UNIT 11

동시성 제어

동시성이란 다수의 사용자가 동시에 데이터베이스 오브젝트를 액세스할 수 있도록 허용하는 메커니즘입니다. DB2는 테이블과 행 수준의 잠금을 이용하여 동시성과 데이터의 무결성을 함께 보장합니다.

RR, RS, CS, UR, CC 등의 분리 수준으로 데이터 조회에 적용되는 잠금을 조절합니다.

DB2 9.7 운영자 가이드

Administrator Edition

- 동시성과 무결성
- 분리 수준
- 분리 수준 지정 방법
- 잠금의 대상
- 테이블 잠금 유형
- 테이블 잠금 지정 방법
- 행 잠금 유형
- 잠금 변환 현상
- 잠금 상승 현상
- 잠금 대기 현상
- 교착 상태

TOPIC 1101 동시성과 무결성

Point

동시성이란 다수의 사용자가 동시에 데이터베이스 오브젝트를 액세스할 수 있도록 허용하는 매커니즘입니다. 데이터베이스 엔진은 동시성을 지원하면서 데이터의 무결성을 보장할 수 있어야 합니다.

Tip

실제 응용프로그램의 구현 시에는 업무 특성에 따라 바람직하지 않은 현상을 의도적으로 허용할 수도 있습니다.

1 여러 사용자가 동시에 데이터베이스를 액세스할 때에는 다음과 같은 데이터베이스의 무결성을 위반하는 바람직하지 않은 현상이 발생될 수 있습니다.

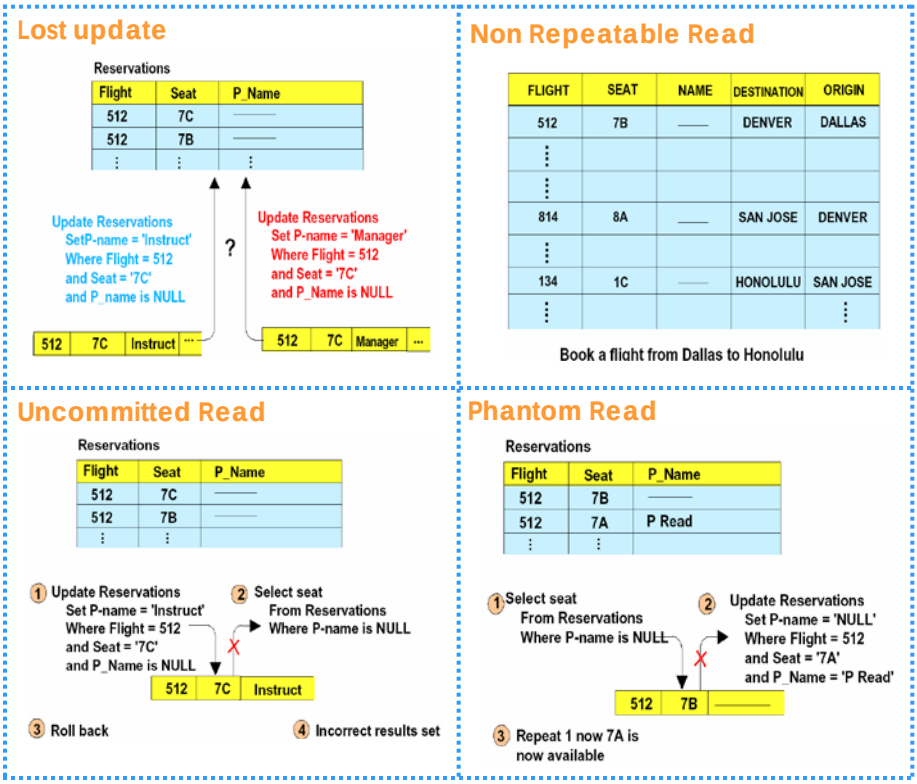

Figure 1101A ••• 동시성을 지원할 때 발생하는 현상

2 A와 B라는 사용자가 동일한 데이터를 액세스한다면, 다음과 같은 현상이 발생할 수 있습니다.

현 상	설 명
갱신 유실 Lost Update	A와 B가 동일한 데이터를 동시에 조회하고 변경하면, A가 변경한 값은 유실되고, B가 변경한 값이 남게 됩니다.
미확약 읽기 Uncommitted Read	A가 추가 또는 갱신한 후에 COMMIT 하지 않은 데이터를 B가 조회할 수 있도록 허용할 때, A가 commit으로 트랜잭션을 종료되었다면, B가 조회한 데이터는 정확한 데이터가 되지만, A가 rollback으로 트랜잭션을 종료한다면, B가 이미 조회한 데이터는 잘못된 데이터가 됩니다.
반복 읽기 불가능 Non Repeatable Read	A가 SELECT문을 실행하여 결과 집합을 조회하고, 동일한 트랜잭션에서 동일한 조건의 SELECT 문을 다시 요청하면 이전의 결과 집합이 그대로 반환되지 않습니다.
팬텀 읽기 Phantom Read	A가 SELECT문을 실행하여 10 건의 결과 집합을 조회하고, 동일한 트랜잭션에서 동일한 조건의 SELECT 문을 다시 요청하면 이전의 결과 집합인 10건은 그대로 반환되고, B가 추가한 1 건의 행이 추가적으로 반환됩니다. B가 새로 추가한 행을 팬텀 행이라고 합니다.

TOPIC 11-02 분리 수준

Point

한 응용프로그램이 SELECT문을 실행하여 결과 집합이 반환되면, 다른 응용프로그램은 해당 결과 집합에 영향을 주는 액세스를 제한 받게 됩니다. 액세스가 제한되는 대상의 범위와 SQL문의 종류는 분리 수준에 의해 조절됩니다.

Tip

- 분리 수준은 SELECT문에 대해서만 적용됩니다. 데이터를 변경시키는 INSERT,UPDATE,DELETE 문에 의해 잠금이 적용된 행은 COMMIT 또는 ROLLBACK이 될 때까지 액세스가 제한됩니다.

1 분리 수준은 다음과 같이 4 가지로 구분됩니다.

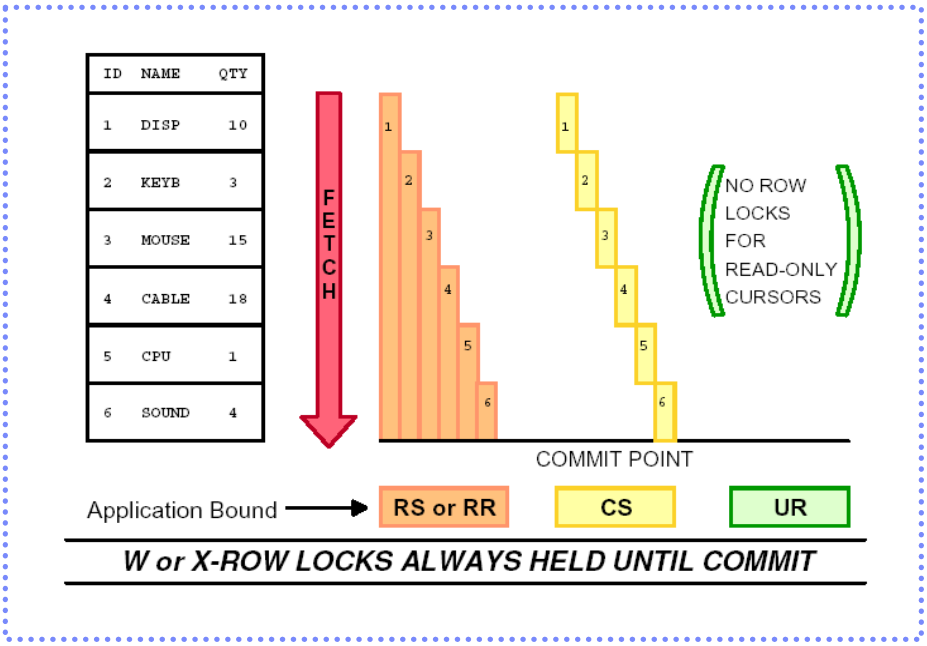

Figure 1102A • • • 분리 수준별 잠금 대상의 범위와 유지 기간

2 분리 수준의 유형별 의미는 다음과 같습니다.

Tip

- CC는 DB2 9.7에서 새롭게 추가되었습니다.

유형	설명
CC	Currently Committed의 약자입니다. 조회하는 UOW는 동시에 변경하는 다른 사용자에 의해 변경되고 있더라도 수행에 방해를 받지 않고 즉시 결과를 리턴 받게 됩니다. DB2 9.7에서 기본 분리 수준입니다.
CS	Cursor Stability의 약자입니다. 한 UOW 에서 동일한 조건의 SELECT문을 여러 번 실행하면 이전의 결과 집합이 반환되는 것을 보장하지 않습니다. 현재 처리 중인 한 개의 행에 대해서만 NS 잠금을 설정합니다. 다른 응용프로그램은 결과 집합의 데이터를 추가, 변경, 삭제할 수 있습니다. DB2 9.5까지 기본 분리수준 입니다.
RR	Repeatable Read 의 약자입니다. 한 UOW 에서 동일한 조건의 SELECT문을 여러 번 실행하면, 이전의 결과 집합과 항상 동일한 결과 집합이 반환되는 것을 보장하므로 반복 읽기가 가능합니다. 반환된 결과 집합의 모든 행과 결과 집합을 추출하기 위해 액세스된 모든 행에 대해 S 잠금을 적용합니다. 다른 응용프로그램은 결과 집합의 데이터를 추가, 변경, 삭제할 수 없습니다.
RS	Read Stability의 약자입니다. 한 UOW 에서 동일한 조건의 SELECT문을 여러 번 실행하면, 이전의 결과 집합을 항상 포함하고, 팬텀 행이 추가적으로 반환되는 것을 허용합니다. 반환된 결과 집합의 모든 행에 NS 잠금을 적용합니다. 다른 응용프로그램은 결과 집합의 데이터를 추가는 할 수 있지만, 변경과 삭제는 할 수 없습니다.
UR	Uncommitted Read의 약자입니다. 한 UOW 에서 동일한 조건의 SELECT문을 여러 번 실행하면 이전의 결과 집합이 반환되는 것을 보장하지 않습니다. 잠금을 전혀 적용하지 않습니다. 다른 응용프로그램은 결과 집합의 데이터를 추가, 변경, 삭제할 수 있습니다. 다른 응용프로그램에서 처리 중인 UOW의 COMMIT 되지 않은 데이터를 조회할 수 있고, COMMIT 하지 않은 데이터가 액세스되는 것을 허용합니다.

TOPIC 1103 분리 수준 지정 방법

Point

분리 수준은 개별 SQL문, 세션, 패키지, DB2 CLI 에서 지정할 수 있습니다. 개별 SQL문에서 WITH 옵션으로 분리 수준을 지정하는 것이 가장 우선적으로 적용됩니다.

Tip

WITH UR 을 지정하면, X 잠금이 설정된 데이터도 액세스할 수 있습니다.

1 <분리 수준>을 지정할 때는 RR, RS, CS, UR 이라는 키워드를 이용합니다. 대소문자는 구별하지 않습니다.

2 분리 수준의 유형을 지정하는 방법은 다음과 같습니다.

대 상	설 명
SQL문	SQL문의 마지막에 WITH 옵션을 사용하여 지정합니다. 패키지 수준과 세션 수준의 설정값보다 우선적으로 적용됩니다. SELECT 문, SELECT INTO 문, INSERT 문, FOR 문, searched DELETE 문, searched UPDATE 문, DECLARE CURSOR 문 등은 WITH 옵션을 사용할 수 있습니다.
현재 세션	SET CURRENT ISOLATION 문을 이용하여 지정합니다. 패키지에 저장된 동적 SQL문에 대해서 패키지 수준의 설정값보다 우선적으로 적용됩니다.
패키지	PREP 명령어의 ISOLATION 옵션, BIND 명령어의 ISOLATION 옵션으로 지정하며, 패키지에 포함된 모든 SELECT문에 적용됩니다.
DB2 CLI	CHANGE ISOLATION LEVEL 명령어, SQLSetConnectAttr 함수, db2cli.ini 파일의 TXNISOLATION 키워드를 이용하여 지정하며, CLI 와 ODBC 드라이버를 통한 실행시에 적용됩니다.
DB2 CLP	CHANGE ISOLATION LEVEL 명령어를 이용하며, 데이터베이스의 재접속시에 반영됩니다.

Tip

패키지에 저장된 SELECT문의 분리 수준이 패키지 수준의 분리 수준과 다른 경우에는 해당 SELECT문에만 적용됩니다.

3 다양한 환경에서 분리 수준을 UR로 설정하는 예는 다음과 같습니다.

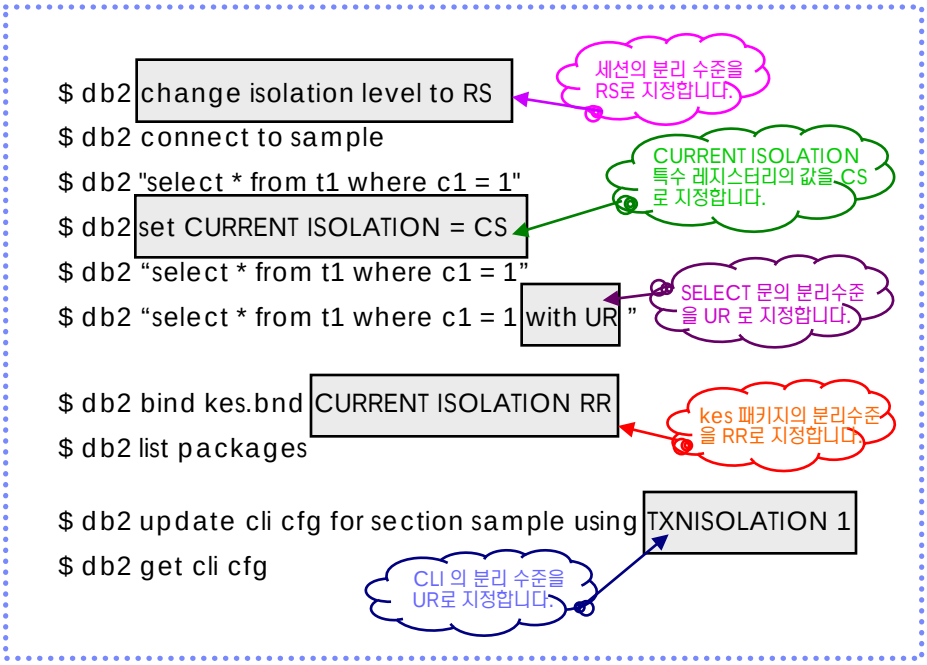

Figure 1103A ••• 다양한 환경에서 분리 수준을 UR로 설정하는 방법

Tip

CLI 구성 파일인 db2cli.ini 파일에 저장되는 TXNISOLATION 키워드의 값은 1(UR), 2(CS), 4(RS), 8(RR) 중에서 한 가지로 지정합니다.

TOPIC 1104 잠금의 대상

Point

QUIESCE 명령어, CONNECT 명령어, LOCK TABLE문으로 인스턴스, 데이터베이스, 테이블스페이스, 테이블에 사용자가 명시적으로 잠금을 적용할 수 있습니다. SQL문을 실행하면 엔진에 의해 테이블과 인덱스의 행에 자동으로 잠금이 적용됩니다.

Tip

인스턴스, 데이터베이스, 테이블스페이스에 대한 잠금은 데이터베이스 관리 차원의 잠금입니다.

Tip

일반적인 응용프로그램에서 SQL문으로 데이터를 액세스하므로 테이블, 행, 인덱스에 대한 잠금이 적용됩니다.

1 잠금이 적용되는 데이터베이스 오브젝트는 다음과 같습니다.

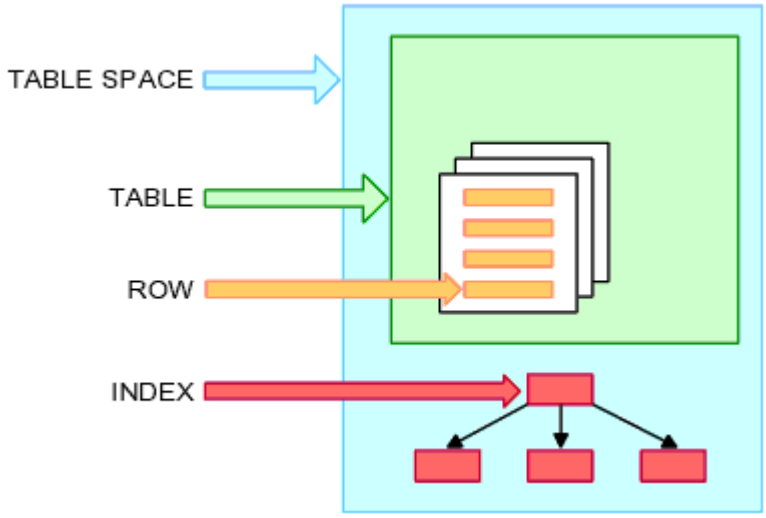

Figure 1104A ••• 잠금이 적용되는 데이터베이스 오브젝트

2 잠금을 적용하는 방법은 다음과 같습니다.

대 상	설 명
인스턴스	QUIESCE INSTANCE 명령어를 실행하면, SYSADM / SYSCTRL / SYSMAINT 권한을 가진 사용자와 액세스가 허용된 사용자만 인스턴스와 데이터베이스를 액세스할 수 있습니다.
데이터베이스	CONNECT 문을 이용하여 데이터베이스에 접속할 때 배타적 모드의 접속이 가능합니다. 기본 접속 모드는 공유적 모드이므로 여러 사용자가 동시에 접속할 수 있습니다. QUIESCE DATABASE 명령어를 실행하면, SYSADM / SYSCTRL / SYSMAINT / DBADM 권한을 가진 사용자와 액세스가 허용된 사용자만 데이터베이스를 액세스할 수 있습니다.
테이블스페이스	QUIESCE TABLESPACES FOR TABLE 명령어를 실행하면, LOAD 명령어를 실행하는 동안 해당 테이블스페이스에 잠금을 적용합니다.
테이블	LOCK TABLE 문을 이용하여 배타적 또는 공유적 모드로 테이블 전체에 잠금을 설정할 수 있습니다. 한 테이블의 행 잠금의 비율이 지정한 기준을 초과하면, 내부적으로 행잠금이 테이블 잠금으로 변환될 수도 있습니다.
행	SQL문을 이용하여 데이터를 액세스하면 기본적으로 행 수준의 잠금이 적용됩니다.
인덱스	SQL문을 이용하여 데이터를 액세스할 때, 특정한 인덱스를 이용하여 액세스하게 되면, 해당 인덱스의 행에 잠금이 적용됩니다.

TOPIC 1105 테이블 잠금의 유형

Point

테이블 잠금의 유형에는 IN, IS, IX, SIX, S, U, X, Z 등이 있습니다. 테이블에 잠금이 적용되면, 행에 대한 잠금 정보는 별도로 관리되지 않습니다. 테이블의 모든 행에 대해서 액세스가 유형별로 제한됩니다.

1 테이블 잠금 유형과 잠금 유형간의 호환표은 다음과 같습니다.

IN	Intent None
IS	Intent Share
IX	Intent eXclusive
S IX	Share with Intent eXclusive
S	Share
U	Update
X	eXclusive
Z	superexclusive

MODE OF LOCK A	MODE OF LOCK B							
	IN	IS	S	IX	SIX	U	X	Z
IN	YES	YES	YES	YES	YES	YES	YES	NO
IS	YES	YES	YES	YES	YES	YES	NO	NO
S	YES	YES	YES	NO	NO	YES	NO	NO
IX	YES	YES	NO	YES	NO	NO	NO	NO
SIX	YES	YES	NO	NO	NO	NO	NO	NO
U	YES	YES	YES	NO	NO	NO	NO	NO
X	YES	NO	NO	NO	NO	NO	NO	NO
Z	NO	NO	NO	NO	NO	NO	NO	NO

Figure 1105A ··· 테이블 잠금 모드와 호환표

Tip

분리 수준 UR을 사용하면 Z 잠금을 제외한 모든 잠금을 무시하고 액세스할 수 있습니다.

Tip

IS는 테이블의 일부 행들이 SELECT 문에 의해 S 또는 NS 잠금이 설정되었다는 것을 알려주기 위해 테이블에 적용되는 지시성 잠금 유형입니다.

Tip

IX는 테이블의 일부 행들이 INSERT, UPDATE, DELETE문에 의해 X 또는 W 잠금이 설정되었다는 것을 알려주기 위해 테이블에 적용되는 지시성 잠금 유형입니다.

2 테이블 잠금 유형의 특성은 다음과 같습니다.

모 드	설 명
IN	잠금 소유자는 언커미트된 데이터를 포함하여 오브젝트의 어떤 데이터든지 읽을 수 있지만, 갱신할 수는 없습니다. 다른 동시 응용프로그램은 테이블을 읽거나 갱신할 수 있습니다. 분리 수준이 UR로 설정된 경우에 적용됩니다. IN, IS, S, IX, SIX, U, X 잠금과 호환됩니다.
IS	잠금 소유자는 잠긴 테이블에서 데이터를 읽을 수는 있지만, 이 데이터를 갱신할 수는 없습니다. 다른 응용프로그램은 테이블을 읽거나 갱신할 수 있습니다. IN, IS, S, IX, SIX, U 잠금과 호환됩니다.
IX	잠금 소유자 및 동시 응용프로그램은 데이터를 읽고 갱신할 수 있습니다. 다른 동시 응용프로그램은 테이블을 읽고 갱신할 수 있습니다. IN, IS, IX 잠금과 호환됩니다.
SIX	잠금 소유자는 데이터를 읽고 갱신할 수 있습니다. 다른 동시 응용프로그램은 테이블을 읽을 수 있습니다. 이미 IX 모드가 적용된 테이블에 대해 S 잠금을 적용해야 하는 경우에 사용됩니다. IN, IS 잠금과 호환됩니다.
S	잠금 소유자와 모든 동시 응용프로그램은 잠긴 데이터를 읽을 수는 있지만, 갱신할 수는 없습니다. IN, IS, S, U 잠금과 호환됩니다.
U	잠금 소유자는 데이터를 갱신할 수 있습니다. 다른 UOW는 잠긴 오브젝트에서 데이터를 읽을 수는 있지만, 갱신할 수는 없습니다. FOR UPDATE 옵션이 있는 커서를 이용한 SELECT문을 실행한 경우에 적용됩니다. IN, IS, S 잠금과 호환됩니다.
X	잠금 소유자만 데이터를 읽고 갱신할 수 있습니다. IN 잠금과 호환됩니다.
Z	테이블이 변경되거나 삭제된 경우, 테이블의 인덱스가 작성되거나 삭제되는 경우에 적용되며, 다른 동시 응용프로그램은 테이블을 읽거나 갱신할 수 없습니다.

TOPIC 1106 테이블 잠금 지정 방법

Point

데이터를 액세스하면 기본적으로 행 수준의 잠금이 적용됩니다. 특정 테이블의 행의 수가 많고, 대부분의 행을 액세스한다면, 행 잠금을 관리하기 위한 리소스가 많이 소모되어 응답 시간이 지연될 수 있으므로 명시적으로 테이블 잠금을 지정하는 것이 유리합니다.

Tip

기본 잠금 수준을 테이블 잠금으로 높이면 동시성이 저하시키는 단점이 있지만, 잠금을 관리하기 위한 비용이 줄어들 수 있으므로 실제적인 응답 시간은 빨라질 수 있습니다.

1 데이터의 기본 잠금 수준은 행 잠금입니다. 기본 잠금 수준을 테이블 잠금으로 지정하면, SQL문을 실행할 때 항상 테이블 잠금이 적용됩니다.

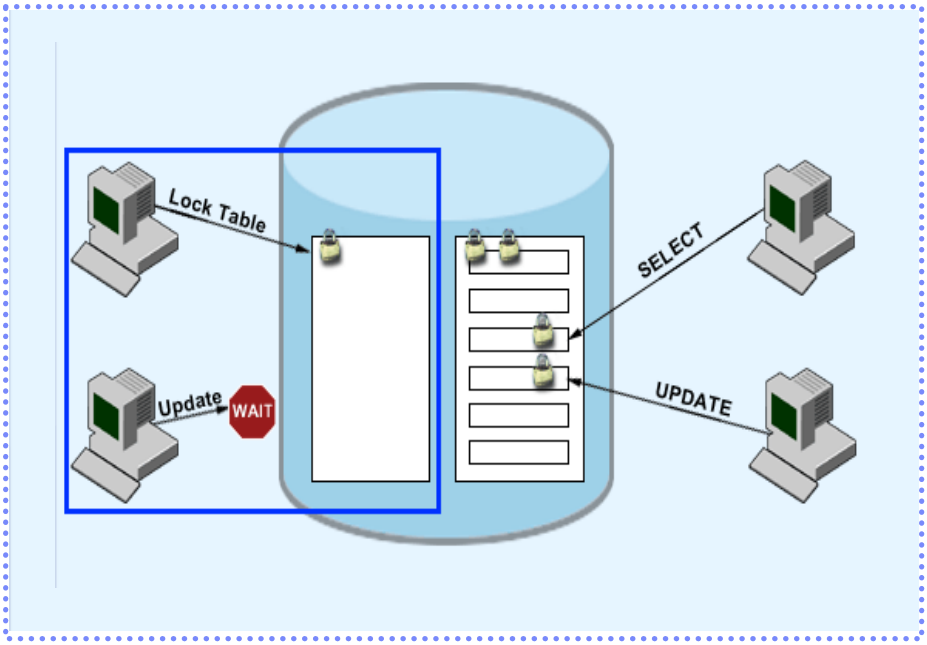

Figure 1106A • • • 명시적인 테이블 잠금

2 alter table 문에서 LOCKSIZE 옵션을 이용하여 지정합니다.

```
$ db2 "alter table <테이블명> LOCKSIZE TABLE"
$ db2 "alter table <테이블명> LOCKSIZE ROW"
```

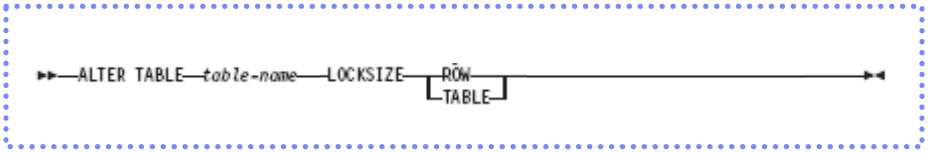

Figure 1106B • • • ALTER TABLE 문의 LOCKSIZE 옵션

3 lock table 문에서 LOCKSIZE 옵션을 이용하여 지정합니다.

```
$ db2 "lock table <테이블명> in SHARE mode"
$ db2 "lock table <테이블명> in EXCLUSIVE mode"
```

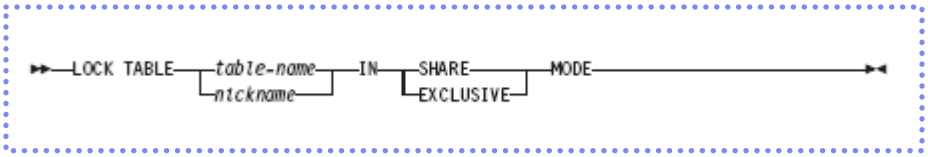

Figure 1106C • • • LOCK TABLE 문

TOPIC 1107 행 잠금 유형

Point

행 잠금의 유형에는 S, U, X, W, NS, NX, NW 등이 있습니다. 행에 잠금이 적용되면, 테이블에는 지시성 잠금이 적용됩니다. 잠금이 적용된 행에 대해서 액세스가 유형별로 제한됩니다.

Tip

S, NS 행 잠금이 적용된 테이블에는 자동으로 IS 테이블 잠금이 적용됩니다.

Tip

U, X, W, NW 행 잠금이 적용된 테이블에는 자동으로 IX 테이블 잠금이 적용됩니다.

Tip

분리 수준 UR을 사용하는 응용프로그램만은 어떤 유형의 잠금이라도 무시하고 액세스할 수 있습니다.

Tip

데이터를 조회한 후에 해당 데이터를 변경하는 로직을 가진 응용프로그램을 여러 사용자가 동시에 실행하려면 FOR UPDATE 옵션이 있는 커서를 사용하여 교착 상태가 발생하는 것을 방지하는 것이 좋습니다.

1 행 잠금 유형과 잠금 유형간의 호환표은 다음과 같습니다.

Row Lock		Minimum Supporting Table Lock
S	Share	IS
U	Update	IX
X	eXclusive	IX
W	Weak exclusive	
NS	Next key Share	
NW	Next key Weak exclusive	

LOCK A MODE	MODE OF LOCK B					
	S	U	X	W	NS	NW
S	YES	YES	NO	NO	YES	NO
U	YES	NO	NO	NO	YES	NO
X	NO	NO	NO	NO	NO	NO
W	NO	NO	NO	NO	NO	YES
NS	YES	YES	NO	NO	YES	YES
NW	NO	NO	NO	YES	YES	NO

Figure 1107A ••• 행 잠금 모드와 호환표

2 행 잠금 유형의 특성은 다음과 같습니다.

모 드	설 명
S	잠금 소유자와 모든 동시 응용프로그램은 잠긴 데이터를 읽을 수는 있지만, 갱신할 수는 없습니다. 분리 수준이 RR인 SELECT문을 실행한 경우에 조건에 맞는 행과 액세스에 사용된 모든 행에 대해 적용됩니다. S, U, NS 잠금과 호환됩니다.
U	잠금 소유자는 데이터를 갱신할 수 있습니다. 다른 UOW는 잠긴 오브젝트에서 데이터를 읽을 수는 있지만, 갱신할 수는 없습니다. FOR UPDATE 옵션이 있는 커서를 이용한 SELECT문을 실행하여 FETCH한 경우에 적용됩니다. S, NS 잠금과 호환됩니다.
X	잠금 소유자는 잠긴 오브젝트의 데이터를 읽고 갱신할 수 있습니다. UPDATE, INSERT, DELETE문을 실행한 경우에 적용됩니다. 다른 응용프로그램들은 X 잠금이 적용된 행에 대한 액세스를 할 수 없습니다.
W	type-2 인덱스가 없는 테이블로 행이 삽입되는 경우에 적용됩니다. 잠금 소유자는 잠긴 행을 변경할 수 있습니다. 중복 값 발견시 중복 값을 커미트했는지 판별하기 위해, 고유한 인덱스로의 삽입 중에도 이 잠금이 사용됩니다. NW 잠금과 호환성이 되는 것을 제외하고 이 잠금은 X 잠금과 유사합니다. NW잠금과 호환됩니다.
NS	잠금 소유자와 모든 동시 응용프로그램은 잠긴 행을 읽을 수는 있지만, 갱신할 수는 없습니다. S 잠금과 유사하며, 분리 수준이 RS 또는 CS인 경우에 적용됩니다. S, U, NS, NW 잠금과 호환됩니다.
NW	행이 인덱스에 삽입되면, NW 잠금은 다음 행에서 획득됩니다. type-2 인덱스의 경우, 다음 행이 현재 RR 스캔에 의해 잠긴 경우에만 발생합니다. 잠금 소유자는 잠긴 행을 읽을 수는 있지만, 갱신할 수는 없습니다. 이 잠금 모드는 W 및 NS 잠금과 호환되는 것을 제외하고는 X 잠금과 유사합니다. W, NS 잠금과 호환됩니다.

TOPIC 1108 잠금 변환 현상

Point

이미 잠금이 적용된 행 또는 테이블에 다시 잠금이 요청되면, 기존의 잠금 정보와 수위를 비교합니다. 기존의 잠금이 더 제한적이라면 새로운 잠금 정보는 무시됩니다. 새로운 잠금이 더 강력하다면 기존의 잠금 정보는 새로운 잠금 정보로 변환됩니다.

Tip

테이블 잠금과 행 잠금의 변환은 개별적으로 적용됩니다.

1 테이블 잠금과 행 잠금의 단계는 다음과 같습니다. 하위 잠금이 적용된 상태에서 상위 잠금이 요청되면, 하위 잠금 정보는 상위 잠금 정보로 갱신됩니다.

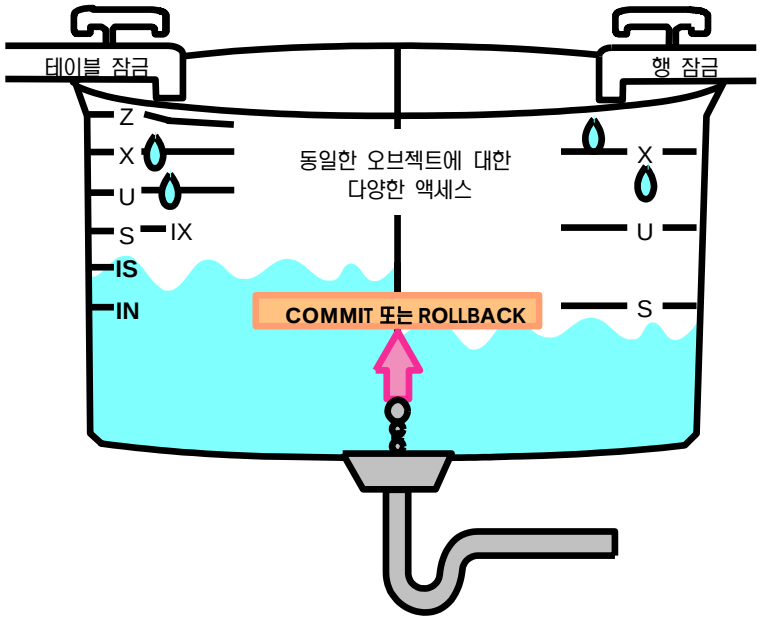

Figure 1108A ••• 잠금 변환

2 SELECT문의 조건문에서 고유한 인덱스가 있는 컬럼을 이용하여 데이터를 조회하면, 해당 행에 대해 NS 또는 S 모드의 잠금이 적용됩니다. 동일한 행에 대해 UPDATE 문을 다시 실행하면, X 잠금이 적용되고, LOCKLIST에 저장된 해당 행에 대한 NS 또는 S 모드의 잠금 정보는 X 모드의 잠금으로 갱신됩니다.

```
SELECT   *
FROM     kes.test_taken
WHERE    seat_no = '1'
AND      date_take = CURRENT DATE
WITH RR;
```

```
UPDATE   kes.test_taken
SET      seat_no = '2'
WHERE    seat_no = '1'
AND      date_take = CURRENT DATE;
```

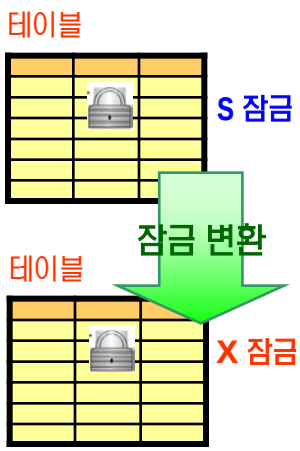

Figure 1108B ••• 행 잠금 유형의 변환

TOPIC 1109 잠금 상승 현상

Point

행잠금이 테이블 잠금으로 전환되는 현상을 잠금 상승 현상이라고 합니다. 잠금 상승 현상은 교착 상태를 유발시킬 수 있으며, 동시성을 저하시키므로, 가급적이면 발생하지 않도록 데이터베이스 구성 변수인 LOCkLIST와 MAXLOCKS을 조절합니다.

Tip

LOCKLIST 는 '온라인 구성 가능 변수' 이므로 동적으로 변경할 수 있습니다.

Tip

LOCKLIST를 AUTOMATIC으로 설정하면 자동으로 DBMS가 크기를 조정합니다.

Tip

MAXLOCKS 는 '온라인 구성 가능 변수' 이므로 동적으로 변경할 수 있습니다.

1. update db cfg 명령어로 데이터베이스 구성 변수인 LOCKLIST 구성 변수를 이용하여 잠금 정보를 저장하는 메모리 영역의 크기를 조절합니다. <크기>는 4K 페이지 단위로 지정합니다.

```
$ db2 update db cfg for <데이터베이스명> using LOCKLIST <크기>
```

2. 잠금 정보는 데이터베이스별로 공유 메모리에 있는 LOCKLIST 영역에 저장됩니다. 잠금 정보의 양이 LOCKLIST의 크기를 초과하게 되면, 가장 많은 행 잠금을 가진 테이블을 식별하여 그 테이블에 대한 행 잠금을 모두 취소하고 테이블 잠금을 적용합니다.

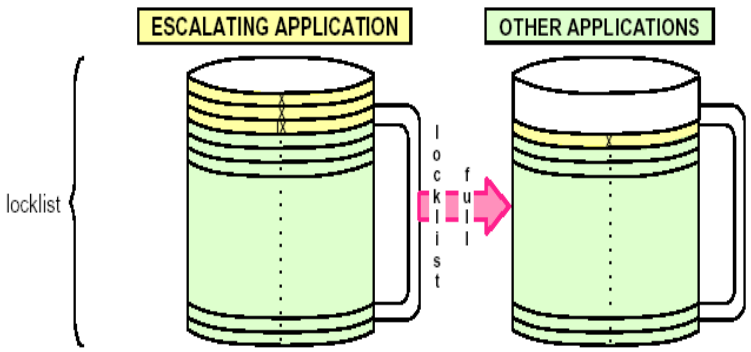

Figure 1109A ••• LOCKLIST 구성 변수와 잠금 상승 현상

3. MAXLOCKS 구성 변수는 LOCKLIST 구성 변수가 지정한 크기에서 한 응용프로그램이 사용할 수 있는 최대 비율을 지정합니다. <비율>은 백분율로 표시합니다.

```
$ db2 update db cfg for <데이터베이스명> using MAXLOCKS <비율>
```

4. 한 응용프로그램에 대한 잠금 정보의 양이 MAXLOCKS의 비율을 초과하게 되면, 해당 응용프로그램에서 가장 많은 행 잠금을 가진 테이블을 식별하여 그 테이블에 대한 행 잠금을 모두 취소하고, 테이블 잠금을 적용합니다.

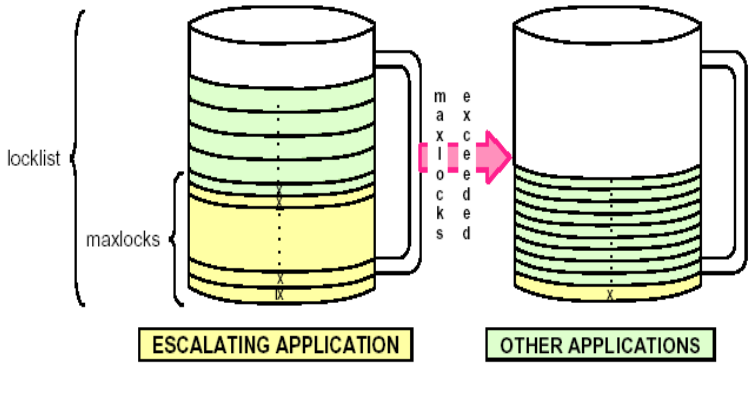

Figure 1109B ••• MAXLOCKS 구성 변수와 잠금 상승 현상

TOPIC 11-10 잠금 대기 현상

Point

한 응용프로그램이 잠금을 적용한 행을 다른 응용프로그램에서 액세스하려면, 그 잠금이 해제될 때까지 대기해야 합니다. 기본 잠금 대기 시간은 무한대이며, 데이터베이스 구성 변수인 LOCKTIMEOUT 을 이용하여 대기 시간을 조절할 수 있습니다.

Tip

LOCKTIMEOUT 구성 변수의 변경값을 반영하려면 데이터베이스의 재활성화가 필요합니다.

1. update db cfg 명령어로 LOCKTIMEOUT 데이터베이스 구성 변수를 설정합니다. <잠금 대기 시간>은 1초 단위로 표시합니다.

```
$ db2 update db cfg for <데이터베이스명> using LOCKTIMEOUT <잠금 대기 시간>
```

2. LOCKTIMEOUT 데이터베이스 구성 변수의 기본값은 -1로서 무한대로 대기하는 것을 의미합니다. 값을 0 으로 설정하면, SQL문 요청 시점에서 잠금을 획득하지 못하면 즉시 중단되게 합니다. 일반적인 OLTP 환경에서는 잠금 대기 시간을 30초 이내로 설정하도록 합니다. 잠금 대기 시간 이내에 필요한 잠금을 획득하면, 응용프로그램은 작업을 계속할 수 있습니다.

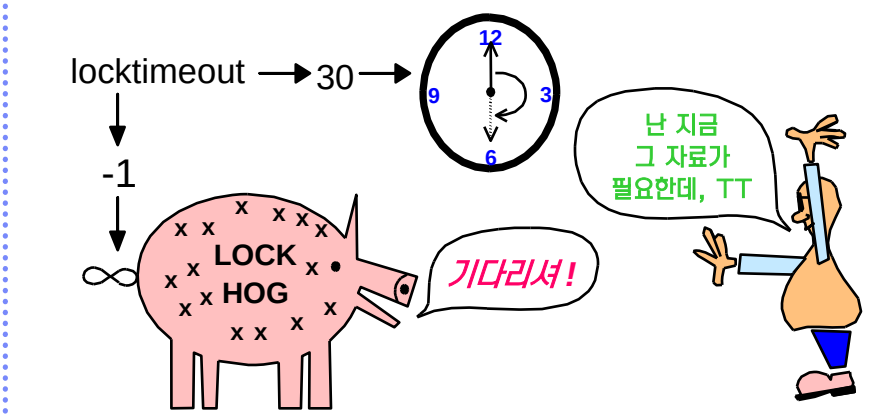

Figure 1110A • • • LOCKTIMEOUT 구성 변수

3. LOCKTIMEOUT 구성 변수의 값을 초과할 때까지 필요한 잠금을 획득하지 못하면, 응용프로그램은 중단되어 SQL0911N, SQLSTATE 40001과 이유 코드 68 이 반환됩니다.

```
<세션 A>
$ db2 connect to sample
$ db2 "create table t1 (c1 int not null primary key, c2 int)"
$ db2 "insert into t1 values (1,10),(2,20),(3,30)"
$ db2  +c "update t1 set c2 = c2 + 100 where c1= 2"
```

```
<세션 B>
$ db2 connect to sample
$ db2  "select * from t1 where c1 = 2"
SQL0911N  현재의 트랜잭션이 교착 상태 또는 시간종료로 인해 롤백되었습니다.
이유 코드 "68".  SQLSTATE=40001
```

Figure 1110B • • • 잠금 대기 시간 초과

TOPIC 11.11 교착 상태

Point

⑩두 개의 응용프로그램이 서로 잠금 대기 상태가 되는 것을 교착 상태라고 합니다. 데이터베이스 구성 변수인 DLCHKTIME을 주기로 교착 상태에 있는 응용프로그램을 파악하고, 한 응용프로그램을 강제로 종료시키면, 다른 응용프로그램은 작업을 계속합니다.

Tip

DLCHKTIME 은 '온라인 구성 가능 변수' 이므로 attach 명령어를 이용하여 인스턴스에 접속한 후 변경하면 즉시 반영될 수 있습니다.

1. update db cfg 명령어로 DLCHKTIME 데이터베이스 구성 변수를 설정합니다. <점검 간격>은 밀리초(1000분의 1초) 단위로 표시합니다.

```
$ db2 update db cfg for <데이터베이스명> using DLCHKTIME <점검 간격>
```

2. DLCHKTIME 구성 변수의 기본값은 10000 ms로 10초 간격으로 교착 상태가 점검됩니다.

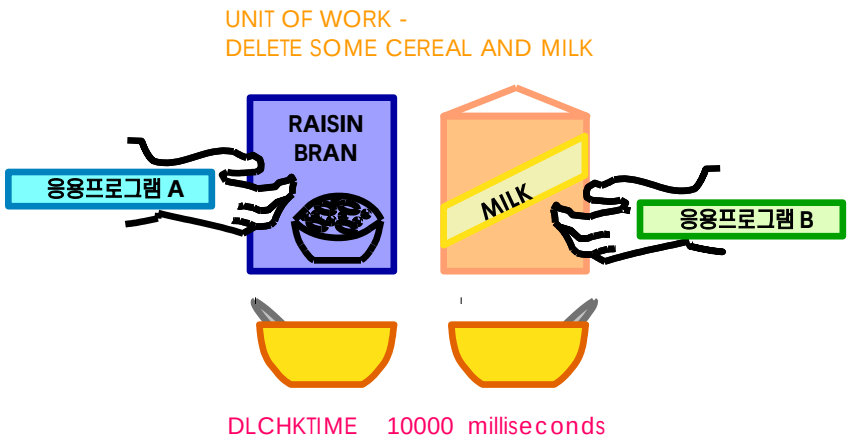

Figure 1111A ••• DLCHKTIME 구성 변수

Tip

교착 상태를 해결하기 위해 희생자 프로세스를 선택하는 기준은 엔진 내부의 알고리즘으로 사용자가 조절할 수 없습니다.

3. 교착 상태가 감지되면, 희생자(victim)로 선정된 한 쪽의 응용프로그램은 강제로 종료되고, SQL0911N, SQLSTATE 40001과 이유 코드 2가 반환됩니다.

```
<세션 A>
$ db2 connect to sample
$ db2 "create table t1 (c1 int not null primary key, c2 int)"
$ db2 "insert into t1 values (1,10),(2,20),(3,30)"
$ db2 +c "select * from t1 where c1 = 1 with rs"
$ db2 +c "update t1 set c2 = c2 + 100 where c1= 2"
```

```
<세션 B>
$ db2 connect to sample
$ db2 +c "select * from t1 where c1 = 2 with rs"
$ db2 +c "update t1 set c2 = c2 + 100 where c1= 1"
SQL0911N 현재의 트랜잭션이 교착 상태 또는 시간종료로 인해 롤백되었습니다.
이유 코드 "2". SQLSTATE=40001
```

Figure 1111B ••• 교착 상태로 인한 트랜잭션의 롤백

Memo