

المحاضرة الثانية

التعامل مع أغراض قاعدة البيانات DDL

بناء قاعدة معطيات – الجداول – الجداول المؤقتة

لغة تعريف أغراض البيانات **DDL Data Definition Language** هي جزء من لغة **SQL** تتضمن التعليمات الخاصة بإنشاء وتعديل وحذف أغراض البيانات مثل:

CREATE: وهي مخصصة لإنشاء أغراض جديدة في قاعدة البيانات.

DROP: وهي مخصصة لحذف غرض من قاعدة البيانات.

ALTER: وهي مخصصة لتعديل غرض في قاعدة البيانات.

الهدف من الجلسة

1. التعامل مع قواعد البيانات **databases** انشاء/تعديل/حذف.
2. التعامل مع الجداول **Tables**.
3. التعامل مع الجداول المؤقتة **Temporary Tables**.

التعامل مع قواعد البيانات

إن قاعدة البيانات هي بنية منطقية **Logical Structure** تتم إدارتها من قبل برنامج متخصص نطلق عليه اسم محرك قواعد البيانات **Database Engine** بحيث يقوم هذا المحرك بإنشاء عدة ملفات فيزيائية موافقة لكل قاعدة ليقوم هذا المحرك بعدها بإدارة عمليات التخزين والاستعلام والحذف والتعديل للبيانات المخزنة في هذه الملفات بشكل كامل.

تحتوي كل قاعدة بيانات على العديد من الأغراض **Database Objects** ومن أهم هذه الأغراض:

1. الجداول
2. القيود
3. الفهارس
4. المناظير
5. الاجرائيات المخزنة
6. التوابيع المعرفة من قبل المستخدم
7. القادحات
8. المستخدمون

يتم التواصل مع مختلف محركات قواعد البيانات العلائقية عبر لغة برمجية موحدة هي لغة الاستعلامات المهيكلية **SQL** وهي اللغة التي تسمح لنا بكتابة التعليمات اللازمة للتعامل مع قواعد البيانات ومختلف أغراضها.

البنية الفيزيائية لقاعدة البيانات

سنقتصر على شرح لمحة موجزة للبنية الفيزيائية لقواعد البيانات المدارة من قبل المحرك Microsoft SQL Server (سنطلق عليه من الآن وصاعداً اسم "محرك مايكروسوفت" أو محرك البيانات اختصاراً) حيث تختلف البنية الفيزيائية الموافقة لكل قاعدة ما بين محرك وآخر.

تتكون كل قاعدة بيانات في محرك مايكروسوفت من الملفات التالية:

1. الملف الأساسي Primary File

يجب أن تحتوي أي قاعدة بيانات على ملف أساسي **Primary** واحد، وواحد فقط (ملف لاحقته المقترحة من مايكروسوفت **mdf**) ويحتوي هذا الملف البيانات الأساسية لقاعدة البيانات أو ما نطلق عليه اسم البيانات السامية لقاعدة البيانات Metadata التي تشتمل على تعريف مختلف الأغراض المخزنة في قاعدة البيانات. كما يمكن للملف الأساسي أن يحتوي أي عدد من أغراض قاعدة البيانات. إن احتواء هذا الملف على البيانات السامية للقاعدة يجعل من سلامة هذا الملف أمراً جوهرياً لضمان عمل القاعدة الموافقة وإن

تعرض هذا الملف للعطب لأي سبب من الأسباب يجعل القاعدة الموافقة معطوبة ويكون من الصعب، ومن المستحيل أحيانا، إصلاحها. سمحت اعتبارا من الإصدار 2012 من محرك مايكروسوفت إمكانية إضافة عدة ملفات **mdf** لنفس القاعدة حيث يقوم المحرك عندها بتخزين البيانات السامية في أول ملف منها.

2. الملفات الثانوية Secondary Files

يمكن لأي قاعدة بيانات أن تحوي ملف ثانوي واحد أو أكثر وهي ملفات مخصصة لتخزين بيانات أغراض قاعدة البيانات (بيانات الجداول والفهارس على سبيل المثال) ولكنها لا تحوي على بيانات سامية. لاحقة هذه الملفات **ndf**

3. ملفات المناقلات Transaction Log Files

وهي ملفات لتخزين تسلسل العمليات المنفذة على القاعدة ونسخ البيانات قبل وبعد أي عملية. تتميز هذه الملفات بصغر حجمها عموما الأمر الذي يسرع من عمليات القراءة والكتابة فيها مقارنة بالملفات الأساسية والثانوية التي قد تكون ذات حجوم كبيرة تصل حتى مرتبة التيرابايت. يقوم المحرك بتخزين أي عملية نقوم بها على ملف المناقلات (تخزين مؤقت) ليقوم بنقله تلقائيا إلى ملفات البيانات (تخزين دائم) بشكل دوري (دون أي تدخل من المستخدم). تساعد هذه الملفات بتسريع عمليات التعامل مع قواعد البيانات من جهة ومن جهة أخرى تسمح باسترجاع جميع البيانات المفقودة الناتجة عن توقف المحرك بشكل مفاجئ نتيجة لانقطاع التيار الكهربائي أو أي عطل آخر للمخدم. تحوي أي قاعدة بيانات على ملف مناقلات واحد على الأقل وفي حال وجود عدة ملفات يتم التخزين فيها تسلسليا أي بعد ملئ الملف الأول يخزن المحرك في الملف الثاني حتى يمتلئ وهكذا حتى يمتلئ آخر ملف ليقوم المحرك بعدها بالتخزين بدءا من بداية الملف الأول (مع حذف أي معلومات سابقة مخزنة فيه) لاحقة هذا النوع من الملفات **ldf**.

ملاحظة: إن لاحقة الملف **mdf**, **ndf**, **ldf** تساعد على تمييز نوع الملف لكن في الحقيقة هي لاحقات مقترحة من قبل مايكروسوفت ويمكننا تعريف ملفات لقاعدة البيانات بحيث تكون من أي لاحقة **XYZ**. مثلا ونحن نحدد نوع الملف المضاف (ملف بيانات أو ملف مناقلات).

يتم إنشاء قاعدة بيانات بالتعليمة Create Database وحذفها بالتعليمة Drop Database وتعديلها بالتعليمة Alter Database. وسنعرض فيما يلي أمثلة عن الإنشاء بأشكاله المختلفة إضافة إلى مثال عن الحذف والتعديل.

إنشاء قاعدة بيانات

1. إنشاء قاعدة بيانات بالإعدادات الافتراضية

أبسط طريقة لإنشاء قاعدة بيانات هي باستخدام الإعدادات الافتراضية للمخدم وفيها لا نحدد أي ملفات ليقوم عندها المخدم تلقائياً بإنشاء الملفات الأساسية الكافية لعمل هذه القاعدة.

لنفترض أننا نريد إنشاء قاعدة اسمها **HelloDB** بالإعدادات الافتراضية. التعليمة الموافقة:

Create Database HelloDB

يقوم المخدم تلقائياً بإنشاء ملفين موافقين للقاعدة الجديدة بحيث يشتق اسم هذين الملفين من اسم القاعدة وهذين الملفين هما:

(a) الملف الرئيسي HelloDB.mdf

(b) ملف المناقلات HelloDB_Log.ldf

يتم إنشاء الملفات السابقة ضمن المجلد MSSQL/DATA الواقع ضمن مسار تثبيت محرك قواعد البيانات.

2. إنشاء قاعدة بيانات ضمن مجلد خاص مع تسمية الملفات المكونة لها

لنفترض أننا نريد أن ننشئ قاعدة بيانات منطقية اسمها Sales بحيث تكون ملفات مخزنة على المسار الفيزيائي D:\. نريد أن تحوي هذه القاعدة على ملف أساسي وحيد اسمه MySales وملف مناقلات وحيد اسمه MySalesLog.

CREATE DATABASE Sales
ON

(**NAME** = Sales_dat ,
 FILENAME = 'D:\MySales.mdf' ,
 SIZE = 10 ,
 MAXSIZE = UNLIMITED,
 FILEGROWTH = 5)

LOG
ON

(**NAME** = 'Sales_log',
 FILENAME = 'D:\MySalesLog.ldf',
 SIZE = 5MB,
 MAXSIZE = 25MB,
 FILEGROWTH = 10%)

نلاحظ أن لكل ملف العديد من المواصفات أهمها:

1. الاسم المنطقي Logical Name الذي تحدده الوصفة NAME وهو اسم للاستخدام المنطقي للملف ضمن المحرك بصرف النظر عن اسمه الفيزيائي ومكان تخزينه
2. الاسم الفيزيائي للملف File Name وهو ما تحدده الوصفة FileName وتحتوي المسار الكامل للملف.
3. الحجم البدائي للملف Size ويمثل الحجم الأولي المحجوز للملف مقدرة افتراضيا بالميجابايت.
4. الحجم الأعظمي Max Size وهو ما تحدده الوصفة MaxSize مقدرة بالميجابايت، وإذا أردنا أن يكون حجم هذا الملف غير محدودا نستخدم القيمة الخاصة Unlimited لنسمح للملف بالنمو تلقائيا بما تحدده الوصفة FileGrowth .
5. معدل النمو File Growth وهو ما تحدده الوصفة FileGrowth والتي يمكن أن تكون قيمتها إما مطلقة أو نسبية. تتم زيادة حجم الملف بعد امتلاء الحجم الحالي كاملا بالبيانات ويتم حساب الحجم الجديد وفق إحدى القاعدتين:

- i. إذا كان النمو مطلقا: الحجم الجديد = الحجم الحالي + معدل النمو
- ii. إذا كان النمو نسبيا: الحجم الجديد = الحجم الحالي * (1 + معدل النمو)

تنمو جميع ملفات البيانات نموا متجانسا بحيث يتم التخزين بشكل يتناسب مع حجوم كل من هذه الملفات.

3. إنشاء قاعدة بيانات بعدة ملفات بيانات ومناقلات

نريد إنشاء القاعدة Archive بحيث تكون جميع ملفاتها ضمن المسار D:\ ولتحتوي 3 ملفات بيانات (رئيسية وثانوية) و 2 ملف مناقلات لتكون بالواصفات التالية:
ملفات البيانات

الاسم المنطقي	الاسم الفيزيائي	الحجم الأولي	الحجم الأعظمي	معدل النمو
Arch1	ArcData1.mdf	MB 10	غير محدود	MB 5
Arch2	ArcData2.ndf	MB 5	MB 15	MB 3
Arch3	ArcData3.ndf	MB 3	MB 12	%20

ملفات المناقلات

الاسم المنطقي	الاسم الفيزيائي	الحجم الأولي	الحجم الأعظمي	معدل النمو
ArcLog1	ArcLog1.ldf	MB 5	MB 10	MB 2
ArchLog2	ArcLog2.ldf	MB 5	MB 20	MB 5

تعليمة الإنشاء الموافقة:

```
CREATE DATABASE Archive
ON
( NAME = Arch1,
  FILENAME = 'D:\archdat1.mdf',
  SIZE = 10MB,
  MAXSIZE = UNLIMITED,
  FILEGROWTH = 5
),
( NAME = Arch2,
  FILENAME = 'D:\archdat2.ndf',
  SIZE = 5MB,
  MAXSIZE = 15,
  FILEGROWTH = 3
```

```

),
( NAME = Arch3,
  FILENAME = 'D:\archdat3.ndf',
  SIZE = 3MB,
  MAXSIZE = 12,
  FILEGROWTH = 20%
)
LOG
ON
( NAME = Archlog1,
  FILENAME = 'D:\archlog1.ldf',
  SIZE = 5MB,
  MAXSIZE = 10,
  FILEGROWTH = 2),
( NAME = Archlog2,
  FILENAME = 'D:\archlog2.ldf',
  SIZE = 5MB,
  MAXSIZE = 20,
  FILEGROWTH = 5
)

```

تعديل قاعدة بيانات

لتعديل أي قاعدة بيانات نستخدم التعليمة **Alter Database**.

مثال 1: لتعديل قاعدة البيانات السابقة لتعديل الاسم نستخدم الأمر التالي:

Alter Database HelloDB

Modify Name = TestDB

مثال 2: لتعديل قاعدة البيانات السابقة لتعديل Collation نستخدم الأمر التالي:

Alter Database TestDB

COLLATE French_CI_AI

حذف قاعدة بيانات

لحذف أي قاعدة بيانات نستخدم التعليمة **Drop Database**.

مثال: لحذف قاعدة البيانات السابقة نستخدم الأمر التالي:

Drop Database TestDB

استخدام توابع النظام في SQL Server للتعامل مع قواعد البيانات:

1- تغيير اسم قاعدة البيانات:

نستخدم تابع النظام **SP_renamedb** مثلا لتغيير اسم القاعدة HelloDb إلى NewHelloDb ننفذ الأمر التالي:

sp_renamedb 'HelloDb' , 'NewHelloDb'

2- تعليمات الاستعلام عن ملفات قواعد البيانات:

لاصدار قائمة بأسماء القواعد المدارة على المحرك الحالي وملفات كل منها نستخدم تابع النظام **sp_helpdb** الذي يعيد قائمة بجميع القواعد المخدومة من المحرك الحالي مع حجم كل منها وللحصول على معلومات خاصة بقاعدة محددة ولتكن Pubs نستخدم نفس التابع السابق ونمرر له اسم القاعدة المطلوبة كما يلي:

sp_helpdb Pubs

لاصدار قائمة ملفات قاعدة البيانات الحالية نستخدم تابع النظام **sp_helpfile**

الذي يعرض بيانات ملفات القاعدة الحالية

use Northwind

Go

sp_helpfile

أو

sp_helpfile Northwind_Log

يقوم هذا التابع بدون أي متحولات بعرض بيانات ملفات القاعدة الحالية في حالتنا هنا Northwind وإذا أردنا عرض بيانات ملف محدد من ملفات هذه القاعدة فنمرر الاسم المنطقي لهذا الملف كمتحول لتابع النظام.

لمعرفة المسار الافتراضي للملفات:

SELECT

SERVERPROPERTY('instancedefaultdatapath') AS

[DefaultFile],

SERVERPROPERTY('instancedefaultlogpath') AS

[DefaultLog]

الجدول Tables:

انشاء الجدول:

تستخدم تعليمة create Table لانشاء الجدول.

مثال:

deptNo	deptName	depStartDate

CREATE Database Test1

Use Test1

Go

```
CREATE TABLE Department  
(  
    deptNo int identity(1,1) ,  
    deptName varchar (50) ,  
    [depStartDate] datetime  
)
```

لاحظ:

لبناء ترقيم تلقائي على أحد الحقول **IDENTITY (1 , 1)**

مثال:

```
CREATE TABLE employee  
(  
    empNo INT PRIMARY KEY,  
    FirstNAME VARCHAR(10),  
    LastName VARCHAR(10),  
    empMgr INT NULL,  
    hiredate DATETIME,  
    sal NUMERIC(7,2),  
    tax INT,  
    dept INT  
)
```

تعديل الجدول:

تستخدم تعليمة **alter Table** لتعديل الجدول كإضافة/تعديل/حذف حقل أو قيد ...

بعض الأمثلة:

إضافة حقل جديد:

```
ALTER TABLE Department  
ADD MinSalary int , employeeNb int
```

تعديل حقل:

```
ALTER TABLE Department  
ALTER COLUMN deptName varchar (100) NOT NULL
```

إذا كان الجدول فارغ يمكن تعديل نمط البيانات لأي حقل إلى أي نمط، أما إن كان هناك بيانات فلا يمكن التعديل إلا إلى نمط بيانات يشمل البيانات الموجودة

حذف حقل:

```
ALTER TABLE Department  
DROP COLUMN MinSalary , employeeNb
```

تعديل اسم الحقل sal من جدول employee:

```
EXEC sp_rename ' employee.sal', 'salary', 'COLUMN'
```

حذف الجدول:

لحذف الجدول السابق

```
drop table [Department]
```

الجدول المؤقتة Temporary Table

الجدول المؤقتة هي جداول يتم إنشاؤها لغرض تخديم عمليات الاستعلام و بالأخص تلك التي قد تتضمن استخدام نتيجة استعلام ضمن استعلام. وبالتالي تشغل مساحة تخزين.

يبدأ اسم الجدول المؤقت المحلي local Temp Table بالرمز # (الجدول مرئي من قبل المستخدم الذي أنشأه في الجلسة الحالية فقط ويتم حذفه تلقائياً عند الخروج من الجلسة)

يبدأ اسم الجدول المؤقت العام global Temp Table بالرمز ### (الجدول مرئي من قبل جميع المستخدمين في جميع الجلسات بعد انشاؤه ويتم حذفه عند خروج جميع المستخدمين الذين يأشرون عليه من جميع الجلسات).

مثلا نريد إنشاء جدول مؤقت #FullName ليحوي حقلي اسم وكنية كل موظف

Use northwind

Go

CREATE TABLE #FullName

```
(  
    FirstName    varchar(50)  
    , LastName   varchar(50)  
)
```

نلاحظ أنه لا يوجد أي اختلاف بصيغة التعريف إلا ببداية اسم الجدول #.

لنفرض أننا نريد أن نخزن في الجدول السابق اسم وكنية جيع الموظفين الذين تم توظيفهم قبل العام 1994. نستخدم لذلك الاستعلام التالي:

INSERT INTO #FULLNAME

SELECT

FIRSTNAME

,LASTNAME

FROM

EMPLOYEES

WHERE

YEAR(HIREDATE) < 1994

نلاحظ أن التعليمات السابقة قد أضافت مجموعة سجلات إلى الجدول المؤقت ولذلك نسمي هذه العملية بعملية إضافة كمية (كمية من السجلات) `bulk insert`.
يمكننا الاستعلام في الجدول المؤقت بشكل مماثل للجدول الدائم.

```
SELECT * FROM #FULLNAME
```

نلاحظ أنه في الطريقة السابقة قمنا بتعريف بنية الجدول `#FullName` (اسم كل عمود ونمط بياناته) قبل أن نقوم بإضافة بيانات إلى هذا الجدول.

يمكننا أن نعرف بنية جدول مؤقت لتتوافق مع بنية الأعمدة الناتجة عن استعلام ما بالتعليمة التالية:

```
SELECT
```

```
FIRSTNAME,
```

```
LASTNAME
```

```
INTO
```

```
#ANOTHER_FULLNAME
```

```
FROM EMPLOYEES
```

نلاحظ أننا لم نضطر هنا لتعريف بنية الجدول `#ANOTHER_FULLNAME` حيث قام محرك مايكروسوفت تلقائياً بتحديد بنيته لتتطابق مع بنية الأعمدة الناتجة عن الاستعلام المنفذ.

انتهت المحاضرة