

使って覚える 組み込み データベース入門(後編, 最終回)

高橋 君夫

この短期連載も今回で3回目, 最終回となります。データベースのプログラミングの仕方に始まり, ちょっと面白いデータベースの使い方をお話ししてきましたが, 最後は「いかにもデータベースを使った」というようなアプリケーションの話をしたと思います。

一昔前と異なり, 最近のカー・オーディオではCDの音楽を再生するとき, トラック番号のような単純な表示ではなく, 「どのアーティストの, 何のアルバムの, 何という曲」という情報を表示してくれます。さらに, アルバム名から聞きたい曲を選んだり, アーティスト名で曲を選んだりなど, 豊富な検索機能をもっています。

このような分野にデータベースを使えば, C言語で開発するよりも, 機能アップが簡単だということは容易に想像

できます。そこで今回はMP3プレーヤの楽曲検索を取り上げます。

曲検索機能をデータベースを使って実装する場合, どんなデータを持っていればよいかを考える必要があります。ここでは, 音楽データのソースとして, MP3ファイルが持っているメタ情報を前提として, 検索に適したデータのフォーマットを考えます。

1. C言語でデータ操作を考える

CDをデータ・ソースとして考える場合, 「アルバム」を単位にすると考えやすいでしょう。CDのアルバムには「アルバムのタイトル」, 「アーティストの名前」, 「曲名」など, 楽曲を検索する上で必要なデータがそろっています。

これらの情報は表1のように表せます。このようなデータをC言語で扱う場合, 図1のようにアルバムを単位とした構造体を作り, そのアルバムに収められた曲をリスト構造で扱うという方法が思い付きます。また検索の順序として,

表1 アルバムを表で表す

アルバム名	アーティスト名	曲名
ハロウ	たむらばん	ヘイヨーメイヨー お前ただな~送らぬ手紙~ 責めないデイ ⋮
パールピアス	松任谷由美	ようこそ輝く時間へ 真珠のピアス ランチタイムが終わるころ ⋮
ひとりあそび	たむらばん	みんな生きている アイノウ ありがとなんです ⋮

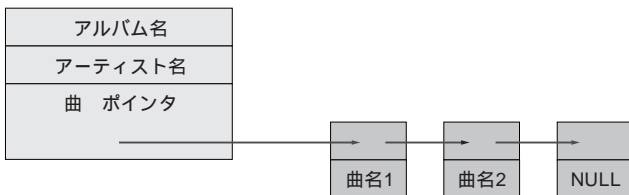


図1 曲名をリスト構造で表す

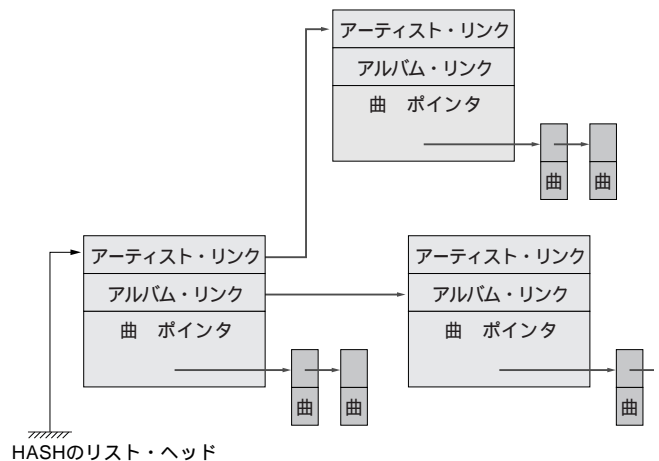


図2 検索速度を向上させようとした場合のデータ構造

表2 重複したフィールドのある表

アルバム名	アーティスト名	曲名
ハロウ	たむらばん	heyoh-meioyo
ハロウ	たむらばん	責めないデイ
パールピアス	松任谷由美	ようこそ輝く時間へ
パールピアス	松任谷由美	真珠のピアス
ひとりあそび	たむらばん	みんな生きている
ひとりあそび	たむらばん	アイノウ

- 1) アーティストが指定される
- 2) アルバムが指定される
- 3) 曲が指定される

というような順序であるとしましょう。そうすると図2のようなデータ構造にして、検索速度を向上させようとするかもしれません。

このようなデータ構造の場合、上記のような検索しか実行されないなら、メモリも無駄に使いませんし、そこそこの速度を得るためのチューニングも簡単に行えるため、何の問題もないと感じるでしょう。しかし、「曲名を直接指定する」という条件を検索ルールに入れると、非常に効率の悪いデータの持ち方であることは明らかです。その場合、資源の消費量とパフォーマンスのバランスを取りながら、データの構造を見直す必要があります。

このようにC言語でデータ操作を記述すると、ある検索条件に特化した場合、非常に高速に記述できることは事実です。しかし、検索条件が変わったり、データの項目が増えたりすると、その度にプログラムを書き換える必要があります。簡単に言うと柔軟性がないのです。

アルバム名	アーティスト名	曲名
ハロウ	たむらばん	heyoh-meioyo
ハロウ	たむらばん	責めないデイ
パールピアス	松任谷由美	ようこそ輝く時間へ
パールピアス	松任谷由美	真珠のピアス
ひとりあそび	たむらばん	みんな生きている
ひとりあそび	たむらばん	アイノウ

アーティストID	アーティスト名
1	たむらばん
2	松任谷由美

図3 アーティストの重複を排除する

2. 表によるデータの保持と正規化

これに対し、データベース使用時に先ほどの表1のようなデータがあるとし、これをまず表2のように、重複しているフィールドを取り除きます。このようにして、データのアクセス単位(レコード)を単純化します。

表2のテーブルを見ると、アーティスト名が重複しています。これを別のアーティスト表(アーティスト・テーブル)に追い出し、さらに重複を排除します(図3)。またアルバムからもアーティストを特定できます。アルバム名の重複を解消した上で、アーティストは(曲ではなく)アルバムと関係付けすると、図4のような三つのテーブルが完成します。このように冗長性を排除していく作業を「正規化」と呼びます。

データベースではこのような正規化を行うことにより、データの重複をなくします。あるデータを削除してもほかのデータに矛盾が発生しないようにテーブルを構成することで、より効率的にデータを扱えるようになるのです。

C言語でも同じようなデータ構造を作れると思われるでしょうが、「何かデータの項目を加える」、「検索する条件が加わる」などの新たな項目が発生すると、プログラムの作り直しになります。

もちろんデータベースを使用したアプリケーションも、上記のような変更が発生した場合に、全くプログラムを書

アルバムID	曲名
1	heyoh-meioyo
1	責めないデイ
2	ようこそ輝く時間へ
2	真珠のピアス
3	みんな生きている
3	アイノウ

曲名テーブル

アーティストID	アーティスト名
1	たむらばん
2	松任谷由美

アーティスト・テーブル

アルバムID	アルバム名	アーティストID
1	ハロウ	1
2	パールピアス	2
3	ひとりあそび	1

アルバム・テーブル

図4 完成した三つのテーブル