**✍در مورد نحوۀ توصیف گرامر برای پیروی از تقدم های ذکر شده، توضیحاتی داده شود.**

ubprogram ::= procedure | function

procedure ::= CREATE PROCEDURE name (parameters) AS BEGIN statements END

function ::= CREATE FUNCTION name (parameters) RETURNS type AS BEGIN RETURN expression END

parameters ::= parameter | parameter, parameters

parameter ::= name type

statements ::= statement | statement; statements

statement ::= assignment | control | query | call

assignment ::= name := expression

control ::= IF condition THEN statements ELSE statements END IF | WHILE condition LOOP statements END LOOP

query ::= SELECT columns FROM tables WHERE condition

call ::= name (arguments)

columns ::= column | column, columns

column ::= name | name AS alias

tables ::= table | table, tables

table ::= name | name AS alias

condition ::= expression comparison expression | condition AND condition | condition OR condition | NOT condition

comparison ::= = | <> | < | > | <= | >=

expression ::= term | term + expression | term - expression

term ::= factor | factor \* term | factor / term

factor ::= (expression) | name | literal | function (arguments)

arguments ::= argument | argument, arguments

argument ::= expression

type ::= INT | FLOAT | CHAR | VARCHAR | DATE | BOOLEAN

name ::= identifier

alias ::= identifier

identifier ::= letter | letter identifier

letter ::= A | B | ... | Z | a | b | ... | z

literal ::= number | string | date | boolean

number ::= digit | digit number

digit ::= 0 | 1 | ... | 9

string ::= 'character' | 'character string'

character ::= any printable ASCII character

date ::= 'YYYY-MM-DD'

boolean ::= TRUE | FALSE

CREATE FUNCTION average\_salary (dept\_id INT) RETURNS FLOAT AS

BEGIN

RETURN (SELECT AVG(salary) FROM employees WHERE department\_id = dept\_id);

END

قسمتی که با آبی هایلایت شده برای نشان دادن تقدن عملگرها بجای دستورات زیر قرار گرفته است:

expression ::= term | term + term | term - term | term \* term | term / term | term % term | (expression) | name | literal | function (arguments)

term ::= name | literal | function (arguments)

تصحیح و جایگذاری انجام شده در توصیف گرامر برای پیروی از تقدم‌های مختلف، شامل ساخت یک گرامر بدون ابهام برای زبان است. این روش به این صورت عمل می‌کند که با استفاده از توسعه یا تغییر گرامر، تقدم و هم‌سطحی عملگرها را در درخت‌های نحوی خود منعکس می‌کند. به عبارت دیگر، این روش با ایجاد سطوح مختلف برای عبارات، اولویت ارزیابی آن‌ها را مشخص می‌کند. برای مثال، در این گرامر عبارت a + b \* c به شکل زیر درخت نحوی می‌سازد:

این درخت نشان می‌دهد که عبارت b \* c اول ارزیابی می‌شود و سپس نتیجه‌ی آن با a جمع می‌شود. به همین ترتیب، عبارت (a + b) \* c به شکل زیر درخت اشتقاق می‌سازد:

این درخت نشان می‌دهد که عبارت a + b اول ارزیابی می‌شود و سپس نتیجه‌ی آن با c ضرب می‌شود. این روش باعث می‌شود که هر عبارت فقط یک درخت اشتقاق داشته باشد و ابهامی در تفسیر آن وجود نداشته باشد.