111第二學期資料庫管理–期末專案

**指導教授 : 柯士文教授**

**助教：盧俊吉、李育慈、陳佳辰**

**Hash :**[**http://www.lorem-ipsum.co.uk/hasher.php?fbclid=IwAR19\_Ytf1P\_Vd49tp95rhrXFum-z7MqFBXuZgwUSKBFmuu12pqfMH8vNj58**](http://www.lorem-ipsum.co.uk/hasher.php?fbclid=IwAR19_Ytf1P_Vd49tp95rhrXFum-z7MqFBXuZgwUSKBFmuu12pqfMH8vNj58)

**Salt :** [**https://www.grc.com/passwords.htm**](https://www.grc.com/passwords.htm)

[Adding salt to Hashing: A Better Way to Store Passwords](https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/)

**OWASP Cheat Sheet Series:**

* [Cryptographic Storage Cheat Sheet](https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html)
* [Password Storage Cheat Sheet](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html)

[W3Schools PHP Online Compiler](https://www.w3schools.com/php/php_compiler.asp)

[PHP password\_hash API](https://www.php.net/manual/en/function.password-hash.php): Creates a password hash

[PHP random\_bytes API](https://www.php.net/manual/en/function.random-bytes.php): Generates cryptographically secure pseudo-random bytes

# 作業繳交注意事項

1. **(請注意檔名請務必完全相符！！)**
2. **ER Diagram (PDF 檔) →** (檔名:db\_學號.pdf，e.g. db\_110423043.pdf)
3. **ER Model 匯出檔 (.mwb 檔) →** (檔名:db\_學號.mwb，e.g. db\_110423043.mwb)
4. **DataBase 的 schema (.sql 檔)** → (檔名:db\_學號.sql，e.g. db\_110423043.sql)

# 情境說明

本專案之題目設計係根據「選課系統」的運作方式進行設計，主要包含學生以及老師於選課系統上的相關操作，例如學生查詢歷年選課結果、登記初選課程、查詢開課課程以及老師修改授課資訊(上課教室、時間)等情境。本專案之設計並不如正式的選課系統完整，因此選課系統的設計限制請依本專案所給予的條件為主，並著重於資料庫的設計。

# 

# ER-Model, Tables & Stored Procedures

1. 請將自己試想為 DBA(Database Administrator)，依照以下需求設計資料庫中的 Stored procedures。
2. 以下為資料庫中的 Stored procedures 內容與描述，總共有 12 個 stored procedures(**加分題為[11] , [12]** )。實作時請依照以下 Stored procedures 的命名進行。
3. SP的input：不管是Update, Create, Get的input，都須符合題目要求的input
4. SP的output：
   1. Update、Create的SP：須符合affected\_row\_num，影響多少的資料筆數會看各位資料庫的設計來判斷是否正確
   2. Get的SP：須至少符合文件內result set所列出的欄位內容，其他想多加的資訊各位可自行發揮
5. 假資料**主要提供給各位理解題目**，若完全以提供的假資料執行Stored Procedures可能會產生邏輯上的衝突或是無法反映出題目的目標，**建議各位可以根據題目的需求建立自己的假資料**。
6. **Stored procedures 之命名必須完全相符，有任何命名不一的情況，該題將以 0 分計算。**

# Stored Procedures

## [1] sp\_Login (8%)

1. 透過 sp\_Login (student\_id (int), 加密後的 pwd) 學生登入系統
2. **input:** student\_id(int), hashedpwd(varchar(200))
3. **output:** status\_code (int) → 登入狀態，為了確認 Store procedure 有效的執行
4. **Definition of status\_code**:

status\_code = 1 --> 登入成功

status\_code = 2 --> 登入失敗 (學生存在但密碼錯誤)

status\_code = 3 --> 登入失敗 (學生不存在)

## 

## [2] sp\_UpdatePwd (10%)

1. 透過 sp\_UpdatePwd (student\_id, 加密後的 pwd, salt) 更改學生密碼
2. **input:** student\_id (int), hashedPwd (varchar(200)), salt (char(64))
3. **output:** affected\_row\_num (int) → 被影響的列數，為了確認 Store procedure 有效的執行

**[3] sp\_GetSelectionCourse (10%)**

1. 透過 sp\_GetSelectionCourse 於During course selection的期間，讓學生(student\_id) 取得目前其已經初選(Preliminary)以及中選(Enrolled)**的**課程
2. **input :** student\_id (int), semester (int)
3. **output:** number of rows in the result set (int)

**results set: (以 student\_id=110123123, semester=1112為例，假設目前學期為1111)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **student\_id** | **semester** | **course\_code** | **course\_name** | **status** |
| 110123123 | 1112 | IM1023 | Finance Management | Preliminary |
| 110123123 | 1112 | IM1022 | Data Science | Enrolled |
| 110123123 | 1112 | IM1025 | Machine Learning | Preliminary |

**[4] sp\_GetCourseStudent (10%)**

1. 透過 sp\_GetCourseStudent 輸出某一學期的某一課程的修課名單，修課名單只會包含已中選(Enrolled)的學生
2. **input:** course\_id (int)
3. **output:** number of rows in the result set (int)

**results set: (以 course\_id=9 為例)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **semester** | **degree\_program** | **course\_code** | **course\_name** | **student\_id** | **student\_name** |
| 1112 | 3 | IM2056 | Introduction to Blockchain | 110123123 | Kevin Wang |
| 1112 | 3 | IM2056 | Introduction to Blockchain | 110123120 | Mei |

**[5] sp\_UpdateWithdrawCourse (10%)**

1. 透過 sp\_UpdateWithdrawCourse 讓學生退選課程
2. 請注意此SP不需負責檢查是否能夠退選，只需負責退選課程即可，亦即input若輸入進來則表示已經是確定可以退選的情況，這個SP只須完成操作資料庫(update withdraw)這件事即可
3. **input:** student\_id (int), course\_id (int)
4. **output:**  affected\_row\_num (int) → 被影響的列數，為了確認 Store procedure 有效的執行

**result set: (以 student\_id=110562123, course\_id=6 為例)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **student\_id** | **degree**  **program** | **course**  **code** | **course**  **name** | **semester** | **status** |
| 110562123 | 1 | IM560 | Programming(Basic) | 1121 | Withdrawn |

**[6] sp\_GetCourseWithSemesterAndTeacher (12%)**

1. 透過 sp\_GetCourseWithSemesterAndTeacher 查詢某一老師在某一學期所開設的課程
2. 此SP只取學期以及老師做為輸入，輸出是該老師於該學期所教授的所有課程
3. **input:** semester (int), teacher\_id (int)
4. **output:** number of rows in the result set (int)

**result set: (以 semester=1121, teacher\_id=2356 為例)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **semester** | **teacher\_id** | **course\_code** | **course\_name** | **degree\_program** |
| 1121 | 2356 | IM1023 | Finance Management | 1 |
| 1121 | 2356 | IM1023 | Finance Management | 2 |
| 1121 | 2356 | IM1023 | Finance Management | 3 |
| 1121 | 2356 | IM1023 | Finance Management | 4 |
| 1121 | 2356 | IM1050 | Marketing Management | 2 |
| 1121 | 2356 | IM1050 | Marketing Management | 3 |
| 1121 | 2356 | IM1050 | Marketing Management | 4 |

**[7] sp\_GetDegreeProgramCourse (10%)**

1. 透過 sp\_GetDegreeProgramCourse 取得某一學程的所有必選修(*required and elective*)課程，並根據必選修進行排序
2. 此SP的輸入就是學程，輸出就是該學程下的所有必選修課程
3. **必選修排序：必修(required) → 選修(elective)，輸出的結果必修需再選修前面**
4. **input:** degree\_program\_id (int)
5. **output:** number of rows in the result set (int)

**result set: (以 degree\_program\_id=1 為例)**

|  |  |  |  |
| --- | --- | --- | --- |
| **degree\_program\_name** | **course\_code** | **course\_name** | **required or elective** |
| Bachelor of Information Management Degree Program | IM1023 | Finance Management | required |
| Bachelor of Information Management Degree Program | IM560 | Programming(Basic) | required |
| Bachelor of Information Management Degree Program | IM2022 | Data Science | elective |

**[8] sp\_GetPrerequisiteCourse (8%)**

1. 透過 sp\_GetPrerequisiteCourse 取得某課程的先修課程(prerequisite courses)
2. **input:** course\_code (varchar(45))
3. **output:** number of rows in the result set (int)

**result set: (以 course\_code=IM2056 為例)**

|  |  |  |  |
| --- | --- | --- | --- |
| **course\_code** | **course\_name** | **Prerequisite\_Course\_code** | **Prerequisite\_Course\_name** |
| IM2056 | Introduction to Blockchain | IM2060 | Information Security |

**[9] sp\_GetSelectableCourses (10%)**

1. 透過 sp\_GetSelectableCourses 根據學生(student\_id) 所修習之學程(Degree Program + Liberal Education Program)，取得該學生目前選課期間所能選擇之課程
2. 請注意**PPT內可選課程的條件**的問題
3. 若現在學期是1111，學生只能選1112的課程，因此此SP的輸出課程也只會是1112的課程
4. 不須擔心衝堂的問題，只需將符合可選課程條件的課程輸出即可(參考112問題整理的第9個SP)，系統後續會自動針對衝堂做處理
5. **input :** student\_id (int), semester (int)
6. **output:** number of rows in the result set (int)

**result set: (以 student\_id=110123123, semester=1112為例)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **student\_id** | **semester** | **degree program name** | **course code** | **course**  **name** |
| 110123123 | 1112 | Bachelor of Information Management Degree Program | IM1022 | Data Science |
| 110123123 | 1112 | Bachelor of Information Management Degree Program | IM2056 | Introduction to Blockchain |
| 110123123 | 1112 | Bachelor of Information Management Degree Program | IM2063 | Practical Course of Cloud Computing |

**[10] sp\_GetStudentCourseWithCourseCard (12%)**

1. 透過 sp\_GetStudentCourseWithCoursecard 取得某一學期某一課程中透過課程密碼卡選課的學生選課紀錄，**輸出需包含課程密碼卡流水號**
2. 不論學生的選課狀態為何，只要其當初選課有使用到密碼卡，就要輸出
3. **input :**course\_id (int)
4. **output:** number of rows in the result set (int)

**result: (以 course\_id=5 為例)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **semester** | **degree**  **program**  **id** | **course\_code** | **course\_name** | **student\_id** | **student\_name** | **serial\_number** |
| 1102 | 1 | IM1022 | Data Science | 110123123 | Kevin Wang | 50001 |
| 1102 | 1 | IM1022 | Data Science | 110123120 | Mei | 50002 |

**[11] sp\_OfferCourse (10%)**

1. 透過 sp\_OfferCourse 開設下一學期的課程，若開設的課程為該學程的必修課，須注意是否與該學程已開設的必修課時間上衝堂(schedule conflict)
2. 只須有(112問題整理的第11個SP)列的3個衝突檢查即可
3. **input :** degree\_program\_id (int), course\_code (varchar(45)), semester (int), teacher\_id (int), course\_time (varchar(45)), course\_room(varchar(45))
4. **output:** affected\_row\_num (int) → 被影響的列數，為了確認 Store procedure 有效的執行

**result set: (以degree\_program\_id=3, semester=1121, course\_code=IM560, teacher\_id=2356, 3120, course\_time=(Fri) 678, course\_room=I1-407為例)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **degree**  **program**  **id** | **semester** | **course**  **code** | **course**  **name** | **teacher**  **name** | **course**  **time** | **course**  **room** |
| 3 | 1121 | IM560 | Programming(Basic) | LIlly Chang、Thor Chang | (Fri) 678 | I1-407 |

**[12] sp\_GetRequiredCourse (10%)**

1. 透過 sp\_GetRequiredCourse 系統根據學生的Degree Program以及所屬年級取得學生下學期需修的必修課程
2. 學生**只能修低於(包含)自己所屬年級的必修課**，若輸出會有低於自己年級的必修課的可能情況為該生當初修習該必修課未通過或因衝堂而當初未修習
3. **輸出的必修課程不能衝堂：**同一學程同一年級的必修課不會衝堂，會衝堂的情況為該生先前有必修課未過，**衝堂的篩選優先順序：低於該生年級的未通過必修課(重修) → 該生當年級的所有必修課**
4. **input:** student\_id (int), semester (int)
5. **output:** number of rows in the result set (int)

**result set: (以 student\_id=110123123, semester=1121 為例)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **course\_id** | **course\_code** | **couse\_name** | **time** | **prerequisite\_year** |
| 1 | IM1023 | Finance Management | (Wed) 234 | 1 |
| 6 | IM560 | Programming(Basic) | (Fri) 678 | 1 |
| 11 | IM1025 | Machine Learning | (Mon) 567 | 2 |