【DB13 PHP】

PHP 變量名以$開頭，可包括字符，字母，數字和_。

-不能以數字開頭 / 變量不是鍵入(Type)的

<u>*4. PHP statement:</u> **<u>print uncords(strtolower ('JOHN smith'))?</u>**	-output : John Smith.
-strtolower()	字串改小(lower case)
-ucwords()	字首改大(uppercases)

PHP Arrays Can be numeric or associative	
<u>*-"Numeric array"</u> is based on a numeric index, starting from 0	基於數字索引，從 0 開始
-"Associative array " is based on a key => value relationship	基於 key => value 關係

【DB14 資料正規化(Normaliztion)】

*Define(a) Normalization 正規化

1.降低資料重複性(Data Redundancy)。

2.避免資料更新異常(Anomalies)。

3.提高關聯性資料庫的效能。

*Define (b) Functional Dependency 功能相依性

1.Full 完全 / Partial 部分 / Transitive 遞移

2. X -> Y , 表如果兩個 X 值相同，則它們必須具有相同的 Y 值

3.Which are correct regarding database design?
-Shall not generate spurious tuples.
-Shall not have too many null values in tuples.
-Shall avoid redundant information in tuples.

2NF/數據庫的表中不存在部分函數依賴

3nf/數據庫的表中不存在傳遞函數依賴

【BCNF】

*What is the definition of BCNF ? (5%)

- X->Y , X is a superkey of R.
- 如果 R 中存在 FD X->A，都滿足 X 是 R 的 Superkey，那麼 R 就屬於 BCNF。

Ex,

Property_id#	County_name	Lot#	Area

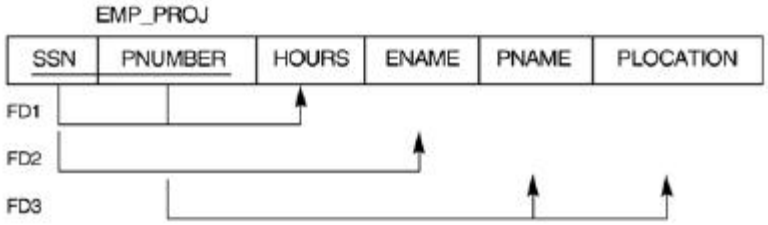
1.Property_id# -> {County_name,Lot#,Area};
2{County_name,Lot#} -> {Property_id#,Area};
3.Area->County_name;

很明顯 3.違反了 BC 正規化的要求，Area 不是關係模式 R 的主鍵，

Property_id#	Area	Lot# _____

Area	County_name

9. The relation EMP_PROJ is not a good relation. Please explain how this relation will cause (a)insert anomaly(不規則) and (b) delete anomaly. (10%)



10. Please normalize the above EMP_PROJ relation to 2NF. (10%)

DB25

【Dell-LaPadula Model】	
(d) Simple security property of Dell-LaPadula Model	

【Threats 威脅】	【countermeasures 對策】
*-Loss of integrity 誠信/對未經授權數據更改	-access control 訪問控制
-Loss of availability 可用/合法用戶無法取得數據	-inference control 推理控制
-Loss of confidentiality 信心/保護數據	-flow control 流量控制
	-encryption 加密

【Database Audit 數據審計】	
1.Database Audit (特定時間段內應用)	reviewing the log to examine all accesses and operations applied to the database during a certain time period.
2.Audit Trail (用於安全目的)	A database log that is used mainly for security purposes
【Frequent Attacks】	
*-SQL Injection	SQL 注入
-Unauthorized Privilege Escalation	未授權的權限升級
*-Denial of Service Attack	DOS 攻擊
-Weak Authentication	弱認證
*Please explain 3 protection techniques against SQL injection:	
(1) bind variables	

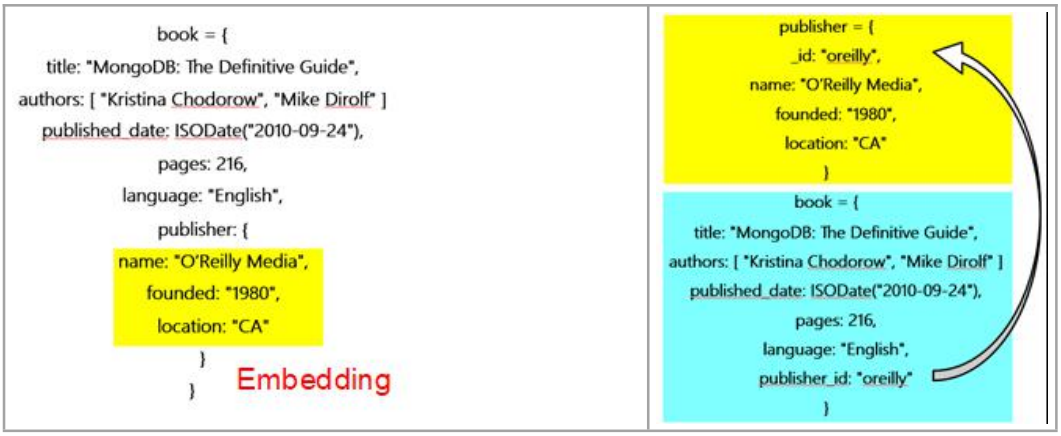
(2) filtering input	
(3) function security.	
【Statistical Queries 統計查詢規範】	
functions to a population of tuples	函數應用於群組
not allowed to retrieve individual data	不允許檢索個人資訊
allowing only statistical aggregate fun	僅允許統計聚合函數(COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION.)
【Public Key Encryption 公鑰加密】	
1.非對稱式加密，就是 pair 有兩個鑰匙	
2. A 有他自己的私鑰跟 B 的公鑰	
3.A 要傳東西給 B 就用 B 的公鑰加密	
4.B 拿到之後用 B 自己的私鑰解密	

【Ingredients of Public Key Encryption】	【Digital Certificate】
1.Plaintext 明文	The information in the certificate 信息
2.Ciphertext 密文	The certificate owner information 擁有者
3.Encryption 加密	The public key of the owner 公鑰
4.Decryption 解密算法	The date of issue of the certificate 日期
5.Public and private keys	The validity period specified 有效期
Plaintext -> (Encryption) -> Ciphertext	The digital signature of CA (identifier) 發行人
	must different for each by the same signer

*For public key encryption, which key shall A use to encrypt his plaintext if A wants to text to B?
-B's public key

*A population is a set of tuples of a relation that satisfy some selection condition.
*(x)The star property states that a subject S is not allowed read access to an object O unless class(S) >= class(O).

【DB98 Mongo DB】	
*two ways to establish the connection:	
A."Embedding" when many" obj always appear with their parents.	B."Linking" when you need more flexibility



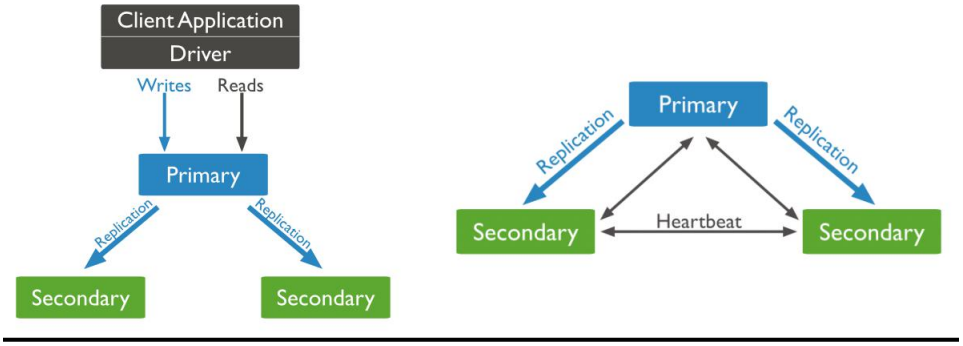
【Problems with SQL】	
-Rigid schema	剛性架構
-Not easily scalable	不易擴展
-Requires unintuitive joins	需不直觀的連接

【Replication】

提供一份資料庫副本，讓主資料庫發生異常時可以接手工作。

當主連線發生故障，會選出一台 Secondary 接手 Primary ，這能力為 Automatic Failover

Purpose : Fault tolerance(容錯) /Availability(可用)



Members

-Primary(Read, Write) / Secondary(Replication)

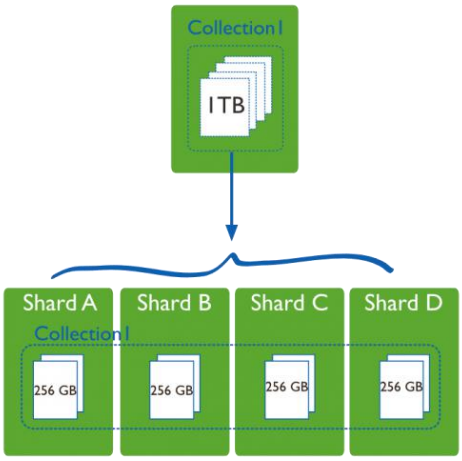
-Arbiter(Voting) (Can’ t be primary)

-Delayed Secondary (Can’ t be primary)

【Sharding】

為一種資料分散處理架構，簡單的說就是將資料片 (Shard) 儲存到不同的機器中

Purpose : Horizontal scaling out 橫向縮小



【CAP theorem 】
-C (Strong Consistency): 任何時候，任兩個節點得到的狀態都是一樣的。
-A (Availability): 若一個節點沒有壞掉，就必須要能正常服務。
-**P (Partition Tolerance):** 若叢集故障被切割多個 sub cluster 時，系統還能正常運作。

SQL	MongoDB
table	collection
row	**document**
column	**field**
table joins (e.g. select queries)	Embedded / linking
foreign Key	reference
primary keys	_id field (is always the pkey)
aggregation / group by	aggregation / pipeline
relational schema	schema-less

【Characteristics of NOSQL 特點】	
Related to distributed database systems	
-Scalability	可擴展性
-Availability, replication and eventual consistency	可用性，複製和最終一致性
-Replication models	複製模型
-Sharding of files	文件分片
-High-performance data access	高性能數據訪問
Related to data models and query languages	
-Not requiring a schema	不需要架構
-Less powerful query language	功能較弱的查詢語言
-Versioning	版本

- Creation index
  - db.users.ensureIndex({ score: 1 })
- Show existing indexes
  - db.users.getIndexes()
- Drop index
  - db.users.dropIndex({score: 1})
- Explain—Explain
  - db.users.find().explain()
  - Returns a document that describes the process and indexes
- Hint
  - db.users.find().hint({score: 1})
  - Overide MongoDB's default index selection

*Mongo DB is a type of wide column-store databases.(x)
*In MongoDB, how many documents will be isolated in a write operation? (Only a single document.)
*MapReduce is a type of aggregation operations which MongoDB provides. (o)

【DB99 Python】

NumPy	能使用一些高階的數學函數
pandas	分析資料結構的好幫手 **DataFrame #類似關聯式資料庫資料表的 Data type**
matplotlib	能簡易的分析數據並繪製成圖表
scikit-learn	機器學習軟體套件
Keras	深度學習的類神經網路元件

名稱	描述
abs(x)	回傳 x 的絕對值
len(s)	回傳 s 中的元素個數，s 可以是字串、list、set 等等
pow(x, y)	回傳 x 的 y 次方
range([start], stop, [step])	產生整數 list，start 省略時預設為 0，step 省略時預設為 1
round(x, n)	將 x 四捨五入到第 n 位

去除陣列中重複資料	
> A = [3, 4, 5, 3, 4, 5, 5, 6, 8, 9] > newA = set(A) > B = {6, 8, 9, 10, 11} > print (newA & B)	# newA = {3, 4, 5, 6, 8, 9} # set B # result of newA ∩ B {6, 8, 9}
矩陣相乘	

In [53]: import numpy as np ...: a = np.matrix('1 2; 3 4') ...: b = np.matrix('4 3; 2 1') ...: print (a*b) ...: [[8 5] [20 13]]	$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \times \begin{bmatrix}4 & 3\\2 & 1\end{bmatrix}$

mysql.connector.connect()	與資料庫連線
mysql.connector.cursor()	回傳 MySQLCursor 物件，必須使用此物件對資料庫下指令
MySQLCursor.execute()	執行指令，並儲存回傳結果
MySQLCursor.fetchall()	將結果以[(result1), (result2)…]格式回傳
cursor.close()	關閉 MySQL 連線 #避免佔用過多資源
connect.close()	關閉 MySQL 連線 #避免占用對 MySQL 的連線
Describe ( )	- statistics of numeric columns

與MySQL連線

```
# -*- coding: utf-8 -*-
import mysql.connector as DB

connect = DB.connect(user="your account", password="your password", host="127.0.0.1")
```

將此套件重新命名為DB

```
# -*- coding: utf-8 -*-
import mysql.connector as DB

config = {
    "user": "your account",
    "password": "your password",
    "host": "127.0.0.1",
    #"database": "database name",
    #"port": 3306,
    #...
    }
connect = DB.connect(**config)
```

建立Database、Table

使用「cursor.execute(sqlString)」對MySQL下指令

```
connect = DB.connect(**config)
cursor = connect.cursor()

sql = "CREATE DATABASE company;"
cursor.execute(sql)

# set connecting database
connect.database = "company"

sql = """
CREATE TABLE `employees`(
    `emp_no` INT(11) NOT NULL AUTO_INCREMENT,
    `birth_date` DATE NOT NULL,
    `first_name` VARCHAR(14) NOT NULL,
    `last_name` VARCHAR(16) NOT NULL,
    `gender` ENUM('M', 'F') NOT NULL,
    `hire_date` DATE NOT NULL,
    PRIMARY KEY(`emp_no`)
) ENGINE = InnoDB
"""
cursor.execute(sql)
```

新增資料-INSERT INTO

```
# easy way to execute SQL, but may not consider in security issue
addEmployeeByString = "INSERT INTO employees (first_name, last_name, hire_date,
gender, birth_date) VALUES ('Wang', 'Andy', '"+ str(date(2018, 1, 1)) +"',
'M', '"+ str(date(1990, 12, 31)) +"');"

cursor.execute(addEmployeeByString)

# recommended way to execute SQL when you get parameter from user
addEmployeeByParameter = "INSERT INTO employees (first_name, last_name,
hire_date, gender, birth_date) VALUES (%s, %s, %s, %s, %s);"
employeeData = ("Lin", "Helen", date(2018, 1, 1), "F", date(1991, 1, 23))

cursor.execute(addEmployeeByParameter, employeeData)
```

參數存放在tuple中

將DataFrame寫入MySQL

```
from sqlalchemy import create_engine
engine = create_engine('mysql://username:password@host:port/database')

companyInDataframe.loc[companyInDataframe.loc[:, "first_name"] == "Wang", "first_name"] = "Chen"
companyInDataframe.to_sql(name="employees", con=engine, if_exists="replace", index=False)

companyInDataframe = pandas.read_sql("SELECT * FROM employees", con=connect)
print(companyInDataframe)
'''output:
   emp_no  birth_date first_name last_name gender  hire_date
0       1  1990-12-31       Chen      Andy      M  2018-01-01
1       2  1991-01-23        Lin     Helen      F  2018-01-01
'''
```

需先建立engine物件，to_sql()才能根據此engine將DataFrame寫入

是否寫入DataFrame中的Index，預設值為True

if_exists參數決定了若employees資料表存在時採取甚麼動作。
fail: 不採取動作（預設值）
replace: drop資料表，create資料表，新增DataFrame資料
append: 直接新增DataFrame資料

Group By In DataFrame_1

```
companyInDataframe = pandas.read_sql("SELECT * FROM employees", con=connect)

print(companyInDataframe)
'''output:
   emp_no  birth_date first_name last_name gender  hire_date  salary
0       1  1990-12-31       Chen      Andy      M  2018-01-01   28000
1       2  1991-01-23        Lin     Helen      F  2018-01-01   30000
...
(total have 15 record)
'''

print(companyInDataframe.groupby(["first_name", "gender"]).agg({"salary": "mean"}))
'''output:
                      salary
first_name gender
Chen       F       34800.000000
           M       29783.333333
Lin        F       33750.000000
           M       32666.666667
Wang       F       30283.333333
           M       30833.333333
'''
```

針對salary欄位使用「mean」aggregate function取得平均

72

----------QUIZ 3

7. Please describe the limitations of XML DTD (Document Type Definition). (10%)

8. What element is the variable $x bound to in XQuery 1 and XQuery 2, respectively? (10%)

XQuery 1.

```
FOR  $x  IN
     doc(www.company.com/info.xml)
     //employee [employeeSalary gt 70000]/employeeName
     RETURN  <res>  $x/firstName, $x/lastName  </res>
```

XQuery 2.

```
FOR  $x  IN
     doc(www.company.com/info.xml)/company/employee
     WHERE  $X/employeeSalary gt 70000
     RETURN  <res>  $x/employeeName/firstName,
             $x/emplyeeName/lastName  </res>
```