



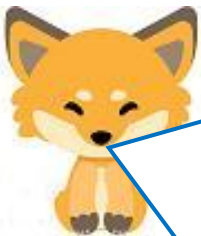
Episode 18 (Sales System II)



Tanuki, when dealing with databases, **you have to design the database well** before you learn to operate the database server. A database is designed to handle hundreds of thousands of data, so if you make a mistake in the initial design, it will be difficult to add or change items later. In the worst case, the database design will have to be redone. In that case, you will be in a situation where you will wonder what to do with the data you have already accumulated.



How difficult is it to design a database?



A full-scale study would take one book on databases alone. If you are interested, you can deepen your study on your own. It is my role to create an opening for you to study. I think this is a very important role.

The MySQL database is a relational database. This means that it is a relational database in which multiple tables in tabular form (like an Excel table) are interconnected by key items.

A table consists of records (data for one case), fields (items), and item attributes (character type and numeric type). The database design used in this case is shown on the next page.

Note that MySQL databases have a hierarchical directory structure and cannot be easily copied and moved.

Relational Database Design

The first thing that must be done is to create a database on the database server(MySQL). The database name, table name, field (item) name, and attributes should be as follows.

Database Name:cdfox

Table Name:cdshohin

Product code (primary key)	item name	scode	Attribute	4 digit	half-width character
CD Name	item name	sname	Attribute	100 digit	Japanese (language)
Jacket image data	item name	jaketto	Attribute	100 digit	half-width character
musical data	item name	music	Attribute	100 digit	half-width character
Inventory	item name	zaiko	Attribute	5 digit	numerical value
Price	item name	kakaku	Attribute	5 digit	numerical value

※By creating a database of image and music file names, it is possible to handle files in any format.

For example, X-ray images can be handled.

Table Name:cduser

Customer number (primary key)	item name	ucode	Attribute	4 digit	half-width character
Customer Name	item name	uname	Attribute	100 digit	Japanese (language)
Purchase item number (external key)	item name	rireki	Attribute	4 digit	half-width character
Purchase Quantity	item name	kazu	Attribute	5 digit	numerical value

The rireki in the user table can be linked to the scode in the shohin table to call a record in the shohin table. In this case, rireki is called a external key.



Is the table above the database design? Is the database name cdfox or is it cdraccoon in the case of Ira? The two tables in the database are cdshohin (rental CDs, etc.), cduser (customer information), or the fields jaketto and music, where only the file names of still images and music are stored.

OK, OK, I understand.

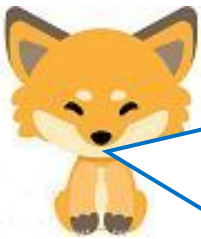
By the way, what's the difference between a primary key and a external key?



The primary key is used to identify a single piece of data (record). So, basically, they should not be used in duplicate. External keys are keys used to connect tables to tables. However, multiple foreign keys may be combined into a single non-overlapping primary key. So, there are many other things to learn about databases, such as normalization.

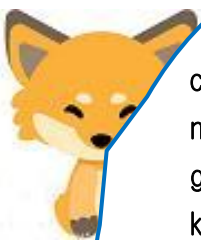


Kitsune, studying the database is going to take some time, so I'll do that when I have more time, what's next?



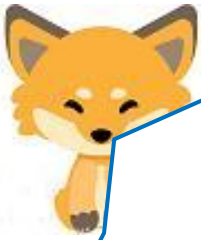
Yes, we would have to create the cdshohin data according to the database design. The basic idea is to input one case at a time into a table, but if there is a large amount of existing data, such as patient records in a hospital, it would take years to open and run a business if we had to input each case one at a time. In this case, let's say that the existing data exists as a text file, and we want to fill it into the corresponding table. cduser is a table to record customer data, so we can leave it empty for this case.

Here is a sample text file to be created.



cdshohin.txt on the right is the name of the text file, created by gedit. A001" is the primary key. k1.jpg" is the image file name and "m1.mid" is the music file name for viewing. 10 and 200 are the number of rental CDs in stock and the rental fee. This is one record. Save the file in UTF8 format in /home/fox.

```
cdshohin.txt ( /home/fox ) - gedit
ファイル(E) 編集(E) 表示(V) 検索(S) ツール(I) ドキュ
新規 開く 保存 印刷... 元に戻す やり直し 切り
cdshohin.txt x
A001, サザエさん, k1. jpg, m1. mid, 10, 200
A002, 新世界, k2. jpg, m2. mid, 5, 300
B001, 運命, k3. gif, m3. wav, 20, 100
B002, 白鳥, k4. gif, m4. wav, 15, 250
C001, ガンダーラ, k5. bmp, m5. mld, 30, 320
C002, 学生街の喫茶店, k6. bmp, m6. mld, 25, 150
D001, イチゴ白書, k7. tif, m7. mid, 50, 400
D002, 第九, k8. png, m8. wav, 45, 350
```



Tanuki, do you remember the connection function to MySQL in line 9 of rtest02.php on page 115 of episode 17? This function has the following meaning

```
mysqli_connect("hostname", "allowed-connection-username",  
"user's-password", "allowed-database-name");  
$abc = mysqli_connect("localhost", "ftpuser", "ftpuser", "cdfox");
```

The hostname can be an online domain name or an IP address, but since the connection between PHP and MySQL is made inside the server, localhost is recommended for safety. Next, the username is ftpuser and the password is ftpuser, so we have to add this as an allowed login user on CentOS7 as shown. I present it in the figure below. Also, remember that the database name is cdfox, **because if one of them is different, the connection between PHP and MySQL will fail. This is the most important part.**

```
root@server25:~  
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)  
[root@server25 ~]# useradd ftpuser  
[root@server25 ~]# passwd ftpuser  
Changing password for user ftpuser  
New UNIX password:  
Retype new UNIX password:  
passwd: all authentication tokens updated successfully.  
[root@server25 ~]#
```



Kitsune, adding a user must be done as root or you will get an error. As I recall, to become root, you have to type \$ su – and then enter fox's password, which will change the prompt from \$ to the root privileged user #.



Tanuki, you're right!
Now it's time to operate MySQL. **First, we need to log in to MySQL.** The server is to be used with a login, you know.

Login to database server

```
# mysql -u root -p
```

Enter password: Enter root password



You've logged in! What's up, I'm getting another prompt. What's next?



MariaDB [(none)]> is the MySQL prompt. The next step is to create a database (cdfox), which is a large vessel.

Creation of database (cdfox)

```
MariaDB [(none)]>create database cdfox;
```

```
MariaDB [(none)]>show databases ;
```

```
MariaDB [(none)]>use cdfox ;
```



Create is what is known as DDL (Data Definition Language)
Create Drop Alter (add).

It is distinguished from the four SQL statements Select, Insert, Update, and Delete.

The other two commands, show and use, display all existing databases and specify the database to be used from among the multiple databases. The next step is to create two tables in cdfox.

Create tables (cdshohin, cduser)

```
MariaDB [(cdfox)]>create table cdshohin(scode varchar(4), sname varchar(100),  
jaketto varchar(100), music varchar(100), zaiko int,kakaku int);
```

```
MariaDB [(cdfox)]>create table cduser(ucode varchar(4), uname varchar(100),  
rireki varchar(4),kazu int);
```




Kitsune! I remember when you created the table as shown on the right, you were always going to make sure to check it with [show tables;], right? It's done!

```
MariaDB [ cdfox ]> show tables;
+-----+
| Tables_in_cdfox |
+-----+
| cdshohin        |
| cduser          |
+-----+
2 rows in set (0.00 sec)
```



The next operation is to pour the "cdshohin.txt" that you made on page 121 into cdshohin. If there is a large amount of data, the computer will stop for a moment, so wait for that moment.

Import into database

```
MariaDB [(cdfox)]> load data infile "/home/fox/cdshohin.txt" into table cdshohin
fields terminated by "," ;
```

confirmation

```
MariaDB [(cdfox)]> select * from cdshohin ;
```

scode	sname	jaketto	music	zaiko	kakaku
A001	サザエさん	k1.jpg	m1.mid	10	200
A002	新世界	k2.jpg	m2.mid	5	300
B001	運命	k3.gif	m3.wav	20	100
B002	白鳥	k4.gif	m4.wav	15	250
C001	ガンダーラ	k5.bmp	m5.mld	30	320
C002	学生街の喫茶店	k6.bmp	m6.mld	25	150
D001	イチゴ白書	k7.tif	m7.mid	50	400
D002	第九	k8.png	m8.wav	45	350

8 rows in set (0.00 sec)



Well, this is easy!

The "terminated by "," " part is a comma-separated list of items. Or " " if it's separated by spaces.

Fox, is that it?



Tanuki, there's a little more to it.

It's the setting of the login user to MySQL and the permission to use the "cdfox" database for that user. This is a very important part of security. If they get through, all the customer data will be stolen. In the case of MySQL, the permission settings are stored in a database called mysql in the form of records. If you delete this database, you will not be able to start the MySQL server itself. First, we will use mysql. Switching databases is done with the "use" command, right?

Switching databases

```
MariaDB [(cdfox)]>use mysql ;
```

Register users with local access to the MySQL server

Format :

```
MariaDB [mysql]> CREATE USER 'username'@'localhost' IDENTIFIED BY 'password' ;
```

Actual input example:

```
MariaDB [mysql]> CREATE USER 'ftpuser'@'localhost' IDENTIFIED BY 'ftpuser' ;
```

Registration of a user with full privileged access to the MySQL server database (cdfox).

However, it is invalid only after the database (cdfox) and necessary tables have been created.

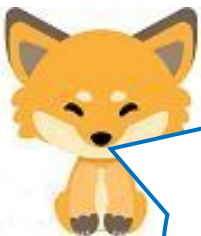
```
MariaDB [mysql]> GRANT ALL PRIVILEGES ON cdfox.* TO 'ftpuser'@'localhost' ;
```

Format :

```
MariaDB [mysql]> CREATE USER 'username'@'localhost' IDENTIFIED BY 'password' ;
```

Actual input example:

```
MariaDB [mysql]> CREATE USER 'ftpuser'@'localhost' IDENTIFIED BY 'ftpuser' ;
```



Tanuki, this is the end.

If you execute page 114 of episode 17, you will see the display on the right and Sazae-san's music will play. However, m1.mid requires you to prepare appropriate music by yourself.





It was a long shot, but I now understand what it means to have a database in the background. With this, I can create all kinds of online shopping.

The amount of data flowing over the Internet is also kept to a minimum. 3-tier C/S is amazing.



Tanuki, now that we've crossed the mountain, all that's left is to explain the rest of the SQL statement.

I'll do that in **Episode 19**.

Oh yeah, when you exit the MySQL server, you have to type EXIT.

Translated at DeepL