



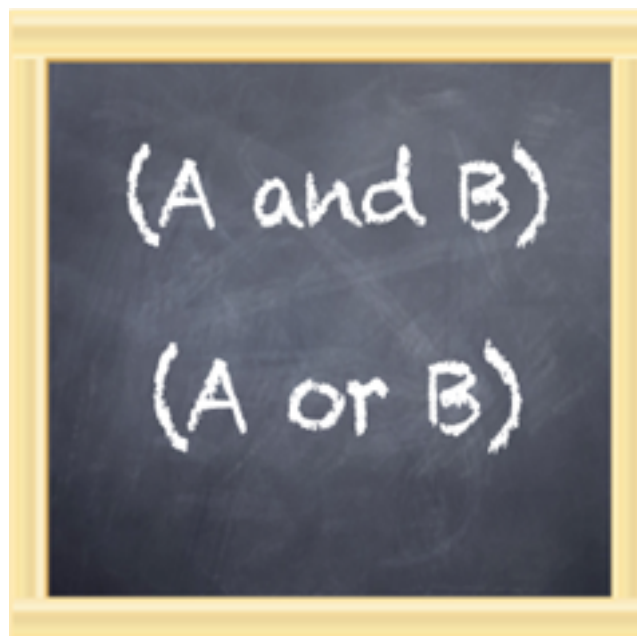
چگونه با ایندکس گذاری حرفه ای سرعت Database  
و گزارشات را به اوج برسانیم؟



مسعود طاهری

### درس اول

## آشنایی با مفاهیم اولیه SQL Inside



**در این درس با مباحث زیر آشنا خواهید شد:**

۱- آشنایی با فایل‌های تشکیل دهنده یک بانک اطلاعاتی

۲- Page (صفحه) چیست؟

۳- Extent چیست؟

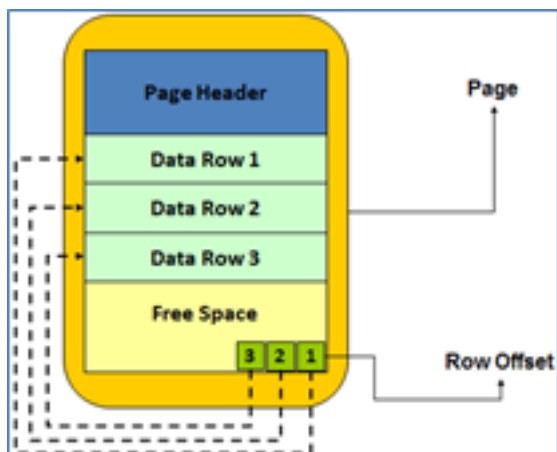
۴- اطلاعات درون LogFile چگونه ذخیره می‌شود؟

### آشنایی با فایل‌های تشکیل دهنده یک بانک اطلاعاتی

به طور کلی هر بانک اطلاعاتی در SQL Server حداقل از دو فایل (Data File , Log File) تشکیل شده است. در Data File کلیه اشیاء مربوط به یک بانک اطلاعاتی مانند جداول، فیلدها، رکوردها و... ذخیره می‌شود و در Log File (فایل ثبت وقایع) کلیه وقایع مربوط به بانک اطلاعاتی مانند ایجاد اشیاء، درج رکورد، حذف رکورد، تغییرات مربوط به بانک اطلاعاتی و... ثبت می‌گردد. وجود Log File در یک بانک اطلاعاتی به قدری حیاتی است که در صورت بروز اشکالات سیستمی برای بانک اطلاعاتی SQL Server با استفاده از این فایل امکان برگردان بانک اطلاعاتی به وضعیت پایدار را خواهد داشت.



### Page (صفحه) چیست؟



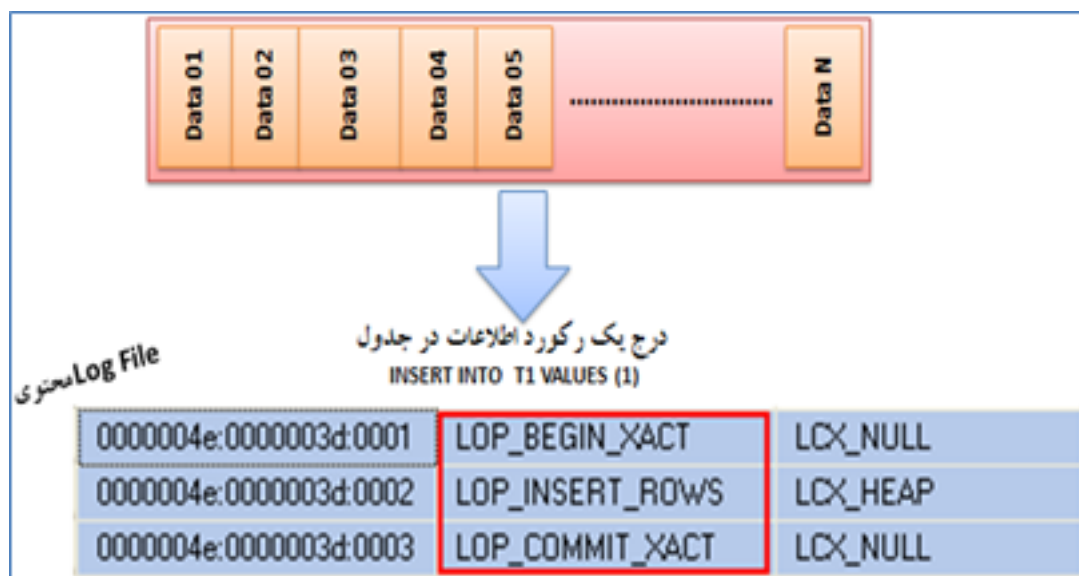
اساسی‌ترین واحد ذخیره اطلاعات در SQL Server صفحه (Page) نام دارد. هر Page در SQL Server هشت کیلوبایت (۸KB) یا ۸۱۹۲ بایت ظرفیت دارد که از این مقدار ۸۰۹۶ بایت جهت ذخیره اطلاعات مورد استفاده قرار می‌گیرد. درون Data File پر از Page‌های مختلف است که SQL Server از آن جهت ذخیره کردن اطلاعات جداول، فیلدها، ایندکس‌ها و... استفاده می‌کند.

### Extent چیست؟

هشت Page کنار هم (به هم پیوسته) یک Extent نام دارد. در حقیقت Extent ابتدایی‌ترین قسمتی است که SQL Server امکان مدیریت بر آن را دارد.

### اطلاعات درون Log File چگونه ذخیره می شود؟

همانطور که قبلاً اشاره شد در Log File (فایل ثبت وقایع) کلیه وقایع مربوط به بانک اطلاعاتی مانند ایجاد اشیاء، درج رکورد، حذف رکورد، تغییرات مربوط به بانک اطلاعاتی و... ثبت می گردد. رکوردهای Log موجود در Log File به صورت ترتیبی ذخیره می شوند. هر رکورد شامل یک ID است که مشخص کننده Transactionی است که رکورد متعلق به آن است.



## آشنایی با ساختار بانک اطلاعاتی Northwind



### در این درس با مباحث زیر آشنا خواهید شد:

- ۱- آشنایی با نحوه نصب و راه اندازی بانک اطلاعاتی Northwind
- ۲- بررسی جداول مهم بانک اطلاعاتی Northwind
- ۳- بررسی روابط بین جداول بانک اطلاعاتی Northwind

### آشنایی با نحوه نصب و راه اندازی بانک اطلاعاتی Northwind

برای نصب بانک اطلاعاتی Northwind می توانید از فایل Script آن استفاده کنید. این فایل Script حاوی دستورات TSQL مربوط به ایجاد بانک اطلاعاتی، ایجاد جداول، رکوردهای جداول و سایر اشیاء مربوط به بانک اطلاعاتی می باشد.



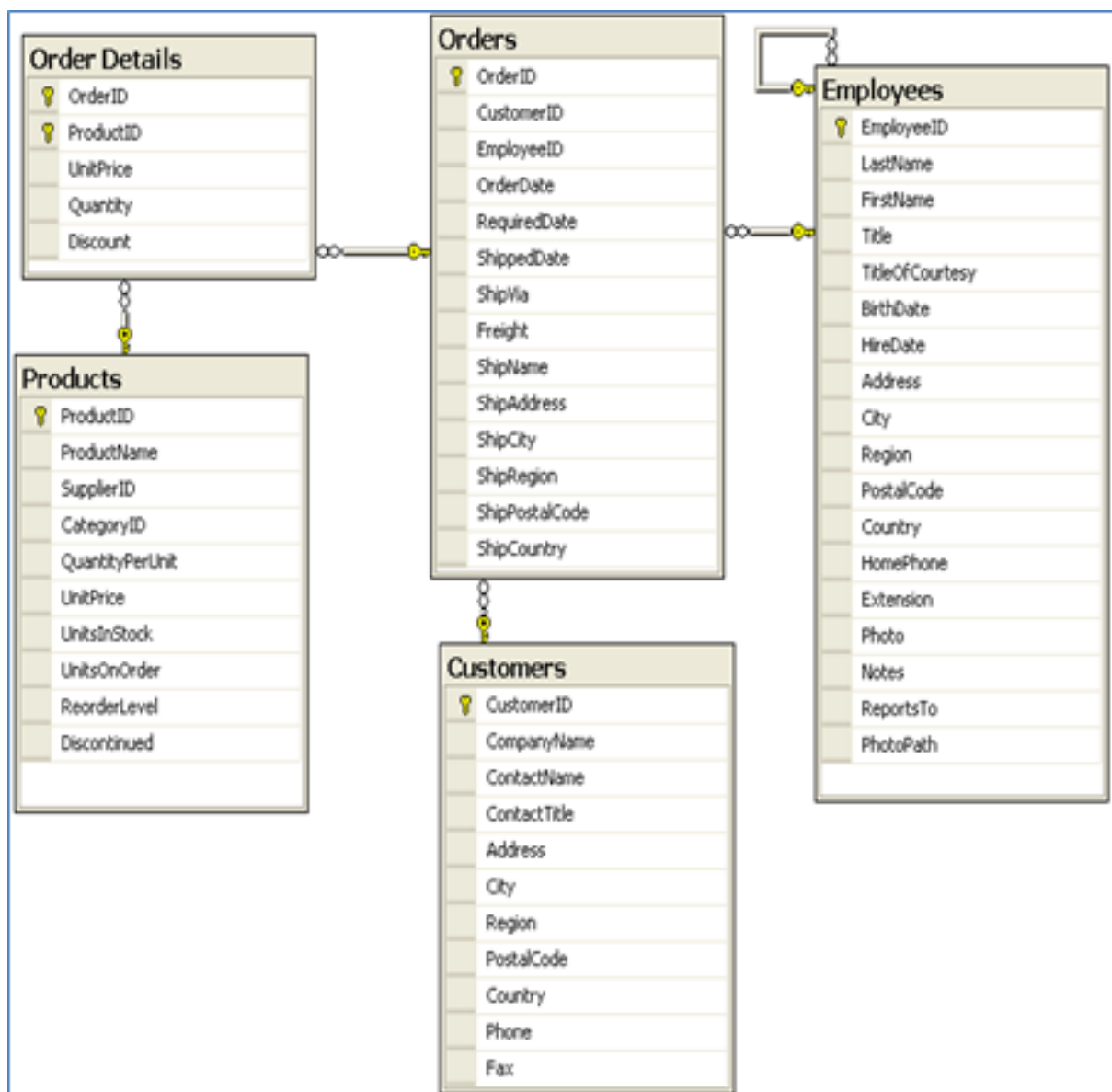
### بررسی جداول مهم بانک اطلاعاتی Northwind

بانک اطلاعاتی Northwind حاوی پنج جدول مهم و کاربردی است که محتوای این دوره بر اساس آن بنا شده است این جداول عبارتند از :

- جدول Orders: حاوی اطلاعات سفارشات شرکت Northwind
- جدول Order Details: حاوی جزئیات سفارشات
- جدول Employees: حاوی اطلاعات کارمندان شرکت Northwind
- جدول Customers: حاوی اطلاعات مشتریان شرکت Northwind
- جدول Products: حاوی محصولات شرکت Northwind

### بررسی روابط بین جداول بانک اطلاعاتی Northwind

همانند سایر بانک‌های اطلاعاتی هر کدام از جداول اشاره شده دارای روابط Master/Detail می‌باشند. در تصویر صفحه بعد جزئیات مربوط به روابط بین این پنج جدول را مشاهده خواهید کرد.



### درس ۳

## آشنایی با مفهوم Index



**در این درس با مباحث زیر آشنا خواهید شد:**

- ۱- بررسی مفهوم Index
- ۲- بررسی دلایل استفاده از Index
- ۳- بررسی انواع Indexها در SQL Server



### بررسی مفهوم Index

لیستی از داده‌ها که بر اساس موضوعی خاص (حروف الفبا، ترتیب عددی و...) مرتب شده باشند.

### بررسی دلایل استفاده از Index

مهمترین دلایل استفاده از Index عبارتند از:

- افزایش سرعت در بازیابی رکوردها.
- افزایش سرعت در عملیات Join (پیوند) بین دو جدول.
- Unique (یکتا) کردن رکوردها با استفاده از ایجاد Unique Index.

### بررسی انواع Indexها در SQL Server

به طور کلی در SQL Server پنج نوع ایندکس وجود دارد.

**Clustered Index:** این نوع ایندکس‌ها بر روی داده‌های عددی، رشته‌ای، تاریخی ایجاد می‌شوند. هنگام استفاده از این نوع ایندکس‌ها ترتیب فیزیکی رکوردها بر اساس کلید ایندکس تغییر پیدا می‌کند پس با در نظر گرفتن این نکته یک جدول صرفاً یک ایندکس از این نوع می‌تواند داشته باشد.

**NonClustered Index:** این نوع ایندکس‌ها بر روی داده‌های عددی، رشته‌ای، تاریخی ایجاد می‌شوند. اما برخلاف Clustered Indexها ترتیب فیزیکی رکوردها را تغییر نمی‌دهند بلکه در فضایی جدا از فضای جدول صرفاً کلید (های) ایندکس مرتب شده و ارتباطی مابین ایندکس و اصل جدول به وجود می‌آید. لازم به ذکر است که تعداد این نوع ایندکس‌ها به ازای یک جدول می‌تواند ۹۹۹ عدد باشد.

**XML Index:** همانطور که از نوع این نوع ایندکس‌ها مشخص است بر روی داده‌ها XMLی ایجاد می‌شوند.

• **Spatial Index**: این نوع ایندکس‌ها بر روی داده‌های فضایی (جغرافیایی) ایجاد می‌شوند. لازم به ذکر است در SQL Server دو نوع داده جغرافیایی وجود دارد:

**Geometry**: این نوع داده بر اساس هندسه دو بعدی (اقلیدسی) بوده و برای نگهداری داده‌هایی که در سطوح مسطح (نقطه، منحنی، چند ضلعی و...) هستند به کار می‌روند.

**Geography**: این نوع داده‌ها بر اساس شکل بیضی بوده و برای ذخیره‌سازی داده‌هایی که در سطح کره (مانند طول و عرض جغرافیایی) هستند به کار می‌روند.

• **Column Stored Index**: این نوع ایندکس‌ها در SQL Server 2012 معرفی شده‌اند. تفاوت این نوع ایندکس‌ها با ایندکس‌ها قبلی در معماری ذخیره‌سازی اطلاعات می‌باشد که این موضوع باعث شده است که عملیات جستجوی اطلاعات با سرعت بیشتری انجام شود.

## آشنایی با ساختار Clustered Index ها (ایندکس های خوشه ای)



**در این درس با مباحث زیر آشنا خواهید شد:**

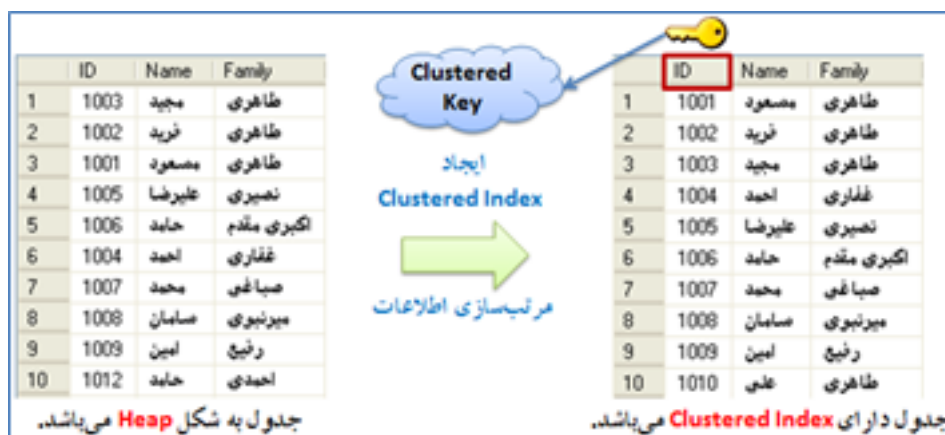
۱- Clustered Index چیست؟

۲- Heap چیست؟

۳- مراحل ایجاد Clustered Index در SQL Server

### Clustered Index چیست؟

نوع خاصی از ایندکس است که در آن ترتیب رکوردهای فیزیکی موجود در جدول بر اساس کلید ایندکس خواهد بود. لازم به ذکر است که با توجه به تعریف فوق یک جدول فقط و فقط می تواند یک نوع ایندکس از نوع Clustered Index داشته باشد.



### Heap چیست؟

Heap (توده، فله) جدولی است که فاقد Clustered Index می باشد. ترتیب رکوردها در این نوع جداول بر اساس هیچ کلید خاصی نمی باشد. لازم به ذکر است این نوع جدول یکی از بدترین نوع جداول در SQL Server می باشد.

### مراحل ایجاد Clustered Index در SQL Server

ایجاد Clustered Index در SQL Server طی چهار مرحله انجام می شود:

۱. ایجاد یک کپی از جدول به همراه اطلاعات آن.
۲. مرتب سازی جدول کپی شده بر اساس کلید ایندکس.
۳. ایجاد ساختار ایندکس (صفحات ریشه و برگ).
۴. حذف جدول اصلی و جایگزینی جدول کپی شده با آن.

### درس ۵

## بررسی IO & Time Statistics و Execution Plan هنگام استفاده از Clustered Index

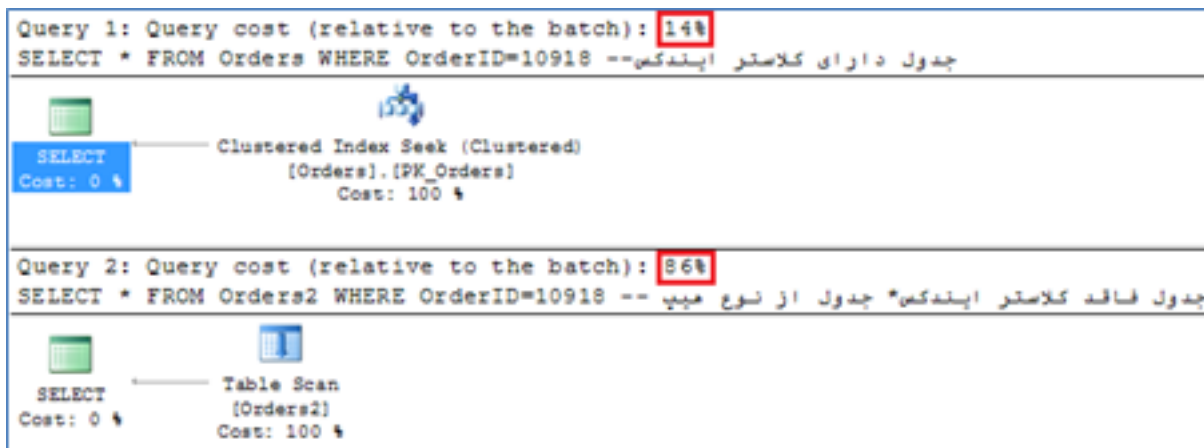


**در این درس با مباحث زیر آشنا خواهید شد:**

- ۱- مقایسه Execution Plan (نقشه اجرایی) Clustered Index با Heap
- ۲- مقایسه IO Statistics (آمار Clustered Index) با IO Heap
- ۳- مقایسه Time Statistics (آمار زمان اجرا) Clustered Index با Heap

### مقایسه Execution Plan (نقشه اجرایی) Clustered Index با Heap

جداولی که دارای Clustered Index هستند باعث تغییر ترتیب رکوردها می‌شوند بنابراین در این نوع جداول ترتیب رکوردها بر اساس کلید ایندکس می‌باشد. اما چنانچه جدول به شکل Heap باشد رکوردهای موجود در آن هیچ ترتیب خاصی ندارند. حال چنانچه Execution Plan (نقشه اجرایی) جدولی که دارای Clustered Index است با Execution Plan (نقشه اجرایی) جدولی که به شکل Heap می‌باشد را مقایسه کنید مشاهده خواهید کرد هزینه اجرای کوئری‌هایی که دارای Clustered Index هستند به مراتب پایین‌تر از جداول Heap می‌باشد.



### مقایسه IO Statistics (آمار IO) ایندکس با Clustered با Heap

همانطور که در تصاویر زیر مشاهده می‌کنید اجرای کوئری در جدولی که دارای Clustered Index باشد IO کمتری نسبت به جدولی که Heap است دارد.

«جدول دارای Clustered Index»

```
SELECT * FROM Orders WHERE OrderID=10918 -- جدول دارای کلاستر ایندکس
GO
(1 row(s) affected)
Table 'Orders'. Scan count 0, logical reads 2, physical reads 2, read-ahead reads 0,
lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
```

### «جدول از نوع Heap»

```
SELECT * FROM Orders2 WHERE OrderID=10918 -- جدول فاقد کلاستر ایندکس* جدول از نوع هیپ
GO

(1 row(s) affected)
Table 'Orders2'. Scan count 1, logical reads 22, physical reads 0, read-ahead reads 27,
lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
```

## مقایسه Statistics Time (آمار زمان اجرا) با Clustered Index

همانطور که در تصاویر زیر مشاهده می کنید اجرای کوئری در جدولی که دارای Clustered Index باشد زمان کمتری نسبت به جدولی که Heap است دارد.

### «جدول دارای کلاستر ایندکس»

```
SELECT * FROM Orders WHERE OrderID=10918 -- جدول دارای کلاستر ایندکس*
GO

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 43 ms.
```

### «جدول از نوع Heap»

```
SELECT * FROM Orders2 WHERE OrderID=10918 -- جدول فاقد کلاستر ایندکس* جدول از نوع هیپ
GO

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 64 ms.
```

## Primary Key & Unique Key و تأثیر استفاده از آنها در ایندکس



**در این درس با مباحث زیر آشنا خواهید شد:**

- ۱- بررسی ویژگی‌های Unique Key
- ۲- بررسی ویژگی‌های Primary Key



### بررسی ویژگی‌های Unique Key

در SQL Server چنانچه بخواهیم مقدار موجود در یک فیلد و یا تعدادی از فیلدها یکتا باشد کافی است که فیلد و یا فیلدهای مورد نظر را به شکل Unique Key در نظر بگیریم.

**استفاده از Unique Key باعث می‌شود که:**

- مقادیر موجود در داخل فیلد و یا فیلدها، یکتا (غیر تکراری) باشد.
- فیلد و یا فیلدهایی که به عنوان Unique Key هستند یکبار مقدار Null را به عنوان مقدار مجاز دریافت نمایند.

نکته اصلی که باید درباره Unique Key بدانید این است که کنترل یکتا بودن مقادیر فیلدها در Unique Key توسط Unique Non Clustered Index مدیریت می‌شود.

```
CREATE TABLE Students
(
    ID INT IDENTITY PRIMARY KEY,
    NationalCode BIGINT UNIQUE,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(70)
)
```

استفاده از Unique Non Clustered Index جهت پیاده‌سازی Unique Key

index_name	index_description	index_keys
1 PK_Students__3214EC272312112D	clustered, unique, primary key located on PRIMARY	ID
2 UQ_Students__3DFA4106EFF50DF6	nonclustered, unique, unique key located on PRIMARY	NationalCode

### بررسی ویژگی‌های Primary Key

مطابق قواعد نرمال‌سازی کلیه جداول باید دارای یک کلید اصلی باشد که با استفاده از آن به راحتی بتوان به رکورد مورد نظر دسترسی پیدا کرد

**استفاده از Primary Key باعث می‌شود که:**

- مقادیر موجود در داخل فیلد و یا فیلدها، یکتا (غیر تکراری) باشد.
- فیلد و یا فیلدهایی که به عنوان Primary Key هستند مقدار Null را نمی‌توانند به عنوان مقدار مجاز دریافت نمایند.

# جنون سرعت در

## SQL Server 2012

### دوره کامل آموزشی :



# جنون سرعت در



Microsoft®  
SQL Server® 2012

### ۵ راز فوق پیشرفته تکنیک‌های Index گذاری حرفه‌ای

یکماه عضویت  
الماسی رایگان



کتابچه تکمیلی  
نحوه ایندکس  
گذاری حرفه‌ای



بیش از  
۶ ساعت  
فیلم آموزشی  
فارسی

جهت **دانلود رایگان** سه درس (بیش از یک ساعت) این دوره کافی است بر روی لینک زیر کلیک کنید.

<http://www.NikAmooz.com/SpeedinSQLServer.html>