

# Label Security in DB2 und Oracle

---

Artikel von Heinz Axel Pürner, veröffentlicht in IT-Focus 01/02-2007

**Die steigenden Anforderungen an die Sicherheit der Daten haben zu verbesserten Lösungsansätzen geführt. So hat gerade IBM in DB2 UDB V9.1 mit der *Label Based Access Control* seinen Ansatz der Mandatory Access Control (MAC) implementiert. Das ist ein Anlaß, im folgenden diese Sicherheitsfunktion in DB2 UDB V9.1 im Vergleich zu der vergleichbaren in Oracle 10g vorzustellen und zu zeigen, wie sie einfach genutzt werden kann.**

Einführung in die Problematik

Strengere gesetzliche Regelungen und weitergehende, geschäftlich begründete Anforderungen an den Schutz von Informationen verlangen danach, dass Daten über die bisherigen Absätze hinaus geschützt und ihre Nutzung nachgewiesen werden müssen. Dieses führt zu mehrstufigen Sicherheitsverfahren (*Multilevel Security*), die die Klassifikation von Objekten (Daten) und Subjekten (Benutzern) basierend auf einem System von hierarchischen Sicherheitsstufen und nicht hierarchischen Sicherheitskategorien erlauben. Dabei sollen unautorisierte Benutzer am Zugriff und an der Herabstufung der Sicherheitsklassifikation von Daten gehindert werden.

Der klassische Ansatz im Standard-SQL ist die Discretionary Access Control (DAC). Dabei steht ein Objekt unter der Kontrolle seines Eigentümers, also des Benutzers, der es erstellt hat. Dieser vergibt oder widerruft die Berechtigungen an dem Objekt. Er kann es verändern oder löschen.

Dieser Ansatz kennt auch nur die Vergabe von Berechtigungen auf Objekt-Ebene, nicht aber auf der Ebene von Zeilen oder Spalten und auch nicht in Abhängigkeit von den Dateninhalten. Zwar kann man mit Hilfe von Datensichten Zeilen und Spalten ausblenden, bei Zeilen auch in Abhängigkeit von Datenwerten, aber diesen Möglichkeiten sind Grenzen gesetzt. Außerdem steigt mit wachsenden Anforderungen an die Zugriffssteuerung die Komplexität der Objektstrukturen erheblich an.

Ein beliebter Ansatz zur Implementierung der Sicherheitsanforderungen an Anwendungssysteme ist die Realisierung dieser Anforderungen fast ausschließlich im Code der Anwendungen: Hier können beliebige Anforderungen flexibel, aber mit entsprechendem Aufwand realisiert werden. Wird aber die Anwendung umgangen oder wird über andere Werkzeuge auf die Daten zugegriffen, so sind die implementierten Berechtigungssteuerungen nicht wirksam. Beliebte ist es in diesem Zusammenhang auch, von dem Anwendungsserver mit internen Benutzer-IDs auf die Datenbank zuzugreifen. Diese Benutzer-IDs verfügen dann über den vollen Berechtigungsumfang für den Zugriff auf die Daten, unabhängig davon, welche Berechtigungen der reale Benutzer am Bildschirm eigentlich hätte. Werden diese Benutzer-IDs gehackt (missbräuchlich oder fahrlässig bekannt), so ist die Datenbank offen wie ein Scheunentor!

Der Lösungsansatz

Mit der MAC (Mandatory Access Control) weicht man von der Objektkontrolle durch den Eigentümer ab. Sie beruht auf Richtlinien (Policies), die nicht durch einzelne Benutzer verändert werden können. Jedem Objekt der Datenbank wird eine Sicherheitsklassifikation zugewiesen und jedem Subjekt (Benutzer oder Programm) eine Freigabe für bestimmte Sicherheitsklassifikationen. Durch Abgleich von Klassifikation und Freigabe wird ermittelt, ob ein Benutzer auf ein Objekt zugreifen darf. Üblicherweise wird MAC in Ergänzung zu DAC eingesetzt.

Ein Weg, ein MAC-Sicherheitssystem zu realisieren, ist die Nutzung von Kennsätzen (*Labels*). Die Kennsätze enthalten die Klassifikation der Daten entsprechend ihrer Sensitivität (*object labels*) oder die Berechtigungen des Benutzers (*subject labels*). Über diese Labels werden lesende und schreibende Zugriffe auf der Ebene von Zeilen und Spalten der Tabellen einer Datenbank gesteuert.

Ein Sicherheitsadministrator erstellt Sicherheitsrichtlinien (*Policies*) und definiert die Kennsätze für Objekte und Benutzer.

Die führenden Hersteller von Datenbank-Managementsystemen Oracle und IBM bieten heute kennsatzgesteuerte Sicherheitslösungen als Bestandteil ihrer Datenbank-Produkte an. Diese Lösungen sollen ohne eigenen Programmieraufwand durch den Anwender nutzbar sein.

Die Vorteile dieser Sicherheitsverfahren sind:

- Zentrale Implementierung in der Datenbank
- Nicht zu umgehen
- Kein Codier-Aufwand

Für den Einsatz der Label Security empfiehlt sich folgende Vorgehensweise:

- Erhebung der Sicherheitsanforderungen durch
  - Analyse der Anwendungen
  - Analyse der verarbeiteten Informationen
  - Analyse der Benutzeranforderungen

- Erstellung eines Sicherheitskonzepts mit
  - Klassifikation der Daten entsprechend ihrer Sensitivität
  - Zugriffskriterien der Benutzer
  - Festlegung der Benutzer-Freigaben

Die technische Implementierung erfolgt in den Schritten:

1. Definition von Policy und deren Komponenten
2. Definition der Daten-Labels
3. Definition der Benutzer-Labels
4. Anwenden der Policy auf die Objekte (Tabellen)
5. Erzeugen der konkreten Label-Werte in den Objekten

In den folgenden Abschnitten werden die Implementierungen der kennsatz-gesteuerten Sicherheitsfunktionen in DB2 UDB for LUW und Oracle näher betrachtet.

#### DB2 UDB V9.1 LBAC

Zu den neuen Funktionen in DB2 for LUW Version 9 zählt auch die Berechtigungssteuerung *Label Based Access Control (LBAC)*. Dazu gehört auch ein neues Berechtigungsprofil, der *Security Administrator (SECAM)*.

Nur der *Security Administrator* definiert die Sicherheitsrichtlinien (*Security Policies*) und Kennsätze (*Labels*). Er vergibt oder widerruft Label-Berechtigungen. Dazu ist kein anderes Profil befugt, auch nicht der System-Administrator (*SYSADM*). Er ordnet die Sicherheitsrichtlinien den Objekten (Tabellen) zu. Ein Objekt kann nur eine Sicherheitsrichtlinie besitzen.

Die *Security Policies* bestehen aus Komponenten, die jeweils eines der Kriterien abbilden, nach denen der Zugriff gesteuert wird. Solche Kriterien können Vertraulichkeitsstufen von Daten, Abteilungszugehörigkeit oder Niederlassungsort sein. Für die Komponenten stehen drei Strukturen zur Verfügung:

- *Set* (Aufzählung)
- *Array* (geordnete Aufzählung, einfache Hierarchie)
- *Tree* (Baumstruktur, komplexe Hierarchie)

Eine *Security Policy* kann aus maximal 16 Komponenten bestehen.

Die *Security Labels* werden Tabellenspalten oder Tabellenspalten zugeordnet. Durch den Vergleich der Labels von Daten und Benutzern wird festgestellt, ob diese zum Zugriff berechtigt sind. Ist ein Benutzer nicht zum Zugriff berechtigt, behandelt DB2 die Daten in einem solchen Fall, als ob sie nicht existieren. Auch Funktionen wie *COUNT()* oder *SUM()* berücksichtigen nur die Daten, für deren Zugriff der Benutzer berechtigt ist. Greift allerdings ein Benutzer auf Spalten zu, für die er nicht berechtigt ist, erhält er eine Fehlermeldung.

#### ORACLE

Oracle verfügt schon länger über weitergehende Lösungsansätze zur Datensicherheit. Schon in der Version 8 stellte Oracle mit seiner Technologie der *Virtual Private Database (VPD)* einen Werkzeugkasten (Toolkit) zur Implementierung von individuellen Zugriffskontrollen vor. Einfach dargestellt ermöglicht VPD für definierte Tabellen die automatische Generierung von Prädikaten, die den *WHERE*-Klauseln der *SQL*-Befehle eines Benutzer von diesem unbemerkt hinzugefügt werden. Zur Nutzung von VPD müssen eigene Routinen in Oracle erstellt werden.

Leider findet man bei Oracle mehr als einen Begriff für diese Sicherheitsfunktionen. Neben VPD wird auch der Begriff *Fine Grained Access Control (FGAC)* verwendet oder von *Row Level Security (RLS)* gesprochen.

In der Version 9 hat Oracle mit *OLS (Oracle Label Security)* eine fertige Lösung mit *Security Labels* auf Basis der VPD-Technologie zur Verfügung gestellt. Sie erfordert keine individuelle Programmierung mehr.

Auch in Oracle werden Sicherheitsrichtlinien (*Policies*) und Komponenten definiert. Allerdings bestehen bei ORACLE die *Security Labels* aus drei vorgegebenen Komponenten:

- der hierarchischen Stufe (*Level*)
- der optionalen Aufteilung (*Compartment*) - nicht hierarchisch
- der optionalen Gruppe (*Group*) - kann hierarchisch definiert werden

Zur Verwaltung der *Label Security* besitzt Oracle die Rolle *LBAC\_DBA*, die der User-ID des Sicherheitsadministrators zugeordnet wird. Außerdem wird bei Aktivierung der *Label Security* in der Datenbank die User-ID *LBACSYS* als Eigentümer der entsprechenden Objekte definiert.

Im Gegensatz zu DB2 stehen in Oracle die Systemadministratoren - bei Anmeldung als *SYSDBA* - oberhalb der *Label Security*. Sie haben also uneingeschränkten Zugriff auf die sonst durch Labels geschützten Daten.

Zur Unterstützung bei der Einrichtung und Pflege der Sicherheitsrichtlinien verfügt Oracle mit dem *OPM (Oracle Policy Manager)* über ein grafisches Werkzeug, mit dem in den meisten Fällen die Definitionen komfortabel und ohne *SQL*-Kenntnisse angelegt werden können.

## Ein Beispiel

Statt theoretischer Erläuterungen über die Unterschiede zwischen den Implementierungen in DB2 for LUW und Oracle soll ein kleines Beispiel die Funktionsweise von Label Security sowie die Vor- und Nachteile der jeweiligen Implementierungen deutlich machen:

Die kleine Republik von Costa Banana hat die Zeichen der Zeit erkannt und eine zentrale Datei zur Terroristenbekämpfung eingerichtet. In diese Datei bringen Polizei und Geheimdienst ihre registrierten Terrorverdächtigen mit persönlichen Stammdaten, Bewegungsdaten, Verweisen auf Kriminal- und Geheimdienst-Dateien und sonstigen Kennzeichen ein. Die Datei soll Polizisten und Geheimdienstlern zur Verfügung stehen. Da allerdings Costa Banana in 4 autonome Provinzen (Nord, Süd, West, Ost) aufgeteilt ist, dürfen die Polizisten nur die Daten der Verdächtigen sehen, für die Dienststellen ihrer Provinz zuständig sind. Nur der zentrale Geheimdienst hat Zugriff auf alle Daten.

Verantwortlich für die Pflege der von der Polizei eingebrachten Daten sind die Sonderkommissionen "Anti-Terror" in den vier Provinzen.

Da die Datenfelder von unterschiedlicher Sensitivität sind, stehen sie auch nicht allen Polizisten zur Verfügung.

Neben der regionalen Einteilung gibt es die Beschränkung, dass normale Polizisten nur als terrorverdächtig Eingestufte (TV) sehen dürfen, ihre vorgesetzten Kriminaldirektoren auch die als aktiv eingestuft (AT) und nur die Sonderkommissionen die Anführer (Top Terroristen - TT).

Die Polizei darf die Datenfelder IDGEHDienst (ID DB Geheimdienst) und VMANNKZ (V-Mann-Kennzeichen) nicht sehen. Aus Sicherheitsgründen ist VMANNKZ auch im Geheimdienst nur ausgewählten Beamten zugänglich.

Innerhalb der Polizei sind die Felder GEFEINSTUFUNG (Gefährdungseinstufung), IDKRIMREG (ID im Kriminal-Register), IDTERRORNETZ und BEZTERRORNETZ (ID und Bezeichnung in der Datei "Terrornetzwerke") nur den Sonderkommissionen zugänglich.

In Vertretung der möglichen Benutzer betrachten wir die folgenden:

- "Harry" - ein einfacher Polizist der Nordprovinz
- "Derrick" - ein Kriminaldirektor der Ostprovinz
- "Soko" - ein Mitglied der Sonderkommission der Westprovinz
- "Schlphut" - ein anonymer Angehöriger des Geheimdiensts
- "Bond" - ein Special Agent des Geheimdiensts mit V-Mann-Berechtigung

Die Zugriffsberechtigungen sollen mit Hilfe der kennsatz-gesteuerten Sicherheitsfunktionen in DB2 und Oracle implementiert werden.

Die Implementierung der Anti-Terror-Datei erfolgt als Tabelle eines relationalen DBMS. Sie hat folgenden Inhalt:

- PNR - Eindeutige Identifikation
- NAME - Name
- VORNAME - Vorname
- NAMENSZUSATZ - Namenszusatz, Geburtsname oder Deckname
- HERKUNFTSLAND - Staat, in der die Person geboren wurde
- GESCHLECHT - Kennzeichen für Geschlecht (M, F)
- GEBDAT - Geburtsdatum
- ERSTEINDAT - Datum der ersten Einreise
- LETZTEINDAT - Datum der letzten Einreise
- LETZTAUSDAT - Datum der letzten Ausreise
- STAATSANGEH - aktuelle Staatsangehörigkeit
- ZUST\_DIENSTS\_ST - zuständige Polizei-Dienststelle
- PROVINZ - Provinz der zuständige Polizei-Dienststelle
- GEFEINSTUFUNG - Einstufung der von der Person ausgehenden Gefährdung
  - TV - Terrorverdächtig
  - AT - Aktiver Terrorist
  - TT - Rädelsführer, Top Terrorist
- VMANNKZ - Kennzeichen, ob als V-Mann tätig
- IDKRIMREG - Identifikation im Kriminalregister
- IDTERRORNETZ - Identifikation des Terroristen-Netzwerks, dem die Person angehört
- BEZTERRORNETZ - Kurzbezeichnung des Terroristen-Netzwerks, dem die Person angehört
- IDGEHDienst - Identifikation in der Datenbank des Geheimdiensts

## Die DB2-Implementierung

Zuerst betrachten wir die Implementierung in DB2:

IBM hat die bekannten SQL-Befehle wie ALTER, CREATE, GRANT oder REVOKE zur Implementierung der LBAC-Funktionalität erweitert. Damit sind die neuen Befehle leicht erlernbar. Das grafische Werkzeug für den Administrator, das Control Center, unterstützt bisher nicht die Definition der LBAC-Funktionalitäten.

Die Tabelle hat in DB2 folgende Struktur:

```
CREATE TABLE "DB2INST9"."ATDATEICB" (
    "PNR" BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY (
        START WITH +1
        INCREMENT BY +1
        MINVALUE +1
        MAXVALUE +9223372036854775807
        NO CYCLE
        CACHE 25
        NO ORDER ) ,
    "NAME" CHAR(24) NOT NULL ,
    "VORNAME" CHAR(24) NOT NULL ,
    "NAMENSZUSATZ" CHAR(24) ,
    "HERKUNFTSLAND" CHAR(4) NOT NULL ,
    "GESCHLECHT" CHAR(1) NOT NULL ,
    "GEBDAT" DATE ,
    "ERSTEINDAT" DATE NOT NULL ,
    "LETZTEINDAT" DATE NOT NULL ,
    "LETZTAUSDAT" DATE ,
    "STAATSANGEH" CHAR(4) NOT NULL ,
    "ZUST_DIENSTS_ST" CHAR(16) NOT NULL ,
    "PROVINZ" CHAR(4) NOT NULL ,
    "GEFEINSTUFUNG" CHAR(2) NOT NULL ,
    "VMANNKZ" CHAR(1) ,
    "IDKRIMREG" BIGINT ,
    "IDTERRORNETZ" BIGINT ,
    "BEZTERRORNETZ" CHAR(16) ,
    "IDGEHDienst" BIGINT )
IN "USERSPACE1" ;

COMMENT ON TABLE "DB2INST9"."ATDATEICB" IS 'Anti-Terror-Datei Costa Banana';
```

Zur Vergabe einer eindeutigen Identifikation PNR wird die automatische Generierung einer fortlaufenden Zahl gewählt. Diese ist zugleich Primärschlüssel:

```
ALTER TABLE "DB2INST9"."ATDATEICB"
    ADD CONSTRAINT "PK_ATDATEICB" PRIMARY KEY
    ("PNR");
```

Die Security Policy und ihre Komponenten sowie die Security Labels werden von Sicherheitsadministrator (Profil SECADM) definiert. Vor der Definition der Policy müssen ihre Komponenten angelegt werden. Eine Policy kann bis zu 16 Komponenten besitzen. Einer Tabelle kann aber nur eine Policy zugeordnet werden.

Zum Schutz der Zeilen gegen unberechtigten Zugriff werden 2 Komponenten benötigt. Mit der ersten werden die Gefährdungseinstufungen hierarchisch abgebildet:

```
CREATE SECURITY LABEL COMPONENT SLC_GKZ
ARRAY ['TT', 'AT', 'TV'];
```

Mit der zweiten wird die staatliche Struktur von Costa Banana (Zentralstaat mit 4 Provinzen) modelliert:

```
CREATE SECURITY LABEL COMPONENT SLC_STAAT
TREE ('Zentralstaat' ROOT,
'Nordprovinz' UNDER 'Zentralstaat',
'Ostprovinz' UNDER 'Zentralstaat',
'Suedprovinz' UNDER 'Zentralstaat',
'Westprovinz' UNDER 'Zentralstaat'
);
```

Für den Schutz der Tabellenspalten wird eine dritte Komponente mit hierarchischer Struktur benötigt:

```
CREATE SECURITY LABEL COMPONENT SLC_LVL
ARRAY ['streng-geheim', 'geheim', 'vs', 'frei'];
```

Die Sicherheitsrichtlinie umfasst diese drei Komponenten:

```
CREATE SECURITY POLICY ANTI_TERROR_POLICY
COMPONENTS SLC_LVL, SLC_STAAT, SLC_GKZ
WITH DB2LBACRULES
RESTRICT NOT AUTHORIZED WRITE SECURITY LABEL;
```

Dann sind die Kennsätze (Labels) mit den Werten zu definieren, die in der weiteren Verarbeitung benötigt werden. Für die zu schützenden Spalten werden folgende Labels angelegt:

```
CREATE SECURITY LABEL ANTI_TERROR_POLICY.VS
COMPONENT SLC_LVL 'vs';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.GH
COMPONENT SLC_LVL 'geheim';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.SGH
COMPONENT SLC_LVL 'streng-geheim';
```

Für die zu schützenden Zeilen werden Labels mit den Kombinationen aus SLC\_STAAT und SLC\_GKZ gebildet.

```
CREATE SECURITY LABEL ANTI_TERROR_POLICY.NPROV_POL
COMPONENT SLC_STAAT 'Nordprovinz', COMPONENT SLC_GKZ 'TV';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.OPROV_POL
COMPONENT SLC_STAAT 'Ostprovinz', COMPONENT SLC_GKZ 'TV';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.SPROV_POL
COMPONENT SLC_STAAT 'Suedprovinz', COMPONENT SLC_GKZ 'TV';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.WPROV_POL
COMPONENT SLC_STAAT 'Westprovinz', COMPONENT SLC_GKZ 'TV';
```

```
CREATE SECURITY LABEL ANTI_TERROR_POLICY.NPROV_KD
COMPONENT SLC_STAAT 'Nordprovinz', COMPONENT SLC_GKZ 'AT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.OPROV_KD
COMPONENT SLC_STAAT 'Ostprovinz', COMPONENT SLC_GKZ 'AT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.SPROV_KD
COMPONENT SLC_STAAT 'Suedprovinz', COMPONENT SLC_GKZ 'AT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.WPROV_KD
COMPONENT SLC_STAAT 'Westprovinz', COMPONENT SLC_GKZ 'AT';
```

```
CREATE SECURITY LABEL ANTI_TERROR_POLICY.NPROV_SK
COMPONENT SLC_STAAT 'Nordprovinz', COMPONENT SLC_GKZ 'TT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.OPROV_SK
COMPONENT SLC_STAAT 'Ostprovinz', COMPONENT SLC_GKZ 'TT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.WPROV_SK
COMPONENT SLC_STAAT 'Westprovinz', COMPONENT SLC_GKZ 'TT';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.SPROV_SK
COMPONENT SLC_STAAT 'Suedprovinz', COMPONENT SLC_GKZ 'TT';
```

Der höchste Kennsatz für einen Benutzer ist:

```
CREATE SECURITY LABEL ANTI_TERROR_POLICY.GHD
COMPONENT SLC_STAAT 'Zentralstaat', COMPONENT SLC_GKZ 'TT',
COMPONENT SLC_LVL 'streng-geheim';
```

Dieser wird dem Datenbank Administrator vorübergehend zugeordnet, um die notwendigen Änderungen an Tabelle und Datendurchführen zu können. Außerdem wird er außerhalb der Regeln für die Policy gestellt:

```
GRANT SECURITY LABEL ANTI_TERROR_POLICY.GHD
TO USER db2inst9 FOR ALL ACCESS;
```

```
GRANT EXEMPTION ON RULE ALL
FOR ANTI_TERROR_POLICY TO USER db2inst9;
```

Nun kann der DBA die nötigen Änderungen durchführen. Der Zeilenschutz wird aktiviert durch:

```
ALTER TABLE "DB2INST9"."ATDATEICB"
ADD COLUMN POL_TAG DB2SECURITYLABEL
ADD SECURITY POLICY ANTI_TERROR_POLICY;
```

Der Schutz der Spalten wird aktiviert durch:

```
ALTER TABLE "DB2INST9"."ATDATEICB"
ALTER COLUMN GEFEINSTUFUNG SECURED WITH VS
ALTER COLUMN IDKRIMREG SECURED WITH VS
ALTER COLUMN IDTERRORNETZ SECURED WITH VS
ALTER COLUMN BEZTERRORNETZ SECURED WITH VS
ALTER COLUMN IDGEHDienst SECURED WITH GH
ALTER COLUMN VMANNKZ SECURED WITH SGH;
```

Die neue Label-Spalte muss noch mit den gewünschten Werten versorgt werden:

```
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'NPROV_POL')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'TV';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'OPROV_POL')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'TV';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'WPROV_POL')
where PROVINZ='West' and GEFEINSTUFUNG = 'TV';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'SPROV_POL')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'TV';
```

```

UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'NPROV_KD')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'AT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'OPROV_KD')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'AT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'WPROV_KD')
where PROVINZ='West' and GEFEINSTUFUNG = 'AT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'SPROV_KD')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'AT';

UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'NPROV_SK')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'TT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'OPROV_SK')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'TT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'WPROV_SK')
where PROVINZ='West' and GEFEINSTUFUNG = 'TT';
UPDATE ATDATEICB
set POL_TAG= SECLABEL_BY_NAME ('ANTI_TERROR_POLICY', 'SPROV_SK')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'TT';

```

Ob die Kennsätze richtig gesetzt wurden, kann überprüft werden mit:

```

select name, provinz, gefeinstufung,
       seclabel_to_char('ANTI_TERROR_POLICY', POL_TAG)
from atdateicb;

```

Wenn alles richtig ist, hat der DB-Administrator seine Pflicht getan und kann auf seine Label-Berechtigung verzichten. Der Sicherheitsadministrator muss noch die Benutzer-Berechtigungen vergeben, bevor diese die Tabelle bearbeiten können:

```

CREATE SECURITY LABEL ANTI_TERROR_POLICY.NPOL
COMPONENT SLC_STAAT 'Nordprovinz', COMPONENT SLC_GKZ 'TV',
COMPONENT SLC_LVL 'frei';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.OKD
COMPONENT SLC_STAAT 'Ostprovinz', COMPONENT SLC_GKZ 'AT',
COMPONENT SLC_LVL 'frei';
CREATE SECURITY LABEL ANTI_TERROR_POLICY.WSK
COMPONENT SLC_STAAT 'Westprovinz', COMPONENT SLC_GKZ 'TT',
COMPONENT SLC_LVL 'vs';

CREATE SECURITY LABEL ANTI_TERROR_POLICY.GHM
COMPONENT SLC_STAAT 'Zentralstaat', COMPONENT SLC_GKZ 'TT',
COMPONENT SLC_LVL 'geheim';

GRANT SECURITY LABEL ANTI_TERROR_POLICY.NPOL
TO USER harry FOR READ ACCESS;
GRANT SECURITY LABEL ANTI_TERROR_POLICY.OKD
TO USER derrick FOR READ ACCESS;
GRANT SECURITY LABEL ANTI_TERROR_POLICY.WSK
TO USER soko FOR ALL ACCESS;
GRANT SECURITY LABEL ANTI_TERROR_POLICY.GHM
TO USER schlphut FOR ALL ACCESS;

GRANT SECURITY LABEL ANTI_TERROR_POLICY.GHD
TO USER bond FOR ALL ACCESS;

```

Nur die Benutzer aus den Sonderkommissionen und vom Geheimdienst erhalten auch eine Schreibberechtigung.

Mit einfachen Abfragen kann nun getestet werden, wie die Policy funktioniert. Bezogen auf den Zeilenschutz dürfen nur die Zeilen angezeigt werden, deren Kennsätze gleich oder hierarchisch unter dem Kennsatz des Benutzers liegen. Bezogen auf den Spaltenschutz erhalten die Benutzer eine Fehlermeldung (SQLCODE - 20264 bzw. SQLSTATE 42512), wenn sie eine Spalte sehen wollen, für die sie keine Berechtigung besitzen.

1) Benutzer Harry - Polizist Nordprovinz:

```

select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST
from DB2INST9.atdateicb;

```

NAME	NAMENSZUSATZ	GESCHLECHT	GEBDAT	PROVINZ	ZUST_DIENSTS_ST
HAAS	-	F	1963-08-24	Nord	1.Kommissariat
THOMPSON	-	M	1978-02-02	Nord	1.Kommissariat
KWAN	-	F	1971-05-11	Nord	1.Kommissariat
STERN	-	M	1975-07-07	Nord	1.Kommissariat
PULASKI	-	F	2003-05-26	Nord	1.Kommissariat
HENDERSON	-	F	1971-05-15	Nord	1.Kommissariat
SPENSER	-	M	1980-12-18	Nord	1.Kommissariat
OCONNELL	-	M	1972-10-18	Nord	1.Kommissariat

NICHOLLS	-	F	1976-01-19	Nord	1. Kommissariat
ADAMSON	-	M	1977-05-17	Nord	1. Kommissariat
PIANKA	-	F	1980-04-12	Nord	1. Kommissariat
SCOUTTEN	-	F	1979-02-21	Nord	1. Kommissariat
BROWN	-	M	1971-05-29	Nord	1. Kommissariat
LUTZ	-	F	1978-03-19	Nord	1. Kommissariat
JEFFERSON	-	M	1980-05-30	Nord	1. Kommissariat
MARINO	-	M	2002-03-31	Nord	1. Kommissariat
JOHNSON	-	F	1976-10-05	Nord	1. Kommissariat
PEREZ	-	F	2003-05-26	Nord	1. Kommissariat
PARKER	-	M	1985-07-09	Nord	1. Kommissariat
SMITH	-	M	1976-10-27	Nord	1. Kommissariat
SETRIGHT	-	F	1961-04-21	Nord	1. Kommissariat
LEE	-	M	1971-07-18	Nord	1. Kommissariat
GOUNOT	-	M	1956-05-17	Nord	1. Kommissariat
HEMMINGER	-	F	1973-08-14	Nord	1. Kommissariat
ORLANDO	-	M	1972-10-18	Nord	1. Kommissariat
NATZ	-	F	1976-01-19	Nord	1. Kommissariat
YAMAMOTO	Kamikaze	M	1981-01-05	Nord	1. Kommissariat
JOHN	-	F	1978-03-19	Nord	1. Kommissariat
MONTEVERDE	-	M	1984-03-31	Nord	1. Kommissariat
SCHWARTZ	-	F	1966-03-28	Nord	1. Kommissariat
SPRINGER	-	F	1961-04-21	Nord	1. Kommissariat
WONG	-	F	1971-07-18	Nord	1. Kommissariat

32 record(s) selected.

## 2) Derrick - Kriminaldirektor Ost:

```
select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST
from DB2INST9.atdateicb;
```

NAME	NAMENSZUSATZ	GESCHLECHT	GEBDAT	PROVINZ	ZUST_DIENSTS_ST
MEHTA	-	M	1962-08-11	Ost	1. Kommissariat
ALONZO	-	M	1956-05-17	Ost	1. Kommissariat

2 record(s) selected

## 3) Soko - Sonderkommission Anti-Terror Westprovinz:

```
select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST,
GEFEINSTUFUNG from DB2INST9.atdateicb;
```

NAME	NAMENSZUSATZ	GESCHLECHT	GEBDAT	PROVINZ	ZUST_DIENSTS_ST	GEFEINSTUFUNG
GEYER		M	1955-09-15	West	1. Kommissariat	TV
WALKER	Die Flasche	M	1982-06-25	West	1. Kommissariat	TV
SMITH		M	1969-11-12	West	1. Kommissariat	TV
Bush	The Warrior	M	1944-02-29	West	3. SK Kölsch	TT
Ente	Duck	M	1872-01-01	West	WaschPo	AT

5 record(s) selected.

## 4) Schlphut - Special Agent Schlapphut vom Geheimdienst:

```
select NAME,NAMENSZUSATZ,GESCHLECHT, Provinz, ZUST_DIENSTS_ST, IDGEHDIENTST
from DB2INST9.atdateicb where IDGEHDIENTST is not null;
```

NAME	NAMENSZUSATZ	GESCHLECHT	PROVINZ	ZUST_DIENSTS_ST	IDGEHDIENTST
PEREZ	-	F	Nord	1. Kommissariat	45628
BinImLaden	Der Araber	M	Sued	2. Kommissariat	4712
Bush	The Warrior	M	West	3. SK Kölsch	876554
Fisher	Moltow-Cocktail	M	Ost	AutobahnPolizei	87621

4 record(s) selected.

## 5) Bond - Special Agent James Bond mit Sonderbefugnissen:

```
select NAME,NAMENSZUSATZ,GESCHLECHT, Provinz, ZUST_DIENSTS_ST, IDGEHDienst
from DB2INST9.atdateicb where VMANNKZ is not null;
```

NAME	NAMENSZUSATZ	GESCHLECHT	PROVINZ	ZUST_DIENSTS_ST	IDGEHDienst
BinImLaden	Der Araber	M	Sued	2.Kommissariat	4712
Fisher	Moltow-Cocktail	M	Ost	AutobahnPolizei	87621
Ente	Duck	M	West	WaschPo	-

3 record(s) selected.

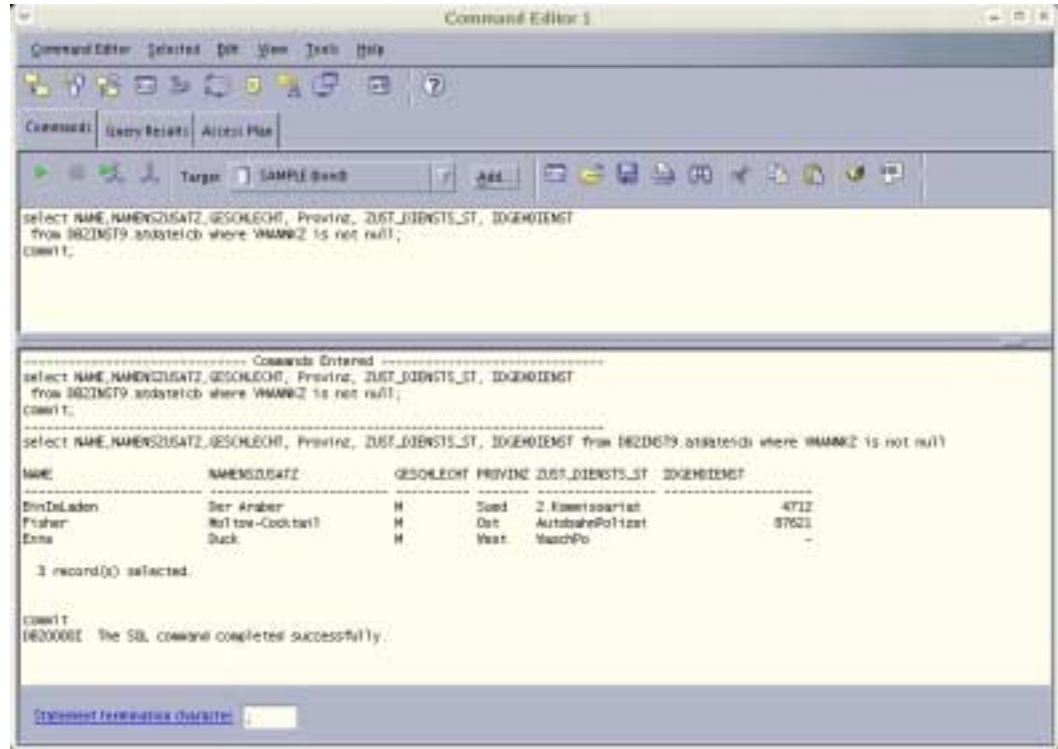


Abbildung 1: Abfrage mit dem DB2 Command Editor (Datei ce001.png)

## Die Oracle-Implementierung

Während IBM die LBAC-Funktionalität als Erweiterungen der bekannten SQL-Befehle implementiert hat, hat Oracle sich für eine Implementierung mit PL/SQL-Routinen entschieden, da OLS ja auf dem VPD-Toolkit aufsetzt. Grundsätzlich können die Definitionen im Policy Manager (OPM) ohne Kenntnis der Routinen festgelegt werden. Zum Vergleich mit der DB2-Lösung finden Sie hier die Prozedur-Aufrufe.

OLS erlaubt im Gegensatz zu DB2 nicht den Schutz von Spalten durch Labels. Es ist zwar möglich, den Zeilenschutz auf bestimmte Spalten zu beschränken, doch schon dazu benötigt man Kenntnisse der VPD-Technik und man verlässt die vorgefertigte Standard-Lösung. Die in unserem Beispiel vorgegebene Kombination aus Schutz auf Zeilenebene und Schutz für bestimmte Spalten ist in Oracle nur möglich, wenn man zu OLS für den Zeilenschutz noch eigene Routinen für den Spaltenschutz hinzufügt. Zuerst implementieren wir den Zeilenschutz über Kennsätze, dann zeigen wir beispielhaft, wie ein Spaltenschutz - allerdings ohne Labels - realisiert werden könnte.

Für die Implementierung der OLS-Funktionen nutzen wir nicht den OPM, damit auch vom Codieraufwand ein Vergleich mit DB2 möglich ist.

Die Tabelle hat in Oracle eine analoge Struktur zu DB2:

```
CREATE TABLE "BI"."ATDATEICB" (
  "PNR" INTEGER,
  "NAME" VARCHAR2(24) NOT NULL ,
  "VORNAME" VARCHAR2(24) NOT NULL ,
  "NAMENSZUSATZ" VARCHAR2(24) ,
  "HERKUNFTSLAND" VARCHAR2(4) NOT NULL ,
  "GESCHLECHT" VARCHAR2(1) NOT NULL ,
  "GEBDAT" DATE ,
  "ERSTEINDAT" DATE NOT NULL ,
  "LETZTEINDAT" DATE NOT NULL ,
  "LETZTAUSDAT" DATE ,
  "STAATSANGEH" VARCHAR2(4) NOT NULL ,
```

```

"ZUST_DIENSTS_ST" VARCHAR2(16) NOT NULL ,
"PROVINZ" VARCHAR2(4) NOT NULL ,
"GEFEINSTUFUNG" VARCHAR2(2) NOT NULL ,
"VMANNKZ" VARCHAR2(1) ,
"IDKRIMREG" INTEGER ,
"IDTERRORNETZ" INTEGER ,
"BEZTERRORNETZ" VARCHAR2(16) ,
"IDGEHDIENT" INTEGER )
;

```

COMMENT ON TABLE "BI"."ATDATEICB" IS 'Anti-Terror-Datei Costa Banana';

PNR ist Primärschlüssel:

```

ALTER TABLE "BI"."ATDATEICB"
  ADD CONSTRAINT "PK_ATDATEICB" PRIMARY KEY
    ("PNR");

```

Für die Vergabe der fortlaufenden Nummern zu PNR definieren wir eine Sequence:

```

CREATE SEQUENCE "BI"."TER_NUM" INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  CACHE 20 ORDER

```

Das Definieren der OLS Security Policy erfolgt mit Angabe der Label-Spalte:

```

begin
  SA_SYSDBA.CREATE_POLICY(
    policy_name => 'ANTI_TERROR_POLICY'
    ,column_name => 'POL_TAG'
    ,default_options => 'ALL_CONTROL,HIDE'
  );
end;
/

```

Da eine Policy immer aus maximal drei fest vorgegebenen Komponenten besteht, ist eine Definition von Komponenten nicht nötig. Es werden daher gleich die Werte zu den Komponenten definiert. Für die Gefährdungseinstufung nutzen wir die Komponente LEVEL:

```

begin
  SA_COMPONENTS.CREATE_LEVEL(
    policy_name => 'ANTI_TERROR_POLICY'
    ,level_num => 1000
    ,short_name => 'TV'
    ,long_name => 'Terror Verdaechtig'
  );
  SA_COMPONENTS.CREATE_LEVEL(
    policy_name => 'ANTI_TERROR_POLICY'
    ,level_num => 2000
    ,short_name => 'AT'
    ,long_name => 'Aktiver Terrorist'
  );
  SA_COMPONENTS.CREATE_LEVEL(
    policy_name => 'ANTI_TERROR_POLICY'
    ,level_num => 4000
    ,short_name => 'TT'
    ,long_name => 'Top Terrorist'
  );
end;
/

```

Für die Abbildung der Organisationsstruktur von Costa Banana bietet sich die Komponente GROUP an:

```

begin
  SA_COMPONENTS.CREATE_GROUP(
    policy_name => 'ANTI_TERROR_POLICY'
    ,group_num => 0
    ,short_name => 'ZS'
    ,long_name => 'Zentralstaat'
    ,parent_name => NULL
  );
  SA_COMPONENTS.CREATE_GROUP(
    policy_name => 'ANTI_TERROR_POLICY'
    ,group_num => 10
    ,short_name => 'NP'
    ,long_name => 'Nordprovinz'
    ,parent_name => 'ZS'
  );
  SA_COMPONENTS.CREATE_GROUP(
    policy_name => 'ANTI_TERROR_POLICY'
    ,group_num => 20
    ,short_name => 'OP'
    ,long_name => 'Ostprovinz'
    ,parent_name => 'ZS'
  );
end;

```

```

SA_COMPONENTS.CREATE_GROUP(
  policy_name => 'ANTI_TERROR_POLICY'
  ,group_num => 30
  ,short_name => 'WP'
  ,long_name => 'Westprovinz'
  ,parent_name => 'ZS'
);
SA_COMPONENTS.CREATE_GROUP(
  policy_name => 'ANTI_TERROR_POLICY'
  ,group_num => 40
  ,short_name => 'SP'
  ,long_name => 'Suedprovinz'
  ,parent_name => 'ZS'
);
end;
/

```

Sicherheitsrichtlinien können auf Schema- oder Tabellenebene zugeordnet werden. Die Sicherheitsrichtlinie in unserem Beispiel wird der Tabelle zugeordnet:

```

begin
SA_POLICY_ADMIN.APPLY_TABLE_POLICY(
  policy_name => 'ANTI_TERROR_POLICY'
  ,schema_name => 'BI'
  ,table_name => 'ATDATEICB'
);
end;
/

```

Wie in DB2 können die sinnvollen Werte-Kombinationen der Security Policy mit ihren Komponenten als Labels definiert werden. In Oracle werden ihnen numerische *Label-Tags* zugeordnet, die so gewählt werden können, dass sie für Gruppierung oder Sortierung in Auswertungen nutzbar sind.

```

begin
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 501
  ,label_value => 'TV::NP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 502
  ,label_value => 'AT::NP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 503
  ,label_value => 'TT::NP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 511
  ,label_value => 'TV::OP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 512
  ,label_value => 'AT::OP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 513
  ,label_value => 'TT::OP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 521
  ,label_value => 'TV::WP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 522
  ,label_value => 'AT::WP'
  ,data_label => TRUE
);
end;
/

```

```

SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 523
  ,label_value => 'TT::WP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 541
  ,label_value => 'TV::SP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 542
  ,label_value => 'AT::SP'
  ,data_label => TRUE
);
SA_LABEL_ADMIN.CREATE_LABEL(
  policy_name => 'ANTI_TERROR_POLICY'
  ,label_tag => 543
  ,label_value => 'TT::SP'
  ,data_label => TRUE
);
end;
/

```

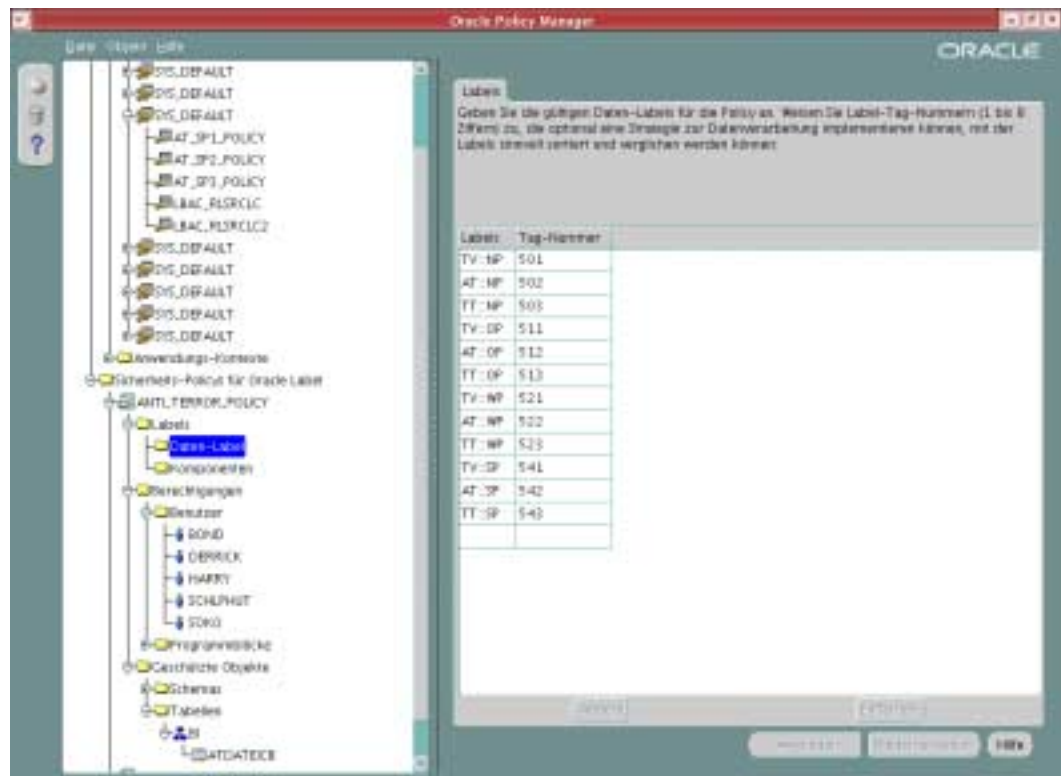


Abbildung 2: Label-Definitionen im Oracle Policy Manager (Datei opm005.png)

Damit die Benutzer arbeiten können, werden ihnen entsprechend den vorgesehenen Berechtigungen Labels zugeordnet:

```

begin
  SA_USER_ADMIN.SET_USER_LABELS(
    policy_name => 'ANTI_TERROR_POLICY'
    ,user_name => 'HARRY'
    ,max_read_label => 'TV::NP'
  );
  SA_USER_ADMIN.SET_USER_LABELS(
    policy_name => 'ANTI_TERROR_POLICY'
    ,user_name => 'DERRICK'
    ,max_read_label => 'AT::OP'
  );
end;

```

```

SA_USER_ADMIN.SET_USER_LABELS(
  policy_name => 'ANTI_TERROR_POLICY'
  ,user_name => 'SOKO'
  ,max_read_label => 'TT::WP'
);
SA_USER_ADMIN.SET_USER_LABELS(
  policy_name => 'ANTI_TERROR_POLICY'
  ,user_name => 'SCHLPHUT'
  ,max_read_label => 'TT::ZS'
);
SA_USER_ADMIN.SET_USER_LABELS(
  policy_name => 'ANTI_TERROR_POLICY'
  ,user_name => 'BOND'
  ,max_read_label => 'TT::ZS'
);
end;
/

```

Analog zu DB2 müssen nun die Werte in der Label-Spalte erzeugt werden:

```

UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TV::NP')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'TV';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TV::OP')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'TV';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TV::WP')
where PROVINZ='West' and GEFEINSTUFUNG = 'TV';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TV::SP')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'TV';

UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'AT::NP')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'AT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'AT::OP')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'AT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'AT::WP')
where PROVINZ='West' and GEFEINSTUFUNG = 'AT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'AT::SP')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'AT';

UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TT::NP')
where PROVINZ='Nord' and GEFEINSTUFUNG = 'TT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TT::OP')
where PROVINZ='Ost' and GEFEINSTUFUNG = 'TT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TT::WP')
where PROVINZ='West' and GEFEINSTUFUNG = 'TT';
UPDATE BI.ATDATEICB
set POL_TAG= CHAR_TO_LABEL ('ANTI_TERROR_POLICY', 'TT::SP')
where PROVINZ='Sued' and GEFEINSTUFUNG = 'TT';

```

Ob die Kennsätze richtig gesetzt wurden, kann überprüft werden durch:

```

select pnr, name, provinz, gefeinstufung, substr(label_to_char(POL_TAG),1,16)
from BI.ATDATEICB;

```

Damit wäre der Zeilenschutz implementiert. Es fehlt noch der Schutz der Spalten. Dieser ist mit OLS nicht zu erreichen, wohl aber mit einer individuellen Lösung. Zur Demonstration, wie es gehen könnte, wählen wir die einfachste Variante. Eine Individual-Lösung könnte allerdings beliebig komplex ausfallen - mit Logon-Triggern, Applikationskontexten oder unterschiedlichen Policy-Gruppen.

In unserer einfachen Version unterstellen wird eine Benutzer-Tabelle USER\_ACCESS mit mindestens den Spalten

- UID - Benutzername
- VSSTUFE - Verschlusssachenermächtigung

Wir vernachlässigen dabei, daß diese Tabelle sicher auch einem besonderen Zugriffsschutz unterliegen wird.

```

CREATE TABLE "BI"."USER_ACCESS" (
  "UID"          VARCHAR2(24) NOT NULL PRIMARY KEY,
  "VSSTUFE"     VARCHAR2(16) NOT NULL )
;

```

Für die drei unterschiedlichen Sicherheitsstufen der Spalten (vs, geheim, streng-geheim) definieren wir drei Filter-Funktionen in einem Paket:

```

create or replace package at_spalten
as
function at_sp1(p_schema in varchar2, p_table in varchar2)
return varchar2;
function at_sp2(p_schema in varchar2, p_table in varchar2)
return varchar2;
function at_sp3(p_schema in varchar2, p_table in varchar2)
return varchar2;
end;
/

create or replace package body at_spalten
as
function at_sp1 (
  p_schema in varchar2,
  p_table in varchar2
)
return varchar2
is
  retstr varchar2(200):='';
  zahl integer:= 0;
begin
  select count(*) into zahl
  from bi.user_access
  where "UID" = SYS_CONTEXT('USERENV','SESSION_USER')
  and VSSTUFE in ('streng-geheim', 'geheim', 'vs');
  retstr := '1 = '||zahl;
  return retstr;
end;
--
function at_sp2 (
  p_schema in varchar2,
  p_table in varchar2
)
return varchar2
is
  retstr varchar2(200):='';
  zahl integer:= 0;
begin
  select count(*) into zahl
  from bi.user_access
  where "UID" = SYS_CONTEXT('USERENV','SESSION_USER')
  and VSSTUFE in ('streng-geheim', 'geheim');
  retstr := '1 = '||zahl;
  return retstr;
end;
--
function at_sp3 (
  p_schema in varchar2,
  p_table in varchar2
)
return varchar2
is
  retstr varchar2(200):='';
  zahl integer:= 0;
begin
  select count(*) into zahl
  from bi.user_access
  where "UID" = SYS_CONTEXT('USERENV','SESSION_USER')
  and VSSTUFE = 'streng-geheim';
  retstr := '1 = '||zahl;
  return retstr;
end;
end;

```

Die Filter-Funktionen werden als Sicherheitsrichtlinie für die jeweiligen Spalten registriert:

```

begin
DBMS_RLS.add_policy (
  object_schema => 'BI',
  object_name => 'ATDATEICB',
  policy_name => 'AT_SP1_POLICY',
  function_schema => 'SYSMAN',
  policy_function => 'AT_SPALTEN.AT_SP1',
  sec_relevant_cols => 'GEFEINSTUFUNG, IDKRIMREG, IDTERRORNETZ, BEZTERRORNETZ'
);

```

```

DBMS_RLS.add_policy (
  object_schema => 'BI',
  object_name   => 'ATDATEICB',
  policy_name   => 'AT_SP2_POLICY',
  function_schema => 'SYSMAN',
  policy_function => 'AT_SPALTEN.AT_SP2',
  sec_relevant_cols => 'IDGEHDIENTST'
);
DBMS_RLS.add_policy (
  object_schema => 'BI',
  object_name   => 'ATDATEICB',
  policy_name   => 'AT_SP3_POLICY',
  function_schema => 'SYSMAN',
  policy_function => 'AT_SPALTEN.AT_SP3',
  sec_relevant_cols => 'VMANNKZ'
);
end;
/

```

Damit ist ein Schutz auf Spaltenebene eingerichtet, der der Funktionalität in DB2 entspricht. Im Gegensatz zu DB2, bei dem eine Fehlermeldung ausgegeben wird, gibt Oracle keine Zeilen aus (SQLCODE +1403), wenn der Benutzer auf eine Spalte zugreifen will, für die er nicht autorisiert ist.

#### 1) Benutzer Harry - Polizist Nordprovinz:

```

select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST
  from BI.atdateicb;

```

NAME	NAMENSZUSATZ	G	GEBDAT	PROV	ZUST_DIENSTS_ST
PEREZ		F	26-MAY-03	Nord	1.Kommissariat
KWAN		F	11-MAY-71	Nord	1.Kommissariat
HAAS		F	24-AUG-63	Nord	1.Kommissariat
THOMPSON		M	02-FEB-78	Nord	1.Kommissariat
STERN		M	07-JUL-75	Nord	1.Kommissariat
PULASKI		F	26-MAY-03	Nord	1.Kommissariat
HENDERSON		F	15-MAY-71	Nord	1.Kommissariat
SPENSER		M	18-DEC-80	Nord	1.Kommissariat
CONNELL		M	18-OCT-72	Nord	1.Kommissariat
NICHOLLS		F	19-JAN-76	Nord	1.Kommissariat
ADAMSON		M	17-MAY-77	Nord	1.Kommissariat
PIANKA		F	12-APR-80	Nord	1.Kommissariat
SCOUTTEN		F	21-FEB-79	Nord	1.Kommissariat
BROWN		M	29-MAY-71	Nord	1.Kommissariat
LUTZ		F	19-MAR-78	Nord	1.Kommissariat
JEFFERSON		M	30-MAY-80	Nord	1.Kommissariat
MARINO		M	31-MAR-02	Nord	1.Kommissariat
JOHNSON		F	05-OCT-76	Nord	1.Kommissariat
PARKER		M	09-JUL-85	Nord	1.Kommissariat
SMITH		M	27-OCT-76	Nord	1.Kommissariat
SETRIGHT		F	21-APR-61	Nord	1.Kommissariat
LEE		M	18-JUL-71	Nord	1.Kommissariat
GOUNOT		M	17-MAY-56	Nord	1.Kommissariat
HEMMINGER		F	14-AUG-73	Nord	1.Kommissariat
ORLANDO		M	18-OCT-72	Nord	1.Kommissariat
NATZ		F	19-JAN-76	Nord	1.Kommissariat
YAMAMOTO	Kamikaze	M	05-JAN-81	Nord	1.Kommissariat
JOHN		F	19-MAR-78	Nord	1.Kommissariat
MONTEVERDE		M	31-MAR-84	Nord	1.Kommissariat
SCHWARTZ		F	28-MAR-66	Nord	1.Kommissariat
SPRINGER		F	21-APR-61	Nord	1.Kommissariat
WONG		F	18-JUL-71	Nord	1.Kommissariat

32 rows selected.

#### 2) Derrick - Kriminaldirektor Ost:

```

select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST
  from BI.atdateicb;

```

NAME	NAMENSZUSATZ	G	GEBDAT	PROV	ZUST_DIENSTS_ST
MEHTA		M	11-AUG-62	Ost	1.Kommissariat
ALONZO		M	17-MAY-56	Ost	1.Kommissariat

2 rows selected.

## 3) Soko - Sonderkommission Anti-Terror Westprovinz:

```
select NAME,NAMENSZUSATZ,GESCHLECHT,GEBDAT, Provinz, ZUST_DIENSTS_ST,
GEFEINSTUFUNG from BI.atdateicb;
```

NAME	NAMENSZUSATZ	G	GEBDAT	PROV	ZUST_DIENSTS_ST	GE
GEYER		M	15-SEP-55	West	1.Kommissariat	TV
WALKER	Die Flasche	M	25-JUN-82	West	1.Kommissariat	TV
SMITH		M	12-NOV-69	West	1.Kommissariat	TV
Ente	Duck	M	01-JAN-72	West	WaschPo	AT
Bush	The Warrior	M	29-FEB-44	West	3.SK Koelsch	TT

5 rows selected.

## 4) Schlphut - Special Agent Schlapphut vom Geheimdienst:

```
select NAME,NAMENSZUSATZ,GESCHLECHT, Provinz, ZUST_DIENSTS_ST, IDGEHDIENST
from BI.atdateicb where IDGEHDIENST is not null;
```

NAME	NAMENSZUSATZ	G	PROV	ZUST_DIENSTS_ST	IDGEHDIENST
BinImLaden	Der Araber	M	Sued	2.Kommissariat	4712
PEREZ		F	Nord	1.Kommissariat	45628
Fisher	Moltow-Cocktail	M	Ost	AutobahnPolizei	87621
Bush	The Warrior	M	West	3.SK Koelsch	12345

4 rows selected.

## 5) Bond - Special Agent James Bond mit Sonderbefugnissen:

```
select NAME,NAMENSZUSATZ,GESCHLECHT, Provinz, ZUST_DIENSTS_ST, IDGEHDIENST
from BI.atdateicb where VMANNKZ is not null;
```

NAME	NAMENSZUSATZ	G	PROV	ZUST_DIENSTS_ST	IDGEHDIENST
BinImLaden	Der Araber	M	Sued	2.Kommissariat	4712
Ente	Duck	M	West	WaschPo	
Fisher	Moltow-Cocktail	M	Ost	AutobahnPolizei	87621

3 rows selected.

## Fazit

Das Beispiel zeigt, wie mit den kennsatz-gesteuerten Sicherheitsrichtlinien fein abgestimmte, von den Datenwerten abhängige Zugriffsrechte definiert werden können. Der Vorteil dieser Implementierungen ist, dass sie nicht oder nur als SYSDBA (Oracle) umgangen werden können.

Das Beispiel führt bei DB2 und Oracle zu gleichen Ergebnissen. Die Art, wie es in DB2 oder Oracle zu definieren ist, unterscheidet sich sehr voneinander.

Auf der Ebene der Befehlszeile sind die Definitionen in DB2 einfacher zu formulieren als in Oracle. Der Schutz von Spalten ist in LBAC integriert und kann somit sehr einfach realisiert werden. DB2 verfügt mit maximal 16 Komponenten einer Policy und dem Schutz von Zeilen und Spalten standardmäßig über deutlich mehr Möglichkeiten als Oracle. Zur Implementierung benötigt ein Sicherheitsadministrator Grundkenntnisse in SQL.

Oracle besitzt für die Definition von Sicherheitsrichtlinien ein grafisches Werkzeug, das DB2 nicht bieten kann. Mit diesem Werkzeug kann der Sicherheitsadministrator mit OLS arbeiten, ohne dass er ein Oracle-Spezialist mit profunden SQL-Kenntnissen sein muss. OLS hat mit drei vordefinierten Komponenten und fehlendem Schutz von Spalten als Standard Grenzen, die schnell erreicht sein können. Andererseits ist es in Oracle möglich, die Standard-Anwendung OLS zu verlassen und über den VPD-Toolkit eine individuelle, beliebig komplexe Sicherheitslösung zu implementieren.

Abschließend muss noch darauf hingewiesen werden, dass diese wichtigen Sicherheitsfunktionen bei beiden Herstellern den *Enterprise Editions* vorbehalten bleiben. Für mehr Sicherheit verlangen die Hersteller auch mehr Geld.