**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**مفاهیم مرتبط با طراحی Database در SQL سرور**

یک پایگاه داده یا Database به اندازه ای کاربردی و مفید است که طراحی آن مفید و کاربردی است.

طراحی ها همیشه متنوع هستند و هیچوقت یک طراحی جامع و کامل وجود ندارد که بتوان از آن در تمامی Database ها استفاده کرد. ممکن است برخی از طراحی ها نسبت به سایر طراحی ها از ساختار بهتری برخوردار باشند اما شما نمی توانید این را ملاک قرار دهید. در طراحی Database ها دو عامل اصلی نقش دارند ، ابتدا دانش طراحی Database است و سپس خلاقیت و تجربه طراح است.

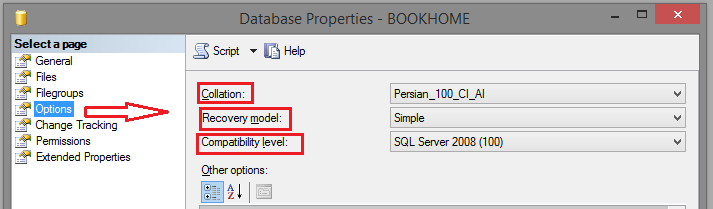
هیچوقت شما در طراحی Database ها چیزی به نام الزام مشاهده نمی کنید (به جز مواردی که به ندرت پیش می آید) و این شما هستید که با استفاده از دانشی که با استفاده از مطالعه و تجربه بدست آورده اید می توانید یک طراحی زیبا را ارائه کنید.

یکی از مواردی که در طراحی Database ها بسیار مهم است این است که شما بتواند در بدو طراحی Database به درستی اینکار را انجام دهید ، اگر طراحی اولیه درست انجام نشود ، بازگشتن و طراحی مجدد آن کار ساده ای نیست و مشکلات بسیاری را در بر خواهد داشت ،قبل از طراحی یک database برای یک نرم افزار بهتر است با برنامه نویس و طراحی نرم افزار به درستی ارتباطات لازم برقرار شود و اطلاعات لازم دریافت شود و از طرفی نیز انجام اینکار نیاز بسیاری به انجام پیشبینی ها و آینده نگری می باشد ، آینده نگری و برنامه ریزی درست برای Database ها در واقع هسته اصلی طراحی یک Database می باشد.

باید به این موضوع دقت کنید که طراحی یک چیز ثابت و از پیش تعریف شده نیست و یک طراح Database موفق در طراحی های خود قطعا فاکتوری به نام تجربه را دارد که بسیار به وی کمک خواهد کرد ، پاسخ داده به سئوالاتی از قبیل :

چرا باید اینکار انجام شود ؟ چرا باید از انجام اینکار پرهیز شود ؟ چرا باید در خصوص فلان مورد تحقیق شود ؟ چرا باید پیشبینی چنین مواردی انجام شود ؟و امثال اینگونه سئوالات صرفا زمانی امکانپذیر است که شما تجربه لازم در خصوص طراحی Database ها را داشته باشید.

**بررسی Option**



* Collation را براي زبان فارسي بهتر است Persian\_100\_CI\_AI قرار دهيد.
* ورژن های Compatibility Level

|  |  |
| --- | --- |
| **Value** | **Description** |
| 80 | SQL Server 2000 |
| 90 | SQL Server 2005 |
| 100 | SQL Server 2008 and SQL Server 2008 R2 |
| 110 | SQL Server 2012 |
| 120 | SQL Server 2014 |

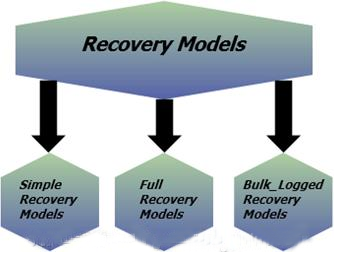
**منظور از Transaction Log در SQL سرور چیست ؟**

ترجمه لغوی این مفهوم به شکل Transaction یا تبادلات و Log یا صورت عملیات می باشد.

در واقع اگر بخواهیم بصورت کلی این واژه را تعریف کنیم Transaction Log ها رکوردهایی هستند که در آنها فعالیت ها یا Action هایی که در پایگاه های داده انجام می شود ثبت می شوند. در این Log ها تمامی فعالیت ها و عملیات هایی که بر روی پایگاه های داده شما انجام می شود یا حتی توسط خود پایگاه داده انجام می شود ثبت شده و نتایج در قالب یک فایل مجزا از پایگاه داده اصلی ذخیره می شود.

برای مثال اگر شما پایگاه داده ای به نام Testایجاد کنید در محل ایجاد این پایگاه داده دو فایل به نام های Test.mdf و Test.ldf ایجاد می شوند که mdf اطلاعات اصلی و ldf مربوط به Transaction Log های این Database می باشد.

**Backup گیری از SQL Server: معرفی انواع Recovery Model در SQL سرور**



در SQL Server ما سه نوع Recovery Model مختلف داریم که روش مدیریت Log File ها و آماده سازی آنها برای بروز حوادث احتمالی را مشخص می کنند. بنابراین وجود یک Recovery Model مناسب ( و البته استفاده درست از آن ) باعث آسودگی خاطر شما در زمان بروز مشکل برای Database ها خواهد بود.

**Full Recovery Model**

این مدل که پیشفرض هم می باشد, کاملترین نوع  هست که کمترین گمشدگی داده رو داریم .

در اینجا ما PIT (Point In Time Restore) رو داریم   یعنی میتونیم به هر لحظه ای از زمان دیتابیس رو برگردونیم .

**Simple**

در حالت Simple Recovery Model شما قابلیت بازیابی اطلاعات به یک نقطه زمانی خاص را ندارید و فقط می توانید اطلاعات را بصورت کامل با استفاده از Full یا Differential Backup به زمانی برگردانید که این Backup ها گرفته شده اند.

مثلا فرض کنید که شما همیشه آخر شب Back up می گیرید و یک Failure در ساعت 10 صبح برای شما رخ میده. در صورتی که مد Simple باشید , داده های بین ساعات 12 شب تا 10 صبح رو از دست خواهید داد .

**Bulk-logged**

این مد با Full تفاوت چندانی ندارد , تنها تفاوت آنها در اعمال Bulk  انبوه هست .

مثلا فرض کنید یک میلیون رکورد رو به صورت انبوه درج کرده ایم , به ازای همه اون یک میلیون ما تنها یک تراکنش در فایل لاگ ذخیره میکردیم  در حالی که در مد Full یک میلیون تراکنش ثبت می کند .

بعضا دیتابیس هایی دیده شده است که علیرغم اینکه حجم فایل mdf آن کم است , حجم فایل log آن چندین برابر mdf آن هست .

**کاربرد Log Shrinking در SQL سرور**

 ممکن است ظرفیت زیادی از دیسک شما به یکباره اشغال شود و حجم Log File شما نیز افزایش پیدا کند . در چنین مواقعی شما باید بصورت دستی طی فرآیندی به نام Shrinking فضای اشغال شده توسط Log را رها کنید ، واژه Shrink به معنی چروک کردن ، کوچک کردن و جمع کردن است و این کاری است که دقیقا شما قرار است انجام دهید ، در صورتیکه با چنین مواردی مواجه شدید می توانید با استفاده از دستوری شبیه به دستور زیر Transaction Log خود را Shrink کنید :

|  |  |
| --- | --- |
|  | **DBCC SHRINKFILE(<filename>,<desired\_shrink\_size>)** |

توجه کنید که در دستور بالا قسمت desired\_Shrink\_Size فضایی که شما می خواهید در نهایت برای Log File شما بر روی دیسک وجود داشته باشد ، این فضا بر اساس مگابایت در نظر گرفته می شود و بهترین زمان برای اینکه کمترین میزان حجم را در این قسمت قرار دهید زمانی است که Transaction Log شما Truncate شدن است و شما مطمئن هستید اطلاعاتی در Transaction Log وجود ندارد.

**ایجاد یک دیتابیس با کد نویسی**

|  |
| --- |
| CREATE DATABASE [BOOK] ON PRIMARY  ( NAME = N'BOOK', FILENAME = N'D:\BOOK.mdf' , SIZE = 3200KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )  LOG ON  ( NAME = N'BOOK\_log', FILENAME = N'D:\BOOK\_log.LDF' , SIZE = 832KB , MAXSIZE = 2048GB , FILEGROWTH = 10%) |
| CREATE DATABASE BOOK2  ON ( NAME = 'BOOK2',FILENAME = 'D:\BOOK2.mdf' ) |

**جهت مشاهده جزئیات یک دیتابیس**

|  |
| --- |
| SELECT \* FROM sys.database\_files |

**انجام یک طراحی ساده Database و Normalize کردن آن**

خوب در این سناریو ما می خواهیم سفارش های مشتریان ( Customer Orders ) را در Database ذخیره کنیم

یک مشتری قاعدتا می تواند در هر سیستم فروشی چندین سفارش ( Order ) داشته باشد

یک سفارش ( Order ) می تواند شامل چندین کالا ( Product ) باشد

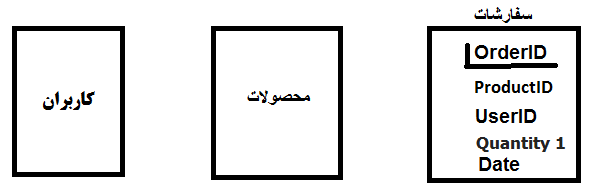
و در نهایت یک کالا ( Product) می تواند چندین بار سفارش ( Order ) داده شود.



همانطور که در جدول بالا مشاهده می کنید ما تمامی موجودیت های خود را در قالب یک Table در کنار هم قرار می دهیم. خوب جدول پیشفرضی که در بالا مشاهده می کنید شاید در حال حاضر کوچک به نظر برسد اما براحتی می تواند تبدیل به یک جدول بسیار بسیار پر حجم و پر از داده های تکراری شود.

* خوب همانطور که مشاهده می کنید در اولین Column ما تاریخ سفارش ( Order Date ) را قرار داده ایم.
* دومین Column شما نام مشتری ( Customer ) را مشاهده می کنید که اطلاعات هویتی اشخاصی می باشد که از شما خرید می کنند.
* سومین Column مربوط به نوع کالا یا محصول اول ( Product ) سفارش داده شده است.
* چهارمین Column تعداد سفارش های مربوط به کالا یا محصول اول را نمایش می دهد که ما به عنوان Quantity 1 آن را نمایش داده ایم ، به همین ترتیب پنجمین Column نوع محصول و کالای دوم و ششمین Column تعداد سفارش محصول دوم را به شما نمایش می دهد.

**در حالت ساده به صورت زیر است**



**عبارت CREATE TABLE برای ساختن یک جدول**

|  |  |
| --- | --- |
| CREATE TABLE table\_name  (  column\_name1 data\_type,  column\_name2 data\_type,  column\_name3 data\_type,....  ) | CREATE TABLE [Forums](  [ID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,  [CatID] [int] NULL references Categories(ID),  [ForumsName] [nvarchar](50) NULL,  [Subject] [nvarchar](50) NULL  ) |
| CREATE TABLE Persons  (  ID int,  LastName NVARCHAR(255),  FirstName varchar(255),  Address NCHAR(255)  ) | CREATE TABLE Persons  (  ID INT IDENTITY(10,1),  LastName NVARCHAR(255),  FirstName varchar(255),  Address NCHAR(255)  ) |
| CREATE TABLE Persons  (  ID NUMERIC(10,0) PRIMARY KEY,  ID2 INT IDENTITY(10,1),  LastName NVARCHAR(255),  FirstName varchar(255),  Address NCHAR(255)  ) | CREATE TABLE Persons  (  ID INT IDENTITY(1,1),  LastName NVARCHAR(255),  FirstName varchar(255),  Address NCHAR(255) DEFAULT (N'بوشهر')  ) |

**آشنایی باConstraint قوانین**

Constraint ها قوانین درج داده در Database ها هستند.

|  |
| --- |
| CREATE TABLE Persons  (  ID INT IDENTITY(1,1),  LastName NVARCHAR(255),  FirstName varchar(255),  Address NCHAR(255) DEFAULT (N'بوشهر'),  Age TINYINT CONSTRAINT [CK\_Persons] CHECK (Age>=18)  ) |
| CREATE TABLE Persons  (  ID INT IDENTITY(1,1),  LastName NVARCHAR(255),  FirstName varchar(255) CONSTRAINT [CK\_Persons\_FirstName] CHECK (FirstName <> 'Ali'),  Address NCHAR(255) DEFAULT (N'بوشهر'),  Age TINYINT CONSTRAINT [CK\_Persons\_Age] CHECK (Age>=18)  ) |
| CREATE TABLE Persons  (  ID INT IDENTITY(1,1),  LastName NVARCHAR(255),  FirstName varchar(255) CONSTRAINT [CK\_Persons\_FirstName] CHECK (FirstName <> 'Ali'),  Address NCHAR(255) DEFAULT (N'بوشهر'),  Age TINYINT CONSTRAINT [CK\_Persons\_Age] CHECK (Age>=18 AND Age IS NOT NULL)  ) |
| CREATE TABLE Persons  (  ID INT IDENTITY(1,1),  LastName NVARCHAR(255),  Code NVARCHAR(20) CONSTRAINT [CK\_Persons\_Code] CHECK (LEN(Code)=10 AND Code IS NOT NULL)  ) |

**جهت حذف جدول از دستور**

DROP TABLE Persons

**عملگر های And و Or**

* عملگر And برای اجرای دستور نياز دارد تا تمام شرط های تعيين شده برای آن درست باش
* عمگر Or فقط نياز دارد که حداقل يکی از شرط ها درست باشد .
* ترکیب عملگرهای And و Or

|  |
| --- |
| SELECT \* FROM Test WHERE Product=N'دفتر' AND ID=6 |
| SELECT \* FROM Test WHERE Product = N'مداد' Or Product = N'ماشین' |
| Select \* From Test  Where ((ID = 1+1 AND Product = N'مداد' ) OR (ID = 12 AND Product = N'ماشین')) |

**دستورات Insert – Update – Delete – Select**

**Insert**

|  |
| --- |
| INSERT INTO [Persons] ([LastName],[Code],[Age])  VALUES (N'Mohamad Hoosein','1236547899',20) |
| INSERT INTO [Persons] VALUES (N'Mohamad Hoosein','1236547899',20) |
| **درج 10 رکورد**  INSERT INTO [Persons] VALUES (N'Mohamad Hoosein','1236547899',20)  go 10 |
| INSERT INTO Customers1 (Name,[Address]) VALUES  (N'Mohamad Hoosein','1236547899',20),  (N'Mohamad Hoosein','1236547891',21),  (N'Mohamad Hoosein','1236547892',22) |

**Update**

|  |
| --- |
| update [Persons] SET LastName=N'Mohamad Hoosein',Code='1236547899',Age=29 WHERE ID=5 |
| update [Persons] SET Code='1236547899',Age=29 WHERE ID=5 AND LastName=N'Mohamad Hoosein' |

**Delete**

|  |
| --- |
| DELETE FROM [Persons] WHERE ID=5 AND LastName=N'Mohamad Hoosein' |
| DELETE FROM [Persons] |

**[**

**Select**

|  |
| --- |
| SELECT [ID],[LastName],[Code],[Age] FROM [Persons] |
| SELECT \* FROM [Persons]  WHERE LastName=N'Mohamad Hoosein' |

**Alias**

برای در نظر گرفتن يک نام مستعار و مجازی برای قرار گرفتن در بخش عنوان هر ستون در خروجی دستور Select

|  |
| --- |
| SELECT [ID] AS ‘کد',[LastName],[Code] AS ‘شماره دانشجویی’,[Age] FROM [Persons] |
| SELECT 'asd' AS sad,'asd' AS [محمد] ,'asdasd'AS N'میلاد','asd' AS [محمد رضا] |