syntegris information solutions

Der perfekte 12c Trigger

Author: Sven Weller

www.syntegris.de

Der perfekte 12c Trigger

NEW 12C FEATURES

Autor: Sven Weller, Syntegris, Neu-Isenburg

Der perfekte 12c Trigger

BEFORE ROW INSERT TRIGGER

- ✓ Feuert beim Insert
- ✓ Für jede Zeile
- ✓ Setzt Spaltenwerte

Traditional Trigger

ID AND AUDIT COLUMNS

```
1   create table swe_demo (id number primary key
2                         ,col1 number
3                         ,col2 varchar2(30)
4                         ,inserted_date date not null
5                         ,inserted_from varchar2(30) not null);
6
7   create sequence swe_demo_seq cache 10000;
8
9   create or replace trigger swe_demo_bri_trg
10    BEFORE INSERT ON swe_demo
11    FOR EACH ROW
12  BEGIN
13     -- record needs always a key
14     IF :new.id IS NULL
15     THEN
16        :new.id := swe_demo_seq.NEXTVAL;
17     END IF;
18
19     -- timestamp of the last changes
20     :new.inserted_date := SYSDATE;
21     :new.inserted_from := NVL(v('APP_USER'), user);
22  END swe_demo_bri_trg;
23  /
```

Der perfekte 12c Trigger

TOPICS

Der “perfekte” Trigger

=

?

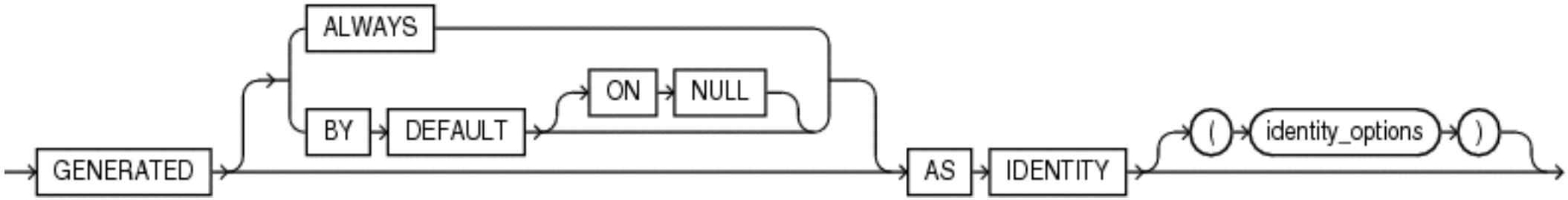
Der perfekte 12c Trigger

TOPICS

- ✓ Neue 12c Features
- ✓ Performance
- ✓ Tipps und Tricks

Identity column (12R1)

SYNTAX




```
"ID" generated always as identity
```

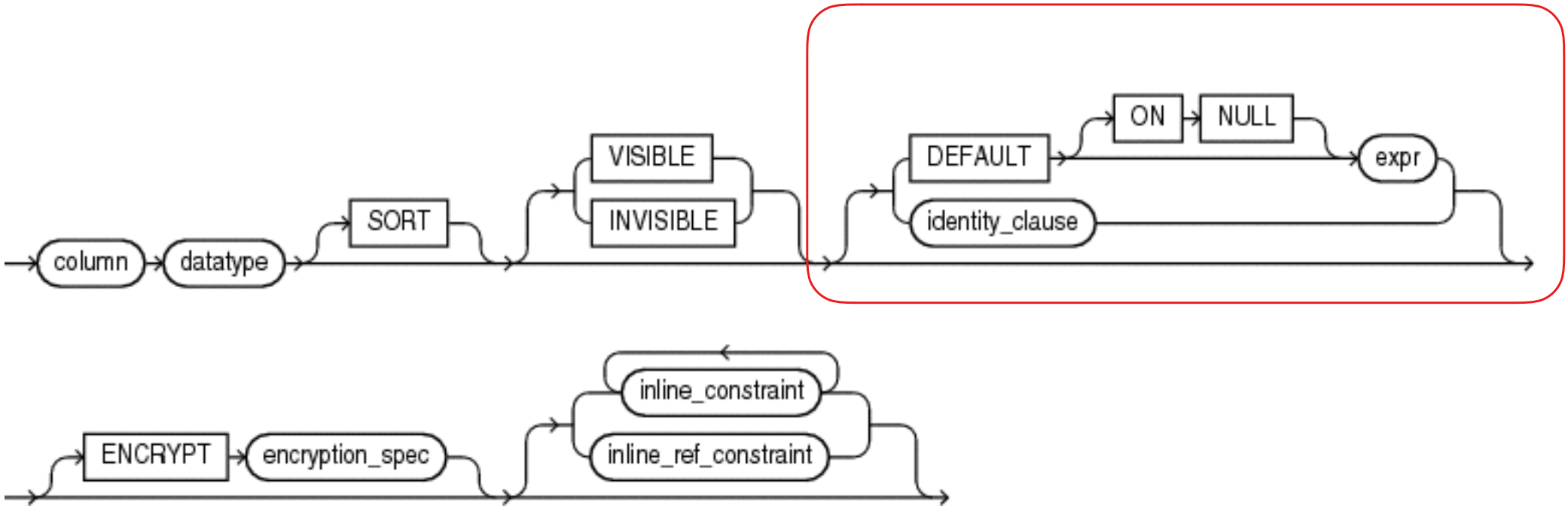
oder

```
"ID" generated by default on null as identity
```

Sequence wird automatisch erzeugt und verwaltet

Default value columns (12R1)

SYNTAX



✓ **expression => sequence.nextval**

Default value oder Identity?

ENTSCHEIDUNG

- ✓ *Default value auch für andere non-ID Spalten*
- ✓ *Identity Columns haben mehr "Nebeneffekte"*

```
1    set serveroutput on
2    declare
3      v_id demo@remoteDB.id%type;
4    begin
5      insert into demo@remoteDB (col1)
6      values ('abc')
7      returning id into v_id;
8      dbms_output.put_line('new ID='||v_id);
9    end;
10   /
ORA-22816: unsupported feature with RETURNING clause
ORA-06512: at line 4
22816. 00000 -  "unsupported feature with RETURNING clause"
*Cause:    RETURNING clause is currently not supported for object type
           columns, LONG columns, remote tables, INSERT with subquery,
           and INSTEAD OF Triggers.
*Action:   Use separate select statement to get the values.
```

12c solution (ohne trigger)

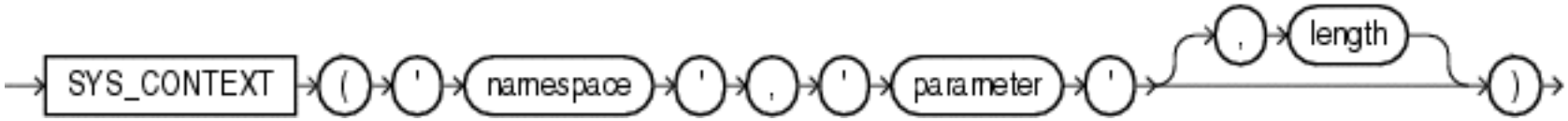
DEFAULT COLUMNS

```
create sequence swe_demo_seq cache 10000;
create table swe_demo
  (id number default on null swe_demo_seq.nextval primary key
   ,col1 number
   ,col2 varchar2(30)
   ,inserted_date date default sysdate not null
   ,inserted_from varchar2(30) default NVL(v('APP_USER'), user) not null);
```

Tipp: user => langsam

Was ist ein Kontext?

SYS_CONTEXT



- Lokale oder globale “Variable”
- vordefiniert: USERENV
- Security Feature

Aktueller Benutzer

USERENV

```
SYS_CONTEXT('USERENV', 'parameter')
```

Parameter	Beschreibung
client_identifier	Returns an identifier that is set by the application through the DBMS_SESSION.SET_IDENTIFIER procedure, the OCI attribute OCI_ATTR_CLIENT_IDENTIFIER, or the Java class Oracle.jdbc.OracleConnection.setClientIdentifier. This attribute is used by various database components to identify lightweight application users who authenticate as the same database user.
current_schema	Name of the default schema being used in the current schema. This value can be changed during the session with an ALTER SESSION SET CURRENT_SCHEMA statement.
current_user	Deprecated – Use the SESSION_USER parameter instead.
session_user	For enterprises users, returns the schema. For other users, returns the database user name by which the current user is authenticated. This value remains the same throughout the duration of the session.
dblink_info	Returns the source of a database link session. Specifically, it returns a string of the form:SOURCE_GLOBAL_NAME=dblink_src_global_name, DBLINK_NAME=dblink_name, SOURCE_AUDIT_SESSIONID=dblink_src_audit_sessionid

Aktueller Benutzer

APEX KONTEXTE

```
SYS_CONTEXT('APEX$SESSION','APP_USER')
SYS_CONTEXT('APEX$SESSION','APP_SESSION')
SYS_CONTEXT('APEX$SESSION','WORKSPACE_ID')
```

✔ *neu in Apex 5*

✔ *Extrem performant*

```
SYS_CONTEXT('USERENV','client_identifier')
```

APP_USER:APP_SESSION => SVEN:0123456789

12c Ergebnis - ohne Trigger

DEFAULT COLUMNS + KONTEXT

```
create sequence swe_demo_seq cache 10000;
create table swe_demo
  (id number default on null swe_demo_seq.nextval primary key
   ,col1 number
   ,col2 varchar2(30)
   ,inserted_date date default sysdate not null
   ,inserted_from varchar2(30) default coalesce(
          sys_context('APEX$SESSION','app_user')
         ,regexp_substr(sys_context('userenv','client_identifier'),'^[^:]*')
         ,sys_context('userenv','session_user')
         )
      not null);
```

12c ohne Trigger

VORTEILE

- ✓ Weniger Code
- ✓ Performance
- ✓ Sequence mit Tabelle verbunden

syntegris information solutions

Fragen?

www.syntegris.de

Sonstiges

BEFORE ROW UPDATE TRIGGER

Keine 12c Alternative bisher!

Neuer Syntax Vorschlag: DEFAULT ON UPDATE

https://community.oracle.com/ideas/15760

Sonstiges

PERFORMANCE TEST - TC1

Test Skript

```
1     set time on
2     set timing on
3     declare
4       v_result varchar2(100);
5     begin
6       for i in 1..1000000 loop
7         v_result := ##EXPRESSION##;
8       end loop;
9     end;
10    /
```

Sonstiges

PERFORMANCE TEST - TC1

Expression	Time (in s)
sys_context('userenv','client_identifier')	2,4
substr(sys_context('userenv','client_identifier'),1 ,instr(sys_context('userenv','client_identifier'),':')-1)	4,3
substr(sys_context('userenv','client_identifier'),1, coalesce(nullif(instr(sys_context('userenv','client_identifier'),':'),0)-1, length(sys_context('userenv','client_identifier'))))	6,3
regexp_substr(sys_context('userenv','client_identifier'),'^[^:]*')	3,5
translate(sys_context('userenv','client_identifier'),'A0123456789','A')	5,6
user	20,6
sys_context('APEX$SESSION','app_user')	1,5
v('APP_USER')	5,6

Sonstiges

CONTEXT WITH DBLINK

#	Beschreibung	Inserted From	App User	Apex	
1	apex remote insert	SVEN	–	–	
2	apex local insert	SVEN	SVEN	SVEN	
5	direct remote insert	REMOTE_B	–	–	–
6	direct local insert	LOCAL_A	–	–	–
7	direct insert current_schema	IAMDBA	–	–	–

The following tests were done:
1) An insert from an apex page into a view that included a db link to REMOTE_B
2) An insert from an apex page, directly into a local view (LOCAL_A) which was mapped to demo.swe_demo.
5) A direct database insert in the view with the DB link (remote insert)
6) A direct database insert in the view without the db link
7) an insert from a DBA “IAMDBA” using alter session set current_schema

#	Current Schema	Current User	User	Session User	Authenticated Identity
1	DEMO	DEMO	REMOTE_B	REMOTE_B	REMOTE_B
2	DEMO	DEMO	APEX_PUBLIC_USER	APEX_PUBLIC_USER	APEX_PUBLIC_USER
5	DEMO	DEMO	REMOTE_B	REMOTE_B	REMOTE_B
6	DEMO	DEMO	LOCAL_A	LOCAL_A	LOCAL_A
7	DEMO	DEMO	IAMDBA	IAMDBA	IAMDBA

syntegris information solutions

The End!

www.syntegris.de