

dd-7. テーブルの 分解と結合

リレーショナルデータベースの
基本（短縮版）（全7回）

基本を把握したい人へ

<https://www.kkaneko.jp/data/dd/index.html>

金子邦彦



謝辞：この資料では「かわいいフリー素材集 いらすとや」のイラストを使用しています

第7回のアウトライン

次のことについて, Paiza.IO を用いた演習

- テーブルの分解
- テーブルの結合

Paiza.IO の URL: <https://paiza.io/>

テーブル定義 scores

1から6行目に、次の **SQL** を書いて、「実行」をクリック。エラーメッセージが出ないことを確認

【SQLプログラム】

```
CREATE TABLE scores (  
  id INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  teacher_name TEXT NOT NULL,  
  student_name TEXT NOT NULL,  
  score INTEGER);
```

```
1 CREATE TABLE scores (  
2   id INTEGER PRIMARY KEY NOT NULL,  
3   name TEXT NOT NULL,  
4   teacher_name TEXT NOT NULL,  
5   student_name TEXT NOT NULL,  
6   score INTEGER);
```

レコードの挿入



7から11行目に、次のSQLを書き加え、「実行」をクリック。エラーメッセージが出ないことを確認

【SQLプログラム】

```
INSERT INTO scores VALUES(1, 'db', 'k', 'kk', 85);
INSERT INTO scores VALUES(2, 'db', 'k', 'aa', 75);
INSERT INTO scores VALUES(3, 'db', 'k', 'nn', 90);
INSERT INTO scores VALUES(4, 'python', 'a', 'kk', 85);
INSERT INTO scores VALUES(5, 'python', 'a', 'nn', 75);
```

```
7  INSERT INTO scores VALUES(1, 'db', 'k', 'kk', 85);
8  INSERT INTO scores VALUES(2, 'db', 'k', 'aa', 75);
9  INSERT INTO scores VALUES(3, 'db', 'k', 'nn', 90);
10 INSERT INTO scores VALUES(4, 'python', 'a', 'kk', 85);
11 INSERT INTO scores VALUES(5, 'python', 'a', 'nn', 75);
```

【SQL プログラム】

```
SELECT * FROM scores;
```

id	name	teacher_name	student_name	score
1	db	k	kk	85
2	db	k	aa	75
3	db	k	nn	90
4	python	a	kk	85
5	python	a	nn	75

【SQL プログラム】

```
SELECT name, teacher_name  
FROM scores;
```

name	teacher_name
db	k
db	k
db	k
python	a
python	a

【SQL プログラム】

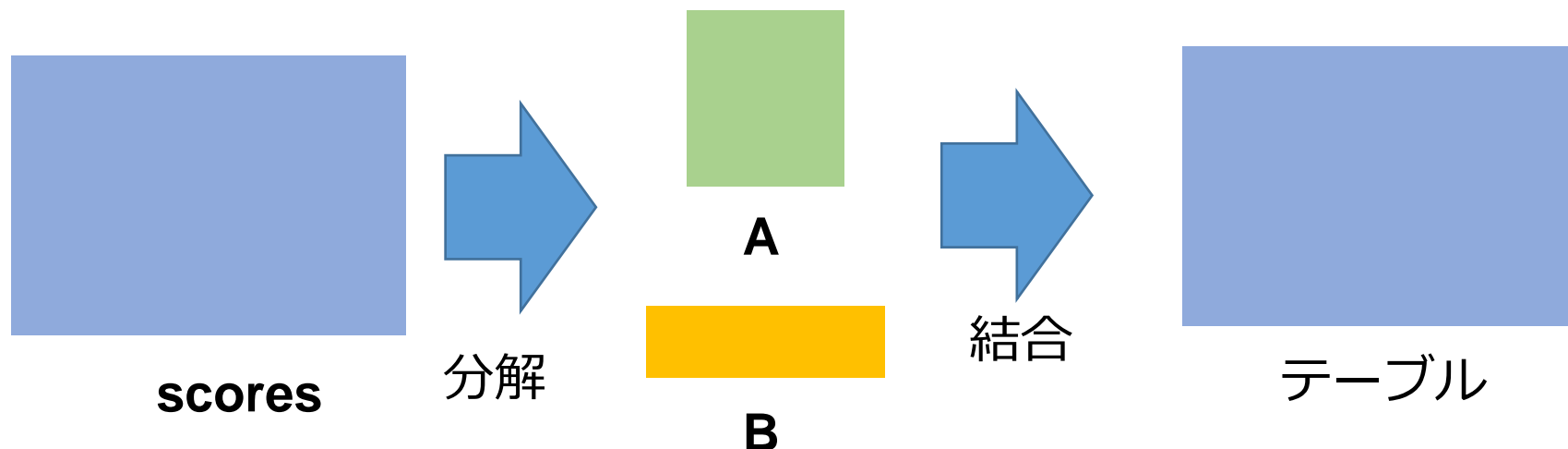
```
SELECT DISTINCT name, teacher_name  
FROM scores;
```

name	teacher_name
db	k
python	a

テーブルの分解



いまから、テーブル scores を、テーブル A, B に**分解**する



**問い合わせの結果を、
テーブルとして保存**

テーブルへの保存の方法

マイクロソフト Access では: INTO

その他のシステム(世界標準): CREATE TABLE ... AS

テーブルAの生成



【SQL プログラム】

```
CREATE TABLE A AS  
SELECT DISTINCT name, teacher_name  
FROM scores;  
SELECT * FROM A;
```

```
name      teacher_name  
db        k  
python    a
```



テーブル B の生成

テーブル A の生成の SQL を消さずに、次を書き加える

【SQL プログラム】

```
CREATE TABLE B AS  
SELECT DISTINCT id, name, student_name, score  
FROM scores;  
SELECT * FROM B;
```

id	name	student_name	score
1	db	kk	85
2	db	aa	75
3	db	nn	90
4	python	kk	85
5	python	nn	75



テーブル A, B の結合

テーブル A の生成, テーブル B の生成の SQL を消さずに、次を書き加える

【SQL プログラム】

```
SELECT B.id, A.name, A.teacher_name, B.student_name, B.score
FROM A, B
WHERE A.name = B.name;
```

id	name	teacher_name	student_name	score
1	db	k	kk	85
2	db	k	aa	75
3	db	k	nn	90
4	python	a	kk	85
5	python	a	nn	75