

# 第9回 今日の目標

## § 3.4 データベース

- リレーショナルデータベースの特徴を示せる
- ロールバックやコミットを説明できる
- データベースのACID特性を説明できる
- デッドロックについて説明できる
- 関係代数について説明できる
- リレーショナルのキーについて説明できる
- SQLについて例示できる

## データとは

人が扱いやすいように表現した基礎となる事実

例：納品書の場合

受注日付、納品先、納品日付、商品名、  
数量、単価、金額など

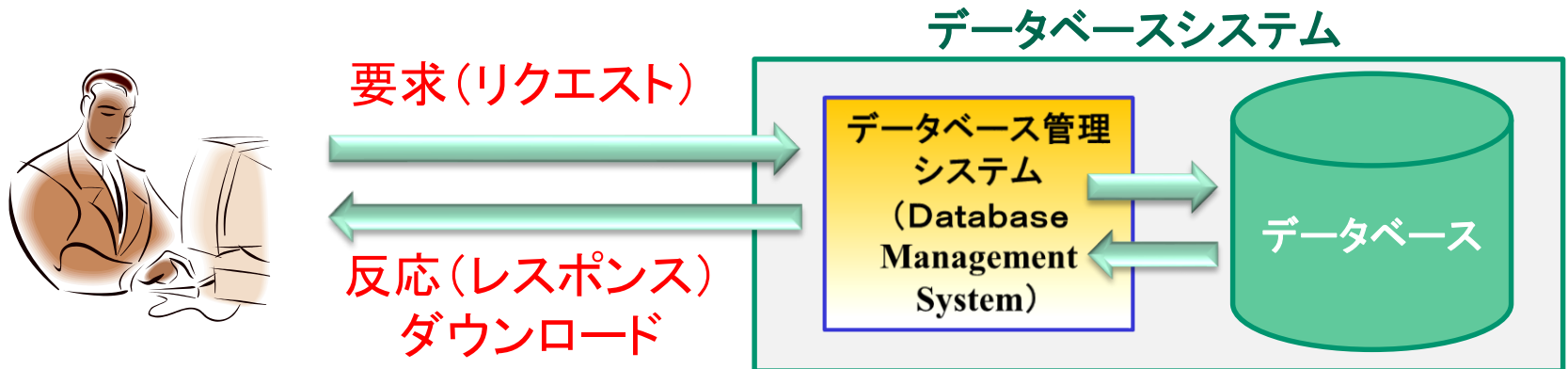
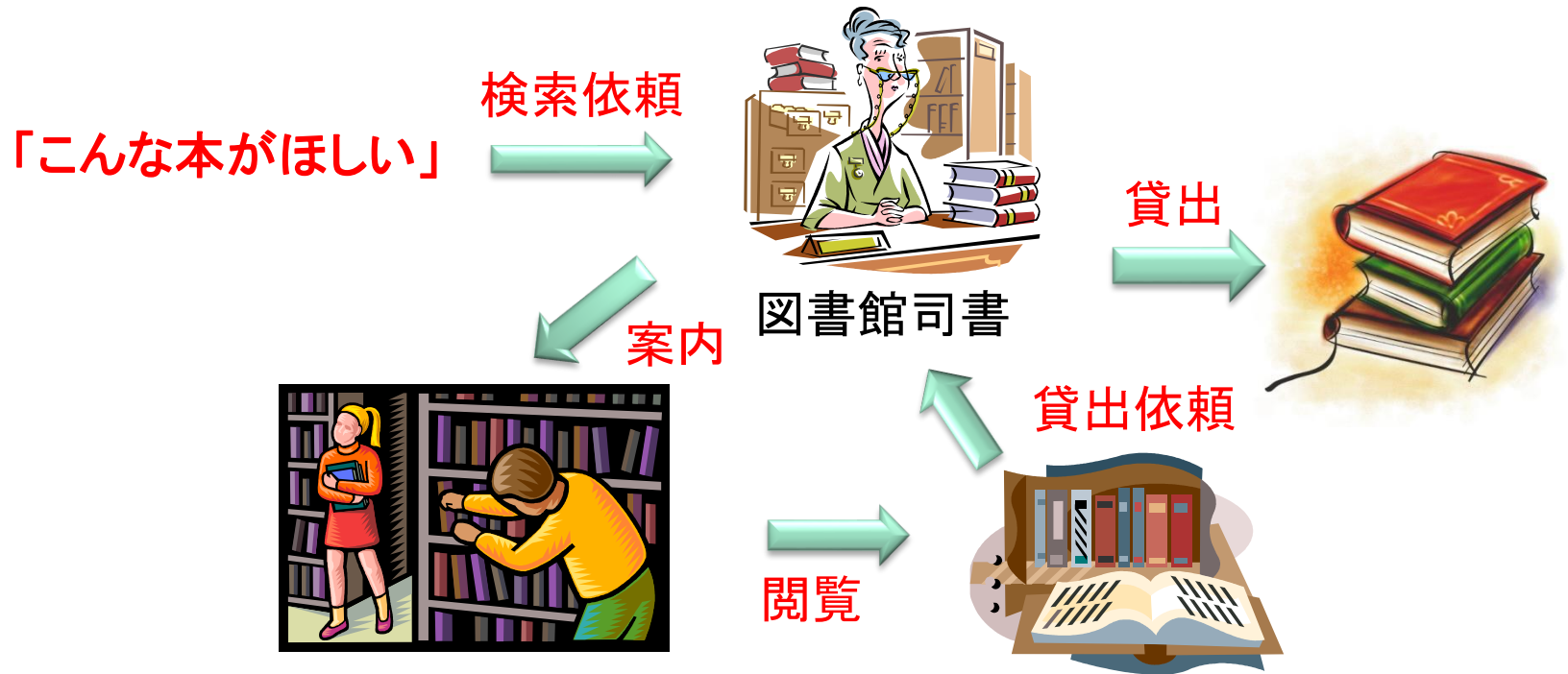
## データベースとは

広義 ⇒ 関連するデータの集まり

狭義 ⇒ データを管理・保守するためのしくみが備わっているシステム、もしくはそのシステムに格納されているデータの集まり

データを一定のルールで蓄積し、必要に応じて取り出せるようにしたもの

# データベースのしくみ = 図書館のしくみ



# データモデルについて

現実社会のデータ

↓  
データモデリング (コンピュータ上で扱える形に置換える作業)

データモデル

(コンピュータ上で扱える形に置換えられたデータ)

Step 1: 概念データモデル

↓ (現実社会の構造をデータ化して記述したもの)

Step 2: 論理データモデル

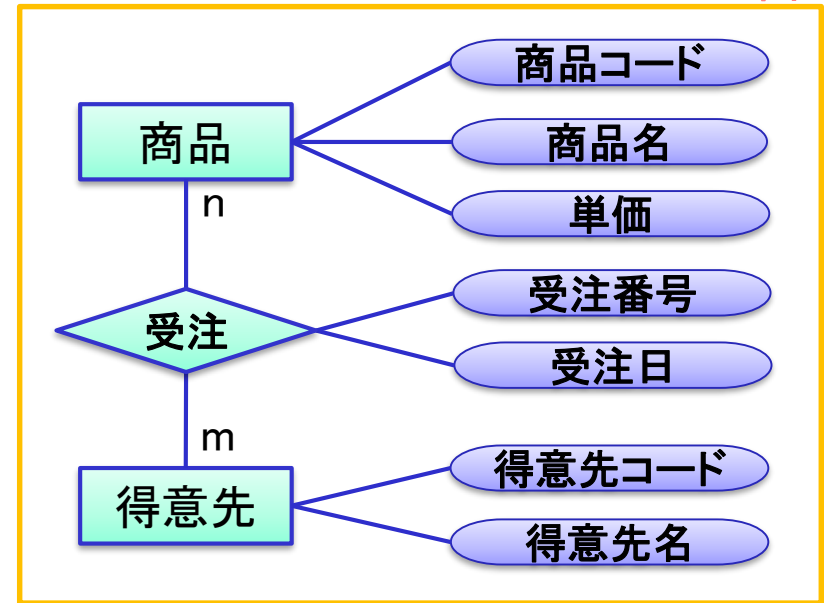
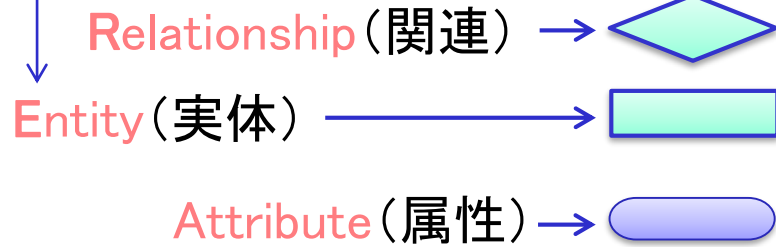
(コンピュータで扱いやすいデータ表現にしたもの)

# 概念データモデル

ER図

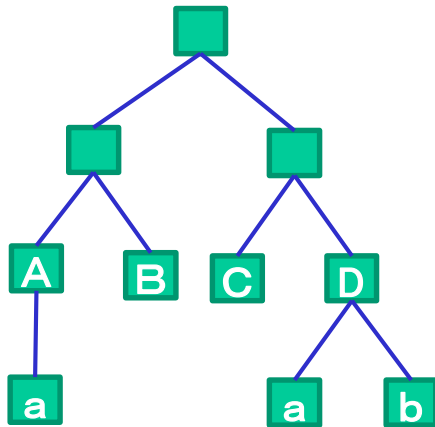
ER図による表現

(記号)

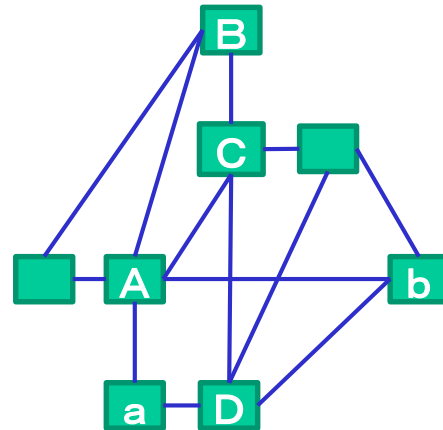


# 論理データモデル

## 階層型データモデル



## ネットワーク型データモデル



## リレーショナル型データモデル

コード	名前	性別	血液型	誕生日	部門コード
101	山口	m	A	1972/02/01	10
102	鈴木	m	AB	1974/09/10	20
103	佐藤	f	B	1976/04/21	10
104	小林	m	O	1978/12/24	30
105	佐々木	f	A	1980/07/04	30

コード	名称
10	総務部
20	営業部
30	開発部

データの冗長化が発生！

アプリのデータ構造依存が大きい！

# リレーショナルデータベース

社員テーブル ←

テーブル(表)名

フィールド(列)名

コード	名前	性別	血液型	誕生日	部門コード
101	山口	m	A	1972/02/01	10
102	鈴木	m	AB	1974/09/10	20
103	佐藤	f	B	1976/04/21	10
104	小林	m	O	1978/12/24	30
105	佐々木	f	A	1980/07/04	30

部門テーブル

コード	名称
10	総務部
20	営業部
30	開発部

フィールド(列)

レコード(行)

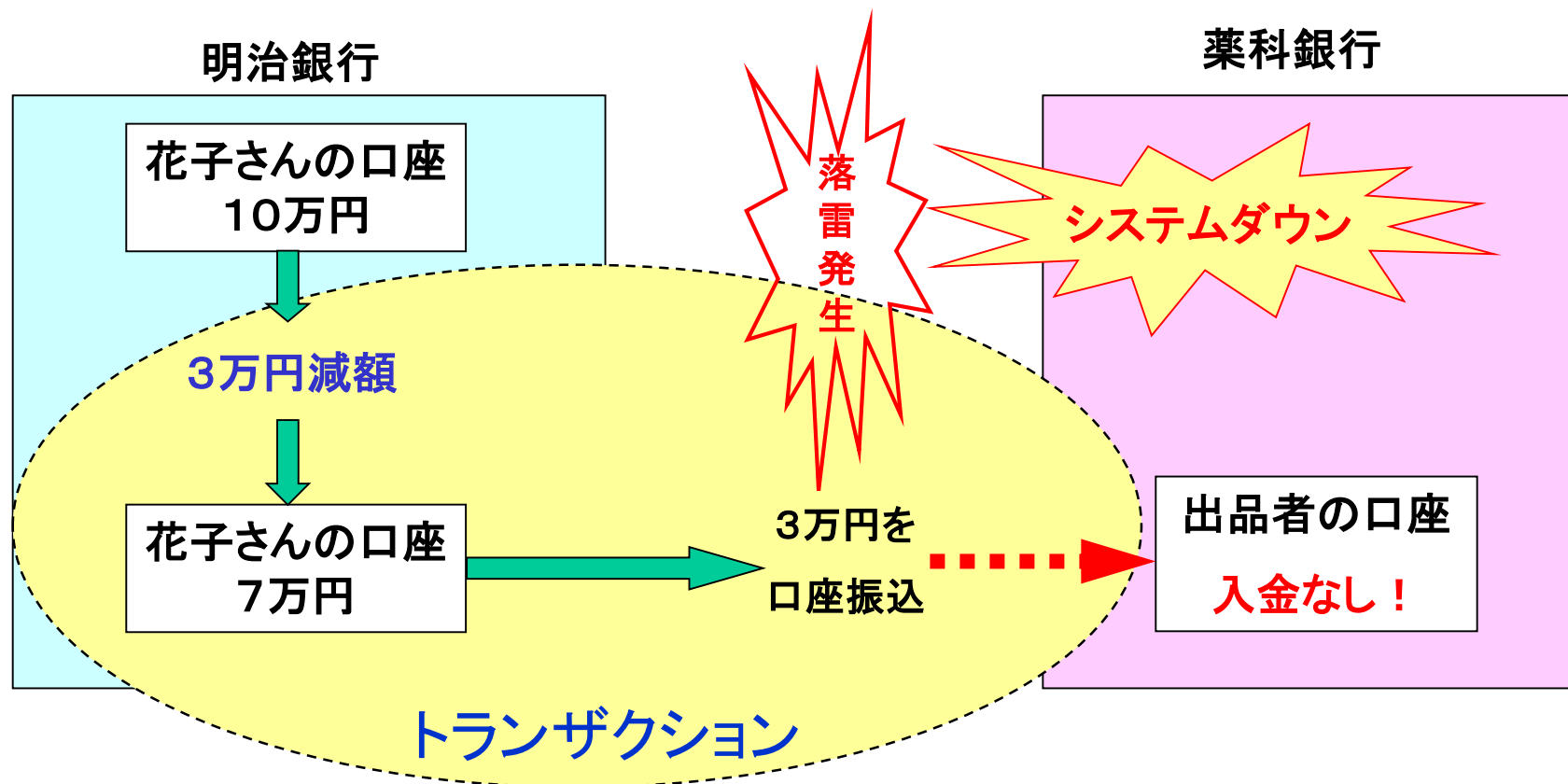
## リレーショナルデータベースの特徴

発明者: 1969年 エドガー・F・コッド博士

- ・複数の2次元の表(テーブル)から構成
- ・テーブルは複数のレコードから成り、レコードは複数のフィールドから成る
- ・テーブル(データ構造)の変更に伴うアプリケーションの変更修正が容易
- ・データ構造が単純でわかりやすいため、アプリケーション作成が容易
- ・複数のテーブルに同じ値を持つフィールドによって、レコード間の関連付けが可能
- ・現在の主流のデータモデル 例) 商用 ⇒ Oracle, Microsoft SQL Server, IBM DB2, Sybase, Microsoft Access  
フリー ⇒ PostgreSQL, MySQL, SQLite

# トランザクションについて

事例) 花子さんはオークションサイトでオークションに参加して、めでたく希望の商品を落札した。3万円の支払いは、銀行振込みで行おうとしたのだが...



トランザクション(Transaction)： データベース上の処理の単位

ロールバック(Rollback)：トランザクションのキャンセル

コミット(Commit)：トランザクションの確定

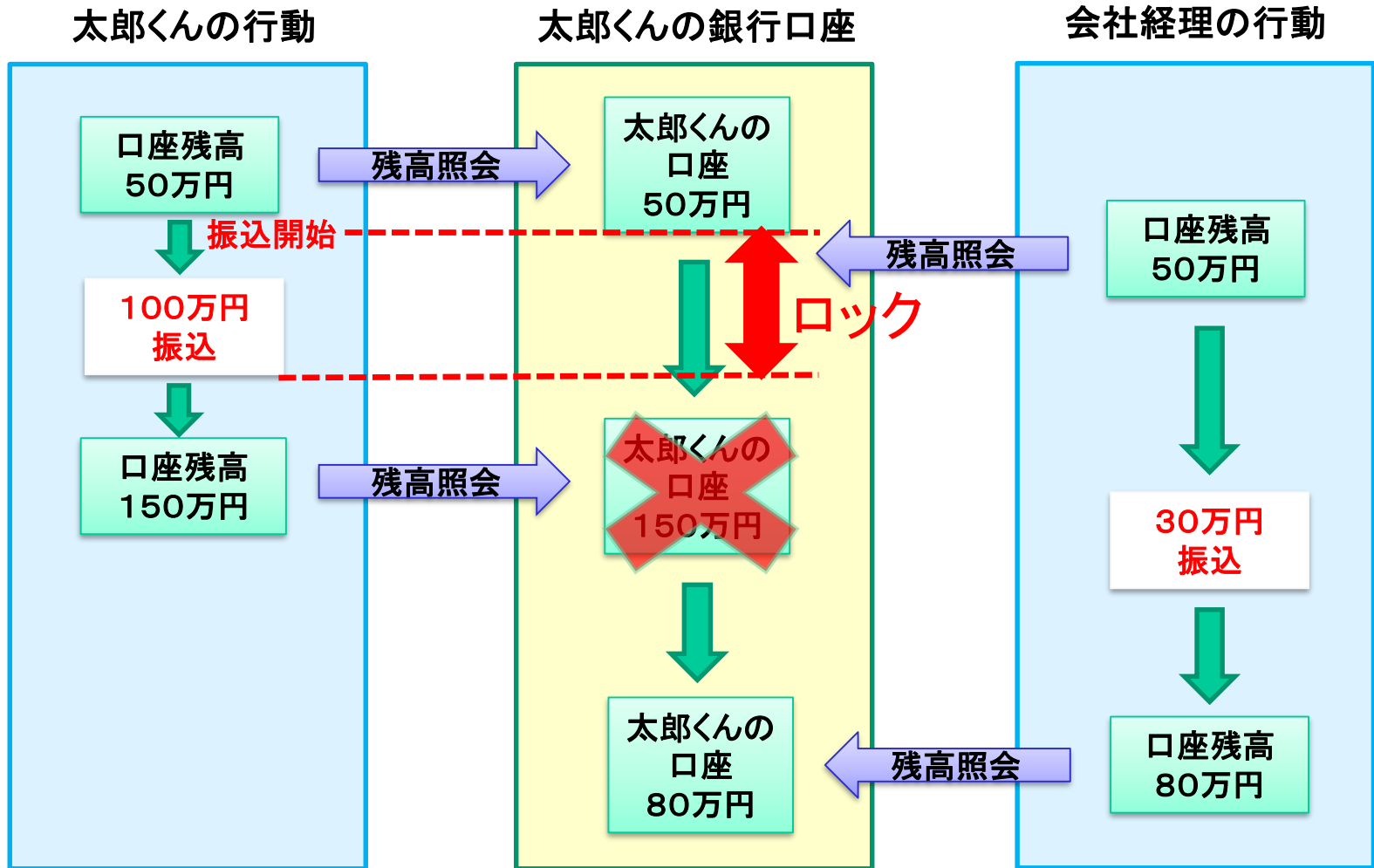
## データベースのACID特性とは

- A** ⇒ **Atomicity (原子性)**: トランザクションは処理が完結しているか、まったく処理されていないかのどちらかでなければならない。
- C** ⇒ **Consistency (一貫性)**: トランザクションの状態にかかわらず、データベースの内容には矛盾がない状態で行われなければならない。
- I** ⇒ **Isolation (隔離性)**: 複数のトランザクションを同時に実行した場合と、それぞれを1つずつ順番に実行した場合とで、結果は同じでなければならない。
- D** ⇒ **Durability (耐久性)**: 処理が終了したトランザクションは、その後にデータベースに障害が発生した場合でも、データの内容が損なわれてはならない。



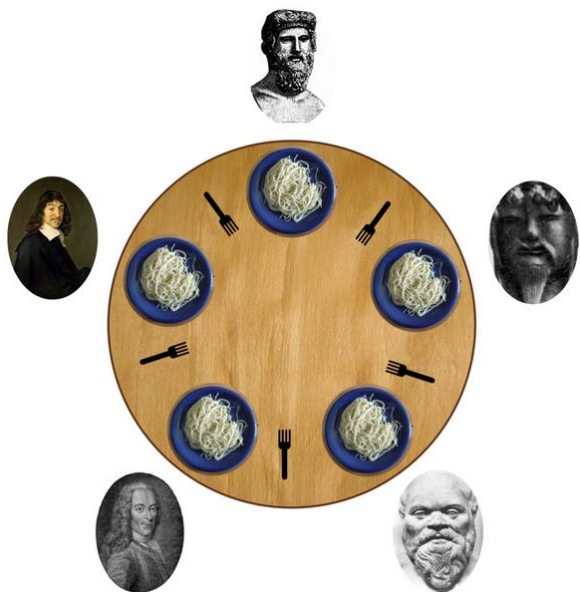
# ロックについて

事例) 太郎くんは株で儲けた100万円を振込んだが、ちょうどその時...



太郎くんの100万円はどこに消えた! ?

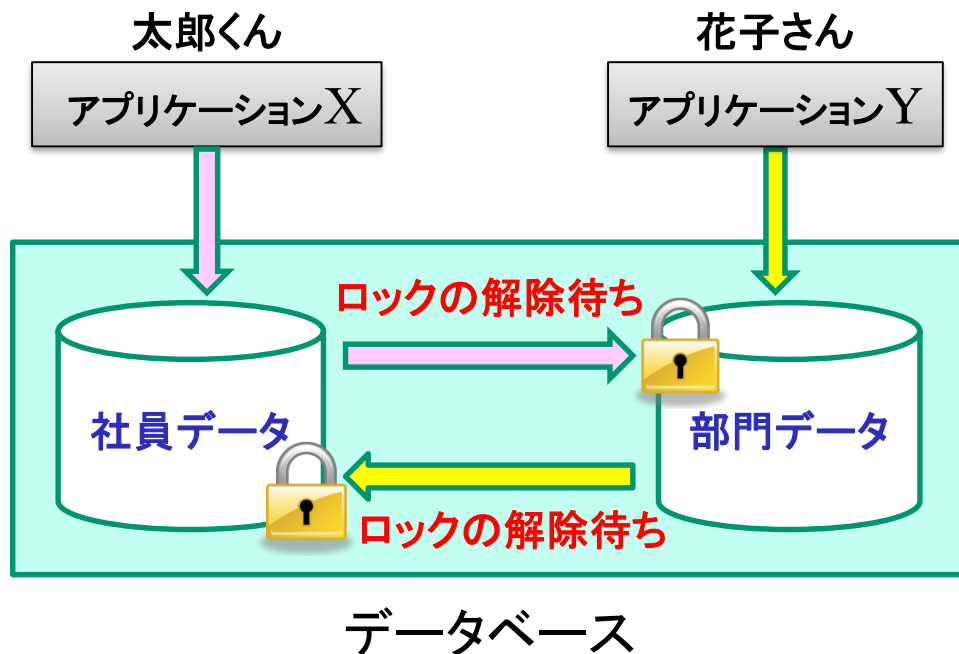
# デッドロックについて



条件：2本のフォークが揃うと食事できる

## 食事する哲学者の問題 (Dining Philosophers Problem)

② ロックする順序がループする場合

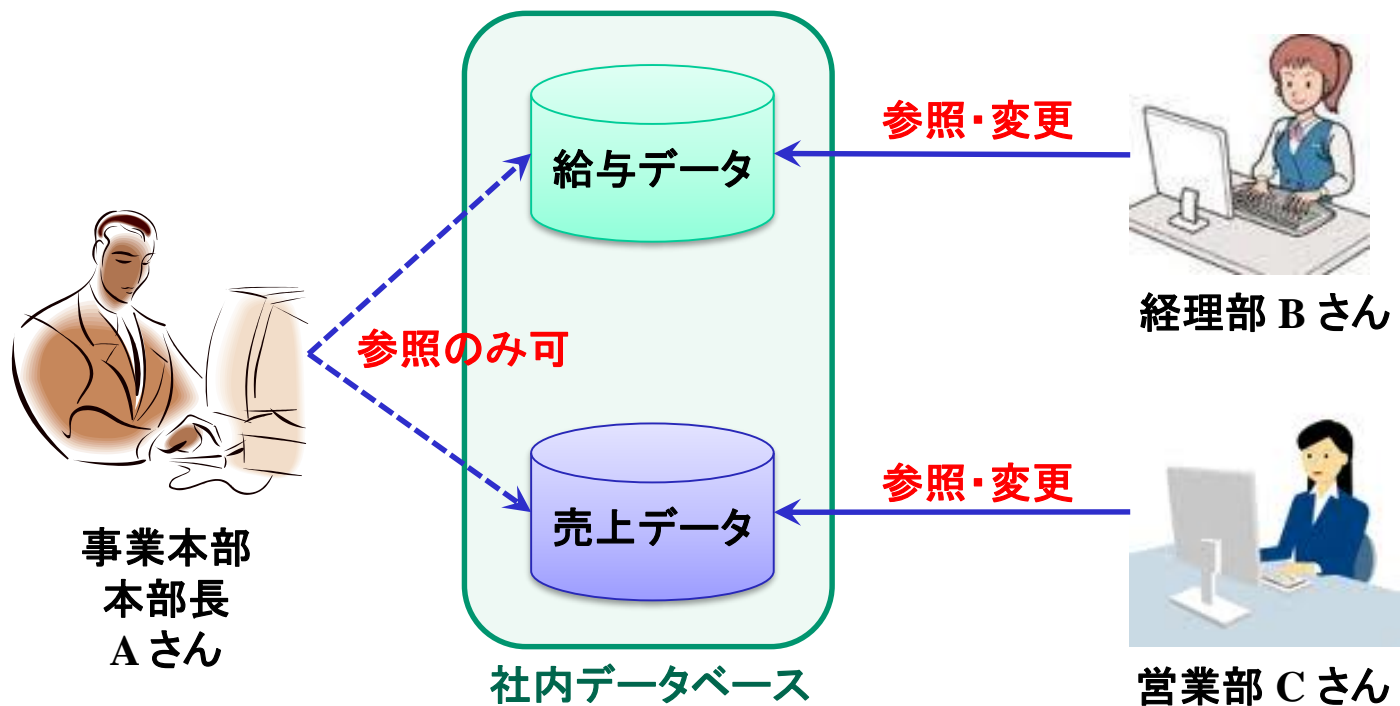


① ロックする順序が逆になる場合

デッドロックを最小化するには

1. 適度に譲り合う
2. ルールを設ける
3. 第三者によって監視する

# ユーザー権限について



ユーザーの社内的なポスト、所属部門によって、データベースへのアクセス権限を変えることで、情報の保護が可能となる。

# 関係演算について

<学生テーブル>

学籍番号	名前	住所
1	田中	東京
2	木村	神奈川
3	伊藤	神奈川
4	鈴木	埼玉
5	佐藤	千葉

選択演算  
(Selection)

Ⓐ

学籍番号	名前	住所
2	木村	神奈川
3	伊藤	神奈川

射影演算  
(Projection)

Ⓑ

学籍番号	名前
1	田中
2	木村
3	伊藤
4	鈴木
5	佐藤

<履修テーブル>

学生名	授業
田中	データベース
木村	ネットワーク
伊藤	ハードウェア
鈴木	ネットワーク
佐藤	データベース
三浦	データベース

Ⓒ

結合演算  
(Join)

学生名	授業	教師名
田中	データベース	前田
木村	ネットワーク	増田
伊藤	ハードウェア	長東
鈴木	ネットワーク	増田
佐藤	データベース	前田
三浦	データベース	前田

<担当テーブル>

授業	教師名
ネットワーク	増田
ハードウェア	長東
データベース	前田

# 集合演算について

<学生テーブル1>		<学生テーブル2>	
学籍番号	名前	学籍番号	名前
0001	田中	0001	田中
0002	木村	0003	伊藤
0003	伊藤	0005	小林
0004	鈴木		

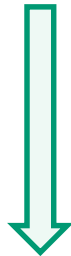
差集合演算  
(Difference)



学籍番号	名前
0002	木村
0004	鈴木

学生テーブル1 - 学生テーブル2

和集合演算  
(Union)

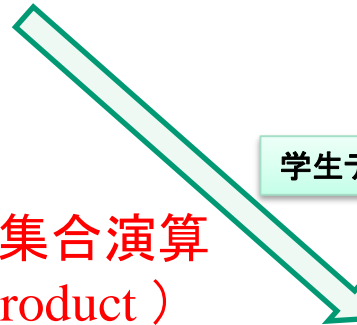


学籍番号	名前
0001	田中
0002	木村
0003	伊藤
0004	鈴木
0005	小林

学生テーブル1 + 学生テーブル2

積集合演算  
(Product)

学生テーブル1 × 学生テーブル2



学籍番号	名前
0001	田中
0003	伊藤

# テーブル上のキーについて

## 社員テーブル

コード	名前	性別	血液型	誕生日	部門コード
101	山口太郎	m	A	1972/02/01	10
102	鈴木一郎	m	AB	1974/09/10	20
103	佐藤輝彦	f	B	1976/04/21	10
104	小林信子	m	O	1978/12/24	30
105	佐々木昭	f	A	1980/07/04	30

## 部門テーブル

コード	名称
10	総務部
20	営業部
30	開発部
20	経理部

部門コードが重複している...

### プライマリーキー (主キー、Primary Key)

テーブル上のレコードを一件に絞り込むためのフィールド。

### ユニークキー (Unique Key)

重複したデータを格納できないようにしたフィールド。  
プライマリーキーはユニークキーの1つである。

### 外部キー (Foreign Key)

他のテーブルと関連付けるためのカギとなるフィールド。外部キーとして設定できるフィールドは、参照される側のプライマリーキーのみとなる。  
他のテーブルで外部キーとして使用されているデータは削除できない。

# SQLについて

リレーショナルデータベースと対話するための言語

1970年 IBM SystemR: 初のリレーショナルデータベースに実装された  
SEQUEL (Structured English Query Language: シークエル)

1986年～  
**ANSI,**  
**ISO規格化**  
**SQL**

SQLの種類 {  
データ定義言語 (DDL: Data Definition Language)  
データ制御言語 (DCL: Data Control Language)  
データ操作言語 (DML: Data Manipulation Language)

## SELECTの例)

SELECT : データの取得  
INSERT : データの追加  
UPDATE : データの更新  
DELETE : データの削除

① SELECT 学籍番号, 名前 FROM 学生

表示したいフィールドのリスト、  
\*ならすべてを表示

テーブル名のリスト

② SELECT 学籍番号, 名前, 住所 FROM 学生  
WHERE 住所 = '神奈川'

検索条件、不等号による大小比較、ORやANDによる論理演算も可能

③ SELECT 学生名. 履修, 授業. 履修, 教師名. 担当  
FROM 履修, 担当  
WHERE 授業. 担当 = 授業. 履修

# 演習

1. これまでに経験したデッドロックと思われる事象を記述しなさい。
2. 以下のEMPLOYEE表から①と②の結果が得られるようなSQLコマンド(ここでは、SELECT文)をそれぞれ記述せよ。

EMPLOYEE表

EMPNO	EMPNAME	JOB	TEL	DEPTNO	SALARY
100	佐藤	社長	03-5213-6666	10	5000
200	鈴木	技術部長	06-6457-6777	20	1250
300	高橋	営業部長	022-716-6000	30	1600
400	渡辺	技術	03-5213-6666	20	1100



②

EMPNAME	SALARY
鈴木	1250
渡辺	1100

①



EMPNO	EMPNAME	JOB	TEL	DEPTNO	SALARY
200	鈴木	技術部長	06-6457-6777	20	1250
400	渡辺	技術	03-5213-6666	20	1100