

2021/11/4 木 2時限

内容（講義と演習）

内容

1. MySQL 複合クエリー group by + having
2. wineShopAplus へのデータクエリー (20 点)

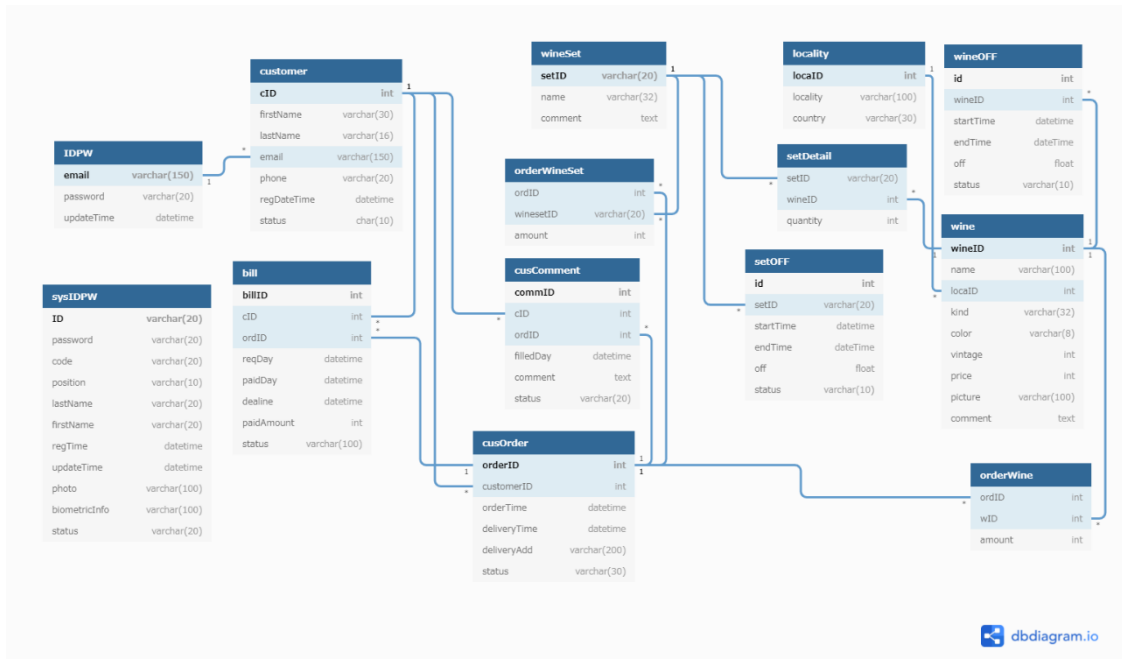


図 1. wineShopAplus の ERD

➤ 複合クエリーの利用における注意点

複合クエリーを利用して、次のようなより複雑な検索ができます。ただし、グループ化を入れた複合クエリーを使う際、集計やグループ計算などできるが、特別な注意を払わなければなりません。

例：ビンテージ別にワインの名前、値段、ビンテージを示せ。

```
mysql> select name,price,vintage from wine order by vintage;
```

name	price	vintage
サンテミリオン	5800	1997
オーメドック	2200	1997
ジュヴレシャンベルタン	3000	1998
シャンパン	4000	1999

トカイ・ソモロドニ・ドライ EV	1728	2000
シャブリ	2400	2001
サンセール	2800	2001
トカイ・アッス・3 ブットニョッシュ EV	2484	2007
コレッツィオーネ チンクアンタ	2580	2011
シュロス・フォルラーツ	2400	2012
マス・デ・タンヌ	1690	2014

11 rows in set (0.00 sec)

例 1. 製造年別で一番高いワインのワイン名, 色, ビンテージと値段を示せ.

以下のクエリーは正しい結果は得られません. 原因は, max(price) は name と必ずしも結びつかないのです.

mysql> select name,max(price),vintage from wine group by vintage; ✗

name	max(price)	vintage	
サンテミリオン	5800	1997	○
ジュヴレシャンベルタン	3000	1998	○
シャンパン	4000	1999	○
トカイ・ソモロドニ・ドライ EV	1728	2000	○
シャブリ	2800	2001	✗
トカイ・アッス・3 ブットニョッシュ EV	2484	2007	○
コレッツィオーネ チンクアンタ	2580	2011	○
シュロス・フォルラーツ	2400	2012	○
マス・デ・タンヌ	1690	2014	○

9 rows in set (0.00 sec)

一般的に, 以下のように複合クエリーを使い, 一番高いワインの名前, 値段, ビンテージを示します.

mysql> select name,price,vintage from wine where price = (select max(price) from wine);

name	price	vintage
------	-------	---------

```
| サンテミリオン | 5800 | 1997 |
```

```
+-----+-----+
```

```
1 row in set (0.00 sec)
```

しかし、これは一つデータしか得られないので、不便だと感じられます。まず、以下のよう
に group by を用いて製造年別にそれぞれの一番高い値段（最大値）を求めます。

```
mysql> select max(price),vintage from wine group by vintage;
```

```
+-----+-----+
```

```
| max(price) | vintage |
```

```
+-----+-----+
```

```
| 5800 | 1997 |
```

```
| 3000 | 1998 |
```

```
| 4000 | 1999 |
```

```
| 1728 | 2000 |
```

```
| 2800 | 2001 |
```

```
| 2484 | 2007 |
```

```
| 2580 | 2011 |
```

```
| 2400 | 2012 |
```

```
| 1690 | 2014 |
```

```
+-----+-----+
```

```
9 rows in set (0.00 sec)
```

そして、「製造年別で一番高いワインのワイン名、色、ビンテージと値段を示せ」との課題
に対して、以下のように複合クエリ組んでみます。結果をみると、2001年は2回現れ、
最大値に対応したワイン名などを示せといった要求に満たしていません。間違えた原因は、
製造年別の値段最大値の集合は (5800, 3000, 4000, 1728, 2800, 2484, 2580, 2400, 1690)
になっており、下記のクエリはすべてのワインの値段を、先に求めてきた最大値集合の
全要素をみながら、**どちらかにマッチすると抽出してしまうわけです**。つまり、2400は2001
年の最大値ではなく2012年の最大値になっているが、最大値集合には入っているので、最
大値として出してしまいました。このクエリ-の表示は間違っていないものの、課題に
合っていない結果になりました。

```
mysql> select name,price,vintage from wine where price in (select max(price) from  
wine group by vintage); ✖
```

```
+-----+-----+
```

name	price	vintage	
シャブリ	2400	2001	×
ジュヴレシャンベルタン	3000	1998	
サンテミリオン	5800	1997	
サンセール	2800	2001	○
シャンパン	4000	1999	
コレッツィオーネ チンクアンタ	2580	2011	
マス・デ・タンヌ	1690	2014	
トカイ・ソモロドニ・ドライ EV	1728	2000	
トカイ・アッス・3 ブットニョッシュ EV	2484	2007	
シュロス・フォルラーツ	2400	2012	

10 rows in set (0.01 sec)

データ次第で同様の問題は下記の例2と例3にも起こりうると思います。

例2. 国別において一番高いワインの名前, 色, 値段を示せ.

```
mysql> select name, price, color, country from wine, locality where
wine.locaID=locality.locaID and price in (select max(price) from wine, locality
where wine.locaID=locality.locaID group by country); ×
```

name	price	color	country
サンテミリオン	5800	赤	フランス
サンセール	2800	白	ドイツ
コレッツィオーネ チンクアンタ	2580	赤	イタリア
トカイ・アッス・3 ブットニョッシュ EV	2484	白	ハンガリ

4 rows in set (0.00 sec)

ただし, もし, 10番目のワインのデータ wineID=10, country='ハンガリ' を次のように変更した後再度検索しますと, 結果が変わりました. フランスは3個の検索結果が出て, 例1と似て結果は間違いました.

```
mysql> update wine set price=2400 where wineID=10;
```

```
Query OK, 1 row affected (0.01 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select wineID,name, price,color,country from wine,locality where  
wine.locaID=locality.locaID and price in (select max(price) from wine,locality  
where wine.locaID=locality.locaID group by country); ×
```

wineID	name	price	color	country	
1	シャブリ	2400	白	フランス	×
3	サンテミリオン	5800	赤	フランス	○
5	サンセール	2800	白	ドイツ	○
7	コレッツィオーネ チンクアンタ	2580	赤	イタリア	○
10	トカイ・アッス・3 ブットニョッシュ EV	2400	白	ハンガリ	○
11	シュロス・フォルラーツ	2400	白	フランス	×

```
6 rows in set (0.00 sec)
```

例3. 色別に一番高いワインの名前, 値段, 色, 国名を示せ.

```
mysql> mysql> select name, price,color,country from wine,locality where  
wine.locaID=locality.locaID and price in (select max(price) from wine group by  
color); ×
```

name	price	color	country
サンテミリオン	5800	赤	フランス
シャンパン	4000	白	フランス

例3もデータ次第で結果が変わってくると予想できます.

特別に指定されるグループにおける値段最大値の集合に対して, ワインテーブル wine には, それぞれにマッチできる複数の値段がある場合, 以上のやり方では検索条件が不十

分（不完全）であるため、検索は不具合が出てきます。たまに検索結果は正しいかもしれませんが、間違った結果がある限り信用性はなくなります。

上記の問題を解決する方法は以下の3つあります。

方法1. 複合クエリーを使う際、group by と having を同時に使う。having はグループ化するときに付き加えた条件です。例1の場合は、最大値と同じビンテージかつ、最大値と同じ値段になるのが条件とします。

```
mysql> select name,price, vintage as va from wine where price in (select max(price) from wine group by vintage having va = vintage);
```

name	price	va
ジュヴレシャンベルタン	3000	1998
サンテミリオン	5800	1997
サンセール	2800	2001
シャンパン	4000	1999
コレツィオーネ チンクアンタ	2580	2011
マス・デ・タンヌ	1690	2014
トカイ・ソモロドニ・ドライ EV	1728	2000
トカイ・アッス・3 ブットニョッシュ EV	2484	2007
シュロス・フォルラーツ	2400	2012

9 rows in set (0.00 sec)

或は、テーブルの別名を使い、自己参照を利用します。

```
mysql> select name,price, vintage from wine as a where price in (select max(price) from wine group by vintage having a.vintage = vintage);
```

name	price	vintage
ジュヴレシャンベルタン	3000	1998
サンテミリオン	5800	1997
サンセール	2800	2001
シャンパン	4000	1999

```

| コレッツィオーネ チンクアンタ | 2580 | 2011 |
| マス・デ・タンヌ | 1690 | 2014 |
| トカイ・ソモロドニ・ドライ EV | 1728 | 2000 |
| トカイ・アッス・3 ブットニョッシュ EV | 2484 | 2007 |
| シュロス・フォルラーツ | 2400 | 2012 |

```

9 rows in set (0.00 sec)

データを変更した例2は、以下のように書き換えます。ここでは、表の自然結合を使いません。

```

mysql> select name, price, color, country from wine natural join
locality as a where price in (select max(price) from wine natural join
locality group by country having a.country=country);

```

```

+-----+-----+-----+-----+
| name | price | color | country |
+-----+-----+-----+-----+
| サンテミリオン | 5800 | 赤 | フランス |
| サンセール | 2800 | 白 | ドイツ |
| コレッツィオーネ チンクアンタ | 2580 | 赤 | イタリア |
| トカイ・アッス・3 ブットニョッシュ EV | 2400 | 白 | ハンガリ |
+-----+-----+-----+-----+

```

4 rows in set (0.00 sec)

或は、

```

mysql> select name, price, color, country from wine, locality as a where
wine.localID=a.localID and price in (select max(price) from wine natural
join locality group by country having a.country=country);

```

```

+-----+-----+-----+-----+
| name | price | color | country |
+-----+-----+-----+-----+
| サンテミリオン | 5800 | 赤 | フランス | ○
| サンセール | 2800 | 白 | ドイツ |
| コレッツィオーネ チンクアンタ | 2580 | 赤 | イタリア |
| トカイ・アッス・3 ブットニョッシュ EV | 2400 | 白 | ハンガリ |
+-----+-----+-----+-----+

```

4 rows in set (0.00 sec)

方法2. 幾つかの部分集合を使う. 検索条件は部分集合の共通する部分を一致するのが条件とする. すべてワインの値段とビンテージの集合と, ビンテージごとに最大値を求めた値段とそのビンテージの集合からデータを探し, 二つの集合にて両方の価格とビンテージが一致しているワインを抽出する.

```
mysql> select name, price, vintage from wine, (select max(price) as mp, vintage as vt from wine group by vintage) as ta where vintage= vt and price = mp order by vintage;
```

name	price	vintage
サンテミリオン	5800	1997
ジュヴレシャンベルタン	3000	1998
シャンパン	4000	1999
トカイ・ソモロドニ・ドライ EV	1728	2000
サンセール	2800	2001
トカイ・アッス・3 ブットニョッシュ EV	2484	2007
コレツィオーネ チンクアンタ	2580	2011
シュロス・フォルラーツ	2400	2012
マス・デ・タンヌ	1690	2014

9 rows in set (0.00 sec)

同様に, データを変更した例2は以下のクエリーで探索します.

```
mysql> select name, price, country from wine, locality, (select max(price) as mp, country as ct from wine, locality where wine.localID=locality.localID group by country) as ta where wine.localID=locality.localID and country = ct and price = mp order by price desc;
```

name	price	country
サンテミリオン	5800	フランス

```

| サンセール | 2800 | ドイツ |
| コレッツィオーネ チンクアンタ | 2580 | イタリア |
| トカイ・アッス・3 ブットニョッシュ EV | 2400 | ハンガリ |
+-----+-----+-----+

```

4 rows in set (0.00 sec)

例3は、以下のように書き直します。結果は先述したクエリーと同じですが、探索エラーを防ぐことができました。

```

mysql> select name, price, color, country from wine, locality, (select max(price) as
mp, color as ct from wine group by color) as ta where wine.localID=locality.localID
and color = ct and price = mp order by price desc;

```

```

+-----+-----+-----+-----+
| name | price | color | country |
+-----+-----+-----+-----+
| サンテミリオン | 5800 | 赤 | フランス |
| シャンパン | 4000 | 白 | フランス |
+-----+-----+-----+-----+

```

2 rows in set (0.00 sec)

方法3. view を使う. ビンテージごとに最大値を求め、その集合を収める仮想のテーブル (view) を作っておきます。その後、ワインのテーブルと照合します。両方の値段とビンテージが一致するデータを抽出します。

```

mysql> create view maxP as select max(price) as mp, vintage as vt from wine group
by vintage;

```

```

mysql> select name, price, vintage from wine, maxP where price=mp and vintage=vt;

```

```

+-----+-----+-----+-----+
| name | price | vintage |
+-----+-----+-----+-----+
| ジュヴレシャンベルタン | 3000 | 1998 |
| サンテミリオン | 5800 | 1997 |
| サンセール | 2800 | 2001 | ○
| シャンパン | 4000 | 1999 |

```

コレツィオーネ チンクアンタ	2580	2011
マス・デ・タンヌ	1690	2014
トカイ・ソモロドニ・ドライ EV	1728	2000
トカイ・アッス・3 ブットニョッシュ EV	2484	2007
シュロス・フォルラーツ	2400	2012

-----+-----+-----

9 rows in set (0.00 sec)

以上の例題のように、同じ課題に対して解決方法は複数があります。自分が使い勝手のよい方法を選びましょう。また、自作したクエリーが正しいかどうかを、様々なデータをなるべく多めに用意して、丁寧にテストをしましょう。間違ったら、即時に直してみましよう。

➤ 演習課題 dbxx_1104(20 点, クエリーごとに 2 点)

課題ごとに課題番号と内容を含め、wineShopAplus へのクエリー命令およびその実行結果を収めたテキストファイル dbxx_1110.txt を sningping@kumamoto-nct.ac.jp へ添付で提出ください。

課題 : group by having の利用演習。

1. セット ID は 's=3' とするワインセットの価格を示せ。
2. 顧客 ID は 1 番とする注文されたワインの総本数と金額を示せ。
3. 6000 円～10000 円のワインセットの名前と価格を示せ。
4. 10000 円以上のワインセットの名前, 価格を示せ。
5. ワイン本数の和が 3 本以上のワインセットの名前, 価格, 本数の和を示せ。
6. 注文 ID は 3 番とする注文されたワインの総額を示せ。
7. 注文 ID 別に注文されたワインの金額は 3000 円以上となる顧客名, ワインの金額を示せ。
8. 注文 ID は 3 番とする注文されたワインセットの総額を示せ。
9. 注文 ID 別に注文されたワインセット総数が 2 以上となる顧客名, セットの総数を示せ。
10. 指定された注文日での店のワインの売り上げ金額を示せ。