Introduction to Database HW4(Final Project) Report

一　組員：0416324胡安鳳　0316323薛世恩

二　貢獻度：各佔三分之一

三

Query2（胡安鳳）:select team\_team\_id,count(player\_player\_id) from team,player where

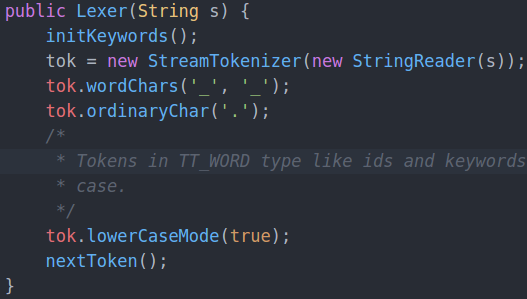
　　team\_team\_id=player\_team\_id and category='badminton' group by

　　team\_team\_id

這個query是用來查詢對於所有打羽毛球的球隊中，其隊伍中的球員總數。

當我們輸入的query其實是被當作一個很長的字串載入sql processor 中

●step1首先以空格切開來研究每一個子字串的，交由parse/Lexer執行，有點類似利用c++中StringStream(sstream)函式庫中切割的方式，同樣都是利用string token(deliminator)這種東西來剖析，如下圖



tok就是用來切割字串的識別符號。其中wordchar這句代表說字詞之間只能用底線連結或是點號連結，例如player\_id , player.name之類命名法則

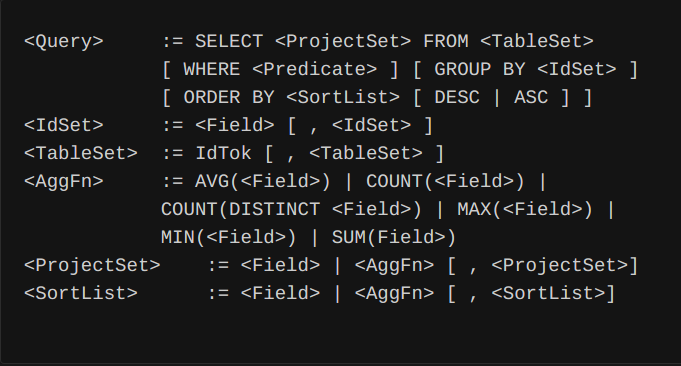
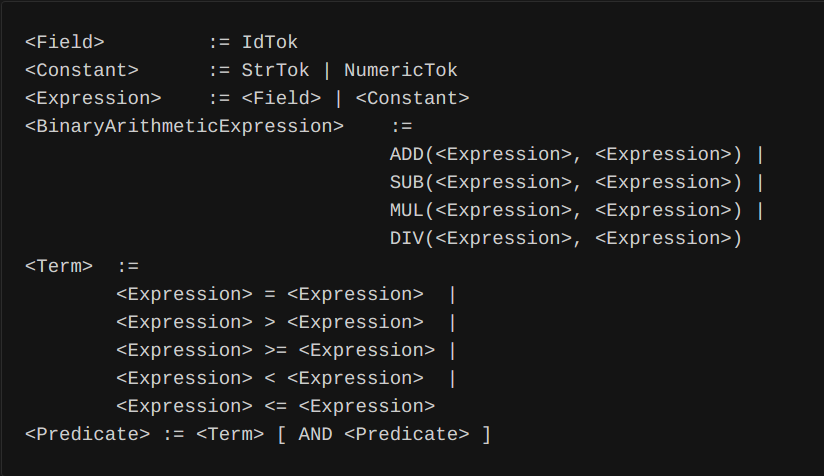
，其他一律切割。最後將tok一律在小寫的ascii code以免混淆，造成錯誤。

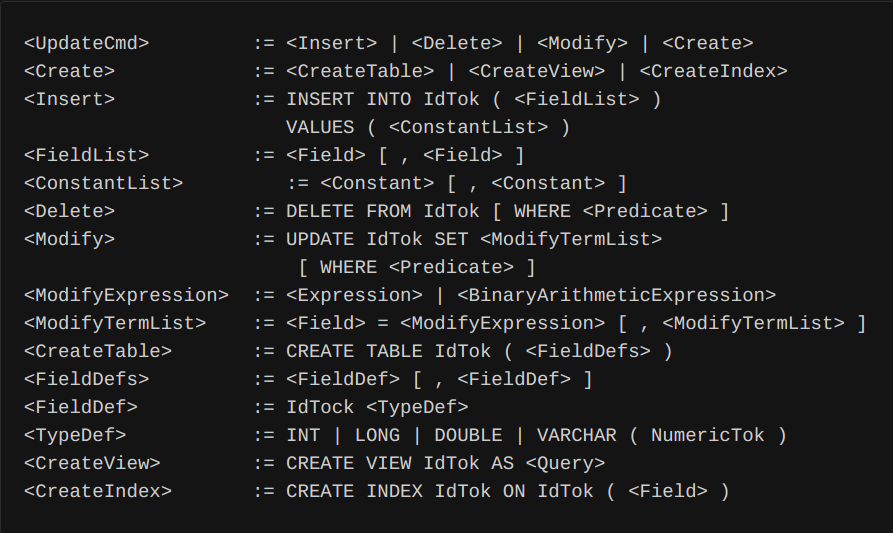
然後以nextToken繼續執行到文末，繼續切割，直到整個輸入的eof(end of file 才會停下來)

以在同一個Lexer.java中有建立一個字典表，代表會用到的SQL關鍵字



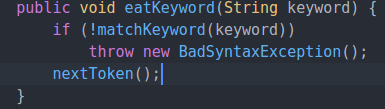
如果有不符合VanillaDBSQL的語法就會報錯，丟出BadSyntaxException這種例外，推斷是運用了『正規語言』的『上下文無關文法』(Context Free Grammar)





以上三者分別說明，關鍵字(Predicates)，查詢(Queries)，以及更新(Updates的語言規範)

，在Lexer中便會識別輸入的語句是屬於那一種然後找對應以上三者之一的文法歸範，如果不符合便會報錯，BadSyntaxException();　例如eatKeyword吃到不符合規範的關鍵自就會報錯，如下圖所示



切割完關鍵字後，便會是一個沒有語法錯誤的SQL語句，此時變可以送到Verifier.java中執行，在前置處理完成後進行要真正做的事情。

●step2切割後當然就會把切割好的東西送去分類以及查詢，