

5.5 SQL Subquery

- 副問合せ
- 複数行副問合せ
- 相関複問合せ

目次

- 1 副問合せ
- 2 複数行副問合せ
- 3 相関副問合せ

テーブルの作成

- 副問合せを学ぶ前にまずは、使用するテーブルとデータの作成をします。今回は4つのテーブルを作成してもらいます。

Try 
subquery.sql

副問合せ


- 副問合せ（サブクエリ）とは、
「SELECT文の中に別のSELECT文を入れて書くことで、複数のSELECT文を一度で実行できるようになる文」のことです。
- SELECT文の中にSELECT文といわれてもイメージが付きにくいと思うので、例を見ながら考えていきましょう。
- 例えば先ほど作成したテーブルを使用し、「priceの平均以上のレコードをとってくる」には、どうしたらよいかを考えていきます。

次へ 

- まずは、「priceの平均以上を取得する必要」があるので、今までの知識でそのSQL文を再現するなら、以下のようなSQL文になります。

```
SELECT AVG(price) FROM productorder;
```



	avg numeric 
1	700.00000000

- 次にpriceの平均値がわかったのでそれを使用して「priceの平均以上のレコードを取得する」ことを今まで習った知識で表現すると以下のようなSQL文になります。

```
SELECT order_id,price FROM productorder WHERE price >= 700;
```



	order_id [PK] integer	price integer
1	6	700
2	4	700
3	3	1400
4	1	840

- 「priceの平均値以上のレコードをとってくる」という作業のために、今まで習った知識を使用すると2つのSELECT文を書く必要があります。しかし、副問合せを使うことで、これらをまとめて1つのSELECT文で書くことができます。

```
SELECT AVG(price) FROM productorder;
```

700

+

```
SELECT  
  order_id, price  
FROM  
  productorder  
WHERE  
  price >= 700;
```

?

- 以下が、副問合せを使用した「priceの平均値以上のレコードをproductorderテーブルからとってくる」のSQL文です。

```
SELECT
  order_id, price
FROM
  productorder
WHERE
  price >= (
    SELECT
      AVG(price)
    FROM
      productorder
  );
```

主問合せと副問合せ

SELECT **カラム名** FROM テーブル名
WHERE **カラム名** 演算子 (SELECT ~);

↑ 主問合せ

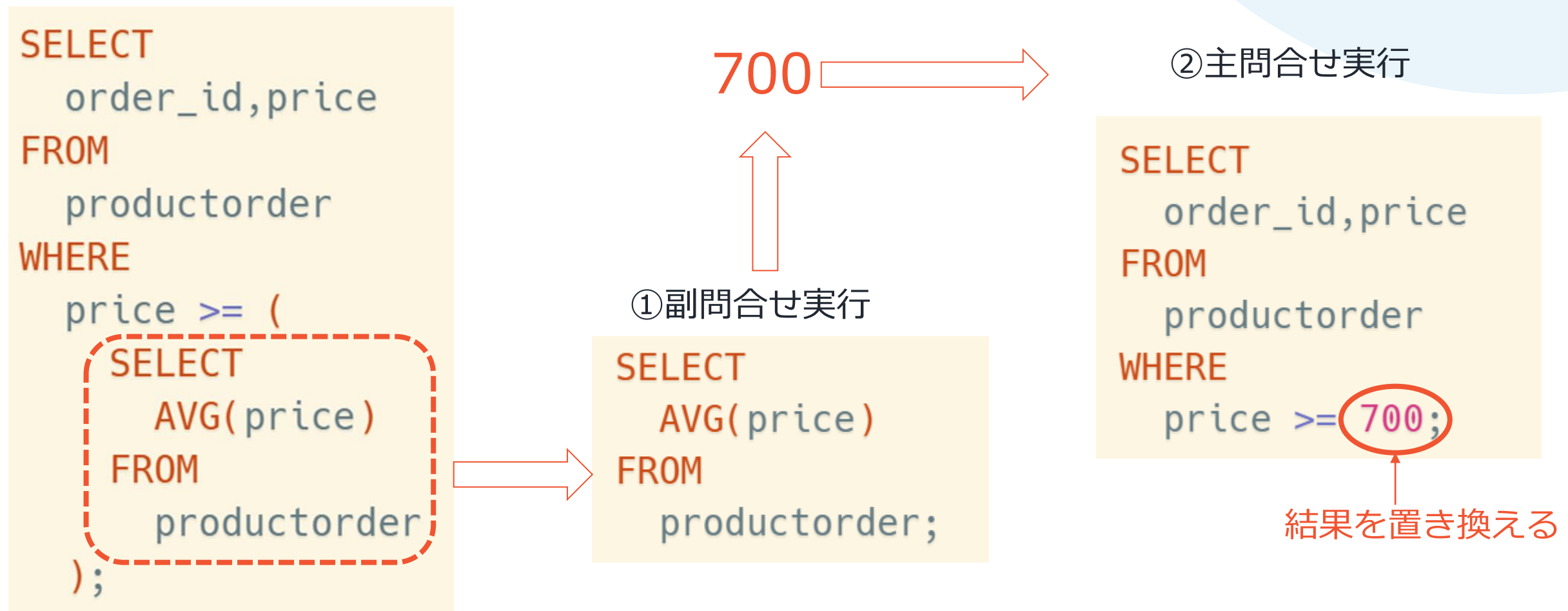
↑ 副問合せ

Note !

集約関数は、WHERE句の中につけないため
WHERE price >= AVG(price) はダメ！

副問合せの実行順番

- 副問合せ部分は、主問合わせ部分より先に実行されます。
- 先ほどの例だと、副問合せ部分「SELECT AVG(price) FROM productorder」が真っ先に実行され、結果の700と置き換わり、その後に主問合せが実行されます。



練習問題

- productorderテーブルをcustomer_idでグループ化する。
グループごとのpriceの平均値が、全レコードのprice平均値よりも小さいグループのcustomer_idとpriceの平均値をとってきてください。

実行結果は下記となります。

	customer_id integer	avg numeric
1	5	650.00000000
2	1	330.00000000

自分が書いたSQL文と実行結果をスクリーンショットを撮り
Slackの「週テスト」に張りつけてください！

Q&A

目次

- 1 副問合せ
- 2 複数行副問合せ
- 3 相関副問合せ

副問合せの結果

- 今までの例や問題に出てきた副問合せは、すべて1つの値を返すものでした。
- 1つの値を返す副問合せを「**単一行副問合せ**」といいます。
- 問合せは、SELECT文を使うので、当然結果が複数行になることもあります。
結果が複数行のレコードを返す副問合せを「**複数行副問合せ**」
といいます。

1つの値のみ

--

単一行副問合せ

1カラムで複数レコード 複数カラム複数レコード=テーブル

複数行副問合せ

次へ



- 例えばproductテーブルでpriceが150以上のproduct_idと一致するproductorderテーブルのレコードをとってくるSQLを実行するとエラーが起こります。

```
SELECT
  order_id,product_id
FROM
  productorder
WHERE
  product_id =
```

```
SELECT
  product_id
FROM
  product
WHERE
  price >= 150
```

```
SELECT
  product_id
FROM
  product
WHERE
  price >= 150;
```

副問合せの結果

product_id
2
3

どれと比較?

ERROR: 式として使用された副問い合わせが
2行以上の行を返しました
SQL 状態: 21000

複数行副問合せ

- 副問合せの結果が複数レコードになる場合があることを先ほどの例で学びました。

副問合せが複数レコード返す場合、主問合せの中で副問合せの結果をどう扱えばよいかをここでは、学習します。

- まずは、副問合せの結果が複数レコードで、カラムが1つだけの場合を一緒に考えていきますが、その前に複数レコードを取扱い演算子を見ておきましょう。

次へ 

- 副問合せの1カラム複数レコードの結果に対して、以下の演算子を使うことができます。

演算子	使い方	意味
IN	a IN (副問合せ)	aが副問合せの結果のいずれかに一致すれば1を返す
NOT IN	a NOT IN (副問合せ)	aが副問合せの結果のいずれにも一致しなければ1を返す
ANY	a 演算子 ANY(副問合せ)	副問合せの結果のいずれかと、aの演算結果が1ならば1を返す
ALL	a 演算子 ALL(副問合せ)	副問合せの結果のすべてと、aの演算結果が1ならば1を返す

- 先ほどの演算子を使った例を1つ見てみましょう。
- 下記のSQLはproductorderテーブルの中にあるpriceの値が700以上のcustomer_idを副問合せでとってきて、重複レコードがあることを考えてDISTINCTを使用。
副問合せの結果とIN演算子を使って、customerテーブルから顧客の情報をとってきます。

```
SELECT
  customer_id, customer_name
FROM
  customer
WHERE
  customer_id IN
  (
    SELECT
      DISTINCT customer_id
    FROM
      productorder
    WHERE
      price >= 700
  );
```



	customer_id [PK] integer	customer_name character varying (30)
1	5	大川裕子
2	4	原和成
3	3	竹村ひとみ
4	2	石川幸江

```
SELECT
  DISTINCT customer_id
FROM
  productorder
WHERE
  price >= 700
```

子查询的结果



customer_id
5
4
3
2



```
SELECT
  customer_id, customer_name
FROM
  customer
WHERE
  customer_id IN
```

- ANY演算子を使った例を見てみましょう
- productorderテーブルで、product_idごとのquantityの合計値を副問合せでとります。Productテーブルから、副問合せの結果のいずれの値よりstockが小さいレコードをとってきます。

```
SELECT
*
FROM
product
WHERE stock < ANY
```

```
(
SELECT
SUM(quantity)
FROM
productorder
GROUP BY
product_id
);
```

SUM(quantity)
12
5
2
3
4

副問合せの結果

product_idごとのquantityの合計値

	product_id [PK] integer	product_name character varying (100)	stock integer	price integer
1	1	薬用入浴剤	100	70
2	2	薬用ハンドソープ	23	700
3	3	温泉のもと草津	4	120
4	4	石鹸	23	120
5	5	イチゴ牛乳	10	150
6	6	バナナ牛乳	15	140

この副問合せの結果のいずれの値よりもstockが小さいproductテーブルのレコードは以下の2つだけです。

	product_id [PK] integer	product_name character varying (100)	stock integer	price integer
1	5	イチゴ牛乳	10	150
2	3	温泉のもと草津	4	120


主問い合わせの結果

副問合せの結果がテーブルになるとき

- 今度は、複数行副問合せの中でも、結果が複数のカラムを持つ場合の使い方を紹介します。
- productorderテーブルをcustomer_idでグループ化し、グループごとの購入回数をカウントし、さらにその平均値を求めます。その際、グループ化の部分を副問合せで行います。

```
SELECT
  AVG(c)
FROM
  (
    SELECT
      customer_id,
      COUNT(*) AS c
    FROM
      productorder
    GROUP BY
      customer_id
  ) AS a;
```



	avg numeric 
1	1.4000000000

次へ 




- 副問合せで、productorderテーブル customer_idでグループし、グループごとの購入回数をカウントします。
- 副問合せのSELECT句にcustomer_idの カラム を入れる必要はないですが、確認のために入れておきます。

```
SELECT
  customer_id,
  COUNT(*) AS c
FROM
  productorder
GROUP BY
  customer_id;
```

- 副問合せの結果のテーブルになります。

customer_id	c
1	2
2	1
3	1
4	1
5	2

- 問合せの結果テーブルから、主問合せでグループごとの購入回数の平均値を計算します。

	avg numeric 
1	1.4000000000

Q&A

目次

- 1 副問合せ
- 2 複数行副問合せ
- 3 相関副問合せ

相関副問合せ

- 今までは、副問合せ実行後、主問合せをしてきましたが、副問合せの中でも、主問合せと連携して実行する場合があります。これを「相関副問合せ」いいます。

今までの副問合せ

②次に副問合せ

SELECT ~

(SELECT ~)

①先に副問合せ

相関副問合せ

①主問合せを実行しつつ

SELECT ~

 (SELECT ~)

②副問合せ実行

- 相関副問合せでは、先に「主問合せ」を実行して、「主問合せの1レコードごとに副問合せを実行」します。普通の副問合せとは、全く違う動作になります。

次へ

- 相関副問合せの例を見ていきましょう。
- productテーブルにある商品の中で、productorderテーブルの情報から、商品ごとの合計売上個数が3より大きい商品の情報をとってきます。

```
SELECT
  product.product_id,
  product.product_name
FROM
  product
WHERE
  3 < (
    SELECT
      SUM(quantity)
    FROM
      productorder
    WHERE
      product.product_id = productorder.product_id
  );
```

相関副問合せ

	product_id [PK] integer	product_name character varying (100)
1	3	温泉のもと草津
2	2	薬用ハンドソープ
3	1	薬用入浴剤

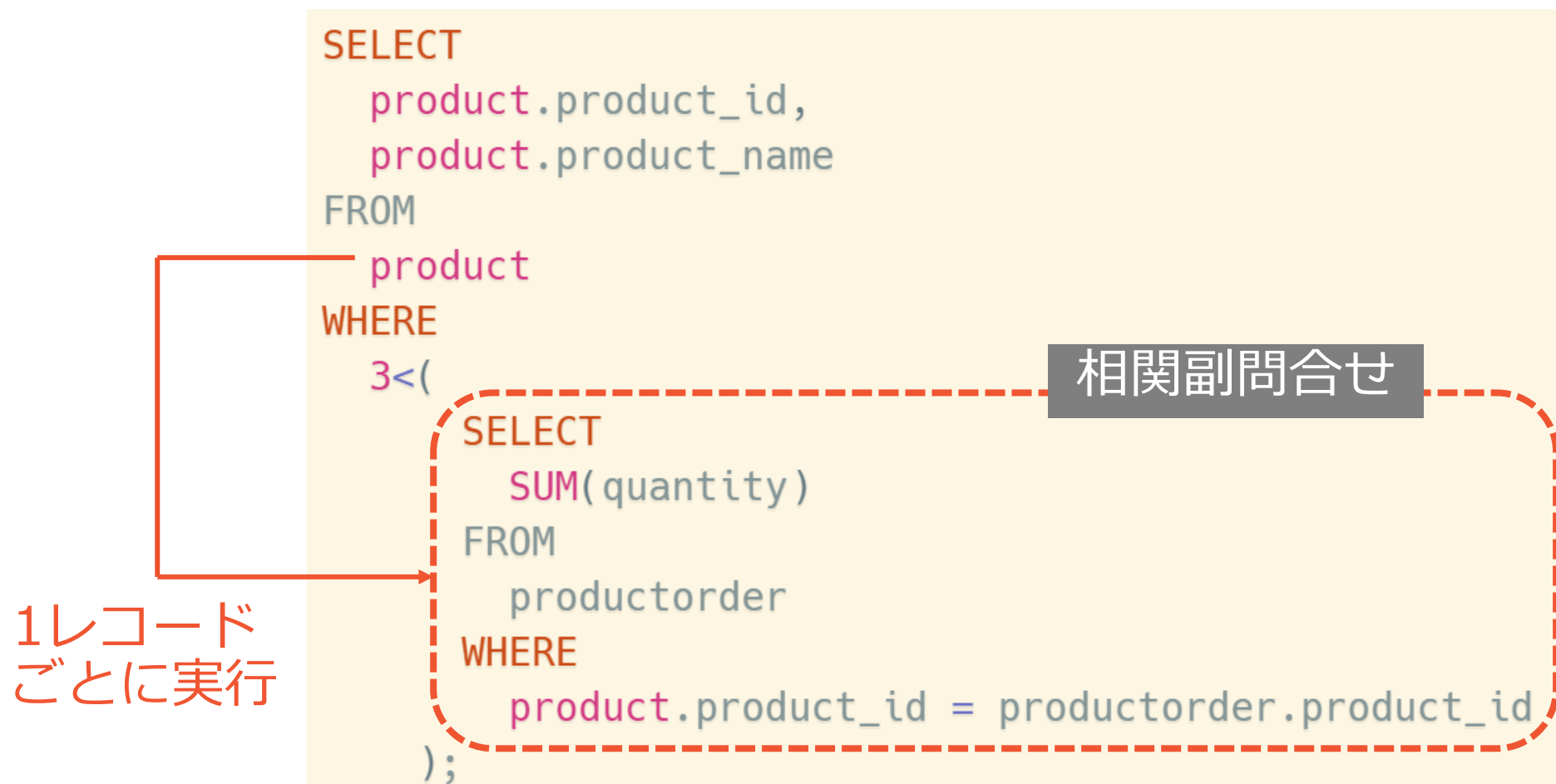
- 主問合せの対象は、productテーブルです。
productテーブルの1レコードごとに副問合せを実行します。
- 副問合せは、そのとき実行している「productテーブルのレコードのproduct_id」と一致する「productorderテーブルのproduct_id」を検索条件としています。副問合せの中で、主問い合わせの値を参照しています。
- 主問合せと副問合せで、テーブルが2種類出てきます。カラムがどちらのテーブルのものかを示すために、テーブル名とカラム名を「.」でつなげて書きます。

ここでは、相関副問合せの仕組みについてみていきます。

相関副問合せは、主問合せの中で1レコードずつ副問合せを実行します。

主問合せは、まず最初にFROM句、次にWHERE句が実行されます。

相関副問合せはWHERE句の中にあります。相関副問合せは、WHERE句で主問い合わせの1レコードずつ実行されます。





- Product テーブルの product_id が1のとき、次の副問合せを実行します。

```
(  
    SELECT  
        SUM(quantity)  
    FROM  
        productorder  
    WHERE  
        1 = productorder.product_id  
)
```

- この結果は12なので、主問い合わせのWHERE句の条件「 $3 < 12$ 」はtrueになり、product_idが1のレコードをとってきます。



```
SELECT
  product.product_id,
  product.product_name
FROM
  product
WHERE
  3 < (
```

	product_id [PK] integer	product_name character varying (100)
1	1	薬用入浴剤
2	2	薬用ハンドソープ
3	3	温泉のもと草津
4	4	石鹸
5	5	イチゴ牛乳
6	6	バナナ牛乳

```
(
  SELECT
    SUM(quantity)
  FROM
    productorder
  WHERE
    product.product_id = productorder.product_id
);
```

productorder
テーブル

実行

	order_id [PK] integer	customer_id integer	product_id integer	quantity integer
1	1	4	1	12
2	2	5	3	5
3	3	2	2	2
4	4	3	2	1
5	5	1	4	3
6	6	5	2	1
7	7	1	5	2

まとめ

Sum Up



1. 副問合せの概念。
2. 複数行副問合せの考え方と書き方
3. 相関副問合せの考え方と書き方



Light in Your Career.
THANK YOU!