

آموزش Sql Server

فصل اول

معرفی SQL و دستورات عمومی آن
توسط SQL میتوان درون یک بانک اطلاعاتی پرس و جو کرده (Query) و نتیجه را برگرداند. بانک اطلاعاتی شامل آیچکتی به نام جدول (Table) میباشد. رکوردها در بانکهای اطلاعات در جداول ذخیره میگرددند. جدول شامل سطر و ستون میباشد.

در زیر میتوان یک جدول را مشاهده کرد :
Name Address City Last Name First

مثال:

Select Last name from persons

پر کاربرد ترین دستورات SQL شامل موارد زیر است :

SELECT استخراج یک داده از بانک اطلاعاتی

UPDATE به روز رسانی یک داده درون بانک

DELETE پاک کردن یک داده از بانک اطلاعاتی

INSERT وارد کردن یک داده جدید به بانک اطلاعاتی

همچنین در SQL میتوان داده هایی نیز تعریف کرد :

CREATE TABLE ایجاد یک جدول جدید

TABLE ALTER تغییر دادن یک جدول

DROP TABLE پاک کردن یک جدول

CREATE INDEX ایجاد یک اندیس

DROP INDEX پاک کردن یک اندیس (کلید جستجو)

SQL و ASP Active - Server Pages :

SQL یکی از قسمتهای خیلی مهم ASP میباشد زیرا در ASP برای کار با بانکهای اطلاعاتی از SQL استفاده میشود. و توسط تکنولوژی ADO میتوان از SQL در ASP استفاده کرد.

دستور SELECT :

SELECT column-name(s) FROM table-name

مثال :

نام جدول Persons است

Last Name First Name Address City

Hasani Ali Esfahan

دستور Select همراه با شرط

column FROM table WHERE column condition value SELECT

عملگر ها در SQL :

عملگر مفهوم

= معادل بودن

<> برابر نبودن

< بزرگتر

دستور DISTINCT (جداسازی) :

FROM table-name (SELECT DISTINCT column-name(s)

مثال :

Company Order Number است Order
۳۴۱۲ sega

دستور Order By :

برای مرتب کردن سطرها
نام جدول Order است Company Order Number
۳۴۱۲sega
۵۶۷۸ABS Shop
۳۳۱۲s ۳w
۶۷۷۸S ۳W
مثال :
SELECT Company , OrderNumber

دستور Insert :

) VALUES۲, Column ۱ INTO Table-Name (Column INSERT

مثال :
last name First name است Persons نام جدول Person WHERE Lastname='hasani DELETE FROM
city Adress
Esfahan ۴۰alian hasani NO
Tehran ۱۰Hasani Ali No

نتیجه :
last name First name Adress city
Esfahan ۴۰alian hasani NO
SEGA
S۳W
Trio

دسترسی به یک پایگاه داده از یک صفحه ASP :

۱- ایجاد یک ADO Connection به یک پایگاه داده
۲- باز کردن Connection پایگاه داده
۳- ایجاد یک
ADO Record Set ۴- باز کردن
Record set ۵- گرفتن داده هایی که نیاز داریم از
Record Set ۶- بستن
Record Set ۷- بستن
Connection

۱- ایجاد یک ADO Connection به یک پایگاه داده

۱-۱- روش C:/InetPub/wwwroot/nor.mdb DSN-LESS مسیری فایل نمونه
> .CreateObject ("ADODB .var conn=Server %
" Conn.Open,OLEDB. . Connection") Conn. Provider="Microsoft . Jet

< % ("C:/InetPub/wwwroot/nor.mdb

۲- روش ODBC

۲- روش ODBC : در این روش ابتدا باید يك ODBC Connection به Data Base ایجاد کنیم و سپس از طریق ADO به فایل DSN به طریق زیر Connect کنیم.
var conn=Server. CreateObject % >

ایجاد يك ODBC Connection به پایگاه داده MS Access :

۱- وارد شدن به ODBC از Control Panel
۲- انتخاب System DSN -۳- کلیک کردن روی دکمه ADD
۴- انتخاب Microsoft Access Driver و کلیک کردن روی دکمه Finish
۵- در مرحله بعد کلیک کردن بر روی دکمه Select و تعیین محل پایگاه داده
۶- دادن يك نام در قسمت Data Source Name
۷- کلیک کردن روی دکمه OK برای اینکه قادر باشیم اطلاعات يك پایگاه داده را بخوانیم
اطلاعات باید ابتدا در Record Set ، Load شوند. بنابر این بعد از ساختن يك Connection باید يك Record Set ایجاد کنیم.
مثال :
نام Data Base = nor.mdb
نام جدول =
conn= Server. CreateObject % > مسیر فایل C:/Inetpub/wwwroot/nor.mdb Data Base Customer ("ADODB.Connection")
ایجاد Connection
Connection (""" conn . Open ("C:/WebData/ nor.mdb", conn.Provider="Microsoft.Jet.OLEDB.
("rs=Server. CreateObject ("ADODB.Recordset
Record Set ایجاد Record Set rs. Open("Customer", conn) بازکردن % < در این مثال تمام محتویات جدول Customer به rs ریخته میشود.
حال میخواهیم در انتخاب محتویاتی از Customer که میخواهند به rs انتقال داده شوند از دستور SQL استفاده کنیم.

ساختن يك Connection و Record Set و به کار بردن SQL :

"conn=Server. CreateObject ("ADODB .Connection set % >
." conn, .Provider="Microsoft.jet.OLEDB. .conn
C:/Inetpub/wwwroot/nor.mdb") (Record Se") Open

گرفتن داده هایی که نیاز داریم از Record Set :

بعد از اینکه Record Set را باز کردیم میتوانیم به داده هایی که نیاز داریم دسترسی داشته باشیم.
مثال :
دسترسی به فیلد name از جدول rs
rs(name

Response.write(rs(n rs از جدول name

بستن Connection و Recordset :

- براي بستن Recordset
- rs.close () - براي بستن Connection

بخش دوم

خلاصه سازی داده ها

هنگامی که دو جدول در یک پیوند یک به چند سهیم هستند. وسیله محرک Query مقادیری از سطر را از یک طرف برای ارتباط دادن سطرها در طرفهای دیگری تکرار می کند. بعضی مواقع آن دقیقاً چیزی است که شما می خواهید، اما اغلب شما می خواهید تکرار کردن سطرها را از چندین طریق دسته بندی یا خلاصه سازی. در این درس، ما به دو روش برای انجام آن که عبارتند از کلید واژه DISTINCT و شرط GROUP BY نگاه خواهیم کرد.

شناختن SELECT DISTINCT

یکی از اهداف طرح پایگاه داده ارتباطی برداشتن وابستگی داده ها می باشد. اما بیشتر پایگاه داده به طور ضروری مقادیر واقعی در چندین سطر را شامل خواهد شد. یک جدولی که شامل اطلاعات آدرس مشتری می باشد برای مثال احتمالاً کد ایالتی و کشوری برای چندین سطر خواهد داشت که نه اشتباه و نه تکراری می باشد. نظر به اینکه هر کد ایالتی نسبتی از یک مشتری می باشد. به همین نحو یک جدول از چندین جهت از یک ارتباط یک به چند ممکن است هر مقدار کلید خارجی معلوم که چندین بار تکرار شده را داشته باشد. آن برای یک ارتباط یکپارچه از پایگاه داده لازم می باشد. اگر چه این تکرار می تواند بعضی مواقع نتایج Query را نامطلوب سازد. یک جدول مشتری با ۱۰۰۰ سطر با ۹۰ درصد مشتری از کالیفرنیا، Query زیر کد CA را ۹۰۰۰ بار خواهد باز گرداند که اصلاً یک نتیجه مفیدی نمی باشد.

```
State From Customer SELECT
```

کلید واژه Distinct در این موقعیت شما را کمک می کند. Distinct که درست بعد از SELECT قرار می گیرد به SQL Server دستور داده که سطرهایی چندگانه در قرارگیری نتایج را حذف نماید. بنابراین Query زیر هر کد ایالتی را فقط یک بار باز می گرداند به طور وضوح لیستی که شما جستجو می کنید.

```
State From Customer SELECT DISTINCT
```

راهنمایی: همتای کلید واژه All، Distinct، می باشد که Server SQL را برای بازگرداندن همه سطرها آگاه می سازد خواه آن واحد باشد یا خیر. از موقعی که این یک عملکرد پیش فرض از یک عبارت SELECT می باشد All به طور معمول استفاده نمی گردد. اما شما ممکن تصمیم بگیرید به در برداشتن آن اگر ساختار دستور Query را بیشتر قابل فهم سازید.

استفاده کردن از SELECT DISTINCT

کلید واژه Distinct می تواند در عبارت SQL از Query Distinct یا به وسیله تنظیمات Properties از Query مشخص گردد.

ایجاد کردن Distinct Query با استفاده از قاب دیاگرام

- ۱- 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.
- ۲- قاب دیاگرام را به وسیله کلیک کردن روی دکمه قاب دیاگرام در نوار ابزار Query Designer نشان می دهیم.
- ۳- دکمه Add Table را کلیک می کنیم. Designer Query کادر محاوره ای Add Table را نشان می دهد.
- ۴- Plantparts را در لیست جدول انتخاب می کنیم و سپس Add را کلیک می کنیم. Query Designer جدول را به Query اضافه می کند.
- ۵- Close را برای بستن کادر محاوره ای کلیک می کنیم.
- ۶- دکمه قاب SQL را در نوار ابزار Query Designer کلیک می کنیم. قاب SQL، Query Designer را نشان می دهد.
- ۷- علامت * بعد از کلید واژه SELECT را حذف می کنیم.

۸- دکمه قاب SQL را در نوار ابزار Query Designer کلیک می کنیم (OK) را اگر Query Designer یک متن خطا درباره ساختار دستور عبارت SELECT نشان می دهد کلیک می کنیم). Query Designer قاب SQL را پنهان می سازد.

مهم: زمانی که شما Designer Query را باز می کنید حالت SQL پیش فرض معمولاً SELECT* می باشد. ستون ویژه در قاب دیاگرام به سبب اینکه آنها به لیست ستون اضافه می شوند انتخاب می گردند.

Plantpart را از جدول Plantparts برای نشان دادن انتخاب می کنیم.

۱۰- دکمه Run را برای اجرای مجدد Query کلیک می کنیم. Query Designer هر مقدار Plantpart را چندین بار لیست می کند.

۱۱- در یک ناحیه خالی از قاب دیاگرام کلیک راست کرده و Properties را انتخاب می کنیم. Query Designer کادر محاوره ای Properties را نشان می دهد.

۱۲- گزینه مقادیر Distinct را انتخاب می کنیم.

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

۱۴- دکمه Run را برای اجرای مجدد Query کلیک می کنیم. Query Designer هر مقدار را فقط یک بار نشان می دهد.

ایجاد کردن یک SELECT DISTINCT با استفاده از قاب SQL

۱- قاب دیاگرام را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- عبارت SELECT موجود را با متن زیر جایگزین می کنیم.

```
SELECT DISTINCT Plant Types Plant Type
```

```
INNER Join FROM Oils
```

```
Plant Types ON Oils Plant Type ID=Plant Types Plant Type ID
```

۳- دکمه Run را برای اجرا مجدد Query کلیک می کنیم. Query Designer مقدار PlantType متمایز که به وسیله جدول Oils بازگشت شده را نشان می دهد.

شناختن GROUP BY

کلید واژه Distinct امر می کند. SQL Server را به بازگرداندن سطرهای واحد نظر به اینکه شرط GROUP BY، SQL Server را به ترکیب سطرها با مقادیر یکسان در ستون یا ستونهای مشخص شده در شرط در داخل سطر تکی امر می کند.

مهم: هر ستونی که در شرط GROUP BY شامل می باشد باید در خروجی Query شامل گردد.

شرط GROUP BY اغلب با یک aggregate Function استفاده می گردد. یک Function aggregate

محاسبات در مجموعه از مقادیر را به انجام می رساند و یک نتیجه تک مقداری را باز می گرداند. رایج ترین گزینه های به هم پیوسته استفاده شده در پرس و جویهای GROUP BY، MIN می باشد که کوچکترین مقادیر را در مجموعه باز می گرداند. MAX که بزرگترین مقادیر در مجموعه را باز می گرداند و COUNT که تعدادی از مقادیر را در یک مجموعه باز می گرداند.

استفاده کردن از GROUP BY

شرط GROUP BY می تواند مشخص شود با استفاده کردن از هر یک از قابها در Query Designer، اما قابهای SQL و Grid بیشترین کنترل را فراهم می سازد.

ایجاد کردن یک Query GROUP BY با استفاده از قاب Grid

۱- قاب SQL را پنهان ساخته و قاب Grid را نشان می دهیم.

۲- ستون OilName را برای Query اضافه می کنیم.

۳- دکمه Group By را در نوار ابزار Query Designer کلیک می کنیم. Query Designer یک ستون Group By برای شبکه اضافه می کند و هر دوی مقادیر را برای Group By قرار می دهد.

۴- سل Group By را برای تغییر دادن سطر OilName به Count تغییر می دهیم.

۵- دکمه Run را برای اجرا مجدد Query کلیک می کنیم. Query Designer تعدادی از Oils را برای هر PlantType نشان می دهد.

SQL با استفاده از قاب Group By Query
۱- قاب شبکه را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- عبارت SELECT موجود را با متن زیر جایگزین می کنیم.
part Count (Oils Oil Name) As Number Of Oils SELECT Plant parts Plant
FROM Oils INNER Join
parts ON Oils Plant part ID=Plant parts Plant part ID Plant
Plant part GROUP BY Plant parts

۳- دکمه Run را برای اجرای مجدد Query کلیک می کنیم. Designer Query مقدار oils برای هر Plantpart نشان می دهد.

استفاده کردن از شرط HAVING
شرط Having سطرهای بازگردانده شده به وسیله شرط GROUP BY را از همان راهی که یک شرط Where سطرهای بازگردانده شده را به وسیله شرط SELECT محدود می سازد. هر دو شرط Where و Having می تواند در یک عبارت SELECT شامل شود که شرط Where قبل از گروه بندی عملیات و شرط Having بعد از آن به کار برده می شود. ساختار دستور شرط Having همانند شرط Where می باشد به استثنای اینکه شرط Having می تواند شامل یکی از توابعی که در لیست ستونها از شرط SELECT قرار می گیرد. اگر چه شما باید توابع به هم پیوسته را تکرار کنید. برای مثال شرط Having که در حالت زیر استفاده شده صحیح می باشد.

Name) As Number Of Oils SELECT Plant parts Plant part Count (Oils Oil
FROM Oils INNER Join
part ID=Plant Parts Plant part ID Plant parts ON Oils Plant
GROUP BY Plant parts Plant part
<Count (Oils Oil Name HAVING
اگر چه شما نمی توانید از اسم مستعار برای تابع Count در شرط Having استفاده کنید. بنابراین شرط Having زیر درست نخواهد بود.
<HAVING Number Of Oils

ایجاد کردن یک Query با استفاده از HAVING در قاب Grid
۱- قاب SQL را پنهان ساخته و قاب Grid را نمایش می دهیم.

۲- Addo < را برای سل مورد نظر از ستون oil Name قرار می دهیم.

۳- دکمه Run در نوار ابزار Query Designer را برای اجرای مجدد Query کلیک می کنیم.

ایجاد کردن یک Query با استفاده از HAVING در قاب SQL
۱- قاب Grid را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- شرط Having را برای (Count (Oils Oil Name) HAVING) تغییر می دهیم.

۳- دکمه Run در نوار ابزار Query Designer را برای اجرای مجدد Query کلیک می کنیم. Query Designer فقط آن Plantparts که Oils اشتراکی کمتر از ۵ دارد را نشان می دهد.

بخش سوم

مرتب سازی و انتخاب کردن سطرها در فصل گذشته ما بیشترین فرمهای مقدماتی از حالت SELECT را مرور کرده و آنها را برای انتخاب ستونهای یک جدول استفاده کردیم. اما بیشتر مواقع شما بازگشت به سطرها در جدول پس زمینه یا نمایش در یک قاعده خاص و فقط بازگشت به یک زیر مجموعه از آنها را خواهید خواست. شرطهای ORDER BY و WHERE در این درس بررسی شده و اجرای آن برای شما فراهم شود.

شرط ORDER BY
شرط ORDER BY The یک ترکیب گزینه ای از یک حالت SELECT می باشد. آن به شما برای مشخص کردن ترتیب در آن سطرهایی که بازگشت خواهند کرد اجازه می دهد. ستونهای چندگانه می توانند مشخص شوند و سطرها می توانند به صورت ترتیب افزایشی یا کاهششی باز گردانده شوند.

مرتب سازی سطرها ساده ترین فرم در شرط ORDER BY فراهم می سازد یک نام ستون تکی که برای مرتب کردن سطرها که به وسیله Query باز گردانده می شود استفاده خواهد شد.

مرتب کردن سطرها با استفاده از قاب Grid
۱- Query Designer را برای جدول Oils به وسیله کلیک راست کردن نام آن در قاب Details باز می کنیم. به زیر منوی جدول Open رفته و همه ستونهای بازگشتی را انتخاب می کنیم. SQL Server ، Query Designer را برای جدول باز می کند.
۲- قاب Grid را به وسیله کلیک راست کردن دکمه قاب Grid در نوار ابزار Query Designer نشان می دهیم.

۳- فقط ستونهای OilID ، Oil Name ، LatinName را برای نمایش انتخاب می کنیم. Designer Query محتویات قاب Results را که کم رنگ شده برای نشان دادن اینکه با مشخصات Query زیاد مربوط نمی باشد نشان می دهد.

۴- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer فقط ستونهای مشخص شده را نشان می دهد.

۵- نوع Sort فیلد Oil Name را صعودی قرار می دهیم.

۶- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer سطرهای مرتب شده با Oil Name را نشان می دهد.

مرتب سازی سطرها با استفاده از قاب SQL
۱- قاب Grid را پنهان ساخته و قاب SQL را به وسیله کلیک کردن دکمه ها روی نوار ابزار Query Designer نشان می دهیم.

۲- DESC را بعد از شرط ORDER BY OilName اضافه می کنیم.

راهنمایی: کلید واژه DESC به SQL Server برگرداندن سطرها به ترتیب نزولی را ابراز می کند. کلید واژه ASC که گزینه ای می باشد سطرها را به ترتیب صعودی برمی گرداند.

۳- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer نتایج را که با OilName ذخیره شده به صورت نزولی نشان می دهد.

مرتب سازی ستونهای چندگانه
شما می توانید ستونهای چندگانه را در شرط ORDER BY مشخص کنید. زمانی که ستونهای چندگانه مشخص می گردند ترتیب ستونها نتایج SQL Server را که به وسیله اولین ستون و سپس به وسیله دومین ستون و بنابراین چهارمین ستون مرتب خواهد شد مشخص می کند.
راهنمایی: تمرینها در این بخش از جدول OilOdors استفاده می کند که به صورت جدول الحاقی انجام وظیفه می کند که ارتباط چندبه چند بین جدول Oils و Odors را تجزیه و تحلیل می کند. معمولاً شما

Grid

- ۱- پنجره شماره ۱ را از منوی Window برای برگشت به درخت Console انتخاب می کنیم.
- ۲- Query Designer را برای جدول Oil Orders به وسیله کلیک راست کردن نام آن در قاب Details باز کرده روی زیر منوی جدول Open رفته و همه سطرهای بازگشتی را انتخاب می کنیم. SQL Server، Query Designer را برای جدول باز می کند.
- ۳- قاب Grid را به وسیله کلیک کردن دکمه قاب Grid روی نوار ابزار Query Designer نشان می دهیم.
- ۴- * را در قاب Grid با نامهای دو فیلد جایگزین می کنیم. SQL Server محتویات از قاب Results را برای نشان دادن اینکه آن با مشخصات Query زیاد در ارتباط نیست کم رنگ می کند.
- ۵- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer فقط ستونهای مشخص شده شما را نشان می دهد.
- ۶- نوع Sort هر دوی ستونها را صعودی قرار می دهیم.

۷- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer سطرهای مرتب شده را با اولین OilID و سپس با OdorID در داخل OilID نشان می دهد.

مرتب کردن سطرها با استفاده از قاب SQL

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

۲- ستونها را در شرط ORDER BY معکوس می کنیم.

۳- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer نتایج مرتب شده را با اولین OdorID و سپس با OilID نشان می دهد.

۴- پنجره Query Designer را می بندیم.

شرط WHERE

با استفاده از شرط اختیاری WHERE از حالت SELECT شما می توانید یک زیر مجموعه از سطرها که باز گردانده می شوند را مشخص کنید. برای مثال شما ممکن است بخواهید فقط مشتریهایی که بیش از \$۱۰۰۰ در ۱۲ ماه قبل خرج کرده اند را ببینید یا اینکه فقط نامهای Oil که با حرف R شروع می شوند را ببینید. شما این ملاکها را با استفاده از شرط WHERE مشخص خواهید کرد.

شرط WHERE BASIC

Operator Meaning

مساویست با = بزرگتر از > کوچکتر از < مساویست با = کوچکتر یا مساویست با <= مساوی نیست با <>

کلید برای شرط WHERE یک ملاک انتخابی می باشد که مشخص می کند که کدام سطرها باز خواهند گشت. ساختار پایه ای از یک شرط WHERE، WHERE می باشد. SQL Server یک حدود کاملی از اپراتورهای مقایسه ای را به طوری که در جدول ۱B نشان داده شده فراهم می سازد.

مشخص شده در شرایط WHERE می تواند یک ارزش دائمی باشد مانند Red "" یا ۱۰۰۰۰ یا می تواند باشد یک عبارتی که یک ارزش مانند GETDATE. را باز گرداند. به طور شبیه ارزش می تواند دستی ساخته شوند با استفاده از تابعهای Transact-SQL مانند LEFT که یک تعدادی از کاراکترهای مشخص شده از چپ یک رشته را باز می گرداند، ما تمام جزئیات توابع را در درس ۲۴ مرور خواهیم کرد.

مشخص کردن یک شرط WHERE با استفاده از قاب Grid

۱- پنجره شماره ۲ را از منوی Window برای بازگشت به پنجره Query Designer که ما زودتر در این درس

WHERE با استفاده از عبارت Format شما می توانید همچنین عبارات را با استفاده از عملگر منطقی OR یا AND ترکیب کنید. یک شرط WHERE از فرمت FormatOR سطرهایی که به ملاک مربوط می گردند خواهد بازگرداند، نظر به اینکه یک شرط WHERE از Format And سطرهایی که به هر دو ملاکها مربوط می گردند را باز خواهد گرداند.

مشخص کردن ملاک پیچیده با استفاده از OR
۱- شرط WHERE رادقاب SQL به (WHERE (OilName='Rosemary' OR OilName='Sagy) تغییر می دهیم.

۲- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer دو سطر را نشان می دهد.

مشخص کردن ملاک پیچیده با استفاده از AND
۱- شرط WHERE را در قاب SQL به (WHERE Oil Name LIKE 'Rose' WHERE)% تغییر می دهیم.

۲- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Query Designer، ۴ سطر را نشان می دهد.

۳- شرط WHERE را در قاب SQL به (OilID > ۴۰) AND (OilName LIKE Rose WHERE)% تغییر می دهیم.

۴- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می کنیم. Designer Query سطر Roso Otto را شامل می گردد.

بخش چهارم

بازیابی سطرها

بیشتر برنامه های پایگاه داده از یک برنامه ریزی محیطی از قبیل مایکروسافت Access یا مایکروسافت Visual Basic برای فراهم آوردن رابط کاربر استفاده خواهند کرد. اگر چه Manager Enterprise یک ابزار ساده به نام Query Designer فراهم می آورد که به شما به آسانی اجازه می دهد داده ها را در یک یا چندین جدول در پایگاه داده Aromatherapy استفاده می کنید.

استفاده کردن از Query Designer

حتی با وجود اینکه داده در یک پایگاه داده SQL Server ذخیره شده به سختی در حالت Raw استفاده شده بدون اینکه ذخیره شود یا فیلتر شود در چندین طریق، بعضی مواقع آن به توانا بودن برای کنترل سریع محتویات یک جدول مفید واقع می شود. Query Designer مدیریت Enterprise آن را برای دیدن و ویرایش سطرها آسان ساخته و حتی سطرهای جدید دیگر اضافه می کند.

دیدن سطرها در یک جدول

ساده ترین راه برای دیدن سطرها در یک جدول بازکردن Query Designer از طریق منوی Context جدول می باشد.

دیدن همه سطرها در یک جدول

۱- در Manager Enterprise ، پوشه Tables را از پایگاه داده Aromatherapy راهبری می کنیم. SQL Server یک سیستمی از جداول در قاب Details را نشان می دهد.

۲- جدول Plant Parts را کلیک راست کرده روی زیر منوی Open Table رفته و همه سطرهای برگردانده شده را انتخاب می کنیم. SQL Server ، Query Designer را با همه سطرهای نشان داده شده در جدول باز می کند.

محدود کردن تعدادی از سطرهای نشان داده شده

۱- Query Designer را اگر هنوز از تمرین قبل بازمانده است می بندیم.
۲- جدول Odors را در قاب Details کلیک راست کرده روی Open Table Submenu رفته و Return Top را انتخاب می کنیم. SQL Server یک کادر محاوره ای را نشان می دهد که از شما درخواست می کند که تعدادی از سطرها را که مشخص کرده اید بیاورید.

۳- ۵ را به عنوان تعداد ماکزیمم برای آوردن سطر تایپ می کنیم.

۴- OK را کلیک کرده Query Designer را باز می کنیم. ۵ سطر اول را در جدول نشان می دهد.

۵- پنجره Query Designer را می بندیم.

به هنگام سازی سطرها در یک جدول Query Designer همچنین می تواند برای تغییر مقادیر از سطرهای موجود در یک جدول یا اضافه کردن سطرها استفاده گردد.

ویرایش یک سطر در Query Designer

۱- جدول Plantparts را در قاب Details کلیک راست کرده روی Open Table Submenu رفته و سپس همه سطرهای بازگشتی را برای بازکردن Query Designer برای جدول Plantparts انتخاب می کنیم.
۲- ستون Plantparts از سطر را با یک Plantparts ID ۱۲ از برای خواندن Fruit Kernel به جای Fruit Kernal تغییر می دهیم. Query Designer یک آیکن Edit برای انتخابگر سطر اضافه می کند. برای نشان دادن اینکه رکورد ویرایش شده است اما تغییرات هنوز ذخیره نشده است.

۳- سطر دیگر را کلیک می کنیم. Query Designer آیکن Edit را ذخیره، تغییر و برمی دارد.

راهنمایی: قبل از اینکه شما به دیگر سطر حرکت کنید، شما می توانید کلید Esc را برای اعمال نکردن تغییراتان فشار دهید.

Query Designer

۱- در Query Designer در ستون Plantparts از آخرین سطر (که با آیکن جدید در انتخابگر سطر علامتگذاری شده) کلیک می کنیم.
۲- Fruit را تایپ می کنیم. Query Designer سطر را با آیکن Edit علامتگذاری کرده و یک سطر جدید در پایین جدول اضافه می کند.

۳- در هر سطر دیگر در جدول کلیک می کنیم. Query Designer یک PlantpartsID را برای سطر جدید تخصیص داده و آن را برای جدول ذخیره می سازد.

راهنمایی: شما نمی توانید یک PlantpartsID را به سطر جدید اضافه کنید زیرا آن به صورت یک ستون موجودیت تغییر داده است. اگر شما آن را انجام دهید Query Designer یک متن Error نشان خواهد داد.

شناخت Designer Query

زمانی که شما Query Designer را با استفاده از دستور Open Table باز می کنید. سطرها در جدول در یک شبکه شبیه به Data Sheet View یک جدول در Access نشان داده می شود که قاب Query Designers Results می باشد. Query Designer سه قاب کلی دارد. قاب Diagram Grid و قاب SQL. شما صفحه نمایشگر قابهای گوناگون را در Query Designer با استفاده از دکمه های Toolbar کنترل می کنید. شما می توانید قابها را در هر ترکیب نشان دهید. Query Designer از به هنگام سازی هر صفحه نمایش قاب مواظبت خواهد کرد زمانی که شما تغییراتی را برای Query می سازید.

قاب Diagram

دکمه قاب Diagram را در نوار ابزار Query Designer که قاب Diagram را نشان می دهد کلیک می کنیم. قاب Diagram یک نمایشی از Query برای یک نمودار پایگاه داده نشان می دهد. به طور کلی برای همه مزیتها از یک نمایش گرافیکی، قاب Diagram مفید می باشد. برای مشخص کردن جداول و نماهایی که روی هر کدام از Query هایتان پایه ریزی خواهد شد و ارتباطی بین آنها به خوبی ستونهایی که نشان داده شده است.

قاب Grid

دکمه قاب Grid را در نوار ابزار Query Designer که قاب Grid را نشان می دهد کلیک می کنیم. قاب Grid شبیه به Query Designer Grid در Access می باشد. آن یک راه سریع برای تغییر سر ستون از یک ستون Query را فراهم می آورد و دستوراتی را که در هر کدام از سطرها نشان داده خواهد شد را مشخص می کند. انتخاب ملاک مانند Surname = Jones قرار می دهد.

قاب SQL

دکمه قاب SQL را روی نوار ابزار Query Designer که قاب SQL را نشان می دهد کلیک می کنیم. قاب SQL عبارت Transact-SQL واقعی را که Query را تولید خواهد کرد نشان می دهد. شما می توانید مستقیماً عبارت Transact-SQL را با استفاده از قاب SQL وارد و ویرایش کنید. راهنمایی: زمانی که شما اولین Transact-SQL را می آموزید، قاب SQL یک ابزار بزرگ آموزش می باشد. Query را با استفاده از قاب Diagram و Grid راه اندازی کرده و Query Designer عبارت Transact-SQL را برای شما خواهد ساخت.

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

۱- 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 Rows Return All را انتخاب می کنیم. SQL Server، Query Designer را برای جدول باز می کند.

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

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

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

راهنمایی: شما می توانید بیشتر سطرها را در قاب Results به وسیله درج کردن دراپور 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

Column-List از شرط Select به

انجام می رسد.

انتخاب ستونها با استفاده از قاب SQL

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

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

انتخاب ستونها با استفاده از قاب Diagram

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

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

۳- 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 ۳۲۷۱۳-Colo۰PK-Varchar- 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

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

۲- 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 SQL Server 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.

۳- 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

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

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

۳- 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 ۲۱ "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
۱- قاب SQL را پنهان کرده و قاب Grid را به وسیله کلیک کردن دکمه روی نوار ابزار Query Designer نشان می دهیم.

۲- در هر سل ستون خالی در قاب Grid کلیک کرده و ' + Latin Name - ' + Oil 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.

۳- کلید Tab را فشار می دهیم. Server SQL ، Expr ۱ را به عنوان نام مستعار ستون پیشنهاد می دهد.

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

۵- 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
۱- قاب Grid را پنهان کرده و قاب SQL را به وسیله کلیک کردن دکمه ها روی نوار ابزار Query Designer نشان می دهیم.

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

راهنمایی: شما را قبل از GETDATE فراموش نکنید.

۳- 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

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

۲- دکمه Run را در نوار ابزار Query Designer برای به کارگیری مجدد Query کلیک می کنیم. SQL Server فقط ۵ سطر اول را نشان می دهد.

نشان دادن Top درصد از سطرها

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

۲- دکمه Run را در نوار ابزار Query Designer برای به کارگیری Query کلیک می کنیم. SQL Server فقط ۵ درصد اول از سطرها را در Server SQL نشان می دهد.

بخش ششم

شناخت عبارت INSERT

ساختار دستور عبارت Insert شبیه به عبارت Select می باشد، بیشترین شکل مینای 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 UNION ، GROUP BY ، Top 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.

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 عملیتهای 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 ۲۷ بحث خواهیم کرد.

ایجاد کردن جداول

جدول 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 قبیل German 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 ۱-۵ 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 ۹ انجام دهید.

مقادیر قابل قبول انواع داده

مقادیر عددی

مقادیر اعداد صحیح از $2^{63}-1$ تا 2^{63}

مقادیر اعداد صحیح از $2^{31}-1$ تا 2^{31}

مقادیر اعداد صحیح از $2^{15}-1$ تا 2^{15}

مقادیر اعداد صحیح از ۰ تا ۲۵۵

مقادیر اعداد صحیح با ارزش ۰ و ۱

مقادیر مقیاس با دقت ثابت شده از $10^{28}+1$ تا $10^{28}-1$

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

مقادیر Monetary (مالی) از $2^{63}-1$ تا 2^{63} (مقادیر Money تا $0,0001$ از هر واحد دقت دارد). Money

مقادیر Monetary از $3648 \cdot 748 \cdot 214$ تا $3647 \cdot 748 \cdot 214$ می باشد Smallmoney

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

مقادیر صحیح شناور (متغیر) از $1,79E+208$ تا $1,79E+208$ (مقادیر Float فقط تقریبی می باشد) Float

مقادیر صحیح شناور متغیر از $2,40E+28$ تا $2,40E+28$ 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 ۳ هزارم ثانیه یا $3,33$ میلی ثانیه دقت دارد) Smalldatetime

مقادیر تاریخ و زمان از ۱ ژانویه و ۱۹۰۰ تا ۶ ژوئن و 2079 می باشد (مقادیر Smalldatetime تا ۱ دقیقه دقت دارند)

مقادیر کاراکتری

مقادیر حرفی کدگذاری نشده با طول ثابت با طول حداکثر ۸۰۰۰ حرف می باشد. Char

Varchar

داده کدگذاری شده با طول متغیر با طول حداکثر ۱-۲۸۳۱ (۶۴۷ . ۷۴۱ . ۰۷۳ . ۱) حرف می باشد. Text
داده کدگذاری شده با طول ثابت با طول حداکثر ۴۰۰۰ حرف می باشد Nchar
داده کدگذاری نشده با طول متغیر با حداکثر طول ۴۰۰۰ حرف می باشد nvarchar
داده کدگذاری شده با طول متغیر با حداکثر طول ۱-۲۸۳۰ (۸۲۳ . ۷۴۱ . ۰۷۳ . ۱) حرف می باشد ntext
مقادیر باینری (۱ و ۰)

داده باینری با طول ثابت با حداکثر طول ۸۰۰۰ بایت می باشد binary
داده باینری با طول متغیر با حداکثر طول ۸۰۰۰ بایت می باشد Varbinary
داده باینری با طول متغیر با حداکثر طول ۱-۲۸۳۱ (۶۴۷ . ۷۸۳ . ۱۴۷ . ۲) بایت می باشد Image
مقادیر دیگر

یک مرجع مبناء برای یک Cursor می باشد (یک Cursor 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 SQL Server Timestamp نامیده می شود) rowversion
مقادیری از هر نوع غیر از text ، ntext ، timestamp (rowversion) و variant - sql می باشد. یک معرف واحد کلیدی GUID می باشد. Uniqidentifier

ایجاد کردن یک جدول جدید

جدول 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

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

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

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

۴- نام جدول را به Lesson ۵ تغییر می دهیم.

۵- 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

۱- 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.

۲- نوع داده را برای decimal تغییر می دهیم. SQL Server طول ستون را به ۹ تغییر داده و صحت، مقیاس و مشخصات فیلدها را تأیید می کند.

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

۴- Precision ستون را به ۵ و Scale را به ۲ تغییر می دهیم. SQL Server طول ستون را به ۵ برای نمایش دادن Precision جدید تغییر می دهد.

enterprise Manager در ۲۰۰۰ SQL Server 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 مقداری از یک extended Properties مقادیر variant - sql می باشد که می تواند شامل بیش از ۷۵۰۰ 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 ۲۸ رجوع کنید.

Scale و Precision (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 ۳۱۱ . ۳۶۴۷ يك دقت ۷ دارد (مجموع شماره رقمها) و يك مقیاس ۳ (رقمهای سمت راست از نقطه Decimal) را دارا می باشد. آن مهم می باشد که بدانیم دقت و مقیاس ارزش يك عدد بر روی طول يك 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.

افزافه کردن يك ستون Identity به جدول

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

۲- نوع داده ها به decimal تغییر می دهیم. SQL Server طول ستون را به ۹ تغییر داده و فیلدهای Identity و Scale و Precision را تأیید می کند.

۳- Allow Nulls را تیک می کنیم.

Nulls

ارزش Nulls يك نوع خاصی از يك ارزش در تکنولوژی رابطه ای می باشد که استفاده می شود برای نشان دادن اینکه يك ارزش ناپیدا یا غیر موجود می باشد استفاده کردن از Nulls قدری پیچیده و مسئله ساز و قطعاً مورد بحث می باشد.

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

۵- 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 ۱ را پیشنهاد می دهد.

ارزشهای Identity

زمانی که شما خصوصیات Identity يك ستون را تنظیم می کنید، شما به SQL Server می گوید که يك 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 يك سطر را در داخل يك جدول که دارای يك ستون Identity است قرار می دهد. آن به طور خودکار ارزشی برای ستون مبني 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 يك ستون Identity به عنوان يك Small int با Identity Seed، ۵۰ و یک Identity Increment، ۵ تعریف شود اولین سطر عدد ۵۰ و دومین سطر ۵۵ و سومین سطر ۶۰ و غیره تخصیص داده می شود فقط يك ستون در يك جدول می تواند تنظیمات Identity Property را داشته باشد.

GUID به جدول

- ۱- در يك سل خالي در ستون Column Name كليك كرده، My Guid را تايپ كرده و سپس Tab را فشار مي دهيم. SQL Server، نوع داده char را پيشنهاد مي دهد.
- ۲- نوع داده را به Uniqueidentifier را تغيير مي دهيم. SQL Server طول ستون را به ۱۶ تغيير داده و فيلد IsrowGuid را تاييد مي كند.
- ۳- در سل Description، Sample Guid تايپ مي كنيم.
- ۴- IsrowGuid را به Yes تغيير مي دهيم. SQL Server يك مقدار پيش فرض به Newid () مي دهد.

GUIDS

Guid كه از GloballyUniqueIdentifier گرفته شده، يك مقادير باينري (۱ و ۰) ۱۶ بابتي مي باشد كه هيچ كامپيوتر ديگري در دنيا نخواهد مقدار آن را توليد كند. نوع داده uniqueidentifier براي ذخيره كردن Guid استفاده مي شود. SQL Server به طور خودكار مقادير Guid را از همان راهي كه مقادير Identity ايجاد مي شود، فراهم نمي كند. زيرا يك جدول مي تواند شامل Guidس چندگانه باشد، اما فقط يك Identity منحصر به فرد باشد. اگر چه، تابع NEWID كه SQL Server آن را به صورت پيش فرض در نظر مي گيرد زماني كه خصوصيات IsrowGuid، Yes مي شود. يك Guid جديد بر خواهد گشت زماني كه سطر قرار داده مي شود.

افزافه كردن يك ستون Data در جدول

- ۱- در يك سل خالي در ستون Column Name كليك كرده، Mydata را تايپ كرده و سپس Tab را فشار مي دهيم. SQL Server، char را به عنوان نوع داده در نظر مي گيرد.
- ۲- نوع داده را به datetime تغيير مي دهيم. SQL Server طول ستون را به ۸ تغيير مي دهد.
- ۳- در سل Sample Date Column Description را تايپ مي كنيم.

افزافه كردن يك ستون Character به جدول

- ۱- در يك سل خالي در ستون Column Name كليك كرده، Mychar را تايپ و سپس Tab را فشار مي دهيم. SQL Server، char را به عنوان نوع داده در نظر مي گيرد.

انواع داده هاي Character

SQL Server دو نوع مختلف از ستونها Character را پشتيباني مي كند. طول ثابت شده و طول متغير كه هر کدام در دو "Flavors" مختلف Unicode و Non-Unicode و ۳ طول متفاوت مي باشند. Unicode يك روش از علامت كد گذاري مي باشد كه انواع باينهاي دابل را پشتيباني مي كند. اگر يك ستون به طور طول متغير نمايان شود (براي مثال، varchar يا text براي داده هاي Non-Unicode و nvarchar و ntext براي داده هاي Unicode) و SQL Server فقط خصوصيت داده هاي معين وارد شده را ذخيره خواهد ساخت. از طرف ديگر اگر ستون به صورت طول ثابت شده نمايان شود (char براي Non-Unicode داده nchar براي داده هاي Unicode) و SQL Server مقادير وارد شده را با فضاهای Pad خواهد كرد.

- ۲- طول ستون را به ۲۵ تغيير مي دهيم.
- ۳- در سل Description، Character Column Sample را تايپ مي كنيم.
- ۴- Unknown را در سل Default Value تايپ مي كنيم (اطمينان حاصل كنيد كه اطراف كلمه ویرگول داشته باشد).

مقادير پيش فرض

يك Default Value يك عددي مي باشد كه در داخل يك ستون قرار خواهد گرفت اگر كاربر به طور آشكار

SQL default Values به وسیله

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 ۱۲۳ توابعی از قبیل NEWID یا GETDATE یا عبارات ریاضی مانند ۳+۵.

ذخیره کردن و بستن جداول

- ۱- دکمه Save را در نوار ابزار Table Designer کلیک می کنیم. Server SQL تعریف جدول را ذخیره می سازد.
- ۲- 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

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

۲- جدول Lesson ۵ را در قاب Details کلیک راست کرده و Design Table را انتخاب می کنیم. SQL Server ، Table Designer را باز می کند.

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

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

برداشتن یک ستون

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

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

۳- کلید Save را برای ذخیره کردن تغییرات کلیک می کنیم.

۴- 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

۱- 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.

۲- جدول Lesson ۵ را در قاب Details کلیک راست کرده و Rename را انتخاب می کنیم.

۳- Lesson ۵ New 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

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.

۵- Close را برای مرخص کردن کادر محاوره ای کلیک می کنیم.
۶- 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

۱- New Lesson را در قاب Details انتخاب می کنیم.

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

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

۳- 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.

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^۱ 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 سطر 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.

ایجاد کردن 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
۱- پوشه Defaults را از پایگاه داده Aromatherapy از درخت Console جستجو می کنیم. SQL Server لیستی از Default ها را در قالب Details نشان می دهد. (چیزی در پایگاه داده Sample وجود ندارد).

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

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

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

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

رابط دادن یک پیش فرض به یک ستون
۱- 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.

۲- 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.

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

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

قطع پیوند یک پیش فرض

۱- 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.

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

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

۴- دکمه Save را کلیک 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 Server SQL تعریف شده به کار برده شود. برخلاف Check Constraint، یک نقش نمی تواند مبنایی برای یک ستون به طور مستقیم باشد. در عوض، مفادیری 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 ۲۴ 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

۱- 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)

۲- دکمه New را کلیک کرده SQL Server کادر محاوره ای را باز می کند.

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

۴- LEN(@FIdalue) < ۳ را به عنوان متن نقش تایپ می کنیم.

راهنمایی: به خاطر داشته باشید LEN یک تابع SQL Transact می باشد که تعدادی از کاراکترها را در یک متن رشته ای برمی گرداند و اینکه @ قبل از یک برچسب عبارت SQL Transact یک متغیر را نشان می دهد، یک مقدار که برای عبارت داده می شود. بنابراین در این حالت نقش True می گردد اگر طول ستون بزرگتر از ۳ باشد.

۵- 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

۱- کادر محاوره ای Properties Rule را برای Sample 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.

۲- ستون Bind را کلیک کرده SQL Server کادر محاوره ای Bind Rule to Columns را نشان می دهد.

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

۴- 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.

۵- SQL Server کادر محاوره ای Columns Bind Rule to را می بندد.

۶- OK را مجدداً برای بستن کادر محاوره ای Rule Properties کلیک می کنیم.

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 Types ها فراهم می آورد. Data User-Defined 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 Type به کار برده شوند. زمانی که یک ستون بر مبنای یک User-Defined Data Type ایجاد می گردد، ستون جدول همه خصوصیات مشخص شده را برای آن نمونه به ارث خواهد برد. زمانی که مشخصات از User-Defined Data Types تغییر می 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.

ایجاد کردن 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 Type

۱- پوشه User-Defined Data Types را از پایگاه داده Aromatherapy راهبری می کنیم. Server SQL یک لیستی از User-Defined Data Types را نشان می دهد (چیزی در پایگاه داده Sample وجود ندارد).

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

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

۴- نوع داده پایگاه را varchar و طول آن را ۲۰ قرار می دهیم.

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

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

تخصیص یک ستون برای یک User-Defined Data Type

۱- 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.

۲- ستون Sample را انتخاب کرده و MySample را از Data Type Combo Box انتخاب می کنیم. Data Type , SQL Server را برای MySample قرار می دهد. راهنمایی: User-Defined Data Types در پایین لیست Data Type می باشد.

۳- دکمه Save را کلیک کرده SQL Server جدول را با تعریف جدید ذخیره می سازد.

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

- ۱- دکمه علامت سؤال جدید را روی Toolbar تجزیه کننده سؤال کلیک کنید تا یک پنجره سؤال جدید ظاهر شود.
- ۲- روی دکمه Script Loud روی Toolbar تجزیه کننده سؤال کلیک کنید. تجزیه کننده سؤال جعبه پرونده گفتگو را نشان خواهد داد.
- ۳- نسخه خطی Create Local را انتخاب کنید و Open را کلیک کنید. تجزیه کننده سؤال نسخه خطی را پر می کند.
- ۴- دکمه Execute Query را در Toolbar تجزیه کننده سؤال کلیک کنید. تجزیه کننده سؤال جدول موقت ایجاد می کند.
- ۵- User Table Folder را که در پایگاه داده Tempdb در Object browser می باشد را انتخاب کنید.
- ۶- F۵ 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

- ۱- 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.
- ۲- بر روی دکمه Loud Script روی Toolbar Query Analyzer کلیک کنید. Query Analyzer نشانگر جعبه گفتگوی فایل Query خواهد بود.
- ۳- خطی را که در آن Create Loud وجود دارد را انتخاب کنید و Open را کلیک کنید. تجزیه کننده سؤال خط انتخابی را Loud می کند.
- ۴- دکمه Execute Query را که روی Query Analyzer Toolbar می باشد را کلیک کنید. Query Analyzer جدول موقت را ایجاد می کند.
- ۵- User Folder Table که مربوط به Tempdb Database در Object browser را انتخاب کنید.
- ۶- F۵ را برای راه اندازی مجدد نمایشگر Object browser فشار دهید. Query Analyzer نمایشگر Loud Table #.dbo لیست خواهد بود.

بخش دهم

ارتباط دادن جداول

Query هایی که در فصل ۳ 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. SQL Server 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 = 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 ۲۵۶ جدولی برای هر Query وجود دارد، اما آن به حد زیاد غیر محتمل است که شما اصلاً نیاز به ۵ یا ۶ نیاز ندارید و ۲ یا ۳ بیشتر معمول می باشد. در حقیقت اگر شما نیاز به اتصال بیش از ۱۰ 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 پروندها جداسازی می گردند. بنابراین نام واجد شرایط از جدول Oils در سیستم من Bunny.Aromatherapy.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.

TRUE را باز می گرداند باز خواهد گرداند.

پیوند دو جدول با استفاده از قاب دیاگرام

۱- 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.

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

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

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

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

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

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

۸- 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.

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

۱۰- 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

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

۲- 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
```

۳- 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

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

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

۳- جدول Planttypes را در لیست جداول انتخاب می کنیم. Add را کلیک کرده SQL Server جدول را به Query اضافه می کند.

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

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

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

ارتباط جداول چندگانه با استفاده از قاب SQL

۱- قاب دیاگرام را پنهان کرده و قاب SQL را نشان می دهیم.

۲- 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
```

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

۴- 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

۱- Query Designer را برای جدول Oils به وسیله کلیک راست کردن نام جدول در قاب Details باز کرده،

Open رفته و همه سطرهای بازگشتی را انتخاب می کنیم.
۲- قاب دیاگرام را نشان می دهیم.

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

۴- 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.
۵- 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.

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

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

۸- 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. میکروسافت این را به عنوان یک ویژگی در نظر می گیرد.

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

۱۰- 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.

۱۱- 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.

۱۲- 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.
۱۳- Close را برای بستن کادر محاوره ای کلیک می کنیم. Query Designer خط ارتباطی را برای انعکاس مشخصات ارتباطی جدید تغییر می دهد.

۱۴- دکمه Run را 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
۱- قاب دیاگرام را پنهان کرده و قاب SQL را در Query Designer نشان می دهیم.

۲- 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

```

۳- 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.

۴- 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 حالتها UNION SELECT را همان طوری که شما برای یک Query دوست دارید اضافه کنید. (این موضوع برای ۲۵۶ 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 Union All را مشخص کنید، ولی سطرهای چندگانه حفظ خواهد شد.

ایجاد کردن یک UNION

۱- 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.
 ۲- قاب SQL را نشان می دهیم.

۳- 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

```

۴- 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.

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 ۷ ما درباره یکپارچگی رابطه ای مطالعه کردیم که تضمین می کند اجتماع بین جداول ایجاد و به طور صحیح نگهداری می گردند.

Check Constraints برای اجرای دو فرم اضافی از یکپارچگی پایگاه داده استفاده می شود. domain Integrity و entity Integrity. در اصطلاحات رابطه ای که Domain محدودی 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 ۳۲/۷۶۸ تا ۳۲/۷۶۸ 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 ۱۹۰۰ و سال جاری می باشد. شما از یک Check Constraints استفاده می کنید، برای تخمین اینکه هیچ کس به طور واقعی مقادیر ۱۵۴۳ یا ۲۰۷۵ 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 Boolean Expressions برای مقادیر True یا False ارزیابی می گردد. Boolean Expressions را در درس ۱۳ می آموزیم. در این درس ما از عبارت LEN()=ε استفاده می کنیم. LEN یک تابع Transact-SQL می باشد که تعدادی کارکترهای یک رشته را برمی گرداند. بنابراین عبارت LEN()=ε ارزش آن اگر شامل ε یا بیشتر کارکتر باشد ارزش آن True و اگر کمتر از ε 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.

۱- 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.

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

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

۴- Len(Oil Name)=ε را 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 بگویید به SQL Server که از ایجاد کردن داده به وسیله چک نکردن Existing Data On Creation Check از محدودیت چشم پوشی کند.

۵- Close را کلیک کرده SQL Server کادر محاوره ای Properties Table Designers را می بندد.

۶- دکمه Save را کلیک کرده SQL Server کنترل می کند که همه سطرها در جدول Check Constraint مواجه می شوند و سپس محدودیت را ذخیره می کنیم.

مدیریت Check Constraints

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
۱- 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 و Design Table را انتخاب می کنیم. SQL Server Table , Designer را باز می کند.

۲- دکمه Constraints را کلیک کرده SQL Server کادر محاوره ای Table Designer را با صفحه خصوصیات Check Constraint باز می کند.

۳- اطمینان حاصل کنید که Oils - Ck در Constraint Combo Box انتخاب شده نشان داده شده است.
۴- متن محدودیت را برای (Oil Name) LEN < ۲ به عنوان عبارت محدودیت جدید تغییر می دهیم.

۵- Close را کلیک کرده SQL Server کادر محاوره ای Table Designer's Properties را می بندد.
۶- 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 از Design Table نگهداری می گردد.

تغییر نام یک Check Constraints
۱- 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.

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

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

۴- Close را کلیک کرده SQL Server کادر محاوره ای Properties را می بندد.
۵- دکمه Save را کلیک کرده SQL Server همه سطرها در جدول که با Check Constraint مواجه شده اند را کنترل می کند و سپس Constraint ذخیره می گردد.

حذف یک Check Constraint
۱- Designer Table 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.

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

۳- 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.

۴- Close را کلیک کرده SQL Server کادر محاوره ای Properties را می بندد.

Save را کلیک می کنیم. SQL Server محدودیت را برمی دارد.
۶- Table Designer را می بندیم.