

データベース設計論

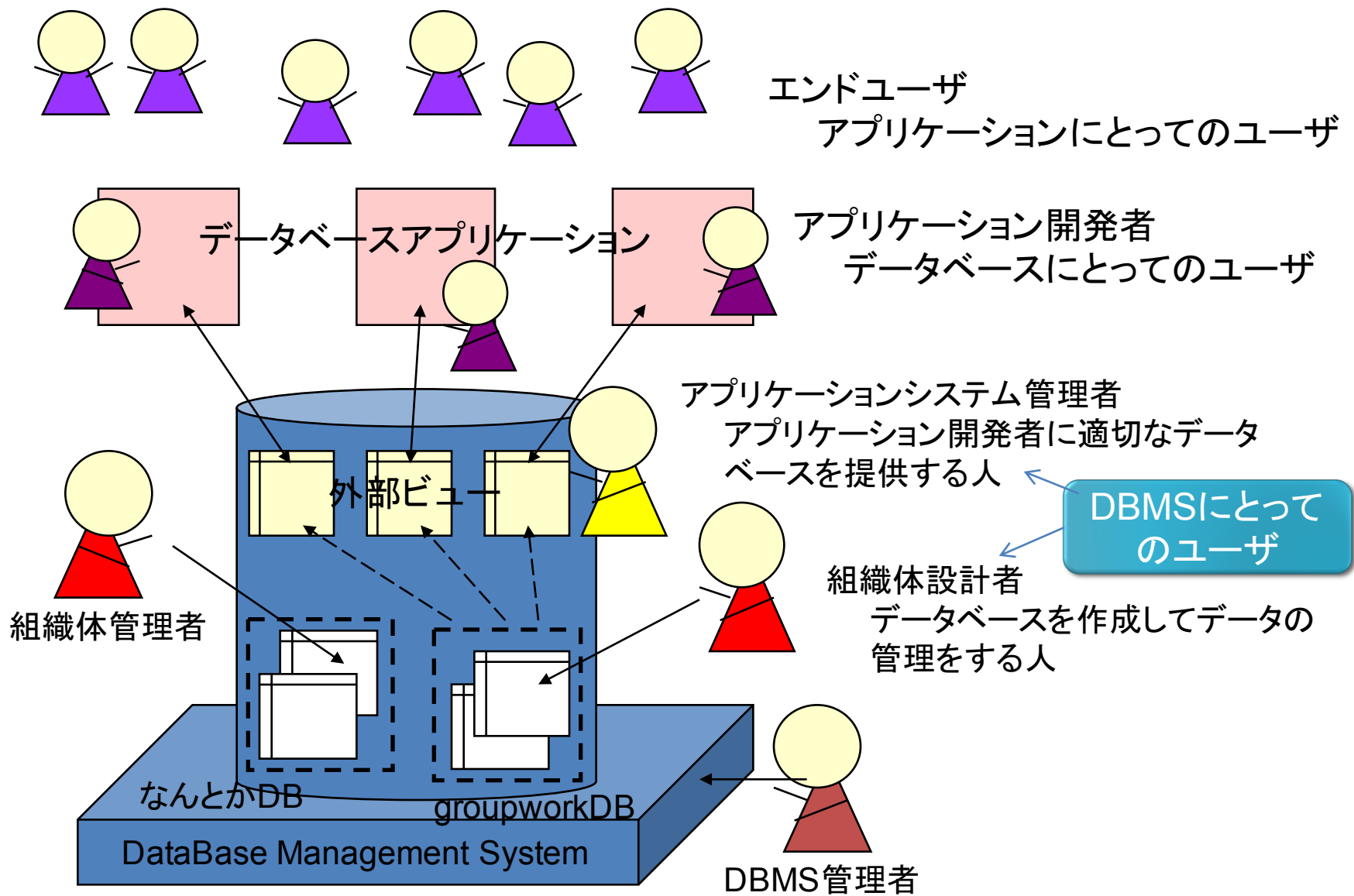
第10回

アプリケーションとデータベースの 関係

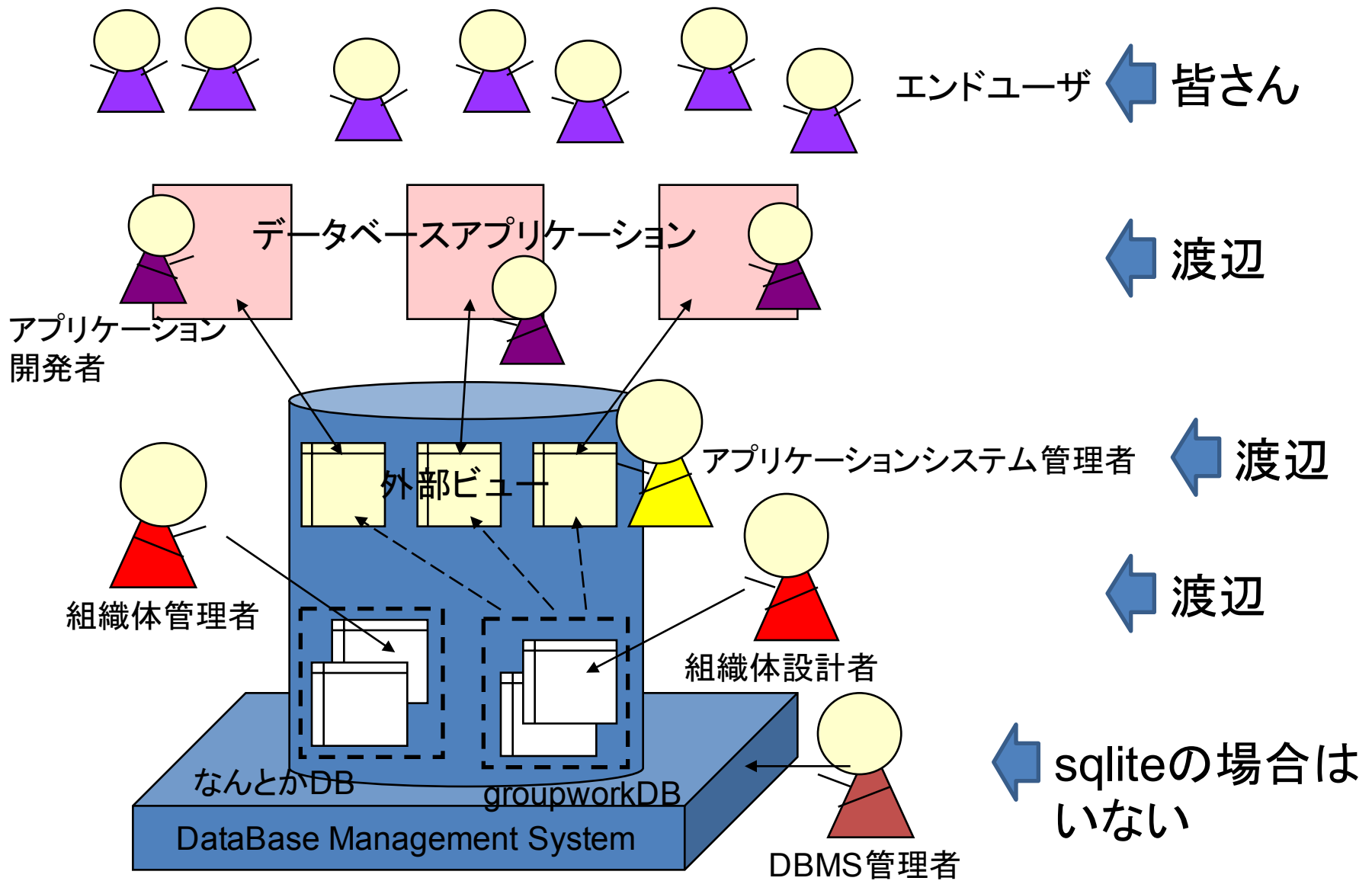
本日の内容

- 今日は、問合せからちょっと外れて「アプリケーションとデータベースの関係」について勉強します
- Keywords
 - DBMSの利用者
 - データ独立性
 - ビュー(外部スキーマ)
 - ODBC, JDBC

DBMSの利用者

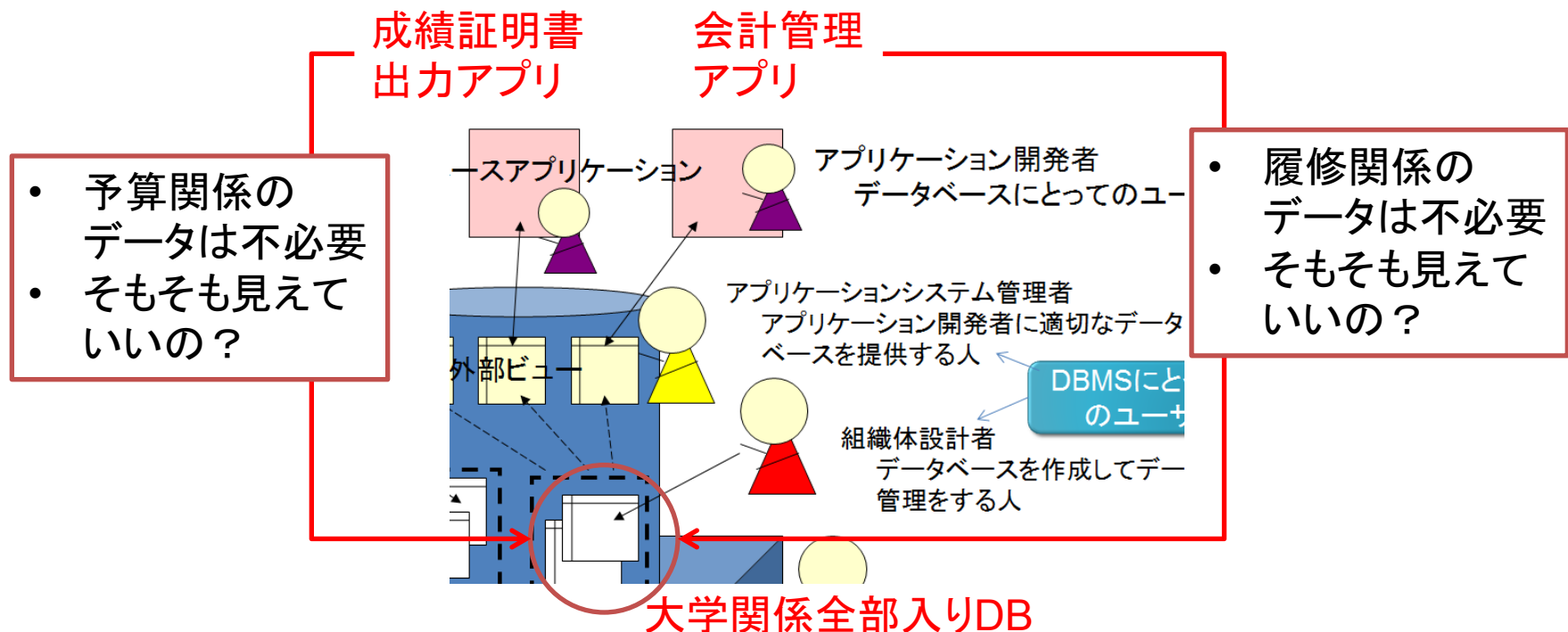


Groupwork データベースの場合



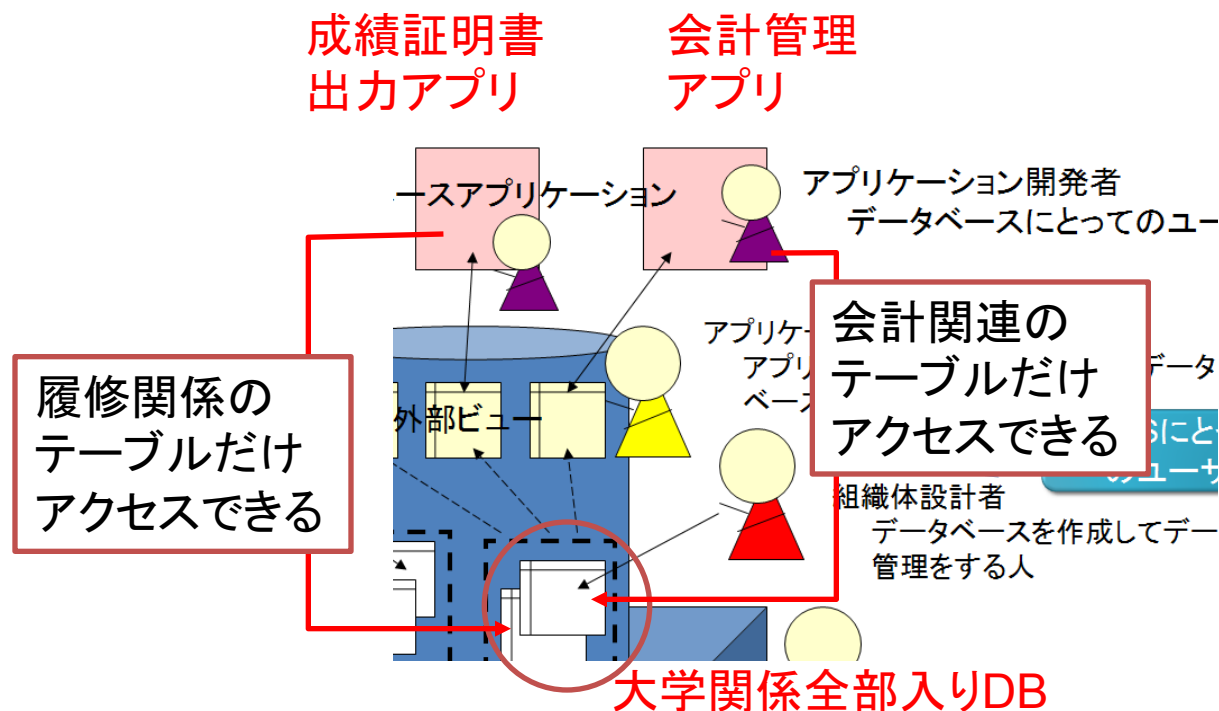
さて...

- アプリケーションシステム管理者はなぜ必要？
- 何でも入っている巨大なデータベースがあった時
 - 例) 大学関係(教員名簿、予算、学生名簿、履修、成績)が全部入っているデータベース



アプリケーションシステム管理者のお仕事

- アプリケーション開発者に適切なDBやテーブルのアクセス権を与える



一部のテーブルだけ見せたい場合は？

- Groupworkデータベースの例
 - studentsに教員も学生も全員入ってしまっている
(リレーション名的にそれはまずいのだが...)
- 学生が主体で開発するアプリが
Groupworkのメンバー覧を観れるようにしたい
 - 教員はいらない
 - パスワードもいらない

元データ (students)

stid	name	grade	password
1	山田花子	3	x;;jkaj;ajklak
2	渡辺知恵美	20	4ej;sj;j;sjkld
3	浅田紀子	2	sj;ekejrhjwk

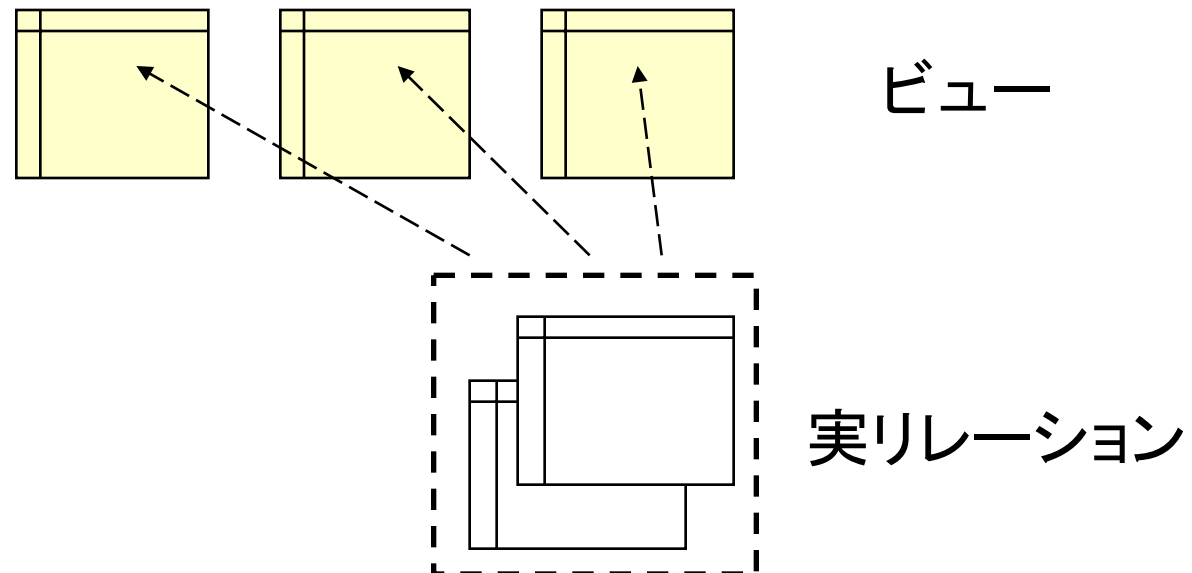
アプリ開発者に提供するデータ

stid	name	grade
1	山田花子	3
3	浅田紀子	2

さてどうする？

ビュー(仮想リレーション)

- 実リレーションから導出される結果リレーションをビューとして保存できる
- ビューが保存するのは結果リレーション本体でなく, ビューを導出した問合せ文
 - ビューの利用者は一見普通のリレーションに見えるが, 実はそのつど実リレーションに問合せして, 実態を生成する



ビューの作成

- CREATE VIEW構文を使う

CREATE VIEW ビュー名 AS 問合せ文

学生の名簿情報のみを抽出したビューを作る

```
CREATE VIEW groupwork_students  
AS  
SELECT stid, name, grade  
FROM students  
WHERE grade between 1 and 4
```

ビューはなぜ必要なのか

- 実リレーションから必要なデータだけを抽出しておきビューとして保存できる
 - 実リレーションが多数存在し、お互いが複雑に関係しているとき、ビューを作ることで問合せ文を単純化することができる
- アクセス権の制御ができる
 - 利用者やアプリケーション開発者には、実リレーションでなくビューだけにアクセスできるように設定すればセキュリティを保護することができる

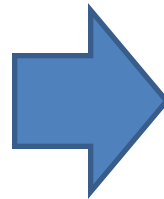
もう一つ大事な使い方

- リレーションスキーマの変更を余儀なくされたとき問題になるのは、すべてのアプリケーションに影響がでること

ビューを使えば回避できる

- 例) これまで属性に年齢をつけていたが生年月日に変更することにした場合

id	name	age
1	Mao Asada	23
2	Elena Radionova	14
3	Adelina Sotnicova	17



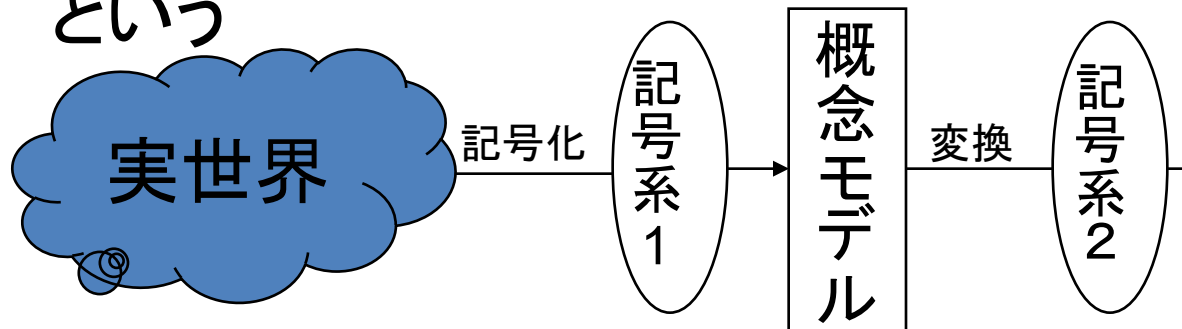
id	name	birthdate
1	Mao Asada	1990/9/25
2	Elena Radionova	1999/1/6
3	Adelina Sotnicova	1996/7/1

さてどうする？

※年齢計算は以下の関数でできることとします
calcage(birthdate, today)

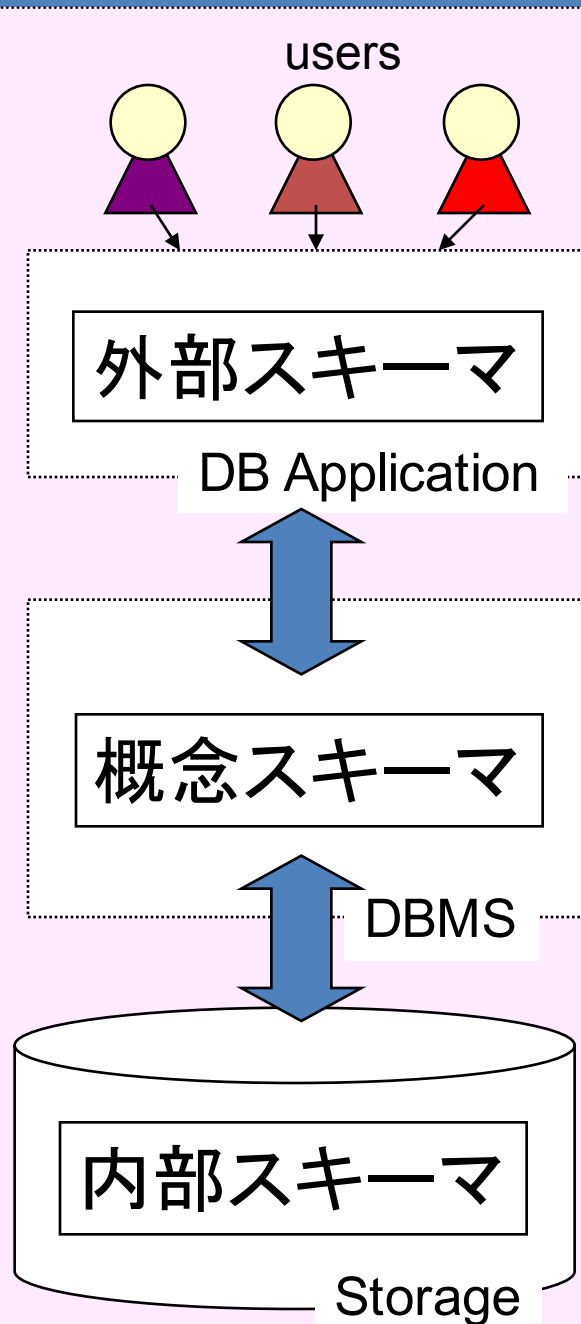
データベースの基本的概念： データ独立性

データベースとアプリケーション，
ストレージの関係を独立して扱う
ことができることをデータ独立性
という



Universe of Discourse(UoD):
対象世界, 対象領域

ANSI/X3/SPARCの3層スキーマ

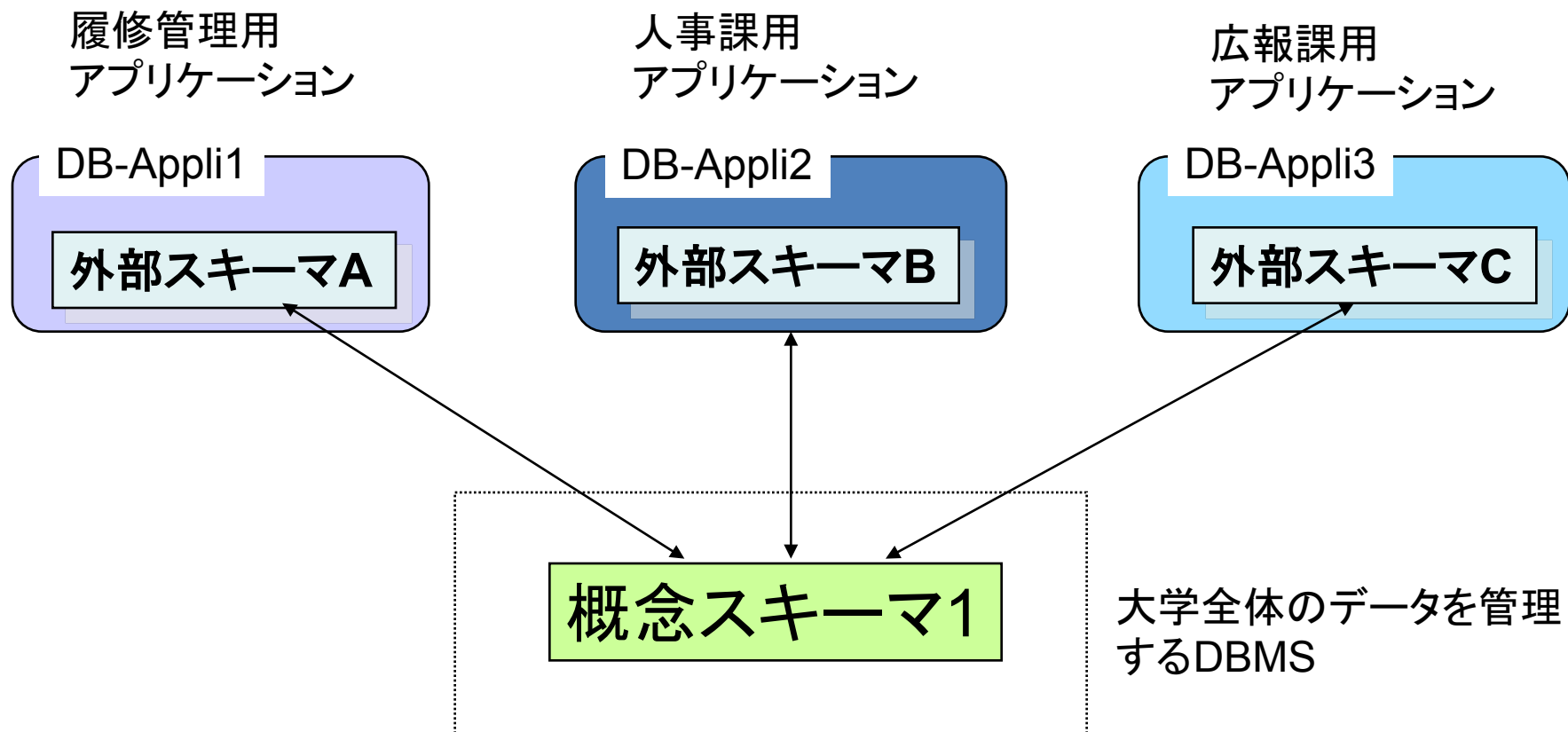


三層スキーマ

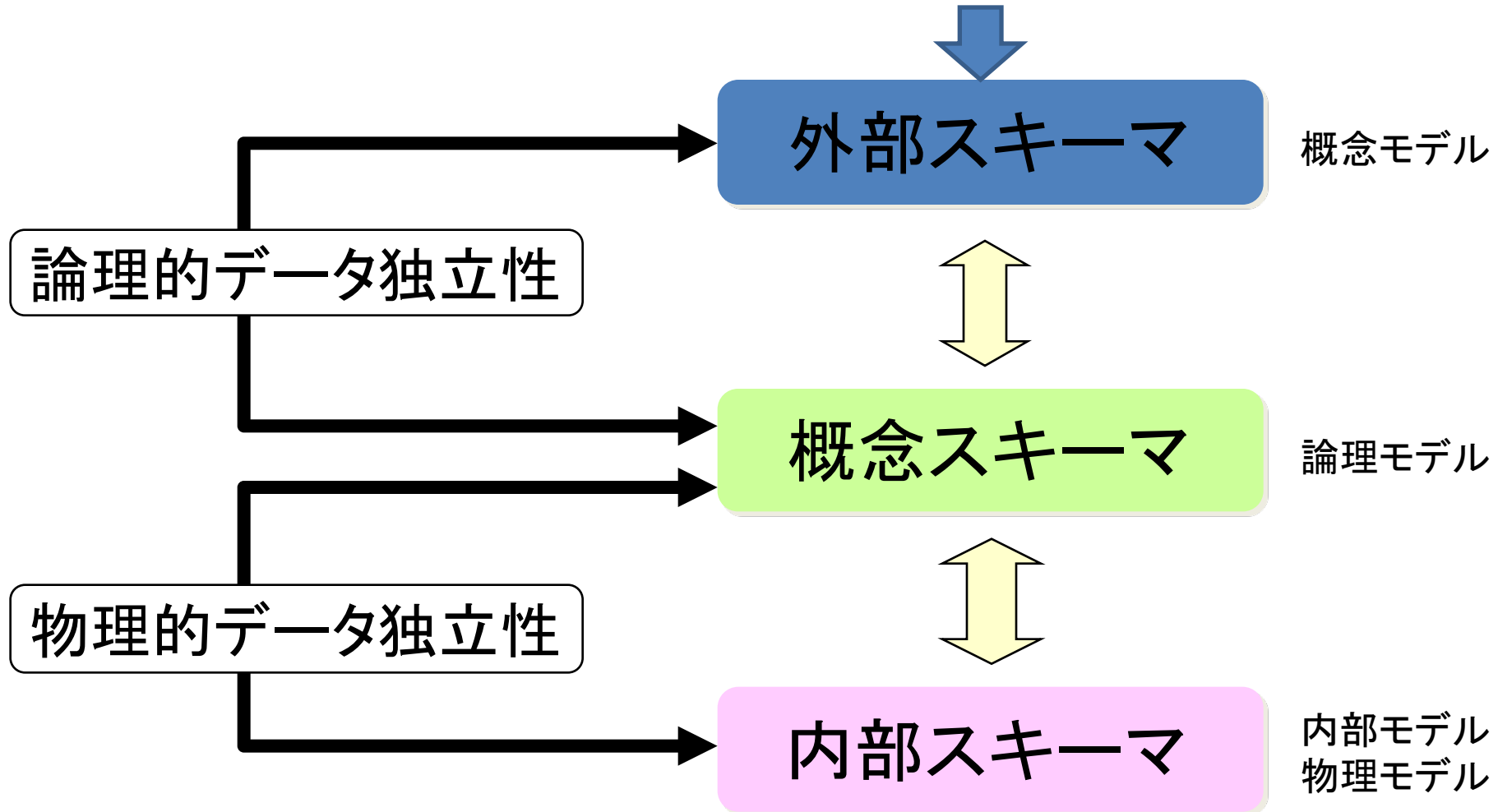
- 概念スキーマ(conceptual schema)
 - データモデルによりUoDをデータベースに「論理的に」記述したもの
 - 前述の「論理スキーマ」と同等のものである
- 内部スキーマ(internal schema)
 - 実際のストレージ(ディスク装置, ファイルなど)に格納されている形式
- 外部スキーマ(external schema)
 - データベースアプリケーションが利用するデータのデータ形式
 - ビューとよぶ

外部スキーマ(ビュー)

- DBアプリケーションはDBMSに管理されている大きなデータベースに対して、自分の必要なデータだけを「ビュー」として利用することが出来る

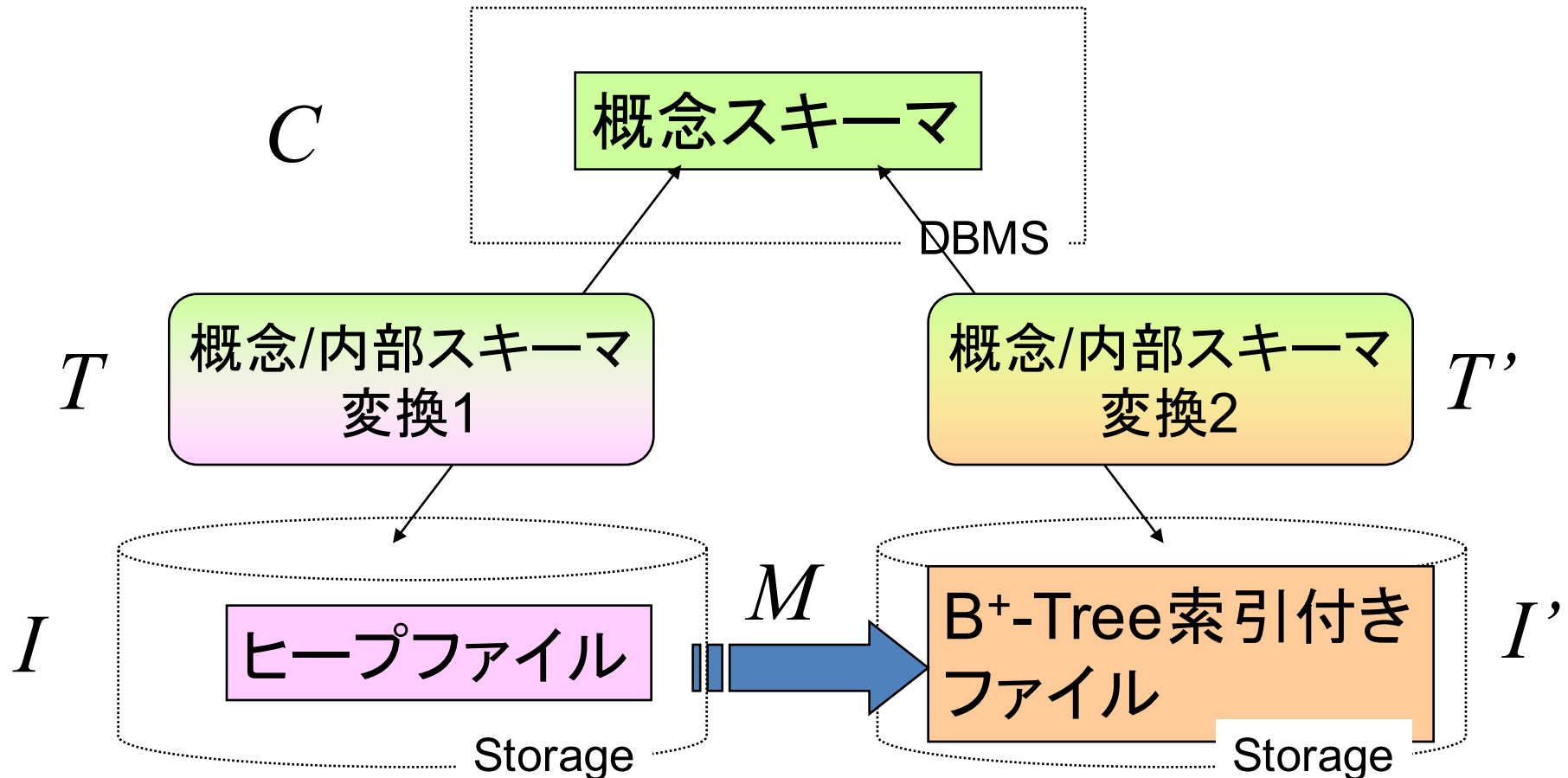


ANSI/X3/SPARCの3階層スキーマ



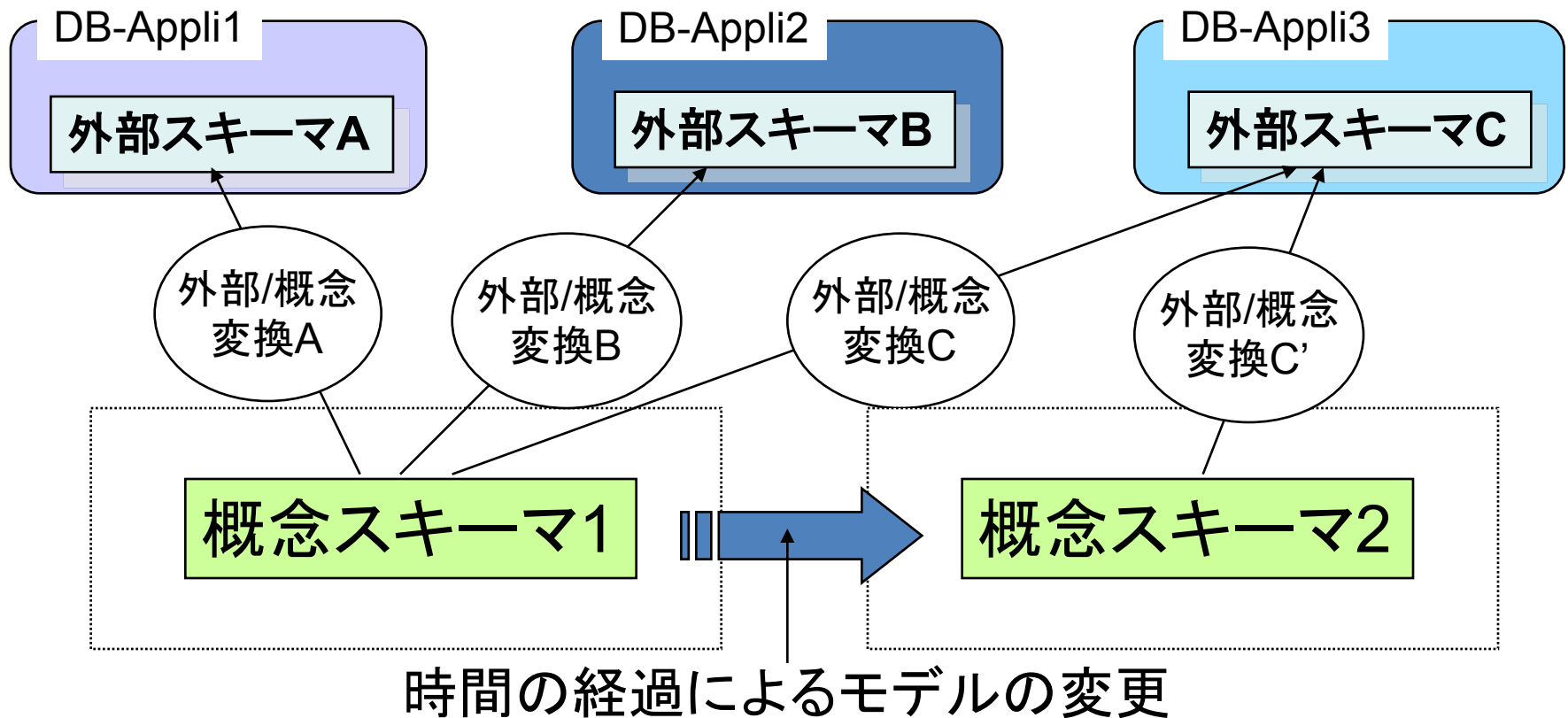
物理的データ独立性

- 内部スキーマの変更が概念スキーマに影響を与えないこと



論理的データ独立性

- 概念スキーマの変化がDBアプリケーションに影響を与えないこと



データベースアプリケーションの開発

データベースに接続するプログラムを作る方法

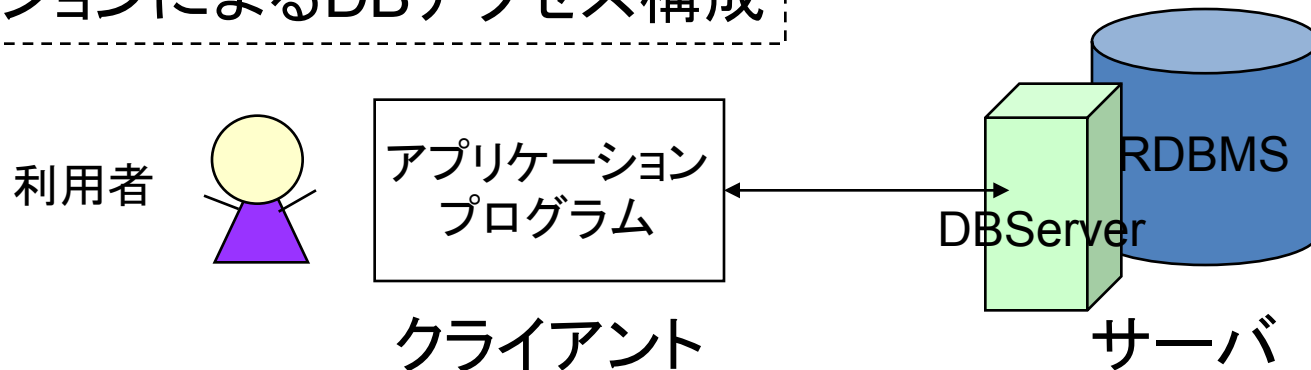
- 埋め込みSQL(Embedded SQL)
 - 各RDBMSベンダーがプログラミングからデータベースにアクセスするためのライブラリ(API)を提供している
 - それぞれの埋め込みSQLにより関数や使い方が異なる
 - Native Interface
- ODBC
 - Microsoftが提唱した, RDBMSに対する共通のアクセスインタフェース
 - 各RDBMSベンダーはODBCへのドライバを提供する
- JDBC
 - ODBCのJava版
- DBI(perl), PDO(PHP), ActiveRecord(Ruby)

SQLite3にアクセスできるプログラミング言語

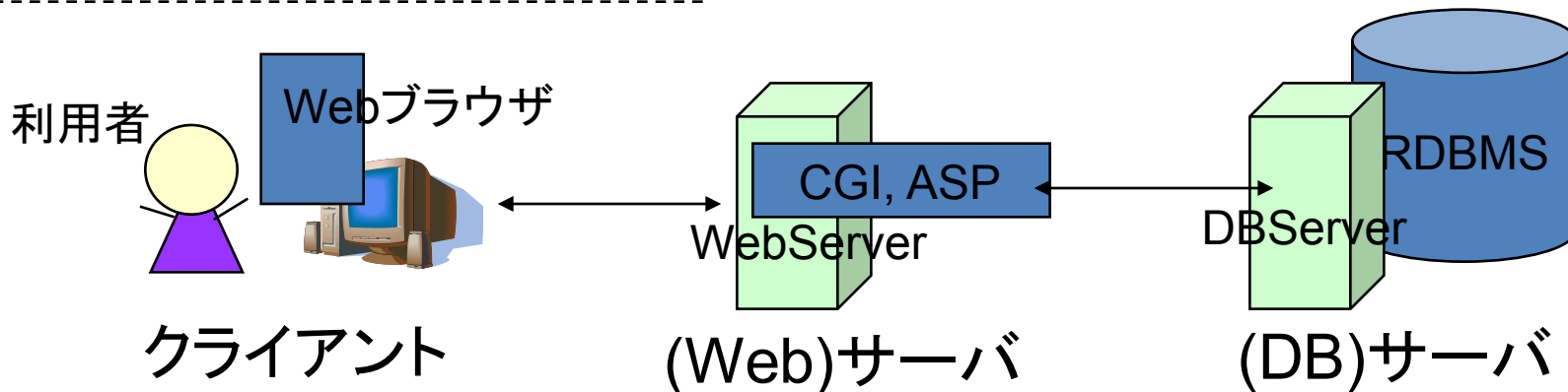
- ODBCが利用できるプログラミング言語
 - C, C++
 - Microsoft .NET系 (Visual C++, C#, ASPなど)
 - なでしこ (日本語プログラミング言語)
- Java
 - JDBC経由
- Perl
 - DBI/DBD : Perl用の汎用インタフェース
- Ruby
- PHP

データベースアプリケーションの構成

アプリケーションによるDBアクセス構成

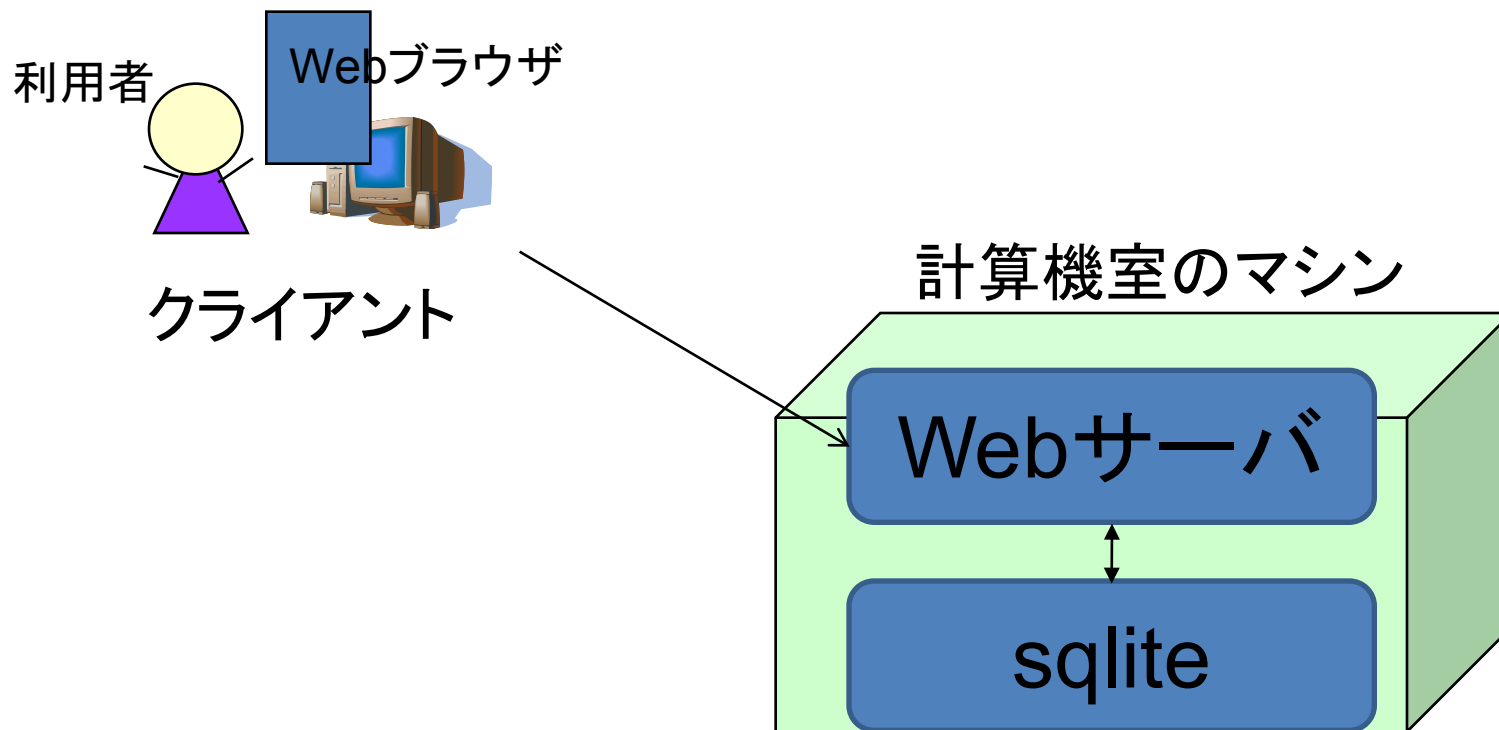


WEB+DBによる三層構成



演習: Web+DBアプリケーションを作ろう

- 構成



PHPに関して

- PHPとは

- 動的にWebページを生成するスクリプト言語
- HTMLに埋め込む形で書くことができる

- 文法などはWebで調べるといろいろ出てくるのでそちらで調べてください

- 参考:

- 初心者用PHP入門
- http://www.standpower.com/php_grammar.html

```
<html>
<body>
<?php
    $name = "Chiemi"
?>
こんにちは
<?php
    print $name
?>
さん
</body>
</html>
```

PDO (PHP Data Object)

- PHPからデータベースに接続するためのライブラリ
- DBへ接続
 - `$dbh = new PDO($dsn, $user, $password);`
- 問合せ
 - `$stmt = $dbh->prepare("select * from s");`
 - `$stmt->execute();`
- データの取得
 - `$cols = $stmt->fetch(PDO::FETCH_NUM)`
 - `$cols`は配列になる
- DBから切断
 - `$dbh = null`
- 他詳しくは「PHP PDO」で調べると出てきます ♪
参考: <http://www.phpbook.jp/database/pdo/index.html>