UNIT 17

테이블 파티셔닝

대용량 데이터의 처리 시 테이블 파티셔닝은 관리 및 성능상 많은 장점을 줄 수 있습니다. Range 파티션 테이블의 생성 방법과 유지 보수 방법을 소개합니다.

DB2 9.7 운영자 가이드

Administrator Edition

- 테이블 파티셔닝 개요
- 파티션 추가
- 파티션 제거
- 파티션 테이블 생성
- Detach/Attach/Add 구문

TOPIC 17 01 테이블 파티셔닝 개요

Point

테이블 파티셔닝은 대용량 테이블의 효과적인 저장 및 처리를 위해 사용되며 데이터의 조회 시 파티션 Elimination을 통하여 성능을 향상 시킬 수 있습니다.

Tip

파티션 테이블의 유연한 유지 보수를 위해서 파티션 별로 별도의 테이블 공간에 저장하는 것이 좋습니다.

Tip

파티셔닝 컬럼은 여러 컬럼으로 구성 될 수 있습니다.

1 테이블 파티셔닝은 다음과 같습니다.

테이블 파티셔닝 은 새로운 범위의 데이터가 계속 추가되어 테이블이 대용량화 되었을 때 Roll-in, Roll-out 기술을 사용하여, 데이터 조회 시 옵티마이저에 의해 불필요한 범위의 스캔을 하지 않음으로 쿼리 성능을 향상시킵니다.

➲ 테이블 파티셔닝

- ➢ 테이블을 범위 단위(일반적으로 주, 월, 분기 또는 년)로 나눕니다.
- ➢ 각 Range는 서로 다른 테이블스페이스에 둘 수 있습니다.
- ➢ Range는 독립적으로 스캔이 가능합니다.
- ➢ ATTACH/DETACH 명령으로 테이블의 roll-in/roll-out이 가능합니다.

일반 테이블 조회

```
SELECT ID, HDATE FROM IBMDB2.SAWON
    WHERE HDATE >= '2009-01-01' AND HDATE < '2010-01-01' ;
```

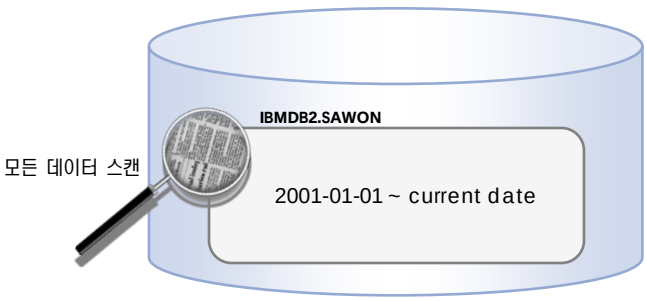


Figure 1701A ••• 일반 테이블 조회 시 모든 데이터 범위 스캔

파티션된 테이블 조회

```
SELECT ID, HDATE FROM IBMDB2.SAWON
    WHERE HDATE >= '2009-01-01' AND HDATE < '2010-01-01' ;
```

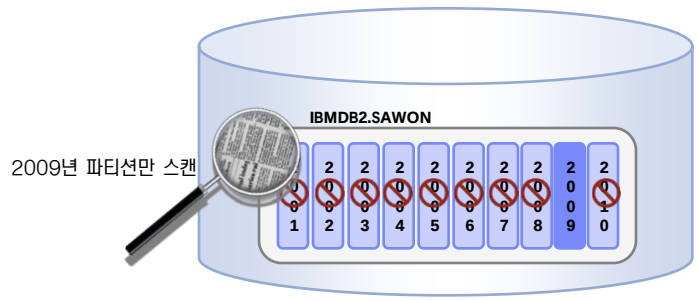


Figure 1701B ••• 파티션된 테이블 조회 시 해당 년도 파티션만 스캔

TOPIC 17 02 파티션 추가

Point

파티션 테이블에 특정 파티션 데이터 추가를 Roll-in이라고 합니다. 관련 명령어는 Attach 입니다.

Tip

Attach 명령을 통해 파티션을 Roll-in 하였다면 SET INTEGRITY 명령을 발행해야 Roll-in 한 데이터를 사용 할 수 있습니다.

Tip

SET LOCK TIMEOUT WAIT 명령을 사용하면 lock 으로 인한 실패로부터 SET INTEGRITY 를 보호할 수 있습니다.

Tip

기회비용을 줄이기 위해 새로운 range partition을 추가 할 때에는 텅빈 파티션으로 Attach 명령을 실행합니다.

1 Roll-in

파티션 된 테이블은 물리적으로 분리된 저장공간에 저장되어 독립된 오브젝트의 데이터를 논리적인 하나의 오브젝트로 만들어진 테이블입니다. 파티션 된 테이블에서 각 파티션을 구분 짓는 range는 기존 데이터와 중첩되지 않는다면 ATTACH 명령으로 새로운 파티션을 Roll-in 할 수 있습니다.

- Rolling in 작업의 장점
  - 테이블의 가용성이 높다. ATTACH 명령의 사용으로 짧은 시간 안에 테이블 사용 가능 합니다.
  - SET INTEGRITY 명령 실행 즉시 모든 데이터를 사용할 수 있습니다.
  - MQTs, constraints, generated columns 를 사용할 수 있습니다.

2 파티션 ATTACH 절차

- ALTER TABLE DETACH PARTITON
  - source와 target 테이블의 Row는 SYSIBM.SYSTABLES 에 X-lock.
  - detach된 파티션의 Row는 SYSIBM.SYSDATAPARTITIONS 에 X-lock.
  - source 테이블에는 Z-lock 이 생성됩니다.
- COMMIT
  - 작업 중 발생된 모든 Lock 이 release 됩니다.
- ONLINE SET INEGRITY
  - ATTACH된 파티션에 Read/Write 가 가능하게 됩니다.

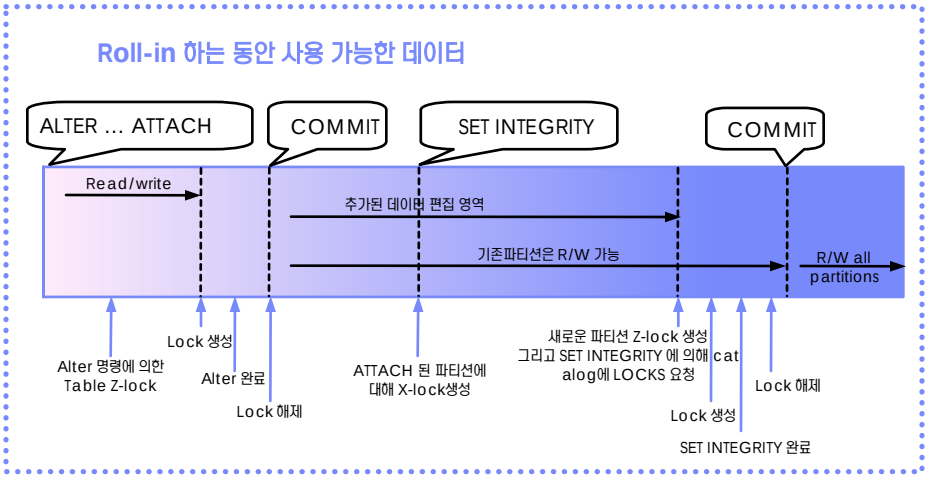


Figure 1702A ••• Roll-in 내부 절차

TOPIC 17 03 파티션 제거

Point

파티션 테이블에 특정 파티션 데이터의 제거를 Roll-out이라고 합니다. 관련 명령어는 Detach 입니다.

Tip

분리된 파티션 테이블은 데이터 변경 이후 다시 Attach될 수 있습니다.

1 Roll-out

파티션 된 테이블에서 기존의 정의된 range 를 DETACH 명령으로 분리 할 수 있습니다. Roll-out된 파티션은 기존 파티션에서 분리되어 사용할 수 없으므로 굳이 DELETE작업을 수행 할 필요가 없습니다.

➲ Rolling out 의 고려사항

- ➢ DETACH 명령이 실행되는 동안 테이블은 offline이 됩니다.
- ➢ 인덱스는 비동기적으로 정리가 된다. 그러므로 따로 인덱스 작업을 따로 수행할 필요가 없습니다.
- ➢ 파티션된 인덱스의 Roll-out 시, 인덱스 클린업 작업이 필요 없습니다.

① 시점 별 가용한 테이블 의 데이터

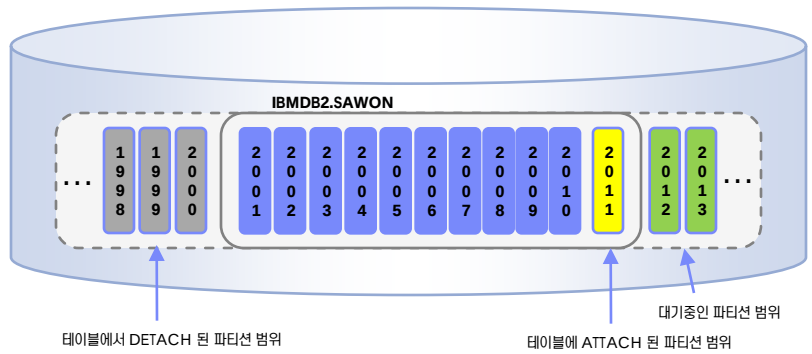


Figure 1703A • • • Roll-out된 테이블

Tip

DETACH, COMMIT 실행 후 SET INTEGRITY 명령은 따로 실행할 필요가 없습니다. 그러나 MQT를 사용할 경우는 SET INTEGRITY ... FULL ACCESS 명령을 실행하여야 이용 가능합니다.

2 파티션 DETACH 절차

➲ ALTER TABLE DETACH PARTITON

- ➢ source와 target 테이블의 Row는 SYSIBM.SYSTABLES에 X-lock.
- ➢ detach된 파티션의 Row는 SYSIBM.SYSDATAPARTITIONS 에 X-lock.
- ➢ source 테이블에는 Z-lock 이 생성됩니다.

➲ COMMIT

- ➢ 작업 중 발생된 모든 Lock 이 release 됩니다.

② Roll-out 하는 동안 사용 가능한 데이터

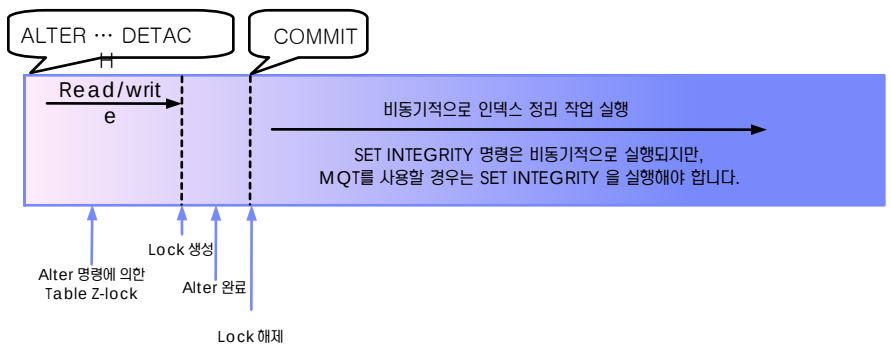


Figure 1703B • • • Roll-out 내부 절차

TOPIC 17 04 파티션 테이블 생성

Point

파티션 테이블의 작성방법은 짧은 구문과 긴 구문으로 작성될 수 있습니다.

Tip

파티션 된 테이블의 각 파티션은 서로 다른 테이블스페이스에 위치시킬 수 있습니다.

Tip

DB2 9.7에서는 서로 다른 파티션에 대해 파티션 인덱스를 생성 할 수 있습니다.

Tip

INCLUSIVE 옵션은 포함된 값이며 EXCLUSIVE 옵션은 제외된 값입니다. STARTING, ENDING에 아무런 표시가 없는 값은 INCLUSIVE 를 의미합니다.

1 파티션 된 테이블 작성

테이블의 데이터는 CREATE TABLE문의 PARTITION BY절에 의해 여러 개의 서로 다른 물리적인 저장 공간에 저장 할 수 있습니다. Partition range는 PARTITION BY절의 STARTING FROM 및 ENDING AT 값으로 지정 할 수 있습니다.

```
CREATE TABLE <table name>
      ...
      IN <table space name1>
      INDEX IN <table space name2>
      LONG IN <table space name3>
   PARTITIONED BY ... PARTITION <partition name> | boundary specification |
                IN <table space name4>
                INDEX IN <table space name5>
                LONG IN <table space name6>
```

2 파티션 테이블 생성을 위한 구문은 아래와 같습니다.

짧은 구문

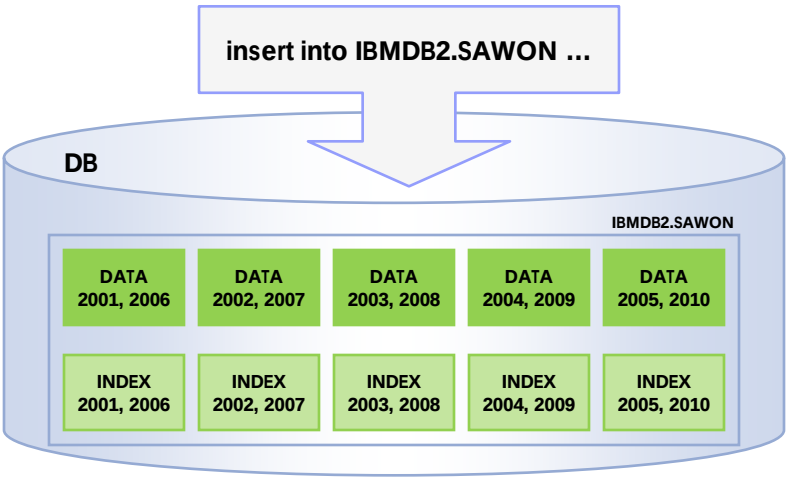
```
CREATE TABLE T1(C1 INT) IN tbsp1, tbsp2, tbsp3
 PARTITION BY RANGE(C1)
 STARTING FROM (1) ENDING (100) EVERY (33)
```

긴 구문

```
CREATE TABLE T1(C1 INT)  PARTITION BY RANGE(C1)
 STARTING FROM (1) ENDING (34) IN tbsp1,
                   ENDING(67)   IN tbsp2,
                   ENDING(100) IN tbsp3
```

➲ 아래는 각 년도 별로 파티션을 만들어 데이터들을 원하는 저장 장소로 모으는 예제입니다.

	1년과 6년	2년과 7년	3년과 8년	4년과 8년	5년과 0년
데이터저장	TSD_IBM01	TSD_IBM02	TSD_IBM03	TSD_IBM04	TSD_IBM05
인덱스저장	TSI_IBM01	TSI_IBM02	TSI_IBM03	TSI_IBM04	TSI_IBM05



TOPIC 17 05 Detach/Attach/Add 구문

Point

파티션 테이블에서 Detach후 Archive를 하거나 새로운 파티션을 추가하는 경우 구문입니다.

Tip

이미 만들어진 새로운 영역의 테이블을 파티션 된 테이블에 삽입 하려면 ATTACH PARTITION 명령을 사용하고 기존에 테이블로 만들어지지 않은 영역은 ADD PARTITION 명령을 사용합니다.

1 Archive 시킬 파티션은 먼저 DETACH합니다.

```
connect to TESTDB;

ALTER TABLE IBMDB2.SAWON
  DETACH PARTITION P_SAWON_2000
  INTO IBMDB2.SAWON_OLD_DETACH;

connect reset;
terminate;
```

2 기존 파티션 테이블에 새로운 파티션의 추가 – 데이터를 가진 기존 테이블인 경우

```
connect to TESTDB;

ALTER TABLE IBMDB2.SAWON
    ATTACH PARTITION P_SAWON_2011
        STARTING FROM ('2011-01-01') INCLUSIVE
        ENDING AT ('2011-12-31') INCLUSIVE
    FROM IBMDB2.SAWON_2011_ATTACH ;

connect reset;
terminate;
```

3 기존 파티션 테이블에 새로운 파티션의 추가 – 빈 파티션만 추가

```
connect to TESTDB;

ALTER TABLE IBMDB2.SAWON
      ADD PARTITION P_SAWON_2012
      STARTING ('2012-01-01') ENDING('2012-12-31') IN TSD_IBM02 INDEX IN
TSI_IBM02 ;

connect reset;
terminate;
```

Memo