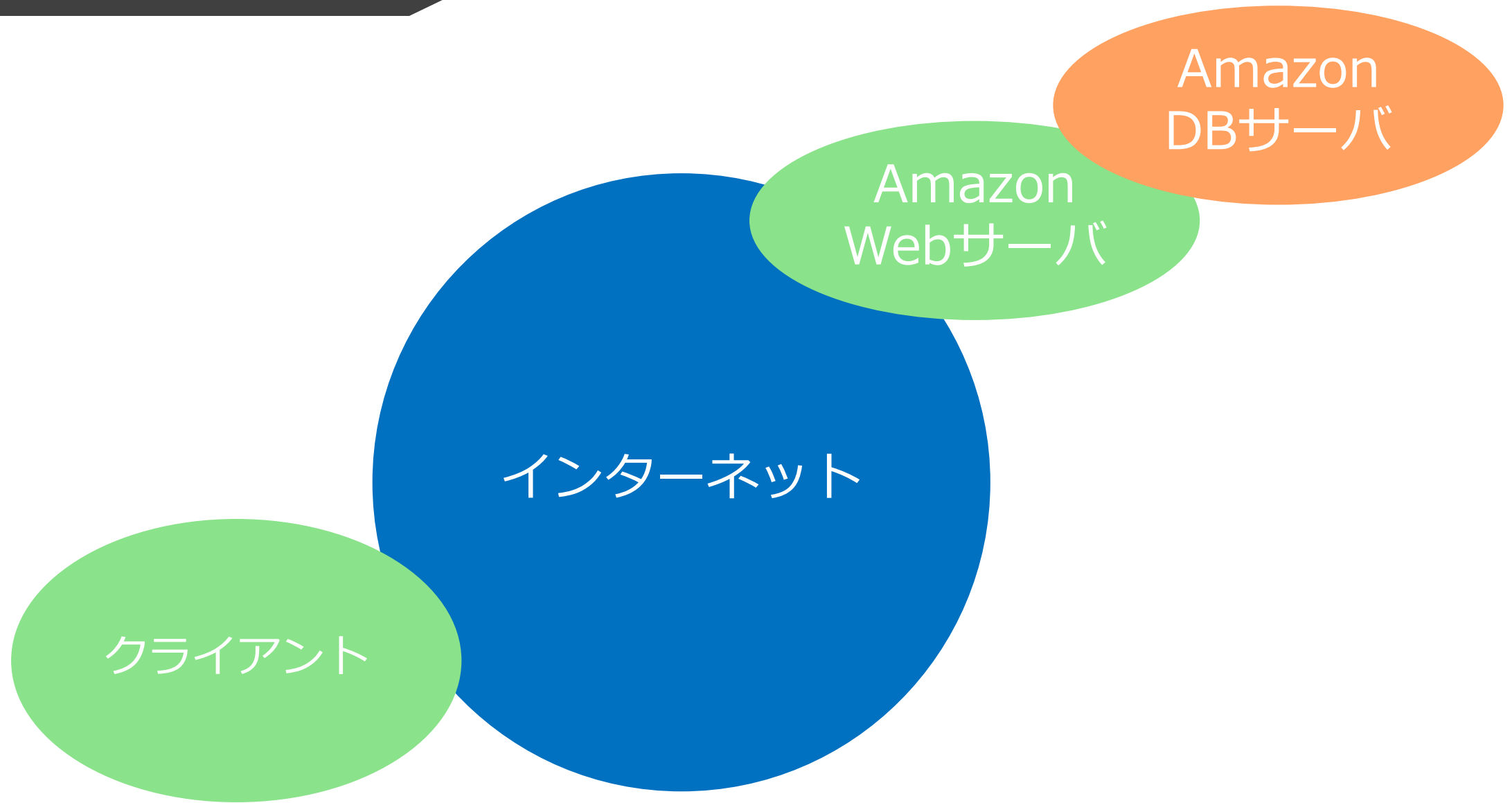


第23講 データベースとは

データベース、RDB
DBMS

Amazonを例に



1. データベースとは

「データベース」 Database

一定の形式で整理された
データの集まり

<目的>

- ・ 複数で共有、利用する
- ・ 検索、加工する

目的が達成できないものは
データベースではない

コンピュータ上で管理するデータを
データベースと呼ぶことが多いが
紙の上で管理する

「電話帳」や「住所録」なども
データベース

(1) 大量のデータをまとめて管理できる

management

(2) 目的のデータを簡単に探すことができる

search

(3) 簡単に編集して使うことができる

edit

データは活用して初めて価値が生まれる



データは資産となる

どうやって活用するか、が重要

DataBase Magement System

DB = 大量のデータを
自動的に処理できる

その管理を行うシステムが**DBMS**

DataBase Magement System

ex.

データの並べ替え（ソート）

データの検索（サーチ） 等

入力は人手に頼るところも多い

ネット経由でユーザから届くデータも多い

DBMS上でデータの追加や削除、
並べ替えなどを行うよう
コンピュータに命令するための言語

基本的に一行ずつ入力して処理させる

まとめて処理する

「ストアードプロシージャ」も使われる

DBMSが異なっても、基本はすべて共通

if文の条件分岐やfor文のループを
SQLと組み合わせて
通常のプログラミング言語のような感覚で
データベースを操作することができる
DBMSの機能

2. データベースの型の種類

基本的には全てが

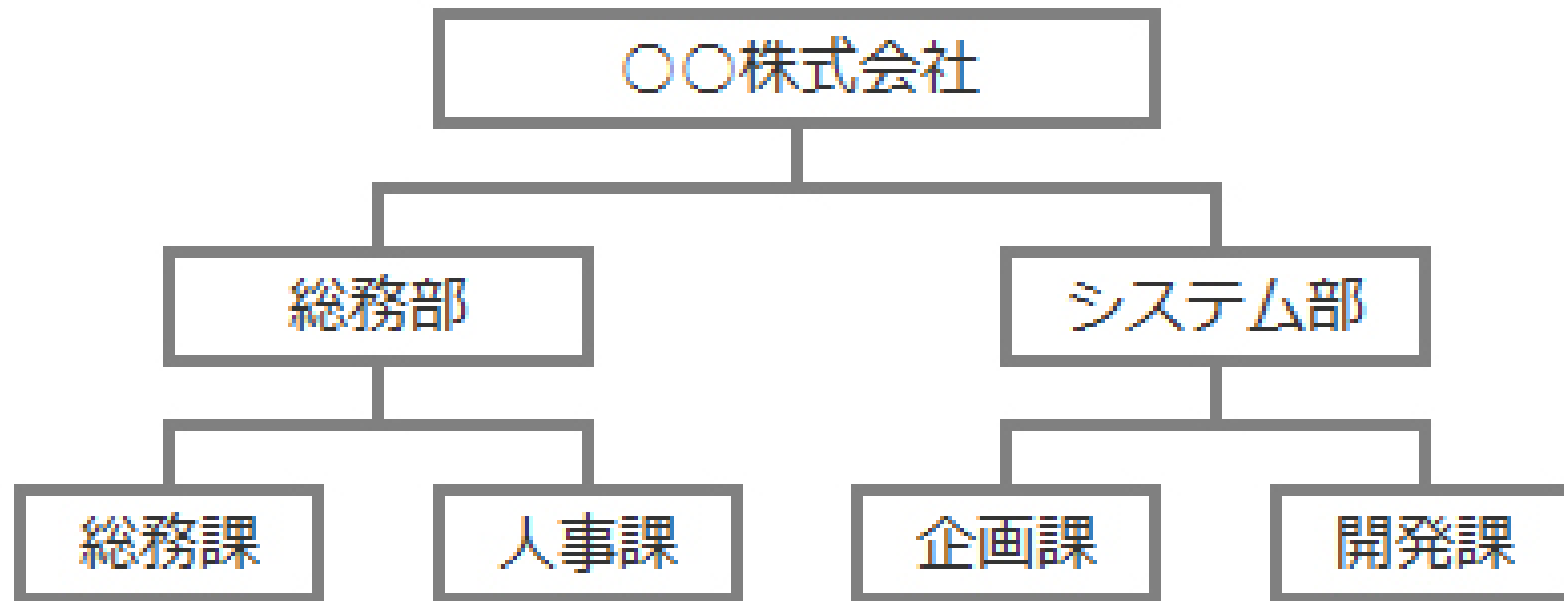
「リレーショナル型・データベース」

RDB

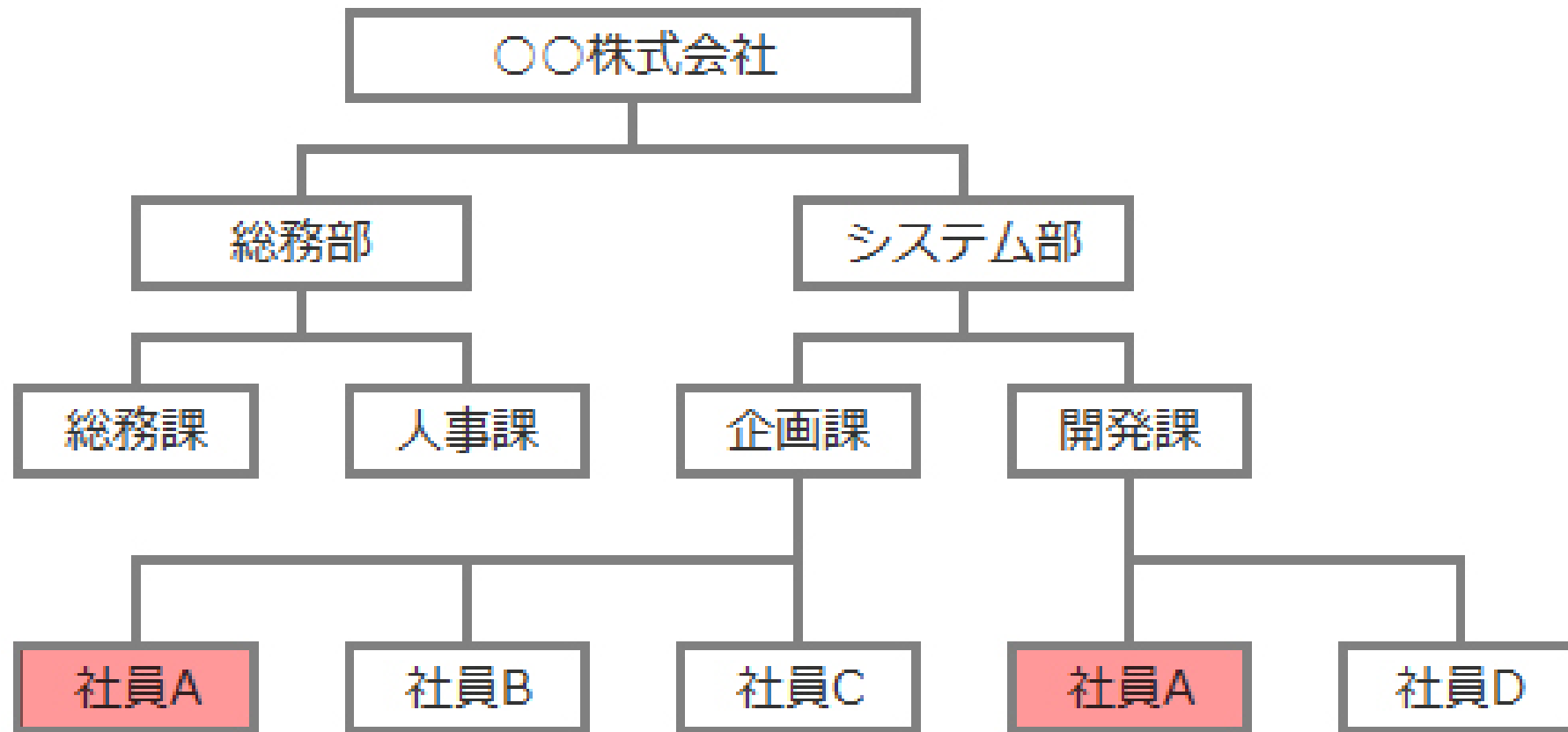
ほかにも「型」がある（参考程度に）

1:nのツリー構造

あるデータまでの経路が 1 つだけなので検索が高速



同じデータが**重複して登録**される可能性がある
(社員Aは二つの課に所属)



データの追加・削除により
ルート你再登録が必要になる
など、処理が煩雑になりやすい

ツリー構造（木構造）

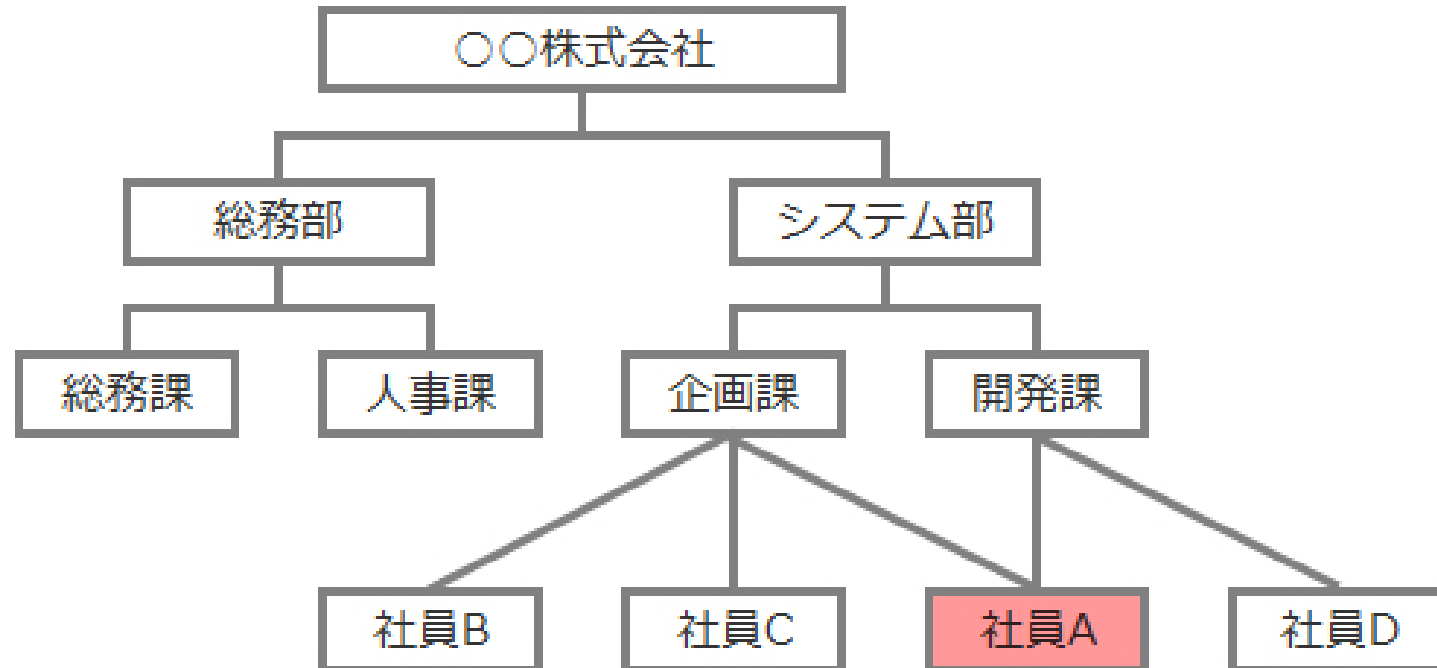
末端の「データ」、分岐点は「ノード」
結ぶ線は「リンク」

ネットワーク型データベース

n:mのネットワーク構造のデータベース

「網の目」のような構成

上位のノードが1つに限定されないので柔軟に処理が可能
より柔軟なリレーショナル型に移行した



テーブル構造（表のイメージ）のデータベース

RDB

（リレーション＝関係）

リレーショナル型データベース

行と列からなる「表」のイメージ
列には**重複しない項目名**が並ぶ
そこにデータを**行単位で追加**していく

社員番号	氏名	主要部署	兼務部署
0001	社員A	企画課	開発課
0002	社員B	企画課	
0003	社員C	企画課	
0004	社員D	開発課	

行 row と 列 column

行			列
社員番号	氏名	主要部署	
0002	社員B	企画課	
0003	社員C	企画課	
0004	社員D	開発課	

リレーショナル型データベース

全体が「テーブル」（表）

1行＝レコード（組、タプル）

1列＝フィールド（属性）

社員番号	氏名	主要部署	兼務部署
0001	社員A	企画課	開発課
0002	社員B	企画課	
0003	社員C	企画課	
0004	社員D	開発課	

メリット

データの重複が避けられる

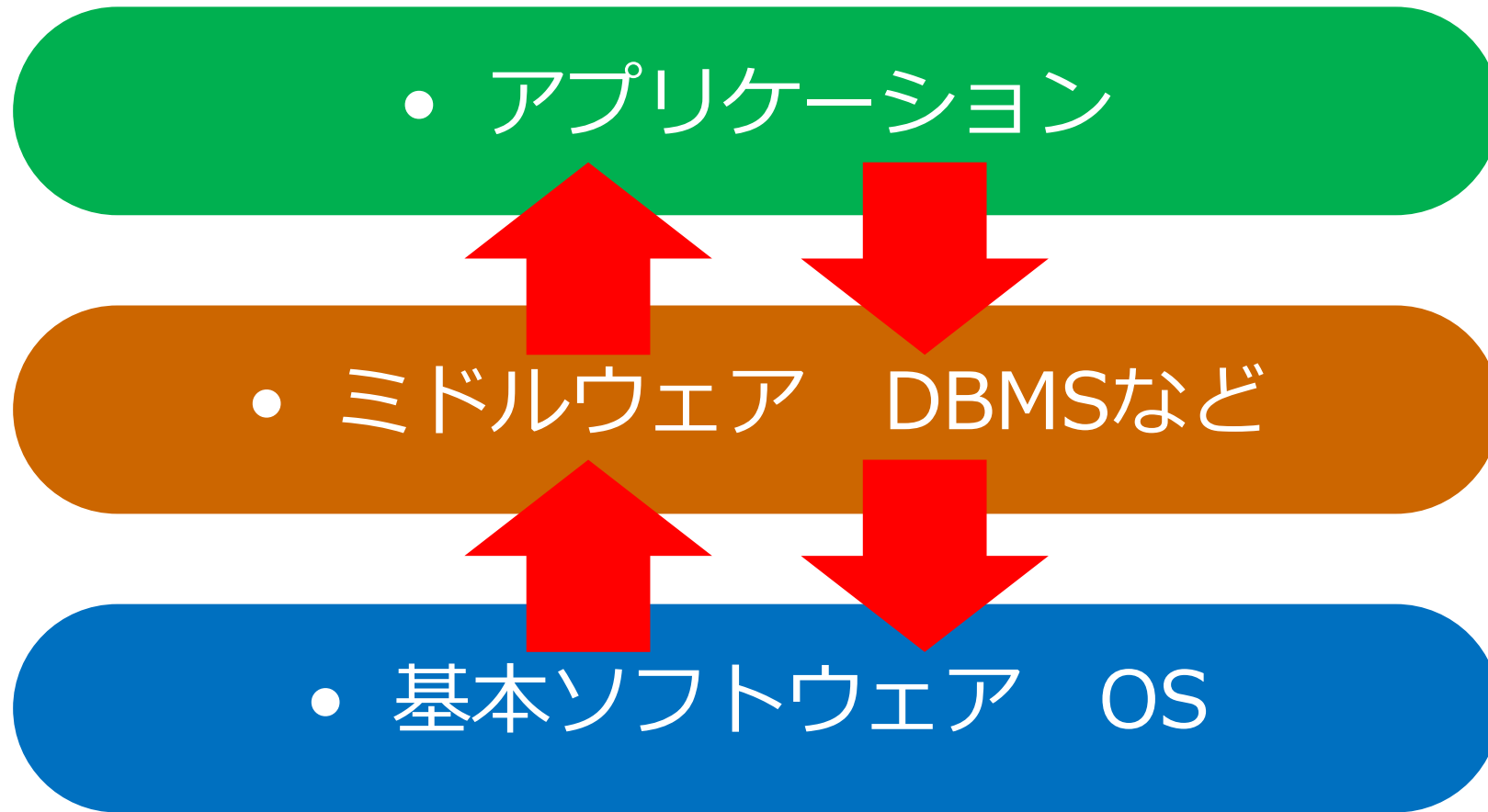
検索能力が高い

**データの追加・削除は
レコード（行）単位**

3 . DBMSとSQL

データベースを効率よく管理、
運用するソフトウェア

RDBの場合は、RDBMSとも呼ぶ



DBMS

DBを利用したいアプリケーションのために、データベース機能を提供する

データベースへの
データ追加、更新、削除を管理する

人手で行うのは危険



SQLで操作

4. 正規化

蓄積されているデータに
矛盾、重複が発生しないように
表を「最適化」すること

社員テーブル

社員番号	氏名	部署
2009001	田中一郎	営業部
2009002	山本二郎	開発部
2009003	佐藤三郎	営業部
2009004	小山四郎	総務部

「営業部」が「法人営業部」
に変更になったら



「営業部」を
全て書き換えなければならない

社員テーブル

社員番号	氏名	部署
2009001	田中一郎	01
2009002	山本二郎	02
2009003	佐藤三郎	01
2009004	小山四郎	03

部署テーブル

部署ID	部署名
01	営業部
02	開発部
03	総務部

部署名だけの
変更でOK

テーブルを分けると…



テーブルの数が増え…



一つ一つのテーブルだけでは
あまり意味がなくなる



複数のテーブルを連携して処理する

5. 関係演算

①テーブルから

特定のレコードを取り出す

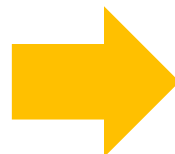
②テーブルをくっつけて

新しいテーブルを作る などの操作

1. 選択
2. 射影 (しゃえい)
3. 結合

社員テーブル

社員番号	氏名	部署ID
2009001	田中一郎	01
2009002	山本二郎	02
2009003	佐藤三郎	01
2009004	小山四郎	03



社員番号	氏名	部署
2009001	田中一郎	営業部
2009003	佐藤三郎	営業部

「営業部」だけ抜き出した

社員テーブル

社員番号	氏名	部署ID
2009001	田中一郎	01
2009002	山本二郎	02
2009003	佐藤三郎	01
2009004	小山四郎	03

「社員番号」 「氏名」 を
抜き出した



社員番号	氏名
2009001	田中一郎
2009002	山本二郎
2009003	佐藤三郎
2009004	小山四郎

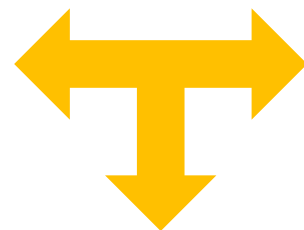
関係演算 「結合」

社員テーブル

社員番号	氏名	部署ID
2009001	田中一郎	01
2009002	山本二郎	02
2009003	佐藤三郎	01
2009004	小山四郎	03

部署テーブル

部署ID	部署名
01	営業部
02	開発部
03	総務部



社員番号	氏名	部署ID	部署名
2009001	田中一郎	01	営業部
2009002	山本二郎	02	開発部
2009003	佐藤三郎	01	営業部
2009004	小山四郎	03	総務部

第23講 データベースの基礎

- データベースとは
- RDBとは
- DBMSとは
- 正規化
- 関係演算