

dd-6. さまざまな SQL 問い合わせ

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6回のアウトライン



次のことについて, Paiza.IO を用いた演習

- 問い合わせ (クエリ)
- 結合, 結合条件
- 並べ替え (ソート)
- 数え上げ
- 範囲指定
- 重複除去

実習



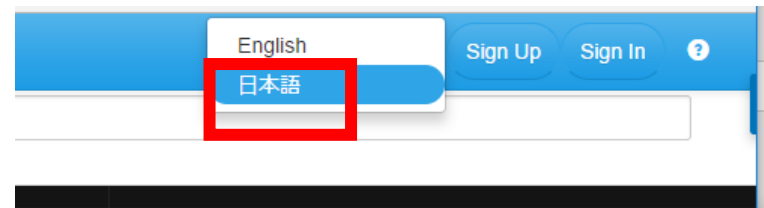
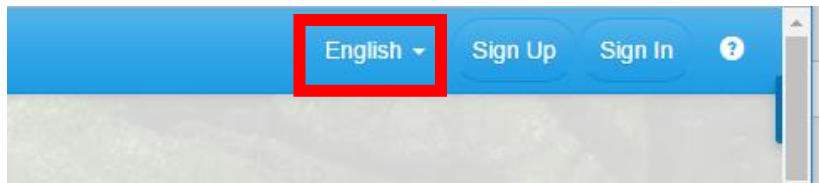
① ウェブブラウザを起動する

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



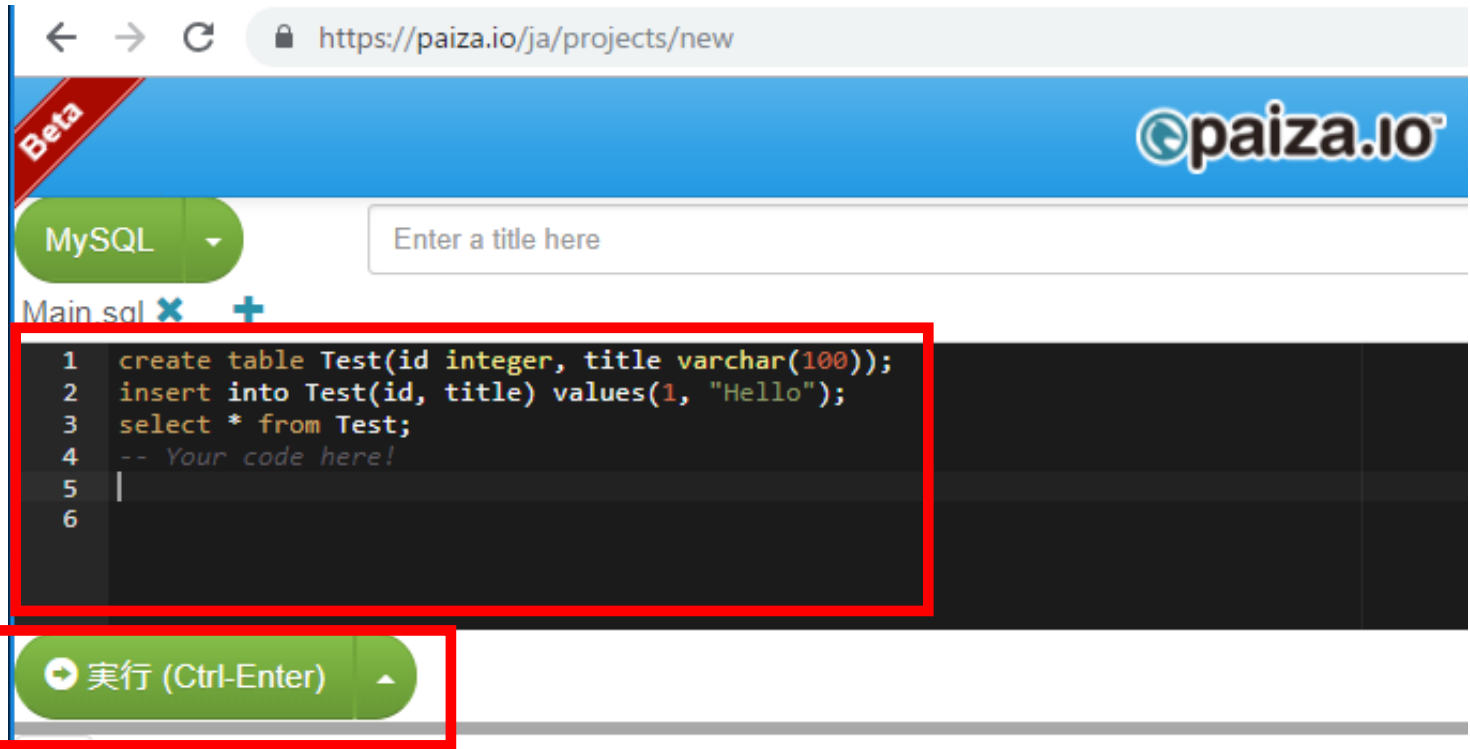
④ 「コード作成を試してみる」をクリック



⑤ 「MySQL」を選ぶ (左上のボタンをクリックするとproductsが出る)



プログラムの
編集画面



paiza.io

Beta

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)

プログラムを
書き換えること
ができる

実行ボタン

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。

```
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
```

作成するテーブル



テーブル名: products

id	name	price
1	orange	50
2	apple	100
3	melon	500

テーブル名: sales

id	customer	pid	num
1	X	1	2
2	Y	1	3
3	X	3	1
4	Y	2	4

テーブル定義 products

1から4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 products (  
  id INTEGER PRIMARY KEY NOT NULL,  
  name TEXT NOT NULL,  
  price REAL);
```



The screenshot shows a MySQL database management interface. At the top, there is a dropdown menu for 'MySQL' and a search bar with the placeholder text 'Enter a title here'. Below this, a tab labeled 'Main.sql' is active. A green banner indicates a 'Success' message with social media sharing options for 'ツイート' (Twitter) and 'Share 0'. The main area displays the following SQL code:

```
1 CREATE TABLE products (  
2   id INTEGER PRIMARY KEY NOT NULL,  
3   name TEXT NOT NULL,  
4   price REAL);
```

At the bottom, there is a green button labeled '実行 (Ctrl-Enter)' (Execute) and a link to 'MySQLのおすすめ本' (Recommended MySQL books). Below the button, there are icons for a user profile, information, and settings. At the very bottom, there is a section for '出力' (Output), '入力' (Input), and 'コメント' (Comments) with a counter showing '0'.

テーブル定義 sales

5から9行目に、次のSQLを書き加えて、「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 sales (  
  id INTEGER PRIMARY KEY NOT NULL,  
  customer TEXT NOT NULL,  
  pid INTEGER NOT NULL,  
  num INTEGER NOT NULL);
```

```
5 CREATE TABLE sales (  
6   id INTEGER PRIMARY KEY NOT NULL,  
7   customer TEXT NOT NULL,  
8   pid INTEGER NOT NULL,  
9   num INTEGER NOT NULL);
```

レコードの挿入と確認



10から 13行目に、次の **SQL** を書き加えて、「実行」をクリック。結果を確認

【SQLプログラム】

```
INSERT INTO products VALUES( 1, 'orange', 50 );  
INSERT INTO products VALUES( 2, 'apple', 100 );  
INSERT INTO products VALUES( 3, 'melon', 500 );  
SELECT * FROM products;
```

id	name	price
1	orange	50
2	apple	100
3	melon	500

テーブルへのレコードの挿入と確認

14から18行目に、次のSQLを書き加えて、「実行」をクリック。結果を確認

【SQLプログラム】

```
INSERT INTO sales VALUES(1, 'X', 1, 2);  
INSERT INTO sales VALUES(2, 'Y', 1, 3);  
INSERT INTO sales VALUES(3, 'X', 3, 1);  
INSERT INTO sales VALUES(4, 'Y', 2, 4);  
SELECT * FROM sales;
```

id	customer	pid	num
1	X	1	2
2	Y	1	3
3	X	3	1
4	Y	2	4

```
1 CREATE TABLE products (  
2   id INTEGER PRIMARY KEY NOT NULL,  
3   name TEXT NOT NULL,  
4   price REAL);  
5 CREATE TABLE sales (  
6   id INTEGER PRIMARY KEY NOT NULL,  
7   customer TEXT NOT NULL,  
8   pid INTEGER NOT NULL,  
9   num INTEGER NOT NULL);  
10 INSERT INTO products VALUES( 1, 'orange', 50 );  
11 INSERT INTO products VALUES( 2, 'apple', 100 );  
12 INSERT INTO products VALUES( 3, 'melon', 500 );  
13 SELECT * FROM products;  
14 INSERT INTO sales VALUES(1, 'X', 1, 2);  
15 INSERT INTO sales VALUES(2, 'Y', 1, 3);  
16 INSERT INTO sales VALUES(3, 'X', 3, 1);  
17 INSERT INTO sales VALUES(4, 'Y', 2, 4);  
18 SELECT * FROM sales;
```

いまからは、**1行目から18行目に入れたプログラムをそのまま残して実行する**

【SQL プログラム】

```
SELECT * FROM products;
```

id	name	price
1	orange	50
2	apple	100
3	melon	500

【SQL プログラム】

```
SELECT * FROM products WHERE name = 'orange';
```

id	name	price
1	orange	50

【SQL プログラム】

```
SELECT * FROM products WHERE price > 70;
```

id	name	price
2	apple	100
3	melon	500

【SQL プログラム】

```
SELECT * FROM products, sales;
```

id	name	price	id	customer	pid	num
3	melon	500	1	X	1	2
2	apple	100	1	X	1	2
1	orange	50	1	X	1	2
3	melon	500	2	Y	1	3
2	apple	100	2	Y	1	3
1	orange	50	2	Y	1	3
3	melon	500	3	X	3	1
2	apple	100	3	X	3	1
1	orange	50	3	X	3	1
3	melon	500	4	Y	2	4
2	apple	100	4	Y	2	4
1	orange	50	4	Y	2	4

行（レコード）の順序が違っている場合がある

【SQL プログラム】

```
SELECT * FROM products, sales  
WHERE products.id = sales.pid;
```

id	name	price	id	customer	pid	num
1	orange	50	1	X	1	2
1	orange	50	2	Y	1	3
3	melon	500	3	X	3	1
2	apple	100	4	Y	2	4

並べ替え (ソート)



【SQLプログラム】

```
SELECT * FROM products ORDER BY price;
```

```
..  
id      name      price  
1       orange   50  
2       apple    100  
3       melon    500
```

並べ替え (ソート)



【SQL プログラム】

```
SELECT * FROM products ORDER BY price DESC;
```

id	name	price
3	melon	500
2	apple	100
1	orange	50

【SQL プログラム】

```
SELECT customer, COUNT(*) FROM sales  
GROUP BY customer;
```

customer	COUNT(*)
X	2
Y	2

【SQL プログラム】

```
SELECT * FROM products  
WHERE price BETWEEN 50 AND 200;
```

id	name	price
1	orange	50
2	apple	100

【SQL プログラム】

```
SELECT DISTINCT customer FROM sales;
```

customer

X

Y

問い合わせの結果をテーブルに保存



【SQL プログラム】

```
CREATE TABLE T AS  
SELECT DISTINCT name FROM products;  
SELECT * FROM T;
```

```
name  
orange  
apple  
melon
```