6.4 Erleichterung der Benutzerverwaltung

Hier werden Möglichkeiten erläutert, die die übersichtliche Umsetzung eines systematischen Sicherheitskonzepts für verschiedene Benutzergruppen erleichtern.

6.4.1 Profile

Ein **Profil** ist eine benannte Menge von Resourcengrenzen.
Ein Benutzer kann nur soviele Datenbank- und Instance-Resourcen belegen, wie in seinem Profil angegeben sind.

Beispiel:
```
CREATE PROFILE clerk LIMIT
   SESSIONS_PER_USER=2
   CPU_PER_SESSION unlimited
   CPU_PER_CALL 6000
   LOGICAL_READS_PER_SESSION unlimited
   LOGICAL_READS_PER_CALL 100
   IDLE_TIME 30
   CONNECT_TIME 480;
```

Ein Standardprofil `DEFAULT` legt die Werte für nichtgenannte Resourcengrenzen fest.

Zum Erstellen eines Profils benötigt man die Funktionsberechtigung CREATE PROFILE.

Im **Datenwörterbuch** (*data dictionary*) wird zu jedem Benutzer gespeichert:

- Benutzername, Id-Nummer, Standard-Speicherbereich für Daten bzw. Standard-Speicherbereich für temporäre Relationen, Erzeugungsdatum, Profil und (verschlüsseltem) Passwort
  (in den Datenwörterbuch-Sichten: DBA_USERS, USER_USERS, ALL_USERS),

- Speicherbereich-Quoten (in: USER_TS_QUOTAS, DBA_TS_QUOTAS),

- Profile und Ressourcengrenzen (in: DBA_PROFILES, USER_RESOURCE_LIMITS),

- Kosten der verschiedenen Systemressourcen (in: RESOURCE_COST),

- Speicherverbrauch der aktuellen Sitzungen (in: V$SESSION, V$SESSTAT, V$STATNAME).

Sie können abgefragt werden z.B. mit: `SELECT * FROM SYS.USER_USERS;`

6.4.2 Rollen und Benutzergruppen

Datenbankbenutzer haben i.a. unterschiedliche Aufgaben.
Daher benötigen sie auch unterschiedliche Rechte bzgl. der Datenbank.

Gibt es viele Datenbankbenutzer, so ist es einfacher und übersichtlicher, Benutzerberechtigungen zu strukturieren:

Eine **Rolle** fasst eine Menge von Berechtigungen zusammen.

Eine **Benutzergruppe** fasst eine Menge von Benutzern zusammen.
Oracle unterstützt aber *nur* die Benutzergruppe `PUBLIC`, die alle Benutzer des Datenbanksystems enthält.

Praktisch geht man wie folgt vor:

Zunächst werden die Rollen deklariert (und evtl. zusätzlich mit einem Paßwort geschützt):

```
CREATE ROLE rolename;
```

Entsprechend können Rollen mit `DROP ROLE rolename;` gelöscht werden.

Anschließend werden die gewünschten Objektberechtigungen und (getrennt) Systemberechtigungen einer Rolle zugewiesen.

```
GRANT list_of_object_privileges ON object TO rolename;

GRANT list_of_system_privileges TO rolename;
```

Benutzern kann man dann auf einfache und übersichtliche Art alle Rechte einer Rolle zuweisen; entsprechend kann man allen Benutzern einer Benutzergruppe Rechte zuweisen.

```
GRANT rolename TO list_of_accounts [WITH ADMIN OPTION];
```

Änderungen betreffen dann i.d.R. nur eine Rolle bzw. analog nur eine Benutzergruppe.

Rollen können Benutzern (aber nicht Rollen) `WITH ADMIN OPTION` zugeteilt werden:

Dann kann der Rechteempfänger die Rolle einem beliebigen Benutzer oder einer beliebigen anderen Rolle in der Datenbank zuteilen oder entziehen.

Er kann dabei die ADMIN OPTION weitergeben.

Außerdem kann er die Rolle ändern oder löschen.

Der Erzeuger einer Rolle erhält die ADMIN OPTION automatisch.

Zum Zuteilen einer Rolle benötigt der Absender die ADMIN OPTION für alle Rollen, die zugeteilt werden, oder die generelle Funktionsberechtigung `GRANT ANY ROLE`.

Man kann darüberhinaus alle Rechte einer Rolle einer anderen Rolle zuweisen und so eine Rolle aus anderen Rollen und ggf. weiteren Rechten zusammensetzen:

```
GRANT rolenames TO rolename;
```

Auf diese Art entsteht eine Rollenhierarchie.

Während einer Datenbanksitzung werden Rollen mit `SET ROLE list_of_rolenames;` aktiviert.

Man kann jedoch **Standardrollen** auszeichnen (`ALTER USER account DEFAULT ROLE rolename`), die automatisch aktiviert werden, wenn der Benutzer eine Sitzung (*session*) startet.

Alternativ kann man festlegen, daß der Benutzer alle direkt zugeteilten Rollen als Standardrolle haben soll (`DEFAULT ROLE ALL`),

es dazu Ausnahmen geben soll (`DEFAULT ROLE ALL EXCEPT rolename`), oder

dass er keine Standardrolle haben soll (`DEFAULT ROLE NONE`).

Änderungen betreffen erst die nächste Sitzung.

Im Datenwörterbuch werden zu jedem Benutzer Rollen und Berechtigungen gespeichert:

- Berechtigungen bzw. Rollen, die der Benutzer gegenwärtig hat (in den Datenwörterbuch-Sichten: SESSION_PRIVS, SESSION_ROLES)

- den Benutzern zugeteilte Funktionsberechtigungen (in: USER_SYS_PRIVS, DBA_SYS_PRIVS),

- Rollen der Datenbank (in: DBA_ROLES),
- den Benutzern zugeteilte Rollen (in: USER_ROLE_PRIVS, DBA_ROLE_PRIVS),
- anderen Rollen zugeteilte Rollen, Funktionsberechtigungen, Objektberechtigungen auf Relationsebene (in: ROLE_ROLE_PRIVS, ROLE_SYS_PRIVS, ROLE_TAB_PRIVS)

- Objektberechtigungen auf Relationebene, die der Benutzer (oder die Benutzergruppe PUBLIC) erhalten hat (in: ALL_TAB_PRIVS, USER_TAB_PRIVS, DBA_TAB_PRIVS),
- Objektberechtigungen auf Relationebene, bei denen der Benutzer Eigentümer oder Zuteiler ist: (in: ALL_TAB_PRIVS_MADE, USER_TAB_PRIVS_MADE),
- analog (in: ALL_TAB_PRIVS_RECD, USER_TAB_PRIVS_RECD),

- Objektberechtigungen auf Attributebene, die der Benutzer (oder die Benutzergruppe PUBLIC) erhalten hat (in: ALL_COL_PRIVS, USER_COL_PRIVS, DBA_COL_PRIVS),
- analog (in: ALL_COL_PRIVS_MADE, USER_COL_PRIVS_MADE),
- analog (in: ALL_COL_PRIVS_RECD, USER_COL_PRIVS_RECD),
- Objektberechtigungen auf Operationsebene pro Attribut (in: COLUMN_PRIVILEGES)

6.5 Protokollierung

6.5.1 Regelung durch Datenbankadministratoren

Oracle erlaubt, die Aktivitäten der Datenbankbenutzer zu protokollieren.

Dabei werden keine veränderten Daten der Datenbank aufgezeichnet.

Ein Datenbanksystemadministrator kann die Protokollierungsmöglichkeit an- oder ausstellen (Standardeinstellung ist aus).

Sind sie angestellt (durch Setzen des Parameters AUDIT_TRAIL auf TRUE in INIT.ORA beim Starten einer Instanz), so kann jeder Datenbankadministrator (und hier reichen wohl DBA-Rechte)

- erfolgreiche und/oder nicht erfolgreiche Login/Logout-Versuche aufzeichnen lassen,
- Rechteweitergabe (durch GRANT und REVOKE) aufzeichen lassen,
- Protokollierungsmöglichkeiten für die Oracle-Benutzer zur Verfügung stellen und dafür die Standardeinstellungen setzen.

Wenn man den Handbüchern glauben darf, gelten diese Standardeinstellungen systemweit; da sie jeder datenbankweite DBA setzen darf, dürften sie -systematisch gedacht- eigentlich nur datenbankweit gelten. Sonst könnte jeder Datenbankadministrator die Protokollierungsoptionen für eine andere Datenbank (für die er nicht Administrator ist) beeinflussen!

```
AUDIT {what-list | ALL } [WHENEVER [NOT] SUCCESSFUL]

NOAUDIT {what-list | ALL } [WHENEVER [NOT] SUCCESSFUL]
```

Die what-list kann enthalten:

`CONNECT`	protokolliert logins / logouts zur Datenbank
`DBA`	protokolliert `GRANT, REVOKE, AUDIT, NOAUDIT`-Statements (nur DBA-Versionen?) sowie `CREATE/DROP PUBLIC SYNONYM / DBLINK`
`NOT EXISTS`	protokolliert alle Verweise auf Granule, die mit einem "…does not exist"-Fehler enden.
`RESOURCE`	protokolliert `CREATE/ALTER/DROP TABLE/VIEW/SYNONYM/CLUSTER/ SEQUENCE/TABLESPACE/ROLLBACK SEGMENT/INDEX` Operationen

Die gegenwärtigen systemweiten Protokollierungsoptionen stehen in `DBA_SYS_AUDIT_OPTS`. Die Protokolldaten stehen in `DBA_AUDIT_TRAIL`; es gibt weitere Sichten darauf.

`AUDIT CONNECT WHENEVER NOT SUCCESSFUL`
protokolliert nicht-erfolgreiche login/logout-Versuche

6.5.2 Regelung durch Eigentümer von Datenbankgranulen

Wenn der Datenbankadministrator entsprechende Optionen gesetzt hat, können die Eigentümer von Relationen oder Sichten die Protokollierung veranlassen

- von erfolgreichen (`SUCCESSFUL`) oder
- nicht erfolgreichen Zugriffen (`NOT SUCCESSFUL`) oder
- allen Zugriffen (`WHENEVER`-Klausel weglassen)

auf Tabellen (tablename / `DEFAULT` (nur für DBA zum Setzen der Standardprotokollierungsparameter)).

Die Protokollierung kann auf vorgegebene Typen von SQL-Operationen eingeschränkt werden (`ALL` // `ALTER` / `AUDIT` / `COMMENT` / `CREATE` / `DELETE` / `EXECUTE` / `GRANT` / `INDEX` / `INSERT` / `LOCK` / `READ` / `REFERENCE` / `RENAME` / `SELECT` / `UPDATE` / `WRITE`).

Außerdem kann der Detaillierungsgrad angegeben werden (`SESSION` / `ACCESS`).
`BY SESSION` liefert einen Eintrag pro Oracle-Sitzung,
nämlich eine Statistik der erfolgreichen und nicht erfolgreichen Zugriffe.
`BY ACCESS` liefert einen Eintrag pro Zugriff auf ein Datenbankgranul.

Achtung: Die Protokollierungseinträge werden nach dem Parsen des Statements, aber vor seiner Ausführung gemacht. Falls also ein Statement zurückgesetzt werden muss, so wird es trotzdem protokolliert.

```
AUDIT ALTER ON EMP BY ACCESS WHENEVER SUCCESSFUL

NOAUDIT ALL ON EMP
```

Das Ergebnis der Protokollierung kann über die vordefinierte Data Dictionary Sicht `USER_AUDIT_TRAIL` angeschaut werden und ist wie folgt codiert:

Für jeden Typ wird ein Zeichen geschrieben.

Die Operation wurde
- gar nicht (`-`),
- erfolgreich (`S`),
- nicht erfolgreich (`F`),
- mal erfolgreich mal nicht erfolgreich (`B`)

ausgeführt.

Die Standardeinstellung für die Protokollierung steht in der Data Dictionary Sicht `ALL_DEF_AUDIT_OPTS`.

Die vom Benutzer spezifizierten Protokollierungsanweisungen stehen in `USER_TAB_AUDIT_OPTS`.

```
SELECT * FROM USER_TAB_AUDIT_OPTS;

TABLE_NAME O_TYPE ALT AUD COM CRE DEL EXE GRA IND INS LOC REA REN SEL UPD
---------- ------ --- --- --- --- --- --- --- --- --- --- --- --- --- ---
EMP        TABLE  -/- -/- -/- -/- -/- -/- -/- -/- -/- -/- -/- -/- -/- -/-
ACCOUNTS   TABLE  S/S -/- S/S -/- S/S -/- S/S -/- S/S A/- -/- -/- S/S S/S
```

- keine Protokollierung, A Protokollierung pro Zugriff, S Protokollierung pro Sitzung

Vor dem Schrägstrich steht die Option für erfolgreiche Zugriffe,

nach dem Schrägstrich die Option für nicht erfolgreiche Zugriffe.

(`REF` und `WRI` aus Platzgründen weggelassen)