

データベース

MySQL(XAMPP)編

データベースについて

DBMSとは？

データベースとは、複数のユーザが情報を共有して使用するための場所の事です。1950年頃、米軍が各地に点在した膨大な量の情報を一ヶ所の基地に集めて、そこにアクセスすれば全ての情報が得ることが出来るように効率化を図る目的でこのシステムを作りました。「情報(データ)の基地(ベース)」ということで、このシステムのことを「情報基地(データベース)」と呼ぶようになったと言われています。

それでは、この「データベース」をイメージするために「図書館」を例にして考えてみます。

図書館には膨大な量の本があり、色々なジャンルの本があります。そして図書館には、多くの利用者がさまざまな目的を持って訪れます。ここで、小学生が夏休みの宿題の読書感想文を書くために、「太宰治」の「走れメロス」という本を借りに来ました。さて、小学生はどうやってこの本を探すでしょうか？

きっとその小学生は図書館の司書に、作者は「太宰治」、作品名は「走れメロス」の本の場所を尋ねることで本のある棚の場所や本の分類等を教えてもらうことになるでしょう。また、その本が「貸出可能」か「貸出中」なのかも教えてくれます。

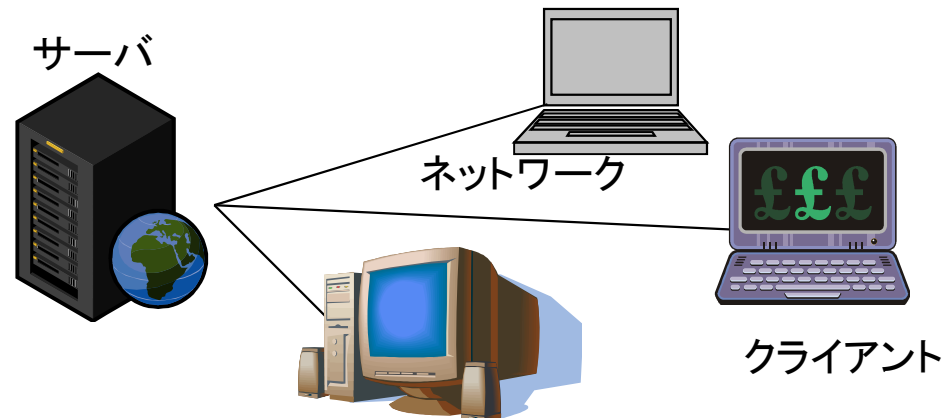
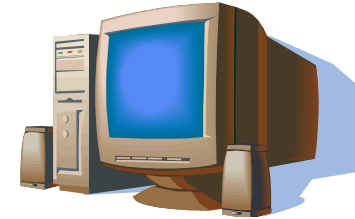
図書館の司書の仕事としては、本棚に本を追加したり、別の本棚に移動したり、また貸出状況を更新しておくなど、常に情報を最新に保つことです。情報が最新ではなかった場合、実際の場所には全然違う本が置いてあるかもしれません。

図書館を快適に皆で使用するためには、ただ漠然と本棚に本があるだけではだめで、情報の追加・更新・削除などデータを管理する必要があります。この時の情報を格納する本棚がデータベースに当たり、本棚(データベース)を管理する司書が、データベース管理システム(DBMS:Database Management System)に当たります。

データベースの構成

スタンドアロン構成

データベースとアプリケーションを同一マシンで動作させる構成です。ネットワークを経由することなく直接、データベースを操作することができます。個人または少数のユーザで使用します。

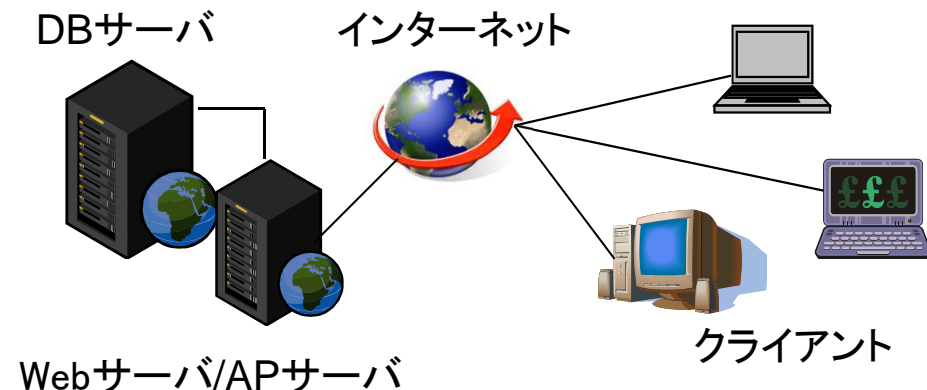


クライアント／サーバ構成

データベースをサーバマシン上で動作させ、クライアントアプリケーションからDBサーバ上のデータベースへアクセスをする構成です。ネットワークはLANやWANで構成されるのが一般的です。

ウェブ構成

現在主流の構成で、クライアントはブラウザを利用してインターネットにアクセスし、Webサーバに接続します。WebサーバはAPサーバ上で動作しているアプリケーションを使用して、DBサーバ上のデータベースに接続します。インターネットを利用することで、世界中どこからでも接続可能です。



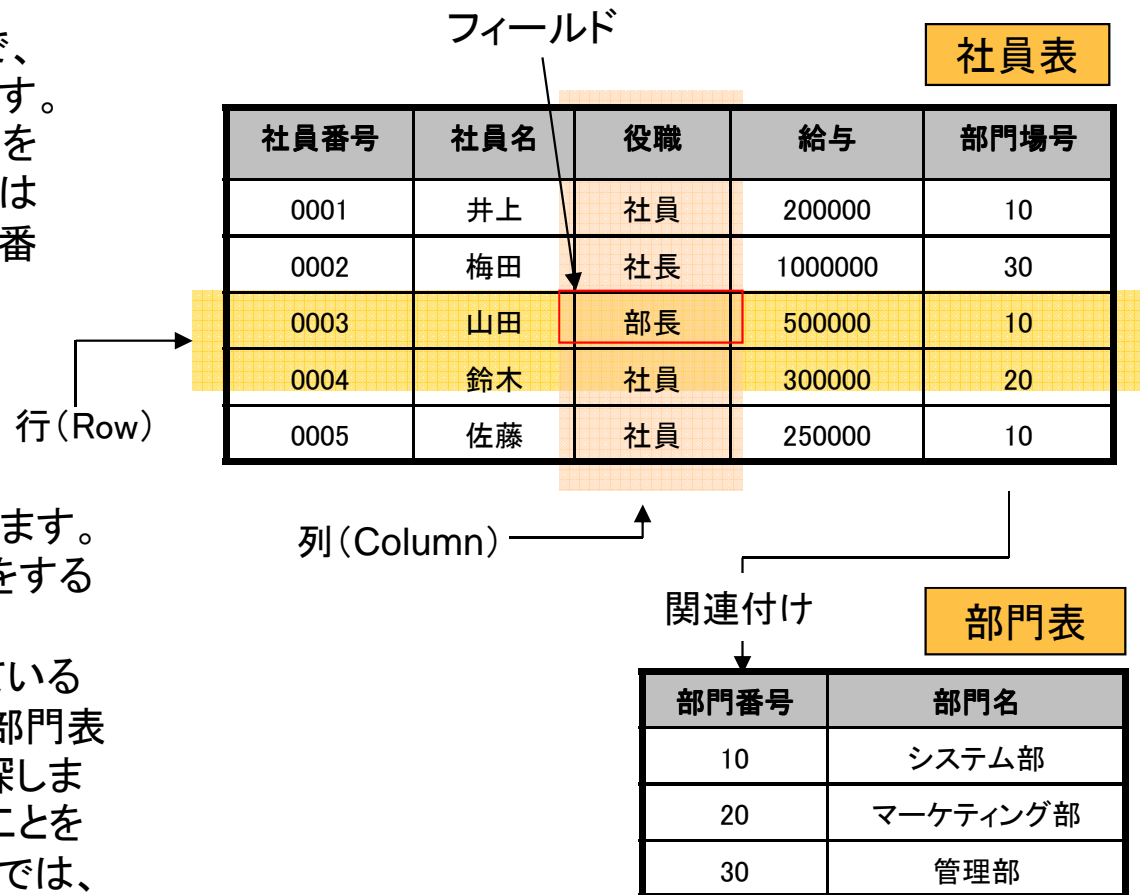
表とは？

表

表(テーブル)はデータを格納するオブジェクトで、リレーショナルデータベースの基本となるものです。Excelのように縦軸と横軸の2次元形式で、縦軸を列(Column)、横軸を行(Row)といいます。縦軸はデータの属性をあらわします。図の例では「社員番号」や「社員名」になります。横軸はデータです。

図の例では会社には社員が100人いたとしたら、社員表には100行のデータが存在することになります。1行のデータを「1レコード」や「1件」という言い方をすることもあります。

社員表には部門番号があります。社員が属している部門名を知りたい場合は、社員表の部門番号と部門表の部門番号が同じデータになっている部門名を探します。社員表と部門表を部門番号で関連付けすることをリレーションシップを張るなどと言います。図の例では、「社員番号」が「0001」のデータは「部門番号」が「10」なので、「部門名」は「システム部」ということになります。



※解説書によって「フィールド」の意味にブレがあります。

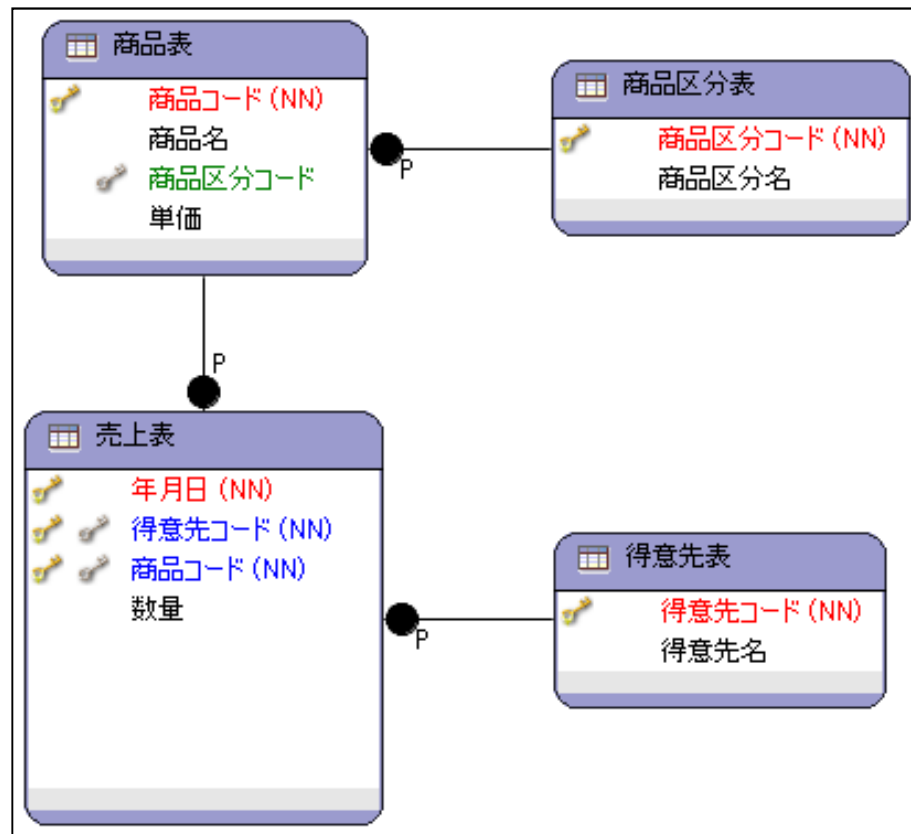
各種制約

制約とは、テーブルに格納するデータを制限する方法です。データ型によって整数であるか、文字列であるかといったデータの種類を限定することはできますが、それでは正数のデータのみを受け付けるとか、同じデータが重複してはいけないとかといった制限を行うことができません。制約はこのような制限を列やテーブルに対して定義することができます。制約に違反するデータを格納しようとするとエラーとなります。

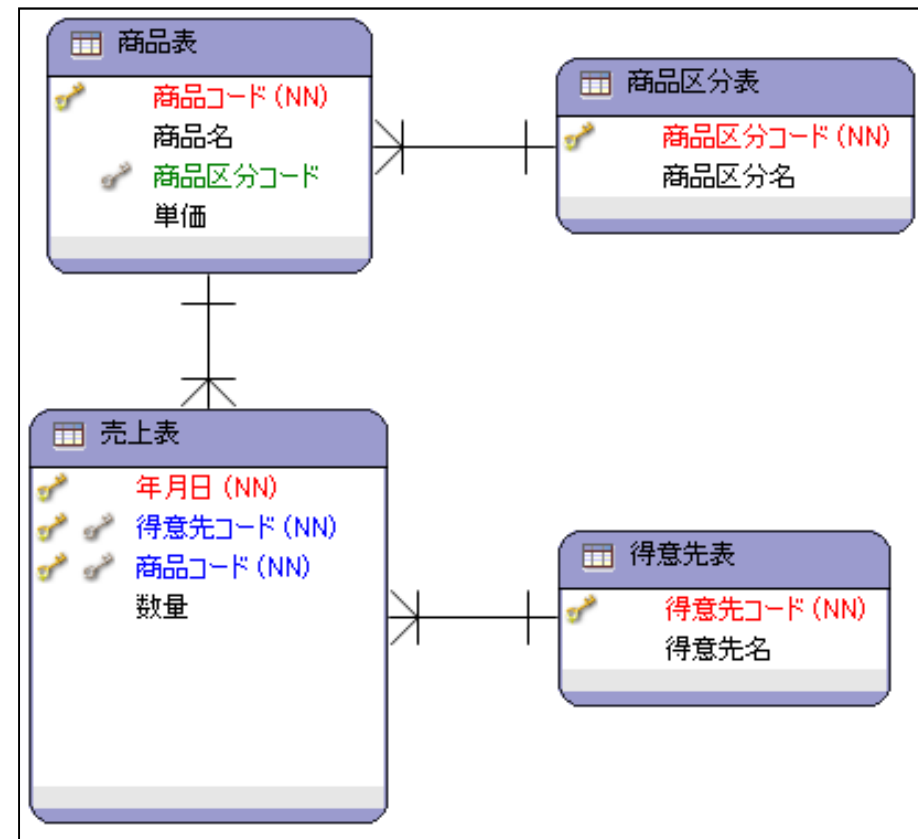
制約名	解説
NOT NULL	NOT NULL制約は、単一の列に対して定義されるルールで、この列のNULL値（値がない）の入力を禁止するという制約です。即ち、テーブルのすべての行で必ずその列にNULL以外の値が格納されるように制限する制約です。
一意性 (UNIQUE)	一意性制約は、列、または複数の列の組み合わせに対して定義されるルールで、それらの列で値が一意であるデータに限りデータの挿入・更新を許可する制約です。ユニークキーとも呼ばれます。
主キー (PRIMARY KEY)	主キー制約は、列、または複数の列の組み合わせに対して定義されるルールで、表の中の行を一意に識別するための制約です。一意性制約と違って、主キーは1つの表に1つしか設定出来ません。また主キー制約は、一意性制約にNOT NULL制約が加わったものとして見ることができます（つまり定義された列、または複数の列は重複が許されずNULLも許されない）。
外部キー (FOREIGN KEY)	外部キー制約は、表と表の間を表内の列（または列の組み合わせ）で参照という関連付けを定義するルールといいます。外部キー制約は表同士の関連付けを行い、子表は親表を参照し、子表の参照列を外部キーと呼びます。親表と子表の間でデータの整合性を保ちます。子表に入力される値は親表に存在する値かNULLでなければなりません。参照整合性制約やREFERENCE KEYなどとも呼ばれます。
表チェック (CHECK)	表チェック制約は、入力条件を定義した制約で、列に条件を満たす値のみ入力が許可されます。

ER図(Entity-Relationship Diagram、ERD)とはデータ項目の集まりであるエンティティと、エンティティ同士の繋がりであるリレーションシップを図示したものです。ER図の表記方法としてはChen表記、IDEF1X表記(アイデフ・ワンエックス)、IE表記などがあります。

IDEF1X表記



IE表記



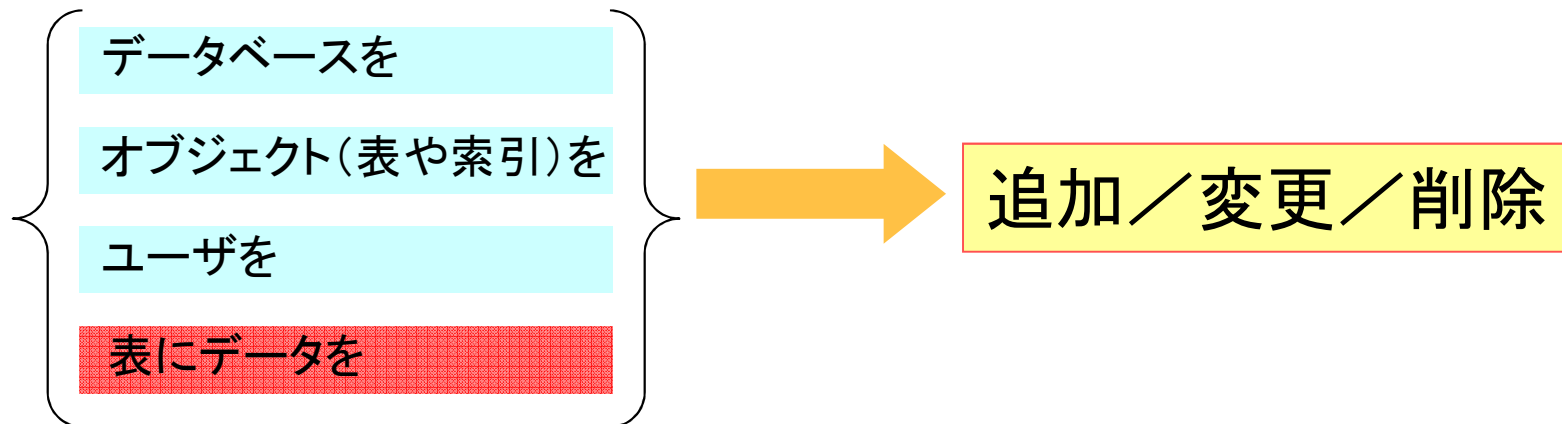
DB操作

ユーザ

データベースを扱うためには、データベースに接続する必要があります。データベースに接続するには、データベースを利用するユーザとなって認証をする必要があります。認証には、通常「ユーザ名」と「パスワード」が必要です。Oracleデータベースではユーザごとに「スキーマ」が割り当てられ、表などのオブジェクトはスキーマに格納されます。一方でMySQLでは「スキーマ」はデータベースと同じ意味として扱われ、表などのオブジェクトはデータベースに格納されます。Oracleでは、スキーマ名はユーザ名と同じ名前が設定されるためしばしば両者は同じように扱われます。MySQLでは、前述したようにスキーマとデータベースは同じ意味として扱われます。

SQL

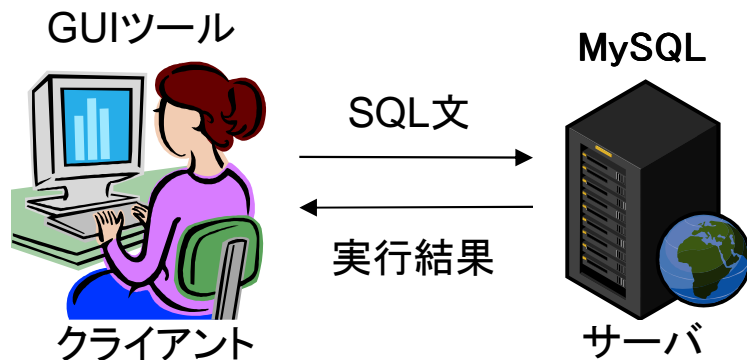
リレーショナルデータベース管理システムを利用するには、SQL（昔はStructured Query Languageと呼ばれていた）という言葉語を利用します。データの操作や定義をします。SQLを実行することで次のようなことができます。



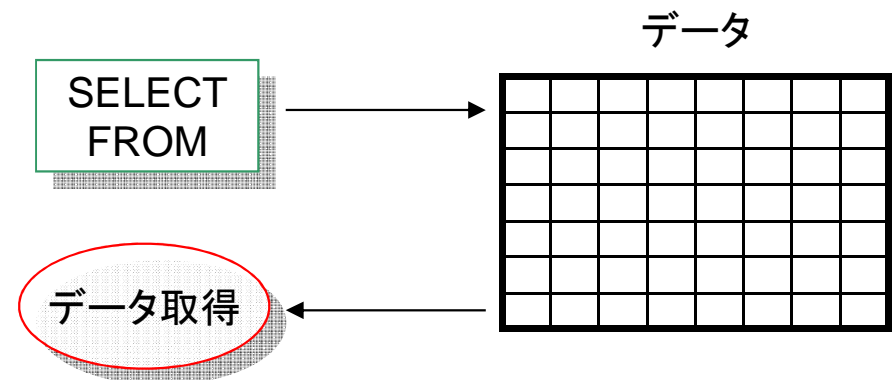
SQLとは？

SQL(エスキューエル)は、データベースを操作するための言語です。
データベースを扱うのに、SQLが分からないと何もすることができません。
ですので、SQLは必要最低限の知識として必ず覚えるようにしましょう。
また、SQLは、ANSI(米国規格協会)やISO(国際標準化機構)で言語仕様の標準化が行われているため、MySQL以外のデータベースでもほとんど同じように利用することができます(多少の違いはあります)。

●GUIツールを使って、データを取得



●SELECT文を使ってデータを取得



SQLの機能

SQLの機能として、以下の機能があります。

- ・データベースのデータを検索・追加・更新・削除する
- ・データベースのオブジェクト(表や索引など)と作成・削除する
- ・データベースのトランザクション(データを変更する作業単位)の制御を行う

それぞれの機能は、SQLの中で次のように分類されます。

データ操作言語(DML:Data Manipulation Language)文

SELECT文	表に格納されているデータを取得する
INSERT文	表にデータを追加する
UPDATE文	表に格納されているデータを更新する
DELETE文	表に格納されているデータを削除する

データ定義言語(DDL:Data Definition Language)文

CREATE文	データベースのオブジェクトを作成する
ALTER文	データベースのオブジェクトの定義を変更する
DROP文	データベースのオブジェクトを削除する

トランザクション制御文

COMMIT文	トランザクションを確定する
ROLLBACK文	トランザクションを取り消す

データ制御言語(DCL:Data Control Language)文

GRANT文	オブジェクトを操作する権限をユーザに与える
REVOKE文	オブジェクトを操作する権限をユーザから奪う

データ定義言語 (DDL) 文

CREATE文 (データベース)

書式

```
CREATE DATABASE database_name [create_specification]
```

引数

database_name データベース名 (最大64バイト)

create_specification 作成するデータベースの特徴

解説

CREATE DATABASE文を実行すると、汎用的なデータベースが作成される。CREATE DATABASEに続いて、作成するデータベース名 (*database_name*) を記述する。作成するデータベースの特徴は、デフォルトの文字セット、デフォルトのデータベース照合順序を指定することができます。

CREATE文 (テーブル)

書式

```
CREATE TABLE table_name [column_definition [, column_definition]]
```

引数

table_name テーブル名 (最大64バイト)

column_definition 列定義

解説

CREATE TABLE文を実行すると、テーブルが作成される。CREATE TABLEに続けて、表名 (*table_name*) を、カッコ内に列名、型、制約などを記述する。

データ定義言語 (DDL) 文

ALTER TABLE 文

書式

ALTER TABLE *table_name* {ADD | MODIFY} *column_definition*

ALTER TABLE *table_name* {ADD | MODIFY} CONSTRAINT *constraint_definition*

ALTER TABLE *table_name* DROP *column_name*

ALTER TABLE *table_name* DROP CONSTRAINT *constraint_name*

ALTER TABLE *table_name* MODIFY *column_definition*

引数

<i>table_name</i>	表名
<i>column_definition</i>	列定義
<i>column_name</i>	列名
<i>constraint_definition</i>	制約定義

解説

列の定義を変更する。変更方法はおもにADD,DROP,MODIFYの3つに分けられる。ALTER TABLEに続き、表名を記述する。ADDまたはMODIFYに続き列定義(*column_name*)を記述し、追加または修正する列定義を指定する。列定義は列名とデータ型を1つ以上指定可能である。複数の列を追加する場合はかっこでくる。また、CONSTRAINTにより制約を追加、修正することもできる。

DROPに続き列名(*column_name*)を記述し、列を削除する。列名は1つ以上指定可能であるが、複数の場合はかっこでくる。CONSTRAINTにより制約の削除も可能である。

データ定義言語 (DDL) 文

DROP文

書式

```
DROP TABLE table_name [CASCADE CONSTRAINTS]
```

引数

table_name 表名

解説

表と表に含まれるすべてのデータをデータベースから削除する。DROP TABLEに続けて、表名 (*table_name*) を記述する。CASCADE CONSTRAINTSを指定した場合は、主キーや一意キーを参照しているすべての参照整合性制約を削除する。
※DROP TABLE文はロールバックできないので、注意が必要である。

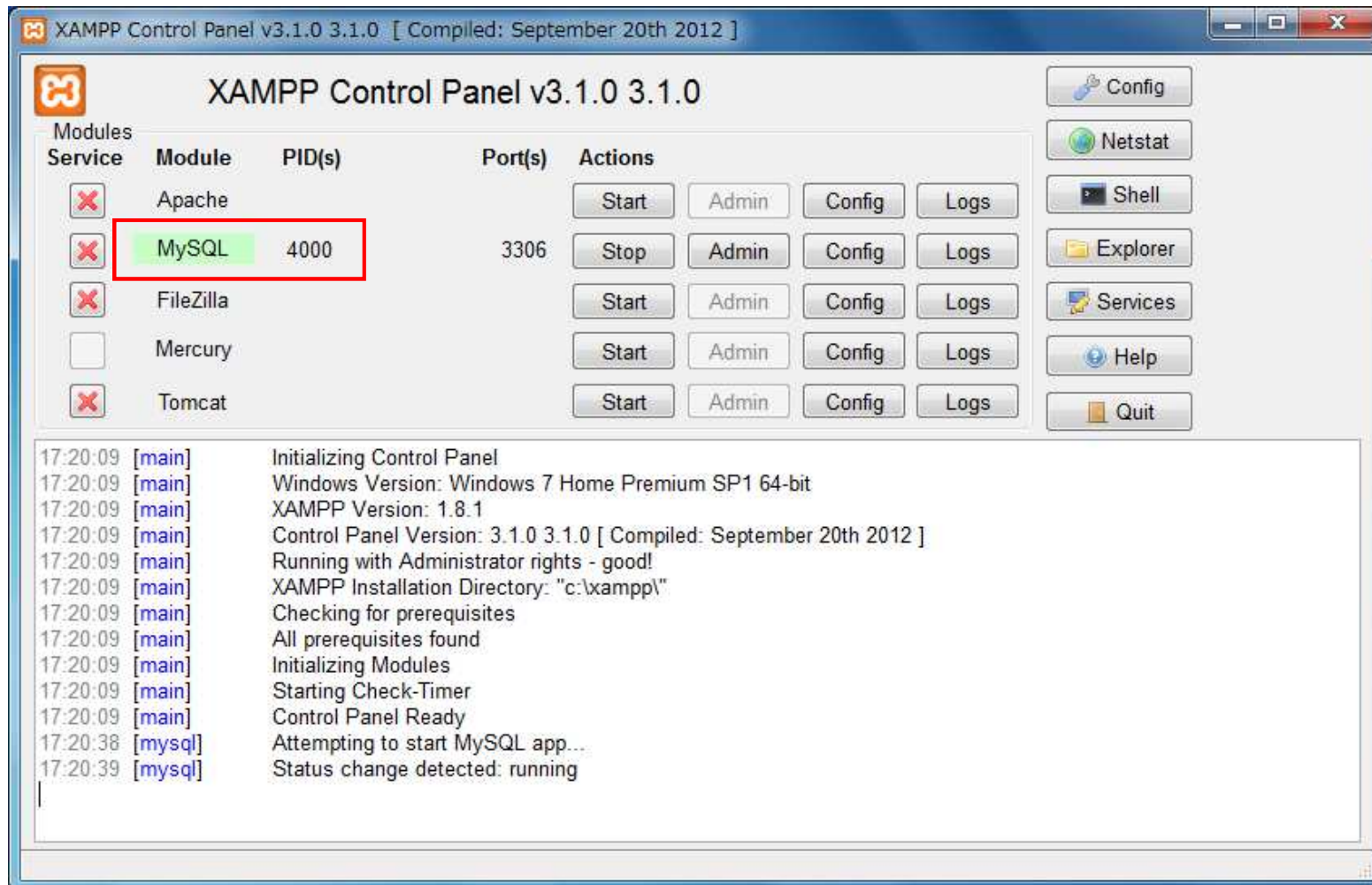
各SQLの詳細な構文については公式のリファレンスを参照してください。

<http://dev.mysql.com/doc/refman/5.1/ja/data-definition.html>

MySQLの設定

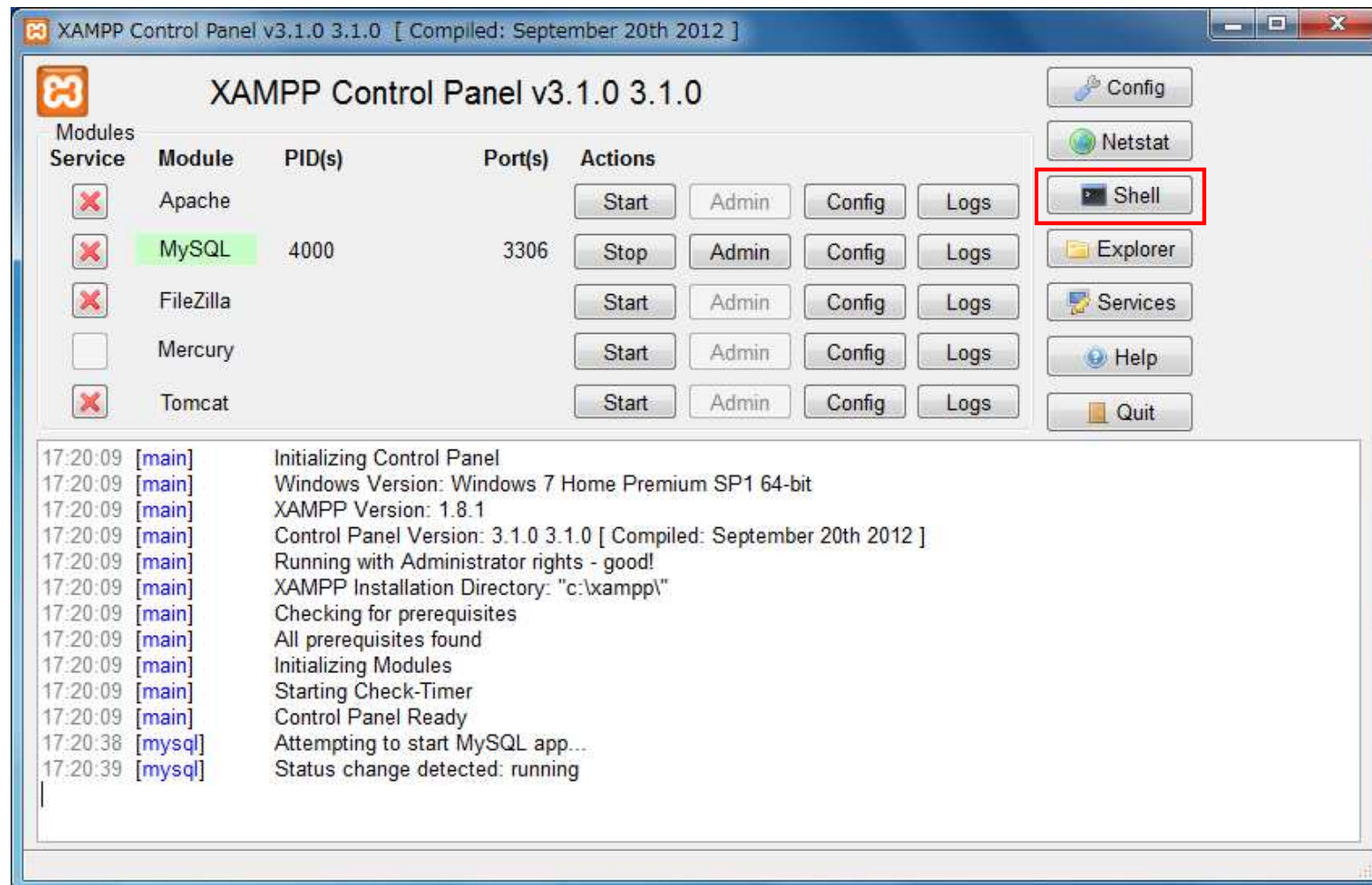
まずはMySQLに接続をします。

XAMPPコントロールパネルを開き、起動していることを確認します。



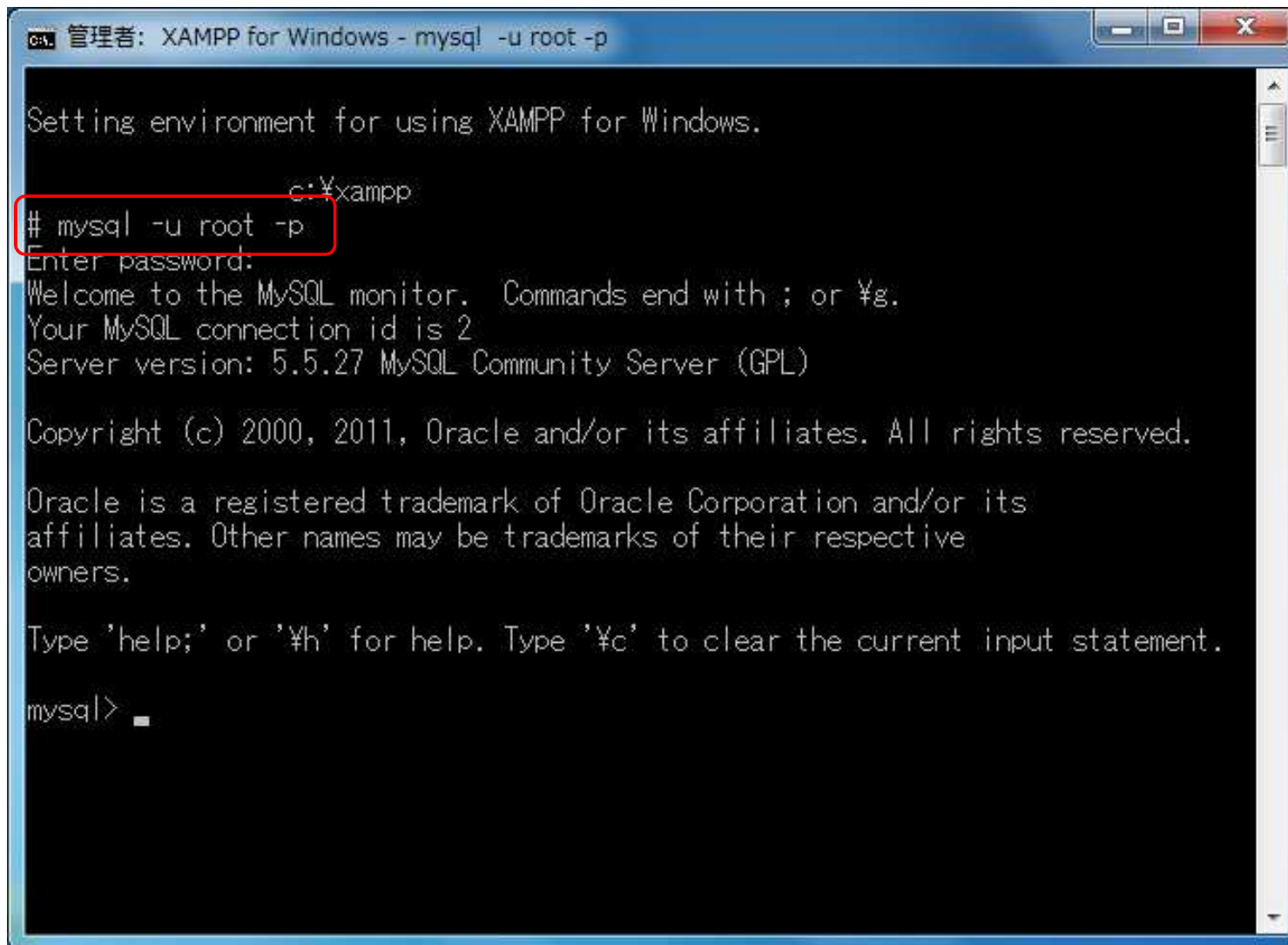
MySQLの設定

XAMPPコントロールパネルからShellを起動します。



MySQLの設定

Shellが起動したら「mysql -u root -p」とコマンドを打ち込みます。パスワードは空エンターをします。



```
管理者: XAMPP for Windows - mysql -u root -p

Setting environment for using XAMPP for Windows.

c:\xampp> # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.27 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

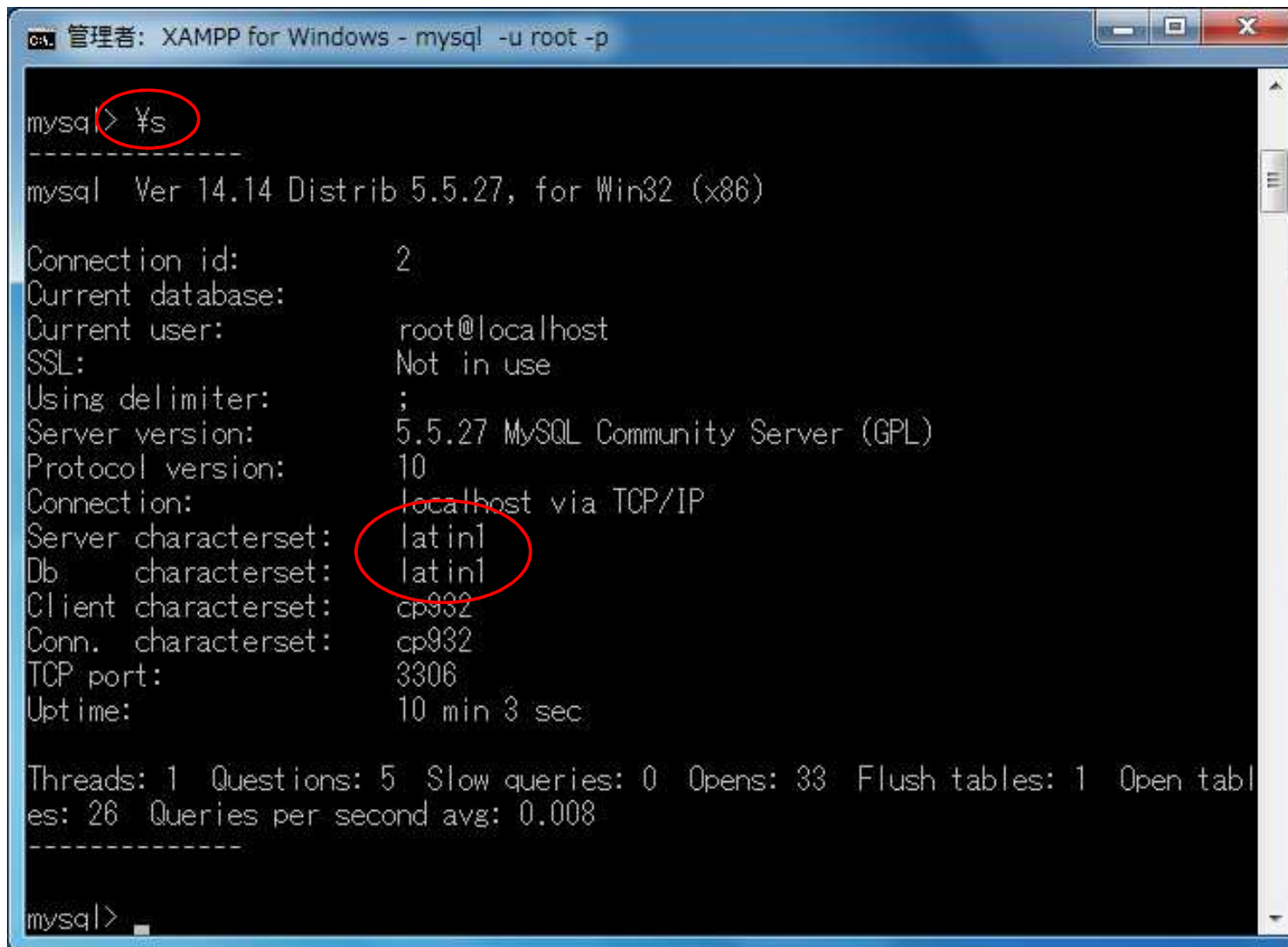
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '?h' for help. Type '?c' to clear the current input statement.

mysql> _
```


MySQLの設定

¥s と入力して現在の設定を確認します。文字コードがlatin1となっています。



```
管理者: XAMPP for Windows - mysql -u root -p

mysql> ¥s
-----
mysql  Ver 14.14 Distrib 5.5.27, for Win32 (x86)

Connection id:          2
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.5.27 MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db characterset:        latin1
Client characterset:    cp932
Conn. characterset:     cp932
TCP port:               3306
Uptime:                 10 min 3 sec

Threads: 1  Questions: 5  Slow queries: 0  Opens: 33  Flush tables: 1  Open tabl
es: 26  Queries per second avg: 0.008
-----

mysql>
```

MySQLの設定

latin1では日本語を正しく扱えないため、文字セットをUTF-8に変更します。

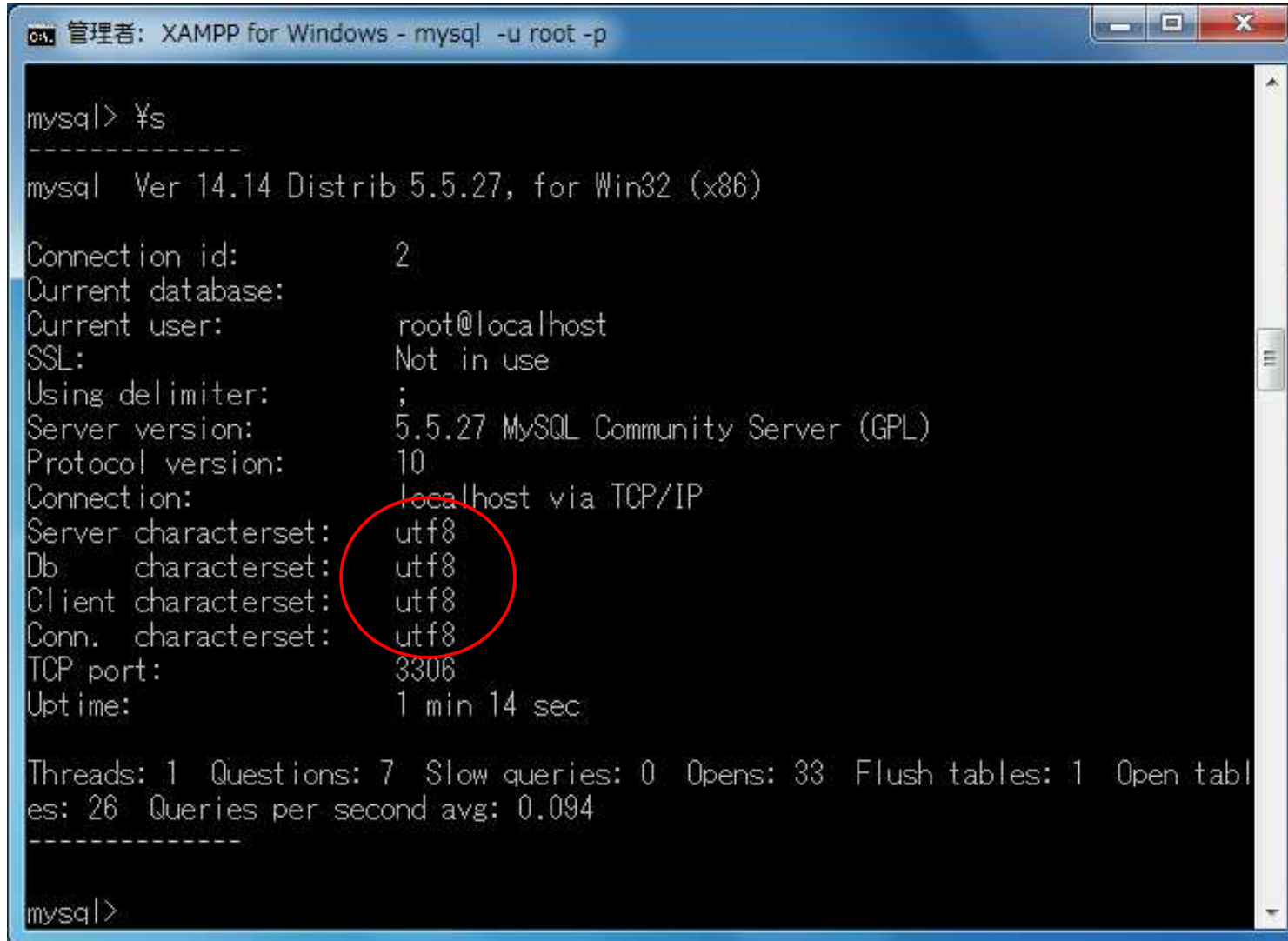
MySQLの設定ファイルは XAMPP_PATH¥mysql¥bin¥my.ini で、[client]セクション及び [mysqld]セクションに以下のように追記します。

```
18 [client] ←
19 # password      = your_password ←
20 port            = 3306 ←
21 socket          = "D:/xampp/mysql/mysql.sock" ←
22 ←
23 # Here follows entries for some specific programs ←
24 default-character-set = utf8 ←
```

```
27 [mysqld] ←
28 port= 3306 ←
29 socket= "D:/xampp/mysql/mysql.sock" ←
30 basedir="D:/xampp/mysql" ←
31 tmpdir="D:/xampp/tmp" ←
32 datadir="D:/xampp/mysql/data" ←
33 pid_file="mysql.pid" ←
34 skip-external-locking ←
35 key_buffer = 16M ←
36 max_allowed_packet = 1M ←
37 table_cache = 64 ←
38 sort_buffer_size = 512K ←
39 net_buffer_length = 8K ←
40 read_buffer_size = 256K ←
41 read_rnd_buffer_size = 512K ←
42 myisam_sort_buffer_size = 8M ←
43 log_error="mysql_error.log" ←
44 #bind-address="127.0.0.1" ←
45 ←
46 character-set-server = utf8 ←
47 skip-character-set-client-handshake ←
```

MySQLの設定

MySQLを再起動(停止して起動)し、Shellも再起動(閉じて再接続)し、`¥s` コマンドで変更が反映されていれば成功です。



```
管理者: XAMPP for Windows - mysql -u root -p

mysql> ¥s
-----
mysql  Ver 14.14 Distrib 5.5.27, for Win32 (x86)

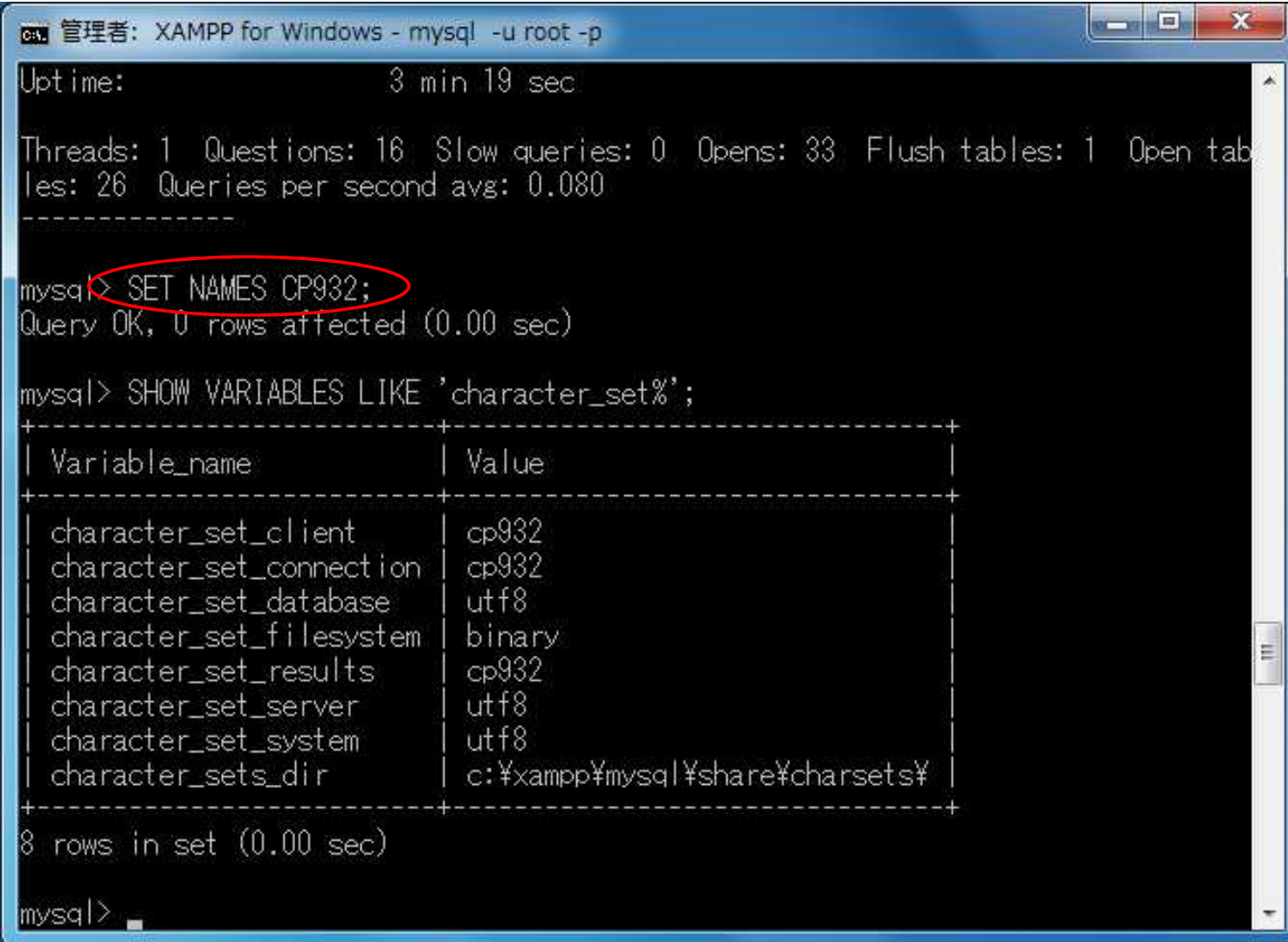
Connection id:          2
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.5.27 MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    utf8
Db      characterset:   utf8
Client characterset:    utf8
Conn.  characterset:    utf8
TCP port:               3306
Uptime:                 1 min 14 sec

Threads: 1  Questions: 7  Slow queries: 0  Opens: 33  Flush tables: 1  Open tabl
es: 26  Queries per second avg: 0.094
-----

mysql>
```

MySQLの設定

WindowsのコマンドプロンプトはShift_JIS(より正確にはWindows-31JまたはCP932)という文字コードを使用しているため正しく日本語が扱えるよう設定をする必要があります。



The screenshot shows a Windows command prompt window titled "管理者: XAMPP for Windows - mysql -u root -p". The prompt is at the mysql> level. The output shows the MySQL status: Uptime: 3 min 19 sec, Threads: 1, Questions: 16, Slow queries: 0, Opens: 33, Flush tables: 1, Open tables: 26, Queries per second avg: 0.080. The command SET NAMES CP932; is entered and circled in red. The output is "Query OK, 0 rows affected (0.00 sec)". The command SHOW VARIABLES LIKE 'character_set%' is entered, and the output is a table with 8 rows and 2 columns: Variable_name and Value.

```
管理者: XAMPP for Windows - mysql -u root -p

Uptime:                 3 min 19 sec

Threads: 1  Questions: 16  Slow queries: 0  Opens: 33  Flush tables: 1  Open tables: 26  Queries per second avg: 0.080
-----
mysql> SET NAMES CP932;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE 'character_set%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | cp932 |
| character_set_connection | cp932 |
| character_set_database | utf8 |
| character_set_filesystem | binary |
| character_set_results | cp932 |
| character_set_server | utf8 |
| character_set_system | utf8 |
| character_sets_dir | c:\xampp\mysql\share\charsets\ |
+-----+-----+
8 rows in set (0.00 sec)

mysql>
```

データ操作言語(DML)

SELECT文

SELECT * FROM usertable
 選択する 全て から テーブル名



userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮

SELECT username FROM usertable
 選択する 名前を から テーブル名



userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮

SELECT username FROM usertable
 選択する 名前を から テーブル名



WHERE userid = 1
 ただし useridが『1』のみ

複数のカラム名を指定する場合
カンマで区切ることが可能

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮

データ操作言語(DML)

INSERT文

```
INSERT INTO usertable  
VALUES('10', '伊藤', '01011112222')
```

テーブル名の後にカラムを指定しない場合
全てのカラムに対応するデータを格納する

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮
10	伊藤	01011112222

```
INSERT INTO usertable(userid, username)  
VALUES('20', '松本')
```

テーブル名の後にカラムを指定した場合
VALUES句の後ろには対応するデータのみ

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮
20	松本	

データ操作言語(DML)

UPDATE文

```
UPDATE usertable SET tel = '111122223'
WHERE userid = 10;
```

WHERE句で指定しない場合は変更する
カラムを全件更新するので注意

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮
10	伊藤	111122223

DELETE文

```
DELETE FROM usertable
WHERE userid = 10;
```

WHERE句で指定しない場合は全件を
削除するので注意が必要

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
10	伊藤	0101112222

データ操作言語(DML)

WHERE 欲しい情報の条件を指定する

```
SELECT * FROM usertable WHERE userid= '111'
```

AND OR 条件を複数指定する場合に使用する

```
SELECT * FROM usertable WHERE userid= '111' AND username= 'alice'
```

```
SELECT usertable WHERE userid= '111' OR username= 'bob'
```

LIKE 抽出条件に部分一致や前方一致を行いたい場合に使用する

```
SELECT * FROM usertable WHERE userid LIKE '%1%'
```

IN 列の値が複数の候補のいずれかと一致する場合の抽出条件

```
SELECT * FROM usertable WHERE userid IN ( '111', '222' )
```

BETWEEN 列の値がある範囲に存在する場合の抽出条件

```
SELECT * FROM usertable WHERE userid BETWEEN '333' AND '666'
```

DISTINCT 重複しているデータを1つにまとめて表示する

```
SELECT DISTINCT username FROM usertable
```

COUNT レコードの合計数を表示する

```
SELECT COUNT(*) FROM usertable
```

SUM 合計値を求める

GROUP BY 集計関数をSELECTで指定した場合に使用する

```
SELECT SUM(userid), username FROM usertable
```

```
GROUP BY username
```

ORDER BY 問合せ結果を並び替えする(昇順)

```
SELECT * FROM usertable ORDER BY userid
```

データ操作言語(DML)

結合(JOIN)

deptテーブル

deptno	dept_name
10	engineer
20	sales

empテーブル

empno	emp_name	deptno
1	alice	10
2	bob	10
3	chris	20

```
SELECT e.empno, e.emp_name, d.dept_name FROM emp e
JOIN dept d ON e.deptno=d.deptno
```

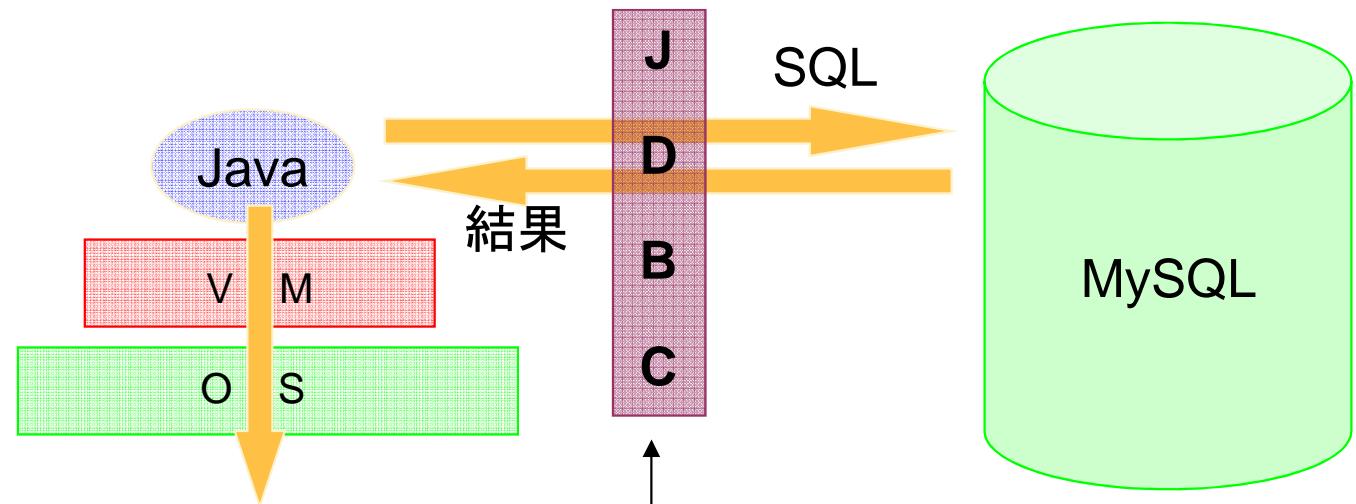
```
SELECT e.empno, e.emp_name, d.dept_name FROM emp e, dept d
WHERE e.deptno=d.deptno
```

empno	emp_name	dept_name
1	alice	engineer
2	bob	engineer
3	chris	sales

結果

JavaからDB操作

これまでは、データベースについてお話してきましたが、実際にはJavaのプログラム経由でアクセスする必要があります。そのためには、新しくJDBCドライバが必要になります。



JDBCドライバは各DBによって違う。
使用するDBに合わせて取得する必要がある。

userid	username	tel
1	田中	0311112222
2	鈴木	0571112222
⋮	⋮	⋮

JavaからDB操作

DbAccess.java

```
import java.sql.*;

public class DbAccess {
    public static void main(String[] args) {
        Connection conn = null;

        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(
                "jdbc:mysql://localhost/axiz","root","");

            Statement stmt = conn.createStatement();
            String sql = "SELECT * FROM users";
            ResultSet rs = stmt.executeQuery(sql);

            while (rs.next()) {
                int id = rs.getInt("id");
                String name = rs.getString("name");
                String tel = rs.getString("tel");

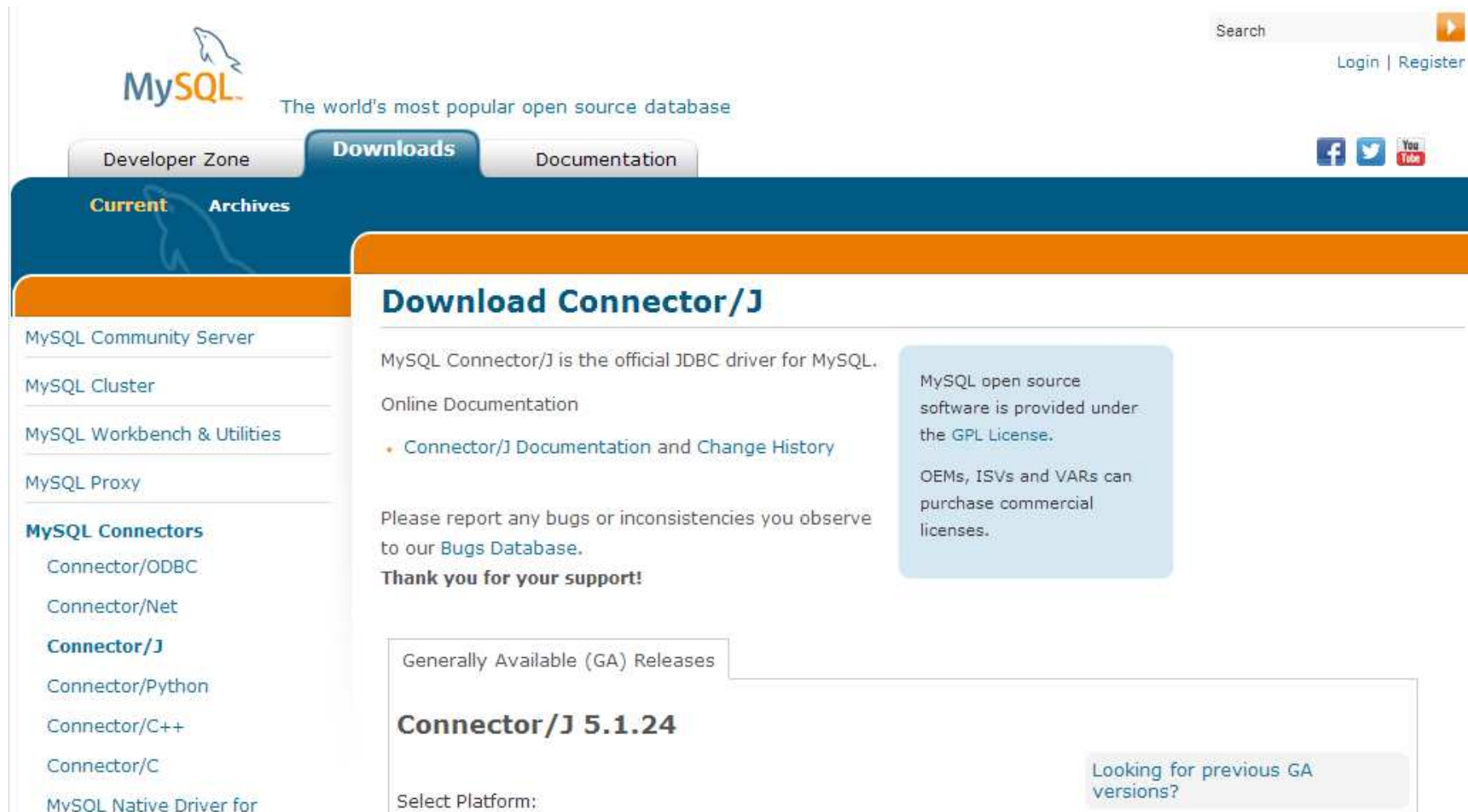
                System.out.println(id + "¥t" + name + "¥t" + tel);
            }
        }
    }
}
```

```
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (stmt != null) {
            try {
                stmt.close();
            } catch (SQLException e) {}
        }
        if (conn != null) {
            try {
                conn.close();
            } catch (SQLException e) {}
        }
    }
}
```

rs.next()・・・次のレコードがあるかどうかを返す

	id	name	tel
<i>rs.next()</i>	1	田中	0311112222
<i>rs.next()</i>	2	鈴木	0571112222
<i>rs.next()</i>			

インターネットで「mysql jdbc」と検索してみてください。
JDBC Driver(Connector/J)のページがヒットします。



The screenshot shows the MySQL website's 'Downloads' section for the 'Connector/J' driver. The page has a blue header with the MySQL logo and tagline 'The world's most popular open source database'. Navigation tabs include 'Developer Zone', 'Downloads' (selected), and 'Documentation'. A sidebar on the left lists various MySQL products, with 'MySQL Connectors' expanded to show 'Connector/J' as the selected option. The main content area is titled 'Download Connector/J' and contains the following text: 'MySQL Connector/J is the official JDBC driver for MySQL.', 'Online Documentation', a link to 'Connector/J Documentation and Change History', and a request to report bugs to the 'Bugs Database'. A blue box on the right states: 'MySQL open source software is provided under the GPL License. OEMs, ISVs and VARs can purchase commercial licenses.' Below this, a section for 'Generally Available (GA) Releases' highlights 'Connector/J 5.1.24' and includes a 'Select Platform:' dropdown. A button at the bottom right says 'Looking for previous GA versions?'.

MySQL The world's most popular open source database

Search Login | Register

Developer Zone Downloads Documentation

Current Archives

MySQL Community Server

MySQL Cluster

MySQL Workbench & Utilities

MySQL Proxy

MySQL Connectors

- Connector/ODBC
- Connector/Net
- Connector/J**
- Connector/Python
- Connector/C++
- Connector/C
- MySQL Native Driver for

Download Connector/J

MySQL Connector/J is the official JDBC driver for MySQL.

Online Documentation

- Connector/J Documentation and Change History

Please report any bugs or inconsistencies you observe to our [Bugs Database](#).

Thank you for your support!

MySQL open source software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

Generally Available (GA) Releases

Connector/J 5.1.24

Select Platform:

Looking for previous GA versions?

ZIP ArchiveのDownloadをクリックします。

Generally Available (GA) Releases

Connector/J 5.1.24

Select Platform:

Platform Independent

[Looking for previous GA versions?](#)

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-5.1.24.tar.gz)	5.1.24	3.5M	Download
MD5: cba8fbf00025e5f24086840c8bd60547 Signature			
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-5.1.24.zip)	5.1.24	3.8M	Download
MD5: f934df6a7c106eb8160d2e570ff2c853 Signature			

ユーザアカウントを持っていない、または登録しない場合は No thanks… をクリックします。

Begin Your Download - mysql-connector-java-5.1.24.zip

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

Login »

using my Oracle Web account

Sign Up »

for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download

JDBC

ダウンロードできたzipファイルを解凍します。

生成されたフォルダにmysql-connector-java-5.1.24-bin.jarというファイルが見つかります。これがJDBCドライバの本体になります。

このjarファイルを XAMPP_PATH¥mysql¥lib にコピーします。

名前	更新日時	種類	サイズ
docs	2013/04/06 17:53	ファイル フォル...	
src	2013/04/06 17:53	ファイル フォル...	
build.xml	2013/03/11 7:20	XML ドキュメント	45 KB
CHANGES	2013/03/11 7:20	ファイル	211 KB
COPYING	2013/03/11 7:20	ファイル	18 KB
mysql-connector-java-5.1.24-bin.jar	2013/03/11 7:20	Executable Jar ...	827 KB
README	2013/03/11 7:20	ファイル	63 KB
README.txt	2013/03/11 7:20	TXT ファイル	65 KB

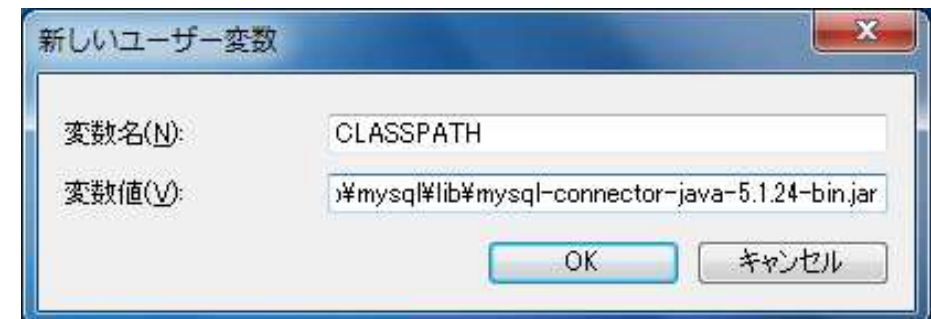
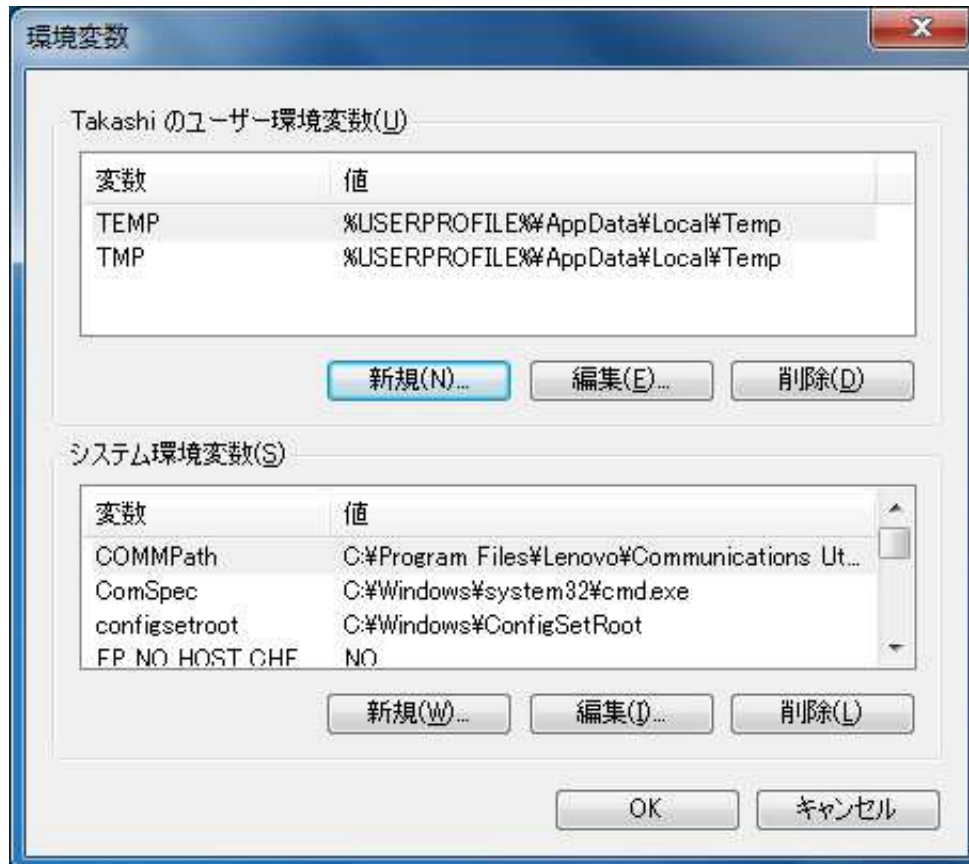
環境変数の設定画面を開き、新しく

変数名: CLASSPATH

変数値: .;C:¥xampp¥mysql¥lib¥mysql-connector-java-5.1.24-bin.jar

を追加します。

(変数値のパスは出来るだけエクスプローラのアドレスなどをコピーし、手入力は控えましょう)



正規化

非正規形

ある売上传票は、次のように1枚の伝票に3つの商品を連記することができると思います。

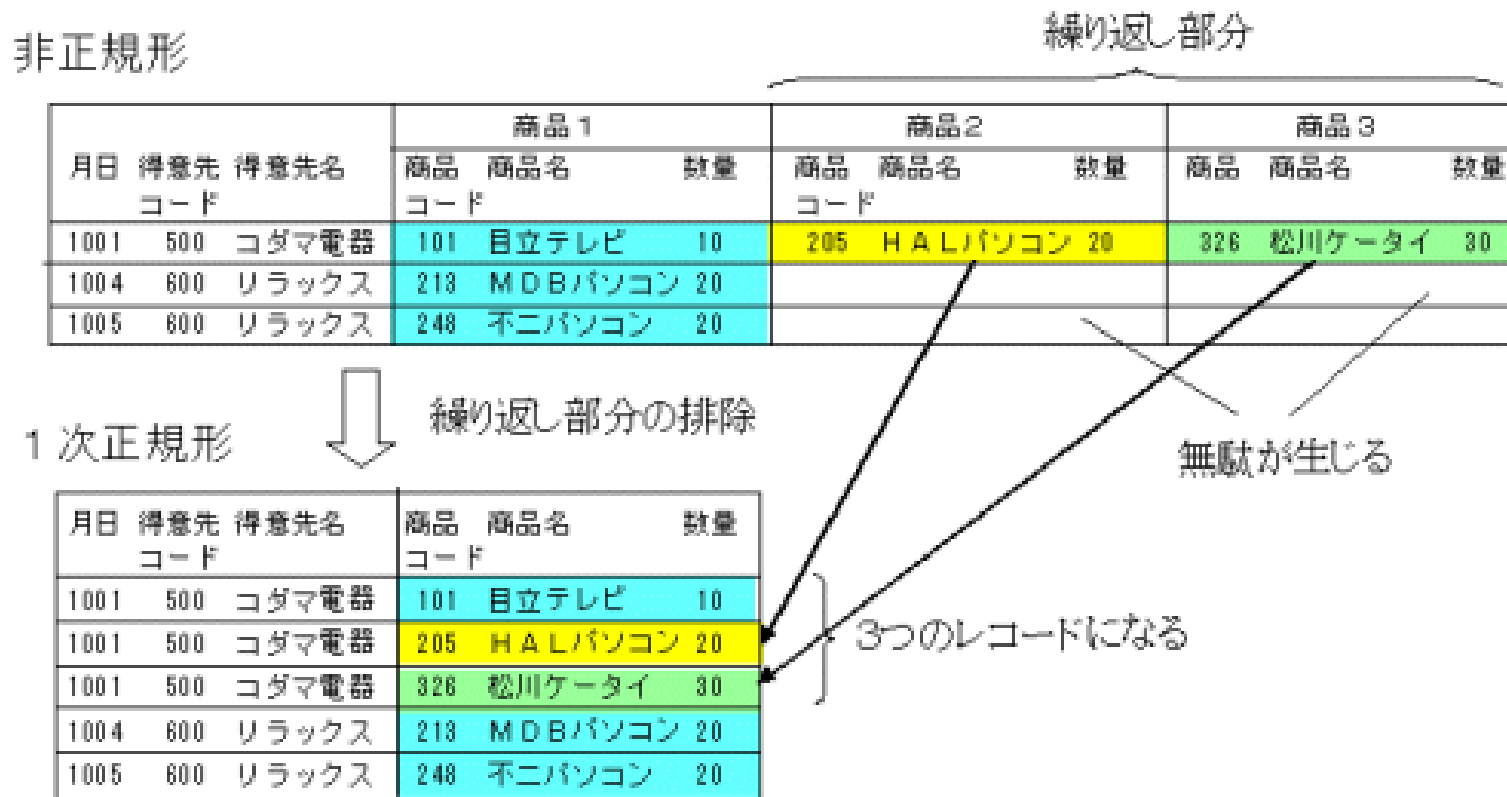
売上传票		
売上日: ××年10月01日		
得意先コード: 500		
得意先名: コダマ電器株式会社		
商品コード	商品名	数量
101	目立テレビ	10
205	HALパソコン	20
326	松川ケータイ	10

これをそのままファイルにすると、図の「非正規形」のようになります(ここでは、図を単純にするために、商品区分や単価を省略しています)。伝票に記載された商品数が少ないときは、レコードの右側に空白がならび無駄になってしまいます。

正規化

第1正規化

第1正規化とは、繰り返しの部分を複数のレコードにして、繰り返しを排除する操作です。
第1正規化をした結果を第1正規形といいます。



正規化

主キーを見つける

第1正規形のファイルが次のようであるとします。

売上ファイル＝年月日
 ＋得意先コード＋得意先名
 ＋商品コード＋商品名
 ＋商品区分コード＋商品区分名
 ＋数量＋単価



このとき、年月日、得意先コード、商品コードのすべてが定まると、そのレコードが決まります。そのとき、これらを主キーといいます。

得意先コードが定まると得意先名が定まるというように、ある属性(項目)が定まると他の属性が定まるという関係を関数従属といいます。

そして、数量は年月日、得意先コード、商品コードのすべての主キーによって定まりますが、このように主キー全体に従属することを完全関数従属といい、得意先コードと得意先名の関係のように、主キーの一部に従属することを部分関数従属といいます。

また、商品コードから商品区分コードがわかり、それにより商品区分名がわかるというように、間接的に従属することを推移関数従属といいます。

正規化

第2正規化

第2正規化とは、部分関数従属する項目を分離することです。その結果のファイルを第2正規形といいます。主キーは年月日、得意先コード、商品コードですが、他の項目とは次の関係があります。

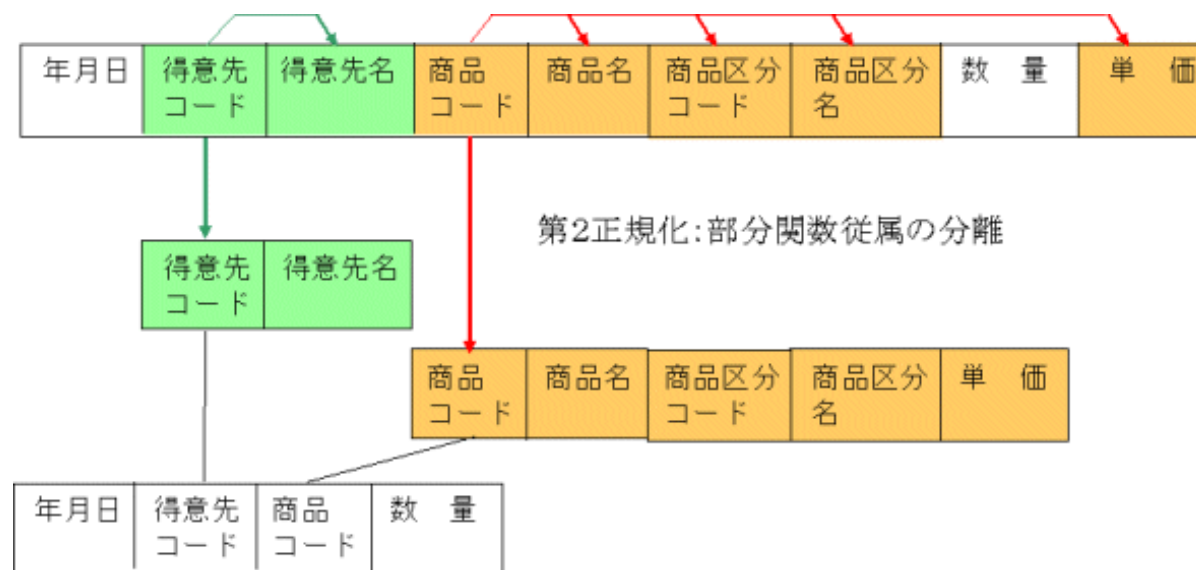
- ・年月日が決まっても、決まる項目はない。
- ・得意先コードが決まれば得意先名が決まる。
- ・商品コードが決まれば商品名、商品区分コード、商品区分名、単価が決まる。
(単価はどの得意先でも同一とする)
- ・数量は、すべての主キーが決まらなければ決まらない(完全関数従属)。

得意先マスタ＝得意先コード＋得意先名

商品マスタ＝商品コード＋商品名＋
商品区分コード＋商品区分名＋
単価

が分離されます。残った項目と分離したファイルのユニークキーからなるファイルは、

売上ファイル＝年月日＋
得意先コード＋
商品コード＋数量



となります(ファイル名は任意)。

正規化

第3正規化

第3正規化は、主キー以外のキーに関数従属する項目を分離します。その結果を第3正規形といいます。得意先マスタは、得意先名は主キーである得意先コードに從属しているから完全関数従属であり、売上ファイルでは、数量がすべての主キーに完全関数従属しているから、これ以上は分割できません。ここでは、商品マスタが対象になります。

商品マスタ＝商品コード＋商品名＋商品区分コード＋商品区分名＋単価

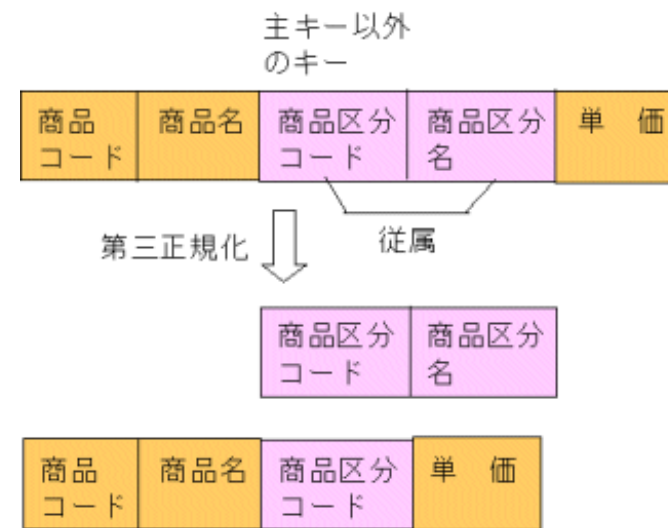
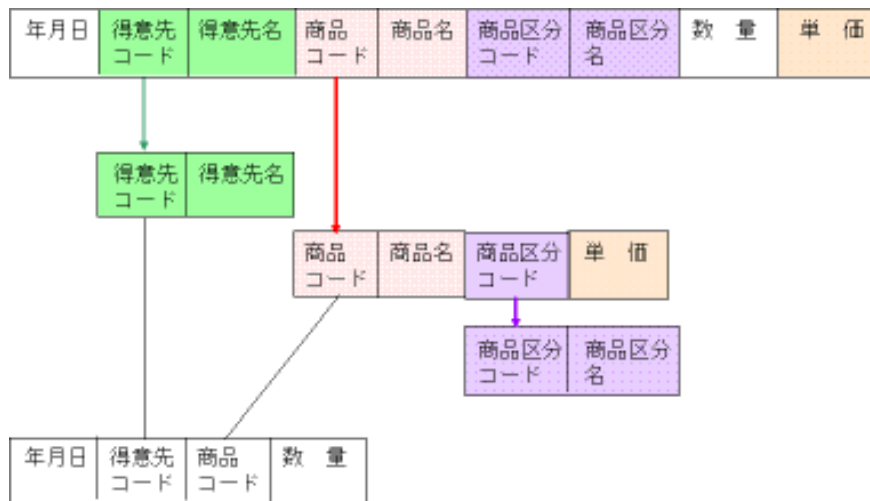
において、主キー以外のキーとは商品区分コードであり、商品区分コードが決まれば商品区分名が決まるので、

商品区分マスタ＝商品区分コード＋商品区分名

が分離できます。それで残った項目（分離ファイルの主キーを含む）で、

商品マスタ＝商品コード＋商品名＋商品区分コード＋単価

の2ファイルになります。



すなわち、第3正規形のファイル群は、左の4ファイルになります。

売上ファイル＝年月日＋得意先コード＋商品コード＋数量
 得意先マスタ＝得意先コード＋得意先名
 商品マスタ＝商品コード＋商品名＋商品区分コード＋単価
 商品区分マスタ＝商品区分コード＋商品区分名

正規化

なお、データの内容は次のようになります。

商品表			
商品コード	商品名	商品区分コード	単価
101	目立テレビ	10	40
122	アニーテレビ	10	50
205	HALパソコン	20	200
213	MDB パソコン	20	180
248	不二パソコン	20	150
326	松川ケータイ	30	10
352	目立ケータイ	30	8
386	MDB ケータイ	30	8

商品区分表	
商品区分コード	商品区分名
10	テレビ
20	パソコン
30	ケータイ

得意先表	
得意先コード	得意先名
500	コダマ電器
600	リラックス
700	淀川カメラ

売上表			
月日	得意先コード	商品コード	数量
1001	500	101	10
1001	500	205	20
1001	500	326	10
1004	600	213	30
1005	600	248	10
1006	700	205	20
1007	700	213	30
1008	500	101	30
1009	500	205	20
1010	600	352	10
1011	600	386	30
1012	600	326	20
1013	700	352	20
1014	700	386	10
1015	500	101	30
1016	500	122	10
1017	600	205	10
1018	700	122	20
1019	700	213	10
1020	500	205	20