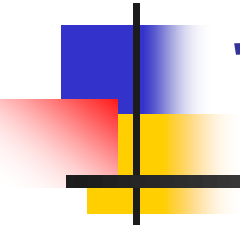


データベース(第8回)

情報工学科 木村昌臣

ANSI/X3/SPARC の3層スキーマ構造





DBMSはどうあるべき？

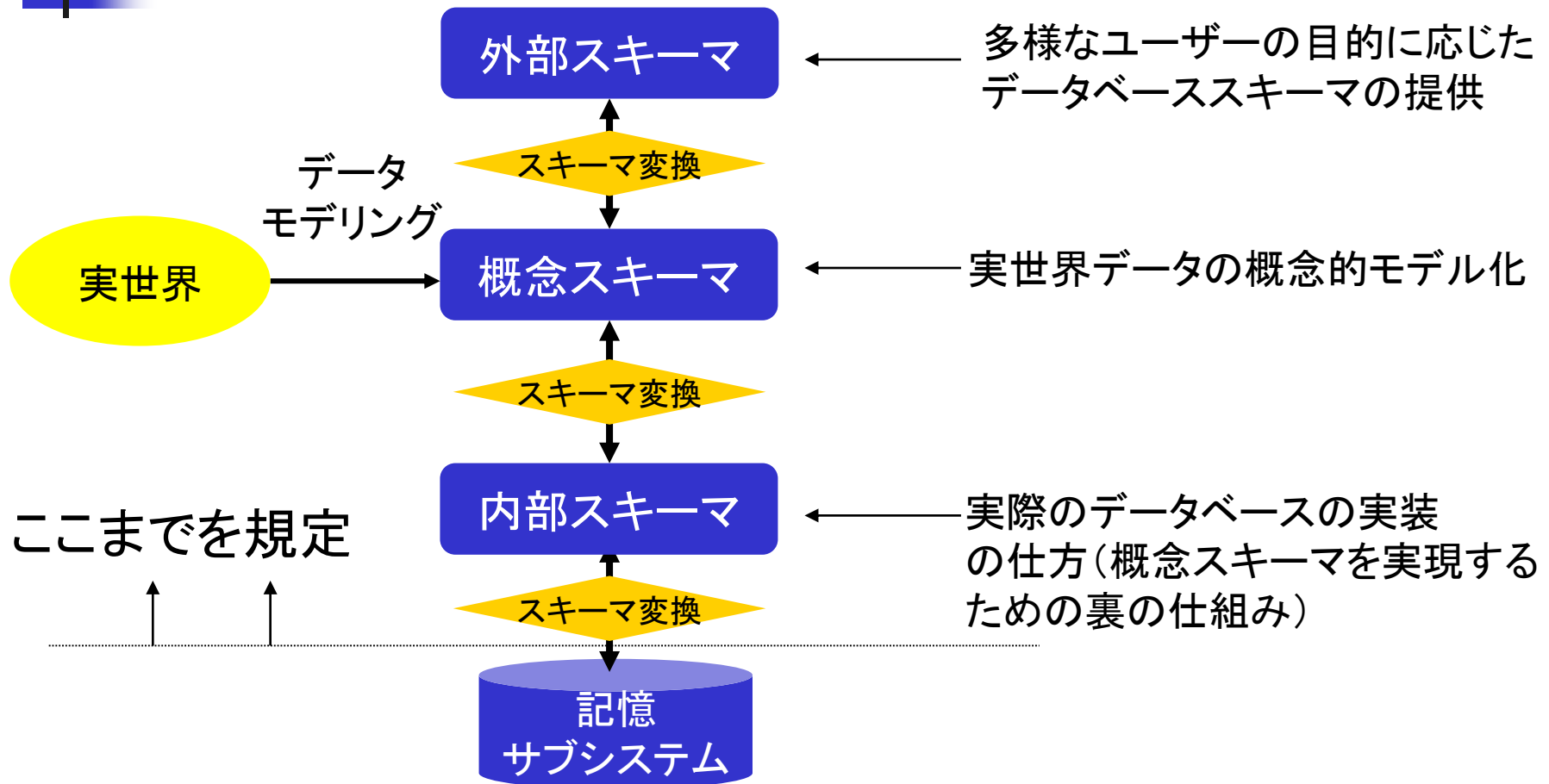
- DBを構築する場合、どのような構成にすべきか



ANSI/X3/SPARC の 3層スキーマ構造

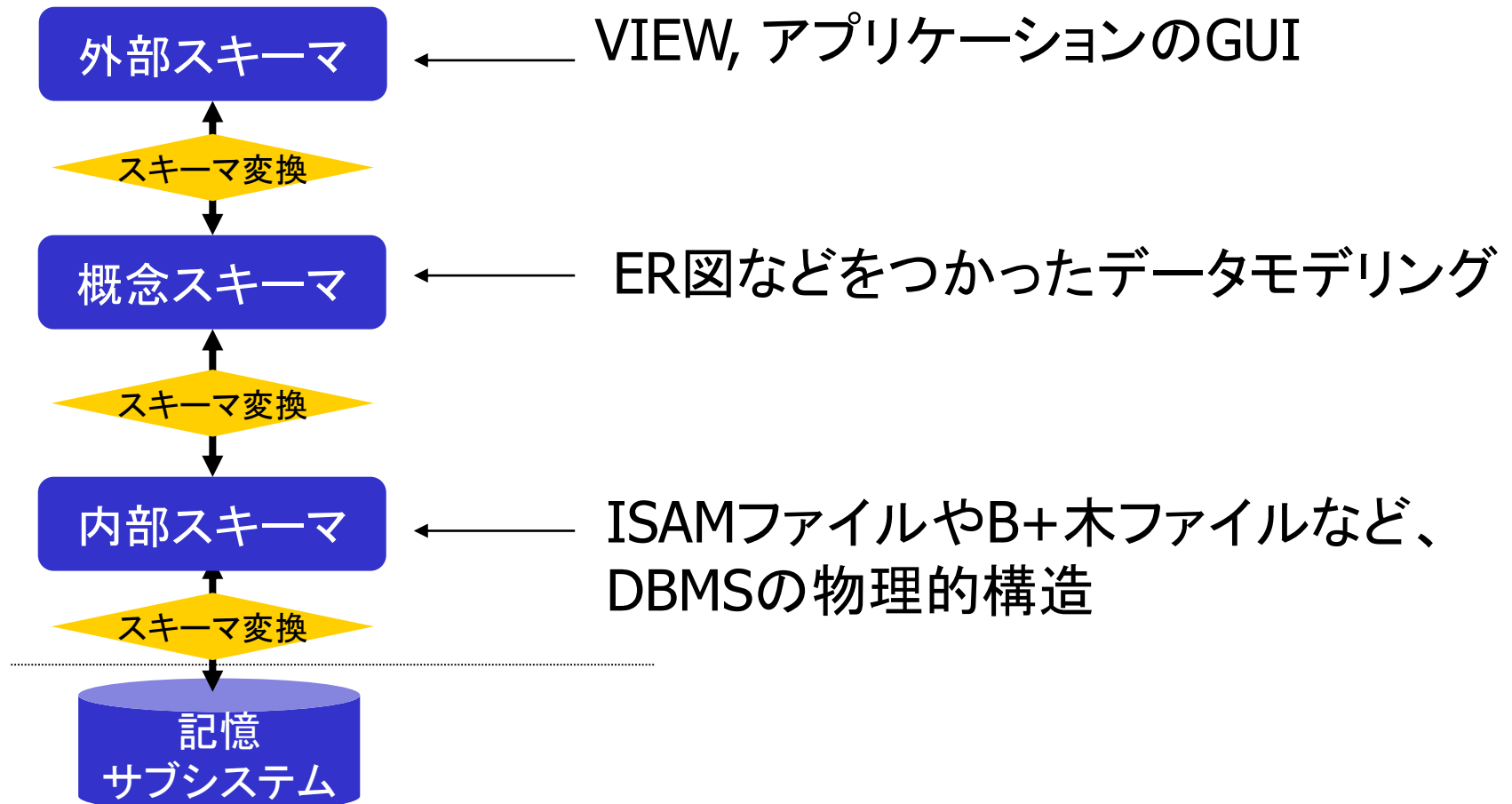
- RDBMSだけでなくDBMSであれば満たすべき構造
- 1973 審議開始
- 1978 報告書にまとめる

ANSI/X3/SPARC 3層スキーマ構造

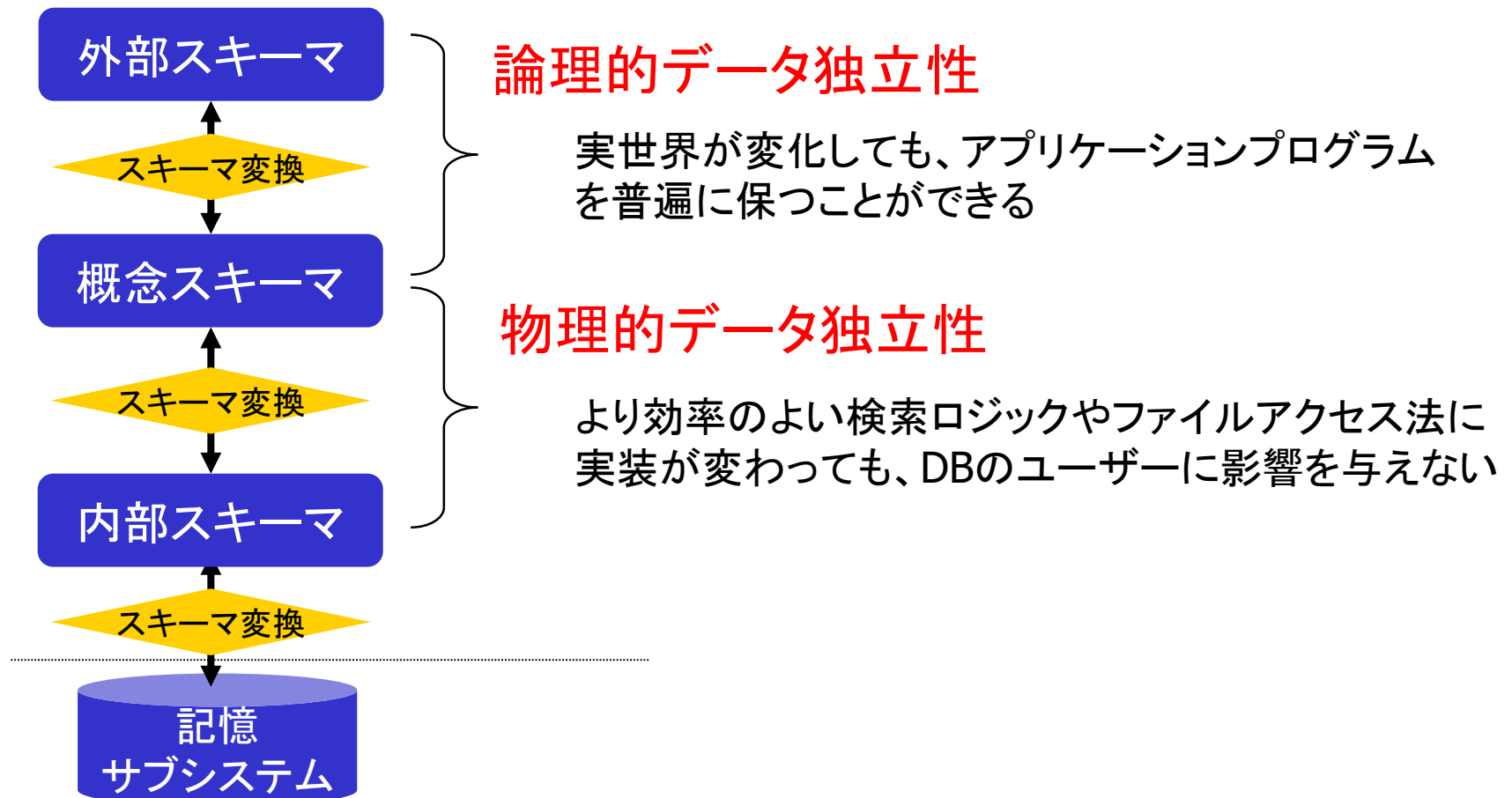


ANSI/X3/SPARC

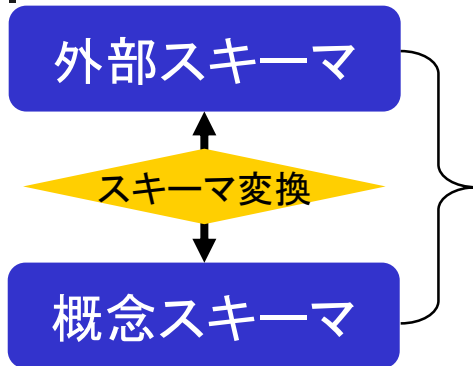
3層スキーマ構造(例)



3層スキーマの意義



3層スキーマの意義

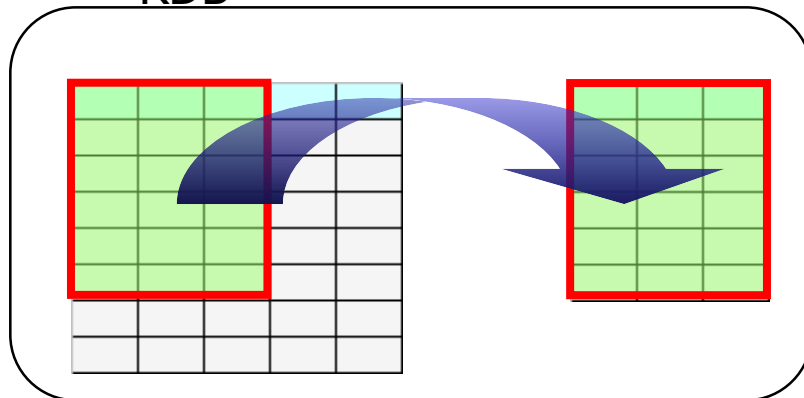


論理的データ独立性

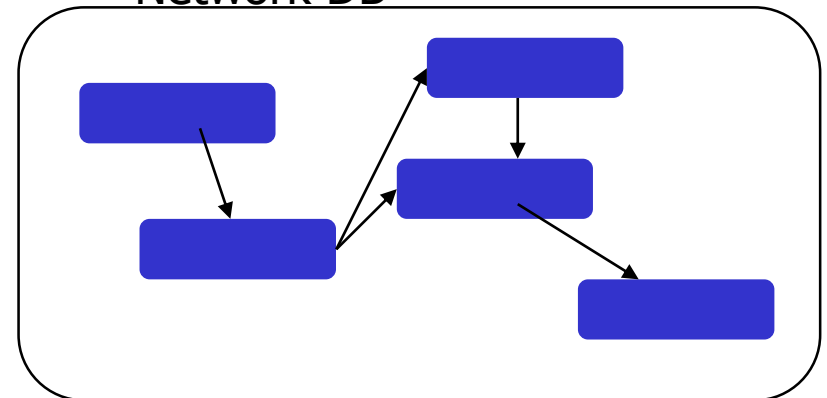
✓リレーショナルデータベースでは、問い合わせ結果と元の表の構造がおなじため、問い合わせ結果として得られる表を外部スキーマとして利用することが可能

✓ネットワークデータベースや階層データベースは概念スキーマと内部スキーマの区別がはっきりしないため、外部スキーマのサポートは困難

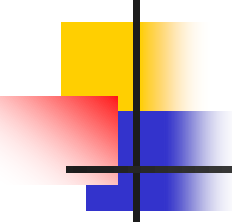
RDB



Network DB







ビュー(View)

- 実際にデータを持つリレーションを実リレーションという。
- 問い合わせの結果できるリレーションのことを結果リレーションという。
- ビューとは、結果リレーションを仮想的なリレーションとして定義したもの。
 - ビューの定義は、結果リレーションを得るための定義（実リレーションへの問い合わせ）のみ。
 - ビューには、実データが存在しない。（実データがあるのはあくまで実リレーション）



VIEW

View作成のSQL文

```
CREATE VIEW View名  
AS  
SELECT文
```

SELECT文の結果をViewという
仮想的な表としてユーザーに
提供する

例)

選択ビュー

```
CREATE VIEW 中堅社員  
AS  
SELECT *  
FROM 社員  
WHERE 年齢 BETWEEN 30 AND 40
```

社員テーブルから該当レコードを
選択して表示

和ビュー

```
CREATE VIEW 全社社員  
AS  
SELECT *  
FROM 東京本社社員  
UNION  
SELECT *  
FROM 大阪支社社員
```



VIEWの種類

- 選択ビュー
 - 元のテーブルから条件に合うレコードを選択してビューとして定義
- 和ビュー
 - 複数テーブルのレコードの和集合をビューとして定義
- 結合ビュー
 - 複数のテーブルを結合(JOIN)したものをビューとして定義
- 値式を使って定義されるビュー
 - SELECTリストの列名の代わりに値式をつかってできた「表」をビューとして定義

VIEWの更新可能性

```
CREATE VIEW 取引  
AS  
SELECT X.仕入れ先, Y.納入先  
FROM 供給 X, 需要 Y  
WHERE X.部品 = Y.部品
```

(XX工業, YY商店)を
削除したい
どうしたらいい？

1. 供給テーブルから
(XX工業, 部品ZZ)
を削除
2. 需要テーブルから
(部品ZZ, YY商店)
を削除
3. 1.2.の両方を削除

1.~3.は、意味としては全く異なるが
これらのどれにしたらいいか、SQL
からは判別できない。

SQL規格におけるVIEWの更新可能性

- DISTINCTを使わない
- 値式はすべて列参照
- FROM句には単一のテーブル(実表もしくはは更新可能なVIEW)が指定されている
- 副照会をしているときは、相関がないこと
- GROUP BYやHAVINGを使用していないこと

つまり

一枚のテーブルに対する射影や選択演算で切り出された結果表である場合のみ更新可能



体現VIEW

- 通常のVIEWに問い合わせや更新を行う場合には、ビュー定義がそれらに組み込まれて実行される。
- ただし、それではパフォーマンスがよくないことがある
- あたかも実テーブルのようにデータベースに格納した(実データを伴う)VIEWのことを
体現VIEWという。