کردن مقادیر پیش فرض برای یک سطر استفاده کنید. زمانی که لیست ستون گنجانده می شود. آن یک فرمت شبیه از لیست ستون در عبارت Select می گیرد: یک لیستی از نامهای ستون که با کاما جدا شده است. از زمانی که یک عبارت Insert می تواند یک سطر را فقط برای یک جدول اضافه کند، شما معمولاً نیاز به استفاده کردن مشخصه نام جدول برای نام ستون نخواهید داشت.

استفاده کردن از عبارت INSERT

یک عبارت Insert می تواند با استفاده کردن از قاب Grid برای مشخص کردن ستونها یا با استفاده کردن قاب SQL برای وارد کردن مستقیم عبارت ایجاد گردد.

درج کردن سطرها با استفاده از قاب Grid

قاب Grid احتمالاً آسانترین راه برای ایجاد یک عبارت Insert می باشد از زمانی که یاد آوردن هر ساختار دستور مورد درخواست شما نباشد.

درج کردن یک سطر با استفاده از قاب Grid

پوشه Tables را از پایگاه داده Aromatherapy راهبری کرده، جدول Oils را در قاب Details کلیک راست می کنیم. در زیر منوی Open Table رفته و Query را انتخاب می کنیم. Query Designer همه چهار قاب نشان داده شده را باز می کند.

آموزش SQLserver (بخش هفتم)

ایجاد کردن جداول و تغییر دادنشان

در محیط SQL Server میکروسافت، در هر پایگاه داده مرتبط، اطلاعاتی در داخل جداول سازماندهی می شوند بر طبق ترتیب سطرها و ستونها که داده ها را برای شئی های تکی ذخیره می کند. در این درس، شما می آموزید که چگونه یک جدول جدید را ایجاد کنید و معین کنید ستونهایی که آن جدول در بردارد.

راهنمایی: در نظر گرفتن محتویات جدول به عنوان یک شبکه (Grid) مانند یک صفحه گسترده ایده آل می باشد. اما این موضوع مهم می باشد به تشخیص اینکه رکوردها در یک جدول نظم درونی ندارند. ایده های Previous و Next در یک رکورد جدول به کار برده نمی شوند. اگر شما نیاز به انجام عملیاتهای متوالی نداشته باشید، شما می توانید به وسیله ای به نام Cursor آن را انجام دهید. یک ماهیتی که به یک سطر ویژه در یک مجموعه از رکوردها اشاره می کند. در مورد Cursor در درس 27 بحث خواهیم کرد.

ایجاد کردن جداول

جداول یک واحد تابعی از انبار داده ها در یک پایگاه داده های مرتبط می باشند. به عنوان یک قاعده کلی، هر نوع از موجودیتها از قبیل Essential Oil در پایگاه داده نمونه مان به وسیله یک جدول نشان داده می شود، نظر به اینکه هر نمونه از آن موجودیتها از قبیلGerman Chamomile Clary Sage به وسیله یک سطر در جدول نشان داده می شود.

طراحی پایگاه داده

در بیشترین روش رایج برای تعریف پایگاه داده ها از راهکارهای (مفهومهای) Entities و Attributes استفاده می کنیم زمانی که شما از طراحی منطقی به طراحی فیزیکی پیش می روید. موجودیتها معمولاً به صورت جداول پیاده سازی می شوند و مشخصه ها (Attributes) در ستونها پیاده سازی می شوند. (همچنین در فیلدها شناخته می شوند)

شناخت انواع داده ها

هر ستون در جدول دارای خصوصیات معین می باشد که آن را برای SQL Server تعریف می کند. مهمتر از این خصوصیات، نوع داده های ستون می باشد، که تعریفی از نوع اطلاعاتی که درستونها ذخیره خواهند شد می باشد. SQL Server یک محدوده وسیعی از انواع داده ها را فراهم آورد که در جدول 5-1 نشان داده شده است. به طور کلی انواع داده ها به وسیله SQL Server فراهم می گردد، همچنین شما می توانید خودتان تعریف کنید. شما خواهید آموخت که چگونه این را در درس 9 انجام دهید.

مقادیر قابل قبول انواع داده

مقادیر عددی

مقادیر اعداد صحیح از 2^63- تا 2^63-1 bight

مقادیر اعداد صحیح از 2^31- تا 2^31-1 Int

مقادیر اعداد صحیح از 2^15- تا 2^15-1 Smallint

مقادیر اعداد صحیح از 0 تا 255 Tinyint

مقادیر اعداد صحیح با ارزش 0 و 1 bit

مقادیر مقیاس با دقت ثابت شده از 10^38+1 تا 10^38-1 decimal

مقادیر Decimal همچنین می تواند تعریف شود به صورتNumeric ، دامنه مقادیر یکسان می باشد.

مقادیر Monetary (مالی) از 2^63- تا 2^63-1 (مقادیر Money تا 0.0001 از هرواحد دقت دارد).

Money

مقادیرMonetary از 3648 . 748 . 214- تا 3647 . 748 . 214 می باشد Smallmoney

(مقادیر Small Monetary تا 0/0001 واحد دقیق می باشد).

مقادیر صحیح شناور (متغیر) از E1.79+308- تا E1.79+308 (مقادیر Float فقط تقریبی می باشد)

Float

مقادیر صحیح شناور متغیر از E3.40+38- تا E3.40+38 می باشند (مقادیر real فقط تقریبی می باشد) real

مقادیر تاریخ و زمان از 1.1753 ژانویه تا 31.9999 دسامبر می باشد datetime

(مقادیر Date Time تا 3 هزارم ثانیه یا 3.33 میلی ثانیه دقت دارد) Smalldatedtime

مقادیر تاریخ و زمان از 1 ژانویه و 1900 تا 6 ژوئن و 2079 می باشد (مقادیرSmalldatetime تا 1 دقیقه دقت دارند)

مقادیر کاراکتری

مقادیر حرفی کدگذاری نشده با طول ثابت با طول حداکثر 8000 حرف می باشد. Char

مقادیر حرفی کدگذاری نشده با طول متغیر با طول حداکثر 8000 حرف می باشد. Varchar

داده کدگذاری شده با طول متغیر با طول حداکثر 2^31-1 (647 . 741 . 073 . 1) حرف می باشد. Text

داده کدگذاری شده با طول ثابت با طول حداکثر 4000 حرف می باشد Nchar

داده کدگذاری نشده با طول متغیر با حداکثر طول 4000 حرف می باشد nvarchar

داده کدگذاری شده با طول متغیر با حداکثر طول 2^30-1 (823 . 741 . 073 . 1) حرف می باشد ntext

مقادیر باینری (1 و 0)

داده باینری با طول ثابت با حداکثر طول 8000 بایت می باشد binary

داده باینری با طول متغیر با حداکثر طول 8000 بایت می باشد Varbinary

داده باینری با طول متغیر با حداکثر طول 2^31-1 (647 . 783 . 147 . 2) بایت می باشد Image

مقادیر دیگر

یک مرجع مبناء برای یک Cursor می باشد (یکCursor یک ماهیتی است که یک مرجع مبناء را برای یک سطر مشخص در یک Result Set نشان می دهد. Cursor

یک شمار واحد پایگاه داده است که به هنگام می شود هر زمانی که یک سطر به هنگام شود. (نوع داده rowversion در نسخه قبلی از Timestamp SQL Server نامیده می شود) rowversion

مقادیری از هر نوع غیر از text، ntext، rowversion (timestamp) و variant - sql می باشد.

یک معرف واحد کلی GUID می باشد. Uniqveidentifier

ایجاد کردن یک جدول جدید

جداول تهیه و با استفاده از طراح جدول Enterprise Manager نگهداری می شوند. اولین گام ایجاد و نامگذاری جدول به وسیله باز کردن Designer Table برای یک جدول جدید می باشد.

ایجاد یک جدول جدید

1- در پایگاه داده Aromatherapy بر روی پوشه Table می رویم. SQL Server یک لیستی از جداول موجود را نشان می دهد.

2- دکمه New را روی نوار ابزار کلیک می کنیم. SQL Server ، Table Designer را باز می کند.

3- دکمه Properties را در نوار ابزار کلیک می کنیم. SQL Server کادر محاوره ای Tables Properties را باز می کند.

4- نام جدول را به Lesson 5 تغییر می دهیم.

5- Close را کلیک می کنیم. SQL Server کادر محاوره ای Properties را می بندد.

اضافه کردن ستونها به یک جدول

اگر چه یک جدول یک Properties معین برای خود دارد. از قبیل نامی که ما در تمرین آخر به کار بردیم. یک جدول اصولاً به وسیله ستونها که در آن موجود است تعریف می شود.

اضافه کردن یک ستون عددی به جدول

1- My Number را در سل Column Name تایپ کرده و سپس Tab را فشار می دهیم. SQL Server، char را به عنوان نوع داده پیشنهاد می دهد.

2- نوع داده را برای decimal تغییر می دهیم. SQL Server طول ستون را به 9 تغییر داده و صحت، مقیاس و مشخصات فیلدها را تأیید می کند.

3- در سل Description، Sample Numeric Column را تایپ می کنیم.

4- Precision ستون را به 5 و Scale را به 2 تغییر می دهیم. SQL Server طول ستون را به 5 برای نمایش دادن Precision جدید تغییر می دهد.

شرح ستون

توانایی اضافه کردن یک شرح برای یک ستون در enterprise Manager در SQL Server 2000 جدید می باشد که بخشی از عاملیت جدید است که به طور extended Properties شناخته می شود. مایکروسافت مقداری از خصوصیات گسترده شده را از قبیل Column description به عنوان بخشی از نصب Server استاندارد ایجاد کرده است. شما می توانید به طور کلی خصوصیات گسترده شده را برای ذخیره اطلاعات ویژه سایت یا اطلاعات ویژه کاربردی در حدود شئی های پایگاه داده ایجاد کنید. extended Properties نام کاربر تعریف شده و یک مقدار دارد که مقادیری از یک extended Properties مقادیر sql - variant می باشد که می تواند شامل بیش از 7500 بایت از داده ها باشد. شما می توانید extended Properties چندگانه را برای هر شئی با استفاده از روشهای ذخیره شده تعریف کنید. برای اطلاعات بیشتر در مورد روشهای ذخیره شده (Stored Procedures) به درس 28 رجوع کنید.

Precision و Scale (دقت و مقیاس)

دقت یک ارزش عددی مقدار ماکزیمم از رقمهای decimal است که ارزش آن را نشان می دهد. برای سمت چپ و راست از نقطه Decimal . مقیاس یک ارزش عددی رقمهای سمت راست از نقطه Decimal می باشد. برای مثال، مقدار عددی 311 . 3647 یک دقت 7 دارد (مجموع شماره رقمها) و یک مقیاس 3 (رقمهای سمت راست از نقطه Decimal) را دارا می باشد. آن مهم می باشد که بدانیم دقت و مقیاس ارزش یک عدد بر روی طول یک ستون تأثیر نمی گذارد. نوع داده طول ستون را تعیین می کند. دقت و مقیاس تعیین می کند که چگونه SQL Server داده های ذخیره شده در ستونها را تفسیر می کند.

اضافه کردن یک ستون Identity به جدول

1- در یک سل خالی در ستون Column Name کلیک می کنیم.Myidentity را تایپ کرده و سپسTab را فشار می دهیم. SQL Server، char را به عنوان یک نوع داده پیشنهاد می کند.

2- نوع داده ها به decimal تغییر می دهیم. SQL Server طول ستون را به 9 تغییر داده و فیلدهای Identity و Scale و Precision را تأیید می کند.

3- Allow Nulls را تیک می کنیم.

Nulls

ارزش Nulls یک نوع خاصی از یک ارزش در تکنولوژی رابطه ای می باشد که استفاده می شود برای

نشان دادن اینکه یک ارزش ناپیدا یا غیر موجود می باشد استفاده کردن از Nulls قدری پیچیده و مسئله ساز و قطعاً مورد بحث می باشد.

4- در سل Description و Sample Identity Column را تایپ می کنیم.

5- فیلد Identity را به Yes تغییر می دهیم (نه برای SQL Server(Replication برای هر دوی از فیلد Identity Seed و فیلد Increment Identity مقدار 1 را پیشنهاد می دهد.

ارزشهای Identity

زمانی که شما خصوصیات Identity یک ستون را تنظیم می کنید، شما به SQL Server می گویید که یک مقدار را در داخل ستون که به طور واحد هر سطر را مشخص می کند قرار دهد. نوع داده انتخاب شده ماهیت دقیق از ستون را تعیین می کند. ستونهای Identity می توانند نوع داده هایی مانند int و Smallint و tinyint یا decimal داشته باشند. زمانی که SQL Server یک سطر را در داخل یک جدول که دارای یک ستون Identity است قرار می دهد. آن به طور خودکار ارزشی برای ستون مبنی بر ارزش استفاده شده اخیر (که با Identity Seed شروع می شود) و Identity Increment مشخص شده زمانی که جدول ایجاد شده بود ایجاد می کند. برای مثال، اگر یک ستون Identity به عنوان یک Small int با Identity Seed، 50 و یک Identity Increment، 5 تعریف شود اولین سطر عدد 50 و دومین سطر 55 و سومین سطر 60 و غیره تخصیص داده می شود فقط یک ستون در یک جدول می تواند تنظیمات Identity Property را داشته باشد.

اضافه کردن ستون GUID به جدول

1- در یک سل خالی در ستون Column Nameکلیک کرده، My Guid را تایپ کرده و سپس Tab را فشار می دهیم. SQL Server ، نوع داده char را پیشنهاد می دهد.

2- نوع داده را به Uniqueidentifier را تغییر می دهیم. SQL Server طول ستون را به 16 تغییر داده و فیلد IsrowGuid را تأیید می کند.

3- در سل Description، Sample Guid تایپ می کنیم.

4- IsrowGuid را به Yes تغییر می دهیم. SQL Server یک مقدار پیش فرض به Newid( ) می دهد.

GUIDS

Guid که از GloballyUniqueIdentifier گرفته شده، یک مقادیر باینری (1 و 0) 16 بایتی می باشد که هیچ کامپیوتر دیگری در دنیا نخواهد مقدار آن را تولید کند. نوع داده uniqueidentifier برای ذخیره کردن Guids استفاده می شود. SQL Server به طور خودکار مقادیر Guid را از همان راهی که مقادیر Identity ایجاد می شود، فراهم نمی کند. زیرا یک جدول می تواند شامل Guids چندگانه باشد، اما فقط یک Identity منحصر به فرد باشد. اگر چه، تابع NEWID که Server SQL آن را به صورت پیش فرض در نظر می گیرد زمانی که خصوصیات IsrowGuid، Yes می شود. یک Guid جدید بر خواهد گشت زمانی که سطر قرار داده می شود.

اضافه کردن یک ستون Data در جدول

1- در یک سل خالی در ستون Column Name کلیک کرده، Mydata را تایپ کرده و سپس Tab را فشار می دهیم. SQL Server، char را به عنوان نوع داده در نظر می گیرد.

2- نوع داده را به datetime تغییر می دهیم. SQL Server طول ستون را به 8 تغییر می دهد.

3- در سل Sample Date Column Description را تایپ می کنیم.

اضافه کردن یک ستون Character به جدول

1- در یک سل خالی در ستون Column Name کلیک کرده، Mychar را تایپ و سپس Tab را فشار می دهیم. SQL Server، char را به عنوان نوع داده در نظر می گیرد.

انواع داده های Character

SQL Server دو نوع مختلف از ستونها Character را پشتیبانی می کند. طول ثابت شده و طول متغیر که هر کدام در دو "Flavors" مختلف Unicode و Non-Unicode و 3 طول متفاوت می باشند. Unicode یک روش از علامت کد گذاری می باشد که انواع بایتهای دابل را پشتیبانی می کند. اگر یک ستون به طور طول متغیر نمایان شود (برای مثال، varchar یا text برای داده های Non-Unicode و nvarchar و ntext برای داده هایUnicode) و SQL Server فقط خصوصیت داده های معین وارد شده را ذخیره خواهد ساخت. از طرف دیگر اگر ستون به صورت طول ثابت شده نمایان شود (char برای Non-Unicode داده nchar برای داده های Unicode) و SQL Server مقادیر وارد شده را با فضاهای Pad خواهد کرد.

2- طول ستون را به 25 تغییر می دهیم.

3- در سل Description، Character Column Sample را تایپ می کنیم.

4- Unknown را در سل Default Valve تایپ می کنیم (اطمینان حاصل کنید که اطراف کلمه ویرگول داشته باشد).

مقادیر پیش فرض

یک Default Valve یک عددی می باشد که در داخل یک ستون قرار خواهد گرفت اگر کاربر به طور آشکار یکی را فراهم نکند. ما تاکنون دو نوع ویژه از مقادیر پیش فرض را دیده ایم default Values به وسیله SQL Server تهیه می شود زمانی که شما Property Identity را تنظیم می کنید و تابع NEWID به وسیله SQL Server تهیه می شود زمانی که شما IsrowGuid را تنظیم می کنید. در حقیقت، شما می توانید مقادیر پیش فرض را برای هر ستون مشخص کنید. مقادیر پیش فرض می تواند دائمی باشد مانند "Unknown" یا 123 توابعی از قبیل NEWID یا GETDATE یا عبارات ریاضی مانند 5+3 .

ذخیره کردن و بستن جداول

1- دکمه Save را در نوار ابزار Table Designer کلیک می کنیم. SQL Server تعریف جدول را ذخیره می سازد.

2- پنجره را می بندیم.

مدیریت جداول

اگر چه "Best Practice" امر می کند که طراحی پایگاه داده تان باید ثابت باشد قبل از اینکه شما پیاده سازی را شروع کنید. خوشبختانه SQL Server به انجام رسانی وظایف نگهداری را آسان می سازد.

تغییر دادن ستونها

شما می توانید مجدد Table Designer را برای یک جدول به وسیله کلیک راست کردن نام جدول در قاب Details باز کنید و Design Table را از منوی Context انتخاب کنید. همان موقعی که Table Designer باز می شود شما می توانید خصوصیات ستونهای موجود را تغییر دهید آنها را حذف و یا ستون جدیدی را اضافه کنید.

تغییر نام یک ستون

1- پوشه Tables را برای پایگاه داده Aromatherapy در درخت Console انتخاب می کنیم. SQL Server یک لیستی از جداول در قاب Details را نشان می دهد.

2- جدول Lesson 5 را در قاب Details کلیک راست کرده و Design Table را انتخاب می کنیم. SQL Server ، Table Designer را باز می کند.

3- Mychar را در سل Column Name انتخاب و MyCharacter را تایپ می کنیم. SQL Server، Column Name را تغییر می دهد.

4- دکمه Save را در نوار ابزار Table Designer کلیک کرده و تغییرات را ذخیره می سازیم.

برداشتن یک ستون

1- ستون Mydate را به وسیله کلیک کردن روی gray Boy در سمت چپ از Column Name انتخاب می کنیم.

2- کلید Delete را فشار می دهیم. SQL Server ستون را برمی دارد.

3- کلید Save را برای ذخیره کردن تغییرات کلیک می کنیم.
4- پنجره Table Designer را می بندیم.

تغییر دادن جداول

به طور کلی برای تغییر تعریف ستونهای یک جدول، Enterprise Manager تغییر نام دادن جداول را آسان می سازد و جداول را از پایگاه داده حذف می کند.

تغییر نام یک جدول

1- پوشه Table را در پایگاه داده Aromatherapy در درخت Console راهبری می کنیم. SQL Server یک لیستی از جداول در قاب Details نشان می دهد.

2- جدول Lesson 5 را در قاب Details کلیک راست کرده و Rename را انتخاب می کنیم.
3- New Lesson 5 را تایپ و Enter را فشار می دهیم. SQL Server کادر محاوره ای Rename را نشان می دهد که به شما اخطار می کند که تغییر در نام جدول هر رابط برای آن را در شئی های دیگر، باطل خواهد کرد.

4- View Dependencies را برای نشان دادن هر شئی که ممکن است به وسیله تغییرات تأثیر یابد را کلیک می کنیم. SQL Server کادر محاوره ای Dependencies را باز می کند.

5- Close را برای مرخص کردن کادر محاوره ای کلیک می کنیم.
6- Yes را در کادر محاوره ای Rename برای تأیید تغییر نام کلیک می کنیم. SQL Server نشان می دهد یک متنی را که تکمیل موفقیت آمیز تغییر نام را تأیید می کند.

برداشتن یک جدول

1- New Lesson 5 را در قاب Details انتخاب می کنیم.

2- کلید Delete را فشار می دهیم. SQL Server کادر محاوره ای Drop Object را نشان می دهد.

راهنمایی: شما می توانید دکمه Dependencies Show را برای نشان دادن هر Objects که متأثر خواهد شد به وسیله حذف جدول کلیک کنید.

3- Drop All را کلیک می کنیم. SQL Server جدول را از پایگاه داده برمی دارد.

مهم: زمانی که شما یک جدول را حذف می کنید، جدول و همه داده هایش به طور دائمی از پایگاه داده برداشته می شود و تنها راه برای بازگرداندن آن نسخه پشتیبان پایگاه داده می باشد.

آموزش SQLserver (بخش هشتم)

ایجاد کردن شئی های جدول

در درسهای گذشته، شما آموختید که چگونه خصوصیات گوناگون از قبیل مقادیر پیش فرض و Check Constraints برای ستونهای خاص از یک جدول تخصیص دهیم. بعضی مواقع اگر چه، یک نوع خاص از ستون در چندین جدول مختلف استفاده می گردد. در این وضعیت، آن اغلب مفید می باشد به ایجاد Properties در یک جای جداگانه به طوری که آنها را برای هر جدول به کار ببریم.

پیش فرضها، نقشها و انواع داده تعریف شده کاربر مکانیزمی را برای ایجاد و نگهداری این شئی ها در یک مکان جداگانه فراهم می آورند. برای مثال شما یک مدل پایگاه داده برای جوابگویی به ارزیابی مشتری می سازید. شما در ابتدا تصمیم می گیرید که مقدار پیش فرض برای هر سؤالی که جواب داده نشده باید "Unknown" شوند. اگر شما یک پیش فرض ایجاد کنید و پیش فرضها را برای ستون مناسب پیوند دهید. شما می توانید بعداً پیش فرض را به Unanswered تغییر دهید. بدون هیچ تغییری هر ستون، آن پیش فرض را استفاده می کند.

شناخت پیش فرضها

توابع پیش فرض از همان راهی که خصوصیات پیش فرض که شما مشخص می کنید زمانی که شما یک ستون در Table Designer ایجاد می کنید آنها مقادیری هستند که به طور خودکار به وسیله SQL Server تخصیص داده می شوند. اگر کاربر یک مقدار را زمانی که سطری را ایجاد می کند مشخص نکند. اگر چه پیش فرض، شئی های سطح پایگاه داده می باشند که می تواند برای چندین ستونها به کار برده شوند.

ایجادکردن پیش فرضها

از موقعی که پیش فرض، شئی های مستقل در داخل پایگاه داده هستند، شئی باید پیش فرض را ایجاد کند قبل از اینکه شئی بتواند آن را برای یک ستون جدول متصل سازد.

ایجاد کردن یک پیش فرض

1- پوشه Defaults را از پایگاه داده Aromatherapy از درختConsole جستجو می کنیم. SQL Server یک لیستی از Default ها را در قالب Details نشان می دهد. (چیزی در پایگاه داده Sample وجود ندارد).

2- دکمه Newرا کلیک کرده، SQL Server کادر محاوره ای Default Properties را نشان می دهد.

3- در فیلد Name، Default Unknown را تایپ می کنیم.

4- "Unknown" را در فیلد Value تایپ می کنیم.

5- OK را کلیک کرده، SQL Server پیش فرض را ایجاد می کند.

ربط دادن یک پیش فرض به یک ستون

1- پوشه Tables را راهبری کرده، Table Designer را برای جدول Oil با کلیک راست کردن نام جدول در قاب Details باز کرده و Table Design را انتخاب می کنیم.

2- یک ستون جدید برای جدولی که Sample نامیده شده اضافه می کنیم. انواع داده پیش فرض را پذیرفته و طول آن به وسیله SQL Server پیشنهاد می گردد.

3- فیلد Valve Default را برای ستون کلیک کرده و سپس dbo.DefaultUnknown را از لیست انتخاب می کنیم.

4- دکمه Save را کلیک کرده SQL Server جدول را ذخیره می کند.

قطع پیوند یک پیش فرض

1- اگر Table Designer برای جدول Oils از تمرین قبل باز نیست، آن را به وسیله کلیک راست کردن نام جدول در قاب Pet ail> باز کرده و Table Design را انتخاب می کنیم. SQL Server، Table Disdainer را باز می کند.

2- ستون Sample را انتخاب کرده، Table Designer خصوصیات این ستون را نشان می دهد.

3- dbo.DefaultUnknown را در فیلد Default Value انتخاب کرده و کلید Delete را برای برداشتن مقدار فشار می دهیم.

4- دکمه Save راکلیک کرده SQL Server تغییرات برای تعریف ستون را ذخیره می سازد.

شناخت نقشها

نقشها مانند پیش فرضها سطوح شئی های پایگاه داده می باشند که می تواند برای ستونها در جداول چندگانه به کار برده شوند. یک نقش Check Constraint موجود می باشد که مشخص می کند مقادیر داده ها در یک ستون مورد قبول می باشند، اما استفاده از آن بیشتر محدود شده است. یک ستون می تواند چندین Check Constraint داشته باشد که برای آن به کار برده می شود.

راهنمایی: مایکروسافت نقشها و توصیه ها را که با Check Constraint جایگزین می شود را درست نمی داند. اگر چه نقشها هنوز جایگاهشان را در پایگاه داده های SQL Server دارند از زمانی که فقط یک وظیفه بتواند برای نوع داده هایی که به وسیله SQL Server تعریف شده به کار برده شود.

برخلاف Check Constraint ، یک نقش نمی تواند مبنایی برای یک ستون به طور مستقیم باشد. در عوض، مقادیری که یک نقش به کار می برد به نقشی که در یک متغیری که فرمت@Variable Name

می گیرد داده می شود. در مورد متغیرها در Detail در فصل 24 بحث خواهیم کرد.

ایجاد کردن نقشها

از زمانی که نقشها مانند Defaults مستقل از شئی های پایگاه داده می باشند، شما باید آنها را قبل از اینکه شما بتوانید آنها را برای یک ستون در جدول به کار ببرید ایجاد کنید.

ایجاد یک نقش

1- پوشه نقش را از پایگاه داده در درخت Console راهبری کرده SQL Server یک لیستی از نقشها در پایگاه داده نشان می دهد. (لیست در پایگاه داده Sample خالی می باشد)

2- دکمه New را کلیک کردهSQL Server کادر محاوره ای را باز می کند.

3- Sample Rule را به عنوان نام نقش تایپ می کنیم.

4- LEN(@Fldalue)>3 را به عنوان متن نقش تایپ می کنیم.

راهنمایی: به خاطر داشته باشید LEN یک تابعTransact SQL می باشد که تعدادی از کاراکترها را در یک متن رشته ای برمی گرداند و اینکه @ قبل از یک بر چسب عبارتTransact SQL یک متغیـــر را نشان می دهد، یک مقدار که برای عبارت داده می شود. بنابراین در این حالت نقش Trueمی گردد اگر طول ستون بزرگتر از 3 باشد.

5-OK را کلیک کرده، SQL Server کادر محاوره ای Rule Properties را می بندد و نقش را ایجاد می کند.

ربط دادن یک Rule به یک ستون

1- کادر محاوره ای Properties Rule را برایSample Rule به وسیله دابل کلیک کردن نام Rule در قاب Details باز می کنیم. SQL Server کادر محاوره ای Rule Properties را نشان می دهد.

2- ستونBind را کلیک کرده SQL Server کادر محاوره ایBind Rule to Columns را نشان می دهد.

3- [dbo].[Oils] را درTable Combo Box انتخاب می کنیم.Server SQL فیلدها را در جدول Oils نشان می دهد.

4- ستون Sample را در لیست Unbound Columns انتخاب کرده و سپس Add را کلیک می کنیم. SQL Server ستون را به لیست Bound Columns حرکت می دهد.

5- SQL Server کادر محاوره ایColumns Bind Rule to را می بندد.
6-OK را مجدداً برای بستن کادر محاوره ای Rule Properties کلیک می کنیم.

شناخت User-Defined Data Types
نقشها و پیش فرضها مکانیزم مفیدی برای نگهداری محدودیتهای پایگاه داده ها می باشند، اما SQL Server حتی مکانیزم قوی تری در User-Defined Data Typesها فراهم می آورد. User-Defined Data Types بر مبنای هیچ نوع از پایگاه داده حقیقی مشخص نمی گردد و شامل مشخصاتی از طول ستون می باشد. به طور کلی نقشها و پیش فرضها ممکن است به طور انتخابی برای یک User-Defined Data Types به کار برده شوند. زمانی که یک ستون بر مبنای یک User-Defined Data Typesایجاد می گردد، ستون جدول همه خصوصیات مشخص شده را برای آن نمونه به ارث خواهد برد. زمانی که مشخصات از User-Defined Data Typesتغییرمی یابد،نقشها برای ستونها بر اساس آن نمونه همچنین تغییر خواهد کرد.
راهنمایی: اگر یک User-Defined Data Type در پایگاه داده Model ایجاد گردد، همه پایگاه داده جدید به طور خودکار به آن نوع دسترسی خواهد داشت.

ایجادکردن Data Types User-Defined
User-Defined Data Types از شئی های پایگاه داده مستقل می باشند و باید در داخل پایگاه داده تعریف شوند قبل از اینکه آنها بتوانند به ستونها تخصیص داده شوند.

ایجاد کردن یک User-Defined Data Types
1- پوشه User-Defined Data Types را از پایگاه داده Aromatherapy راهبری می کنیم. SQL Server یک لیستی ازUser-Defined Data Types را نشان می دهد (چیزی در پایگاه داده Sample وجودندارد).

2- دکمه New را کلیک کرده SQL Server کادر محاوره ای User-Defined Data Types را نشان می دهد.

3- MySample را به عنوان نام User-Defined Data Types تایپ می کنیم.

4- نوع داده پایگاه را varchar و طول آن را 20 قرار می دهیم.

5- Dbo.Sample Rule را در Rule Combo Box انتخاب می کنیم.

6- مقادیر پیش فرض برای Allow Nulls و گزینه های پیش فرض را می پذیریم و OKرا کلیک می کنیم. SQL Server، User-Defined Data Types را ایجاد می کند.

تخصیص یک ستون برای یک User-Defined Data Types

1- Designer Table را برای جدول Oils به وسیله کلیک راست کردن نام آن در قاب Details باز می کنیم و Details Table را انتخاب می کنیم. SQL Server، Table Designer را باز می کند.

2- ستون Sample را انتخاب کرده وMySample را از Data Type Combo Boxرا انتخاب می کنیم. SQL Server , Data Type را برای MySampleقرار می دهد.
راهنمایی: User-Defined Data Types در پایین لیست Data Type می باشد.

3- دکمهSave را کلیک کردهSQL Server جدول را با تعریف جدید ذخیره می سازد.

آموزش SQLserver (بخش نهم)

جداول موقت

جداول موقت مثل جداول عادی هستند با این تفاوت که آنها فقط زمانی وجود دارند که از آنها استفاده می شود. آنها به طور اتوماتیک وقتی که تمام کاربران دیگر با آنها کاری ندارند توسط Microsoft SQL حذف می شود. نکته: ایجاد یک جدول موقت یک رویه تقریباً پر هزینه می باشد که مربوط به هزینه منابع Server و چرخه های CPU می باشد. بسیاری از استفاده های مرسوم از جداول موقت هم اکنون می توانند با استفاده از جدول متغیرها جایگزین شوند.

استفاده از جدول موقت

جداول موقت با استفاده از همان فرمان Create و Select INTO به عنوان جداول عادی ایجاد می شوند. بعد از تهیه جدول در سؤال قابل دسترسی به ارتباط خواهد بود. (این است که این یک جدول محلی تهیه شده توسط ارتباط متفاوت نیست)، این عمل همچنین مثل جداول عادی قابل استفاده می باشد.

ایجاد جدول موقت محلی

1- دکمه علامت سؤال جدید را روی Toolbar تجزیه کننده سؤال کلیک کنید تا یک پنجره سؤال جدید ظاهر شود.

2- روی دکمه Script Loud روی Toolbar تجزیه کننده سؤال کلیک کنید. تجزیه کننده سؤال جعبه پرونده گفتگو را نشان خواهد داد.

3- نسخه خطی Create Local را انتخاب کنید و Open را کلیک کنید. تجزیه کننده سؤال نسخه خطی را پر می کند.

4- دکمه Execute Query را در Toolbar تجزیه کننده سؤال کلیک کنید. تجزیه کننده سؤال جدول موقت ایجاد می کند.

5- User Table Folder را که در پایگاه داده Tempdb در Object browser می باشد را انتخاب کنید.

6- F5 را برای به کارگیری مجدد نمایشگر Object browser فشار دهید و User Table Folder را گسترش دهید. نمایشگر سؤال جدول محلی dbo.#### را در لیست نشان خواهد داد.

ایجاد جدول جهانی موقت

1- بدون بستن پنجره شامل نسخه خطی Create Local بر روی دکمه New Query روی Query Analyzer Toolbar برای باز کردن یک پنجره جدید خطی، کلیک کنید.

2- بر روی دکمه Loud Script روی Toolbar Query Analyzer کلیک کنید. Query Analyzer نشانگر جعبه گفتگوی فایل Query خواهد بود.

3- خطی را که در آن Create 6 Loud وجود دارد را انتخاب کنید و Open را کلیک کنید. تجزیه کننده سؤال خط انتخابی را Loud می کند.

4- دکمه Execute Query را که روی Query Analyzer Toolbar می باشد را کلیک کنید.

Query Analyzer جدول موقت را ایجاد می کند.

5- User Folder Table که مربوط به Tempdb Database در browser Object را انتخاب کنید.

6- F5 را برای راه اندازی مجدد نمایشگر Object browser فشار دهید. Query Analyzer نمایشگر dbo##6 Loud Table لیست خواهد بود.

آموزش SQLserver (بخش دهم)

ارتباط دادن جداول

Query هایی که در فصل 3 بررسی گردید سطرهایش از یک جدول تکی ترسیم گردیده است اما Query ها به ویژه می توانند مفید باشند برای ترکیب ستونهایی از چندین جدول یا نما که Joining Tables نامیده می شود و آن در شرطهای FROM یا WHERE از حالت SELECT انجام می گیرد. در این فصل ما بر روی ایجاد ارتباطها با استفاده از شرط FROM که روشی توصیه شده است متمرکز خواهیم شد.

شناختن شرط FROM

همان طوری که ما دیده ایم، ساختار پایه ای از شرط FROM به آسانی نام از یک جدول یا نما ساده را فراهم می سازد. اما برای دسترسی پیدا کردن به توانایی از مدل ارتباطی، ما باید به بازیابی ستونها از جداول چندگانه و نماها در یک Query خاص قادر باشیم. شرط FROM یک مکانیزمی برای انجام آن با استفاده از ساختار دستور زیر:FROMON

اپراتور پیوند انواع پیوند برای به انجام رسیدن را تشریح می کند. Server SQL پیوندهای داخلی و خارجی همه نوسانات را پشتیبانی می کند، همان طوری که در بخش بعدی خواهیم دید. شرایط پیوند یک تعبیری می باشد شبیه به ملاک که در شرط WHERE استفاده شده است. آن مشخص می کند که چگونه سطرها در دو جدول ارتباط خواهند یافت. بیشتر پردازشگرهای ربطی روی پایه ای از عبارتهای برابری مانند B ستون = A ستون به انجام می رسند. اما SQL Server هر اپراتور منطقی را پشتیبانی کرده و شرایط پیوند می تواند به طور دلخواه پیچیده باشد، با عبارات چندگانه پیوند یافته که از حرف ربط AND یا OR از همان راهی که یک شرط WHERE می تواند ملاک انتخابی چندگانه را شامل باشد استفاده کند. عبارت پیوند می تواند برای اضافه کردن جداول و نماهای اضافی برای Query تکرار شود. ساختار دستور برای پیوند جداول چندگانه عبارت است از: FROM

ON

یک حدود فرضی 256 جدولی برای هر Query وجود دارد، اما آن به حد زیاد غیر محتمل است که شما اصلاً نیاز به 5 یا 6 نیاز ندارید و 2 یا 3 بیشتر معمول می باشد. در حقیقت اگر شما نیاز به اتصال بیش از 10 جدول در یک Query باشید، شما باید به دقت طرح پایگاه داده آن را نگاه کرده برای اینکه مطمئن شوید که آن به طور صحیح به حالت عادی در آمده است.

ایجاد کردن پیوندها

پیوندها می توانند در Designer Query با استفاده هر کدام از قاب Grid یا قاب SQL ایجاد گردند. قاب Grid اغلب آسانتر می باشد اگر شما جداولی که رسماً در طرح پایگاه داده مربوط شده اند پیوند دهید، از موقعی که Query Designer پیوندی بین آنها به طور خودکار ایجاد خواهد کرد. اما به طور معمول قاب SQL با انعطاف پذیری بیشتر برای شما فراهم می گردد.

نامگذاری شئی ها

زمانی که شما با یک جدول یا نمای تکی کار می کنید آنجا می تواند هیچ ابهامی در حدود منابع از یک ستون نداشته باشد از موقعی که همه نام ستونها در یک جدول باید منحصر به فرد باشند. هنگامی که شما برای بار اول کار کردن با چندین جدول در یک Query را شروع می کنید یا شما باید برای مشخص کردن نام ستونها صریحاً مواظب باشید. مشخصات کامل برای هر شئی پایگاه داده چهار معرفه را در بردارد. نام سرور، نام پایگاه داده، نام صاحب، نام شئی. معرفه ها به وسیله پریودها جداسازی می گردند. بنابراین نام واجد شرایط از جدول Oils در سیستم من Bunny.Aromatherary.dbo.Oil می باشد. مقداری از شئی ها مانند نماها و جداول شامل شئی های دیگر می باشد. برای رجوع به یکی از این شئی های گنجانده شده (در این حالت، ستونها). شما به آسانی نام آن را به نام شئی ضمیمه می کنید. نام واجد شرایط کامل از ستون OilID از جدول Oils (Bunny.Aromatherapy.dbo.Oils.Oil ID) می باشد. خوشبختانه شما فقط نیاز به مشخص کردن کافی از درجه بندی ابهام اجتناب پذیر دارید.

در یک Query مبنی بر یک جدول تکی، برای مثال نام ستون به وسیله خودش دارای مشخصات کافی می باشد. اگر یک Query به بیش از یک جدول مربوط گردد، اگر چه جداول دارای ستونهایی با یک نام باشد. شما باید نام جدول را در نام شئی (Oils.OilID (Object ، OilPropertise.OilID که تمایز را به طور کامل روشن می سازد لحاظ کنید.

پیوندهای داخلی

بیشترین فرمهای رایج پیوند یک پیوند داخلی می باشد. یک پیوند داخلی فقط آن سطرهایی که شرایط پیوند TRUE را باز می گرداند باز خواهد گرداند.

پیوند دو جدول با استفاده از قاب دیاگرام

1- Query Designer جدول Oils را به وسیله کلیک راست کردن نام آن در قاب Details باز کرده روی جدول Open رفته و همه سطرهای بازگشتی را انتخاب می کنیم.

2- قاب دیاگرام را به وسیله کلیک کردن دکمه قاب دیاگرام روی نوار ابزار Query Designer نشان می دهیم.

3- دکمه Add Table را روی نوار ابزار Query Designer کلیک می کنیم. Query Designer کادر محاوره ای Add Table را نشان می دهد.

4- جدول PlantTypes را در لیست جدول انتخاب کرده و Add را کلیک می کنیم. SQL Server جدول را به Query اضافه می کند.

5- Close را برای بستن کادر محاوره ای Add Table کلیک می کنیم.

6- دکمه قاب SQL را در نوار ابزار Query Designer کلیک می کنیم. Designer Query

قاب **SQL** را نشان می دهد.

7- علامت × را بعد از کلید واژه **SELECT** حذف می کنیم.

8- دکمه قاب **SQL** را در نوار ابزار **Query Designer** کلیک می کنیم. (**OK** را کلیک کرده اگر **Query Designer** یک متن خطا درباره ساختار دستور **SELECT** نشان دهد). **Query Designer** قاب **SQL** را پنهان می سازد.
مهم: زمانی که شما **Designer Query** را باز می کنید، حالت **SQL** معمولاً × را انتخاب می کند. انتخاب کردن ستونهای مشخص در قاب دیاگرام سبب می شود که آنها به لیست ستون اضافه شوند. مایکروسافت آن را به صورت یک خصیصه در نظر می گیرد.

9- در قاب دیاگرام ستونهای **OilID** و **OilName** را در جدول **Oils** و ستون **PlantType** را در جدول **PlantType** انتخاب می کنیم.

10- دکمه **Run** روی نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** مقادیر **Planttype** را برای هر **Oil** نشان می دهد.

ارتباط دو جدول به وسیله قاب **SQL**

1- قاب دیاگرام را پنهان کرده و قاب **SQL** را به وسیله کلیک کردن دکمه ها روی نوار ابزار **Query Designer** نشان می دهیم.

2- حالت **SQL** موجود را با عبارت زیر جایگزین می کنیم.

```
SELECT Oils.OilID,Oils.Oil Name,PlantParts.PlantPart
FROM OilsINNER Join
PlantParts ON Oils.PlantPartID=PlantParts.PlantPart ID
```

3- دکمه **Run** را در نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** مقادیرها برای هر سطر **Oils Plant Part** را نشان می دهد.

ارتباط جداول چندگانه با استفاده از قاب دیاگرام

1- قاب **SQL** را پنهان کرده و قاب دیاگرام را نمایش می دهیم.

2- دکمه **Add Table** را روی نوار ابزار **Query Designer** کلیک می کنیم. **Query**

Designer کادر محاوره ای **Add Table** را نشان می دهد.

3- جدول **Planttypes** را در لیست جداول انتخاب می کنیم. **Add** را کلیک کرده **SQL Server** جدول را به **Query** اضافه می کند.
4- **Close** را برای بستن کادر محاوره ای **Add Table** کلیک می کنیم.

5- در قاب دیاگرام ستون **Planttype** را در جدول **Planttypes** برای اضافه کردن ستون به **Query** کلیک می کنیم.

6- دکمه **Run** در نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** هر دوی ستونهای **PlantPart** و **Planttype** را برای هر **Oil** نشان می دهد.

ارتباط جداول چندگانه با استفاده از قاب SQL
1- قاب دیاگرام را پنهان کرده و قاب **SQL** را نشان می دهیم.
2- عبارت **SELECT** موجود را با عبارت زیر جایگزین می کنیم.

```
SELECT Oils.Oil ID.Oils.Oil Name.odors.odor
FROM Oils
INNER Join Oilodors on Oils.OilID=Oil odors.OilID
INNER Join odors on Oilodors.odor ID=odors.odorID
```

3- دکمه **Run** را در نوار ابزار **Designer Query** برای اجرای **Query** کلیک می کنیم.

4- پنجره **Query Designer** را می بندیم.

ارتباطات خارجی
بعضی مواقع شما می خواهید که یک **Query** همه سطرها یک یا چندین جدول را باز گرداند، خواه آنها سطرهای ارتباطی در جداول دیگر داشته باشند و یا خیر. که با استفاده از یک ارتباط خارجی به انجام رسیده که می تواند سه گونه باشد: چپ، راست و کامل. یک ارتباط خارجی همه سطرهایی از جدول چپ در شرط **JOIN** و فقط آن سطرهایی از جدول راست برای اینکه شرایط ارتباط **TRUE** می باشد را باز خواهد گرداند.
دستور ساختار برای یک ارتباط خارجی عبارت است از:
Left Outer Join Right able on FROM Left Table
برای مثال عبارت **SELECT** زیر، همه سطرها در جدول **Oils** را باز می گرداند و مقادیر **PlantPart** از جدول آنجایی که **PlantPart** مشخص شده جفت می شود. آنجا سطرهای ارتباطی در جدول

Plant Parts وجود ندارد و **Query**، **Null** را به عنوان مقدار **PlantPart** برای آن سطر باز می گرداند.

```
SELECT Oils.Oil Name.Plant Parts.Plant Part
FROM Oils Left Outer Join
Plant Parts on Oils.Plant Part ID=Plant Parts.Plant Part ID
```

یک ارتباط خارجی سمت راست مقابل یک ارتباط خارجی سمت چپ می باشد. آن همه سطرها از جدول راست در شرط **JOIN** را باز می گرداند و مقادیر ارتباطی را از جدول چپ ارتباط می دهد. نظر به اینکه یک ارتباط کاملاً خارجی همه سطرها از دو جدول را با هم هماهنگ می سازد آنجایی که امکان پذیر باشد.

ایجاد کردن یک ارتباط خارجی چپ با استفاده از قاب دیاگرام

1- **Query Designer** را برای جدول **Oils** به وسیله کلیک راست کردن نام جدول در قاب **Details** باز کرده، روی جدول **Open** رفته و همه سطرهای بازگشتی را انتخاب می کنیم.

2- قاب دیاگرام را نشان می دهیم.

3- دکمه **Add Table** را در نوار ابزار **Query Designer** کلیک می کنیم. **Query Designer** کادر محاوره ای **Add Table** را نشان می دهد.

4- **Cautions** و **Oil Cautions** در لیست جدول را انتخاب و سپس **Add** را کلیک می کنیم. **Query Designer** جدولی برای **Query** اضافه می کند.

راهنمایی: شما می توانید یک ارتباط خارجی چپ با دو جدول ایجاد کنید. ما حالت سوم را در ایجاد استفاده می کنیم با جدول **OilCautions** که به عنوان یک جدول الحاقی عمل می کند که ارتباط چندگانه بین **Oils** و **Cautions** را حل می کند.

5- **Close** را برای بستن کادر محاوره ای کلیک می کنیم.

راهنمایی: شما می توانید جداول را در قاب دیاگرام برای پاک کردن نمایشگر درج کنید.

6- دکمه قاب **SQL** را در نوار ابزار **Query Designer** کلیک می کنیم. **Query Designer** قاب **SQL** را نشان می دهد.

7- علامت × را در کلید واژه **SELECT** حذف می کنیم.

8- دکمه قاب **SQL** را در نوار ابزار **Query Designer** کلیک می کنیم. (**OK** را کلیک کرده اگر **Query Designer** یک متن خطا درباره ساختار از عبارت **SELECT** نشان دهد) **Query**

Designer قاب **SQL** را پنهان می سازد.
مهم: زمانی که شما **Query Designer** را باز می کنید عبارت **SQL** پیش فرض معمولاً × را انتخاب می کند. ستون ویژه که در قاب دیاگرام انتخاب شده سبب می شود که آنها برای لیست ستون اضافه شوند. مایکروسافت این را به عنوان یک ویژگی در نظر می گیرد.

9- در قاب دیاگرام، ستونها از **OilName** و **OilID** را از جدول **Oils** و ستون **Caution** را از جدول **Cautions** برای خروجی انتخاب می کنیم.

10- دکمه **Run** را در نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** فقط آن **Oils** که **Cautions** دارد را نشان می دهد.

11- خط ارتباطی بین جداول **Oil Cautions** و **Oils** را به وسیله کلیک کردن آن انتخاب می کنیم و سپس دکمه **Properties** را در نوار ابزار **Query Designer** کلیک می کنیم. **Query Designer** کادر محاوره ای **Join Properties** را نشان می دهد.

12- **All Rows From Oils** را انتخاب می کنیم.

راهنمایی: همه سطرها از **Oil Cautions** یک ارتباطی خارجی سمت راست ایجاد خواهد کرد و هر دوی گزینه ها را انتخاب کرده که یک ارتباط کامل خارجی ایجاد می کند.
13- **Close** را برای بستن کادر محاوره ای کلیک می کنیم. **Designer Query** خط ارتباطی را برای انعکاس مشخصات ارتباطی جدید تغییر می دهد.

14- دکمه **Run** در نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** همه سطرها در جدول **Oils** را نشان می دهد و مقادیر از جدول **Cautions** را ارتباط می دهد.

ایجاد یک ارتباط خارجی سمت راست با استفاده از قاب **SQL**
1- قاب دیاگرام را پنهان کرده و قاب **SQL** را در **Query Designer** نشان می دهیم.

2- حالت **SELECT** موجود را با عبارت زیر جایگزین می کنیم.

```
SELECT Oils.Oil Name.Properties.Property
FROM Oils
Right OUTER JOIN Oil Properties on
Oils.Oil ID=Oil Properties.Oil ID
INNER JOIN Properties on
Oil Properties.Property ID=Properties.Property ID
```

3- دکمه **Run** را در نوار ابزار **Query Designer** برای اجرای **Query** کلیک می کنیم. **Query Designer** شامل همه سطرها از جدول **Oil Properties** با مقادیر ارتباطی از جدول **Oils** می باشد.

4- پنجره **Query Designer** را می بندیم.

UNIONS

آخرین نوع از ارتباط به صورت **Union** شناخته می گردد. یک **Union** نتایجی از دو عبارت **SELECT** متمایز را در داخل یک تنظیم از سطرها ترکیب می کند. ارتباطات داخلی و خارجی ستونهایی از دو جدول درگیر شده در یک سطر تکی را با هم ترکیب می کند یک **Union** سطرهایی از دو جدول در یک ستون تکی را به هم ترکیب می کند. شما می توانید تصور کنید که تنظیمات دو سطر را گرفته و یکی را در بالای دیگری باز گردانید. اگر چه قاعده اصلی از سطرها باز گردانده شده به وسیله شرط **ORDER BY** مشخص می گردد. ساختار دستور یک **Union** از پیوندها متفاوت می باشد. یک **Union** ساختار دستوری به صورت زیر دارد:

```
SELECT FROM
UNION [All]
SELECT FROM
[ORDER BY ]
```

شما می توانید بسیاری از حالتهای **UNION SELECT** را همان طوری که شما برای یک **Query** دوست دارید اضافه کنید. (این موضوع برای **256** جدول محدود می باشد) اما همه عبارتهای **SELECT** باید همان تعداد از ستونها را از انواع سازگار یا شبیه در همان دستور باز گرداند. اولین عبارت **SELECT** نامهای ستون را مشخص خواهد کرد و شرط **BY ORDER** از آخرین عبارت **SELECT** دستور **Sort** را تعیین خواهد کرد. به طور پیش فرض، **Server SQL** سطرهای چندگانه ای از نتایج را از یک **Union Query** برمی دارد. اگر شما **All Union** را مشخص کنید، ولی سطرهای چندگانه حفظ خواهد شد.

ایجاد کردن یک **UNION**

1- **Query Designer** را به وسیله کلیک کردن جدول **Properties** در قاب **Details** باز کرده روی جدول **Open** رفته و همه سطرهای بازگشتی را انتخاب می کنیم.

2- قاب **SQL** را نشان می دهیم.

3- حالت **SQL** موجود را با عبارت زیر جایگزین می کنیم.

```
SELECT Property Table AS Table Name.Property ID AS ID
Property AS Quality From Properties
UNION
SELECT odor Table.odor ID odor From odors
ORDER BY Quality
```

4- دکمه **Run** در نوار ابزار **Designer Query** را برای اجرای **Query** کلیک می کنیم. **Query Designer** نتایجی از دو عبارت **SELECT** را با هم ترکیب می کند.

آموزش SQLserver (بخش یازدهم)

ایجاد کردن Check Constraint

شناخت Check Constraint

یکی از مهمترین بازرسی های طراحی پایگاه داده یکپارچه سازی داده ها می باشد. قاعده یکپارچگی داده ها تضمین می کند که داده های ایجاد شده در پایگاه داده اگر درست نباشد حداقل قابل قبول می باشد. چندین سطح از یکپارچگی داده ها وجود دارد. در درس 7 ما درباره یکپارچگی رابطه ای مطالعه کردیم که تضمین می کند اجتماع بین جداول ایجاد و به طور صحیح نگهداری می گردند.
Check Constraintsبرای اجرای دو فرم اضافی از یکپارچگی پایگاه داده استفاده می شود. domain Integrity و entity Integrity . در اصطلاحات رابطه ای که Domain محدوده ای از مقادیری است که یک ستون می تواند داشته باشد. نوع داده های یک ستون یکی از ویژگیهای از یک Domain می باشد، اما تعریف نوع داده معمولاً کافی نیست. برای مثال، یک ستون Smallint می تواند شامل مقادیر صحیح از 32/768- تا 32/767 می باشد که ممکن نوع داده مناسبی برای یک ستونی که شامل سالی است که یک کارمند مدرک دانشگاهی را در دریافت می کند باشد. اما حدود اصلی مقادیر ستون Year Degree Awarded بیشتر محدود شده و بین 1900 و سال جاری می باشد. شما از یک Check Constraints استفاده می کنید، برای تخمین اینکه هیچ کس به طور واقعی مقادیر 1543 یا 2075 را به عنوان مقدار ستون وارد نمی کند. Entity Integrity Constraints جامعیت از موجودی خودش را اجرا می کند. مهمترین محدودیتهای یکپارچگی موجودیت آن است که هر موجودیت Entity Integrity باید به طور واحد قابل شناسایی باشد. این محدودیت به وسیله مشخص کردن یک کلید اصلی برای جدول انجام می گیرد. یکپارچگی موجودیت همچنین می تواند درگیر ارزیابی شرطی از چندین ستون در یک جدول باشد و این نوع از محدودیت اغلب بیشتر با استفاده از Check Constraints انجام می گیرد. برای مثال، اگر یک جدول شامل ستونهای کشور و ایالت باشد شما ممکن از یک Check Constraints برای مشخص کردن اینکه ارزش ستون State، AZ" معتبر می باشد فقط اگر ستون Country شامل مقادیر USA باشد. Check Constraint به عنوان Boolean Expressions شناخته می شود یک Expressions Boolean برای مقادیر True یا False ارزیابی می گردد. Boolean Expressions را در درس 13 می آموزیم. در این درس ما از عبارت LEN()>=4 استفاده می کنیم. LEN یک تابع Transact-SQL می باشد که تعدادی کارکترهای یک رشته را برمی گرداند. بنابراین عبارت LEN()>=4 ارزش آن اگر شامل 4 یا بیشتر کارکتر باشد ارزش آن True و اگر کمتر از 4 باشد ارزش آن False می گردد.

ایجاد کردن Check Constraint)

مانند ایندکس ها و پیوندها شما می توانید Check Constraints را با استفاده از کادر محاوره ای Properties از Table Designer ایجاد کنید.
1- Table Designer را برای جدول Oils به وسیله کلیک راست کردن نام جدول در قاب Details باز

کرده و Table Design را انتخاب می کنیم. SQL Server , Table Designer را باز می کنیم.

2- دکمه Constraints را کلیک کرده SQL Server کادر محاوره ای Table Designer Properties را با صفحه خصوصیات Constraints Check نشان داده شده باز می کنیم.

3- New را کلیک کرده SQL Server , Ck-Oils را به عنوان نام محدودیت پیشنهاد می دهد. برای این مثال این نام را می پذیریم.

4- Len(Oil Name)>=4را به عنوان عبارت محدودیت وارد می کنیم.

راهنمایی: اگر شما یک Check Constraint جدید را ایجاد می کنید و مراقب نیستید که آیا داده های موجود متابعت می شوند، شما می توانید بگوید به SQL Server که از ایجاد کردن داده به وسیله چک نکردنExisting Data On Creation Check از محدودیت چشم پوشی کند.
5- Close را کلیک کرده Server SQL کادر محاوره ای Designers Properties Table را می بندد.
6- دکمه Save را کلیک کرده SQL Server کنترل می کند که همه سطرها در جدول با Check Constraint مواجه می شوند و سپس محدودیت را ذخیره می کنیم.

مدیریت Check Constraints
به عنوان بخشی از طرح پایگ اه داده، Check Constraints نباید تحت شرایط نرمال نیاز به مقدار زیاد نگهداری داشته باشد. شما آنها را یکباره تعریف کرده زمانی که پایگاه داده را ایجاد می کنید. اگر چه طرحهای پایگاه داده به تدریج تغییر خواهد کرد. تغییرات Check Constraints تغییر خواهد کرد. Enterprise Manager نگهداری محدودیتها را آسان می سازد.

تغییر دادن Check Constraint
Table Designer مکانیزمی برای تغییر متن از یک Check Constraint از همان کادر محاوره ای که شما برای ایجاد آن استفاده کردید فراهم می کند.

تغییر متن محدودیت
1- اگر Designer Table برای جدول Oils هنوز از تمرین قبل باز نمی باشد به وسیله کلیک راست کردن نام جدول در قاب Details آن را باز کرده وDesign Table را انتخاب می کنیم. Table Designer , SQL Sarver را باز می کند.

2- دکمه Constraints را کلیک کرده SQL Sarver کادر محاوره ایTable Designer را با صفحه خصوصیات Check Constraint باز می کند.

3- اطمینان حاصل کنید کهCk – Oilsدر Constraint Combo Box انتخاب شده نشان داده شده است.
4- متن محدودیت را برایLEN (Oil Name) >2 به عنوان عبارت محدودیت جدید تغییر می دهیم.

5- Close را کلیک کردهSQL Server کادر محاوره ایTable Designer's Properties را می بندد.
6- دکمه Save را کلیک کرده SQL Server همه سطرها در جدول را که با Check Constraint جدید مواجه است را کنترل کرده و سپس محدودیت را ذخیره می سازد.

نگهداری Check Constraints
مانند دیگر خصوصیات جدول دیگر، Check Constraints در کادر محاوره ای Properties ازTable Design نگهداری می گردد.

تغییر نام یک Check Constraints
1- Designer Table برای جدول Oils به وسیله کلیک راست کردن نام جدول در قاب Details باز کرده و Design Table را انتخاب می کنیم. SQL Server , Table Designer را باز می کنیم.

2- دکمه Constraints را کلیک کرده SQL Server کادر محاوره ای Designers Table Properties را با صفحه خصوصیات Check Constraint نشان داده شده باز می کند.

3- Ck-Oils را در فیلد Constraint Name انتخاب کرده و آن را به Ck-Deleteme تغییر می دهیم.

4- Close را کلیک کرده SQL Server کادر محاوره ای Properties را می بندد.
5- دکمه Save را کلیک کردهSQL Server همه سطرها در جدول که باCheck Constraint مواجه شده اند را کنترل می کند و سپسConstraint ذخیره می گردد.

حذف یک Check Constraint
1- Table Designer برای جدول Oils را به وسیله کلیک راست کردن نام جدول در قاب Details باز کرده و جدول Design را انتخاب می کنیم. SQL Server , Table Designer را باز می کند.

2- دکمه Constraints را کلیک کرده SQL Server کادر محاوره ای Table Designers Properties را با صفحه خصوصیاتCheck Constraint نشان داده شده باز می کنیم.

3- اطمینان حاصل کنید که Ck-Deleteme در فیلد محدودیت انتخاب شده می باشد و سپس Delete را کلیک می کنیم. SQL Server محدودیت را برمی دارد.

4- Close را کلیک کرده SQL Server کادر محاوره ای Properties را می بندد.
5- دکمه Save را کلیک می کنیم. SQL Server محدودیت را برمی دارد.
6- Table Designer را می بندیم.