



網路程式設計實習

Unit14 - SQL UPDATE & DELETE

班級：資訊二乙

座號：26

姓名：陳維基



單元內容描述

1. SQL Delete
2. SQL Update

1 SQL Delete

Delete 也是操作資料庫不可或缺的項目之一，而在SQL 中 Delete 的語法與 Select 有些相似。一般來說 Delete 也會搭配 Where 使用，不過為了避免誤刪資料，一般都是以主鍵、唯一鍵值來判斷，比較不會有問題。

功能	SQL 指令
將 table1 的所有資料刪除	Delete from table1
將 table1 的 id 為 1 的資料刪除	Delete from table1 where id = 1

2 SQL Update

資料更新，對於各資料的狀態更新也是常見的操作之一，而Update 的語法跟 Insert 有多少相似，同樣對欄位操作，不過還是有些許不一樣，Insert 會是一次給欄位再給所有值，Update 則是給一個欄位後直接賦值，有點像變數賦值的概念。Update 基本上就是得綁著 Where 了，不然整張資料表都會被更動

功能	說明
將 table1 中 id 為 1 的項目的 column1 改為 60	Update set column1 = 60 where id = 1
將 table1 中 id 為 1 的項目的 column1 改為 60, column2 改為 20	Update set column1 = 60, column2 = 20 where id = 1



雜記

1. Password Check
2. Data Get
3. Data Update / Insert
4. Format Check Alert
5. Data Delete
6. Data Refresh
7. Data Sort
8. Data Analysis



1 Password Check

這次因為是對資料庫操作，做了個簡單的防止其他人無故操作系統的登入系統，不過只有針對輸入的密碼。

做法主要是透過重整頁面的概念，雖然說是重整頁面，但還是有向自己傳輸資料，接者再透過PHP判斷是否登入成功。且為了防止透過變更Html的Style就可以操作物件，我還是有將密碼每次都傳遞給連接資料庫的PHP。

1 Password Check

輸入密碼的Dialog

```
<div id="login" class="dialog">
  <div class="inputBox">
    <input id="pwd" class="input" placeholder=" " type="password" autocomplete="off">
    <label class="hint">Password</label>
    <i></i>
  </div>
  <p>密碼在php檔內，若要使用此系統請輸入該密碼</p>
  <div class="buttonParent"><button onclick="login()">Login</button></div>
</div>
```

登入重新導向自己

```
function login(){
  var form = document.createElement("form")
  form.innerHTML = `<input id="password" type="hidden" name="password" value=${document.getElementById("pwd").value}>`
  form.action = "hw01.php"
  form.method = "post"
  form.style.visibility = "hidden"
  document.body.appendChild(form)
  form.submit()
}
```

2 Data Get

這次連接資料庫的方法同為使用XMLHttpRequest(), 不過這周有做密碼的判斷, 所以要多處理密碼錯誤的問題。

```
function getAll() {
    var data = new FormData()
    data.append("mode", 0)
    data.append("password", password)
    var req = new XMLHttpRequest()
    var url = "select.php"
    req.open("POST", url, true)
    console.log(password)

    req.onreadystatechange = () => {
        if (req.readyState == 4 && req.status == 200){
            console.log(req.responseText)
            var res = JSON.parse(req.responseText)
            try{
                if (res.state == "fail"){
                    dbFail()
                    return
                }
            }
            catch {}
            db = JSON.parse(req.responseText)
            refresh()
        }
    }
    req.send(data)
}
```


3 Data Update / Insert

這次的更新以及新增都是用自訂的Dialog 來處理，畫面的呈現上主要是差在有沒有預先給資料，跟欄位是否可輸入而已，跟資料庫溝通的方式其實也只差在送的一項參數不同而已，不過還沒有想到方法可以一次將兩者整合就是了。以下為訂定的 Dialog 片段。

```
<div id="editBoxBackground" class="dialogBackground close">
  <div id="editBox" class="dialog close">
    <div class="inputBox">
      <input id="studentName" class="input" placeholder=" " type="text" autocomplete="off">
      <label class="hint">姓名</label>
      <i></i>
    </div>
    <div class="inputBox">
      <input id="seatNo" class="input sendCheck" placeholder=" " type="text" autocomplete="off" pattern="^(?:0[1-9]|[1-9][0-9])$"
    >
      <label class="hint">座號</label>
      <i></i>
    </div>
    ...
  </div>
</div>
```

3 Data Update / Insert

JavaScript 處理這兩個功能的程式片段，而其中是靠mode來區分要做的事。

```
function editRequest(studentData){
    var data = new FormData()
    data.append("mode", mode)
    data.append("password", password)
    data.append("name", studentData.name)
    data.append("seatNo", studentData.seatNo)
    data.append("chinese", studentData.chinese)
    data.append("english", studentData.english)
    data.append("math", studentData.math)
    data.append("pro1", studentData.pro1)
    data.append("pro2", studentData.pro2)
    var req = new XMLHttpRequest()
    var url = "select.php"
    req.open("POST", url, true)

    req.onreadystatechange = () => {
        if (req.readyState == 4 && req.status == 200){
            console.log(req.responseText)
            refresh()
        }
    }
    req.send(data)
}
```

4 Format Check Alert

在 unit08 的作業中就有做過類似的功能，不過這次用來判斷輸入的分數是否符合範圍，以及資料是否完整，再透過Alert 顯示。以下為處理Alert 的程式片段

```
var name = document.getElementById("studentName")
if (name.value.trim() === ""){
  newAlert({
    content : `姓名欄位不可為空`,
    mode: "error",
    during : 5000,
    behavior : {
      smoothIn : true,
      float: false
    }
  })
  return
}
```

```
var elements = [...document.getElementsByClassName("sendCheck")]

for (const element of elements) {
  if (!new RegExp(element.getAttribute("pattern")).test(element.value)){
    element.focus()
    newAlert({
      content : `${element.id}欄位的格式不正確`,
      mode: "error",
      during : 5000,
      behavior : {
        smoothIn : true,
        float: false
      }
    })
    return
  }
}
```

5 Data Delete

這次因為會有需要刪除資料，不過通常刪除資料都會要求使用者再次確認，所以這次做出了一個簡單的確認Dialog，確保使用者不是誤刪資料。

```
<div id="deleteConfirmBackground" class="dialogBackground close">
  <div id="deleteConfirm" class="dialog close">
    <h2>確定刪除?</h2>
    <div class="buttonParent">
      <button onclick="sendDelete()">Submit</button>
      <div class="divider"></div>
      <button class="close" onclick="toggleDialog
('deleteConfirm', 'deleteConfirmBackground')">Close</button>
    </div>
  </div>
</div>
```

6 Data Refresh

這次的作業中因為是將資料庫的資料直接存值本地端，所以在處理資料每次接近來的資料都要花上許多時間。且還要重新刷新表格以及排序等等，在這部分是花了很多時間撰寫。至於寫法與動態生成表格沒有差多少，只不過多了處理 Icon 的部分。

```
tr.innerHTML += `<td></td>`  
tr.innerHTML += `<td></td>`  
tr.innerHTML += `<td></td>`
```

7 Data Sort

這次的排序也都是應用了之前作業所撰寫過的，不過這次的資料會刷新的緣故，所以在每次刷新資料都必須重新整理排序一次才行。

```
if (index == 0){  
    sortMode = 1  
    data = data.sort((a, b) => sortMode*(a.no - b.no  
    })
```

8 Data Analysis

這也是運用了過去作業的功能，這次是用側欄的方式顯示圖表。

```
chart = new Chart("chart",{
  type: "radar",
  data: {
    labels: ["國文", "數學", "英文", "專一", "專二"]
  },
  options: {
    legend: {
      labels: {
        fontColor: "ffff",
        fontSize: 18
      }
    },
    scale: {
      ticks: {
        stepSize: 10,
        max: 100,
        min: 0
      },
      angleLines: {
        color: "#e1e1e5",
        lineWidth: 1,
      },
      gridLines: {
        color: "#e1e1e5",
        lineWidth: 1
      },
      pointLabels: {
        fontColor: "ffff",
        fontSize: 18
      }
    }
  }
})
```



心得

在過去我一直對撰寫網頁就有很深的興趣，不過一直沒有機會將他學好，透過這學期的課程，還有新增了一些神奇的功能，讓我知道在撰寫網頁是一件非常有趣的事。

從最基礎的Html 到 CSS 甚至到 PHP 及 JavaScript, 都可以發現許多有趣的部份，無論是Html class 屬性與 CSS 運用，還是 PHP 與 JavaScript 的溝通，都是十分的吸引我。



心得

或許是因為有選手訓練的經驗吧？總可以依些經驗尋找到許多有趣功能寫法，而且對於資料庫的理解很高，所以在撰寫SQL、產生表格的過程中都是很順利的。

透過這學期所學的，可以運用原先就學過的技術，在與新學到的知識結合，希望可以為未來工程師的生活奠下基礎。