**SQL基礎語法**

* 創建一個table employee

Create Table Store\_Information

(Store\_Name char(50),

Txn\_Date datetime2,

Sales integer);

* 加資料進去table

INSERT INTO Store\_Information ("Store\_Name","Txn\_Date","Sales")

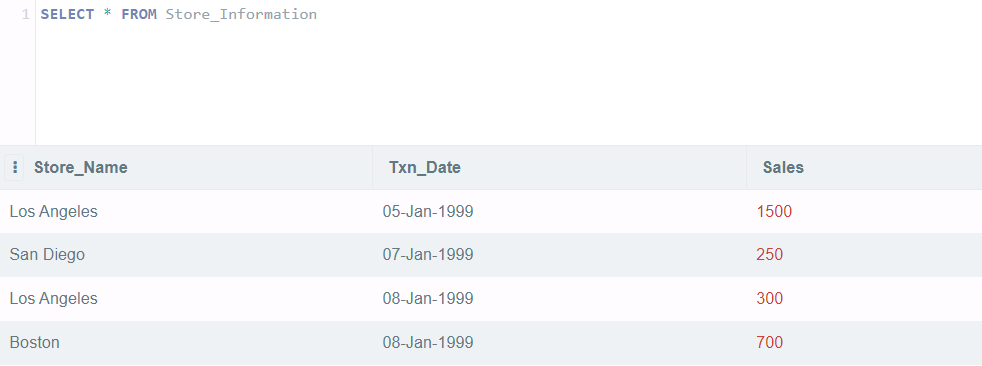
VALUES ("Los Angeles","05-Jan-1999",1500),

("San Diego","07-Jan-1999",250),

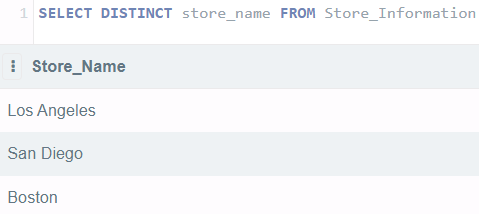
("Los Angeles","08-Jan-1999",300),

("Boston","08-Jan-1999",700);

* Table結果



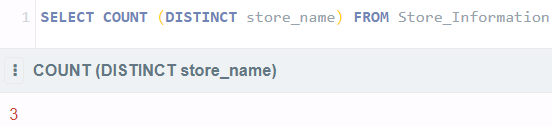
* Distinct (取唯一值)



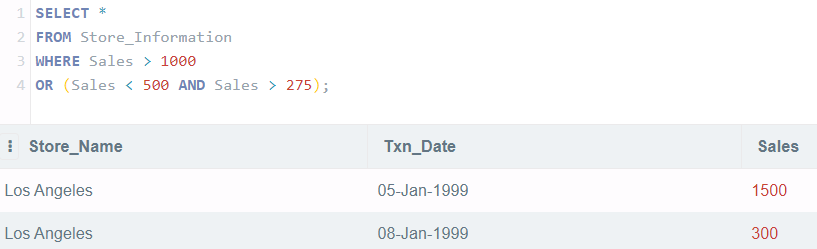
* Count



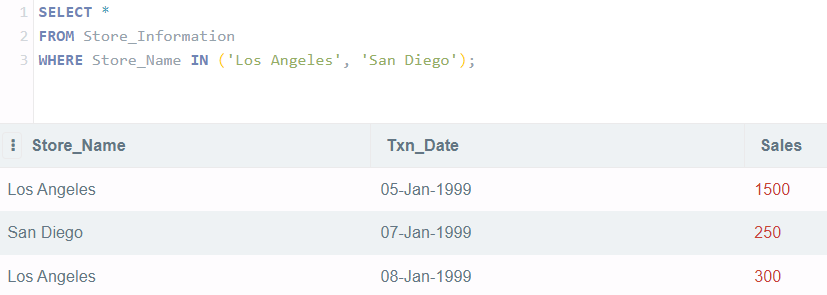
* Distinct with Count



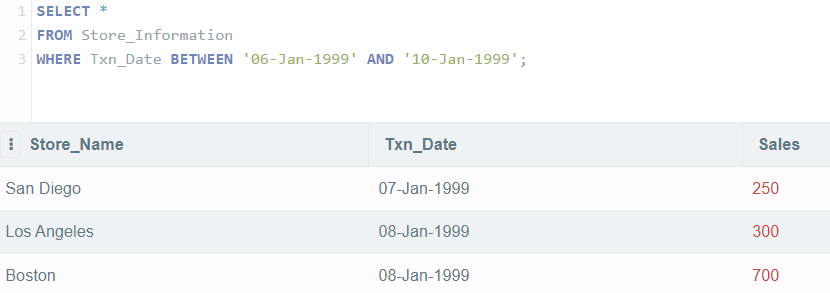
* Where AND OR



* Where IN (完全不合字串)/ NOT IN (完全不符合字串)



* Between(運用一個範圍 (range) 內抓出資料庫中的值)



* SQL萬用字元

有的時候，我們需要依照由字串模式中找出相符的資料。要滿足這個需求，我們就需要用到萬用字元 (wildcard) 的做法。SQL 中有兩個萬用字元：

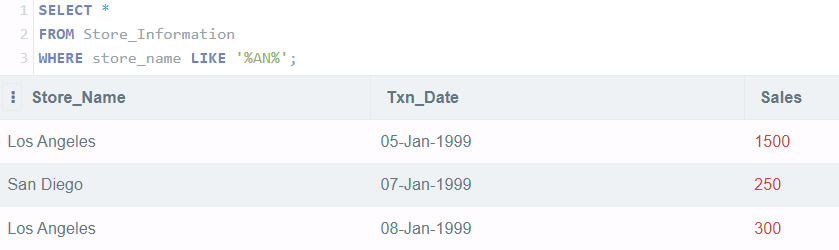
% (百分比符號)：代表零個、一個、或數個字母。

\_ (底線)：代表剛好一個字母。

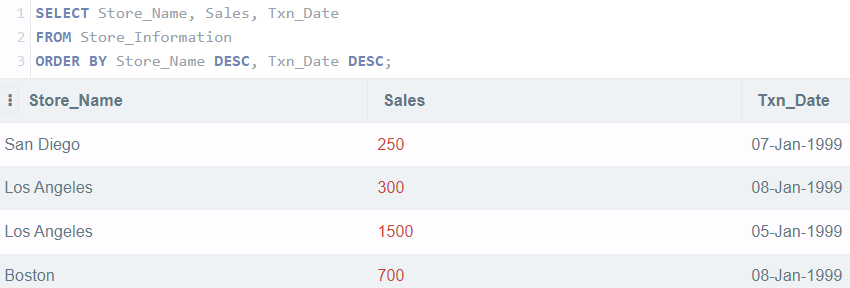
萬用字元是與 LIKE 關鍵字一起使用的。

以下是幾個萬用字元的例子：

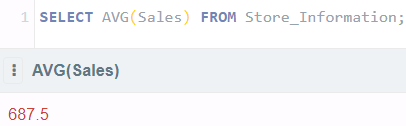
* 'A\_Z': 所有以 'A' 起頭，另一個任何值的字原，且以 'Z' 為結尾的字串。 'ABZ' 和 'A2Z' 都符合這一個模式，而 'AKKZ' 並不符合 (因為在 A 和 Z 之間有兩個字元，而不是一個字元)。
* 'ABC%': 所有以 'ABC' 起頭的字串。舉例來說，'ABCD' 和 'ABCABC' 都符合這個模式。
* '%XYZ': 所有以 'XYZ' 結尾的字串。舉例來說，'WXYZ' 和 'ZZXYZ' 都符合這個模式。
* '%AN%': 所有含有 'AN'這個模式的字串。舉例來說， 'LOS ANGELES' 和 'SAN FRANCISCO' 都符合這個模式。
* '\_AN%'： 所有第二個字母為 'A' 和第三個字母為 'N' 的字串。舉例來說，'SAN FRANCISCO' 符合這個模式，而 'LOS ANGELES' 則不符合這個模式。
* LIKE (上面有解說可參考)



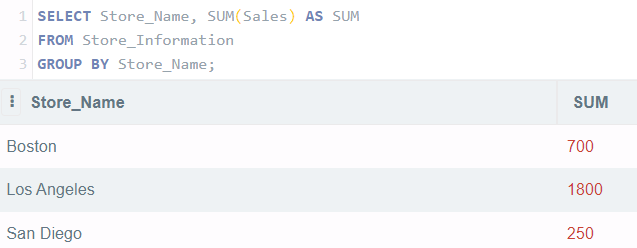
* Order by (最後也可以用數字1,2…表示變數，如果照兩個變數排序，兩個都要寫DESC/ASC，預設是ASC)



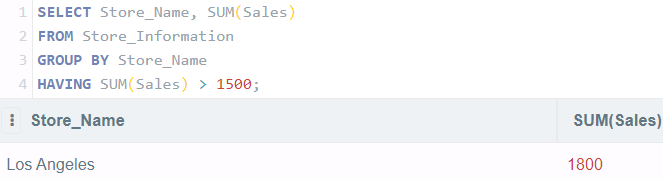
* AVG/COUNT/MAX/MIN/SUM (方式都一樣)



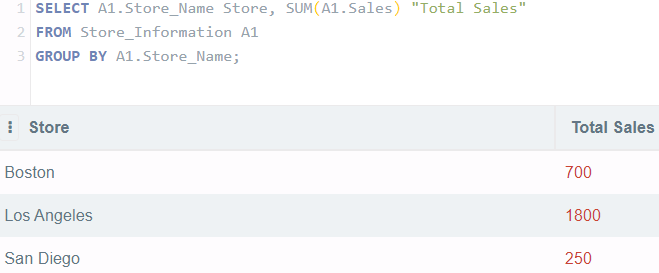
* GROUP BY



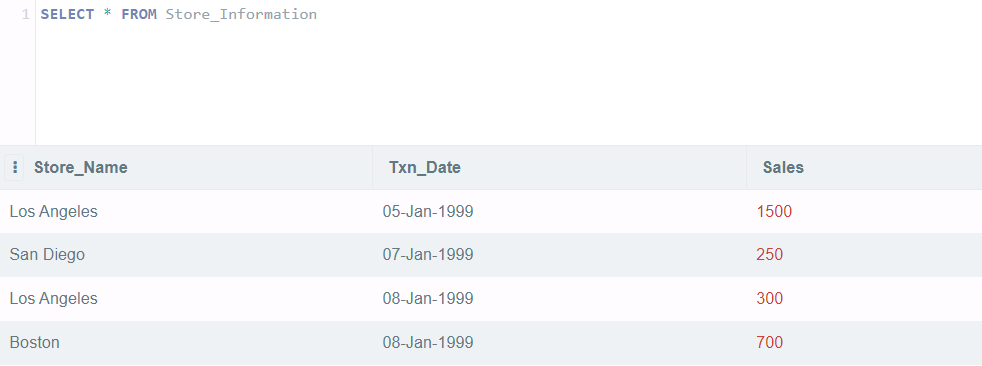
* HAVING



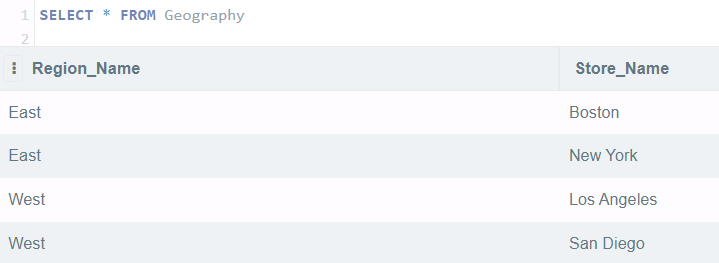
* 別名 (或加as)



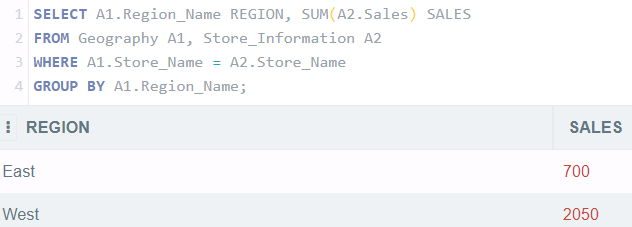
Table\_1



Table\_2:



* inner/left join (先把兩表格Store\_Name一樣地找出還後再做Groupby)



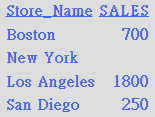
* outer join

SELECT A1.Store\_Name, SUM(A2.Sales) SALES

FROM Georgraphy A1, Store\_Information A2

WHERE A1.Store\_Name = A2.Store\_Name (+)

GROUP BY A1.Store\_Name;



* Concatenate

SELECT Region\_Name || ' ' || Store\_Name FROM Geography

WHERE Store\_Name = 'Boston';



* Substring



* Trim



* Length



* Replace

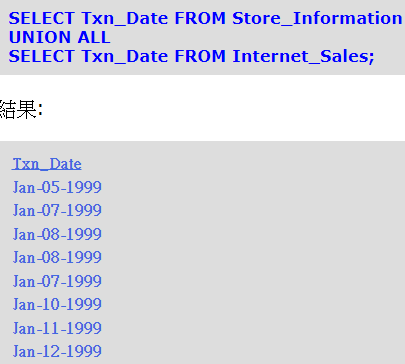


**SQL 進階語法**

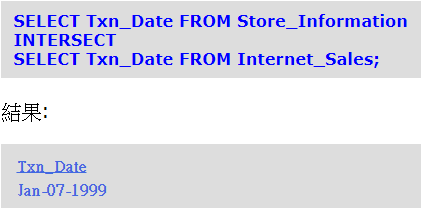
* UNION(重複取唯一)



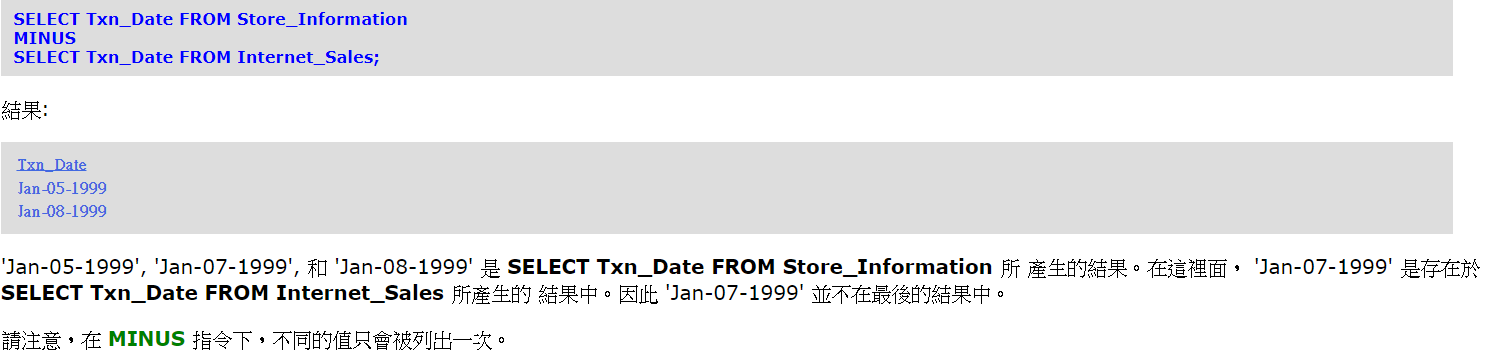
* Union all (重複會列出)



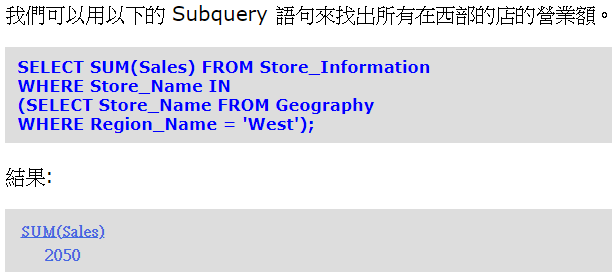
* Intersect(都有的列出來)



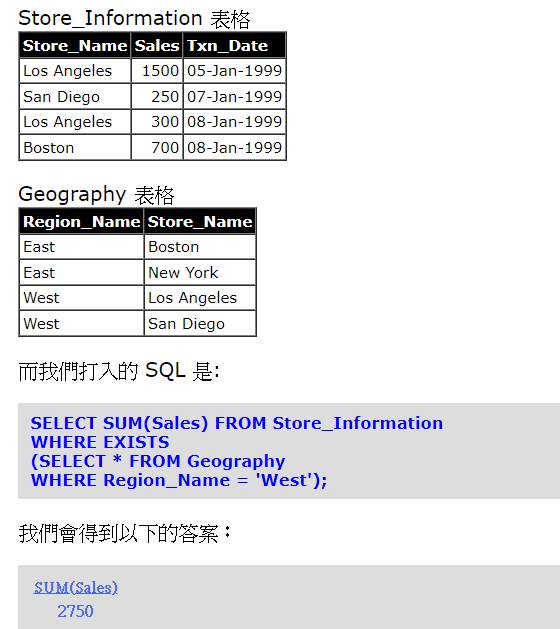
* Minus (以左方的表格為主，取差值的唯一值)



* 子查詢



* EXISTS(用來測試內查詢有沒有產生任何結果。如果有的話，系統就會執行外查詢中的 SQL。若是沒有的話，那整個 SQL 語句就不會產生任何結果。)



* Case(decode有異曲同工之處)



* 算排名 (From同一個表格會變成最下面那張表)

