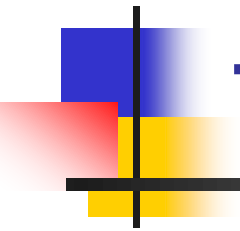
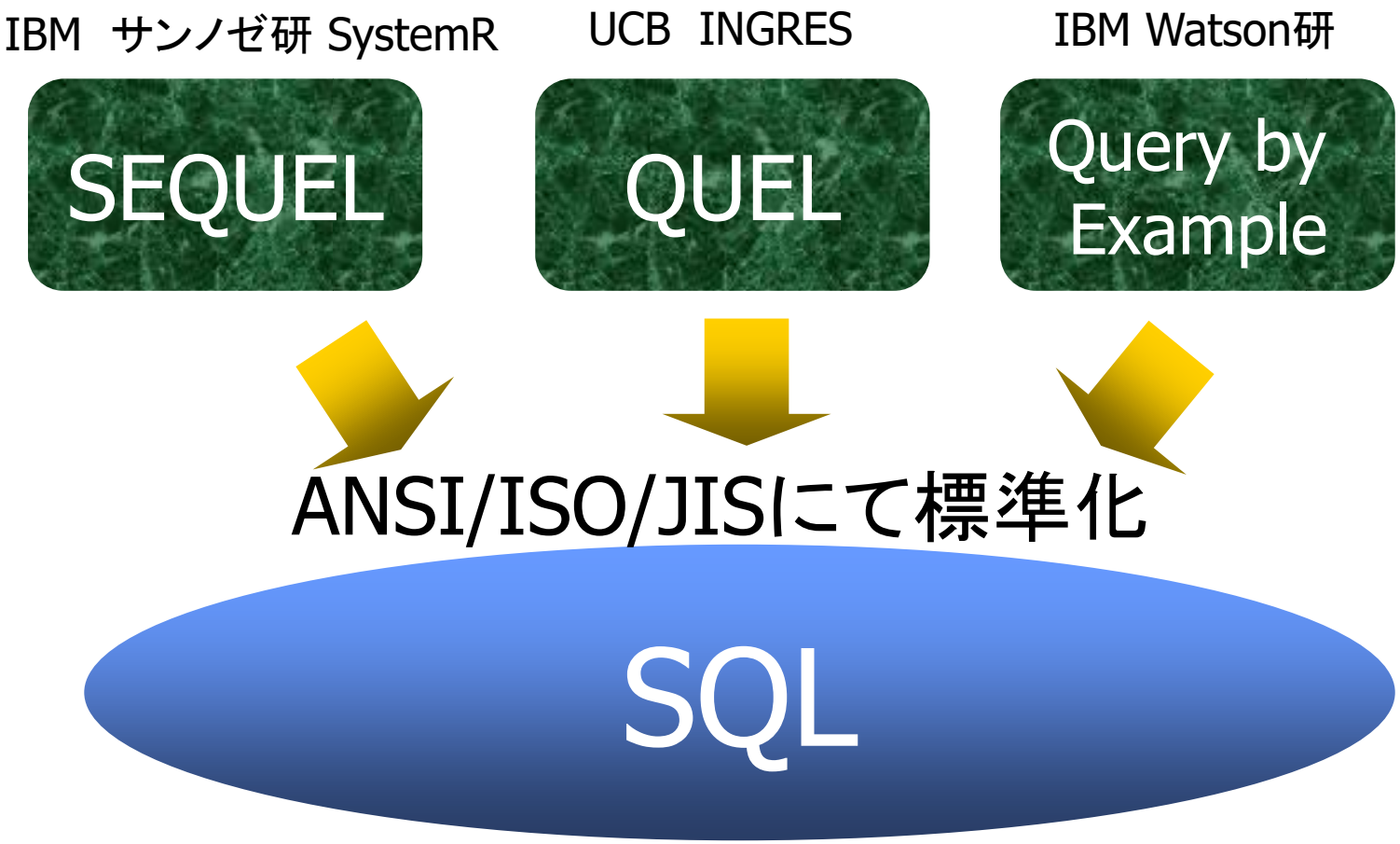
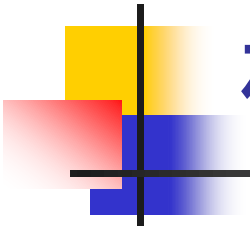


データベース(第6回)

情報工学科
木村昌臣

リレーショナルデータベース言語の標準化



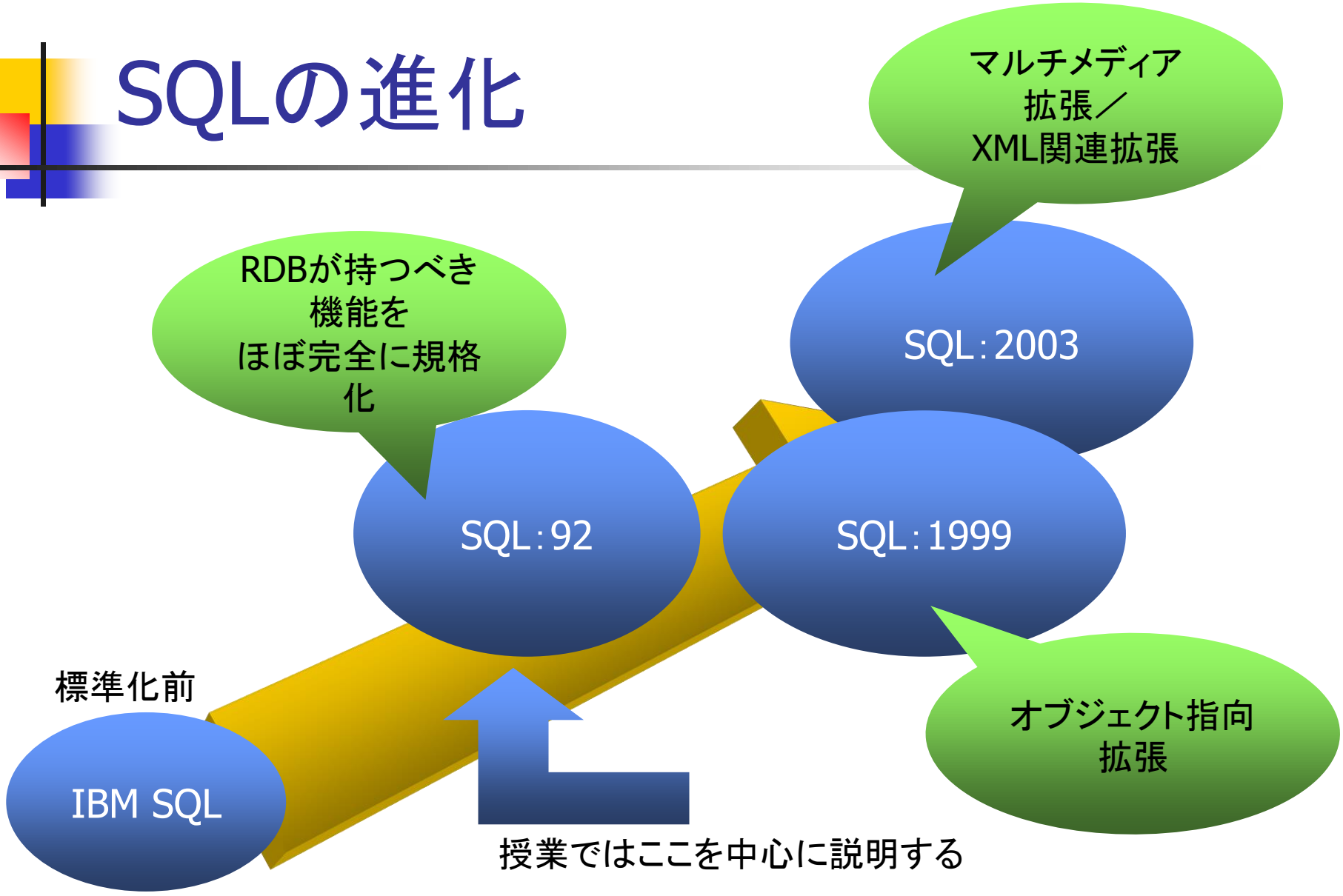


標準化されていないと・・・

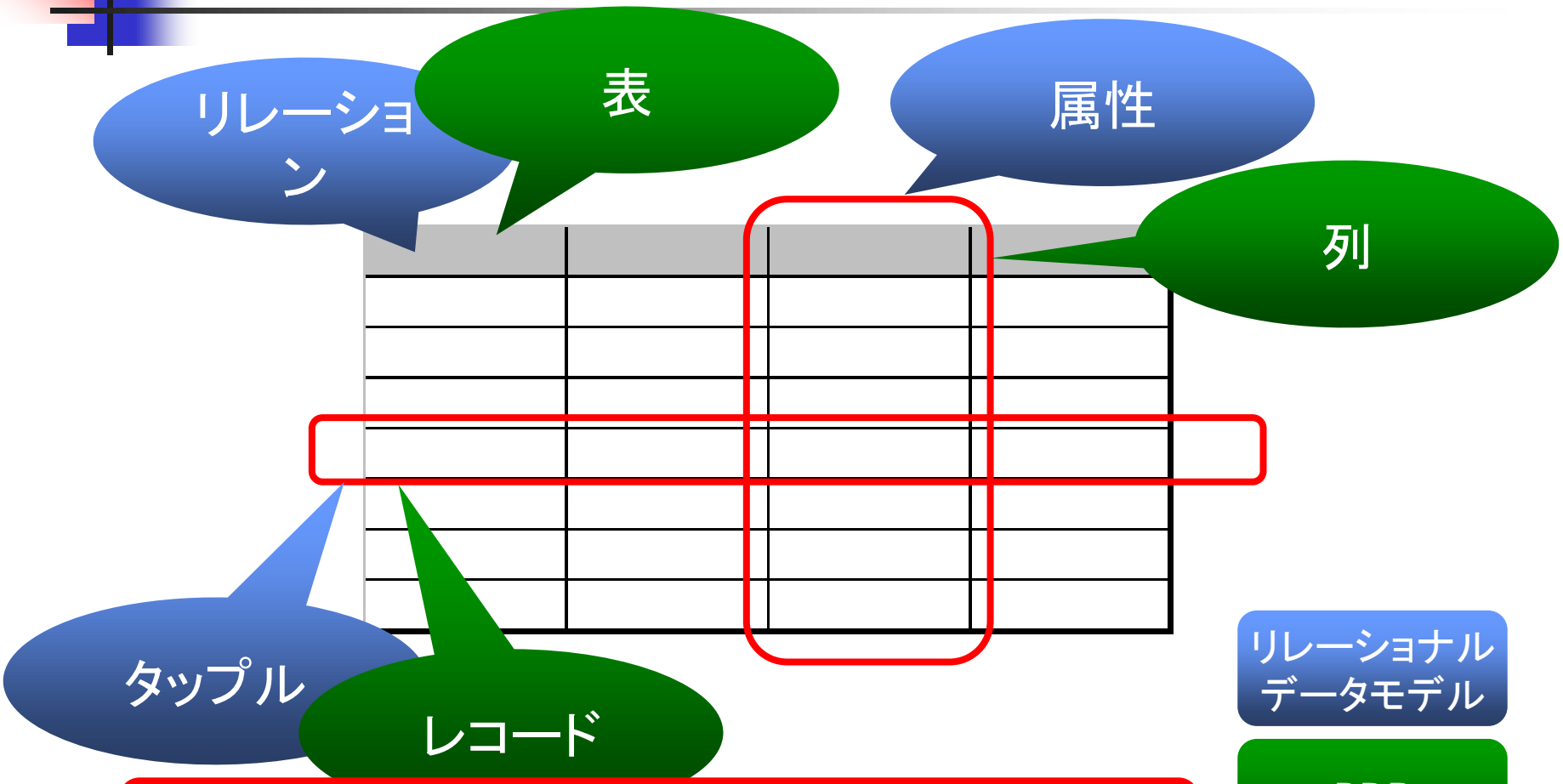
- あるデータベースについて習熟しても、別のデータベースを扱うときにまた勉強しなおさなければならない
- DBMSを変えたら、データベースを扱うソフトウェアも書き換えなければならない

など、困ることが多い

SQLの進化


マルチメディア
拡張／
XML関連拡張
RDBが持つべき
機能を
ほぼ完全に規格
化
SQL：2003
SQL：92
SQL：1999
標準化前
IBM SQL
オブジェクト指向
拡張
授業ではここを中心に説明する

RDBとリレーショナルデータモデルとの比較



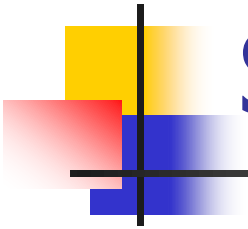
SQLの例

テーブルT_EMPから部署コードがD001な社員名を取り出す

```
SELECT 社員名
FROM   T_EMP
WHERE 部署コード='D001'
```

T_EMP

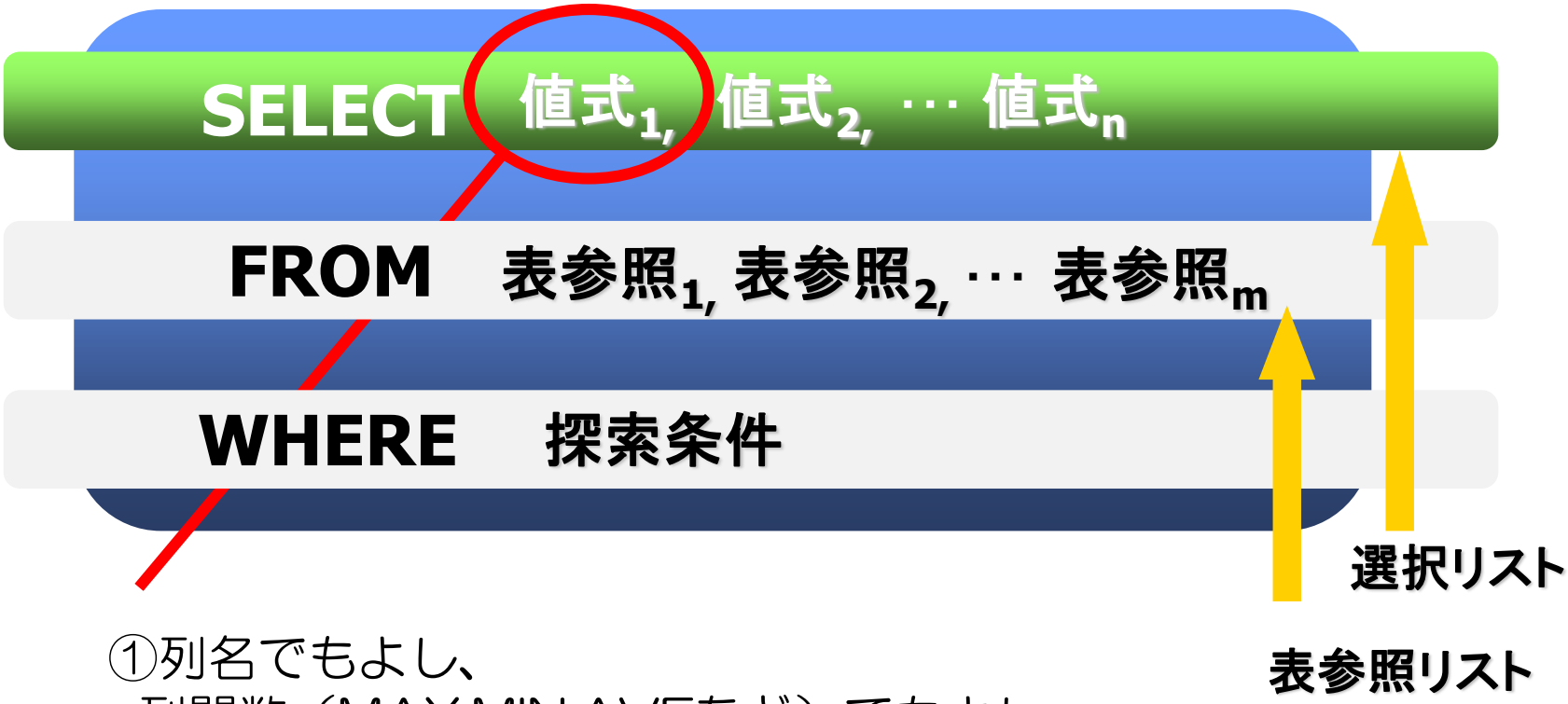
社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00003	劇団ひとり	D003
E00004	高木ブー	D004
E00005	仲本工事	D003
E00006	いかりや長介	D006
E00007	妻夫木聡	D003
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002



SQLはプログラミング言語

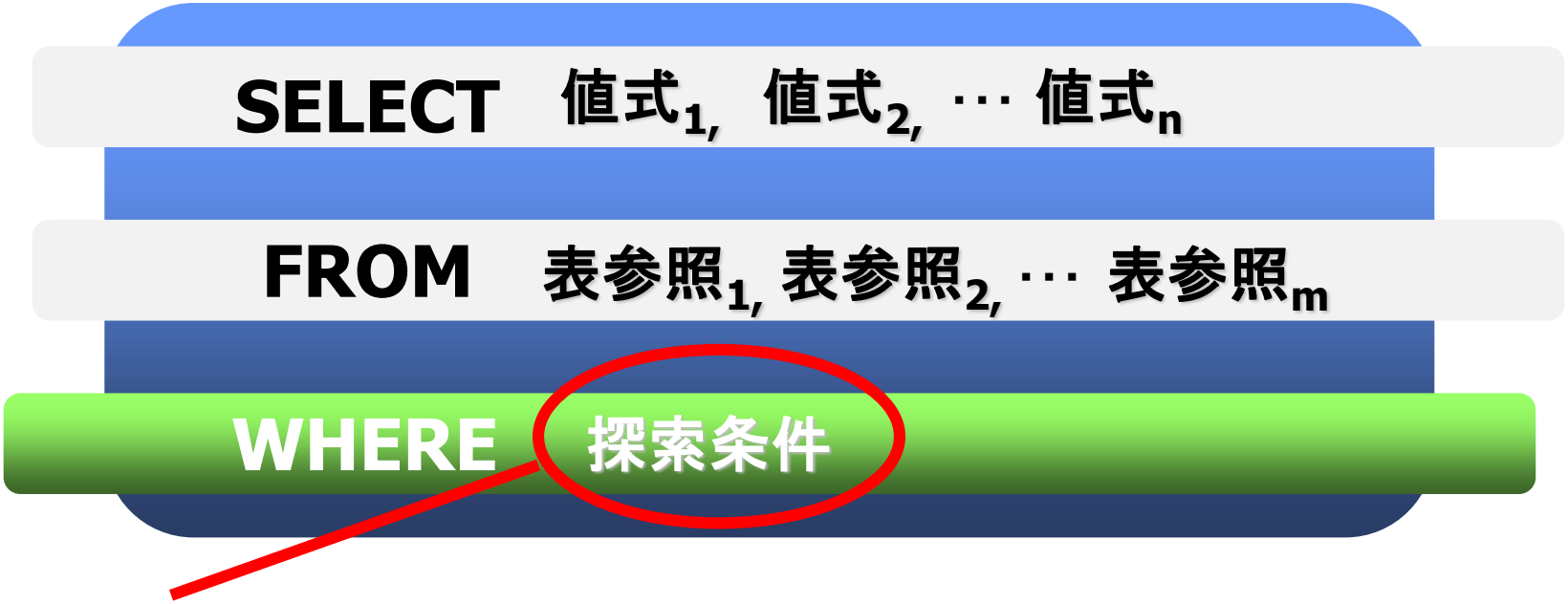
- C言語やFORTRANは手続き型言語
  - 手続き・制御構造などを駆使してプログラムを作る
  - 「どういう方法で処理するか」を記述
- SQLは非手続き型言語
  - 制御構造は基本的に記述せず、レコードを指定する条件などを記述する
  - 「何を抽出するか」を記述する

SQLによる問い合わせ（SELECT文）



①列名でもよし、
　列関数（MAX,MIN,AVEなど）でもよし、
　列名に対して演算したものでもよし
②INTEGER, CHAR, VARCHARなどの型がある

SQLによる問い合わせ（SELECT文）



列名に対し、比較演算子 >, >=, <, <=, <> だけでなく、
BETWEEN、IN、LIKE、IS (NOT) NULL、EXISTSが使える

単純質問(2)

```
SELECT  県名, 人口*1000
FROM    T_PREF
```

列名だけではなく、
式を含めることができる

T_PREF

県名	人口
宮城県	2371
埼玉県	7047
山形県	1223
神奈川県	8732
長野県	2211

結果表

県名	人口*1000
宮城県	2371000
埼玉県	7047000
山形県	1223000
神奈川県	8732000
長野県	2211000

条件(BETWEEN)

```
SELECT  県名,人口
FROM    T_PREF
WHERE 人口 BETWEEN 2000 AND 5000
```

T_PREF

県名	人口
宮城県	2371
埼玉県	7047
山形県	1223
神奈川県	8732
長野県	2211

結果表

県名	人口
宮城県	2371
長野県	2211

条件(Like)

```
SELECT　＊
FROM　　T_EMP
WHERE 社員名 like ‘加藤%’
```

T_EMP

社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00003	劇団ひとり	D003
E00004	高木ブー	D004
E00005	仲本工事	D003
E00006	いかりや長介	D006
E00007	妻夫木聡	D003
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002

結果表

社員番号	社員名	部署コード
E00002	加藤茶	D001
E00009	加藤ローサ	D002

条件(IN)

```
SELECT　＊
FROM　　T_EMP
WHERE 部署コード IN (‘D001’,’D002’)
```

T_EMP

社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00003	劇団ひとり	D003
E00004	高木ブー	D004
E00005	仲本工事	D003
E00006	いかりや長介	D006
E00007	妻夫木聡	D003
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002

結果表

社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002

ORDER BY

```
SELECT  県名,人口
FROM    T_PREF
WHERE 人口 >2000
ORDER BY 人口
```

T_PREF

県名	人口
宮城県	2371
埼玉県	7047
山形県	1223
神奈川県	8732
長野県	2211

結果表

県名	人口
長野県	2211
宮城県	2371
埼玉県	7047
神奈川県	8732

GROUP BY

```
SELECT 部署コード, count(社員名)
FROM   T_EMP
GROUP BY 部署コード
```

T_EMP

社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00003	劇団ひとり	D003
E00004	高木ブー	D004
E00005	仲本工事	D003
E00006	いかりや長介	D006
E00007	妻夫木聡	D003
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002

結果表

部署コード	COUNT(社員名)
D001	3
D002	1
D003	3
D004	1
D006	1

HAVING

```
SELECT 部署コード, count(社員名)
FROM   T_EMP
GROUP BY 部署コード
HAVING count(社員名)>2
```

T_EMP

社員番号	社員名	部署コード
E00001	志村けん	D001
E00002	加藤茶	D001
E00003	劇団ひとり	D003
E00004	高木ブー	D004
E00005	仲本工事	D003
E00006	いかりや長介	D006
E00007	妻夫木聡	D003
E00008	矢田亜希子	D001
E00009	加藤ローサ	D002

結果表

部署コード	count(社員名)
D001	3
D003	3

結合質問

商品マスタテーブル

商品コード	商品名	単価
10001	携帯電話	20000
10002	パソコン	200000
23333	プリンタ	35000
43990	ハードディスク	10000

納品テーブル

商品コード	納品先	納品数量	納
10001	A百貨店	250	
10001	B電気店	100	
10002	B電気店	100	7月22日
23333	A百貨店	50	7月21日
23333	B電気店	100	7月22日
23333	Cショップ	50	7月22日
43990	A百貨店	40	7月21日

二つのテーブルを
商品コードによって
つなげて見たい
＝結合演算

```
SELECT A.* , B.*
FROM 商品マスタ A, 納品 B
WHERE A.商品コード=B.商品コード
```

A,Bは相関名という

結合質問

商品コードを重ならないようにSELECTリストを選ぶと自然結合になる

二つのテーブルを
商品コードによって
つなげて見たい
＝結合演算

```
SELECT A.* , B.*
FROM 商品マスタ A, 納品 B
WHERE A.商品コード=B.商品コード
```

商品コード	商品名	単価	商品コード	納品先	納品数量	納品日時
10001	携帯電話	20000	10001	A百貨店	250	7月21日
10001	携帯電話	20000	10001	B電気店	100	7月22日
10002	パソコン	200000	10002	B電気店	100	7月22日
23333	プリンタ	35000	23333	A百貨店	50	7月21日
23333	プリンタ	35000	23333	B電気店	100	7月22日
23333	プリンタ	35000	23333	Cショップ	50	7月22日
43990	ハードディス	10000	43990	A百貨店	40	7月21日

商品マスタテーブル部分

納品テーブル部分

結合質問

結合質問(JOIN)は、三つ以上のテーブルに対しても可能

```
SELECT   A.商品コード, A.商品名, B.納品先, C.納品先住所
FROM 商品マスタ A, 納品 B, 住所マスタ  C
WHERE  A.商品コード=B.商品コード  AND
          B.納品先名=C.納品先名
```

結合質問(JOIN)は、ひとつのテーブルだけでも可能

```
SELECT    A.商品名, A.単価, B.商品名, B.単価
FROM 商品マスタ A, 商品マスタ B
WHERE          A.単価>B.単価
```

たとえば、単価
の高いものと
安いものを並べる場合

入れ子型質問(副照会)

- SELECT文の条件部分(WHERE句)に SELECT文が入れ子で入っているもの。
- どっちかというと、普通は副照会という。

```
SELECT   顧客番号,顧客名
FROM     顧客マスタ
WHERE    顧客番号 IN (SELECT 顧客番号
                      FROM 納品
                      WHERE 商品番号='10001')
```

納品テーブルの中にある商品’10001’を購入した顧客を調べる(納品テーブルでひっかかった顧客番号をもつ顧客マスタのレコードを調べ、顧客番号と顧客名を表示する)

データの挿入(INSERT文)

商品マスタテーブル

商品コード	商品名	単価
10001	携帯電話	20000
10002	パソコン	200000
23333	プリンタ	35000
43990	ハードディスク	10000

新製品　MP3プレーヤー（単価25000円）を登録したい

INSERT　INTO　商品マスタ　VALUES
(10003, 'MP3プレーヤー', 25000)

単価はまだ決まっていないがMP3プレーヤーを登録したい

INSERT　INTO　商品マスタ(商品コード,商品名）　VALUES
(10003, 'MP3プレーヤー')

データの挿入(INSERT文)

商品マスタテーブル

商品コード	商品名	単価
10001	携帯電話	20000
10002	パソコン	200000
23333	プリンタ	35000
43990	ハードディスク	10000

P社製製品テーブルから、データを商品マスタテーブルにとってきたい

```
INSERT  INTO  商品マスタ
(SELECT  製品コード,製品名,希望小売価格
 FROM    P社製品)
```

INSERT文にSELECT文を含めることができる
VALUESがないところに注意

データの変更(UPDATE文)

商品コード	商品名	単価
10001	携帯電話	20000
10002	パソコン	200000
23333	プリンタ	35000
43990	ハードディスク	10000

商品コード’10001’の商品の単価を15000円に修正したい

```
UPDATE 商品マスタ
SET 単価=15000
WHERE 商品コード=‘10001’
```

同じレコードの二つ以上のフィールドを変更したい場合は
SET 列名1=‘XXXX’, 列名2=‘YYYY’
のように列記する。

データの削除(DELETE文)

商品コード	商品名	単価
10001	携帯電話	20000
10002	パソコン	200000
23333	プリンタ	35000
43990	ハードディスク	10000

商品コード’10001’の商品はWITHDRAWになったので
マスタから削除したい

DELETE　FROM 商品マスタ
WHERE 商品コード=‘10001’