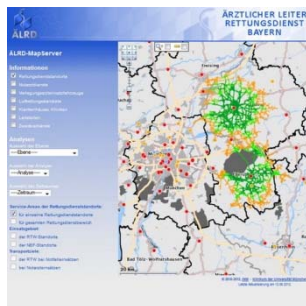




VIRTUAL PRIVATE DATABASE

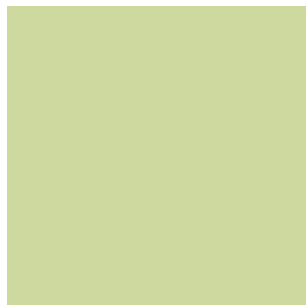
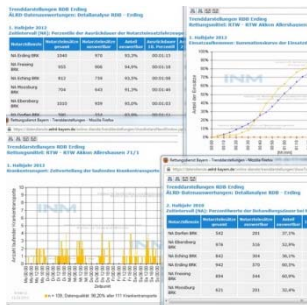
Markus Geis & Mathias Weber

03.06.2014



Trendanalysen RIB Erlang
 Erlang: 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030

Indikator	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
...



AGENDA

- **Institut für Notfallmedizin und Medizinmanagement - INM**
- **Was ist Virtual Private Database - VPD ?**
- **Aufbau einer VPD-Umgebung**
- **DEBUG / Performance / Probleme / Sicherheit**
- **Fazit**

INSTITUT FÜR NOTFALLMEDIZIN UND MEDIZIN-MANAGEMENT

■ Markus Geis



markus.geis@med.uni-muenchen.de

- IT – Informationstechnologie
- Datenbankadministration
- Datenbankentwicklung

■ Mathias Weber



mathias.weber@med.uni-muenchen.de

- SysPro – Systemanalyse und Prozessoptimierung
- Geoinformationssysteme
- MapServer-Applikationen
- Datenbankentwicklung

KLINIKUM DER UNIVERSITÄT (LMU)

- zwei Standorte: Innenstadt und Großhadern
- ca. 500.000 Patienten (ambulant, teilstationär und stationär)
- 45 Fachkliniken, Instituten und Abteilungen
- 45 interdisziplinären Zentren
- 10.000 Beschäftigte (1.800 Mediziner und 3.400 Pflegekräfte)
- WS 2012/2013: 5.637 Studenten

INSTITUT FÜR NOTFALLMEDIZIN UND MEDIZIN-MANAGEMENT

- gegründet 2002
- erstes notfallmedizinisches Institut an einer deutschsprachigen Universität
- Interdisziplinäre Forschung und Lehre in Notfallmedizin, Rettungswesen und Management in der Medizin
- www.inm-online.de
- Eingesetzte SW/HW
 - Oracle 3 Knoten RAC 11gR2
 - Partitioning
 - Spatial
 - NetApp-Metro-Cluster



AGENDA

- **Institut für Notfallmedizin und Medizinmanagement - INM**
- **Was ist VPD ?**
- **Aufbau einer VPD-Umgebung**
- **DEBUG / Performance / Probleme / Sicherheit**
- **Fazit**

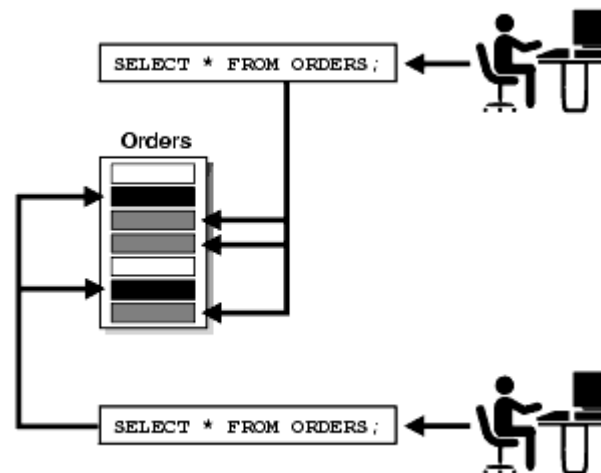
WAS IST VPD (IST-ZUSTAND) ?

- Wie ist der „normale“ Status in einer Oracle-DB:
- Tables, User,
- User greifen über Objektprivilegien
-> select, insert, update, delete
auf Tables zu
- Daten einer Table werden „komplett“ angezeigt
- Einschränkung der Zugriffe:
über die Applikation ->
eine Möglichkeit (Statementänderung / Userverwaltung
in der Applikation)

- *Provokante Frage/Feststellung: „ist dann alles OKAY ?“*
- Applikationsänderung / Sicherheitkonzept

WAS IST VPD (GRUNDGEDANKE) ?

- gibt es eine andere Möglichkeit ?
- Einschränkung der Usersicht **ohne** Applikations- und Statementänderung (sql-statements bleiben immer GLEICH)
- **Virtual Private Database**
- **Daten-Zugriffe bei Tabellen auf Zeilenebene beschränkt**



WAS IST VPD ?

- **ergänzt zur Laufzeit automatisch die Statements mit einer zusätzlichen -> WHERE-Bedingung**
- Anzahl der zu selektierten Zeilen wird dadurch (VPD-WHERE) eingeschränkt
- VPD hat eine TABLE zur BASIS
- VPD wird über eine „Policy“ an eine Table gehängt
- mehrere „Policys“ können pro Table genutzt werden
- Daten können gruppen-/userbezogen dargestellt werden (mandantenfähig)

- zusätzlich: VPD ermöglicht ebenfalls Spalten aus einer Table für gewisse User zu „maskieren“ bzw. „nicht anzuzeigen“

WAS IST VPD ?

- Mehrere Begriffe werden synonym verwendet:
 - VIRTUAL PRIVATE DATABASE (VPD)
 - ROW LEVEL SECURITY (RLS)
 - FINE GRAINED ACCESS CONTROL

- Gesamt-“VPD“ besteht aus zwei Teilen
 - Fine Grained Access Control
 - Context
(Parameter + Werte im Memory (SGA))

WAS IST VPD ?

- Feature der Oracle Enterprise Edition
- Anfang: Trusted Oracle 7 / Label Security
- relativ altes Feature -> ab Version 8i
- Einsatz im INM ab der Version 8.2.0.3

- LABEL-Security -> LIZENZ

AGENDA

- **Institut für Notfallmedizin und Medizinmanagement - INM**
- **Was ist VPD ?**
- **Aufbau einer VPD-Umgebung**
- **DEBUG / Performance / Probleme / Sicherheit**
- **Fazit**

AUFBAU EINER VPD-UMGEBUNG

- Ein Tabellenmodell, welches als Basis für den Aufbau dient (Planung)
- Objekt in welchem die „User/Gruppe/Rechte“ und „Wer darf was sehen“ (INM: Userverwaltungs-Table)
- Function, welche die Rechte für die User zusammenstellt und einen Rückgabe-String für die Nutzung von VPD aufbaut
- Policy, welche über die Function mit einem Rückgabewert versorgt wird, um die Einschränkungen durchzuführen

AUFBAU EINER VPD-UMGEBUNG

- Beispiel APEX/VPD

<http://www.oracle.com/webfolder/technetwork/de/community/apex/tips/virtual-private-database/index.html>

- VPD-Table: Userverwaltung zur Rechtesteuerung

```
select * from my_users;  
USERID      CLASS      DEPTS  
-----  
SYSTEM      ADMIN  
SCOTT        DEPTADM    40  
BLAKE        DEPTADM    20  
MILLER       DEPTADM    30  
KING         ADMIN
```

```
select ename, sal, deptno from emp;  
ENAME          SAL      DEPTNO  
-----  
MARTIN          1250      30  
BLAKE           2850      30  
CLARK           2450      10  
SCOTT           3000      20  
KING            5000      10  
TURNER          1500      30  
ADAMS           1100      20  
JAMES           950       30  
FORD            3000      20
```

AUFBAU EINER VPD-UMGEBUNG

- Function in PL/SQL programmiert / ohne setzen eines CONTEXT / mit DB-User / 2-Tier Application

```

CREATE OR REPLACE FUNCTION "F_ZUGRIFF_RDB_XXX"
  (owner in varchar2, objname varchar2)
  return varchar2
as
  v_id_rdb number :=0;
  v_id_reg_bez number :=0;
  out_string varchar2(400) default '1=2';
begin
  select id_rdb, nvl(id_reg_bez, 0) into v_id_rdb, v_id_reg_bez
  from userverwaltung
  where upper (user_name) = upper(sys_context('userenv','session_user'));
if user = 'RDB_OWNER' then
  return '';
else
  if v_id_reg_bez = 0 then -- Benutzer ist VB
    if v_id_rdb = 0 then
      out_string := '1=1';
    else
      out_string:='id_rdb='||v_id_rdb;
    end if;
  else
    out_string:='id_rdb in'||
      '(select id_rdb from id_reg_bez where id_reg_bez='||v_id_reg_bez||')';
  end if;
end if;
return out_string;
exception
  when others then -- wenn account nicht in der Table, dann wird dieser Fehler erzeugt
    return '1=2';
end;

```

AUFBAU EINER VPD-UMGEBUNG

- POLICY anlegen

```
DBMS_RLS.ADD_POLICY (  
  object_schema      IN VARCHAR2 NULL,  
  object_name        IN VARCHAR2,  
  policy_name        IN VARCHAR2,  
  function_schema    IN VARCHAR2 NULL,  
  policy_function     IN VARCHAR2,  
  statement_types    IN VARCHAR2 NULL,  
  update_check       IN BOOLEAN  FALSE,  
  enable             IN BOOLEAN  TRUE,  
  static_policy       IN BOOLEAN  FALSE      TRUE: same predicate string for anyone  
                                              accessing the object  
  policy_type        IN BINARY_INTEGER NULL  NULL-> static_policy wird genutzt  
                                              (STATIC, DYNAMIC, CONTEXT_SENSITIVE)  
  long_predicate      IN BOOLEAN  FALSE      4000 bytes / TRUE: 32K  
  sec_relevant_cols  IN VARCHAR2,  
  sec_relevant_cols_opt IN BINARY_INTEGER NULL);
```

- Funktion für die Erstellung des VPD Strings wird normalerweise bei jedem PARSE/EXECUTE ausgeführt (Automatic Reparsing)
- Caching des VPD Strings

AUFBAU EINER VPD-UMGEBUNG

■ POLICY anlegen (Beispiel)

```
BEGIN
  SYS.DBMS_RLS.ADD_POLICY      (
    ,object_name              => 'TAB1'
    ,policy_name              => 'ZUGRIFFSKONTROLLE_TAB1'
    ,policy_function          => 'F_ZUGRIFF_RDB_XXX'
    ,statement_types         => 'SELECT,INSERT,UPDATE,DELETE'
    ,policy_type              => dbms_rls.dynamic
    ,long_predicate          => FALSE
    ,update_check            => TRUE
    ,static_policy           => FALSE
    ,enable                  => TRUE );
END;
```

```
BEGIN
  SYS.DBMS_RLS.DROP_POLICY (
    object_schema    => Null
    ,object_name     => 'TAB1'
    ,policy_name     => 'ZUGRIFFSKONTROLLE_TAB1');
END;
```

AUFBAU EINER VPD-UMGEBUNG

- Einbindung bei WEB-Applikationen (IST-Zustand)
 - 3-Tier Application
 - technischer DB-User
 - connection-pool
 - kein DB-User-Session
 - stateless
 - Applikation händelt Connects/Rechte
- Was passiert bei Veränderungen der Rechtestruktur (Statements / Einschränkung der Sichten über SQL)?
- Auditung
- usergesteuerte Trigger
- direkter DB-Zugriff (ODBC, OLE, DOT-Net, JDBC, sql*plus)

- Bietet Oracle auch dafür eine Lösung ?

AUFBAU EINER VPD-UMGEBUNG

- CONTEXT:
- Eine Anzahl von Variablen, welche Informationen zwischen Applikationen und Usern abgleichen und setzten
`dbms_session.set_context`
- Informationen werden im CACHE gehalten
- Speziell für VPD entwickelt (Performance)
- Typen: USERENV: User-Session-Infos: Name, IP, ..
 GLOBAL: connection-pool / shareable /
 Zugriff von verschiedenen DB-Sessions
 LOCAL: enthält Applikation-Infos
- `CREATE OR REPLACE CONTEXT RDB_CONTEXT USING RDB_LOGIN_PACKAGE;`
- Namespace "RDB_CONTEXT"
- nur das Package "RDB_LOGIN_PACKAGE" darf die Werte ändern / trusted

AUFBAU EINER VPD-UMGEBUNG

■ POLICY Aufbau mit CONTEXT

■ CREATE OR REPLACE package **rdb_login_package**

is

```
procedure set_context(p_userid in varchar2);
```

```
end rdb_login_package;
```

■ CREATE OR REPLACE package body **rdb_login_package**

is

```
procedure set_context(p_userid in varchar2) is
```

```
    v_rdb wasserw_verwaltung.userverwaltung.ID_RDB%TYPE := 1;
```

```
begin
```

```
    for cl in (select id_rdb from
```

```
        wasserw_verwaltung.userverwaltung where upper(user_name) = upper(p_userid)) loop
```

```
        v_rdb := cl.id_rdb;
```

```
    end loop;
```

```
        dbms_session.set_context('rdb_context','ID_RDB', v_rdb);
```

```
    end set_context;
```

```
end rdb_login_package;
```

■ `rdb_login_package.set_context('xyz-user');` (z.B.: after-logon-trigger/Applikation)

AUFBAU EINER VPD-UMGEBUNG

- POLICY Aufbau mit CONTEXT

- ```
CREATE OR REPLACE function WASSERW.f_rdb_policy(p_schema varchar2,
 p_object varchar2) return varchar2 is
 v_sql varchar2(32767);
begin
 if sys_context('rdb_context','ID_RDB') = 0 then
 null;
 elsif sys_context('rdb_context','ID_RDB') != 0 then
 v_sql := ' ID_RDB = (' || sys_context('rdb_context','ID_RDB') || ')';
 end if;
 return v_sql;
end;
/
```

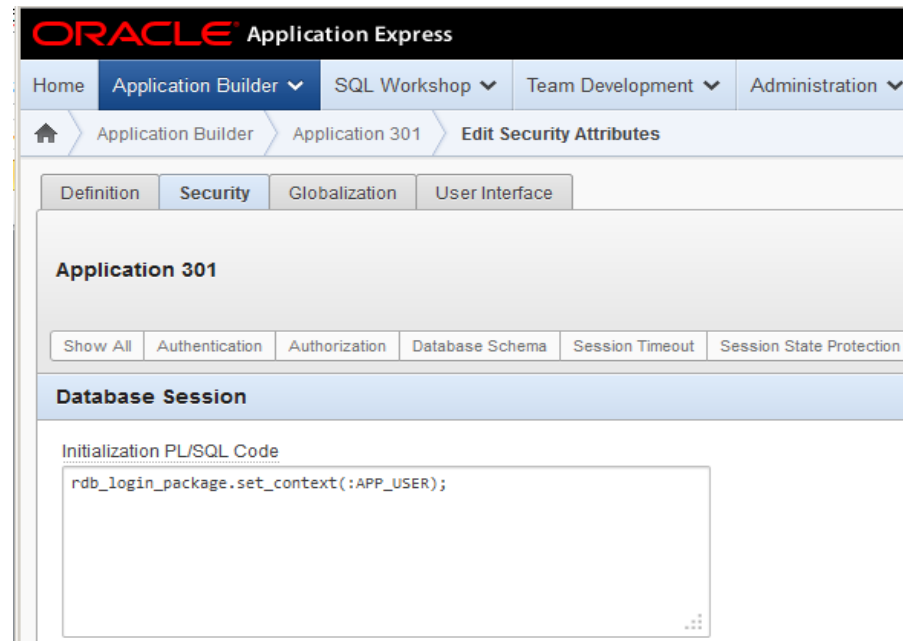
- ```
BEGIN SYS.DBMS_RLS.ADD_POLICY (-----)
```

AUFBAU EINER VPD-UMGEBUNG

- POLICY Aufbau mit CONTEXT
- mehrere Möglichkeiten:
- Connection-Pool / PROXY_USER
- LDAP
- Oracle Internet Directory
- APEX-User

AUFBAU EINER VPD-UMGEBUNG

- POLICY Aufbau mit CONTEXT / 3-Tier Application / APEX
- Shared Components -> Security Attributes -> Security / Database Session (mit APEX-User)



AUFBAU EINER VPD-UMGEBUNG

- Zusätzlich VPD auf Spalten anwenden: „column-level VPD“
- `sec_relevant_cols => 'SAL'`
Welche Spalten sollen “maskiert” werden
- `sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS`
Anzeigen aller Zeilen – Werte der eingestellten COLUMNS werden nicht angezeigt
(default ist “NULL”, Zeile, welche den Wert (Spalte) enthält, werden nicht angezeigt)

- ```
SQL> select name,sal from emp;
NAME SAL

SMITH
MILLER 2000
WARD
JONES
```



## AGENDA

- **Institut für Notfallmedizin und Medizinmanagement - INM**
- **Was ist VPD ?**
- **Aufbau einer VPD-Umgebung**
- **DEBUG / Performance / Probleme / Sicherheit**
- **Fazit**

## DEBUG

- Frage: Was enthält der RETURN-Wert, welcher an die POLICY zurückgegeben wird ?
- “Event 10730” (trace row level security policy predicates) to dump the rewritten SQL to a trace file in the *user\_dump\_dest* )
- Sicherheit: Leserechte auf TRACE-File bzw. Verzeichnisse

```
alter session set sql_trace=true
exec scott.my_login_package.SET_CONTEXT ('SCOTT');
alter session set events '10730 trace name context forever';
select * from scott.emp;
alter session set events '10730 trace name context off';
alter session set sql_trace=false;
```

## DEBUG

```
PARSING IN CURSOR #3 len=39 dep=1 uid=84 oct=47 lid=84 tim=1397549833573845 hv=3519430458 ad='11b19fc98' qlid='5n2dgnv8wccgtu'
```

```
begin :con := MY_POLICY(:sn, :on); end;
```

```
END OF STMT
```

```
begin :con := MY_POLICY(:sn, :on); end;
```

```
END OF STMT
```

```
PARSE #4:c=0,e=39,p=0,cr=0,cu=0,mis=0,r=0,dep=1,og=1,plh=0,tim=1397569281171110
```

```
EXEC #4:c=0,e=202,p=0,cr=0,cu=0,mis=0,r=1,dep=1,og=1,plh=0,tim=1397569281171390
```

```
CLOSE #4:c=0,e=14,dep=1,type=3,tim=1397569281171446
```

```

Logon user : SYSTEM
```

```
Table/View : SCOTT.EMP
```

```
Policy name : MY_EMP_POLICY
```

```
Policy function: SCOTT.MY_POLICY
```

```
RLS view :
```

```
SELECT "EMPNO" , "ENAME" , "JOB" , "MGR" , "HIREDATE" , "SAL" , "COMM" , "DEPTNO"
```

```
FROM "SCOTT"."EMP" "EMP" WHERE (DEPTNO in (40))
```

## DEBUG

- DBA-View
- `select * from dba_policies  
where object_owner like 'WASSER%';`

| OBJECT_OWNER | OBJECT_NAME | POLICY_GROUP | POLICY_NAME | PF_OWNER   | PACKAGE | FUNCTION     | SEL | INS | UPD | DEL | IDX | CHK_OPTION | ENABLE | STATIC_POLICY | POLICY_TYPE | LONG_PREDICATE |
|--------------|-------------|--------------|-------------|------------|---------|--------------|-----|-----|-----|-----|-----|------------|--------|---------------|-------------|----------------|
| WASSERW_STR  | RDB         | SYS_DEFAULT  | RDB_POLICY  | WASSERW_AN |         | F_RDB_POLICY | YES | YES | YES | YES | NO  | NO         | YES    | NO            | DYNAMIC     | NO             |

# SICHERHEIT

- Sicherheit muß im Zusammenhang betrachtet werden
- VPD ist nicht die alleinige Lösung

```
0x0E1FE60: 1F 2C 00 08 03 C2 4C 16 04 57 41 52 44 08 53 41 ÄL.WARD.SA
0x0E1FE70: 4C 45 53 4D 41 4E 03 C2 4D 63 07 77 B5 02 16 01 LESMAN.ÄMc.wµ...
0x0E1FE80: 01 01 03 C2 0D 33 02 C2 06 02 C1 1F 2C 00 08 03 ...Ä.3.Ä...Ä....
0x0E1FE90: C2 4C 43 05 4A 4F 4E 45 53 07 4D 41 4E 41 47 45 ÄLC.JONES.MANAGE
0x0E1FEA0: 52 03 C2 4F 28 07 77 B5 04 02 01 01 01 03 C2 1E R.ÄO(.wµ.....Ä.
0x0E1FEB0: 4C FF 02 C1 15 2C 00 08 03 C2 4D 37 06 4D 41 52 Ly.Ä....ÄM7.MAR
0x0E1FEC0: 54 49 4E 08 53 41 4C 45 53 4D 41 4E 03 C2 4D 63 TIN.SALESMAN.ÄMc
0x0E1FED0: 07 77 B5 09 1C 01 01 01 03 C2 0D 33 02 C2 0F 02 .wµ.....Ä.3.Ä..
0x0E1FEE0: C1 1F 2C 00 08 03 C2 4D 63 05 42 4C 41 4B 45 07 Ä....ÄMc.BLAKE.
0x0E1FEF0: 4D 41 4E 41 47 45 52 03 C2 4F 28 07 77 B5 05 01 MANAGER.ÄO(.wµ..
0x0E1FF00: 01 01 01 03 C2 1D 33 FF 02 C1 1F 2C 00 08 03 C2 Ä.3y.Ä....Ä
0x0E1FF10: 4E 53 05 43 4C 41 52 4B 07 4D 41 4E 41 47 45 52 NS.CLARK.MANAGER
0x0E1FF20: 03 C2 4F 28 07 77 B5 06 09 01 01 01 03 C2 19 33 .ÄO(.wµ.....Ä.3
0x0E1FF30: FF 02 C1 0B 2C 00 08 03 C2 4E 59 05 53 43 4F 54 y.Ä....ÄNY.SCOT
0x0E1FF40: 54 07 41 4E 41 4C 59 53 54 03 C2 4C 43 07 77 BB T.ANALYST.ÄLC.w»
0x0E1FF50: 04 13 01 01 01 02 C2 1F FF 02 C1 15 2C 00 08 03 Ä.y.Ä....
0x0E1FF60: C2 4F 28 04 4B 49 4E 47 09 50 52 45 53 49 44 45 ÄO(.KING.PRESIDE
0x0E1FF70: 4E 54 FF 07 77 B5 0B 11 01 01 01 02 C2 33 FF 02 NTy.wµ.....Ä3y.
0x0E1FF80: C1 0B 2C 00 08 03 C2 4F 2D 06 54 55 52 4E 45 52 Ä....ÄO-.TURNER
0x0E1FF90: 08 53 41 4C 45 53 4D 41 4E 03 C2 4D 63 07 77 B5 .SALESMAN.ÄMc.wµ
0x0E1FFA0: 09 08 01 01 01 02 C2 10 01 80 02 C1 1F 2C 00 08 Ä...I.Ä....
0x0E1FFB0: 03 C2 4F 4D 05 41 44 41 4D 53 05 43 4C 45 52 4B .ÄOM.ADAMS.CLERK
0x0E1FFC0: 03 C2 4E 59 07 77 BB 05 17 01 01 01 02 C2 0C FF .ÄNY.w»....Ä.y
0x0E1FFD0: 02 C1 15 2C 00 08 02 C2 50 05 4A 41 4D 45 53 05 .Ä....ÄP.JAMES.
0x0E1FFE0: 43 4C 45 52 4B 03 C2 4D 63 07 77 B5 0C 03 01 01 CLERK.ÄMc.wµ....
0x0E1FFF0: 01 03 C2 0A 33 FF 02 C1 1F 3C 01 00 03 06 E0 DF ..Ä.3y.Ä.<....àB
0x0E20000: 00 A2 00 00 10 07 00 01 00 00 00 00 00 01 05 .ç.....
0x0E20010: 10 A1 00 00 00 00 00 00 00 00 00 00 00 00 00 .i.....
```

# SICHERHEIT

```
[oracle@apex-live ~]$ sqlplus system
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Thu Apr 10 15:29:55 2014
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Enter password:
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> select * from scott.emp;
```

```
no rows selected
```

```
SQL> █
```

```
[oracle@apex-live ~]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Thu Apr 10 15:32:03 2014
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
SQL> set linesize 200
SQL> select * from scott.emp;
```

| EMPNO | ENAME | JOB      | MGR  | HIREDATE  | SAL  | COMM | DEPTNO |
|-------|-------|----------|------|-----------|------|------|--------|
| 7369  | SMITH | CLERK    | 7902 | 17-DEC-80 | 800  |      | 20     |
| 7499  | ALLEN | SALESMAN | 7698 | 20-FEB-81 | 1600 | 300  | 30     |
| 7521  | WARD  | SALESMAN | 7698 | 22-FEB-81 | 1250 | 500  | 30     |

## EXPORT

- logische Datenbanksicherung (export full database)  
VPD verhindert export der Daten
- `ORA-39181: Only partial table data may be exported due to fine grain access control on "SCOTT"."CONTROL_ITEMS"`
- Zwei Möglichkeiten:
  - **EXEMPT ACCESS POLICY (sehr weitgehendes Recht)**  
`grant EXEMPT ACCESS POLICY to user_export;`
  - Policy erweitern

## PERFORMANCE

- VPD hat sehr wenig (negativen) Einfluß auf die Performance
- VPD nutzt die Möglichkeiten des Optimizer (transparent zum CBO) – Policy nutzt dadurch Statistiken, Histogramme
- z.T. wird die Performance der Applikation besser, wenn entsprechenden Einschränkungen über VPD durchgeführt werden



## PERFORMANCE

### CONTEXT-SCOTT:

```
select *
from scott.emp
```

| call    | count | cpu  | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse   | 1     | 0.00 | 0.00    | 0    | 0     | 0       | 0    |
| Execute | 1     | 0.00 | 0.00    | 0    | 0     | 0       | 0    |
| Fetch   | 2     | 0.00 | 0.00    | 0    | 4     | 0       | 1    |
| total   | 4     | 0.00 | 0.00    | 0    | 4     | 0       | 1    |

Misses in library cache during parse: 1

Optimizer mode: ALL\_ROWS

Parsing user id: 5 (SYSTEM)

Rows Row Source Operation

```

1 TABLE ACCESS BY INDEX ROWID EMP (cr=4 pr=0 pw=0 time=0 us cost=681 size=3426681 card=92613)
1 INDEX RANGE SCAN EMP_IDX (cr=3 pr=0 pw=0 time=0 us cost=68 size=0 card=34816)(object id 114217)
```

## PERFORMANCE

### CONTEXT-SYSTEM:

```
select *
from scott.emp
```

| call    | count | cpu  | elapsed | disk | query | current | rows   |
|---------|-------|------|---------|------|-------|---------|--------|
| Parse   | 1     | 0.00 | 0.00    | 0    | 0     | 0       | 0      |
| Execute | 1     | 0.00 | 0.00    | 0    | 0     | 0       | 0      |
| Fetch   | 37139 | 0.90 | 0.79    | 0    | 40192 | 0       | 557057 |
| total   | 37141 | 0.90 | 0.79    | 0    | 40192 | 0       | 557057 |

Misses in library cache during parse: 1

Optimizer mode: ALL\_ROWS

Parsing user id: 5 (SYSTEM)

Rows Row Source Operation

-----  
557057 TABLE ACCESS FULL EMP (cr=40192 pr=0 pw=0 time=286592 us cost=893 size=20560160 card=555680)

## AGENDA

- **Institut für Notfallmedizin und Medizinmanagement - INM**
- **Was ist VPD ?**
- **Aufbau einer VPD-Umgebung**
- **DEBUG / Performance / Probleme / Sicherheit**
- **Fazit**

## FAZIT

- VPD ist einfach einzusetzen (trivial)
- durchgängig bei allen Oracle-Produkten (exp/imp, DataGuard..)
- VPD bietet viele Möglichkeiten die Datenbestände vor neugierigen Blicken zu schützen
- VPD ist für die Applikation völlig transparent
- VPD ist sehr flexibel
- Nutzung aller DB-Features (Trigger, AUDIT,...)
- Berechtigungskonzepte können ohne Applikationsänderungen verändert werden (SQL-Statements)
- DB sichert die Daten ab / "sql-injection"
- auch nachträglicher Aufbau möglich (prüfen)

## FAZIT

- SAP ? / andere ?
- Nachteil -> Enterprise Edition
- Billig-Version über VIEWS möglich / Performance

## BUCHEMPFEHLUNG

- Oracle® Database Security Guide 11g Release 2
- Effective Oracle Database 10g Security by Design by Knox, David



# VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

## ANSPRECHPARTNER:

Markus Geis / Mathias Weber

Klinikum der Universität München

INM - Institut für Notfallmedizin und Medizinmanagement

Telefon: 089 / 4400-57101

E-Mail: [markus.geis@med.uni-muenchen.de](mailto:markus.geis@med.uni-muenchen.de)

[mathias.weber@med.uni-muenchen.de](mailto:mathias.weber@med.uni-muenchen.de)

Internet: [www.inm-online.de](http://www.inm-online.de)



