

Q1 List2-7

Shohin テーブルで、shohin_bunrui 列を重複を省いた形で出力する SELECT 文

A1

```
SELECT DISTINCT shohin_bunrui FROM Shohin;
```

Q2 List2-10

Shohin テーブルで、shohin_bunrui 列が'衣類'となっている行を選択する SELECT 文

A2

```
SELECT * FROM Shohin WHERE shohin_bunrui='衣類';
```

Q3 List3-13

Shohin テーブルで、shohin_bunrui 列の値ごとの行数を数える SELECT 文

要は、shohin_bunrui 列に入る値は数通りの選択肢があつて、各選択肢が選ばれた行がいくつあるのかを数えたい。

A3

```
SELECT shohin_bunrui, COUNT(*) FROM Shohin GROUP BY shohin_bunrui;
```

解説:GROUP BY によって、shohin_bunrui 列がまとめられる。そのため、集約関数である COUNT が SELECT 句にあるにもかかわらず、この列だけは選択可能になる。

Q4 List3-29

Shohin テーブルを、hanbai_tanka 列の降順で並び替えて取得する SELECT 文

A4

```
SELECT * FROM Shohin ORDER BY hanbai_tanka DESC;
```

Q5 List4-4

テーブルにデータを追加する SQL 文の構文を示せ

A5

```
INSERT INTO テーブル (列 1, 列 2, ...) VALUES (値 1, 値 2, ...);
```

Q6 List4-14

Shohin テーブルで、hanbai_tanka 列が 4000 以上の行を削除する SQL 文を示せ

A6

```
DELETE FROM Shohin WHERE hanbai_tanka >= 4000;
```

Q7 List4-16

Shohin テーブルで、shohin_bunrui 列が「キッチン用品」の行に限って hanbai_tanka 列を 10 倍に変更する SQL 文

A7

```
UPDATE Shohin
```

```
SET hanbai_tanka = hanbai_tanka * 10
```

```
WHERE shohin_bunrui = 'キッチン用品';
```

Q8 List5-12

hanbai_tanka 列の値が平均以上の行を選択する SELECT 文。サブクエリ使え

A8

```
SELECT * FROM 商品 WHERE hanbai_tanka > (SELECT AVG(hanbai_tanka) FROM  
Shohin);
```

Q9 List6-23

列 C が正規表現「.*東京電.大学.*」にマッチする文字列であるような行を、表 T から抽出する SELECT 文

A9

```
SELECT * from T WHERE C LIKE '%東京電_大学%';
```

Q10

(結合は省略)

Q11 P36

テーブルを作成する例を挙げよ

A11

```
CREATE TABLE Shohin
```

```
(shohin_id CHAR(4) NOT NULL,--列名、型、(NOT NULL)?
```

```
shohin_mei VARCHAR(100) NOT NULL,
```

```
...
```

```
PRIMARY KEY (shohin_id));--not null かつ重複不可
```

Q12

ACID 特性とは

A12

DBMS のトランザクションの約束事

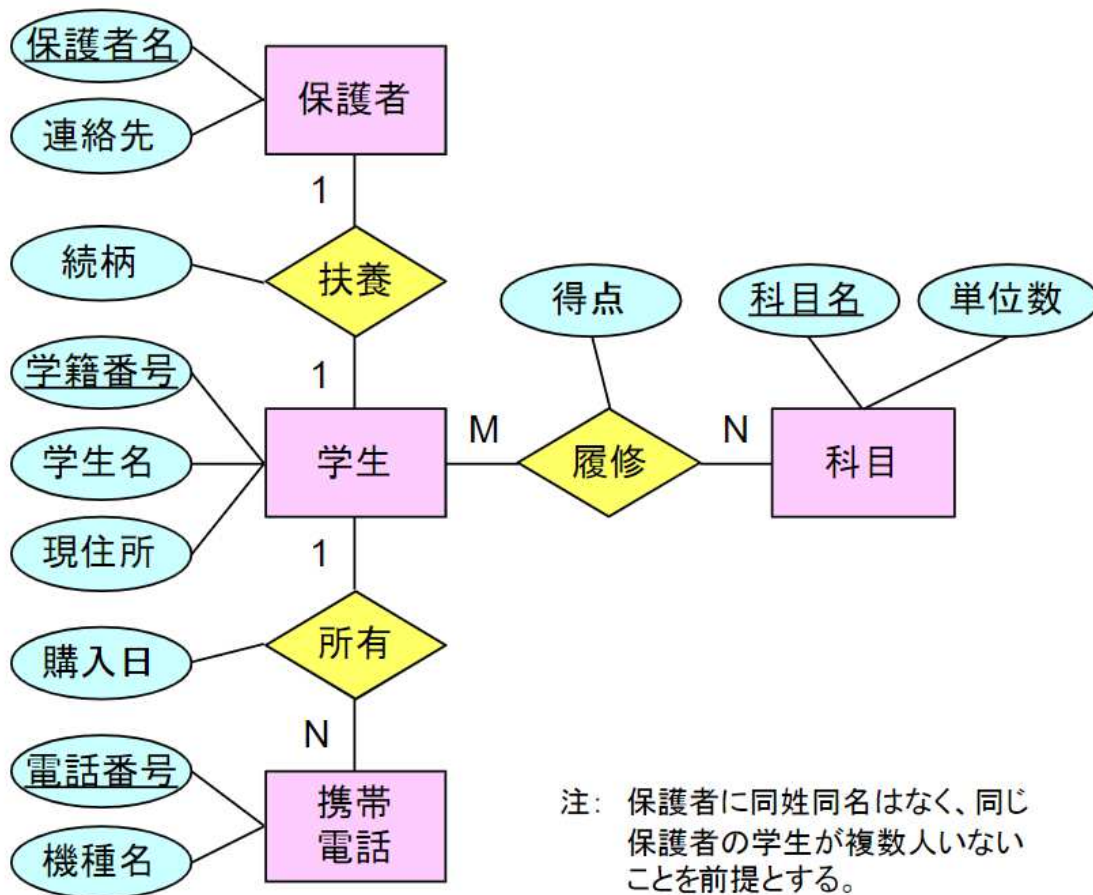
A は原子性。COMMIT か ROLL BACK

C は一貫性。違反のある SQL はロールバックされる。

I は独立性。トランザクション同士が互いに干渉しない。(よって、トランザクションの入れ子は作れない)

D は永続性。

学生が科目を履修するケース



Q14

<http://gomocool.net/gomokulog/?p=514>

表 14.1 (ア)標準形

取引予定 取引番号	取引予定日	顧客コード	顧客名	明細番号	商品コード	名称	単価	注文数	小計	合計
M002345	2015-05-01	T00113	はじまりの村 よろずや	1	W00001	ひのきの棒	100	20	2000	11900
				2	W00002	棍棒	300	10	3000	
				3	W00008	銅の剣	800	8	6400	
				4	ITM00004	薬草	50	10	500	
M002346	2015-05-01	T00255	リバー王国 騎士団	1	W00010	鋼の剣	5000	25	125000	340000
				2	A00002	鋼の鎧	7000	20	140000	
				3	D00008	鋼の盾	3000	20	60000	
				4	ITM00008	上薬草	200	50	10000	
				5	ITM00004	薬草	50	100	5000	

上の表のようなのを(ア)正規形という。

(イ)があるため、このままデータベースに登録することはできない。

そこで、(イ)を(ウ)ることで、(エ)にした次のような表を(オ)正規形という。

表 14.2 (オ)標準形

取引予定 取引番号	取引予定日	顧客コード	顧客名	明細番号	商品コード	名称	単価	注文数	小計	合計
M002345	2015-05-01	T00113	はじまりの村 よろずや	1	W00001	ひのきの棒	100	20	2000	11900
M002345	2015-05-01	T00113	はじまりの村 よろずや	2	W00002	棍棒	300	10	3000	11900
M002345	2015-05-01	T00113	はじまりの村 よろずや	3	W00008	銅の剣	800	8	6400	11900
M002345	2015-05-01	T00113	はじまりの村 よろずや	4	ITM00004	薬草	50	10	500	11900
M002346	2015-05-01	T00255	リバー王国 騎士団	1	W00010	鋼の剣	5000	25	125000	340000
M002346	2015-05-01	T00255	リバー王国 騎士団	2	A00002	鋼の鎧	7000	20	140000	340000
M002346	2015-05-01	T00255	リバー王国 騎士団	3	D00008	鋼の盾	3000	20	60000	340000
M002346	2015-05-01	T00255	リバー王国 騎士団	4	ITM00008	上薬草	200	50	10000	340000
M002346	2015-05-01	T00255	リバー王国 騎士団	5	ITM00004	薬草	50	100	5000	340000

が、よみにくくなってしまった。

ここで、(カ)を探そう。(カ)とは、(キ)のことであり、今回は(ク)だ。

(ク)で、(ケ)情報がある。それを切り出したものが(コ)標準形だ。

表 14.3 (コ)標準形

取引明細 取引番号	明細番号	商品コード	名称	単価	注文数	小計
M002345	1	W00001	ひのきの棒		100	2000
M002345	2	W00002	棍棒		300	3000
M002345	3	W00008	銅の剣		800	6400
M002345	4	ITM00004	薬草		50	500
M002346	1	W00010	鋼の剣		5000	125000
M002346	2	A00002	鋼の鎧		7000	140000
M002346	3	D00008	鋼の盾		3000	60000
M002346	4	ITM00008	上薬草		200	10000
M002346	5	ITM00004	薬草		50	5000
取引予定 取引番号	取引予定日	顧客コード	顧客名	合計		
M002345	2015-05-01	T00113	はじまりの村 よろずや	11900		
M002346	2015-05-01	T00255	リバー王国 騎士団	340000		

(コ)を、さらに「積極的に」切り分けてみよう。

例えば、取引番号と明細番号が決まれば商品コード、名称、単価、注文数、小計が決まるが、そもそも(サ)わけじゃないか。ここが無駄なので、切り分けよう。これをやったものが、(シ)正規形である。

表 14.4 (シ) 正規形

取引明細			
取引番号	明細番号	商品コード	注文数
M002345	1	W00001	20
M002345	2	W00002	10
M002345	3	W00008	8
M002345	4	ITM00004	10
M002346	1	W00010	25
M002346	2	A00002	20
M002346	3	D00008	20
M002346	4	ITM00008	50
M002346	5	ITM00004	100
取引予定			
取引番号	取引予定日	顧客コード	
M002345	2015-05-01	T00113	
M002346	2015-05-01	T00255	
商品			
商品コード	名称	単価	
W00001	ひのきの棒	100	
W00002	棍棒	300	
W00008	銅の剣	800	
ITM00004	薬草	50	
W00010	鋼の剣	5000	
A00002	鋼の鎧	7000	
D00008	鋼の盾	3000	
ITM00008	上薬草	200	
ITM00004	薬草	50	
顧客			
顧客コード	顧客名		
T00113	はじまりの村 よろずや		
T00255	リバー王国 騎士団		

A14

(ア): 非

(イ): 結合したセル

(ウ): 1 行 1 列ずつにわけ

(エ): 無理やりデータベースに登録できるよう

(オ): 第 1

(カ): 主キー

(キ): 各行を一意に特定するために最低限必要な情報

(ク): 取引番号と明細番号の組み合わせ

(ケ): その一部のキーだけでも特定できる

(コ): 第 2

(サ): 商品コードが決まれば、名称も単価も決まる

(シ): 第 3

Q15

複数のトランザクションをスケジュールAに従って実行したときに、その結果がスケジュールBに(ア)になる場合に、二つのスケジュールA、Bは等価であると定義する。

直列化可能とは、トランザクションを同時実行するスケジュール((イ))と(ウ)が(エ)な場合に、その非直列スケジュールは直列化可能という。

一般に直列スケジュールは複数あるので、等価な直列スケジュールが一つでもあれば直列化可能とする。

A15

(ア): 従って実行したときと全く同じ

(イ): 非直列スケジュール

(ウ): トランザクションを直列実行するスケジュール (直列スケジュール)

(エ): 等価

Q16

時刻	トランザクション T_1	トランザクション T_2
t_1	read(x)	
t_2		read(y)
t_3	write(x)	
t_4		read(x)
t_5		write(y)

直列化せよ。

A16

時刻	トランザクション T_1	トランザクション T_2
t_1	read(x)	
t_2	write(x)	
t_3		read(y)
t_4		read(x)
t_5		write(y)

Q17

次のものが直列化可能か、直列化可能なら等価な直列スケジュールの順番を求めたい。

表 17.1

時刻	T_1	T_2	T_3	T_4	T_5
t_1					read(x)
t_2		read(x)			
t_3					write(x)
t_4	read(x)				
t_5				read(x)	
t_6		read(y)			
t_7	write(x)				
t_8			read(x)		
t_9		write(y)			
t_{10}				read(y)	
t_{11}			write(x)		
t_{12}				write(y)	

これを求めるためには、(ア)を考える必要がある。

(ア)とは、(イ)だ。

例えば(ウ,エ|順不同)は(ア)といえる。

(ア)なものを(オ)ものを(カ)ものを(ア)グラフという。

(ア)グラフで、(キ)ることができれば直列化可能であるし、その時の順番こそ等価な直列スケジュールだ。

A17

(ア): 相反

(イ): 実行順序によって処理結果が変わりうる基本操作の対

(ウ): read(x)と write(x)

(エ): write(x)と write(x)

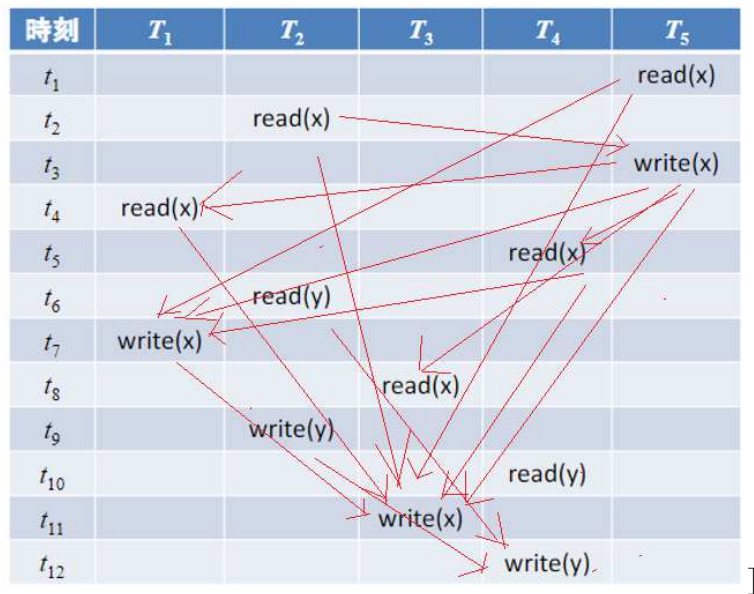
(オ): スケジュール表上で、下向き矢印でつないだ

(カ): 参考に、Ti の状態遷移図(自分に戻るのは不要)っぽく Ti と矢印だけで表したグラフ

(キ): 矢印に従って一筆書きしたときにすべての Ti をつなげ

例:表 17.1 に相反を結ぶ下向き矢印を書き加えると次のようになる。

表 17.2



相反グラフを作ると

$$T_2 \rightarrow T_3$$

$$T_2 \rightarrow T_3$$

$$T_4$$

$$T_5$$

$$T_3 \rightarrow X \text{ (これはおかしい)}$$

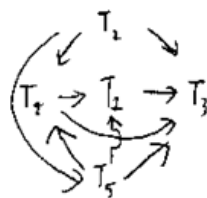
$$T_4 \rightarrow T_1$$

$$T_3$$

$$T_5 \rightarrow T_1$$

$$T_3$$

$$T_4$$



よって、 $T_2 \rightarrow T_5 \rightarrow T_4 \rightarrow T_1 \rightarrow T_3$