SQL Server
Part: Indexing & Transaction

آزمایشگاه پایگاه داده
استاد درس: حمیدرضا رمضانی
h.ramezany72@gmail.com

انواع INDEX

- ساختاری است در SQL که موجب بازیابی سریع اطلاعات خواسته شده، در یک جدول می‌شود.
  - B-tree

- برای ساخت Index از یک یا چند ستون جدول استفاده می‌شود که به آن‌ها، Index Key و Key Columns گفته می‌شود.

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع INDEX

- Indexها در SQL Server به دو دسته کلی تقسیم می‌شوند
- Clustered Indexes
  1. در هر جدول فقط یک Clustered Index وجود دارد
  2. داده‌ها برحسب Clustered Key مرتب می‌شوند
  3. برای دسترسی به داده‌ها در گره‌های برگ نیازی به اشاره گر نداریم (داده‌ها مستقیم در گره‌های برگ ذخیره می‌شوند)
  4. مرتب سازی و ذخیره داده‌ها بصورت فیزیکی می‌باشد
  5. با افزودن کلید اصلی به جدول بصورت اتوماتیک یک Clustered Index بر روی جدول ایجاد می‌شود
- Non-Clustered Indexes

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع INDEX

- Non-Clustered Indexes

1. مرتب سازی و ذخیره داده‌ها بصورت فیزیکی نمی‌باشد
2. در سطح گره‌های برگ در B-tree اشاره گری به داده‌های اصلی تشکیل می شود.
3. محتوای صفحه‌های برگ شامل اشاره‌گری با یک شماره صفحه و یک شماره سطر می‌باشد
4. تعریف چندین Non-Clustered Indexes در یک جدول ممکن می‌باشد
5. امکان تعریف بصورت ساده و ترکیبی از ستون‌ها می‌باشد
6. حداکثر می‌توان ۲۴۹ Non-Clustered Indexes بر روی یک جدول داشت

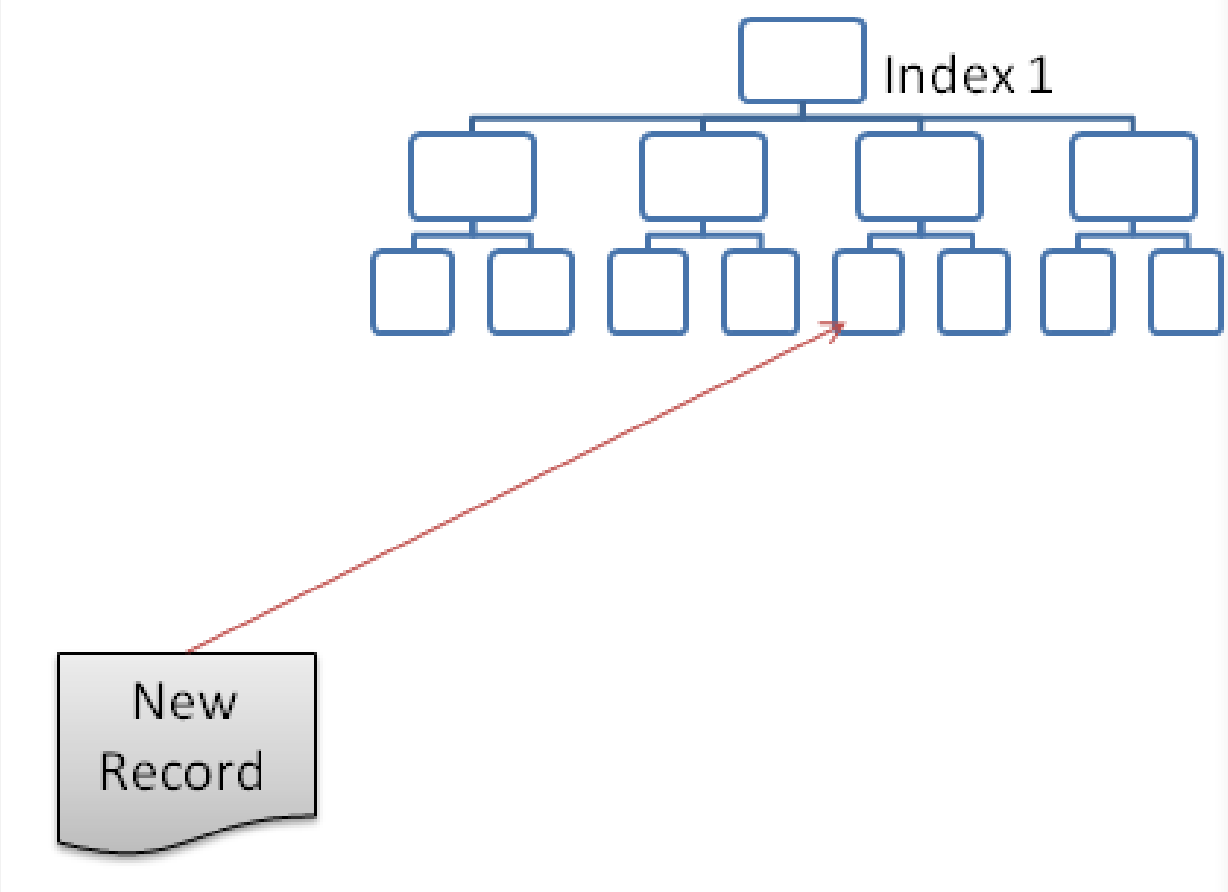
آزمایشگاه پایگاه داده
حمیدرضا رمضانی

قالب کلی تعریف INDEX

```
Create  [Unique] [Clustered | Non Clustered] Index  نام
    ON    نام جدول یا دیدگاه  (نام ستونها)
    [With  [pad_Index] [,] FillFactor = x]
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

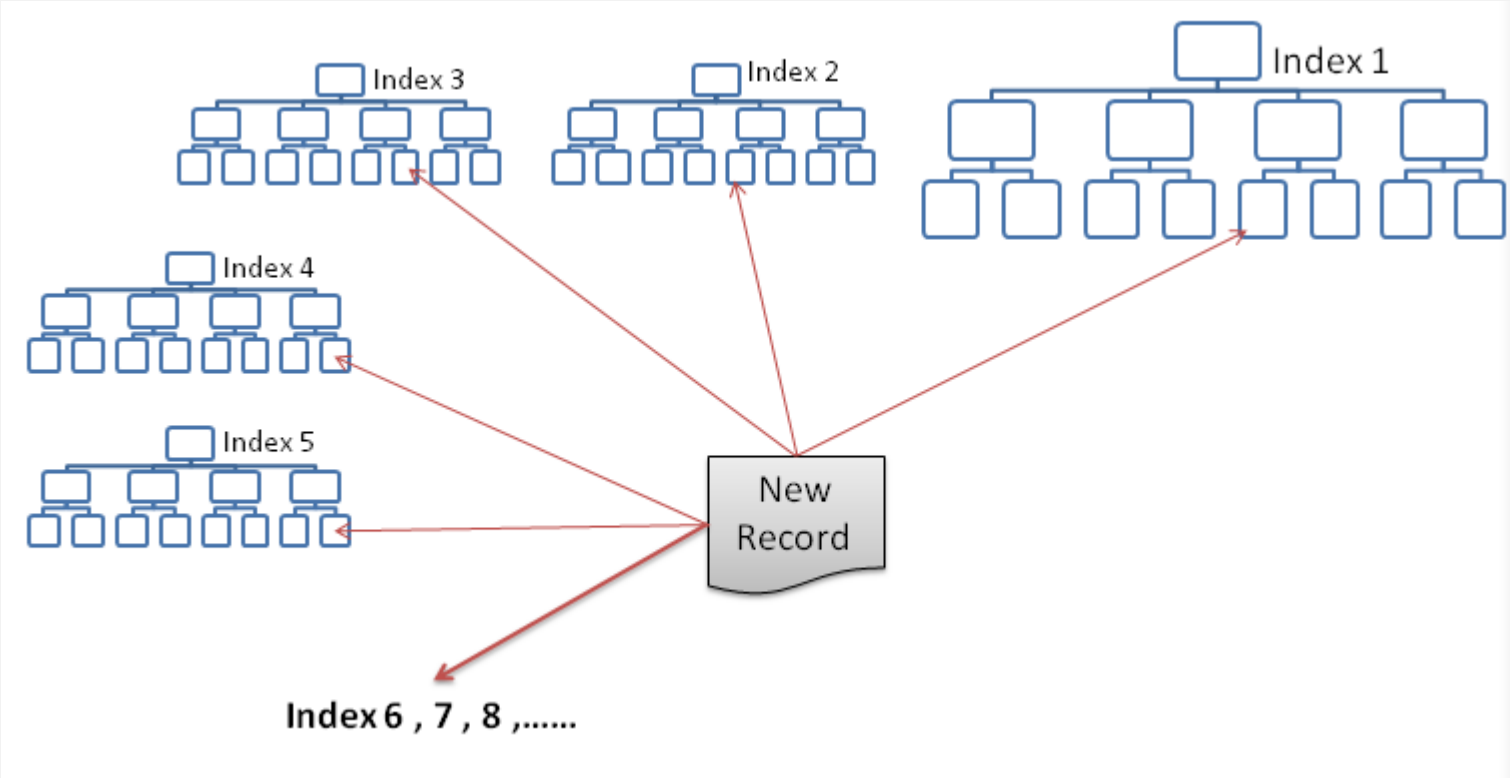
مقدمه‌ای بر پارامترهای INDEX



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

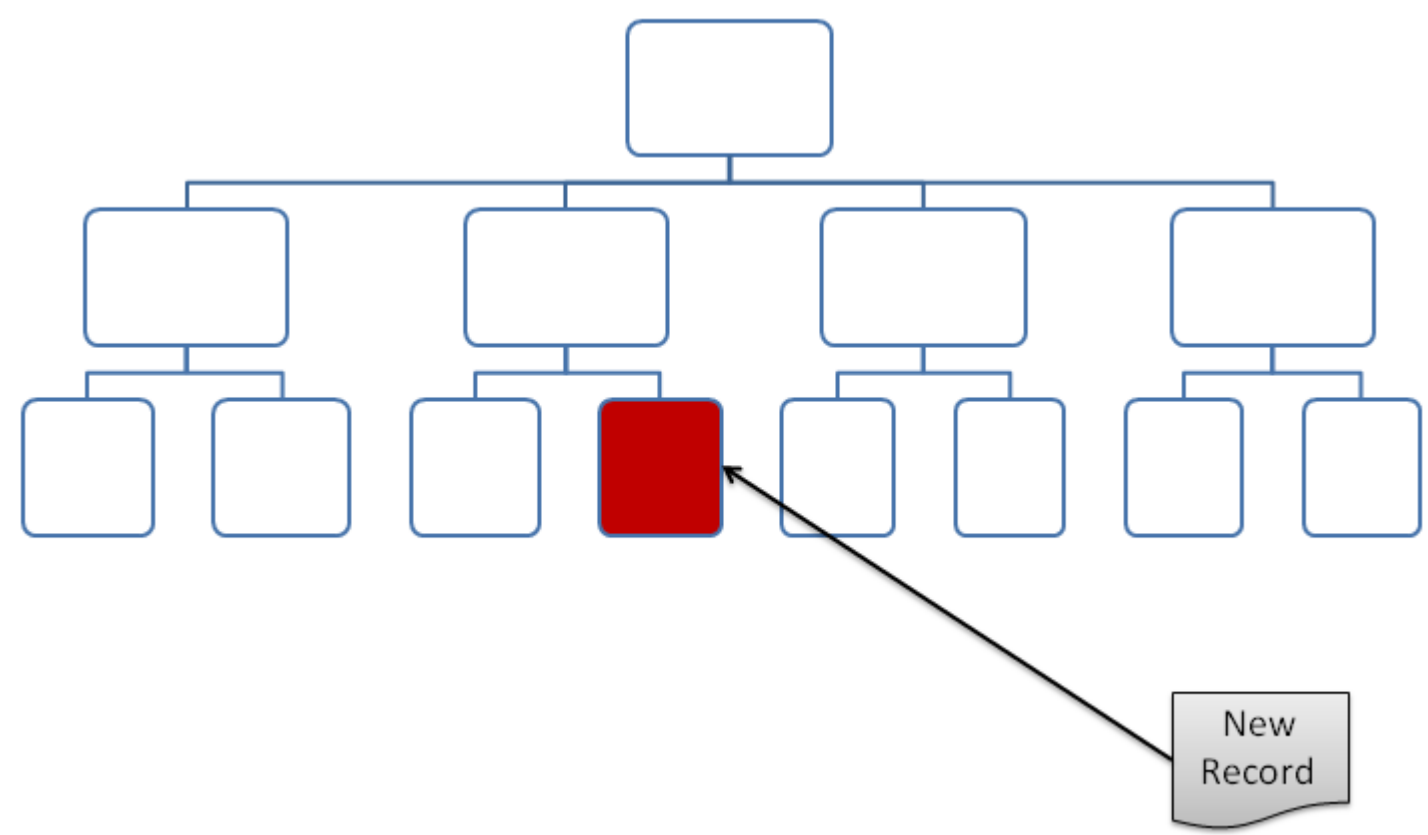
مقدمه‌ای بر پارامترهای INDEX



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

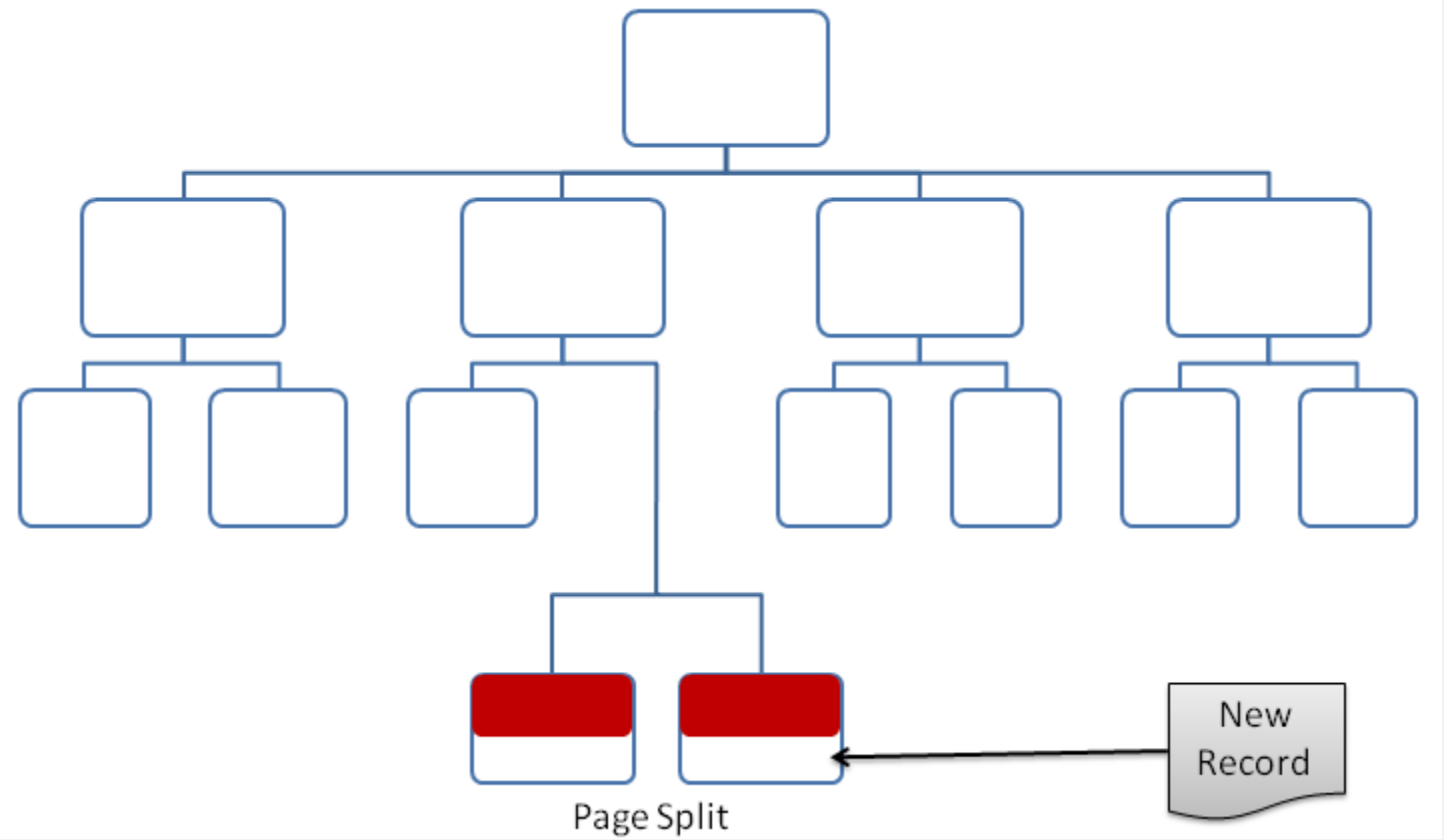
مقدمه‌ای بر پارامترهای INDEX



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

مقدمه‌ای بر پارامترهای INDEX



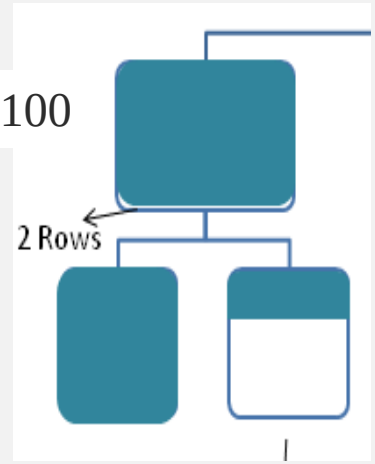
آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

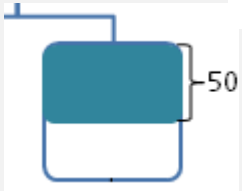
قالب کلی تعریف INDEX

```
Create  [Unique] [Clustered | Non Clustered] Index  نام
    ON    نام جدول یا دیدگاه  (نام ستونها)
    [With  [pad_Index] [,] FillFactor = x]
```

Fillfactor = 0 or 100

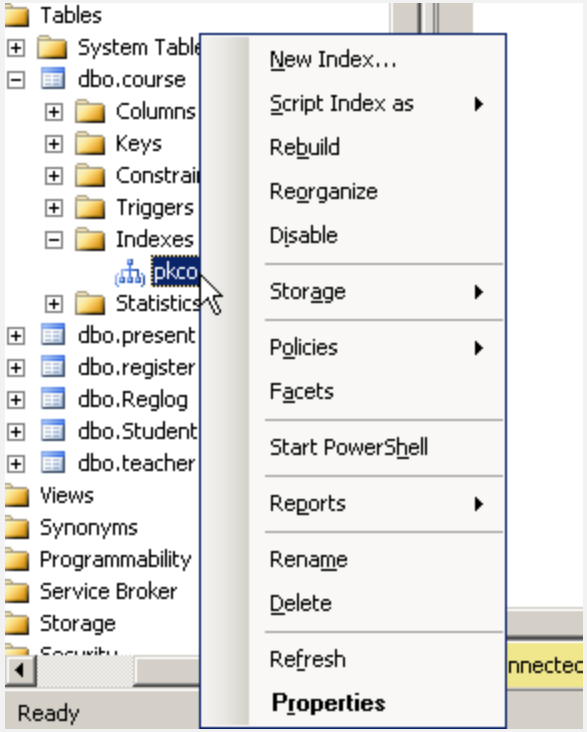


Fillfactor = 50



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

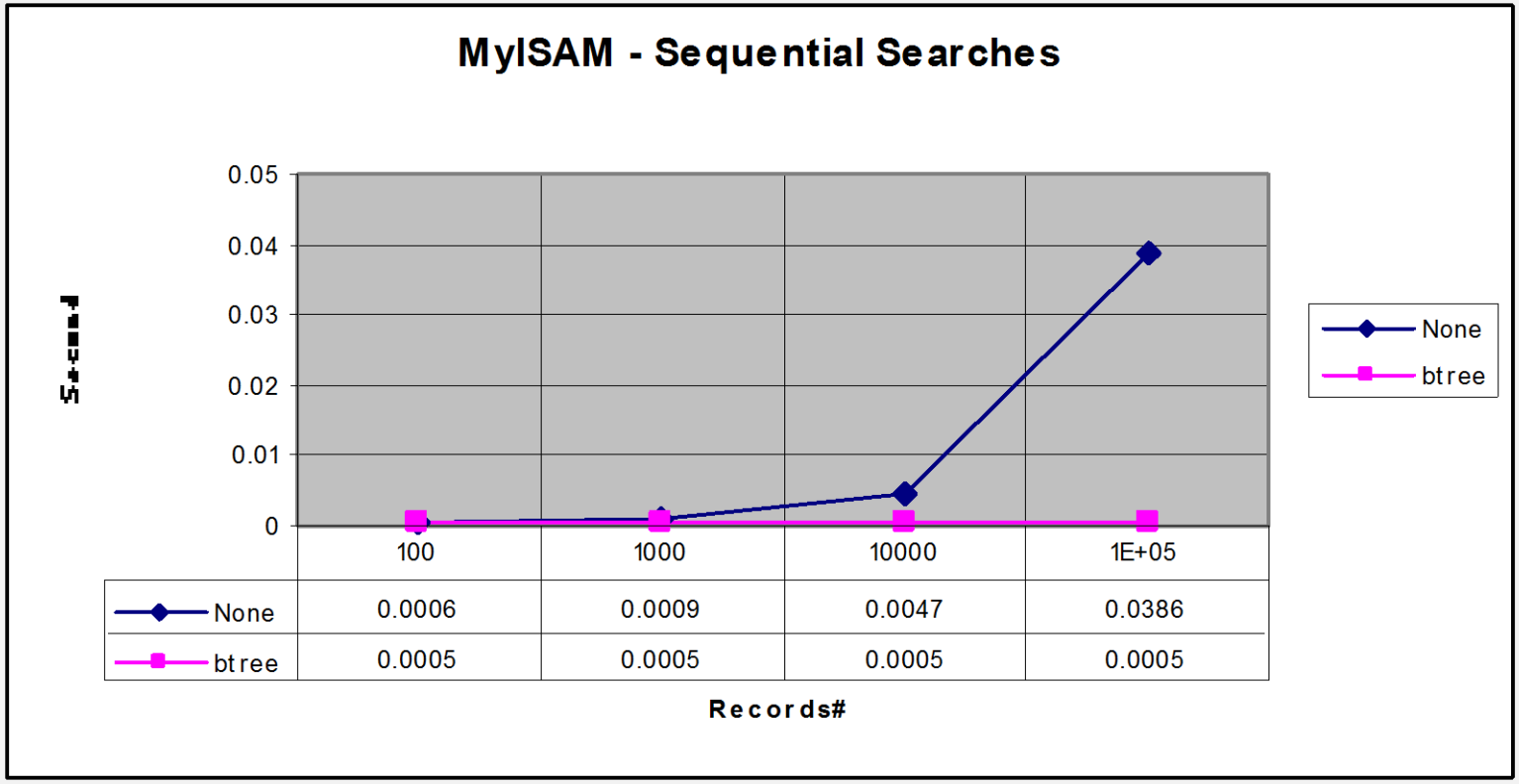
ایجاد INDEX بصورت گرافیکی



آزمایشگاه پایگاه داده

حمیدرضا رمضانی

کارایی استفاده از INDEX

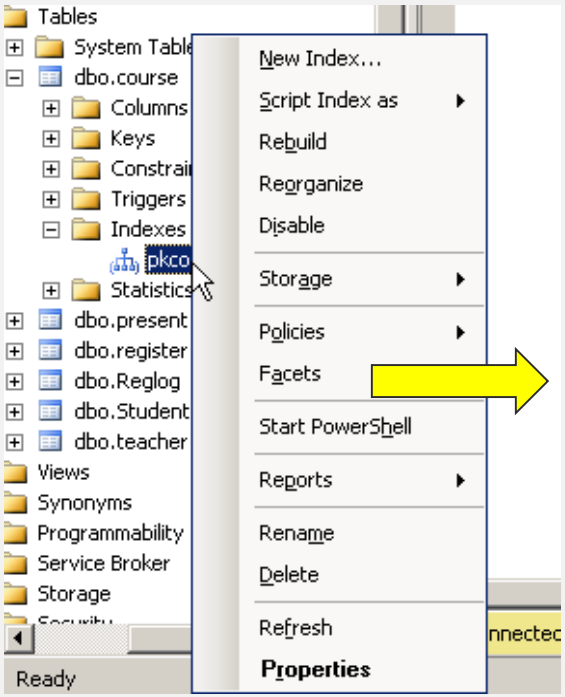


آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

REBUILD INDEX

- گرافیکی :



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

تراکنشها در SQL Server

به مجموعه‌ای از دستورات که می بایست یا همگی با هم با موفقیت اجرا شوند و یا هیچ‌کدام اجرا نشوند، تراکنش گفته می‌شود . تراکنش‌ها باعث حفظ جامعیت داده‌های ذخیره شده در بانک اطلاعاتی می شوند .

مزایای اصلی استفاده از تراکنشها در بانک‌های اطلاعاتی که مختصراً ACID نامیده می شوند، به شرح زیر هستند :

1) **Atomicity (تجزیه ناپذیری)** : تعریف دستورات در قالب یک فعالیت عملیاتی را به صورتیکه یا کلیه عملیات با هم اجرا شوند ویا هیچکدام اجرا نشوند را Atomicity می نامند .
2) **Consistency (پایداری)** : یک تراکنش، پس از خاتمه، می بایست داده‌ها را در یک وضعیت پایدار قرار دهد، به عنوان مثال در یک بانک اطلاعاتی رابطه‌ای، پس از خاتمه یک تراکنش، کلیه قوانین جامعیت داده‌ها، بایستی به داده‌های تغییر یافته توسط تراکنش اعمال گردد و همچنین ساختارهای داخلی داده‌های ذخیره شده مانند Indexها بایستی پس از اعمال تغییرات بازسازی و به وضعیت پایدار برسند.
3) **Isolation (جدا سازی)** : در هنگام کار با تراکنش‌ها، یکی از مهمترین موارد، امکان دسترسی همزمان یک یا چند کاربر به یک منبع داده مشترک است. تغییرات در یک تراکنش همزمان می بایست از تغییرات در تراکنش همزمان دیگر، جدا باشد .

تراکنشها در SQL Server

4) **Durability (مقاومت یا دوام)** : یک تراکنش پس از خاتمه می‌بایست دارای تاثیرات دائمی و ماندگار باشد. این بدان معنی است که عدم سازگاری ناشی از خرابی سیستم مانند قطع Power یا قطع شبکه و ... توسط تراکنش قابل کنترل و تصمیم گیری باشد.

در SQL Server سه دسته امکانات برای رسیدن به اهداف فوق وجود دارد :

1) امکانات قفل گذاری (Locking) که محیط را برای رسیدن به Isolation مناسب، مهیا می سازد .
2) امکانات واقعه نگاری (logging) که در صورت هر نوع خرابی ناشی از سیستم عامل، شبکه، سخت افزار، برق، یا حتی نسخه بانک اطلاعاتی، با شروع مجدد، وضعیت داده‌ها را به حالت قبل از شروع تراکنش باز می گرداند و در جهت رسیدن به Durability بکار گرفته می شود .
3) امکانات مدیریت تراکنشها (Transaction Management) که اصولاً جهت پیش برد اهداف Atomicity و Consistency بکار گرفته می شود . در واقع پس از آغاز، یک تراکنش، بایستی بطور موفقیت آمیزی خاتمه یابد، یا اینکه نسخه جاری مدیریت بانک اطلاعاتی، همه داده‌های تغییر یافته در طول تراکنش را به وضعیت قبل از شروع تراکنش باز گرداند .

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

تراکنشها در SQL Server

در هنگام بروز هر نوع خطا در هنگام اجرای یک تراکنش، عملیاتی تحت عنوان Recovery آغاز می شود .

عملیات Recovery معمولاً توسط SQL Server مدیریت می شوند که Automatic Recovery نامیده می‌شود . این عملیات خود سه دسته هستند :

۱- با شروع مجدد سرویس SQL Server

۲- با در خواست کاربر و اجرای دستور Rollback

۳- سرویس خودکار مدیریت تراکنشها

SQL Server هنگام شروع یک تراکنش، وضعیت جاری را در فایل‌های Log ذخیره، و چنانچه نیاز به بازیابی بود، در هنگام Recovery، اطلاعات مورد نظر را از فایلهای Log بازخوانی می‌نماید .

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع تراکنشها در SQL Server

دو دسته اصلی از تراکنشها در SQL Server وجود دارند :

۱- Single Transactions

۲- Distributed Transactions

دسته اول تراکنشها، فقط روی یک بانک اطلاعاتی قابل استفاده هستند. در صورتیکه حوزه عملیاتی یک تراکنش بیش از یک بانک اطلاعاتی باشد، بایستی از دسته دوم تراکنشها، استفاده شود. در این حالت حتی این امکان وجود دارد که تراکنشها بر روی دو Server نیز استفاده شوند. این امکان با استفاده از سرویس MSDTC (Microsoft Distributed Transaction Coordinator) قابل انجام است .

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع تراکنشها در SQL Server

سه دسته اصلی از تراکنشهای Single Transactions در SQL Server وجود دارد :

۱- تراکنشهای خودکار Auto commit Transactions

۲- تراکنشهای صریح Explicit Transactions

۳- تراکنشهای ضمنی Implicit Transactions

دسته اول تراکنشهای خودکار :

این دسته از تراکنشها، توسط SQL Server و در هنگام اجرای دستورات Insert، Update و Delete آغاز می شوند . در صورت موفقیت آمیز بودن و عدم وجود خطا بصورت خودکار Commit می شوند و در غیر اینصورت عملیات آنها، لغو می‌گردد .

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع تراکنشها در SQL Server

Begin Transaction ← شروع تراکنش

دسته دوم تراکنشهای صریح :

این دسته از تراکنشها، دارای نقطه شروع و خاتمه مشخصی هستند .

به شکل مقابل توجه کنید :

Commit Transaction ← موفقیت آمیز بودن تراکنش

Rollback Transaction ← **Automatic Recovery**

مثال :

```
Begin      Transaction                          نقطه شروع
      Begin      Try
            Insert      Into      Content   (…)
                  Values    (…)
            Insert      Into      News      (…)
                  Values    (…)
            Commit   Transaction
      End Try
      Begin      Catch
            RollBack Transaction                نقطه خاتمه
      End        Catch
```

آزمایشگاه پایگاه داده

حمیدرضا رمضانی

SAVE POINT

```
Begin try
  Begin transaction
        T-SQL Command 1
        Save transaction S1
        T-SQL Command 2
  Commit transaction
End try
Begin catch
  Rollback transaction S1
  Commit transaction
End catch
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

انواع تراکنشها در SQL Server

دسته سوم تراکنشهای ضمنی:

این دسته از تراکنشها، شباهت زیادی به تراکنشهای صریح دارند، با این تفاوت که یکبار Begin می شوند و می توانند چندین بار commit شوند.

شروع تراکنش صریحاً توسط دستور Begin Transaction تصریح نمی شود و تراکنش‌های ضمنی با دستور زیر فعال یا غیر فعال می شوند :

Set Implicit_Transactions {On|Off}

```
Set Implicit_Transactions ON
        T-SQL Command
        T-SQL Command
Commit Transaction
        T-SQL Command
        T-SQL Command
Rollback Transaction
Set Implicit_Transactions OFF
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

انواع تراکنشها در SQL Server

مثال :

```
Use [Lab-SimpleCMS]
Declare     @ID         Int
Begin       Transaction
   Begin Try
            Insert  Into [Content]
               Values  (1,GetDate(),'SQL Server 2008 Released')
            Set         @ID         =   SCOPE_IDENTITY()
            Insert  Into News
               Values  (@ID,
                        'New Version of SQL Server Is Released ,.... ' ,
                        GetDate() ,
                        DateAdd(Month,1, GetDate()) ,
                        1)
            Commit      Transaction
   End Try
   Begin   Catch
            Print Error_Message()
            RollBack    Transaction
   End         Catch
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین

- بانک اطلاعاتی برای پرواز ایجاد کنید – Flight D.B
- جداول زیر برای آن ایجاد کنید: Ticket & Reserve
- تعدادی پرواز درون آن درج کنید.
- Stored Procedure بنویسید که رزرو بلیط F.K F.K

  در هر دو جدول ثبت شود.
- Stored Procedure بنویسید که کاربر بتواند بلیط

  خرید شده را در هر دو جدول کنسل کند.

Ticket Table

Column Name	Data Type
Tno	Varchar(50)
Passno	Varchar(50)
Fname	Nvarchar(50)
Lname	Nvarchar(50)
Fno	Int
Fdate	datetime

Reserve Table

Column Name	Data Type
Fno	Int
Fdate	Datetime
Source	Varchar(50)
Target	Varchar(50)
Capacity	Smallint
Number	smallint

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)