

# コンピュータ基礎

[テクノロジ系]  
データベースのお話し

中村勝則

武庫川女子大学

# データベース

## ■ まずは前提から...

- データを登録するシステム.
- しかし「ファイルの置き場所」ではない...
- 質問に答えてくれるシステム
- 喩え話：「物知りボックス」

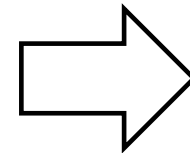


# データベース

## ■「“情報の登録”あるところにデータベースあり」

- インターネットの情報サイト
- 動画／音楽配信サイト
- POSの会員登録
- 銀行
- Etc.

とにかく何でも彼に  
覚えておいてもらう...



# データベースの能力

## ■ 記憶力

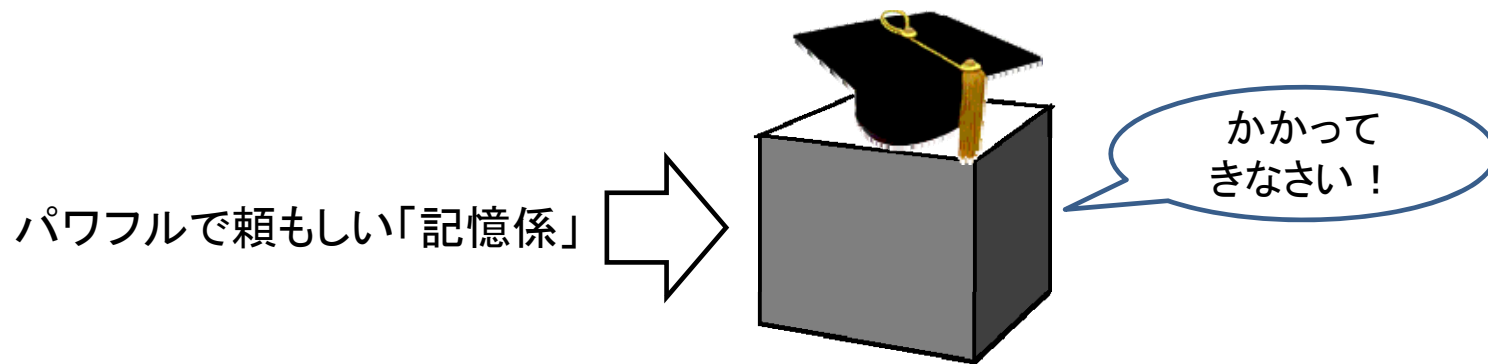
- 数億件(レコード)～

## ■ 読み書きの速度

- 一回の応答は数ミリ秒(あるいはそれ以下)

## ■ 対応力

- 一度にたくさんのクライアントの相手をする(数十人～)



# データベース

- ここまでは「前提」
  - 試験には出ません.
  - でも理解の基礎(重要)

# データベース

## ■ ITシステムの「記憶係」

- 大量の情報を記憶し、問い合わせに答える
- 正体：
  - ◆ データ管理用のソフトウェア (DBMS) をインストールしたコンピュータ (サーバ)
- データの関連性にもとづいて回答する機能

## ■ DBの記憶のしかた P.280～281

- どうやって膨大な情報を記憶しているのか？  
→ 「データベースのモデル」
- 1) リレーショナル型
- 2) ネットワーク型
- 3) ツリー型



# リレーショナルDB (RDB)

## ■ 重要な前提:

- あらゆる情報を「~の...は△である」の形で表現

◆ ~の: 「~」をキーという (喩え:「主語」)

◆ ...は: 「...」を属性という  
DB用語では項目と呼ぶ

◆ △である: 「△」を値という (具体的な数値や文字データ)

◆ 例: 「中村の体重は65kgである」

- 表(テーブル)にしてデータを記録している



氏名	体重	身長
中村	65	164
田中	73	172

# リレーショナルDB

## ■ 1つの表だけでは単純すぎる...

- 1つの表には決まった項目しか記録できない
- 柔軟で高度な応答を実現するためには表を連鎖的に使用する

## ■ 用語:

- キー
  - ◆ データの検索に使う項目  
(複数の項目をキーにできる)

氏名	体重	身長
中村	65	164
田中	73	172



## 例) RDBの表の関連 (P.288の補足)

受注テーブル

受注No.	受注年月日	得意先コード
0001	2011/1/10	A-1
0002	2011/1/10	B-1
0003	2011/1/11	A-20

得意先テーブル

得意先コード	得意先名	所在地
A-1	南北電気	東京都...
A-20	いろは電子	兵庫県...
B-1	日本工業	大阪府...

通常、左端のキーが「主キー」

受注明細テーブル

受注No.	商品No.	数量
0001	1-001	2
0001	2-004	3
0002	5-012	6
0002	1-002	10

商品テーブル

商品No.	商品名	単価
1-001	テレビ(液晶)	200,000
1-002	テレビ	15,000
2-004	DVDレコーダ	80,000
5-012	ラジオ	3,000

たくさんの表を保持しており、つなぎあわせて情報を取り出す

## こんな例も可能, しかし...

受注 No.	受注 年月日	得意先 コード	得意 先名	所在地	商品 No.	商品名	単価	数量	受注 小計	受注 合計
0001	2011/1/10	A-1	南北電気	東京都...	1-001	テレビ(液晶)	200000	2	400000	640000
					2-004	DVDレコーダ	80000	3	240000	
0002	2011/1/10	B-1	日本工業	大阪府...	5-012	ラジオ	3000	6	18000	168000
					1-002	テレビ	15000	10	150000	

### ■ 複雑で無駄が多い (P.287の表)

- 同じ情報を繰返して登録する部分が多発する...

- → まとまった項目たちを分離して  
複数の表にするほうが良い

#### ◆重要: 「正規化」

繰返しによる無駄を省き, 矛盾が起こらないようにする工夫  
→ 第一正規化から第三正規化まで

## 重要な用語

受注No.	受注年月日	得意先コード	得意先コード	得意先名	所在地
0001	2011/1/10	A-1	A-1	南北電気	東京都...
0002	2011/1/10	B-1	A-20	いろは電子	兵庫県...
0003	2011/1/11	A-20	B-1	日本工業	大阪府...

↑  
主として検索に使うキーを「主キー」  
という

↑  
別の表の主キー  
に連結するための  
項目を「外部キー」  
という

### ■ その他:

表の中の行のことを「レコード」という

外部キーと次の表との連結が必ず存在することを「参照制約」という

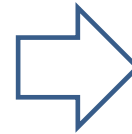
「インデックス」(索引)は検索を高速化するための工夫

## その他の重要な事柄

### ■ データの構造の表現

- 表の絵を書くのは面倒...

氏名	体重	身長
中村	65	164
田中	73	172



氏名	体重	身長
----	----	----

見出し行だけ書き表す  
ことが一般的

### ■ DBMSの役割 (P.282の表)

## データベースの機能のまとめ p.290

### ■ 選択

- 目的のレコード(行)を絞り込むこと

### ■ 射影

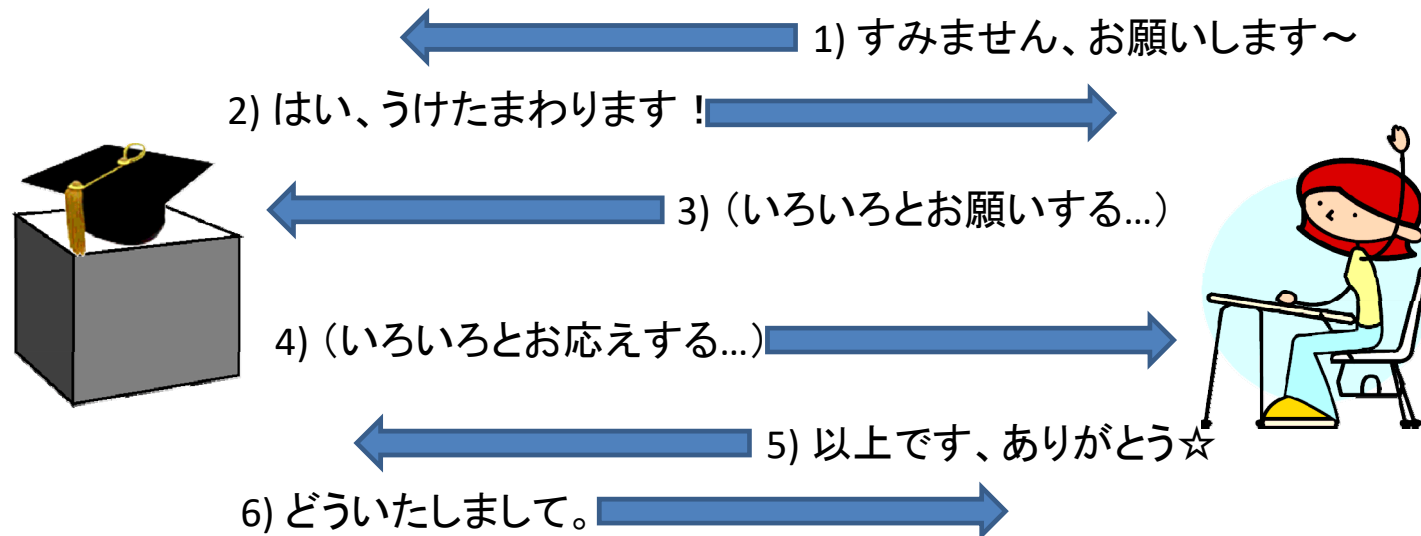
- 必要な項目(列)だけを取り出すこと

### ■ 結合

- 複数の表を関連付けて連結すること

# データベースとの付き合い方

## ■ トランザクション p.292 - (重要)



一連のまとまった問い合わせ処理を  
「トランザクション」という

## データベースの排他制御 p.292 -

### ■ 大勢同時の読み書きに対応するには

#### ● 例： 列車の指定席予約

- ◆ 「18-A」の席を予約する場合...
- ◆ 複数の窓口で複数の客がこの席を押さえようとしたら...
- ◆ → 同時に複数の要求がデータベースに来る！
- ◆ → 最初に「18-A」を押さえに来た人が占有！

#### ● 「ロック」の重要性

- ◆ ダブルブッキングの防止
- ◆ 2種類のロック
  - 占有ロック： 一番乗りの人だけが読み書きできるロック
  - 共有ロック： 一番乗りの人だけが変更できるロック
    - 他の人は見るだけならOK

## データベースの排他制御 p.292 -

### ■ 困った問題...

- 「18-A」の席を予約する場合の例
- 本当に同時に2人の人が予約に来たらどうなるか？
- → 「デッドロック」が起こってしまう...
- → 現在ではデッドロックを回避する方法がいくつか存在する



## データベースのバックアップに関すること p.295

### ■ ログファイル

- データベース更新処理の記録が克明に記録されている

### ■ ロールバック

- データベースの内容を以前の状態に戻す

### ■ ロールフォワード

- データベースの内容を進める(???)
- 障害発生時にバックアップからリストアした後に重要
- → ログファイルを元に再度処理をやり直してデータベースの状態を更新する