

با افزایش حجم بانک‌های اطلاعاتی دسترسی سریع به داده‌های مطلوب به یک معضل تبدیل می‌شود. بهمین دلیل نیاز به مکانیزم‌هایی برای بازیابی سریع داده‌ها احساس می‌شود. یکی از این مکانیزم‌ها اندیس گذاری (indexing) است. اندیس گذاری مکانیزمی است که به ما امکان دسترسی مستقیم (direct access) را به داده‌های بانک اطلاعاتی می‌دهد.

عمل اندیس گذاری وظیفه طراح بانک اطلاعاتی است که با توجه به دسترسی‌هایی که در آینده به بانک اطلاعاتی وجود دارد مشخص می‌کند که بر روی چه ستون‌هایی می‌خواهد اندیس داشته باشد. بعنوان مثال با تعیین کلید اصلی اعلام می‌کند که بیشتر دسترسی‌های آینده من بر اساس این کلید اصلی است و بنابراین بانک اطلاعاتی بر روی کلید اصلی اندیس گذاری را انجام می‌دهد. علاوه بر کلید اصلی می‌توان بر روی هر ستون دیگری از جدول نیز اندیس گذاشت که همانطور که گفته شد این مسئله بستگی به تعداد دسترسی آینده ما از طریق آن ستون‌ها دارد.

پس از اندیس گذاری بر روی یک ستون بسته به نوع اندیس فایلی در پایگاه اطلاعاتی ما ایجاد می‌شود که به آن فایل اندیس (index file) گفته می‌شود. این فایل یک فایل مبتنی بر رکورد (record-based) است که هر رکورد آن محتوی زوج کلید جستجو - اشاره گر می‌باشد. کلید جستجو را مقدار ستون مورد نظر و اشاره گر را اشاره گری به رکورد مربوط به آن می‌تواند در نظر گرفت.

توجه داشته باشید که اندیس گذاری و مدیریت اندیس‌ها، همانطور که در این مقاله آموزشی گفته خواهد شد سر بارهایی (از نظر حافظه و پردازش) را بر سیستم تحمیل می‌نمایند. بعنوان مثال با اندیس گذاری بر روی هر ستونی یک فایل اندیس نیز ایجاد می‌شود بنابراین اگر اندیس‌های ما بسیار زیاد باشد حجم زیادی از بانک اطلاعاتی ما را خواهند گرفت. مدیریت و بروز نگهداری فایل‌های اندیس نیز خود مسئله ایست که سر بار پردازشی را بدنبال دارد. بنابراین توصیه می‌شود در هنگام اندیس گذاری حتما بررسی‌ها و تحلیل‌های لازم را انجام دهید و تنها بر روی ستون‌هایی اندیس بگذرید که در آینده بیشتر دسترسی‌های شما از طریق آن ستون‌ها خواهد بود.

عموما در بانک‌های اطلاعاتی دو نوع اندیس می‌تواند بکار گیری شود که عبارتند از :

اندیس‌های مرتب (ordered indices) : در این نوع کلیدهای جستجو (search-key) بصورت مرتب نگداری می‌شوند.

اندیس‌های هش (Hash indices) : در این نوع از اندیس‌ها کلیدهای جستجو در فایل اندیس مرتب نیستند. بلکه توسط یک تابع هش (hash function) توزیع می‌شوند.

در این مقاله قصد داریم به اندیس‌های مرتب بپردازیم و بخشی از مفاهیم مطرح در این باره را پوشش دهیم.

اندیس‌های متراکم (dense index) :

اولین و ساده‌ترین نوع از اندیس‌های مرتب **اندیس‌های متراکم (dense)** هستند. در این نوع از اندیس‌ها وقتی بر روی ستونی می‌خواهیم عمل اندیس گذاری را انجام دهیم می‌بایست به ازای هر کلید - جست و جو (search-key) غیر تکراری در ستون مورد نظر، یک رکورد در فایل اندیس مربوط به آن ستون اضافه کنیم. برای روشن شدن بیشتر موضوع به شکل زیر توجه کنید.

Brighton		A-217	Brighton	750	
Downtown		A-101	Downtown	500	
Mianus		A-110	Downtown	600	
Perryridge		A-215	Mianus	700	
Redwood		A-102	Perryridge	400	
Round Hill		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

شکل 1 - اندیس متراکم (sparse index)

همانطور که در تصویری مشاهده می‌کنید بر روی ستون دوم از این جدول (جدول سمت راست)، اندیس متراکم (dense) گذاشته شده است. بر همین اساس به ازای هر کدام از اسامی خیابان‌ها یک رکورد در فایل اندیس (جدول سمت چپ) آورده شده است. در فایل اندیس می‌بینید که در کنار کلید جستجو یک اشاره گر نیز به جدول اصلی وجود دارد که در هنگام دسترسی مستقیم (direct access) از این اشاره گر استفاده خواهد شد. دقت کنید که کلیدهای جستجو در فایل اندیس بصورت مرتب نگهداری شده اند که نکته ای کلیدی در اندیس‌های مرتب می‌باشد.

مرتب بودن فایل اندیس موجب می‌شود که ما در هنگام جستجوی کلید مورد نظرمان در جدول اندیس بتوانیم از روش‌های جستجویی نظری جست و جوی دو دویی استفاده کنیم و در نتیجه سریع‌تر کلید مورد نظر را پیدا کنیم. این مسئله باعث بهبود کارایی می‌شود. بعنوان مثال فرض کنید در فایل اندیس یک میلیون رکورد داریم. در این صورت برای یافتن کلید مورد نظرمان در جدول اندیس بروش جست و جوی دو دویی تنها کافی است 20 عمل مقایسه انجام دهیم. بنابراین می‌بینید که مرتب نگهداشتن جدول اندیس چقدر در سرعت بازیابی، تاثیر دارد.

نکته مهمی که در اندیس‌های متراکم باید به آن دقت شود اینست که ما به ازای کلیدهای جستجوی غیر تکراری یک رکورد در جدول اندیس نگهداری می‌کنیم. برای مثال در شکل بالا در ستون مورد نظر ما دو رکورد برای Downtown و سه رکورد برای Perryridge وجود دارد. این در حالی است که در فایل اندیس فقط یک Downtown و Perryridge داریم.

در اندیس‌های متراکم ما امکان دو نوع دسترسی را داریم :

دسترسی مستقیم (direct access)

دسترسی ترتیبی (sequential access)

دسترسی مستقیم :

توجه داشته باشید که در هنگام کار با یک جدول، فایل‌های اندیس آن به حافظه اصلی آورده می‌شوند (البته ممکن است که بخشی از فایل‌های اندیس به حافظه اصلی نیایند). این در حالی است که فایل اصلی جدول در حافظه جانبی قرار دارد. بنابراین در هنگام

بازیابی یک رکورد از برای یافتن محل آن رکورد نیازی به مراجعه زیاد به حافظه جانبی نیست. بلکه در حافظه اصلی بسرعت با یک عمل جستجو اشاره گر مربوط به رکورد مورد نظر در حافظه جانبی پیدا شده و مستقیماً به آدرس همان رکورد می‌رویم و آن را می‌خوانیم. به این دسترسی، دسترسی مستقیم (direct access) می‌گوییم.

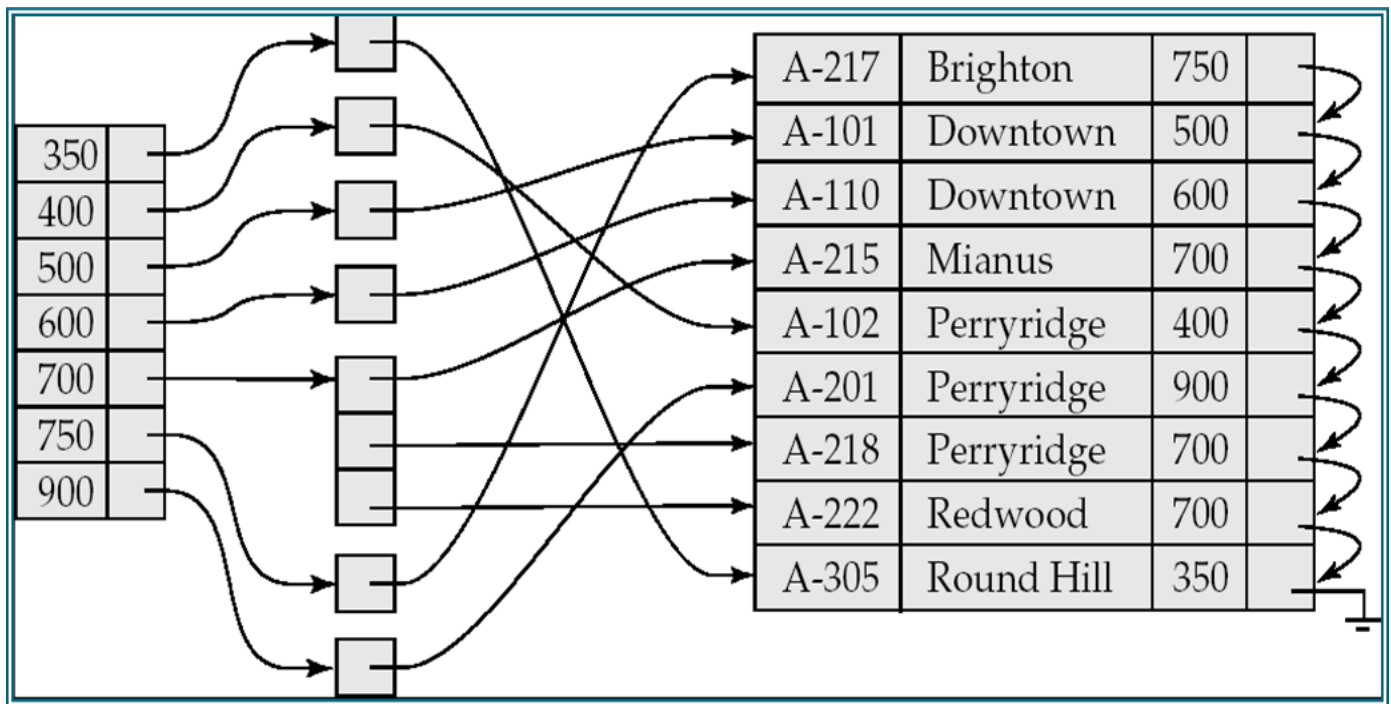
دسترس‌ی ترتیبی :

در برخی از روش‌های اندیس گذاری علاوه بر دسترسی مستقیم امکان دسترسی بصورت ترتیبی نیز وجود دارد. در دسترسی ترتیبی این امکان وجود دارد که از یک رکورد خاص در جدول اصلی بتوانیم رکوردهای بعد از آن را به ترتیبی منطقی پیمایش کنیم. برای روشن‌تر شدن موضوع به شکل شماره 1 توجه کنید. در انتهای هر رکورد اشاره گری به رکورد منطقی بعدی مشاهده می‌کنید. این اشاره گرها امکان پیمایش و دسترسی ترتیبی را به ما می‌دهند. بعنوان مثال فرض کنید قصد داریم تمامی رکوردهای حاوی کلید Perryridge را بازیابی نماییم. از آنجایی که در جدول اندیس تنها برای یکی از رکوردهای حاوی این کلید اندیس داریم، برای بازیابی باقی رکوردها چه باید کرد؟ در چنین شرایطی ابتدا با دسترسی مستقیم اولین رکورد حاوی Perryridge را پیدا کرده و آن را بازیابی می‌کنیم. سپس از طریق اشاره گر انتهای آن رکورد، می‌توان به رکورد بعدی آن دست یافت و به همین ترتیب می‌توان یک به یک به رکوردهای دیگر دسترسی ترتیبی پیدا نمود.

دقت کنید که رکوردهای جدول ما بصورت فیزیکی مرتب نیستند. اما اشاره گرهای انتهای رکوردها طوری مقدار دهی شده اند که بتوان آنها را بصورت مرتب شده پیمایش نمود.

اندیس اولیه (primary index) و اندیس ثانویه (secondary index) :

بر روی ستون‌های یک جدول می‌توان چندین اندیس را تعریف نمود. اولین اندیسی که بر روی یک ستون از یک جدول گذاشته می‌شود اندیس اولیه (primary index) نامیده می‌شود. عموماً این اندیس به کلید اصلی نسبت داده می‌شود، چراکه اولین اندیسی است که بر روی جدول زده می‌شود. توجه داشته باشید که رکوردهای جدول اصلی بر اساس کلیدهای جستجوی اندیس اولیه بصورت منطقی (با استفاده اشاره گرهای انتهای رکورد که توضیح داده شد) مرتب هستند. بنابراین امکان دسترسی بصورت ترتیبی وجود دارد. وقتی پس از اندیس اولیه اقدام به اندیس گذاری‌های دیگری می‌کنیم، اندیس‌های ثانویه را ایجاد می‌کنیم که اندکی با اندیس‌های اولیه متفاوت می‌باشند. در اندیس‌های ثانویه دیگر امکان پیمایش و دسترسی ترتیبی وجود ندارد چراکه اشاره گرهای انتهای رکوردها بر اساس اندیس اصلی (اولیه) مرتب شده اند. بنابراین ما در اندیس‌های ثانویه تنها دسترسی مستقیم خواهیم داشت. شکر زیر نمونه ای از یک اندیس ثانویه را نشان می‌دهد.



شکل 2 - اندیس ثانویه

همانطور که مشاهده می‌کنید علاوه بر اندیس اصلی (بر روی ستون 2) بر روی سومین ستون این جدول اندیس ثانویه متراکم زده شده است. دقت کنید که هر اشاره گر از جدول اندیس به یک باکت (bucket) اشاره دارد. در هر باکت اشاره گر هایی وجود دارد که به رکورد هایی از جدول اصلی اشاره می‌کنند. فلسفه وجود باکت‌ها اینست که در اندیس‌های ثانویه امکان دسترسی ترتیبی وجود ندارد. بنابراین برای مقادیری تکراری در جدول (مثلا عدد 700) نمی‌توان از اشاره گرهای انتهایی رکوردها استفاده نمود. در چنین شرایطی در باکت‌ها اشاره گر مربوط به تمامی رکوردهای حاوی مقادیر تکراری یک کلید را نگهداری می‌کنیم تا بتوان به آنها دسترسی مستقیم داشت. همانطور که مشاهده می‌کنید برای بازیابی رکوردهای حاوی مقدار 700 ابتدا از جدول اندیس (که مرتب است) باکت مربوطه را پیدا کرده و سپس از طریق اشاره گرهای موجود در این باکت به رکوردهای حاوی مقدار 700 دستیابی پیدا می‌کنیم.

اندیس‌های تنک (sparse index) :

در این نوع از اندیس‌ها بر خلاف اندیس‌های متراکم، تنها به ازای برخی از کلیدهای جستجو در جدول اندیس اشاره گر نگهداری می‌کنیم. بهمین دلیل فایل اندیس ما کوچکتر خواهد بود (نسبت به اندیس متراکم). در مورد اندیس‌های تنک نیز امکان دسترسی ترتیبی وجود دارد. در شکل زیر نمونه از اندیس تنک (sparse) را مشاهده می‌کنید.

Brighton		A-217	Brighton	750	
Mianus		A-101	Downtown	500	
Redwood		A-110	Downtown	600	
		A-215	Mianus	700	
		A-102	Perryridge	400	
		A-201	Perryridge	900	
		A-218	Perryridge	700	
		A-222	Redwood	700	
		A-305	Round Hill	350	

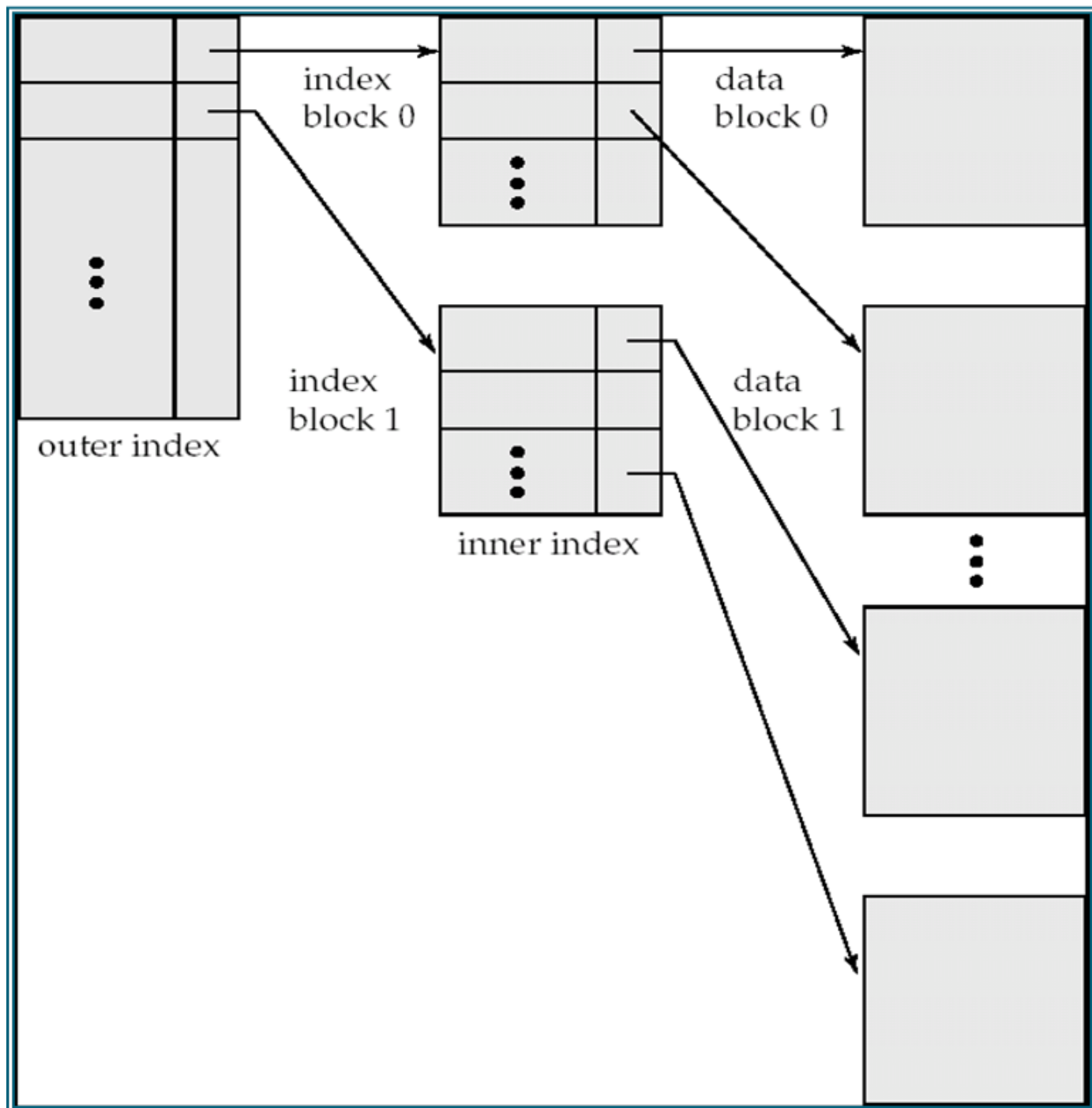
شکل 3 - اندیس تنک (sparse index)

همانند شکل 1، در این شکل نیز اندیس اولیه بر روی ستون دوم زده شده است. اما این بار از اندیس تنک استفاده گردیده است. مشاهده می‌کنید که از میان مقادیر مختلف این ستون تنها برای سه کلید Brighton، Perryridge و Redwood در جدول اندیس رکورد درج شده است. بنابراین برای دست یابی به کلیدهای دیگر باید ابتدا محل تقریبی آن را با جستجو بر روی جدول اندیس پیدا نمود و سپس از طریق پیمایش ترتیبی به رکورد مورد نظر دست یافت. بعنوان مثال برای بازیابی رکورد حاوی مقدار Mianus ابتدا در جدول اندیس کلیدی که از Mianus کوچکتر باشد (یعنی Brighton) را پیدا می‌کنیم. سپس به رکورد حاوی Brighton می‌رویم و از آنجا با استفاده از اشاره گرهای انتهایی رکوردها به سمت رکورد حاوی Mianus حرکت می‌کنیم تا به آن برسیم.

نکته بسیار مهمی که در مورد اندیس‌های تنک مطرح می‌شود اینست که سیستم چگونه باید تشخیص دهد که کدام کلیدها را در جدول اندیس نگهداری کند. این تصمیم به مفهوم بلاک‌های حافظه و اندازه آنها باز می‌گردد. می‌دانیم که واحد خواندن اطلاعات از حافظه بر اساس بلاک‌ها می‌باشد. این بدان معنی است که در هنگام خواندن رکوردهای جداول بانک اطلاعاتی، عمل خواندن بصورت بلاکی انجام می‌شود. هنگامی که بر روی یک جدول می‌خواهیم اندیس تنک بنیم ابتدا باید ببینیم این جدول چند بلاک از حافظه را اشغال کرده است. سپس رکوردهای اول هر بلاک را پیدا کرده و به ازای هر بلاک آدرس و کلید جستجوی رکورد اول آن را در جدول اندیس نگهداری کنیم. بدین ترتیب ما به ازای هر بلاک از جدول یک رکورد در فایل اندیس خواهیم داشت و با تخصیص بلاک‌های جدید به آن، طبیعی است که اندیس‌های جدید نیز در فایل اندیس ذخیره خواهند شد.

اندیس‌های چند سطحی (multi-level index)

در دنیایی واقعی معمولا تعداد رکوردهای جداول مورد استفاده بسیار بزرگ است و این اندازه دائما در حال زیاد شدن می‌باشد. افزایش اندازه جداول باعث می‌شود که اندازه فایل‌های اندیس نیز رفته رفته زیاد شود. گفتیم برای کارایی هرچه بیشتر باید جدول اندیس مورد استفاده به حافظه اصلی آورده شود تا تعداد دسترسی‌های ما به حافظه جانبی تا حد امکان کاهش یابد. اما اگر اندازه فایل اندیس ما بسیار بزرگ باشد ممکن است حجم زیادی از حافظه اصلی را بگیرد یا اینکه در حافظه اصلی فضای کافی برای آن وجود نداشته باشد. در چنین شرایطی از اندیس‌های چند سطحی استفاده می‌شود. به بیان دیگر بر روی جدول اندیس نیز اندیس زده می‌شود. تعداد سطوح اندیس ما بستگی به اندازه جدول اصلی دارد و هر چه این اندازه بزرگ‌تر شود، ممکن است باعث افزایش تعداد سطوح اندیس شود. در شکل زیر ساختار یک اندیس دو سطحی را مشاهده می‌کنید.



نکته مهم در مورد اندیس‌های چند سطحی اینست که اندیس‌های سطوح خارجی (outer index) از نوع تنک هستند. این مسئله به این دلیل است که اندازه اندیس‌ها کوچک‌تر شود. چراکه اگر اندیس خارجی از نوع متراکم باشد به این معناست که به ازای هر رکورد غیر تکراری باید یک رکورد در فایل اندیس نیز آورده شود و این مسئله باعث بزرگ شدن اندیس می‌شود. بهمین دلیل سطوح خارجی را در اندیس‌های چند سطحی از نوع تنک می‌گیرند. تنها آخرین سطحی که مستقیماً به جدول اصلی اشاره می‌کند از نوع متراکم است. به این سطح از اندیس، اندیس داخلی (inner index) گفته می‌شود.

بروز نگهداشتن اندیس‌ها :

با انجام عملیات درج و حذف بروی جداول، جداول اندیس مربوطه نیز باید بروز رسانی شوند. در این بخش قصد داریم به نحوه بروز رسانی جداول اندیس در زمان حذف و درج رکورد بپردازیم.

بروز رسانی در زمان حذف :

اندیس متراکم :

هنگامی که رکوردی از جدول اصلی حذف می‌شود، در صورتی که بر روی ستون‌های آن اندیس‌های متراکم داشته باشیم، پس از حذف رکورد اصلی باید ابتدا کلید جستجوی ستون مربوط را در جدول اندیس پیدا کنیم. در صورتی که از این کلید تنها یک مقدار در جدول اصلی وجود داشته باشد، اندیس آن را از فایل اندیس حذف کرده و اشاره گرهای انتهای رکوردها را بروز رسانی می‌کنیم. اما اگر از کلید مورد نظر چندین مورد وجود داشته باشد نباید رکورد مورد نظر در جدول اندیس پاک شود. بلکه تنها ممکن است نیاز به ویرایش اشاره گر اندیس باشد. ویرایش در زمانی رخ می‌دهد که اشاره گر جدول اندیس مستقیماً به رکوردی اشاره کند که حذف شده باشد، در این صورت باید اشاره گر اندیس را ویرایش نمود تا به رکورد بعدی اشاره نماید.

اندیس تنک :

همانند روش قبل ابتدا رکورد اصلی را از جدول حذف می‌کنیم. سپس در فایل اندیس بدنبال کلید جستجوی مربوط به رکورد حذف شده می‌گردیم. در صورتی که کلید مورد نظر در جدول اندیس پیدا شد کلید جستجوی رکورد بعدی در جدول اصلی را جایگزین آن می‌کنیم. چنانچه کلید مربوط به رکورد بعدی در جدول اندیس وجود داشته باشد نیازی به جایگزینی نیست و باید فقط عمل حذف اندیس را انجام داد.

اگر کلید مورد جستجو در جدول اندیس وجود نداشته باشد نیاز به انجام هیچ عملی نیست. در پایان باید اشاره گرهای انتهای رکوردها را ویرایش نمود تا ترتیب منطقی برای پیمایش ترتیبی حفظ شود.

بروز رسانی در زمان درج:

اندیس متراکم:

در هنگام درج یک رکورد جدید، ابتدا باید کلید موجود در رکورد جدید را در جدول اندیس جستجو نمود. در صورتی که کلید مورد نظر در جدول اندیس یافت نشد، باید رکوردی جدیدی در فایل اندیس درج کرد و اشاره گر آن طوری مقدار دهی نمود تا به رکورد جدید اشاره نماید. اگر کلید مورد نظر در جدول اندیس وجود داشته باشد دیگر نیازی به بروز رسانی اندیس‌ها نیست و تنها کافی است اشاره گرهای انتهای رکوردها بروز رسانی شوند.

اندیس تنک :

در مورد اندیس‌های تنک کمی پیچیدگی وجود دارد. در صورتی که رکورد جدید باعث تخصیص بلاک (block) جدیدی از حافظه به جدول شود، باید به ازای آن بلاک یک اندیس در جدول اندیس‌ها ایجاد شود و آدر آن بلاک را (که در واقع آدرس رکورد جدید نیز می‌شود) در اشاره گر اندیس قرار داد. اما درغیز این صورت (در صورتی که رکورد در بلاک‌های موجود ذخیره شود) نیازی به بروز رسانی جدول اندیس‌ها وجود ندارد.

نوع دیگری از اندیس‌های مرتب نیز وجود دارد که اندیس های B-Tree هستند که در سیستم‌های اطلاعاتی دنیای واقعی بیشتر از آنها استفاده می‌شود. به امید خدا در مطالب بعدی این اندیس‌ها را نیز مورد بررسی قرار خواهیم داد.

موفق و پیروز باشید.

نظرات خوانندگان

نویسنده: مجید_فاضلی نسب
تاریخ: ۱۳:۲۹ ۱۳۹۲/۰۸/۰۳

سلام و درود.
فرق Indices و Index چیست ؟

نویسنده: حامد خسروجردي
تاریخ: ۹:۵۱ ۱۳۹۲/۰۸/۰۴

سلام. indices همون جمع index هستش.

نویسنده: وحید نصیری
تاریخ: ۱۳:۳۸ ۱۳۹۲/۰۸/۰۴

Indices مستقیماً از زبان لاتین گرفته شده است. آمریکایی‌ها بیشتر indexes را بکار می‌برند بجای Indices. هر دو هم صحیح هستند.