

1. wineShopAplus のスキーマ

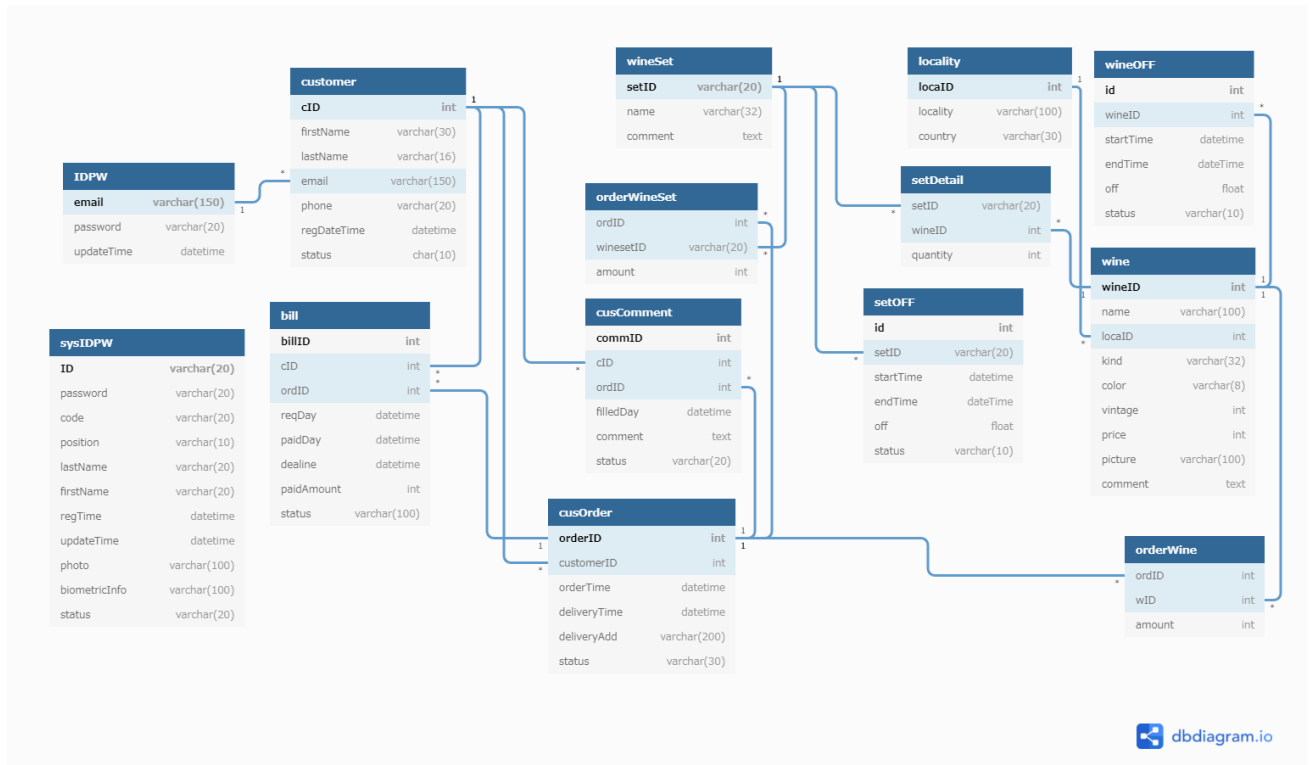


図 2. wineShopAplus の ERD

➤ group by を使ってデータをグループ化する

GROUP BY 句を使用することで SELECT 文でデータを取得する時に指定したカラム名の値を基準にグループ化することができます。集計やデータ抽出などに使われます。

select 属性 1, 属性 2, … from テーブル名 (結合も含む) where 条件 group by 属性名;

GROUP BY 句の後に指定したカラム名の値が同じデータをグループとしてまとめます。複数のカラム名を指定した場合には、カラムの値の組み合わせが同じデータをグループとしてまとめます。

グループ化すると SELECT 文で取得できるデータの数はグループの数だけとなります。そこでグループ化を行った場合には、グループ化に指定したカラムの値や、用意された関

数を使ってカラムの値をグループ単位で集計した結果などを取得することができます。

例題 1. 産地ごとにワインの最高価格, 最低価格, 平均価格を示せ. 但し, 日本語でカラム名を示す.

```
select locality as '産地', max(price) as '最高価格', min(price) as '最低価格', avg(price) as '平均価格' from wine natural join locality group by locality;
```

```
mysql> select locality as '産地', max(price) as '最高価格', min(price) as '最低価格', avg(price) as '平均価格' from wine natural join locality group by locality;
```

産地	最高価格	最低価格	平均価格
シャンパーニュ	4000	4000	4000.0000
ブルゴーニュ	3000	2400	2700.0000
ボルドー	5800	2200	4000.0000
ロワール	2800	2800	2800.0000
山形	1800	1800	1800.0000
山梨	15276	1800	4447.7143
熊本	2913	2913	2913.0000
長野	17820	886	6911.2000

例題 2. 色ごとにワインの最高価格, 最低価格, 平均価格を示せ. 但し, 日本語でカラム名を示す.

```
select color as '色', max(price) as '最高価格', min(price) as '最低価格', avg(price) as '平均価格' from wine natural join locality group by color;
```

```
mysql> select color as '色', max(price) as '最高価格', min(price) as '最低価格', avg(price) as '平均価格' from wine natural join locality group by color;
```

色	最高価格	最低価格	平均価格
白	6090	1800	3345.5455
赤	17820	886	5978.0000

2 rows in set (0.00 sec)

例題 3. ワインセットの価格を示せ.

```
select wineSet.setID as 'セット ID', wineSet.name as 'セット名', sum(price*quantity) as
```

'セット価格' from wineSet inner join setDetail using(setID) inner join wine using(wineID) group by setID;

```
mysql> select wineSet.setID as 'セット ID', wineSet.name as 'セット名', sum(price*quantity) as 'セット価格'
from wineSet inner join setDetail using(setID) inner join wine using(wineID) group by setID;
```

セット ID	セット名	セット価格
KW-1	甲州白ワインセット	6100
KW-2	甲州紅白ワインセット	4900
s-1	ブルゴーニュセット	5400
s-2	ボルドーセット	11400
s-3	白ワインセット	6400
s-4	赤ワインセット	17400

6 rows in set (0.00 sec)

例題 4. 各顧客が注文したワインの総額を示せ。

select cID,concat(lastName,firstName) as '顧客名', sum(price*amount) as '総額' from orderWine inner join cusOrder on cusOrder.orderID=orderWine.ordID inner join customer on customer.cID=cusOrder.customerID inner join wine on orderWine.wID=wine.wineID group by cusOrder.customerID;

```
mysql> select cID,concat(lastName,firstName) as '顧客名', sum(price*amount) as '総額' from orderWine inner
join cusOrder on cusOrder.orderID=orderWine.ordID inner join customer on
customer.cID=cusOrder.customerID inner join wine on orderWine.wID=wine.wineID group by
cusOrder.customerID;
```

cID	顧客名	総額
1	東山桜	6700
2	山村健	11600
3	伊集院紫	4000

3 rows in set (0.00 sec)

確認：

```
mysql> select * from orderWine;
```

ordID	wID	amount
1	3	2
2	8	1

```

| 4 | 10 | 1 |
| 4 | 9 | 1 |
| 5 | 6 | 1 |
| 5 | 8 | 1 |

```

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

6 rows in set (0.00 sec)

```
mysql> select * from cusOrder;
```

```

+-----+-----+-----+-----+
| orderID | customerID | orderTime          | deliveryTime      |
+-----+-----+-----+-----+
| 1 | 2 | 2020-06-30 12:57:56 | 2020-07-03 12:57:56 |
| 2 | 1 | 2020-06-30 12:58:18 | 2020-07-02 12:58:18 |
| 3 | 3 | 2020-06-30 12:58:29 | 2020-07-05 12:58:29 |
| 4 | 1 | 2020-07-01 14:10:23 | 2020-07-04 14:10:23 |
| 5 | 3 | 2020-07-01 14:22:06 | 2020-07-04 14:22:06 |
+-----+-----+-----+-----+

```

5 rows in set (0.00 sec)

```
mysql> select wineID, price from wine;
```

```

+-----+-----+
| wineID | price |
+-----+-----+
| 1 | 2400 |
| 2 | 3000 |
| 3 | 5800 |
| 4 | 2800 |
| 5 | 4000 |
| 6 | 2200 |
| 7 | 1800 |
| 8 | 1800 |
| 9 | 2400 |
| 10 | 2500 |
| 11 | 886 |
| 12 | 4620 |
| 13 | 17820 |
| 14 | 6090 |
| 15 | 2913 |
| 16 | 1800 |
| 17 | 5140 |
| 18 | 5400 |
| 19 | 1958 |
| 20 | 15276 |
+-----+-----+

```

20 rows in set (0.00 sec)

```
mysql> select cID, concat(lastName,firstName) from customer;
+-----+-----+
| cID | concat(lastName,firstName) |
+-----+-----+
| 1 | 東山桜 |
| 2 | 山村健 |
| 3 | 伊集院紫 |
+-----+-----+
3 rows in set (0.00 sec)
```

➤ 複合クエリー

2つ以上のクエリーを組み合わせ、集合演算を利用してデータを探し出します。同一の問い合わせに異なるクエリーで求められます。

例題5. 最高価格となる日本産ワインの情報を示せ。

```
select wineID,name,color,locality,price from wine natural join locality where price =(select max(price) from wine natural join locality where country='日本');
```

```
mysql> select wineID,name,color,locality,price from wine natural join locality where price =(select max(price) from wine natural join locality where country='日本');
+-----+-----+-----+-----+-----+
| wineID | name | color | locality | price |
+-----+-----+-----+-----+-----+
| 13 | 桔梗ヶ原メルロー | 赤 | 長野 | 17820 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

例題5. 値段はワインの平均価格以下となるワイン情報を示せ。

```
select wineID,name,color,locality,price from wine natural join locality where price <=(select avg(price) from wine) order by price desc;
```

```
mysql> select wineID,name,color,locality,price from wine natural join locality where price <=(select avg(price) from wine) order by price desc;
+-----+-----+-----+-----+-----+
| wineID | name | color | locality | price |
+-----+-----+-----+-----+-----+
| 5 | シャンパン | 白 | シャンパーニュ | 4000 |
| 2 | ジュヴレシャンベルタン | 赤 | ブルゴーニュ | 3000 |
| 15 | 菊鹿シャルドネ | 白 | 熊本 | 2913 |
| 4 | サンセール | 白 | ロワール | 2800 |
| 10 | 茅ヶ岳甲州 | 白 | 山梨 | 2500 |
```

```

| 1 | シャブリ | 白 | ブルゴーニュ | 2400 |
| 9 | ラフィーユ 樽ベリー-A | 赤 | 山梨 | 2400 |
| 6 | オーメドック | 赤 | ボルドー | 2200 |
| 19 | 甲州ドライ | 白 | 山梨 | 1958 |
| 8 | いろ甲州 | 白 | 山梨 | 1800 |
| 16 | バレルセレクションルージュ | 赤 | 山形 | 1800 |
| 7 | グリド甲州 | 白 | 山梨 | 1800 |
| 11 | 雪花 | 赤 | 長野 | 886 |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)

```

例題 6. 顧客に注文されなかったワインの情報を示せ.

```
mysql> select wineID, name, locality, color, price from wine natural join locality where
wineID not in(select wID from orderWine) order by wineID;
```

```
mysql> select wineID, name, locality, color, price from wine natural join locality where wineID not in(select wID
from orderWine) order by wineID;
```

```

+-----+-----+-----+-----+
| wineID | name | locality | color | price |
+-----+-----+-----+-----+
| 1 | シャブリ | ブルゴーニュ | 白 | 2400 |
| 2 | ジュヴレシャンベルタン | ブルゴーニュ | 赤 | 3000 |
| 4 | サンセール | ロワール | 白 | 2800 |
| 5 | シャンパン | シャンパーニュ | 白 | 4000 |
| 7 | グリド甲州 | 山梨 | 白 | 1800 |
| 11 | 雪花 | 長野 | 赤 | 886 |
| 12 | 塩尻 | 長野 | 赤 | 4620 |
| 13 | 桔梗ヶ原メルロー | 長野 | 赤 | 17820 |
| 14 | 北信シャルドネ RGC | 長野 | 白 | 6090 |
| 15 | 菊鹿シャルドネ | 熊本 | 白 | 2913 |
| 16 | バレルセレクションルージュ | 山形 | 赤 | 1800 |
| 17 | ヴィニュロンズリザーブ シャルドネ | 長野 | 白 | 5140 |
| 18 | ブリリヤンテ | 山梨 | 白 | 5400 |
| 19 | 甲州ドライ | 山梨 | 白 | 1958 |
| 20 | 絢 | 山梨 | 赤 | 15276 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

➤ 演習課題 dbxx_1021 (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取めたテキストファイル **dbxx_1021.txt** を sningping@kumamoto-nct.ac.j へ添付で提出ください.

A. group by の利用演習.

1. 国別にワインの最高価格, 最低価格, 平均価格を示せ.
2. 各顧客が注文されたワインセットのセット注文総数を示せ.
3. ワインセットのセット ID, セット名, 価格を示せ.
4. 注文 ID 別に注文されたワインの顧客名, ワインの金額を示せ.
5. 注文 ID 別に注文されたワインセットの顧客名, セットの総数を示せ.

B. 複合クエリーの利用演習.

1. 最安値となるワインの名前, 産地, 色, 値段を示せ.
2. 平均価格以下のフランス産ワインの名前, 産地, 色, 値段を示せ.
3. 平均価格以上の赤ワインの名前, 産地, 値段を示せ.
4. 顧客に注文されなかったワインセットのセット ID, セット名を示せ.
5. 顧客に注文されなかったワインの数を示せ.