

عنوان مقاله: ایندکس گذاری در SQL server چیست؟ معرفی ۱۲ نوع از پرکاربردترین ایندکس ها

نویسنده مقاله: تیم فنی نیکآموز

تاریخ انتشار: ۱۸ آبان ۱۴۰۲

منبع: <https://nikamooz.com/what-is-indexing>



ایندکس گذاری در SQL Server ، با هدف بهینه‌سازی عملیات استخراج داده‌ها از پایگاه داده (Database) و بهبود کارایی کوئری‌ها استفاده می‌شود. در این مطلب، ابتدا به این سؤال پاسخ داده می‌شود که مفهوم ایندکس گذاری چیست و سپس در بخش‌های آتی، انواع ایندکس گذاری به همراه مزایا و معایب هر یک از آن‌ها مورد بررسی قرار می‌گیرند. به‌عنوان مکمل این مقاله، پیشنهاد می‌کنیم [ویدئو بررسی مفهوم ایندکس در SQL Server](#) را نیز تماشا کنید.

### مفهوم ایندکس گذاری چیست ؟

به‌طور کلی، مفهوم ایندکس گذاری (Indexing | شاخص گذاری) فرآیندی است که در آن، اطلاعات یا داده‌ها در قالب خاصی سازماندهی می‌شوند. این سازماندهی باید به گونه‌ای باشد که دسترسی به اطلاعات تسهیل پیدا کند. به‌عنوان مثال، فرض کنید قصد دارید بخش مشخصی از یک کتاب جامع را مطالعه کنید. در چنین شرایطی، کافی است به فهرست مطالب کتاب مراجعه کنید و صفحه مرتبط با آن موضوع را پیدا کنید. در عمل، ایندکس گذاری نیز مشابه همین مثال ساده، به‌عنوان یک راه حل کمکی به کار می‌رود. با این دید مقدماتی از مفهوم Indexing، در ادامه به بررسی ایندکس گذاری در پایگاه داده (Database) پرداخته می‌شود.

### Index در پایگاه داده چیست ؟

ایندکس گذاری در پایگاه داده روشی است که با کمک آن می‌توان سرعت و کارایی عملیات استخراج داده‌ها، شامل جستجو (Search)، کوئری‌نویسی و مرتب سازی (Sort) را بهبود داد. در فرآیند ایندکس گذاری ، یک ساختمان داده یا همان Index ایجاد می‌شود که به‌واسطه آن، می‌توان به‌راحتی به اطلاعات درون پایگاه داده دسترسی داشت. در عمل، ایندکس گذاری در SQL Server با هدف بهینه‌سازی کارایی کوئری‌ها استفاده می‌شود؛ زیرا این فرآیند، یک راه حل کارآمد و سریع برای مشاهده سطرهای خاصی از جدول‌ها به شما ارائه می‌دهد و دیگر نیازی نیست [سیستم مدیریت پایگاه داده \(RDBMS\)](#) کل جدول پایگاه داده را به‌ترتیب بررسی کند.

معمولاً ایندکس‌ها بر روی یک یا چند ستون (Columns | Fields) از یک جدول پایگاه داده ایجاد می‌شوند. در چنین شرایطی، یک نسخه از داده‌های ایندکس گذاری شده همراه با یک اشاره‌گر (Pointer) به جایی که واقعاً سطرها در جدول قرار دارند، ذخیره خواهد شد. زمانی که بخواهید کوئری خاصی را اجرا کنید که دارای ستون‌های ایندکس گذاری شده باشد، سیستم مدیریت پایگاه داده می‌تواند با سرعت بیشتری به داده‌های مرتبط با آن کوئری دسترسی پیدا کند.



### کاربرد ایندکس گذاری در SQL Server چیست ؟

ایندکس گذاری در SQL Server در بهینه سازی کارایی (Performance) پایگاه های داده نقش کلیدی دارد؛ زیرا با استفاده از آن، این امکان به وجود می آید که فرآیند استخراج داده ها سریع و کارآمد انجام شود. مواردی همچون کاهش عملیات I/O، افزایش سرعت دسترسی به داده ها، یکپارچگی داده ها، بهینه سازی کوئری ها و **عملیات Join** و Filtering، همگی کاربرد ایندکس گذاری در SQL Server محسوب می شوند. از آنجایی که ایندکس گذاری در SQL Server از کاربردهای قابل توجهی دارند، لذا پیشنهاد می کنیم نگاهی به **مقاله قاتلین ایندکس در SQL Server** ببیندازید.

### مقایسه روش های Clustered و Nonclustered

پیش از معرفی انواع ایندکس گذاری در SQL Server، قصد داریم به دسته بندی آن ها به دو گروه بپردازیم. به طور کلی، ایندکس های خوشه ای (Clustered Indexes)، مشخصه هایی همچون نیاز به محیط ذخیره سازی جداگانه، تعیین ترتیب فیزیکی قرارگیری سطرهای داده را دارند و می توان به ازای هر جدول، فقط یک ایندکس گذاری Clustered ایجاد کرد. در نقطه مقابل، در ایندکس گذاری NonClustered یا غیرخوشه ای، از یک ساختمان داده مستقل برای ذخیره سازی اشاره گرها و مقادیر کلید ایندکس استفاده می شود و این امکان وجود دارد که برای هر یک از جدول ها، چند Non-clustered Index ایجاد کرد. با این دیدگاه کلی از ایندکس گذاری در SQL Server، در ادامه انواع مختلف ایندکس گذاری را به طور مفصل بررسی می کنیم.

## پرکاربردترین انواع ایندکس ها

انواع ایندکس گذاری در پایگاه داده و مزایا و معایب آن ها در این بخش مورد بررسی قرار می گیرند.

### Clustered index

Clustered index نوعی ایندکس گذاری در پایگاه داده رابطه ای محسوب می شود که ترتیب فیزیکی سطرهای داده در جدول را تعیین می کند. این روش ایندکس گذاری در SQL Server که یک RDBMS است، مورد استفاده قرار می گیرد. طبیعتاً هر جدول تنها می تواند یک ایندکس گذاری خوشه ای یا همان کلاستری داشته باشد.

### مزایای Clustered index

مهم ترین مزیت های ایندکس گذاری کلاستری عبارتند از:

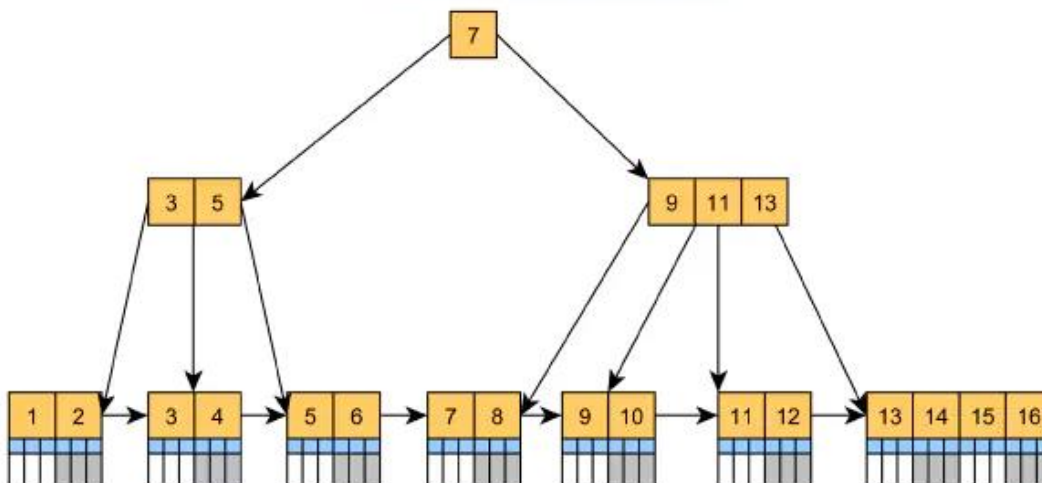
- نگهداری ترتیب فیزیکی داده های جدول و سرعت بالا در استخراج
- بهینگی و کارایی بالا برای Range Query ها
- عدم نیاز به بررسی تمام جدول در زمان استخراج داده ها
- سربار ذخیره سازی حداقلی

### معایب Clustered index

هرچند این نوع از ایندکس گذاری در SQL Server ، مزیت هایی دارا است، اما کاستی هایی نیز دارد. برخی از معایب ایندکس گذاری کلاستری به شرح زیر است:

- اثرگذاری مستقیم ایندکس گذاری به فضای ذخیره سازی فیزیکی داده ها (دیسک)
- محدودیت قرار دادن یک ایندکس به ازای هر جدول از پایگاه داده
- سربار بالا برای درج و به روز رسانی جداول و همچنین افزایش Fragmentation

## Clustered Indexes



## Non-Clustered Index

Non-Clustered Index نوع دیگری از ایندکس گذاری در SQL Server محسوب می‌شوند که یک راه کار جایگزین برای استخراج داده‌ها از جداول پایگاه داده ارائه می‌دهند. برخلاف ایندکس گذاری کلاستری، در این روش محل ذخیره‌سازی فیزیکی سطرهای جدول نگهداری نمی‌شود. بلکه در Nonclustered Indexing، یک ساختمان داده جداگانه ذخیره‌سازی می‌شود که در آن، زیرمجموعه‌ای از ستون‌های جدول و اشاره‌گر به سطرهاى واقعی داده‌ها وجود دارند. در [مقاله نحوه صحیح ایجاد Nonclustered Index](#) به این موضوع پرداخته‌ایم و برای انجام صحیح آن در [پایگاه داده SQL Server](#) پیشنهاد می‌کنیم مقاله را مطالعه نمایید.

## مزایای Non-Clustered Index

مزیت‌های ایندکس گذاری غیرخوشه ای در ادامه لیست شده‌اند:

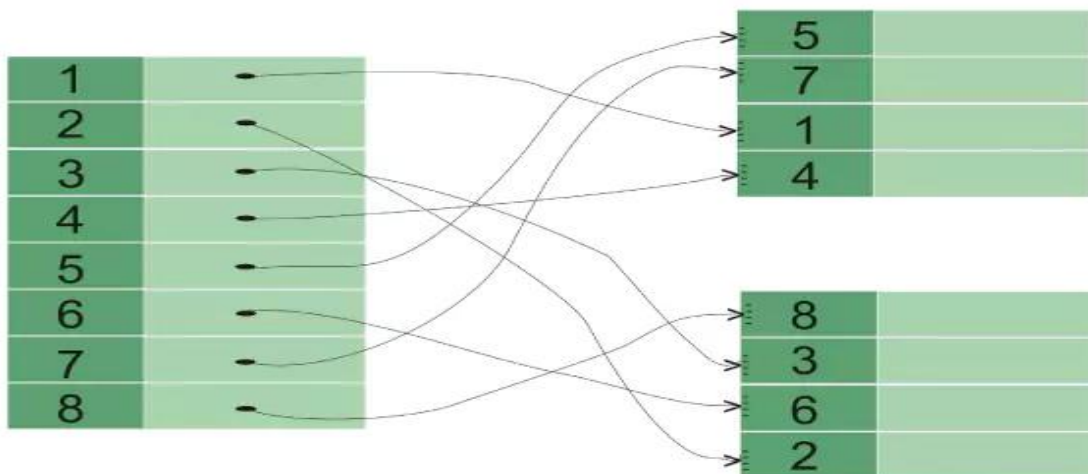
- وجود یک ساختار داده جداگانه و مستقل بودن آن از محیط ذخیره‌سازی فیزیکی جداول
- امکان ایجاد چندین ایندکس گذاری برای هر جدول
- قابلیت استخراج سریع داده‌ها
- انعطاف‌پذیری بالا و پشتیبانی از الگوهای کوئری مختلف

## معایب Non-Clustered Index

لیست زیر به معایب ایندکس گذاری Non Clustered اشاره دارد:

- نیاز به فضای ذخیره‌سازی اضافه
- احتمال افزایش پیچیدگی هنگام استفاده از چند ایندکس گذاری Non-Clustered
- کاهش کارایی در عملیات نوشتن (شامل درج، به‌روزرسانی و حذف داده‌ها)
- وجود سربار مربوط به نگهداری و اثرگذاری آن بر روی کارایی کوئری‌ها

## NonClustered Indexes



## Filtered Index

یکی از روش‌های مناسب ایندکس گذاری در SQL Server ، استفاده از Filter Index است. Filter Index نوعی ایندکس گذاری غیرخوشه ای به‌شمار می‌رود و برای ایجاد آن، مجموعه‌ای از سطرهای جدول، براساس یک شرط فیلترسازی مشخص به کار می‌روند. برخلاف روش‌های سنتی ایندکس گذاری Non-cluster ، که در آن‌ها تمام سطرهای جدول پوشش داده می‌شوند، در Filtered Index ، فقط سطرهای مطابق با معیار فیلترسازی در نظر گرفته خواهند شد.

### مزایای Filtered Index

موارد زیر به‌عنوان مزیت های Filtered Index شناخته می‌شوند:

- بهبود کارایی کوئری‌هایی که به‌طور خاص به داده‌های فیلتر شده مرتبط هستند.
- کاهش اندازه ایندکس گذاری و سربار نگهداری در مقایسه با سایر روش‌های سنتی Non-Clustered
- امکان استفاده از آن برای کوئری‌های انتخابی، **پارتیشن سازی داده ها** (Data Partitioning)، آرشیو دیتا و اعمال قوانین برای کیفیت داده‌ها

### معایب Filtered Index

موارد زیر معایب Filtered Index محسوب می‌شوند:

- کارایی محدود شده به کوئری‌های هدف
- نیاز به استفاده از محیط ذخیره‌سازی اضافی
- نیاز به نگهداری دائم و اطمینان از کارایی در طول زمان

## Filtered Index



## Unique Index

استفاده از Unique Index به عنوان روش ایندکس گذاری در SQL Server ، باعث می‌شود تا مقادیر یک یا چند ستون جدول، یکتا باشد. این یکتایی (Uniqueness) یعنی هرکدام از مقادیرهای موجود در ستون ایندکس گذاری شده، امکان تکرارپذیری را نخواهند داشت. در واقع، Unique Index ها با جلوگیری از درج مقادیر تکراری یا همان غیریکتا، باعث می‌شوند که یکپارچگی داده‌ها (Data Integrity) حفظ شود. شایان ذکر است که Unique Index می‌تواند با هدف **اعمال محدودیت کلید اصلی** (Primary Key Constraint) روی یک جدول مورد استفاده قرار بگیرد. شما می‌توانید برای آشنایی با این Constraint و اسکریپت آن، [مقاله اسکریپت های پرکاربرد SQL](#) را مطالعه کنید.

## مزایای Unique Index

مشخصه‌های مثبت Unique Index ها به شرح زیر است:

- تضمین یکپارچگی داده‌ها
- امکان پیاده‌سازی مفهوم کلید اصلی (PK) با استفاده از آن
- قابلیت اجرای محدودیت‌های یکتایی بر روی **کلیدهای کاندید** (Candidate Keys) و **کلیدهای طبیعی** (Natural Keys)
- امکان ایجاد محدودیت یکتایی روی ترکیبی از مقادیر ستون‌ها (Composite Unique Index)
- استخراج داده‌ها به صورت کارآمد

## معایب Unique Index

کاستی‌های Unique عبارتند از:

- نیاز به محیط ذخیره‌سازی اضافه
- احتمال کاهش کارایی در عملیات های درج، به‌روزرسانی و حذف داده‌ها
- نیاز به نگهداری مستمر با هدف مدیریت Fragmentation

## Covering Index

یک نوع جالب از ایندکس گذاری در SQL Server ، روش Covering Index است که با نام‌های دیگری همچون Covered Index و Included Index نیز به آن اشاره می‌شود. Covering Index یک روش ایندکس گذاری غیرخوشه ای (Non-clustered) محسوب می‌شود که با استفاده از آن، تمام ستون‌های موردنیاز برای یک کوئری خاص، بدون دسترسی به جدول واقعی ارائه می‌شوند. به بیان دیگر، این ایندکس گذاری در SQL Server ، می‌تواند همه ستون‌هایی را «پوشش دهد» یا اصطلاحاً Cover کند که برای اجرای کوئری به آن‌ها نیاز دارید. هدف از این تکنیک ایندکس گذاری، بهینه سازی کارایی کوئری‌ها است و می‌توان از آن به منظور مرتب سازی کارآمد نیز بهره برد.

## مزایای Covering Index

موارد زیر، مزیت‌های اصلی Covering Index محسوب می‌شوند:

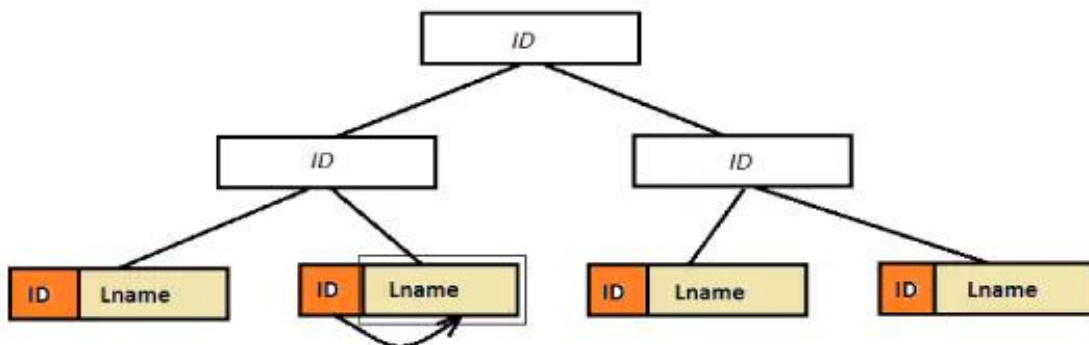
- بهبود کارایی کوئری‌ها و استخراج سریع داده‌ها
- کاهش عملیات ورودی / خروجی (I/O)
- امکان استفاده از آن برای مرتب‌سازی کارآمد داده‌ها

## معایب Covering Index

معایب Covering Index عبارتند از:

- نیاز به محیط ذخیره‌سازی اضافی
- کاهش سرعت در کارایی عملیات نوشتن (Delete و Update، Insert)
- امکان افزایش پیچیدگی در [شمای پایگاه داده](#) (Database Schema)

## Covering Index



## Columnstore Index

یکی رویکرد دیگر از ایندکس گذاری در SQL Server، روش Columnstore Index محسوب می‌شود که با هدف بهینه‌سازی ذخیره‌سازی و استخراج داده‌ها از مجموعه داده‌های وسیع طراحی شده است. در Columnstore Index، به جای اینکه ذخیره‌سازی و مدیریت داده‌ها با فرمت مبتنی بر سطرها انجام شود، این امور در قالب ستونی انجام خواهد شد. به‌طور خاص، این ایندکس گذاری برای انبارسازی داده‌ها (Data Warehousing) و لود کاری تجزیه و تحلیل مطلوب است. به‌عنوان مثال، اگر روی Dataset‌های گسترده، عملیات تجمیعی (Aggregation)، پالایش (Filtering) یا گزارش‌گیری انجام می‌دهید، این روش ایندکس گذاری در SQL Server به کارتان می‌آید.



## مزایای Columnstore Index

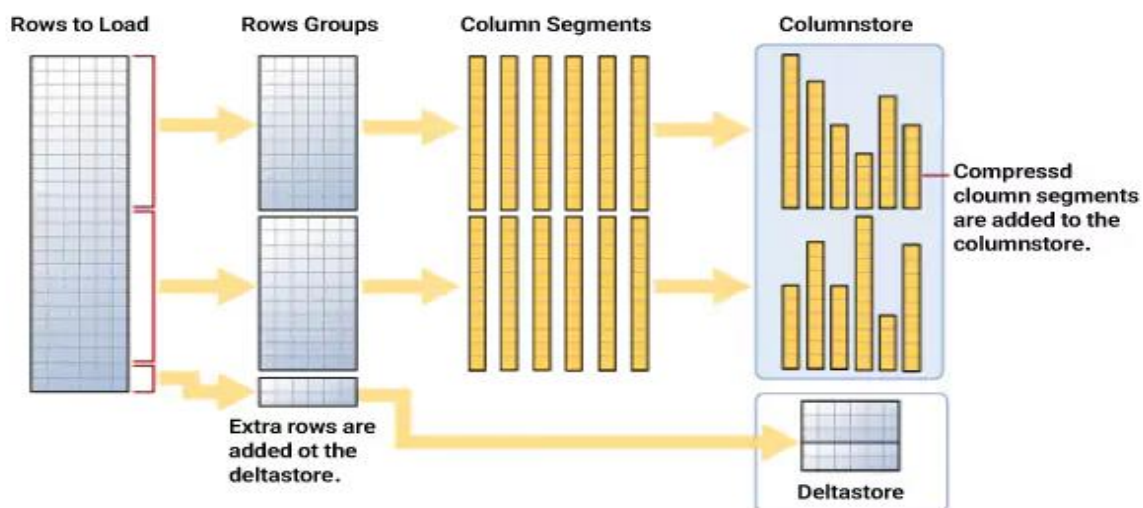
مزیت‌های Columnstore Index در ادامه فهرست شده‌اند:

- امکان فشرده‌سازی داده‌ها (Data Compression) و کاهش فضای ذخیره‌سازی مورد نیاز
- بهره‌وری از **قابلیت پردازش موازی** (Parallel Processing)
- بهبود کارایی کوئری‌ها (به ویژه در Aggregation و Filtering)
- امکان به‌کارگیری آن در Data Warehousing، گزارش‌گیری و یکپارچه‌سازی کلان داده‌ها (Big Data)

## معایب Columnstore Index

موارد زیر، از کاستی‌های روش ایندکس گذاری Columnstore Index به‌شمار می‌روند:

- کارایی محدود شده و مناسب نبودن آن‌ها برای لود کاری تراکنشی (Transactional)
- نیاز به نگهداری با هدف مدیریت Fragmentation



## Bitmap Index

Bitmap Index در SQL Server، نوعی روش ایندکس گذاری است که برای بهینه‌سازی کارایی کوئری‌ها استفاده می‌شود. Bitmap Index به‌طور خاص برای بهینه‌سازی کوئری‌هایی به‌کار می‌رود که در آن‌ها Cardinality ستون‌های جدول پایین است. این یعنی، تعداد محدودی از ستون‌ها دارای مقادیر متمایز (Distinct) هستند. به‌دلیل اینکه در این ایندکس گذاری، از نمایش دودویی داده‌ها یا اصطلاحاً Bitmap استفاده می‌شود، نامگذاری آن نیز با همین عنوان انجام شده است.



## مزایای Bitmap Index

مهم‌ترین مزیت‌های Bitmap Index به شرح زیر است:

- مناسب برای ستون‌های با Cardinality کم
- پشتیبانی از عملگرهای منطقی (شامل AND، OR و NOT)
- ایده‌آل برای کوئری‌های مبتنی بر چند شرط یا ترکیب‌های منطقی
- استفاده کارآمد از حافظه و فشردگی بالای آن‌ها
- امکان استفاده در محیط‌های انبارسازی داده‌ها

## معایب Bitmap Index

معایب Bitmap Index عبارتند از:

- انتخاب نامناسب برای ایندکس گذاری ستون‌های دارای کاردینالیته بالا
- عملکرد کند در عملیات‌های نوشتن (درج، به‌روزرسانی و حذف)
- نیاز به نگهداری دائم (شامل ساخت مجدد یا بازسازی Bitmap Index)
- کاربرد محدود به نوع‌های داده و کوئری‌های خاص



## Full-Text Index

Full-Text Index یک تکنیک کاربردی در DBMS ها است که به شما امکان جستجو و استخراج کارآمد داده‌های متنی از درون یک یا چندین داکيومنت را می‌دهد. به واسطه این ایندکس گذاری در SQL Server ، می‌توانید جستجوی مبتنی بر متن پیچیده را روی متون بدون ساختار یا نیمه ساختاریافته، همچون مقاله، کتاب، گزارش و ایمیل، اجرا کنید. ایندکس گذاری تمام‌متنی یا همان Full-Text، به‌طور خاص برای مواقعی مناسب است که در اپلیکیشن، به جستجوی متنی زبان طبیعی (Natural Language) نیاز دارید.

### مزایای Full-Text Index

موارد زیر، مزیت‌های Full-Text Index محسوب می‌شوند:

- پشتیبانی از Partial Matching
- جستجوی متنی کارآمد و سریع داده‌های بدون ساختار و نیمه‌ساختاریافته
- پشتیبانی از کوئری‌های زبان طبیعی (NLQ)
- کارایی گسترده در حوزه‌های مختلف (شامل پلتفرم‌های تجارت الکترونیک، موتورهای جستجو)
- بهبود کارایی کوئری‌های مبتنی بر متن

### معایب Full-Text Index

برخی از کاستی‌های Full-Text Index به شرح زیر است:

- پیچیدگی مربوط به پیاده‌سازی و نگهداری آن
- ایجاد Fragmentation با افزایش حجم دیتاست
- زمان بر بودن فرآیند اولیه ایندکس گذاری
- مصرف بالای حافظه و CPU

## Hash Index

یک روش ایندکس گذاری در SQL Server ، تکنیک Hash Index است که در آن با به‌کارگیری یک تابع Hash، مقادیر داده‌ها به کلیدهای ایندکس یا محل قرارگیری آن‌ها در یک ساختمان داده (معمولاً آرایه) نگاشت می‌شوند. این نوع ایندکس گذاری ، در درجه اول به‌منظور دسترسی سریع و مستقیم به داده‌ها، به‌ویژه در مواقعی که به جستجو براساس کلید نیاز باشد، مورد استفاده قرار می‌گیرد.

## مزایای Hash Index

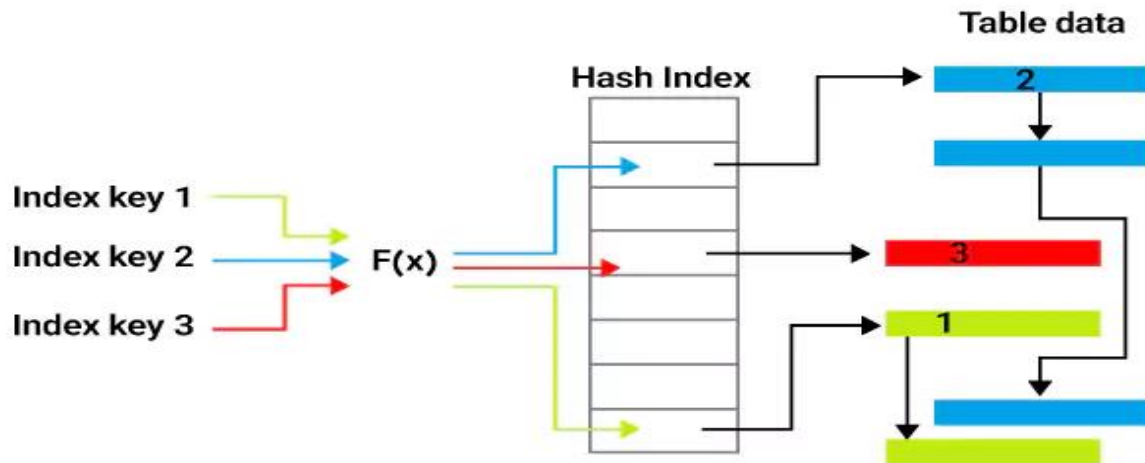
مزیت‌های Hash Index عبارتند از:

- دسترسی مستقیم و سریع به داده‌ها
- کارایی بالا برای جستجوی مورد خاص در یک Dataset گسترده
- تولید مقادیر Hash به صورت قطعی (Deterministic)
- دارای پیچیدگی زمانی ثابت (O1)
- کارایی در مصرف فضای ذخیره‌سازی

## معایب Hash Index

برخی از معایب Hash Index در ادامه لیست شده‌اند:

- امکان وقوع تصادف (Collision) هنگام تولید یک Hash Value یکسان
- نامناسب برای Range Query و Inequality Search
- دشواری‌های مربوط به انتخاب تابع Hash مناسب
- احتمال به هدررفتن فضای ذخیره‌سازی در داده‌های پراکنده (Sparse Data)



## Spatial Index

Spatial Indexing تکنیکی است که در [پایگاه‌های داده](#) و سیستم اطلاعات جغرافیایی (GIS)، با هدف ذخیره‌سازی، مدیریت و کوئری نویسی فضایی یا جغرافیایی داده‌ها به صورت کارآمد مورد استفاده قرار می‌گیرد. Spatial Index کارایی عملیات داده‌های فضایی را بهبود می‌دهد و به دنبال آن، کار کردن با اطلاعات جغرافیایی و اجرای کوئری‌های Spatial را تسهیل می‌بخشد.

## مزایای Spatial Index

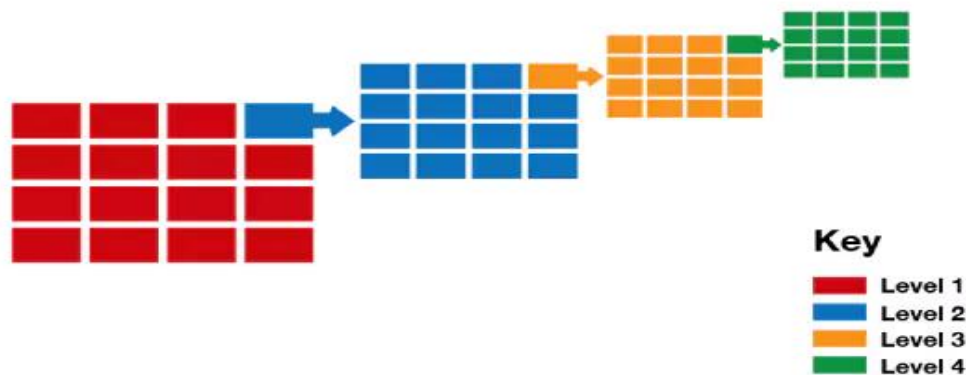
موارد زیر، مزایای Spatial Index به شمار می‌روند:

- کاهش سربار محاسباتی
- کارآمد در رسیدگی به مجموعه داده‌ها گسترده
- پشتیبانی از تجزیه و تحلیل جغرافیایی (Geographic Analysis)
- بهبود کارایی Geocoding

## معایب Spatial Index

هرچند این ایندکس گذاری در SQL Server دارای نکات مثبت مختلفی است، اما چالش‌ها و معایبی دارد که در این قسمت به آن‌ها می‌پردازیم:

- پیچیدگی پیاده‌سازی و نگهداری
- چالش در انتخاب ساختار Spatial Index و Index Parameter مناسب
- عملیات و کوئری‌های پیچیده Spatial
- استاندارد نبودن آن در انواع سیستم Database



## XML Index

XML Index یک ساختار داده دیتابیس محسوب می‌شود که به‌طور خاص، با هدف بهینه‌سازی اموری همچون ذخیره‌سازی و استخراج داده‌های XML درون سیستم پایگاه داده طراحی شده است. این نوع ایندکس گذاری در SQL Server، به‌منظور بهبود کارایی انواع عملیات مرتبط با XML، از جمله کوئری نویسی، جستجو و پردازش داکيومنت‌های XML، کاربردی است.

## مزایای XML Index

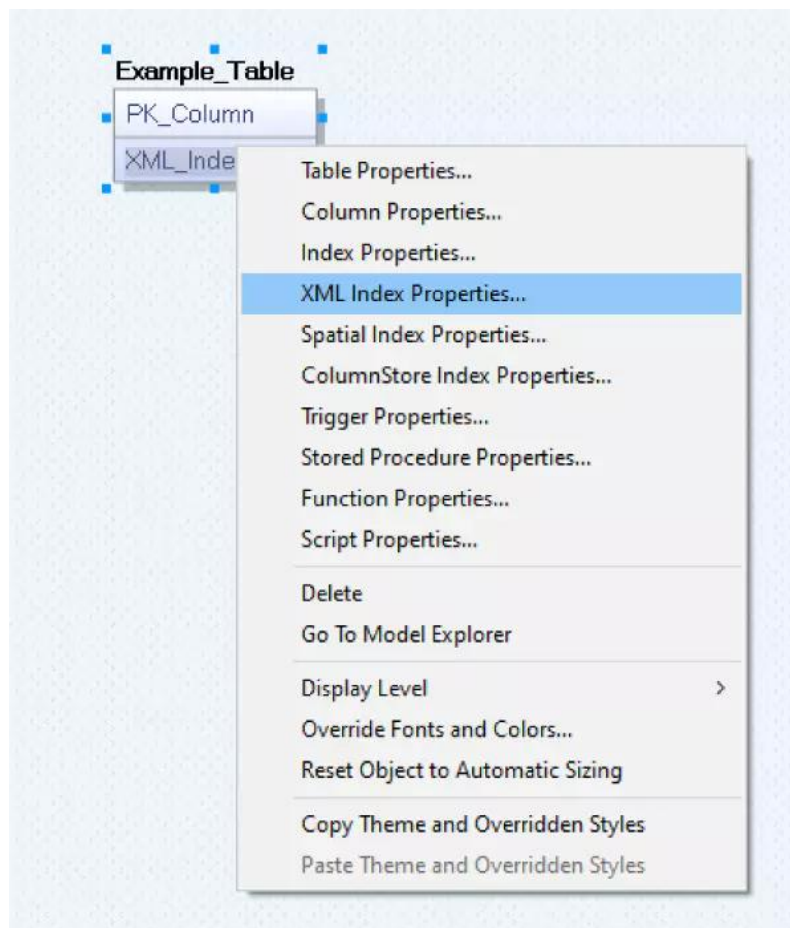
مزیت‌های XML Index عبارتند از:

- بهبود کارایی کوئری‌های مرتبط با XML
- افزایش کارایی در استخراج عناصر (Elements) درون XML
- بهینه‌سازی کوئری‌های XPath

## معایب XML Index

معایب XML Index به شرح زیر است:

- پیچیدگی پیاده‌سازی
- سرشار فضای ذخیره‌سازی و نگهداری
- مصرف بالای منابع ذخیره‌سازی و CPU



## In-Memory OLTP Index

برخلاف سایر روش‌های سنتی ایندکس گذاری در SQL Server که در بخش‌های قبلی به آن‌ها پرداختیم، **In-Memory OLTP Index** نوع متمایزی از Indexing است. به‌طور خاص In-Memory OLTP Index برای جداول بهینه شده از نظر حافظه (Memory-Optimized) در SQL Server طراحی شده است.

در ایندکس گذاری مرسوم و سنتی که شامل ایندکس‌های B Tree و BitMap است، از جداول مبتنی بر دیسک استفاده می‌شود و در درجه اول، به منظور بهینه‌سازی فضای ذخیره‌سازی و استخراج داده‌ها از دیسک، مورد استفاده قرار می‌گیرند. در نقطه مقابل، In-Memory OLTP Index برای جداول Memory-Optimized که تماماً در RAM قرار دارند، به کار می‌رود. این ایندکس گذاری به‌منظور ارائه دسترسی سریع به داده‌ها و بدون متحمل شدن تأخیر مربوط به I/O مورد استفاده قرار می‌گیرد.

### مزایا In-Memory OLTP Index

مزیت‌های In-Memory OLTP Index به شرح زیر است:

- بهبود کارایی کوئری‌ها
- مقیاس‌پذیری بالا
- کاهش Bottleneck دیسک I/O
- پردازش موازی و همچنین پشتیبانی از لود کاری OLAP

### معایب In-Memory OLTP Index

معایب In-Memory OLTP Index عبارتند از:

- موارد استفاده (Use-Cases) های محدود
- نیاز به حجم زیادی از RAM (زیرا جداول و ایندکس‌ها، همگی در آن ذخیره می‌شوند).
- اطمینان از ماندگاری داده‌ها (Data Durability) و همچنین نیاز به استراتژی نظارت و نگهداری مخصوص
- عدم پشتیبانی یا پشتیبانی محدود از برخی Feature ها و **نوع های داده SQL Server**

هرچند ایندکس گذاری در SQL Server یا سایر سیستم‌های مدیریت پایگاه داده، کاربردهای خاص خود را دارد، اما باید یک Trade-off یا اصطلاحاً حد توازن میان ایجاد ایندکس و نگهداری در نظر گرفته شود. با توجه به اینکه معمولاً ایندکس‌ها، فضای ذخیره‌سازی قابل توجهی استفاده می‌کنند و در عملیات ویرایش داده‌ها تأثیرگذار هستند، لازم است ایندکس گذاری در SQL Server با احتیاط و در نظر داشتن نیازمندی‌های اپلیکیشن انجام شود.



## کلام آخر: معرفی انواع ایندکس گذاری در SQL Server

ایندکس گذاری در SQL Server یکی از روش‌های کارآمد برای دسترسی و استخراج سریع داده‌ها و بهینه‌سازی کوئری‌ها تلقی می‌شود. در این مطلب، مفهوم ایندکس گذاری و ایندکس گذاری، انواع Indexing به همراه مزایا و معایب هر یک شرح داده شدند. در عمل، لازم است **مدیر پایگاه داده (DBA)** و توسعه‌دهندگان، ایندکس گذاری را با برنامه‌ریزی و طراحی دقیق عملی کنند و در فرآیند آن، موارد استفاده و الگوهای کوئری نویسی مربوط به دیتابیس‌های سازمان را در نظر بگیرند. در چنین شرایطی، طراحی و نگهداری ایندکس گذاری می‌تواند به شکل قابل توجهی بر روی کارایی کلی و همچنین پاسخگویی اپلیکیشن‌های مبتنی بر پایگاه داده تاثیرگذار باشد.