

TP2: Réalisation d'un Analyseur Lexical

Nombre de Séances : 2

L'objectif de ce TP est de réaliser en C un analyseur lexical du langage de requête de consultation SQL (Select) décrit par la grammaire suivante :

```
query -> select [from] [where] [orderby]
select-> SELECT ('*' | proplist )
from -> FROM ntlist
where -> WHERE whereexp
orderby -> ORDER BY proptime [DESC|ASC] {' proptime [DESC|ASC]}
proplist -> proptime {';' proptime}
ntlist -> tablename {';' tablename}

whereexp -> proptime op value |
           proptime IS NULL |
           proptime IS NOT NULL |
           proptime IN '(' value {';' value} ')' |
           like |
           contains |
           whereexp AND whereexp |
           whereexp OR whereexp |
           NOT whereexp |
           '(' whereexp ')'
```

```
op -> '='>'<'>='<='<>'
like -> proptime LIKE likepattern [ escape ]
likepattern -> likechar { likepattern }
likechar -> char | '%' | '_'
escape -> ESCAPE' likechar
contains -> CONTAINS '(scope ',' exp )'
```

```
scope -> proptime {';' proptime} | '*'
exp -> word {OR word}
```

Les mots clés apparaissent en gras majuscule. Le terminal **proptime** représente le nom d'une colonne, **tablename** représente le nom d'une table, **value** représente la valeur d'une colonne, elle peut être caractère, entière, réelle ou chaîne. Le terminal **char** représente un caractère et **word** une chaîne de caractère.

Bon courage