SQL 查詢總是先從SELECT開始的嗎?

SQL數據庫開發 今天

SQL專欄

SQL基礎知識第二版

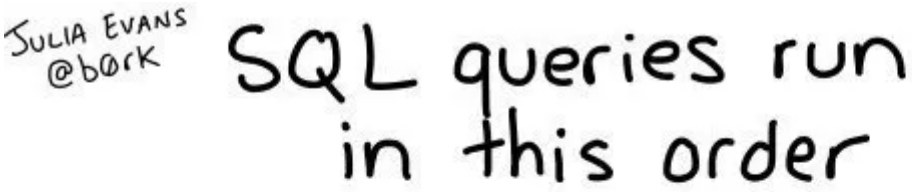
很多SQL 查詢都是以SELECT 開始的。不過，最近我跟別人解釋什麼是窗口函數，我在網上搜索”是否可以對窗口函數返回的結果進行過濾“這個問題，得出的結論是”窗口函數必須在WHERE 和GROUP BY 之後，所以不能”。

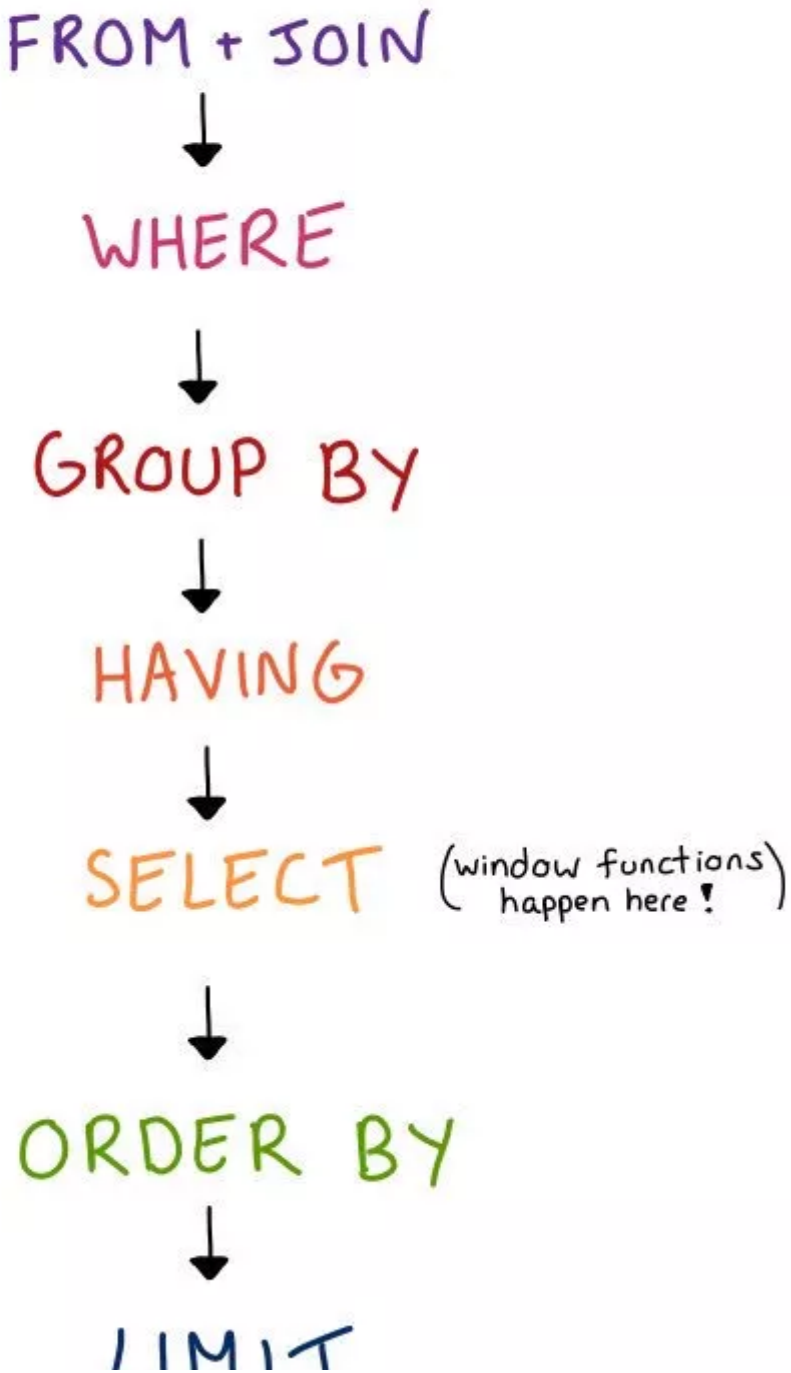
於是我又想到了另一個問題：

好像這個問題應該很好回答，畢竟自己已經寫了上萬個SQL 查詢了，有一些還很複雜。但事實是，我仍然很難確切地說出它的順序是怎樣的。

SQL 查詢的執行順序

於是我研究了一下，發現順序大概是這樣的。SELECT 並不是最先執行的，而是在第五個。




FROM + JOIN
WHERE
GROUP BY
HAVING
SELECT
(window functions happen here!)
ORDER BY
LIMIT

這張圖回答了以下這些問題

這張圖與SQL 查詢的語義有關，讓你知道一個查詢會返回什麼，並回答了以下這些問題：

- 可以在GRROUP BY 之後使用WHERE 嗎？（不行，WHERE 是在GROUP BY 之後！）
- 可以對窗口函數返回的結果進行過濾嗎？（不行，窗口函數是SELECT 語句裡，而SELECT 是在WHERE 和GROUP BY 之後）
- 可以基于 GROUP BY 里的东西进行 ORDER BY 吗？（可以，ORDER BY 基本上是在最后执行的，所以可以基于任何东西进行 ORDER BY）
- LIMIT 是在什么时候执行？（在最后！）

但数据库引擎并不一定严格按照这个顺序执行 SQL 查询，因为为了更快地执行查询，它们会做出一些优化，这些问题会在以后的文章中解释。

所以：

- 如果你想要知道一个查询语句是否合法，或者想要知道一个查询语句会返回什么，可以参考这张图；
- 在涉及查询性能或者与索引有关的东西时，这张图就不适用了。

混合因素：列别名

有很多 SQL 实现允许你使用这样的语法：

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name, count(*)
FROM table
GROUP BY full_name
```

从这个语句来看，好像 GROUP BY 是在 SELECT 之后执行的，因为它引用了 SELECT 中的一个别名。但实际上不一定要这样，数据库引擎可以把查询重写成这样：

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name, count(*)
FROM table
GROUP BY CONCAT(first_name, ' ', last_name)
```

这样 GROUP BY 仍然先执行。

数据库引擎还会做一系列检查，确保 SELECT 和 GROUP BY 中的东西是有效的，所以会在生成执行计划之前对查询做一次整体检查。

数据库可能不按照这个顺序执行查询（优化）

在实际当中，数据库不一定会按照 JOIN、WHERE、GROUP BY 的顺序来执行查询，因为它们会进行一系列优化，把执行顺序打乱，从而让查询执行得更快，只要不改变查询结果。

这个查询说明了为什么需要以不同的顺序执行查询：

```
SELECT * FROM
owners LEFT JOIN cats ON owners.id = cats.owner
WHERE cats.name = 'mr darcy'
```

如果只需要找出名字叫“mr darcy”的猫，那就没必要对两张表的所有数据执行左连接，在连接之前先进行过滤，这样查询会快得多，而且对于这个查询来说，先执行过滤并不会改变查询结果。

数据库引擎还会做出其他很多优化，按照不同的顺序执行查询，不过我并不是这方面的专家，所以这里就不多说了。

LINQ 的查询以 FROM 开头

LINQ（C#和 VB.NET 中的查询语法）是按照 FROM…WHERE…SELECT 的顺序来的。这里有一个 LINQ 查询例子：

```
var teenAgerStudent = from s in studentList
             where s.Age > 12 && s.Age < 20
             select s;
```

pandas 中的查询也基本上是这样的，不过你不一定要按照这个顺序。我通常会像下面这样写 pandas 代码：

```
df = thing1.join(thing2)    # JOIN
df = df[df.created_at > 1000] # WHERE
df = df.groupby('something', num_yes = ('yes', 'sum')) # GROUP BY
df = df[df.num_yes > 2]     # HAVING，对 GROUP BY 结果进行过滤
df = df[['num_yes', 'something1', 'something']] # SELECT，选择要显示的列
df.sort_values('sometthing', ascending=True)[:30] # ORDER BY 和 LIMIT
df[:30]
```

这样写并不是因为 pandas 规定了这些规则，而是按照 JOIN/WHERE/GROUP BY/HAVING 这样的顺序来写代码会更有意义些。不过我经常会先写 WHERE 来改进性能，而且我想大多数数据库引擎也会这么做。

作者 | Julia Evans　译者 | 无明

infoq.cn/article/Oke8hgilga3PTZ3gWvbg

推荐阅读

- 国产数据库建模工具，看到界面第一眼，良心了！
- 机房布线的最高境界 ，最后的暗黑系，真是亮瞎眼
- 人脸识别的时候，一定要穿上衣服啊！否则...
- 终于有人把「内卷」和「努力」的区别讲明白了
- 二本学生连发10篇SCI直博香港城大，被质疑「灌水」，本人回应！