

de-13. SQL 総合演習, トランザクション

(データベース演習)

URL: <https://www.kkaneko.jp/cc/de/index.html>

金子邦彦



アウトライン

番号	項目
13-1	演習① SQL によるテーブル定義
13-2	演習② SQL によるレコード挿入, トランザクション, ロールバック, コミット
13-3	演習③ SQL による問い合わせ
13-4	演習④ SQL による結合

各自、資料を読み返したり、課題に取り組んだりも行う

リレーショナルデータベースシステム

データベースシステムの一種

コンピュータ



記憶装置

リレーショナルデータベース管理システム リレーショナルデータベース

ID	名前	単価
1	みかん	50
2	りんご	100
3	りんご	150

ID	購入者	商品ID	数量
1	X	1	10
2	Y	2	5

たくさんの**テーブル**が格納される

あわせて

リレーショナルデータベースシステム

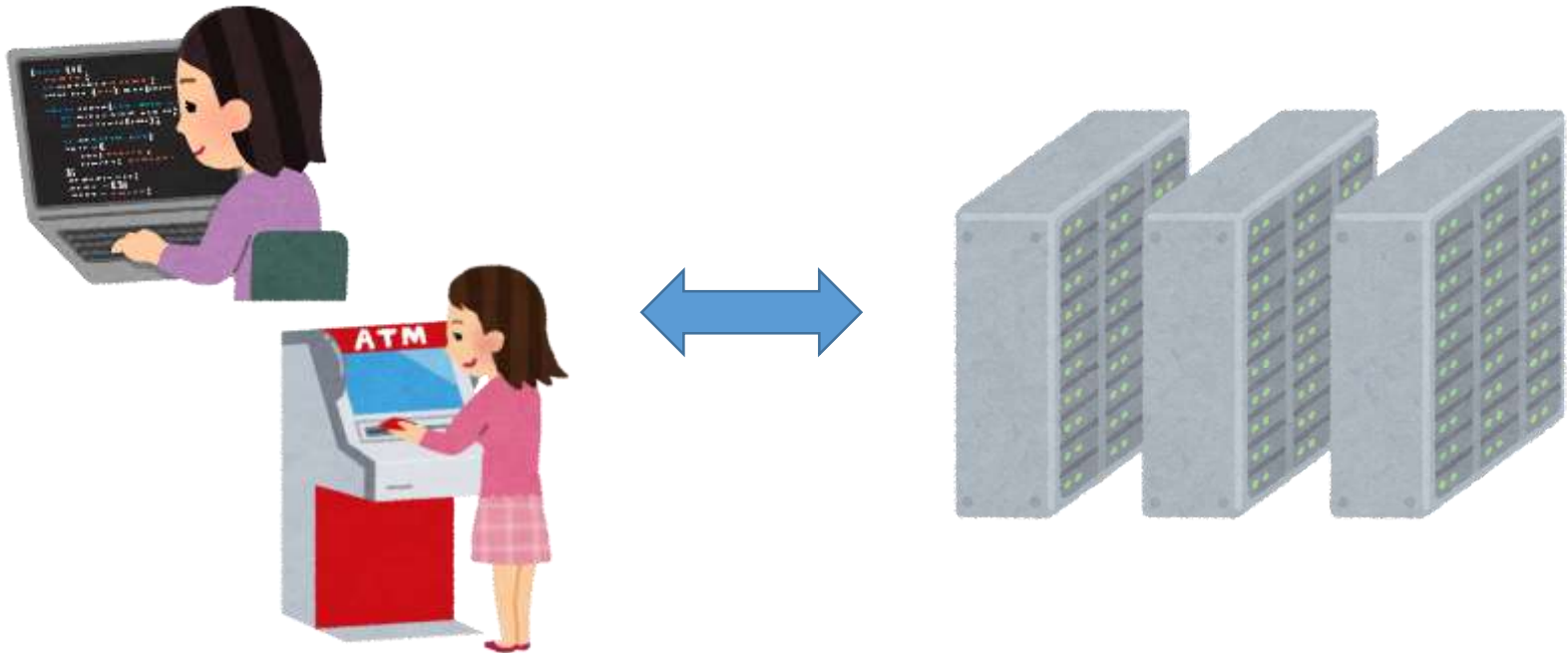
SQL

- **SQL** は、**リレーショナルデータベースシステム**のさまざまな機能を使える言語

問い合わせ（クエリ）、

テーブル定義、

その他の操作



SQL による問い合わせ（クエリ）の例

- ① **SELECT * FROM** 商品;
- ② **SELECT** 名前, 単価 **FROM** 商品;
- ③ **SELECT** 名前, 単価 **FROM** 商品 **WHERE** 単価 > 80;
- ④ **SELECT** 受講者, **COUNT(*) FROM** 成績 **GROUP BY** 受講者;
- ⑤ **SELECT * FROM** 米国成人調査データ **ORDER BY** 年齢;
- ⑥ **SELECT * FROM** T, S;
- ⑦ **SELECT * FROM** T, S **WHERE** a = b;

テーブル定義

リレーショナルデータベースのテーブル定義は、

- ・ **テーブル名**
- ・ 各属性の**属性名**
- ・ 各属性の**データ型**

などを定義すること。

主キーの指定 (primary key) を行う場合がある。参照整合性制約などの**制約**の指定を行う場合がある

SQL コマンドを用いたテーブル定義

■ SQLite システムの場合

```
1 create table 商品 (  
2     ID integer,  
3     商品 char,  
4     単価 integer  
5 );  
6
```

テーブル名：商品

属性名とデータ型
ID INTEGER
商品 CHAR
単価 INTEGER

■ マイクロソフト Access の場合

```
クエリ1  
create table 商品 (  
ID integer,  
商品名 char,  
単価 integer  
)
```

属性のデータ型

Access の主なデータ型	SQL のキーワード	
	NULL	空値
短いテキスト	CHAR	文字列
長いテキスト	TEXT	文字列
数値	INTEGER, REAL	整数や浮動小数点数
日付／時刻	DATETIME	日付や時刻など
Yes／No	BIT, BOOL	ブール値

※ **整数**は INTEGER, **浮動小数点数**（小数付きの数）は REAL

※ **短いテキスト**は半角 255文字分までが目安
それ以上になる可能性があるときは**長いテキスト**

挿入

テーブル T

名前	昼食	料金

空



テーブル T

名前	昼食	料金
A	そば	250
B	カレーライス	400
C	カレーライス	400
D	うどん	250

○ 実行例（テーブル定義， 3 行の挿入， 確認）

```
1 create table T(名前 text, 昼食 text, 料金 integer);
2 insert into T values('A', 'そば', 250);
3 insert into T values('B', 'カレーライス', 400);
4 insert into T values('C', 'カレーライス', 400);
5 insert into T values('D', 'うどん', 250);
6 select * from T;
```

名前	昼食	料金
A	そば	250
B	カレーライス	400
C	カレーライス	400
D	うどん	250

トランザクション

- データベースを扱うプログラムは、普通のプログラムと違う
 - A : 作業途中の結果をそのまま残さない
 - C : データベースの異状を防ぐ仕組み
 - I : データベースを複数人が同時使用できる。
 - D : 作業が完了したら、そのデータは残る
- 特性 A, C, I, D を持つプログラムのことを**トランザクション**と呼んだりする

ロールバック : データベースをトランザクション開始時点に戻す

コミット : トランザクションの正常終了

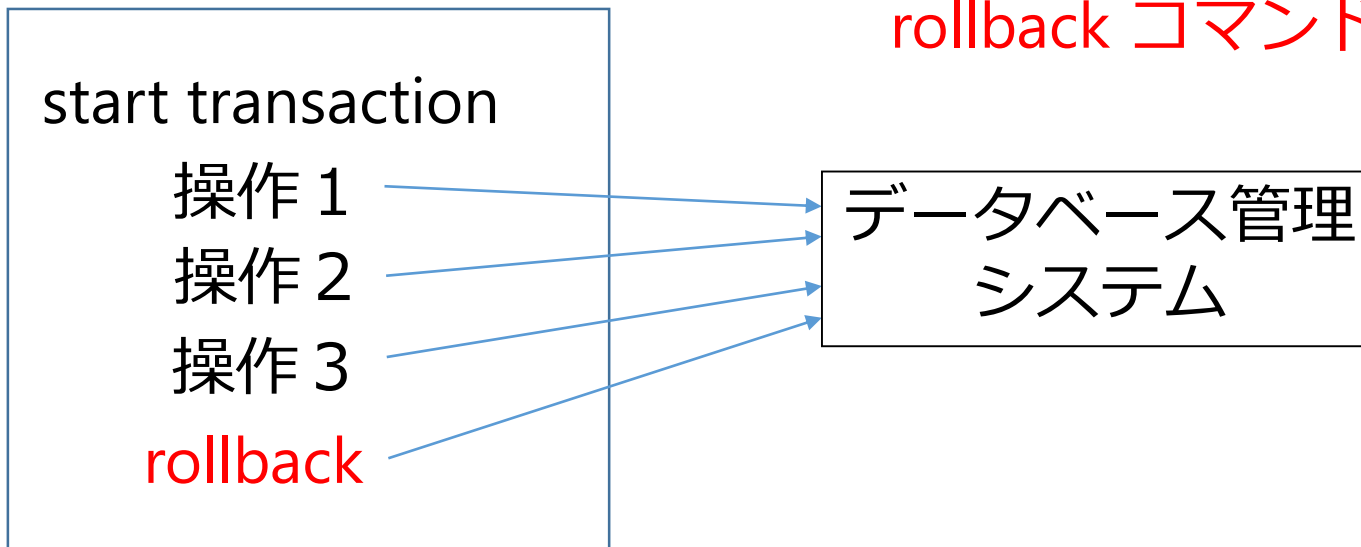
トランザクションの開始, ロールバック, コミット

- MySQL でのトランザクションの開始
start transaction
※ 他のシステムでは「**begin transaction**」となることもある
- ロールバック (データベースをトランザクション開始時点に戻す)
rollback
- コミット (トランザクションの正常終了)
commit

ロールバック (rollback) のイメージ

操作 1、操作 2、操作 3
と操作していて、
最初に戻したくなったら . . .

rollback コマンド



Paiza.IO の使い方

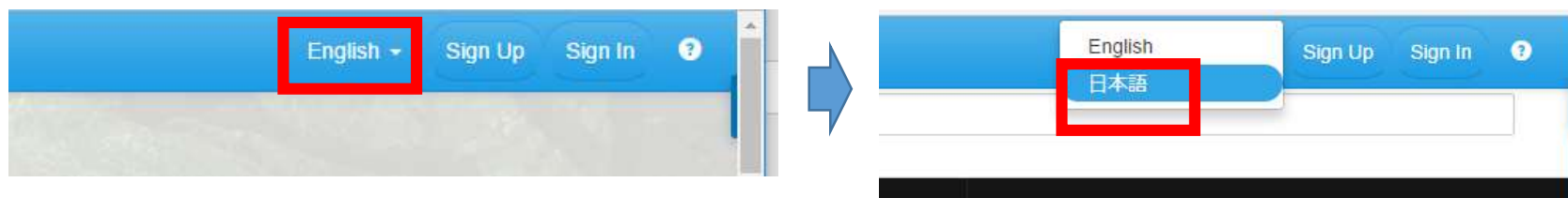
① ウェブブラウザを起動する

② 次の URL を開く

<https://paiza.io/>



③ もし、表示が英語になっていたら、**日本語**に切り替える



④ 「コード作成を試してみる」をクリック



⑤ 「MySQL」を選ぶ (左上のボタンをクリックするとメニューが出る)



プログラムの
編集画面

プログラムを
書き換えること
ができる

実行ボタン

← → ↻ <https://paiza.io/ja/projects/new>

Beta paiza.io

MySQL Enter a title here

Main sql

```
1 create table Test(id integer, title varchar(100));
2 insert into Test(id, title) values(1, "Hello");
3 select * from Test;
4 -- Your code here!
5 |
6
```

実行 (Ctrl-Enter)

編集画面を確認する。

すでに、**SQLが入っている**が、使わないので**消す**。

```
main.sql
1 create table Test(id integer, title varchar(100));
2 insert into Test(id, title) values(1, "Hello");
3 select * from Test;
4 -- Your code here!
5 |
6
```


13-1. 演習① SQL によるテーブル定義

いまから作成するテーブル products

id	name	type	price	at
1	apple	red	100	2023-01-06 02:31:54
2	apple	yellow	200	2023-01-06 02:31:54
3	orange	orange	300	2023-01-06 02:31:54

整数

テキスト

テキスト

整数

日時

↑ 実行した日時を格納する
(世界標準時. 9時間の時差)

- テキスト (文字属性) には **TEXT** を使用
- 整数には **INTEGER** を使用
- 日時には **DATETIME** を使用

SQL を用いたテーブル定義

【SQLプログラム】

```
create table products (  
  id integer primary key,  
  name text,  
  type text,  
  price integer,  
  at datetime  
) engine = innnoDB;
```

実行し、エラーが表示されないことを確認せよ（プログラムを消さないこと）



```
1 create table products (  
2   id integer primary key,  
3   name text,  
4   type text,  
5   price integer,  
6   at datetime  
7 ) engine = innnoDB;  
8
```

実行 (Ctrl-Enter)

出力 入力 コメント 0

データ型

空値 **NULL**

テキスト（文字属性） **char, text**

数値 **integer, real**

日付や時刻 **datetime**

ブール値 **bool** など

primary key

主キーの指定

engine = innnoDB

MySQL で制約の機能を使いたいとき

13-2. 演習② SQLによるレコード挿入, トランザクション, ロールバック, コミット

SQL を用いた行の挿入

【SQL プログラム】

次のプログラムを書き加える

```
insert into products values(1, 'apple', 'red', 100, now());  
insert into products values(2, 'apple', 'yellow', 200, now());  
insert into products values(3, 'orange', 'orange', 300, now());  
select * from products;
```

次ページに続く

レコード (行) の挿入
insert into, values
現在日時の取得
now

SQL を用いた行の挿入

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innnoDB;  
8  
9 insert into products values(1, 'apple', 'red', 100, now());  
10 insert into products values(2, 'apple', 'yellow', 200, now());  
11 insert into products values(3, 'orange', 'orange', 300, now());  
12 select * from products;
```

実行 (Ctrl-Enter)

MySQLを学ぶ | プログラミングカ診断

出力 入力 コメント 0

id	name	type	price	at
1	apple	red	100	2023-01-06 02:31:54
2	apple	yellow	200	2023-01-06 02:31:54
3	orange	orange	300	2023-01-06 02:31:54

トランザクションとロールバック

【SQLプログラム】

次のプログラムを書き加える

```
start transaction;  
insert into products values(4, 'melon', 'green', 500, now());  
select * from products;  
rollback;  
select * from products;
```

次ページに続く

MySQLでのトランザクションの開始

start transaction

ロールバック

rollback

コミット（トランザクションの正常終了）

commit

トランザクションとロールバック

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innodb;  
8  
9 insert into products values(1, 'apple', 'red', 100, now());  
10 insert into products values(2, 'apple', 'yellow', 200, now());  
11 insert into products values(3, 'orange', 'orange', 300, now());  
12 select * from products;  
13  
14 start transaction;  
15 insert into products values(4, 'melon', 'green', 500, now());  
16 select * from products;  
17 rollback;  
18 select * from products;  
19
```

実行 (Ctrl-Enter)

MySQLを学ぶ | プログラミング力診断

出力 入力 コメント 0

id	name	type	price	at
1	apple	red	100	2023-01-06 02:35:07
2	apple	yellow	200	2023-01-06 02:35:07
3	orange	orange	300	2023-01-06 02:35:07

id	name	type	price	at
1	apple	red	100	2023-01-06 02:35:07
2	apple	yellow	200	2023-01-06 02:35:07
3	orange	orange	300	2023-01-06 02:35:07
4	melon	green	500	2023-01-06 02:35:07

id	name	type	price	at
1	apple	red	100	2023-01-06 02:35:07
2	apple	yellow	200	2023-01-06 02:35:07
3	orange	orange	300	2023-01-06 02:35:07



結果が変わる
(元に戻る)

トランザクションとコミット

【SQLプログラム】

「rollback」を「commit」に書き換える

```
start transaction;  
insert into products values(4, 'melon', 'green', 500, now());  
select * from products;  
commit;  
select * from products;
```

次ページに続く

MySQLでのトランザクションの開始

start transaction

ロールバック

rollback

コミット（トランザクションの正常終了）

commit

トランザクションとコミット

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innodb;  
8  
9 insert into products values(1, 'apple', 'red', 100, now());  
10 insert into products values(2, 'apple', 'yellow', 200, now());  
11 insert into products values(3, 'orange', 'orange', 300, now());  
12 select * from products;  
13  
14 start transaction;  
15 insert into products values(4, 'melon', 'green', 500, now());  
16 select * from products;  
17 commit;  
18 select * from products;  
19
```

実行 (Ctrl-Enter) MySQLを学ぶ | プログラミング力診断

出力 入力 コメント 0

id	name	type	price	at
1	apple	red	100	2023-01-06 02:36:00
2	apple	yellow	200	2023-01-06 02:36:00
3	orange	orange	300	2023-01-06 02:36:00

id	name	type	price	at
1	apple	red	100	2023-01-06 02:36:00
2	apple	yellow	200	2023-01-06 02:36:00
3	orange	orange	300	2023-01-06 02:36:00
4	melon	green	500	2023-01-06 02:36:00

id	name	type	price	at
1	apple	red	100	2023-01-06 02:36:00
2	apple	yellow	200	2023-01-06 02:36:00
3	orange	orange	300	2023-01-06 02:36:00
4	melon	green	500	2023-01-06 02:36:00



結果が変わらない 26

13-3. 演習③ SQL による問い 合わせ

問い合わせ

【前準備】

次の SQL は残す。 その下はすべて消す。

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innnoDB;  
8 insert into products values(1, 'apple', 'red', 100, now());  
9 insert into products values(2, 'apple', 'yellow', 200, now());  
10 insert into products values(3, 'orange', 'orange', 300, now());
```

【SQL プログラム】

次のプログラムを 書き加える

```
select * from products where price > 200;
```

次ページに続く

問い合わせ

実行し、結果を確認せよ（実行後、プログラムを消さないこと）

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innodb;  
8 insert into products values(1, 'apple', 'red', 100, now());  
9 insert into products values(2, 'apple', 'yellow', 200, now());  
10 insert into products values(3, 'orange', 'orange', 300, now());  
11 select * from products where price > 200;  
12
```

▶ 実行 (Ctrl-Enter)

MySQLを学ぶ | プログラミングカ診断

出力 入力 コメント 0

id	name	type	price	at
3	orange	orange	300	2023-01-06 02:38:54

文字列のマッチング

【SQL プログラム】

次のプログラムを書き加える

```
select * from products where name like 'ap%';
```

次ページに続く

文字列のマッチング

実行し、結果を確認せよ

「like 'ap%」は、
先頭が「ap」の文字列

```
1 create table products (  
2     id integer primary key,  
3     name text,  
4     type text,  
5     price integer,  
6     at datetime  
7 ) engine = innodb;  
8 insert into products values(1, 'apple', 'red', 100, now());  
9 insert into products values(2, 'apple', 'yellow', 200, now());  
10 insert into products values(3, 'orange', 'orange', 300, now());  
11 select * from products where price > 200;  
12 select * from products where name like 'ap%';  
13
```

実行 (Ctrl-Enter)

MySQLを学ぶ | プログラミングカ診断

出力 入力 コメント 0

id	name	type	price	at
3	orange	orange	300	2023-01-06 02:39:47
id	name	type	price	at
1	apple	red	100	2023-01-06 02:39:47
2	apple	yellow	200	2023-01-06 02:39:47

ここまでのまとめ

- テーブル定義

CREATE TABLE ...

- 問い合わせ (クエリ)

SELECT ... FROM ...

SELECT ... FROM ... WHERE ...

- 文字列のマッチング

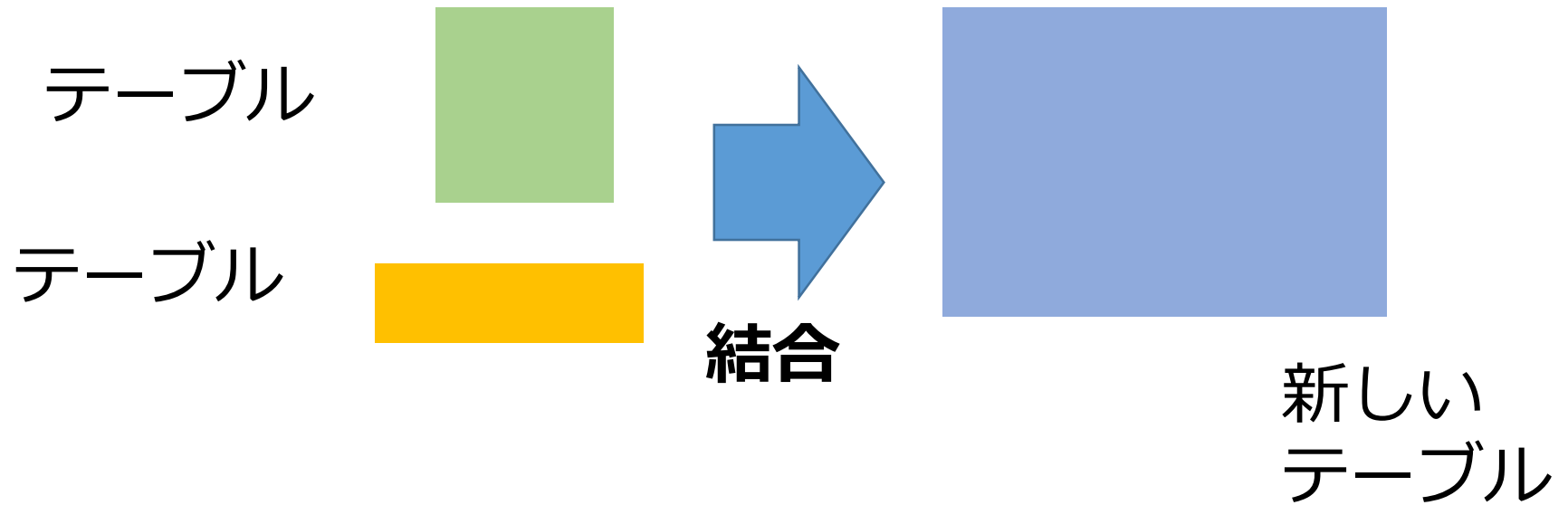
%, LIKE を使用

- レコードの挿入

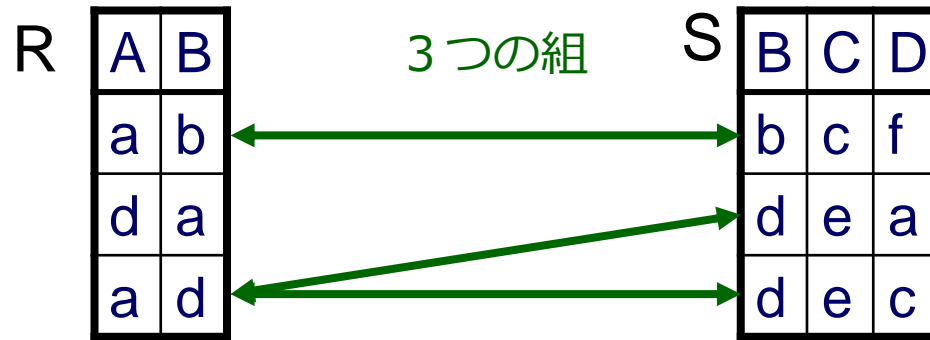
INSERT INTO ...

13-4. 演習④ SQL による結合

結合のイメージ



結合の例



結合条件 $R.B = S.B$



結合

A	R.B	S.B	C	D
a	b	b	c	f
a	d	d	e	a
a	d	d	e	c

テーブル R

【前準備】

今まで使っていた SQL プログラムは、もう使わないので、すべて消す

【SQL プログラム】

```
create table R (  
  A text,  
  B text  
) engine = innoDB;  
insert into R values('a','b');  
insert into R values('d','a');  
insert into R values('a','d');  
select * from R;
```

次ページに続く

テーブル R

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table R (  
2   A text,  
3   B text  
4 ) engine = innodb;  
5 insert into R values('a','b');  
6 insert into R values('d','a');  
7 insert into R values('a','d');  
8 select * from R;  
9
```

 実行 (Ctrl-Enter) [MySQLを学ぶ](#)

出力	入力	コメント
A	B	
a	b	
d	a	
a	d	

R

A	B
a	b
d	a
a	d

テーブル S

【SQL プログラム】

次のプログラムを書き加える

```
create table S (  
  B text,  
  C text,  
  D text  
) engine = innoDB;  
insert into S values('b', 'c', 'f');  
insert into S values('d', 'e', 'a');  
insert into S values('d', 'e', 'c');  
select * from S;
```

次ページに続く

テーブル S

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table R (  
2   A text,  
3   B text  
4 ) engine = innodb;  
5 insert into R values('a','b');  
6 insert into R values('d','a');  
7 insert into R values('a','d');  
8 select * from R;  
9  
10 create table S (  
11   B text,  
12   C text,  
13   D text  
14 ) engine = innodb;  
15 insert into S values('b','c','f');  
16 insert into S values('d','e','a');  
17 insert into S values('d','e','c');  
18 select * from S;  
19
```

実行 (Ctrl-Enter) MySQLを学ぶ | プ

出力 入力 コメント 0

A	B	
a	b	
d	a	
a	d	
B	C	D
b	c	f
d	e	a
d	e	c

S	B	C	D
	b	c	f
	d	e	a
	d	e	c

結合の例① 結合条件なしの結合

【SQL プログラム】

次のプログラムを書き加える

```
select * from R, S;
```

次ページに続く

結合の例① 結合条件なしの結合

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table R (  
2   A text,  
3   B text  
4 ) engine = innodb;  
5 insert into R values('a','b');  
6 insert into R values('d','a');  
7 insert into R values('a','d');  
8 select * from R;  
9  
10 create table S (  
11   B text,  
12   C text,  
13   D text  
14 ) engine = innodb;  
15 insert into S values('b','c','f');  
16 insert into S values('d','e','a');  
17 insert into S values('d','e','c');  
18 select * from S;  
19 select * from R, S;
```

実行 (Ctrl-Enter) MySQLを学ぶ | プロ

出力 入力 コメント 0

A	B	B	C	D
a	b	b	c	f
d	a	b	c	f
a	b	b	c	f
a	d	d	e	a
d	a	d	e	a
a	b	d	e	a
a	d	d	e	c
d	a	d	e	c
a	b	d	e	c

結合の結果

結合の例② 結合条件は「R.B = S.B」

【SQL プログラム】

最後の行のプログラムを次のように書き換える

```
select * from R, S where R.B = S.B;
```

次ページに続く

結合の例② 結合条件は「R.B = S.B」

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table R (  
2   A text,  
3   B text  
4 ) engine = innodb;  
5 insert into R values('a','b');  
6 insert into R values('d','a');  
7 insert into R values('a','d');  
8 select * from R;  
9  
10 create table S (  
11   B text,  
12   C text,  
13   D text  
14 ) engine = innodb;  
15 insert into S values('b','c','f');  
16 insert into S values('d','e','a');  
17 insert into S values('d','e','c');  
18 select * from S;  
19 select * from R, S where R.B = S.B;
```

実行 (Ctrl-Enter) MySQLを学ぶ | プログラミング力診断

出力 入力 コメント 0

A	B
a	b
d	a
a	d

B	C	D
b	c	f
d	e	a
d	e	c

A	B	B	C	D
a	b	b	c	f
a	d	d	e	a
a	d	d	e	c

結合の結果

結合の例③ 結合条件は「R.B = S.B」、 そして「C = 'e'」で選択

【SQL プログラム】

最後の行のプログラムを次のように書き換える

```
select * from R, S where R.B = S.B and C = 'e';
```

次ページに続く

結合の例③ 結合条件は「R.B = S.B」、 そして「C = 'e'」で選択

実行し、結果を確認せよ（プログラムを消さないこと）

```
1 create table R (  
2   A text,  
3   B text  
4 ) engine = innodb;  
5 insert into R values('a','b');  
6 insert into R values('d','a');  
7 insert into R values('a','d');  
8 select * from R;  
9  
10 create table S (  
11   B text,  
12   C text,  
13   D text  
14 ) engine = innodb;  
15 insert into S values('b', 'c', 'f');  
16 insert into S values('d', 'e', 'a');  
17 insert into S values('d', 'e', 'c');  
18 select * from S;  
19 select * from R, S where R.B = S.B and C = 'e';
```

実行 (Ctrl-Enter)

MySQLを学ぶ | プログラミング

出力 入力 コメント 0

A	B			
a	b			
d	a			
a	d			
B	C	D		
b	c	f		
d	e	a		
d	e	c		
A	B	B	C	D
a	d	d	e	c
a	d	d	e	a

結合の結果