

リレーショナルデータベース入門

SRA OSS, Inc. 日本支社

「データベース」とは？

- データ(Data)の基地(Base)
- 実世界のデータを管理する**いれもの**

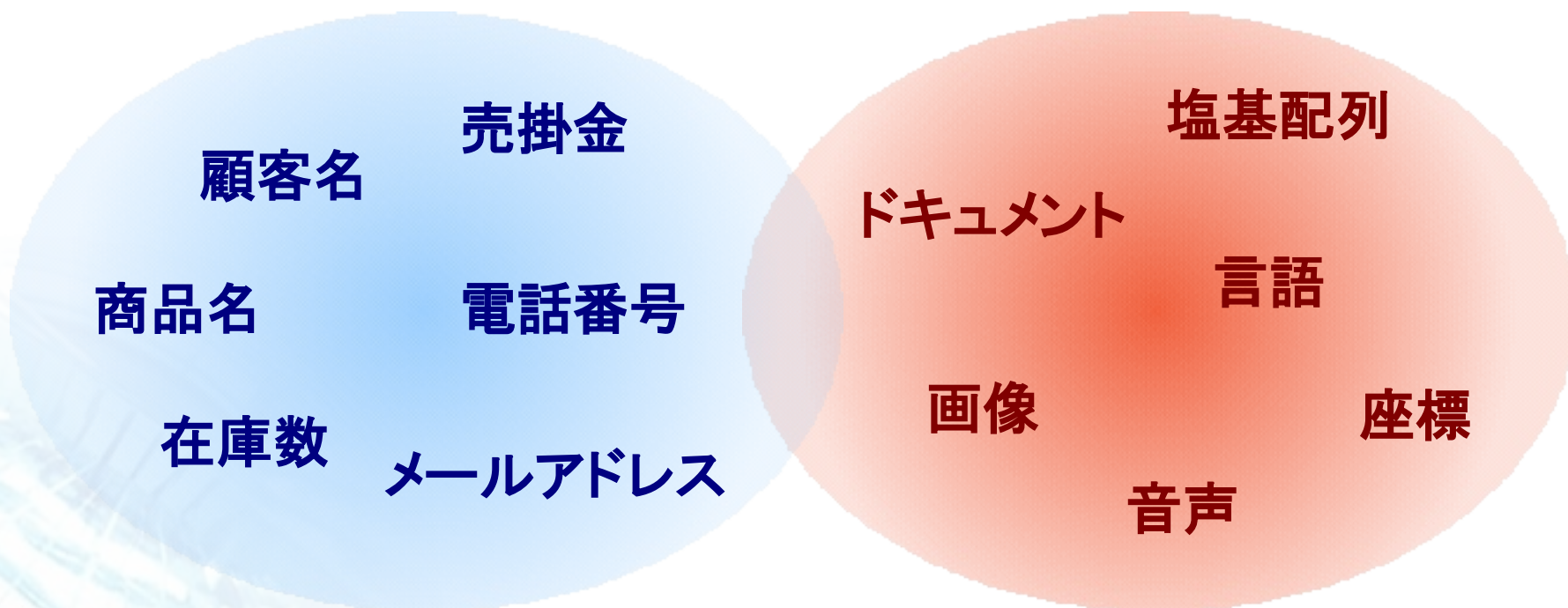
例えば・・・

電話帳 辞書 メーカー 検索エンジン

もデータベースである

データ

- データベースに保存するデータ



- データを加工して必要な情報を取り出す

「DBMS」とは？

- データベースをコンピュータ上で管理するシステム
- 「システム」としていろいろな機能が提供される

データの操作(収集、蓄積、加工、提供)

トランザクション(操作一貫性)の管理

同時実行制御

セキュリティ管理

障害回復機能

DBMS のメリット

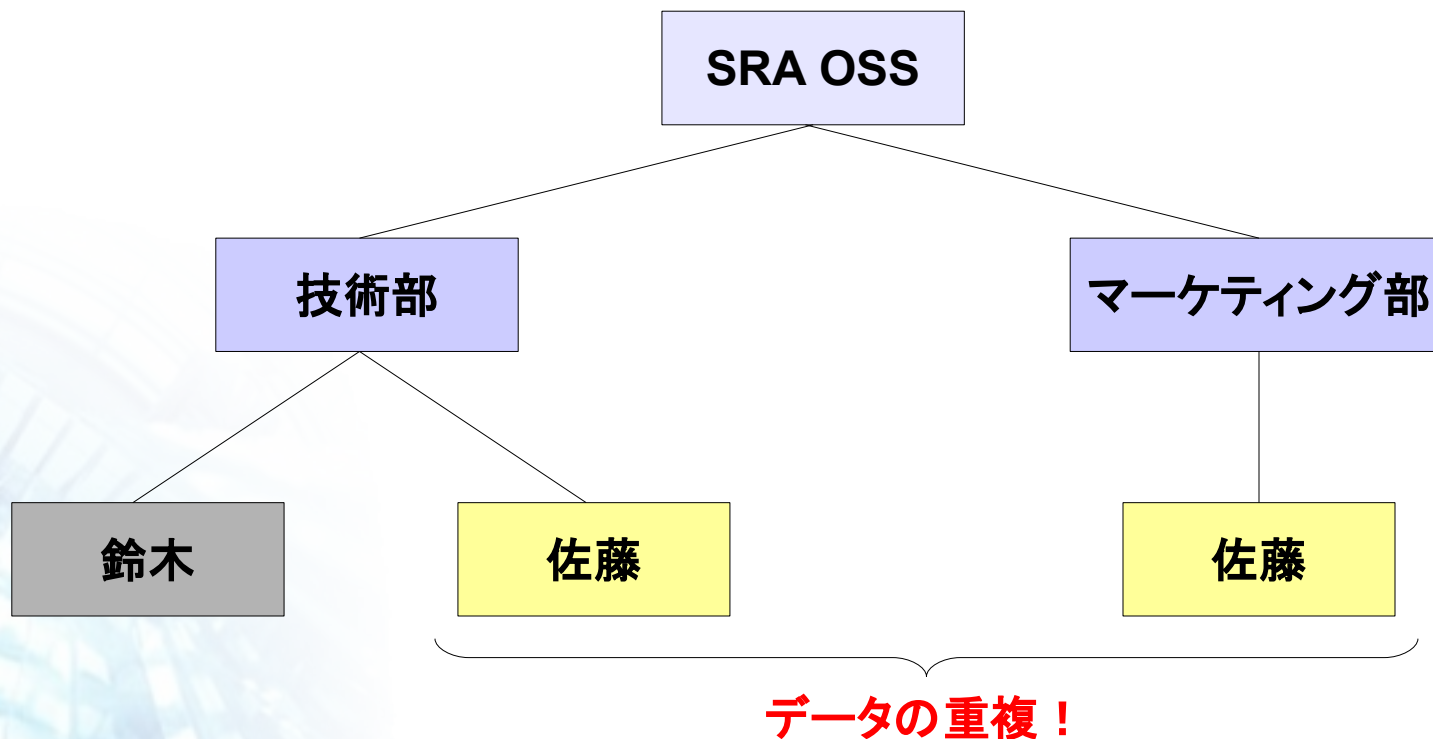
- ユーザはデータの状態を気にする必要がない
 - データ構造
 - 物理的なデータ配置
 - OS、ファイルシステム、メモリ、I/O、etc...
- アプリケーションとデータを分離できる
 - アプリケーション内でデータを管理しようとする、上記のデータの状態(物理的構造)を理解する必要がある

データモデル

- DBMSに格納されるデータの論理構造
 - ハイアラキカルデータモデル
 - ネットワークデータモデル
 - リレーショナルデータモデル

ハイアラキカル(階層型)データモデル

- データを木構造で表現



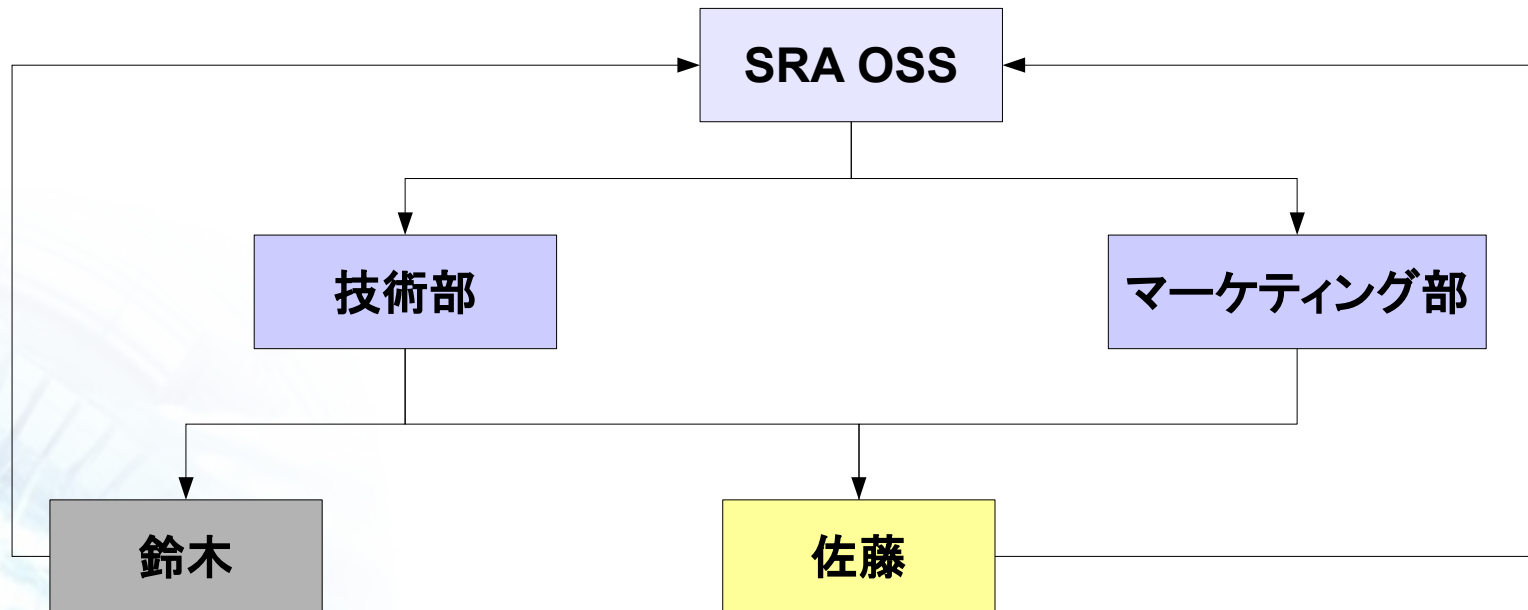
ハイアラキカル(階層型)データモデル

- 目的のデータに到達するパスは1つのみ
 - データ抽出には階層構造を理解する必要がある

- 親と子は「1対N」の関係にあるので…
 - 複数の親に所属する子データが重複してしまう
 - 更新は重複するデータ全てを書き換える必要がある

ネットワークデータモデル

- データを「N対N」の構造で表現

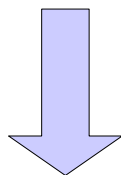


ネットワークデータモデル

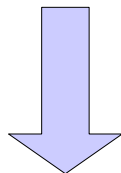
- 目的のデータに到達するパスは複数あるが・・・
 - 親子関係を表現するポインタや多重リンクが複雑
 - データ抽出にはネットワーク構造を理解する必要がある

階層型、ネットワーク型ともに

- 物理構造と論理構造に密接な繋がりがある



- 構造が変わるとデータ抽出方法が変わる



- アプリケーションに変更が必要になる

そこで...

リレーショナルデータモデル

- 数学の集合論に基づいてデータを表現

SRA OSS 組織テーブル

部署番号	所属社員
1	1
1	2
2	2

社員テーブル

社員番号	社員名
1	鈴木
2	佐藤

部署テーブル

部署番号	部署名
1	技術部
2	マーケティング部

データとデータ間に**関連**を持たせ、2次元の表で表現する

リレーショナルデータモデル

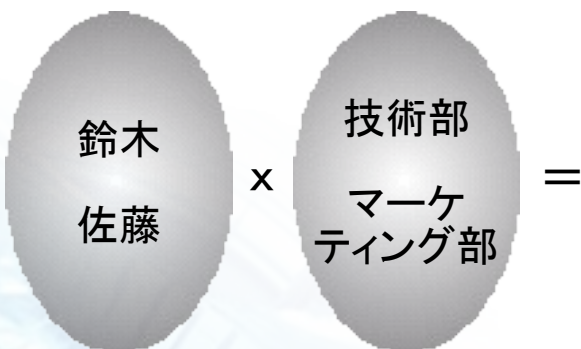
- 階層構造のようなデータの重複を防げる
- ネットワーク構造よりも直感的に構造を理解できる
- 物理構造と論理構造が完全に独立している
- データ操作方法が論理構造に縛られない
- ただしネットワーク構造に比べて処理が遅い

リレーショナルデータモデルの歴史

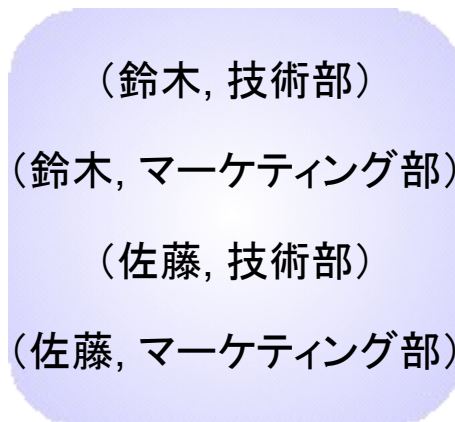
- 1970年に E.F.Codd 氏が提案
- 当時はコンピュータの性能が悪く、リレーショナルデータベースは実用化が難しかった
- 1980年代からリレーショナルデータモデルの利点が認知されだし、実用にも耐えられるものが出てくる
- 現在では一番普及しているデータモデル
 - PostgreSQL, Oracle, DB2, MySQL, SQL Server...

リレーショナルデータベースモデルの用語

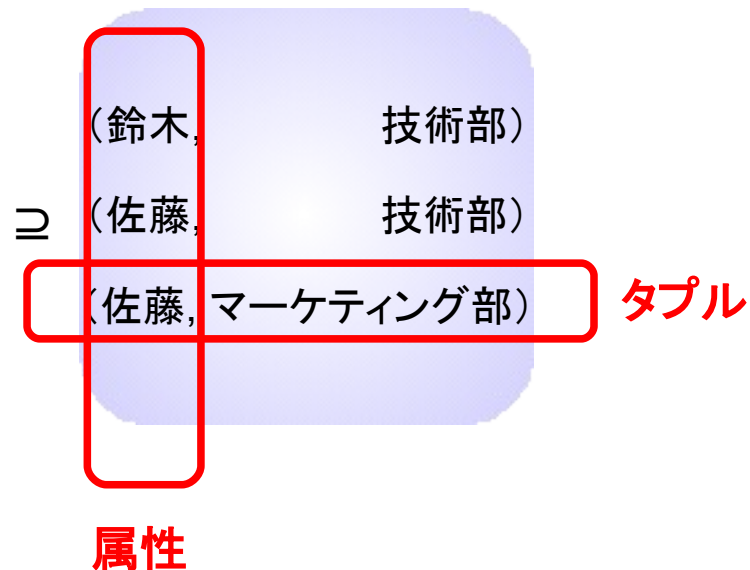
社員ドメイン × 部署ドメイン



直積



リレーション



リレーショナルデータベース

- リレーション → 表 / テーブル
- 属性 → 列 / カラム
- タプル → 行 / レコード
- ドメイン → データ型

外部参照

部署番号	所属社員
1	1
1	2
2	2

社員番号	社員名
1	鈴木
2	佐藤

外部キー

主キー

リレーショナル代数

- リレーショナルデータベースを操作する言語
- リレーショナル代数は5つの演算から成り立つ

和集合

差集合

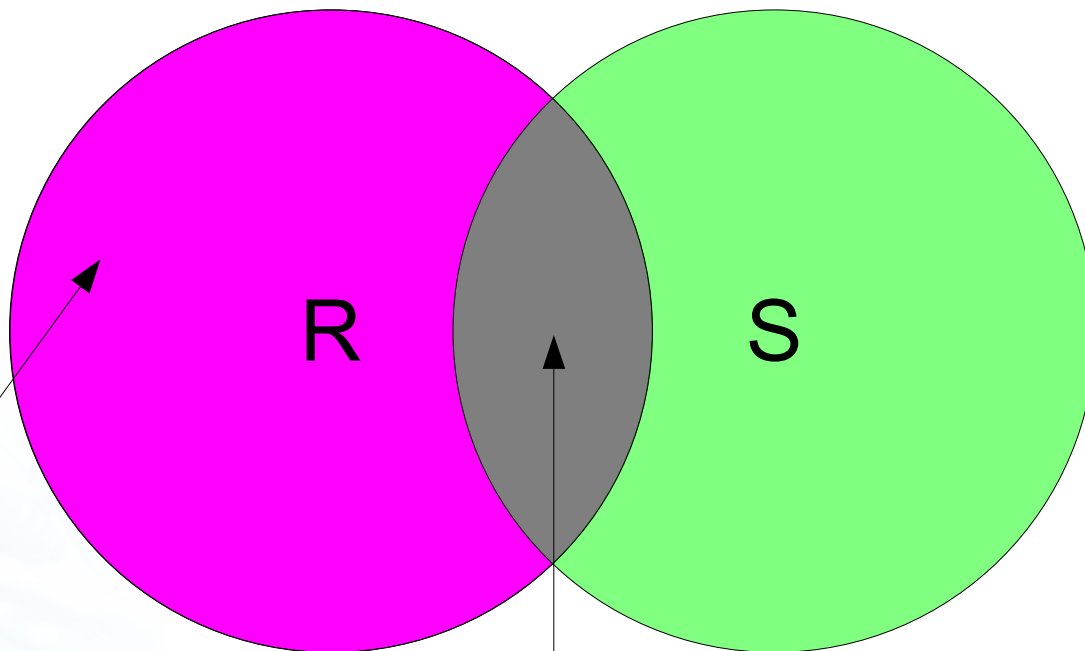
直積集合

射影

選択

リレーショナル代数

和集合 (全体)
 $R \cup S$



差集合
 $R - S$

積集合
 $R \cap S = R - (R - S)$

リレーショナル代数

R

部署番号	所属社員
1	1
1	2
2	2

S

社員番号	社員名
1	鈴木
2	佐藤

直積集合
R X S

部署番号	所属社員	社員番号	社員名
1	1	1	鈴木
1	1	2	佐藤
1	2	1	鈴木
1	2	2	佐藤
2	2	1	鈴木
2	2	2	佐藤

全てのタプルの組み合わせ

リレーショナル代数

T

部署番号	所属社員	社員番号	社員名
1	1	1	鈴木
1	1	2	佐藤
1	2	1	鈴木
1	2	2	佐藤
2	2	1	鈴木
2	2	2	佐藤

射影

社員名
鈴木
佐藤
鈴木
佐藤
鈴木
佐藤

縦・横方向の切り取り

選択

部署番号	所属社員	社員番号	社員名
1	1	1	鈴木
1	2	2	佐藤
2	2	2	佐藤

DBMS の機能 (1)

データの操作

SQL

- リレーショナルデータベースを操作する言語
- ISO標準規格
 - 多くのRDBMSで共通規格として取り入れられている
- DDL テーブルなどの構造を定義する
- DML データを操作する
- DCL データ操作の確定、制御などを行う

DML (Data Manipulation Language)

- データの検索、演算結果の抽出
 - **SELECT** [カラム, ...] FROM [テーブル, ...] WHERE
- データの挿入
 - **INSERT** INTO [テーブル] VALUES (値, ...)
- データの更新
 - **UPDATE** [テーブル] SET [列=値, ...] WHERE
- データの削除
 - **DELETE** FROM [テーブル] WHERE

SELECTの種類

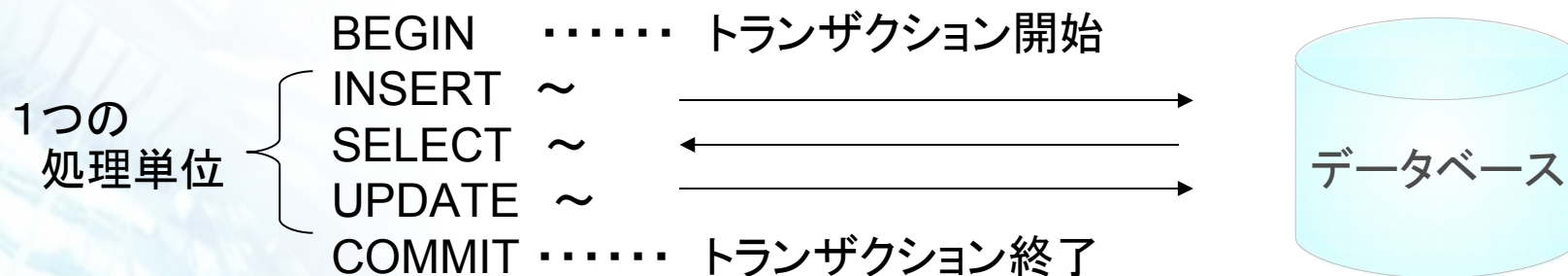
- Simple Query
 - SELECT ... FROM ... WHERE ...
- Join Query (結合)
 - SELECT ... FROM t1 JOIN t2 ON ... WHERE ...
- Nested Query (サブクエリー)
 - SELECT ... FROM ... WHERE ... IN (SELECT ...)
 - SELECT ... FROM (SELECT ...) WHERE ...

DBMS の機能 (2)

トランザクション

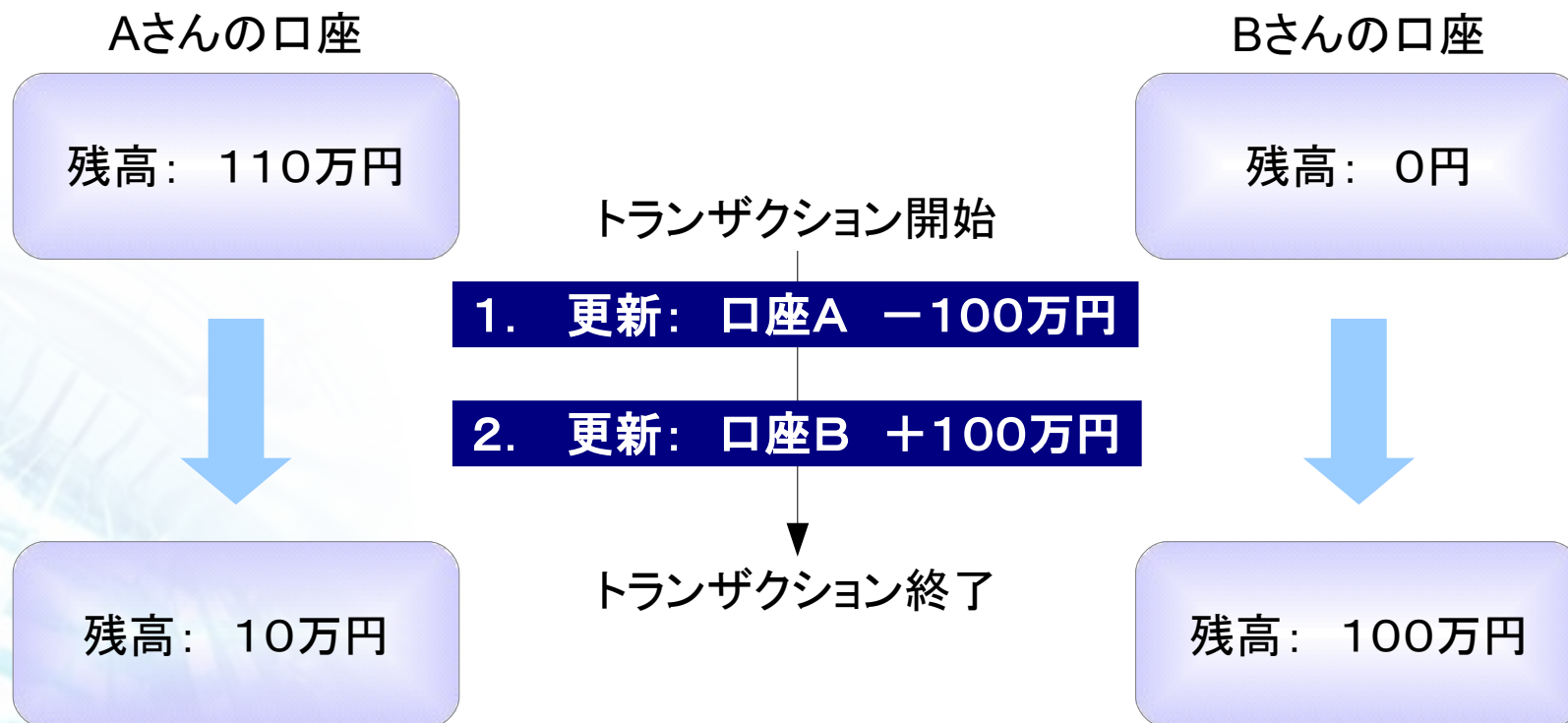
トランザクションとは？

- 複数のSQL文を、1つの処理単位にまとめたもの
- 1トランザクション内のSQLは全て成功か、全て失敗かのいずれかであることが保証される



トランザクションを利用した例 1

Aさんの口座から、Bさんの口座へ100万円振り込む



トランザクションを利用しなかった例

Aさんの口座から、Bさんの口座へ100万円振り込む

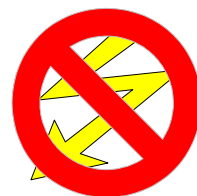
Aさんの口座

残高: 110万円

Bさんの口座

残高: 0円

1. 更新: 口座A -100万円



停電!

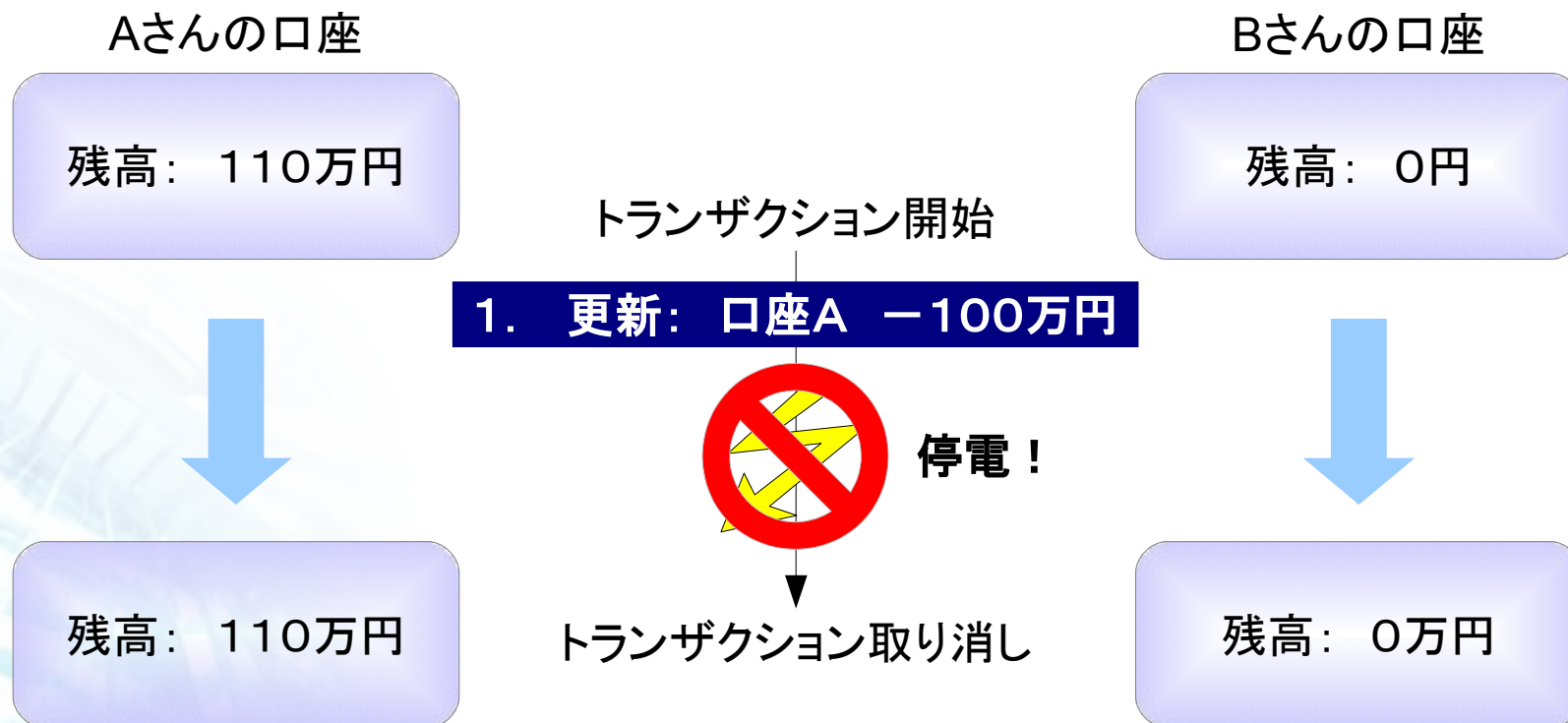
お金が消えた!

残高: 10万円

残高: 0万円

トランザクションを利用した例 2

Aさんの口座から、Bさんの口座へ100万円振り込む



DBMS の機能 (3)

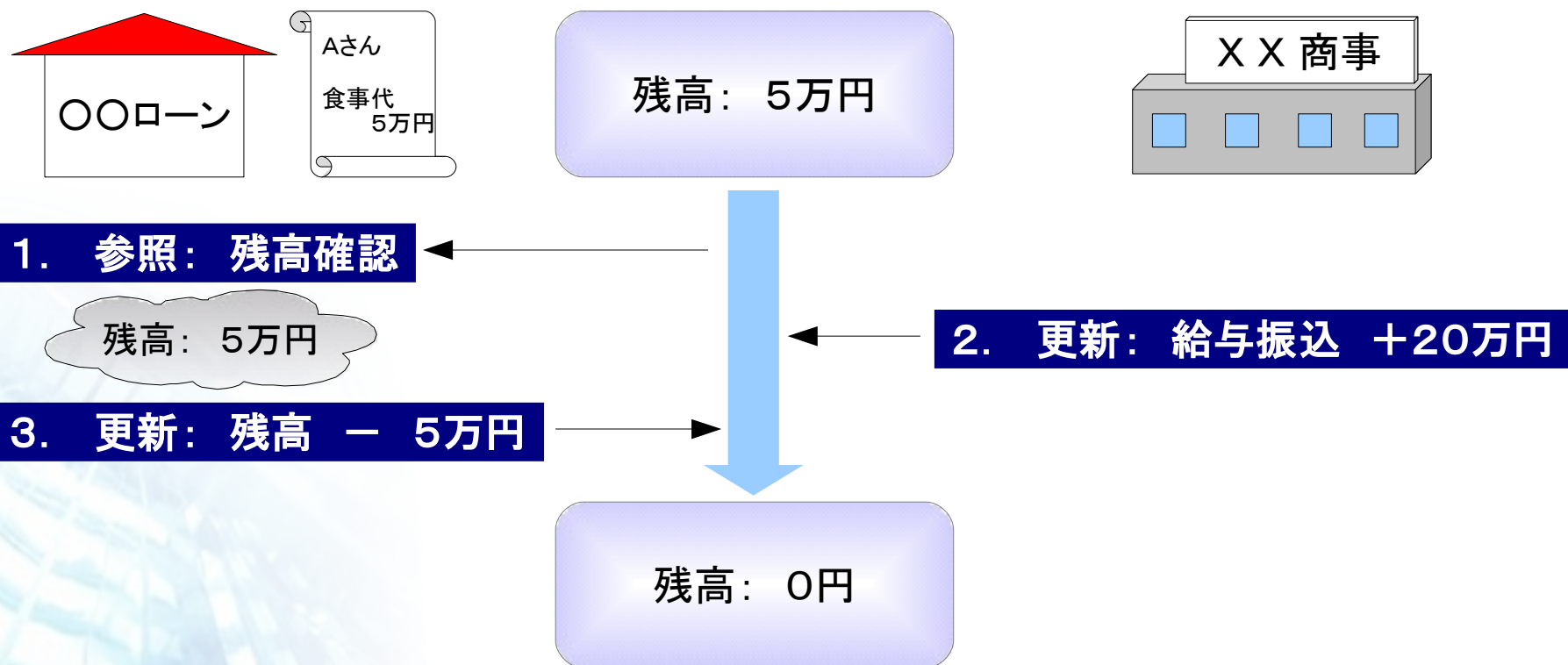
同時実行制御

同時実行制御とは？

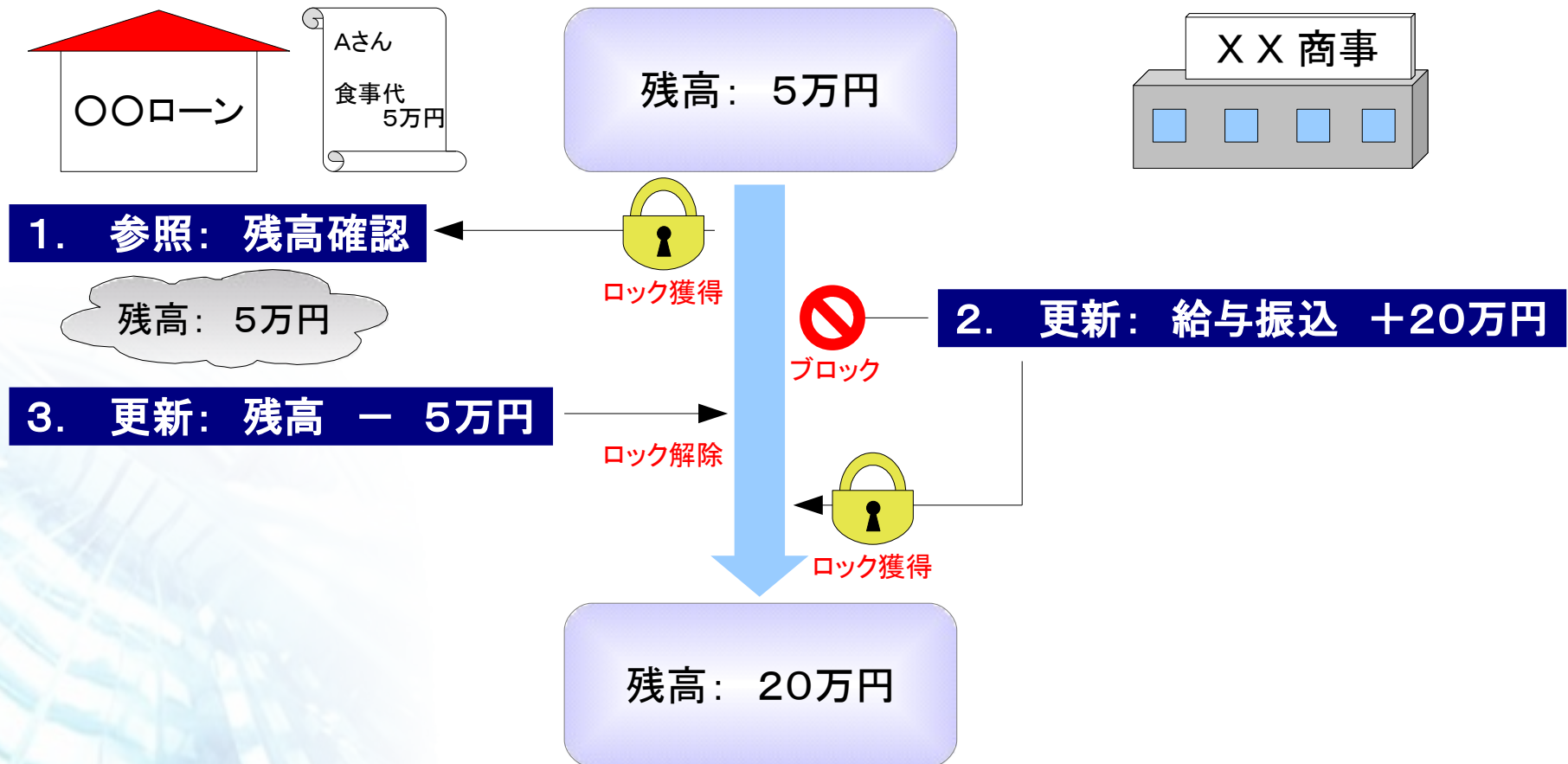
- 複数のユーザが同じデータを操作した際に、矛盾が生じないようにする
 - ロックによる制御が暗黙的に行われる
 - 1行をロックする「行ロック」
 - テーブル全体をロックする「テーブルロック」
- がある

同時実行制御をしていない例

給料日前のAさんの口座



同時実行制御をしている例



DBMS の機能 (4)

セキュリティ管理 障害回復機能

... はこの後の「PostgreSQL 入門」で

ご清聴ありがとうございました