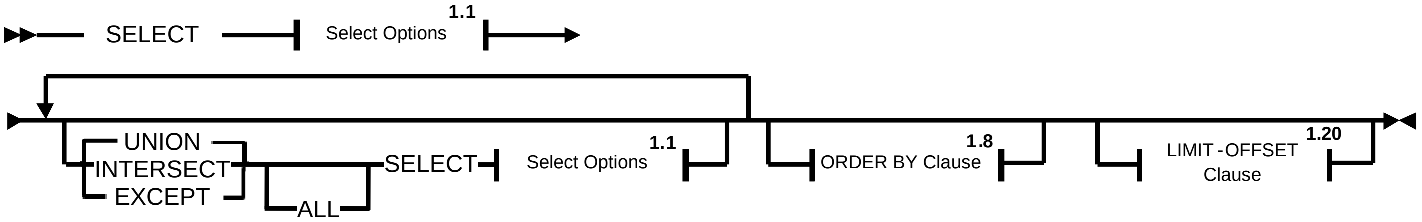
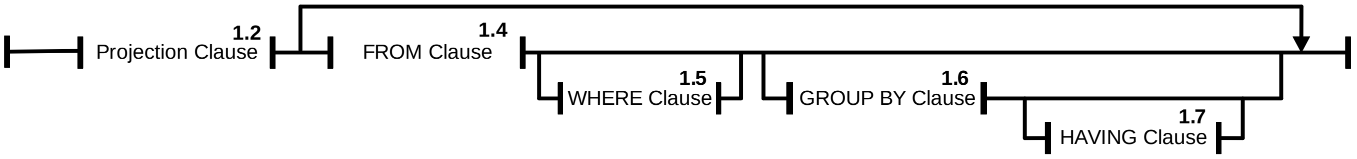
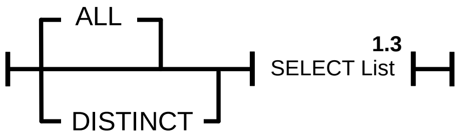
1. SELECT Statement



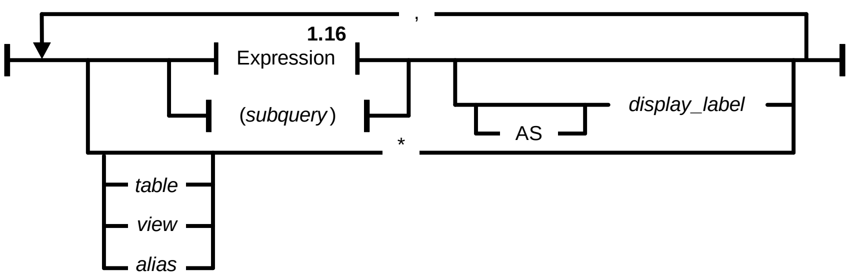
1.1 SELECT Options



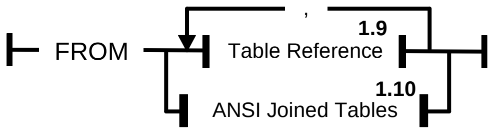
1.2 Projection Clause



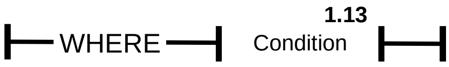
1.3 SELECT List



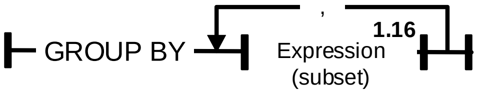
1.4 From Clause



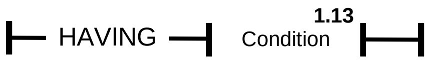
1.5 WHERE Clause



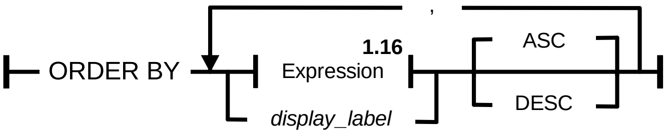
1.6 GROUP BY Clause



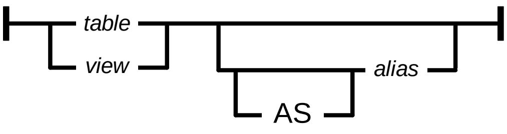
1.7 HAVING Clause



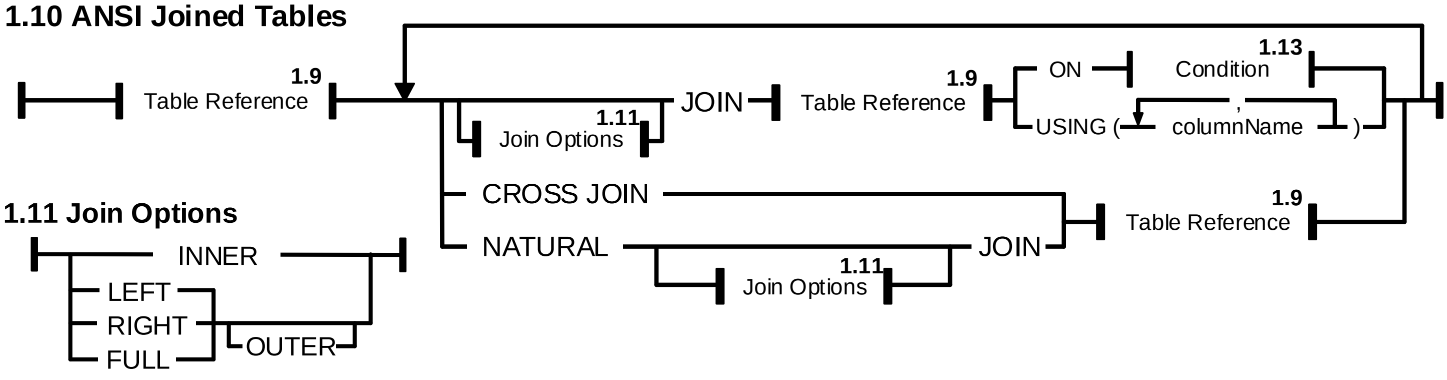
1.8 ORDER BY Clause



1.9 Table Reference

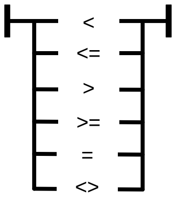


1.10 ANSI Joined Tables

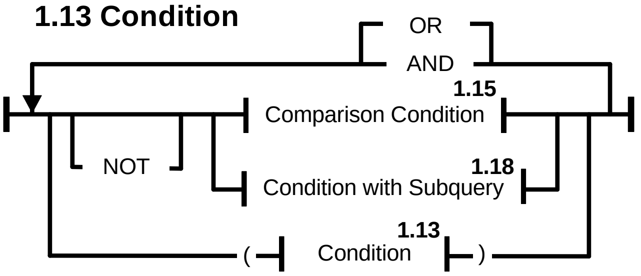


1.11 Join Options

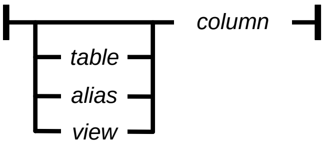
1.12 Relational Operator



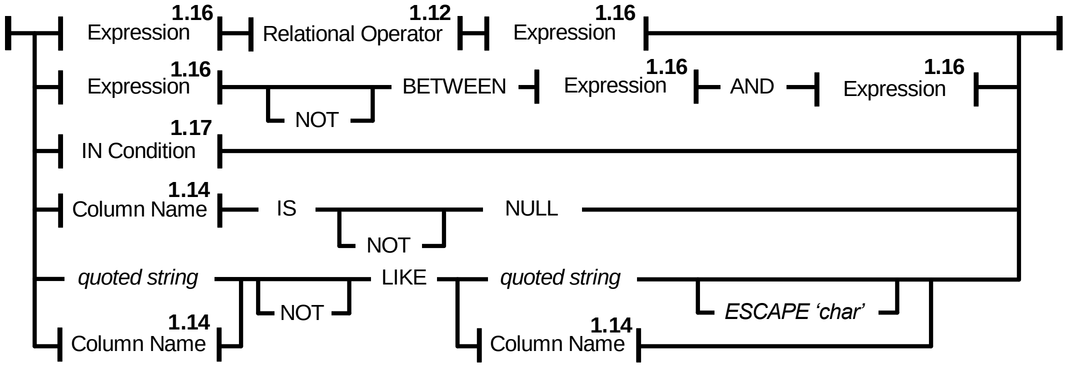
1.13 Condition



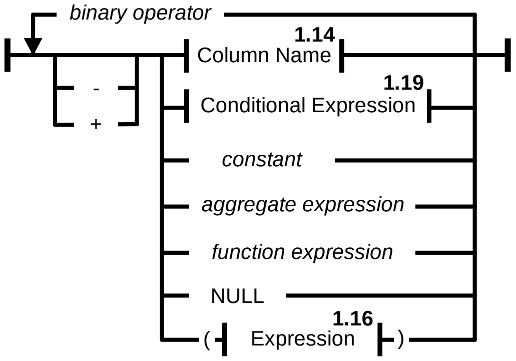
1.14 Column name



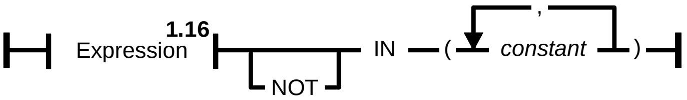
1.15 Comparison Condition



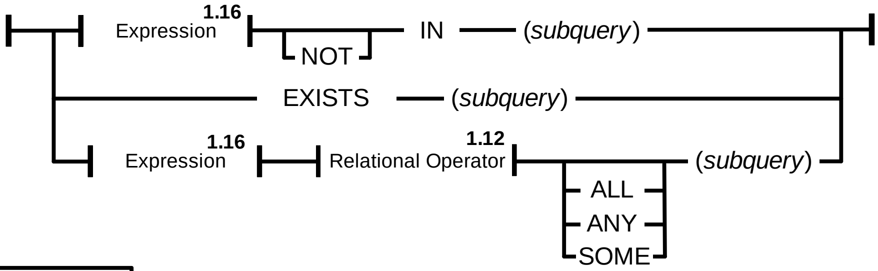
1.16 Expression



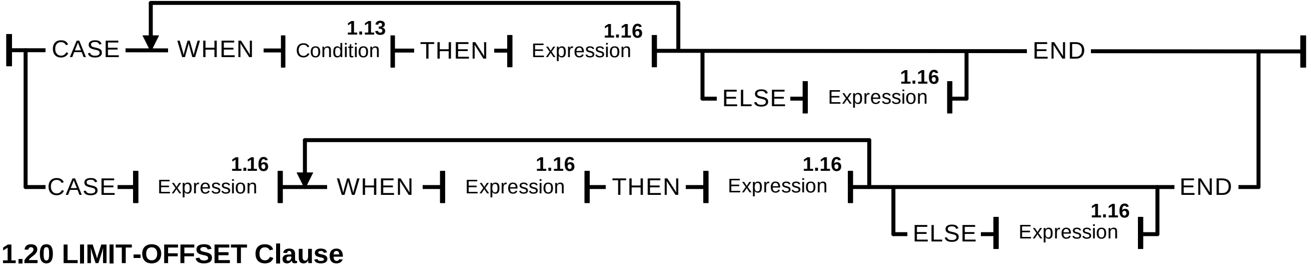
1.17 IN Condition



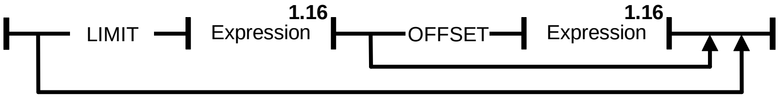
1.18 Condition with Subquery



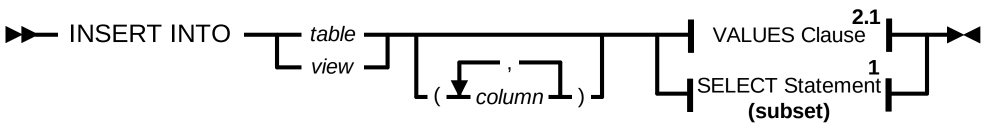
1.19 Conditional Expression



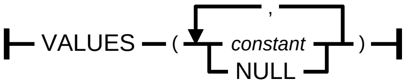
1.20 LIMIT-OFFSET Clause



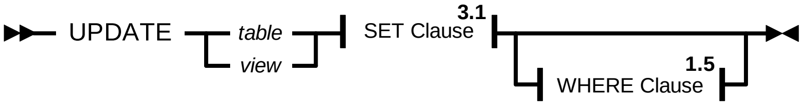
2. INSERT Statement



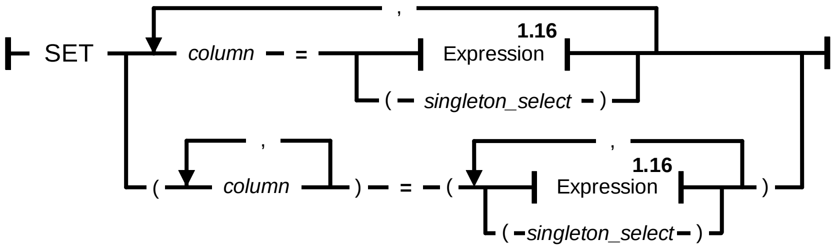
2.1 VALUES Clause



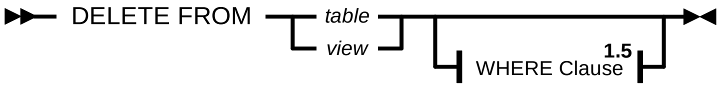
3. UPDATE Statement



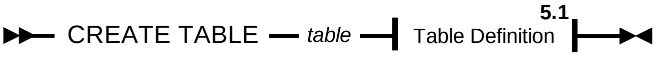
3.1 SET Clause



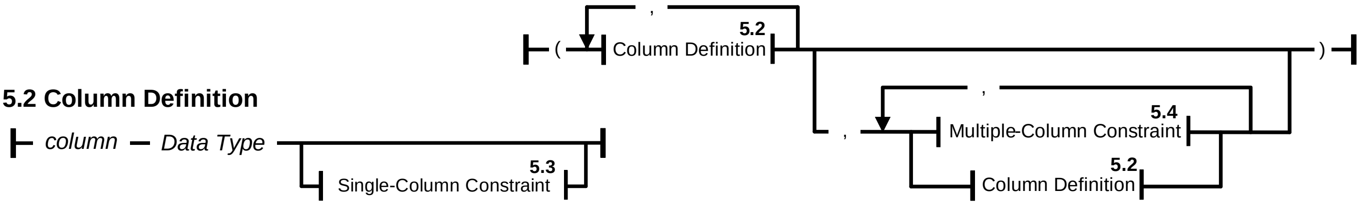
4. DELETE Statement



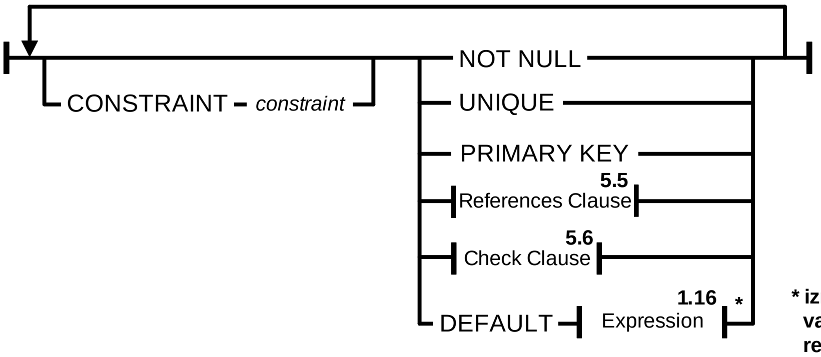
5. CREATE TABLE Statement



5.1 Table Definition

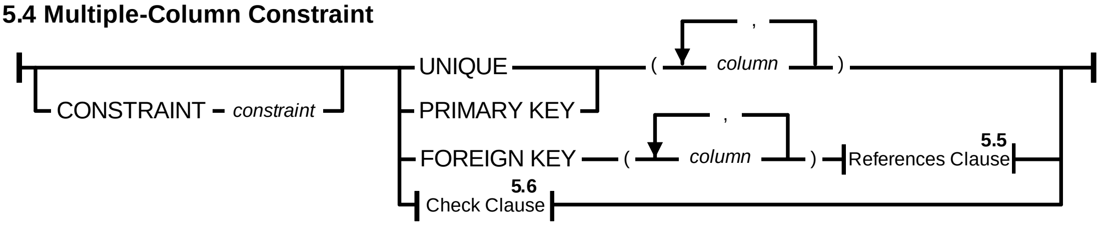


5.3 Single-Column Constraint

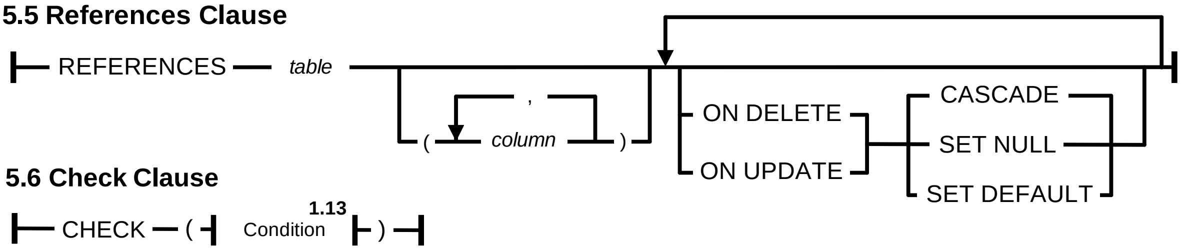


*** izraz *default_expr* ne smije sadržavati varijable i podupite, niti smije referencirati stupce trenutne tablice**

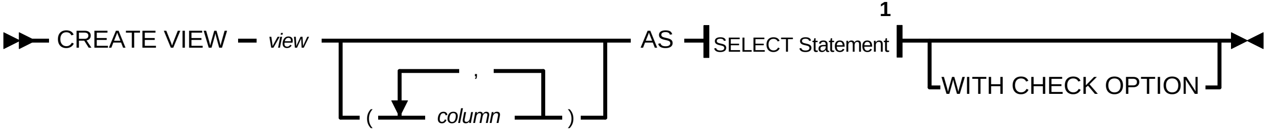
5.4 Multiple-Column Constraint



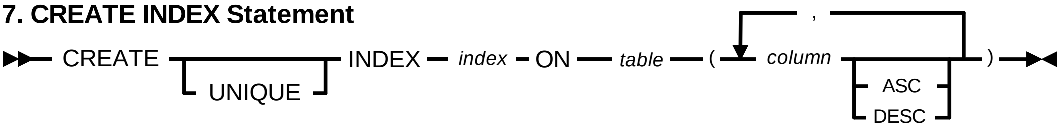
5.5 References Clause



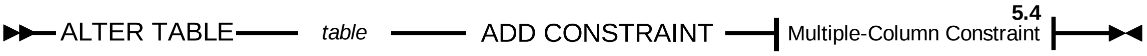
6. CREATE VIEW Statement



7. CREATE INDEX Statement



8. ALTER TABLE Statement



CAST (expression AS type)
ili **expression::type**

ABS (*num_expression*)
num_expression – izraz numeričkog tipa

MOD (*dividend, divisor*)
dividend (djeljenik) – izraz numeričkog tipa
divisor (djelitelj) – izraz numeričkog tipa

ROUND (*expression[, rounding_factor]*)
expression – izraz numeričkog tipa
rounding_factor – izraz cjelobrojnog tipa

SUBSTRING (*source_string FROM start_position [FOR length]*)
source_string – izraz tipa niza znakova
start_position – izraz cjelobrojnog tipa
length(duljina) – izraz cjelobrojnog tipa

UPPER (*expression*)
LOWER (*expression*)
expression – izraz tipa niza znakova

TRIM(*expression*)
expression – izraz tipa niza znakova

CHAR_LENGTH(*expression*)
OCTET_LENGTH(*expression*)
expression – izraz tipa niza znakova

CURRENT_USER
CURRENT_DATE
CURRENT_TIME
HH:MI:SS.xxxxxx
CURRENT_TIMESTAMP
DD.MM.YYYY HH:MI:SS.xxxxxx±HH

EXTRACT (*field* FROM *source*)
field ∈ {year, month, day, hour, minute, second, week, dow, doy}
source – izraz tipa TIMESTAMP ili INTERVAL

Za B$^+$-stablo reda n vrijedi

korijen:
min. broj kazaljki na podređene čvorove: **2**
maks. broj kazaljki na podređene čvorove: **n**

ostali interni čvorovi:
min. broj kazaljki na podređene čvorove: ⌈**n/2**⌉
maks. broj kazaljki na podređene čvorove: **n**

listovi:
min. broj kazaljki na zapise u blokovima s podacima: ⌈**(n-1)/2**⌉
maks. broj kazaljki na zapise u blokovima s podacima: **n-1**

Posjetnik na PL/pgSQL sintaksu

```
CREATE OR REPLACE FUNCTION fname ( arg type, ...)
    RETURNS ( type | VOID ) AS $$
DECLARE var [CONSTANT] type [NOT NULL]
        [ {DEFAULT | := | = } expression ];
BEGIN
var1 = var2 + var3;
UPDATE rel SET atr = var...;
SELECT atr1, atr2, ... INTO var1, var2, ...
  FROM rel
  WHERE ...;

RAISE EXCEPTION 'Error description'
  USING HINT = 'Instruction';

IF (condition) THEN
    statements;
ELSIF (condition) THEN
    statements;
    ...
ELSE
    statements;
END IF;

FOR i IN 1..m LOOP
    statements;
    ...
    END LOOP;
RETURN var;
END;
$$ LANGUAGE plpgsql
   [SECURITY (INVOKER | DEFINER)];

SELECT fname (arg1, arg2, ...);
DROP FUNCTION fname;
```

Posjetnik na sintaksu naredbe za kreiranje okidača

```
CREATE FUNCTION fname() RETURNS trigger AS $$
BEGIN
   [ SELECT... | INSERT... | UPDATE... | DELETE... ]
   ...;
  IF ( TG_OP = {'INSERT' | 'UPDATE'| 'DELETE'} ) THEN
  ...
  END IF;
  RAISE EXCEPTION 'message';

  RETURN { NEW | NULL };
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER name
  { BEFORE | AFTER | INSTEAD OF }  { event [OR ...] }
    ON table_name
  [ FOR [EACH] { ROW | STATEMENT } ]
  [ WHEN (condition) ]
 EXECUTE PROCEDURE fname( arguments );
event ∈ { INSERT | UPDATE  | DELETE }
```

```
CREATE {USER | ROLE} name
    [ [ WITH ] option [...] ];
DROP {USER | ROLE} roleName;
SET ROLE roleName;
where option can be:
   SUPERUSER | NOSUPERUSER
 | CREATEDB | NOCREATED
 | CREATEUSER | NOCREATEUSER
 | [ENCRYPTED | UNENCRYPTED] PASSWORD 'pwd'
 | CREATEROLE | NOCREATEROLE
 | INHERIT | NOINHERIT
 | LOGIN | NOLOGIN

GRANT dbPrivilege ON DATABASE name
   TO {roleList | PUBLIC};

REVOKE dbPrivilege ON DATABASE name
 FROM {roleList | PUBLIC};
dbPrivilege: CONNECT, CREATE


GRANT schemaPrivilege ON SCHEMA name
   TO {roleList | PUBLIC};

REVOKE schemaPrivilege ON SCHEMA name
 FROM {roleList | PUBLIC};
schemaPrivilege: ALL, USAGE, CREATE



GRANT tablePrivilegeList ON
  {ALL TABLES|tableName|viewName}
     [ IN SCHEMA name ]
  TO {roleList | PUBLIC}  [WITH GRANT OPTION];


REVOKE tablePrivilegeList
       ON {tableName|viewName}
 FROM {roleList | PUBLIC}  [CASCADE|RESTRICT];

tablePrivilege:  SELECT [(columnList)]
                 UPDATE [(columnList)]
                 INSERT [(columnList)]
                 DELETE
                 ili ALL (zamjenjuje sve op.)

CREATE SCHEMA schName;
DROP SCHEMA schName [CASCADE];


GRANT EXECUTE ON {
 FUNCTION function_name([atype [, ...] ] ) }
 TO role_specification [, ...]
    [ WITH GRANT OPTION ]


REVOKE [ GRANT OPTION FOR ] EXECUTE ON {
 FUNCTION function_name([atype [, ...] ] ) }
 FROM { [ GROUP ] role_name | PUBLIC }
 [ CASCADE | RESTRICT ]
```

Varijabla	Opis
NEW	nova vrijednost n-torke; ta n-torka će u konačnici biti zapisana (moguće ju je mijenjati). NULL, ako je DELETE ili STATEMENT lvl
OLD	stara vrijednost n-torke; NULL ako je INSERT ili STATEMENT lvl. (podsjetnik: UPD = INS + DEL)
TG_NAME	ime okidača
TG_WHEN	BEFORE, AFTER, ili INSTEAD OF
TG_LEVEL	ROW ili STATEMENT
TG_OP	INSERT, UPDATE, DELETE ili TRUNCATE