# 6 Sicherheitskonzepte in Oracle

Datenbanksysteme sind für viele Benutzer(-gruppen) mit unterschiedlichen Nutzungswünschen geschaffen, aber nicht alle Daten des Informationssystems sind für alle Benutzer bestimmt.

Daher trifft man Maßnahmen, um die Daten nur bestimmten Personen zugänglich machen, insbesondere vergibt man Berechtigungen

- zur Benutzung des Datenbankssystems (**Zugangsberechtigung**),
- zur Benutzung bestimmter Datenbankoperationen (**Funktionsberechtigung**), und
- zur Benutzung bestimmter Datenbankobjekte (**Objektberechtigung**).

Oracle enthält eine Reihe dieser konventionellen Sicherheitskonzepte in Datenbanksystemen.

# 6.1 Zugangsberechtigungen

Der Zugriff auf die Datenbank sollte nur berechtigten Benutzern vorbehalten sein.

**Zugangsberechtigungen** sollen die Datenbank vor Zugriffen von unberechtigten Personen sichern.

Zugangsberechtigungen bestehen z.Zt. i.a. aus einer **Datenbankkennung** (*account, user name*) und einem **Paßwort** für das Datenbanksystem[1].

Hat ein Benutzer sein Paßwort nicht selber bestimmt, so sollte er sein Paßwort sofort ändern:

```
ALTER USER username IDENTIFIED BY neues_paßwort
```

Gute Systeme verlangen auch das alte Paßwort zur Paßwortänderung
(so können mißbräuchliche Paßwortänderungen z.B. bei kurzfristig verlassenen Terminals verhindert werden).

Die Zugangserlaubnis eines Datenbank-Benutzers wird wahlweise geprüft

- vom Betriebssystem (dann verlässt man sich auf die Überprüfung der Zugangsberechtigung des Benutzers für das Betriebssystem) (`IDENTIFIED EXTERNALLY`) oder
- direkt vom Datenbanksystem (`IDENTIFIED BY` <passwort>).

---

[1] Vorsicht: In Oracle z.B. kann das Datenbanksystem statt mit `sqlplus` auch dialog-abkürzend gestartet werden mit
`sqlplus account/password`
Diese Kommandozeile kann aber unter UNIX von jedem anderen Benutzer mittels `ps` gelesen werden!!!

In Oracle kann bei der Vergabe einer Datenbankkennung an neue Datenbankbenutzer zugleich festgelegt werden,

- ob eine Authentifikation durch das Datenbanksystem erfolgen soll oder über das Betriebssystem,
- ein Standard-**Speicherbereich** für Relationen festgelegt werden (`DEFAULT TABLESPACE`) und
- ein Speicherbereich für kurzlebige Datenbankobjekte (`TEMPORARY TABLESPACE`),
- Speicherobergrenzen, sog. **Quoten**, für Speicherbereiche festgelegt werden, sowie
- ein Profil (siehe Kapitel 6.4.1):

**Beispiel**:

```
CREATE USER OPS$jward
IDENTIFIED EXTERNALLY
DEFAULT TABLESPACE data_ts
TEMPORARY TABLESPACE temp_ts
QUOTA 100M ON test_ts
QUOTA 500K ON data_ts
PROFILE clerk;
```

Allerdings kann ein neuer Benutzer erst auf eine Datenbank zugreifen, nachdem er Funktionsberechtigungen erhalten hat, die `CREATE SESSION` miteinschließt.

Soll sich das Datenbanksystem auf die Überprüfung der Zugangserlaubnis des Betriebssystems verlassen, so muß man (den Oracle-Parameter `REMOTE_OS_AUTHENT` auf true setzen und) als Datenbankkennung `OPS$`<BS-account> benutzen.

Dieser Präfix `OPS$` kann mit Hilfe des Parameters `OS_AUTHENT_PREFIX` (z.B. in die leere Zeichenkette `""`) geändert werden.

# 6.2 Funktionsberechtigungen für Benutzer

Oracle unterscheidet zwischen zwei Arten von Berechtigungen. Beide Rechte werden in Oracle durch syntaktisch sehr ähnliche Kommandos vergeben.

- Eine **Funktionsberechtigung** (*system privilege*) erlaubt die Ausführung einer (Klasse von) Datenbankoperation(en).

- Eine **Objektberechtigung** (*object privilege*) erlaubt den Zugriff auf ein Datenbankobjekt.

Funktionsberechtigungen werden in Oracle vergeben bzw. entzogen durch das Kommando

```
GRANT list_of_system_privileges_or_roles TO list_of_accounts_or_roles
[WITH ADMIN OPTION];

REVOKE list_of_system_privileges_or_roles FROM list_of_accounts_or_roles;
```

Werden Funktionsberechtigungen mit dem Zusatz `WITH ADMIN OPTION` zugeteilt, so kann der Rechteempfänger die Funktionsberechtigung einem beliebigen Benutzer oder einer beliebigen anderen Rolle in der Datenbank zuteilen oder entziehen.
Er kann dabei die ADMIN OPTION weitergeben.

Zum Zuteilen einer Funktionsberechtigung benötigt der Absender die ADMIN OPTION für alle Funktionsberechtigungen (und Rollen), die er zuteilt.

Grob eingeteilt, unterscheidet man Datenbankbenutzer mit vier hierarchisch geordneten Rechtestufen:

- **Datenbanksystemadminstrator** (DBSA),
- **Datenbankadministrator** (DBA),
- Benutzer, die Relationen anlegen dürfen (Eigentümer von Datenbankobjekten) und
- Benutzer, die Relationen nutzen dürfen (Nutzer von Datenbankobjekten).

Typische Funktionsberechtigungen von Oracle sind für

- **Benutzer, die Relationen nutzen dürfen**:

  | | |
  |---|---|
  | eigene Sitzungen erzeugen: | [CREATE \| ALTER] SESSION, |
  | Sichten, Synonyme etc. erzeugen: | CREATE [DATABASE LINK \| SYNONYM \| VIEW], |
  | Entscheidung bei Transaktion erzwingen: | FORCE TRANSACTION, |

- **Benutzer, die Relationen anlegen dürfen**:

  | | |
  |---|---|
  | eigene Relationen, Cluster etc. anlegen: | CREATE [CLUSTER \| PROCEDURE \| SEQUENCE \| SNAPSHOT \| TABLE \| TRIGGER], |

- **Datenbankadministratoren**:

| | |
|---|---|
| öffentliche Synonyme etc. anlegen: | [CREATE \| DROP] PUBLIC [DATABASE LINK \| SYNONYM |
| fremde Relationen etc. anlegen / löschen: | [CREATE \| DROP] ANY [CLUSTER \| INDEX \| PROCEDURE SEQUENCE \| SNAPSHOT \| SYNONYM \| TABLE \| TRIGGER VIEW], |
| fremde Relationen etc. ändern: | ALTER ANY [CLUSTER \| INDEX \| PROCEDURE \| SEQUENCE \| SNAPSHOT \| TABLE], |
| fremde Prozeduren ausführen: | EXECUTE ANY PROCEDURE, |
| Anfragen auf fremden Daten stellen: | SELECT ANY [SEQUENCE \| TABLE], |
| fremde Daten modifizieren etc.: | [BACKUP \| LOCK \| COMMENT \| INSERT \| UPDATE \| DELETE] ANY TABLE, |
| fremde Transaktionen … | FORCE ANY TRANSACTION, |
| fremde Relationen etc. analysieren: | ANALYSE ANY, |
| fremde Relationen etc.protokollieren | AUDIT ANY, |

| | |
|---|---|
| Benutzeridentität (für Import) wechseln: | BECOME USER |
| Datenbankparameter ändern: | ALTER DATABASE, |

- **Datenbanksystemadministratoren**:

| | |
|---|---|
| Benutzer verwalten: | [CREATE \| ALTER \| DROP] USER, |
| Rollen verwalten: | CREATE ROLE, [ALTER \| DROP \| GRANT] ANY ROLE, |
| Berechtigungen weitergeben: | GRANT ANY PRIVILEGE, |
| Resourcekosten und -grenzen verwalten: | [CREATE \| ALTER \| DROP] PROFILE,<br>ALTER RESOURCE COST, |
| Speicherbereiche verwalten: | [CREATE \| ALTER \| MANAGE \| DROP] TABLESPACE,<br>UNLIMITED TABLESPACE, |
| Protokollierungsoptionen einstellen: | AUDIT SYSTEM |
| Datenbank-Fehlerbehandlung: | [CREATE \| ALTER \| DROP] ROLLBACK SEGMENT |
| für STARTUP RESTRICT: | RESTRICTED SESSION, |
| Systemparameter ändern: | ALTER SYSTEM |

# 6.3 Objektberechtigungen für Benutzer

Nach der Erzeugung einer Relation (oder Sicht) hat der Erzeuger als **Eigentümer** das Recht, die Relation zu lesen oder zu verändern (im weitesten Sinn).

Ein Rechteinhaber kann ein Recht an andere Benutzer weitergeben (GRANT-Statement) oder zurückrufen (REVOKE-Statement).

Objektberechtigungen werden vergeben bzw. entzogen durch

```
GRANT list_of_object_privileges ON object TO list_of_accounts_or_roles
[WITH GRANT OPTION];

REVOKE list_of_object_privileges ON object FROM list_of_accounts_or_roles;
```

**Beispiel**: Benutzer mit dem DB-Account `scott` ändert die Zugriffsberechtigung für seine Relation `emp`:

```
GRANT SELECT, INDEX, UPDATE ON emp TO jennifer

REVOKE INDEX ON emp FROM jennifer
```

Jennifer kann auf diese Relation unter dem Namen `scott.emp` zugreifen.

**Wer** kann das Recht bekommen?

- jeder einzelne Benutzer (durch Angabe des Oracle-Benutzeraccounts)
- alle Benutzer (durch Angabe des Schlüsselwortes PUBLIC)
- eine Rolle (siehe Kapitel 6.4.2)

Für welche **Operation** kann man das Recht bekommen?

- zum Lesen (SELECT) (anwendbar auf Relationen, Sichten, Snapshots, Sequenzen)
- zum Einfügen (INSERT) (anwendbar auf Relationen, Sichten)
- zum Ändern (UPDATE) (anwendbar auf Relationen, Sichten)
- zum Löschen (DELETE) (anwendbar auf Relationen, Sichten)
- zum Ausführen (EXECUTE) (anwendbar auf Prozeduren)
- zum Indexanlegen (INDEX) (anwendbar auf Relationen)
  (und so als Eigentümer des angelegten Index implizit das Recht, den Index zu löschen)
- für Schemaerweiterungen oder -änderungen (ALTER) (anwendbar auf Relationen, Sequenzen)
- zur Verwendung von Verweisen innerhalb von Relationen oder Attributbeschränkungen (REFERENCES) (anwendbar auf Relationen)

ALL oder ALL PRIVILEGES teilt alle verfügbaren Operationen für ein Objekt zu bzw. entzieht sie.

Viele der hier nicht aufgeführten Schema-Objekte (z.B. Cluster, Indizes, Trigger, Database Links) werden nur durch Funktionsberechtigungen oder das Recht des Eigentümers gesteuert.

Die Berechtigung für UPDATE und REFERENCES kann außerdem gezielt für einzelne Attribute vergeben werden (z.B. UPDATE(ename, empno) ).

Für Sichten sind nur die Operationen SELECT, INSERT, UPDATE, DELETE sinnvoll;
für SEQUENZEN nur SELECT und ALTER.

Achtung: Änderungsoperationen auf Sichten sind aber nicht möglich, wenn die Sicht durch Ausdrücke, Verbundoperationen oder Mengenoperationen gebildet wurde.

Eigentümer einer Sicht können ein GRANT SELECT auf die Sicht nur dann vergeben, wenn sie auch das Recht auf die zugrundeliegende Relation (noch) haben.
D.h. durch Erzeugung einer Sicht kann man fehlende Rechte nicht ersetzen.

Für **welches Datenbankobjekt** kann man das Recht bekommen?

- für Relationen
- für Sichten
- für persistente Prozeduren

**Sichten** sind ein wichtiges Hilfsmittel, um den Datenausschnitt, auf den zugegriffen werden darf, zu begrenzen, sowohl

- horizontal (durch geeignete Bedingungen in der WHERE-Klausel) wie auch
- vertikal (durch Auswahl der Attribute in der SELECT-Klausel) oder gar nur
- auf Aggregationen der Relation (durch Verwendung von Aggregatfunktionen in der SELECT-Klausel).

Es werden dann die Rechte nur für die Sicht und nicht für die zugrundeliegenden Relationen vergeben.

```
CREATE VIEW sales AS
   SELECT ename, empno, mgr
   FROM emp
   WHERE job='SALESMAN'

GRANT SELECT, UPDATE ON sales TO salesmgr
```

Mit **persistenten Prozeduren** kann die Benutzung von Operationen beschränkt werden.

Statt den Lese- und/oder Schreibzugriff allgemein zu erlauben, erlaubt man nur die Ausführung vordefinierter Prozeduren (z.B. Überweisung statt Schreibe Kontostand)

Wer darf (**Teil-**)**Rechte** an andere Benutzer **weitergeben**?

- der Erzeuger des Datenbankobjekts stets;

- andere Benutzer (nicht aber Rollen) nur, wenn sie das Recht ausdrücklich mit der **Weitergabeoption** `WITH GRANT OPTION` erhalten haben;

```
GRANT SELECT, INDEX, UPDATE ON emp TO jennifer WITH GRANT OPTION
```

In manchen anderen Datenbanksystemen (z.B. Sybase) können weitergegebene Rechte ihrerseits nicht weitergegeben werden.

## Zugriffsbeschränkung auf ausgezeichnete Terminals

Mit Hilfe der Funktion `USERENV` kann der Datenzugriff auf bestimmte Terminals eingeschränkt werden:

```
CREATE VIEW myview AS
SELECT * FROM mysalary
WHERE USERENV(´TERMINAL´)=´RTA0´;

GRANT SELECT ON myview TO evelyn;
```

Die Sicht `myview` ist jetzt gleich der Relation `mysalary`, wenn `evelyn` am Terminal `RTA0` eingeloggt ist und leer sonst!

## Zugriffsbeschränkung auf ausgezeichnete Oracle-Benutzer (innerhalb von SQL-Anweisungen)

Es gibt ein Pseudoattribut `USER`, das den Oracle-Benutzernamen des gegenwärtigen Oracle-Benutzers liefert. Mit Hilfe dieses Pseudoattributs `USER` kann der Zugriff auf bestimmte Oracle-Benutzer eingeschränkt werden, z.B.:

```
SELECT * FROM salary
WHERE USER=´JIMMY´;
```