

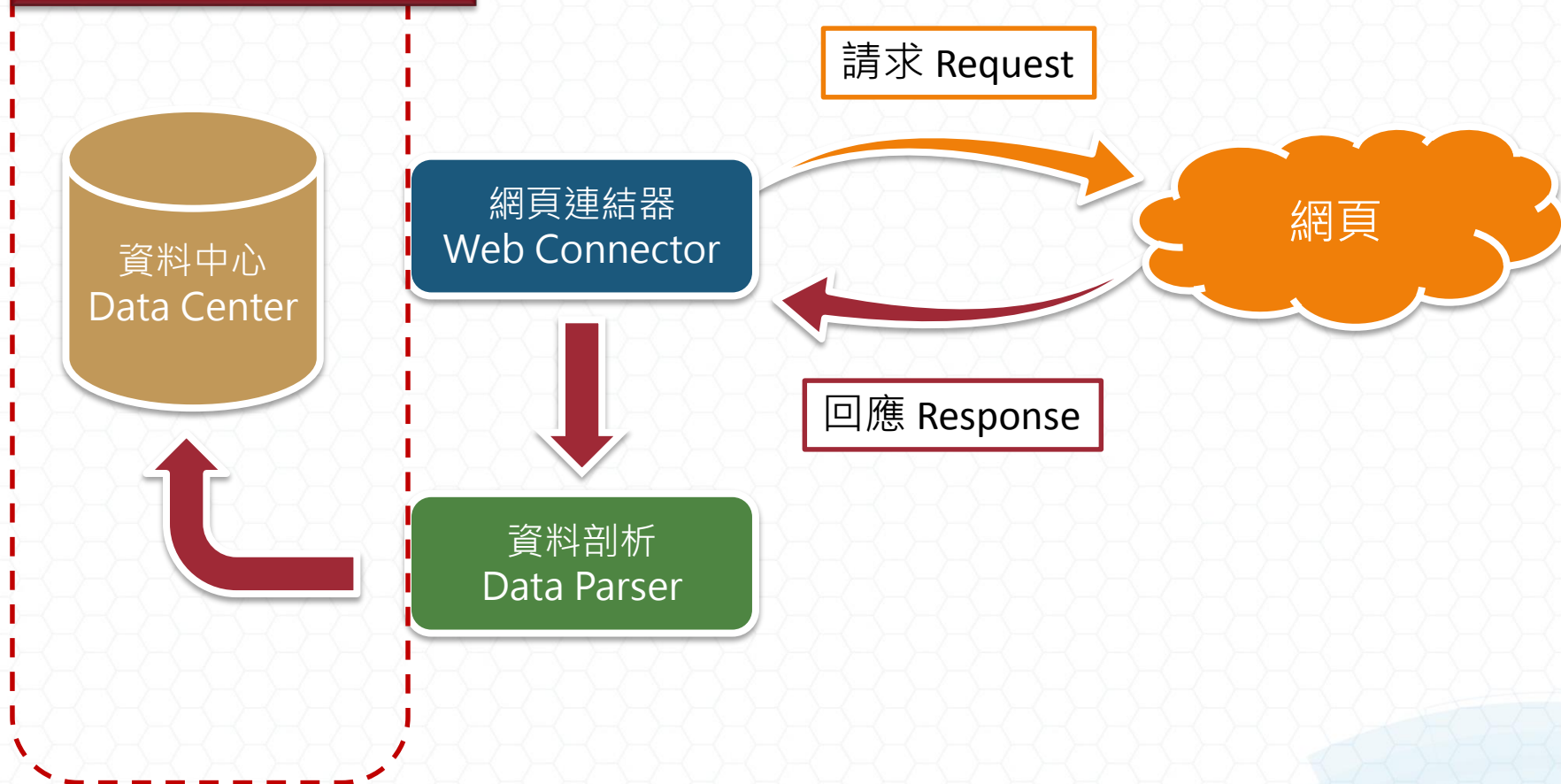
# 數據分析師假日精修班

## Lab4

David Chiu  
2016/07/24

# 網路爬蟲架構

如何將爬取到的  
文字資料放到資料庫中



# 資料儲存



# 關聯式資料庫

- 安全存儲、管理資料
  - 有效管理磁碟上的資料
- 保持資料的一致性
  - ACID 四原則
- 可以透過標準模型整合資料
  - 使用SQL 操作資料



# 最受歡迎的開源資料庫



## 網頁服務解決方案

Linux  
Apache  
MySQL  
PHP

# 以往如何存放資料?

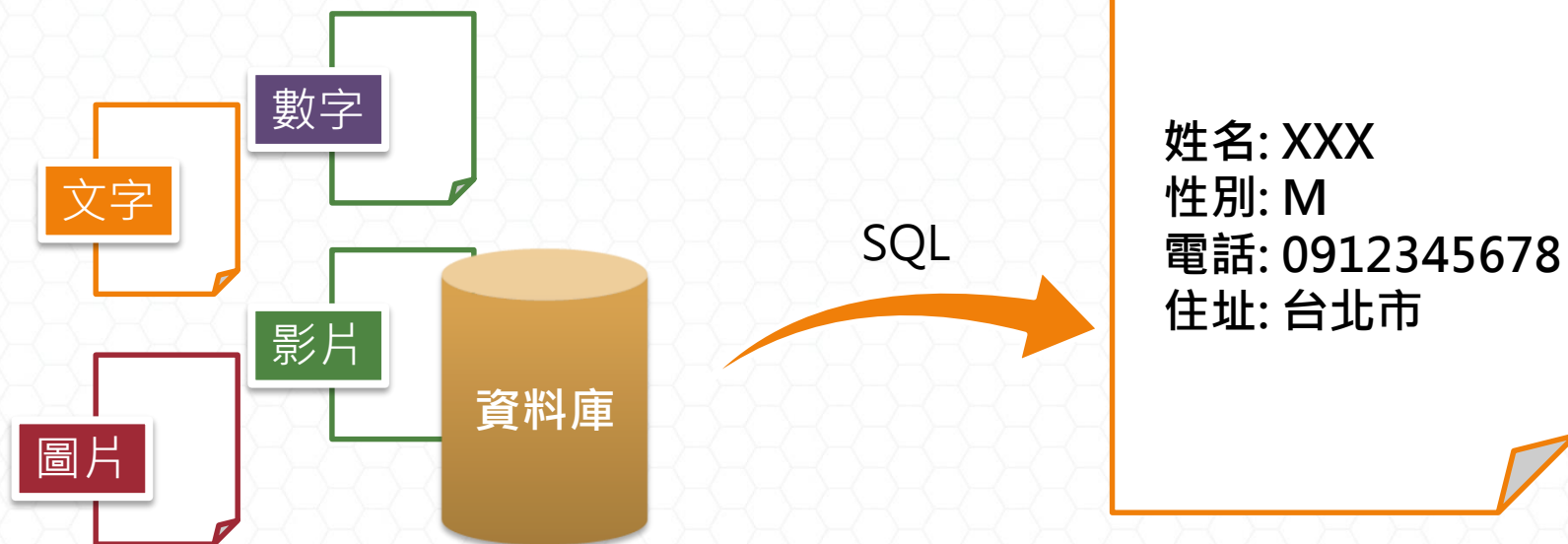


姓名: XXX  
性別: M  
電話: 0912345678  
住址: 台北市

但該怎麼搜尋資料?

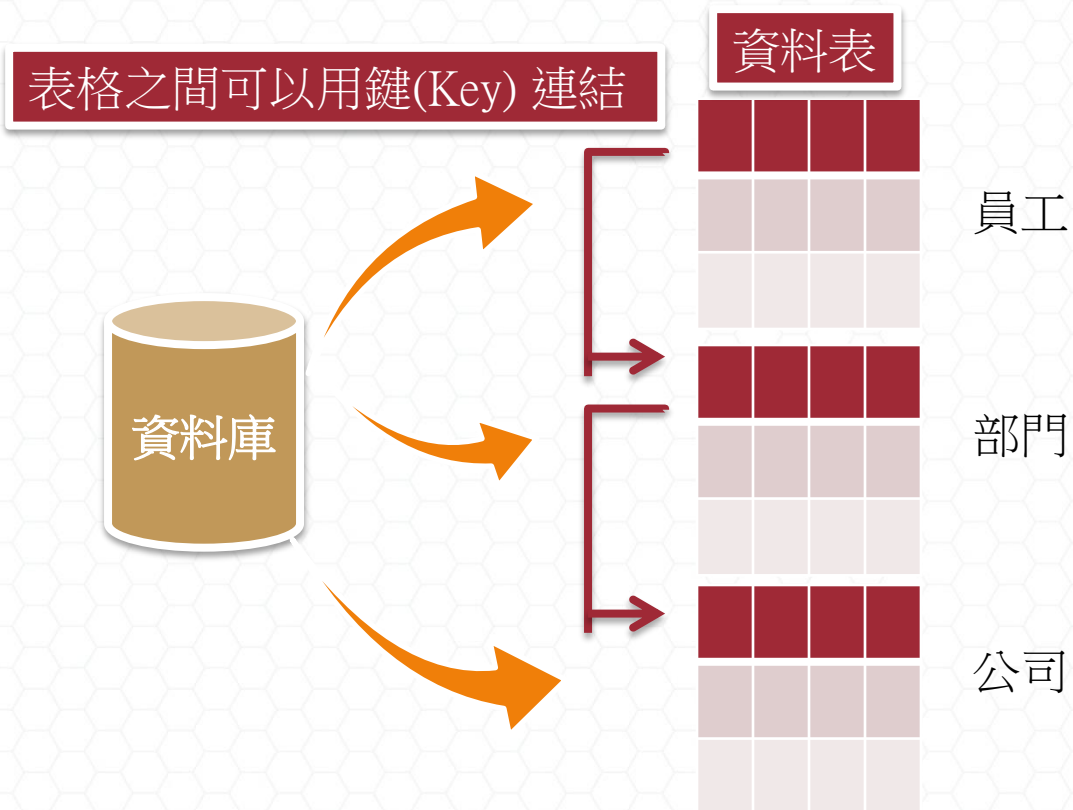
# 資料庫

- 將資料以結構化方式做存儲，讓使用者可以透過結構化查詢語言(Structured Query Language, 簡稱SQL)快速取用及維護資料





# 資料庫中含多個資料表





# 資料表包含的內容

主鍵	id	名字	性別	住址	欄位名稱
列	1	John	M	Taipei	資料內容
	2	Mary	F	Tainan	

# 資料庫的ACID

- 不可分割性 (Atomicity)
  - 交易必須全部完成或全部不完成
  - (e.g. 轉帳)
- 一致性 (Consistency)
  - 交易開始到結束，資料完整性都符合既設規則與限制
  - (e.g. 帳號)
- 隔離性 (Isolation)
  - 並行的交易不會引響彼此
  - (e.g. 餘額查詢)
- 持久性 (Durability)
  - 進行完交易後，對資料庫的變更會永久保留在資料庫
  - (e.g. 系統毀損)

# 安裝MySQL 資料庫




# 下載MySQL 5.7

■ <http://dev.mysql.com/downloads/mysql/>

## MySQL Installer 5.7 for Windows

**All MySQL Products. For All Windows Platforms.  
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.

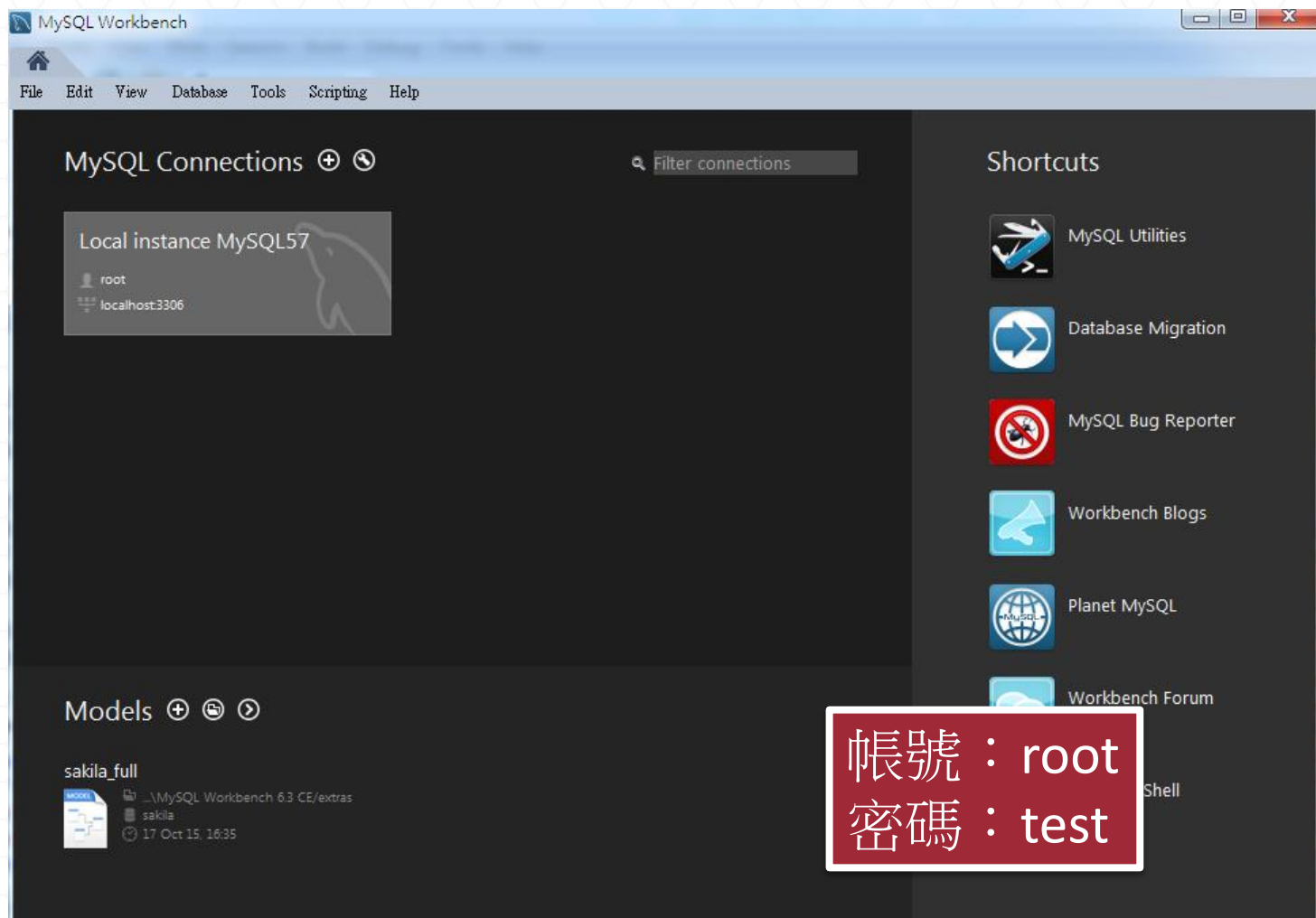


**Windows (x86, 32-bit), MySQL Installer MSI** [Download](#)

Other Downloads:

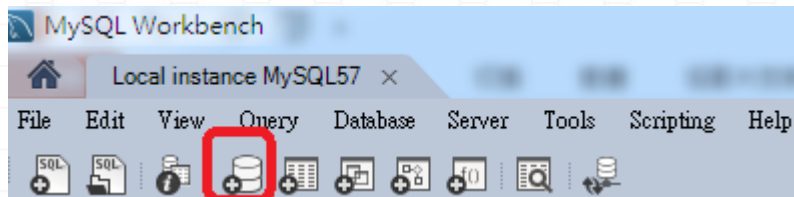
<b>Windows (x86, 32-bit), ZIP Archive</b> (mysql-5.7.11-win32.zip)	5.7.11	323.5M	<a href="#">Download</a>
<b>Windows (x86, 64-bit), ZIP Archive</b> (mysql-5.7.11-winx64.zip)	5.7.11	336.7M	<a href="#">Download</a>

# 使用SQL Workbench 操作資料庫

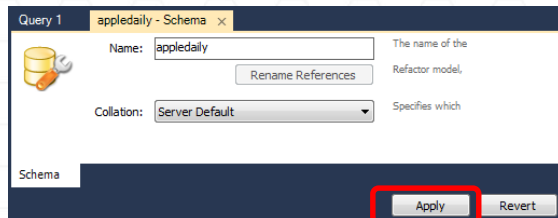


# 建立資料庫

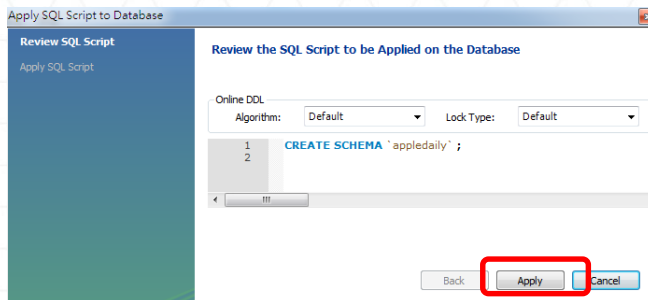
## ■ 點選資料庫圖示



## ■ 填入資料庫名稱



## ■ 建立資料庫





# 使用R 連結MySQL

## ■ RMySQL

- 透過DBI Interface 連結MySQL
- `install.packages("RMySQL")`
- `library(RMySQL)`

## ■ RODBC

- 透過ODBC 連結MySQL
- `install.packages("RODBC")`
- `library(RODBC)`

## ■ RJDBC

- 透過JDBC 連結MySQL
- `install.packages("RJDBC")`
- `library(RJDBC)`

先下載JDK 以及將jvm.dll 所在的位置  
加到path 以後再安裝RJDBC

# 下載JDK

- <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

## Java SE Development Kit 8u101

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.77 MB	<a href="#">jdk-8u101-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.72 MB	<a href="#">jdk-8u101-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	160.28 MB	<a href="#">jdk-8u101-linux-i586.rpm</a>
Linux x86	174.96 MB	<a href="#">jdk-8u101-linux-i586.tar.gz</a>
Linux x64	158.27 MB	<a href="#">jdk-8u101-linux-x64.rpm</a>
Linux x64	172.95 MB	<a href="#">jdk-8u101-linux-x64.tar.gz</a>
Mac OS X	227.36 MB	<a href="#">jdk-8u101-macosx-x64.dmg</a>
Solaris SPARC 64-bit	139.66 MB	<a href="#">jdk-8u101-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	98.96 MB	<a href="#">jdk-8u101-solaris-sparcv9.tar.gz</a>
Solaris x64	140.33 MB	<a href="#">jdk-8u101-solaris-x64.tar.Z</a>
Solaris x64	96.78 MB	<a href="#">jdk-8u101-solaris-x64.tar.gz</a>
Windows x86	188.32 MB	<a href="#">jdk-8u101-windows-i586.exe</a>
Windows x64	193.68 MB	<a href="#">jdk-8u101-windows-x64.exe</a>

下載Windows x86 版本

# 將jvm.dll 的位置加進PATH 之中

## ■ 將jvm.dll 的位置加到PATH之中

□ e.g. C:\Program Files\Java\jre1.8.0\_91\bin\server

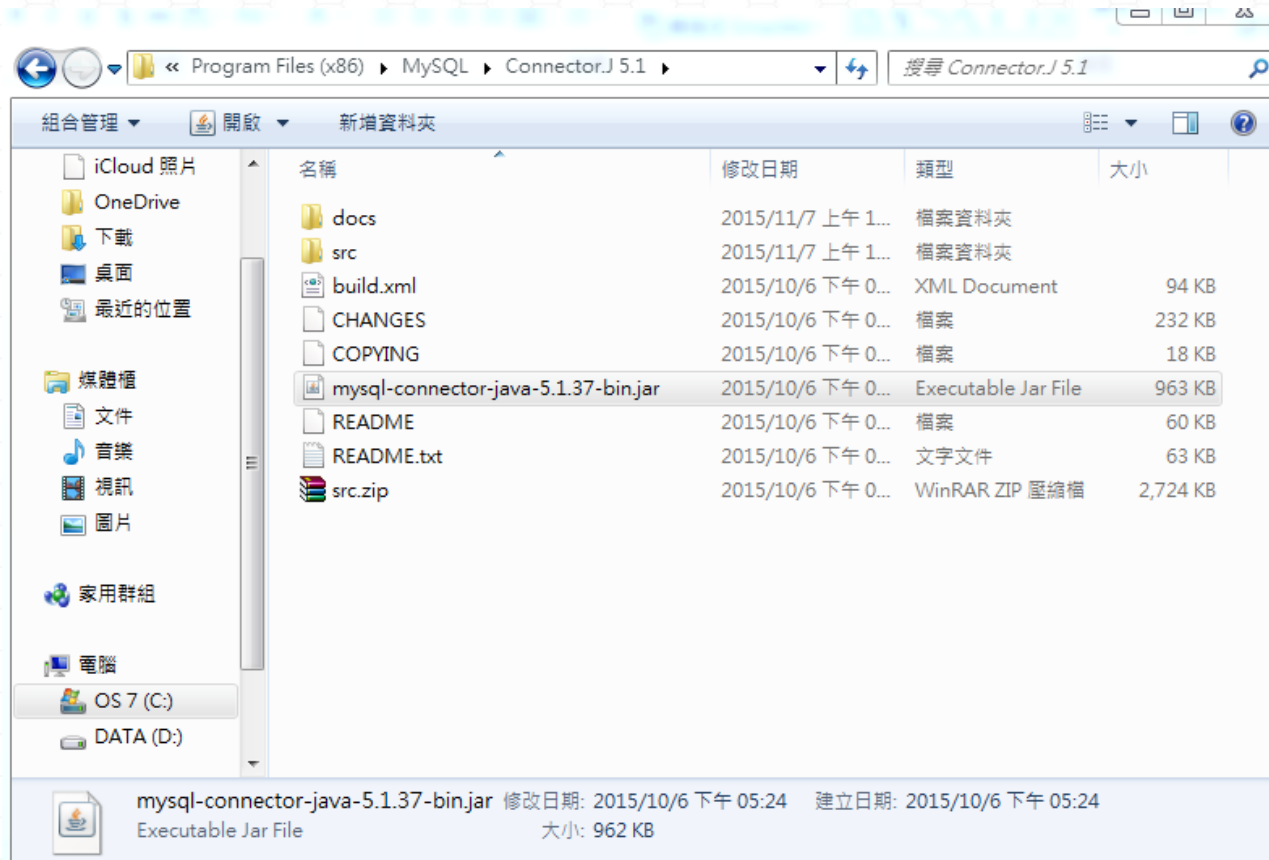


使用; 分隔路徑



# 找到JDBC 的JAR檔

## ■ C:\Program Files (x86)\MySQL\Connector.J 5.1



# 完整的範例操作

```
library(RJDBC)
jar.loc <- 'C:\\Program Files (x86)\\MySQL\\Connector.J 5.1\\mysql-connector-java-5.1.37-bin.jar'
drv <- JDBC("com.mysql.jdbc.Driver",
            jar.loc,
            identifier.quote="`)")
conn <- dbConnect(drv, "jdbc:mysql://localhost/appliedaily", "root",
                  "test")

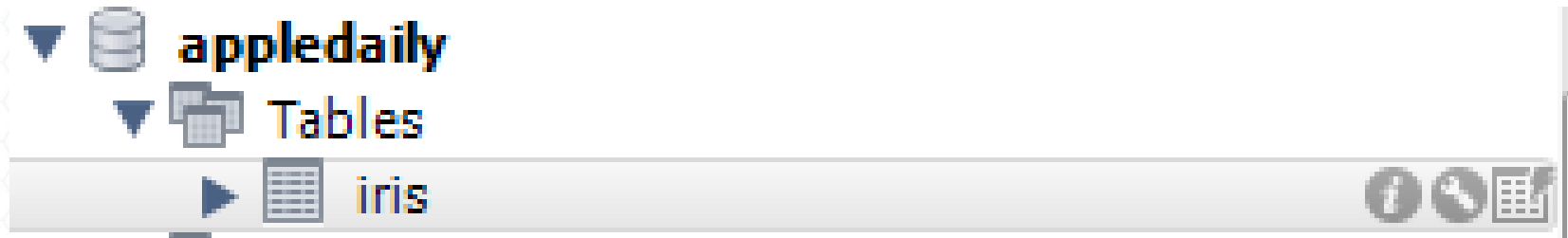
dbDisconnect(conn)
```

# 建立、表列、讀取表格資訊

```
dbWriteTable(conn, "iris", iris)
```

```
dbListTables(conn)
```

```
dbReadTable(conn, "iris")
```





# 使用Query 查詢數據

```
res <- dbGetQuery(conn, "SELECT * FROM iris")  
res
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

# 捨棄已存在表格

```
if(dbExistsTable(conn,'iris')){  
    dbRemoveTable(conn, 'iris')  
}  
dbListTables(conn)
```

# 讀取蘋果新聞資料

## ■ 下載新聞資料

```
download.file('https://github.com/ywchiu/  
rtibame/raw/master/Data/applenews.Rdata',  
'applenews.RData')
```

## ■ 讀取新聞資料

```
load('news.RData')
```

## ■ 儲存新聞資料

```
save(file='applenews.RData', x=applenews)
```



# 插入整理完的蘋果新聞表格

dbWriteTable(conn, "applenews", applenews)

dbListTables(conn)

dbReadTable(conn, "applenews")

Column	Type	Default Value	Nullable	Character Set
◇ article	varchar(255)		YES	utf8
◇ title	varchar(255)		YES	utf8
◇ dt	varchar(255)		YES	utf8
◇ category	varchar(255)		YES	utf8
◇ clicked	varchar(255)			
◇ view_cnt	varchar(255)			
◇ content	varchar(255)		YES	utf8
◇ content_clean	varchar(255)		YES	utf8
◇ dt1	varchar(255)		YES	utf8

預設建立字串長度255的欄位

# 建立新聞資料表

## ■ 表格名稱

□ news\_main

## ■ 欄位名稱

□ id

□ Title

□ category

□ view\_cnt

□ content

□ dt

```
CREATE TABLE `news_main` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `title` varchar(1000) DEFAULT NULL,  
  `category` varchar(10) DEFAULT NULL,  
  `view_cnt` int(11) DEFAULT NULL,  
  `content` text,  
  `dt` datetime DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

# 使用Append 新增資料

## ■ 寫入資料

```
dbWriteTable(conn, 'news_main', applenews,  
append=TRUE,row.names=FALSE,overwrite=FALSE)
```

overwrite 需設為FALSE

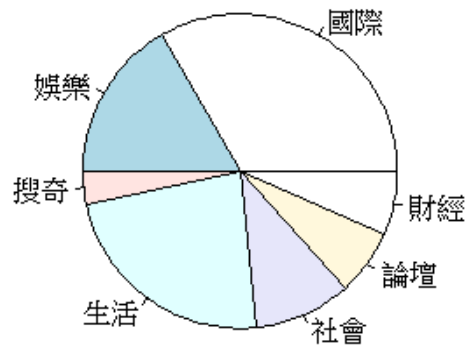
## ■ 讀取表格資料

```
dbReadTable(conn, 'news_main')
```



# 使用SQL 作基本的資料分析

```
res <- dbGetQuery(conn, "SELECT category,  
count(*) FROM news_main group by category")  
res  
names(res) = c('category', 'cnt')  
pie(res$cnt, labels = res$category)
```



# 使用dplyr 做資料分析

# 敘述性統計分析

- 多數資料分析，80% 在於如何加總與平均
  - 銷售份額
  - 客戶數量
  - 業績成長量
- 用SQL做敘述性統計
  - `select * from tb1 where col1 >= 100 limit 3`
- R有類似的工具嗎？
  - `plyr`
  - `reshape2`
  - `dplyr`





# 如何操作資料

- 關於操作資料，你需要
  - 可以分割資料(Split)
  - 可以轉換資料(Transformation)
  - 可以聚合資料(Aggregation)
  - 可以探索資料(Exploration)
- 需要如同SQL的語法操作



# 為什麼要使用dplyr

- 提供操作資料的基本語法
  - filter, select, arrange, mutate, summarise, group\_by
- 提供資料合併功能(JOIN)
  - Inner join, left join
- 可以操作資料表(data table) 或資料庫 (Database) 的資料

# 安裝與使用dplyr

## ■ 安裝dplyr

- `install.packages("dplyr")`

## ■ 使用dplyr

- `library(dplyr)`

## ■ 觀看說明頁

- `help(package='dplyr')`

# 過濾資料

## ■ 原先R 提供的過濾功能

```
applenews[applenews$category == "娛樂",]
```

## ■ dplyr 的過濾功能

```
applenews$dt <- as.POSIXct(applenews$dt)  
filter(applenews, category == "娛樂")
```

# 可以使用 AND, OR 與 IN 來過濾資料

- 找出娛樂以及點閱數超過1000的新聞

```
filter(applenews, category == "娛樂" & view_cnt > 1000)
```

- 找出娛樂或點閱數超過1000的新聞

```
filter(applenews, category == "娛樂" | view_cnt > 1000)
```

- 找出娛樂與社會新聞

```
filter(applenews, category %in% c("娛樂", "社會"))
```



# 選擇欄位

- 原先R 提供的欄位選取

```
applenews[, c("category","view_cnt")]
```

- dplyr 的欄位選取

```
select(applenews,category,view_cnt)
```

# 但如果我想同時選擇欄位又過濾資料呢？

## ■ 鏈接(Chaining)

- %>% (Then)

- 來自 magrittr

## ■ 使用Then (%>%)

applenews %>%

select(category,view\_cnt) %>%

filter(category == "社會")

# 資料做排序

- 使用Arrange 可以將資料做排序

```
applenews %>%
```

```
  select(category,view_cnt) %>%
```

```
    filter(category == "社會") %>%
```

```
    arrange(view_cnt)
```

- 由大到小排序 (desc)

```
applenews %>%
```

```
  select(category,view_cnt) %>%
```

```
    filter(category == "社會") %>%
```

```
    arrange(desc(view_cnt))
```

如同

```
SELECT category, view_cnt  
FROM applenews  
WHERE category = "社會"  
ORDER BY view_cnt
```

# 新增欄位 (mutate)

## ■ 計算總和

```
freqsum = applenews %>%  
  select(view_cnt) %>%  
  sum()
```

## ■ 使用mutate 新增欄位

```
applenews %>%  
  select(title, category, view_cnt) %>%  
  mutate(portion= view_cnt/freqsum)
```

## ■ 儲存新欄位

```
applenews = applenews %>% mutate(portion=  
view_cnt/freqsum)
```



# 分組計算 (group\_by, summarise)

## ■ 分組計算函式

- group\_by: 分組依據
- summarise: 依組別計算結果

## ■ 統計各新聞點閱數的總和

```
applenews %>%
```

```
  group_by(category) %>%
```

```
  summarise(view_sum = sum(view_cnt,  
na.rm=TRUE))
```

# 統計多個欄位

- 使用summarise\_each 統計portion 與 view\_cnt的總和

```
applenews %>%
```

```
  group_by(category) %>%
```

```
  summarise_each(funs(sum), view_cnt, portion)
```

# 針對多個欄位做統計

- 可針對不同資料同時做統計

```
applenews %>%
```

```
  group_by(category) %>%
```

```
  summarise_each(funs(min(., na.rm=TRUE), max(.,  
na.rm=TRUE)), matches("view_cnt"))
```

# 資料計數

## ■ 一般計數

```
applenews %>%  
  select(category) %>%  
  summarise_each(funs(n()))
```

## ■ 不重複計數

```
applenews %>%  
  select(category) %>%  
  summarise_each(funs(n_distinct(category)))
```



# 使用直方圖顯示新聞點閱總和

## ■ 取得統計數

```
cat_stat = applenews %>%
```

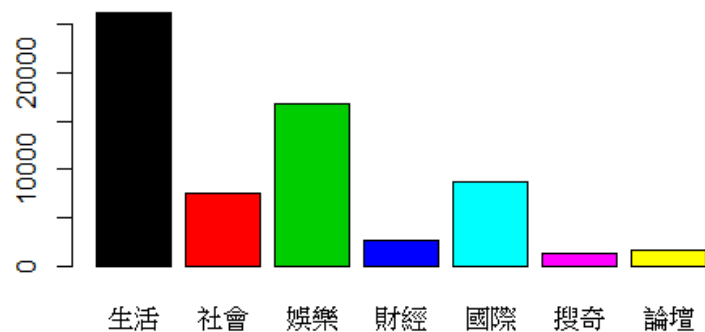
```
  group_by(category) %>%
```

```
  summarise(view_sum = sum(view_cnt))
```

```
cat_stat$category = as.factor(cat_stat$category)
```

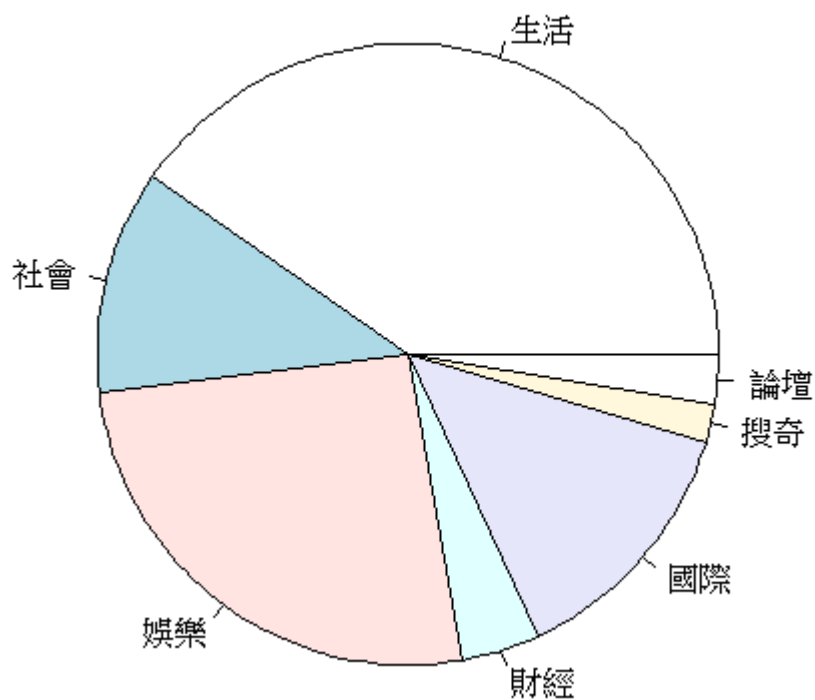
## ■ 繪圖

```
barplot(cat_stat$view_sum, names.arg=cat_stat$category,  
col=cat_stat$category)
```



# 使用圓餅圖顯示新聞點閱比例

```
pie(cat_stat$view_sum, label = cat_stat$category)
```



The background features a light gray hexagonal grid pattern. Overlaid on this is a large, faint, light blue circular graphic composed of several concentric rings. These rings are not solid but are made of segments, creating a spiral or tunnel-like effect that draws the eye towards the center. The overall aesthetic is clean, modern, and technical.

**THANK YOU**