**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**Data Definition Language**

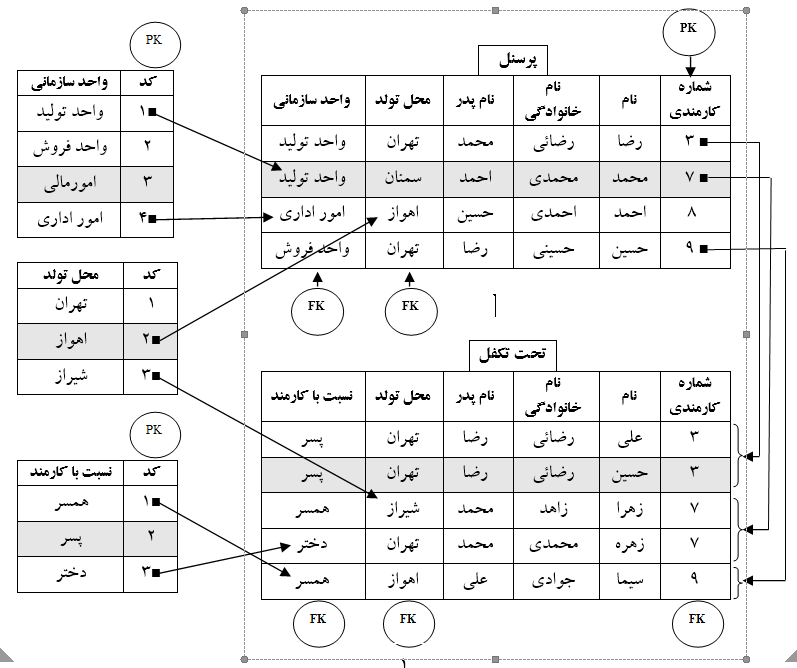
ایجاد، تغییر و حذف آبجکتها

**Table چیست؟**

داده ها در دیتابیس یا پایگاه داده داخل جدول یا [table](http://pars-soft.ir/../index.aspx?Article=What-is-a-Table) ها ذخیره میشوند. هر database  میتواند شامل چندین table باشد .هر جدول از تعدادی سطر و ستون تشکیل شده است.

* ستون اول نام فیلد با رعایت قوانین نامگذاری (Column Name)
* ستون دوم نوع داده فیلد (Data Type)
* ستون سوم (Allow Nulls)
* در ستو ن سوم مشخص میکنیم که آیا این فیلد می تواند مقدار null را در خود ذخیره کند یا خیر Allow nulls ایا فیلد اجازه ذخیره null را میدهد یا خیر ؟

برای تمام ستون ها در دیتابیس بسته به نوع کارکرد آن میتوانید نوع مورد نظر خود را تعریف کنید ، مقدارهایی که درون هر فیلد ذخیره میشود بایستی با نوع انتخابی فیلد مطابقت داشته باشد. **روابط بین جدول ، فیلد و رکورد را در مثال ذیل ملاحظه فرمائید:**

****

**Field چیست ؟**

به هر يک از خانه های ستون يک جدول [فيلد ( Field )](http://pars-soft.ir/../index.aspx?Article=What-is-a-Field) می گويند .

هر فيلد در بر گيرنده يک صفت و ويژگی برای موجوديت می باشد ، که دارای 2 جزء اصلی است :

1. اسم صفت خاصه : نام صفت مورد نظر را تعيين می کند . برای مثال فيلد نام ، نام خانوادگی ،  و ... در جدول اطلاعات مربوط به شخص.
2. مقدار صفت خاصه : در برگيرنده مقدار برای صفت مورد نظر است . برای مثال سعید " به عنوان مقدار برای فيلد نام .

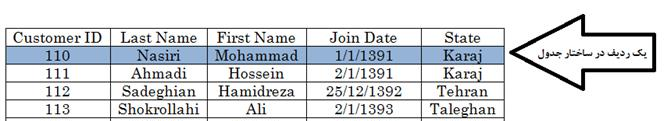


**Record چیست؟**

به سطرهاي يک جدول رکورد (Record) گفته میشود. هر رکورد مجموعه ای از اطلاعات طبقه بندی شده درباره يک موجوديت خاص است .

یک رکورد نشان دهنده یک ورودی در جدول است. یک جدول می تواند هر تعداد رکورد داشته باشد.

مثال از رکورد در پایگاه داده: اگر شما جدول “Students” برای ذخیره اطلاعات دانش آموزان داشته باشید، در این جدول یک رکورد نشان دهنده یک دانش آموز خواهد بود.



**ساخت جدول CREATE TABLE**

|  |
| --- |
| CREATE TABLE Persons  (PersonID int,LastName nvarchar(255),FirstName nvarchar(255)); |

**تغییر در جدول ALTER TABLE**

|  |  |
| --- | --- |
| ALTER TABLE table\_name ADD column\_name datatype | **Add Column** |
| ALTER TABLE Persons1 ALTER COLUMN id2 VARCHAR(50) | **Edit Column** |
| ALTER TABLE table\_name DROP COLUMN column\_name | **DROP Column** |

**حذف جدول** **DROP TABLE**

|  |
| --- |
| DROP TABLE table\_name |

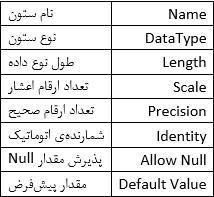
**ساخت جدول با استفاده از Designer**

* نظیم را انجام میدهیم کلید ذخیره و یا Ctrl+ S را میزنیم. با پیغام اخطاری مواجه می شویم

Saving changes is not permitted. The changes you have made require table to be dropped and re-created.

**راه حل:**

به منو Tools رفته بروی گزینه Option کلیک کنید. انگاه Designers را باز کنید و بروی ایتم Table and Database Designers. کلیک کنید و تیک Prevent saving changes that require the table to be re-created را بردارید.



**قید (Constraint)**

Constraint به معنای قید و محدودیت است. قید ها به ما اجازه میدهند قوانینی را تعیین کنیم که موتور پایگاه داده بطور اتوماتیک بر صحت اطلاعات وارد شده نظارت داشته باشد. و اجازه وارد کردن مقادیری که (integrity) (صحت، جامعیت) دیتابیس را بر هم میزنند ندهد .

**Constraintها شامل موارد ذیل می باشند:**

* NOT NULL: مشخص میکند که فیلد مقادیر Null را نمی پذیرد.
* Check: جهت محدود کردن مقادیر ورودی در فیلد مثلا محدود کنیم که مقدار ورودی در فیلد بین 100 تا 2000 باشد.
* UNIQUE: مقدار فیلد مورد نظر باید در تمامی رکورد های یک جدول منحصر به فرد باشد. مثلا هنگامی که جدولی برای نگهداری اطلاعات اشخاص ایجاد میکنیم. بروی فیلد کد ملی Unique Constraint میگذاریم تا اس کیو ال سرور اجازه وارد کردن یک کد ملی را برای دو نفر ندهد .
* PRIMARY KEY: کلید اصلی یک Constraint است هیچ دو رکوردی در جدول دارای مقدار یکسان در کلید اصلی نیستند.
* FOREIGN KEY: رابطه (Relationship) بین جداول را تحمیل میکند هنگامی که رابطه والد و فرزندی بین رکورد های دو جدول برقرار است foreign key اجاره نمیدهد رکورد والدی حذف شود که دارای فرزند است (در ادامه تشریح خواهد شد) و یا اینکه رکورد فرزندی درج شود بدون اینکه کلید اصلی والد ان درج شود.
* Default: تعیین مقدار پیشفرض برای فیلد. اگر مقداری در فیلد وارد نشد این مقدار بصورت اتوماتیک در فیلد درج میشود .

|  |  |
| --- | --- |
| CREATE TABLE Persons  (PersonID INT,LastName nvarchar(255) NULL,FirstName nvarchar(255) NOT NULL); | [**NOT NULL**](http://pars-soft.ir/index.aspx?Article=sql-not-null-constraint) |
| CREATE TABLE Persons (  ID int NOT NULL,  FirstName varchar(255),  Age int,  CHECK (Age>=18)  ) | **CHECK** |
| ALTER TABLE [dbo].[Persons] WITH CHECK ADD CHECK (([Age]>=(18))) | **CHECK** |
| ALTER TABLE Persons ADD CONSTRAINT uc\_PersonID UNIQUE (LastName) | **UNIQUE** |
| CREATE TABLE Persons  (  P\_Id int NOT NULL,  City varchar(255),  CONSTRAINT uc\_PersonID UNIQUE (P\_Id,LastName)  ) | **UNIQUE** |
| CREATE TABLE Persons  (  P\_Id int NOT NULL IDENTITY(1,1),  City varchar(255),  CONSTRAINT [PK\_Persons] PRIMARY KEY NONCLUSTERED (City)  ) | **UNIQUE** |
| CREATE TABLE Persons (PersonID INT PRIMARY KEY,LastName nvarchar(255) NULL); | [**PRIMARY KEY**](http://pars-soft.ir/index.aspx?Article=sql-rimary-key-constraint) |
| CREATE TABLE Persons  (  P\_Id int NOT NULL IDENTITY(1,1),  City varchar(255),  CONSTRAINT [PK\_Persons] PRIMARY KEY CLUSTERED (P\_Id)  ) | [**PRIMARY KEY**](http://pars-soft.ir/index.aspx?Article=sql-rimary-key-constraint) |
| ALTER TABLE dob.[OptionsValue] WITH CHECK ADD CONSTRAINT [FK\_OptionsValue\_OptionsType] FOREIGN KEY([IdOptionsType])  REFERENCES dob.[OptionsType] ([Id]) | **FOREIGN KEY** |
| ALTER TABLE [dbo].[Persons] ADD DEFAULT ('Sandnes') FOR [City] | **DEFAULT** |
| CREATE TABLE Persons  (PersonID INT PRIMARY KEY,LastName nvarchar(255) NULL DEFAULT('xxx')) | **DEFAULT** |
| ALTER TABLE [dbo].[Persons] DROP CONSTRAINT [DF\_\_Persons\_\_City\_\_1BE81D6E] | **DROP** |

**فعال و غیر فعال کردن CONSTRAINT ها**

|  |
| --- |
| -- Disable all table constraints  ALTER TABLE YourTableName NOCHECK CONSTRAINT ALL  -- Enable all table constraints  ALTER TABLE YourTableName CHECK CONSTRAINT ALL  -- ----------  -- Disable single constraint  ALTER TABLE YourTableName NOCHECK CONSTRAINT YourConstraint  -- Enable single constraint  ALTER TABLE YourTableName CHECK CONSTRAINT YourConstraint  -- ----------  -- Disable all constraints for database  EXEC sp\_MSforeachtable "ALTER TABLE ? NOCHECK CONSTRAINT all"  -- Enable all constraints for database  EXEC sp\_MSforeachtable "ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all"  select \* from sys.all\_objects where [name] like 'sp\_MSforeach%' |

**جدول موقت، Temporary Table**

هنگامی که می خواهید دیتایی در جداول اس کیوال سرور بطور دائم ذخیره نشود   
یا ممکن است دیتای محاسباتی داشته باشید و بخواهید بطور موقت انها را ذخیره کنید . از جداول موقت استفاده میشود .

**SQL Session**

وقوع تعامل کاربر با پایگاه داده از طریق دستورات اس کیو ال می باشد .   
هنگامی که یک کاربر به پایگاه داده متصل میشود از طریق management studio و یا یک application یک session برقرار میگردد . در ضمن این sql session کاربر می تواند دستورات tsql را اجرا نماید ، ساختار بانک اطلاعاتی ، جداول و سایر آبجکت ها را تغییر دهد یا اطلاعات جداول را دستکاری نماید.

هربار که یک پنجره new Query را باز میکنید در واقع یک session جدید ایجاد می شود و با بستن آن این session بسته خواهد شد.

|  |
| --- |
| CREATE TABLE #teststudent  (  ID int primary key,  FirstName nvarchar(50),  LastName nvarchar(50)  )  insert into #teststudent  values(1,N'رضا',N'احمدی'),(2,N'محمد',N'احمدی'),(3,N'وحید',N'احمدی')  select \* from #teststudent  drop table #teststudent |

* #با اسکپ لوکال
* ## با اسکپ گلوبال
* اگر با ## جدول teststudent ایجاد شود در new session دیگر نیز قابل استفاده است

**Temp Table**

در SQL server یک پایگاه داده از پیش تعریف شده با نام Temp db وجود دارد، در بعضی مواقع شما به یک جدول موقت نیاز دارید این جداول در این پایگاه داده ذخیره می شود و Temp Table نام دارد.

برای استفاده از جداول موقت دو حالت وجود دارد اگر قبل از این جداول از یک # استفاده شود این جدول Local بوده مثلا اگر در Stored Procedure دستور ساخت این جدول نوشته شده باشد به ازای هر کاربر شما این جدول ساخته می شود و در صورتی که کانکشن این کاربر از بین برود جدول وی نیز از بین می رود.

در حالت دوم اگر قبل از نام جدول از دو علامت ## استفاده شود این جدول Global خواهد بود. در این صورت سایر کاربران از همین جدول استفاده می کنند اگر کاربر به وجود آورنده جدول، جدول را حذف کند طبیعتا جدول دیگر توسط هیچ کاربری قابل دسترس نخواهد بود.

نمونه دستورات :

|  |
| --- |
| CREATE TABLE #Table (id INT , name NVARCHAR(MAX))  DECLARE @i INT =0  WHILE (@i <20)  BEGIN  INSERT INTO #Table ([id],[name]) VALUES (@i,'ESH')  SET @i+=1;  END  SELECT \* FROM #Table  DROP TABLE #Table |

استفاده از جدول موقت به طور کلی برای پردازش مقدار زیادی از اطلاعات است زیرا می تواند کار ها را به صورت موازی انجام دهد. در حالی که جدول متغیرها بهترین راه حل برای مقدار کمی از داده ها است.

**موارد استفاده از Temp Table ها :**

حتی الامکان می بایست از temp table ها استفاده نشود. زیرا علاوه بر این که سر بار زیادی بر روی sql می گذارند. Performance را نیز پایین می اورند. تنها در برخی استفاده از temp table ها مانعی ندارد. برخی از این استثنا ها عبارتند از :

* هنگامی که بخواهیم نتیجه یک Query را از یک nested stored procedure به stored procedure دیگری منتقل کنیم که در این صورت تنها راه حل ممکن استفاده از temp table ها است .
* در صورتی که بخواهیم در stored procedure از cursor استفاده کنیم . با توجه به سرعت پایین cursor ها می توان از Temp Table ها به عنوان راه حل استفاده نمود .
* سعی کنید فقط فیلد هایی را که نیاز دارید در Temp Table درج نمایید.
* از دستور Select Into برای ایجاد Temp Table استفاده نکنید بلکه ابتدا با استفاده از دستور Create Table جدول مجازی را ایجاد کنید و سپس با استفاده از دستور Insert Into رکورد های مورد نظر را در آن درج نمایید .
* حتما سعی کنید از clustered و non-clustered Index ها در Temp Table ها استفاده نمایید . ( مخصوصا هنگامی که تعداد رکوردهای موجود در Temp Table ها بسیار زیاد باشد)
* حتما پس از استفاده از Temp Table ها ان را حذف کنید ( در این صورت فضای Tempdb ازاد می شود) و هرگز منتظر نباشید تا با قطع شدن connection بطور اتوماتیک tempdb حذف شود .
* در صورتی که استفاده از Temp Table ها در برنامه هایتان زیاد است بهتر است با تنظیمات sql server مسیر ان را به یک دیسک مجزا منتقل کنید تا فضای اصلی کمتر اشغال گردد.
* هرگز Temp Table ها را در Transaction ها ایجاد نکنید چرا که در این صورت برخی از جداول سیستمی (sysindex , syscolumn , syscomments) در حالت Lock باقی می مانند و عملا تا پایان Transaction ازاد نمی شوند و بنابراین بسیاری از دستورات که توسط کاربران دیگر صادر می شود تا پایان Transaction غیر قابل اجرا خواهد بود . برای جلوگیری از این مشکل بهتر است قبل از Transaction ان را ایجاد کنید تا جداول سیستمی Lock نشوند .