Phân đoạn bảng trong DB2 9

Phân đoạn bảng trong DB2 9

Việc quản lý các cơ sở dữ liệu lớn được cải thiện

Richard Hewitt, Chuyên gia IT, Tư vấn DB2 LUW, IBM UK

Paul Read, Giám đốc giới thiệu sản phẩm của IBM, IBM

Tóm tắt: Hướng dẫn này được thiết kế để chứng tỏ các chức năng phân đoạn theo dải giá trị (Range Partitioning) của DB2® 9. Các học viên sẽ tiếp thu kinh nghiệm thực hành về cách sử dụng các đặc tính phân đoạn theo dải giá trị với các bảng của DB2. Phân đoạn bảng là cách sắp xếp dữ liệu theo hệ thống, trong đó dữ liệu bảng được chia ra trên nhiều đối tượng lưu trữ dữ liệu được gọi là các phân đoạn dữ liệu, hoặc các dải, theo các giá trị trong một hoặc nhiều cột bảng. Mỗi phân đoạn dữ liệu được lưu trữ riêng rẽ. Các đối tượng lưu trữ này có thể ở trong các vùng bảng khác nhau, trong cùng một vùng bảng hay là kết hợp của cả hai cách trên.

Trước khi bạn bắt đầu

Dữ liệu bảng được phân đoạn như đã chỉ rõ trong mệnh đề PARTITION BY của câu lệnh CREATE TABLE. Các cột được sử dụng trong định nghĩa này được gọi là các cột khóa phân đoạn bảng (tablepartitioning key).

Bạn có thể đọc thêm các mô tả về đặc tính này tại bài viết "Phân đoạn bảng trong DB2 9" (developerWorks, tháng Năm năm 2006).

Phân đoạn bảng cung cấp cho ta các ưu điểm sau:

- Dễ dàng cuộn vào (roll-in) và cuộn ra (roll-out) các dữ liệu bảng
- Dễ dàng hơn khi quản trị các bảng lớn

- Sắp xếp linh hoạt các chỉ mục
- Cải thiện hiệu suất cho các truy vấn theo phong cách nghiệp vụ thông minh

Về hướng dẫn này

Các bài tập sau đây cho phép bạn làm việc với các đặc tính phân đoạn bảng và chứng tỏ các khả năng: Dễ dàng cuộn vào (roll-in) và cuộn ra (roll-out) các dữ liệu bảng, Dễ dàng hơn khi quản trị các bảng lớn, sắp xếp linh hoạt các chỉ mục và cải thiện hiệu suất cho các truy vấn theo phong cách nghiệp vụ thông minh (BI).

Các bài tập đó đã được phát triển để trình diễn một hoặc nhiều tác vụ trong từng vấn đề nói trên.

Mục đích của bài hướng dẫn

Mục tiêu của bài hướng dẫn này là để khám phá các đặc tính và các lợi ích của việc phân đoạn theo dải giá trị trong DB2 9 trong các vấn đề sau:

- Tạo các bảng được phân đoạn theo dải giá trị
- Cuộn vào (roll-in) và cuộn ra (roll-out) các phân đoạn
- Quản lý bảng được phân đoạn
- Quản lý và xếp đặt các chỉ mục

Các yêu cầu cần có trước

Hướng dẫn này được viết cho các chuyên gia DB2 có kỹ năng và kinh nghiệm ở mức độ từ mới bắt đầu đến trung cấp. Bạn cần phải có những hiểu biết chung về việc sử dụng dòng lệnh của DB2, các công cụ quản trị của DB2 và có một số kiến thức làm việc với SQL.

Yêu cầu về hệ thống

Để chạy các ví dụ trong hướng dẫn này, bạn cần phải có:

- Máy chủ dữ liệu DB2 9
- Hệ điều hành Windows® 2000 của Microsoft® hoặc mới hơn và một tài khoản với các đặc quyền của quản trị viên, hoặc hệ điều hành Linux® (ấn bản có hiệu lực) truy cập của root.
- Bạn phải cài đặt môi trường chạy thi hành Java phiên bản 1.4.2 hoặc mới hơn tại máy tính của bạn.
- Tham khảo trang Yêu cầu về hệ thống của DB2 9 để đảm bảo rằng phần cứng của bạn thoả mãn các yêu cầu.

Sản phẩm DB2 9 Express C có sẵn từ liên kết ở trên. Đối với các bước cài đặt DB2, xin tham khảo mục "Hướng dẫn đánh giá DB2 XML" (developerWorks, tháng Sáu 2006). Trừ khi cấu hình DB2 bị thay đổi, DB2 sẽ tự động khởi chạy sau khi cài đặt.

Bạn hãy sử dụng các kịch bản lệnh mẫu và các dữ liệu ví dụ mẫu được cung cấp trong tệp tin partition.zip để giải thích các khái niệm trong hướng dẫn này. Giải nén nội dung của tệp tin vào một thư mục con có tên là script (C:\scripts hoặc home/userid/scripts). Thư mục này sau đây được gọi tắt là stmm_scripts trong suốt

hướng dẫn này. Hướng dẫn này giả định rằng bạn đã sử dụng các thư mục mặc định cho việc cài đặt DB2, và tất cả các thao tác DB2 được thực hiện với ID của quản trị viên cơ sở dữ liệu.

Tạo các bảng được phân đoạn

Bài thực hành này sẽ xem xét một số tùy chọn cho việc tạo bảng được phân đoạn, nạp dữ liệu vào các bảng được phân đoạn, và cách sử dụng lệnh describe để minh họa cho các dải giá trị trong bảng:

1. Bạn sẽ đăng nhập vào và thiết lập môi trường cơ sở cho tất cả các bài tập.
2. Bạn sẽ tạo một bảng được phân đoạn theo một số định dạng và nạp dữ liệu.
3. Bạn sẽ sử dụng các lệnh DB2 và SQL để xem lại các kết quả.
4. Việc này sẽ cho bạn một tổng quan về việc phân đoạn bảng theo dải giá trị trong DB2 9.

Đăng nhập và các lệnh cơ bản

Hình 1. Thiết lập cơ bản

```
[db2inst1@localhost /]$ cd /scripts
[db2inst1@localhost scripts]$ db2start
11/07/2006 10:56:03     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
[db2inst1@localhost scripts]$ db2 connect to sample

   Database Connection Information

 Database server        = DB2/LINUX 9.0.0
 SQL authorization ID   = DB2INST1
 Local database alias   = SAMPLE
```

1. Đăng nhập vào máy tính của bạn, trong hình 1, đã sử dụng db2inst1.
2. Mở một cửa sổ đầu cuối (terminal - với Linux) hoặc cửa sổ lệnh DB2 (với Windows).
3. Chuyển sang thư mục con của các kịch bản lệnh.

Liệt kê 1. Thay đổi thư mục

cd /scripts hay

cd c:\scripts

4.

5. Khởi chạy DB2 bằng cách sử dụng lệnh db2start và kết nối tới cơ sở dữ liệu SAMPLE.

Liệt kê 2. Lệnh StartDB2

db2start

db2 connect to SAMPLE

6.

Tạo phân đoạn bảng cơ bản

Phần này của bài hướng dẫn đề cập đến các cơ sở trong tạo và nạp các bảng phân đoạn. Bạn sẽ tạo ra các bảng theo một số định dạng, xác nhận việc tạo bảng, nạp dữ liệu và truy vấn bảng.

1. Tạo bảng LINEITEM với bốn dải giá trị bằng cách sử dụng ngôn ngữ định nghĩa dữ liệu (DDL) sau đây:

Liệt kê 3. Tạo bảng

```
CREATE TABLE LINEITEM

( l_orderkey             DECIMAL(10,0)
NOT NULL,

 l_partkey          INTEGER,

 l_suppkey          INTEGER,

 l_linenumber       INTEGER,

 l_quantity         DECIMAL(12,2),

 l_extendedprice    DECIMAL(12,2),

 l_discount         DECIMAL(12,2),

 l_tax              DECIMAL(12,2),
```

```
l_returnflag       CHAR(1),
l_linestatus       CHAR(1),
l_shipdate         DATE,
l_commitdate       DATE,
l_receiptdate      DATE,
l_shipinstruct     CHAR(25),
l_shipmode         CHAR(10),
l_comment          VARCHAR(44))
PARTITION BY RANGE(l_shipdate)
( STARTING '1/1/1992' ENDING
'30/06/1992',
 STARTING '1/7/1992' ENDING
'31/12/1992',
 STARTING '1/1/1993' ENDING
'30/6/1993',
 STARTING '1/7/1993' ENDING
'31/12/1993')
```

2. SQL để tạo bảng nằm trong tệp tin EX1-6.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 4. Chạy tệp tin EX1-6

db2 –vtf EX1-6.sql

3.

4. Sử dụng lệnh sau để minh họa cho các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM:

Liệt kê 5. Lệnh desricbe

db2 describe data partitions for table
LINEITEM

5.

Hình 2. Minh họa các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM

```
[db2inst1@localhost scripts]$ db2 describe data partitions for table LINEITEM

PartitionId Inclusive (y/n)                 Inclusive (y/n)
              Low Value                       High Value
----------- - ------------------------------ - ------------------------------
          0 Y '1992-01-01'                   Y '1992-06-30'
          1 Y '1992-07-01'                   Y '1992-12-31'
          2 Y '1993-01-01'                   Y '1993-06-30'
          3 Y '1993-07-01'                   Y '1993-12-31'

  4 record(s) selected.
```

Lưu ý: Bốn phân đoạn dữ liệu đã được tạo ra. Đây là các dải giá trị bao gồm cả giá trị ở đầu mút.

6. Nhập khẩu một số dữ liệu vào bảng LINEITEM. Lệnh nhập khẩu cho hoạt động này nằm trong tệp tin EX1-8.sql và có thể được chạy bằng lệnh sau đây:

Liệt kê 6. Nạp bảng có loại bỏ

db2 –vtf EX1-8.sql

7.

Hình 3. Nhập khẩu dữ liệu vào bảng LINEITEM

```
SQL3221W  ...Begin COMMIT WORK. Input Record Count = "1000".

SQL3222W  ...COMMIT of any database changes was successful.

SQL3110N  The utility has completed processing.  "1000" rows were read from
the input file.

SQL3221W  ...Begin COMMIT WORK. Input Record Count = "1000".

SQL3222W  ...COMMIT of any database changes was successful.

SQL3149N  "1000" rows were processed from the input file.  "271" rows were
successfully inserted into the table.  "729" rows were rejected.


Number of rows read         = 1000
Number of rows skipped      = 0
Number of rows inserted     = 271
Number of rows updated      = 0
Number of rows rejected     = 729
Number of rows committed    = 1000
```

Lưu ý: Có 729 hàng bị từ chối nhập khẩu, vì chúng không có giá trị l_shipdate trong các dải giá trị hiện tại được định nghĩa trong các phân đoạn dữ liệu của bảng LINEITEM.

8. Có sẵn một hàm vô hướng để hiển thị số hiệu phân đoạn dữ liệu (datapartitionnum) mà một hàng của bảng nằm trong phân đoạn đó. Hãy thi hành SQL ví dụ sau đây để xem kết quả đầu ra của hàm vô hướng này:

Liệt kê 7. Truy vấn – So khớp ngày tháng với phân đoạn

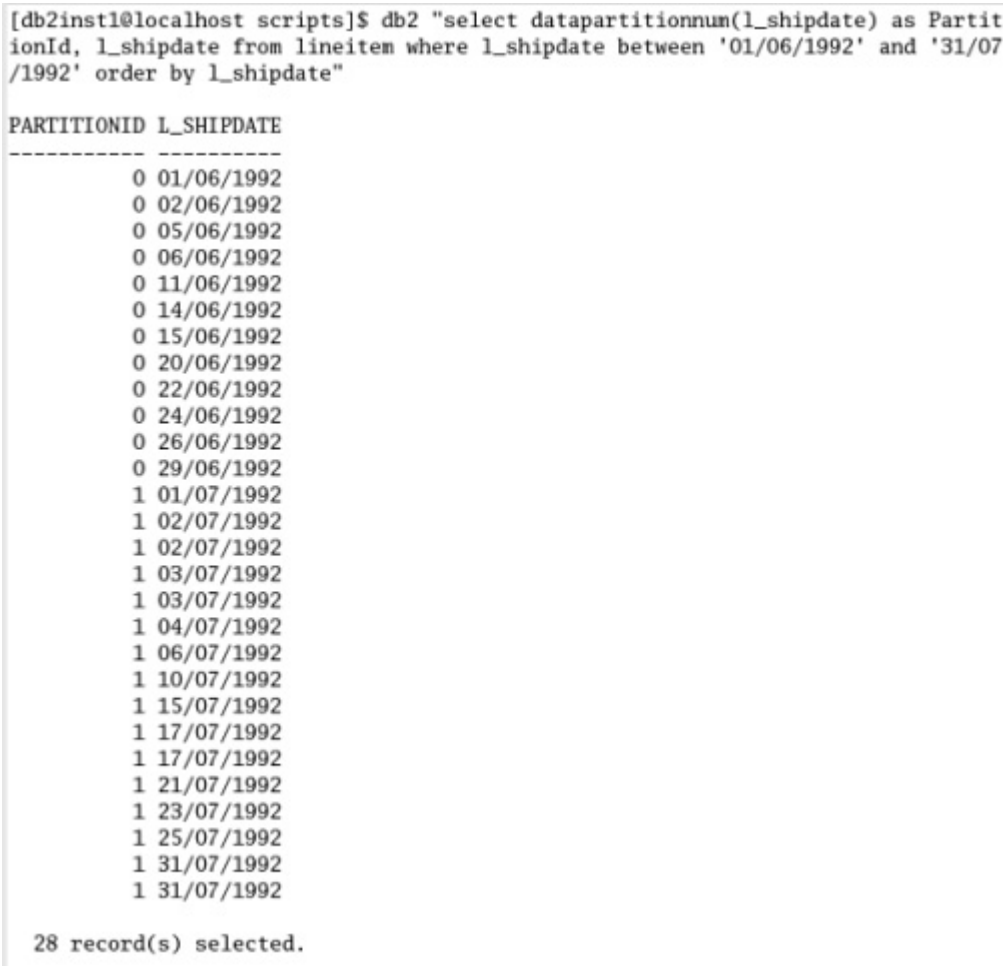
db2 “select datapartitionnum(l_shipdate)
as PartitionId, l_shipdate from lineitem

where l_shipdate

between ’01/06/1992’ and ‘31/07/1992’

order by l_shipdate”

Hình 4. Đầu ra của hàm vô hướng



```
[db2inst1@localhost scripts]$ db2 "select datapartitionnum(l_shipdate) as Partit
ionId, l_shipdate from lineitem where l_shipdate between '01/06/1992' and '31/07
/1992' order by l_shipdate"

PARTITIONID L_SHIPDATE
----------- ----------
          0 01/06/1992
          0 02/06/1992
          0 05/06/1992
          0 06/06/1992
          0 11/06/1992
          0 14/06/1992
          0 15/06/1992
          0 20/06/1992
          0 22/06/1992
          0 24/06/1992
          0 26/06/1992
          0 29/06/1992
          1 01/07/1992
          1 02/07/1992
          1 02/07/1992
          1 03/07/1992
          1 03/07/1992
          1 04/07/1992
          1 06/07/1992
          1 10/07/1992
          1 15/07/1992
          1 17/07/1992
          1 17/07/1992
          1 21/07/1992
          1 23/07/1992
          1 25/07/1992
          1 31/07/1992
          1 31/07/1992

  28 record(s) selected.
```

Lưu ý: Giá trị được trả lại bởi hàm vô hướng này (datapartitionnum) cũng chính là giá trị PartitionId được trả về từ lệnh describe describe. Vị từ range

được sử dụng trong mệnh đề between của câu lệnh này cắt qua đường biên giới giữa PartitionId 0 và PartitionId 1.

Phân đoạn bảng thu giữ tất cả các dải giá trị

1. Tạo một bảng LINEITEM mới với hai phân đoạn dữ liệu bổ sung, một để thu giữ các giá trị thấp hơn dải giá trị hiện thời và một để thu giữ các giá trị cao hơn dải giá trị hiện thời. Trước tiên bạn hủy bỏ bảng được phân đoạn LINEITEM hiện có bằng cách sử dụng lệnh sau:

   Liệt kê 8. Hủy bảng

   db2 drop TABLE LINEITEM

2. Sau đó bạn tạo phiên bản mới của bảng LINEITEM bằng cách sử dụng DDL sau:

   Liệt kê 9. Tạo bảng

   CREATE TABLE LINEITEM

   (l_orderkey DECIMAL(10,0)

```
NOT NULL,
 l_partkey          INTEGER,
 l_suppkey          INTEGER,
 l_linenumber       INTEGER,
 l_quantity         DECIMAL(12,2),
 l_extendedprice    DECIMAL(12,2),
 l_discount         DECIMAL(12,2),
 l_tax              DECIMAL(12,2),
 l_returnflag       CHAR(1),
 l_linestatus       CHAR(1),
 l_shipdate         DATE,
 l_commitdate       DATE,
 l_receiptdate      DATE,
 l_shipinstruct     CHAR(25),
 l_shipmode         CHAR(10),
 l_comment          VARCHAR(44))
 PARTITION BY RANGE(l_shipdate)
```

(STARTING MINVALUE,

STARTING '1/1/1992' ENDING '30/06/1992',

STARTING '1/7/1992' ENDING '31/12/1992',

STARTING '1/1/1993' ENDING '30/6/1993',

STARTING '1/7/1993' ENDING '31/12/1993',

ENDING MAXVALUE)

3. SQL để tạo bảng nằm trong tệp tin EX1-10.sql và có thể chạy bằng lệnh sau:

Listing 10. Run EX1-10

```
db2 –vtf EX1-10.sql
```

4.

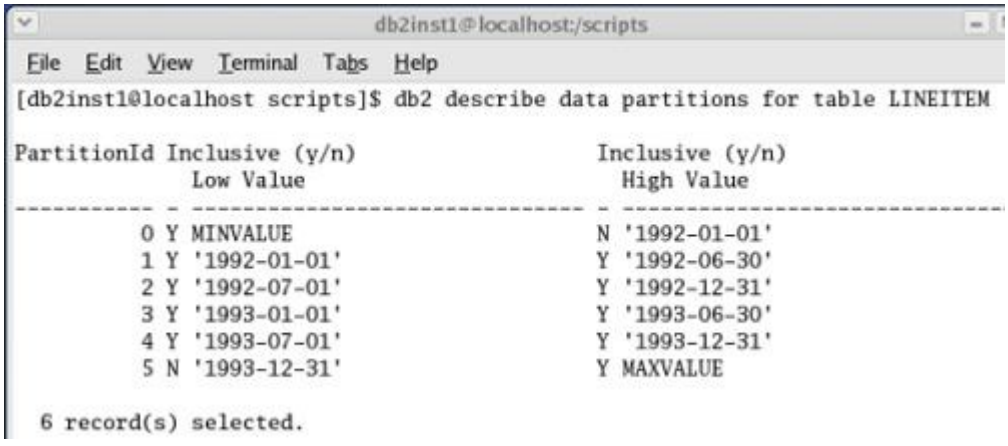
5. Sử dụng lệnh sau đây để minh họa cho các dải giá trị của các phân đoạn được tạo ra cho bảng LINEITEM.

Liệt kê 11. Lệnh describe

db2 describe data partitions for table
LINEITEM

6.

Hình 5. Minh họa các dải giá trị của các phân đoạn được tạo ra cho bảng LINEITEM



Lưu ý: Dải giá trị MINVALUE mới có giá trị ở đầu cao, bằng với giá trị bắt đầu của phân đoạn dữ liệu tiếp theo, nhưng nó không phải là một giá trị cao được tính gồm vào. Dải giá trị MAXVALUE có giá trị ở đầu thấp, bằng

với giá trị kết thúc của dải giá trị trước đó nhưng nó cũng không phải là một giá trị thấp được tính gồm vào. Điều này tạo ra một dải giá trị liên tục mà không có bất kỳ khoảng trống nào.

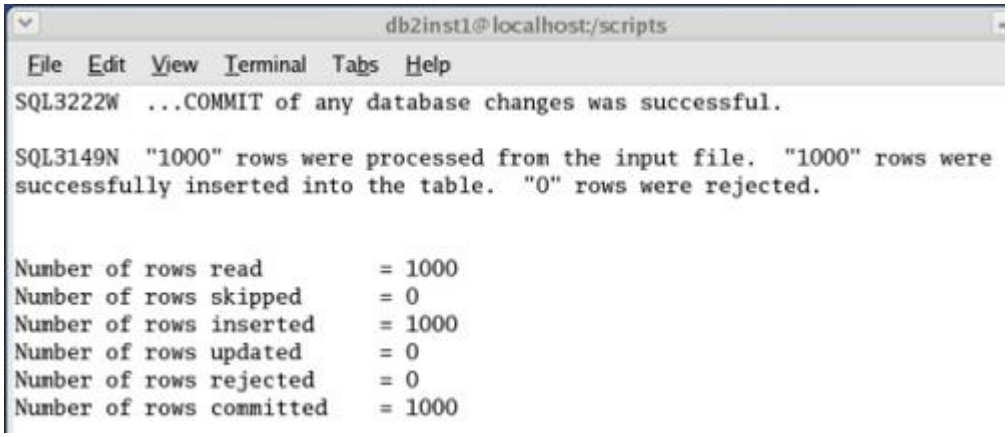
7. Nhập khẩu một số dữ liệu vào bảng LINEITEM. Lệnh nhập khẩu dành cho hoạt động này nằm trong tệp tin EX1-8.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 12. Nạp bảng và thu giữ tất cả các dải giá trị

db2 –vtf EX1-8.sql

8.

Hình 6. Nhập khẩu dữ liệu vào bảng LINEITEM



Phân đoạn bảng với các dải giá trị được sinh ra

1. Tạo một bảng LINEITEM mới với dải giá trị phân đoạn dữ liệu được sinh ra theo tháng từ 1 tháng Một năm 1992 đến ngày 31 tháng Mười hai năm 1998. Bạn cũng thêm dải giá trị minvalue và maxvalue để lưu giữ các hàng có giá trị l_shipdate bên ngoài dải giá trị trên. Trước tiên bạn hủy bỏ bảng được phân đoạn LINEITEM hiện có bằng cách sử dụng lệnh sau:

Liệt kê 13. Hủy bảng

```
db2 drop TABLE LINEITEM
```

2. Sau đó bạn tạo ra phiên bản mới của bảng LINEITEM bằng các sử dụng DDL sau:

Liệt kê 14. Tạo bảng

```
CREATE TABLE lineitem

(l_orderkey            DECIMAL(10,0)
NOT NULL,

 l_cpartkey      INTEGER,
```

```
l_suppkey       INTEGER,
l_linenumber    INTEGER,
l_quantity      DECIMAL(12,2),
l_extendedprice DECIMAL(12,2),
l_discount      DECIMAL(12,2),
l_tax           DECIMAL(12,2),
l_returnflag    CHAR(1),
l_linestatus    CHAR(1),
l_shipdate      DATE,
l_commitdate    DATE,
l_receiptdate   DATE,
l_shipinstruct  CHAR(25),
l_shipmode      CHAR(10),
l_comment       VARCHAR(44))
PARTITION BY RANGE(l_shipdate)
(STARTING MINVALUE,
STARTING '1/1/1992' ENDING
```

'31/12/1998'

EVERY 1 MONTH,

ENDING MAXVALUE);

3. SQL để tạo bảng nằm trong tệp tin EX1-13.sql và có thể chạy bằng lệnh sau:

Liệt kê 15. Chạy tệp tin EX1-13

db2 –vtf EX1-13.sql

4.

5. Sử dụng lệnh sau để minh họa cho các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM:

Liệt kê 16. Lệnh describe

db2 describe data partitions for table

LINEITEM

6.

Hình 7. Minh họa cho các dải giá trị của các phân đoạn được tạo ra cho bảng LINEITEM

```
[db2inst1@localhost scripts]$ db2 describe data partitions for table LINEITEM

PartitionId Inclusive (y/n)                 Inclusive (y/n)
              Low Value                       High Value
----------- - ------------------------------ - ------------------------------
          0 Y MINVALUE                       N '1992-01-01'
          1 Y '1992-01-01'                   N '1992-02-01'
          2 Y '1992-02-01'                   N '1992-03-01'
          3 Y '1992-03-01'                   N '1992-04-01'
          4 Y '1992-04-01'                   N '1992-05-01'
          5 Y '1992-05-01'                   N '1992-06-01'
          6 Y '1992-06-01'                   N '1992-07-01'
          7 Y '1992-07-01'                   N '1992-08-01'
          8 Y '1992-08-01'                   N '1992-09-01'
          9 Y '1992-09-01'                   N '1992-10-01'
```

Lưu ý: Có 86 phân đoạn dữ liệu được tạo ra. Các giá trị ở đầu cao của các dải giá trị này không được tính gồm vào vì chúng sẽ chồng lên giá trị ở đầu thấp của phân đoạn dữ liệu tiếp sau.

7. Nhập khẩu một số dữ liệu vào bảng LINEITEM. Lệnh nhập khẩu dành cho hoạt động này nằm trong tập tin EX1-8.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 17. Nạp bảng với các dải giá trị được sinh ra

db2 –vtf EX1-8.sql

8.

Hình 8. Nhập khẩu dữ liệu vào bảng LINEITEM

```
SQL3149N  "1000" rows were processed from the input file.  "1000" rows were
successfully inserted into the table.  "0" rows were rejected.


Number of rows read         = 1000
Number of rows skipped      = 0
Number of rows inserted     = 1000
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 1000
```

9. Sử dụng lệnh SQL sau để kiểm tra số đếm các hàng trong mỗi phân đoạn dữ liệu của bảng LINEITEM:

Liệt kê 18. Truy vấn dữ liệu

db2 “select year(l_shipdate) as year,
month(l_shipdate) as month,

count(*) as count from lineitem

group by year(l_shipdate),

month(l_shipdate)

order by 1, 2”

10.

Sử dụng SQL sau để kiểm tra số đếm các hàng trong mỗi phân đoạn dữ liệu của bảng LINEITEM

Liệt kê 19. Kịch bản lệnh truy vấn dữ liệu

db2 –vtf EX1-16.sql

11.

Hình 9. Kiểm tra các số đếm các hàng

```
1998           2           12
1998           3           13
1998           4            9
1998           5           12
1998           6           21
1998           7           16
1998           8           11
1998           9            2
1998          10            8
1998          11            2

82 record(s) selected.
```

Lưu ý: Có 82 trong số 86 dải giá trị chứa một hay nhiều hàng sau hoạt động nạp dữ liệu.

Xếp đặt các bảng được phân đoạn

Bài thực hành này sẽ xem xét một số tùy chọn cho việc xếp đặt các bảng được phân đoạn và cách sử dụng lệnh describe để minh họa cho các dải giá trị và cách xếp đặt bên trong bảng:

1. Bạn sẽ tạo các vùng bảng mới cho hoạt động xếp đặt dữ liệu.
2. Bạn sẽ tạo một bảng được phân đoạn theo một số định dạng.
3. Bạn sẽ sử dụng các lệnh db2 và SQL để xem lại kết quả.

Thiết lập môi trường cơ sở

1. Sử dụng lệnh describe data partitions (mô tả phân đoạn dữ liệu) với tùy chọn show detail (hiển thị chi tiết) để hiển thị sự xếp đặt các phân đoạn vào vùng bảng.

   Liệt kê 20. Tạo bảng

   db2 describe data partitions for table
   LINEITEM show detail

2.

   Hình 10. Xếp đặt các phân đoạn vào vùng bảng

```
PartitionId PartitionName                  TableSpId   PartObjId   LongTblSpId
AccessMode

  Status
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 PART0                                  3           5           3
F
          1 PART1                                  3           6           3
F
          2 PART2                                  3           7           3
F
          3 PART3                                  3           8           3
F
```

Lưu ý: Cột TableSpID cho biết số ID của vùng bảng chứa phân đoạn đó. Trong ví dụ này giá trị của TableSpID là '3 '.

3. Sử dụng lệnh list tablespaces (liệt kê vùng bảng) để xác định vùng bảng kết hợp với TableSpId.

Liệt kê 21. Lệnh describe

db2 list tablespaces

4.

Hình 11. Xác định vùng bảng

```
         Tablespaces for Current Database

Tablespace ID                        = 0
Name                                 = SYSCATSPACE
Type                                 = Database managed space
Contents                             = All permanent data. Regular table space.
State                                = 0x0000
  Detailed explanation:
    Normal

Tablespace ID                        = 1
Name                                 = TEMPSPACE1
Type                                 = System managed space
Contents                             = System Temporary data
State                                = 0x0000
  Detailed explanation:
    Normal

Tablespace ID                        = 2
Name                                 = USERSPACE1
Type                                 = Database managed space
Contents                             = All permanent data. Large table space.
State                                = 0x0000
  Detailed explanation:
    Normal
```

Lưu ý: Cột TableSpID tương ứng với giá trị '2' là USERSPACE1 hay là vùng bảng mặc định.

5. Bây giờ có 5 vùng bảng sẽ được tạo ra để minh họa cho các tùy chọn xếp đặt khác nhau. Sử dụng các lệnh sau đây:

Liệt kê 22. Lệnh describe

```
db2 create tablespace dms_d1 managed
by database using (file 'c:\ts1' 10000);

db2 create tablespace dms_d2 managed
```

by database using (file 'c:\ts2' 10000);

db2 create tablespace dms_d3 managed
by database using (file 'c:\ts3' 10000);

db2 create tablespace dms_d4 managed
by database using (file 'c:\ts4' 10000);

db2 create tablespace dms_i1 managed
by database using (file 'c:\ts5' 10000);

6.

SQL để tạo các vùng bảng nằm trong tệp tin EX2-3.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 23. Kịch bản lệnh truy vấn dữ liệu

db2 –vtf EX2-3.sql

7.

8. Tạo một bảng LINEITEM mới với một tập hợp các phân đoạn được sinh ra và đặt trong các vùng bảng dms_d1 và dms_d2. Trước tiên, hãy hủy bỏ bảng được phân đoạn LINEITEM hiện có bằng cách sử dụng lệnh sau:

Liệt kê 24. Huỷ bảng

```
db2 drop TABLE LINEITEM
```

9.

Sau đó, bạn tạo ra phiên bản mới của bảng LINEITEM bằng cách sử dụng DDL sau:

Liệt kê 25. Tạo bảng

```
CREATE TABLE LINEITEM

(l_orderkey      DECIMAL(10,0) NOT
NULL,

 l_partkey       INTEGER,

 l_suppkey           INTEGER,

 l_linenumber    INTEGER,

 l_quantity      DECIMAL(12,2),
```

```
 l_extendedprice DECIMAL(12,2),
 l_discount      DECIMAL(12,2),
 l_tax           DECIMAL(12,2),
 l_returnflag    CHAR(1),
 l_linestatus    CHAR(1),
 l_shipdate      DATE,
 l_commitdate    DATE,
 l_receiptdate       DATE,
 l_shipinstruct  CHAR(25),
 l_shipmode      CHAR(10),
 l_comment       VARCHAR(44))
 IN DMS_D1, DMS_D2
 PARTITION BY RANGE(l_shipdate)
(STARTING MINVALUE,
 STARTING '1/1/1992'
     ENDING '31/12/1998' EVERY 1
MONTH,
```

ENDING MAXVALUE);

10.

SQL để tạo bảng nằm trong tệp tin EX2-4.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 26. Chạy tệp tin EX2-4

db2 –vtf EX2-4.sql

11.

12. Sử dụng lệnh sau để minh họa cho các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM:

Liệt kê 27. Lệnh Describe

db2 describe data partitions for table
LINEITEM show detail

13.

Hình 12. Minh họa cho các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM

```
[db2inst1@localhost scripts]$ db2 describe data partitions for table LINEITEM sh
ow detail

PartitionId Inclusive (y/n)                 Inclusive (y/n)
              Low Value                       High Value
----------- - ------------------------------ - ------------------------------
          0 Y MINVALUE                       N '1992-01-01'
          1 Y '1992-01-01'                   N '1992-02-01'
          2 Y '1992-02-01'                   N '1992-03-01'
          3 Y '1992-03-01'                   N '1992-04-01'
          4 Y '1992-04-01'                   N '1992-05-01'
          5 Y '1992-05-01'                   N '1992-06-01'
          6 Y '1992-06-01'                   N '1992-07-01'
          7 Y '1992-07-01'                   N '1992-08-01'
          8 Y '1992-08-01'                   N '1992-09-01'
          9 Y '1992-09-01'                   N '1992-10-01'
         10 Y '1992-10-01'                   N '1992-11-01'
```

Hình 13. Các phân đoạn

```
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 PART0                                    4           4           4
F
          1 PART1                                    5           4           5
F
          2 PART2                                    4           5           4
F
          3 PART3                                    5           5           5
F
          4 PART4                                    4           6           4
F
          5 PART5                                    5           6           5
F
          6 PART6                                    4           7           4
F
          7 PART7                                    5           7           5
F
          8 PART8                                    4           8           4
F
```

Lưu ý: Cột TableSpID cho biết số ID của vùng bảng có chứa phân đoạn đó. Trong ví dụ này, giá trị của TableSpID hoặc là 4 (tương ứng với DMS_D1) hoặc là 5 (tương ứng với DMS_D2). Với ví dụ này các phân đoạn được sinh ra đã được phân bổ xoay vòng lần lượt (round robin) vào các vùng bảng đã chỉ định.

Xếp đặt tường minh các phân đoạn

1. Tạo một bảng LINEITEM mới với bốn phân đoạn dữ liệu, mỗi phân đoạn dữ liệu được đặt trong một vùng bảng một cách tường minh. Trước tiên, hãy hủy bỏ bảng được phân đoạn LINEITEM hiện có bằng cách sử dụng lệnh sau:

   Liệt kê 28. Huỷ bảng

   db2 drop TABLE LINEITEM

2.

   Sau đó bạn tạo phiên bản mới của bảng LINEITEM bằng cách sử dụng DDL sau đây:

   Liệt kê 29. Tạo bảng

```
CREATE TABLE LINEITEM

 (l_orderkey          DECIMAL(10,0)
NOT NULL,

 l_partkey           INTEGER,

 l_suppkey           INTEGER,

 l_linenumber        INTEGER,

 l_quantity          DECIMAL(12,2),

 l_extendedprice     DECIMAL(12,2),

 l_discount          DECIMAL(12,2),

 l_tax               DECIMAL(12,2),

 l_returnflag        CHAR(1),

 l_linestatus        CHAR(1),

 l_shipdate          DATE,

 l_commitdate        DATE,

 l_receiptdate       DATE,

 l_shipinstruct      CHAR(25),
```

```
 l_shipmode      CHAR(10),
 l_comment       VARCHAR(44))
PARTITION BY RANGE(l_shipdate)
( STARTING MINVALUE IN
DMS_D1,
  STARTING '1/1/1992' ENDING
'31/12/1992' IN DMS_D2,
  STARTING '1/1/1993' ENDING
'31/12/1993' IN DMS_D3,
  ENDING MAXVALUE IN DMS_D4
);
```

3.

SQL để tạo bảng nằm trong tệp tin EX2-6.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 30. Chạy tệp tin EX2-6

```
db2 –vtf EX2-6.sql
```

4.

5. Sử dụng lệnh sau đây để minh họa các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM:

Liệt kê 31. Lệnh Describe

```
db2 describe data partitions for table
LINEITEM show detail
```

6.

Hình 14. Minh họa các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM

```
PartitionId Inclusive (y/n)                  Inclusive (y/n)
             Low Value                         High Value
----------- - ------------------------------ - ------------------------------
          0 Y MINVALUE                       N '1992-01-01'
          1 Y '1992-01-01'                   Y '1992-12-31'
          2 Y '1993-01-01'                   Y '1993-12-31'
          3 N '1993-12-31'                   Y MAXVALUE

  4 record(s) selected.


PartitionId PartitionName                TableSpId   PartObjId   LongTblSpId
AccessMode

  Status
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 PART0                                  4         281           4
F
          1 PART1                                  5         281           5
F
          2 PART2                                  6           4           6
F
          3 PART3                                  7           4           7
F
```

Lưu ý: Trong ví dụ này, mỗi phân đoạn đã được đặt trong một TableSpID khác, tương ứng với vùng bảng đã được chỉ rõ trong tệp tin DDL tạo bảng.

7. Tạo một bảng LINEITEM mới với bốn phân đoạn dữ liệu, mỗi phân đoạn dữ liệu được đặt trong một vùng bảng một cách tường minh và chỉ mục được đặt trong vùng bảng DMS_I1. Tại bước này, bạn sẽ được giới thiệu khái niệm về cách đặt tên cho các phân đoạn của bạn, thay vì dùng tính năng sinh tên mặc định. Trước tiên, hãy hủy bỏ bảng được phân đoạn LINEITEM hiện có bằng cách sử dụng lệnh sau:

Liệt kê 32. Huỷ bảng

db2 drop TABLE LINEITEM

8.

Sau đó bạn tạo phiên bản mới của bảng LINEITEM bằng cách sử dụng DDL sau:

Liệt kê 33. Tạo bảng

```
CREATE TABLE LINEITEM

(l_orderkey           DECIMAL(10,0)
NOT NULL,

 l_partkey            INTEGER,

 l_suppkey            INTEGER,

 l_linenumber         INTEGER,

 l_quantity           DECIMAL(12,2),

 l_extendedprice      DECIMAL(12,2),

 l_discount           DECIMAL(12,2),

 l_tax                DECIMAL(12,2),
```

```
 l_returnflag        CHAR(1),
 l_linestatus        CHAR(1),
 l_shipdate          DATE,
 l_commitdate        DATE,
 l_receiptdate       DATE,
 l_shipinstruct      CHAR(25),
 l_shipmode          CHAR(10),
 l_comment           VARCHAR(44))
INDEX IN DMS_I1
PARTITION BY RANGE(l_shipdate)
( PART JAN1992 STARTING '1/1/1992'
ENDING '30/6/1992' IN DMS_D1,
  PART JULY1992 STARTING
'1/7/1992' ENDING '31/12/1992' IN
DMS_D2,
  PART JAN 1993 STARTING
'1/1/1993' ENDING '30/6/1993' IN
DMS_D3,
  PART JULY1993 STARTING
'1/7/1993' ENDING '31/12/1993' IN
```

DMS_D4);

9.

SQL để tạo bảng nằm trong tệp tin EX2-8.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 34. Chạy tệp tin EX2-8

db2 –vtf EX2-8.sql

10.

11. Tạo một chỉ mục trong bảng LINEITEM, và đặt nó vào vùng bảng DMS_I1. Sử dụng SQL sau đây:

Liệt kê 35. Chỉ mục

db2 “create index I_LINEITEM on
LINEITEM(L_SHIPDATE) in DMS_I1”

12.

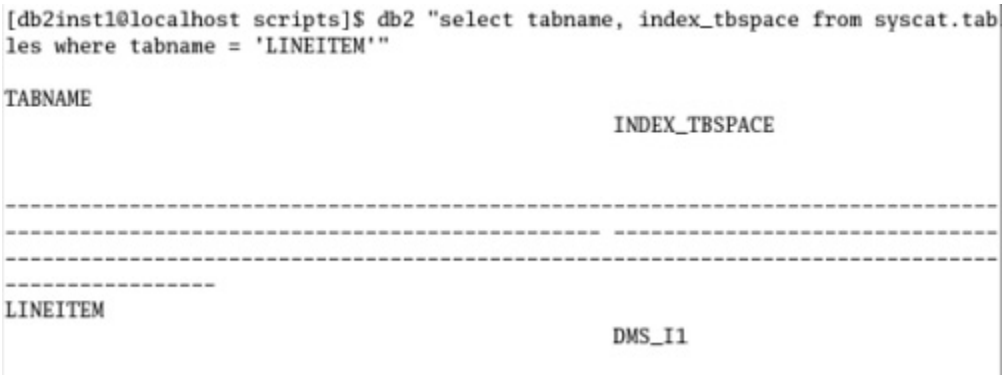
13. Sử dụng SQL sau để kiểm tra vị trí của các chỉ mục kết hợp với bảng này:

Liệt kê 36. Lệnh Describe

```
db2 “select tabname, index_tbspace from
syscat.tables where tabname =
‘LINEITEM’”
```

14.

Hình 15. Kiểm tra vị trí của các chỉ mục



```
[db2inst1@localhost scripts]$ db2 "select tabname, index_tbspace from syscat.tab
les where tabname = 'LINEITEM'"

TABNAME
                                                  INDEX_TBSPACE


------------------------------------------------------------------------------
------------------------------------------- ------------------------------
------------------------------------------------------------------------------
------------------
LINEITEM
                                                  DMS_I1
```

Lưu ý: Vùng chỉ mục là DMS_I1. Nếu không có vùng bảng nào được gán cho một bảng được phân đoạn, thì theo mặc định các chỉ mục sẽ nằm tại vùng bảng gắn theo đầu tiên. Một cách làm thực tế tốt là định nghĩa vùng bảng này trong lệnh CREATE TABLE. Tuy nhiên, cho dù bạn có xác định rõ vùng bảng cho chỉ mục hay không trong câu lệnh tạo bảng, thì nó cũng sẽ không hạn chế nơi mà bạn có thể đặt các chỉ mục trong tương lai. Bạn có

thể xác định tường minh vùng bảng cho chỉ mục trong chính bản thân câu lệnh INDEX CREATE. Các chỉ mục khác nhau trên cùng một bảng được phân đoạn có thể được đặt trong các vùng bảng khác nhau.

Quản lý các bảng được phân đoạn

Bài thực hành này sẽ xem xét cách mà bạn có thể quản lý và thao tác các bảng được phân đoạn:

1. Bạn sẽ thêm và xoá các phân đoạn.
2. Bạn sẽ thực hiện cuộn vào (roll-in) và cuộn ra (roll-out) các phân đoạn.
3. Bạn sẽ sử dụng các lệnh DB2 và SQL trong quá trình hoạt động để xem sự tiến triển.

Thêm một phân đoạn mới bằng cách sử dụng một bảng hiện có

1. Nhập khẩu một số dữ liệu vào bảng LINEITEM. SQL để tạo các vùng bảng nằm trong tệp tin EX3-1.sql và có thể chạy bằng lệnh sau đây:

   Liệt kê 37. Nhập khẩu dữ liệu

   db2 –vtf EX3-1.sql

2.

   Hình 16. Nhập khẩu dữ liệu vào bảng LINEITEM

```
SQL3221W  ...Begin COMMIT WORK. Input Record Count = "271".

SQL3222W  ...COMMIT of any database changes was successful.

SQL3149N  "271" rows were processed from the input file.  "271" rows were
successfully inserted into the table.  "0" rows were rejected.


Number of rows read         = 271
Number of rows skipped      = 0
Number of rows inserted     = 271
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 271
```

Lưu ý: Số đếm của các bản ghi trong bảng được phân đoạn, và những bản ghi mà bạn sẽ gắn vào và sẽ tháo ra là rất quan trọng để minh họa khi nào dữ liệu trở nên sẵn sàng trong cơ sở dữ liệu.

3. Tạo một bảng mới có tên là NP_LINEITEM. Kịch bản lệnh EX3-2.sql tạo ra một bảng mới có tên là NP_LINEITEM với 87 hàng:

Liệt kê 38. Phân đoạn mới

db2 –vtf EX3-2.sql

4.

Hình 17. Nhập khẩu dữ liệu vào bảng NP_LINEITEM

```
SQL3221W  ...Begin COMMIT WORK. Input Record Count = "87".

SQL3222W  ...COMMIT of any database changes was successful.

SQL3149N  "87" rows were processed from the input file.  "87" rows were
successfully inserted into the table.  "0" rows were rejected.


Number of rows read         = 87
Number of rows skipped      = 0
Number of rows inserted     = 87
Number of rows updated      = 0
Number of rows rejected     = 0
Number of rows committed    = 87
```

5. Sử dụng lệnh sau để minh họa cho các dải giá trị của các phân đoạn đã được tạo ra cho bảng LINEITEM:

Liệt kê 39. Lệnh Describe

db2 describe data partitions for table
LINEITEM show detail

6.

Hình 18. Các phân đoạn cho bảng LINEITEM

```
PartitionId PartitionName                   TableSpId   PartObjId   LongTblSpId
AccessMode

  Status
----------- ------------------------------- ----------- ----------- -----------
- ------
          0 JAN1992                                   4         282           4
F
          1 JULY1992                                  5         282           5
F
          2 JAN1993                                   6           4           6
F
          3 JULY1993                                  7           4           7
F

  4 record(s) selected.
```

Lưu ý: Bốn phân đoạn dữ liệu hiện đang tồn tại trong bảng LINEITEM.

7. Sử dụng câu lệnh Alter để gắn thêm (roll-in) một phân đoạn mới vào bảng LINEITEM hiện tại.

Liệt kê 40. Phân đoạn mới

```
ALTER TABLE LINEITEM ATTACH
PARTITION JAN1994

STARTING '1/1/1994' ENDING
'30/6/1994'

FROM NP_LINEITEM
```

8.

Kịch bản lệnh EX3-4 có thể chạy bằng lệnh sau đây:

Liệt kê 41. Phân đoạn mới

db2 –vtf EX3-4.sql

9.

Hình 19. Kịch bản lệnh EX3-4

```
[db2inst1@localhost scripts]$ db2 -vtf EX3-4.sql
alter table lineitem attach partition JAN1994 STARTING '1/1/1994' ENDING '30/6/1
994' from np_lineitem
SQL3601W  The statement caused one or more tables to automatically be placed
in the Set Integrity Pending state.  SQLSTATE=01586
```

Lưu ý: Bảng LINEITEM được đặt ở trạng thái SET INTEGRITY PENDING.

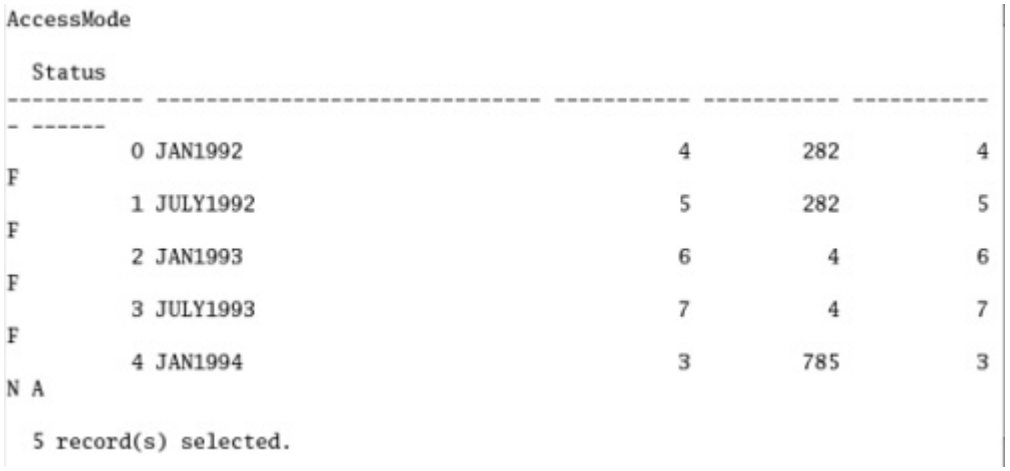
10. Sử dụng lệnh describe data partitions (mô tả các phân đoạn dữ liệu) để minh họa cho các dải giá trị của các phân đoạn trong bảng LINEITEM sau khi gắn thêm phân đoạn:

Liệt kê 42. Lệnh Describe

db2 describe data partitions for table
LINEITEM show detail

11.

Hình 20. Các phân đoạn cho bảng LINEITEM



```
AccessMode

  Status
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 JAN1992                                  4         282           4
F
          1 JULY1992                                 5         282           5
F
          2 JAN1993                                  6           4           6
F
          3 JULY1993                                 7           4           7
F
          4 JAN1994                                  3         785           3
N A

  5 record(s) selected.
```

Lưu ý: Phân đoạn dữ liệu mới PartitionId 4 (JAN1994) bây giờ được gắn thêm vào bảng LINEITEM. Tuy nhiên, giá trị của AccessMode cho phân đoạn vừa được gắn vào là 'N' và giá trị của Status là 'A'. Các giá trị có thể của AccessMode là:

- D = Không di chuyển dữ liệu
- F = Truy cập đầy đủ
- N = Không có quyền truy cập

- R = Truy cập chỉ đọc (Read-only)

Các giá trị có thể của Status là:

- A = Phân đoạn dữ liệu vừa được gắn vào
- D = Phân đoạn dữ liệu được tháo ra
- I = Phân đoạn dữ liệu đã tháo ra mà mục tương ứng của nó trong danh mục chỉ được duy trì trong quá trình dọn sạch các chỉ mục không đồng bộ; các hàng với giá trị STATUS là "I" được gỡ bỏ khi tất cả các bản ghi chỉ mục tham chiếu đến phân đoạn đã tháo ra đã bị xóa.
- Empty string (Xâu rỗng) = Phân đoạn dữ liệu được hiển thị (trạng thái bình thường)

12. Chạy hai câu lệnh select count (lấy số đếm bản ghi) để kiểm tra tính sẵn có để sử dụng của dữ liệu trong hai bảng liên quan trong câu lệnh gắn thêm vào.

Liệt kê 43. Đếm Lineitem

db2 "select count(*) from lineitem"

13.

Hình 21. Kết quả từ câu lệnh select count

```
[db2inst1@localhost scripts]$ db2 "select count(*) from lineitem"

1
-----------
        271

  1 record(s) selected.
```

Lưu ý: Các phân đoạn gốc trong bảng LINEITEM có sẵn nhưng ta chưa nhìn thấy các dữ liệu mới trong phân đoạn PartitionId 4.

Liệt kê 44. Đếm np-lineitem

db2 “select count(*) from np_lineitem”

14.

Hình 22. Kết quả từ câu lệnh select count

```
[db2inst1@localhost scripts]$ db2 "select count(*) from np_lineitem"
SQL0204N  "DB2INST1.NP_LINEITEM" is an undefined name.  SQLSTATE=42704
```

Lưu ý: Bảng NP_LINEITEM bây giờ là một đối tượng chưa được xác định và chỉ sẵn sàng với vai trò là một phân đoạn trong bảng LINEITEM.

15. Tạo một bảng ngoại lệ để sử dụng với câu lệnh SET INTEGRITY. Câu lệnh DDL này nằm trong tệp tin EX3-7.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 45. Bảng ngoại lệ

```
db2 –vtf EX3-7.sql
```

16.

17. Chạy lệnh set integrity cho bảng LINEITEM đã phân đoạn.

Liệt kê 46. Lệnh Set integrity

```
SET INTEGRITY FOR LINEITEM

ALLOW WRITE ACCESS

IMMEDIATE CHECKED
```

FOR EXCEPTION IN LINEITEM USE
LINEITEM_EX

18.

SQL nằm trong tệp EX3-8.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 47. Lệnh Set integrity

db2 –vtf EX3-8.sql

19.

Hình 23. Tệp tin EX3-8.sql

```
[db2inst1@localhost scripts]$ db2 -vtf EX3-8.sql
set integrity for lineitem allow write access immediate checked for exception in
 lineitem use lineitem_ex
DB20000I  The SQL command completed successfully.

commit work
DB20000I  The SQL command completed successfully.
```

Lưu ý: Lệnh SET INTEGRITY là cần thiết để kiểm tra xem dữ liệu vừa được gắn vào đã ở trong dải chưa. Lệnh này cũng làm mọi việc bảo trì cần thiết với các chỉ mục và các đối tượng phụ thuộc khác, chẳng hạn như bảng truy vấn được vật chất hóa (MQTs). Cho đến khi câu lệnh SET

INTEGRITY được giao kết, dữ liệu mới không nhìn thấy được. Tuy nhiên, dữ liệu hiện tại trong bảng LINEITEM là hoàn toàn có thể truy cập được cho cả đọc và viết trong khi lệnh SET INTEGRITY đang chạy. Người sử dụng phải giao kết (commit) giao dịch SET INTEGRITY để làm cho toàn bộ bảng sẵn sàng để sử dụng. Khi câu lệnh SET INTEGRITY đang chạy, bạn không thể thực hiện các DDL hoặc các hoạt động kiểu tiện ích trên bảng. Trong bài tập này, tất cả các hàng được tạo ra trong bảng NP_LINEITEM, vừa được gắn thêm vào bảng LINEITEM, đều nằm trong dải giá trị được xác định trong câu lệnh gắn thêm vào. Nếu bất kỳ một trong các hàng đó nằm ngoài dải giá trị, thì bạn cần phải có một bảng ngoại lệ trong câu lệnh SET INTEGRITY để tránh cho câu lệnh khỏi thất bại. Người ta khuyến cáo rằng bạn nên luôn luôn đưa một bảng ngoại lệ vào câu lệnh SET INTEGRITY. Nếu không có bảng ngoại lệ thì bất kỳ hành vi vi phạm nào được câu lệnh SET INTEGRITY phát hiện sẽ làm cho câu lệnh thất bại và tất cả các công việc sẽ bị cuộn ngược lại (rolled back). Đây rất có thể là một hoạt động kéo dài nếu một khối lượng lớn dữ liệu được sử dụng. Ta cũng cần lưu ý rằng nếu hoạt động SET INTEGRITY không thực hiện được, thì tất cả công việc sẽ bị cuộn trở lại, điều này trái ngược với hoạt động LOAD chỉ đơn giản loại bỏ các hàng sai.

20. Chạy SQL select count (lấy số đếm bản ghi) trên bảng LINEITEM để kiểm tra tính sẵn có của dữ liệu trong phân đoạn đã gắn vào:

Liệt kê 48. Đếm Lineitem

db2 “select count(*) from lineitem”

21.

Hình 24. Kết quả từ câu lệnh select count

```
[db2inst1@localhost scripts]$ db2 "select count(*) from lineitem"

1
-----------
        358

  1 record(s) selected.
```

Lưu ý: Sau khi hoàn tất hoạt động SET INTEGRITY thành công, bây giờ bảng LINEITEM sẽ chứa các dữ liệu từ phân đoạn PartitionId 4.

Tháo ra một phân đoạn từ một bảng được phân đoạn

1. Sử dụng lệnh describe data partitions show detail (mô tả phân đoạn dữ liệu, hiển thị chi tiết) để xác định PartitionName của phân đoạn mà bạn sẽ tháo ra (roll-out) từ LINEITEM được phân đoạn.

Liệt kê 49. Lệnh Describe table

db2 describe data partitions for table
LINEITEM show detail

2.

Hình 25. Các phân đoạn cho bảng LINEITEM

```
PartitionId PartitionName                  TableSpId   PartObjId   LongTblSpId
AccessMode

  Status
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 JAN1992                                  4         282           4
F
          1 JULY1992                                 5         282           5
F
          2 JAN1993                                  6           4           6
F
          3 JULY1993                                 7           4           7
F
          4 JAN1994                                  3         785           3
F

  5 record(s) selected.
```

Lưu ý: Bạn sẽ tháo ra dải phân đoạn cũ nhất PartitionId 0. PartitionName của phân đoạn này là JAN1992. Nó sẽ được sử dụng trong hoạt động DETACH (tháo ra) để xác định phân đoạn được tháo ra. Bạn cũng lưu ý rằng vì hoạt động SET INTEGRITY đã thành công, nên giá trị AccessMode của phân đoạn JAN1994 là 'F' và giá trị Status là xâu rỗng. Các kết quả cho các mục TableSpId, PartObjId và LongTblSpId có thể khác với các kết quả hiển thị ở đây.

3. Sử dụng câu lệnh Alter để tháo ra (roll-out) phân đoạn JAN1992 khỏi bảng LINEITEM.

Liệt kê 50. Lệnh Alter table

ALTER TABLE LINEITEM DETACH
PARTITION JAN1992 INTO
LINEITEM_JAN1992

4.

SQL để tạo tệp tin nằm trong tệp EX3-11.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 51. Lệnh Alter table

db2 –vtf EX3-11.sql.

5.

Lưu ý: Việc tháo ra phân đoạn JAN1992 hoàn tất thành công và một bảng mới LINEITEM_JAN1992 được tạo ra. Không có dịch chuyển dữ liệu nào liên quan đến hoạt động DETACH này và bảng mới nằm trong cùng vùng bảng như trước đây khi nó là một phần của bảng phân đoạn LINEITEM. Không cần thiết phải chạy câu lệnh SET INTEGRITY trên bảng LINEITEM tại điểm này bởi vì không có MQT nào được định nghĩa trên bảng đó. Cũng cần lưu ý rằng nếu một bảng được tạo ra là kết quả của một hoạt động tháo ra một phân đoạn từ bảng phân cụm nhiều chiều (Multi-Dimensional Clustering -MDC) được phân đoạn, thì bảng đó cũng sẽ là một

MDC. Quy tắc này cũng được áp dụng khi bạn tháo ra một phân đoạn từ một bảng phân tán để tạo ra một bảng phân tán nằm trong cùng nhóm phân đoạn. Bảng được tạo ra từ hoạt động DETACH không được định nghĩa với bất kỳ chỉ mục nào khác với các chỉ mục của MDC. Đối với các MDC, thì việc xây dựng lại chỉ mục sẽ xảy ra tại lần truy cập đầu tiên tới các bảng được tách ra. Việc làm sạch chỉ mục của một phân đoạn được tháo ra sẽ tự động được thực hiện tại nền sau. Lược đồ, các đặc quyền, và vùng bảng của các chỉ mục được thừa hưởng từ ID của người dùng đã thực hiện hoạt động DETACH.

6. Chạy hai câu lệnh select count để kiểm tra tính sẵn có của dữ liệu trong hai bảng liên quan đến câu lệnh DETACH.

Liệt kê 52. Đếm số bản ghi trong bảng Lineitem_jan1992

db2 “select count(*) from
lineitem_jan1992”

7.

Hình 26. Kết quả từ câu lệnh select count

```
[db2inst1@localhost scripts]$ db2 "select count(*) from lineitem_jan1992"

1
-----------
         38

  1 record(s) selected.
```

Lưu ý: Bảng LINEITEM_JAN1992 được tạo ra và chứa 38 hàng trong phân đoạn JAN1992 của bảng được phân đoạn LINEITEM.

Liệt kê 53. Đếm lineitem

db2 “select count(*) from lineitem”

8.

Hình 27. Kết quả từ câu lệnh select count

```
[db2inst1@localhost scripts]$ db2 "select count(*) from lineitem"

1
-----------
        320

  1 record(s) selected.
```

Lưu ý: Bảng LINEITEM hoàn toàn có sẵn để dùng tại thời điểm này và không còn chứa các dữ liệu từ PART0.

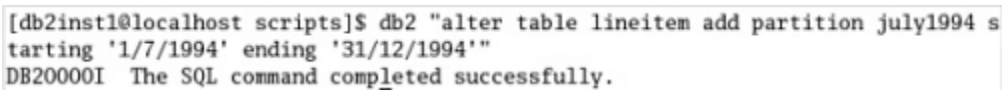
9. Khi dữ liệu được nhỏ giọt đưa vào một bảng phân đoạn, hoặc khi bạn muốn nạp hoặc chèn dữ liệu trực tiếp vào một bảng phân đoạn, thì có thể sẽ thích hợp hơn nếu ta thêm một phân đoạn rỗng vào một bảng được phân đoạn hiện có. Bạn sử dụng lệnh sau đây để thêm một phân đoạn rỗng vào bảng LINEITEM hiện tại:

Liệt kê 54. Lệnh describe

```
db2 “ALTER TABLE LINEITEM
ADD PARTITION JULY1994
STARTING '1/7/1994' ENDING
'31/12/1994'”
```

10.

Hình 28. Thêm một phân đoạn rỗng vào bảng LINEITEM hiện tại



11. Sử dụng lệnh describe data partitions show detail để kiểm tra xem phân đoạn với tên phân đoạn là JULY1994 đã được thêm vào bảng LINEITEM chưa:

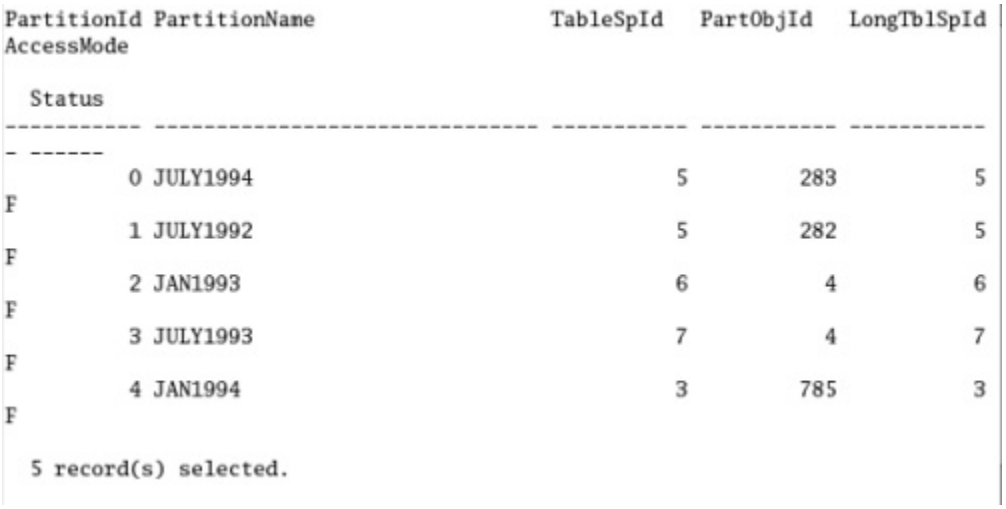
Liệt kê 55. Lệnh describe

```
db2 describe data partitions for table
```

LINEITEM show detail

12.

Hình 29. Các phân đoạn cho bảng LINEITEM



```
PartitionId PartitionName                  TableSpId   PartObjId   LongTblSpId
AccessMode

  Status
----------- ------------------------------ ----------- ----------- -----------
- ------
          0 JULY1994                               5         283           5
F
          1 JULY1992                               5         282           5
F
          2 JAN1993                                6           4           6
F
          3 JULY1993                               7           4           7
F
          4 JAN1994                                3         785           3
F

  5 record(s) selected.
```

Các kế hoạch truy cập cho các bảng được phân đoạn

Bài thực hành này nghiên cứu cách các bảng được phân đoạn được miêu tả như thế nào trong các kế hoạch truy cập:

1. Bạn sẽ cập nhật các số liệu thống kê trên một bảng được phân đoạn.
2. Bạn sẽ sử dụng lệnh db2expln và phân tích kết quả.
3. Bạn sẽ sử dụng các lệnh DB2 và SQL trong các quá trình hoạt động để xem lại tiến độ.

1. Thực hiện hoạt động RUNSTATS trên bảng LINEITEM:

Liệt kê 56. Lệnh Runstats

db2 runstats on table db2inst1.lineitem

2.

3. Chạy công cụ giải nghĩa (Explain) lệnh SQL sau và xem xét kết quả đầu ra giải nghĩa:

Liệt kê 57. Công cụ giải nghĩa

db2 “select l_shipdate,sum(l_quantity)
from LINEITEM group by l_shipdate”

4.

SQL cần giải nghĩa nằm trong tệp tin EX4-2.sql và có thể chạy bằng lệnh sau:

Liệt kê 58. Đầu ra của công cụ giải nghĩa

db2expln –d SAMPLE –t –f EX4-2.sql

5.

Hình 30. Đầu ra của công cụ giải nghĩa

```
Access Table Name = DB2INST1.LINEITEM  ID = -6,-32768
|  #Columns = 2
|  Data-Partitioned Table
|  All data partitions will be accessed
|  Relation Scan
|  |  Prefetch: Eligible
|  Lock Intents
|  |  Table: Intent Share
|  |  Row  : Next Key Share
|  Sargable Predicate(s)
|  |  Insert Into Sorted Temp Table  ID = t1
|  |  |  #Columns = 3
|  |  |  #Sort Key Columns = 1
|  |  |  |  Key 1: L_SHIPDATE (Ascending)
|  |  |  Sortheap Allocation Parameters:
|  |  |  |  #Rows     = 249.000000
|  |  |  |  Row Width = 37
|  |  |  Piped
|  |  |  Buffered Partial Aggregation
Sorted Temp Table Completion  ID = t1
Access Temp Table  ID = t1
|  #Columns = 3
|  Relation Scan
|  |  Prefetch: Eligible
Final Aggregation
|  Group By
|  Column Function(s)
Return Data to Application
|  #Columns = 2

End of section
```

Lưu ý: SQL này thực hiện hoạt động quét chỉ mục của bảng LINEITEM. Những điểm cần lưu ý trong kết quả đầu ra của công cụ giải nghĩa, về khía cạnh phân đoạn bảng, là bảng đã truy cập được phân đoạn và rằng tất cả các phân đoạn dữ liệu được truy cập trong quá trình quét.

6. Chạy công cụ giải nghĩa (Explain) lệnh SQL sau và xem xét kết quả đầu ra giải nghĩa:

Liệt kê 59. Công cụ giải nghĩa

```
db2 “select l_shipdate, l_partkey,
l_returnflag

from LINEITEM

where l_shipdate between '01/01/1993'
and '31/08/1993'

and l_partkey = 49981”
```

7.

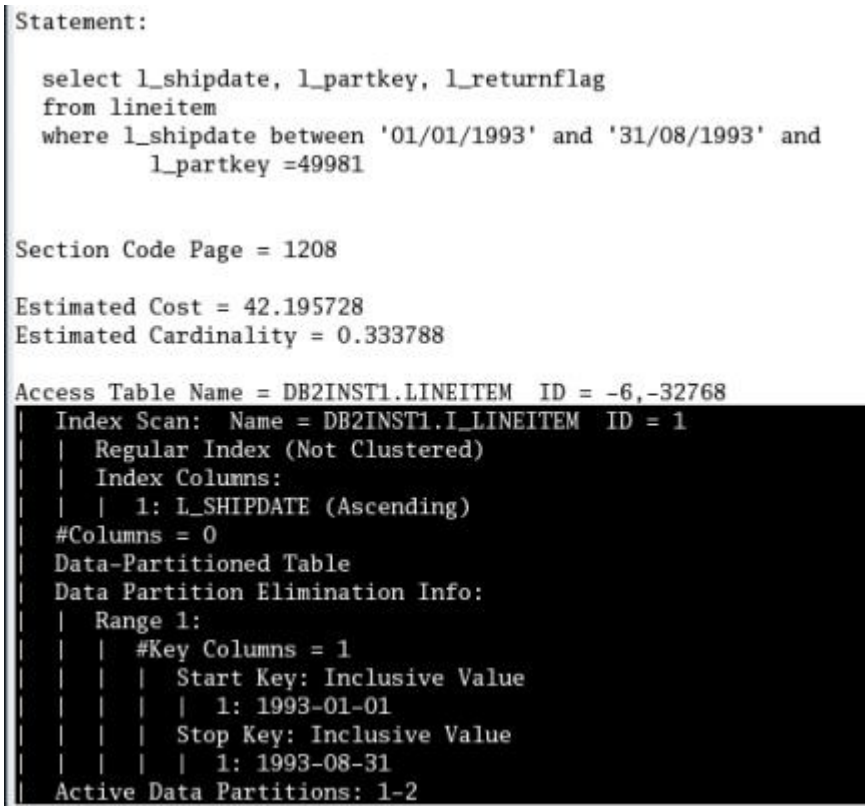
SQL cần giải nghĩa nằm trong tệp tin EX4-3.sql và có thể chạy bằng lệnh sau đây:

Liệt kê 60. Đầu ra của công cụ giải nghĩa

```
db2expln –d SAMPLE –t –f EX4-3.sql
```

8.

Hình 31. Đầu ra giải nghĩa



```
Statement:

  select l_shipdate, l_partkey, l_returnflag
  from lineitem
  where l_shipdate between '01/01/1993' and '31/08/1993' and
          l_partkey =49981


Section Code Page = 1208

Estimated Cost = 42.195728
Estimated Cardinality = 0.333788

Access Table Name = DB2INST1.LINEITEM  ID = -6,-32768
|  Index Scan:  Name = DB2INST1.I_LINEITEM  ID = 1
|  |  Regular Index (Not Clustered)
|  |  Index Columns:
|  |  |  1: L_SHIPDATE (Ascending)
|  #Columns = 0
|  Data-Partitioned Table
|  Data Partition Elimination Info:
|  |  Range 1:
|  |  |  #Key Columns = 1
|  |  |  |  Start Key: Inclusive Value
|  |  |  |  |  1: 1993-01-01
|  |  |  |  Stop Key: Inclusive Value
|  |  |  |  |  1: 1993-08-31
|  Active Data Partitions: 1-2
```

Lưu ý: SQL này thực hiện hoạt động quét chỉ mục của bảng LINEITEM. Trong ví dụ này, bạn thấy rằng trình tối ưu hóa là có thể thực hiện việc xoá bỏ phân đoạn dữ liệu. Những điểm cần lưu ý trong đầu ra giải nghĩa, về khía cạnh phân đoạn bảng, đó là bảng đã truy cập được phân đoạn, đã diễn ra việc xóa bỏ phân đoạn và các giá trị của các phân đoạn dữ liệu đang hoạt động. Trong ví dụ này, các phân đoạn dữ liệu đang hoạt động là 1-2. Đây là các số thứ tự (seqno) trong danh mục hệ thống syscat.datapartitions chứ không phải là PartitionId trong câu lệnh describe data partitions (mô tả phân đoạn dữ liệu).

9. Sử dụng SQL sau đây để xác định các tên của phân đoạn là các phân đoạn đang hoạt động trong ví dụ giải nghĩa ở trên:

Liệt kê 61. Lệnh describe

db2 “select seqno,datapartitionname

from syscat.datapartitions

where tabname = ‘LINEITEM’ order by
seqno”

10.

Hình 32. Các tên của phân đoạn

```
[db2inst1@localhost scripts]$ db2 "select seqno, datapartitionname from syscat.d
atapartitions where tabname= 'LINEITEM' order by seqno"

SEQNO       DATAPARTITIONNAME

----------- ------------------------------------------------------------------
------------------------------------------------------------

          0 JULY1992

          1 JAN1993

          2 JULY1993

          3 JAN1994

          4 JULY1994

  5 record(s) selected.
```

Lưu ý: Các số thứ tự 1 và 2 ánh xạ đến phân đoạn có tên là JAN1993 và JULY1993.

Kết luận

Hướng dẫn này được dựa trên đặc tính phân đoạn dữ liệu của DB2 9 của IBM. Bạn đã hoàn thành các bài tập bằng việc làm các bài thực hành các vấn đề sau:

- Làm thế nào để định nghĩa các bảng được phân đoạn
- Làm thế nào để đặt các bảng được phân đoạn vào hệ thống con của ổ đĩa ở phía dưới
- Làm thế nào để bảo trì các bảng được phân đoạn
- Làm thế nào để diễn giải các bảng được phân đoạn với DB2 Explain

Phân đoạn theo dải giá trị ánh xạ dữ liệu đến các phân đoạn dựa trên các dải giá trị khóa mà người sử dụng thiết lập cho mỗi phân đoạn. Ví dụ, các công ty thường thích chia các dữ liệu bán hàng thành các phân đoạn theo tháng. Khi được sử dụng kết hợp với các khả năng MDC, thì việc phân đoạn theo dải giá trị cho phép lấy ra một cách nhanh chóng các thông tin từ các truy vấn phức tạp, bằng cách làm cho dữ liệu trở nên được định vị dễ dàng hơn.