

## Tipps & Tricks: Privilegien prüfen

Bereich:	DBA	Erstellung:	07/2013 MP
Versionsinfo:	12.1	Letzte Überarbeitung:	07/2013 MP

### Privilegien prüfen

Welche Benutzer haben zu viele Rechte und nutzen diese gar nicht?

Bei dieser Fragestellung hilft uns ab Version 12c das Package `DBMS_PRIVILEGE_CAPTURE`.

Um damit arbeiten zu können, benötigen Sie die `CAPTURE_ADMIN` Rolle:

```
GRANT capture_admin TO scott;
```

#### Ablauf der Rechteerfassung

1. Sie sagen, was Sie aufzeichnen wollen:

```
DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE
```

2. Sie starten die Aufzeichnung:

```
DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE
```

3. Sie führen Ihre Aufgaben in der DB mit dem gewünschten Benutzer (bzw. Ihrer App) aus:

```
SELECT  
UPDATE  
CREATE  
ALTER  
DROP
```

4. Sie stoppen die Aufzeichnung:

```
DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE
```

5. Die Reporterzeugung wird angestoßen:

```
DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
```

6. Für die Auswertung stehen Ihnen folgende Views zur Verfügung:

•

- DBA\_USED\_PRIVS
- DBS\_UNUSED\_PRIVS

**Beispiel 1 - Alle Rechte der Datenbank erfassen**

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
    name => 'ALL_PRIVS_ANALYSIS',
    type => DBMS_PRIVILEGE_CAPTURE.G_DATABASE);
END;
/
```

**Beispiel 2 - Alle Rechte von PUBLIC erfassen**

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
    name  => 'public_analysis',
    type  => DBMS_PRIVILEGE_CAPTURE.G_ROLE,
    roles => role_name_list('PUBLIC'));
END;
/
```

**Beispiel 3 - Die genutzten Rechte des Benutzers SCOTT erfassen**

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
    name      => 'scott_priv_analysis',
    type      => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
    condition => q'!SYS_CONTEXT('USERENV', 'SESSION_USER')='SCOTT'!');
END;
/
```

**Beispiel 4 - Alle genutzten Rechte des Moduls sap\_modul**

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
    name      => 'sap_modul_priv_analysis',
    type      => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
    condition => q'!SYS_CONTEXT('USERENV', 'MODULE') = 'sap_modul'!');
END;
/
```

Aufzeichnung starten (für Beispiel 1):

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ('ALL_PRIVS_ANALYSIS');
END;
```

/

Befehle in der DB ausführen (wie an einem ganz normalen Tag):

```
SELECT * FROM ;
CREATE TABLE ;
DROP VIEW ;
```

Aufzeichnung beenden:

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ( 'ALL_PRIVS_ANALYSIS' );
END;
/
```

Report erzeugen:

```
BEGIN
  DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ( 'ALL_PRIVS_ANALYSIS' );
END;
/
```

Nun können wir sehen, welche Rechte zwischen `enable_capture` und `disable_capture` genutzt wurden, oder noch besser, welche der Benutzer zwar hat, aber nicht braucht:

```
SELECT * FROM dba_unused_privs
WHERE capture = 'ALL_PRIVS_ANALYSIS';
```

CAPTURE	USERNAME	ROLENAME	SYS_PRIV	OBJ_PRIV	USER_PRIV	OBJECT_OWNER	OBJECT_NAME
ALL_PRIV_CAPTURE	SCOTT	(null)	(null)	READ	(null)	SYS	DATA_PUMP_DIR
ALL_PRIV_CAPTURE	SCOTT	(null)	(null)	WRITE	(null)	SYS	DATA_PUMP_DIR
ALL_PRIV_CAPTURE	SCOTT	(null)	(null)	EXECUTE	(null)	SYS	DBMS_REDACT
ALL_PRIV_CAPTURE	SCOTT	(null)	(null)	EXECUTE	(null)	SYS	DBMS_PRIVILEG
ALL_PRIV_CAPTURE	SCOTT	(null)	UNLIMITED TABLESPACE	(null)	(null)	(null)	(null)
ALL_PRIV_CAPTURE	SCOTT	(null)	CREATE DATABASE LINK	(null)	(null)	(null)	(null)
ALL_PRIV_CAPTURE	SCOTT	(null)	CREATE VIEW	(null)	(null)	(null)	(null)
ALL_PRIV_CAPTURE	SCOTT	(null)	AUDIT SYSTEM	(null)	(null)	(null)	(null)

```
SELECT os_user, userhost, module, username, used_role, sys_priv,
       obj_priv, user_priv, object_owner, object_name, object_type,
       column_name
FROM dba_used_privs
WHERE capture = 'ALL_PRIVS_ANALYSIS';
```

OS_USER	USERHOST	MODULE	USERNAME	USED_ROLE	SYS_PRIV	OBJ_PRIV	USER_PRIV	OBJECT_OWNER	OBJECT
marco	goofymarco	SQL Developer	SCOTT	PUBLIC	(null)	SELECT	(null)	SYS	USER_OB
marco	goofymarco	SQL Developer	SCOTT	PUBLIC	(null)	SELECT	(null)	SYS	ALL_SYN
marco	goofymarco	SQL Developer	SCOTT	PUBLIC	(null)	SELECT	(null)	SYS	DUAL
marco	goofymarco	SQL Developer	SCOTT	PUBLIC	(null)	SELECT	(null)	SYS	ALL_TAB

marco	goofymarco	SQL Developer	MDSYS	PUBLIC	(null)	EXECUTE	(null)	SYS	DBMS_ST
marco	goofymarco	SQL Developer	MDSYS	PUBLIC	(null)	SELECT	(null)	SYS	DUAL
marco	goofymarco	SQL Developer	SCOTT	RESOURCE	CREATE TABLE	(null)	(null)	(null)	(null)
marco	goofymarco	SQL Developer	MDSYS	MDSYS	(null)	SELECT	(null)	SYS	DBA_TAB
oracl	localhost.		APEX_040200	PUBLIC	(null)	USE	(null)	SYS	ORA\$BAS
oracl	localhost.	DBMS_SCHEDULE	APEX_040200	PUBLIC	(null)	EXECUTE	(null)	SYS	DEFAULT
oracl	localhost.	DBMS_SCHEDULE	APEX_040200	PUBLIC	(null)	EXECUTE	(null)	SYS	DBMS_OU
oracl	localhost.	DBMS_SCHEDULE	APEX_040200	PUBLIC	(null)	EXECUTE	(null)	SYS	DBMS_OU

Diese und noch mehr als 100 weitere Neuerungen erwarten Sie im [Oracle 12c Neuheiten Kurs](#).