Large Objects

Modulvortrag Datenbankimplementierungstechniken

Sven Walter

July 9, 2014

Was soll das?

Large Objects sind Datentypen um umfangreiche Datenmengen zu speichern. [1]

- Mögliche Anwendungen
  - Speicherung von Dateien
    - müssen nicht im Dateisystem abgelegt werden
    - vereinfacht Datensicherung
    - feinere Granulierung der Rechte
    - z.B. Nutzerbilder von Webseiten
  - Auslagerung von Daten aus dem Arbeitsspeicher
    - Serialisierung komplexer Objekte
    - z.B. CAD-Anwendungen

Inhalt

Typen (Oracle)

BLOB *Binary Large Object*. Datentyp für alle Arten von Daten in binärem Format (Bilder, Videos).

CLOB *Character Large Object*. Datentyp für Zeichenketten in einheitlicher Kodierung (XML-Dateien).

NCLOB Äquivalent von CLOB für *National Character Set*.

BFILE *External Binary File*. Eine Datei, die außerhalb der Datenbank gespeichert wird, aber über das DBMS aufrufbar ist. Ist nicht beschreibbar.

Arten von Daten

- *Structured data*
  - logisch strukturiert
  - von DBMS interpretiert
- *Semi-structured data*
  - logisch strukturiert
  - nicht von DBMS interpretiert
- *Unstructured data*
  - kann logisch nicht weiter unterteilt werden
  - nicht von DBMS oder Anwendung interpretiert
- bei semisstruktirierten und unstrukturierten Daten, LOB interessant

Speicherung (Oracle) – Zellen

- Zelleninhalt
  - Zelleninhalt wird LOB Instance genannt
  - LOB Instanzen haben einen Wert und einen Locator
  - Wert ist eigentliches Objekt
  - Locator ist Referenz auf physikalischen Speicherort
- Zellenzustand
  - NULL
    - Zelle wurde erstellt
    - enthält weder Wert noch Locator
  - Empty
    - enthält Locator, aber keinen Wert
  - Populated
    - enthält Locator und Wert

Speicherung (Oracle) – Intern / Extern

- Internal
  - Speicherung im Tablespace
  - nur für *BLOB*/*CLOB*/*NCLOB* verfügbar
  - effizienter Zugriff
- External
  - Speicherung im Dateisystem
  - nur für *BFILE* verfügbar
  - kann nur gelesen werden

Speicherung (Oracle) – Inline / Out-of-Line

- Inline: Speicherung in der Zeile
- Out-of-Line: Speicherung außerhalb der Zeile
- verschiedene Kriterien
  - Konfiguration der Tabelle (default: Out-of-Line)
  - ab 3964 Bytes Out-of-Line (konfigurierbar)
  - keine Änderung bei neuer Länge des LOB

Speicherung (Oracle) – Chunks

- nur für Out-of-Line Speicherung relevant
- Chunk belegt einen oder mehrere Blöcke (aka Pages)
- Chunksize konfigurierbar
  - LOB belegt immer ein vielfaches der Größe
  - Einfluss auf Speicherverbrauch
  - Einfluss auf Lesegeschwindigkeit
  - Konfiguration für Tabelle und Datenbank
  - kann später nicht mehr geändert werden

Speicherung (Oracle) – Temporär / Persistent

- ein internes LOB kann temporär oder persistens sein
- temporär
  - existiert nur im Kontext der lokalen Anwendung
- persistent
  - existiert in einer Tabellenzeile
  - es gelten ACID Eigenschaften
- ein temporäres LOB wird persistent, wenn die Instanz in einer Zeile gespeichert wird

Besonderheiten (Oracle*)

- Einschränkungen
  - nicht als Primärschlüssel nutzbar
  - nicht nutzbar in ORDER BY, GROUP BY oder bei Aggregationen
  - in Queries mit DISTINCT oder UNIQUE nicht nutzbar
  - viele weitere Einschränkungen

Tabellen erstellen

- einfach Datentyp anpassen

Beispieltabelle

```
CREATE TABLE images
( image_id          NUMBER(6)
, content           BLOB
);
```

Zugriff mittels SQL (Oracle)

- einfaches Mapping der Typen
  - CLOB → CHAR, LONG oder VARCHAR2
  - BLOB → LONG RAW oder RAW

Beispielabfrage

```
INSERT INTO images (image_id, content)
    VALUES(42, 0x89504e470d0a1a...);
SELECT content FROM images WHERE image_id=42;
```

- Limits
  - SQL: 4000 Bytes mit SQL
  - PL/SQL: 32KB - 1B
- Alternativen
  - Oracle Call Interface (OCI)
  - Datenbanktreiber für Programmiersprache (z.B. JDBC)

Zugriff mittels API

- ziemlich alle APIs bieten LOB-Unterstützung
- meist Umwandlung in einen Datenstrom
- Beispiel JDBC

Beispielabfrage

```
Connection c = ...
ResultSet rs = c.prepareStatement("SELECT content "+
    "FROM images WHERE image_id = 2").executeQuery();
rs.next();
InputStream content = rs.getBinaryStream("content");
```

- Schreiben äqvivalent:
  - `stmt.setBinaryStream(int, InputStream)`

PL/SQL Funktionen

- Auswertung
  - READ: auslesen ab angegebener Stelle
  - SUBSTR: angeg. Teil auslesen
  - GETLENGTH: Länge des LOB
- Modifikation
  - APPEND: anhängen von Daten
  - ERASE: angeg. Abschnitt löschen
  - TRIM: LOB auf angeg. Länge kürzen
  - WRITE: Wert an eine angeg. Stelle schreiben
- weitere ... [1,Abs. 6]

Quellen

[1] Large Objects (Oracle Docs) http://docs.oracle.com/cd/B19306_01/appdev.102/b14249/toc.htm