【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

|  |  |
| --- | --- |
| **\*4. PHP statement:**  **print uncords(strtolower (‘JOHN smith’))?** | -output :  John Smith. |
| -strtolower() | 字串改小(lower case) |
| -ucwords() | 字首改大(uppercases) |

|  |  |
| --- | --- |
| PHP Arrays Can be numeric or associative |  |
| **\*-"Numeric array"**  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%)

EMP_PROJ 
SSN PLOC.ATK>N 

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 |
| book z  title: • MongoDB: lhe Definitive Guide•,  authors: I •Kristina Chodorow•, •Mike Dirolf• I  published date; tsopateC2mo-09-24•b  pages: 216,  language: •English•,  publisher  name: •CYReilly Media",  founded: •1980•,  location: • CN  Embedding | publisher = 丨  name: ' 0 ' R Media'.  title; · M0n90D8 ; The Definitwe de'.  | •Kristina 0 。 ro , , •Mike Dirolf• |  d 一 d 』 e : 阝 00 e 「 10 • 4 ' J,  pages: 2 | |

|  |  |
| --- | --- |
| 【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)

ClientApplication 
Driver 
Writes Reads 
Secondary 
Primary 
Secondary 
Secondary 
Primary 
Heartbeat 
Secondary 
Election for New Primary 
Hea rtbeat 
Secondary 
Secondary 
New Primary Elected 
Re lication 
Primary 
Secondary 
Heartbeat 

Members

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

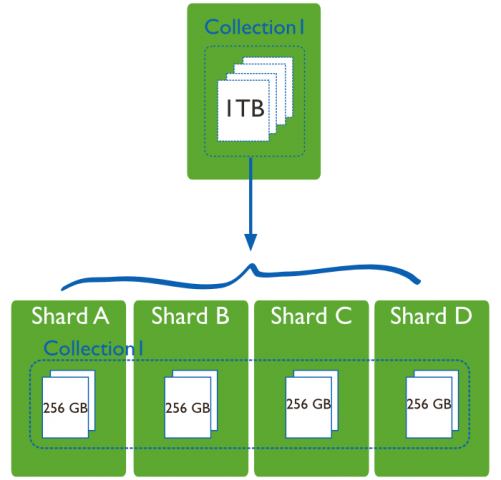
-Arbiter(Voting) (Can’t be primary)

-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.ensurelndex({ score: 1}) 
• Show existing indexes 
— db.users.getlndexes() 
• Drop index 
— db.users.droplndex( {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 |  |
| (3, 4, 5, 3, 4, 5, 5, 6, 8, 91  > newA = set(A)  > 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  np.matrix('1 2;  np.matrix('4 3;  . print (a*b)  [2e 13] ]  21') | 1 2 4 3  [3 4]x[2 1] |
|  |  |

|  |  |
| --- | --- |
| 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 |

IS coding: 
utf-8 - * _ 
2 import • mysql. connector • as DB 
3 
4 connect • • • DB 
. connect user-" our • account" 
assword•" 
our 
assword" 
• host-"127. 
1 # -*- coding: utf-8 -*- 
2 import • mysql . connector • as DB 
3 
4 config 
5. 
"user" • "your • account" 
" password" : 
"your password" , 
• • "host . " 
127 .e.e.l", 
8. 
• • • • • • •#"database": "database name" 
9 • 3306, 
10 • 
11 
12 connect DB. connect **confi 

Table 
12 connect • = • DB. 
13 cursor • = • connect. cursor() 
14 
cu rsor.execute(sq IString) 
15 sql z. 
"CREATE - DATABASE company; " 
HMySQLF$fi+ 
16 cursor. execute(sql) 
17 
18# set connecting database 
19 connect. database • "company" 
20 
21 sq1 
22 CREATE. TABLE employees* ( 
emp_no- INT(II) -NOT. NULL AUTO INCREMENT, 
23 • 
24 • 
birth date* -DATE 410TOUJLL, 
• first_name• .VARCHAR(14) -NOT -NULL, 
25 • 
last_name• .VARCHAR(16) • NOT. NULL, 
26 • 
gender* ENUM( 'M • , 
• F • ) .NOT• NULL, 
27 • 
28 • 
*hire date' •DATE-NOT. NULL, 
29 • .PRIMARY KEYC emp_no• ) 
30 ) ENGINE 
. InnoDB 
31 
32 cursor. execute(sql) 

NSERT I NT O 
164 # easy way to execute SQL, but may not consider in security issue 
165 addEmp10yeeByString• "INSERT INTO • employees (first_name, - last_name, - hire_date, 
gender, • birth_date) • VALUES ( 'Wang ' , 
• Andy • 
•w, .12, .31)). " 
166 
167 cursor. execute(addEmp10yeeByString) 
168 
169 # recommended way to execute SQL when you get parameter from user 
170 addEmp10yeeByParameter• — • "INSERT • INTO • employees • (first_name, • last_name, 
hire_date, gender, -birth_date) TVALIJES- (%s, v%s, • ; " 
("Lin", 
, • date(2018, .1, .1), 
.date(1991, .1, .23)) 
171 employeeData• = • 
• "Helen" 
172 
173 cursor. execute(addEmp10yeeByParameter, • employeeData) 

ae4 from • sqlalchemy • import • create_engine 
4e5 engine • create_engine( mysql : / [username : password@host:port/database • ) 
7 "first_name"] "Wang" 
• "first_name"].-. 
8 companyInDataframe. employees " , 
• con—engine, • • index—False) 
"Chen" 
410 companyInDataframe • • pandas 
411 print(conpanyInDataframe) 
412 • 
• •output: 
413 - • •emp_no 
. employees" , 
on—connect) 
last_name gender • 
•hire dat 
name 
414 w 
415 1 
416 
-2 
•birth date•first 
199e-12-31 
.1991 -el .23 
gine'VJf#: , 
Chen • 
• Lin• 
Andy 
Helen 
M. -2018-01. 
ADataFrame 
Index 
fiDataFrame 
employees 
fail: ) 
replace: create 
append: fi$Ffi%DataFrame 

401 e 
402 1 
483 
Group By In DataFrame_l 
396 companyInDataframe • pandas . read_sql( "SELECT v employees" , • con—connect) 
397 
398 print (companyInDataframe) 
34800. eeeeee 
, 3e283.333333 
. 3e833.333333 
399 •output: 
• • •emp_no 
birth_date first_name last_name gender 
1 
-2 
1990-12-31 
1991-01-23 
15 record) 
Chen 
- Lin 
Andy 
- Helen 
•hire date 
2e18-ø1-01 
, 2e18-ø1-01 
salary 
28øee 
3øøøø 
484 (total - have 
405 
406 
407 print(companyInDataframe.groupby(["first_name", "gender")) "mean } ) ) 
408 • • •output: 
409 • 
410 first_name gender • 
411 Chen • 
412 
413 Lim 
414 
415 Wang 
416 • 
4 
••••••salary 
29783. 333333 
33750. eeeeee 
32666.666667 
"-H salaryffåflLfLiH4 
mean aggregate 
fun ction 

------------------------------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 700001/employeeName 
RETURN <res> Sx/firstName, $x/lastName 
XQuery 2. 
FOR $x IN 
doc(www.company.com/info.xml)/company/employee 
WHERE SX/employeeSalary gt 70000 
RETURN $x/employeeName/firstName, 
$x/emplyeeName/lastName 