CSCE 315

TEACHING ASSISTANT: JAY CHEN JAYCHEN@CSE.TAMU.EDU

Project 1: **Database Management System**

- Design Document
- DB Engine
- Lexical Parser
- DB Engine and Lexical Parser Integration
- DB Application and integration

Grammar

- **Baccus Naur Form (BNF):**
  - **=** : definition
  - **,** : concatenation
  - **; or .** : termination
  - **|** : alternation, or
  - **::=** : the symbol on the left must be replaced with the expression on the right
  - **[]** : Optional items enclosed in square brackets
  - **{}** :Items repeating 0 or more times

Grammar

insertcmd::=

INSERT INTO relationname VALUES FROM (literal { , literal }); **OR**

INSERT INTO relationname VALUES FROM RELATION expr ;

E.g.,

INSERT INTO relationname VALUES FROM (literal { , literal });

INSERT INTO animals VALUES FROM ("Joe", "cat", 4);

animals		
Spot	Dog	10
Snoopy	Dog	3
Tweety	Bird	1
Joe	Cat	4

Grammar

E.g.,

INSERT INTO relationname VALUES FROM RELATION expr

INSERT INTO species VALUES FROM RELATION project (kind) animals;

species
Pig
dog
fish

animals		
Spot	Dog	10
Snoopy	Dog	3
Tweety	Bird	1
Joe	Cat	4

species
Pig
Rat
fish
Dog
Dog
Bird
Cat

Note: both tables should have a “Primary Key” attribute, but it is ignored here

Grammar

expr ::= *atomicexpr*

| *selection*

| *projection*

| *renaming*

| *union*

| *difference*

| *product*

selection **::= select (*condition*) *atomicexpr***

atomicexpr::= *relationname* | (*expr*)

condition ::= *conjunction* { || *conjunction* }

conjunction ::= *comparison* { && *comparison* }

comparison ::= *operand op operand* | (*condition*)

operand ::= *attributename* | *literal*

projection ::= project (*attributelist*) *atomicexpr*

A example expr:

Select (kind == “dog” || kind == “cat”) animal ;

Select (kind == “dog” || kind == “cat”) (project (kind, years) animal);

animals		
Spot	Dog	10
Snoopy	Dog	3
Tweety	Bird	1
Joe	Cat	4

Grammar

Spot	Dog	10
Snoopy	Dog	3
Joe	Cat	4

Select (kind == “dog” || kind == “cat”) animal ;

Dog	10
Dog	10
Dog	3
Cat	4

Select (kind == “dog” || kind == “cat”) (project (kind, years) animal)

Continue the previous example, but more complicated command:
INSERT INTO species VALUES FROM RELATION project (kind) animals;

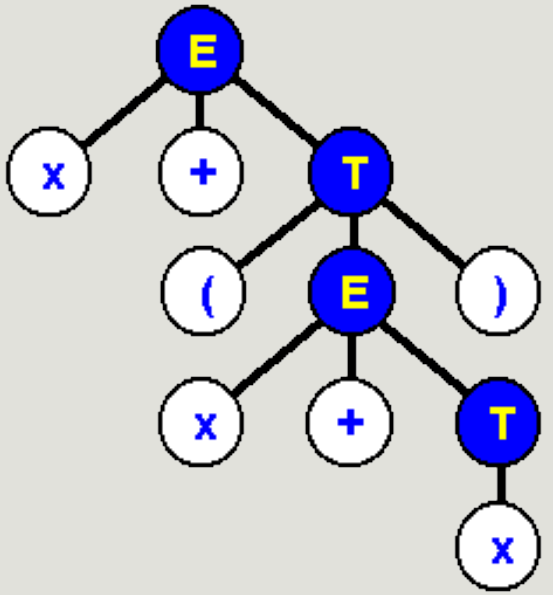
INSERT INTO species VALUES FROM RELATION Select (kind == “dog” || kind == “cat”) (project (kind, years) animal);

Recursive Descent Parser

Grammar:
E ::= x+T
T ::= (E) | x

Example:
x+(x+x)

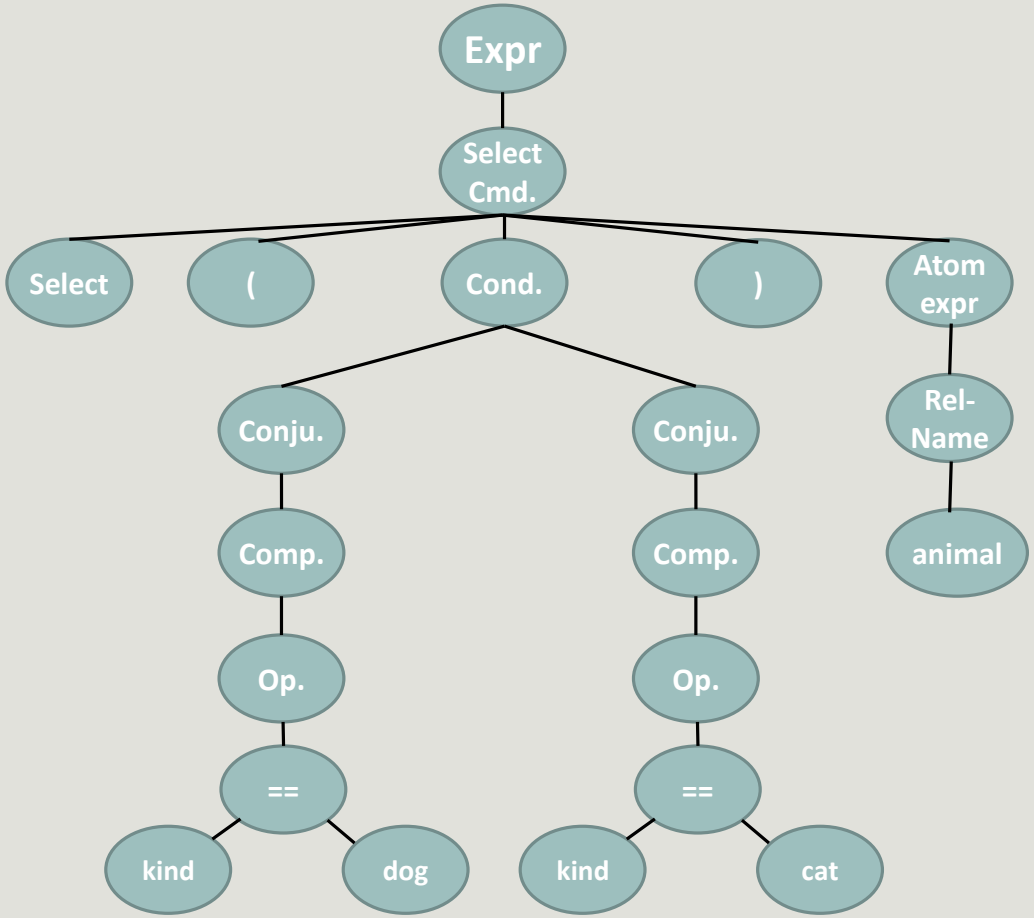
- Entire x+(x+x) is an Expression
- (x+x) is a Token
- x+x is an Expression



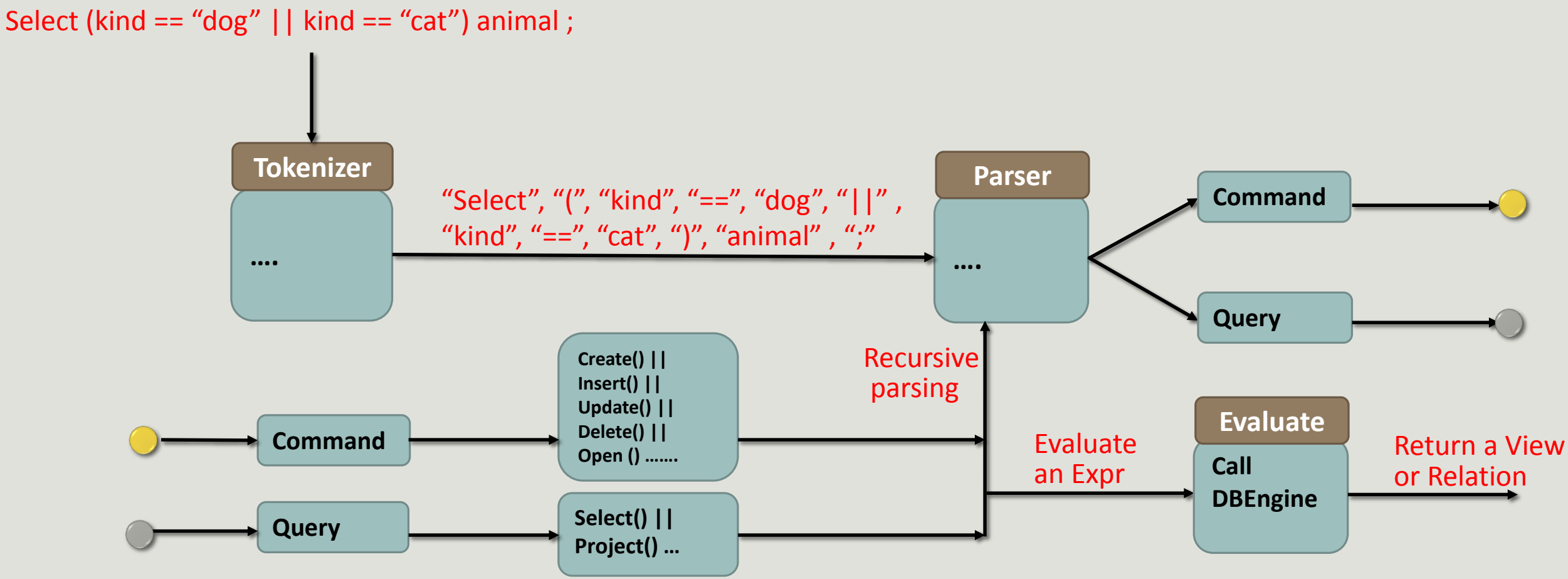
Recursive Descent Parser

Grammar:
selection **::=** select (*condition*) *atomicexpr*

Example:
Select (kind == “dog” || kind == “cat”) animal ;



Recursive Descent Parser



References

- https://en.wikipedia.org/wiki/Extended_Backus%E2%80%93Naur_Form
- https://en.wikipedia.org/wiki/Recursive_descent_parser
- http://www.cs.engr.uky.edu/~lewis/essays/compilers/rec-des.html