

内容（講義と演習）

内容

1. MySQL の簡単なクエリー(query)
2. wineShopAplus へのデータクエリー (10点)

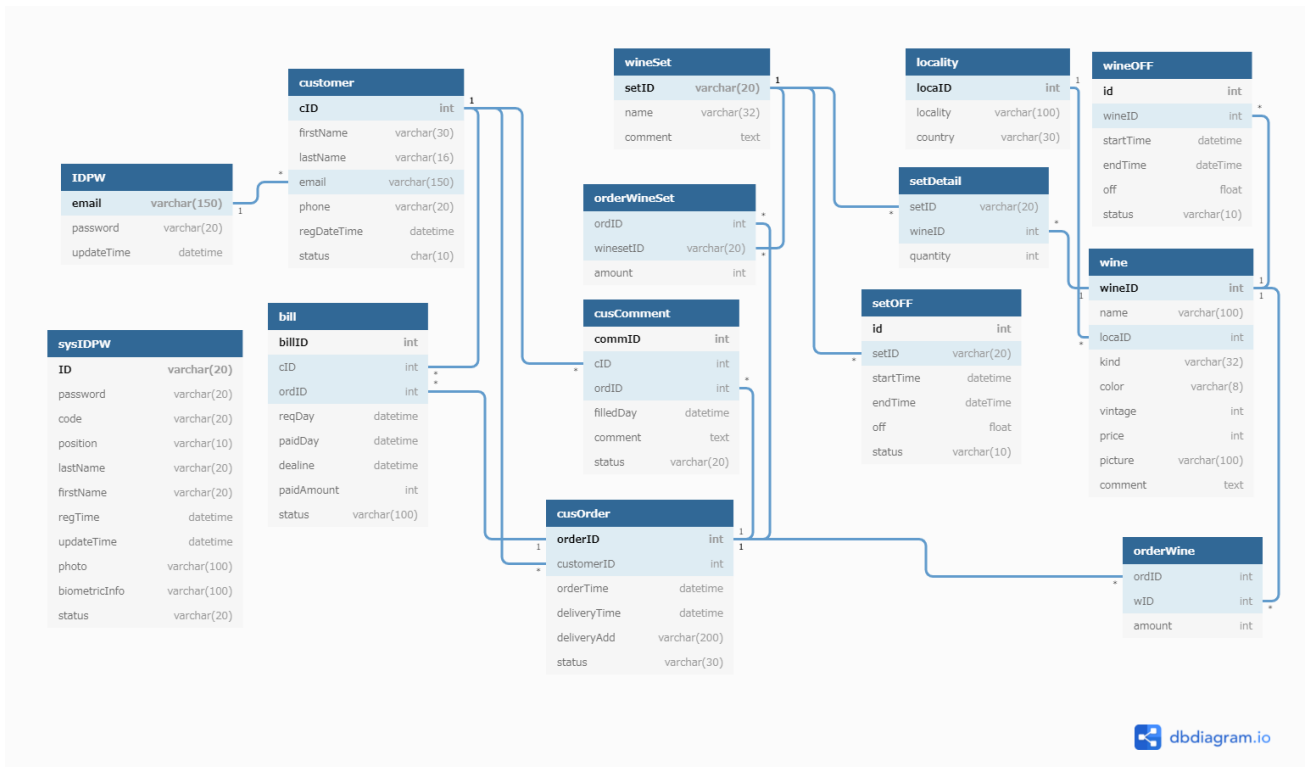


図1. wineShopAplus の ERD

1. SQL とは

SQLは「Structured Query Language」で、リレーショナルデータベース管理システム（RDBMS）において、データの操作や定義を行うためのデータベース言語（問い合わせ言語）、ドメイン特化言語（domain-specific language、DSL）で、特定のタスク向けに設計されたコンピュータ言語です。基本的に、会話的な命令の形使われていて、簡単なプロシージャや関数を組めます。プログラミングにおいてデータベースへのアクセスのために、プログラミング言語と併用されるが、SQL そのものはプログラミング言語ではありません。

SQLは90年代の初め頃から今までは、商用データベースだけでなくオープンソース・データベースを併用するケースも増えており、選択肢は多様化しています。SQLはRDBMS共通の言語ですが、実際は細かな記述の違いやRDBMS独自の機能が多数存在します。そのため、例えば商用データベースからオープンソース・データベ

スに移行したり併用したりすると、アプリケーションの改修コストや、意図した通りに動作しないといった問題が発生する場合があります。

世の中によく知られている SQL を取り扱い RDBMS は以下に示します。

- Oracle Database (Oracle)
- MySQL
- Microsoft SQL Server
- PostgreSQL
- MariaDB
- MongoDB

2. SQL クエリーの基本形

`select` カラムリスト (属性名リスト) `from` 表リスト `where` 条件;

① 単一の表へのクエリ

Ex1. `wine` の ID, 名前, 色, 産地 ID, 価格を示せ。

`select wineID, name, color, locaID, price from wine;`

実行結果：

```
mysql> select wineID, name, color, locaID, price from wine;
```

wineID	name	color	locaID	price
1	シャブリ	白	4	2400
2	ジュヴレシャンベルタン	赤	4	3000
3	サンテミリオン	赤	5	5800
4	サンセール	白	7	2800
5	シャンパン	白	1	4000
6	オーメドック	赤	5	2200
7	グリド甲州	白	8	1800
8	いろ甲州	白	8	1800
9	ラフィーユ 樽ベリー-A	赤	8	2400
10	茅ヶ岳甲州	白	8	2500
11	雪花	赤	9	886
12	塩尻	赤	9	4620
13	桔梗ヶ原メルロー	赤	9	17820
14	北信シャルドネ RGC	白	9	6090
15	菊鹿シャルドネ	白	10	2913
16	バレルセレクションルージュ	赤	11	1800

17	ヴィニユロンズリザーブ シャルドネ	白	9	5140
18	ブリリヤンテ	白	8	5400
19	甲州ドライ	白	8	1958
20	絢	赤	8	15276

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

20 rows in set (0.00 sec)

Ex2. wineSet の ID, 名前を示せ.

`select setID, name from wineSet;`

実行結果

```
mysql> select setID,name from wineSet;
+-----+-----+
| setID | name                |
+-----+-----+
| KW-1  | 甲州白ワインセット |
| KW-2  | 甲州紅白ワインセット |
| s-1   | ブルゴーニュセット |
| s-2   | ボルドーセット     |
| s-3   | 白ワインセット     |
| s-4   | 赤ワインセット     |
+-----+-----+
6 rows in set (0.00 sec)
```

Ex3. 2000 円以上の赤ワインの名前, 価格を降順に示せ.

`select name, price from wine where color='赤' and price>=2000 order by price desc;`

実行結果：

```
mysql> select name,price from wine where color='赤' and price>=2000 order by price desc;
+-----+-----+
| name                | price |
+-----+-----+
| 桔梗ヶ原メルロー   | 17820 |
| 絢                  | 15276 |
| サンテミリオン     | 5800  |
| 塩尻                | 4620  |
| ジュヴレシャンベルタン | 3000  |
| ラフィーユ 樽ベリーA | 2400  |
| オーメドック       | 2200  |
+-----+-----+
7 rows in set (0.01 sec)
```

Ex4. 白ワインの最高額, 最低額, 平均額を示せ.

```
select max(price), min(price), avg(price) from wine where color='白';
```

実行結果

```
mysql> select max(price), min(price), avg(price) from wine where color='白';
+-----+-----+-----+
| max(price) | min(price) | avg(price) |
+-----+-----+-----+
|          6090 |          1800 | 3345.5455 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Ex 5. ワイン産地の数を示せ.

```
select count(distinct(localID)) from wine;
```

実行結果

```
mysql> select count(distinct(localID)) from wine;
+-----+
| count(distinct(localID)) |
+-----+
|                          8 |
+-----+
1 row in set (0.00 sec)
```

② 2つの表へのクエリ (primary key と foreign key を持って結合する)

Ex1. 白ワインの中で, 2500 円以上のワインの名前, 産地, 国名, 価格を降順で示せ.

テーブル locality の主キー localID と, テーブル wine の外部キー localID が 2つのテーブルを繋げさせるので, inner join を利用して結合を行い探索します.

```
select name, locality, country, price from wine inner join locality
using(localID) where color='白' and price>=2500 order by price desc;
```

実行結果

```
mysql> select name, locality, country, price from wine inner join locality using(localID) where color='白' and price>=2500 order by price desc;
+-----+-----+-----+-----+
| name                | locality    | country | price |
+-----+-----+-----+-----+
| 北信シャルドネ RGC | 長野       | 日本    | 6090 |
+-----+-----+-----+-----+
```

ブリリヤンテ	山梨	日本	5400
ヴィニュロンズリザーブ シャルドネ	長野	日本	5140
シャンパン	シャンパーニュ	フランス	4000
菊鹿シャルドネ	熊本	日本	2913
サンセール	ロワール	ドイツ	2800
茅ヶ岳甲州	山梨	日本	2500

+-----+-----+-----+-----+
7 rows in set (0.00 sec)

Ex2. ワイン名前の中で一部分が“サン”となっているワインの名前, 色, 価格, 国名を示せ.

条件判断: name like '%サン' 末尾が サン
'san%' 先頭が サン
'%san%' 間中が サン

```
select name,color,price,country from wine inner join locality using(localID)
where name like '%san%';
```

実行結果

```
mysql> select name,color,price,country from wine inner join locality using(localID) where name like '%san%';
```

name	color	price	country
サンテミリオン	赤	5800	フランス
サンセール	白	2800	ドイツ

2 rows in set (0.00 sec)

Ex3. setID が 「s-1」となっているワインセットの構成 (ワインの名前, 色, 値段) とそれぞれの本数を示せ.

テーブル wine の主キー wineID と, テーブル setDetail の外部キー wineID が 2 つのテーブルを繋げさせるので, inner join を利用して結合を行い探索します.

```
select name, color, price, quantity from setDetail inner join wine using(wineID) where
setDetail.setID='s-1';
```

実行結果 1

```
mysql> select name,color,price,quantity from setDetail inner join wine using(wineID) where setDetail.setID='s-1';
```

name	color	price	quantity
シャブリ	白	2400	1

```
| ジュヴレシヤンベルタン | 赤 | 3000 | 1 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

実行結果 2

```
mysql> select name,color,price,quantity from wine inner join setDetail using(wineID) where setDetail.set
ID='s-1';
```

```
+-----+-----+-----+
| name | color | price | quantity |
+-----+-----+-----+
| シャブリ | 白 | 2400 | 1 |
| ジュヴレシヤンベルタン | 赤 | 3000 | 1 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Ex4. IDPW に登録した顧客の氏名とメールアドレスを示せ.

テーブル IDPW の主キーemail と、テーブル customer の外部キーemail が 2 つのテーブルを繋げさせるので、inner join を利用して結合を行い探索します。concat 関数は文字列を結合させ、as name は結果表示名を'name'にしてくれます。

```
select concat(lastName, firstName) as name, email from IDPW inner join
customer using(email);
```

実行結果

```
mysql> select concat(lastName,firstName) as name, email from IDPW inner join customer using(email);
```

```
+-----+-----+
| name | email |
+-----+-----+
| 東山桜 | Alpha@where.com |
| 山村健 | Bega@where.com |
| 伊集院紫 | Gama@where.com |
+-----+-----+
3 rows in set (0.00 sec)
```

Ex5. 注文した顧客の氏名とメールアドレス, 注文日時, 希望する配達日時を示せ.

テーブル customer の主キーは cID とし、テーブル cusOrder の外部キーcustomerID が customer の cID に参照するが、名前が違うので、inner join を利用して結合を行うことはできません。この場合は、where 条件の中に結合条件を入れます。

```
select concat(lastName,firstName), email, orderTime, deliveryTime from cusOrder,
```

```
customer where customer.cID=cusOrder.customerID;
```

実行結果

```
mysql> select concat(lastName,firstName), email, orderTime,deliveryTime from cusOrder,customer where customer.cID=cusOrder.customerID;
```

```
+-----+-----+-----+-----+
| concat(lastName,firstName) | email          | orderTime          | deliveryTime      |
+-----+-----+-----+-----+
| 東山桜                     | Alpha@where.com | 2020-06-30 12:58:18 | 2020-07-02 12:58:18 |
| 山村健                     | Bega@where.com  | 2020-06-30 12:57:56 | 2020-07-03 12:57:56 |
| 伊集院紫                   | Gama@where.com  | 2020-06-30 12:58:29 | 2020-07-05 12:58:29 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

****番外編****

また、deliveryTime を記入する際、orderTime に時間の加算方法は使いました。

テーブルに新しいデータを追加する時、次の命令を使います。

```
insert into cusOrder(customerID, orderTime, deliveryTime) values(1,now(),(now()+interval 3 day));
```

データ更新する時に、次の命令を使います。

```
update cusOrder set deliveryTime=(orderTime+interval 5 day) where orderID=1;
```

interval の前にマイナス・プラス+記号をつけることで、減算・加算ができます。

+(-) INTERVAL 数値 単位 という構文になります。単位は、

MICROSECOND
SECOND
MINUTE
HOUR
DAY
WEEK
MONTH など

➤ 演習課題 dbxx_1007(10 点)

以下の課題ごとに課題番号と内容を含め、wineShopAplus へのクエリ命令およびその実行結果を収めたテキストファイル **dbxx_1007.txt** を sningping@kumamoto-nct.ac.j へ添付で提出ください。

➤ 単一の表に対して

1. ワインの名前, 色, 価格を価格の降順で示せ.
 2. ワインの最高価格, 最低価格, 価格の平均を示せ.
 3. ワイン種類の数を示す.
 4. 2500 円以下の赤ワインの名前, 種類, 価格を降順で示せ.
 5. 2000 年以前のワインの名前, 色, 価格を降順で示せ.
- 二つの表の結合に対して
6. 日本産のワインの wineID, 名前, 色, 産地, 価格を示せ.
 7. フランス産のワインの最高価格, 最低価格, 価格の平均を示せ.
 8. 5000 円以下の白ワインの名前, 価格, 産地, 種類を示せ.
 9. ID とパスワードおよび各対応した顧客 customer の情報を lastName のアルファベット昇順に示せ. *昇順は asc
 10. 注文した customer 顧客の情報示せ.