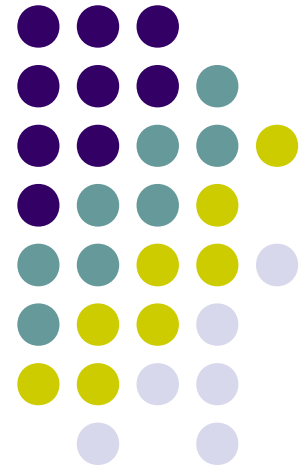
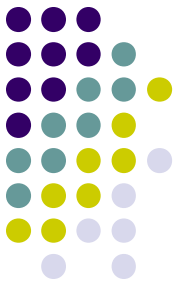


# JSP & MySQL

---





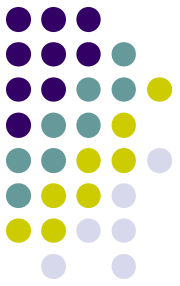
# SQL 操作教學

- **Insert a record**

**SQL> insert into personal\_data (name, passwd)  
values ('kclin', 'test1234');**

- **List all the records**

**select \* from personal\_data;**



# SQL 操作教學

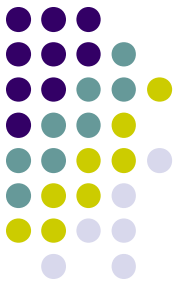
- **Updating existing records**

```
SQL> update personal_data set passwd='test5678'  
where field01='kclin';
```

- **Deleting records**

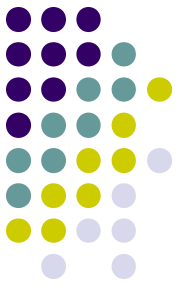
```
SQL> delete from personal_data where  
name='kclin';
```

# JDBC



- <http://www.mysql.com/downloads/connector/j/>
- copy mysql-connector-java-5.1.22-bin.jar 到 Tomcat 目錄 \common\lib, 或
- Tomcat 目錄 \webapps\ 你的 web\WEB-INF\lib

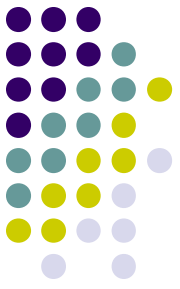
# 資料庫連結的建立



- 在正式連結資料庫時，請先利用 **Class.forName** 方法載入驅動程式類別：

```
Class.forName("org.gjt.mm.mysql.Driver").newInstance();
```

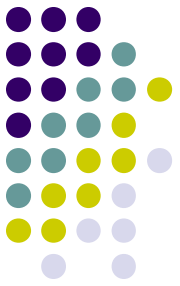
# 資料庫連結的建立



- 驅動程式載入記憶體後，將自動利用驅動程式管理員 - `java.sql.DriverManager` 類別，註冊載入的驅動程式，使其狀態為可使用（**available**），便可利用 `DriverManager` 類別開啓資料庫連結，語法如下：

```
Connection connection 物件變數 =  
                                DriverManager.getConnection  
(" 資料來源名稱 ", " 使用者帳號 ", " 密碼 ")
```

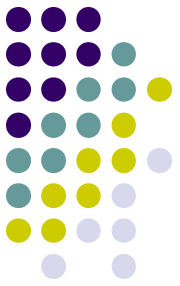
# JSP connect MYSQL (db\_ini.jsp)



- ```
<%@ page import="java.sql.*"%>
<%!
    String user = "root";
    String pass = "viva19640814";
    String database = "peresonal";
    String url = "jdbc:mysql://127.0.0.1/" + database + "?useUnicode=true&characterEncoding=big5";
    // 建立一個聯結物件
    Connection conn;
    // 建立 PreparedStatement 物件
    PreparedStatement pstmt = null;
%>
<%
    try{
        // 定義驅動程式與資料來源之間的連結
        Class.forName("org.gjt.mm.mysql.Driver").newInstance();
        // 建立一個聯結物件
        conn = DriverManager.getConnection(url,user,pass);
    }catch(SQLException sqle){
        out.println("SQL Exception : " + sqle);
    }
%>
```

# 從資料庫取得資料

## - 記錄集的觀念



- 什麼是記錄集 (**ResultSet**)

在程式裡，當透過資料庫連結，執行 **SQL** 敘述進行資料的查詢時，回傳的查詢結果稱之為記錄集。

- 記錄集的使用與記錄指標

記錄集是由一組記錄所組成，一次僅能存取一筆記錄。而此記錄即記錄指標所指向的記錄，該筆記錄亦稱之為目前記錄（**Current Record**）。



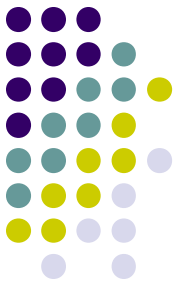
# 從資料庫取得資料

- 記錄集的觀念

| 紀錄指標 | index | name | category | address                        | other    |
|------|-------|------|----------|--------------------------------|----------|
|      | 2     | 陳易於  | 朋友       | 台北市新生南路 125 之 3 號              |          |
| →    | 2     | 蔣玉冰  | 同事       | 台北市中正路一段 25 號                  |          |
|      | 6     | 吳意倩  | 同學       | 高雄市五福一路 125 巷 25 弄 10 號 3 樓之 1 | 台大<br>土木 |
|      | 10    | 郭尚君  | 朋友       | 基隆市政容街 120 號                   |          |

# 從資料庫取得資料

## - 從資料來源取得資料



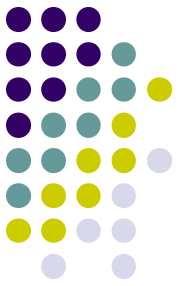
- 建立 **Statement** 物件

執行查詢動作前，需在程式內呼叫 **Connection** 物件的 **createStatement** 方法建立 **Statement** 物件，以代表執行的 **SQL** 敘述。

語法如下：

```
Statement statement 物件變數 =  
    connection 物件變數 .createStatement()
```

# 從資料庫取得資料



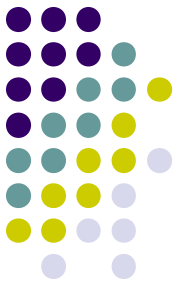
## - 從資料來源取得資料

- 若希望設定 **Statement** 物件所產生 **ResultSet** 物件的型態，則產生 **Statement** 物件時，必須使用以下的語法：

**Statement statement 物件變數 =**  
**connection 物件變數 .createStatement(int 指標型**  
**態 , int 資料一致性 )**

# 從資料庫取得資料

## - 從資料來源取得資料



- 下面是語法各部份的說明：

**statement** 物件變數：

由 **Connection** 物件建立的 **Statement** 物件。

**connection** 物件變數：

代表資料庫連結的 **Connection** 物件。

# 從資料庫取得資料

## - 從資料來源取得資料

指標型態：設定 **Statement** 物件執行 **SQL** 敘述後，所產生 **ResultSet** 物件的指標特性，設定所使用的參數為 **ResultSet** 物件之類別屬性。

| 類別屬性                    | 說明                                                  |
|-------------------------|-----------------------------------------------------|
| TYPE_FORWARD_ONLY       | 游標僅能由第一筆記錄向最後一筆移動，也就是游標僅能向前移動。                      |
| TYPE_SCROLL_SENSITIVE   | 允許游標向前或向後移動，且其他 <b>ResultSet</b> 物件變動游標時，將影響游標的位置。  |
| TYPE_SCROLL_INSENSITIVE | 允許游標向前或向後移動，但其他 <b>ResultSet</b> 物件變動游標時，並不影響游標的位置。 |

# 從資料庫取得資料

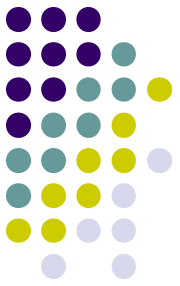
## - 從資料來源取得資料

資料一致性：由 **Statement** 物件執行 **SQL** 敘述後，產生之 **ResultSet** 物件的資料一致性，設定所使用的參數為 **ResultSet** 物件之類別屬性。

| 類別屬性              | 說明                        |
|-------------------|---------------------------|
| CONCUR_READ_ONLY  | ResultSet 物件中的資料僅能讀取不能修改。 |
| CONCUR_UPDATEABLE | ResultSet 物件中的資料可讀取亦可修改。  |

# 從資料庫取得資料

## - 從資料來源取得資料



- 執行查詢動作

- 以下語法將執行查詢動作，並將查詢結果以 **ResultSet** 物件傳回，語法如下：

```
ResultSet resultset 物件變數 =  
    statement 物件變數 .  
    executeQuery(String SQL 敘述 )
```

# 從資料庫取得資料

## - 從資料來源取得資料



- 語法各部份的說明：

- ✓ **resultset** 物件變數：

引用至 `executeQuery` 方法，回傳查詢結果的 `ResultSet` 物件。

- ✓ **statement** 物件變數：

執行 SQL 敘述的 `Statement` 物件。

- ✓ **SQL** 敘述字串：

欲執行的 SQL 敘述。



# 從資料庫取得資料

## - 從資料來源取得資料



- 取得欄位資料

- 欲從 **ResultSet** 物件中，取得記錄指標目前指向記錄各欄位的資料時，可以依照欄位內的資料型別呼叫 **getXXX** 方法。

語法如下：

**resultset** 物件變數 **.getXXX(" 欄位名稱 ")**

或

**resultset** 物件變數 **.getXXX(" 欄位索引 ")**

# 從資料庫取得資料

## - 從資料來源取得資料



- 語法各部份的說明：

**resultset** 物件變數： 取得資料的 **ResultSet** 物件變數。

欄位名稱： 從 **ResultSet** 物件中，欲取得資料的欄位名稱。

欄位索引： 從 **ResultSet** 物件中，欲取得資料的欄位索引，第一欄索引值為 **1**。

# 從資料庫取得資料

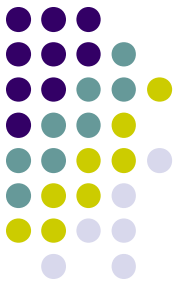
## - 從資料來源取得資料



- 利用 **while** 取得記錄集中的所有記錄
  - 欲從這個虛擬的資料工作表中取得某筆記錄時，必須 先將記錄指標移至該記錄上，再利用 **ResultSet** 物件的 **getXXX** 方法取得欄位資料。
  - 以下語法從 **rs** 物件中，取得記錄指標目前所指向記錄的 **BookID** 欄位值，該欄位資料的型別為 **String**：

**rs.getString("BookID")**

# 從資料庫取得資料



## ■ 從資料來源取得資料

- 最標準的記錄集資料讀取方法：利用 **while** 迴圈，從記錄集的第一筆記錄，向前一筆一筆取出記錄。

語法如下：

```
while(rs.next()){  
    // 若指標移至最後一筆記錄之後時，傳回 false  
    .....  
}
```

# 從資料庫取得資料

## - 從資料來源取得資料



- 取出 **ResultSet** 物件的資料時，先用 **afterLast** 方法，將指標移到 **ResultSet** 物件中最後一筆記錄之後，再呼叫 **previous** 方法，從最後一筆記錄開始，向後取得記錄集中的資料。

```
rs.afterLast(); // 將指標移至最後一筆記錄之後  
while(rs.previous()){ // 若指標移至第一筆記錄之前時  
    , 傳回 false  
    .....  
}
```

# 從資料庫取得資料

## - 從資料來源取得資料



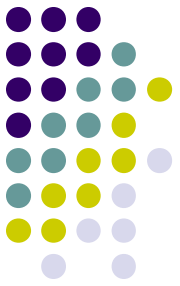
- **ResultSet** 物件與 **Statement** 物件的關閉

使用完 **ResultSet** 物件與 **Statement** 物件後，可以呼叫 **close** 方法，關閉它們，並釋放所佔用的記憶體空間。

語法如下：

**resultset** 物件變數 **.close()**

**statement** 物件變數 **.close()**



# 取得記錄集中欄數與欄位名稱

- 取得記錄集中的欄位數

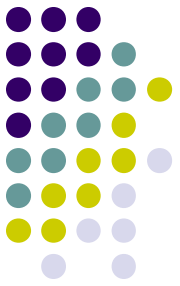
語法如下：

```
ResultSetMetaData resultsetmetadata 物件 =  
                                rs.getMetaData();
```

```
// 取得 ResultSetMetaData 物件
```

```
int ColNum = resultsetmetadata 物件  
                                .getColumnCount();
```

```
// 取得欄位數目
```



# 取得記錄集中欄數與欄位名稱

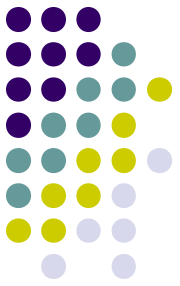
- 取得記錄集中欄位的名稱

✓ 語法如下：

**resultsetmetadata** 物件變數 .

**getColumnName( 欄位索引值 )**





# 取得記錄集中欄數與欄位名稱

✓ 語法中各部份說明如下：

**resultsetmetadata** 物件變數：

已完成建立的 **ResultSetMetaData** 物件。

欄位索引值：

記錄集中代表各欄位的索引值。



# 記錄指標位置的設定 / 取得與記錄筆數

- 記錄指標位置的取得  
語法如下：

**resultset** 物件變數 **.getRow()**

- 記錄指標位置的設定  
語法如下：

**resultset** 物件變數 **.absolute(int 記錄位置 )**



# 記錄指標位置的設定 / 取得與記錄筆數

- 記錄筆數的取得

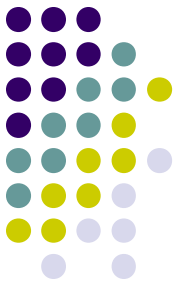
呼叫 **last** 方法，將記錄指標移至最後一筆記錄，再呼叫 **getRow** 方法，取得代表該筆記錄之位置的數字，即為該 **ResultSet** 物件的記錄總筆數。

- 其他 **ResultSet** 的記錄指標操作方法

請參考下一張投影片。

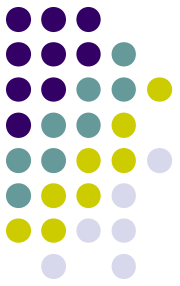
# 記錄指標位置的設定 / 取得與記錄筆數

| 方法            | 傳回值     | 說明                                         |
|---------------|---------|--------------------------------------------|
| first         | boolean | 將記錄指標移至第一筆記錄上，成功則傳回 true ，反之傳回 false 。     |
| isAfterLast   | boolean | 判斷記錄指標是否指向最後一筆記錄之後，是則傳回 true ，反之傳回 false 。 |
| isBeforeFirst | boolean | 判斷記錄指標是否指向第一筆記錄之前，是則傳回 true ，反之傳回 false 。  |
| isFirst       | boolean | 判斷記錄指標是否指向第一筆記錄，是則傳回 true ，反之傳回 false 。    |
| isLast        | boolean | 判斷記錄指標是否指向最後一筆記錄，是則傳回 true ，反之傳回 false 。   |
| last          | boolean | 將記錄指標移至最後一筆記錄上，成功則傳回 true ，反之傳回 false 。    |



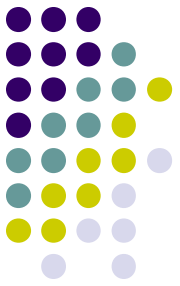
# PreparedStatement-insert

- `PreparedStatement pstmt = null;`
- `String sql = "insert into test (name) values (?);"`  
`pstmt = conn.prepareStatement(sql);`  
`String name = request.getParameter("name");`  
`pstmt.setString(1, name);`  
`pstmt.executeUpdate();`



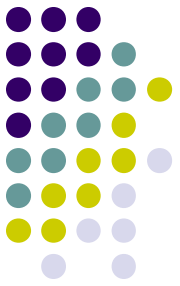
# PreparedStatement-Update

- ```
int num = Integer.parseInt(request.getParameter("num"));  
String name = request.getParameter("name");  
String sql = "update test set name = ? where num = ?";  
pstmt = conn.prepareStatement(sql);  
pstmt.setString(1 , name);  
pstmt.setInt(2 , num);  
pstmt.executeUpdate();
```



# PreparedStatement-delete

- ```
String sql = "delete from test where num = ?";  
pstmt = conn.prepareStatement(sql);  
int num = Integer.parseInt(request.getParameter("num"));  
pstmt.setInt(1 , num);  
pstmt.executeUpdate();
```

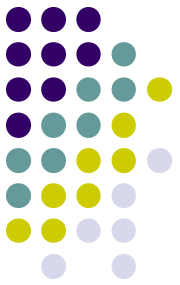


# PreparedStatement-Query

- `// 建立查詢並產生 rs 的資料表`  
    `String sql = "select * from test";`  
    `pstmt = conn.prepareStatement(sql);`  
    `ResultSet rs = pstmt.executeQuery();`  
`// 開始顯示`  
    `while(rs.next()){`  
        `String name = rs.getString("Name");`  
        `String num = rs.getString("Num");`  
        `out.print("Name : " + name + " Num : " + num + "<br>");`

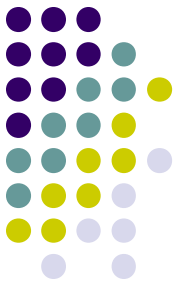


# insert.jsp



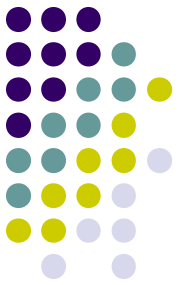
- ```
<%@ include file = "db_ini.jsp"%>
<%
    if(request.getParameter("insert") != null){
        String sql = "insert into test (name) values (?";
        pstmt = conn.prepareStatement(sql);
        String name = request.getParameter("name");
        pstmt.setString(1 , name);
        pstmt.executeUpdate();
        out.print("Insert success !  return index.jsp after 3 seconds !  ");
        //3 秒後將網頁轉跳到首頁
        response.setHeader("Refresh" , "3;url=index.jsp");
    }
%>
<html>
    <head>
        <title>Insert</title>
    </head>
    <body>
        <form action = "insert.jsp" method = "post">
            Name : <input type = "text" name= "name" size = "10">
            <br />
            <input type = "submit" name = "insert" value = " 新增 ">
        </form>
        <a href="index.jsp"> 回首頁 </a>
    </body>
</html>
```

# delete.jsp



- ```
<%@ include file = "db_ini.jsp"%>
<%
    if(request.getParameter("delete") != null){
        String sql = "delete from test where num = ?";
        pstmt = conn.prepareStatement(sql);
        int num = Integer.parseInt(request.getParameter("num"));
        pstmt.setInt(1 , num);
        pstmt.executeUpdate();
        out.print("Delete success ! return index.jsp after 3 seconds ! ");
        //3 秒後將網頁轉跳到首頁
        response.setHeader("Refresh" , "3;url=index.jsp");
    }
%>
<html>
<head>
<title>Update</title>
</head>
<body>
<form action = "delete.jsp" method = "post">
    Num : <input type = "text" name= "num" size = "10" value = "">
    <br />
    <input type = "submit" name = "delete" value = " 刪除 ">
</form>
<a href="index.jsp"> 回首頁 </a>
</body>
</html>
```

# update.htm



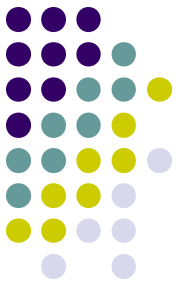
- ```
<html>
  <head>
    <title>Update</title>
  </head>
  <body>
    <form action = "update.jsp" method = "post">
      Num : <input type = "text" name= "num" size = "10">
      <br />
      <input type = "submit" name = "select" value = " 修改 ">
    </form>
    <a href="index.jsp"> 回首頁 </a>
  </body>
</html>
```

# update.jsp

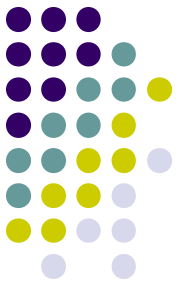
- ```
<%@ include file = "db_ini.jsp"%>
<%
    if(request.getParameter("update") != null){
        int num = Integer.parseInt(request.getParameter("num"));
        String name = request.getParameter("name");
        String sql = "update test set name = ? where num = ?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1 , name);
        pstmt.setInt(2 , num);
        pstmt.executeUpdate();
        out.print("Update success ! return index.jsp after 3 seconds ! ");
        //3 秒後將網頁轉跳到首頁
        response.setHeader("Refresh" , "3;url=index.jsp");
    }
    // 抓取資料
    int num = Integer.parseInt(request.getParameter("num"));
    String sql = "select Name from test where num = " + num;
    pstmt = conn.prepareStatement(sql);
    ResultSet rs = pstmt.executeQuery();
    rs.next();
    String Name = rs.getString("Name");
%>
<html>
    <head>
        <title>Update</title>
    </head>
    <body>
        <form action = "update.jsp" method = "post">
            Name : <input type = "text" name= "name" size = "10" value = "<%=Name%>">
            <br />
            <input type = "submit" name = "update" value = " 修改 ">
        </form>
        <a href="index.jsp"> 回首頁 </a>
    </body>
</html>
```



# index.jsp



- ```
<%@ include file = "db_ini.jsp"%>
<html>
  <head>
    <title>Index</title>
  </head>
  <body>
<%
  // 建立查詢並產生 rs 的資料表
  String sql = "select * from test";
  pstmt = conn.prepareStatement(sql);
  ResultSet rs = pstmt.executeQuery();
  // 開始顯示
  while(rs.next()){
    String name = rs.getString("Name");
    String num = rs.getString("Num");
    out.print("Name : " + name + " Num : " + num + "<br>");
  }
  // 關閉資料表
  rs.close();
%>
  <a href="insert.jsp"> 新增資料 </a><br>
  <a href="delete.jsp"> 刪除資料 </a><br>
  <a href="update.htm"> 修改資料 </a><br>
</body>
</html>
```



# Homework

- 註冊及登入系統
- 第一次登入系統必須先註冊
  - Userid, name, password, birthday, memo
  - 以 jQuery 檢查欄位是否空白
- 註冊完返回首頁重新登入
- 登入後以 jQuery 顯示增刪改查之選單