

勞動部產業新尖兵計畫

人工智慧金融應用與實務培訓班



課程模組： AI 金融科技課程 - 網路爬蟲技術

5.SQLite 資料庫

葉建華 (Yeh, Jian-hua)

tdi.jhyeh@tdi.edu.tw
au4290@gmail.com

講次內容

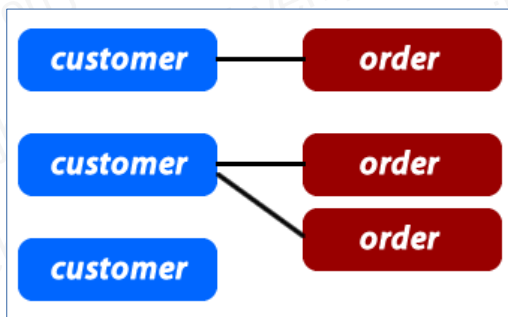
- 關聯式資料庫懶人包
- SQLite 介紹與資料庫連線
- 建立 SQLite 資料表
- 資料表的新增、刪除、修改
- 資料表的查詢

關聯式資料庫懶人包

- Database, DB
 - 資料庫，資料儲存的地方（資料表，Table）
 - 資料表之間可以有關聯（Relation）
- DBMS
 - 資料庫管理系統，DB Management System
- DBMS 可以輔助我們對 DB Table 做 新增、刪除、修改、查詢

資料表與關聯

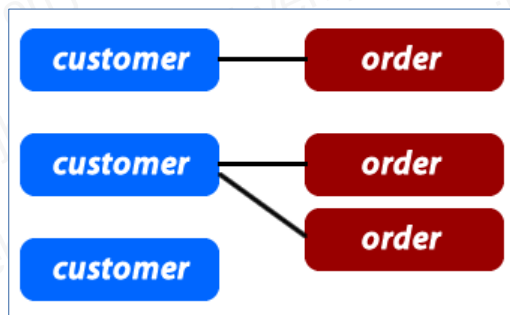
CUSTOMERS			
customer_id	customer_name		
101	John Doe		
102	Bruce Wayne		
ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00



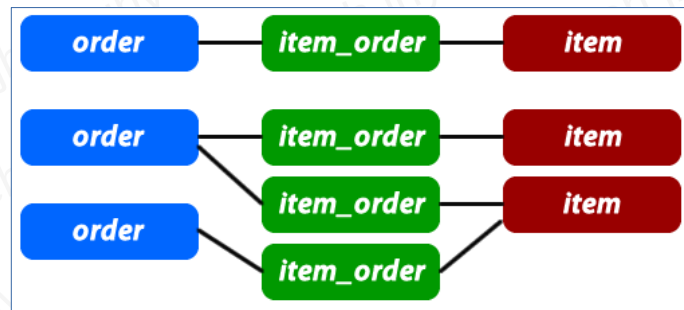
資料表與關聯

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00



ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
ITEMS			
item_id	item_name	item_description	
201	Tickle Me Elmo	It wants to be tickled	
202	District 9 DVD	Awesome sci-fi movie	
203	Batarang	It is very sharp	
ITEMS_ORDERS			
order_id	item_id		
555	201		
555	202		
556	202		
556	203		

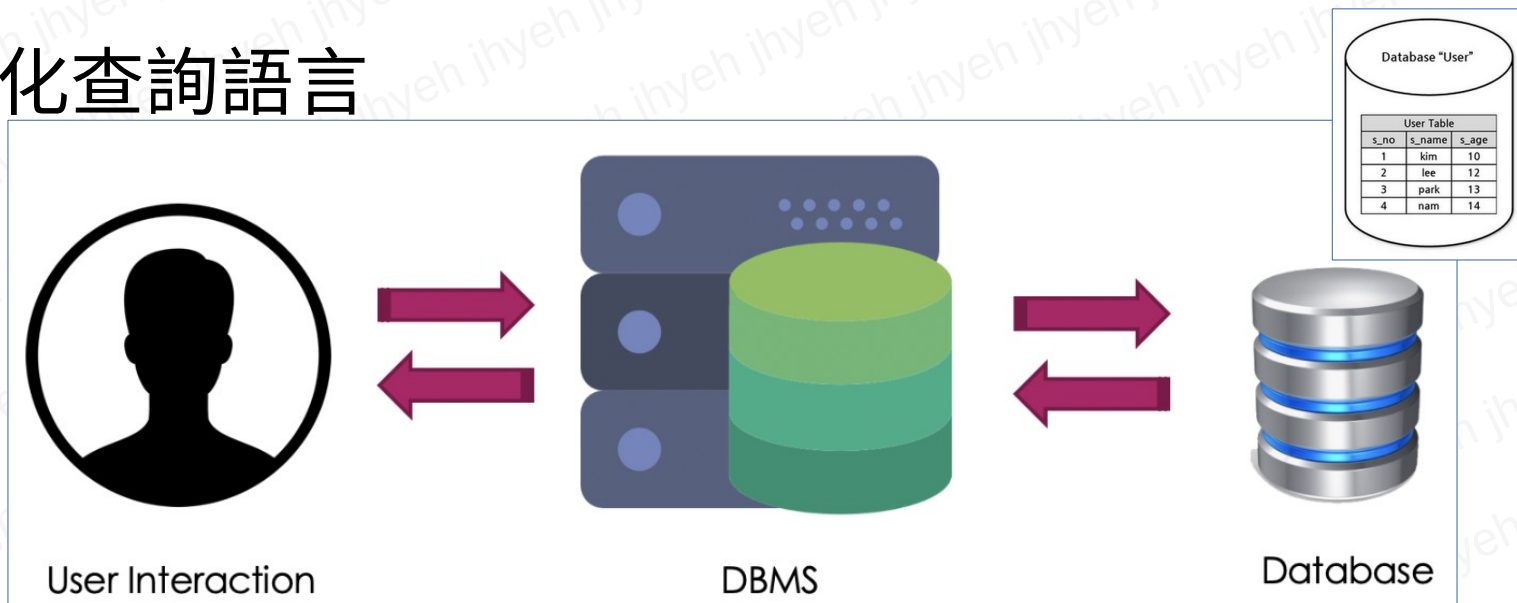


關聯式資料庫懶人包

- 那 SQL 是什麼?
 - Structural Query Language, SQL
 - 結構化查詢語言
 - 查詢 DB 的程式語言

關聯式資料庫懶人包

- 那 SQL 是什麼?
 - Structural Query Language, SQL
 - 結構化查詢語言
 - 查詢

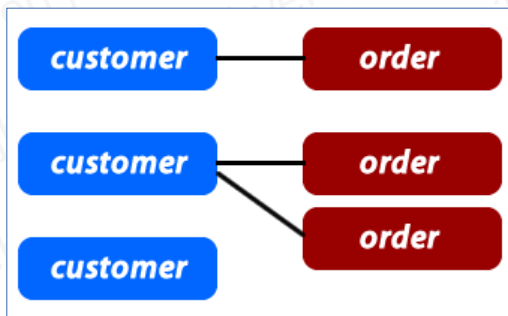


資料表、關聯與 SQL

CUSTOMERS			
customer_id	customer_name		
101	John Doe		
102	Bruce Wayne		
ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

```
CREATE TABLE customers (  
  customer_id INT AUTO_INCREMENT PRIMARY KEY,  
  customer_name VARCHAR(100)  
);
```

```
CREATE TABLE orders (  
  order_id INT AUTO_INCREMENT PRIMARY KEY,  
  customer_id INT,  
  amount DOUBLE,  
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```



SQL scripts!

講次內容

- 關聯式資料庫懶人包
- SQLite 介紹與資料庫連線
- 建立 SQLite 資料表
- 資料表的新增、刪除、修改
- 資料表的查詢

SQLite 介紹

- SQLite 是輕量化的 DBMS
- Python 3.x 版內建
- `import sqlite3`

連線到 SQLite

- 使用 SQLite，要先建立資料庫連線
 - 使用 connect() 方法
 - 取得連線物件
 - 連線用完要關閉

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)

# ... 資料庫操作...

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

講次內容

- 關聯式資料庫懶人包
- SQLite 介紹與資料庫連線
- 建立 SQLite 資料表
- 資料表的新增、刪除、修改
- 資料表的查詢

SQLite 支援的資料類型

資料類型	說明
NULL	空值，對應 None
INTEGER	整數， int
REAL	浮點數， float
TEXT	字串， str
BLOB	byte 區塊資料（二進位資料）

建立 SQLite 資料表

- connect() 物件的方法

connect() 物件方法	說明
close()	關閉連線
commit()	更新資料庫內容
cursor()	cursor 物件，資料庫「游標」，執行 SQL 敘述
execute()	執行資料庫指令，建立、新增、刪除、修改、查詢

建立 SQLite 資料表

- cursor() 物件的方法

cursor() 物件方法	說明
execute()	執行資料庫指令，建立、新增、刪除、修改、查詢

建立 SQLite 資料表

- 那就仿這個吧！

```
CREATE TABLE customers (  
    customer_id INT AUTO INCREMENT PRIMARY KEY,  
    customer_name VARCHAR(100)  
);
```

```
import sqlite3  
  
dbname = 'mytest.db'  
# 取得資料庫連線物件  
conn = sqlite3.connect(dbname)  
  
# ... 資料庫操作...  
sqlstr = 'CREATE TABLE customers (customer_id INTEGER, \  
        customer_name TEXT, PRIMARY KEY(customer_id))'  
conn.execute(sqlstr)  
print('資料表customers已建立')  
  
# 良好的習慣，用完資料庫要關閉連線  
conn.close()  
print('資料庫連線關閉：', dbname)
```

注意！！

1. 同名資料表不能重複建立
2. 要刪除資料表，sqlstr 請用 'DROP TABLE customers'

CREATE/DROP TABLE

```
CREATE TABLE table_name(  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    .....  
    columnN datatype,  
    PRIMARY KEY( one or more  
columns )  
);
```

```
DROP TABLE table_name;
```

講次內容

- 關聯式資料庫懶人包
- SQLite 介紹與資料庫連線
- 建立 SQLite 資料表
- 資料表的新增、刪除、修改
- 資料表的查詢

資料表新增資料紀錄

- 新增 John Doe
- 怎麼做？

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

資料表新增資料紀錄

- 新增 John Doe

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)

# ...資料庫操作...
sqlstr = 'INSERT INTO customers (customer_id, customer_name) values (101, "John Doe")'
conn.execute(sqlstr)
conn.commit()
print(sqlstr)
print('資料表customers紀錄已新增')

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

練習：資料表新增資料紀錄

- 新增 Bruce Wayne
- 怎麼做？

<u>CUSTOMERS</u>			
customer_id	customer_name		
101	John Doe		
102	Bruce Wayne		
<u>ORDERS</u>			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

練習：資料表新增資料紀錄

- 新增 Bruce Wayne

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)

# ...資料庫操作...
sqlstr = 'INSERT INTO customers (customer_id, customer_name) values (102, "Bruce Wayne")'
conn.execute(sqlstr)
conn.commit()
print(sqlstr)
print('資料表customers紀錄已新增')

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

INSERT TABLE RECORD

```
INSERT INTO table_name(column1, column2, ...columnN)  
VALUES (value1, value2, ...valueN);
```

資料表刪除資料紀錄

- 刪除 customer id 102
- 怎麼做？

<u>CUSTOMERS</u>			
customer_id	customer_name		
101	John Doe		
102	Bruce Wayne		
<u>ORDERS</u>			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

資料表刪除資料紀錄

- 刪除 cus
- 怎麼做?

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)
cur = conn.cursor()

# ... 資料庫操作...
sqlstr = 'DELETE FROM customers WHERE customer_id=102'
results = cur.execute(sqlstr)
conn.commit()
print(sqlstr)
if cur.rowcount==1:
    print('資料表customers紀錄已刪除')
else:
    print('資料表customers紀錄刪除失敗')

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

CUSTOMERS

name		
ne		
id	order_date	amount
101	12/24/09	\$156.78
102	12/25/09	\$99.99
101	12/26/09	\$75.00

DELETE TABLE RECORD

```
DELETE FROM table_name  
WHERE {CONDITION};
```

資料表修改資料紀錄

- customer id 101 姓名改為 John Smith
- 怎麼做？

<u>CUSTOMERS</u>			
customer_id	customer_name		
101	John Doe		
102	Bruce Wayne		
<u>ORDERS</u>			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

資料表修改資料紀錄

CUSTOMERS

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)
cur = conn.cursor()

# ... 資料庫操作...
sqlstr = 'UPDATE customers SET customer_name="John Smith" WHERE customer_id=101'
results = cur.execute(sqlstr)
conn.commit()
print(sqlstr)
if cur.rowcount==1:
    print('資料表customers紀錄已修改')
else:
    print('資料表customers紀錄修改失敗')

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

amount
\$156.78
\$99.99
\$75.00

UPDATE TABLE RECORD

UPDATE table_name

SET column1=value1, column2=value2, ...columnN=valueN

[WHERE CONDITION];

講次內容

- 關聯式資料庫懶人包
- SQLite 介紹與資料庫連線
- 建立 SQLite 資料表
- 資料表的新增、刪除、修改
- 資料表的查詢

資料表查詢資料紀錄

- 你可以列出資料表中儲存的資料紀錄
- 怎麼做？

CUSTOMERS	
customer_id	customer_name
101	John Doe
102	Bruce Wayne

ORDERS			
order_id	customer_id	order_date	amount
555	101	12/24/09	\$156.78
556	102	12/25/09	\$99.99
557	101	12/26/09	\$75.00

資料表查詢資料紀錄

- 你可以列料紀錄
- 怎麼做？

```
import sqlite3

dbname = 'mytest.db'
# 取得資料庫連線物件
conn = sqlite3.connect(dbname)
cur = conn.cursor()

# ... 資料庫操作...
sqlstr = 'SELECT * FROM customers'
results = cur.execute(sqlstr)
conn.commit()
for rec in results:
    print('customer id:', rec[0], end=', ')
    print('name:', rec[1])

# 良好的習慣，用完資料庫要關閉連線
conn.close()
print('資料庫連線關閉：', dbname)
```

CUSTOMERS

d	customer_name
01	John Doe
02	Bruce Wayne

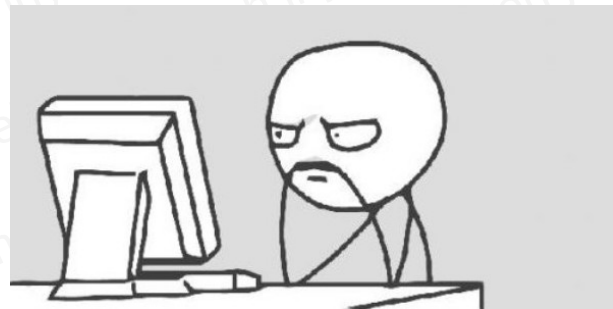
customer id: 101, name: John Smith
customer id: 102, name: Bruce Wayne
資料庫連線關閉: mytest.db

SELECT TABLE RECORD

```
SELECT column1, column2, ...columnN  
FROM   table_name  
[ WHERE CONDITION ];
```

後記

- 更多 SQLite 說明請查詢網路教學
 - 例: <https://www.runoob.com/sqlite/sqlite-python.html>
- 本講次只是關聯式資料庫的一個小小角落
- 關聯式資料庫是門複雜的學問
- 在大學裡是一門課！！
 - 18 週課程，每週三小時！！



這個講次中，你應該學到了 ...

- 認識關聯式資料庫
- 認識 SQLite
- 在 SQLite 中進行資料表的建立與刪除
- 在 SQLite 中進行資料表紀錄的新增、刪除、修改、查詢