



**G's ACADEMY**  
**TOKYO**

# PHP MySQL

## 更新／削除

授業前にApache,MySQLの起動確認



授業用チューター呼び出しサービス



<http://www.call.jp/room/gsacademy-tokyo>

# 前回の課題発表 ブックマークアプリ

登録、表示、テーブル作成

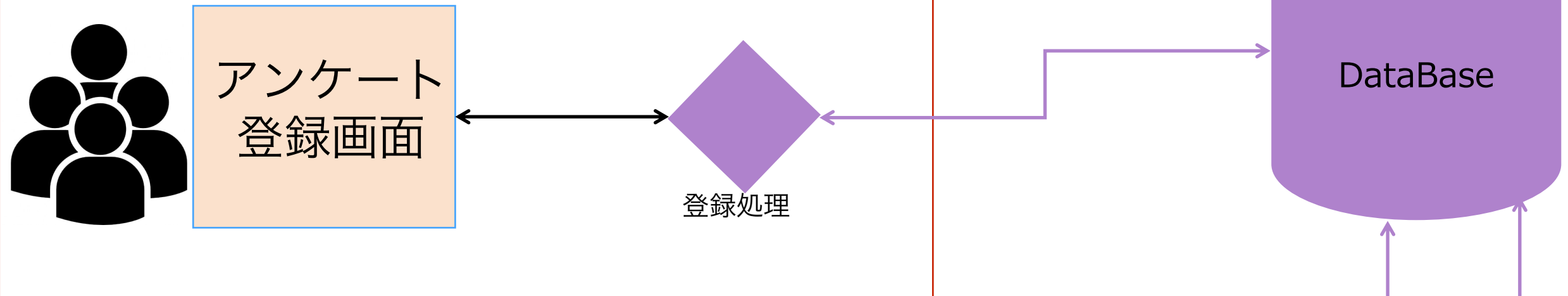
# 本日の授業内容

# アジェンダ

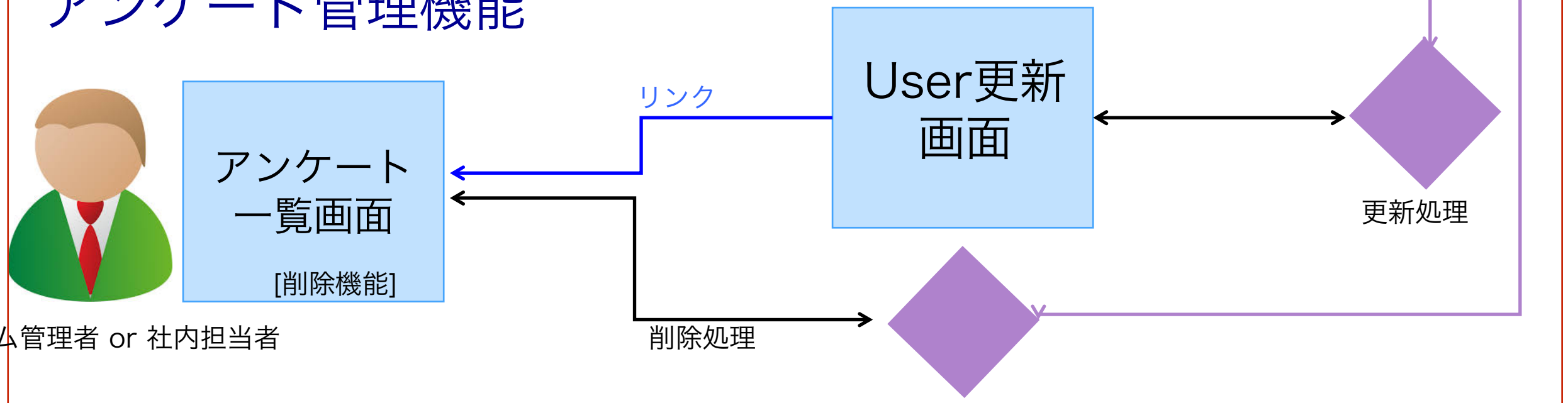
- **PHP&MySQL 更新・削除**
- **演習**
- **次回課題**

# 1. アンケートシステム (1,2回目授業内容)

## 一般公開表示



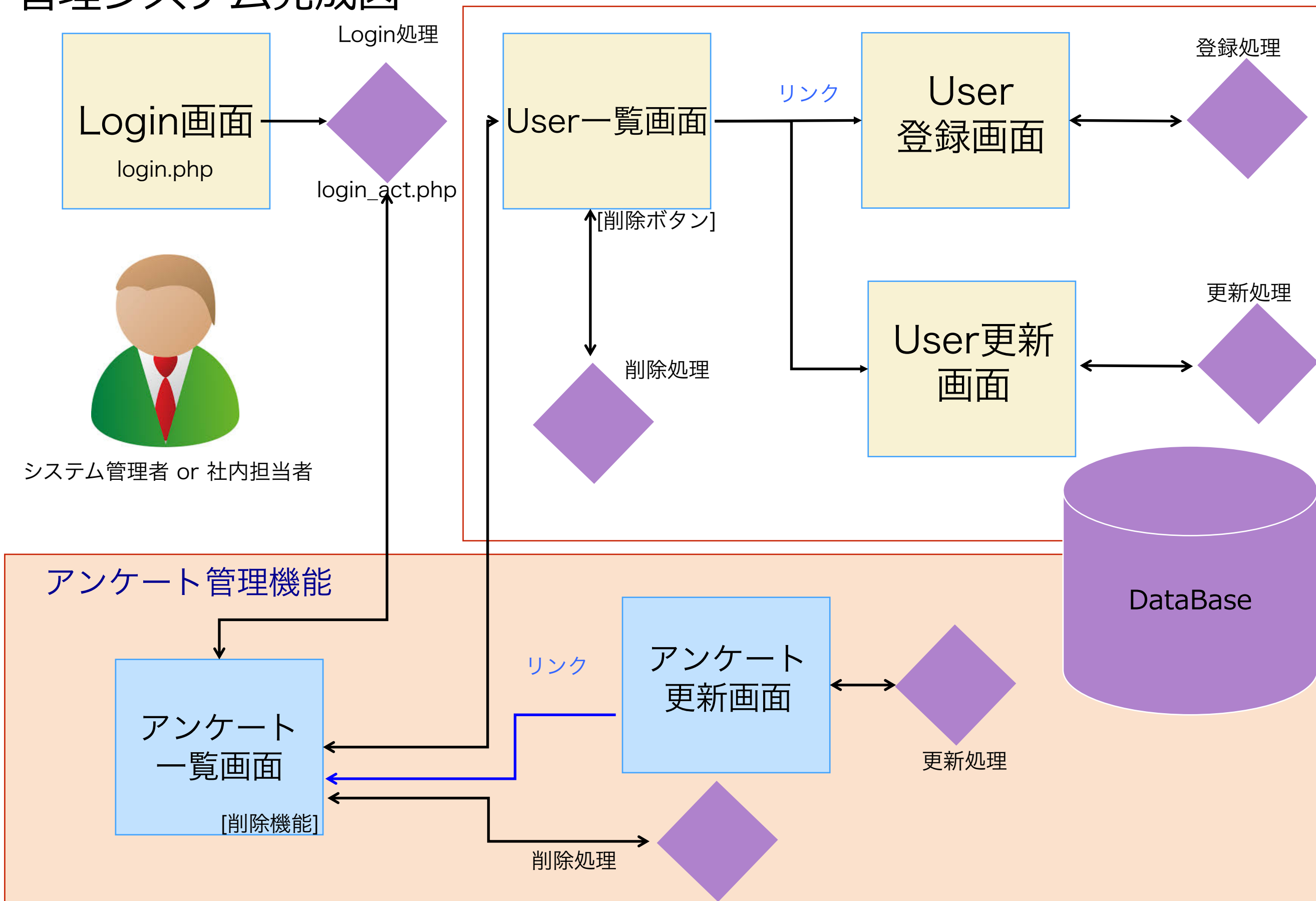
## アンケート管理機能



システム管理者 or 社内担当者

# 管理機能完成図

# 管理システム完成図





# PHPとDB接続

データ更新

# 更新の流れ

## ①リンク作成

select.php

localhost/gs\_code/php/select.php

データ登録

2016-08-09 14:31:55	1	やまざき
2016-08-10 14:53:32	2	ジーズ三郎
2016-08-10 14:54:18	4	ジーズ五郎
2016-08-17 00:00:00	5	ジーズ六郎
2016-08-17 00:00:00	6	ジーズ七郎
2016-08-17 00:00:00	7	ジーズ八郎
2016-08-17 15:16:58	8	テスト 9 9
2016-08-17 15:21:47	9	test100

データ登録

2016-08-09 14:31:55	1	ジーズタロウ
2016-08-10 14:53:32	2	ジーズ三郎
2016-08-10 14:54:18	4	ジーズ五郎
2016-08-17 00:00:00	5	ジーズ六郎
2016-08-17 00:00:00	6	ジーズ七郎

## ②更新ページ作成

detail.php

`$_GET["id"];`で1レコード取得

データ一覧

フリーアンケート

名前:

Email:

## ③更新処理作成

update.php

アンケート更新処理

[detail.php?id=\\*\\*](detail.php?id=**)

<select.php>

## ①リンク作成:「更新ページへのリンク作成」

◇サンプルコード: [select.php](#)

```
$view = "";  
if($status==false) {  
    $error = $stmt->errorInfo(); //Errorがある場合  
    exit("ErrorQuery:".$error[2]); //配列index[2]にエラーコメントあり  
} else {  
    while( $result = $stmt->fetch(PDO::FETCH_ASSOC)){  
        //GETデータ送信リンク作成  
        $view .= '<p>';  
        $view .= '<a href="detail.php?id=' . $result["id"].">';  
        $view .= $result["indate"] . " : " . $result["name"] ;  
        $view .= '</a>';  
        $view .= '</p>';  
    }  
}
```

## ②更新ページ作成: id値をGET→ id値をもとにデータ抽出

◇サンプルコード: [detail.php](#)

```
<?php
```

```
//GET送信されたidを取得 (URLの後ろについてるデータ)
```

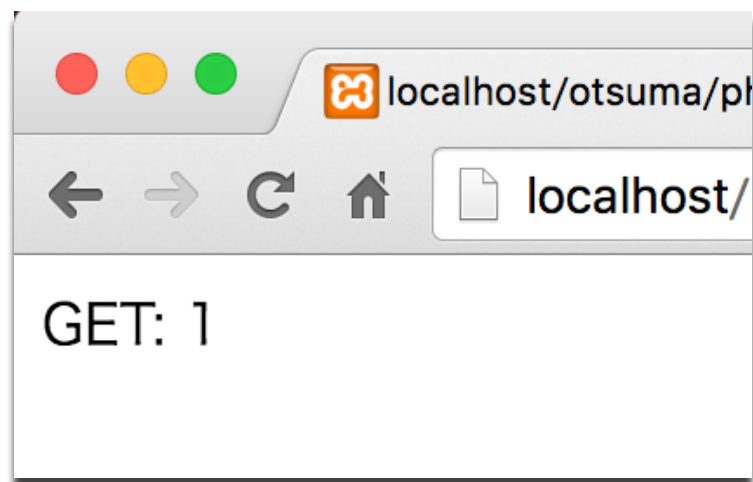
```
$id = $_GET["id"];
```

```
echo "GET: ". $id;
```

```
?>
```

表示結果: detail.php

(注意:必ず [select.php](#) のリンクをクリックして表示！)



ポイント : **\$row** = \$stmt->fetch();

③更新処理作成: データ更新処理 → 処理完了後にデータ一覧ページへ遷移

◇サンプルコード: 更新処理 (update.php)

//データ登録SQL作成

```
$update = $pdo->prepare("UPDATE テーブル名 SET name=:name,  
email=:email, naiyou=:naiyou WHERE id=:id");
```

```
$update ->bindValue(':name', $name, PDO::PARAM_STR);  
$update ->bindValue(':email', $email, PDO::PARAM_STR);  
$update ->bindValue(':naiyou', $naiyou, PDO::PARAM_STR);  
$update ->bindValue(':id', $id, PDO::PARAM_INT);
```

//SQL実行

```
$status = $update->execute();
```

//次のページへリダイレクト

```
header("Location: select.php");
```

# PHPとDB接続

データ削除

# 削除の流れ

select.php

← → ↻ 🏠 ⓘ localhost/gs\_code/php/select.php

データ登録

2016-08-09 14:31:55	1	やまざき	[削除]
2016-08-10 14:53:32	2	ジーズ三郎	[削除]
2016-08-10 14:54:18	4	ジーズ五郎	[削除]
2016-08-17 00:00:00	5	ジーズ六郎	[削除]
2016-08-17 00:00:00	6	ジーズ七郎	[削除]
2016-08-17 00:00:00	7	ジーズ八郎	[削除]
2016-08-17 15:16:58	8	テスト 9 9	[削除]
2016-08-17 15:21:47	9	test100	[削除]

[delete.php?id=\\*\\*](#)

`$_GET["id"];`でデータ取得

delete.php

アンケート削除処理

[select.php](#)

データ登録

2016-08-10 14:53:32	2	ジーズ三郎	
2016-08-10 14:54:18	4	ジーズ五郎	
2016-08-17 00:00:00	5	ジーズ六郎	
2016-08-17 00:00:00	6	ジーズ七郎	

## PHPとデータベースの接続 (リンク作成＞データ削除)

◇サンプルコード：[select.php](#)

// 赤文字のリンクを追加する！！

```
while( $result = $stmt->fetch(PDO::FETCH_ASSOC)){  
    //GETデータ送信リンク作成  
    $view .= '<p>';  
    $view .= '<a href="detail.php?id='.$result["id"].'">';  
    $view .= $result["indate"] . " : ". $result["name"] ;  
    $view .= '</a>';  
    $view .= '  ' ;  
    $view .= '<a href="delete.php?id='.$result["id"].'">';  
    $view .= ' [削除] ' ;  
    $view .= '</a>';  
    $view .= '</p>';  
}
```



## PHPとデータベースの接続 (データ削除)

### ◇サンプルコード：\_削除 (delete.php)

```
//GET値取得
```

```
$id = $_GET["id"];
```

```
//データ登録SQL作成
```

```
$update = $pdo->prepare("DELETE FROM テーブル名 WHERE id=:id");
```

```
//WHERE id=変更するidを指定
```

```
$update->bindValue(':id', $id, PDO::PARAM_INT);
```

```
//SQL実行
```

```
$status = $update->execute();
```

# 課題 1

# 課題テーブル仕様

本をブックマークするDBを造

- **DB名:** `gs_db`
- **Table名:** `gs_bm_table`
- **項目名:**
  1. ユニーク値 (int 12 , PRIMARY, AutoIncrement)
  2. 書籍名 (varChar 64)
  3. 書籍URL (text)
  4. 書籍コメント (text)
  5. 登録日時 (datetime)

前回課題

`index.php`(登録)

`select.php`(更新)

授業で作成したアンケートを模倣して作ること。

フィールド名は  
自分で考えて実際にテーブルを作成しましょう！

# 【今回課題】ブックマークアプリの更新画面&処理作成

## ◇仕様：

- アンケート画面を参考に「ブックマーク”更新”画面」  
bm\_update\_view.phpを作成  
(アンケートのHTMLをコピーして変更使用してもOK)
- bm\_update.php ファイルを作成し「更新処理」を作成。
- 「更新処理」完了後に、自動で bm\_list\_view.php に戻る  
※header関数を利用
- 「削除」も実装。

授業で作成したアンケート  
を模倣して作ること。  
処理ほぼ一緒。

スポーツと一緒に「トレーニング」手と頭を動かす！  
「作る、動作確認」を繰り返すことが習得の早道！

# 課題 2

## ～ブックマークアプリのUSER管理画面を作る～

- ・管理ユーザーを追加 (登録画面、登録処理)
- ・管理ユーザーの一覧表示 (一覧画面、[更新/削除リンク](#))
- ・管理ユーザーを変更 ([更新画面](#)、[更新処理](#))
- ・管理ユーザーを削除 ([削除処理](#))

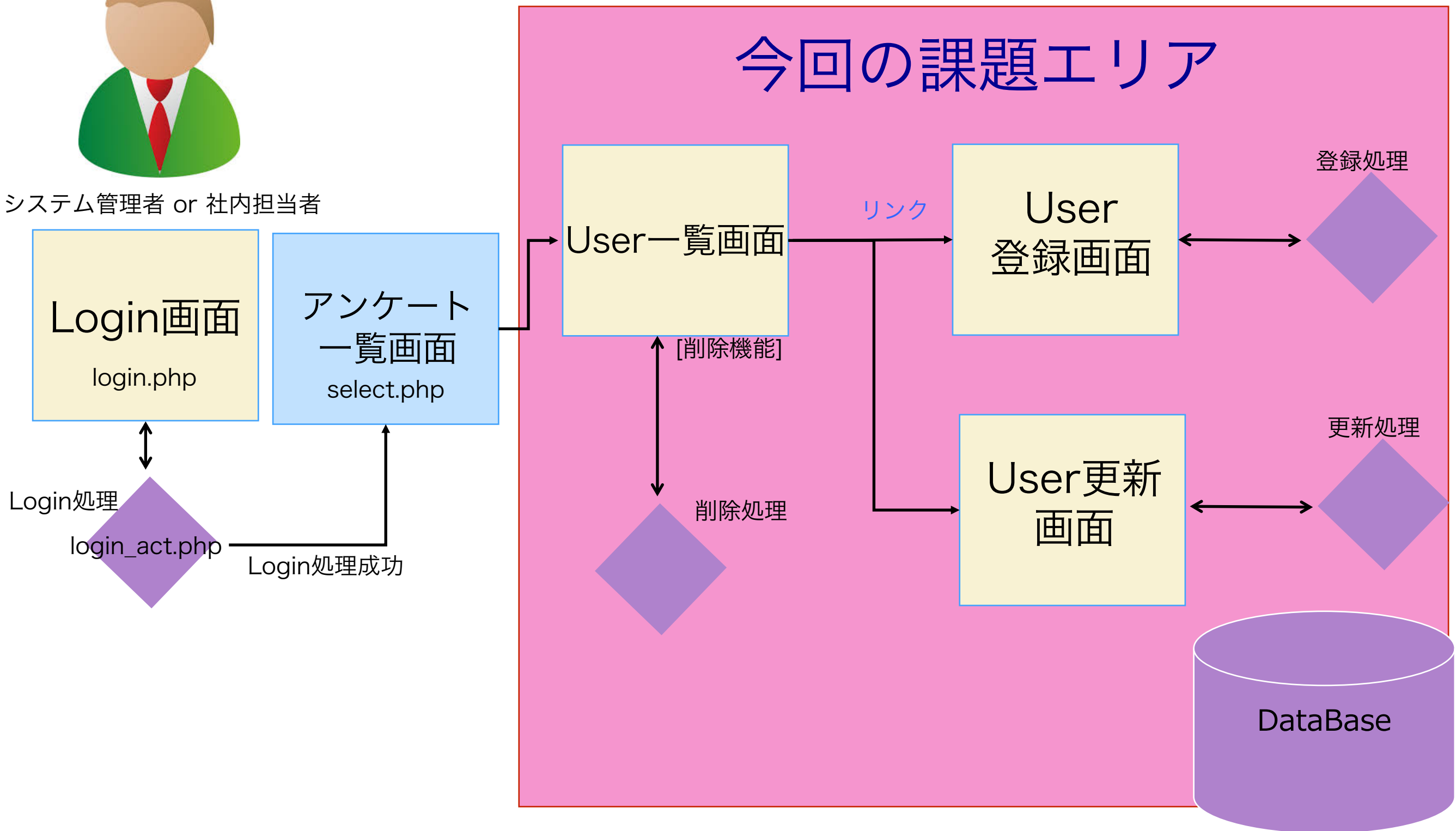
ユーザーテーブルの仕様は20ページを参照。

※認証機能は来週の授業でやります。

## 2. システム管理者機能 (課題)



システム管理者 or 社内担当者



# 管理ユーザー：管理画面

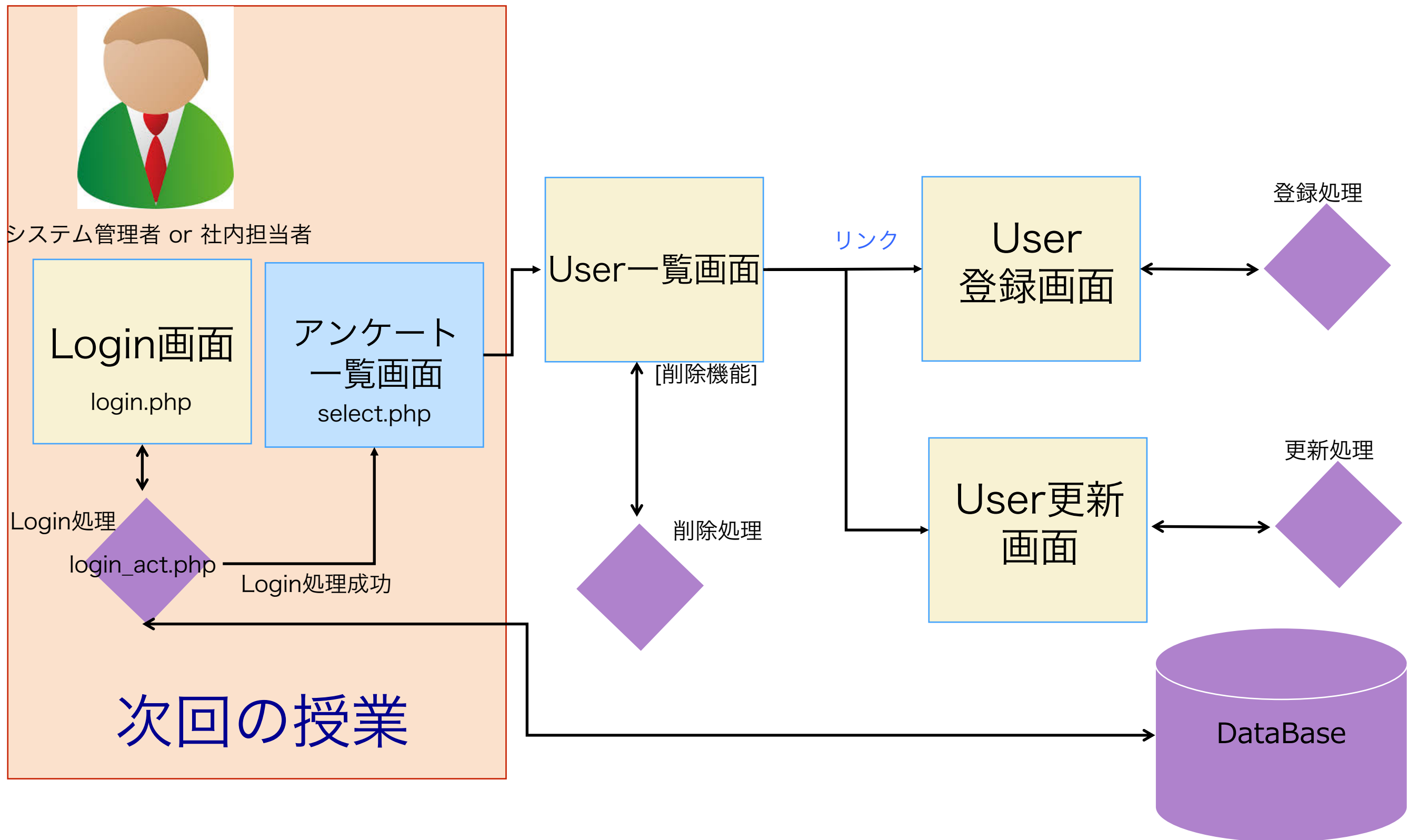
## ◇管理ユーザーテーブルを作成

- **DB名:** `gs_db`
- **Table名:** `gs_user_table`
- **Field名:**
  - id: `int(12) AUTO INCREMENT PRIMARY KEY`
  - name : `var_char(64)`
  - lid: `var_char(128)`
  - lpw: `var_char(64)`
  - kanri\_flg: `int(1) ※0=管理者, 1=スーパー管理者`
  - life\_flg: `int(1) ※0=使用中, 1=使用しなくなった`

※Fieldの右にあるのは、データ型(Type)です。

<http://mysql.akarukutanoshiku.com/category5/entry21.html>

# 3. システム管理者機能(次回の授業)





# 管理機能完成図

# 管理システム完成図

