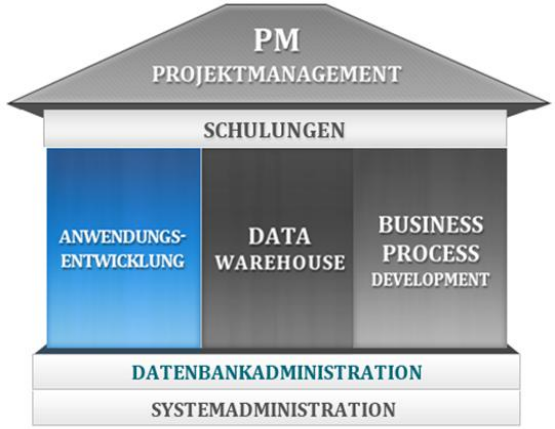
"Instructions for living a life.
Pay attention. Be astonished. Tell about it."
Mary Oliver, US Autorin

Referent Sven-Uwe Weller



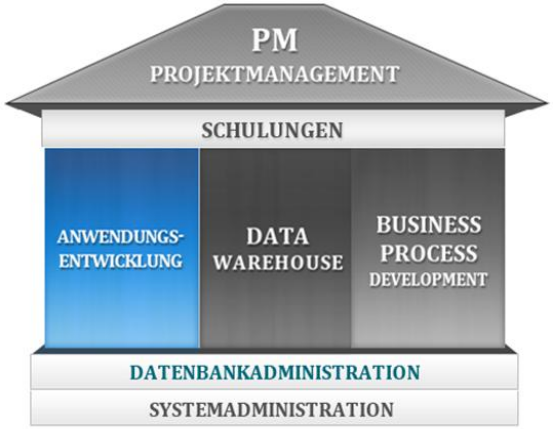
syntegris information solutions

Apex und Datenbanklinks

Einsatz in Produktivumgebungen

"I'm sorry, coffee's out of the question.
When I moved to California I promised my mother that I wouldn't start doing drugs."
Dr. Sheldon Cooper, Big Bang Theory

Firma



ADFFaces AJAX Android ant Apache APEX ApplicationServer Backup BalancedScorecard BPEL BusinessIntelligence BusinessObjects C++ Cognos CommonWarehouseModel ContinuousIntegration DataMart DataWarehouse Design Eclipse EJB ERWin Essbase ETL fop Glassfish GridControl Groovy GWT HealthCheck Hibernate HTML5 IceFaces Informatica ITIL Java JavaScript JAXB JBoss JDeveloper JEE JPA JSF LDAP Linux Migration MobileApps mvn Oracle Oracle11g OracleADF OracleBISuite OracleForms OracleWarehouseBuilder Palo Pentaho PerformanceTuning Perl PHP PLSQL PowerBuilder PrimeFaces RAC Recovery RichFaces Ruby SAS Scala Scrum Seam Shell SOA SOAP Solaris Spring SQL SQLDeveloper Subversion SUN Support Sybase Tapestry TOAD Tomcat UML Unix VMWare WebServices Workflow WSDL XML XPATH XQUERY XSD XSLT

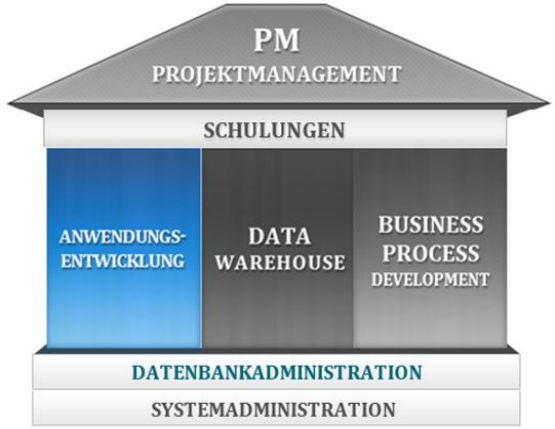
www.syntegris.de

syntegris information solutions GmbH, Hermannstr. 54-56, 63263 Neu-Isenburg

syntegris information solutions

Einleitung

Apex und Datenbanklinks



Basis

- Praktische Projekterfahrungen mit Apex (3.2 – 4.1)
- Anbindung mehrerer Oracle DBs mittels DB-Links

Gliederung

- Systemlandschaft / Einsatzszenarien
- Typische Probleme und Lösungen

Gründe für den Einsatz von APEX

- Moderne Oberfläche (browserbasiert)
- Existierende Kenntnisse (SQL, Datenmodell, etc.) können weiterverwendet werden,
- Neue Sprachen sind (im ersten Moment) nicht nötig.
- Schnell vorzeigbare Ergebnisse
- Geringer Deploymentaufwand (keine Client Installation, einfaches Provisioning)
- Administrationskenntnisse im Oracle Umfeld sind bereits vorhanden.
- Potential für Mobile Apps / läuft auch auf dem Tablett
- Keine Lizenzkosten

syntegris information solutions GmbH, Hermannstr. 54-56, 63263 Neu-Isenburg

Projekte auf der grünen Wiese sind selten!

Nachteile / Hürden

- Änderungen an existierenden System sind nicht gestattet
- DBAs / Sysadmins haben keine Erfahrung mit APEX
- Alles soll in einer einheitlichen integrierten Umgebung laufen

syntegris information solutions

Szenario 1 - Anforderung

Reporting Applikation

Für ein OLTP-Fremdsystem eines Softwareherstellers sollen zusätzliche Reports und Statistiken erstellt werden.

Eine Erweiterung innerhalb des Fremdsystems ist sehr teuer oder dauert sehr lange.

Die Applikationsbetreuer und Entwickler verfügen aber über solide Kenntnisse des zugrundeliegenden Datenmodels.

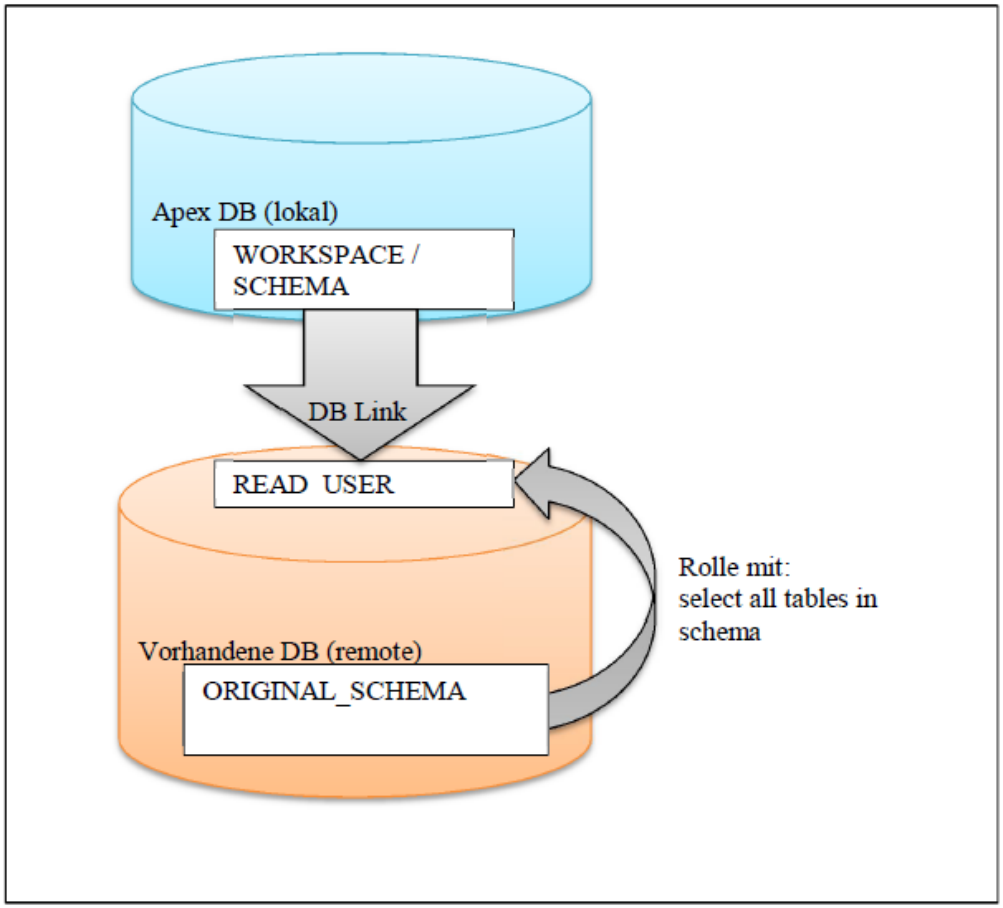
Erweiterungen im Schema sind aber aus lizenzrechtlichen Gründen nicht gestattet.

syntegris information solutions GmbH, Hermannstr. 54-56, 63263 Neu-Isenburg

syntegris information solutions

Szenario 1 - Lösung

Reporting Applikation



Die nötigen Reports werden in einer Apex Applikation zusammengefasst. Diese greift mittels read-only DB Links auf das existierende Schema zu. Lokal werden praktisch keine Daten gehalten.

syntegris information solutions

Szenario 2 - Anforderung

Zugriff auf mehrere DBs gleichzeitig

Datenvergleich oder -abgleich zwischen Test- und Produktivsystemen.

Mittels geeigneter Mechanismen kann der gleiche (Apex-)Report sowohl für den Zugriff auf eine Test Instanz, als auch für den Zugriff auf eine Produktiv Instanz verwendet werden.

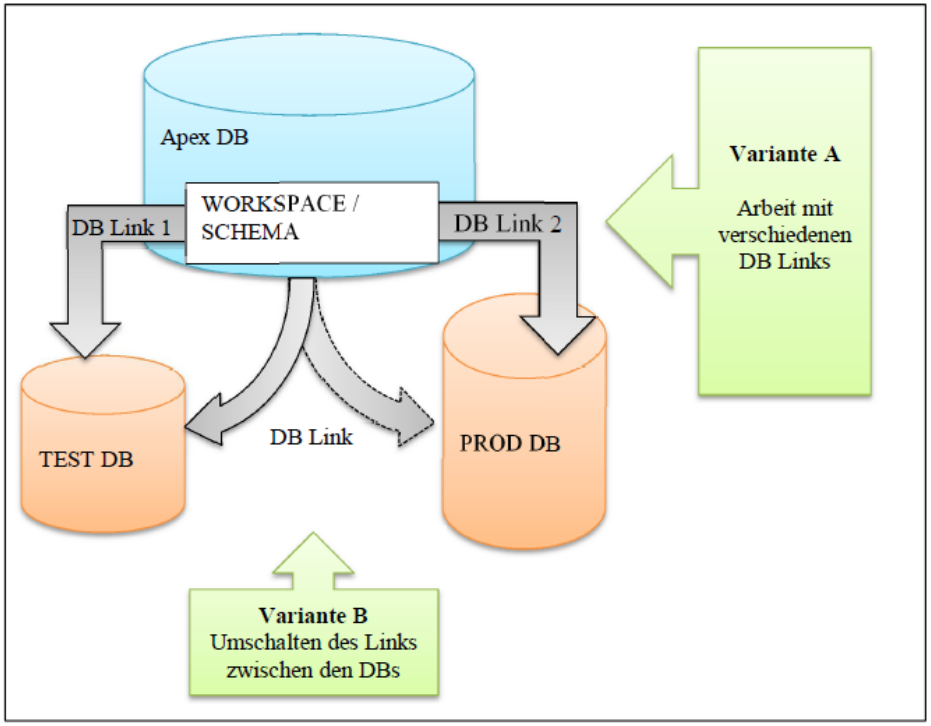
Dies ist insbesondere bei Erweiterungen am vorhandenen System oder bei Vorabuntersuchungen für neue Releases (Tests) sehr praktisch.

Es gibt auch die Möglichkeit Reports zu erstellen, die gleichzeitig die Daten beider Systeme anzeigen, z.B. bei regional verteilten Datenbanken.

syntegris information solutions

Szenario 2 – Lösung A

Zugriff auf mehrere DBs gleichzeitig



Beispiel für Reports mit Daten aus beiden remote-DB-Instanzen.

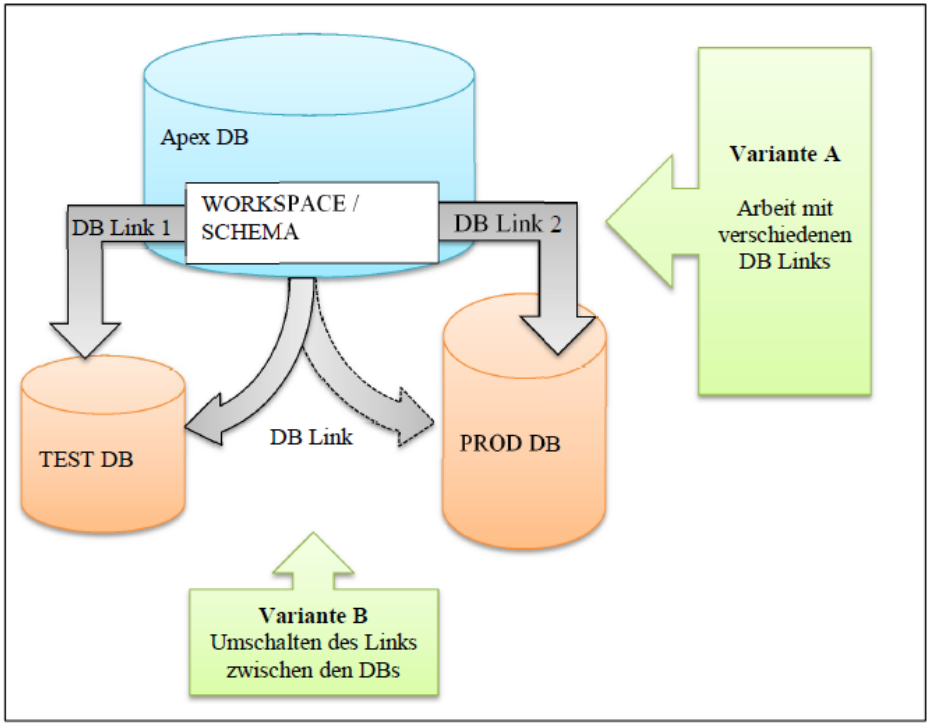
```
Select 'TEST' source, t.* from emp@testdb t
UNION ALL
select 'PROD' source, p.* from emp@proddb p
;
```

***Tipp**: Immer UNION ALL verwenden. Niemals nur UNION.*

syntegris information solutions

Szenario 2 – Lösung B

Zugriff auf mehrere DBs gleichzeitig



Beispiel für Report, der zwischen den Instanzen umschaltbar ist.

```
create synonym empremote on emp@testdb;
select e.* from empremote e;
```

Daten aus der Instanz 1# werden angezeigt.

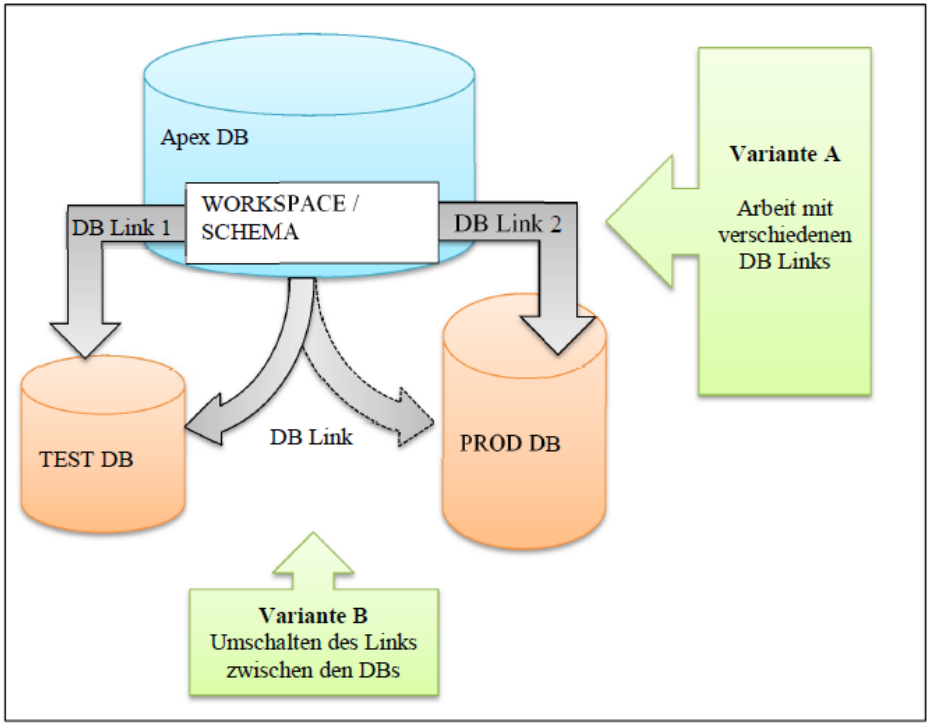
Danach Switch des DB Links oder alternativ umschalten eines Synonyms auf den zweiten Link

```
drop synonym empremote;
create synonym empremote on emp@proddb;
select e.* from empremote e;
```

Nun kommen die Daten aus der Instanz 2#!

Szenario 2 – Fazit

Zugriff auf mehrere DBs gleichzeitig



Vorteile

- Apex steht als GUI / RAD für unterschiedliche Ziel-Datenbanken zur Verfügung.
- Kein weiterer Installationsaufwand
- Hohe Flexibilität bei Erweiterungen
- Einfache Testmöglichkeiten
- Eine Oberfläche für verschiedene Zielsysteme

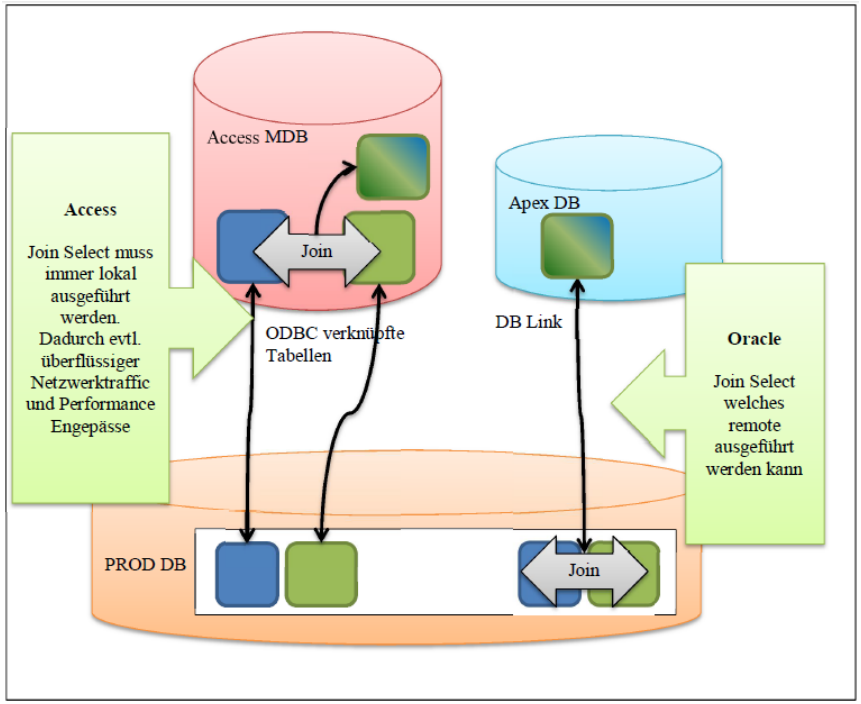
Nachteile

- Performance bei verteilten Abfragen muss sorgfältig beobachtet werden.
- Ggf. erhöhte Netzwerklast.
- DB Link zur Remote DB muss auch erst erzeugt werden.
- Security: Zugang sollte mit einem extra User erstellt werden.

syntegris information solutions

Szenario 3 - Anforderung

Ersatz von Access Zugriffen



Nutzung der Oracle Power anstelle von Fremdsystemen (MS Access).

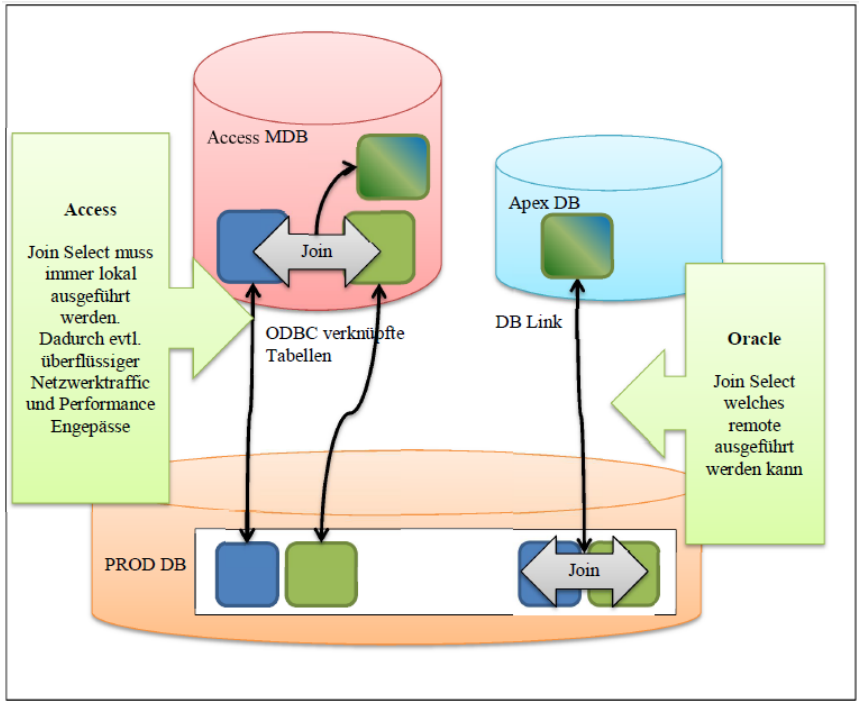
Oracle Datenbanken lassen sich per ODBC in viele Fremdsysteme einbinden (Access, Excel, etc.). Bei Access geht das über sogenannte Verknüpfungen. Wenn dann Joins über verknüpfte Tabellen gemacht werden, dann kommt es schnell zu Performanceproblemen.

Ursache: Die Tabellen werden komplett geladen, danach wird der Join innerhalb von Access gemacht. Dies ist dann inperformant, wenn die Tabellen viele Daten enthalten, die Ergebnismenge des Selects aber verhältnismäßig klein ist.

syntegris information solutions

Szenario 3 - Beispiel

Ersatz von Access Zugriffen



Beispiel:
Ermittle die Anzahl aller Angestellten, je Department

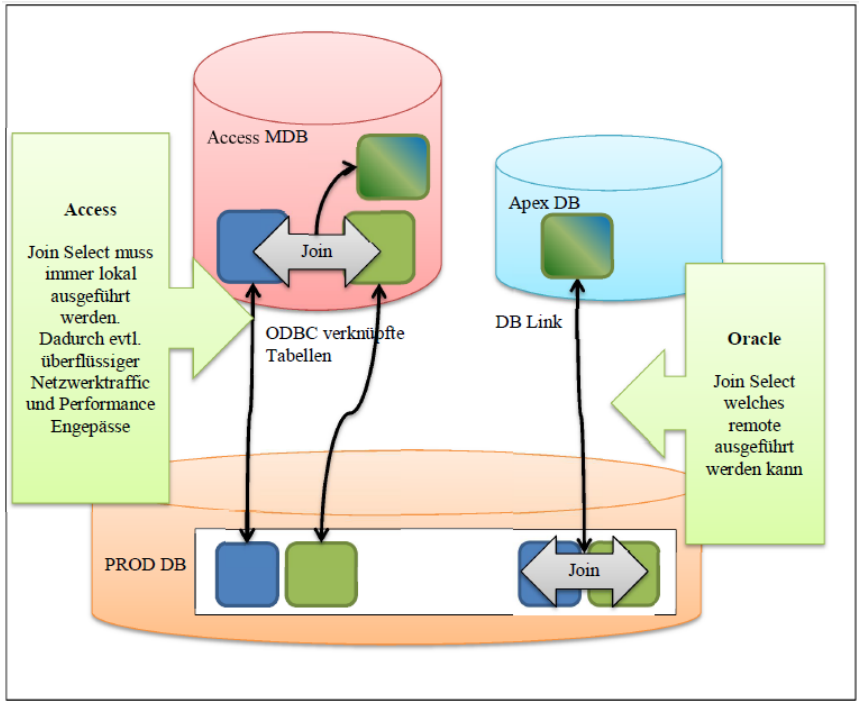
Datenmenge:
bigemp = 100.000 Zeilen,
department = 50 Zeilen

Resultat = 50 Zeilen

```
Select d.dept_name Department
      ,count(e.empno) Anzahl
from department d
left outer join bigemp e
on d.deptno = e.deptno;
```

Szenario 3 – Lösung

Zugriff auf mehrere DBs gleichzeitig



Abfrage remote ausführen

```
Select /*+ driving_site (e) */
       d.dept_name Department
      ,count(e.empno) Anzahl
from department d
left join bigemp e
on d.deptno = e.deptno;
```

syntegris information solutions

Zwischenfazit

Brown Field Projects:

APEX + DB Links ermöglichen Lösungen für schwierige Probleme, die im Zusammenhang mit existierenden System auftreten.

APEX liefert die RAD Umgebung, damit schnell neue Dinge bereitgestellt werden können.

DB Link stellt den minimal-invasiven Zugriff auf das vorhandene System her.

DB Link ist das ideale Werkzeug für die Kommunikation zwischen mehreren Oracle Datenbanken.

syntegris information solutions

Umsetzung

Rechte / Voraussetzungen

```
-- Rechte anlegen
Grant create database link to <user>;
```

Schema (Workspace) muss das Recht haben einen DB Link zu erstellen.

```
-- Link erzeugen
create database link "DOAGREMOTE"
connect to "DOAG"
identified by "doagdoag"
using '192.168.10.xxx:1521/service_name'
;
```

TNS Eintrag auf Serverebene, falls DB Link über TNSNAMES.ora erstellt wird. Alternativen: direct connect oder LDAP Namensauflösung.

```
-- Test
Select * from dual@doagremote;
```

Parameter: **Global_Names** beachten TRUE => Service Name der Datenbank muss als Linkname benutzt werden.

syntegris information solutions

Tipps & Tricks

Fallstricke und Lösungsmöglichkeiten

Im Folgenden werden einzelne Problemfelder beschrieben, die bei der Verwendung von DB Links in Apex auftreten können.

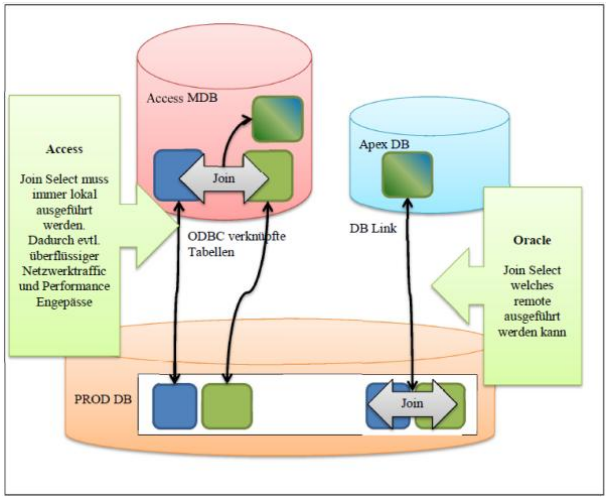
Die Symptome und die Ursachen für diese Probleme werden erklärt.

Danach erfolgt die Vorstellung von Lösungsansätzen oder Strategien zur Vermeidung des Problems.

Teilweise unter Verwendung einer Live Demo.

Tipps & Tricks

JOIN Performance



Problem:
Die Performance eines Joins hängt stark davon ab, ob die Daten lokal oder remote vorliegen.

Meist ist die Ergebnismenge eines SELECTs kleiner als die Menge an Ursprungsdaten.

Bei Remote Abfragen sollte so viel wie möglich auf der remote DB "gejoint" werden und erst das Ergebnis über den DB Link zur lokalen DB kopiert werden.

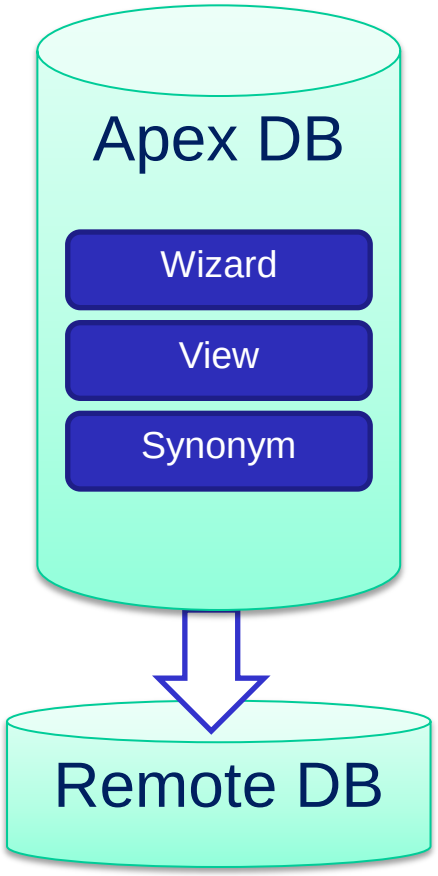
Oracle führt standardmäßig ein SELECT immer lokal aus, selbst wenn alle beteiligten Tabellen remote liegen.

Lösung:
Verwendung des driving_site hints.

syntegris information solutions

Tipps & Tricks

APEX Wizards



Problem: Die APEX Wizards können mit remote Tabellen nichts anfangen. Das stört bei Reports nur wenig, jedoch beim Anlegen von Forms.

Lösung: Jede Tabelle oder View auf die mittels DB Link zugegriffen wird, bekommt ein lokales Synonym und eine View die auf dieses Synonym zeigt.

Das **Synonym** dient nur dazu den DB Link aus den Apex Seiten fern zu halten. Es kommt doch häufig vor, dass der Name des Links sich ändert. Das Synonym bietet eine extra Abstraktionsschicht für minimalen Aufwand.

Die **View** dient dazu, den Apex Wizards ein lokales Objekt zu geben, auf dem der Wizard aufbauen kann. Sie dient als Abstraktionsschicht und bildet Meta Daten im lokalen Data Dictionary. Diese Meta Daten werden an einigen Stellen durch Apex verwendet. Am offensichtlichsten ist es bei den Form Wizards, aber auch andere Stellen in APEX funktionieren nicht oder schlecht, wenn diese Metadaten fehlen.

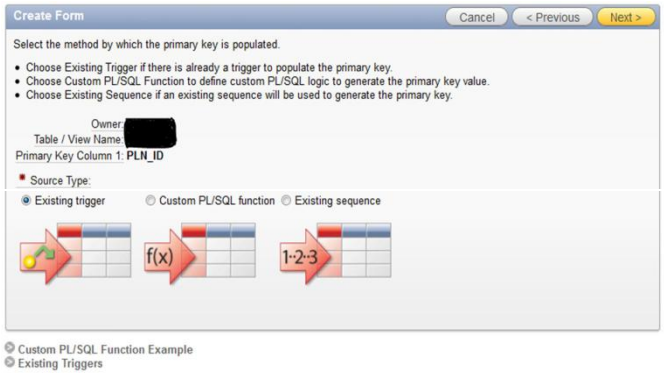
Hinweis: Wenn sich die Struktur der Ursprungstabelle ändert, muss die View auch neu erstellt werden, damit die Spaltenliste im DD aktualisiert wird.

syntegris information solutions

Tipps & Tricks

Error: returning IDs

Identity Spalten (kurz IDs) werden normalerweise mittels einer Oracle Sequence befüllt. Diese Sequence wird üblicherweise in einem BEFORE-ROW-INSERT Trigger auf der jeweiligen Tabelle aufgerufen und befüllt damit die ID Spalte (PK). Apex kann damit sehr gut umgehen. Bei Verwendung des Form Wizards, wird man gefragt, wie die Werte für den PK bestimmt werden sollen.



Problem: Die returning clause funktioniert aber nicht für remote Tabellen.

Apex verwendet die RETURNING clause um den Wert der ID Spalte nach einem Insert in das Feld zu laden. Dies tut Apex (Version 4.1) auch, wenn das Feld RETURNING gar nicht verwendet wird. Meiner Meinung nach ein Bug.

```
ORA-22816: Nicht unterstützte Funktion mit RETURNING-Klausel
ORA-06512: in Zeile x
22816. 00000 - "unsupported feature with RETURNING clause"
*Cause: RETURNING clause is currently not supported for object type
columns, LONG columns, remote tables, INSERT with subquery,
and INSTEAD OF Triggers.
*Action: Use separate select statement to get the values.
```

syntegris information solutions

Tipps & Tricks

returning Ids - Lösung 1: ROWID

Apex 4.1 kann anstelle von PK Spalten auch mittels ROWID den jeweiligen Datensatz bestimmen. Dies ist teilweise ein Workaround. Scheint aber nicht in allen Fällen zu funktionieren.

Bei Apex 4.2 wurden im Forum bereits neue Fehler in Zusammenhang mit ROWIDs berichtet (page load prozesse). Vermutlich wurde die Funktionalität in 4.2 etwas abgeändert.

Wertung: Eigentlich gute Lösung, hat aber nur bei 2 von 3 Masken funktioniert.

Tendenz: Auf Upgrade zu 4.2 warten, evtl. ist das Problem dann behoben.

Apex Patchset 4.1.1 Note: **7.3 Columns Used in Automatic Row Processing (DML) Processes**
Because of bug 12990445, the following changes have been implemented for Automatic Row Processing (DML) process types. The code which performs the INSERT has been changed to determine if the columns should be included in the INSERT statement. Note that these are the same checks which occur before an UPDATE. These new checks include:
Is the source type a DB Column?
Does the specified Build Option have a status of Include or is it not specified at all?
Is the page item contained in the POST request? For example, if the page item is conditional it will not be contained in the POST request if the condition evaluates to FALSE during page rendering.
Is the page item not of type Display Only where Save State is set to No?
The above new behavior is used for all new applications which are created in Oracle Application Express release 4.1.1.

syntegris information solutions

Tipps & Tricks

returning Ids – Lösung 2: INSERT

Ein extra insert process. Der normale DML Prozess wird nur noch für UPDATE und DELETE genommen. Für das Insert gibt es eine eigene Umsetzung ohne die returning clause.

Wertung: Umständlich, aber führt auf jeden Fall zu einer Lösung

"A computer once beat me at chess, but it was no match for me at kick boxing."
Emo Philips, American stand up comedian

returning Ids – Lösung 3: computation

Befüllen des Feldes aus der Sequence mittels einer Computation, kurz bevor das Insert gemacht wird.

Wertung: Diese Lösungsvariante halte ich für sehr schwach. Dafür ist ein extra remote Zugriff (sequence fetch) nötig und der Name der Sequence muss ebenfalls bekannt sein.

Dies hebelt das Konzept eines Triggers etwas aus, dass der Entwickler sich eigentlich nicht mehr um die Sequence kümmern muss.

Vorteil: Ist schnell umgesetzt.

syntegris information solutions

Tipps & Tricks

Audit Spalten und DB Trigger

Beispiel: DB Trigger

```
CREATE OR REPLACE TRIGGER DOAG.DEMOTRG_BRI
BEFORE INSERT ON DOAG.DEMOTRG
FOR EACH ROW
BEGIN
  if :new.id is null then
    :new.id := demotrg_seq.nextval;
  end if;
  :new.createdt := sysdate;
  :new.createuser
      := nvl(v('APP_USER'), user);
END;
/
```

Problem:

Ein DB Trigger schreibt den aktuell angemeldeten Apex-User in eine Spalte "EINGEFUEGT_VON". Der Trigger liegt auf der Remote DB und kann deshalb nicht auf v('APP_USER') zugreifen.

syntegris information solutions

Tipps & Tricks

Audit Spalten und DB Trigger

Lösung 1: Spalte über Apex selbst füllen.

Eine Hidden Column verwenden. Den Default Wert aus :APP_USER setzen.

Vorteil:
einfach, übersichtlich, nachvollziehbar, kein extra remote Code

Nachteil:
extra Programmieraufwand bei jeder Form Page.

*"I say a murder is abstract.
You pull the trigger and after that
you do not understand anything that happens."*
Jean-Paul Sartre

Tipps & Tricks

Audit Spalten und DB Trigger

Beispiel: DB Trigger

```
CREATE OR REPLACE TRIGGER DOAG.DEMOTRG_BRI
BEFORE INSERT ON DOAG.DEMOTRG
FOR EACH ROW
BEGIN
  if :new.id is null then
    :new.id := demotrg_seq.nextval;
  end if;
  :new.createdt := sysdate;
  :new.createuser
      := nvl(regexp_substr(
sys_context('USERENV','CLIENT_IDENTIFIER')
       ,'^[^:]*'),user);
END;
/
```

Lösung 2: Einen Applikations-Kontext für die Remote DB setzen.

Der remote DB-Trigger wertet dann diesen Kontext aus. Apex macht das sogar automatisch für die lokale DB. Für ältere Apex Versionen, kann der Kontext manuell gesetzt werden.

```
DBMS_SESSION.SET_IDENTIFIER(v('APP_USER'));

sys_context('USERENV' ,'CLIENT_IDENTIFIER')
```

Vorteil:
Reine Trigger Lösung, keine extra Logik in Apex.

Nachteil:
Trigger in der Remote-DB muss angepasst werden.
Erst ab Apex 4.x.

syntegris information solutions

Tipps & Tricks

LOBs via DB Link

Problem: Lobs können nicht über DB Links übertragen werden.

Die obige Aussage stimmt zum Glück nicht zu 100%. Es ist nur recht schwierig vernünftig mit LOBs umzugehen. Ein einfaches Select liefert aber einen Fehler, wenn lob Spalten vorhanden sind.

```
select clobColumn from myTable@remoteDb;

ORA-22992: LOB-Locators können nicht von Remote-Tabellen
verwendet werden
22992. 00000 - "cannot use LOB locators selected from remote
tables"
*Cause: A remote LOB column cannot be referenced.
*Action: Remove references to LOBs in remote tables.
```

Dieses Problem hat eigentlich nichts mit APEX zu tun.

Die Lösung basiert darauf das mit einem `insert .. select` Befehl ein LOB übertragen werden kann.

```
Insert into myTable i (clobColumn)
select r.clobColumn from myTable@remoteDb r;
```

syntegris information solutions

Tipps & Tricks

LOBs via DB Link

Beispiel: Dokumente Tabelle

```
CREATE TABLE AX_DOKUMENTE
  (DOK_ID   NUMBER(15,0) NOT NULL ENABLE,
   FILENAME VARCHAR2(100 BYTE),
   QUELLE   VARCHAR2(100 BYTE),
   DOKUMENT BLOB,
   MIMETYPE VARCHAR2(100 BYTE),
   PFAD     VARCHAR2(100 BYTE),
   TYP      VARCHAR2(20 BYTE),
   GEAENDERT_AM DATE,
   GEAENDERT_VON VARCHAR2(50 BYTE),
   FILESIZE      NUMBER(10,0),
   CONSTRAINT AX_DOKUMENTE_PK
      PRIMARY KEY (DOK_ID) USING INDEX
      TABLESPACE IX_MEDIUM ENABLE
   )
   TABLESPACE "DATA_SMALL"
   LOB ("DOKUMENT")
    STORE AS BASICFILE
  (TABLESPACE "DATA_HUGE" ENABLE STORAGE
   IN ROW CHUNK 8192 RETENTION NOCACHE
LOGGING);

COMMENT ON TABLE AX_DOKUMENTE
IS 'In der Tabelle AX_DOKUMENTE werden die
verschiedenen ''Images '', und die PDF''s als
BLOBs gespeichert.';

create sequence AX_DOKUMENTE_SEQ;
```

Lösung: Teil 1 – Report

„Refresh“ Button mit dem die Dokumente aktualisiert werden können.

Prozess:

```
declare
  v_count binary_integer;
begin
  delete from ax_dokumente;
  insert into ax_dokumente select * from
ax_dokumente_remote;
  v_count := sql%rowcount;
  apex_application.g_print_success_message :=
apex_application.g_print_success_message
                    || to_char(v_count)||' Dokumente
wurden erfolgreich abgeglichen!';
  commit;
end;
```

syntegris information solutions

Tipps & Tricks

LOBs via DB Link

Beispiel: Synonym auf die Tabelle und die Sequence in der Remote DB

```
CREATE OR REPLACE
SYNONYM "AX_DOKUMENTE_REMOTE"
FOR "AX_DOKUMENTE"@"DOAGREMOTE";

CREATE OR REPLACE SYNONYM
"AX_DOKUMENTE_REMOTE_SEQ"
FOR "AX_DOKUMENTE_SEQ"@"DOAGREMOTE";
```

Lösung: Teil 2 – Single DML Form

Vor dem DML – Sequence laden

```
begin
  :P6_DOK_ID := AX_DOKUMENTE_REMOTE_SEQ.NEXTVAL;
end;
```

Nach dem DML Prozess – BLOB aktualisieren

```
begin
  insert into ax_dokumente_remote i
  select *
  from ax_dokumente dl
  where dl.dok_id = :P6_DOK_ID;
  commit;
end;
```

Sonstige Tipps

- Schliessen von DB Links/remote sessions
  - `alter session CLOSE DATABASE LINK <dblink>`
- IR Date Spalten Filter Bug (Apex 3.2+10g)
  - u.U. Timestamp Spalten verwenden
- Remote package call
  - `package.procedure@dblink(parameter)`

Beispiel: remote ddl call um neue Spalte einzufügen

```
begin
   DBMS_UTILITY.EXEC_DDL_STATEMENT@DOAGREMOTE ('alter table demotrg add gebdat date');
end;
```

Danach view aktualisieren

```
create or replace view v_demotrg as select * from demotrg;
```

"A fear of heights is illogical.
A fear of falling, on the other hand, is prudent and evolutionary."
Dr. Sheldon Cooper, The Big Bang Theory

Fazit

Apex und DB Links

- Bisher habe ich kein Problem gefunden, welches durch die Kombination von Apex und DB Links entstanden ist und sich nicht lösen liess.
- Die Vorteile überwiegen meiner Erfahrung nach deutlich die Nachteile.
- Sehr empfehlenswert für Reporting Applikationen. Die meisten Schwierigkeiten traten bei Forms/Tabular Forms auf.
- Die Installation einer weiteren Oracle (APEX) DB muss auch unter dem Gesichtspunkt der Lizenzkosten sorgfältig abgewägt werden. Für einige Szenarien ist der Einsatz einer Oracle XE Version denkbar. In vielen Großunternehmen ist jedoch die Installation einer weiteren Oracle DB praktisch lizenzkostenfrei. Entweder weil die entsprechenden Server bzw. VMs bereits eine Oracle DB betreiben oder es gibt sogar eine unlimited license.
- Die organisatorischen Hürden können durch Apex + DB Links deutlich entschärft werden.
- **RAD mittels APEX auch für Brown Field Projects möglich!**

syntegris information solutions GmbH, Hermannstr. 54-56, 63263 Neu-Isenburg

"I would trade all of my technology for an afternoon with Sokrates."
Steve Jobs

syntegris information solutions

Fragen?

Apex und DB Links