




UNIT 10

데이터베이스 운영



데이터베이스 시스템을 운영할 때는 업무 유형에 따라 일정한 주기로 REORGCHK, REORG, RUNSTATS 등의 명령어를 실행하여, 테이블과 인덱스의 현재 저장 상태를 최적화하고, REBIND 등의 명령어를 실행하여 항상 최적의 액세스 플랜을 유지하는 것이 중요합니다.

DB2 9.7 운영자 가이드

Administrator Edition

- 일반적인 운영 방법
- REORGCHK 명령어
- 재구성 필요 여부 판별 방법
- REORG TABLE 명령어
- REORG INDEXES 명령어
- RUNSTATS 명령어
- 액세스 플랜 갱신
- REBIND 명령어



Point



데이터베이스 시스템을 운영할 때는 일정한 주기로 RUNSTATS, REORGCHK, REORG, REBIND 등의 명령어들을 실행하여 항상 효율적인 액세스 플랜이 적용될 수 있도록 하는 것이 중요합니다.

Tip

인덱스 추가 또는 대량의 데이터 로드 등의 작업 후에는 RUNSTATS 명령어로 통계 자료를 갱신하고, REBIND 명령어로 패키지를 갱신하는 것이 권장됩니다.

- 1 REORGCHK, REORG TABLE, REORG INDEXES, RUNSTATS, REBIND 등의 유틸리티를 이용하여 데이터베이스의 구조가 최적화되었는지 정기적으로 점검합니다.

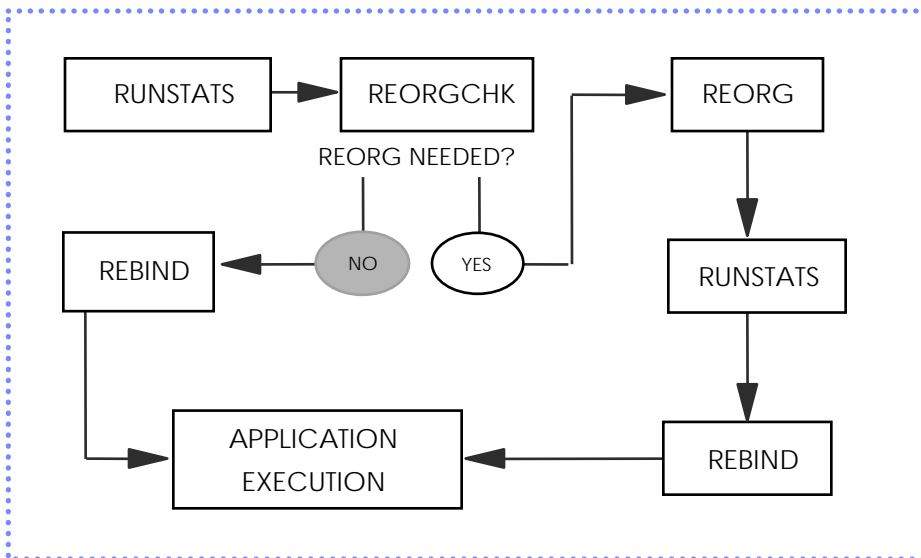


Figure 1001A... 운영 유틸리티를 이용한 데이터 운영

- 2 DB2 사용자로 로그인하여 데이터베이스에 접속합니다.

```
$ login <DB2 사용자>
$ db2 connect to <데이터베이스명>
```

- 3 REORGCHK 명령어는 테이블과 인덱스의 재구성 필요 여부를 판별합니다.

```
$ db2 reorgchk on table <스키마명>.<테이블명>
```

- 4 재구성이 필요하다고 판별되면 REORG TABLE 명령어와 REORG INDEX 명령어를 이용하여 테이블과 인덱스를 재구성합니다.

```
$ db2 reorg table <스키마명>.<테이블명>
$ db2 reorg index <스키마명>.<인덱스명>
```

- 5 REORG 명령어를 실행한 후에는 반드시 RUNSTATS 명령을 이용하여 시스템 카탈로그의 통계 자료를 갱신합니다.

```
$ db2 runstats on table <스키마명>.<테이블명> and indexes all
```

- 6 시스템의 통계 자료가 갱신되면, 더 적합한 액세스 플랜이 생성되도록 모든 패키지를 갱신합니다.

```
$ db2rbind <데이터베이스명> /I <로그파일명> all
```

Tip

테이블명 또는 인덱스명을 명시할 때는 <스키마명>.<오브젝트명>의 형태를 사용하도록 합니다.

Point



REORGCHK 명령어는 8가지의 판별식을 사용하여 테이블과 인덱스의 재구성 필요 여부를 판단합니다. 각 판별식을 적용한 결과가 기준치를 벗어난 경우에는 * 가 표시되어 테이블 또는 인덱스의 재구성이 필요함을 알려줍니다.

Tip

SYSADM, DBADM 권한 또는 테이블에 대한 CONTROL 특권을 가진 사용자가 실행합니다.

Tip

REORGCHK 를 실행하기 전에는 RUNSTATS를 실행하는 것이 권장됩니다. CURRENT STATISTICS 옵션은 최근에 RUNSTATS를 이미 실행한 경우에만 사용하도록 합니다.

- 1 REORGCHK 명령어는 8가지 판별식으로 테이블과 인덱스의 재구성 필요 여부를 판별합니다.

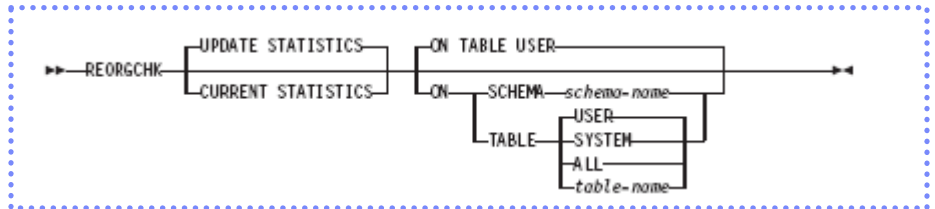


Figure 1002A... REORGCHK 명령어

- 2 현재의 통계 자료를 유지한 상태로 재구성 필요 여부를 점검합니다. 옵션을 지정하지 않으면, 기본값인 UPDATE STATISTICS 가 적용되므로 자동으로 RUNSTATS 명령어가 실행됩니다.

```
$ db2 reorgchk CURRENT STATISTICS on table <스키마명>.<테이블명>
```

- 3 특정한 <스키마명>을 가지는 테이블들에 대해서 재구성 필요 여부를 점검합니다.

```
$ db2 reorgchk ON SCHEMA <스키마명>
```

- 4 시스템 카탈로그 테이블에 대해서만 재구성 필요 여부를 점검합니다.

```
$ db2 reorgchk ON TABLE SYSTEM
```

- 5 모든 테이블에 대해서 재구성 필요 여부를 점검합니다.

```
$ db2 reorgchk ON TABLE ALL
```

- 6 특정 테이블에 대해서 재구성 필요 여부를 점검합니다.

```
$ db2 reorgchk ON TABLE <스키마명>.<테이블명>
```

Table Statistics

SCHEMA	NAME	CARD	OV	NP	FP	TSIZE	F1	F2	F3	REORG
ADMIN	ACCT	1000000	0	27792	27792	1.10e+008	0	97	100	---
...										

"*"
exceeded
set
bounds

INDEX Statistics

SCHEMA NAME	CARD	LEAF	ELEAF	LVLS	ISIZE	NDEL	KEYS	F4	F5	F6	F7	F8	REORG
Table: ADMIN.ACCT													
ADMIN ACCT_GRP	1e+006	1703	0	3	2	0	1000	4	71	20	0	0	---
ADMIN ACCTINDEX	1e+006	3611	0	3	4	0	1e+006	100	87	6	0	0	---

Figure 1002B... REORGCHK 명령어의 실행 결과

Point



총 8가지의 REORGCHK 판별식 중에서 F1~F3 까지의 판별식은 테이블의 재구성 여부를 판별하는데 사용되며, F4~F8 까지의 판별식은 각 인덱스에 대한 재구성 여부를 판별하는데 사용됩니다.

Tip

- 재구성을 실행하더라도 모든 인덱스에 대한 클러스터링 비율이 좋아질 수는 없으므로, 가장 중요한 인덱스를 기준으로 재구성하는 것이 좋습니다.

1

테이블에 대한 3가지 판별식을 이용하여 테이블의 재구성 필요 여부를 판별합니다.

Table Statistics:

F1: $100 * \text{OVERFLOW} / \text{CARD} < 5$
 F2: $100 * \text{TSIZE} / ((\text{FPAGES} - \text{NPARTITIONS}) * (\text{TABLEPAGESIZE} - 68)) > 70$
 MDC 테이블) $100 * \text{TSIZE} / ((\text{ACTBLK} - \text{FULLKEYCARD}) * \text{EXTENTSIZE} * (\text{TABLEPAGESIZE} - 68)) > 70$
 F3: $100 * \text{NPAGES} / \text{FPAGES} > 80$
 MDC 테이블) $100 * \text{activeblocks} / ((\text{fpages} / \text{ExtentSize}) - 1)$

Figure 1003A... 테이블 재구성 필요 여부 판별식

판별식	설 명
F1	오버플로우된 행의 건수는 전체 건수의 5% 미만이어야 합니다.
F2	할당된 전체 공간에서 비어있는 공간의 총합은 30% 이하이어야 합니다.
F3	완전히 비어있는 페이지의 수는 전체 페이지 수의 20% 이하이어야 합니다.

2

인덱스에 대한 5가지 판별식을 이용하여 인덱스의 재구성 필요 여부를 판별합니다.

Index Statistics:

F4: CLUSTERRATIO or normalized $\text{CLUSTERFACTOR} > 80$
 다중파티션), $\text{AVGPARTITION_CLUSTERRATIO}$ or normalized $\text{AVGPARTITION_CLUSTERFACTOR} > 80$
 F5: $100 * (\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}) / ((\text{NLEAF} - \text{NUM_EMPTY_LEAFS} - 1) * (\text{INDEXPAGESIZE} - \text{LEAF_PAGE_OVERHEAD})) > \text{MIN}(50, (100 - \text{PCTFREE}))$
 다중 파티션) $100 * (\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE}) / ((\text{NLEAF} - \text{NUM_EMPTY_LEAFS} - \text{NPARTITIONS}) * (\text{INDEXPAGESIZE} - \text{LEAF_PAGE_OVERHEAD})) > \text{MIN}(50, (100 - \text{PCTFREE}))$
 F6: $(100 - \text{PCTFREE}) * ((\text{FLOOR}((100 - \text{LEVEL2PCTFREE}) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF_PAGE_OVERHEAD}) / (\text{NLEAF_RECSIZE} + \text{NLEAF_RECSIZE_OVERHEAD}))) * (\text{FLOOR}((100 - \text{MIN}(10, \text{LEVEL2PCTFREE})) / 100 * (\text{INDEXPAGESIZE} - \text{NLEAF_PAGE_OVERHEAD}) / (\text{NLEAF_RECSIZE} + \text{NLEAF_RECSIZE_OVERHEAD}))) * (\text{NLEVELS} - 3)) * (\text{INDEXPAGESIZE} - \text{LEAF_PAGE_OVERHEAD})) / (\text{KEYS} * (\text{LEAF_RECSIZE} + \text{LEAF_RECSIZE_OVERHEAD}) + (\text{INDCARD} - \text{KEYS}) * \text{DUPKEYSIZE})) < 100$
 F7: $100 * (\text{NUMRIDS_DELETED} / (\text{NUMRIDS_DELETED} + \text{INDCARD})) < 20$
 F8: $100 * (\text{NUM_EMPTY_LEAFS} / \text{NLEAF}) < 20$

Figure 1003B... 인덱스 재구성 필요 여부 판별식

판별식	설 명
F4	인덱스의 클러스터링 비율은 80% 를 초과해야 합니다.
F5	인덱스에 할당된 공간에서 비어 있는 공간의 총합은 50% 이하이어야 합니다.
F6	재구성 후에 인덱스 트리의 레벨이 한 단계 낮아질 수 있는지를 판별합니다.
F7	삭제된 키가 존재하는 페이지(non-pseudo-empty pages)에서 삭제된 키(pseudo-deleted RIDs)가 차지하는 총 공간은 20% 미만이어야 합니다.
F8	전체 리프 페이지 중에서 완전히 비어 있는 리프 페이지(pseudo-empty leaf pages)의 비율은 20% 미만이어야 합니다.

Point



REORGCHK 수행 결과 테이블 재구성을 권장 받으면, REORG TABLE 명령어를 이용하여 테이블의 데이터가 연속적인 페이지에 저장될 수 있도록 테이블을 재구성합니다. Inplace 옵션을 사용하면 Online 모드로 수행됩니다.

Tip

SYSADM, SYSCTRL, SYSMAINT, DBADM 권한 또는 테이블에 대한 CONTROL 특권을 가진 사용자가 실행합니다.

Tip

REORG TABLE 을 실행한 후에는 RUNSTATS, REBIND 를 실행하는 것이 권장됩니다.

Tip

작업 시에는 현재 데이터의 약 2.5배의 공간이 필요하므로 USE 옵션을 이용하여 충분한 크기의 임시 테이블 공간을 확보하는 것이 좋습니다.

Tip

임시 테이블스페이스의 페이지 크기는 재구성하는 테이블과 동일해야 합니다.

Tip

INPLACE 옵션을 사용하면, 임시 테이블 공간을 지정할 수 없습니다.

1

REORG TABLE 명령어로 테이블의 데이터를 물리적으로 재배포하고, 인덱스를 재 생성합니다.

```
>>REORG----->

>--+TABLE--table-name--| Table clause |-----+
+--+INDEXES ALL FOR TABLE--table-name-----+| Index clause |--+
| '-INDEX--index-name-----+'                  |
|                                     '-FOR TABLE--table-name-' |
|                                     .-ALLOW WRITE ACCESS-.    |
|                                     +--ALLOW READ ACCESS--+   |
|                                     '-ALLOW NO ACCESS-----'  |
```

Figure 1004A... REORG TABLE 명령어

2

테이블은 기본적으로 클러스터 인덱스를 기준으로 재구성됩니다. INDEX 옵션을 지정하면 특정 인덱스의 정렬 순서를 기준으로 재구성합니다. 클러스터 인덱스가 없거나 INDEX 옵션을 지정하지 않으면, 데이터는 임의의 순서로 재구성됩니다.

```
$ db2 reorg table <스키마명>.<테이블명> index <스키마명>.<인덱스명>
```

3

재구성할 테이블의 복사본을 저장할 임시 테이블스페이스를 지정합니다. 지정하지 않으면 테이블이 저장된 테이블스페이스에 복사본을 생성하게 됩니다.

```
$ db2 reorg table <스키마명>.<테이블명> USE <시스템 임시 테이블스페이스명>
```

4

테이블을 재구성하는 동안 다른 사용자는 테이블에 대한 액세스 범위를 제한 받습니다. 액세스 허용 범위는 ALLOW 옵션의 2가지의 모드로 조절합니다.

모드	설 명
ALLOW NO ACCESS	테이블을 액세스할 수 없습니다.
ALLOW READ ACCESS	읽기 액세스만 허용됩니다. 기본 모드입니다.
ALLOW WRITE ACCESS	읽기와 쓰기 액세스가 모두 허용됩니다. INPLACE 옵션을 사용한 온라인 재구성에서만 허용됩니다.

```
$ db2 reorg table <스키마명>.<테이블명>
```

```
$ db2 reorg table <스키마명>.<테이블명> ALLOW NO ACCESS
```

5

INPLACE 옵션을 사용하여 온라인 모드로 테이블을 재구성하면, 재구성이 진행되는 동안에 테이블의 쓰기 액세스를 허용할 수 있습니다. START, STOP, PAUSE, RESUME 옵션을 이용하여 온라인 재구성 작업을 중단시키고 옵션을 변경한 후에 다시 계속할 수 있습니다.

```
$ db2 reorg table <스키마명>.<테이블명> INPLACE ALLOW READ ACCESS
START
```

```
$ db2 reorg table <스키마명>.<테이블명> INPLACE PAUSE
```

```
$ db2 reorg table <스키마명>.<테이블명> inplace ALLOW WRITE ACCESS
NOTRUNCATE RESUME
```

```
$ db2 reorg table <스키마명>.<테이블명> INPLACE STOP
```

Point



REORGCHK 수행 결과 인덱스 재구성을 권장 받으면, REORG INDEXES 명령어를 이용하여 인덱스의 데이터가 연속적인 페이지에 저장될 수 있도록 인덱스만 재구성할 수 있습니다.

Tip

SYSADM, SYSCTRL, SYSMANT, DBADM 권한 또는 테이블에 대한 CONTROL 특권을 가진 사용자가 실행합니다.

Tip

REORG INDEXES 를 실행한 후에는 RUNSTATS, REBIND 를 실행하는 것이 권장됩니다.

Tip

한 페이지의 모든 데이터가 삭제되어 완전히 빈 페이지를 'committed pseudo empty page' 라고 하며, SYSCAT.INDEXES 카탈로그 뷰의 NUM_EMPTY_LEAFS 컬럼의 값으로 확인할 수 있습니다.

Tip

리프 페이지에서 삭제된 인덱스 데이터를 'committed pseudo deleted key' 라고 하며, 카탈로그 뷰인 SYSCAT.INDEXES 의 NUMRIDS_DELETED 컬럼의 값으로 확인할 수 있습니다.

Tip

버전8 은 기본적으로 type-2 인덱스를 생성합니다. 버전7 의 데이터베이스를 마이그레이션하여 사용하면, 기존 인덱스는 type-1 이 됩니다.

1

REORG INDEXES 명령어로 인덱스를 재 생성하거나, 데이터가 삭제된 공간을 반환합니다.

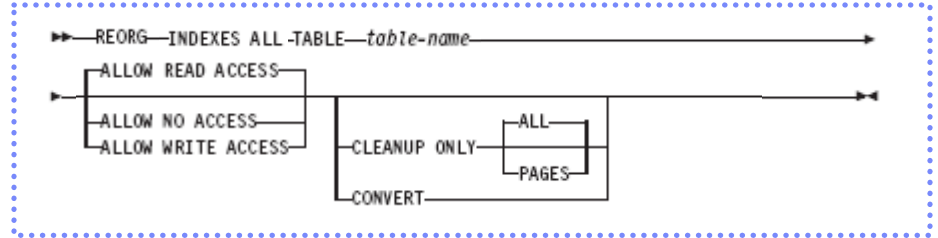


Figure 1005A... REORG INDEXES 명령어

2

인덱스를 재구성하는 동안 다른 사용자는 테이블에 대한 액세스 범위를 제한 받습니다. 액세스 허용 범위는 ALLOW 옵션의 3가지의 모드로 조절합니다.

모드	설명
ALLOW NO ACCESS	테이블을 액세스할 수 없습니다.
ALLOW READ ACCESS	읽기 액세스만 허용됩니다. 기본 모드입니다.
ALLOW WRITE ACCESS	읽기와 쓰기 액세스가 모두 허용됩니다.

```

$ db2 reorg indexes all for table <스키마명>.<테이블명>
$ db2 reorg indexes all for table <스키마명>.<테이블명> ALLOW NO ACCESS
$ db2 reorg indexes all for table <스키마명>.<테이블명> ALLOW WRITE ACCESS
  
```

3

인덱스를 재구성하는 작업은 실제로 인덱스를 재 생성하는 작업입니다. CLEANUP ONLY 옵션으로 인덱스를 재 생성하지 않고, 데이터가 삭제된 영역들을 동일한 테이블의 다른 인덱스가 사용할 수 있도록 할 수 있습니다. CLEANUP ONLY 옵션은 2 가지 모드를 지원합니다.

모드	설명
PAGES	한 페이지의 모든 데이터가 삭제되어 완전히 비어있는 페이지를 인덱스 트리에서 제거합니다.
ALL	완전히 비어있는 페이지를 인덱스 트리에서 제거하고, 일부 영역이 비어 있는 페이지는 인접 페이지와 병합하여 한 페이지를 빈 페이지로 만든 후 인덱스 트리에서 제거합니다. 기본 모드입니다.

```

$ db2 reorg indexes all for table <스키마명>.<테이블명> CLEANUP ONLY
$ db2 reorg indexes all for table <스키마명>.<테이블명> CLEANUP ONLY PAGES
  
```

4

CONVERT 옵션으로 type-1 인덱스를 type-2 인덱스로 변환할 수 있습니다.

```

$ db2 reorg indexes all for table <스키마명>.<테이블명> CONVERT
  
```

Point



DB2는 비용 기반의 옵티마이저를 이용하므로, RUNSTATS명령어를 이용하여 통계정보를 최근 것으로 유지하는 것이 중요합니다. 통계정보는 SYSSTAT 스키마를 가진 시스템 카탈로그 뷰에 그 정보가 저장됩니다.

Tip

SYSADM, SYSCTRL, SYSMAINT, DBADM, LOAD 권한 또는 테이블의 CONTROL 특권을 가진 사용자가 실행합니다.

Tip

RUNSTATS 로 통계 자료를 갱신한 후에는 REBIND를 실행하는 것이 권장됩니다.

Tip

SYSSTAT로 시작되는 뷰의 값이 -1로 된 항목은 통계 자료가 갱신되지 않았음을 의미합니다.

Tip

runstats가 실행되는 동안 목표 테이블에 INSERT가 발생하면 INDEX와의 불일치 경고 메시지가 반환될 수 있지만, 이는 무시해도 좋습니다.

Tip

수집된 대표적인 통계 자료는 다음과 같은 정보를 포함합니다.

- 테이블의 페이지 수
- 비어 있지 않은 페이지의 수
- 페이지 오버플로우의 정도
- 테이블의 행의 수
- 컬럼 내의 구별되는 값의 수
- 색인의 클러스터링 정도
- 색인 레벨의 수
- 각각의 색인에서 리프 페이지의 수
- 자주 사용된 컬럼 값의 발생 수
- 컬럼값의 분산
- 사용자 정의 함수에 대한 비용 예측

Tip

Sampling 중 BERNOULLI는 레코드단위인데 반해 SYSTEM은 페이지 단위로 샘플링하여 성능이 보다 우수합니다.

1

RUNSTATS 명령어로 수집된 데이터베이스에 대한 통계 자료를 시스템 카탈로그에 반영합니다.

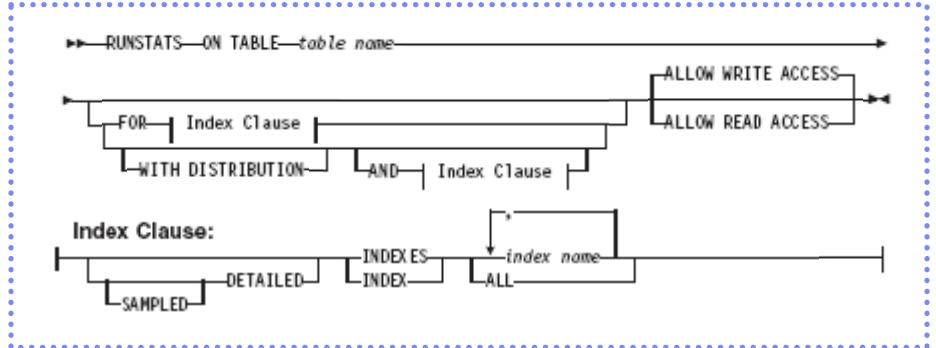


Figure 1006A... RUNSTATS 명령어

2

특정 테이블에 대한 통계 자료를 갱신합니다.

```
$ db2 runstats on table <스키마명>.<테이블명>
```

3

특정 테이블의 인덱스에 대한 통계 자료만 갱신할 때에는 FOR INDEXES ALL 또는 FOR INDEX 옵션을 사용합니다.

```
$ db2 runstats on table <스키마명>.<테이블명> FOR INDEXES ALL
$ db2 runstats on table <스키마명>.<테이블명> FOR INDEX <스키마명>.<인덱스명>
```

4

특정 테이블과 인덱스에 대한 통계 자료를 모두 갱신할 때에는 AND INDEX 옵션 또는 AND INDEXES ALL 옵션을 사용합니다.

```
$ db2 runstats on table <스키마명>.<테이블명> AND INDEXES ALL
$ db2 runstats on table <스키마명>.<테이블명> AND INDEX <스키마명>.<인덱스명>
```

5

DETAILED 옵션을 이용하면 인덱스에 대한 상세한 통계 자료를 갱신합니다.

```
$ db2 runstats on table <스키마명>.<테이블명> for DETAILED indexes all
$ db2 runstats on table <스키마명>.<테이블명> and DETAILED indexes all
```

6

WITH DISTRIBUTION 옵션으로 기본 통계 자료와 컬럼값의 분포에 대한 통계 자료를 함께 수집하여 시스템 카탈로그에 반영할 수 있습니다.

```
$ db2 runstats on table <스키마명>.<테이블명> WITH DISTRIBUTION
```

7

ALLOW READ ACCESS 옵션은 통계 자료를 갱신하는 동안 쓰기 액세스를 차단합니다.

```
$ db2 runstats on table <스키마명>.<테이블명> ALLOW READ ACCESS
```

8

대용량 테이블의 경우, 일부의 Page만 Sampling하여 통계정보를 수집합니다. (예제 1%)

```
$ db2 runstats on table <스키마명>.<테이블명> AND INDEXES ALL
TABLESAMPLE SYSTEM (1)"
```


Point



이전에 바인드 된 응용프로그램의 패키지에 저장된 액세스 플랜을 갱신하는 작업을 리바인드라고 합니다. RUNSTATS 명령어에 의해 갱신된 새로운 통계 자료를 사용하거나 추가적으로 작성된 인덱스를 이용하려면 리바인드 작업을 실행합니다.

Tip

동적 SQL에 대한 액세스 플랜은 시스템 통계 정보가 갱신되면, 실행 시에 자동으로 다시 작성됩니다. 정적 SQL은 BIND 또는 REBIND 명령어를 이용하여 명시적으로 액세스 플랜을 갱신하지 않으면, 기존의 액세스 플랜을 계속 유지하게 되므로 주의해야 합니다.

- 1 SQL문에 대한 액세스 플랜은 시스템 카탈로그의 패키지에 저장됩니다. list packages 명령어를 이용하여 패키지에 대한 현재 상태를 확인하면, VALID 항목에 패키지의 상태에 따라 Y(정상), N(무효), X(작동 불능) 중에 한 가지가 표시됩니다.

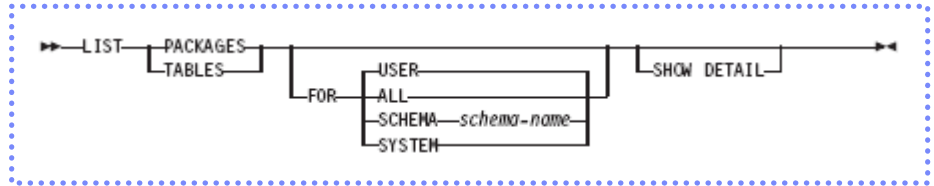


Figure 1007A... LIST PACKAGES 명령어

- 2 패키지 상태는 3가지로 분류되며, 정상 상태가 아닌 경우에는 리바인드가 필요합니다.

상태	설명
NORMAL	정상 상태입니다. 리바인드는 필요하지 않지만, RUNSTATS 명령어를 수행한 후 갱신된 통계를 사용하거나 새로 작성된 인덱스를 이용하고자 한다면 명시적으로 리바인드를 실행할 수 있습니다.
INVALID	패키지에 포함된 SQL문에서 참조했던 테이블, 뷰, 트리거 등의 데이터베이스 오브젝트가 삭제되면, 패키지는 무효 상태가 되어 리바인드 후에 사용할 수 있습니다. 리바인드는 해당 패키지가 다음 번에 실행될 때 데이터베이스 엔진에 의해 자동으로 실행됩니다.
INOPERATIVE	참조하고 있던 UDF가 삭제되면 패키지는 작동 불능 상태가 됩니다. 이 경우에는 반드시 BIND 명령어 또는 REBIND 명령어를 이용하여 사용자가 명시적으로 리바인드를 해야 합니다.

- 3 list packages 명령어를 이용하여 패키지에 대한 현재 상태를 확인하면, VALID 항목에 패키지의 상태에 따라 Y(정상), N(무효), X(작동 불능) 중에 한 가지가 표시됩니다.

\$ db2 list packages

Package	Schema	Version	Bound by	Total sections	Valid	Format	Isolation level	Blocking
F4INS	USERA	VER1	SNOWBELL	221	Y	0	CS	U
F4INS	USERA	VER2.0	SNOWBELL	201	Y	0	RS	U
F4INS	USERA	VER2.3	SNOWBELL	201	N	3	CS	U
F4INS	USERA	VER2.5	SNOWBELL	201	Y	0	CS	U
PKG12	USERA		USERA	12	Y	3	RR	B
PKG15	USERA		USERA	42	Y	3	RR	B
SALARY	USERT	YEAR2000	USERT	15	Y	3	CS	N

Figure 1007B... 패키지의 상태 확인

Point



명시적으로 리바인드를 요청하는 명령어로써 시스템 카탈로그에 저장되어 있는 기존의 특정한 패키지를 갱신합니다. 패키지에 저장된 액세스 플랜은 새로운 통계 자료를 기준으로 다시 생성됩니다.

Tip

- SYSADM, DBADM 권한 또는 스키마에 대한 ALTERIN, 패키지에 대한 BIND 특권을 가진 사용자가 실행합니다.

Tip

- RESOLVE CONSERVATIVE 옵션은 Inoperative 패키지에 대해서는 지원되지 않습니다.

- 1 최신의 데이터베이스에 대한 통계 자료를 기반으로 기 작성된 패키지의 액세스 플랜을 갱신합니다.

```
>>-REBIND-----package-name----->
      '-PACKAGE-'

>-----RESOLVE-----ANY----->
      '-VERSION--version-name-'      '-CONSERVATIVE-'

>-----APREUSE-----YES----->
      '-NO--'      '+-REOPT NONE---+
                  +-REOPT ONCE---+
                  +-REOPT ALWAYS-->
```

Figure 1008A... REBIND 명령어

- 2 옵션에 대한 설명은 아래와 같습니다.

모드	설명
RESOLVE ANY	모든 오브젝트를 참조하여, 비 제한적 바인딩으로 수행됩니다.
RESOLVE CONSERVATIVE	이전 정의된 오브젝트만 참조하여, 제한적 바인딩으로 수행됩니다.
APREUSE YES	패키지 내에서 정적 SQL Plan이 재 사용됩니다
APREUSE NO	패키지 내에서 정적 SQL Plan이 재 사용되지 않습니다.
REOPT NONE	액세스 플랜을 결정할 때 호스트 변수, 매개 변수 표시 문자, 특수 레지스터 변수에 할당된 실제값을 이용하지 않고, 미리 결정된 기본 추정값 (Default Filtering Factor)를 이용합니다. 결정된 액세스 플랜은 캐쉬에 저장되어 계속 사용됩니다. REOPT 옵션을 지정하지 않는 경우에 기본 모드로 제공됩니다.
REOPT ONCE	쿼리가 처음 실행되었을 때 호스트 변수, 매개 변수 표시 문자, 특수 레지스터 변수에 할당된 실제값을 이용하여 액세스 플랜을 결정합니다. 결정된 액세스 플랜은 캐쉬에 저장되어 계속 사용됩니다.
REOPT ALWAYS	쿼리가 실행될 때마다 호스트 변수, 매개 변수 표시 문자, 특수 레지스터 변수에 할당된 실제값을 이용하여 SQL문을 재컴파일하고, 액세스 플랜을 재결정합니다.

- 3 rebind 명령어를 이용하여 패키지를 갱신합니다.

```
$ db2 rebind <스키마명>.<패키지명> RESOLVE ANY REOPT ONCE
```

- 4 db2rbind 명령어는 특정 데이터베이스에 있는 모든 패키지의 액세스 플랜을 한꺼번에 갱신합니다. <로그파일명>은 임의로 지정합니다.

```
$ db2rbind <데이터베이스명> /I <로그파일명> all
```