

3 データベース概論

3.1 データベースとは

データベース (database) はデータの基地と直訳できる。データを有用な情報と考えれば、データベースとは蓄積された有用な情報の基地である。この基地にアクセスすれば、欲しい情報を手に入れることができる。

3.1.1 データベースは目的別に構築される

もし、あらゆる情報を1つのデータベース(例えば音楽情報、スポーツ情報、遊戯スポット情報)に集めたとき、音楽のタイトル情報をキーワードに検索(抽出ともいう)する際に、スポーツや遊戯スポット情報といった明らかに関連のないデータ群も含めて探し出すことになる。これでは検索に余計な時間を費やすことになり効率が悪い。

しかし、予め目的によってデータベースを分けて構築しておけば、検索の際に不要なデータ群への参照が必要なくなり検索に要する時間が軽減されることになる。

そればかりでなくデータベースの構築や管理もしやすくなる。例えば音楽情報に限定してデータベースを構築すれば、

アーティスト名, タイトル名, ジャンル名, CD あるいは DVD の有無などの項目(フィールド)を用意するだけ足りる。しかし、遊戯スポットの情報まで含めたデータベースを構築すると、音楽情報では利用しないような遊戯店の住所や電話番号、営業時間などのフィールドも事前に用意しなければならない。結果的に必要以上のフィールドを用意することになる。データベースは目的ごとに分けるべきであり、フィールド数については必要最小限に留めるに越したことはない。

3.1.2 データベースは管理・運用される (CRUD とは)

情報は日々変化するため、データベース内に新しくテーブルを作成(Create)したり、そのテーブルに新しくデータ(専門用語でレコード。以下レコードという)を追加したり、レコードの内容を更新(Update)したり、あるいは古くなったレコードを削除(Delete)したりといった管理が必要になる。そして蓄積されたデータベースから欲しい情報を探し出す(Read)といった運用も必要になる。

即ち、データベースの管理と運用を行なうためのシステムが必要になる。このシステムをデータベース管理システムといい、DBMS(DataBase Management System)と表す。ユーザーは、このDBMSを介してデータベースの管理と運用を行なう。DBMSは、SQL¹という言語を実装しており、この言語を使用してユーザーは命令を出すことになる。

以上より、データベースとは、データそのものを蓄積する Database とそれを管理・運用するためのシステムである DBMS の2つを統括して指す言葉である。そのため、単にデータベースではなく、データベースシステムと呼ばれることが多い。

以降から、Database とは蓄積されたデータそのものを指し、Database+DBMS についてはデータベースシステムと呼ぶことにする。

¹Structured Query Language の略である。これについては後述する。

Database(情報の基地) + DBMS(DataBase Management System)

本来、データを蓄積する Database には規格があり、共通である。それにも関わらずデータベースを管理・運用するシステムは、それを開発した時期や背景、開発者の考え方によって違いが生じた。そのため DBMS の違いによって、それぞれ幾種類かのデータベースシステムが存在することになる。

現在の代表的なデータベースシステムは、本授業で利用する MySQL だけでなく、PostgreSQL, Oracle Database, SQL Server などがある。前者2つはオープンソースライセンスであるため、マニュアルや文献などを参照しながら利用者自身でサポートしていくのであれば、無償で利用できる。後者2つは、企業による開発が進められているもので、利用にあたってはサポートなども含め、ライセンスを有償で購入する必要がある。それぞれに特徴や若干の仕様の違いがあるが、その多くは共通しており、基盤となる知識体系は同じである。そのため MySQL を学ぶことは、将来どのデータベースシステムにおいても通用する知識となる。

3.2 リレーショナル型 Database

Database は必ず名前をつけて管理する。Database は、Table と呼ばれるデータファイルの集まりによって構成されている。この Table とは、Table 毎に決められたフィールドによって構成されるレコードの集まりである。これらの Table も名前をつけて管理する。大規模な Database になればなるほど、この Table 数は膨大に増える。

Database のイメージ

Database:music

Table1:artistName

artist_code	artist_nm
000001	黒夢
000002	Maximam the ホルモン
000003	Sads
...	...

Table2:musicTitle

artist_code	music_nm
000001	ピストル
000002	爪爪爪
000003	サロメ
...	...

Database には、この Table の構築の仕方によっていくつかの種類があり、現在主流なのがリレーショナル²型と呼ばれる Database である。

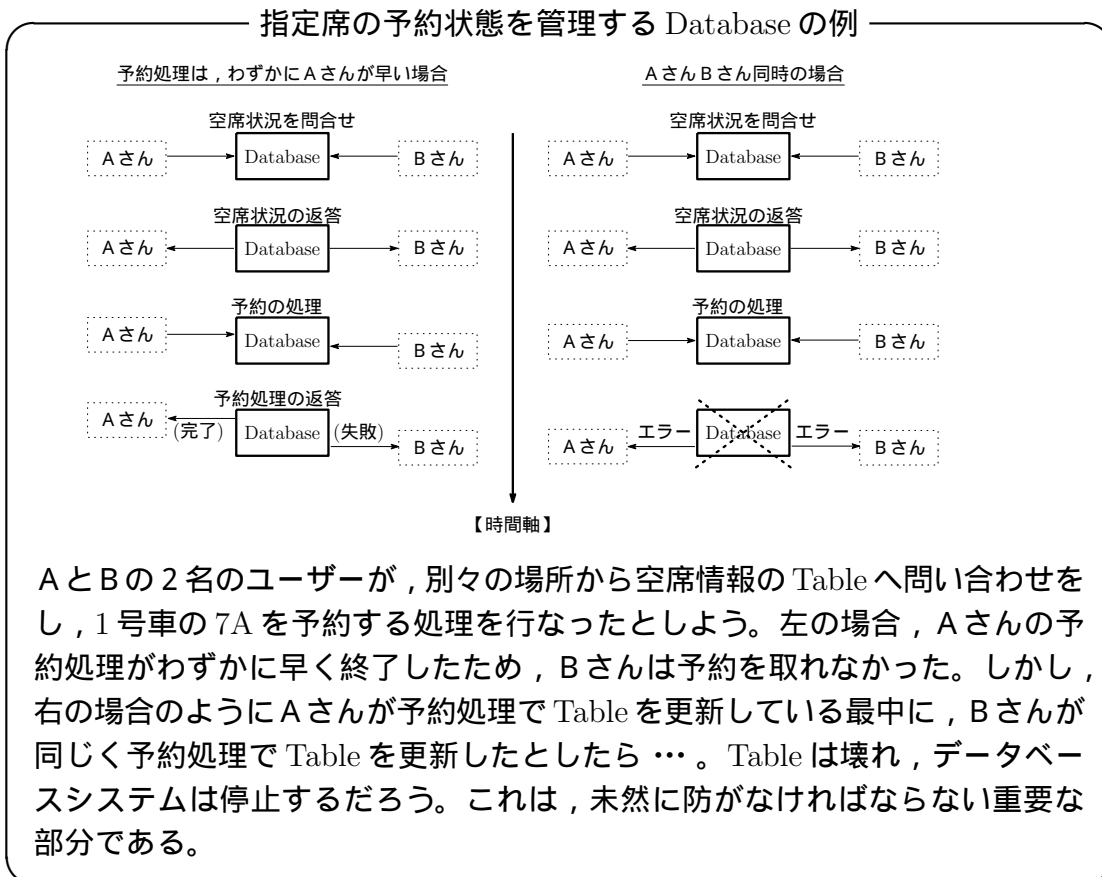
リレーショナル型以外にも階層型とか、ネットワーク型などと呼ばれる方式もある。

²relational: "相互関係の"の意

リレーショナル型 Database の特徴は、複数の Table に関連性を持たせることで Table 間を連結して利用できる点にある。

3.2.1 排他制御処理

DBMS は、Database へ問い合わせをするユーザーが多数いるという想定で開発されている。特にインターネットに公開されたデータベースシステムの場合ならユーザー数は膨大になる。こういった場合、複数のユーザーから同時期に一つの Table に問い合わせが来るということも十分あり得る。次の例を見てみる。



これを防ぐ処理のことを排他制御処理という。Aさんが処理を開始するために Table に問い合わせをした際に、その Table にロックを掛ける処理のことである。ロックの仕方には、共有ロックと占有ロックの2種類があり、前者はロックした Table への参照のみ許可し、更新を許可しない。後者は参照すらも許可しない。

この排他制御処理のおかげで、Table は安全が保たれる。

3.2.2 トランザクションとコミット、ロールバック

DBMS は、途中で途切れることが許されないユーザーからの問合せの開始と終了までをトランザクションと呼び、これを処理の一括りにする概念がある。1つのトランザクションが正常に処理され、Database の整合性が保たれていることを確認したとき、はじめて Database への書込み（更新）を行なう。このことをコミットと呼ぶ。もし、なんらかのトラブルで整合性が保てないときには、トランザクションを破棄し、直前の状態に戻さなければならない。このことをロールバックと呼ぶ。

この一連の機能によって、Database の整合性は保たれる。