

# Syntax Analysis

## ★ Grammar (EBNF)

$\langle \text{program} \rangle ::= \{ \langle \text{statement} \rangle \}$   
 $\langle \text{statement} \rangle ::= \langle \text{load\_stmt} \rangle | \langle \text{filter\_stmt} \rangle | \langle \text{map\_stmt} \rangle |$   
 $\quad \langle \text{aggregate\_stmt} \rangle | \langle \text{for\_stmt} \rangle | \langle \text{print\_stmt} \rangle |$   
 $\quad \langle \text{assign\_stmt} \rangle | \langle \text{expr\_stmt} \rangle$   
 $\langle \text{load\_stmt} \rangle ::= \text{"load"} \text{ IDENTIFIER "from" STRING\_LITERAL}$   
 $\langle \text{filter\_stmt} \rangle ::= \text{"filter"} \text{ IDENTIFIER } \langle \text{block} \rangle$   
 $\langle \text{map\_stmt} \rangle ::= \text{"map"} \text{ IDENTIFIER "on" IDENTIFIER } \langle \text{block} \rangle$   
 $\langle \text{aggregate\_stmt} \rangle ::= \text{"aggregate"} \text{ IDENTIFIER "on" IDENTIFIER } \langle \text{block} \rangle$   
 $\langle \text{for\_stmt} \rangle ::= \text{"for"} \text{ IDENTIFIER "in" IDENTIFIER } \langle \text{block} \rangle$   
 $\langle \text{print\_stmt} \rangle ::= \text{"print"} \langle \text{expr} \rangle \langle \text{expr\_list} \rangle$   
 $\langle \text{assign\_stmt} \rangle ::= \text{IDENTIFIER "=" } \langle \text{expr} \rangle$   
 $\langle \text{block} \rangle ::= \{ \{ \langle \text{block\_stmt} \rangle \} \}$   
 $\langle \text{block\_stmt} \rangle ::= \langle \text{where\_stmt} \rangle | \langle \text{assign\_stmt} \rangle | \langle \text{print\_stmt} \rangle$   
 $\langle \text{where\_stmt} \rangle ::= \text{"where"} \langle \text{expr} \rangle$   
 $\langle \text{expr\_list} \rangle ::= \langle \text{expr} \rangle \{ \text{","} \langle \text{expr} \rangle \}$   
 $\langle \text{expr} \rangle ::= \langle \text{logic\_expr} \rangle$   
 $\langle \text{logic\_expr} \rangle ::= \langle \text{rel\_expr} \rangle \{ (\text{"and"} | \text{"or"}) \langle \text{rel\_expr} \rangle \}$   
 $\langle \text{rel\_expr} \rangle ::= \langle \text{add\_expr} \rangle [(\text{"="} | \text{"!="} | \text{"<" } | \text{">" } | \text{"<=" } | \text{">="}) \langle \text{add\_expr} \rangle]$   
 $\langle \text{add\_expr} \rangle ::= \langle \text{mul\_expr} \rangle \{ (\text{"+"} | \text{"-"}) \langle \text{mul\_expr} \rangle \}$   
 $\langle \text{mul\_expr} \rangle ::= \langle \text{unary\_expr} \rangle \{ (\text{"*"} | \text{" / " } | \text{" % " }) \langle \text{unary\_expr} \rangle \}$   
 $\langle \text{unary\_expr} \rangle ::= [\text{"-"}] \langle \text{primary} \rangle$   
 $\langle \text{primary} \rangle ::= \text{NUMBER} | \text{STRING\_LITERAL} | \text{IDENTIFIER} |$   
 $\quad \text{IDENTIFIER "." IDENTIFIER} | \langle \text{function\_call} \rangle | \text{"(" } \langle \text{expr} \rangle \text{"}"}$   
 $\langle \text{function\_call} \rangle ::= \text{IDENTIFIER " (" } \langle \text{expr\_list} \rangle \text{ ")"}$

### \* Notes:

- ( ) without quotation marks represents grouped grammar elements
- [ ] without quotation marks represents optional grammar elements
- { } without quotation marks represents repetition (0 or more times)



# \* Example Program (For parse tree and derivation)

load employees from "employees.csv"

filter high\_salary {

where salary > 50000

map bonus\_calc on high\_salary {

bonus = salary \* 0.10

aggregate stats on bonus\_calc {

avg\_salary = avg(salary)

avg\_bonus = avg(bonus)

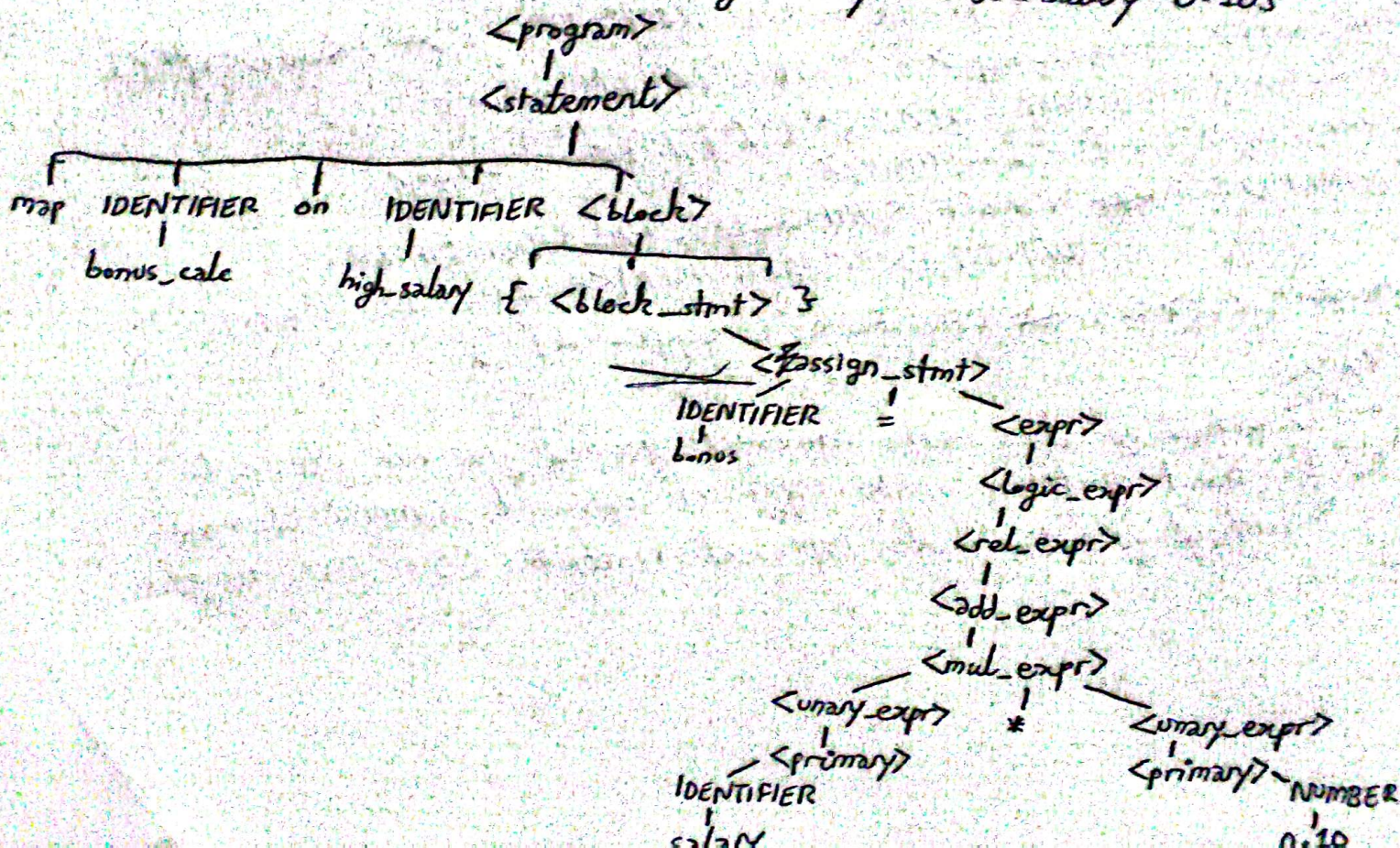
print stats

for row in bonus\_calc {

print row.name, row.salary, row.bonus

}

## \* Parse Tree for "map bonus\_calc on high\_salary { bonus = salary \* 0.10 }





\* Parse Tree for "for row in bonus\_cale {print row.name, row.salary, row.bonus}"

