

データベースS 講義資料 第14回 問い合わせ処理、障害回復

九州工業大学 情報工学部 システム創成情報工学科 講義担当：尾下真樹

1. 問い合わせ処理

1.1. 問い合わせ処理の最適化

一般に、ある問い合わせを実現するためには、複数の処理を内部で行う必要がある。このとき、処理を適用する順番によって、最終的に出力される結果は同じであっても、中間データとして必要になるデータ量や処理の効率（所要時間）は大きく異なる。

問い合わせの内部の処理の順番を図 1(a)のような処理木として表し、以下のルールを順番に適用することで、図 1(b)のような、最適な処理手順の処理木が得られる。

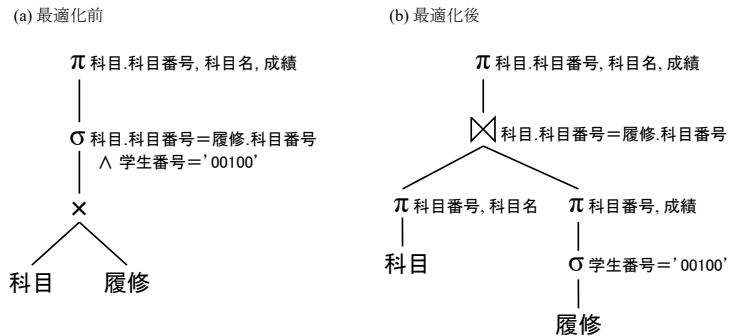


図1 処理木を使った最適化

1. 選択条件を分解
2. 選択を可能な限り下に移す
3. 連続する直積と選択を結合にまとめる
4. 射影を可能な限り下に移す
5. 連続した射影・選択をまとめて一つにする

1.2. 基本データ操作の実行方法

基本データ操作（特に**選択**や**結合**）についても、どのような方法で処理を行うかによって、処理効率が異なる。できるだけ処理効率の良い方法を適用することが望ましい。どの方法を適用できるかは、データが格納されているデータ構造（ソートされた格納されているか、インデックスを持っているか、等）によって異なる。

選択については、データ格納方式の回に学習したように、選択に用いる属性でインデックスがなければ**スキャン法**や**探索法**、インデックスがあれば**インデックス**を使って、処理を実現する。

結合については、結合を行う2つのリレーションがどちらもインデックスを持っていない場合は、2つのリレーションの各データが結合されるかどうかを確認する、**入れ子ループ結合**を用いることになる。この方法は、二重ループでの処理が必要になるため、処理効率は悪くなる。

一方（あるいは両方）のリレーションが結合に用いる属性でのインデックスを持っている場合は、**インデックスを用いた結合**として、インデックスを持っていない方のリレーションの各データについて、もう一方のリレーションの中の対応するデータを、インデックスを用いて探索することができるので、効率的に処理を行える。

もし、両方のリレーションが結合を行う属性でソートされていれば、**マージ結合**として、両方のリレーションを先頭から辿りながら、結合されるデータの組み合わせを取り出ししていくことができるので、両方のリレーションを1回ずつスキャンするだけで済み、最も効率的に処理を行える。

一方（あるいは両方）のリレーションが結合に用いる属性でハッシュを使って格納されている場合は、**ハッシュ結合**を使い、インデックスを用いた結合と同様の方法で、結合を行える。

2. 障害回復

データベースシステムにおいて生じる障害として、トランザクション障害、システム障害、メディア障害がある。**トランザクション障害**は、トランザクションがアボートされるなどして、トランザクションが完了できなくなる障害である。同時実行処理を行うときには、普通に発生する障害となる。**システム障害**、**メディア障害**は、それぞれ、システムやメディア（ハードディスク）に障害が発生して、データベースシステムの動作が継続できなくなる障害である。この3種類の障害の中では、トランザクション障害が最も問題が小さく、メディア障害が最も問題が大きい障害となる。

障害への対応方法としては、全ての操作内容のログを記録しておき障害が発生したらログをもとに状態を復元する**ログ法**、変更を行ったページのコピーを記録しておき障害が生じたらコピーをもとに状態を復元する（障害が生じなければコピーのページは破棄して変更を行ったページを利用するように切り替える）**シャドウページ法**、などがある。どちらの方法も、変更の記録しか残らないため、メディア障害に対応するためには定期的にデータベース全体のバックアップを取っておく必要がある。