

# Information Lifecycle Management

**Dr. Frank Haney**  
**Consultant**  
**Jena**

## **Schlüsselworte:**

Information Lifecycle Management, ILM, Oracle Database, Partitionierung, Komprimierung, Archivierung, Heat Map, Automatic Data Optimization, Temporal Validity, Row Archival

## **Zum Gegenstand**

Geschäftsapplikationen sammeln und verarbeiten Daten. Dabei können enorme Mengen zusammenkommen. Die maximale Datenbankgröße beträgt immerhin 8 EB (64kB-Blöcke und Bigfile Tablespaces vorausgesetzt). Derartige Datenmengen ohne Abstriche ad infinitum verfügbar zu halten, kann Performance und Verfügbarkeit enorm beeinträchtigen, und ist letztlich auch ein Kostenfaktor. Es bedarf der klaren Vorstellung, dass Daten einem Lebenszyklus unterliegen. Der Vortrag soll das näher beleuchten und vor allem die von der Oracle Datenbank bereitgestellten Mittel vorstellen.

## **Was ist Information Lifecycle Management (ILM)?**

Zunächst geht es bei ILM um Anforderungen für die Speicherung von Daten. Die Daten haben einen bestimmten, sich verändernden Wert für die Geschäftstätigkeit des Unternehmens, das reicht von kritischer bis zu eher historischer Bedeutung. Dann muss auch klar sein, wann die Daten auf preisgünstigeren damit weniger performanten Speicher verschoben werden können. Das soll nach Möglichkeit automatisch, ohne menschliche Intervention in jedem einzelnen Fall erfolgen. Für diese Automatismen müssen Regeln definiert und implementiert werden. Entscheidend für diese Regeln sind Wichtigkeit, Wertigkeit und Kosten.

Voraussetzung ist die Einteilung der Daten in Klassen entsprechend den Anforderungen an das Information Lifecycle Management. Diese Unterscheidung von Datenklassen erfordert die Festlegung

- der Relevanz der Daten für das Geschäft,
- des Workflow der Daten innerhalb der Firma,
- der Sichtbarkeit und Manipulierbarkeit der Daten innerhalb des Lebenszyklus,
- des notwendigen Grades an Verfügbarkeit und Sicherheit,
- der erforderlichen Performance beim Zugriff sowie nicht zuletzt
- der Aufbewahrungsfristen aus geschäftlicher und rechtlicher Sicht.

Als Konsequenz müssen für die Datenklassen dann definiert und implementiert werden:

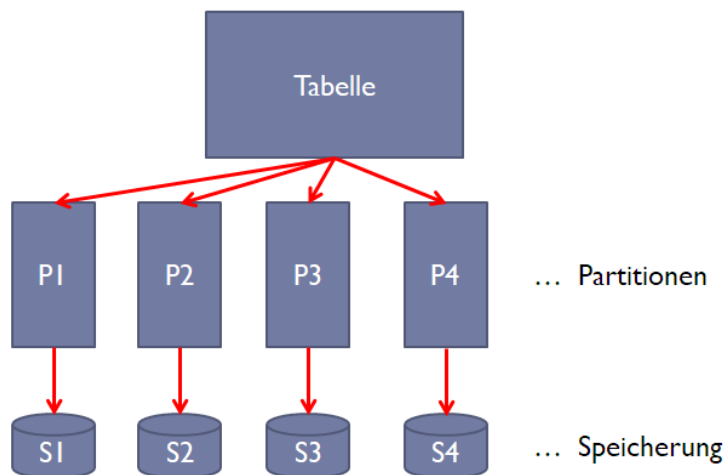
- die Speicherbereiche (logisch und physisch)
- der Lifecycle (Was wird wo, wie und wie lange gespeichert?)

- die Zuweisung des Lifecycle zu Segmenten (Tabellen und Partitionen)
- entsprechende Regeln (Policies), die den Lifecycle automatisch umzusetzen erlauben.

## Oracle Features für die Organisation des Information Lifecycle

An dieser Stelle sollen, ohne ins Detail zu gehen, die Features vorgestellt werden, die bei Oracle die Basis des ILM darstellen: Partitionierung, Komprimierung und Archivierung. Diese sind in der Regel leider nur durch die Lizenzierung zusätzlicher Optionen zur Enterprise Edition verfügbar.

**Partitionierung** (Partitioning Option): Große Segmente (Tabellen und Indizes) werden in kleinere Teile aufgeteilt (partitioniert). Das ist die Grundlage für die Zuweisung unterschiedlichen Speichers zu verschiedenen Datenklassen und hat darüber hinaus die weiteren Vorteile performanteren Zugriffs auf Teile der Daten und besserer Wartbarkeit großer Datenmengen. Partitionierung ist seit Oracle 8 als Range-Partitionierung verfügbar. Später kamen dann Hash- (8i) und List-Partitionierung (9i) dazu. Seitdem hat jedes Release auch neue Partitionierungsmöglichkeiten gebracht, z.B. Referenz- und Intervall-Partitionierung in 11.2 oder die Auto-List-Partitionierung in 12.2. Das Prinzip der Partitionierung kann man folgendermaßen veranschaulichen:



**Komprimierung** (Advanced Compression Option): Es geht hier darum, je nach Ihrer Verwendung Datensegmente (Tabellen und Indizes) oder ganze Bereiche (Tablespaces) in ihrem Speichervolumen zu reduzieren (komprimieren), um letztlich Platz und damit Kosten zu sparen. Je nach Komprimierungsart und -rate hat das natürlich Auswirkungen auch auf die Performance, weswegen man gut überlegen muss, welche Daten man wie komprimiert. Oracle stellt entsprechende Verfahren zur Verfügung. Eine Übersicht bietet die folgende Tabelle:

Method	Kompressionsrate	CPU-Overhead	Verwendung	Bemerkungen
Basic table compression	Hoch	Minimal	DSS	Keine
Advanced row compression	Hoch	Minimal	OLTP, DSS	Keine
Warehouse compression (Hybrid Columnar Compression)	Am höchsten	Höher	DSS	Kompressionsniveau und CPU Overhead hängen von der Einstellung LOW oder HIGH ab.
Archive compression (Hybrid Columnar Compression)	Am höchsten	Am höchsten	Archivierung	Kompressionsniveau und CPU Overhead hängen von der Einstellung LOW oder HIGH ab.

Es ist evident, wie sich im Zusammenwirken von Partitionierung und Komprimierung eine effektive ILM-Strategie entwickeln lässt.

**Flashback Data Archive** (Advanced Compression Option, ohne Komprimierung in allen Editionen frei): Dieses früher unter dem Namen *Total Recall* als eigene Option vermarktete Feature dient der möglichst revisionssicheren Langzeitarchivierung von Daten. Im Prinzip soll es möglich sein alle Transaktionen an ausgewählten Tabellen über einen zu definierenden Zeitraum nachverfolgen zu können. Die Daten stehen für das Reporting zu Verfügung, können aber nicht manipuliert werden und werden nach dem festgelegten Zeitraum automatisch gelöscht. Der Ablauf ist folgender:

- Mit dem Befehl `CREATE FLASHBACK ARCHIVE` wird in einem bestimmten Tablespace das Flashback Data Archive einschließlich Einstellungen zu Komprimierung, Quota und Aufbewahrungsdauer angelegt.
- Mit der Klausel `FLASHBACK ARCHIVE` von `CREATE` oder `ALTER TABLE` wird die Tabelle dem Archiv hinzugefügt. (Ab 12c können Tabellen in einer **Application** zusammengefasst und gemeinsam dem Flashback Data Archive hinzugefügt werden. Außerdem ist jetzt auch der User-Kontext der Änderungen nachvollziehbar.)
- Die historischen Daten stehen dann mit der von Flashback bekannten Klausel `AS OF TIMESTAMP` für Abfragen zur Verfügung. Ebenso ist Flashback Version Query (Klausel `VERSIONS BETWEEN`) möglich.

## Was bietet Oracle 12c an Neuem für das ILM?

Auf der Basis dieser etablierten Features eröffnet Oracle 12c nun ganz neue Möglichkeiten für das Information Lifecycle Management. Zu nennen sind folgende Features:

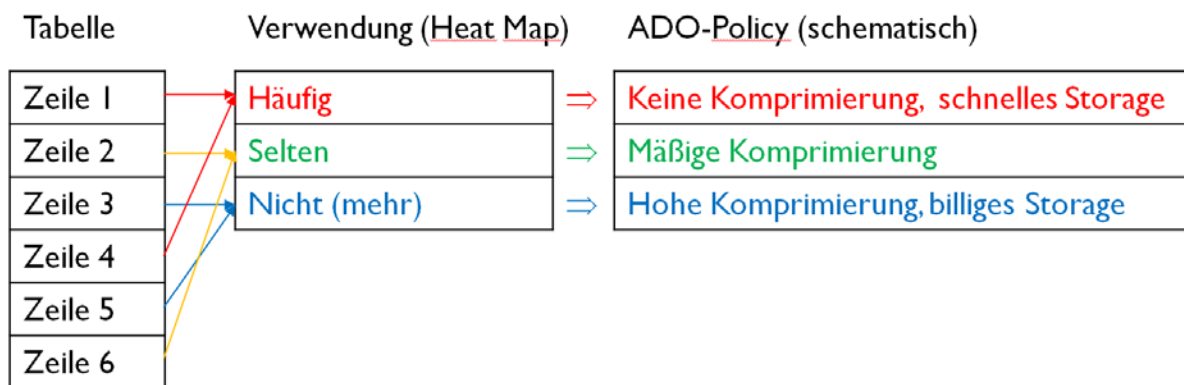
- Heat Map und Automatic Data Optimization (ADO)
- Temporal Validity
- In-Database Archiving

### Heat Map und Automatic Data Optimization (ADO)

Dieses Feature hat zur Grundlage, dass die Datenbank Aufzeichnungen über die Verwendung der Daten führt. In speziellen Katalogobjekten wird die Häufigkeit und der Letzte von Lese- und Schreibzugriffen erfaßt. Das erfolgt auf Segment-, aber auch auf Zeilenebene und unterscheidet Full Table Scans und Index Lookups. Eingestellt wird Heat Map auf Session oder Systemebene:

```
ALTER {SYSTEM|SESSION} SET HEAT MAP = {ON|OFF};
```

Auf der Basis der aufgezeichneten Informationen greift dann die ADO-Policy, die Partitionierung und Komprimierung verwendet. Schematisch muss man sich das so denken:



Die ADO-Policy wird mit CREATE oder ALTER TABLE definiert und erlaubt folgende Einstellungen (Auf Syntax-Beispiele wird hier verzichtet.):

- **Komprimierung:** Der Gebrauch oder Nichtgebrauch von Daten entscheidet ohne Eingriff des DBA über die Komprimierung der Daten bezüglich der Datenklasse.
- **Tiering:** Der Gebrauch oder Nichtgebrauch der Daten entscheidet ohne Eingriff des DBA über die Speicherung in einem Tablespace bezüglich der Datenklasse.
- **In-Memory** (ab 12.2): Der Gebrauch oder Nichtgebrauch der Daten entscheidet ohne Eingriff des DBA über die In-Memory-Attribute bezüglich der Datenklasse.

Allerdings ist für ADO entweder die *Advanced Compression* oder die *In-Memory* Option zu lizenzieren.

## Temporal Validity

Dieses Feature hat die Sichtbarkeit der Daten im Kontext der ILM-Policy im Fokus. Daten (Tabellenzeilen) werden nicht zwangsläufig bei Ablauf gelöscht, sind für die Applikation aber nur in einem definierten Zeitfenster sichtbar.



Organisiert wird das über zwei verborgene Spalten, die Beginn und Ende der Sichtbarkeit eines Datensatzes angeben. Diese werden beim Erzeugen der Tabelle mit der Klausel `PERIOD FOR` angelegt. Explizit können die Spalten auch im `INSERT` und `UPDATE` referenziert werden. In Abfragen kann dann die Klausel `PERIOD FOR` verwendet werden, um die zu einem bestimmten Zeitpunkt gültigen Daten zu sehen. Ansonsten werden nur die aktuell gültigen Datensätze angezeigt.

Mit der Prozedur `DBMS_FLASHBACK_ARCHIVE.ENABLE_AT_VALID_TIME` kann die Sichtbarkeit auf Session-Ebene eingestellt werden, ohne dass explizit die Klausel `PERIOD FOR` im Statement angegeben werden muß. Gültige Werte sind `CURRENT` für die aktuell gültigen Daten, `ASOF` für die zu einem bestimmten Zeitpunkt gültigen Daten und `ALL` für alle jemals gültigen Daten.

## In-Database Archiving

Das verfolgt ein ähnliches Ziel wie Temporal Validity. Zeilen werden entsprechend den ILM-Anforderungen vor der Applikation verborgen, aber nicht gelöscht. Das erfolgt im Gegensatz zu *Temporal Validity* nicht zeitbasiert, sondern auf der Basis einer diskreten Statusspalte `ORA_ARCHIVE_STATE`. Diese wird angelegt, wenn die Tabelle mit der Klausel `ROW ARCHIVAL` deklariert wird. Diese Spalte kann man dann als Partitionierungsschlüssel und zum Storage Tiering verwendet werden:

Tabelle			Speicherung	
Zeile 1	<code>ORA_ARCHIVE_STATE=0</code>	⇒ <b>aktuell</b> ⇒	Partition 0	Tier 0
Zeile 2	<code>ORA_ARCHIVE_STATE=1</code>	⇒ <b>veraltet</b> ⇒	Partition 1	Tier 1
Zeile 3	<code>ORA_ARCHIVE_STATE=0</code>	⇒ <b>aktuell</b> ⇒	Partition 0	Tier 0
Zeile 4	<code>ORA_ARCHIVE_STATE=2</code>	⇒ <b>obsolet</b> ⇒	Partition 2	Tier 2
Zeile 5	<code>ORA_ARCHIVE_STATE=1</code>	⇒ <b>veraltet</b> ⇒	Partition 1	Tier 1

Der Session-Parameter ROW ARCHIVAL VISIBILITY bestimmt, ob historische Zeilen sichtbar sind oder nicht.

ROW ARCHIVAL VISIBILITY=ACTIVE bedeutet, dass alle aktive Zeilen, gleichbedeutend mit ORA\_ARCHIVE\_STATE = 0 angezeigt werden. Bei ROW ARCHIVAL VISIBILITY=ALL bedeutet, werden alle Zeilen angezeigt, auch die mit ORA\_ARCHIVE\_STATE > 0.

### **Fazit**

Es wurde gezeigt, dass die Oracle Datenbank verschiedene Möglichkeiten zum Information Lifecycle Management Out of the box mitbringt. Diese werden durch neue Features weiter erweitert. Der Vorteil ist auf jeden Fall, daß die ILM-Policies von der Datenbank erzwungen werden und nicht extern implementiert werden müssen. Nachteil sind allerdings die in der Regel zusätzlich anfallenden Lizenzkosten. Der Anwender muss nun entscheiden, ob er sich diese leisten will bzw. kann. Löhnen wird sich das sicher nur im Rahmen einer umfassenden ILM-Strategie.

### **Kontaktadresse:**

Dr. Frank Haney  
Anna-Siemsen-Str. 5  
D-07745 Jena

Telefon: +49(0)3641-210224  
E-Mail: [info@haney.it](mailto:info@haney.it)  
Internet: <http://www.haney.it>