

# Löschfunktionen in BALVI iP

Bearbeitet am: 05.08.2014

Version: Stand iP 1.22.0f

## Inhalt

1	Einführung .....	2
1.1	Neuerungen mit dem Paket vom August 2014 .....	2
1.2	Oracle-Kenntnisse vorausgesetzt .....	2
1.3	Installationshinweise .....	3
1.4	Bekannte Probleme während der Ausführung .....	3
2	Funktionsbeschreibungen .....	4
2.1	DELETE_D90 (OlderThan DATE) - Änderungsprotokolltabelle .....	4
2.2	DELETE_D92 (OlderThan DATE) - Leseprotokolltabelle .....	4
2.3	DELETE_D800201 (OlderThan DATE) - TSN-Export-Tabelle .....	5
2.4	DELETE_D91 (OlderThan DATE) - Anwendungsprotokolleinträge .....	6
2.5	DELETE_D0303 (OlderThan DATE) - gelöschte Dokumenten .....	7
2.6	DELETE_WKD_UAMT() -Arbeitstabellen für Untersuchungsamt-Importe (TS).....	7
2.7	DELETE_WKD_AVV() -Arbeitstabellen für Untersuchungsamt-Importe (LM) .....	8
2.8	DELETE_WKD_FM() -Arbeitstabellen für Untersuchungsamt-Importe (FM).....	9
2.9	DELETE_WKD_TSK() -Arbeitstabellen für die Schnittstelle zur Tierseuchen-Kasse (TSK) .....	9
2.10	DELETE_HITLOG (OlderThan DATE) - Löschen von HIT-Batch-Protokollen .....	10
2.11	DELETE_S3103 - Datenübernahme-Protokolle.....	10
3	Hinweise zur Konfiguration und Verwaltung von Oracle Jobs .....	11
3.1	Allgemeine Hinweise zur Ausführung .....	11
3.2	Allgemeine Hinweise zur Verwaltung von Oracle Jobs .....	11
3.2.1	Wrapper zur Ausführung verwenden .....	12
3.2.2	Den Wrapper selber schreiben .....	13
3.2.3	Empfehlungen .....	15
4	Oracle Tuning-Hinweise.....	15
4.1	Tabellen verkleinern - SHRINK SPACE .....	16
4.1.1	Beispiel: Dokumente .....	16
4.1.2	Beispiel: Arbeitstabellen WKD_AVV* .....	17
4.2	Arbeitsschritte nach einer Tabellenverkleinerung .....	17
4.2.1	Reparatur und verkleinern von Indizes .....	17
4.2.2	Compilieren aller ungültigen Objekte .....	19
4.2.3	Datenbank-Statistiken aktualisieren .....	19

# 1 Einführung

Dieses Dokument beschreibt zusätzlich zur regulären Programmfunktion verfügbare Löschfunktionen für Protokoll-Tabellen und Funktionen zur Datenbereinigung.

Die hier beschriebenen Funktionen bzw. die Anwendung dieser Funktionen ist nicht Bestandteil der regulären Auslieferung von BALVI iP, sondern gezielt durch einen technischen Administrator (z.B. Datenbank-Administrator) nach Rücksprache mit dem Fachverfahrensbetreuer einzustellen.

Zudem sind diese Funktionen nicht für die Archivierung von Daten vorgesehen, sondern löschen ausschließlich nicht zu archivierende Daten aus der Anwendung. Dieses kann z.B. erforderlich sein, um das Datenbank-Wachstum zu reduzieren oder wenn aus datenschutzrechtlichen Gründen eine Löschung vorgeschrieben ist.

Für die Nutzung der Löschfunktionen, welche nur als Datenbankfunktionen (nicht Programmfunktionen) verfügbar sind, wird zudem vorausgesetzt, dass der Ausführende die erforderlichen Kenntnisse im DBMS Oracle besitzt, um die folgenden Anweisungen korrekt auszuführen.

## 1.1 Neuerungen mit dem Paket vom August 2014

Mindestvoraussetzung zur Nutzung ist BALVI iP 1.22.0f oder höher.

Zur Initialauslieferung vom Juni 2013 wurden noch folgende weitere Löschfunktionen bereitgestellt:

- 1.) DELETE\_WKD\_AVV()
- 2.) DELETE\_WKD\_FM()
- 3.) DELETE\_WKD\_TSK()

Diese wurden in den Wrapper-Funktionen DELETE\_LOG() und DELETE\_ALL\_LOG() ergänzt. Wenn Sie diese Funktionen schon innerhalb eines Oracle DBMS\_JOB verwenden, sollte nach der Aktualisierung geprüft werden, ob der JOB hinterher weiterhin funktionsfähig ist.

Sollten sie einen eigenen Wrapper in Verwendung haben, so sollte dieser überprüft und ggf. um die neuen Funktionen ergänzt werden.

## 1.2 Oracle-Kenntnisse vorausgesetzt

Da es sich bei den im Paket enthaltenen Funktionen um reguläre Oracle PL/SQL-Procedures handelt, wird zur Ausführung vorausgesetzt, dass ein versierter Datenbank-Administrator die erforderlichen Werkzeuge (z.B. SQL-Editor) besitzt und die Zugangsdaten (Benutzername und Passwort) des jeweiligen BALVI iP Schemas kennt.

Die Procedures sind als Löschfunktionen für Massendaten konzipiert, da der Datenbestand, welcher zu löschen ist, ggf. mehrere Millionen Datensätze beinhalten kann. Die erste

Ausführung jeder einzelnen Funktion kann daher mehrere Stunden Laufzeit erfordern. Daher wird dringend davon abgeraten, die erste Ausführung (Initial-Löschung) im laufenden Betrieb durchzuführen, sondern diese in einen Wartungszeitraum von 1-2 Tagen einzuplanen.

Jede beschriebene Funktion ist zudem in der Lage, als Oracle DBMS Job ausgeführt zu werden, um die neu aufgelaufenen Daten, z.B. jede Nacht oder jedes Wochenende, zu bereinigen. Die Funktionen sind dabei meistens datumsorientiert, was bedeutet, dass die Einstellung eines Datums immer heißt: Alle Datensätze, welche Älter sind als das angegebene Datum. Oracle bietet dafür z.B. die Funktion (SYSDATE - {n Tage}) an, um immer aus der Sicht des aktuellen Datums alle alten Daten zu löschen.

### 1.3 Installationshinweise

Dem Dokument liegt ein Skript „ora\_Functions\_DELETE\_LOG.sql“ bei. Dieses muss als Schema-Besitzer der Anwendung BALVI iP ausgeführt werden. Zur Ausführung wurde bei BALVI der Oracle SQL-Developer verwendet, welcher kostenfrei bei Oracle zum Download bereitgestellt wird. Die Ausführung sollte jedoch mit jedem gängigen SQL-Editor (z.B. TOAD) möglich sein.

Nach der Installation (bzw. der Ausführung des Skriptes) wird keine Aktion in BALVI iP oder der Datenbank ausgeführt. Um die Skripte bzw. Procedures zu verwenden, lesen sie bitte die Hinweise im Abschnitt „Hinweise zur Konfiguration und Verwaltung von Oracle Jobs“.

Es wird empfohlen, min. einen UNDO-Tablespace mit 10GB bereit zu stellen.

### 1.4 Bekannte Probleme während der Ausführung

Probleme bei der Ausführung betreffen primär die Initiallöschung, können jedoch bei großen Datenmengen jederzeit auftreten.

Es wird während der Löschung in regelmäßigen Schritten ein COMMIT abgesetzt, jedoch ist ein starkes Wachstum des UNDO-Tablespaces nicht ausgeschlossen. Das liegt ggf. an der eingestellten UNDO\_RETENTION (Default=900 s). Bitte beachten Sie, dass BALVI keine Anpassungen an der UNDO\_RETENTION vornimmt.

Vgl. : [http://docs.oracle.com/cd/B19306\\_01/server.102/b14231/undo.htm#sthref1477](http://docs.oracle.com/cd/B19306_01/server.102/b14231/undo.htm#sthref1477)

Es kann ggf. der Fehler „ORA-01555: Snapshot zu alt: Rollback-Segmentnummer 9 namens „\_SYSSMU9\$“ ist zu klein“ auftreten. Siehe dazu Oracle Support: [ID 1307334.1]

Auszug aus dem Oracle Support-Artikel:

„LOB Issues:

*Out-of-row LOB undo is maintained in the LOB segment. So the UNDO tablespace and undo retention is not associated with most LOB ORA-1555 issues. ... In environments with high*

*updates, deletes on rows including LOBs, the chances of ORA-1555 on LOB undo is very high.“*

Die Masslöschung von Spalten mit LOB's ist Ziel dieses Skript-Paketes. Sollte trotzdem ein Fehler vom Typ ORA-01555 auftreten, warten Sie bitte bis der UNDO-Tablespace wieder frei ist und starten das gewünschte Skript erneut. Da die Löschung der Datensätze Fehlertolerant entwickelt wurde, sollte im nächsten Durchlauf die betroffene Funktion das Löschen fortsetzen. Sollte es wider erwarten trotzdem in einer Funktion mehrfach Probleme geben, wenden Sie sich an den BALVI Support, damit wir die entsprechende Funktion optimieren können.

## 2 Funktionsbeschreibungen

### 2.1 DELETE\_D90 (OlderThan DATE) - Änderungsprotokolltabelle

Die Tabelle VIS\_D90 wird i.d.R. nicht gefüllt, wenn BALVI iP verwendet wird. Für einzelne Kunden hat BALVI jedoch Datenbank-Trigger erstellt, welche die Änderungen an Datensätzen protokolliert. Wenn Sie diese Protokoll-Trigger nicht aktiv haben, bleibt diese Tabelle leer.

Wenn die Protokollierung aktiviert wurde, werden hier alle Neuanlagen, Änderungen und Löschungen atomar gespeichert. Wenn z.B. ein Anwender in einem vorhandenen Datensatz drei Werte ändert, werden drei Protokolleinträge angelegt, je geändertem Wert ein Datensatz mit Wert Alt und Wert Neu, welche Tabelle, Primary Key des Datensatzes in der Tabelle, welcher Benutzer, Zeit und Änderungshinweise. Daher ist das Datenaufkommen in dieser Tabelle sehr hoch.

Eine regelmäßige Löschung sollte einen Zeitraum von min. 30 Tagen bis max. 90 Tagen an Änderungen dokumentieren, daher sollte die Funktion folgendermaßen verwendet werden:

```
-- Beispiel: alles was älter als 30 Tage ist
BEGIN
  DELETE_D90 (SYSDATE-30);
END;
```

Wenn sie die Änderungsprotokollierung nicht verwenden, dauert die Ausführung der Prozedur nur wenige Sekunden, es ist daher sicherheitshalber immer möglich, die Prozedur auszuführen.

### 2.2 DELETE\_D92 (OlderThan DATE) - Leseprotokolltabelle

Die Tabelle VIS\_D92 wird i.d.R. nicht gefüllt, wenn nicht in BALVI iP (oder dem BALVI Kommunikationsserver) die Leseprotokollierung aktiviert wurde. Die Leseprotokollierung

speichert alle ausgeführten SQL-Statements vollständig ab, nicht die vom Anwender ausgelesenen Daten. Trotzdem ist das Wachstum in der Tabelle sehr groß, da z.B. zum Öffnen der Detailansicht des Objektes „Betrieb“ schon ca. 10-15 Abfragen ausgeführt werden, um die Daten für die Detailmaske zu laden (Betriebsadresse, Fachbereiche, Betriebsarten, Nummern, Hoftierarzt, ... abhängig von den Fachbereichen des jeweiligen Betriebs und Anwenders).

In BALVI iP kann dieses über die Einstellung in der BALVIIP.INI erfolgen, im Kommunikationsserver wird die Einstellung über die CONTEXT.XML vorgenommen. Default von BALVI ist „aus“.

Eine regelmäßige Löschung sollte einen Zeitraum von min. 30 Tagen bis max. 90 Tagen an Ausführungen dokumentieren, daher sollte die Funktion folgendermaßen verwendet werden:

```
-- Beispiel: alles was älter als 30 Tage ist
BEGIN
  DELETE_D92(SYSDATE-30);
END;
```

Wenn sie die Leseprotokollierung nicht verwenden, dauert die Ausführung der Prozedur nur wenige Sekunden, es ist daher sicherheitshalber immer möglich, die Prozedur auszuführen.

## 2.3 DELETE\_D800201 (OlderThan DATE) - TSN-Export-Tabelle

In BALVI iP wird generell die Schnittstelle „Export an TSN (3.0)“ angeboten. Diese erzeugt eine DBF-Datei mit Spalten zu den aktuellen Tierangaben aller Betriebe im Zuständigkeitsbereich. Da die DBF-Datei nach der Erstellung durch TSN3 eingelesen werden muss und dazu ein Pfad fest in TSN hinterlegbar ist, wurde dem Dateinamen kein Datum mitgegeben, die Datei heißt immer „DAVIDTSN.DBF“. bei einer weiteren Ausführung wird die Datei überschrieben. Daher wurde auch auf Anforderung einzelner Kunden eine mit der DBF-Datei identische Tabelle VIS\_D800201 angelegt, um alle exportierten Daten zu protokollieren. Die Protokollierung erfolgt automatisch und kann nicht abgeschaltet werden.

Ein gravierendes Wachstum der Tabelle ist daher zu erwarten, wenn Sie den Export alle Daten (Landesexport) an TSN automatisiert jede Nacht laufen lassen. Bei einem Kunden wurde dabei ein Wachstum der Tabelle um ca. 150.000 Datensätze pro Nacht festgestellt, daher wird eine Löschung in regelmäßigen Abständen angeraten.

Bei der regelmäßigen Löschung sollte einen Zeitraum von min. 7 Tagen bis max. 90 Tagen an Exportdaten vorgehalten und die die Funktion folgendermaßen verwendet werden:

```
-- Beispiel: alles was älter als 7 Tage ist
BEGIN
  DELETE_D800201(SYSDATE-7);
END;
```

## 2.4 DELETE\_D91(OlderThan DATE) - Anwendungsprotokolleinträge

Das Anwendungsprotokoll wurde eingeführt, um herauszufinden, welcher Benutzer spezielle Skripte, Vorlagen oder andere kritische Funktionalitäten in der Anwendung ausgeführt hat. Es gibt keine Gesamtauswertung in der Oberfläche, um diese Informationen auszuwerten. Diverse weitere Funktionen wie z.B. die Hitbatch-Schnittstelle, die Untersuchungsamt-Schnittstelle, die Update.exe, der BALVI Benchmark schreiben innerhalb der Sitzung des angemeldeten Benutzers zusätzliche Informationen in die Protokolle.

Die Löschfunktion entfernt nahezu alle Protokolleinträge, welche innerhalb einer Benutzersitzung aufgetreten sind. Das sind automatisch auch Protokolleinträge, welche z.B. die Untersuchungsamt-Schnittstelle, das TSN-Export-Protokoll oder die HIT-Schnittstelle erzeugt haben. Es wäre also nicht unbedingt erforderlich, diese fachlichen Spezialfunktionen (DELETE\_WKD\_UAMT() und DELETE\_HITLOG() oder DELETE\_D800201()) aufzurufen, um die Einträge der Tabelle VIS\_D91 zu entfernen. Es gibt teilweise jedoch zusätzliche Tabellen, welche mit der entsprechenden Schnittstelle befüllt werden, welche nicht über die Funktion DELETE\_D91() gelöscht werden.

Zudem besteht auch die Möglichkeit, dass die allgemeinen Login- und Skript-Ausführungsinformationen länger aufgehoben werden sollen, als die HIT-, TSN- oder Untersuchungsamt-Protokolle, da das nachvollziehen z.B. der Ausführung von Korrektur-Skripten auch min. 1 Jahr in der Vergangenheit möglich sein soll.

In diesem Fall würden Sie andere Zeitmaßstäbe an die unterschiedlichen Funktionen stellen. Beispiel:

- Alle HIT-Protokolle sollen max. 3 Monate aufgehoben werden
- Alle TSN-Export-Protokolle sollen max. 1 Monat aufgehoben werden
- Alle UAMT-Protokolle sollen max. 6 Monate aufgehoben werden (Default von BALVI)
- Alle anderen Protokolle, z.B. die Ausführung von Korrektur-Skripten, Informationen, wann ein Update eingespielt worden ist, ob SQL-Work benutzt wurde, sollen 1 Jahr aufgehoben werden.

Die Zusammenstellung der Ausführung sieht dann folgendermaßen aus:

```
BEGIN
-- alle (allgemeinen) Systemprotokolleinträge
-- ein Jahr aufheben
DELETE_D91(SYSDATE-365);
-- TSN-Exporte: alles was älter als 30 Tage ist löschen
DELETE_D800201(SYSDATE-30);
-- Untersuchungsamt-Importe löschen
-- Default 6 Monate, kein Datum einstellbar
DELETE_WKD_UAMT();
-- HIT-Protokolle 3 Monate aufheben
DELETE_HITLOG(SYSDATE-90);
END;
```

## 2.5 DELETE\_D0303 (OlderThan DATE) - gelöschte Dokumenten

BALVI iP hat eine integrierte Dokumentenablage, welche es ermöglicht, zu jeglichen Daten in der Oberfläche zusätzlich Dateien hinzu zu fügen, z.B. Kontrollberichte oder auch Bilder. Diese Dokumente verbleiben auch noch in der Datenbank, wenn der Anwender diese aus der Oberfläche zum Löschen markiert, da BALVI iP nur eine logische Löschung durchführt, um ggf. in der Lage zu sein, die Löschung wieder rückgängig zu machen.

Es kommt jedoch öfter vor, dass die Anwender Dokumente löschen und damit ggf. dennoch sehr viel Speicherplatz in der Datenbank beschlagnahmen. Diese gelöschten Dokumente können endgültig (physikalisch) gelöscht werden, wenn die Anwender sich nicht innerhalb von n Tagen melden, um die Löschung rückgängig zu machen. Das Verhalten entspricht hier z.B. dem im Windows bekannten „Papierkorb leeren“.

Nach der ersten Löschung der Tabelleninhalte werden voraussichtlich mehrere Gigabyte an Speicher wieder frei. Diese bleiben jedoch bis auf weiteres bei Oracle als freie Blöcke vorhanden, könnten jedoch bei Bedarf über ein ALTER TABLE wieder freigegeben werden. Diese ist nicht Teil dieser Beschreibung.

Bei der regelmäßigen Löschung sollte einen Zeitraum von min. 7 Tagen bis max. 90 Tagen an gelöschten Dokumenten (Papierkorb) vorgehalten und die die Funktion folgendermaßen verwendet werden:

```
-- Beispiel: alles was älter als 7 Tage ist
BEGIN
  DELETE_D0303(SYSDATE-7);
END;
```

Hinweise für DBA's: Bitte beachten Sie, dass die gelöschten Datensätze in einem Oracle BLOB liegen und zu einer Fragmentierung des LOB-Segments der Tabelle VIS\_D0303 führen können. Eine regelmäßige Defragmentierung oder die Verlegung dieses LOB-Segments in einem separaten TABLESPACE mit UNIFORM SIZE wird empfohlen. BALVI bietet für diese Optimierung die Möglichkeit einer Vor-Ort-Besprechung und technischen Betreuung dieser Umstellung.

## 2.6 DELETE\_WKD\_UAMT() -Arbeitstabellen für Untersuchungsamt-Importe (TS)

Die Schnittstelle Untersuchungsamt-Ergebnisse Tierseuchen wird in vielen Bundesländern eingesetzt, um regelmäßig (teilweise täglich) die vom Labor gelieferten Dateien mit Untersuchungsergebnissen (Befunden) in BALVI iP einzulesen. Da die Schnittstelle mehrstufig arbeitet, also ein landesweiter Import und Aufteilung auf die einzelnen Mandanten vornimmt, werden Arbeitstabellen gefüllt, welche ebenfalls noch einige Zeit nach dem Import verfügbar sein müssen, um ggf. vorliegende Zuordnungsprobleme zu analysieren. Zudem werden für jeden Verarbeitungsschritt Protokolle angelegt. Diese Angaben wurden bislang nicht automatisch gelöscht. Die Prozedur hat keine Möglichkeit,

eine Zeitabgabe zu definieren, da die Vorhaltezeit von BALVI vorerst auf 6 Monate festgelegt wurde. Die Prozedur löscht dabei mehrstufig auch alle entstandenen Protokolle aus der VIS\_D91 (vgl. DELETE\_D91()) und Inhalte der Arbeitstabellen.

Der Anwender sieht diese Löschung z.B. im Objekt „Importverwaltung“, da der Datensatz, welche dort angezeigt wird (Version des Imports, wann gestartet, von wem, Protokollübersicht), ebenfalls gelöscht wird.

Es wird empfohlen, diese Prozedur regelmäßig auszuführen, jedoch nur dann, wenn nicht gleichzeitig gerade ein neuer Import läuft.

```
-- Beispiel:  
-- In dieser Funktion ist kein Datum durch den Kunden einstellbar  
BEGIN  
  DELETE_WKD_UAMT();  
END;
```

Hinweis:

BALVI geht davon aus, dass in Zukunft auch die Protokolle der Schnittstellen „Untersuchungsamt-Import (Lebensmittel)“ und „Untersuchungsamt-Import (Futtermittel)“ über dieselbe Routine gelöscht werden. Diese sind bislang noch nicht verfügbar.

## 2.7 DELETE\_WKD\_AVV() -Arbeitstabellen für Untersuchungsamt-Importe (LM)

Die Schnittstelle Untersuchungsamt-Ergebnisse Lebensmittel wird in vielen Bundesländern eingesetzt, um regelmäßig (teilweise täglich) die vom Labor gelieferten Daten einzulesen.

Äquivalent zur Prozedur DELETE\_WKD\_UAMT hat diese keine Möglichkeit, eine Zeitabgabe zu definieren, da die Vorhaltezeit von BALVI vorerst auf 6 Monate festgelegt wurde. Die Prozedur löscht dabei mehrstufig auch alle entstandenen Protokolle aus der VIS\_D91 (vgl. DELETE\_D91()) und Inhalte der Arbeitstabellen.

Der Anwender sieht diese Löschung z.B. im Objekt „Importverwaltung“, da der Datensatz, welche dort angezeigt wird (Version des Imports, wann gestartet, von wem, Protokollübersicht), ebenfalls gelöscht wird.

Es wird empfohlen, diese Prozedur regelmäßig auszuführen, jedoch nur dann, wenn nicht gleichzeitig gerade ein neuer Import läuft.

```
-- Beispiel:  
-- In dieser Funktion ist kein Datum durch den Kunden einstellbar  
BEGIN  
  DELETE_WKD_AVV();  
END;
```



## 2.8 DELETE\_WKD\_FM() -Arbeitstabellen für Untersuchungsamt-Importe (FM)

Die Schnittstelle Untersuchungsamt-Ergebnisse Futtermittel wird in einigen Bundesländern eingesetzt, um regelmäßig (teilweise täglich) die vom Labor gelieferten Daten einzulesen.

Äquivalent zur Prozedur DELETE\_WKD\_UAMT hat diese keine Möglichkeit, eine Zeitabgabe zu definieren, da die Vorhaltezeit von BALVI vorerst auf 6 Monate festgelegt wurde. Die Prozedur löscht dabei mehrstufig auch alle entstandenen Protokolle aus der VIS\_D91 (vgl. DELETE\_D91()) und Inhalte der Arbeitstabellen.

Der Anwender sieht diese Löschung z.B. im Objekt „Importverwaltung“, da der Datensatz, welche dort angezeigt wird (Version des Imports, wann gestartet, von wem, Protokollübersicht), ebenfalls gelöscht wird.

Es wird empfohlen, diese Prozedur regelmäßig auszuführen, jedoch nur dann, wenn nicht gleichzeitig gerade ein neuer Import läuft.

```
-- Beispiel:  
-- In dieser Funktion ist kein Datum durch den Kunden einstellbar  
BEGIN  
  DELETE_WKD_FM();  
END;
```

## 2.9 DELETE\_WKD\_TSK() -Arbeitstabellen für die Schnittstelle zur Tierseuchen-Kasse (TSK)

Die Schnittstelle TSK wird in vielen Bundesländern eingesetzt, um manuell die gelieferten Daten einzulesen. Da es bei der TSK-Schnittstelle in diversen Bundesländern andere Regelungen gibt, wurde die Vorhaltezeit von BALVI vorerst auf 2 Jahre festgelegt.

Äquivalent zur Prozedur DELETE\_WKD\_UAMT hat diese Funktion daher ebenfalls keine Möglichkeit, eine Zeitabgabe zu definieren. Die Prozedur löscht dabei mehrstufig auch alle entstandenen Protokolle aus der VIS\_D91 (vgl. DELETE\_D91()) und Inhalte der Arbeitstabellen.

Der Anwender sieht diese Löschung z.B. im Objekt „Importverwaltung“, da der Datensatz, welche dort angezeigt wird (Version des Imports, wann gestartet, von wem, Protokollübersicht), ebenfalls gelöscht wird.

Es wird empfohlen, diese Prozedur regelmäßig auszuführen, jedoch nur dann, wenn nicht gleichzeitig gerade ein neuer Import läuft.

```
-- Beispiel:  
-- In dieser Funktion ist kein Datum durch den Kunden einstellbar  
BEGIN  
  DELETE_WKD_TSK();  
END;
```

## 2.10 DELETE\_HITLOG (OlderThan DATE) - Löschen von HIT-Batch-Protokollen

Die Schnittstelle BALVI HitBatch benutzt die von HI-Tier gelieferte Anwendung HIT-Batch, um Daten mit HI-Tier auszutauschen. Da in dieser Anwendung diverse Fehler und Warnungen fachlicher und technischer Natur zurückgegeben werden, werden diese ebenfalls in Protokoll Datensätzen in BALVI iP gespeichert, damit die Anwender die Fehler und Warnungen sichten und bearbeiten können. Diese Protokolle wurden bislang nicht gelöscht, was auch zu einem Wachstum der Tabelle VIS\_D91 (vgl. Prozedur DELETE\_D91()) führt, da die Protokolle oftmals mehrere 100 kb groß werden.

Bei regelmäßigem Abgleich im Nachtbatch können so über mehrere Jahre alle Meldungen aus HIT nachvollzogen werden, jedoch ist eine Größe von ca. 2-8 GB reiner Protokollinformationen schon aufgetreten. Daher wird dazu geraten, diese Protokolle zu löschen, wenn die Tabelle VIS\_D91 zu groß wird.

Die Einträge werden automatisch entfernt, wenn sie die Funktion DELETE\_D91() verwenden, ggf. möchten Sie jedoch wegen der Größe bzw. Menge der Protokolleinträge ein kürzeres Löschintervall für die Hit-Protokolle verwenden.

```
BEGIN
  -- alle (allgemeinen) Systemprotokolleinträge
  -- ein Jahr aufheben
  DELETE_D91(SYSDATE-365);
  -- HIT-Protokolle 3 Monate aufheben
  DELETE_HITLOG(SYSDATE-90);
END;
```

## 2.11 DELETE\_S3103 - Datenübernahme-Protokolle

Die Datenübernahme aus Hamlet 2000 und David 99 sind in BALVI iP in Arbeitstabellen abgelegt worden. Diese sind nur für die Nachbearbeitung relevant und können nach Abschluss der Datenübernahme gelöscht werden. Der Protokoll-Kopfsatz (Eintrag im Objekt „Importverwaltung“) bleibt dabei erhalten.

Die Prozedur hat keine Einstell-Möglichkeit für ein Datum. Ausführung (einmalig):

```
BEGIN
  DELETE_S3103();
END;
```

## 3 Hinweise zur Konfiguration und Verwaltung von Oracle Jobs

Diese kurze Zusammenfassung der Funktionen sollten für jeden Oracle Administrator ausreichend sein, um die Verwaltung von Oracle Jobs zu verstehen. Sollten jedoch weiterführende Fragen zur Konfiguration auftreten, bitten wir Sie, die Dokumentation von Oracle zu konsultieren.

### 3.1 Allgemeine Hinweise zur Ausführung

Die Reihenfolge der Ausführung ist frei wählbar. Es spielt keine Rolle, ob sie zuerst DELETE\_D91() und danach DELETE\_HITLOG() ausführen. Wenn beide angegebenen Datumswerte identisch sind, ist z.B. das Ausführen von DELETE\_HITLOG() nicht unbedingt erforderlich, jedoch problemlos möglich. Das sollte die Gesamtausführungsgeschwindigkeit nicht sonderlich beeinträchtigen. Es sollte jedoch darauf geachtet werden, dass die Funktionen, welche in der Tabelle VIS\_D91 Löschungen vornehmen, nicht parallel ausgeführt werden, sondern ausschließlich sequentiell hintereinander.

BALVI liefert zudem Beispiel- bzw. Standard-Wrapper, welche mehrere Aufrufe bündeln, um das Arbeiten mit Datenbank-Jobs zu vereinfachen. Diese können natürlich so wie geliefert verwendet werden, wenn jedoch granulare Einstellungen erwünscht sind, kann BALVI nicht für jeden Kunden und Fall einen Wrapper bereitstellen. Dieses muss dann vom Administrator über die Datenbank-Job- Konfiguration gelöst werden.

### 3.2 Allgemeine Hinweise zur Verwaltung von Oracle Jobs

Führen sie alle folgenden Befehle als Schema-Besitzer von BALVI iP aus, nicht als SYS oder SYSTEM! Dieses erfordert für den Schema-Besitzer das Grant-Recht CREATE JOB, welches i.d.R. über die Rolle RESOURCE schon zugewiesen sein sollte. Zur Ausführung können sie einen SQL-Editor Ihrer Wahl verwenden, z.B. Oracle SQL-Developer.

Alternativ besteht die Möglichkeit, den Job auch als SYSTEM anzulegen. Dieses wird aus organisatorischer Sicht jedoch nicht empfohlen.

Um die Lösch-Prozeduren automatisch und zeitgesteuert auszuführen, wird ein Oracle User-Job angelegt. Die Uhrzeit für den Beispiel-Job ist auf 23.00 Uhr (23/24) nachts eingestellt. Sollten Sie mehrere BALVI iP Schemata in einer Datenbank-Instanz hosten, führen Sie diesen Befehl mehrmals (für jedes Schema einmal) aus und ändern die Ausführungszeit entsprechend ab, damit es keine Überschneidungen gibt. Beachten Sie bitte auch, dass dieser Job nicht laufen sollte, während eine Datenbanksicherung erfolgt.

Um zu prüfen, ob bereits ein JOB vorliegt, bzw. die Jobnummer und den Status zu erfahren, nutzen Sie bitte folgende Kontrollabfrage:

```
SELECT job, last_date, next_date, next_sec, failures, broken,  
       SUBSTR(what,1,200) DESCRIPTION
```

```
FROM
  user_jobs;
```

Neuen Job hinzufügen:

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  dbms_job.submit( job
    , 'DELETE_LOG(SYSDATE-90);'
    , TRUNC(SYSDATE) + 23/24
    , 'TRUNC(SYSDATE) + 1 + 23/24'
    , FALSE
  );
END;
```

Falls Sie den Job wieder entfernen wollen, nutzen Sie folgenden Befehl, wobei die Jobnummer mittels der Kontrollabfrage ermittelt werden muss:

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  job := 12345;
  dbms_job.remove(job);
END;
```

Den Job sofort ausführen, wobei die Jobnummer mittels der Kontrollabfrage ermittelt werden muss:

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  job := 12345;
  dbms_job.run(job);
END;
```

Falls Sie den Job ändern wollen, nutzen Sie folgenden Befehl, wobei die Jobnummer mittels der Kontrollabfrage ermittelt werden muss:

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  job := 12345;
  dbms_job.change(job
    , 'DELETE_LOG(SYSDATE-[NEUE Anzahl Tage]);' -- neuer Befehl
    , NULL, NULL); -- Keine Änderung im Ausführungszeitpunkt
end;
```

### 3.2.1 Wrapper zur Ausführung verwenden

Der Standard-Wrapper DELETE\_LOG(OlderThan DATE) ist schon eine logisch zusammengefasste Ausführungsreihenfolge, welche als Job ausgeführt werden kann.

Dieser ist folgendermaßen aufgebaut:

```
CREATE OR REPLACE PROCEDURE DELETE_LOG (OlderThan DATE) AS
```

```
BEGIN
  DELETE_D90 (OlderThan);
  DELETE_D800201 (OlderThan);
  DELETE_D91 (OlderThan);
  DELETE_D92 (OlderThan);
  DELETE_WKD_UAMT ();
  DELETE_WKD_AVV ();
  DELETE_WKD_FM ();
  DELETE_WKD_TSK ();
  DELETE_HITLOG (OlderThan);
END;
/
```

Die Logik enthält alle normalen Lösch-Funktionen, welche Protokoll-Objekte beinhalten, jedoch keine Spezialfunktionen wie DELETE\_S3103() oder DELETE\_D0303(). Allen Funktionen wird genau ein einheitliches Datum übergeben. Die Verwendung im Datenbank-Job ist daher auch relativ simpel möglich, wie in der Job-Verwaltung beschrieben.

Beispiel (einfach): Den von BALVI gelieferten Wrapper ausführen:

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  dbms_job.submit( job
    , 'DELETE_LOG (SYSDATE-90);'
    , TRUNC (SYSDATE) + 23/24
    , 'TRUNC (SYSDATE) + 1 + 23/24'
    , FALSE
  );
END;
```

### 3.2.2 Den Wrapper selber schreiben

Wenn Sie eigene Ausführungsschritte in einem Job zusammenfassen möchten, müssen Sie jedoch nicht unbedingt eine eigene Prozedur erstellen. Die Jobverwaltung ermöglicht auch die Ausführung von komplexeren Befehlen, solange diese nicht 4000 Zeichen (max. VARCHAR2) überschreiten.

Im folgenden Beispiel werden die oben beschriebenen Prozeduren selbst zusammengestellt und mit unterschiedlichen Zeiten deklariert.

Beispiel (erweitert):

```
DECLARE
  job BINARY_INTEGER;
BEGIN
  dbms_job.submit( job
    , 'BEGIN ' || chr(10) ||
      ' DELETE_D800201 (SYSDATE-14);' || chr(10) ||
      ' DELETE_WKD_UAMT ();' || chr(10) ||
      ' DELETE_HITLOG (SYSDATE-90);' || chr(10) ||
      ' DELETE_D0303 (SYSDATE-30);' || chr(10) ||
      ' DELETE_D91 (SYSDATE-365);' || chr(10) ||
      'END;'
    , TRUNC (SYSDATE) + 23/24
    , 'TRUNC (SYSDATE) + 1 + 23/24'
```

```
, FALSE  
);  
END;
```

Die blau markierten Textpassagen sind zur Übersichtlichkeit gedacht und erzeugen Zeilenumbrüche im PL/SQL. Wenn Sie dieses nicht möchten, können Sie den gesamten Befehl auch einzeilig schreiben. Beachten Sie jedoch bitte, dass zwischen BEGIN und dem ersten Befehl kein Semikolon eingefügt wird:

```
DECLARE  
  job BINARY_INTEGER;  
BEGIN  
  dbms_job.submit( job  
    , 'BEGIN DELETE_D800201(SYSDATE-14); DELETE_WKD_UAMT(); DELETE_HITLOG(SYSDATE-30); DELETE_D91(SYSDATE-365); END;'  
    , TRUNC(SYSDATE) + 23/24  
    , 'TRUNC(SYSDATE) + 1 + 23/24'  
    , FALSE  
  );  
END;
```

### 3.2.3 Empfehlungen

Das initiale Löschen von Daten kann sehr lange dauern, speziell dann, wenn schon mehrere Jahre mit diversen Schnittstellen und den automatischen Nachtbatch-Durchläufen gearbeitet wurde.

Nutzen Sie im ersten Schritt die Funktion „DELETE\_ALL\_LOG(SYSDATE-365)“, um alle Protokolle zu löschen, welche älter als ein Jahr sind. Je nach Anforderungen sollte dann ein spezieller Job angelegt werden, um die gezielten Lösch-Aufgaben optimal einzustellen. Die folgende Tabelle sollte dabei hilfreich sein:

Funktion	Empfehlung BALVI	Einstellung
DELETE_D90 (Änderungsprotokolle)	Normalerweise inaktiv, daher nicht erforderlich	-
DELETE_D91 (Systemprotokolle)	1 Jahr vorhalten	SYSDATE - 365
DELETE_D92 (Leseprotokolle)	Normalerweise inaktiv, daher nicht erforderlich	-
DELETE_D800201 (TSN-Exporte)	14 Tage vorhalten	SYSDATE - 14
DELETE_WKD_UAMT	Keine Datum einstellbar	(ausführen)
DELETE_WKD_AVV	Keine Datum einstellbar	(ausführen)
DELETE_WKD_FM	Keine Datum einstellbar	(ausführen)
DELETE_WKD_TSK	Keine Datum einstellbar	(ausführen)
DELETE_S3103 (DÜ-Protokolle)	Einmalig (kein Datum)	(ausführen)
DELETE_HITLOG (HIT-Protokolle)	Einstellung nur erforderlich, wenn die Prozedur DELETE_D91 sehr lange Zeiträume (z.B. SYSDATE-365) eingestellt hat und die HIT-Protokolle früher gelöscht werden sollen. 3 Monate vorhalten	SYSDATE-90
DELETE_D0303 (gel. Dokumente)	1 Monate vorhalten	SYSDATE-30

Die hier angegebenen Beispiele sind im Kapitel „Den Wrapper selber schreiben“ als Beispiel-PL/SQL für den Job hinterlegt.

## 4 Oracle Tuning-Hinweise

Das Löschen von Tabelleninhalten verringert die Datenmengen, jedoch wird der bislang verwendete Speicherplatz nicht freigegeben. Oracle bietet diverse Möglichkeiten an, den Speicherplatz auch wieder freizugeben, jedoch werden die erforderliche Schritte hier nur exemplarisch am Beispiel VIS\_D0303 (Dokumente) erläutert.

Da die Rahmenbedingungen zum Tuning nur in Zusammenarbeit zwischen dem Rechenzentrum und BALVI erarbeitet werden können, bietet BALVI für das Tuning spezielle Beratungen an.

## 4.1 Tabellen verkleinern - SHRINK SPACE

### 4.1.1 Beispiel: Dokumente

Die folgenden Schritte sind als Beispiel für die Tabelle VIS\_D0303 bzw. zu Freigabe des freien Speicherplatzes konzipiert, nachdem die Funktion „DELETE\_D0303()“ das erste Mal ausgeführt wurde. Die Ausführung sollte außerhalb des regulären Betriebs erfolgen, nachdem eine Sicherung der Datenbank durchgeführt wurde! Wenn Sie diese Bereinigung nutzen möchten, gehen Sie bitte folgendermaßen vor (ebenfalls als SCHEMA-Besitzer, z.B. BALVI\_PROD):

Freie Extends wieder freigeben, dieses kann je nach Tabellengröße und IO-Performance mehrere Stunden dauern!

```
ALTER TABLE VIS_D0303 ENABLE ROW MOVEMENT;  
ALTER TABLE VIS_D0303 SHRINK SPACE;
```

LOB-Segmente müssen separat bereinigt werden:

```
ALTER TABLE VIS_D0303 MODIFY LOB (CONTENT) (SHRINK SPACE);
```

Row-Movement wieder abschalten:

```
ALTER TABLE VIS_D0303 DISABLE ROW MOVEMENT;
```

Alle Indizes neu aufbauen:

```
ALTER INDEX PK_D0303_ID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_A0303_01_SID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_D0101_ID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_D010200_ID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_D0302_ID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_MANDANT$DAT REBUILD ONLINE;  
ALTER INDEX IDX_D0303_S5200_SID REBUILD ONLINE;  
ALTER INDEX IDX_D0303_SID REBUILD ONLINE;
```

Ggf. gibt es noch weitere INVALID-Objects, bitte prüfen Sie dieses anhand folgender Kontrollabfrage:

```
SELECT OBJECT_TYPE,OBJECT_NAME FROM USER_OBJECTS WHERE STATUS <>  
'VALID'
```

Es sollten nur mehrere Objekte als INVALID markiert sein, die folgenden Schritte stellen die Lauffähigkeit wieder her:

```
ALTER PROCEDURE DELETE_D0303 COMPILE;  
ALTER VIEW VIS_VD030301_ANZ COMPILE;  
ALTER VIEW VIS_VD0303 COMPILE;  
ALTER TRIGGER VIS_D0801_HIST COMPILE;
```



Diese beiden TRIGGER müssen nicht existieren und könnten einen SQL-Fehler verursachen: ORA-04080: Trigger 'VIS\_D0303\_LOG' ist nicht vorhanden. Dieser Fehler kann ignoriert werden:

```
ALTER TRIGGER VIS_D0303_LOG COMPILE;  
ALTER TRIGGER VIS_D0303_LOG_DEL COMPILE;
```

#### 4.1.2 Beispiel: Arbeitstabellen WKD\_AVV\*

Äquivalent zum verkleinern der Dokumententabelle kann auch jegliche andere Tabelle verkleinert werden. Dieses Beispiel ist für die Funktion DELETE\_WKD\_AVV().

Manuelle Verkleinerung:

Auszuführen als Eigentümer des Schemas, nicht als SYS oder SYSTEM! Die Ausführung aus BALVI iP SQL-Work ist möglich.

```
ALTER TABLE WKD_AVV01 ENABLE ROW MOVEMENT;  
ALTER TABLE WKD_AVV02 ENABLE ROW MOVEMENT;  
ALTER TABLE WKD_AVV03 ENABLE ROW MOVEMENT;  
ALTER TABLE WKD_AVV03 SHRINK SPACE;  
ALTER TABLE WKD_AVV02 SHRINK SPACE;  
ALTER TABLE WKD_AVV01 SHRINK SPACE;  
ALTER TABLE WKD_AVV01 MODIFY LOB (KO2) (SHRINK SPACE);  
ALTER TABLE WKD_AVV01 DISABLE ROW MOVEMENT;  
ALTER TABLE WKD_AVV02 DISABLE ROW MOVEMENT;  
ALTER TABLE WKD_AVV03 DISABLE ROW MOVEMENT;
```

## 4.2 Arbeitsschritte nach einer Tabellenverkleinerung

Nach einer Tabellenverkleinerung müssen alle Indizes neu aufgebaut und die Table Statistics aktualisiert werden.

Um sicherzustellen, dass das ROW MOVEMENT keine negativen Auswirkungen auf die Anwendung hat, sollten folgende Kontrollabfragen ausgeführt werden:

```
SELECT * FROM USER_OBJECTS WHERE STATUS <> 'VALID';
```

```
SELECT * FROM USER_INDEXES WHERE STATUS <> 'VALID';
```

Alle INVALID Objects müssen unbedingt repariert werden.

Die folgenden Beispiel-Skripte sind ebenfalls im Kommentar des beiliegenden Skripts „ora\_Functions\_DELETE\_LOG.sql“ enthalten. Es wird nicht empfohlen, den Inhalt des Dokuments per Copy&Paste in einen SQL-Editor einzufügen, sondern die Inhalte aus dem Skript zu kopieren, um Zeichensatz- und Formatierungsprobleme zu vermeiden.

### 4.2.1 Reparatur und verkleinern von Indizes

Die folgende PL/SQL-Beispiele sind auszuführen als Eigentümer des Schemas, nicht als SYS oder SYSTEM!

Die Ausführung aus BALVI iP (SQL-Work) ist möglich, wird jedoch nicht empfohlen, da die Ausgabe der Ergebnisse aus dem PL-SQL-Block nicht möglich ist! Bei der Ausführung in SQL-Work dürfen zudem keine "/" am Ende eines PL-SQL-Blocks angegeben werden, jeder SQL-Block muss einzeln ausgeführt werden.

Dieses Beispiel basiert ebenfalls auf dem Fall „WKD\_AVV“ zugeschnitten, daher sind beim „REBUILD INVALID AND SHRINK Indizes“ Sonderbehandlungen für alle Tabellen mit Namen WKD\_AVV% enthalten. Diese sollten für andere Konstellationen angepasst bzw. entfernt werden.

```

SET SERVEROUTPUT ON SIZE 1000000;
-- Rebuild INVALID and SHRINK Indizes -----
DECLARE
  sSQL          VARCHAR2(4000);
  sProt         VARCHAR2(4000);
BEGIN
  DBMS_OUTPUT.PUT_LINE( 'Rebuild not valid Indexes');
  FOR rec IN ( SELECT INDEX_NAME, TABLE_NAME FROM USER_INDEXES
              WHERE INDEX_TYPE IN ('NORMAL', 'FUNCTION-BASED NORMAL')
                AND (STATUS <> 'VALID' OR TABLE_NAME LIKE 'WKD_AVV%')
              ORDER BY 2,1
  ) LOOP
    sProt := rec.index_name || ' -> rebuild';
    sSQL := 'ALTER INDEX ' || rec.index_name || ' REBUILD';
    EXECUTE IMMEDIATE (sSQL);
    DBMS_OUTPUT.PUT_LINE( sProt );
    IF REC.TABLE_NAME LIKE 'WKD_AVV%' THEN
      sProt := rec.index_name || ' -> shrink';
      sSQL := 'ALTER INDEX ' || rec.index_name || ' SHRINK SPACE CASCADE';
      EXECUTE IMMEDIATE (sSQL);
      DBMS_OUTPUT.PUT_LINE(sProt);
    END IF;
  END LOOP;
END;
/
-- Rebuild INVALID and SHRINK Indizes -----

```

## 4.2.2 Compilieren aller ungültigen Objekte

Das PL/SQL\_Skript „Recompile INVALID Objects“ ist generisch und kann jederzeit verwendet werden. Trotzdem empfiehlt es sich nach der Ausführung zu prüfen, ob noch ungültige Objekte vorhanden sind.

```
-- Recompile VIEWS, TRIGGERS, FUNCTIONS, PACKAGES -----
DECLARE
  sSQL          VARCHAR2(4000);
  sProt         VARCHAR2(4000);
BEGIN
  FOR REC IN (SELECT OBJECT_TYPE, OBJECT_NAME FROM USER_OBJECTS
              WHERE STATUS <> 'VALID' and object_type NOT IN('PACKAGE BODY')
              ORDER BY
                -- Trigger am Schluß,
                -- vorher Views, davor Packages,
                -- Functions und Procedures
                CASE
                  WHEN OBJECT_TYPE='TRIGGER' THEN 3
                  WHEN OBJECT_TYPE='VIEW' THEN 2
                  ELSE 1
                END, OBJECT_TYPE
              )
  LOOP
    sProt := rec.OBJECT_TYPE || ' ' || rec.OBJECT_NAME || ' -> compile';
    sSQL := 'ALTER ' || rec.OBJECT_TYPE || ' ' || rec.OBJECT_NAME || ' COMPILE';
    EXECUTE IMMEDIATE (sSQL);
    DBMS_OUTPUT.PUT_LINE( sProt );
  END LOOP;
END;
/
-- Recompile VIEWS, TRIGGERS, FUNCTIONS, PACKAGES -----
```

## 4.2.3 Datenbank-Statistiken aktualisieren

Es wird empfohlen, nach Massendaten-Korrekturen ein Gather Statistics auszuführen. Sie haben die Möglichkeit, dieses über Schema-Statistiken oder als explizite Statistiken für die geänderten Objekte durchzuführen.

Beispiel: expliziter Statistiken für den die Tabelle „VIS\_D0303“: Finden Sie alle Indizes, welche für die Tabelle VIS\_D0303 gültig sind:

```
SELECT * FROM USER_INDEXES WHERE TABLE_NAME= 'VIS_D0303' AND
INDEX_TYPE <> 'LOB';
```

Für jeden gefundenen Index sollte ein GATHER\_INDEX\_STATS ausgeführt werden. Das Ergebnis könnte folgendermaßen aussehen:

```
BEGIN
  DBMS_STATS.GATHER_TABLE_STATS(user, 'VIS_D0303');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'PK_D0303_ID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_A0303_01_SID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_D0101_ID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_D010200_ID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_D0302_ID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_MANDANT$DAT');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_S5200_SID');
  DBMS_STATS.GATHER_INDEX_STATS(user, 'IDX_D0303_SID');
END;
```

Für die Verwendung von DBMS\_STATS konsultieren Sie bitte die aktuellste Oracle Dokumentation für Ihre DBMS-Version.