

【インデックスを極める】

インデックスの必要性

前提として、テーブル内の行（タプル）は、入力された順番で配置されるため、主キー等の昇順で並んでいるわけではありません。つまり、よく目にする整然とデータが配置されたテーブルは、SELECT文やCREATE VIEWの実行結果なのです。

よって、検索対象行を無策で検索するならば、DBMSはDBからメモリ上に、順番に1ページ（入出力単位はページ単位）ずつ読み込んで、そのページにおいて、順番に検索対象行を検索するため、最悪の場合、全ページを読み込むこととなります。これが**表検索**です。

そこで、**CREATE INDEX**を使用して**インデックス**を定義すると、記憶装置内において、DBとは別の領域に**インデックス**が生成されます。**インデックス**は、検索対象行が存在するテーブルの列の値と**ポインタ（その列の値を持つ行の記憶装置上のアドレス）**で構成されます。よって、検索対象行を検索する際は、まずメモリ上にインデックスだけを読み込み、それから、検索対象行が存在するページだけをメモリに読み込みます。

ただし、インデックスにおいても、アドレス順にメモリ上に読み込んで、検索対象行に対する**ポインタ**を得ることになるため、表検索と比較すると格段に効率はいいのですが、下表（インデックスの種類）の方法によって、インデックス自体の検索効率を高めています。

表：インデックスの種類

ハッシュ インデックス	列の値の ハッシュ値 を求め、その ハッシュ値 のアドレスに ポインタ を格納する。ただし、 ハッシュ値 の衝突（シノニム）が発生するような大量のデータを取り扱う場合は有効ではない。
B木 インデックス	列の値を構成要素とした B木 から目的の列の値を辿り、到達した場所に ポインタ を格納する。特定の一つのデータの検索（ ランダムアクセス ）に適している。また、範囲検索も可能であるが、 検索条件である述語（式）は一つにすることになっている。 （列名 BETWEEN 1 AND 100 はOKであるが、同じ意味の列名 >=1 AND 列名<=100 はNG） ※ BETWEEN句は数値、文字列及び日付が使用可能
B+木 インデックス	B木インデックス との相違点は、構成要素は途中にあるものもあるが、全て ポインタ と共にリーフ（最下層）にある。また、リーフ間は連結され、昇順に辿ることができる。これによって、 ランダムアクセス の他、 シーケンシャルアクセス（順次アクセス） もできる。
ビットマップ インデックス	成人か否かのように1か0の二値で表すことができる場合に適している。当然、対応する ポインタ は多くあるため、 非ユニークインデックス となる。

クラスター／非クラスターインデックス

インデックスの分類方法に下表（クラスター／非クラスターインデックス）があります。

表：クラスター／非クラスターインデックス

クラスター インデックス (クラスタ)	<ul style="list-style-type: none">●シーケンシャルな検索効率を高めるために、インデックスの並びとテーブルの内の行の並びが同じになるようにポインタを操作してテーブルを並べ替えるもの●その結果、1ページに検索対象行がたくさん含まれることになり、極端な例としては、検索対象行が1ページに全て含まれる場合には、ページの読み込みは1回だけになる。●B+インデックスのリーフ部分は、シーケンシャルであるため、格納するポインタをシーケンシャルにすることで、クラスタになる。●ポインタを操作するため、クラスタはテーブルにつき、1つしか定義できない。●以上のことから、インデックスは主キー等の一意性が高いものが適しており、その場合、対応するポインタが1つであるユニークインデックスにもなり得る。
非クラスター インデックス (非クラスタ)	<ul style="list-style-type: none">●クラスタとは異なり、テーブルの並び替えは発生しないため、複数定義することができる。●クラスタにより、テーブルの並び替えが行われていても、その並び替え後のポインタを使用することで、非クラスタを定義すればいい。●以上のことから、クラスタと非クラスタは共存できる。

インデックスの定義方法

CREATE文を使用する場合「**CREATE INDEX INDEX名 ON TABLE名 (列名1, …)**」等でインデックスは定義できますが、実際は、下記のとおり、CREATE TABLE文を使用して、効率的に定義することができます。

- ①主キーの列は、自動的にユニークインデックスが設定され、主索引（プライマリーインデックス）となる。
- ②テーブルにクラスターインデックスが存在しない場合には、主索引がクラスターインデックスとなる。
- ③主索引以外は、副次索引（セカンダリーインデックス）であり、例えば、WHERE句における主索引以外の列又は列の組み合わせも副次索引である。

注意事項

- ①前ページで「**テーブル内の行(タプル)は、入力された順番で配置されるため、主キー等の昇順で並んでいるわけではありません**」と記載しましたが、上記のとおり、主キーを設定することで、自動的にクラスターインデックスが作成されます。ただし、試験において、そこまで言及していなければ、安易にそう考えるのは危険です。
- ②インデックスを設定すると、SELECT文による抽出は早くなりますが、**表の更新（挿入、更新、削除）を実行するたびに、インデックスの再構成は必要となり、その間の処理が遅くなります。**