DB HW4 Report

1. 組員：賴暐倫、賴威仁、張彧豪
2. 貢獻程度
3. 各題說明：
4. 第一題

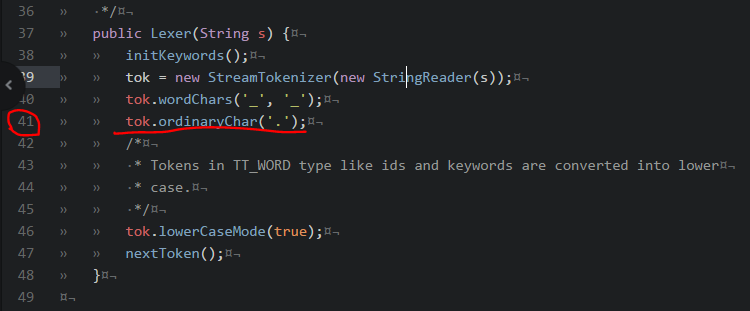
SQL：select certification, count(film\_id) from film group by certification

1. 大致流程：

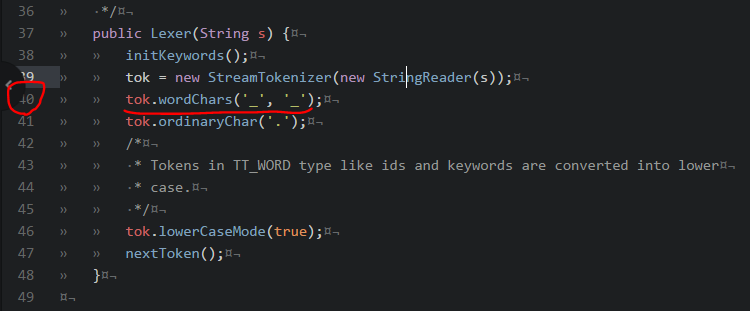
此query會被當成string傳入planner，planner會先將query string做拆解以及解析此query，再來做verify的動作，確認此query沒有語意錯誤，最後才做plan的動作

1. Lexer

lexer是query string第一個會來到的地方，在這邊lexer會以空白為間隔符號，把string給一個一個切開，比較特別的地方是，vanilladb不支援talbe\_name.table\_attribute，是因為這個java寫法的關係：



第41行的意思是會把點當成普通的字元並將它當成一個single-character token，所以會導致talbe\_name.table\_attribute這種語法無法執行，因為table\_name跟table\_attribute會被分成兩個token；取而代之的語法是tablename\_tableattribute，這種語法會被接受是因為這個java寫法的關係：



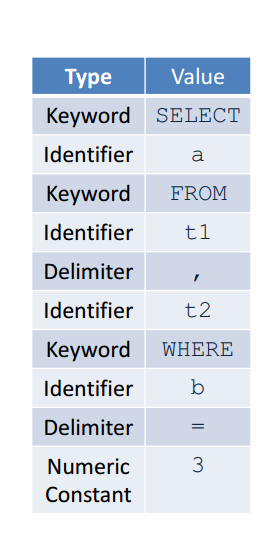
第40行的意思是把底線當作跟一般英文字母一樣的字元，所以遇到底線時不會被拆成兩個token，而是一個完整的token，所以tablename\_tableattribute這種語法可以正確執行。

這邊有個小發現是，每個被割開的token，java有定義不同的描述(description)來描述一個token，像是一般的tablename、keyword之類的都是TT\_WORD，而數字就是TT\_NUMBER。而第46行的lowerCaseMode(true)指的是會把屬於TT\_WORD的token都轉成小寫。Lexer會一直讀直到遇到eof為止。

依照上面的講法，此題的query string會被切成：select, certification, “,”, count(film\_id), from, film, group, by, certification這幾個token。

1. Parser

Parser做的事情就是將剛剛lexer割開的token蒐集起來，要對每個token去決定它到底是什麼樣的token(跟前面的TT\_WORD不一樣，這裡是我們定義有什麼樣的token)。從vanilladb的文件中可以知道，一個token有它對應的type跟value，value就是token的內容，而type則vanilladb自己定義的，如下圖所示：



一個query string則是由Planner此檔案內第57行呼叫位於Parser內的queryCommand指令，一開始會先把關鍵字給parse出來且建立projectlist跟table set，其中projectlist就是要select的attribute name或是aggregation function，而table set就是此query需要的table。

接下來就是要將where後面的條件給選出來，選出來是利用一個叫做predicate的函數，會把所有的term(可以想做就是一個條件)放在一個較過Predicate的物件裡面。

再來要做的是判斷有沒有使用group by，如果有的話就要把要group by的attribute給找出來，放在groupFields這個物件中。

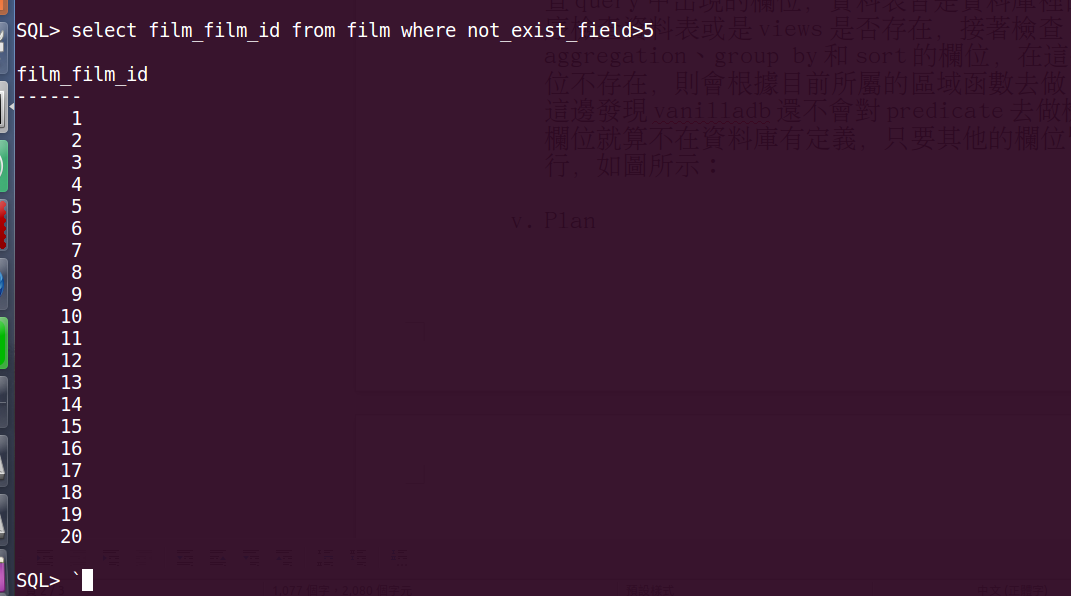
最後判斷有沒有使用order by，如果有的話，會將要拿來sort的attribute找出來放在fieldList物件內，而對於每個要sort的attribute，會有對應的方向(ASC,DESC)，方向資訊則會方在directionList這個物件內。

到這邊大致是queryCommand函數執行完畢，主要回傳的參數有select list、tables、predicate、groupFields、aggregation function和sortFields及sortDirs，這些參數即是QueryData物件所需的參數Planner。

1. Verifier

做完parse之後，Planner會在將傳回的QueryData物件傳給Verifier去做語意的檢查，這邊的檢查與parser不同的是這邊做的是語意的檢查，檢查query中出現的欄位，資料表皆是資料庫裡面有定義的。Verifier會依序檢查資料表或是views是否存在，接著檢查select的欄位，接著檢查aggregation、group by和sort的欄位，在這些檢查當中若發現有某項欄位不存在，則會根據目前所屬的區域函數去做throw exception的動作。

這邊發現vanilladb還不會對predicate去做檢查，所以where後面條件的欄位就算不在資料庫有定義，只要其他的欄位皆對的話，query還是可以執行，如圖所示︰



1. Plan

確定語意正確後，就可以來執行plan。Planner會呼叫createPlan這個函數來進行創立plan的動作。vanilladb有兩種創立plan的方式，一種是basic create plan，這種方法就是最原始的方式，不做任何優化去創立tablePlan、productPlan(多張table合在一起)、selectPlan(predicate的plan)、groupPlan(如果有的話)、projectPlan(select list的plan)和sortPlan(如果有的話)，最後則是如果有使用explain的話，那還要創立explainPlan。

第二種plan的方式是Heuristic plan，這種方式有點像是try-error的方法，每次算一種組合的cost，以此選出最小cost的plan，像是productPlan時，就是每次拿目前最佳的組合和目前的table來做組合，看是否這種組合是否cost有比目前最佳組合還要小，如果是的話它新的最佳組合就換給別人，以此做下去就可以選出最佳組合。

1. Algebra