de-10. 分解と結合

（リレーショナルデータベース演習）

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

金子邦彦

アウトライン

番号	項目
10-1	準備
10-2	テーブル定義
10-3	テーブルへのレコード（行）の挿入
10-4	結合の例
10-5	分解の例

内容は、今までの確認、復習にもなっている。
（次回以降につながる大事な基礎の確認、復習）

各自、資料を読み返したり、課題に取り組んだりも行う

10-1. 準備

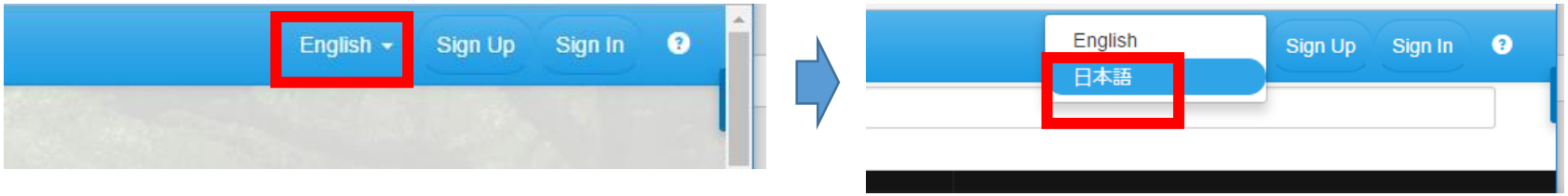
実習

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



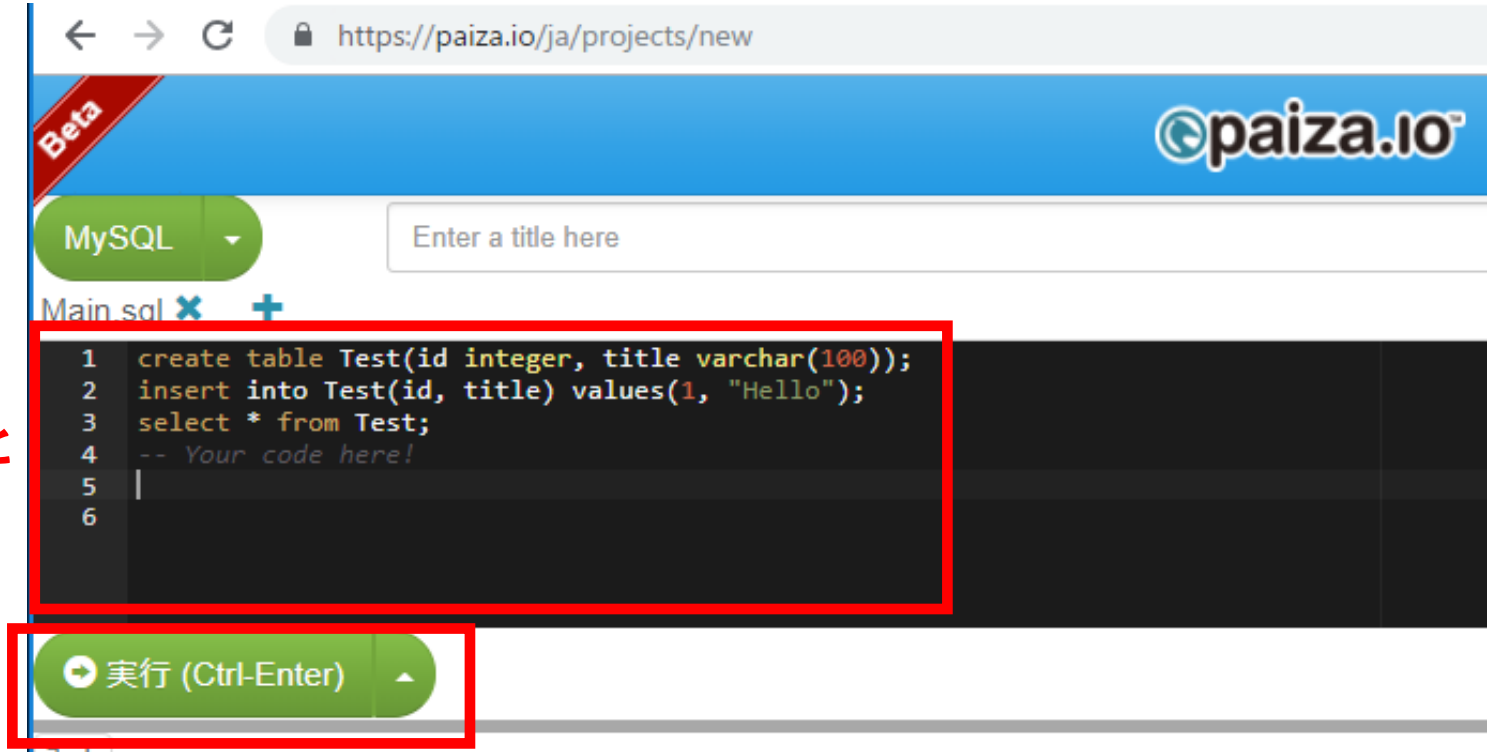
④「コード作成を試してみる」をクリック

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



10-2. テーブル定義

いまから扱うテーブルのイメージ

kamoku (科目) テーブル

ID	name (名前)	teacher (教員)
1	database	K
2	programming	A

主キー

seiseki (成績) テーブル

student (学生)	jukou (受講)	score (スコア)
KK	1	85
AA	1	75
LL	1	90
KK	2	80
LL	2	85

科目テーブルの ID を参照

① テーブル定義

```
1  create table kamoku (
2    ID integer primary key,
3    name text,
4    teacher text
5  );
```

データ型
テキスト（文字列） **text**
数値 **integer**　など
主キー **primary key**

実行

エラーメッセージが出ないことを確認．表示はない．

※　エラーメッセージが出たときは、SQLを修正してから、**再度**実行する（以下同様）

② テーブル定義　次を書き加える

```
);
create table seiseki (
   student text,
   jukou integer,
   score integer,
   foreign key(jukou) references kamoku(ID)
);
```

データ型
テキスト（文字列） **text**
数値 **integer**　など
参照整合性制約
foreign key, references

実行
エラーメッセージが出ないことを確認．表示はない．

10-3. テーブルへのレコード（行）の挿入

③ レコード（行）の挿入　次を書き加える

```
12  insert into kamoku values(1, 'database', 'K');
13  insert into kamoku values(2, 'programming', 'A');
14  insert into seiseki values('KK', 1, 85);
15  insert into seiseki values('AA', 1, 75);
16  insert into seiseki values('LL', 1, 90);
17  insert into seiseki values('KK', 2, 80);
18  insert into seiseki values('LL', 2, 85);
```

レコード（行）の挿入
insert into, values

実行

エラーメッセージが出ないことを確認．表示はない．

④ 確認表示　次を書き加える

```
select * from kamoku;
select * from seiseki;
```

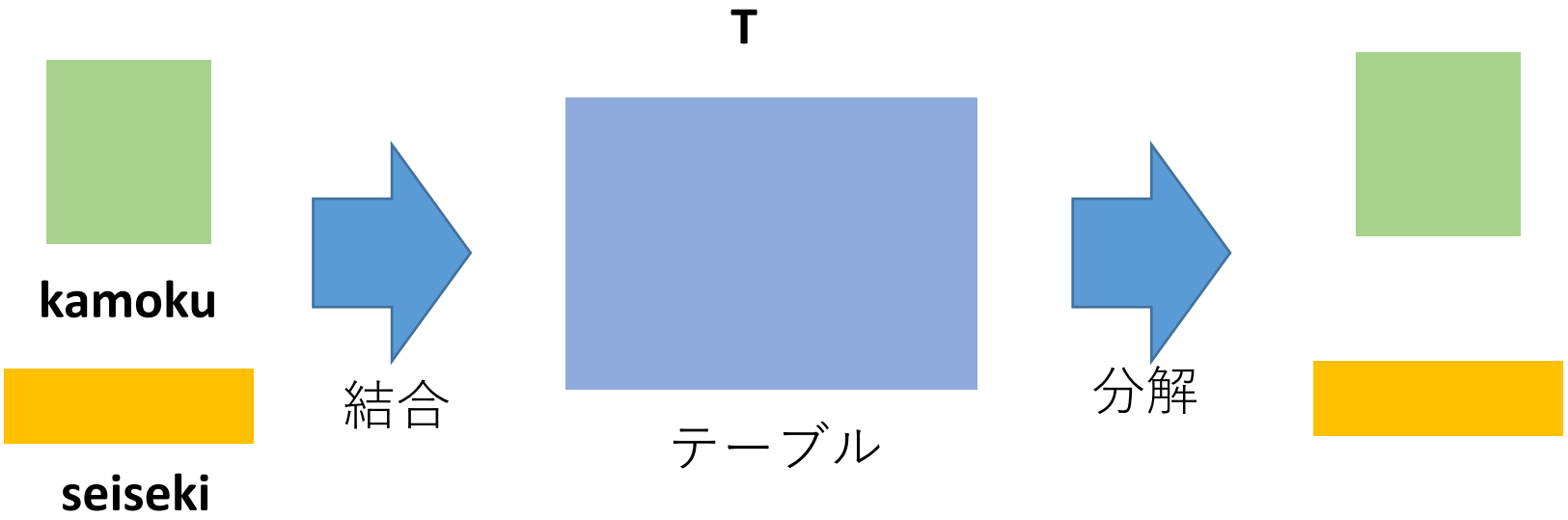
実行

エラーメッセージが出ないことを確認．表示を確認．
（あとで「表示が邪魔」と思ったときは、このプログラムは削除しても問題ない）

```
ID      name    teacher
1       database        K
2       programming     A
student jukou   score
KK      1       85
AA      1       75
LL      1       90
KK      2       80
LL      2       85
```

10-4. 結合の例

テーブルの結合と分解



問い合わせの結果を、
テーブルとして保存

うまく結合、
分解すると
元に戻る

テーブルへの保存の方法

create table … as

Access以外のシステムで動く(世界標準の方法)

⑤ 単純な結合（結合条件なし）　次を書き加える

```
select * from kamoku, seiseki;
```

実行

エラーメッセージが出ないことを確認．表示を確認．
（あとで「表示が邪魔」と思ったときは、このプログラムは削除しても問題ない）

ID	name	teacher	student	jukou	score
2	programming	A	KK	1	85
1	database	K	KK	1	85
2	programming	A	AA	1	75
1	database	K	AA	1	75
2	programming	A	LL	1	90
1	database	K	LL	1	90
2	programming	A	KK	2	80
1	database	K	KK	2	80
2	programming	A	LL	2	85
1	database	K	LL	2	85

⑥ 結合（結合条件あり）　次を書き加える

```
select * from kamoku ,seiseki
where kamoku.ID = seiseki.jukou;
```

実行

エラーメッセージが出ないことを確認．表示を確認．
（あとで「表示が邪魔」と思ったときは、このプログラムは削除しても問題ない）

ID	name	teacher	student	jukou	score
1	database	K	KK	1	85
1	database	K	AA	1	75
1	database	K	LL	1	90
2	programming	A	KK	2	80
2	programming	A	LL	2	85

⑦ 結合の結果をテーブルとして保存　次を書き加える

```
create table T as
select * from kamoku ,seiseki
where kamoku.ID = seiseki.jukou;
```

問い合わせの結果を、
テーブルとして保存
create table, as

実行
エラーメッセージが出ないことを確認．表示はない

⑧ 確認表示　次を書き加える

```
create table T as
select * from kamoku ,seiseki
where kamoku.ID = seiseki.jukou;
select * from T;
```

実行

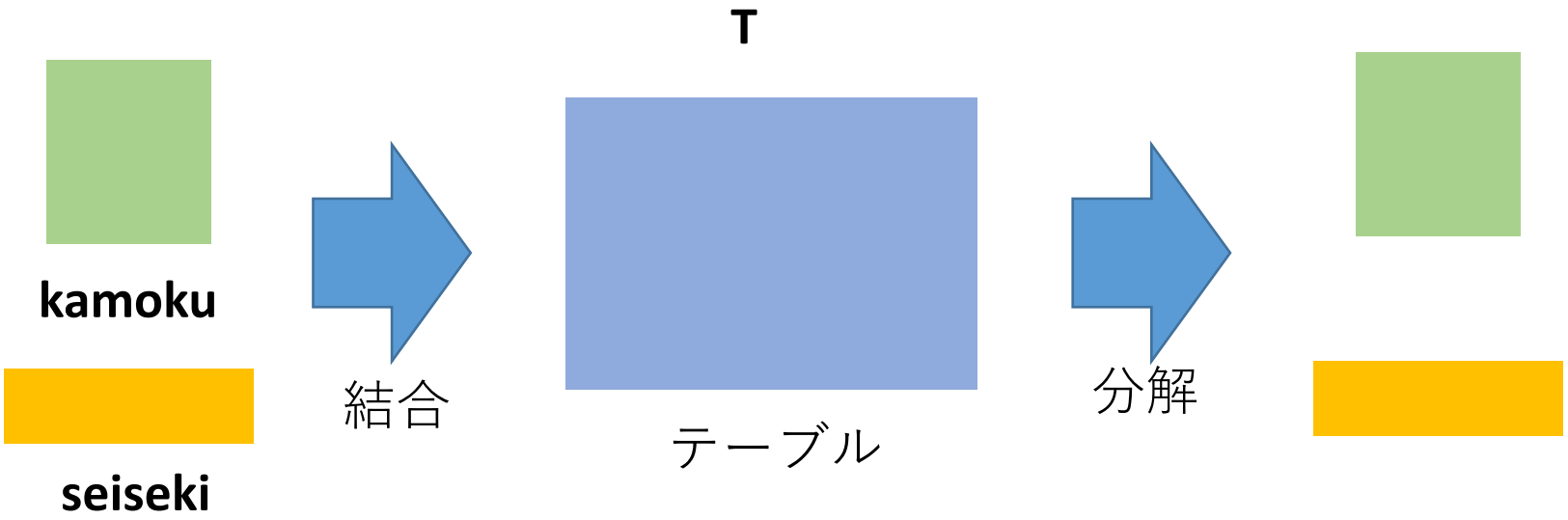
エラーメッセージが出ないことを確認．表示を確認．

このプログラムは後で使うので消さないこと

ID	name	teacher	student	jukou	score
1	database	K	KK	1	85
1	database	K	AA	1	75
1	database	K	LL	1	90
2	programming	A	KK	2	80
2	programming	A	LL	2	85

10-5. 分解の例

テーブルの結合と分解



問い合わせの結果を、
テーブルとして保存

うまく結合、
分解すると
元に戻る

テーブルへの保存の方法

create table … as

Access以外のシステムで動く(世界標準の方法)

⑨ 分解　次を書き加える

```
select distinct ID, name from T;
select distinct student, jukou, score from T;
```

実行

エラーメッセージが出ないことを確認．表示を確認．
（あとで「表示が邪魔」と思ったときは、このプログラムは削除しても問題ない）

```
ID      name
1       database
2       programming
student jukou   score
KK      1       85
AA      1       75
LL      1       90
KK      2       80
LL      2       85
```

⑩ 分解では重複行除去が必要　次を書き加える（わざと distinct を外している）

```
select ID, name from T;
select student, jukou, score from T;
```

実行

エラーメッセージが出ないことを確認．表示を確認．

```
ID      name
1       database
1       database
1       database
2       programming
2       programming
student jukou   score
KK      1       85
AA      1       75
LL      1       90
KK      2       80
LL      2       85
```