Zentralisation & Hochverfügbarkeit im globalen Systembetrieb

Datenbankkonsolidierung und Hot Deployment in der Praxis

Christian Piasecki

PITSS GmbH

19.11.2015 2015

© PITSS GmbH 2015

Agenda

- Fokus und Zielsetzung des Projekts
- Organisatorische und technische Rahmenparameter
- Technische Umsetzung
  - Analyse
  - Herstellung der Mandantenfähigkeit
  - Datenübernahme
  - Edition-Based Redefinition
- Status und Ausblick

© PITSS GmbH 2015

Über mich

- Beratung, Training, Entwicklung
  Oracle Technologie
  - Oracle Apex
  - Oracle BI Suite
  - Oracle BI Publisher
  - Oracle Warehouse Builder
  - Oracle Data Integrator

Christian Piasecki

Consultant

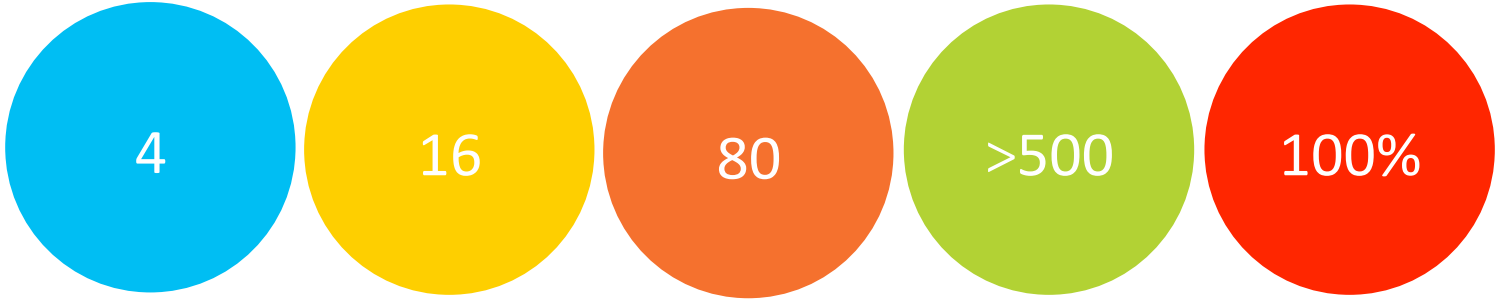
XING

@cpiasecki23

http://pitss.de/blog

© PITSS GmbH 2015

Über PITSS



und...

Locations

Germany 2x
UK
USA

Jahre

1999
gegründet

Kollegen

Kompetente
& erfahrene
Experten

Kunden

Zufriedene
Kunden in
über 40 Ländern

Oracle

Oracle Gold Partner
& spezialisiert auf
Oracle Technologien

© PITSS GmbH 2015

Über PITSS

Tool

für

High Performance

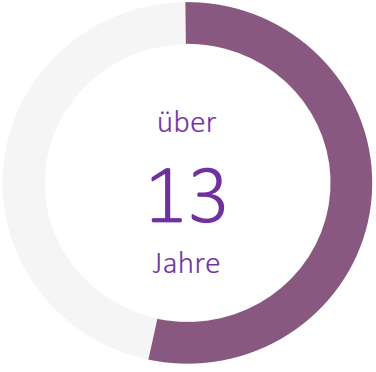
Oracle Applikationen

© PITSS GmbH 2015

Unser Focus

Forms

über Jahre gesammelte Erfahrung: von Forms 2.0... bis 12c



SOA

erfahren mit SOA, WebService Technologien, OAS, WLS, BEPL, BPM

ADF

erfolgreiche ADF Migrationsprojekte seit 2006

Mobile

erfolgreiche Mobile Projekte für Scanner, Handhelds, WinCE, iOS, Android, MAF

© PITSS GmbH 2015

Unsere Referenzen



© PITSS GmbH 2015

Fokus und Zielsetzung des Projekts

- Bestehende Systemlandschaft – dezentrale Struktur:
  - Mehrere Standorte
  - Jeder Standort hat eigene Systemlandschaft (Datenbank & Anwendungen)
  - Gründe:
    - Unabhängigkeit von externen Netzwerken & Netzverfügbarkeit
    - Historisch gewachsen
  - Nachteile:
    - Überwachung und Wartung von mehreren Systemen

    → Vervielfachung von Arbeitsaufwand und Kosten
- Neue Systemlandschaft – zentrale Struktur
  - Mehrere Standorte
  - Eine Systemlandschaft für alle Standorte zusammen
  - Möglich durch:
    - Weiterentwicklung von Hardware
    - neue Möglichkeiten von Datenbanken und Anwendungssoftware

→ Reduzierung von Weiterentwicklungs- und Wartungsaufwand und laufenden Kosten

© PITSS GmbH 2015

Organisatorische und technische Rahmenparameter

- 16 Standorte in unterschiedlichen Zeitzonen
- Systemumgebung(en):
  - pro Standort eine Oracle 11gR2 Datenbank EE → 16 Datenbanken
  - 16 x 1 (gleiche) Forms-Anwendung

→Big Bang Ansatz mit Zusammenführung aller DBs zu einem Zeitpunkt nicht möglich, da sonst Produktionsstillstand an mehreren Standorten

→Einzelne Datenbanken werden im Vorfeld für die Zusammenführung vorbereitet und nach und nach in einer neuen Masterdatenbank konsolidiert

© PITSS GmbH 2015

Mehrwert für Benutzer des System

- Finanzierung des Projekts teilweise durch Fachabteilung benötigt
  → Es muss ein Mehrwert für die Benutzer sichergestellt werden
- Aktuell müssen sich beim Einspielen eines Updates alle Benutzer vom System abmelden und es kann nicht gearbeitet werden, so dass die Zeitfenster für das Ausrollen von Änderungen eingeschränkt sind
- Durch die Konsolidierung und den unterschiedlichen Zeitzonen gibt es quasi diese Fenster nicht mehr

→Einführung einer Hot Deployment-Fähigkeit

- Einspielen von Updates im laufenden Systembetrieb
- Einsatz der Edition-Based-Redefinition Funktion der Datenbank

© PITSS GmbH 2015

Aufgabenverteilung

- PITSS:
  - Analyse der aktuellen Datenbank und Forms-Anwendung und Zurverfügungstellung von Abweichungen
  - Zur Verfügung stellen von Skripten für Herstellung der Mandanten- und Hot-Deployment-Fähigkeit
  - Skripte für die Übernahme der Daten in die neue Master-Datenbank
- Kunde:
  - Korrigieren der Abweichungen in den 16 Datenbanken
  - Anpassen der Forms-Sourcen
  - Erweitern der 16 Datenbanken um die Mandantenfähigkeit
  - Aufsetzen einer neuen Master-Datenbank (Mandanten-, und Hot Deployment-fähig)

© PITSS GmbH 2015

Technische Umsetzung

- Zusammenführen der 16 Schemata in einer neuen Master-Datenbank
- Einführung eines Mandanten für jeden Standort
- Nur noch 1 Forms Anwendung
- Analyse-Phase nötig um die Datenbanken vorher gleichzuziehen, da
  - Bekannt ist, dass es in der jahrelangen Entwicklung zu Abweichungen gekommen ist
  - und es sonst nach der Konsolidierung zu Funktions- und Performanceproblemen kommen kann
- Einsatz von PITSS.CON als Analyse-Tool
- Eine aktuelle QS-Datenbank wurde als Referenz genommen, gegen die alle anderen DBs verglichen wurden

© PITSS GmbH 2015

Analyse

- Datenbank
  - init.ora
  - Tabellen und View Definitionen
  - Indexe
  - Primary, Foreign und Unique Keys
  - Check-Constraints
  - Packages, Proceduren und Funktionen
  - Hints
  - Aufrufe von dbms_alert
  - Aufrufe von dbms_pipe
  - sysdate
  - Select * from... –Aufrufe
  - Insert into Tabelle values.. (ohne Spalten-Angabe)
- Forms
  - Hints
  - Aufrufe von dbms_alert
  - Aufrufe von dbms_pipe
  - sysdate
  - Select * from... –Aufrufe
  - Insert into Tabelle values.. (ohne Spalten-Angabe)

© PITSS GmbH 2015

Analyseergebnisse

Init.ora

- Keine Abweichung

Tabellen und View Definitionen

- Eine Abweichung, die vom Kunden korrigiert wurde

Indexe

- Es wurden einige unterschiedliche Index-Definitionen gefunden, die zu möglichen Performanceproblemen nach der Konsolidierung hätten führen können
- Manuelle Entscheidung und Korrektur nötig

Primary, Foreign und Unique-Keys sowie Check-Constraints

- Unterschiedliche Definitionen würden zu Problemen bei der Datenkonsolidierung führen
- Herausforderung: Analyse musste über die Definition der Objekte und nicht die Namen gehen, da vor allem bei PK und Check-Constraints vom System generierte Namen benutzt wurden
- Die gefundenen Abweichungen mussten von der Kunden IT-Abteilung überprüft und korrigiert werden

© PITSS GmbH 2015

Analyseergebnisse

Packages, Proceduren und Funktionen

- Keine gravierende Abweichungen, nur verschlüsselte Werte und Objekte die für den Zugriff auf die anderen DBs genutzt wurden

Hints

- Nutzung unterschiedlicher Hints könnte nach der Zusammenführung die Performance beeinflußen
- Es wurde ein Hint in den Forms gefunden, dessen Index aber in allen DBs gleich definiert war → keine Anpassung nötig

dbms_alert & dbms_pipe

- Für die korrekte Funktionsweise muss die Information vorhanden sein, aus welchem Standort sie aufgerufen werden → Erweiterung um einen Mandantenparameter
- da es nur gekapselte Aufrufe aus der DB gab, mussten nur geringe Anpassungen gemacht werden

sysdate

- Aufrufe von Sysdate mussten in Forms durch eine eigene Funktion ersetzt zu werden, um in den unterschiedlichen Zeitzonen nicht die Serverzeit zunehmen
- Da es die Funktion schon vorher gab, waren die Änderungen marginal

© PITSS GmbH 2015

Analyseergebnisse

Select * from & Insert into table values

- Für die Konsolidierung werden die Tabellen um eine Mandantenspalte erweitert, so dass die 2 Befehle zu Fehlern führen würden
- Analyse des ganzen DB- und Forms wurde mit PITSS.CON durchgeführt
- 5 betroffene Stellen wurden dann manuell angepasst

© PITSS GmbH 2015

Herstellung der Mandantenfähigkeit

- Tabellen
  - Erweitern der Tabellen um eine Mandantenspalte mit einem ALTER-Befehl
  - Ausnahme: Tabellen mit Standortübergreifenden Daten
- Constraints und Indexe
  - Um die Performance und die Eindeutigkeit der Daten sicherzustellen mussten auch diese Objekte angepasst werden
  - Nutzung von dbms_metadata.get_dll
    - Liefert die Create-Befehle mit aktueller Definition der Objekte
    - Erweitern um die Mandantenspalte
- Zuordnung von Benutzer zum Mandanten
  - Einsatz einer Zuordnungstabelle
  - 1 Benutzer – 1 Standort
  - Erstellen einer Funktion, die zum Benutzer den Mandanten liefert

© PITSS GmbH 2015

Herstellung der Mandantenfähigkeit

- Befüllen der Mandantenspalte
  - Keine Veränderung am Source-Code, deswegen
  - Erstellen von Triggern für jede Tabelle um automatisiert die Mandantenspalte zu füllen
- Datensichtbarkeit
  - Benutzer sollen nur Daten Ihres Standort sehen
  - Einsatz von VPD
  - Setzen des Context im Logon-Trigger
  - Erstellen von Policies für die Tabellen

© PITSS GmbH 2015

Datenübernahme

Mehrstufiger Prozess um die Downtime für die Benutzer möglichst gering zu halten

Phase 1:

- Überprüfen ob die Tabellen- und Constraint-Struktur auf der Export-DB der Struktur auf der Master-DB entspricht

  → sonst würde der Import möglicherweise fehlschlagen und die Benutzer hätten sich unnötig vom System abgemeldet

  → passt die Struktur kann Phase 2 erfolgen, wenn nicht, muss die Struktur korrigiert werden

Phase 2:

- Auf dem Altsystem müssen sich die Benutzer abmelden und die Daten werden per Datapump exportiert
- Auf dem Zielsystem werden die Daten per Datapump in ein Stagingschema importiert

© PITSS GmbH 2015

Datenübernahme

Phase 3:

- Auf dem Zielsystem müssen sich alle Benutzer abmelden
- Deaktivieren von Constraints und Triggern um den Import zu beschleunigen
- Generieren von Insert-Statements für die zu übernehmenden Daten
  - Herausforderung: Berechnen von Offsets für Primary- und Foreign-Keys
- Ausführend der Inserts

Phase 4:

- Aktivieren von Constraints und Triggern
- Berechnung von Statistiken und aktualisieren der Indexe

→ Benutzerzugriff erlauben

© PITSS GmbH 2015

Einführung der Hot Deployment-Fähigkeit

- Zukünftig wird es keine Zeitfenster mehr für Updates geben oder es müssen sich alle Benutzer weltweit abmelden
- Es wird eine Möglichkeit benötigt um Updates im laufenden Betrieb einspielen zu können

→Edition-Based Redefinition

- Forms-Anpassungen/Application Server

© PITSS GmbH 2015

Neuer Upgrade-Prozess

- Vorbereiten des Upgrades
  - Erstellen des neuen Releases parallel zum aktuellen
  - Applikation läuft im alten Release
- Umschalten auf das neue Release
  - Neue Sessions laufen mit dem neuen Release
  - Alte Session (bis sie beendet werden) laufen mit dem alten Release

Doch wie funktioniert EBR und was ist vorzubereiten?

© PITSS GmbH 2015

Edition-Based Redefinition

- Feature der Oracle Datenbank EE
- Verfügbar an Version 11gR2
- Editions
  - Mehrere Versionen (Editions) des gleichen Objekts in der Datenbank
  - Objekt ist eindeutig durch die Kombination aus Objektname, Schema und Edition
  - Datenbank Session läuft immer im Context einer Edition
  - Es gibt immer eine Default-Edition
    - Muss deswegen beim Anmelden nicht mitgeben werden

```
SELECT property_value
FROM database_properties
WHERE property_name = 'DEFAULT_EDITION';

PROPERTY_VALUE
--------------------------------------------------------------------------------
ORA$BASE
```

```
ALTER DATABASE DEFAULT EDITION = edition-name;
```

```
ALTER SESSION SET EDITION = release_v1;

SELECT SYS_CONTEXT('USERENV', 'SESSION_EDITION_NAME') AS edition
FROM dual;
EDITION
------------------------------------------------------------------------------
RELEASE_V1
```

© PITSS GmbH 2015

Edition anlegen

```
CREATE EDITION edition-name;
CREATE EDITION edition-name AS CHILD OF parent-edition;
```

- AS CHILD KLAUSEL optional in 11gR2
- Eine Edition kann nur ein Child haben (für zukünftige DB Versionen)
- Es kann keine Parent Edition gelöscht werden

```
CREATE EDITION release_v1;
CREATE EDITION release_v2 AS CHILD OF release_v1;
CREATE EDITION release_v3;
SELECT * FROM dba_editions;

EDITION_NAME                   PARENT_EDITION_NAME            USA
------------------------------ ------------------------------ ---
ORA$BASE                                                      YES
RELEASE_V1                     ORA$BASE                       YES
RELEASE_V2                     RELEASE_V1                     YES
RELEASE_V3                     RELEASE_V2                     YES
```

© PITSS GmbH 2015

Edition-Based Redefinition

- Editionable Objekte
  - Function
  - Trigger
  - Synonym
  - Procedure
  - Package + Package Body
  - Type + Type Body
  - View
  - Library
- Non Editionable Objekte
  - Tables
  - Public Synonyme
  - Sequences

→ Tabellen sind nicht Editionable, deswegen gibt es Editioning Views und Crossedition Trigger!!!

© PITSS GmbH 2015

Editioning Views

- Umsetzung von Änderungen im Datenmodel mit Hilfe von Editioning Views
- Dienen als Wrapper für Anpassungen an Tabellen, da sie in unterschiedlichen Versionen unterschiedlich definiert sein können.
- Sind einfache Views auf jeweils eine Tabelle
- Alle Operationen sollen auf Views ausgeführt werden
- Trigger sollen auch an Views und nicht an Tabellen hängen (Können so auch pro Edition unterschiedlich sein)

```
CREATE OR REPLACE EDITIONING VIEW view-name AS
SELECT      col1,
            col3,
            col4 AS new_name
FROM table-name;
```

© PITSS GmbH 2015

Crossedition Triggers

- Unterschiedliche Benutzer können parallel in unterschiedlichen Editionen unterwegs sein
- Crossedition Trigger sorgen dafür, das DML-Operationen unabhängig von der Edition funktionieren und die Benutzer konsistente Daten sehen
- Forward Crossedition Trigger überführen die Daten aus den Spalten in der alten Version in die Spalten in der neuen Version
- Reverse Crossedition Trigger überführen die Daten aus den Spalten in der neuen Version in die Spalten in der alten Version
- Sobald alle Benutzer/Sessions nur mit der neuen Version arbeiten und die Trigger nicht mehr benötigt werden, können Sie gelöscht werden
- Trigger werden nur in der neuen Edition angelegt

© PITSS GmbH 2015

Arbeiten mit Editioning

- Alle Aktionen arbeiten auf den Views und nicht auf den Tabellen
- Für jede Tabelle gibt es einen Editioning View, der die Tabelle 1:1 abbildet
- Trigger hängen an Editioning Views
- VPD auf Editioning Views und nicht auf den Tabellen
- Benutzer haben Berechtigungen auf den Editioning Views und keine Berechtigungen an den Tabellen
- Oracle garantiert den identischen Execution Plan für eine Abfrage auf einen Editioning View oder die Basistabelle

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
ALTER SESSION SET EDITION = release_v1;
SELECT SYS_CONTEXT('USERENV', 'SESSION_EDITION_NAME') AS edition FROM dual;

EDITION
 --------------------
RELEASE_V1
```

```
CREATE TABLE employees_tab (
                employee_id NUMBER(5) NOT NULL,
                name VARCHAR2(40) NOT NULL,
                date_of_birth DATE NOT NULL,
                CONSTRAINT employees_pk PRIMARY KEY (employee_id)
                );

CREATE SEQUENCE employees_seq;
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
CREATE OR REPLACE EDITIONING VIEW employees AS
SELECT employee_id,
            name,
            date_of_birth
FROM employees_tab;


CREATE OR REPLACE PROCEDURE create_employee (p_name IN employees.name%TYPE,
                                                p_date_of_birth IN employees.date_of_birth%TYPE) AS
BEGIN
            INSERT INTO employees (employee_id, name, date_of_birth)
            VALUES (employees_seq.NEXTVAL, p_name, p_date_of_birth);
END create_employee;
 /
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
BEGIN
      create_employee('Peter Parker', TO_DATE('01-JAN-2010', 'DD-MON-YYYY'));
      COMMIT;
END;

SELECT * FROM employees;

EMPLOYEE_ID          NAME                                     DATE_OF_B
 -----------         ---------------------------------------- ---------
2                    Peter Parker                             01-JAN-10
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
ALTER SESSION SET EDITION = release_v2;

BEGIN
    create_employee('Clark Kent', TO_DATE('02-JAN-2010', 'DD-MON-YYYY'));
    COMMIT;
END;
 SELECT * FROM employees;
EMPLOYEE_ID          NAME                                    DATE_OF_B
 -----------         ---------------------------------------- ---------
2                    Peter Parker                            01-JAN-10
3                    Clark Kent                              02-JAN-10
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
ALTER TABLE employees_tab ADD
                ( postcode VARCHAR2(20)
);
```

```
ALTER SESSION SET EDITION = release_v1;

BEGIN
    create_employee(Flash Gordon', TO_DATE('03-JAN-2010', 'DD-MON-YYYY'));
    COMMIT;
END;

 SELECT * FROM employees;

EMPLOYEE_ID          NAME                                    DATE_OF_B
 -----------         ---------------------------------------- ---------
2                    Peter Parker                            01-JAN-10
3                    Clark Kent                              02-JAN-10
4                    Flash Gordon                            03-JAN-10
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
ALTER SESSION SET EDITION = release_v2;
CREATE OR REPLACE EDITIONING VIEW employees AS
SELECT      employee_id,
            name,
            date_of_birth,
            postcode
FROM employees_tab;
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
CREATE OR REPLACE PROCEDURE create_employee (p_name IN employees.name%TYPE,
                                             p_date_of_birth IN employees.date_of_birth%TYPE,
                                             p_postcode IN employees.postcode%TYPE) AS
BEGIN
        INSERT INTO employees (employee_id, name, date_of_birth, postcode)
        VALUES (employees_seq.NEXTVAL, p_name, p_date_of_birth, p_postcode);
END create_employee;

BEGIN
        create_employee('Mighty Mouse', TO_DATE('04-JAN-2010', 'DD-MON-YYYY'), 'AA1 2BB');
        COMMIT;
END;
```

© PITSS GmbH 2015

Beispiel 1: Ohne Datentransformation

```
SELECT * FROM employees;

EMPLOYEE_ID  NAME                                   DATE_OF_B  POSTCODE
-----------  ---------------------------------------- ---------  --------------------
2            Peter Parker                           01-JAN-10
3            Clark Kent                             02-JAN-10
4            Flash Gordon                           03-JAN-10
5            Mighty Mouse                           04-JAN-10  AA1 2BB
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
ALTER TABLE employees_tab ADD (
                first_name VARCHAR2(20),
                last_name VARCHAR2(20)
);
```

```
ALTER SESSION SET EDITION = release_v3;

CREATE OR REPLACE EDITIONING VIEW employees AS
SELECT      employee_id,
            first_name,
            last_name,
            date_of_birth,
            postcode
FROM employees_tab;
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
CREATE OR REPLACE PROCEDURE create_employee (p_first_name IN employees.first_name%TYPE,
                                             p_last_name IN employees.last_name%TYPE,
                                             p_date_of_birth IN employees.date_of_birth%TYPE,
                                             p_postcode IN employees.postcode%TYPE) AS
BEGIN
        INSERT INTO employees (employee_id, first_name, last_name, date_of_birth, postcode)
        VALUES (employees_seq.NEXTVAL, p_first_name, p_last_name, p_date_of_birth, p_postcode);
END create_employee;
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation – Forward Trigger

```
CREATE OR REPLACE TRIGGER employees_fwd_xed_trg
   BEFORE INSERT OR UPDATE ON employees_tab
   FOR EACH ROW
   FORWARD CROSSEDITION
   DISABLE
BEGIN
   :NEW.first_name := SUBSTR(:NEW.name, 1, INSTR(:NEW.name, ' ')-1);
   :NEW.last_name := SUBSTR(:NEW.name, INSTR(:NEW.name, ' ')+1);
END employees_fwd_xed_trg;
```

- Beide Trigger müssen in der neuen Edition angelegt werden
- Trigger enablen nicht vergessen

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation – Reverse Trigger

```
CREATE OR REPLACE TRIGGER employees_rvrs_xed_trg
  BEFORE INSERT OR UPDATE ON employees_tab
  FOR EACH ROW
  REVERSE CROSSEDITION
  DISABLE
BEGIN
  :NEW.name := :NEW.first_name || ' ' || :NEW.last_name;
END employees_rvrs_xed_trg;
```

```
ALTER TRIGGER employees_fwd_xed_trg ENABLE;
ALTER TRIGGER employees_rvrs_xed_trg ENABLE;
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
UPDATE employees_tab SET name = name;
```

- Schritt nötig, damit die Alt-Daten in die neuen Spalten überführt werden
- Falls es zu viele Daten sind, hier in Chunks abarbeiten

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
SELECT SYS_CONTEXT('USERENV', 'SESSION_EDITION_NAME') AS edition FROM dual;
EDITION
------------------------------------------------------------------------------------------------
RELEASE_V3
```

```
BEGIN
        create_employee('Wonder', 'Woman', TO_DATE('01-JAN-2010', 'DD-MON-YYYY'), 'A11 2BB');
        COMMIT;
END;
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
SELECT * FROM employees;
EMPLOYEE_ID  FIRST_NAME           LAST_NAME            DATE_OF_B  POSTCODE
-----------  --------------------  -------------------- ---------  --------------------
2            Peter                 Parker               01-JAN-10
3            Clark                 Kent                 02-JAN-10
4            Flash                 Gordon               03-JAN-10
5            Mighty                Mouse                04-JAN-10  AA1 2BB
6            Wonder                Woman                01-JAN-10  A11 2BB
```

```
SELECT * FROM employees_tab;
EMPLOYEE_ID  NAME          DATE_OF_B  POSTCODE   FIRST_NAME           LAST_NAME
-----------  --------------- ---------  ---------- --------------------  --------------------
2            Peter Parker    01-JAN-10             Peter                Parker
3            Clark Kent      02-JAN-10             Clark                Kent
4            Flash Gordon    03-JAN-10             Flash                Gordon
5            Mighty Mouse    04-JAN-10  AA1 2BB    Mighty               Mouse
6            Wonder Woman    01-JAN-10  A11 2BB    Wonder               Woman
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
ALTER SESSION SET EDITION = release_v2;
BEGIN
        create_employee('Inspector Gadget', TO_DATE('01-JAN-2010', 'DD-MON-YYYY'), 'A12 2BB');
        COMMIT;
END;
```

© PITSS GmbH 2015

Beispiel 2: Mit Datentransformation

```
SELECT * FROM employees;
EMPLOYEE_ID NAME                 DATE_OF_B  POSTCODE
----------- ------------------------ --------- ----------
2           Peter Parker         01-JAN-10
3           Clark Kent           02-JAN-10
4           Flash Gordon         03-JAN-10
5           Mighty Mouse         04-JAN-10  AA1 2BB
6           Wonder Woman         01-JAN-10  A11 2BB
7           Inspector Gadget     01-JAN-10  A12 2BB
```

```
SELECT * FROM employees_tab;
EMPLOYEE_ID NAME           DATE_OF_B  POSTCODE   FIRST_NAME           LAST_NAME
----------- -------------- --------- ---------- -------------------- --------------------
2           Peter Parker   01-JAN-10             Peter                Parker
3           Clark Kent     02-JAN-10             Clark                Kent
4           Flash Gordon   03-JAN-10             Flash                Gordon
5           Mighty Mouse   04-JAN-10  AA1 2BB    Mighty               Mouse
6           Wonder Woman 01-JAN-10    A11 2BB    Wonder               Woman
7           Inspector Gadget 01-JAN-10 A12 2BB    Inspector            Gadget
```

© PITSS GmbH 2015

Was ist in der Forms-Anwendung zu tun

Anpassung Forms:

- Im ersten Forms Dialog den Befehl:

  ALTER SESSION SET EDITION = <passende Edition zur Forms-Anwendung>;

  einbauen.
- Parameter kann in der formsweb.cfg mitgegeben werden

© PITSS GmbH 2015

Wie bringe ich die Änderungen ohne Downtime in die Produktion - Ablauf

- Änderungen in Datenbank in neuer Version entwickeln und testen
- Anpassungen in Dialogen entwickeln und testen
- Neue Edition in Produktion zur Verfügung stellen
- Neue Dialoge in Produktion deployen → Benutzer arbeiten noch mit alter Applikation und alter Edition
- Neue Sessions auf neue Dialoge umleiten (Load-Balancer, Config der Formsweb.cfg)
- Sobald alle Benutzer nur noch mit der neuen Version arbeiten, alte Anwendungs-Version und Cross-Edition-Trigger entfernen

© PITSS GmbH 2015

Welche Anpassungen mussten gemacht werden

- Umbenennen der Tabellen (_TAB)
- Anlegen der EDITIONING VIEWS auf die Tabellen
- Rechte auf Tabellen entziehen und auf EDITIONING VIEWS zuweisen
- Trigger rekompilieren
- PL/SQL-Code und Views rekompilieren
- VPD auf EDITIOING VIEWS umstellen

© PITSS GmbH 2015

Projektstatus und Ausblick

- Analysephase abgeschlossen
- Master-DB erstellt
- Mandantenfähigkeit an allen Standorten eingeführt
- 5 von 16 Standorten in Master-DB konsolidiert

- Einführung des Hot Deployment verzögert sich wegen Einwänden vom DB-Betrieb
  - Kein Einsatz von Public Synonymen
  - Security-Einwände (grant create any Edition)
  - Einwand, dass die Struktur zu unübersichtlich wird durch die verschiedenen Edtionen
  - Umsetzung mit mehreren DB-Schemata vorgeschlagen, von uns abgelehnt, das Aufdopplung der Objekte und keine konsistenten Daten gewährleistet

© PITSS GmbH 2015

Q&A

© PITSS GmbH 2015

PITSS auf der DOAG

- Stand 206, 2.OG
- Weitere Vorträge:

  MAF – start small and simple to extend Forms to mobile
  Stephan La Rocca
  Mittwoch, 18.11., 10:00 – 10:45, Raum Kopenhagen

  Und dann kam der Datenschutz
  Stephan La Rocca
  Donnerstag, 19.11., 14:00 – 14:45, Raum Shanghai

  Warum sind meine Abschätzungen so ungenau?
  Jon Erik de Linde
  Donnerstag, 19.11., 15:00 – 15:45, Raum Prag

- Schulungstag:

  Mathias Waedt: Einfacher Start in die mobile Anwendungsentwicklung mit MAF.

© PITSS GmbH 2015