

# 第一回輪講 データベースの基礎

名城大学 理工学部 情報工学科  
渡邊研究室 B4  
三宅 佑佳

# 書籍について

- ▶ これだけはおさえたい データベースの基礎の基礎
  - 著者: 谷尻かおり
  - 発行者: 片岡巖
  - 発行所: 株式会社技術評論社
  - 発行: 2009年10月25日

※本資料は上記の書籍を基にして作成されたものです。文書の内容の正確さは保証できないため、正確な知識を求める方は原文を参照してください。



# データベースとは

- ▶ **ある目的のために**集めたデータを**一定の規則に従って**コンピュータ上に蓄積したもの
  - ある目的のために
    - 何かをしたいという要求からどのようなデータが必要か考えて過不足なくデータを蓄積すること
  - 一定の規則に従って(書籍データベースの場合)
    - ISBN番号、タイトル、著者、出版社、出版年度、ジャンルの登録
    - すべての項目を必ず入力する
    - 同じ書籍を重複して登録しない

※ ISBN(International Standard Book Number) は、世界共通で図書を特定するための番号である。日本語に訳すと国際標準図書番号となる。

# データベースの登場背景

- ▶ ファイルや表計算ソフトでのデータ管理
  - データの二重登録を防ぐことが難しい
  - ファイル間でデータの間係を保つことが難しい

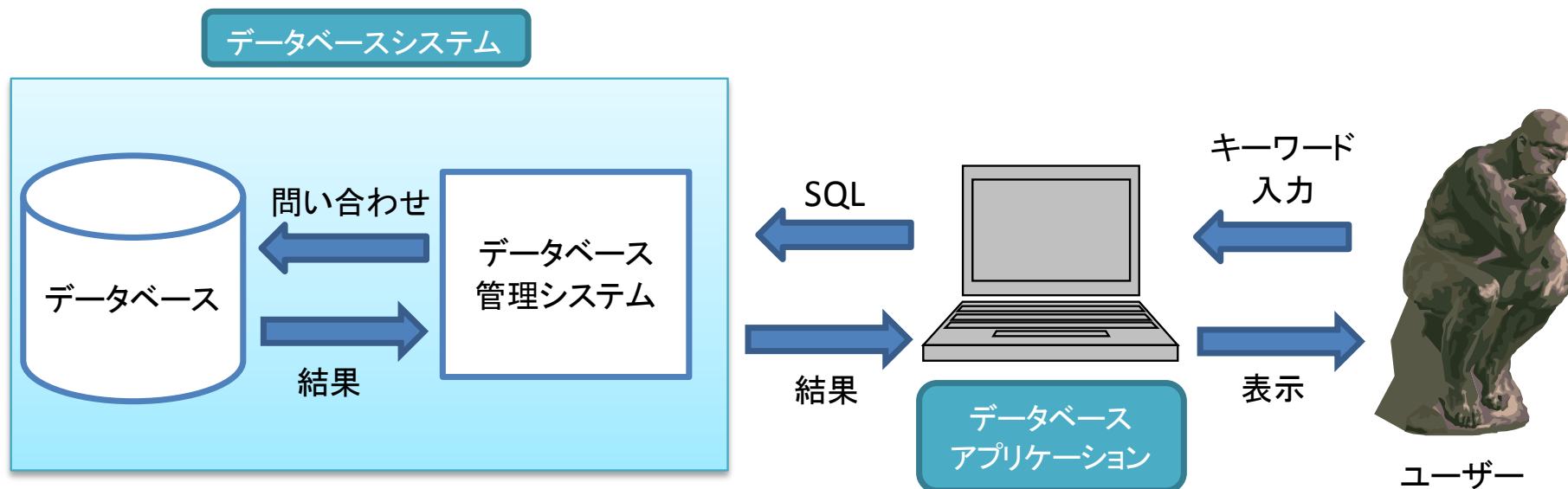


データベースを用いることで  
これらの問題を解決することができる

# データベースの全体像

## ▶ (例) 書籍検索システムの場合

- タイトルや著作名が不明瞭でもキーワードを入力するだけで候補の書籍を一覧表示可能



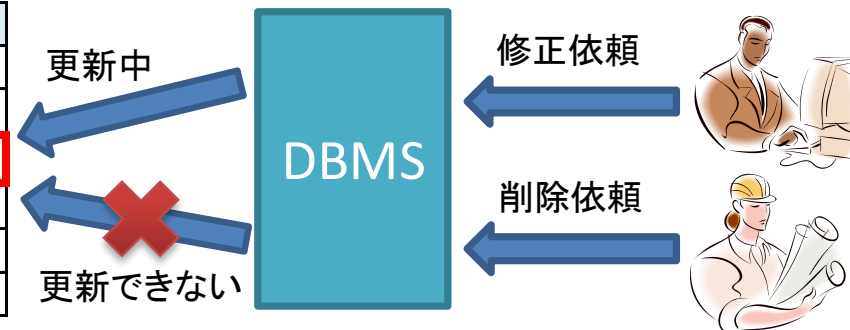
# データベース管理システム

## ▶ データベース管理システム

(DBMS: DataBase Management System)

- 主にデータベースの管理と運用を行う
  - 新しいデータを登録するとき重複するものがないかどうか確認
  - 利用者からの要求に応じて必要なデータを抽出
  - 複数人が同じデータに同時にアクセスしたときにデータの矛盾が生じないように監視

ISBN	タイトル
4-7741-3759-9	接客とルールのマナー らくらくマスターBOOK
4-7741-3758-2	ビジネスルール「ころ」の伝え方教えます
4-7741-3756-8	はじめての「私流！癒しサロン」オープンBOOK
4-7741-3704-9	なぞの金属・レアメタル
4-7741-3738-4	Excel 2007の困った！をいまずぐ解決する本
4-7741-3737-4	Word 2007の困った！をいまずぐ解決する本



# データベースアプリケーション

- ▶ データベースアプリケーション
  - 主にユーザーとデータベースシステムの橋渡しを行う
  - ユーザーが入力したキーワードを用いて書籍検索する場合の内部処理
    - ユーザーが入力したキーワードで検索する為のSQLを作成
    - 作成したSQLをデータベース管理システムに発行する
    - データベース管理システムから受け取った結果をわかりやすい形に編集して画面に出力

# SQLとは

- ▶ データベースを操作するための言語
  - データを削除するときはDELETE
  - データを追加するときはINSERT など

例)「社員という名前の表から山田さんのデータを削除する」

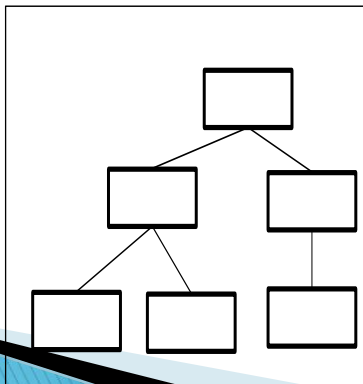
```
DELETE FROM 社員  
WHERE 社員名 = '山田';
```



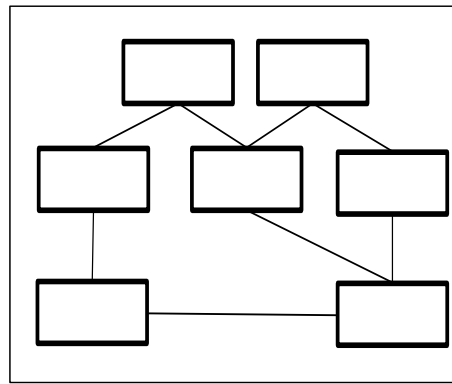
# データベースの種類

- ▶ 階層型
  - 木構造で表しデータを階層的に格納・整理する仕組み
- ▶ ネットワーク型
  - 各データの繋がりを網目の形で表現
- ▶ リレーショナル型
  - 複数の表同士の関係で構成

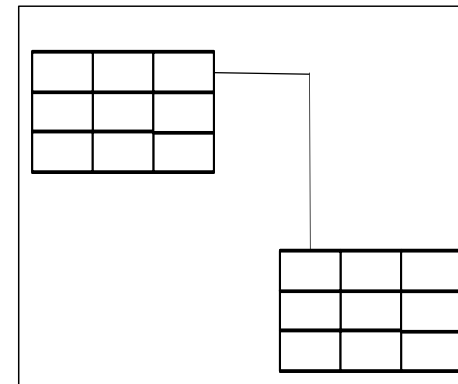
階層型



ネットワーク型



リレーショナル型



# リレーショナル・データベースとは

- ▶ 複数の表を利用してデータの管理を行う
- ▶ 表同士に**関係 (relation)**を持たせることができる

社員番号	名前	部署コード
20011214	山田 太郎	S1
20011215	佐藤 花子	S1
20010023	田中 桃子	D1
19951624	鈴木 一郎	D1
19997932	高橋 五郎	M2

部署コード	部署名	内線番号
S1	営業部	1234
D1	開発部	5678
M1	総務部	101
M2	人事部	102
M3	経理部	103

# リレーショナル・データベースの特徴(1)

- ▶ データはテーブルで管理
  - テーブル : データを蓄積する下図のような表
  - レコード : 1件分のデータの集まり(行)
  - フィールド : 蓄積する項目(列)

フィールド

登録番号	名前	性別	都道府県	郵便番号	住所
1	山田 太郎	男性	東京都	162-0846	新宿区市ヶ谷左内町99-999
2	佐藤 花子	女性	東京都	113-0033	文京区本郷99-999
3	田中 桃子	女性	東京都	171-8553	豊島区高田99-999
4	鈴木 一郎	男性	東京都	170-0052	港区赤坂99-999
5	高橋 五郎	男性	東京都	105-0001	渋谷区神宮前99-999

レコード

# リレーショナル・データベースの特徴(2)

## ▶ テーブルは2次元の表形式で構成

登録番号	名前	住所
1	山田 太郎	162-0846 東京都 新宿区市ヶ谷左内町99-999
2	佐藤 花子	113-0033 東京都 文京区本郷99-999
3	田中 桃子	171-8553 東京都 豊島区高田99-999
4	鈴木 一郎	170-0052 東京都 港区赤坂99-999
5	高橋 五郎	105-0001 東京都 渋谷区神宮前99-999

← 1つのフィールドに複数のデータ

フィールドを  
3つにわけると

登録番号	名前	郵便番号	都道府県	住所
1	山田 太郎	162-0846	東京都	新宿区市ヶ谷左内町99-999
2	佐藤 花子	113-0033	東京都	文京区本郷99-999
3	田中 桃子	171-8553	東京都	豊島区高田99-999
4	鈴木 一郎	170-0052	東京都	港区赤坂99-999
5	高橋 五郎	105-0001	東京都	渋谷区神宮前99-999

# リレーショナル・データベースの注意(1)

- ▶ フィールドには必ず違う名前を付けること

同じ名前が使える



一覧表	登録番号	名前	性別	住所	住所	住所
	1	山田 太郎	男性	東京都	162-0846	新宿区市ヶ谷左内町99-999
	2	佐藤 花子	女性	東京都	113-0033	文京区本郷99-999
	3	田中 桃子	女性	東京都	171-8553	豊島区高田99-999
	4	鈴木 一郎	男性	東京都	170-0052	港区赤坂99-999
	5	高橋 五郎	男性	東京都	105-0001	渋谷区神宮前99-999

異なる名前にしなくてはならない



テーブル	登録番号	名前	性別	都道府県	郵便番号	住所
	1	山田 太郎	男性	東京都	162-0846	新宿区市ヶ谷左内町99-999
	2	佐藤 花子	女性	東京都	113-0033	文京区本郷99-999
	3	田中 桃子	女性	東京都	171-8553	豊島区高田99-999
	4	鈴木 一郎	男性	東京都	170-0052	港区赤坂99-999
	5	高橋 五郎	男性	東京都	105-0001	渋谷区神宮前99-999

# リレーショナル・データベースの注意(2)

- ▶ テーブルには重複した値の登録できない
  - データの二重登録を防ぐためレコードの識別に使用するフィールド(主キー)を設定
    - すでに登録済の値と同じ値は入力不可
    - フィールドには必ず値を入力しなければならない
- ▶ 決められた種類やサイズのデータしか入力できない
  - 同じ数値データでも10と100,000ではメモリサイズが違う
  - 見た目は同じ「10」でも文字データと数値データでは違う

# リレーショナル・データベースの注意(3)

- ▶ 1つのデータは一か所で管理する
  - 所属や内線番号のようなデータを1つのテーブルに繰り返して登録してはいけない
  - 下図のようなデータはテーブルを2つに分けて関連付けて使用

社員番号	名前	部署コード
20011214	山田 太郎	S1
20011215	佐藤 花子	S1
20010023	田中 桃子	D1
19951624	鈴木 一郎	D1
19997932	高橋 五郎	M2

リレーションシップ

部署コード	部署名	内線番号
S1	営業部	1234
D1	開発部	5678
M1	総務部	101
M2	人事部	102
M3	経理部	103

# テーブル間の関連付け

- ▶ リレーションシップ
  - テーブルどうしを関連付けること
  - リレーショナル・データベースの最大の特徴
  - 条件に従ったテーブルでなければ設定できない
- ▶ リレーションシップの種類
  - 一対一
  - 一対多
  - 多対多



# 一対一リレーションシップ

- ▶ テーブルの1レコードが他方テーブルの1レコードに対応
- ▶ フィールド数が多いためやセキュリティ管理のためにテーブルを2つに分けるといった限定的な利用場面

顧客1	登録番号	名前	性別	顧客2	登録番号	電話番号	郵便番号	住所
	1	山田 太郎	男性		1	03-3513-9999	162-0846	新宿区市ヶ谷左内町99-999
	2	佐藤 花子	女性		2	03-3812-9999	113-0033	文京区本郷99-999
	3	田中 桃子	女性		3	03-5959-9999	171-8553	豊島区高田99-999
	4	鈴木 一郎	男性		4	03-5599-9999	170-0052	港区赤坂99-999
	5	高橋 五郎	男性		5	03-5477-9999	105-0001	渋谷区神宮前99-999



1つのテーブルにまとめる

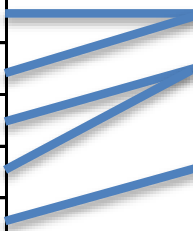
登録番号	名前	性別	電話番号	郵便番号	住所
1	山田 太郎	男性	03-3513-9999	162-0846	新宿区市ヶ谷左内町99-999
2	佐藤 花子	女性	03-3812-9999	113-0033	文京区本郷99-999
3	田中 桃子	女性	03-5959-9999	171-8553	豊島区高田99-999
4	鈴木 一郎	男性	03-5599-9999	170-0052	港区赤坂99-999
5	高橋 五郎	男性	03-5477-9999	105-0001	渋谷区神宮前99-999

# 一対多リレーションシップ

- ▶ テーブルの1レコードが他方テーブルの複数レコードに対応
- ▶ 1つの情報は1か所で管理する形
- ▶ データを効率よく管理できる
- ▶ 必要なデータを取り出しやすい

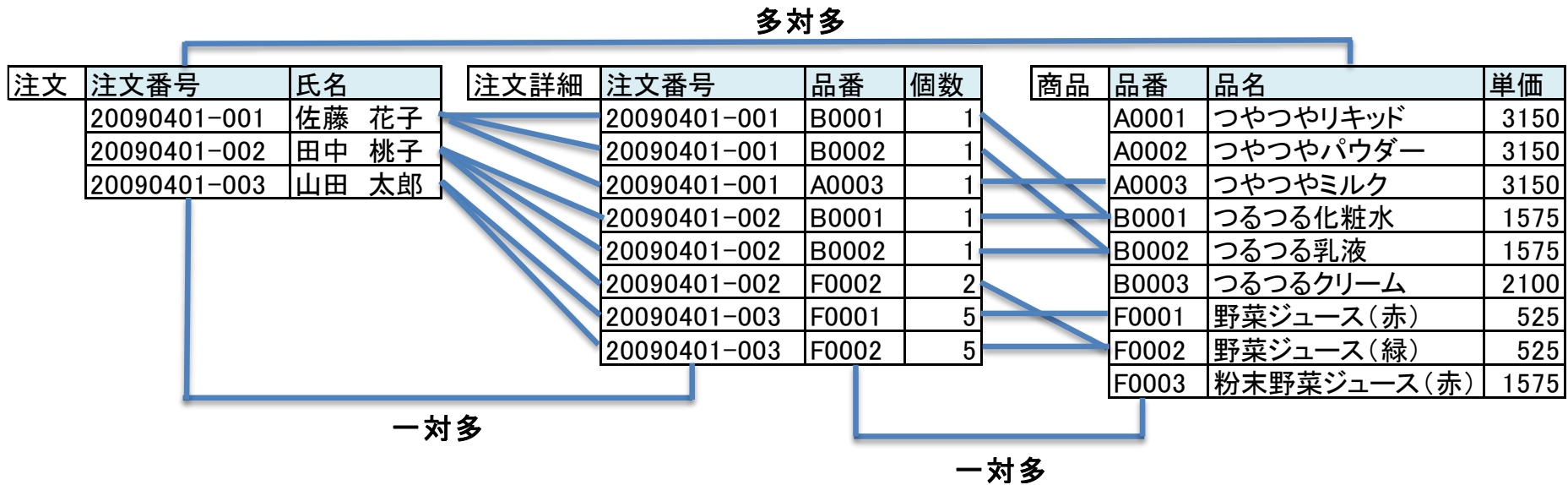
社員番号	名前	部署コード
20011214	山田 太郎	S1
20011215	佐藤 花子	S1
20010023	田中 桃子	D1
19951624	鈴木 一郎	D1
19997932	高橋 五郎	M2

部署コード	部署名	内線番号
S1	営業部	1234
D1	開発部	5678
M1	総務部	101
M2	人事部	102
M3	経理部	103



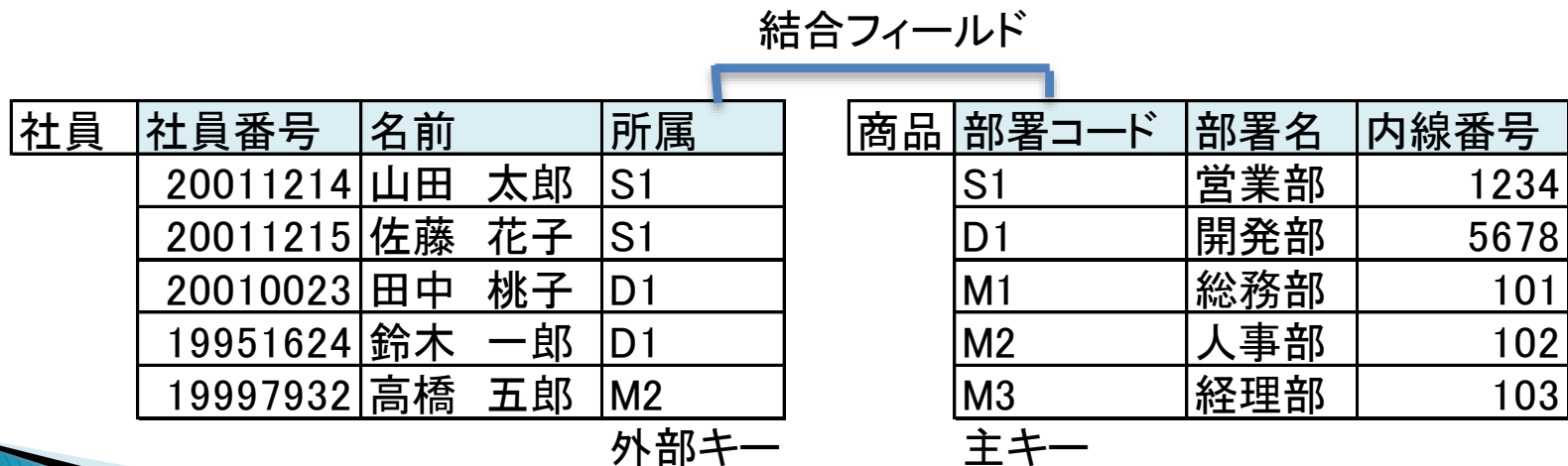
# 多対多リレーションシップ

- ▶ 2つのテーブル間に結合テーブルを設けることで作成
- ▶ 2つのテーブル間に多対多のリレーションシップが直接存在することはない



# リレーションシップの設定条件

- ▶ 両方のテーブルに同じ値を保持するフィールド（結合フィールド）があること
- ▶ 一方のテーブルに主キーが設定されていること
  - 主キー：レコードの識別に使用するフィールド
  - 外部キー：他のテーブルの値を参照するための値



# まとめ

## ▶ データベース

- ある目的のために集めたデータを一定の規則に従ってコンピュータ上に蓄積したもの
- データの二重登録防止やデータの関係保持

## ▶ リレーショナル・データベース

- 多数の表を利用してデータの管理を行う
- 表どうしに関係を持たせることができる

# 補足資料

# 身の回りのデータベース

- 携帯できるデータベース
  - ・ 携帯電話や携帯情報端末のアドレス帳やスケジュール
- 誰もが使えるデータベース
  - ・ インターネットの検索エンジン
- 裏方として活躍するデータベース
  - ・ 買い物時のポイントサービス
  - ・ 電子カルテ
- 企業の中で使うデータベース
  - ・ 顧客管理や売上管理、商品管理

# 表計算ソフトの利点と欠点

## ▶ 利点

- 入力するデータの種類やサイズに対する決まりがない
- 表の構成が自由
- 関数を使ってグラフ化や集計が可能

## ▶ 欠点

- データの矛盾が発生する可能性がある
- 業務として定型化しにくい表ができる可能性が高い
  - 作る人間によって同じ内容の違う表ができる
- メンテナンスの手間がかかる

参考文献:

「Visual Basic For Access入門」

<http://nilgiri.keddy.ne.jp/lib/vb/vbaccess/>



# データベースの利点と欠点

## ▶ 利点

- キーの設定によりデータの矛盾が発生しにくい
- 業務として定型化しやすい(誰でも簡単にデータを扱える)
- 定型化されているためメンテナンスが容易
- 共通する項目の表示や任意の期間のデータ抽出が容易

## ▶ 欠点

- 決められた種類やサイズのデータしか入力できない
- 一度定義したテーブルの構成は変更しにくい
- 簡単にグラフ化や集計ができない

参考文献:

「Visual Basic For Access入門」

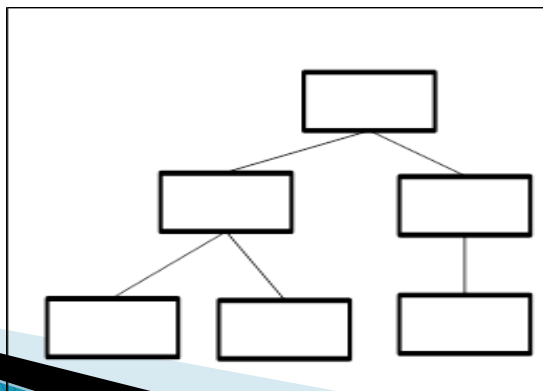
<http://nilgiri.keddy.ne.jp/lib/vb/vbaccess/>

# SQLの命令の種類

- ▶ データ定義言語：データを登録する為の枠組みを定義
  - CREATE : データベースやテーブル、インデックスを作成
  - DROP : データベースやテーブル、インデックスを削除
  - ALTER : データベースやテーブル定義の変更
- ▶ データ操作言語：データの操作の基本
  - SELECT : テーブルから目的のレコードの取得
  - INSERT : テーブルに新たなレコードの追加
  - DELETE : テーブルからレコードを削除
  - UPDATE : 登録済のデータの更新
- ▶ データ制御言語：データベース及びデータに対する制御
  - GRANT : ユーザの権限を設定
  - REVOKE : ユーザーの権限を削除する

# 階層型データベース

- ▶ データを階層的に表現
  - ⇒ データの冗長化・重複が発生しやすくなる
    - 親子(1対多)の関係だけではなく「多対1」の場合や「多対多」の場合もある
    - データの中身は同じだが親の位置が異なることによって重複するデータ所有が発生
- ▶ データベースの構造に変更があった際はそれに合わせてプログラムの改変も必要

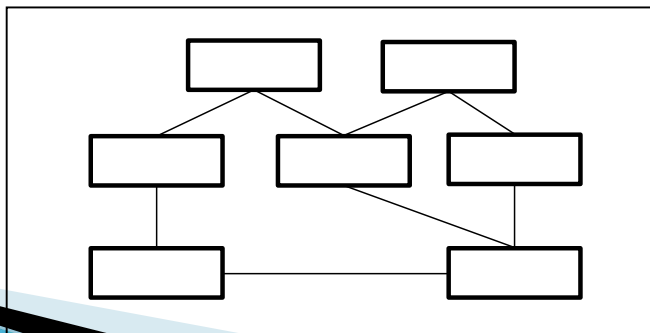


参考文献:

「基礎から始めるデータベース入門セミナー」  
<http://www.oracle.com/technetwork/jp/articles/index-155309-ja.html>

# ネットワーク型データベース

- ▶ 1つのデータに対して複数の親を持つような構造
  - データの冗長化・重複は解消されている
- ▶ データ構造への依存性が強い
  - データの整理や一元化は難しい
  - データ構造を意識したプログラムやデータ構造の変更に伴うプログラム改変の必要性
    - ⇒ 多くのユーザーやプログラムがデータを使用していた場合にメンテナンスに大きな工数と負荷をかける

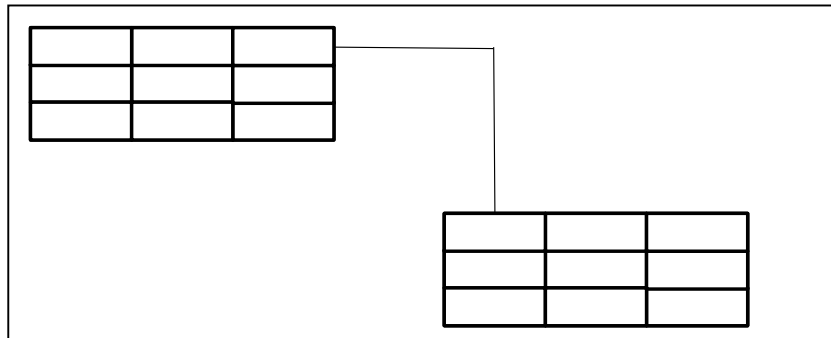


参考文献:

「基礎から始めるデータベース入門セミナー」  
<http://www.oracle.com/technetwork/jp/articles/index-155309-ja.html>

# リレーショナル型データベース

- ▶ プログラムとデータの分離
  - データ構造に修正が入ったとしてもプログラムへの影響は極めて小さい
- ▶ 柔軟かつ容易なデータの取り出しが可能
- ▶ データベース操作の簡略化
  - SQLによりデータベースの構築や問合せが簡略化された



参考文献:

「基礎から始めるデータベース入門セミナー」  
<http://www.oracle.com/technetwork/jp/articles/index-155309-ja.html>

# テーブル内での制約(キー)

## ▶ キー

- テーブルの中でレコードを識別するための情報
- リレーショナル型ではフィールドの値をキーとして使用

社員番号	名前	部署コード
20011214	山田 太郎	S1
20011215	佐藤 花子	S1
20010023	田中 桃子	D1
19951624	鈴木 一郎	D1
19997932	高橋 五郎	M2

リレーションシップ

部署コード	部署名	内線番号
S1	営業部	1234
D1	開発部	5678
M1	総務部	101
M2	人事部	102
M3	経理部	103

# キーについて

## ▶ 候補キー

- 値が重複しておらず、各レコードが識別可能なフィールド

## ▶ 主キー

- 実際にレコードの識別に使うフィールド
- 候補キーの中から1つだけ設定が可能

登録番号	名前	郵便番号	都道府県	住所
1	山田 太郎	162-0846	東京都	新宿区市ヶ谷左内町99-999
2	佐藤 花子	113-0033	東京都	文京区本郷99-999
3	田中 桃子	171-8553	東京都	豊島区高田99-999
4	鈴木 一郎	170-0052	東京都	港区赤坂99-999
5	高橋 五郎	105-0001	東京都	渋谷区神宮前99-999

主キー

識別できない

# 主キーのルール

- ▶ すでに登録した値と同じ値は入力できない
- ▶ フィールドには必ず値を入力しなければならない



データベース管理システムの監視下であり、  
間違った値を登録しようとするときエラーを返す



# 主キーの選定(1)

- ▶ 候補キーの中から主キーを設定
  - 目的のレコードを識別できること
  - 管理しやすい値であること
    - データの並べ替えや検索にも使用される為、文字数の少ないものの方が望ましい。

登録番号	名前	郵便番号	都道府県	住所
1	山田 太郎	162-0846	東京都	新宿区市ヶ谷左内町99-999
2	佐藤 花子	113-0033	東京都	文京区本郷99-999
3	田中 桃子	171-8553	東京都	豊島区高田99-999
4	鈴木 一郎	170-0052	東京都	港区赤坂99-999
5	高橋 五郎	105-0001	東京都	渋谷区神宮前99-999



主キー

## 主キーの選定(2)

- ▶ 複数のフィールドの値を組み合わせて主キーにすることも可能(複合キー)

複合キー

登録番号	名前	電話番号	性別	郵便番号	住所
1	山田 太郎	03-3513-9999	男性	162-0846	新宿区市ヶ谷左内町99-999
2	佐藤 花子	03-3812-9999	女性	113-0033	文京区本郷99-999
3	田中 桃子	03-5959-9999	女性	171-8553	豊島区高田99-999
4	鈴木 一郎	03-5599-9999	男性	170-0052	港区赤坂99-999
5	高橋 五郎	03-5477-9999	男性	105-0001	渋谷区神宮前99-999

# フィールドにおける制約

## ▶ 制約とは

- 属性を設定することにより、テーブルに登録するデータが従う一定の約束事
- 約束事が守られているかどうかを監視するのはデータベース管理システム

# フィールドに設定可能な制約

- ▶ UNIQUE制約(一意性)
  - 重複した値を登録できなくする制約
- ▶ NOT NULL制約(非NULL)
  - 必ず値の入力を要求する制約
- ▶ DEFAULT制約(既定値)
  - フィールドの値として自動的に値が設定される
- ▶ CHECK制約(入力規則)
  - フィールドに入力する値の範囲や書式を制限

# NULL値

## ▶ NULL値

- フィールドに値が何も入力されていない状態
  - 入力する値がわからない(unknown)
  - 入力する値がない(N/A: not applicable)

分類	名前	部署コード
既婚	山田 太郎	さくら
既婚	佐藤 花子	
未婚	田中 桃子	

←unknown

←N/A