

内容（講義と演習）

内容

1. MySQL のデータ入力とテーブル操作
2. wineShopDB へのデータ入力の演習

1. ワインとワインセット管理用のスキーマ

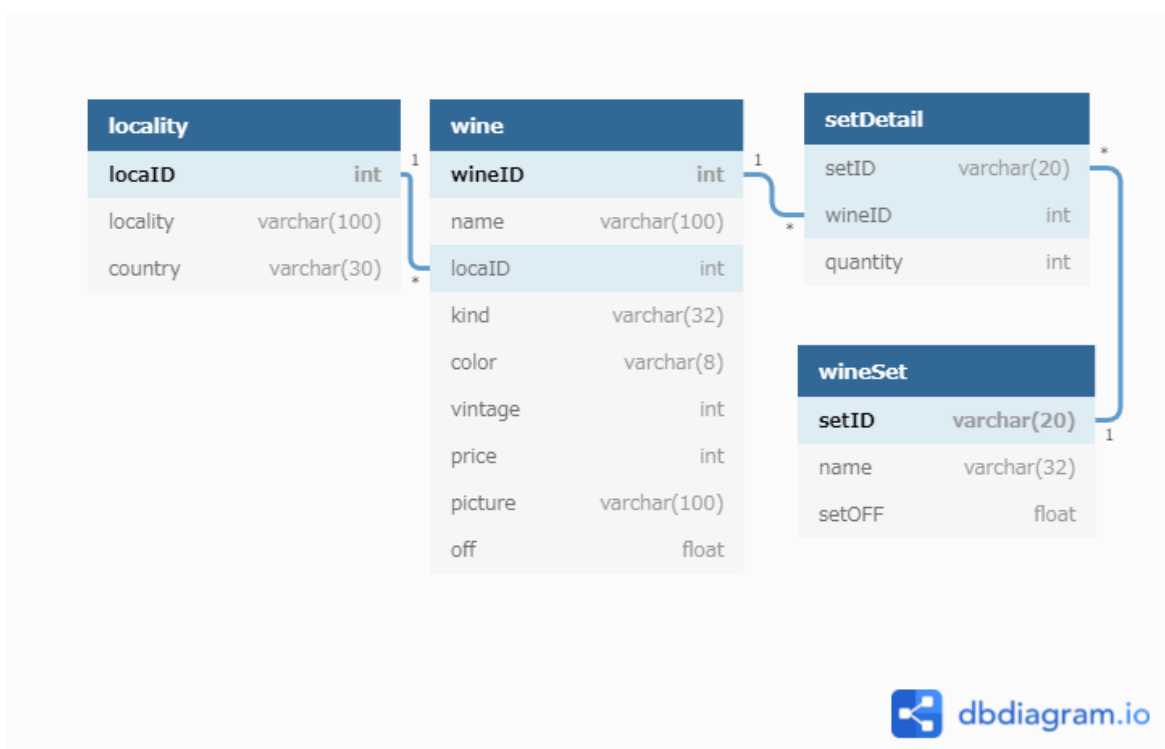


図1 scheme of wine and wineSet

2. データの入力

➤ 原始データ

<http://apricot.hi.kumamoto-nct.ac.jp:8099/~snngping/DB/ERD/wineData.sql>

➤ MySQL データ挿入命令

```
insert into テーブル名(属性名 1, 属性名 2, …) values(値 1, 値 2, …);
```

注意事項：

- ✧ **auto-increment** に指定された **primary key** は値を代入できない。
- ✧ 全ての属性名に値を代入する場合は、属性名を羅列しなくてもよいが、値一覧の順序はスキーマの定義に一致しなければなりません。

```
insert into テーブル values(値 1, 値 2, …);
```

- ✧ **属性の型と値の型は絶対一致すること。**
- ✧ 参照される可能性があるテーブルを先に作り、データ入力も先にやること。

3. テーブルの操作命令(1)

➤ テーブルの属性を確認する

```
desc テーブル名；
```

➤ テーブルの中身を確認する

```
select * from テーブル名；
```

➤ テーブルのデータ（行）を削除する

```
delete from テーブル名 where 削除条件；
```

例えば,

delete from locality where locaID >= 8;

- ✚ 削除後, 今後のデータ挿入はテーブルに残しているデータの直後に引き続きに入れて穴を空けないよう, 主キー(primary key)の auto_increment 値の連番をリセットしなければなりません. 例えば, 削除した後最後の記録の主キーの locaID は7となる時,

alter table locality auto_increment=8;

4. テーブルの操作命令(2)

➤ データの更新

update テーブル名 set 属性名=更新する値 where 更新する条件;

例えば,

```
update wineSet set setID='J-1' where setID='s-5';
```

*注意事項: 挿入済みデータに対して, 値を変えたい場合は update しか使えないこと. また, (where 条件)で条件付き更新を行うことは忘れないでください. 条件を書かないと, テーブルにあるすべての行に対してデータ更新をしてしまい, 元に戻らないこととなります.

➤ テーブルのカラムを追加する (属性 (列名) を増す)

alter table テーブル名 add 新カラム名 新カラムの型;

例えば, オンラインワインショップにワインの写真を掲示したいという要求がありました.

1. まず, 既存のテーブル wine に画像ファイル名を入れるための新しいカラムを追加します.

alter table wine add picture varchar(150);

2. 次に, オンラインでワイン写真を掲示できるように, public_html の下に画像の置き場所を決め, 画像をアップロードしておきます.
3. 他のデータが既に入力済みなので, テーブルのデータ更新を行います.

update wine set picture='../images/wine1.jpg' where wineID=1;

4. PHPなどで写真を探し出し、店のページに掲示します。

➤ 作成済みのテーブルからカラムを削除する

`alter table テーブル名 drop カラム名;`

➤ テーブルのカラムの定義を変更する

`alter table テーブル名 change 旧カラム名 新カラム名 新カラムの型;`

例えば、setDetailの「本数」を示すカラム名は「number」を「quantity」に変更したいと、次の命令を実行します。

`alter table setDetail change number quantity int;`

カラムはテーブルやDBの構造に関連するので、カラムの名前や型を変えたり、カラムを追加したり、削除したりすることで、DBの構造も変えてしまいます。よって、開発においてDBの構造を変更する前に、まずDBのスキーマやERDの変更を行わなければなりません。

付録 MySQLのソースからの実行方法

ステップ1. linuxサーバー上のある場所(public_html/DB)でsqlソースを作成する。

<http://apricot.hi.kumamoto-nct.ac.jp:8099/~sningping/DB/insertWineData.pdf>

のように、viかvimでsqlソースを書いて、ファイル名は「wineData.sql」として保存する。

例えば、<http://apricot.hi.kumamoto-nct.ac.jp:8099/~sningping/DB/wineData.sql>

ステップ2. ステップ1と同じ場所(public_html/DB)でMySQLにログインします。

```
mysql -u db○○ -p
```

-> password:

ログインができれば、データベースに接続する。

`use db○○`

ステップ 3. sql ソースを一回実行する.

```
source wineData.sql
```

「OK」というレスポンスが出れば、実行ができました. Error メッセージが出れば、ソースにミスあるので、やり直しなければなりません. 但し、成功した部分は重複して再度実行しないこと. 間違った部分だけやり直せば結構です.

ステップ 4. 入力後、下記の方法で入力を確認する.

- ① `select * from locality;`
- ② `select * from wine;`
- ③ `select * from wineSet;`
- ④ `select * from setDetail;`