

## Datenbanken II B: DBMS-Implementierung — 8. Übungsblatt: Segmente —

### Hausaufgaben

Geben Sie die Aufgaben dieses Abschnitts bis Mittwoch, 15.12.2021, 18<sup>00</sup>, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösungen in eine `.txt`-Datei bzw. `.sql`-Datei (kein PDF und schon gar kein Word).

- a) **(3 Punkte)** Finden Sie heraus, wie das „Extent Management“ für die Tablespaces unserer Datenbank funktioniert. Schreiben Sie eine Anfrage, die jeweils den Namen des Tablespaces ausgibt, und einen der drei Texte:

- `EXTENT MANAGEMENT DICTIONARY`
- `EXTENT MANAGEMENT LOCAL UNIFORM SIZE <Size>`, wobei Sie für `<Size>` die `INITIAL_EXTENT` Größe einsetzen (ich bin nicht sicher, ob das tatsächlich immer die deklarierte Größe der Extents ist, finde aber keinen anderen Zahlwert). Falls die Zahl groß genug ist, können Sie es gern in entsprechende Einheiten umrechnen, z.B. wäre der Default `1M`. Sie müssen diese Umrechnung aber nicht machen, Angaben in Bytes wären auch legal im `CREATE TABLESPACE`.
- `EXTENT MANAGEMENT LOCAL AUTOALLOCATE`

Sie finden die Information z.B. in der Tabelle `DBA_TABLESPACES`:

- Die Spalte `EXTENT_MANAGEMENT` kann die Werte `DICTIONARY` oder `LOCAL` enthalten.
- Falls `EXTENT_MANAGEMENT LOCAL` ist, kann `ALLOCATION_TYPE` die Werte `SYSTEM` (entspricht `AUTOALLOCATE`) und `UNIFORM` haben.

[[https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/DBA\\_TABLESPACES.html](https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/DBA_TABLESPACES.html)]

- b) **(3 Punkte)** Finden Sie die verschiedenen Extent-Größen heraus, die in unserer Datenbank vorkommen. Geben Sie die Größen in Blöcken, in Bytes und in Kilobytes aus. Sortieren Sie die Ausgabe nach der Größe der Extents. Sie finden die Daten in der Tabelle `DBA_EXTENTS`.

[[https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/DBA\\_EXTENTS.html](https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/DBA_EXTENTS.html)]

Naturgemäß ist die Tabelle sehr lang. Sie sollten daher nicht probierhalber

```
SELECT * FROM DBA_EXTENTS
```

eingeben. Sie können aber mit der `WHERE`-Bedingung `ROWNUM < 50` die Ausgabe auf die ersten 50 Zeilen begrenzen. Der Wert von `ROWNUM` wird nur erhöht, wenn eine Zeile ausgegeben wird. Wenn Sie die ersten Zeilen nach einem Sortier-Kriterium wünschen, müssen Sie das `ORDER BY` in einer Unteranfrage unter `FROM` machen, und in der äußeren Anfrage dann die Zeilenzahl begrenzen (Sonst würden erst 50 beliebige Zeilen ausgewählt und diese dann sortiert.). Seit Version 12c versteht Oracle auch das standardkonforme

```
FETCH FIRST 50 ROWS ONLY
```

am Ende der Anfrage. Das erspart Ihnen die Unteranfrage (es wird erst nach dem `ORDER BY` ausgewertet).

- c) **(3 Punkte)** Welches Segment bzw. welche Segmente in der Datenbank haben mehr als ein Extent? Geben Sie `OWNER`, `SEGMENT_NAME`, `SEGMENT_TYPE` und die Anzahl Extents aus. Beschränken Sie die Ausgabe vorsichtshalber auf 20 Zeilen.
- d) **(3 Punkte)** Legen Sie die folgende Tabelle an:

```
R(A: numeric(7), B: varchar(10), C: varchar(50))
```

Die Tabelle hat bewusst keinen Schlüssel. Alle Attribute können `NULL` sein.

- e) **(3 Punkte)** Schreiben Sie ein kurzes Programm in C++, das eine Datei `insert.sql` mit 25000 `INSERT`-Anweisungen der folgenden Form erzeugt:

```
INSERT INTO R VALUES(123456, NULL,  
                      'ABCDEFGHIJKLMNQRSTUWXYZ1234');
```

Da kein Schlüssel definiert ist, können Sie einfach 25000 Kopien dieser Anweisung in die Datei schreiben. Falls Sie große Schwierigkeiten mit C++ haben, dürfen Sie auch eine andere Programmiersprache verwenden. In C++ geht es so:

- Sie benötigen die Include-Dateien `<iostream>` und `<fstream>` („file stream“). Die Klasse für Ausgabeströme in Dateien ist `std::ofstream`.

[<http://www.cplusplus.com/reference/fstream/ofstream/>]

Etwas lesbarer ist dieses Kapitel:

[<https://stdcxx.apache.org/doc/stdlibug/30.html>]

Sie können ein Objekt der Klasse `ofstream` folgendermaßen anlegen:

```
std::ofstream file;
```

- Sie öffnen die Datei mit der Methode `open`:

```
[http://www.cplusplus.com/reference/fstream/ofstream/open/]
```

Das geht z.B. so:

```
file.open("insert.sql", std::ios::out);
```

Das zweite Argument erlaubt genauere Modus-Angaben, z.B. könnte man festlegen, dass die Ausgaben an den bisherigen Datei-Inhalt hinten angehängt werden sollen. Im Beispiel könnte man das zweite Argument auch weglassen, für Ausgabeströme ist der „out“-Modus der Default. Wenn Sie binäre Daten schreiben wollen (ohne Änderung eventueller Zeilenenden unter Windows) wäre folgender Modus passend:

```
file.open("inserts.sql", std::ios::out|std::ios::binary);
```

Wenn Sie den bisherigen Inhalt der Datei überschreiben wollen, verwenden Sie:

```
file.open("inserts.sql", std::ios::out|std::ios::trunc);
```

Allerdings ist auch `trunc` Default für Ausgabeströme. Eine gute Erklärung zu den Modi finden Sie auch hier:

```
[https://stdcxx.apache.org/doc/stdlibug/30-3.html]
```

- Natürlich sollten Sie testen, ob das Eröffnen der Datei erfolgreich war.

```
if(file.fail()) ...
```

Es geht auch kürzer:

```
if(!file) ...
```

Wie bei der `fail()`-Methode werden `failbit` und `badbit` im Stream-Objekt abgefragt.

- Sie schreiben in die Datei wie auf die Standard-Ausgabe:

```
file << "INSERT ...";
```

- Am Ende sollten Sie den Ausgabestrom schließen:

```
file.close();
```

Gepufferte Ausgaben werden möglicherweise erst dann tatsächlich in die Datei geschrieben. Sie sollten nochmals testen dass kein Fehler aufgetreten ist.

Die so erzeugte Datei können Sie im Adminer mit `IMPORT` ausführen.

- f) **(3 Punkte)** Stellen Sie fest, wie viele Extents welcher Größe Ihre Tabelle hat. Geben Sie für jede Größe von Extents (bitte in KByte und in Anzahl Blöcken) die Anzahl der Extents, die minimale `Extent_ID` und die maximale `EXTENT_ID` aus. Sortieren Sie die Ausgabe nach der Größe oder alternativ nach der minimalen `EXTENT_ID`.

Die Tabelle `DBA_EXTENTS` enthält die Extents von allen Segmenten. Es gibt deswegen keine Spalte `TABLE_NAME`, sondern nur `SEGMENT_NAME`. Im Falle von Segmenten für Tabellen steht darin aber der Tabellen-Name (`R`). Beachten Sie, dass natürlich auch andere DB-Nutzer eine Tabelle `R` erstellen. Sie müssen also den `OWNER` in der `WHERE`-Bedingung angeben. Wenn Sie wollen, können Sie zur Sicherheit noch verlangen, dass `SEGMENT_TYPE` den Wert „`TABLE`“ enthält.

Am besten löschen Sie die Tabelle `R` nach der Abfrage wieder: `DROP TABLE R.`

## Wiederholungsaufgaben

Die „Wiederholungsaufgaben“ brauchen Sie nicht abzugeben. Beschäftigen Sie sich aber bitte auch mit diesen Aufgaben. Notieren Sie sich Fragen, die Sie gerne in der Übung geklärt haben wollen.

- g) Wie würden Sie in einer mündlichen Prüfung auf folgende Fragen antworten?
- Was ist ein „Segment“ in Oracle? Welche verschiedenen Arten von Daten können in einem Segment gespeichert werden?
  - Während der Name „Segment“ spezifisch für Oracle ist, steht doch jedes DBMS vor der gleichen Aufgabe. Kann ein DBMS eine Funktion des Betriebssystems dafür nutzen? Nennen Sie Vor- und Nachteile.
  - Warum ist die Beziehung zwischen Tabellen und Daten-Segmenten in Oracle zwar normalerweise 1:1, aber im Allgemeinen n:m? Nennen Sie die Konstrukte, die Ausnahmen von der 1:1-Regel verursachen.
  - Wenn Sie einen Segment-Manager für Ihr DBMS programmieren müssten, was wäre sein Interface?
  - Was ist ein „Extent“ in Oracle? Warum fordert Oracle Speicher in Extents an, und nicht in einzelnen Blöcken?
  - Wie kann ein Segment-Manager freien Speicherplatz in einer Datei oder einem Tablespace finden, wenn ein neues Segment angelegt wird oder ein Segment wächst? Oracle hat zwei Lösungen:
    - „`EXTENT MANAGEMENT DICTIONARY`“ (die ältere Version), und
    - „`EXTENT MANAGEMENT LOCAL`“ (die neuere Version). Diese hat die Varianten „`UNIFORM SIZE`“ und „`AUTOALLOCATE`“.

Wie würden Sie das lösen?

## Für Interessierte

h) Werfen Sie einen Blick auf die folgenden Dokumente:

- MySQL Internals Manual  
[<https://dev.mysql.com/doc/internals/en/>]
- MySQL Coding Guidelines:  
[[https://dev.mysql.com/doc/dev/mysql-server/latest/\[PAGE\\_CODING\\_GUIDELINES.html](https://dev.mysql.com/doc/dev/mysql-server/latest/[PAGE_CODING_GUIDELINES.html)]  
This also refers to the Google C++ Style Guide:  
[<https://google.github.io/styleguide/cppguide.html>]
- Oracle Database Administrator's Guide: Managing Datafiles  
[[https://docs.oracle.com/cd/B28359\\_01/server.111/b28310/dfiles001.htm](https://docs.oracle.com/cd/B28359_01/server.111/b28310/dfiles001.htm)]  
This contains an explanation of absolute and relative file numbers that appear in the data dictionary tables.
- Rachid Belaid: Introduction to PostgreSQL Physical Storage.  
[<http://rachbelaid.com/introduction-to-postgres-physical-storage/>]
- Heikki Linnakangas: New Free Space Map and Visibility Map.  
[[https://wiki.postgresql.org/images/8/81/FSM\\_and\\_Visibility\\_Map.pdf](https://wiki.postgresql.org/images/8/81/FSM_and_Visibility_Map.pdf)]