

## イントロダクション

第10回でモデル化した情報システム（予約システムの利用者用サブシステム）を，第11回～第14回で作成します．以下のようなスケジュールになります．

第11回：データベース，SQLユーザ，初期データの準備

第12回：基本的なインターフェース，空き状況の確認（必須），施設概要の表示（オプション）

第13回：ログインの処理（必須），自分の予約の確認（オプション）

第14回：新規予約（必須），予約のキャンセル（オプション）

今回は，システムを作成する準備を整えます．やり方は一度学習していますが，データベースの作成とSQLユーザの登録を間違うと，後々うまくいかないのので，それらについては，やり方を示します．一方，初期データの準備の方は，課題として実施してもらいます．

---

## データベースの確認

課題7の(2)で作成したデータベースを用います。実質的な内容があれば、テーブル名やカラム名が異なっても、問題ありません。ただ、第12回以降で示すサンプルプログラムでは以下のようなテーブル名とカラム名、データ型であることを前提として話を進めます。

### テーブル名

```
mysql> use db_reservation;
Database changed
mysql> show tables;
+-----+
| Tables_in_db_reservation |
+-----+
| facility                  |
| reservation              |
| user                     |
+-----+
3 rows in set (0.01 sec)
```

### テーブルuserのカラム名と型

```
mysql> describe user;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_id        | varchar(20)   | YES  |     | NULL    |       |
| user_name      | varchar(30)   | YES  |     | NULL    |       |
| address        | varchar(50)   | YES  |     | NULL    |       |
| telephone      | varchar(15)   | YES  |     | NULL    |       |
| mail_address   | varchar(50)   | YES  |     | NULL    |       |
| password       | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)
```

### テーブルfacilityのカラム名と型

```
mysql> describe facility;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| facility_name  | varchar(30)   | YES  |     | NULL    |       |
| open_time      | time          | YES  |     | NULL    |       |
| close_time     | time          | YES  |     | NULL    |       |
| explanation    | text          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

## テーブルreservationのカラム名と型

```
mysql> describe reservation;
```

Field	Type	Null	Key	Default	Extra
reservation_id	int(11)	NO	PRI	NULL	auto_increment
date	datetime	YES		NULL	
start_time	time	YES		NULL	
end_time	time	YES		NULL	
user_id	varchar(20)	YES		NULL	
facility_name	varchar(30)	YES		NULL	

```
6 rows in set (0.01 sec)
```

## データベースの再作成

データベースのテーブル名やカラム名などをあわせておきたい方は、ここでデータベースを作り直しておきましょう。再作成をしない方はこのページは読み飛ばして結構です。再作成を間違えると、後々大変なので、やり方を示しておきます。

まず、せっかく作ったデータベースですが、以下のようなコマンドで削除してしまいます。

```
mysql> drop database db_reservation;
```

次に、データベースを作成し直します。以下のコマンドです。

```
mysql> create database db_reservation;
```

次に、3つのテーブルを作成します。作成するコマンドを以下に示します。

```
mysql> use db_reservation;
Database changed
mysql> create table user (
  -> user_id varchar(20),
  -> user_name varchar(30),
  -> address varchar(50),
  -> telephone varchar(15),
  -> mail_address varchar(50),
  -> password varchar(20)
  -> );
Query OK, 0 rows affected (0.20 sec)

mysql> create table facility (
  -> facility_name varchar(30),
  -> open_time time,
  -> close_time time,
  -> explanation text
  -> );
Query OK, 0 rows affected (0.52 sec)

mysql> create table reservation (
  -> reservation_id INT AUTO_INCREMENT,
  -> date datetime,
  -> start_time time,
  -> end_time time,
  -> user_id varchar(20),
  -> facility_name varchar(30),
  -> PRIMARY KEY( reservation_id)
  -> );
Query OK, 0 rows affected (0.20 sec)
```

---

## MySQLを操作するためのユーザの登録

一度、SQLモニタをexitで終了した後、もう一度立ち上げ直してから、以下のコマンドを実行します。（前の続きからそのまま実行すると、use db\_reservation が効いているので、以下のコマンドはうまく実行されません。）

```
mysql> GRANT ALL ON db_reservation.* To reservation_user@localhost
-> IDENTIFIED BY 'pass0004';
Query OK, 0 rows affected (0.05 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)
```

reservation\_userが実行できるコマンドはALLを指定していますので、全てのコマンドが実行できます。

---

## 課題11

(1) eclipseで新しいプロジェクトを作成しておきましょう。プロジェクト名はreservation\_systemとして、外部JARファイルを追加で、mysql-connector-javaを追加しておきましょう。

(2)データベースに初期データを追加しましょう。初期データの例については次のページで説明します。以下の方法のいずれかで実施しましょう。課題11Bの方が高い得点になります。

課題11A SQLモニタを使ってINSERTコマンドでデータを追加する。

課題11B 初期データを追加するJavaプログラムを作成し、それを実行して追加する。

(2)についてそれぞれ以下のファイルを作成して提出しなさい。(1)はやっておくこととし、特に何かを提出する必要はありません。

課題11A ワードなどのファイルに、以下を収めて1つのファイルを提出。

- ・ 実行したSQLコマンドの例 (1種類のレコードに対して1例)
- ・ 登録が完了した各テーブルに対してSELECTを実行した結果の画面をキャプチャした図

課題11B プログラムリストのxxxx.javaファイル、登録が完了した各テーブルに対してSELECTを実行した結果の画面をキャプチャした図をワードなどのファイルに収めたものの2つのファイルを提出。

SQLモニタを使ってSELECT文で登録したレコードを表示させると、日本語がきちんと表示されないかもしれませんが、文字化けした状態の画面をキャプチャして提出するのでかまいません。

---

## 初期データ

以降の実習のために、3つのテーブルに初期データを登録します。facilityテーブルについては、第6回で示した施設がありましたから、それらを登録しておきます。userテーブルとreservationテーブルは適当に2つぐらいのレコード登録しておきましょう。

### テーブルfacilityの初期データの例

施設名と開館時刻、閉館時刻は課題で提示されていたので、この通りに入力しましょう。施設概要の説明(explanation)は、それらしいことを自由に記述してかまいません。

facility			
facility_name	open_time	close_time	explanation
小ホール	10:00:00	21:00:00	小ホール 客席数:500 利用可能時間 10時から21時
大会議室1	9:00:00	20:00:00	大会議室1 座席数:80 利用可能時間 9時から20時
大会議室2	9:00:00	20:00:00	大会議室2 座席数:80 利用可能時間 9時から20時
小会議室1	9:00:00	20:00:00	小会議室1 座席数:20 利用可能時間 9時から20時
小会議室2	9:00:00	20:00:00	小会議室2 座席数:20 利用可能時間 9時から20時
小会議室3	9:00:00	20:00:00	小会議室3 座席数:20 利用可能時間 9時から20時
小会議室4	9:00:00	20:00:00	小会議室4 座席数:20 利用可能時間 9時から20時
小会議室5	9:00:00	20:00:00	小会議室5 座席数:20 利用可能時間 9時から20時
小会議室6	9:00:00	20:00:00	小会議室6 座席数:20 利用可能時間 9時から20時

INSERT文で値を指定する際に、time型の値は'時:分:秒'という形式で指定することに注意しましょう。

### テーブルuserの初期データの例

初期データとしてレコードを2以上登録しましょう。以下に例を示しますが、これと同じにならないようにしてください。

user					
user_id	user_name	address	telephone	mail_address	password
watanabe	渡辺博芳	宇都宮市豊郷台1-1	028-627-7111	hiro@ics.teikyo-u.ac.jp	pass0001
teikyo	帝京太郎	東京都板橋区加賀2-11-1	03-3964-1211	teikyo@teikyo-u.ac.jp	pass0002

ユーザ情報は、実在しないものでもかまいません。ユーザIDとパスワードは、後で使います。半角英数字にしておきましょう。

### テーブルreservationの初期データの例

これも初期データとしてレコードを2つ以上登録しましょう。以下の例とは同じにならないように自

分で作成してください。ただし、userに登録されているユーザID、facilityに登録されているfacility\_nameを指定することし、使用の開始時刻と終了時刻はその施設の開館時間にあうようにしましょう。また、2つ以上のレコードを登録する際に、施設と使用時間が重ならないように注意してください。

reservation

date	start_time	end_time	user_id	facility_name
2009-11-15	09:00:00	12:00:00	watanabe	小会議室1
2009-11-15	10:00:00	12:00:00	teikyo	小会議室2

このテーブルでは、reservation\_idというカラムもありますが、INSERT文でデータを挿入すると、自動的に割り当てられるので、ここには示していません。INSERT文では、指定するカラム名を列挙してからVALUESで値を列挙する方法をとることになります。詳しくは第7回を復習しましょう。

---



## 課題11Bの手引き

課題11Bは新規に作成したプロジェクトreservation\_systemのsrcの下にinitializeというパッケージを作成して、その中に新規クラスを作成してプログラムを作りましょう。

今回の学習モジュールで示した通りに、SQLユーザを作成した場合、データベースの接続は、以下のようになります。SQLの接続・準備と切断のその他の部分は第8回のプログラムと同じです。

```
Connection con = DriverManager.getConnection(url,"reservation_user", "pass0004");
```

試しに何度かプログラムを実行することを考えると、INSERTによる挿入の前に、一度、テーブル内のレコードをすべて削除しておく方がよいでしょう。たとえば、userテーブルのレコードをすべて削除するのは、以下のように記述できます。

```
// ユーザテーブルクリア
String sql = "delete from db_reservation.user;";
int rs_int = stmt.executeUpdate(sql);
```

大会議室や小会議室のように、同じようなデータを何度か挿入する場合は、for文を使うことが考えられます。たとえば、大会議室1と大会議室2を登録するfor文は次のようになります。String.valueOf(i)でfor文のカウンタとして使われている変数iを使って、「大会議室1」と「大会議室2」の数字の部分を変えている訳です。

```
// 大会議室
for ( int i=1; i<=2; i++){
    // 設備登録のクエリ
    sql = "insert into db_reservation.facility values ('大会議室";
    sql += String.valueOf(i) + "', '09:00:00', '20:00:00', '大会議室" +
String.valueOf(i)+ " 座席数：80 利用可能時間 9時から20時')";
    // クエリを実行
    rs_int = stmt.executeUpdate(sql);
}
```

登録対象が2つしかない場合は、for文にするメリットはあまり感じられませんが、小会議室の方はfor文にするメリットが感じられると思います。

---