



ایندکس در اوراکل



ایندکس

ایندکس با استفاده از ذخیره آدرس فیزیکی رکوردها باعث می شود که دسترسی سریع تر به رکوردهای جدول داشته باشیم و به جای اینکه تمامی رکوردهای یک جدول را واکشی کند این شی فقط رکوردهایی را که مربوط به پرس و جو می باشند واکشی می کند و باعث کاهش I/O و افزایش سرعت واکشی اطلاعات خواهد شد.

روش های دستیابی به داده های جدول

- 1- Full table scan
- 2- Index

ستون های جدول

- 1- rowid ← آدرس 18 کاراکتری است که به محل آدرس فیزیکی محل ذخیره سازی رکورد اشاره می کند
- 2- ستون های جدول

ساختار ایندکس

ایندکس درختی متشکل از برگهایی است که هر برگ شامل دو مقدار key و value می باشد

Rowid = Object id + Data file id + Data block + Row number

Value = مقدار ستونی که بر روی آن ایندکس تعریف شده است

موارد استفاده از ایندکس

- 1- در SQL هنگامی که در where Clause ها می خواهیم فیلتر انجام دهیم
- 2- در مرتب سازی ها order by
- 3- اگر در دستور select یکی از عبارت های order by ، group by ، distinct و union باشد از ایندکس استفاده خواهد شد.
- 4- در دستور Join وجود ایندکس در ستون هایی که در شرط join شرکت می کنند باعث بهبود کارایی خواهد شد

نکته: هر دستور DML که باعث تغییر در داده های جدول می شود باعث به روز رسانی ایندکس می شود

در چه مواقعی بهتر است از ایندکس استفاده کنیم؟

- ✓ اگر پرس و جو ها نوشته شده 5٪ از رگوردهای جدول را باز می گرداند می توان بر روی ستونی که در where استفاده شده است ایندکس تعریف کرد
- ✓ برای ستون هایی که بصورت مکرر در where استفاده می شوند می توان ایندکس تعریف کرد
- ✓ پیشنهاد می شود که برای جداول PK (کلید اصلی) تعریف کنید ، تعریف PK در یک جدول باعث می شود که بصورت اتوماتیک یک unique index بر روی جدول تعریف شود
- ✓ پیشنهاد می شود برای ستون های FK در جداول ایندکس تعریف گردد، تعریف ایندکس بر روی ستون های FK باعث بهبود عملکرد دستور join خواهد شد
- ✓ برای بهینه سازی عملیاتی مانند order by ، group by ، distinct و union می توان از ایندکس استفاده کرد
- ✓ معمولا در سیستم های انبار داده (data warehouse) ها از ایندکس های بیشتر و در سیستم های OLTP به دلیل آنکه داده ها مرتب به روز رسانی می شود از Index های کمتری استفاده می شود

مواقعی که بهتر است از ایندکس استفاده نشود؟

- ✓ ستون هایی که دارای مقادیر null و تکراری زیاد هستند.
- ✓ ستون هایی که تنوع داده ای کمی دارند مانند زن و مرد (در این حالت بهتر است که از ایندکس bitmap استفاده شود)
- ✓ ستون هایی که نوع داده ای آنها کاراکتری و یا Datetime است
- ✓ جداول کوچک (جداولی که تعداد رکوردهای کمی دارند)
- ✓ جداولی که حجم تغییرات داده در آنها زیاد است (داده ها در بازه زمانی کم دائما در حال اضافه ، اصلاح یا حذف هستند)

انواع ایندکس

Btree -1

- ایندکس Btree یک درخت متوازن سه یا چهار سطحی می باشد که شامل برگ هایی است که هر برگ شامل دو مولفه key و value است
- ✓ اولین کاربرد ایندکس در where clause ها است و دومین کاربرد ایندکس در مرتب سازی است، هنگامی که در یک پرس و جو از order by ، group by ، distinct و union استفاده می کنید اوراکل می تواند از ایندکس استفاده کند
- ✓ استفاده از عملگرهای منطقی and و or در دستورات SQL باعث می شود که از ایندکس btree استفاده نشود

✓ در درخت ایندکس از نوع btree مقادیر null ذخیره نمی‌شوند بنابراین در پرس و جوهای که در آنها از whrer column_name is null استفاده شده است کل جداول مورد پیمایش قرار داده می‌شود.

✓ اوراق هر زمانی که بتواند از ایندکس استفاده می‌کند، در واقع هر زمانی که بتواند با استفاده از ایندکس عملکرد بهتری ارائه دهد از ایندکس استفاده خواهد شد.

✓ کاربرد دیگر ایندکس در Join ها است، معمولا بر روی ستون های FK جداول ایندکس تعریف می‌شود.

```
create index <index_name>
on <table_name> (column_name)
tablespace <tablespace_name>
local
parallel degree (DOP);
```

2- Bitmap

برای محیط های انبار داده (data warehouse) استفاده از ایندکس bitmap مناسب نیست چون بیشتر پرس و جوهای که در سیستم های انبار داده به پایگاه داده ارسال می‌گردد با استفاده از عملگر منطقی or یا and است، در این مواقع بهتر است به جای تعریف Index Btree از ایندکس bitmap استفاده شود

✓ در ایندکس bitmap مقادیر null نیز به عنوان یک مقدار مجزا در نظر گرفته می‌شود

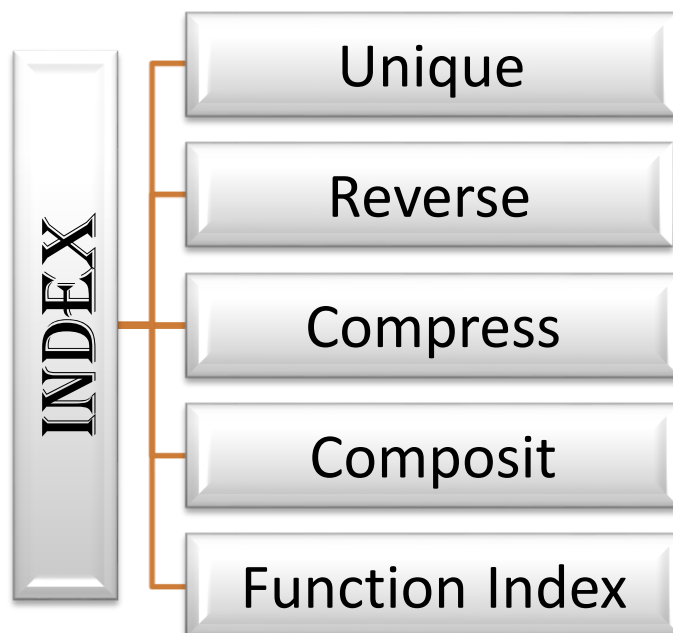
✓ در دستورات SQL که از عملگرهای منطقی or و and استفاده شده است وجود ایندکس bitmap باعث بهبود کارایی خواهد شد

در چه مواقعی که بهتر است از ایندکس bitmap استفاده شود

- ✓ تنوع مقادیر (Cardinality) مقادیر کم است
- ✓ تعداد رکوردها زیاد است
- ✓ در دستورات SQL از عملیات منطقی (and و or) استفاده شده است

```
create bitmap index on <table_name> (column_name)
tablespace <tablespace_name>
local
parallel degree (DOP);
```

انواع Index از لحاظ نوع ایجاد



Unique

اوراکل در زمان ایجاد کلید اصلی (primary key) به صورت اتوماتیک یک unique index تعریف می کند.

```
create unique Index <index_name>
On <table_name> (column_name);
```

Reverse

در مواقعی کاربرد دارد که قصد داریم برای فیلدی که مقدار آن بصورت ترتیبی اضافه می شود Index تعریف کنیم، با استفاده از این گزینه محل قرار گرفتن برگ مربوط به رکورد را در درخت ایندکس پخش می کنیم.

Compress

مقادیر تکراری در ستون Index فقط یکبار ذخیره می شوند برای ایندکس های غیر یکتا (non-unique) که ممکن است مقادیر تکراری زیادی داشته باشند کاربرد دارد.

```
create index <index_name>
on <table_name> (column_name)
tablespace <tablespace_name>
compress level <level_number>;
```

level_number: سطحی که قرار است فشرده سازی براساس آن انجام شود

Composit

در مواقعی است که برای بیش از یک ستون ایندکس ساخته می شود، ترتیب قرار گرفتن ستون ها در ایندکس مهم است و فیلدهایی باید در ابتدا قرار گیرد که به تنهایی بیشترین احتمال استفاده در where clause ها را دارند

Function Index

می توانید با استفاده از یک تابع (Function) بر روی یک فیلد ایندکس تعریف کنید.

Rebuild index

با انجام کارهای DML ایندکس به صورت اتوماتیک به روز رسانی می شود مخصوصاً پس از انجام عملیات حذف (delete)، در صورت نیاز می توان بصورت دستی نیز ایندکس را rebuild کرد

```
alter index <index_name>
rebuild online nologging;
```

online: هنگام rebuild یک جدول ممکن است انجام عملیات DML یا DDL فراهم نباشد با استفاده از گزینه online در زمان rebuild کردن ایندکس امکان انجام دستورات DML در اختیار کاربران قرار خواهد گرفت، انجام دستورات DDL تا پایان rebuild ایندکس امکان پذیر نیست.

nologging: این گزینه باعث می شود کمترین log در online redo log ها ثبت شود، استفاده از این گزینه باعث بهبود سرعت rebuild ایندکس خواهد شد.

انتقال Index به Tablespace جدید

```
alter index <index_name>
rebuild tablespace <new_tablespace_name>
```

ایندکس های unusable

انجام عملیات مانند انتقال داده ها و غیره ممکن است وضعیت یک ایندکس را به حالت unusable تغییر دهد که در این حالت اوراکل از ایندکس فوق استفاده نمی کند، در نسخه های قبلی اوراکل اگر در هنگام اجرای SQL با ایندکس unusable مواجه می شد دستور SQL خطا می داد ولی در نسخه های جدید اوراکل بصورت پیش فرض هنگام مواجه شدن با ایندکس unusable خطا نمایش داده نمی شود

اجرای دستور skip_unusable_Index=false باعث می شود که همانند نسخه های قبلی هنگام مواجه شدن با ایندکس unusable خطا ایجاد شود.

```
select * from dba_indexes where status = 'unusable';
```

تغییر نام Index

از دستور ذیل برای تغییر نام ایندکس استفاده می شود

```
alter index < old_index_name> Rename to <new_index_name>;
```

حذف Index

از دستور ذیل برای حذف ایندکس استفاده می شود

```
drop Index <Index_name>;
```

ایندکس های پارتیشن شده

دو نوع ایندکس برای جداول پارتیشن شده وجود دارد

Global -1

با استفاده از این روش می توان یک ایندکس را به روشی متفاوت از پارتیشن های جدول پارتیشن بندی کرد یا حتی ایندکس را پارتیشن بندی نکرد.

🚩 **نکته:** هنگام اجرای دستورات DDL در جدول پارتیشن بندی شده ایندکس های Global ،

unusable شده و باید مجدداً آن را rebuild کرد برای جلوگیری از این کار می توان از دستور

update global index استفاده کرد.

```
alter table <table_name>
```

```
drop partition <partition_name>
```

```
update global index;
```

Local -2

در هر پارتیشن تعریف می شوند و دارای مزایای ذیل می باشد

✓ با تغییرات داده ای توسط دستورات DML فقط ایندکس های پارتیشن (هایی) که در حال تغییر

است rebuild می شود

✓ اوراق می تواند از Plan های بهتری جهت اجرای SQL ها استفاده کند

✓ با استفاده از ایندکس های local می توان ایندکس های یک پارتیشن خاص را rebuild کرد.

Data Dictionary های مربوط به ایندکس‌ها

user_ind_columns
user_indexes
user_ind_statistics
user_ind_partitions