

ユーティリティーとロック



<第1.00版 2014年 9月>

お断り: 当資料は、DB2 for Linux, UNIX and Windows V10.5をベースに作成されています。

本資料掲載事項は、ある特定の環境・使用状況においての正確性がIBMによって確認されていますが、すべての環境において同様の結果が得られる保証はありません。これらの技術を自身の環境に適用する際には、自己の責任において十分な検証と確認を実施いただくことをお奨めいたします。

© Copyright IBM Japan Systems Engineering Co., Ltd. 2014

はじめに

- 当資料は、DB2 for LUWで、ユーティリティーを実行した際に取得されるロックについて解説したものです。ユーティリティーと並行して実行されるアプリケーションにどのような影響を与えるのか解説しています。
- ユーティリティー実行時のロック競合を防ぐための情報を、読者に提供することを目的としています。
- DB2 10.5環境でテストした結果を元に作成しています。
バージョンに依存して、取得されるロックの内容に変更が出る可能性もありますが、異なるバージョンのDB2 for LUWをお使いの場合でも、当資料を参考にしてください。

目次

1. EXPORT
2. IMPORT
3. LOAD
4. RUNSTATS
5. REORG TABLE
 - オンライン表再編成
 - オフライン表再編成
6. REORG INDEXES
7. REORGCHK
8. ONLINE BACKUP
9. INGEST

各章の最後に、各ユーティリティーに関する参照ページがあります。
この資料はロックに焦点を当てています。各ユーティリティーに関する詳細(技術情報)については、
参考ページに掲載されているリンク先をご参照ください。

オンライン・ユーティリティ

- 各ユーティリティで指定するALLOWオプションにより、排他制御の方法が決まります。
- EXPORT/REORGCHK/ONLINE BACKUP/INGESTには、ALLOWオプションはありません。

ユーティリティ	タイプ	指定できるALLOWオプション		
		NO	READ	WRITE
IMPORT	CREATE*1/ REPLACE/REPLACE_CREATE	◎	×	×
IMPORT	INSERT/INSERT_UPDATE	◎	×	○
LOAD		◎	○*2	×
RUNSTATS		×	○	◎
REORG TABLE	オンライン表再編成	×	○	◎
REORG TABLE	オフライン表再編成	○	◎	×
REORG INDEXES		○	◎	○

*1 V9.5以降、IMPORTのCREATEオプションは非推奨です。

*2 V10.1FP1以降、ALLOW READ ACCESSは非推奨です。
INGESTユーティリティの使用を検討してください。

◎ 指定可能(デフォルト)

○ 指定可能

×

ユーティリティ実行中の該当表に対する参照/更新

- 各ユーティリティで最も高いアクセス・レベルのALLOWオプションを指定した場合に、該当表に対する参照/更新が可能かどうかを一覧で示します(詳細は各章参照)。

ユーティリティ	オプション	ユーティリティ対象表	
		参照 (SELECT)	更新 (INSERT/UPDATE/DELETE)
EXPORT		○	○
IMPORT	ALLOW WRITE ACCESS	○	○
LOAD	ALLOW READ ACCESS*	○ 実行直後、完了直前に瞬間的にアクセス不可	×
RUNSTATS	ALLOW WRITE ACCESS	○	○
REORG TABLE(オンライン)	ALLOW WRITE ACCESS	○ 実行直後、完了直前に瞬間的にアクセス不可	○ 実行直後、完了直前に瞬間的にアクセス不可
REORG TABLE(オフライン)	ALLOW READ ACCESS	△ ソート・フェーズ、ビルド・フェーズのみアクセス可	×
REORG INDEXES	ALLOW WRITE ACCESS	○ 実行直後、完了直前にアクセス不可	○ 実行直後、完了直前にアクセス不可
REORGCHK		○	○
ONLINE BACKUP		○	○
INGEST		○	○

* V10.1FP1以降、ALLOW READ ACCESSは非推奨です。INGESTユーティリティの使用を検討してください。

1. EXPORT

□EXPORTユーティリティの分離レベルCS

- EXPORT実行中でも該当表を**参照可能**

□CUR_COMMIT(DB CFG)=ONの環境であれば更新処理とEXPORTはロック競合にならない

- EXPORT実行中でも該当表を**更新可能**
- CUR_COMMIT(DB CFG)はV9.7からデフォルトでON

□該当表への更新処理中はEXPORTをロック待ちさせた場合に以下のいずれかの対応を行う

- CUR_COMMIT(DB CFG)=OFF
- EXPORTをRSまたはRRでBIND
- SELECT文に WAIT FOR OUTCOME を指定する
 - Select * from xx wait for outcome

解説:EXPORT

- ①Appl1が
テーブルにIXロックを取得
レコードにXロックを取得

- ②Appl2がExport

□CUR_COMMIT=ON

- ①更新処理

Update tbl set c2='Z' where c1=3

表にIXロック

行にXロック

C0	C1	C2
	3	C

- ②EXPORT

更新前イメージをEXPORT
(C1=3、C2='C')

□CUR_COMMIT=DISABLE

- ①更新処理

Update tbl set c2='Z' where c1=3

表にIXロック

行にXロック

C0	C1	C2
	3	C

- ②EXPORT

ロック待ち

①のコミットを待って、
更新後のレコードをEXPORT
(C1=3、C2='Z')

WAIT

EXPORT実行時のロック

- 取得されるロックは、EXPORTの分離レベルとSELECTステートメントのアクセス経路によって異なる
 - EXPORTの分離レベルCS(デフォルト)
 - EXPORTユーティティのパッケージをバインドし直すことにより、他のISOLATIONに変更可能
 - `db2 bind $HOME/sqllib/bnd/db2uexpm.bnd isolation rs`

解説:EXPORT実行時のロック

□取得されるロックは分離レベルやアクセス・プランによって決まる

- 詳細はマニュアルを参照

➤ http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.perf.doc/doc/r0005275.html

□(例)EXPORT ... SELECT * FROM STAFF

- STAFF表にはユニーク索引がある前提
- ISOLATION RRであれば、表に対してSロックを取得しようとする
- ISOLATION CSであれば、表に対してISロック、行に対してNSロックを取得しようとする

分離レベル	読み取り専用および未確定のスキャン	カーソル操作		検索条件付き UPDATE または DELETE	
		スキャン	現在の場所	スキャン	更新または削除
RR	➔ S/-	IX/S	IX/X	X/-	X/-
RS	IS/NS	IX/U	IX/X	IX/X	IX/X
CS	➔ IS/NS	IX/U	IX/X	IX/X	IX/X
UR	IN/-	IX/U	IX/X	IX/X	IX/X

実行例:EXPORTのロック待機(1/2)

□ロック待ち再現シナリオ

- ①Appl1:db2 +c “update tbl set c2= ‘Z’ where c1=3”
- ②Appl2:db2 +c “export to tbl.ixf of ixf select * from tbl wait for outcome”
- ロック状況の取得:db2 “call monreport.lockwait()”

□monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	1930	NS	1914	X	ROW

← Appl2がNSロックを取得したいが、
Xロックが取得されているため、
ロック待ちとなっている

↓ Appl1(1914)とAppl2(1930)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	1930	1914
APPLICATION_ID	*LOCAL. lock105. 140821064806	*LOCAL. lock105. 140821064519
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	X
LOCK_MODE_REQUESTED	NS	-

実行例:EXPORTのロック待機(2/2)

□monreport.lockwait()続き

-- Lock details --

```
LOCK_NAME           = 00030008000000000000000000652
LOCK_WAIT_START_TIME = 2014-08-21-15.49.33.851841
LOCK_OBJECT_TYPE    = ROW
TABSCHEMA           = LOCK105
TABNAME              = TBL
ROWID                = 6
LOCK_STATUS          = W
LOCK_ATTRIBUTES      = 000000000000000000
ESCALATION            = N
```

← LOCK105.TBL表への行ロックの競合

-- Lock requestor waiting agent and activity --

```
AGENT_TID           = 12252
REQUEST_TYPE         = OPEN
ACTIVITY_ID          = 9
UOW_ID               = 3
LOCAL_START_TIME     = 2014-08-21-15.49.33.851811
ACTIVITY_TYPE        = READ_DML
ACTIVITY_STATE       = EXECUTING
```

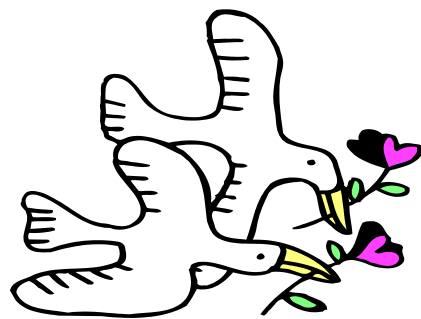
Appl2が発行している statement →
select * from tbl wait for outcome

```
STMT_TEXT           =  
select * from tbl wait for outcome
```

EXPORT参照:

□『DB2 V9.5 運用管理ガイド: データ移動ユーティリティ』

- <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%20%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%E3%83%87%E3%83%BC%E3%82%BF%E7%A7%BB%E5%8B%95%E3%83%A6%E3%83%BC%E3%83%86%E3%82%A3%E3%83%AA%E3%83%86%E3%82%A3%E3%83%BC?lang=j>
a
- EXPORTに関する全般的な情報が記載されています。

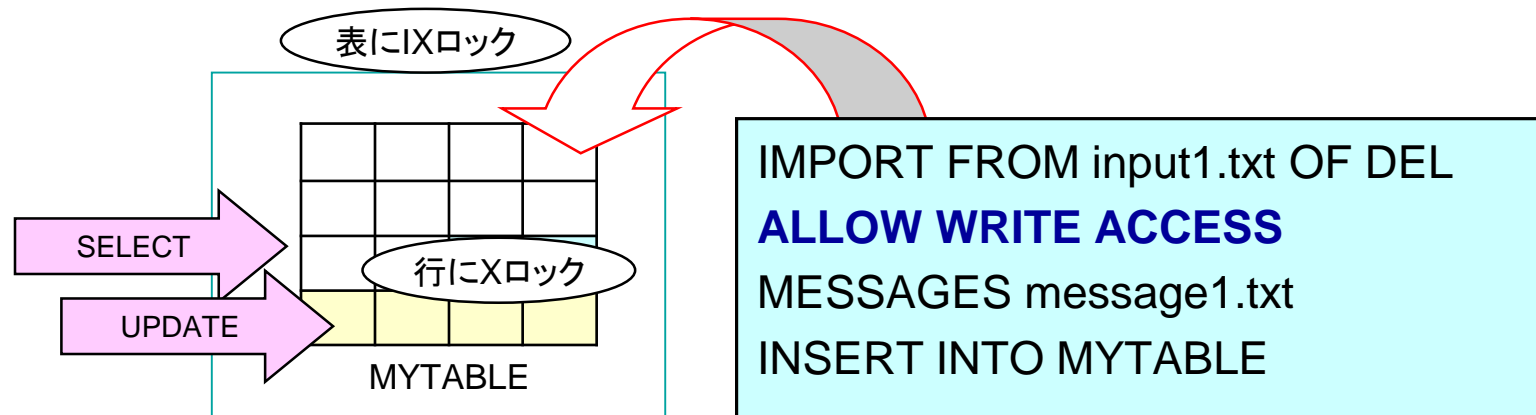


ブランク・ページです。

2. IMPORT

□ アクセス・レベルはALLOWオプションで指定

- ALLOW WRITE ACCESSの場合：
 - IMPORT実行中でも該当表を参照更新可能
 - 1つの表に対して、複数同時のインポートも可能



- ALLOW NO ACCESSの場合：
 - REPLACE/CREATEモードのとき
 - IMPORT中は表へのアクセス不可
 - INSERT/INSERT_UPDATEモードのとき
 - IMPORT中は分離レベルURでのみ読み取り可能

解説:IMPORT

□ALLOW WRITE ACCESS

- IMPORT実行中の表に対して意図的排他ロック(IX)を取得
 - 1つの表に対して複数同時のインポートも可能
 - INSERT/INSERT_UPDATEモードのときに指定可能なオプション
 - 表に対してIXロック、挿入行に対してXロック取得
 - REPLACE/CREATEモードのときに指定するとエラー(SQL27970)
- 非互換の表ロックをかけている他のアプリケーションがある場合、アプリケーションがcommit/rollbackされるまでIMPORTは開始されない
- 非互換の行ロックをかけている他のアプリケーションがある場合、アプリケーションがすべて変更内容をcommit/rollbackするまでIMPORTは待機する
- IMPORT中は、非互換の表/行ロックを要求している他のアプリケーションは、IMPORTがcommit/rollbackされるまで待機する
- ロック・エスカレーションを回避するため、定期的にCOMMITが実行される
 - COMMITCOUNTが明示的に設定されていなければ、インポートは、COMMITCOUNT AUTOMATIC が指定されている場合と同じようにコミットを実行
 - COMMITCOUNT が 0 に設定されていると、コミットは実行されない

□ALLOW NO ACCESS(デフォルト)

- IMPORT開始から終了まで、表レベルの排他ロックを取得
 - REPLACE/CREATEモードのとき
 - IMPORT中は表へのアクセス不可(Zロック)
 - INSERT/INSERT_UPDATEモードのとき
 - IMPORT中は分離レベルURでのみ読み取り可能(Xロック)

IMPORTオプションとロック

IMPORTコマンド	INSERT／INSERT_UPDATE		CREATE ^{*1} ／REPLACE	
	照会	更新	照会	更新
ALLOW NO ACCESS (オフライン・インポート)	△ ^{*2}	×	×	×
	表レベルXロック		表レベルX→Zロック	
ALLOW WRITE ACCESS (オンライン・インポート)	○	○	指定不可(SQL27970)	
	表レベルIXロック 行レベルXロック			

*1 V9.5以降、IMPORTのCREATEオプションは非推奨です。

*2 分離レベルURでのみ読み取り可

IMPORT(ALLOW NO ACCESS + INSERT/INSERT_UPDATE)のロック待機

□ 排他モードの表ロックを取得

- INSERTオプションつきIMPORTでLOCK TABLE xx IN EXCLUSIVE MODEが発行されている

① Appl1が
テーブルにISロックを取得
レコードにNSロックを取得

② Appl2がIMPORT
Appl1のテーブル・ロックに対してロック待ち

① 参照処理(isolation=RS)
select from tbl where c1=100

表にISロック

行にNSロック

C0	C1	C2
>	100	Y

② IMPORT
(isolation=RS)

表にXロック要求
ロック待ち！

WAIT

実行例:IMPORT(ALLOW NO ACCESS + INSERT/INSERT_UPDATE)のロック待機(1/2)

□ロック待ち再現シナリオ

- ①Appl1:db2 +c “select * from tbl where c1=100”(ISOLATION=RS)
- ②Appl2:db2 +c “import from tbl.del of del allow no access insert into tbl”
- ロック状況の取得:db2 call monreport.lockwait()

□monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	1930	X	1914	IS	TABLE

← Appl2がXロックを取得したいが、ISロックが取得されているため、ロック待ちとなっている

↓ Appl1(1914)とAppl2(1930)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	1930	1914
APPLICATION_ID	*LOCAL. lock105. 140821064806	*LOCAL. lock105. 140821064519
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	IS
LOCK_MODE_REQUESTED	X	-

実行例:IMPORT(ALLOW NO ACCESS + INSERT/INSERT_UPDATE)のロック待機(2/2)

monreport.lockwait出力結果(抜粋)つづき

-- Lock details --

```
LOCK_NAME           = 00030008000000000000000000054
LOCK_WAIT_START_TIME = 2014-08-21-16. 42. 53. 557874
LOCK_OBJECT_TYPE     = TABLE
TABSCHEMA            = LOCK105
TABNAME              = TBL
ROWID                =
LOCK_STATUS          = W
LOCK_ATTRIBUTES      = 00000000000400000
ESCALATION            = N
```

← LOCK105.TBL表への表ロックの競合

-- Lock requestor waiting agent and activity --

```
AGENT_TID           = 12252
REQUEST_TYPE        = EXECIMMD
ACTIVITY_ID         = 11
UOW_ID              = 11
LOCAL_START_TIME    = 2014-08-21-16. 42. 53. 557760
ACTIVITY_TYPE       = OTHER
ACTIVITY_STATE      = EXECUTING
```

Appl2が発行している statement →
LOCK TABLEが内部的に発行されている

```
STMT_TEXT           =
LOCK TABLE tbl IN EXCLUSIVE MODE
```

IMPORT(ALLOW WRITE ACCESS + INSERT/INSERT_UPDATE)ロック待機しないケース

□ 排他モードの行ロックを取得

- 表に対してIXロック、挿入行に対してXロックを取得
- IMPORT中の表に対して更新可能

①更新処理(isolation=CS)

update tbl set c2='Z' where c1=100

表にIXロック

行にXロック

C0	C1	C2
	100	A
→	100	Y
	100	B

①Appl1が

テーブルにIXロックを取得

レコードにXロックを取得

②Appl2がIMPORT

Appl1のテーブル・ロックに対してロック待ち

②IMPORT(100, B)
(isolation=RS)

ロック待ちせずに挿入可能

IMPORT(ALLOW WRITE ACCES + INSERT/INSERT_UPDATE)のロック待機

□ 排他モードの行ロックを取得

- 表に対してIXロック、挿入行に対してXロックを取得

①更新処理(isolation=CS)

update tbl set c2='Z' where c1=100

表にIXロック

行にXロック

C0	C1	C2
>	100	<- Y

①Appl1が

テーブルにIXロックを取得

レコードにXロックを取得

②Appl2がIMPORT

Appl1のテーブル・ロックに対してロック待ち

②IMPORT

(isolation=RS)

行にUロック要求
ロック待ち！

WAIT

* C1 にユニーク索引がある前提
すでに表に存在するキー値をINSERT
した例です。

実行例:IMPORT(ALLOW WRITE ACCES + INSERT/INSERT_UPDATE)実行時のロック

□ロック待ち再現シナリオ

- ①Appl1:db2 +c “update tbl set c2= ‘Z’ where c1=100”(ISOLATION=CS)
- ②Appl2:db2 +c “import from tbl.del of del allow write access insert into tbl”
- ロック状況の取得:db2 call monreport.lockwait()

□monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	6616	U	6575	X	ROW

← Appl2がUロックを取得したいが、
Xロックが取得されているため、
ロック待ちとなっている

↓ Appl1(6575)とAppl2(6616)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	6616	6575
APPLICATION_ID	*LOCAL. lock105. 140904043513	*LOCAL. lock105. 140904041628
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	X
LOCK_MODE_REQUESTED	U	-

IMPORT参照:

□ 『DB2 V9.5 運用管理ガイド: データ移動ユーティリティー』

- <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%20%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9A%E3%83%87%E3%83%BC%E3%82%BF%E7%A7%BB%E5%8B%95%E3%83%A6%E3%83%BC%E3%83%86%E3%82%A3%E3%83%AA%E3%83%86%E3%82%A3%E3%83%BC?lang=ja>
 - IMPORTに関する全般的な情報が記載されています。

□ インポートの概要

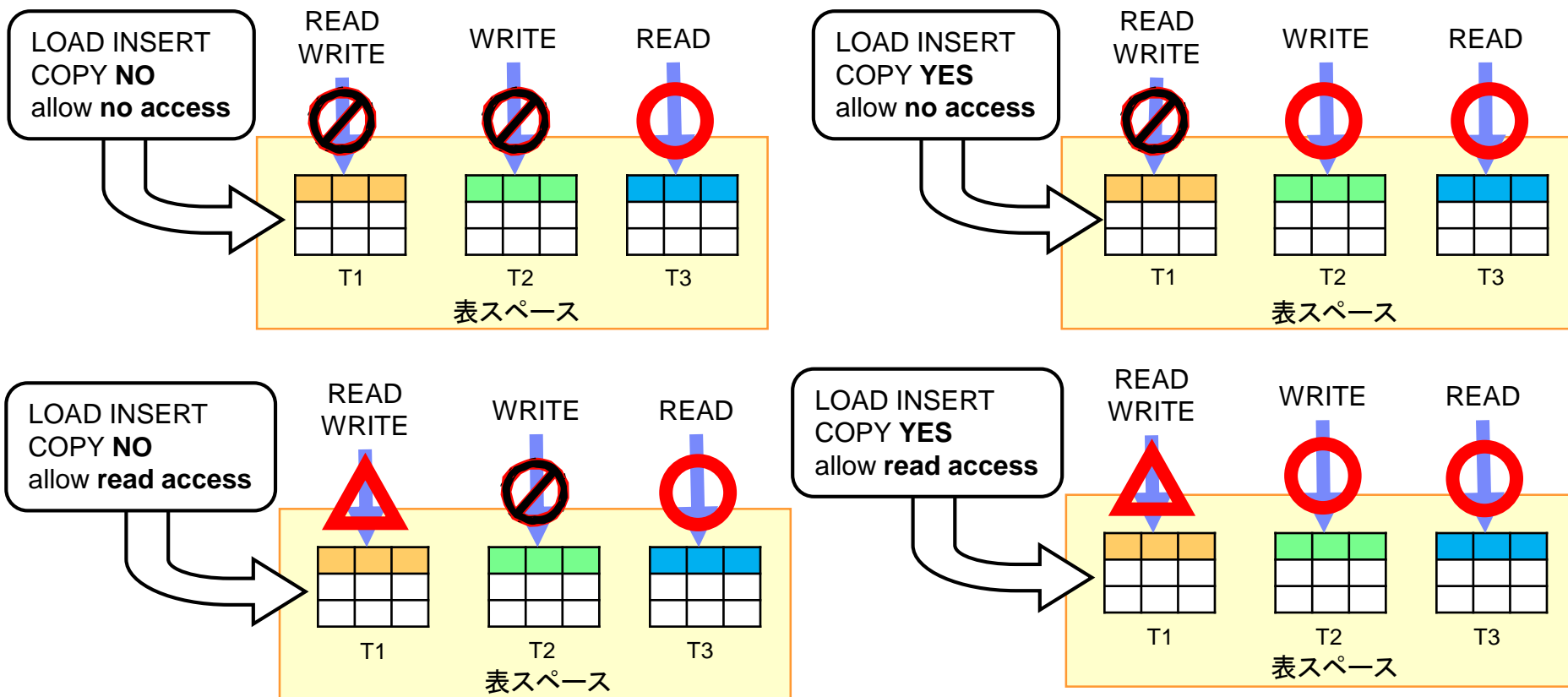
- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0004573.html
 - マニュアルに記載されているIMPORTの概要です。

□ インポート・ユーティリティーでサポートされている表ロックング・モード

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0004583.html
 - IMPORTとロックに関する情報が記載されています。

3. LOAD

□LOADオプションと表へのアクセス



LOAD以前に存在したデータは参照可能
LOAD中は更新不可
LOADしたデータはLOAD完了時に参照更新可能

解説: LOADオプションと表へのアクセス

□ LOAD対象表と同じ表スペース内の他表へのアクセス

- COPY NO: **参照のみ可能**
 - 表スペースがBackup pending状態になるまで、同じ表スペース内の他の表に共有モードでロックが取得される
 - LOCK WITH FORCEオプションが指定されている場合は、同じ表スペース内の他の表に対してロック競合するロックを取得しているアプリケーションは force applicationされる
- COPY YES: **参照更新が可能**
 - 同じ表スペース内の他の表へはロックが取得されないため、他の表の参照更新が可能

□ LOAD対象表に対するアクセス

- ALLOW NO ACCESS
 - 表へのアクセス不可
 - 表に対して超排他ロック(Z)を取得
- ALLOW READ ACCESS(V10. FP1から非推奨)
 - LOAD以前に存在したデータは参照可能
 - LOAD中のデータはLOAD完了後に参照更新可能
 - LOAD INSERT時のみ指定可能なオプション

LOADオプションとロック

LOADコマンド		LOAD中の表		表スペース内の他の表	
		照会	更新	照会	更新
COPY NO (※1)	ALLOW NO ACCESS	×	×	○	×
		Zロック		Sロック	
	ALLOW READ ACCESS	○(※2)	×	○	×
		(開始前)Zロック→Uロック →(完了直前)Zロック		Sロック	
COPY YES	ALLOW NO ACCESS	×	×	○	○
		Zロック		—	
	ALLOW READ ACCESS	○(※2)	×	○	○
		(開始前)Zロック→Uロック →(完了直前)Zロック		—	

※1 LOAD終了後、表スペースはバックアップ保留状態(照会は可能、更新はエラー)

※2 INSERTモード時に既存のデータについて照会可能
LOAD実行前および完了直前に瞬間的にアクセス不可

参考:

□ ALLOW READ ACCESSとUSE <tablespace-name>オプション

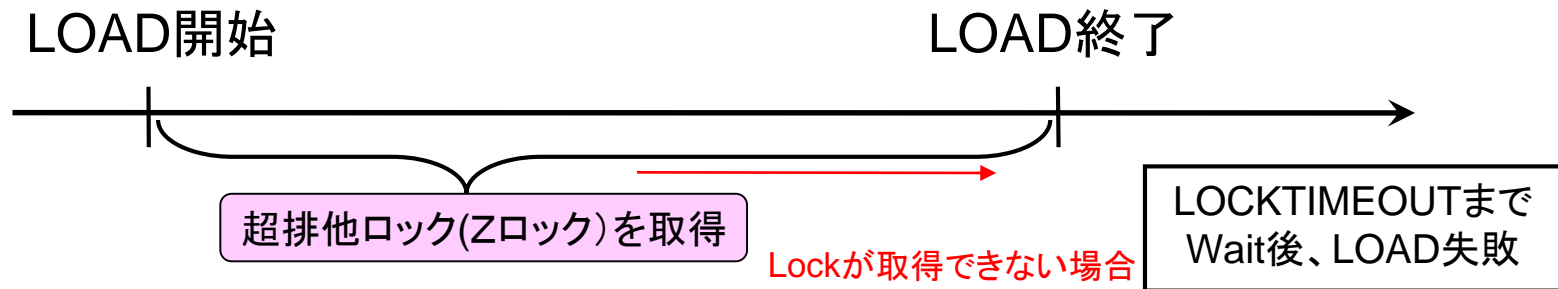
- USE <tablespace-name>を指定した場合は、索引コピー・フェーズが完了するまでZロックが取得され続ける
 - USE <tablespace-name> オプションが指定されている場合、INDEX REBUILD時に、指定された一時表スペースに索引のシャドー・コピーが作成される、LOAD完了時の索引コピー・フェーズで、元の表スペース上にコピーされる
 - I/O時間を要しますが、索引用に十分な表スペースが確保できます。
 - 索引コピー・フェーズの期間ずっと排他ロックが続くのでご注意ください。
 - USE <tablespace-name>オプションを指定しない場合、シャドー索引が索引オブジェクトと同じ表スペース内に作成される
 - パフォーマンス的に有利ですが、表スペースが不足する可能性がありますのでご注意ください。

□ ALLOW READ ACCESSとDPF(Database Partitioning Feature)環境

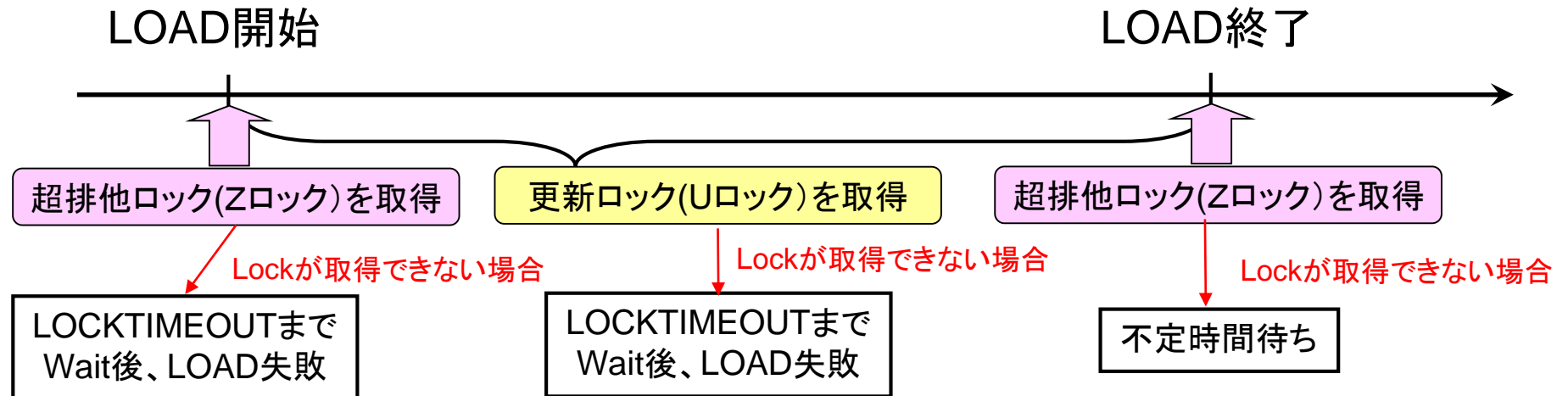
- DPF環境ではデッドロックの可能性がある
- 詳細は、IBM Knowledge Center[読み取りアクセス・ロード操作]を参照してください
 - http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0007909.html

LOAD中の表に対して取得するロック

□ ALLOW NO ACCESS



□ ALLOW READ ACCESS



同時稼動SQLにより、オンラインLOADの失敗、遅延が発生する可能性がある

解説: LOAD中の表に対して取得するロック

□ ALLOW NO ACCESS

- LOAD中の表に対して超排他ロック(Z)を取得する
 - ロックが取得できなければLOAD処理がロック・タイムアウトになる
- LOAD対象表の表スペースにISロックを取得する
- 同じ表スペース内の他の表は参照のみ可能

□ ALLOW READ ACCESS (V10. FP1から非推奨)

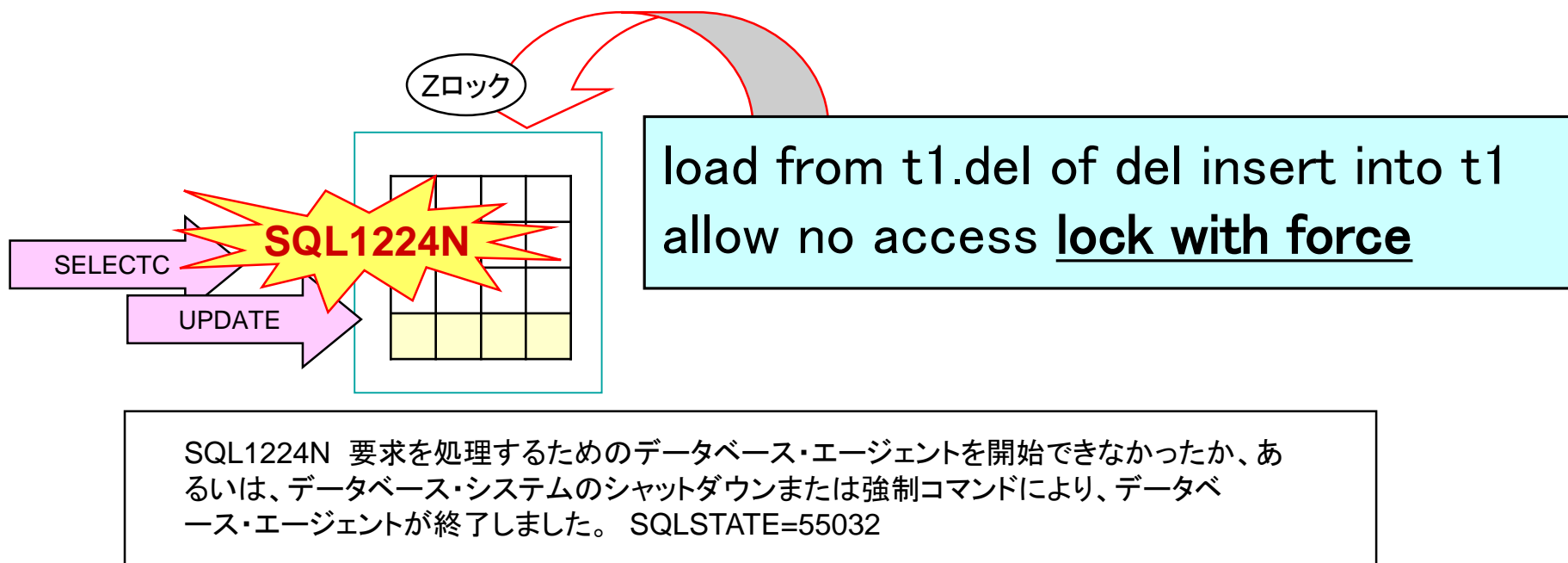
- LOAD開始直前に表に超排他ロック(Z)を取得
 - 開始時のZロックが取得できなければLOAD処理がロック・タイムアウトになる
- LOAD開始直前のZロック取得後、LOADが開始され、Uロックに変わる
 - Uロックが取得できなければLOAD処理がロック・タイムアウトになる
- LOAD終了直前に超排他ロック(Z)を取得
 - 終了時(COMMITのため)のZロックが取得できなければ待機する
 - ロック・タイムアウトにはならない

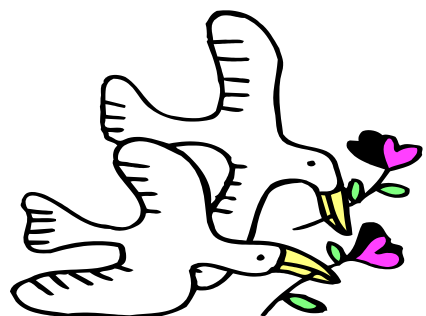
LOADのLOCK WITH FORCEオプション

□ 同時稼動SQLによるオンラインLOADの失敗、遅延を防ぐ

● LOCK WITH FORCEオプション

- LOAD先の表にロックを取得したアプリケーションを強制終了させて、LOADのためのロックを取得して実行する
- FORCEされたアプリケーションには、SQL1224Nが戻される





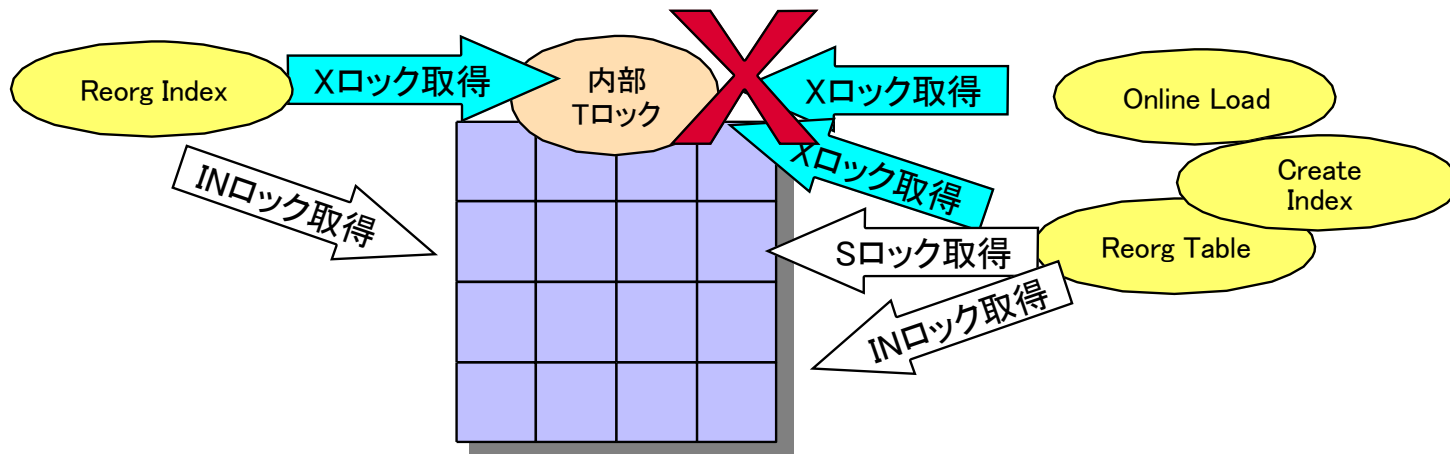
ブランク・ページです。

参考: 内部Tロック

□ 同一表に複数ロードすることは不可能

□ 内部Tロックでシリアライズ

- 「内部Tロック」に排他ロックを取得しシリアライズされるため、複数実行することは出来ない。
- 索引作成、索引の再編成、ロード、表の再編成

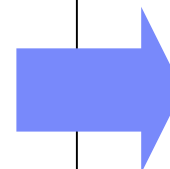


参考: 内部Tロック

□ロック・スナップショットの出力結果(抜粋)

- db2 get snapshot for locks on sample

Lock Name	= 0x0002001300000000000000000074
Lock Attributes	= 0x00000000
Release Flags	= 0x40000000
Lock Count	= 1
Hold Count	= 0
Lock Object Name	= 19
Object Type	= Internal Table Alter Lock
Tablespace Name	= USERSPACE1
Table Schema	= LOCK105
Table Name	= TBL
Mode	= X



内部Tロック

LOAD参照:

□ 『DB2 V9.5 運用管理ガイド: データ移動ユーティリティ』

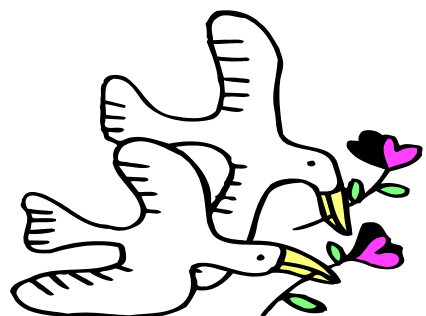
- <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%20%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9A%E3%83%87%E3%83%BC%E3%82%BF%E7%A7%BB%E5%8B%95%E3%83%A6%E3%83%BC%E3%83%86%E3%82%A3%E3%83%AA%E3%83%86%E3%82%A3%E3%83%BC?lang=ja>
 - LOADに関する全般的な情報が記載されています。

□ ロード操作時の表のロックング

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0007938.html
 - LOADとロックに関する情報が記載されています。

□ 読み取りアクセス・ロード操作

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0007909.html
 - ALLOW READ ACCESSのLOADに関する情報が記載されています。



ブランク・ページです。

4. RUNSTATS

□ アクセス・レベルはALLOWオプションで指定

- ALLOW WRITE ACCESSの場合：
 - RUNSTATS実行中でも該当表を**参照更新可能**
 - 他のアプリケーションからの更新処理が少ないタイミングでRUNSTATSすることを推奨
 - 表の統計を収集する時と索引の統計を収集する時の間に、該当表に更新があった場合、レコード内容が異なることで統計に不整合が発生する場合があるため
- ALLOW READ ACCESSの場合：
 - RUNSTATS実行中は該当表を**参照のみ可能**

□ RUNSTATSは、複数のシステム・カタログ表が更新されロックが取得される

- ロック競合を避けるため、RUNSTATSコマンドを並列で実行せず、順次実行すること

解説: RUNSTATS

- ALLOW WRITE ACCESSオプション(デフォルト): RUNSTATS実行中は該当表を更新可能
 - 1. SYSTABLEに対してIS/NS-Lock
 - 2. 該当表に対してIN-Lock(UR)でアクセス
 - 3. カタログ更新(IX, X-Lock)
- ALLOW READ ACCESSオプション: RUNSTATS実行中は該当表を参照のみ可能
 - 1. SYSTABLEに対してIS/NS-Lock
 - 2. 該当表に対してS-Lock(RR)でアクセス
 - 3. カタログ更新(IX, X-Lock)
- RUNSTATSコマンドにより更新されるシステム・カタログ表には次のものがあります。指定するオプションによって更新させる統計情報は変わります。
 - 表統計: SYSCAT.TABLES および SYSSTAT.TABLES
 - 列統計: SYSCAT.COLUMNS および SYSSTAT.COLUMNS
 - 複数列統計: SYSCAT.COLGROUPS および SYSSTAT.COLGROUPS
 - 索引統計: SYSCAT.INDEXES と SYSSTAT.INDEXES
 - 列分散統計: SYSCAT.COLDIST と SYSSTAT.COLDIST
 - 詳細は、IBM Knowledge Center[\[カタログ統計の表\]](#)を参照してください。

RUNSTATSオプションとロック

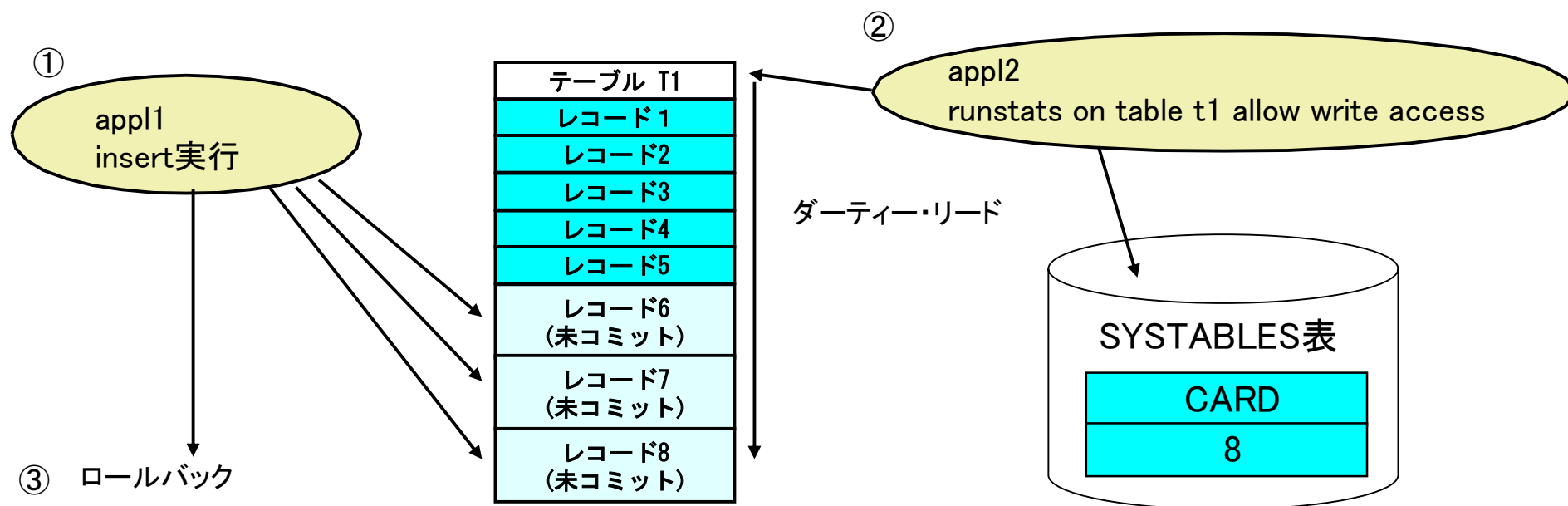
□ RUNSTATSのオプションとロック

RUNSTATSコマンド	該当表		SYSTABLES	カタログ更新
	照会	更新		
ALLOW READ ACCESS	○	×	表レベルISロック 行レベルNSロック	表レベルIXロック 行レベルXロック
	表レベルSロック (分離レベルRR)			
ALLOW WRITE ACCESS	○	○	表レベルISロック 行レベルNSロック	表レベルIXロック 行レベルXロック
	表レベルINロック (分離レベルUR)			

ALLOW WRITE ACCESS実行時の読み取り

□ ALLOW WRITE ACCESS実行時の読み取り

- ALLOW WRITE ACCESSは、該当表をダーティ・READ(UR)しているのでカタログの統計情報が実際と異なる場合もありえる



```
>DB2 SELECT TABNAME, CARD FROM SYSCAT.TABLES WHERE TABNAME=' T1'
```

TABNAME	CARD
T1	8

1 レコードが選択されました。

実際は5件

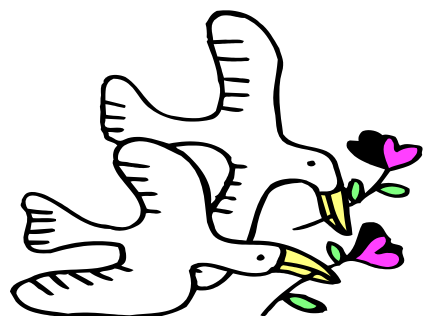
RUNSTATS参照:

□ [RUNSTATS コマンド]

- http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0001980.html?cp=SSEPGG_10.5.0%2F3-6-2-4-113&lang=ja
 - RUNSTATS実行時に取得されるロックに関する記載があります。

□ 『DB2 V9.5 運用管理ガイド:DB2 for LUW V9.5 データベース・メンテナンス』

- <https://www.ibm.com/developerworks/community/wikis/home?lang=ja#!/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%20%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9ADB2%20for%20LUW%20V9.5%20%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%83%BB%E3%83%A1%E3%83%B3%E3%83%86%E3%83%8A%E3%83%B3%E3%82%B9>
 - RUNSTATSに関する全般的な情報が記載されています。



ブランク・ページです。

5. REORG TABLE

□ 表再編成の2つの手法

- オンライン再編成: **参照更新可能**
 - オンライン再編成の実行中に、他のユーザーが該当表に対して参照・更新可能
 - 実行時間の長いトランザクションが、オンライン再編成中の表にアクセスする場合は、オンライン再編成が待機する
- オフライン再編成: 一部のフェーズで**参照可能**
 - オフライン再編成の実行中に、他のユーザーが該当表に対して更新できない
 - 参照可能な期間は限られるため、オフライン再編成中は、該当表へのアクセスを停止するのが一般的



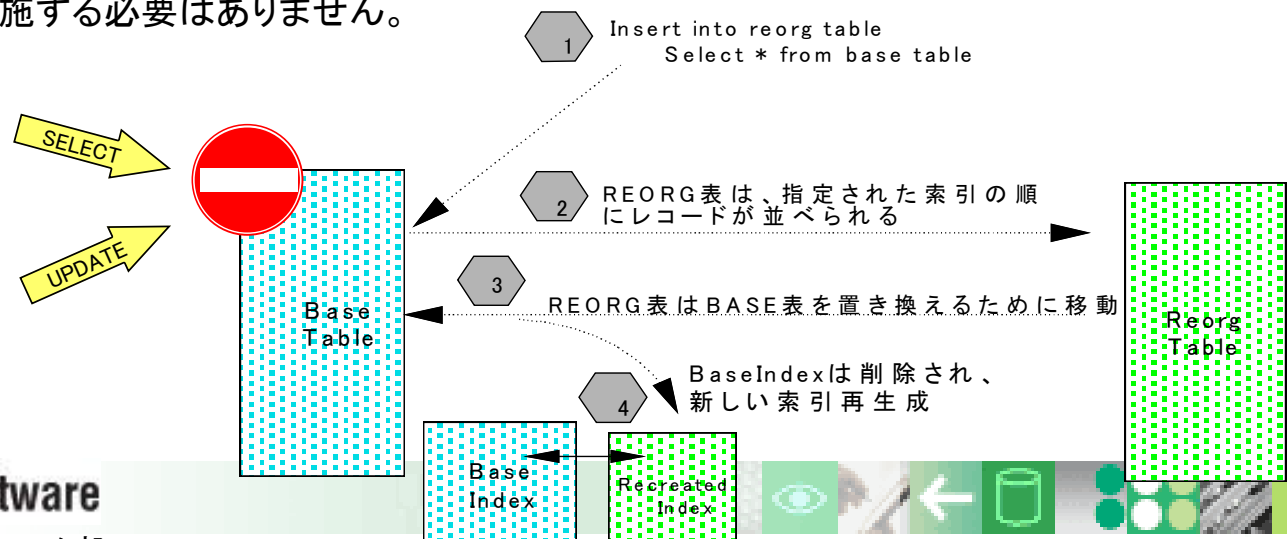
解説: REORG TABLE

□ オンライン再編成(インプレース)

- インプレース・アプローチという手法です。一時表を使わず、既存のエクステント(表スペースにおけるテナ内の領域の割り振り単位で、デフォルトは32ページ)の空き領域を利用して再編成を行います。
- オンライン再編成では、実行中に他のユーザーが該当表に対して参照・更新することができます。
- データを移動している範囲内に、未コミットのトランザクションのレコードがある場合には、そのトランザクションがCOMMITされるまで、再編成はWAITします。
- オンライン再編成の場合、索引は再作成されないため、別途、索引の再編成を実施する必要があります。

□ オフライン再編成(クラシック)

- シャドウ・コピーを作成する手法です。シャドウ・コピーは、表が保管されている表スペースか一時表スペースに一時的に置かれます。REORGする表のデータ量の2-3倍の(一時)表スペース領域を確保する必要があります。
- オフライン再編成中は、並行して表にアクセスすることができないタイミングがあるため、該当表にアクセスするアプリケーションは停止してください。
- オフライン再編成では、表本体の再編成とともに、表に定義されたすべての索引を再作成するため、索引のみの再編成を別途実施する必要はありません。



オンライン表再編成のオプションとロック

REORG TABLEコマンド		該当表に対するロック	
		照会	更新
インプレース	INPLACE ALLOW WRITE ACCESS	○(※)	○(※)
		(実行直後)Zロック→ISロック→ (TRUNCATE時)Sロック→(完了直前)Zロック	
	INPLACE ALLOW READ ACCESS	○(※)	×
		(実行直後)Zロック→Sロック→ (TRUNCATE時)Sロック→(完了直前)Zロック	

※実行直後、完了直前に瞬間的にアクセス不可

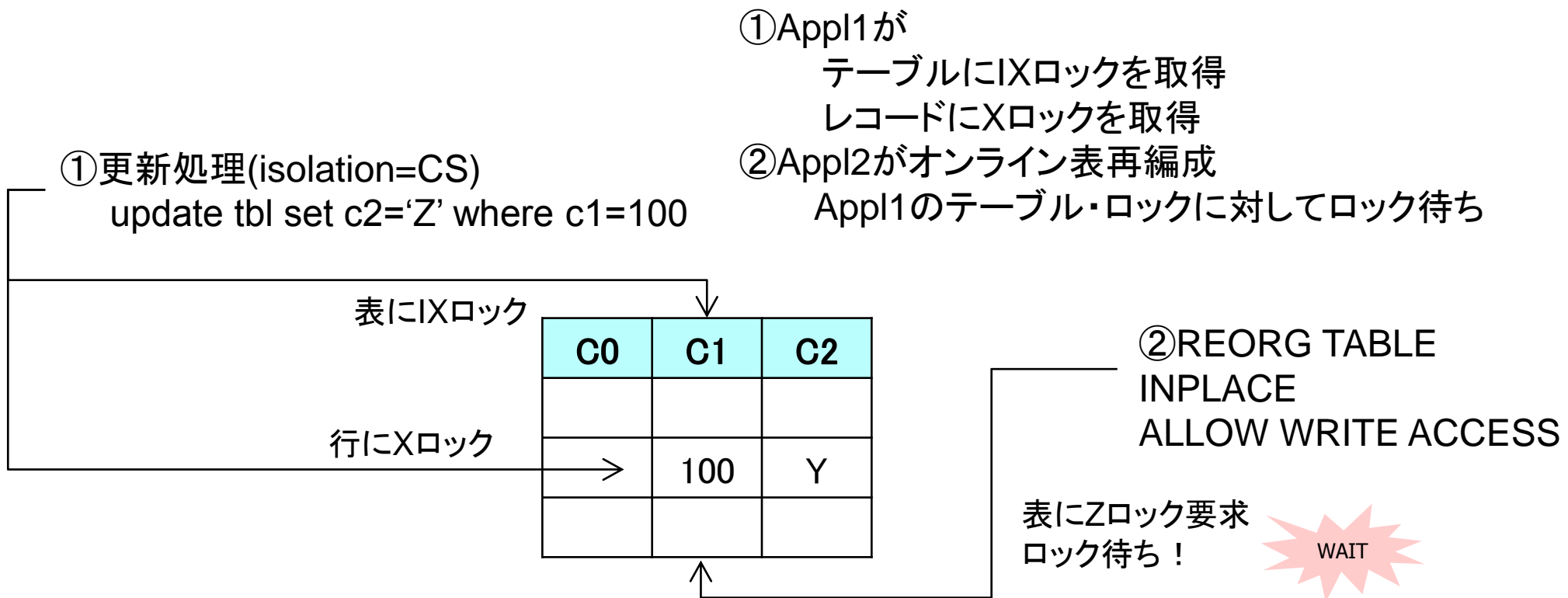
解説: オンライン表再編成のオプションとロック

- オンライン再編成では、ある決まった範囲ごとに再編成しますが、その範囲は、一時点で、新しいRIDと古いRIDの両方を保持しています。その範囲の再編成完了時点で、古いRIDを削除しますが、この時、未コミットのトランザクションで古いRIDを保持しているものがある場合には、そのトランザクションがCOMMITされるまで、再編成はWAITします。ロックタイムアウトにはなりません。従って、インプレイス・アプローチの再編成中には、短いトランザクションによる表へのアクセスは問題ありませんが、長いトランザクションによる表へのアクセスは、再編成のパフォーマンスを下げることになります。
- ALLOW WRITE ACCESS、ALLOW READ ACCESS共に、再編成実行直後、または、RESUMEによる再開直後と、完了直前に表に対する超排他(Z)ロックを取得します。Zロックは保持しつづけるわけではなく、瞬間的に取得し解放します。
- NOTRUNCATE TABLEオプションを指定しない場合(デフォルト)、余分なエクステントの切り捨て(TRUNCATE)時に、表に対する共用(S)ロックを取得します。Sロックは、Zロックと異なり、TRUNCATE処理が完了するまで取り続けます。
- NOTRUNCATE TABLEオプションを指定した場合、余分なエクステントの切り捨ては行われません。このためTRUNCATE処理にかかる時間を短縮することが可能です。余分なエクステントの切り捨て時に取得される表に対する共用(S)ロックも取得されません。ただし、余分なエクステントの切り捨てが行われないため、DELETE処理によって空いたページを開放することはありません。その領域は同一テーブル上のデータのみ利用されることとなります。

REORG TABLE(INPLACE ALLOW WRITE ACCESS) 実行時のロック待機

□ REORG実行直後に瞬間的にZロックを取得

- 表に対してZロックを取得するため瞬間的にアクセス不可



* ロックタイムアウトしないため、
①のトランザクションがCOMMITされる
まで②のオンライン表再編成はWAIT

実行例:REORG TABLE(INPLACE ALLOW WRITE ACCESS)実行時のロック待機

□ ロック待ち再現シナリオ

- 1. Appl1: db2 +c “update tbl set c2= ‘Z’ where c1=100”
- 2. Appl2: db2 +c “reorg table tbl inplace allow write access”
- 3. ロック状況の取得:db2 call monreport.lockwait()

□ monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	14401	Z	14400	IX	TABLE

← Appl2がZロックを取得したいが、IXロックが取得されているため、ロック待ちとなっている

↓ Appl1(14401)とAppl2(14400)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	14401	14400
APPLICATION_ID	*LOCAL. DB2. 140930010421	*LOCAL. lock105. 140930010318
APPLICATION_NAME	db2reorg	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	IX
LOCK_MODE_REQUESTED	Z	-

オフライン表再編成のオプションとロック

REORG TABLEコマンド			
		照会	更新
クラシック	ALLOW READ ACCESS	△(※)	×
		ソート・フェーズ: Uロック ビルド・フェーズ: Uロック 置換フェーズ: Zロック 索引再作成フェーズ: Zロック	
	ALLOW NO ACCESS	×	×
		Zロック	

※ソート・フェーズ、ビルド・フェーズのみ読み取りアクセス可

解説: オフライン表再編成のオプションとロック

- シャドウ・コピー・アプローチでは、4つのフェーズがありますが、置換フェーズと索引再作成フェーズで表に対して超排他(Z)ロックを取得します。
 - ソート・フェーズ: 索引を指定した場合にソートを行う (Uロック)
 - ビルド・フェーズ: シャドウ・コピーを作成 (Uロック)
 - 置換フェーズ: シャドウ・コピーを実表と置換 (Zロック)
 - 索引再作成フェーズ: 索引の再作成を行う (Zロック)
- Zロックを取得しようとする時に、競合するロックが既に取得されている場合には、再編成はロック待機状態になりますが、ロックタイムアウトにはなりません。

REORG TABLE(ALLOW READ ACCESS) 実行時のロック待機

□置換フェーズと索引再作成フェーズでZロックを取得

- 表に対してZロックを取得するため瞬間的にアクセス不可

①表に共用ロックを取得
lock table tbl in share mode

- ①Appl1が
テーブルにSロックを取得
②Appl2がオフライン表再編成
Appl1のテーブル・ロックに対してロック待ち

表にSロック

C0	C1	C2
	100	Y

②REORG TABLE
ALLOW READ ACCESS

表にZロック要求
ロック待ち！

WAIT

* ロックタイムアウトしないため、
①のトランザクションがロックを解放するまで②のオフライン表再編成は
WAIT

実行例:REORG TABLE(ALLOW READ ACCESS)

実行時のロック待機

□ ロック待ち再現シナリオ

- 1. Appl1: db2 +c “lock table tbl in share mode”
- 2. Appl2: db2 +c “reorg table tbl index tbl_idx use temp space1 allow read access”
- 3. ロック状況の取得: db2 call monreport.lockwait()
- 4. 表再編成のモニター: db2pd -db sample -reorgs

□ monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	14534	Z	14417	S	TABLE

← Appl2がZロックを取得したいが、
Sロックが取得されているため、
ロック待ちとなっている

↓ Appl1(14417)とAppl2(14534)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	14534	14417
APPLICATION_ID	*LOCAL. lock105. 140930023938	*LOCAL. lock105. 140930011042
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	S
LOCK_MODE_REQUESTED	Z	-

-- Lock requestor waiting agent and activity --

AGENT_TID	= 13069
REQUEST_TYPE	= REORG

実行例:REORG TABLE(ALLOW READ ACCESS)

実行時のロック待機

db2pd出力結果(抜粋)

表再編成の対象表
の名前が分かる

オフライン表再編成であることを示す

Database Member 0 -- Database SAMPLE -- Active -- Up 0 days 03:07:01 -- Date 2014-09-30-13.17.43.602986

Table Reorg Information:

Address	TbpaceID	TableID	PartID	MasterTbs	MasterTab	TableName	Type	IndexID	TempSpaceID
0x0A00020017D7E8F8	3	8	n/a	n/a	n/a	TBL	Offline	1	1

Table Reorg Stats:

Address	TableName	Start	End	PhaseStart	MaxPhase	Phase Build
0x0A00020017D7E8F8	TBL	2014-09-30 13:15:22	n/a	2014-09-30 13:15:22	4	

CurCount	MaxCount	Status	Completion
32499	69841	Started	0

Zロックが取得できないため、
ビルド・フェーズから置換フェーズに
遷移しない

REORG TABLE参照:

□ [オンライン表再編成のロックおよび並行性に関する考慮事項]

- http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.perf.doc/doc/c0024857.html
 - オンライン再編成実行時に取得されるロックに関する記載があります。

□ [CLASSIC (オフライン) 表再編成]

- http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.perf.doc/doc/c0024781.html
 - オフライン再編成実行時に取得されるロックに関する記載があります。

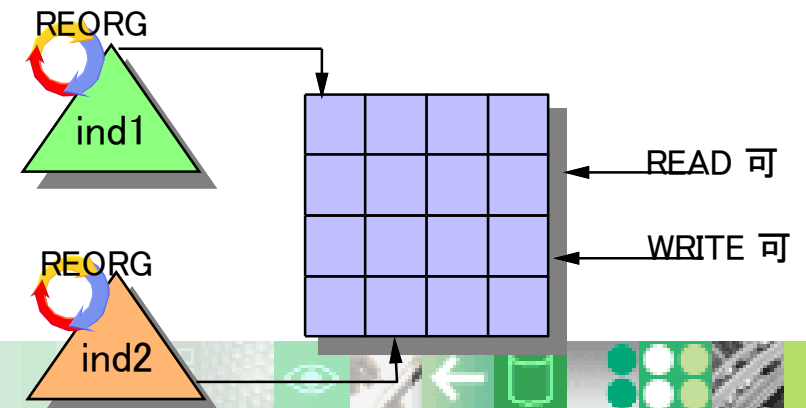
□ 『DB2 V9.5 運用管理ガイド:DB2 for LUW V9.5 データベース・メンテナンス』

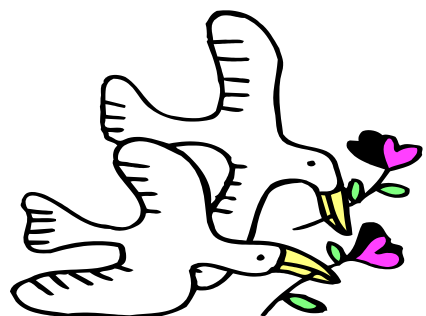
- <https://www.ibm.com/developerworks/community/wikis/home?lang=ja#!/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%20%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9ADB2%20for%20LUW%20V9.5%20%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%83%BB%E3%83%A1%E3%83%B3%E3%83%86%E3%83%8A%E3%83%B3%E3%82%B9>
 - REORGに関する全般的な情報が記載されています。

6. REORG INDEXES

□ 索引再編成

- REORG INDEXES ALL FOR TABLEコマンド
 - 表に対して作成された全ての索引が再編成される
- 排他レベルはALLOWオプションで指定
 - ALLOW NO ACCESSの場合： 表・索引に対する**アクセス不可**
 - 索引はRebuildとマークされ、新規に作成される
 - ALLOW READ ACCESSの場合： 表・索引に対する**参照のみ可能**
 - 既存の索引を残したまま、シャドー索引を作成する
 - ALLOW WRITE ACCESSの場合： 表・索引に対する**参照更新可能**
 - 既存の索引を残したまま、シャドー索引を作成する





ブランク・ページです。

REORG INDEXESオプションとロック

REORG INDEXESコマンド		該当表に対するロック	
		照会	更新
シャドー・アプローチ	ALLOW NO ACCESS	×	×
		Zロック	
	ALLOW READ ACCESS	○(※)	×
		Sロック→(完了直前)Zロック	
	ALLOW WRITE ACCESS	○(※)	○(※)
		(開始直後)Sロック→INロック→ (完了直前)Uロック→Zロック	

※実行直後に瞬間的にアクセス不可

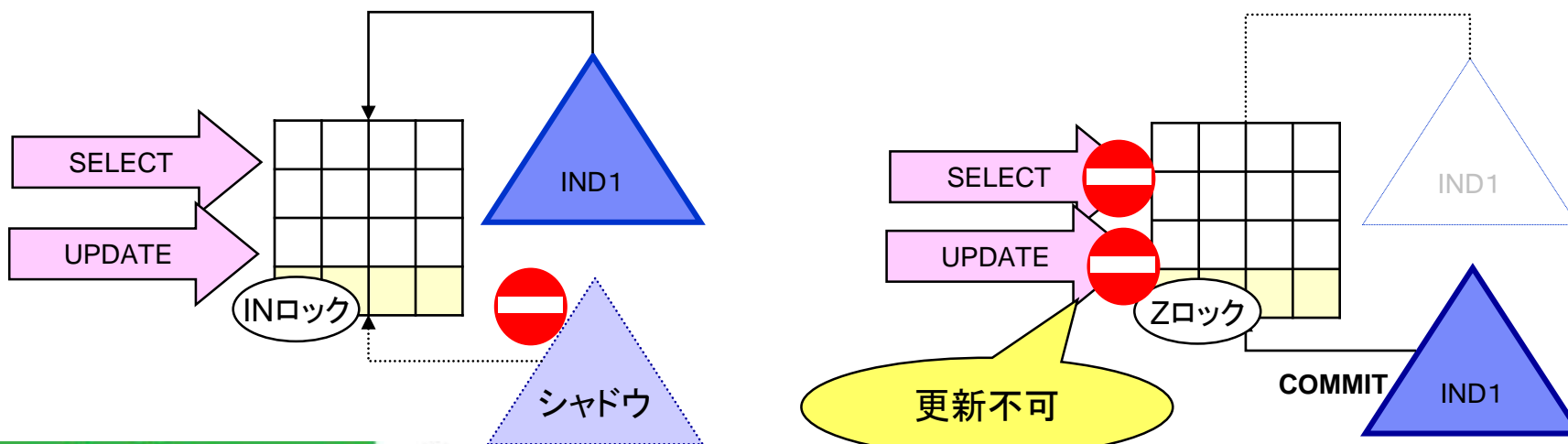
※完了直前にアクセス不可

解説: REORG INDEXESオプションとロック

□ ALLOW READ/WRITE ACCESS指定時

1. 表をスキャン、キーをソート
2. 既存の索引を残したままシャドー索引を作成
 - ALLOW READ ACCESS: 表にSロック
 - ALLOW WRITE ACCESS: 表にINロック
3. 再編成中の更新内容をシャドー索引に適用
 - ALLOW WRITE ACCESS指定時のみ
 - 表に対してUロックが取得される
4. シャドー索引に切り替え、既存の索引を削除
 - 瞬間的に表に対してZロックが取得され、表へのアクセスが拒否される

ALLOW WRITE ACCESSの場合



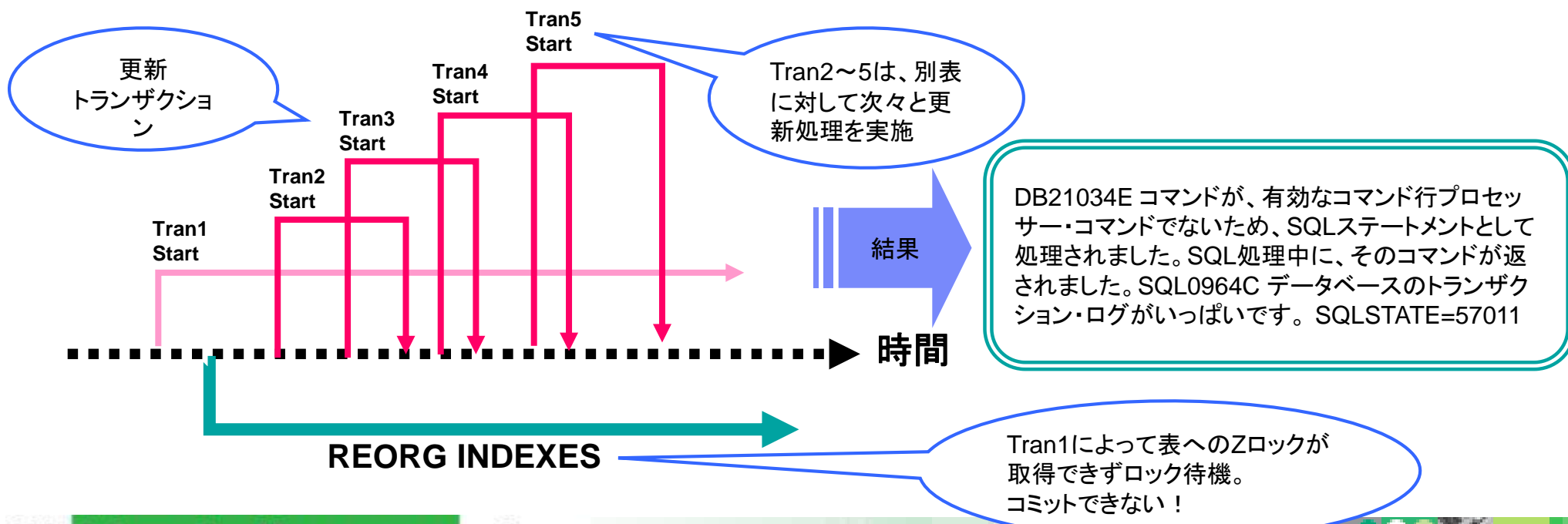
索引再編成(ALLOW WRITE ACCESS)の ロック待機

□ Zロックが取得できずにロック待機した場合 →ロックタイムアウトしない

- シャドー索引を切り替えるときにZロックが取得される
 - 索引再編成がZロック取得待ちでロック待機した場合、ロックタイムアウトでエラーにならずに待ち続ける
- 索引再編成中に、絶え間なく更新処理が発生すると、索引再編成はロック待機状態になり完了しない
 - 索引再編成は、ロング未コミット・トランザクションとして扱われるため、更新処理が重なるとトランザクション・ログがいっぱいのエラーが発生する可能性あり

□ Sロックが取得できずにロック待機した場合 →ロックタイムアウトする

- 索引再編成の開始直後に瞬間的にSロックが取得される
 - 索引再編成がSロック取得待ちでロック待機した場合、ロックタイムアウトする



解説: 索引再編成(ALLOW WRITE ACCESS)のロック待機

□Zロックが取得できずにロック待機した場合

- allow write accessで索引再編成を実行した場合は、以下のタイミングでZロックを取得し、一時的に更新を拒否します。
 - 再編成中の更新内容を適用してシャドウ索引と既存の索引を切り替える際
- この時、索引再編成はZロックが取得できるまで、**ロックタイムアウトでエラーにならずに待ち続けます**。つまり再編成処理が、未コミットのロング・トランザクションとなる可能性があります。
- 頻繁に更新が発生するような時間帯に索引再編成するのは避ける方が良いでしょう。

□Sロックが取得できずにロック待機した場合

- allow write accessで再編成を実行した場合は、索引再編成の開始直後に、瞬間的にSロックが取得されます。
- ロックタイムアウトを設定していた場合、索引再編成はSロックが取得できないと、**ロックタイムアウトとなりロールバックされます**。
- REORG 操作がロールバックされた場合は、元の索引がそのまま維持されます。
- 索引再編成がロック待機となる原因となった更新トランザクションが作業単位を終了(コミットまたはロールバック)した時点で、索引再編成にSQL0911Nエラー理由コード=68が返されます。
- ロングトランザクションが発生するような時間帯に索引再編成するのは避ける方が良いでしょう。

REORG INDEXES(ALLOW WRITE ACCESS) 実行時のロック待機

□再編成中の更新内容をシャドー索引に適用するタイミング でSロックを取得

- Sロックが取得できずに索引再編成が待機した場合ロックタイムアウトする可能性あり

LOCKTIMEOUT(DB構成パラメーター)=30秒
に設定されたデータベース

①更新処理(isolation=CS)

update tbl set c2='Z' where c1=100

(例)40秒経過

表にIXロック

C0	C1	C2
→	100	Y

行にXロック

③更新処理のコミット

①Appl1が

テーブルにIXロックを取得

②Appl2がオンライン索引再編成

Appl1のテーブル・ロックに対してロック待ち

③Appl1が更新処理を開始してから

40秒後にコミット

④Appl2の索引再編成がロックタイムアウトで失敗

②REORG INDEXES ALLOW WRITE ACCESS

表にSロック要求
ロック待ち！

WAIT

④REORGがロックタイムアウトで ロールバック

* ③がコミットされたタイミングで索引
再編成にSQL0911Nエラー理由コード
=68が返される

実行例:REORG INDEXES(ALLOW WRITE ACCESS)

実行時のロック待機

□ ロック待ち再現シナリオ

- 1. Appl1: db2 +c “update tbl set c2= ‘Z’ where c1=100”
- 2. Appl2: db2 +c “reorg indexes all for table tbl allow write access”
- 3. ロック状況の取得: db2 call monreport.lockwait()
- 4. 表再編成のモニター: db2pd -db sample -reorgs index

□ monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	23	S	7	IX	TABLE

← Appl2がSロックを取得したいが、IXロックが取得されているため、ロック待ちとなっている

↓ Appl1(7)とAppl2(23)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	23	7
APPLICATION_ID	*LOCAL. lock105. 141003051026	*LOCAL. lock105. 141003051010
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	IX
LOCK_MODE_REQUESTED	S	-

-- Lock requestor waiting agent and activity --

AGENT_TID = 12853
REQUEST_TYPE = REORG

実行例:REORG INDEXES(ALLOW WRITE ACCESS) 実行時のロック

□ db2pd出力結果(抜粋)

```
Index Reorg Stats:  
Retrieval Time: 2014-10-03 15:57:40  
TbpaceID: 3      TableID: 8  
Schema: LOCK105  TableName: TBL  
Access: Allow write  
Status: Stopped  
Start Time: 2014-10-03 15:57:11   End Time: -  
Total Duration: -  
Prev Index Duration: -  
Cur Index Start: 2014-10-03 15:57:11  
Cur Index: 1      Max Index: 1      Index ID: 1  
Cur Phase: 0      ( - )      Max Phase: 3  
Cur Count: 0      Max Count: 0  
Total Row Count: -
```

索引再編成の対象表
の名前が分かる

アクセス・モードが分かる

Sロックが取得できないため、
索引再編成の処理が開始できない
ことを示す

REORG INDEXES参照:

□ [オンライン索引再編成のロックおよび並行性に関する考慮事項]

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.perf.doc/doc/c0056911.htm
!
 - 索引再編成実行時に取得されるロックに関する記載があります。

□ 『DB2 V9.5 運用管理ガイド:DB2 for LUW V9.5 データベース・メンテナンス』

- <https://www.ibm.com/developerworks/community/wikis/home?lang=ja#!/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9ADB2%20for%20LUW%20V9.5%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%83%BB%E3%83%A1%E3%83%B3%E3%83%86%E3%83%8A%E3%83%B3%E3%82%B9>
 - REORGに関する全般的な情報が記載されています。

7. REORGCHK

□ REORGCHK実行中に、アプリケーションからの参照・更新可能

□ 統計を更新するか否かで取得されるロックが異なる

- UPDATE STATISTICSオプション(デフォルト)
 - 先にRUNSTATSを実行してカタログ統計情報を更新した後、REORGCHKが実行される
 - RUNSTATSはTABLE and INDEXES ALLオプションで実行(SYSCAT.INDEXESの情報も更新される)
 - RUNSTATSはALLOW WRITE ACCESSが使用される
 - ◆ a. SYSTABLESに対してIS/NS-Lock
 - ◆ b. 該当表に対してIN-Lock(UR)でアクセス
 - ◆ c. カatalog更新(IX,X-Lock)
- CURRENT
 - 現在の統計情報をもとにして、再編成が必要かどうかを判定する

解説: REORGCHK

- REORGCHKは、表と索引に対して検査を行いREORG(再編成)の必要性を判断する。
- REORGCHKはデフォルトではRUNSTATSを伴う処理を行う。
- REORGCHKを行い、なおかつ統計情報を更新したくない場合
 - REORGの必要性だけを調べたい場合
 - 例えば、以下のような手順により実現できる
 - db2 +c reorgchk on table staff_reorg
 - db2 rollback

REORGCHK出力結果(抜粋)

```
Table statistics:

F1: 100 * OVERFLOW / CARD < 5
F2: 100 * (Effective Space Utilization of Data Pages) > 70
F3: 100 * (Required Pages / Total Pages) > 80

SCHEMA  NAME                CARD  OV  NP  FP ACTBLK  TSIZE  F1  F2  F3  REORG
-----
Table: LOCK105.STAFF_REORG
LOCK105  STAFF_REORG        148896 171 1709 3252   - 6849216 0 52 52 -**
```

“*”のついた項目を持つ表がREORG対象の候補となる。

REORGCHKオプションとロック

REORGCHKコマンド	該当表に対するロック		SYSTABLES	カタログ更新
	照会	更新		
UPDATE STATISTICS	○	○	表レベルISロック 行レベルNSロック	表レベルIXロック 行レベルXロック
	表レベルINロック (分離レベルUR)			
CURRENT STATISTICS	○	○	表レベルISロック 行レベルNSロック	—
	—			

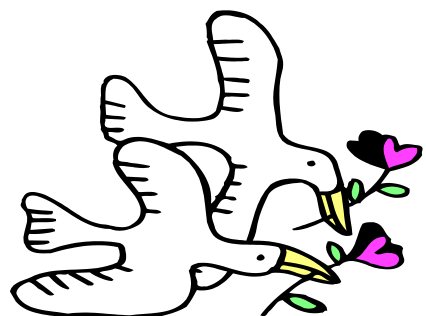
REORGCHK参照:

□ 『DB2 V9.5 運用管理ガイド:DB2 for LUW V9.5 データベース・メンテナンス』

- <https://www.ibm.com/developerworks/community/wikis/home?lang=ja#!/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9.5%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9ADB2%20for%20LUW%20V9.5%E3%83%87%E3%83%BC%E3%82%BF%E3%83%99%E3%83%BC%E3%82%B9%E3%83%BB%E3%83%A1%E3%83%B3%E3%83%86%E3%83%8A%E3%83%B3%E3%82%B9>
 - REORGCHKに関する全般的な情報が記載されています。

8. ONLINE BACKUP

- ONLINE BACKUP中に、アプリケーションからの参照・更新可能
- ONLINE BACKUPと同時に実行できないユーティリティやDDLステートメントがあるため注意(後述)
- ONLINE BACKUP中に取得するロックは、SMS表スペースとDMS表スペースで異なる
 - DMS表スペースの場合には、表に対するロックは取得しない
 - SMS表スペースの場合には、表に対してINロックを取得する
 - SMS表スペース内の表に対して、以下のようなZロックを必要とするDDLを実行し未コミットの状態であると、ONLINE BACKUPは失敗し、ロックタイムアウト(SQL0911Nエラー理由コード=68)となる
 - DROP TABLE, ALTER TABLE, CREATE VIEW, etc
 - V9.5以降、オンライン・バックアップ中に、LOB データが含まれる表に対して保持されていたSロックは取得されない



ブランク・ページです。

ONLINE BACKUP(DMS表スペース)

□ ONLINE BACKUPとロック

- ONLINE BACKUP対象の表スペースがDMSであった場合

BACKUPコマンド	ユーザー表スペース		カタログ表スペース
ONLINE	INロック		INロック
	表		カタログ表
	照会	更新	—
	○	○	
	—		

ONLINE BACKUP(SMS表スペース)

□ ONLINE BACKUPとロック

- ONLINE BACKUP対象の表スペースがSMSであった場合

BACKUPコマンド	ユーザー表スペース		カタログ表スペース
ONLINE	INロック		INロック
	表		カタログ表
	照会	更新	INロック
	○	○	
	INロック		

Drop Tablespace実行(未コミット)後のOnline Backupのロック待機

❑表スペースを削除(未コミット)後にONLINE BACKUPを実行するとロック待機

- ロックタイムアウトに設定した時間の経過後、ONLINE BACKUPがSQL2048Nエラー(理由コード2)で失敗する

LOCKTIMEOUT(DB構成パラメーター)=30秒
に設定されたデータベース

①表スペース削除

drop tablespace testtbs1

表スペースにZロック

表スペース testtbs1		
	100	Y

30秒経過

①Appl1が

表スペースにZロックを取得

②Appl2がオンライン・バックアップ

Appl1の表スペース・ロックに対してロック待ち
(Appl1が表スペースの削除を開始してから30秒経過してもコミットしない)

③Appl2がSQL2048Nエラーで失敗

②ONLINE BACKUP

表スペースにINロック要求
ロック待ち！

WAIT

③ONLINE BACKUPに
SQL2048Nエラー(理由コード2)
が返される

実行例: Drop Tablespace実行(未コミット)後の Online Backupのロック待機

□ ロック待ち再現シナリオ

- ① Appl1: db2 +c “drop tablespace testtbs1”
- ② Appl2: db2 “backup database sample online”
- ロック状況の取得: db2 “call monreport.lockwait()”

□ monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK MODE	LOCK_OBJECT_TYPE
1	3636	IN	3608	Z	TABLESPACE

← Appl2がINロックを取得したいが、
Zロックが取得されているため、
ロック待ちとなっている

← 表スペースに対するロックの競合

↓ Appl1(3608)とAppl2(3636)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	3636	3608
APPLICATION_ID	*LOCAL. lock105. 140811015846	*LOCAL. lock105. 140811014850
APPLICATION_NAME	db2bp	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	Z
LOCK_MODE_REQUESTED	IN	-

オンライン・バックアップと他のユーティリティーの互換性(1/2)

□ オンライン・バックアップと互換性がないユーティリティー

- 下記ユーティリティーを実行中に、オンライン・バックアップを開始すると、バックアップが失敗する
 - REORG TABLE(オフライン)
 - RESTORE DATABASE
 - ROLLFORWARD DATABASE
 - LOAD
 - ALLOW NO ACCESSオプションを指定
 - COPY NO + ALLOW READ ACCESSオプションを指定
 - SET WRITE
 - BACKUP DATABASE(ONLINEオプションを指定)

オンライン・バックアップと他のユーティリティーの互換性(2/2)

□ オンライン・バックアップと並行実行を避けるべきユーティリティー

- オンライン・バックアップとロック競合が発生する可能性があり、並行して実行できない可能性がある
 - IMPORT(REPLACEオプションを指定)
 - LOAD(COPY YES + ALLOW READ ACCESSオプションを指定)
 - REORG TABLE(オンライン)
 - ストレージ・グループDDL
 - CREATE STOGROUP
 - ALTER STOGROUP
 - DROP STOGROUP
 - RENAME STOGROUP
 - ALTER DATABASE
 - SMS表スペース
 - ALTER TABLE、DROP TABLE、DROP INDEX
 - CREATE INDEX
 - REORG INDEX
 - システム・カタログ表スペースがSMS表スペース
 - RUNSTATS
 - DMS表スペース
 - CREATE INDEX
 - REORG INDEX
 - ALTER TABLESPACE
 - AUTORESIZEオプションを指定
 - コンテナサイズを変更する操作
 - TRUNCATE TABLE

ONLINE BACKUP参照:

□ [BACKUP DATABASE コマンド]

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0001933.html?cp=SSEPGG_10.5.0%2F3-6-2-4-8

➤ ONLINE BACKUP実行時に取得されるロックに関する記載があります。

□ [オンライン・バックアップと他のユーティリティの互換性]

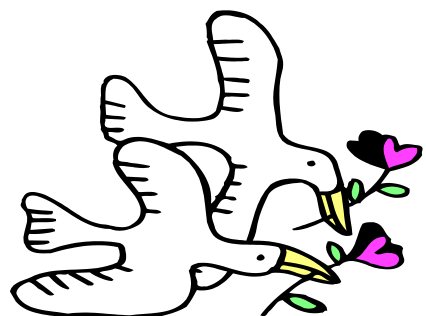
- http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.ha.doc/doc/c0021554.html

➤ ONLINE BACKUPと同時に実行してはいけないユーティリティが分かります。

□ 『DB2 V9 運用管理ガイド: バックアップ・リカバリーの基礎』

- <https://www.ibm.com/developerworks/community/wikis/home?lang=ja#!/wiki/Information%20Management%20Wiki%20-%20Japan/page/DB2%20V9%E9%81%8B%E7%94%A8%E7%AE%A1%E7%90%86%E3%82%AC%E3%82%A4%E3%83%89%EF%BC%9A%E3%83%90%E3%83%83%E3%82%AF%E3%82%A2%E3%83%83%E3%83%97%E3%83%BB%E3%83%AA%E3%82%AB%E3%83%90%E3%83%AA%E3%83%BC%E3%81%AE%E5%9F%BA%E7%A4%8E>

➤ BACKUPに関する全般的な情報が記載されています。



ブランク・ページです。

9. INGEST

□ DB2 V10.1の新機能として登場

- INGESTは、特定のエディションのみ使用可能

➤ [DB2 フィーチャーおよび DB2 製品エディションの機能]

□ 行ロックを使用するため、同じ表に対する他のユーザーからの処理への影響は、最小限に抑えられる

□ 大量データを高速に移動させることが可能

□ 使用可能なDMLオペレーション

- INSERT
- UPDATE
- DELETE
- MERGER
- REPLACE



解説: INGEST

- ❑ IMPORTは順次INSERTを実行するのに対して、INGESTは並列でINSERTを実行するため高速です。
- ❑ LOADは対象表に対して他のアプリケーションからの更新はできませんが、INGESTは更新ができるので、同時実行性が高いと言えます。
- ❑ 一方、INGESTはトランザクション・ログに書き込むため、LOADよりパフォーマンスは劣ります。

- ❑ INGESTが行ロックングを使用する例を説明します。
- ❑ IMPORT(REPLACE)であれば、IMPORTを実行すると内部的にLOCK TABLE xxx IN EXCLUSIVE MODEが発行され、表レベルにXロックを取得します。
- ❑ INGEST(REPLACE)であれば、内部的にLOCK TABLEは発行されません。表レベルのXロックを取得せずに行レベルのXロックを取得してデータを処理します。

- ❑ INGESTは、時間(commit_period)または行数(commit_count)に基づいてコミットの頻度を決定します。デフォルトは、1秒間隔でコミットが実行されます。
 - (実行例)
 - \$ db2 "ingest set commit_count 2000"
 - \$ db2 "ingest set commit_period 0"
 - \$ db2 "ingest from file ingtbl.del format delimited insert into ingtbl"

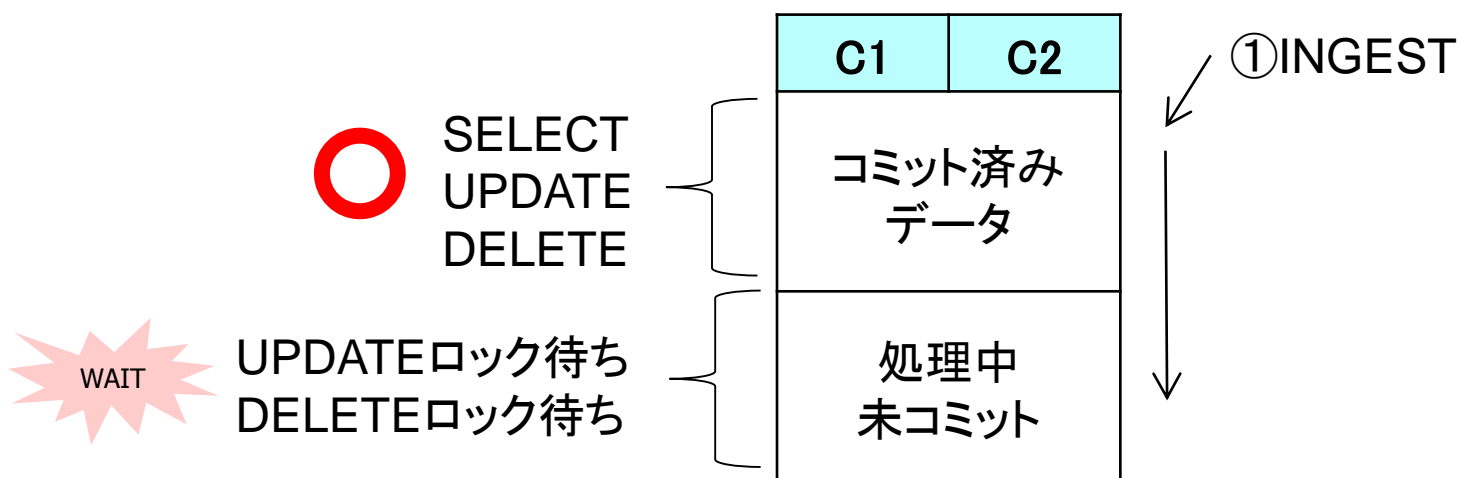
INGESTのロック

□INGEST中は表への参照可能

- INGEST中のデータはコミット済みのデータのみ参照可能

□INGESTとロック競合にならないデータは更新が可能

- INGEST対象表にはIXロック、更新行にはXロックを取得する



INGEST実行時のロック待機

□INGEST対象行とアプリケーション更新対象行のロックが競合したとき

①更新処理(isolation=CS)
update ingtbl set c2='Z' where c1=10

表にIXロック

行にXロック

C1	C2
10	A

- ①Appl1が
テーブルにIXロックを取得
レコードにXロックを取得
- ②Appl2がINGEST REPLACE
Appl1の行ロックに対してロック待ち

②INGEST REPLACE

行にXロック要求
ロック待ち！

WAIT

実行例:INGEST実行時のロック待機(1/2)

□ロック待ち再現シナリオ

- ①Appl1:db2 +c “update tbl set c2=‘Z’ where c1=10”
- ②Appl2:db2 “ingest from file ingtbl.del format delimited replace into ingtbl”
- ロック状況の取得:db2 call monreport.lockwait()

□monreport.lockwait出力結果(抜粋)

Part 1 - Summary of current lock waits

#	REQ_APPLICATION HANDLE	LOCK_MODE REQUESTED	HLD_APPLICATION _HANDLE	LOCK_ MODE	LOCK_OBJECT_TYPE
1	7030	X	6889	X	ROW

← Appl2がXロックを取得したいが、
Xロックが取得されているため、
ロック待ちとなっている

↓ Appl1(6889)とAppl2(7030)の状況

-- Requestor and holder application details --

Attributes	Requestor	Holder
APPLICATION_HANDLE	7030	6889
APPLICATION_ID	*LOCAL. lock105. 140905022006	*LOCAL. lock105. 140905005417
APPLICATION_NAME	DB2_INGEST	db2bp
SESSION_AUTHID	LOCK105	LOCK105
MEMBER	0	0
LOCK_MODE	-	X
LOCK_MODE_REQUESTED	X	-

実行例:INGEST実行時のロック待機(2/2)

monreport.lockwait出力結果(抜粋)つづき

-- Lock details --

LOCK_NAME	=	000300090000000000000000D52
LOCK_WAIT_START_TIME	=	2014-09-05-11. 20. 09. 296077
LOCK_OBJECT_TYPE	=	ROW
TABSCHEMA	=	LOCK105
TABNAME	=	INGTBL
ROWID	=	13
LOCK_STATUS	=	W
LOCK_ATTRIBUTES	=	0000000000400000
ESCALATION	=	N

← LOCK105.INGTBL表への行ロックの競合

-- Lock requestor waiting agent and activity --

AGENT_TID	=	14472
REQUEST_TYPE	=	EXECIMMD
ACTIVITY_ID	=	1
UOW_ID	=	9
LOCAL_START_TIME	=	2014-09-05-11. 20. 09. 295912
ACTIVITY_TYPE	=	WRITE_DML
ACTIVITY_STATE	=	EXECUTING

Appl2(INGEST)が発行している statement →
REPLACEのためDELETE発行されている

STMT_TEXT	=	
DELETE FROM ingtbl WHERE 1 = 1		

INGEST参照:

□ INGEST、インポート、およびロードの各ユーティリティの比較

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0057291.html
 - INGEST、IMPORT、LOADの類似点、相違点が記載されています。

□ INGESTユーティリティ

- http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dm.doc/doc/c0057237.html
 - INGESTの概要が記載されています。

□ INGESTコマンド

- http://www-01.ibm.com/support/knowledgecenter/api/content/SSEPGG_10.5.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0057198.html
 - INGESTのコマンド・シンタックスが記載されています。

□ DB2 10 for LUW 新機能ワークショップ資料

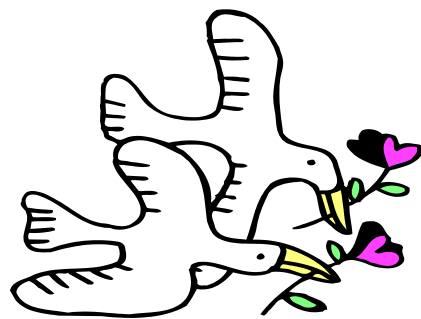
- 5章:新しい運用管理ツールとアプリケーション開発の生産性向上
- http://www.ibm.com/developerworks/jp/data/products/db2/db2_10-workshop/
 - INGESTIに関するワークショップ資料です。

□ Introduction to the DB2 Continuous Data Ingest feature (英語)

- <http://www.ibm.com/developerworks/data/library/techarticle/dm-1304ingestcmd/index.html?ca=dbg-twodw20130503>
 - INGESTの使い方について記載されています。

□ DB2 utilities

- <http://www.ibm.com/developerworks/data/tutorials/db2-cert6115/>
 - DB2ユーティリティに関する情報が記載されています。



ブランク・ページです。