TSBE Bern
Höhere Fachschule

ORACLE®
DATABASE

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Security 1/2

TSBE Bern
Höhere Fachschule

- Security wird im Datenbank Umfeld immer wichtiger, ist jedoch oft nicht vollumfänglich implementiert
- Im Bankenumfeld gibt es zT gesetzliche Bestimmungen betr. Security auf den Datenbanken
- Security auf Datenbanken ist weitläufig. Viele Sachen können zur Security gezählt werden. z.B.
  - Verschiedene Aspekte betr. Passworte z.B.
    - Komplexität
    - Lebensdauer
    - Anzahl Fehlversuche bis der User gesperrt wird etc.
  - Default Port des Listeners
  - Limitieren der Zugriffe auf die Daten
  - Auditing
- Dieses Kapitel soll nur einen kleinen Einblick geben was möglich ist. Es gibt noch einige weitere Möglichkeiten

Security 2/2

TSBE Bern
Höhere Fachschule

- Der Umgang mit der Security beim Kunden oder bei den Software Herstellern zeigt, dass Security immer wichtiger wird
- z.B. das Verkaufen von Bankdaten an Regierungen zeigt, dass die Security noch nicht überall genügend implementiert ist
- Firmen haben immer restriktivere Security Richtlinien, nicht nur im Datenbankumfeld
- SW Hersteller wie z.B. ORACLE arbeiten an immer besseren Möglichkeiten im Bereich Security wie z.B. die Einführung von Unified Auditing

- → Security hat einen immer höheren Stellenwert in der IT

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Passworte 1/3

TSBE Bern
Höhere Fachschule

- Passworte können bei ORACLE in Profilen gesteuert werden
- Bei ORACLE Datenbanken wird jeder User einem Profil zugeordnet
- Per Default ist das das Profile DEFAULT, welches immer vorhanden ist
- Data Dictionary View

```
DBA_PROFILES
```

- Mit ORACLE 11g wurden einige Neuigkeiten betr. Passworte eingeführt
  - Per Default läuft ein Passwort nach 180 Tagen ab
  - Es wurden Case – Sensitive Passworte eingeführt
    - Case Sensitivität kann aus Kompatibilitätsgründen ausgeschaltet werden
- Mit ORACLE 12c wurde der Umgang mit Passworten weiter verschärft:
  - Case Sensitivität kann zwar immer noch ausgeschaltet werden, dies ist aber seit 12c deprecated, wird also in einem Folgerelease nicht mehr möglich sein

Passworte 2/3

TSBE Bern
Höhere Fachschule

- Profile haben 7 unterschiedliche Möglichkeiten mit Passworten umzugehen:

```
SQL> select resource_name, limit
  2  from   dba_profiles
  3  where  profile='DEFAULT'
  4    and  resource_type = 'PASSWORD'
  5  order by 1;

RESOURCE_NAME                    LIMIT
-------------------------------- ----------
FAILED_LOGIN_ATTEMPTS            10
PASSWORD_GRACE_TIME              7
PASSWORD_LIFE_TIME               180
PASSWORD_LOCK_TIME               1
PASSWORD_REUSE_MAX               UNLIMITED
PASSWORD_REUSE_TIME              UNLIMITED
PASSWORD_VERIFY_FUNCTION         NULL

7 rows selected.
```

Passworte 3/3

TSBE Bern
Höhere Fachschule

- FAILED_LOGIN_ATTEMTS
  - Steuert nach wie vielen Fehlversuchen das Account gelockt wird
- PASSWORD_GRACE_TIME
  - Spezifiziert in Tagen wie lange der User vor Ablaufen des Passworts gewarnt wird
  - Logins mit dem alten Passwort sind in dieser Zeit möglich
- PASSWORD_LIFE_TIME
  - Steuert in Tagen wie lange ein Passwort gültig ist
  - Nach Ablauf der Life Time muss der Benutzer beim nächsten Login das Passwort ändern
- PASSWORD_LOCK_TIME
  - Steuert wie lange in Tagen ein User gesperrt bleibt nachdem das Account wegen Überschreiten der FAILED_LOGIN_ATTEMTS gesperrt wurde
- PASSWORD_REUSE_MAX
  - Steuert wie oft ein Passwort gewechselt werden muss, bis ein bereits verwendetes Passwort wieder verwendet werden darf
- PASSWORD_REUSE_TIME
  - Steuert wie lange in Tagen ein Passwort nicht wieder verwendet werden darf
- PASSWORD_VERIFY_FUNCTION
  - Es kann eine PL/SQL Funktion angegeben werden, welche das Passwort prüft
  - In dieser Funktion können Regeln hinterlegt werden, wie ein Passwort auszusehen hat
  - z.B.: PW Länge, erlaubte Sonderzeichen, Zahlen, Gross- und Kleinschreibung etc.
  - Es gibt eine Beispielfunktion, die nach den Richtlinien der Firma angepasst werden kann
    - ${ORACLE_HOME}/rdbms/admin/utlpwdmg.sql

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Default Listener Port

TSBE Bern
Höhere Fachschule

- Der Listener von ORACLE läuft per Default auf Port 1521
- Dieser Port ist weitreichend bekannt und sollte in Security relevanten Umgebungen nicht verwendet werden
- Verwenden eines anderen Ports als 1521 ist eine einfache Möglichkeit die Security der Datenbank zu erhöhen

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Auditing 1/3

TSBE Bern
Höhere Fachschule

- ORACLE bietet eine Vielzahl von Möglichkeiten an die Datenbank zu überwachen
- Bei ORACLE 12c wurde ein komplett neues-, sicheres Auditing System eingeführt (Unified Auditing)
  - Bis ORACLE 12c war es möglich die gesammelten Auditing Daten manuell zu verändern, falls man über die nötigen Berechtigungen verfügte
    - Ab 12c mit Unified Auditing ist dies nicht mehr möglich
  - Unified Auditing überwacht sobald es eingeschalten ist auch Backup- oder Datenexporte und Importe
  - Unified Auditing kann bei 12c aus Kompatibilitätsgründen parallel zum herkömmlichen Auditing verwendet werden
- Die gesammelten Auditing Daten können wahlweise in der Datenbank oder im Filesystem abgelegt werden
- Es gibt unterschiedlich granulare Einstellungen des Auditings
  - DB, EXTENDED sammelt auch die Werte der Bindvariablen

Auditing 2/3

TSBE Bern
Höhere Fachschule

- ORACLE bietet beim klassischen Auditing die Möglichkeit folgende Typen von Operationen zu auditieren:
  - Statement Audit Optionen wie
    - alter session
    - alter any table
    - etc.
  - Data Dictionary View

DBA_STMT_AUDIT_OBTS

  - Privilegien wie
    - grant any table
    - alter profile
    - etc.
  - Data Dictionary View

DBA_PRIV_AUDIT_OBTS

  - Object Auditing
    - Verschiedene Arten von Zugriffen auf Objekte wie Tabellen oder Prozeduren etc.
  - Data Dictionary View

DBA_OBJ_AUDIT_OBTS

Auditing 3/3

TSBE Bern
Höhere Fachschule

- Mit Unified Auditing (seit 12c) lassen sich noch weit mehr Operationen wie z.B. Backup, Importe und Exporte etc. auditieren
- Weiterhin bietet ORACLE die Möglichkeit von «Fine Grade Auditing» (FGA)
- Mit FGA lassen sich Sachen auditieren wie:
  - Zugriff auf die Konten der Bank XY wenn der Kontostand > 1’000’000 ist
- Schreibende- und lesende Zugriffe können auditiert werden
- Zugriffe auf Tabellen und Views können mit FGA auditiert werden
- Die Definition der FGA Policies wird mit dem Package dbms_fga gemacht

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Zugriff auf Schemen 1/2

TSBE Bern
Höhere Fachschule

- Eine Applikation bzw. deren User sollten NIE als Schemaowner auf die Datenbank zugreifen
- Applikationen oder Benutzer sollten immer mit dedizierten, eigenen Benutzern auf die Datenbank zugreifen
  - Diese Benutzer erhalten dann die nötigen Berechtigungen, damit sie ihre Arbeit machen können
- Dieses Konzept setzt sich immer mehr durch, es gibt jedoch immer noch Applikationen, die sich direkt als Schemaowner auf die Datenbank verbinden
- Es gibt keinen Grund, warum man sich im normalen Betrieb als Schemaowner verbinden muss.
  - Alles ist mit Berechtigungen und ggf. mit Synonymen lösbar

Zugriff auf Schemen 2/2

TSBE Bern
Höhere Fachschule

- Nachteile, wenn sich die Applikation als Schemaowner auf die DB verbindet:
  - Das Schemaowner Passwort ist «bekannter»
    - Mehr Leute kennen es
  - Das Schema kann jederzeit, auch aus der Applikation, verändert werden
    - z.B. Tabellen oder deren Daten können verändert- oder gelöscht werden
  - Oft wollen spezielle Benutzer (Entwickler oder Superuser etc.) sich mit anderen Tools wie z.B. SQL*Plus oder ähnlich auf die Datenbank verbinden um Auswertungen zu machen. Machen sie das mit dem Schemaowner, haben sie volle Berechtigung auf das Schema

  - → Der Betrieb kann so nicht garantiert werden!

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Grants 1/11

TSBE Bern
Höhere Fachschule

- Grants sind Berechtigungen
- Auf einer Datenbank darf nicht jeder Benutzer hohe Berechtigungen haben
  - User xy, darf nicht Einstellungen des Systems ändern- oder gar das System neu starten dürfen
- Auf der Datenbank gilt:
  - Der Benutzer hat genau so viele Berechtigungen wie er braucht und nicht mehr
- Berechtigungen können an einzelne Benutzer- oder an Rollen vergeben werden
- Der Einfachheit halber ist es sinnvoller Berechtigungen an Rollen zu vergeben und diese Rollen anschliessend den Benutzern zu vergeben
- Genereller Syntax:

```
grant [Privileg] to [Benutzer | Rolle]
revoke [Privileg] from [Benutzer | Rolle]
```

Grants 2/11

TSBE Bern
Höhere Fachschule

- Data Dictionary Views:
  - Berechtigungen auf Objekte

```
DBA_TAB_PRIVS
```

  - Berechtigungen auf Rollen

```
DBA_ROLE_PRIVS
```

  - Systemprivilegien

```
DBA_SYS_PRIVS
```

Grants 3/11

TSBE Bern
Höhere Fachschule

- Welche Objekt Privilegien gibt es?

```
SQL> select distinct privilege from dba_tab_privs order by 1;

PRIVILEGE
----------------------------------------
ALTER
DEBUG
DELETE
DEQUEUE
EXECUTE
FLASHBACK
INDEX
INSERT
ON COMMIT REFRESH
QUERY REWRITE
READ
REFERENCES
SELECT
UPDATE
USE
WRITE

16 rows selected.
```

Grants 4/11

TSBE Bern
Höhere Fachschule

- Welche System Privilegien gibt es?

```
SQL> select distinct privilege from dba_sys_privs order by 1;

PRIVILEGE
----------------------------------------
ADMINISTER ANY SQL TUNING SET
ADMINISTER DATABASE TRIGGER
ADMINISTER RESOURCE MANAGER
ADMINISTER SQL MANAGEMENT OBJECT
ADMINISTER SQL TUNING SET
ADVISOR
ALTER ANY ASSEMBLY
…
…
…
UNLIMITED TABLESPACE
UPDATE ANY CUBE
UPDATE ANY CUBE BUILD PROCESS
UPDATE ANY CUBE DIMENSION
UPDATE ANY TABLE

202 rows selected.
```

Grants 5/11

TSBE Bern
Höhere Fachschule

- → Berechtigungen können sehr fein vergeben werden
- Oft wollen Benutzer hohe Privilegien wie z.B. die DBA Rolle, «weil es damit immer funktioniert hat»
  - Die DBA Rolle beinhaltet viele System- und Rollenberechtigungen

```
SQL> select count(*)
  2  from dba_sys_privs
  3  where grantee in (select granted_role
  4                    from dba_role_privs
  5                    where grantee = 'DBA'
  6                    union
  7                    select 'DBA' from dual);

  COUNT(*)
----------
       315
```

  - Somit ist die Chance sehr gross, dass das was erreicht werden soll auch funktioniert
  - Es geht aber auch mit weniger Privilegien!

Grants 6/11

TSBE Bern
Höhere Fachschule

- Berechtigungen, die vermieden werden sollten
  - Es gibt einige Berechtigungen, die einem Benutzer NICHT vergeben werden sollten:
    - *ANY* Privilegien
    - DBA Rolle
- *ANY*
  - Privilegien wie «select any table» oder «delete any table» sollten nicht an Benutzer vergeben werden
  - Mit ANY Privilegien hat der Benutzer Rechte auf alle Tabellen in der ganzen Datenbank
  - Dies kann mit Einzelprivilegien gelöst werden
- DBA
  - Die DBA Rolle beinhaltet sehr viele Privilegien wie z.B. alle *ANY* Privilegien
  - Die DBA Rolle sollte NIE an normale Benutzer vergeben werden
  - Trifft man auf Systeme, bei welchen normale Benutzer die DBA Rolle bereits gegrantet haben, bringt man sie fast nicht mehr weg
    - «Es geht nur wenn ich die DBA Rolle habe…»

Grants 7/11

TSBE Bern
Höhere Fachschule

- Wird ein User auf einer ORACLE Datenbank neu erstellt, hat er erstmal gar keine Berechtigungen
  - Er hat nicht einmal die Berechtigung sich mit der Datenbank zu verbinden
- Wichtige Rollen, die neuen DB Usern gegranted werden sollten:
  - Applikationsbenutzer:
    - connect Rolle
      - Damit kann sich der Benutzer auf die Datenbank verbinden
  - Schemaowner:
    - connect Rolle
    - resource Rolle
      - Damit hat der Schemaowner die Berechtigung verschiedene DB Objekte wie z.B. Tabellen etc. anzulegen

Grants 8/11

TSBE Bern
Höhere Fachschule

- Hat ein Benutzer kein Privileg um auf eine Tabelle zuzugreifen, wird bewusst eine unklare Fehlermeldung ausgegeben:

```
SQL> create user test identified by test;

User created.

SQL> grant connect to test;

Grant succeeded.

SQL> connect test/test
Connected.
SQL> select * from webshop.kunden;
select * from webshop.kunden
                      *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Grants 9/11

TSBE Bern
Höhere Fachschule

- Schlussfolgerung Berechtigungen:
  - Berechtigungen im Datenbankumfeld können sehr fein vergeben werden
  - Jeder Benutzer soll nur die Berechtigungen bekommen die er braucht
  - Berechtigungen auf Schemaobjekte werden am Besten mit Rollen vergeben

Grants 10/11

TSBE Bern
Höhere Fachschule

- Übung 1

1. Erstelle eine View, damit das lange Statement zur Abfrage unseres Webshop's nicht immer vollständig getippt werden muss
2. Liste via View alle Rechnungspositionen zu Rechnung Nr. 1 auf

```
select k.name,
       k.vorname,
       r.rg_nr,
       r.rg_datum,
       a.bezeichnung,
       rp.anzahl,
       a.preis,
       a.preis * rp.anzahl totalpreis
from   kunden k,
       rechnungen r,
       artikel a,
       rech_pos rp
where   k.kde_nr = r.kde_nr
  and   r.rg_nr = rp.rg_nr
  and   a.art_nr = rp.art_nr
order by a.bezeichnung;
```

Grants 11/11

TSBE Bern
Höhere Fachschule

- Lösung Übung Grants
  - Erstelle eine View, damit das lange Statement zur Abfrage unseres Webshop's nicht immer vollständig getippt werden muss
  - Liste via View alle Rechnungspositionen zu Rechnung Nr. 1 auf

```
grant create view to webshop;
connect webshop/manager

create or replace view v_rgpos as
select k.name,
       k.vorname,
       r.rg_nr,
       r.rg_datum,
       a.bezeichnung,
       rp.anzahl,
       a.preis,
       a.preis * rp.anzahl totalpreis
from   kunden k,
       rechnungen r,
       artikel a,
       rech_pos rp
where  k.kde_nr = r.kde_nr
  and  r.rg_nr = rp.rg_nr
  and  a.art_nr = rp.art_nr;

select * from v_rgpos
where  rg_nr = 1;
```

Agenda

TSBE Bern
Höhere Fachschule

- Security
  - Passworte
  - Default Listener Port
  - Auditing
- Zugriff auf Schemen
- Grants
- Rollen

Rollen 1/12

TSBE Bern
Höhere Fachschule

- Rollen und Gruppen werden an verschiedenen Stellen in der Informatik verwendet
  - Meist ist das Gleiche damit gemeint
- Rollen sind Gefässe für Privilegien
- Rollen können verschachtelt werden

```
SQL> create role role1;

Role created.

SQL> create role role2;

Role created.

SQL> grant role1 to role2;

Grant succeeded.
```

Rollen 2/12

TSBE Bern
Höhere Fachschule

- Rollen sind Datenbankobjekte, die ähnlich behandelt werden wie User
  - Siehe Fehlermeldung, wenn man eine Rolle mit dem gleichen Namen wie ein User anlegen möchte

```
SQL> create user test identified by test;

User created.

SQL> create role test;
create role test
            *
ERROR at line 1:
ORA-01921: role name 'TEST' conflicts with another user
or role name
```

Rollen 3/12

TSBE Bern
Höhere Fachschule

- In der Applikationsentwicklung ist es sinnvoll ein einfaches Rollenkonzept zu erstellen um auf die Daten zuzugreifen.
- Ein einfaches Rollenkonzept könnte z.B. aus zwei Rollen bestehen:
  - Rolle1 hat nur Lesezugriff auf die Daten
  - Rolle2 hat nur Schreibzugriff auf die Daten, hat aber auch noch die Berechtigung auf Rolle1

```
SQL> create role role1;

Role created.

SQL> create role role2;

Role created.

SQL> grant role1 to role2;

Grant succeeded.
```

Rollen 4/12

TSBE Bern
Höhere Fachschule

- Übung 1:

1. Erstelle für unseren Webshop ein Rollenkonzept, bestehend aus zwei Rollen:
   - Rolle 1: webshop_ro
     - webshop_ro hat Lesezugriff (ro = Read Only) auf alle Tabellen des Schemas webshop
   - Rolle 2: webshop_rw
     - webshop_rw hat Lese- und Schreibzugriff (rw = Read Write) auf alle Tabellen des Schemas webshop
2. Erstelle zwei Benutzer auf der Datenbank:
   - Es dürfen keine direkten Grants verwendet werden, nur Rollen dürfen berechtigt werden
   - Benutzer 1 (Name frei wählbar), hat nur Lesezugriff auf alle Tabellen des Schemas webshop
   - Benutzer 2 (Name frei wählbar), hat Lese- und Schreibzugriff auf alle Tabellen des Schemas webshop
3. Mit beiden Usern auf die Datenbank verbinden und testen ob es so funktioniert wie erwartet

- Erst selber versuchen bevor in der Lösung nachgeschaut wird!

Rollen 5/12

TSBE Bern
Höhere Fachschule

- Übung 2:

1. Die beiden Benutzer, welche in Übung 1 erstellt wurden, sollen auf die Daten zugreifen können ohne das der Schemanamen vorangestellt werden muss
2. Testen ob es funktioniert wie erwartet

Rollen 6/12

TSBE Bern
Höhere Fachschule

- Übung 3:

1. Teste den Zugriff auf die Daten via die View webshop.v_rgpos
2. Mach was nötig ist damit es ohne vorangestellten Schemanamen funktioniert

Rollen 7/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 1 Teil 1:

1. Erstelle für unseren Webshop ein Rollenkonzept, bestehend aus zwei Rollen:
   - Rolle 1: webshop_ro
     - webshop_ro hat Lesezugriff (ro = Read Only) auf alle Tabellen des Schemas webshop
   - Rolle 2: webshop_rw
     - webshop_rw hat Lese- und Schreibzugriff (rw = Read Write) auf alle Tabellen des Schemas webshop
2. Erstelle zwei Benutzer auf der Datenbank:
   - Es dürfen keine direkten Grants verwendet werden, nur Rollen dürfen berechtigt werden
   - Benutzer 1 (Name frei wählbar), hat nur Lesezugriff auf alle Tabellen des Schemas webshop
   - Benutzer 2 (Name frei wählbar), hat Lese- und Schreibzugriff auf alle Tabellen des Schemas webshop
3. Mit beiden Usern auf die Datenbank verbinden und testen ob es so funktioniert wie erwartet

```
create role webshop_ro;
grant select on webshop.artikel to webshop_ro;
grant select on webshop.kunden to webshop_ro;
grant select on webshop.rechnungen to webshop_ro;
grant select on webshop.rech_pos to webshop_ro;

create role webshop_rw;
grant insert, update, delete on webshop.artikel to webshop_rw;
grant insert, update, delete on webshop.kunden to webshop_rw;
grant insert, update, delete on webshop.rechnungen to webshop_rw;
grant insert, update, delete on webshop.rech_pos to webshop_rw;
grant webshop_ro to webshop_rw;
```

Rollen 8/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 1 Teil 2:

1. Erstelle für unseren Webshop ein Rollenkonzept, bestehend aus zwei Rollen:
   - Rolle 1: webshop_ro
     - webshop_ro hat Lesezugriff (ro = Read Only) auf alle Tabellen des Schemas webshop
   - Rolle 2: webshop_rw
     - webshop_rw hat Lese- und Schreibzugriff (rw = Read Write) auf alle Tabellen des Schemas webshop
2. Erstelle zwei Benutzer auf der Datenbank:
   - Es dürfen keine direkten Grants verwendet werden, nur Rollen dürfen berechtigt werden
   - Benutzer 1 (Name frei wählbar), hat nur Lesezugriff auf alle Tabellen des Schemas webshop
   - Benutzer 2 (Name frei wählbar), hat Lese- und Schreibzugriff auf alle Tabellen des Schemas webshop
3. Mit beiden Usern auf die Datenbank verbinden und testen ob es so funktioniert wie erwartet

```
create user web_ro_usr identified by test;
grant connect to web_ro_usr;
grant webshop_ro to web_ro_usr;

create user web_rw_usr identified by test;
grant connect to web_rw_usr;
grant webshop_rw to web_rw_usr;
```

Rollen 9/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 1 Teil 3:

1. Erstelle für unseren Webshop ein Rollenkonzept, bestehend aus zwei Rollen:
   - Rolle 1: webshop_ro
     - webshop_ro hat Lesezugriff (ro = Read Only) auf alle Tabellen des Schemas webshop
   - Rolle 2: webshop_rw
     - webshop_rw hat Lese- und Schreibzugriff (rw = Read Write) auf alle Tabellen des Schemas webshop
2. Erstelle zwei Benutzer auf der Datenbank:
   - Es dürfen keine direkten Grants verwendet werden, nur Rollen dürfen berechtigt werden
   - Benutzer 1 (Name frei wählbar), hat nur Lesezugriff auf alle Tabellen des Schemas webshop
   - Benutzer 2 (Name frei wählbar), hat Lese- und Schreibzugriff auf alle Tabellen des Schemas webshop
3. Mit beiden Usern auf die Datenbank verbinden und testen ob es so funktioniert wie erwartet

```
connect web_ro_usr/test
select count(*) from webshop.kunden;
insert into webshop.kunden (kde_nr, name, vorname, adresse,
                            plz, ort, email, handy)
            values (3, 'Joray', `Robert', 'Hofstrasse 19',
                    3013, 'Bern', `robert.joray@gmx.net',
                    '079/234 56 78');
update webshop.kunden set handy= '079/345 67 89'
where kde_nr=3;
delete webshop.kunden where kde_nr=3;

connect web_rw_usr/test
-- gleiche Statements wie oben ausführen
```

Rollen 10/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 2, Version 1:

1. Die beiden Benutzer, welche in Übung 1 erstellt wurden, sollen auf die Daten zugreifen können ohne das der Schemanamen vorangestellt werden muss
2. Testen ob es funktioniert wie erwartet

```
create synonym web_ro_usr.artikel for webshop.artikel;
create synonym web_ro_usr.kunden for webshop.kunden;
create synonym web_ro_usr.rechnungen for webshop.rechnungen;
create synonym web_ro_usr.rech_pos for webshop.rech_pos;

create synonym web_rw_usr.artikel for webshop.artikel;
create synonym web_rw_usr.kunden for webshop.kunden;
create synonym web_rw_usr.rechnungen for webshop.rechnungen;
create synonym web_rw_usr.rech_pos for webshop.rech_pos;

connect web_ro_usr/test
select count(*) from kunden;

connect web_rw_usr/test
select count(*) from kunden;
```

Rollen 11/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 2, Version 2:
- Oft hat es in einem Schema sehr viele Tabellen. Bei vielen Tabellen ist es einfacher sich die Statements mit dynamischem SQL zu generieren:

1. Die beiden Benutzer, welche in Übung 1 erstellt wurden, sollen auf die Daten zugreifen können ohne das der Schemanamen vorangestellt werden muss
2. Testen ob es funktioniert wie erwartet

```
SQL> select 'create synonym ' || u.username || '.' ||
            t.table_name || ' for ' || t.owner || '.' ||
            t.table_name || ';'
  2  from dba_tables t, dba_users u
  3  where t.owner = 'WEBSHOP'
  4  and u.username like 'WEB%USR'
  5  order by 1;

'CREATESYNONYM'||U.USERNAME||'.'||T.TABLE_NAME||'FOR'||T.OWNER||'.'||T.TABLE_NAM
--------------------------------------------------------------------
create synonym WEB_RO_USR.ARTIKEL for WEBSHOP.ARTIKEL;
create synonym WEB_RO_USR.KUNDEN for WEBSHOP.KUNDEN;
create synonym WEB_RO_USR.RECHNUNGEN for WEBSHOP.RECHNUNGEN;
create synonym WEB_RO_USR.RECH_POS for WEBSHOP.RECH_POS;
create synonym WEB_RW_USR.ARTIKEL for WEBSHOP.ARTIKEL;
create synonym WEB_RW_USR.KUNDEN for WEBSHOP.KUNDEN;
create synonym WEB_RW_USR.RECHNUNGEN for WEBSHOP.RECHNUNGEN;
create synonym WEB_RW_USR.RECH_POS for WEBSHOP.RECH_POS;

8 rows selected.

-- Testen wie in Übung 2, Version 1
```

Rollen 12/12

TSBE Bern
Höhere Fachschule

- Lösung Übung 3:

1. Teste den Zugriff auf die Daten via die View webshop.v_rgpos
2. Mach was nötig ist damit es ohne vorangestellten Schemanamen funktioniert

```
connect web_ro_usr/test

select * from webshop.v_rgpos;

connect / as sysdba

grant select on webshop.v_rgpos to webshop_ro;

create synonym web_ro_usr.v_rgpos for webshop.v_rgpos;
create synonym web_rw_usr.v_rgpos for webshop.v_rgpos;

connect web_ro_usr/test

select * from v_rgpos;
```

Fragen?

TSBE Bern
Höhere Fachschule