

Oracle 10gR2 実行計画の確認

[Oracle]

SQL Plus

http://otndnld.oracle.co.jp/document/products/oracle10_g/102/doc_cd/server.102/B19207-02/optimops.htm#967766

■ 実行例

```
SQL> set lin 1000
SQL> set autotrace trace explain
SQL> select * from oratest.google_content_report
2 left outer join (
3 select proc_time, count(proc_time) ac from oratest.httpd_access_log
4 group by proc_time
5 ) on rep_date = proc_time
6 order by rep_date
7 /
```

実行計画

Plan hash value: 542296894

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		581	24402	1370 (3)	00:00:17
1	SORT ORDER BY		581	24402	1370 (3)	00:00:17
* 2	HASH JOIN RIGHT OUTER		581	24402	1369 (3)	00:00:17
3	VIEW		31	651	1366 (3)	00:00:17
4	HASH GROUP BY		31	248	1366 (3)	00:00:17
5	TABLE ACCESS FULL	HTTPD_ACCESS_LOG	155K	1216K	1347 (1)	00:00:17
6	TABLE ACCESS FULL	GOOGLE_CONTENT_REPORT	581	12201	3 (0)	00:00:01

Predicate Information (identified by operation id):

2 - access("REP_DATE"="PROC_TIME"(+))

- ・ [iSQL Plus から Explain プランを確認](#)

アクセス・パス

http://otndnld.oracle.co.jp/document/products/oracle10_g/102/doc_cd/server.102/B19207-02/optimops.htm#967766

- ・ データベースからデータを取り出す経路
- ・ 一般に、表の行の小さいサブセットを取得する文には索引アクセス・パスを指定する
- ・ 表の大きい部分にアクセスするときは全体スキャンの方が効率がよくなる

アクセス・パス	内容
全表スキャン	表にあるすべての行の読取り、選択基準を満たしていない行のフィルタが実行される
ROWID スキャン	行が含まれているデータファイルおよびデータ・ブロックと該当するブロック内の位置を指定。行の ROWID の特定による行の位置特定は、単一行を取得する最も高速な方法。
索引スキャン	索引付きの列の値を使用して索引が検索され、行が取得される。

クラスタ・アクセス	索引クラスタに格納された表から、クラスタ・キー値の等しい行すべてを取得するときに使用。同一のクラスタ・キー値を持つすべての行が同じデータ・ブロックに格納
ハッシュ・アクセス	ハッシュ値に基づいて行をハッシュ・クラスタに配置するために使用。同一のハッシュ値を持つすべての行が同じデータ・ブロックに格納
サンプル表スキャン	複合 SELECT 文からデータのランダムなサンプルを取り出す。文の FROM 句に SAMPLE 句または SAMPLE BLOCK 句が含まれているときに使用。

サンプル表スキャン

```
SELECT * FROM employees SAMPLE BLOCK (1);
```

■索引スキャンのタイプ

ブロックの I/O（行ではなく）の想定

- ・ Oracle は、ブロック単位で I/O を実行
- ・ 全表スキャンを使用するかどうかのオプティマイザの決定は、行でなくアクセスされるブロックのパーセンテージに影響される
- ・ これを索引クラスタ化係数という
- ・ 低いクラスタ化係数は、個々の行が表の少数のブロック内に集中されることを示す。
- ・ 高いクラスタ化係数は、個々の行が表の複数のブロックによりランダムに分散されることを示す。
- ・ 高いクラスタ化係数の場合はレンジ・スキャンを使用して ROWID で行をフェッチするので、よりコストがかかる。

索引一意スキャン

- ・ 1 つの ROWID しか戻しません。文が単一行にしかアクセスしないことが保証されている UNIQUE 制約または PRIMARY KEY 制約が存在する場合

索引レンジ・スキャン

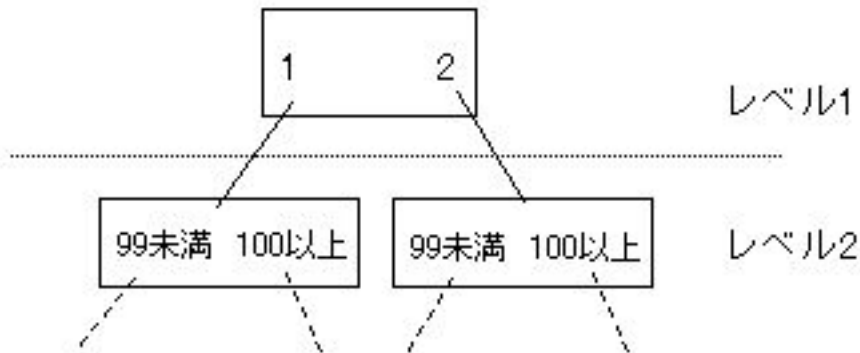
- ・ 選択性の高いデータにアクセスする共通の操作
- ・ 境界（両側で境界付き）スキャンまたは非有界（片側または両側で）スキャンとすることができる
- ・ データは、索引列の昇順に戻される。
- ・ 同じ値を持つ複数の行は、ROWID で昇順にソート

索引レンジ・スキャン降順

- ・ データが降順に戻されること以外、索引レンジ・スキャンと同じ

索引スキップ・スキャン

- ・ 接頭辞の付いていない列による索引スキャンが改善されます
- ・ コンポジット索引をさらに小さい副索引に論理的に分割できます
- ・ コンポジット索引の初期列が問合せで指定されておらず、その列がスキップされる
- ・ 多くの場合、索引ブロックをスキャンする方が、表データ・ブロックをスキャンするより高速



上記レベル 1(col1 とする)、レベル 2(col2 とする) にコンポジット索引が張られていた場合、論理的に、以下の 2 つの副索引に分割される。

- ・ 値 '1' を持つキーがある第 1 の副索引
- ・ 値 '2' を持つキーがある第 2 の副索引

```
select * from hoge where col2 = 123;
```

- ・ 索引の完全なスキャンは行われず、まず値 '1' を持つ副索引が検索され、次に値 '2' を持つ副索引が検索される

全体スキャン

- ・ 述語が索引内の列の 1 つを参照している場合
- ・ 述語がなく、次の条件が 2 つとも満たされる場合
 - ・ 問合せで参照される表の列すべてが索引に含まれている
 - ・ 少なくとも索引列の 1 つが NOT NULL

高速全索引スキャン

- ・ 問合せに必要なすべての列が索引に含まれ、索引キー内の 1 つ以上の列に NOT NULL 制約が存在する場合に、全表スキャンの代用として使用される
- ・ 表にアクセスすることなく索引そのものに存在するデータにアクセス

索引結合

ビットマップ索引