DB2 Training

1. **产品类型、版本和历史**

|  |  |
| --- | --- |
| 版本 | 产品 |
| V9.1 | CLIENT, RTCL, ESE, WSE, EXP, PE, CONSV |
| V9.5 | CLIENT, RTCL, ESE, WSE, EXP, PE, CONSV |
| V9.7 | CLIENT, RTCL, ESE, WSE, EXP, PE, CONSV, AESE |
| V10.5 | CLIENT, RTCL, SERVER, EXP, CONSV |

RTCL: Runtime Client

ESE: Enterprise Server Edition

WSE: Workgroup Edition

EXP: DB2 Express Edition/ Express-C(no-charge)/Express-C FTL

PE: Personal Edition

CONSV: Connect Server

AESE: Advanced ESE

SERVER: ESE, AESE, WSE, AWSE

DB2 Personal Edition（DB2 Personal）是运行在基于桌面的普通硬件上的单用户数据服务器。DB2 Personal 可用在基于 Intel 和 AMD 的 Windows 和 Linux 工作站上。

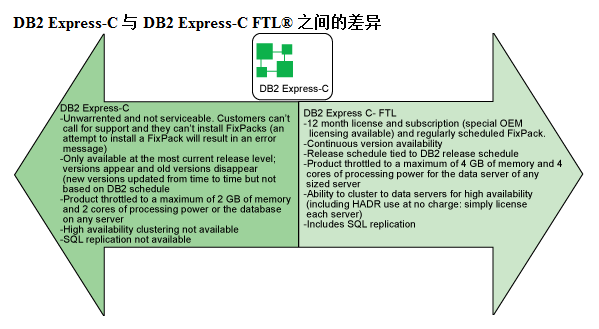
DB2 Personal 具有 DB2 Express Edition 中的大多数特性，但是它无法用作中心数据库服务器，因为远程客户机不能连接到这个 DB2 版本的数据库。

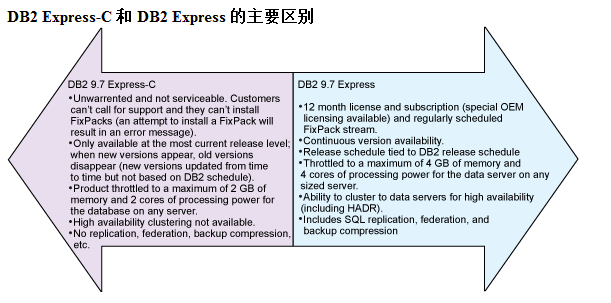
**DB2 Personal Edition (PE) 自 10.1 版开始不再可用。**

DB2 Express-C： 免费 DB2 数据服务器版本，对 RAM 和处理器设置了限制。

DB2 Express-C Fixed Term License（FTL）：FTL 是在DB2 Express-C这个 DB2 版本上增加的订购选项。12个月的License定期合同，提供fixpack，7\*24支持，HADR，复制等DB2 Express-C不支持的特性。因为 DB2 Express-FTL 与 DB2 Express 非常相似，**所以从V9.7开始删除了 DB2 Express-C FTL 包。**

DB2 Express Edition（DB2 Express）是一个入门级的具有完整支持的 DB2 数据服务器。



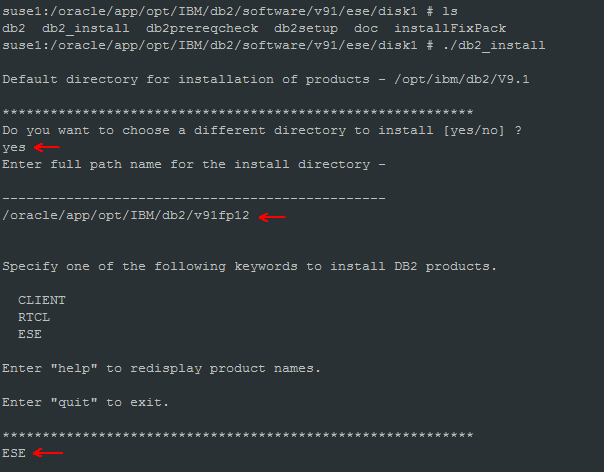


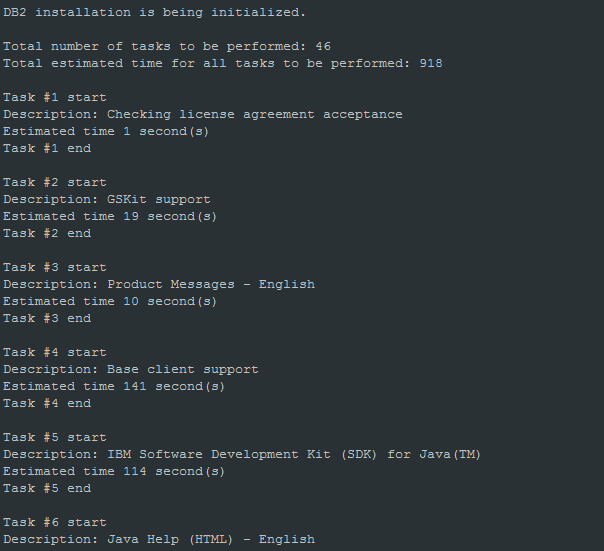
DB2 Workgroup Server Edition : 是提供与 DB2 Express 数据服务器相同的功能的 DB2 数据服务器，但它是为需求更高的工作负载设计的，支持更多内存、更大处理能力和更高的可用性，并提供许多平台部署选项。与 DB2 Express 不同（DB2 Express 只能在 Windows、基于 x86 的 Solaris 和 Linux 操作系统上运行），在 DB2 支持的所有分布式平台上都可以运行这个 DB2 版本，即：Linux、Windows、AIX、Solaris 和 HP-UX。

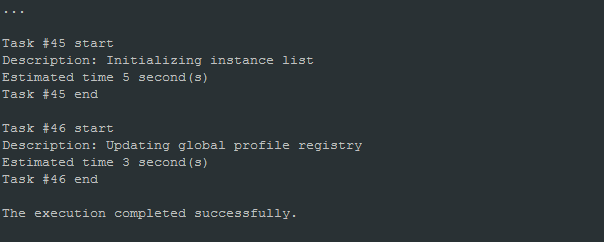
DB2 Enterprise Server Edition: 是 IBM 功能最齐全的支持 Web 的客户机/服务器数据服务器。在 DB2 支持的所有分布式平台上都可以运行这个 DB2 版本，即：Linux、Windows、AIX、Solaris 和 HP-UX。 旨在用作大中型的部门服务器，在所有 DB2 9 版本中它提供的特性和服务最为全面。没有CPU，内存使用限制。

1. **产品安装(以V9.1为例)**

root用户：

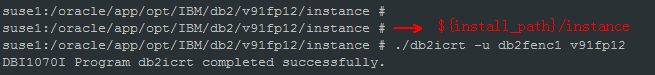






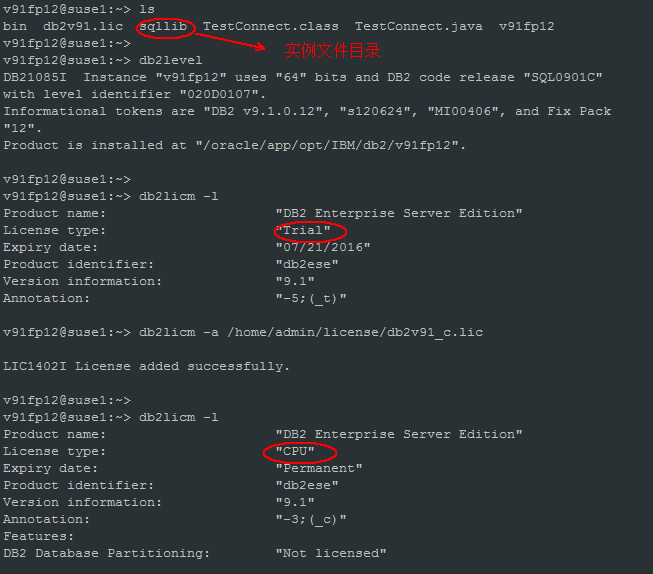
1. **实例创建**

root用户：

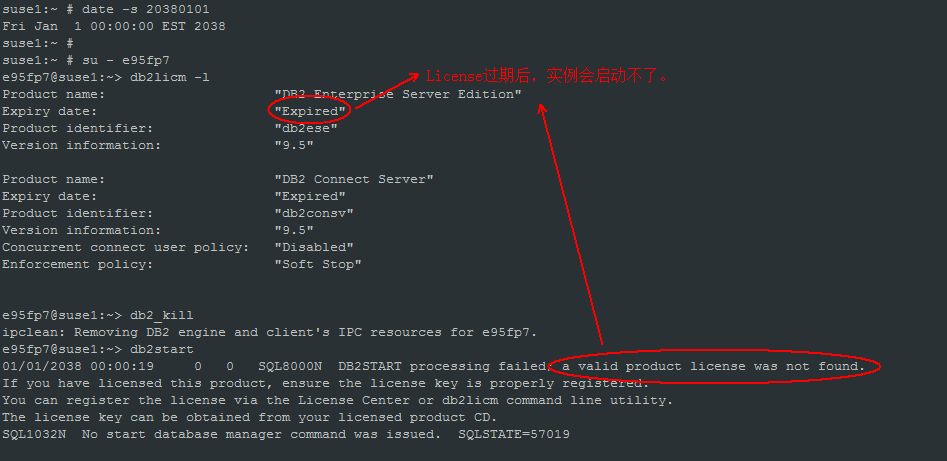


1. **License安装**

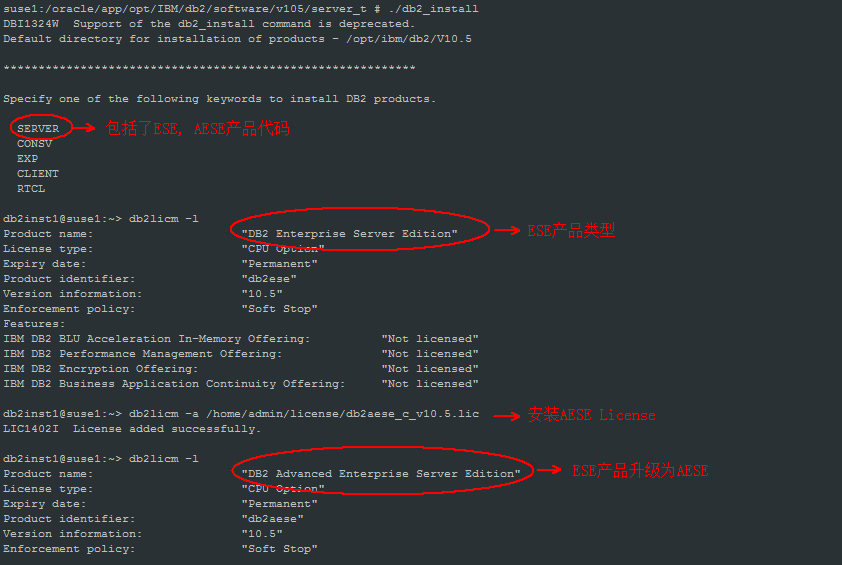
su – v91fp12



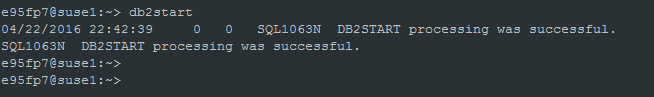
License过期后，实例会启动不了。



**V10.5以前，如果要升级产品，比如V9.7，从ESE升级为AESE， 需要重新安装AESE产品。但V10.5开始，安装产品时选择SERVER类型即包括了ESE,AESE,WSE等产品，从ESE升级到AESE，不需要重新安装AESE产品，只需要安装AESE的License即可。**



1. **实例启动**

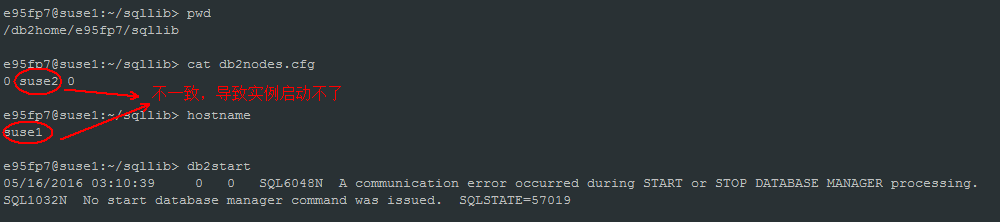


* 1. **实例启动不了异常处理**

1. 清理ipc资源，ipclean –a

实例异常终止后，实例启动不了，可能是IPC资源未清理干净。

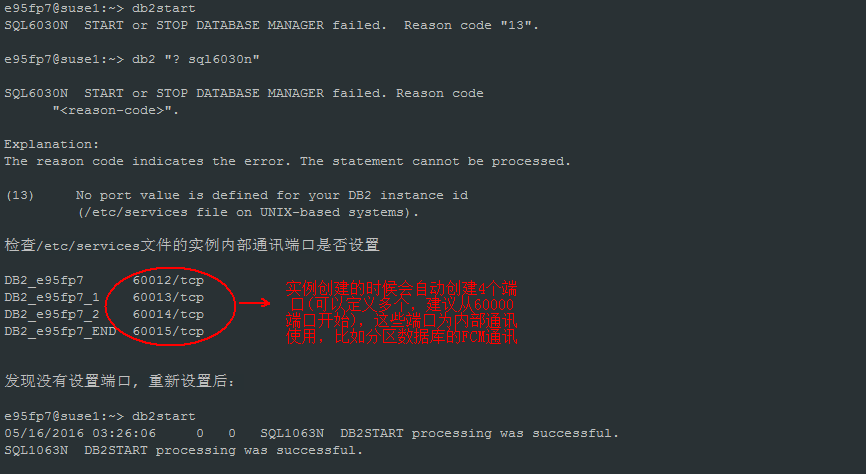
1. 检查db2nodes.cfg文件，主机名称是否正确



1. 检查db2diag.log日志文件

db2diag.log日志可以发现一些问题，比如hostname不一致。

1. 检查/etc/services，实例内部通讯端口是否设置



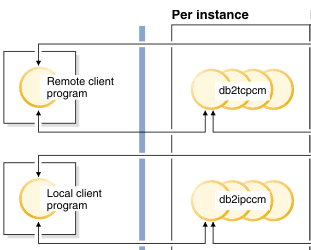
1. License是否过期

License过期，实例会启动不了。

1. 重启服务器

如果尝试多种方法后，实例还是无法启动，可以重启服务器看是否能解决。

* 1. **监听进程**



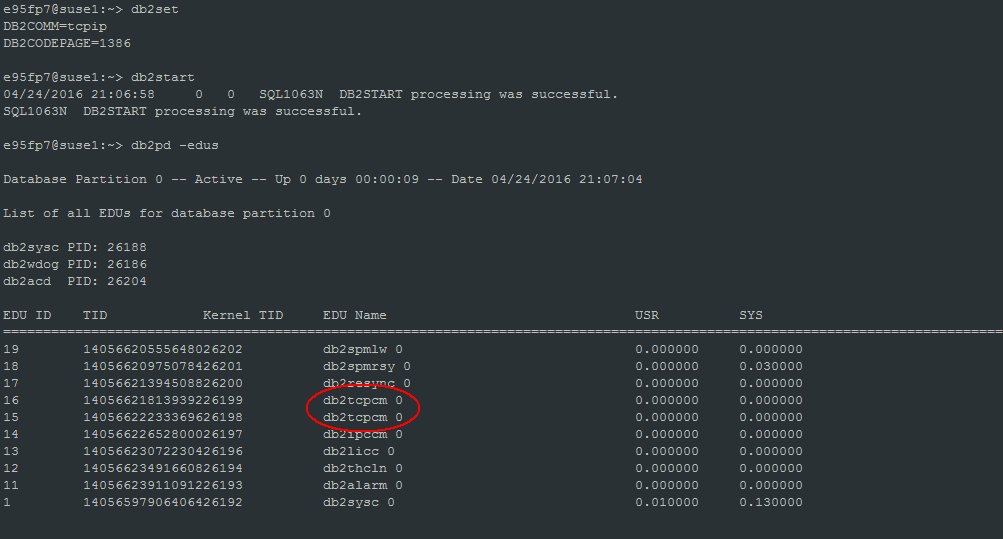
1. 远程db2tcpcm

通过设置DB2COMM=tcpip激活，如果不设置该参数，远程连接访问不了数据库。

禁止远程连接：

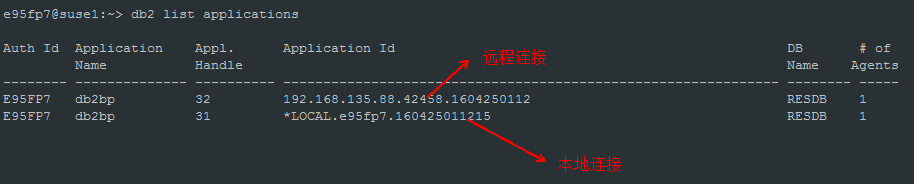
db2set DB2COMM=

重启实例



1. 本地 db2ipccm

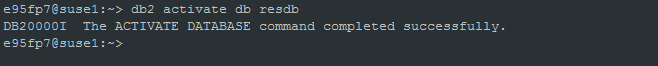
本地连接，指在数据库服务器本机上访问数据库，ipc进程间通信。



1. **数据库创建**



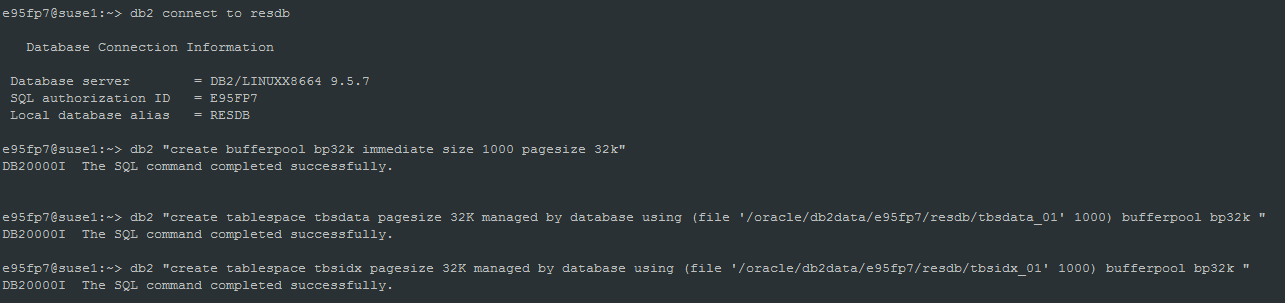
1. **数据库激活**
2. activate db

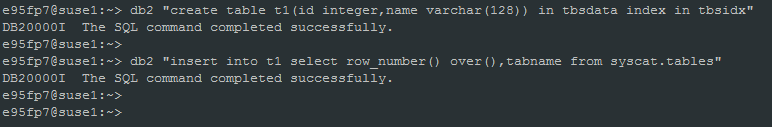


1. 第一次connect to db

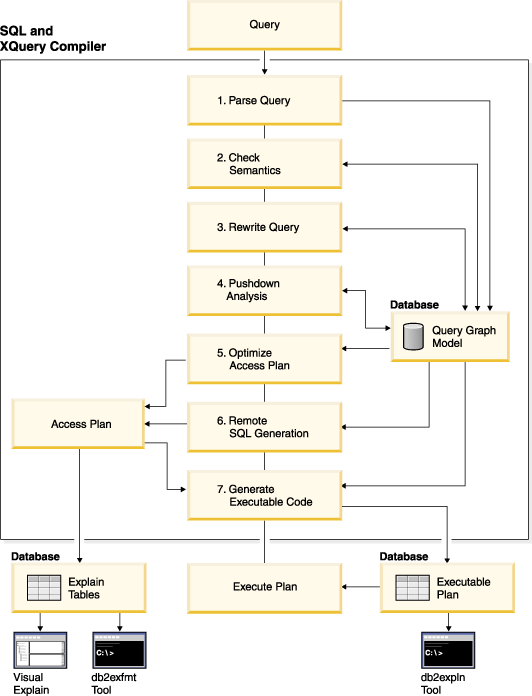
通过这种方式激活的数据库，当所有数据库连接都断开的时候，数据库会deactivate.

1. **缓冲池、表空间**





1. **SQL生命周期 & 后台过程**
   1. **SQL语句执行过程**



1.语法分析（Parse Query）

SQL语句被提交给SQL编译器，编译器分析该语句，检查其语法（Parse Query：语法分析），如果存在语法错误，编译器就停止处理并返回错误信息；如果不存在语法问题，编译器会将SQL语句转换为可被优化器分析的关系代数语句，并据此创建该查询的查询图模型（Query Graph Model，又称语法树）。

2.语义检查（Check Semantics）

语法分析完成后，编译器会根据查询图模型进行语义检查（比如检查语句中的数据类型是否与数据库的表列的数据类型一致），语义检查完成后也会将相关信息添加到查询图模型，包括参考约束，表检查约束，触发器，和视图信息等。

3.查询重写（Rewrite Query）

如果SQL语句的语法语义都没有问题，就可以正式进行查询操作了。这是优化器进行查询优化的开始阶段，DB2优化器三大组件之一的查询重写器（Query Rewriter）就是处理这一工作的。其目的是将提交的SQL语句优化成效率更高的形式，这种优化可以是基于查询成本的考虑，也可以是基于查询规则的考虑。

4.优化访问计划（Optimizer Access Plan）

根据查询图模型提供的信息，优化器会生成许多能够满足查询请求的访问计划（执行方案方案），然后优化器综合系统编目表中关于表，索引，列和函数等等的统计信息，估计每种访问计划的执行成本，并选择具有最小成本的方案作为最终的访问计划（Acess Plan）。这个进行优化选择的工作是由DB2优化器的另两大组件成本评估器（Estimator）和计划生成器（Plan Generator）完成的。需要注意的是：优化器内部有若干优化度不同的优化算法，优化度越高（优化效果好）的算法所需要的优化时间也是越长的。优化器会根据设定的优化级来选择不同的优化算法。通常数据量越大，越复杂的查询会使用优化度更高的算法，因为执行优化算法所花费的时间对于优化后的查询所带来的缩短查询时间的收益而言是微不足道的。

5.生成可执行代码（Generate Executable Code）

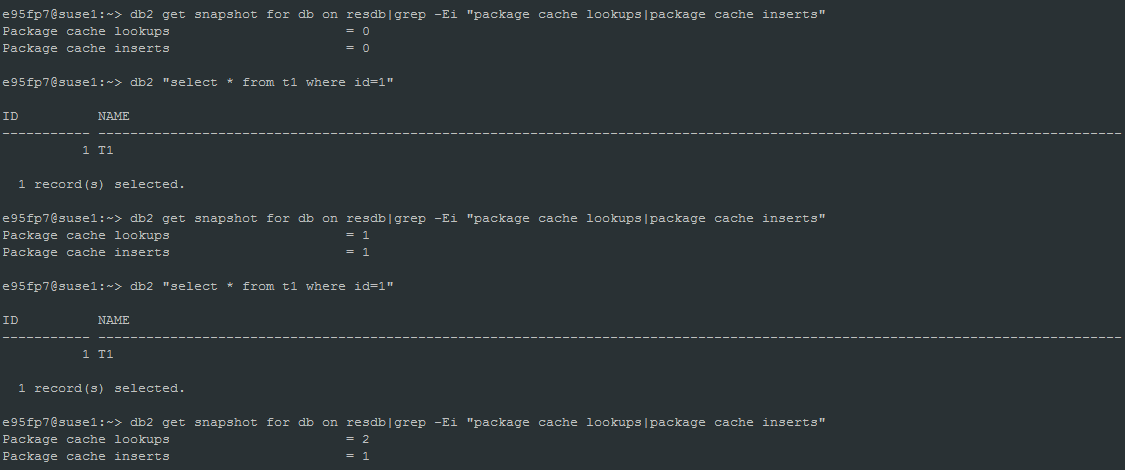
根据最终选定的访问计划生成执行代码，类似C语言编译后生成可被机器识别的机器码一样。

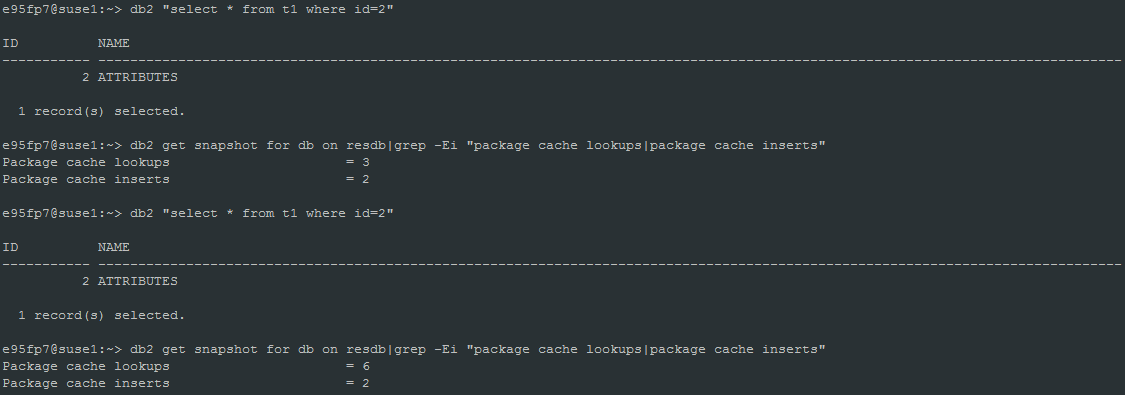
6.执行访问计划（Execute Plan）

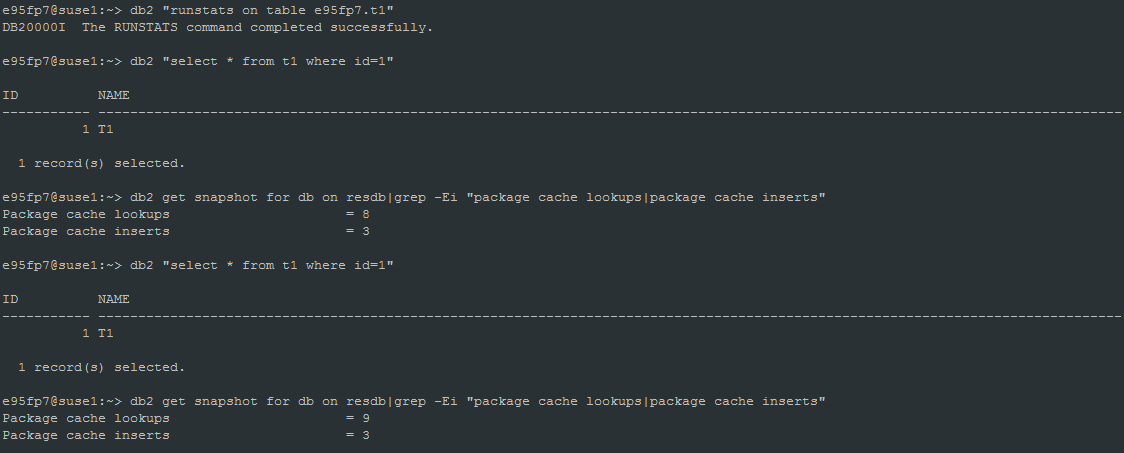
执行可执行代码，获取查询结果集。

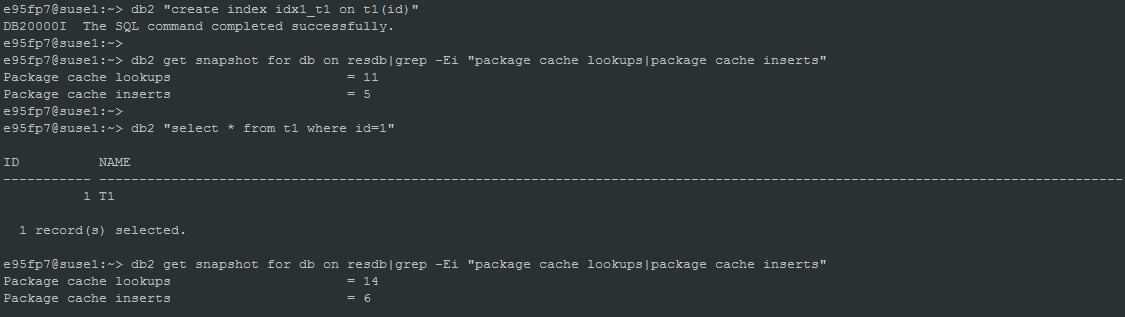
每一条单独的 SQL 请求都将被映射到 DB2 UDB 内一个特定的包(package), 包缓存在package cache里面。

数据结构(索引)，统计信息的变化会导致package cache的SQL信息过期，重新解析SQL。



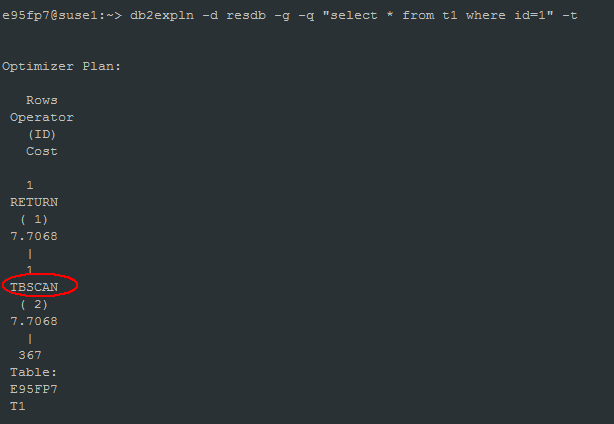




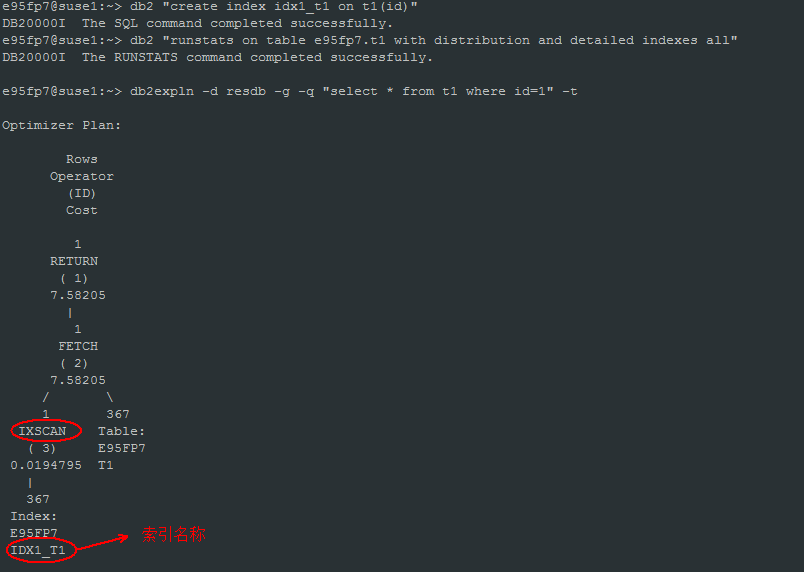


* 1. **全表扫描/索引扫描**

表扫描的方式的执行计划：



索引扫描的执行计划：



* 1. **物理读/逻辑读**

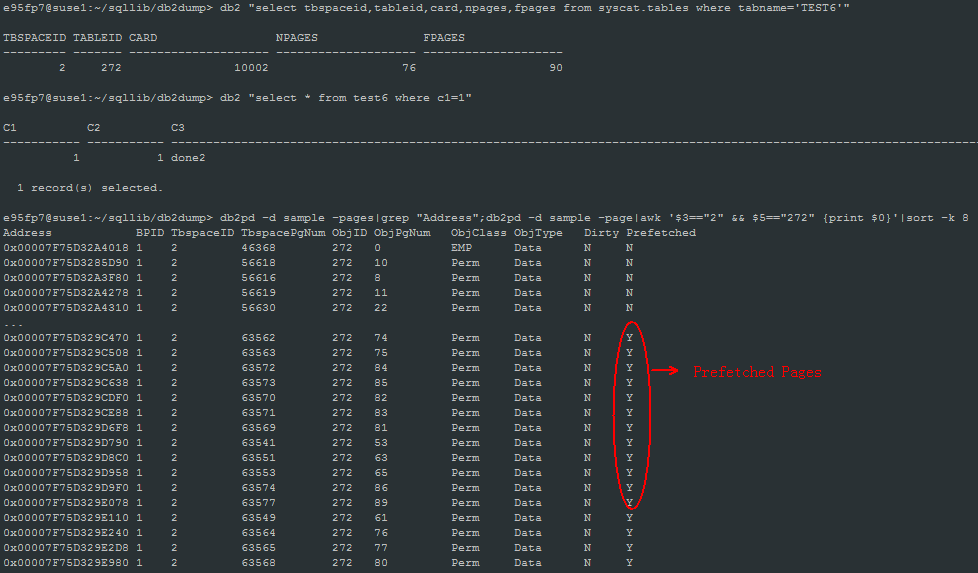
物理读：从磁盘直接读page到缓冲池，单位是页面

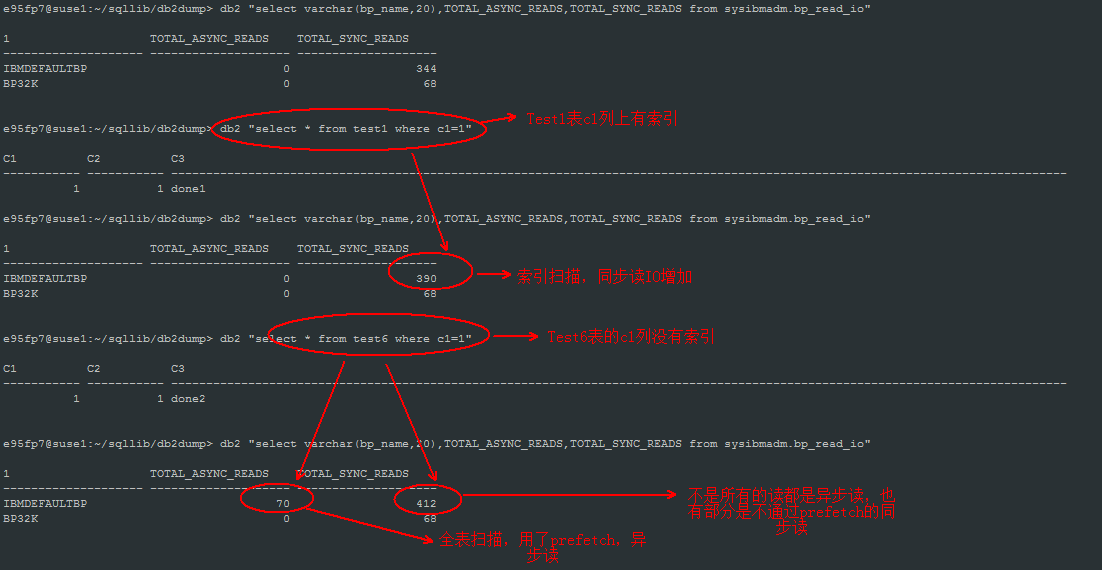
逻辑读：从缓冲池直接读到page数据

数据库先查询缓冲池是否有数据，如果有，直接从缓冲池返回数据，如果没有，才读磁盘。磁盘IO性能比直接从内存读效率低很多。

* 1. **IO预取Prefetch(线程db2pfchr)**

预取: 在查询所需的数据之前读入这些数据，这样查询就不必等待磁盘I/O了。当数据库确定顺序 I/O预取操作可能有助于提高性能时，它就选择预取操作。





I/O预取只有对于大量的I/O才是比较有意义的，少量数据的I/O会直接使用随机I/O方式进行访问。在一个真正的 OLTP 环境中， 建议<20%。但在 DSS 类的工作负载中，预取发挥着重要的作用，一般大于 90%。

另外，I/O预取不仅可以用于数据页的读取，同样也适用于索引页（索引页本质上也是数据页）。

* 1. **日志优先write-ahead logging(线程db2loggw)**

数据库提交的时候，必须保证数据库日志先落地(写回磁盘)，脏数据(修改过的page)会在之后通过触发条件异步或同步写回磁盘。

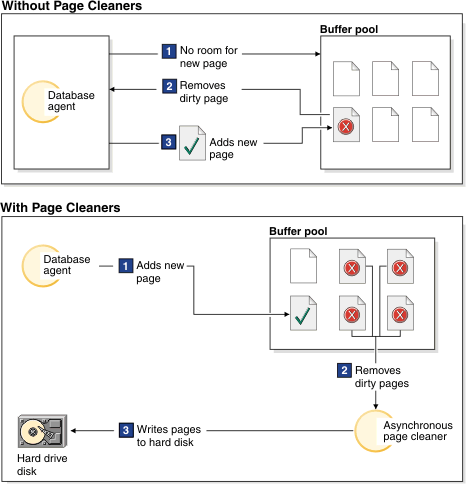
数据库日志先写到日志缓冲池，再写回磁盘。

以下操作必须先等待日志写回磁盘：

1. On COMMIT
2. Until the corresponding data pages are written to disk, because the DB2 server uses write-ahead logging, in which not all of the changed data and index pages need to be written to disk when a transaction completes with a COMMIT statement
3. Until changes (mostly resulting from the execution of data definition language statements) are made to the metadata
4. When the log buffer is full
   1. **脏页写回磁盘(线程db2pclnr)**

In a well-tuned system, it is usually the page-cleaner agents that write changed or dirty pages to disk. Page-cleaner agents perform I/O as background processes and allow applications to run faster because their agents can perform actual transaction work. Page-cleaner agents are sometimes referred to as asynchronous page cleaners or asynchronous buffer writers, because they are not coordinated with the work of other agents and work only when required.

To improve performance for update-intensive workloads, you might want to enable proactive page cleaning, whereby page cleaners behave more proactively in choosing which dirty pages get written out at any given point in time. This is particularly true if snapshots reveal that there are a significant number of synchronous data-page or index-page writes in relation to the number of asynchronous data-page or index-page writes.



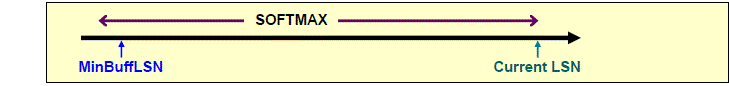
Parameters for asynchronous page cleaning.

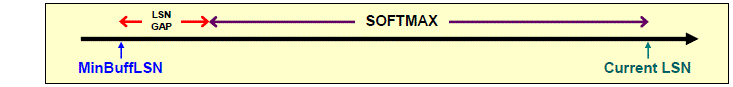
1. CHNGPGS\_THRESH

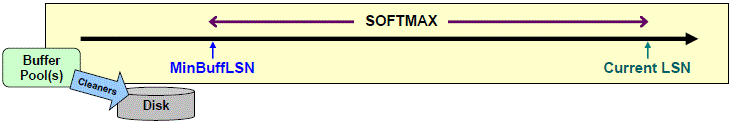
缓冲池脏数据的比率超过了CHNGPGS\_THRESH的值，触发异步写。

1. SOFTMAX

MinBuffLSN, Current LSN的日志量差值超过了SOFTMAX的日志量，触发异步写。

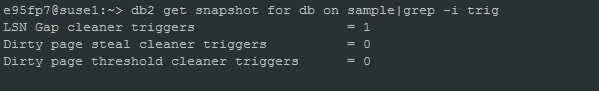
[](http://s3.51cto.com/wyfs02/M01/6B/EF/wKioL1U7EPCTE8AZAAAP1x72yXk874.gif)

[](http://s3.51cto.com/wyfs02/M00/6B/F3/wKiom1U7EB6BvdbSAAARa9oL5W4863.gif)

[](http://s3.51cto.com/wyfs02/M02/6B/EF/wKioL1U7EAXA7o56AAAcJcgGBqM156.gif)

Database recovery after a system crash is faster if more pages have been written to disk.

监控：



LSN Gap cleaner triggers: SOFTMAX触发的次数

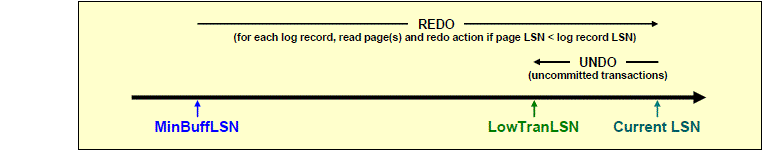
Dirty page threshold cleaner triggers: CHNGPGS\_THRESH触发的次数

Dirty page steal cleaner triggers: 同步清理，由db2agent处理(agent chooses it as a victim)，性能不好。

DB2\_USE\_ALTERNATE\_PAGE\_CLEANING参数： enable proactive page cleaning。 SOFTMAX, CHNGPGS\_THRESH不起作用。

* 1. **Crash Recovery**

**Transactions (or units of work) against a database can be interrupted unexpectedly. If a failure occurs before all of the changes that are part of the unit of work are completed and committed, the database is left in an inconsistent and unusable state. Crash recovery is the process by which the database is moved back to a consistent and usable state. This is done by rolling back incomplete transactions and completing committed transactions that were still in memory when the crash occurred.**

[](http://s3.51cto.com/wyfs02/M00/6B/EF/wKioL1U7Ek7zluG5AAAeJ2Wc_Ps171.gif)

**Min Buff LSN:**

1) Pointer to a log record

2) Tracks oldest change to a data page in the BP that has not been flushed to disk

3) Represents the first log that would need to be redone for crash recovery

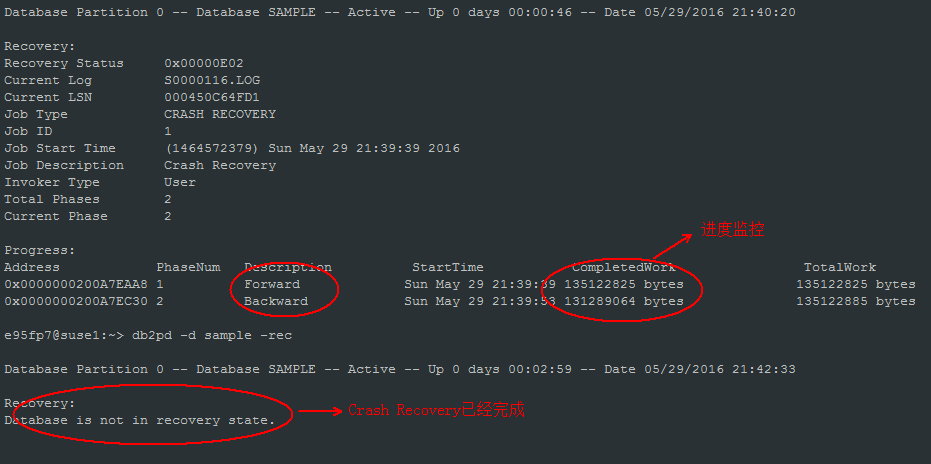
**Low Trans LSN:**

1) Pointer to a log record

2) Oldest uncommitted database change

3) Represents the oldest log record that would need to be undone for crash recovery



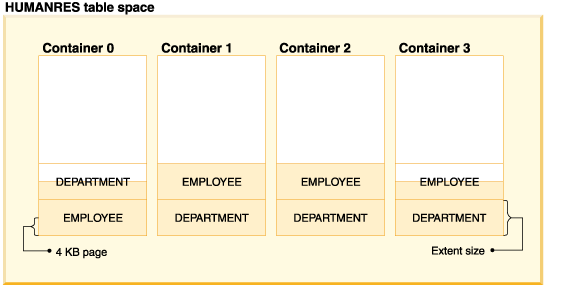


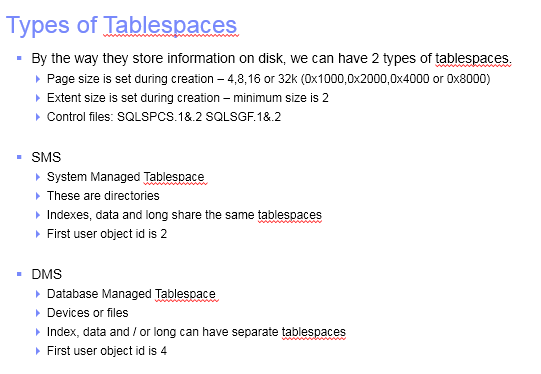
1. **对象结构分析**
   1. **表空间/表/索引**

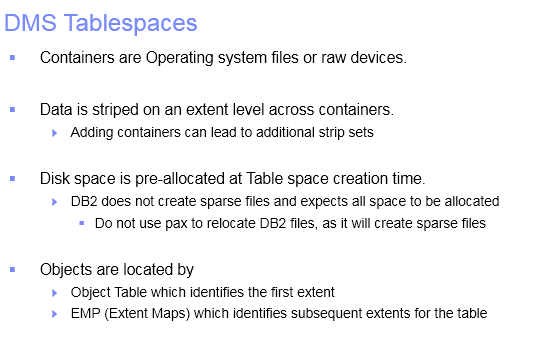
**Table spaces and tables in a database**

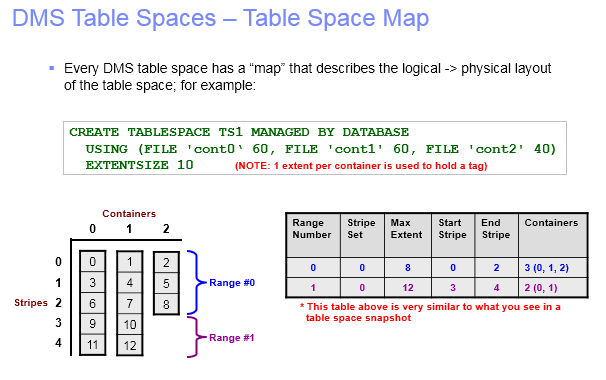


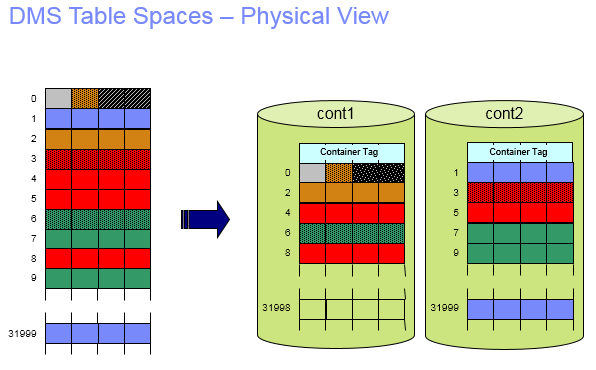
**Containers and extents in a table space**

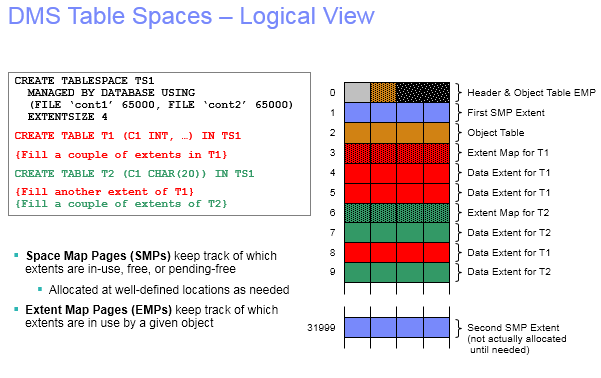






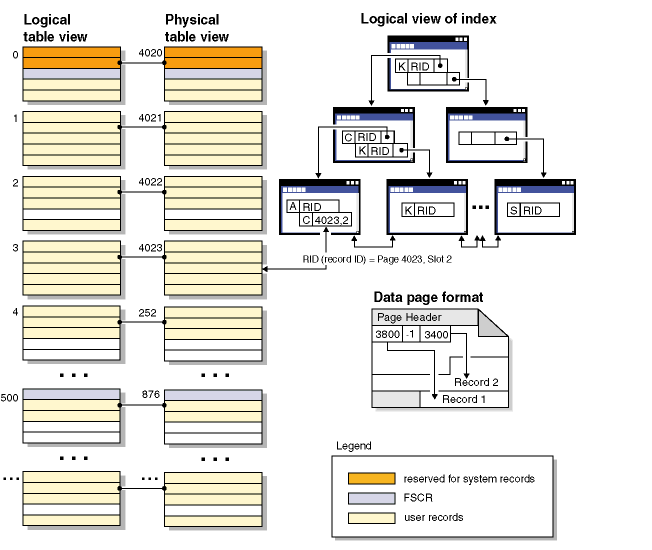






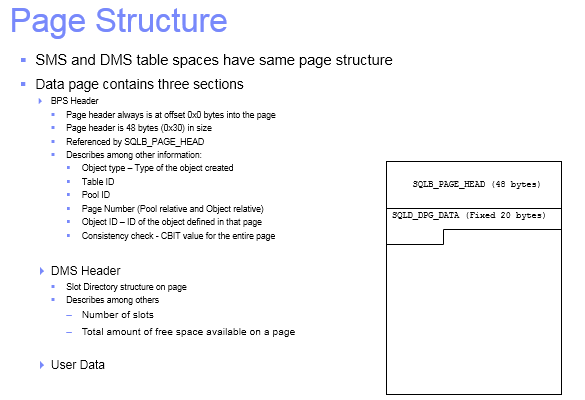


**Logical table, record, and index structure for standard tables**

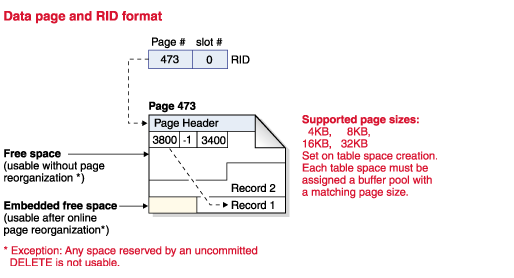


Logically, index pages are organized as a B-tree that can efficiently locate table records that have a specific key value. The number of entities on an index page is not fixed, but depends on the size of the key. For tables in database managed space (DMS) table spaces, record identifiers (RIDs) in the index pages use table space-relative page numbers, not object-relative page numbers. This enables an index scan to directly access the data pages without requiring an extent map page (EMP) for mapping.

Each data page has the same format. A page begins with a page header.

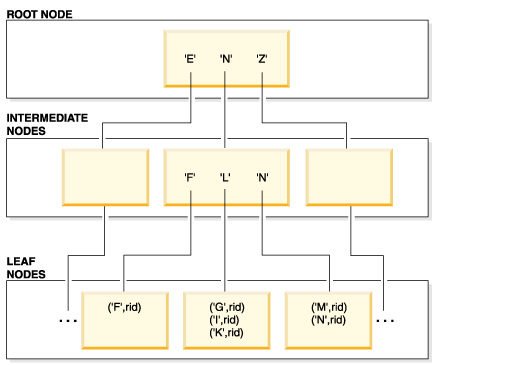


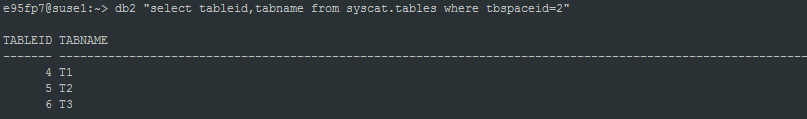
Record identifiers consist of a page number followed by a slot number . Index records contain an additional field called the ridFlag. The ridFlag stores information about the status of keys in the index, such as whether they have been marked deleted. After the index is used to identify a RID, the RID is used to identify the correct data page and slot number on that page. After a record is assigned a RID, the RID does not change until the table is reorganized.

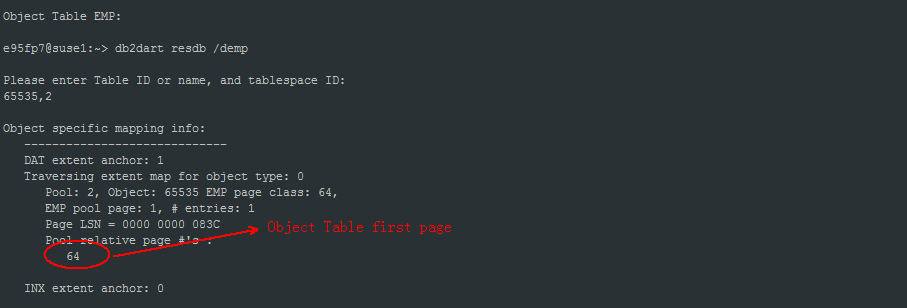


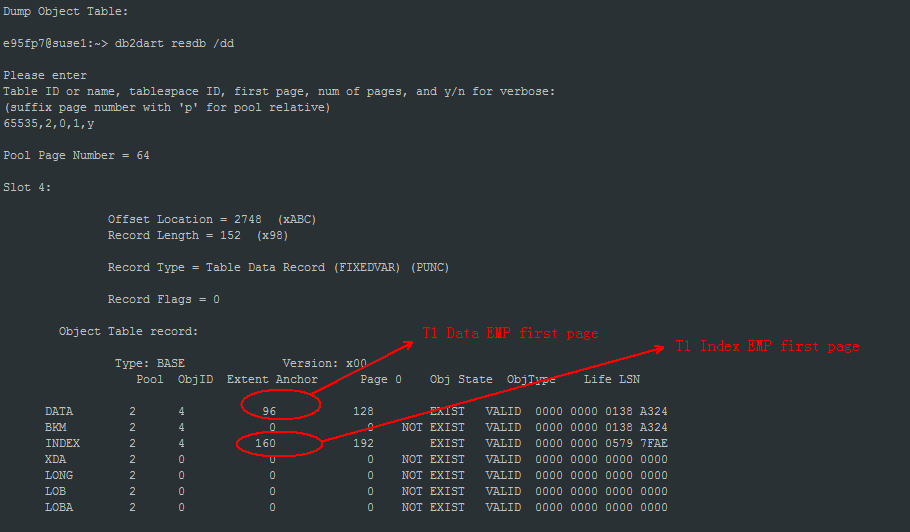
When a table page is reorganized, embedded free space that is left on the page after a record is physically deleted is converted to usable free space.

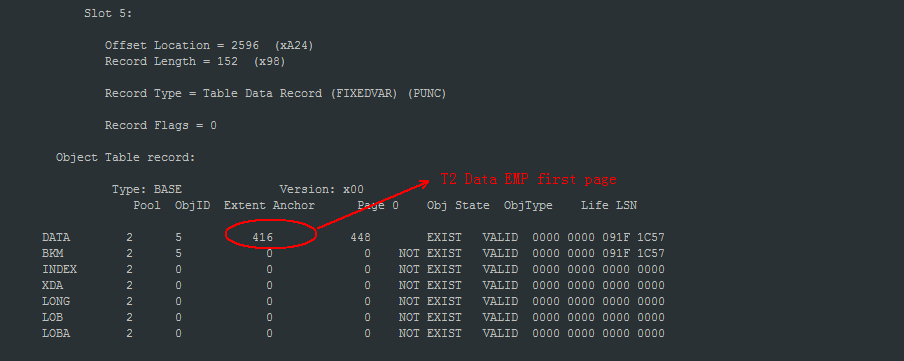
**Structure of a B+ Tree Index**

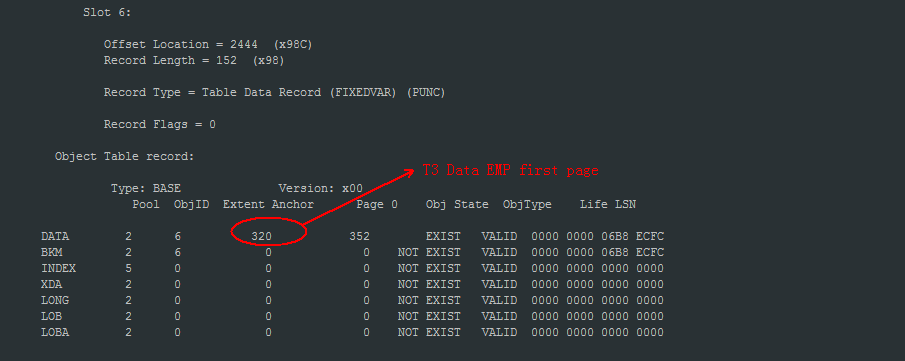


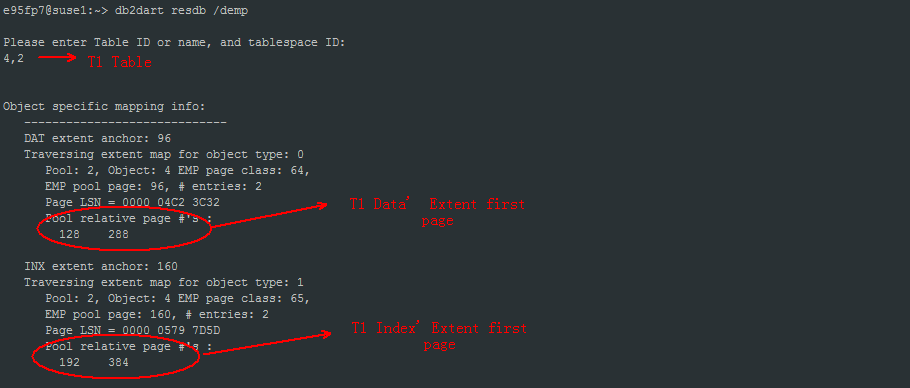


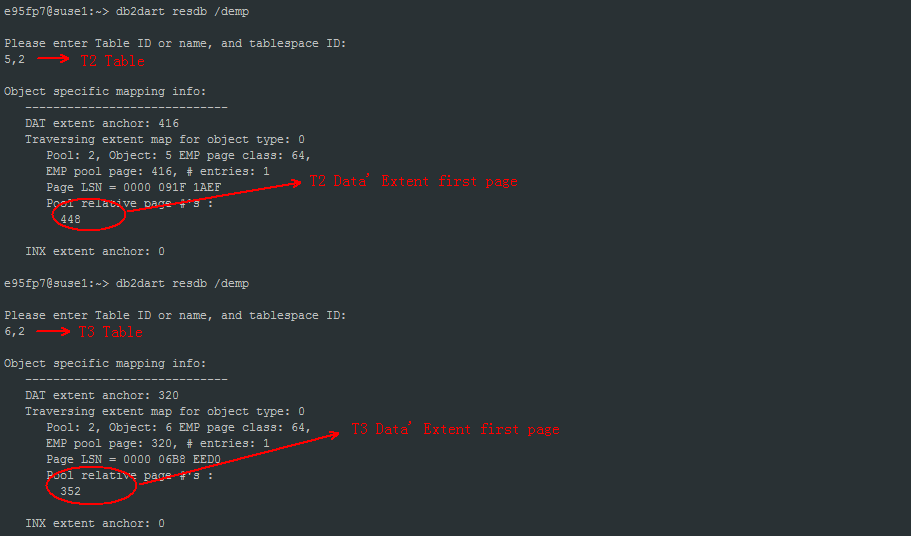
















1. **安全模型**
   1. **认证**
2. SERVER

The user ID and password are sent from the client to the server so that authentication can take place on the server

1. SERVER\_ENCRYPT

The value SERVER\_ENCRYPT provides the same behavior as SERVER, except that any user IDs and passwords sent over the network are encrypted.

1. CLIENT

Specifies that authentication occurs on the database partition where the application is invoked using operating system security. The user ID and password specified during a connection or attachment attempt are compared with the valid user ID and password combinations on the client node to determine if the user ID is permitted access to the instance. No further authentication will take place on the database server.

1. DATA\_ENCRYPT

The server accepts encrypted SERVER authentication schemes and the encryption of user data. The authentication works exactly the same way as that shown with SERVER\_ENCRYPT.

The following user data are encrypted when using this authentication type:

SQL and XQuery statements.

SQL program variable data.

Output data from the server processing of an SQL or XQuery statement and including a description of the data.

Some or all of the answer set data resulting from a query.

Large object (LOB) data streaming.

SQLDA descriptors.

1. KERBEROS

Used when both the DB2 client and server are on operating systems that support the Kerberos security protocol. The Kerberos security protocol performs authentication as a third party authentication service by using conventional cryptography to create a shared secret key. This key becomes a user's credential and is used to verify the identity of users during all occasions when local or network services are requested. The key eliminates the need to pass the user name and password across the network as clear text. Using the Kerberos security protocol enables the use of a single sign-on to a remote DB2 database server.

1. GSSPLUGIN

Specifies that the server uses a GSS-API plug-in to perform authentication.

**Command:**

db2 attach to e95fp7 -> attach to instance

db2 update dbm cfg using AUTHENTICATION server\_encrypt

需要重启实例才生效

* 1. **授权**

1. 实例级别权限

SYSADM、SYSCTRL、SYSMAINT、SYSMON

**SYSADM authority：**

Upgrade a database

Restore a database

Change the database manager configuration file

**SYSCTRL authority:**

Update a database, node, or distributed connection services (DCS) directory

Force users off the system

Create or drop a database

Drop, create, or alter a table space

Use any table space

Restore to a new or an existing database.

**SYSMAINT authority:**

Back up a database or table space

Restore to an existing database

Perform roll forward recovery

Start or stop an instance

Restore a table space

Run a trace, using the db2trc command

Take database system monitor snapshots of a database manager instance or its databases.

Query the state of a table space

Update log history files

Quiesce a table space

Reorganize a table

Collect catalog statistics using the RUNSTATS utility.

**SYSMON authority:**

get database manager monitor switches

get monitor switches

get snapshot

list active databases

list applications

list dcs applications

reset monitor

update monitor switches

1. 数据库级别权限

SECADM

DBADM

BINDADD

CONNECT

CREATETAB

CREATE\_EXTERNAL\_ROUTINE

CREATE\_NOT\_FENCED\_ROUTINE

IMPLICIT\_SCHEMA

QUIESCE\_CONNECT

LOAD

1. Privileges 特权(对象级别权限)

Authorization ID privileges

Schema privileges

Table space privileges

Table and view privileges

Package privileges

Index privileges

Sequence privileges

Routine privileges

Usage privilege on workloads

1. 基于内容的权限(LBAC)
   1. **Restrictive Database权限实验**

If the RESTRICTIVE option is present it causes the RESTRICT\_ACCESS database configuration parameter to be set to YES and no privileges or authorities are automatically granted to PUBLIC.

If the RESTRICTIVE option is not present then the RESTRICT\_ACCESS database configuration parameter is set to NO and all of the following privileges are automatically granted to PUBLIC.

CREATETAB

BINDADD

CONNECT

IMPLSCHEMA

EXECUTE with GRANT on all procedures in schema SQLJ

EXECUTE with GRANT on all functions and procedures in schema SYSPROC

BIND on all packages created in the NULLID schema

EXECUTE on all packages created in the NULLID schema

CREATEIN on schema SQLJ

CREATEIN on schema NULLID

USE on table space USERSPACE1

SELECT access to the SYSIBM catalog tables

SELECT access to the SYSCAT catalog views

SELECT access to the SYSSTAT catalog views

UPDATE access to the SYSSTAT catalog views

创建restrictive的database：

e95fp7@suse1:~> db2 create database resdb restrictive

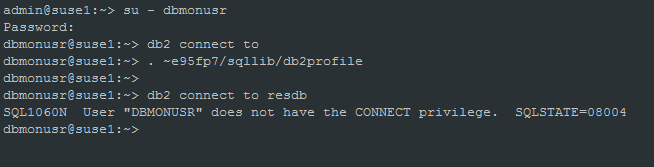
DB20000I The CREATE DATABASE command completed successfully.

e95fp7@suse1:~> db2 get db cfg|grep -i restrict

Restrict access = YES

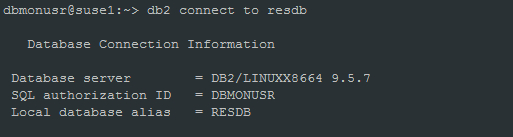
dbmonusr用户：

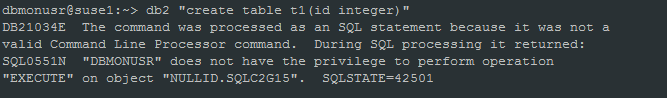
登录没有权限



e95fp7@suse1:~> db2 "grant connect on database to dbmonusr"

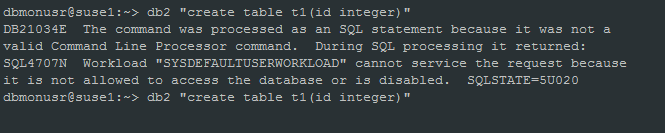
DB20000I The SQL command completed successfully.





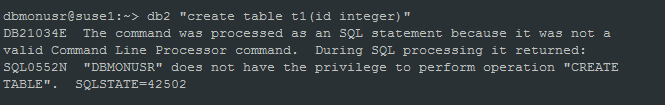
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLC2G15 to dbmonusr"

DB20000I The SQL command completed successfully.



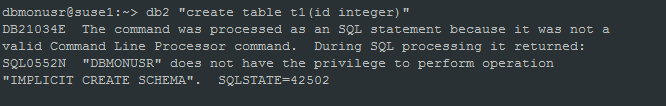
e95fp7@suse1:~> db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to dbmonusr"

DB20000I The SQL command completed successfully.



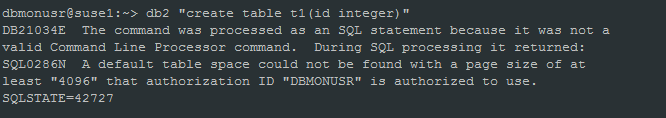
e95fp7@suse1:~> db2 "grant createtab on database to dbmonusr"

DB20000I The SQL command completed successfully.



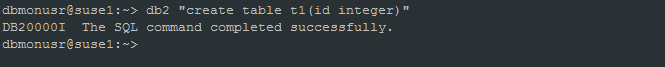
e95fp7@suse1:~> db2 "grant IMPLICIT\_SCHEMA on database to dbmonusr"

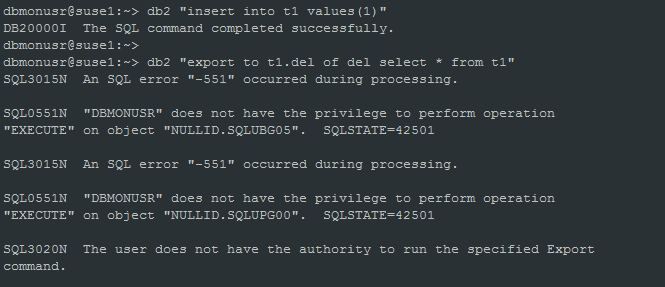
DB20000I The SQL command completed successfully.



e95fp7@suse1:~> db2 "grant use of tablespace userspace1 to dbmonusr"

DB20000I The SQL command completed successfully.



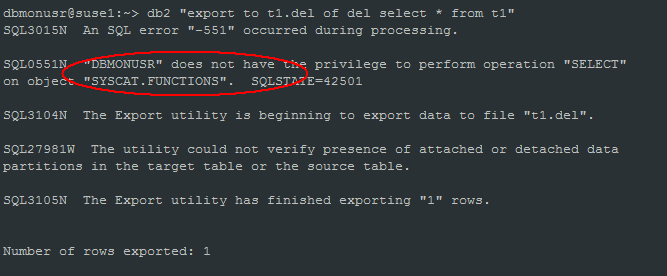


e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUBG05 to dbmonusr"

DB20000I The SQL command completed successfully.

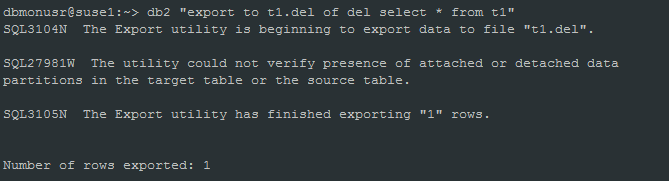
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUPG00 to dbmonusr"

DB20000I The SQL command completed successfully.



e95fp7@suse1:~> db2 "grant select on table SYSCAT.FUNCTIONS to dbmonusr"

DB20000I The SQL command completed successfully.





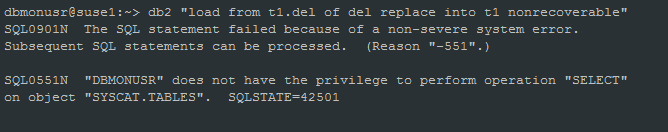
db2diag –A

more db2diag.log



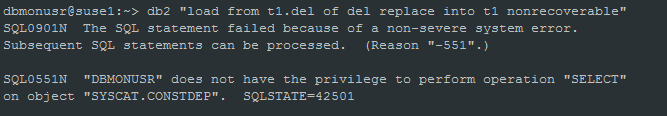
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUJG0A to dbmonusr"

DB20000I The SQL command completed successfully.



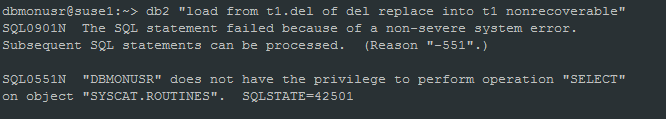
e95fp7@suse1:~> db2 "grant select on table syscat.tables to dbmonusr"

DB20000I The SQL command completed successfully.



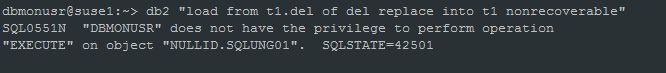
e95fp7@suse1:~> db2 "grant select on table syscat.constdep to dbmonusr"

DB20000I The SQL command completed successfully.



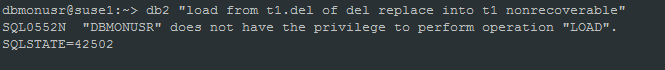
e95fp7@suse1:~> db2 "grant select on table syscat.routines to dbmonusr"

DB20000I The SQL command completed successfully.



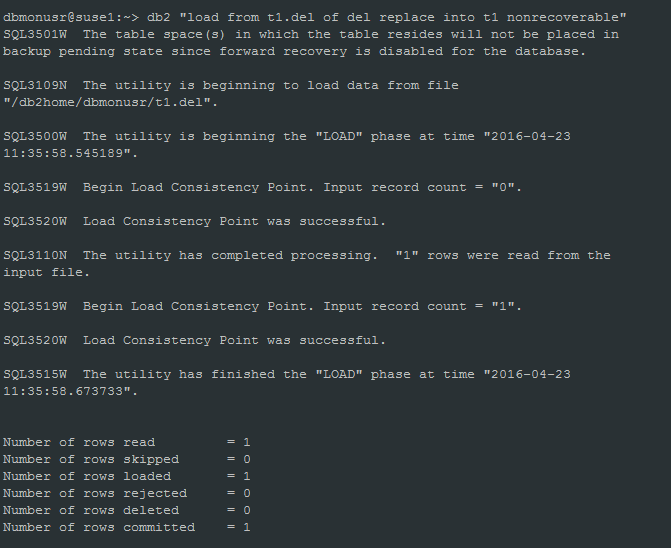
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUNG01 to dbmonusr"

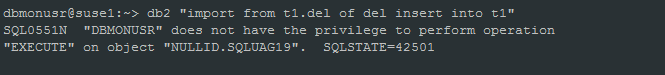
DB20000I The SQL command completed successfully.



e95fp7@suse1:~> db2 "grant load on database to dbmonusr"

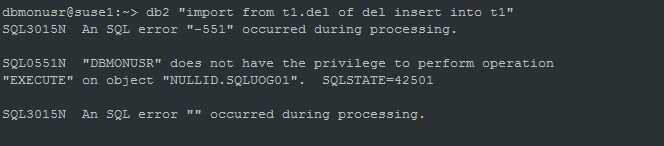
DB20000I The SQL command completed successfully.





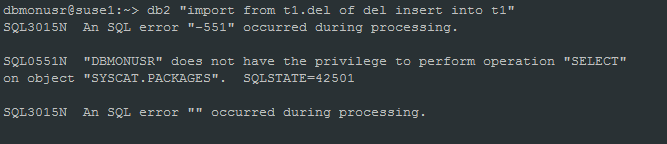
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUAG19 to dbmonusr"

DB20000I The SQL command completed successfully.



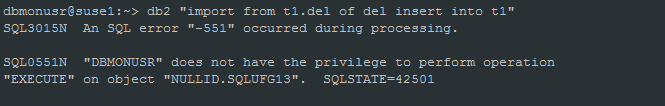
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUOG01 to dbmonusr"

DB20000I The SQL command completed successfully.



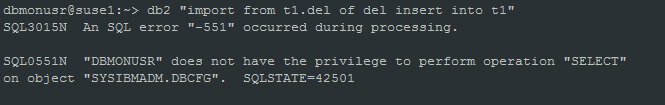
e95fp7@suse1:~> db2 "grant select on table syscat.packages to dbmonusr"

DB20000I The SQL command completed successfully.



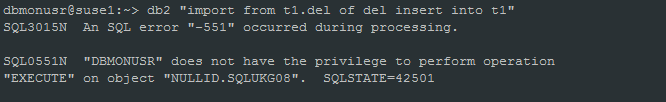
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUFG13 to dbmonusr"

DB20000I The SQL command completed successfully.



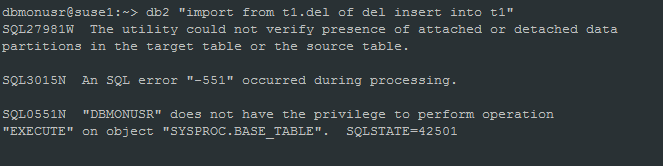
e95fp7@suse1:~> db2 "grant select on table sysibmadm.dbcfg to dbmonusr"

DB20000I The SQL command completed successfully.



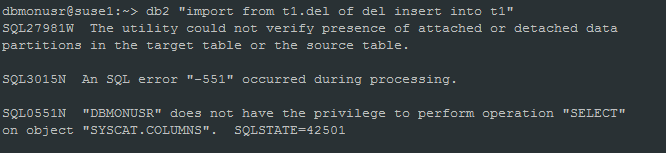
e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLUKG08 to dbmonusr"

DB20000I The SQL command completed successfully.



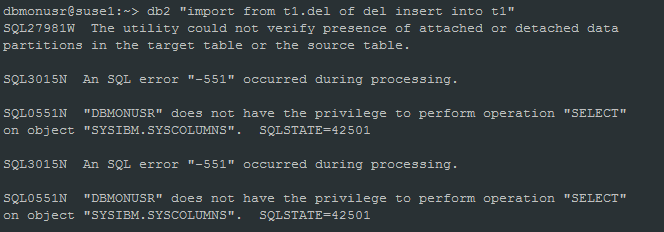
e95fp7@suse1:~> db2 "grant execute on function SYSPROC.BASE\_TABLE to dbmonusr"

DB20000I The SQL command completed successfully.



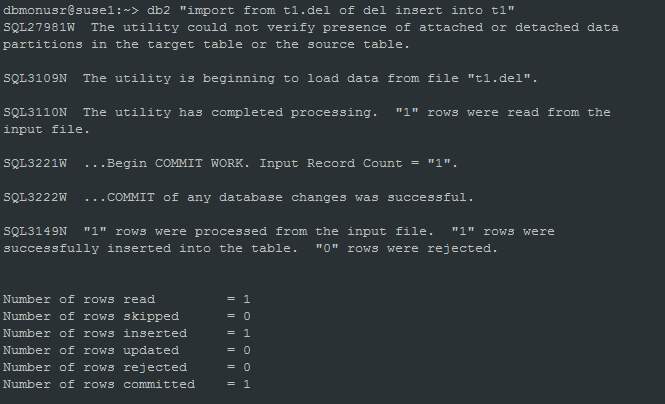
e95fp7@suse1:~> db2 "grant select on table syscat.columns to dbmonusr"

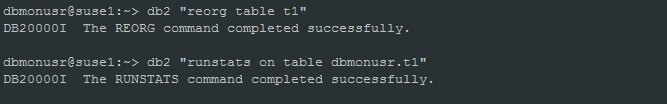
DB20000I The SQL command completed successfully.

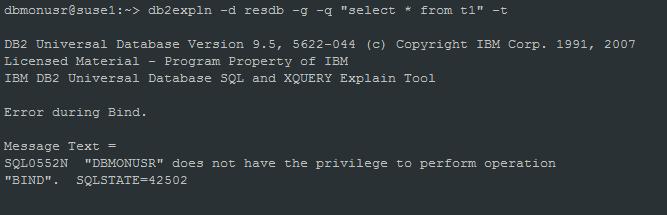


e95fp7@suse1:~> db2 "grant select on table SYSIBM.SYSCOLUMNS to dbmonusr"

DB20000I The SQL command completed successfully.

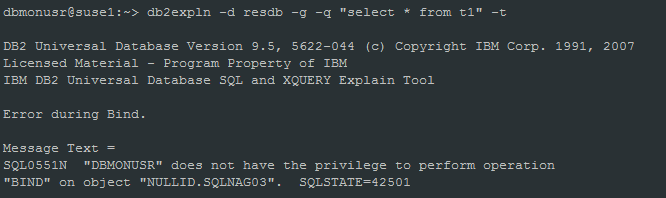






e95fp7@suse1:~> db2 "grant bindadd on database to dbmonusr"

DB20000I The SQL command completed successfully.



e95fp7@suse1:~> db2 "grant execute on package NULLID.SQLNAG03 to dbmonusr"

DB21034E The command was processed as an SQL statement because it was not a

valid Command Line Processor command. During SQL processing it returned:

SQL0204N "NULLID.SQLNAG03" is an undefined name. SQLSTATE=42704

e95fp7@suse1:~> db2 "select pkgname from syscat.packages where pkgname='SQLNAG03'"

PKGNAME

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

0 record(s) selected.

没有找到NULLID.SQLNAG03包.

NULLID.SQLNAG03属于db2expln.bnd, 需要绑包。

e95fp7@suse1:~/sqllib/bnd> db2 "bind db2expln.bnd blocking all sqlerror continue "

LINE MESSAGES FOR db2expln.bnd

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

SQL0061W The binder is in progress.

SQL0091N Binding was ended with "0" errors and "0" warnings.

e95fp7@suse1:~/sqllib/bnd> db2 "select pkgname from syscat.packages where pkgname='SQLNAG03'"

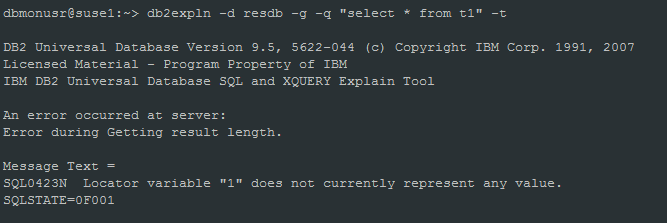
PKGNAME

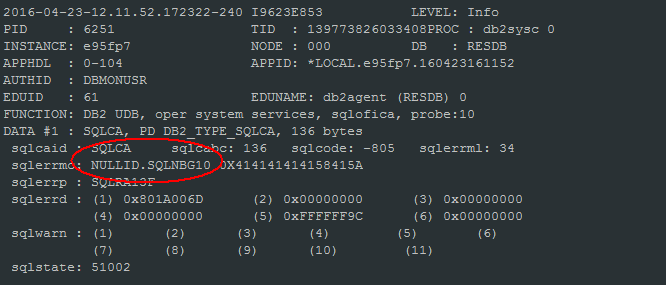
--------------------------------------------------------------------------------------------------------------------------------

SQLNAG03

e95fp7@suse1:~/sqllib/bnd> db2 "grant execute on package NULLID.SQLNAG03 to dbmonusr"

DB20000I The SQL command completed successfully.





e95fp7@suse1:~/sqllib/bnd> db2 "grant execute on package NULLID.SQLNBG10 to dbmonusr"

DB21034E The command was processed as an SQL statement because it was not a

valid Command Line Processor command. During SQL processing it returned:

SQL0204N "NULLID.SQLNBG10" is an undefined name. SQLSTATE=42704

e95fp7@suse1:~/sqllib/bnd>

e95fp7@suse1:~/sqllib/bnd> db2 "grant execute on package NULLID.SQLNBG10 to dbmonusr"

e95fp7@suse1:~/sqllib/bnd> db2 "select pkgname from syscat.packages where pkgname='SQLNBG10'"

PKGNAME

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

0 record(s) selected.

NULLID.SQLNBG10属于db2exsrv.bnd，重绑包。

e95fp7@suse1:~/sqllib/bnd> db2 "bind db2exsrv.bnd blocking all sqlerror continue"

LINE MESSAGES FOR db2exsrv.bnd

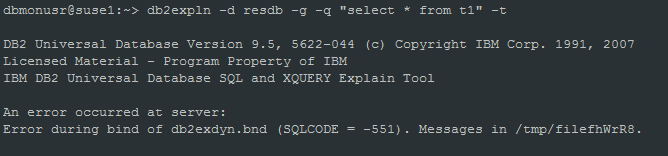
------ --------------------------------------------------------------------

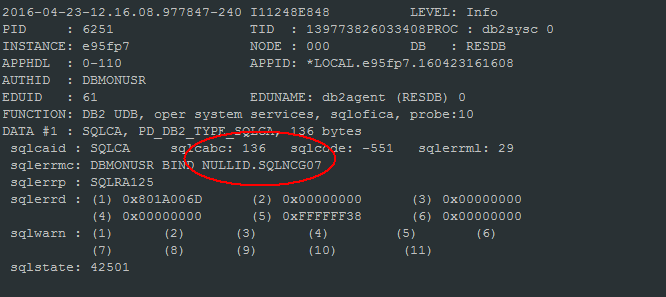
SQL0061W The binder is in progress.

SQL0091N Binding was ended with "0" errors and "0" warnings.

e95fp7@suse1:~/sqllib/bnd> db2 "grant execute on package NULLID.SQLNBG10 to dbmonusr"

DB20000I The SQL command completed successfully.





e95fp7@suse1:~/sqllib/bnd> db2 "select pkgname from syscat.packages where pkgname='SQLNCG07'"

PKGNAME

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

0 record(s) selected.

e95fp7@suse1:~/sqllib/bnd> db2 "bind db2exdyn.bnd blocking all sqlerror continue"

LINE MESSAGES FOR db2exdyn.bnd

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

SQL0061W The binder is in progress.

SQL0091N Binding was ended with "0" errors and "0" warnings.

e95fp7@suse1:~/sqllib/bnd> db2 "grant execute on package NULLID.SQLNCG07 to dbmonusr"

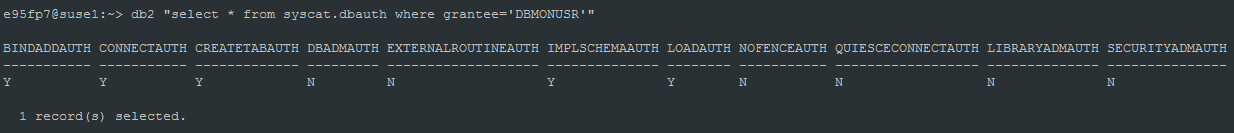
DB20000I The SQL command completed successfully.



e95fp7@suse1:~/sqllib/bnd> db2 "grant select on table sysibm.systables to dbmonusr"

DB20000I The SQL command completed successfully.





1. **隔离级别/锁**
   1. **隔离级别**
2. Uncommitted read (UR)

The uncommitted read isolation level allows an application to access the uncommitted changes of other transactions.

支持脏读、不可重复读、幻象读

1. Cursor stability (CS)

The cursor stability isolation level locks any row being accessed during a transaction while the cursor is positioned on that row. This lock remains in effect until the next row is fetched or the transaction terminates.

支持不可重复读、幻象读

1. Read stability (RS)

The read stability isolation level locks only those rows that an application retrieves during a unit of work.

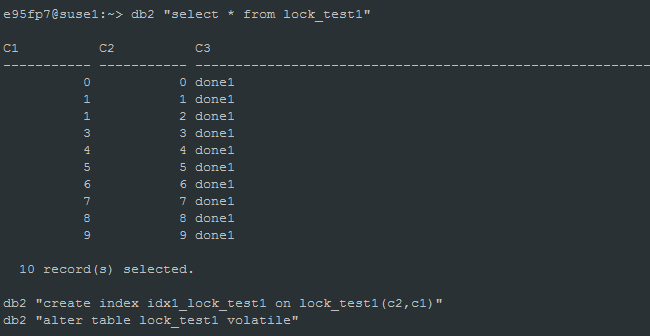
支持幻象读

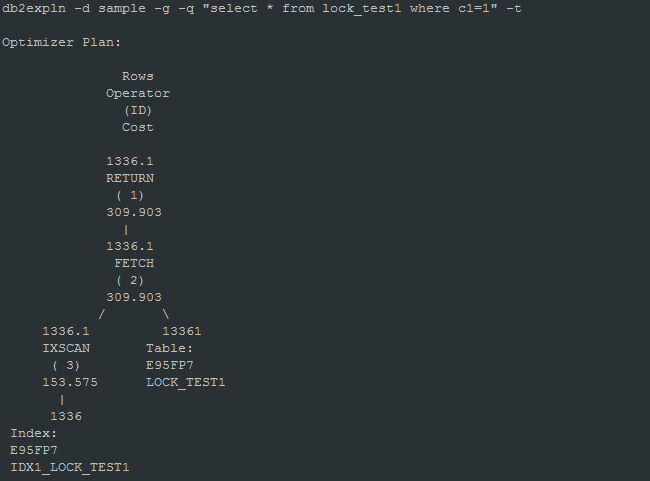
1. Repeatable read (RR)

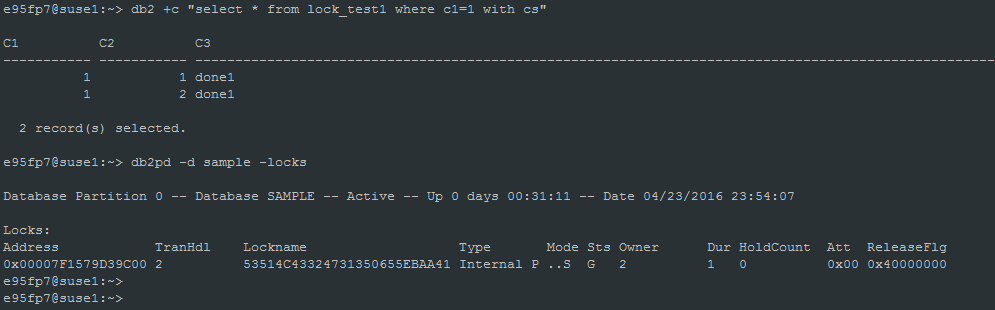
The repeatable read isolation level locks all the rows that an application references during a unit of work (UOW).

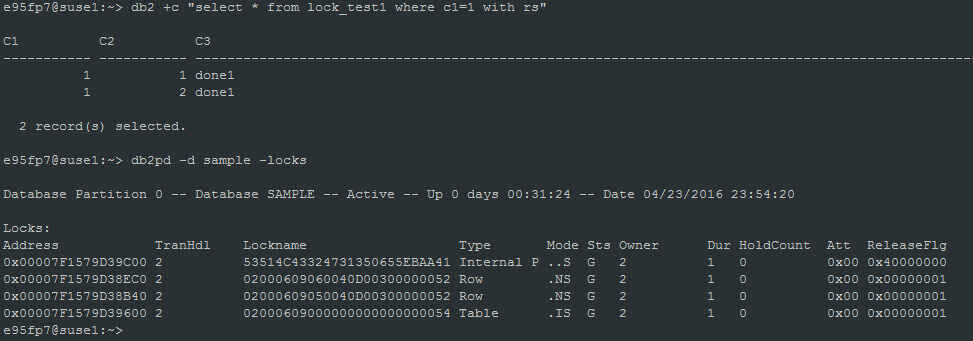
Every referenced row is locked, not just the rows that are retrieved.

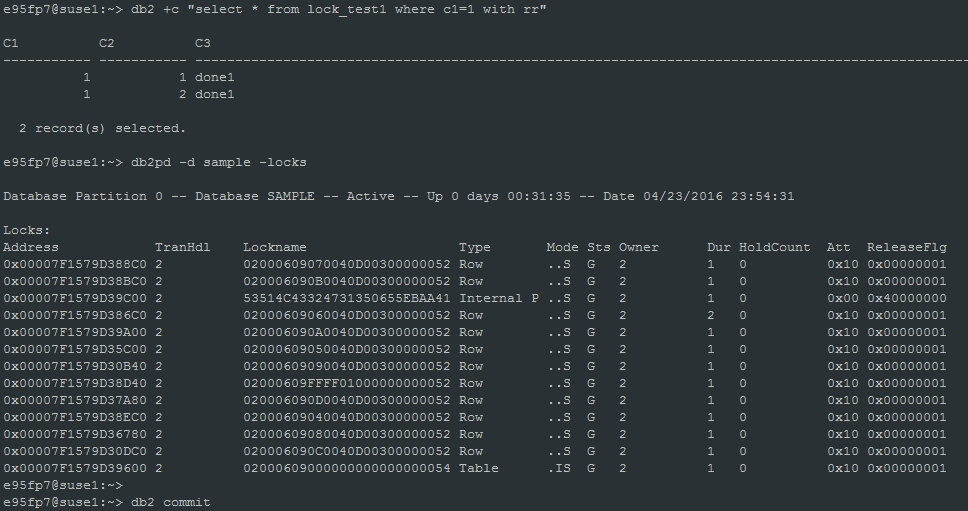
For example, if you scan 10 000 rows and apply predicates to them, locks are held on all 10 000 rows, even if, say, only 10 rows qualify.





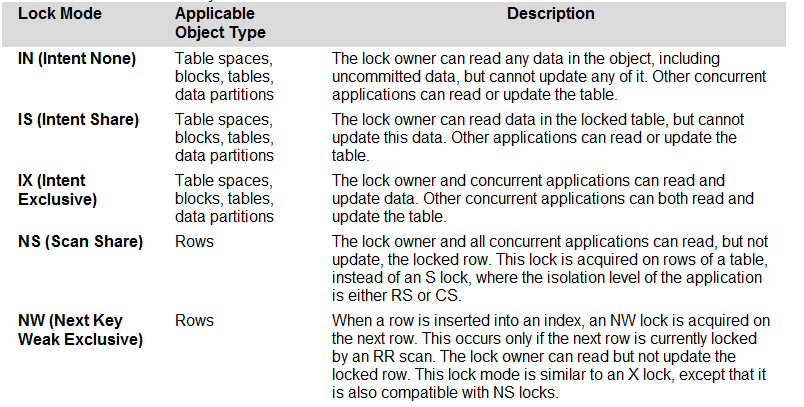


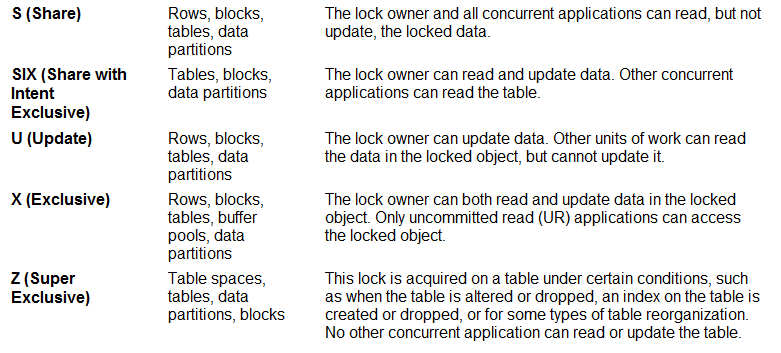




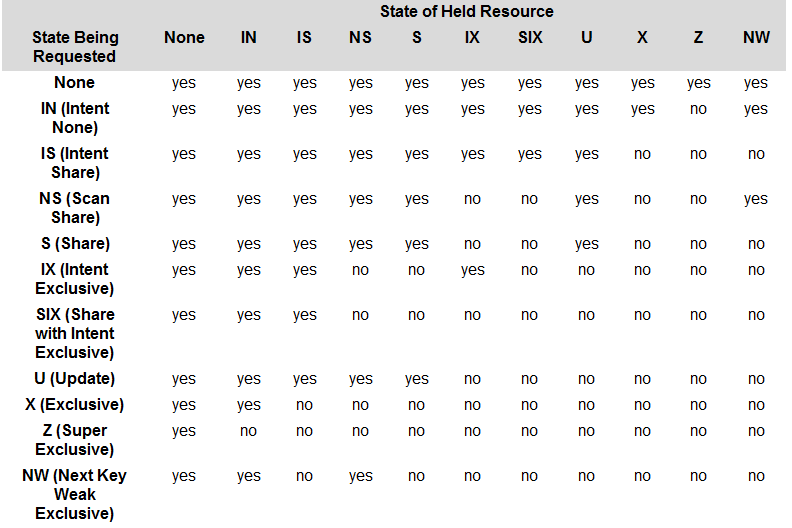
* 1. **锁**

1. Lock Mode



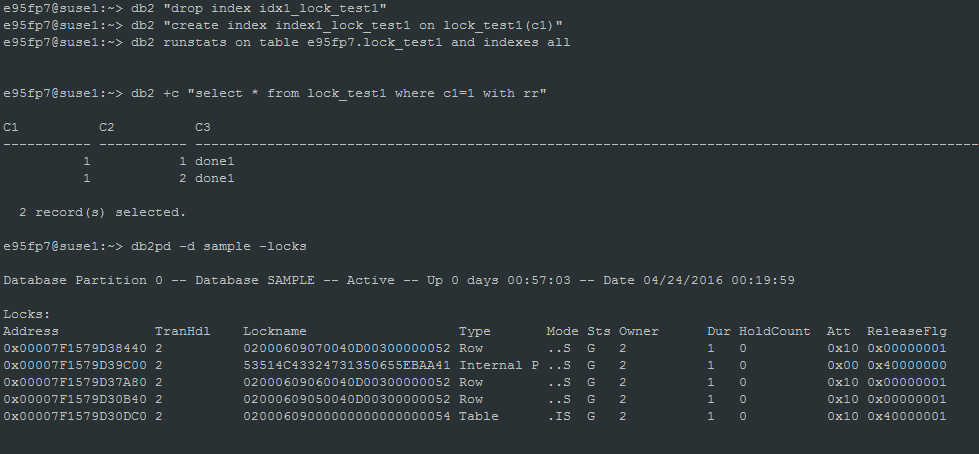


1. Lock Type Compatibility



* 1. **避免幻象读的加锁方式(RR隔离级别)**

SESSION 1：查询只有两条记录，但是行锁(S锁)却有三个，其中一个是NEXT ROW的S锁。



SESSION 2: 新增c1=1的记录。幻象读是允许新增的，但是因为SESSION 1使用的是RR查询，所以SESSION 2有锁等待。

e95fp7@suse1:~> db2 "insert into lock\_test1 values(1,1,'done1')"

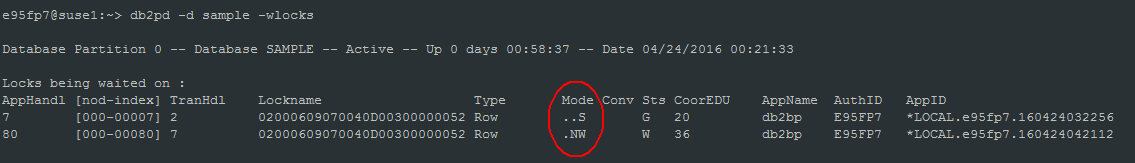
DB21034E The command was processed as an SQL statement because it was not a

valid Command Line Processor command. During SQL processing it returned:

SQL0911N The current transaction has been rolled back because of a deadlock

or timeout. Reason code "68". SQLSTATE=40001

是什么样的锁等待呢？

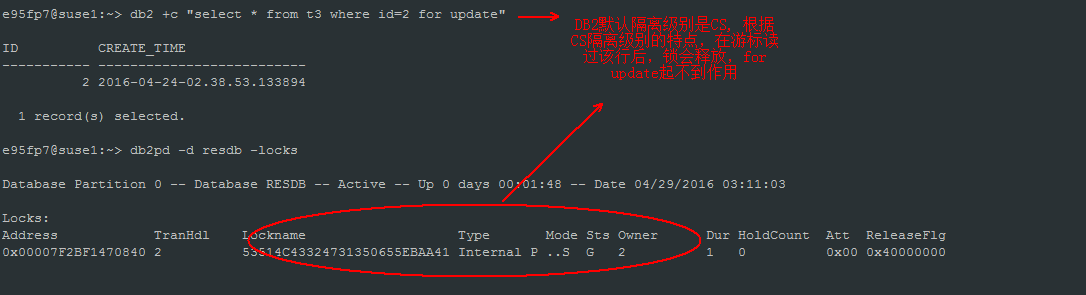


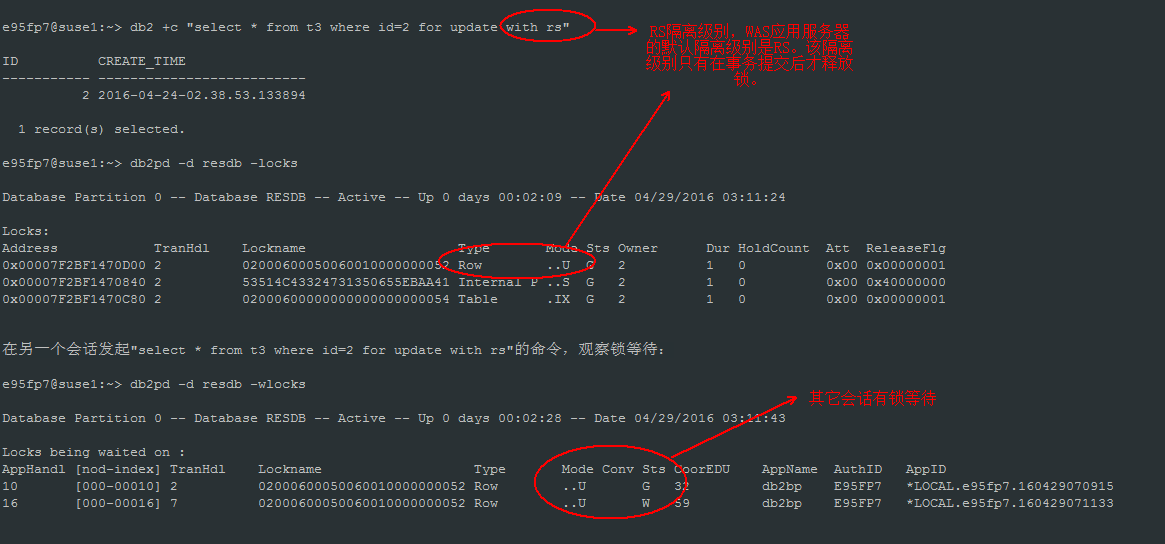
新增记录需要对c1=1的next row加NW锁(有索引的情况下), 但是因为RR隔离级别已经对c1=1的next row加了S锁，S锁与NW锁不兼容，导致锁等待。从而避免了SESSION 1的幻象读。

如果没有索引，那么SESSION1 的RR查询，对表加的是S锁，更加不可能有幻象读了。

* 1. **Select .. for update**

在oracle里，select .. for update会对行加锁，别的事务对该行select .. for update会等锁。但是DB2里，不同的隔离级别处理方式不同，for update可能不起作用。





**从oracle转来的开发人员要弄清楚这点，特别是关键的账务处理。WAS默认的隔离级别是RS，如果需要降低隔离级别，需要慎重考虑！**

* 1. **db2是否存在一致性读?**

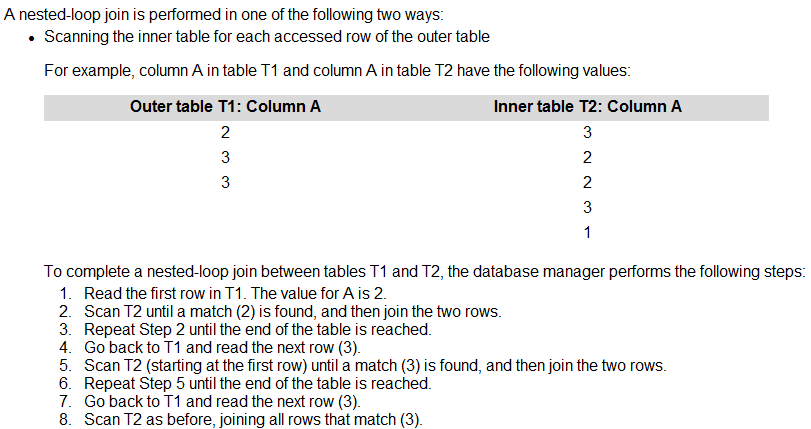
DB2是否存在ORACLE的一致性读呢？

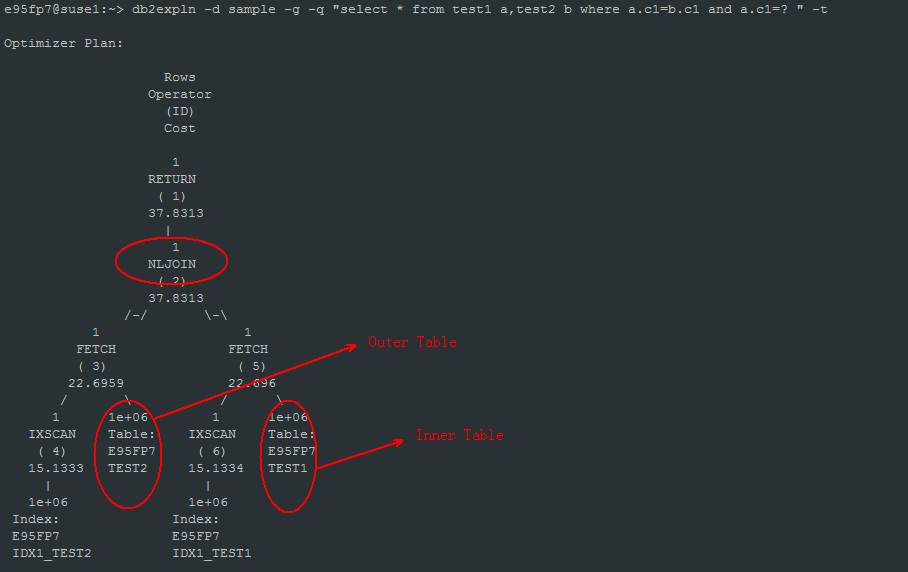
ORACLE通过回滚空间实现多版本的一致性读，查询的数据块都是在事务开始时间点SCN的数据。SCN: system change number, 类似db2的LSN(log sequence number)



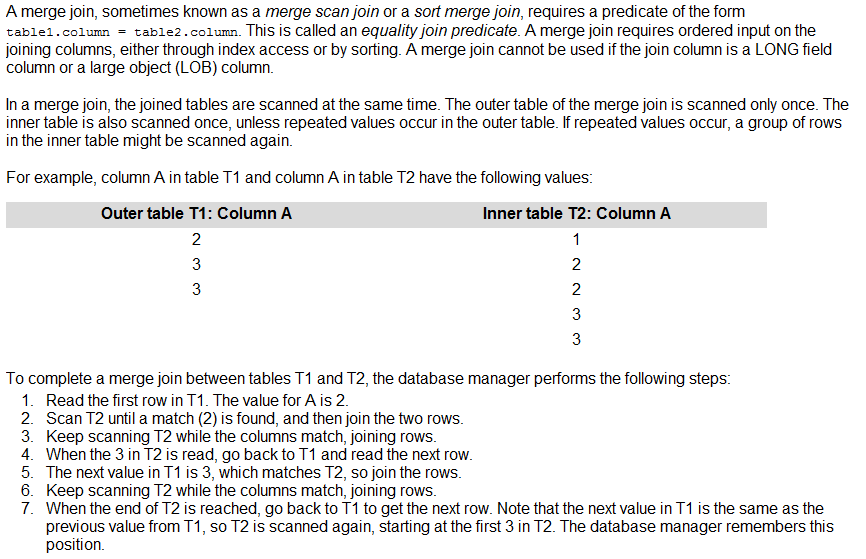


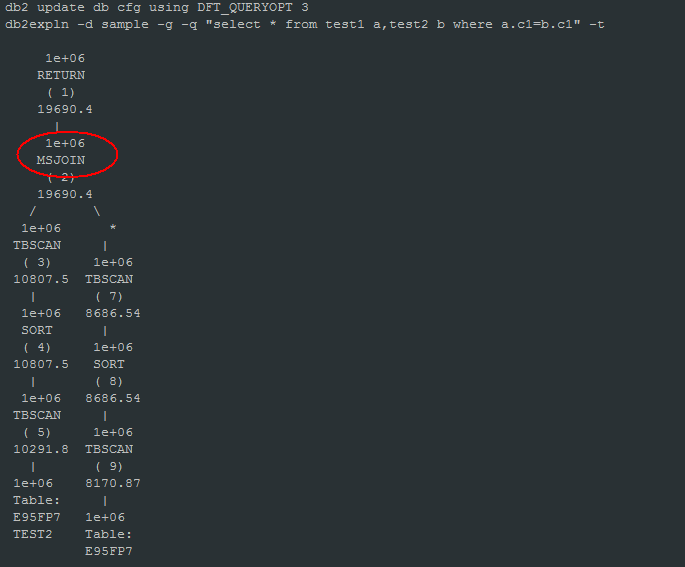
1. **SQL优化**
   1. **表连接方法**
2. **Nested-loop join**



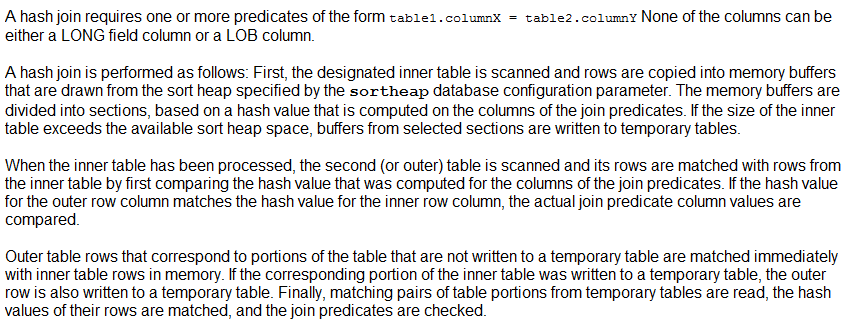


1. **Merge join**

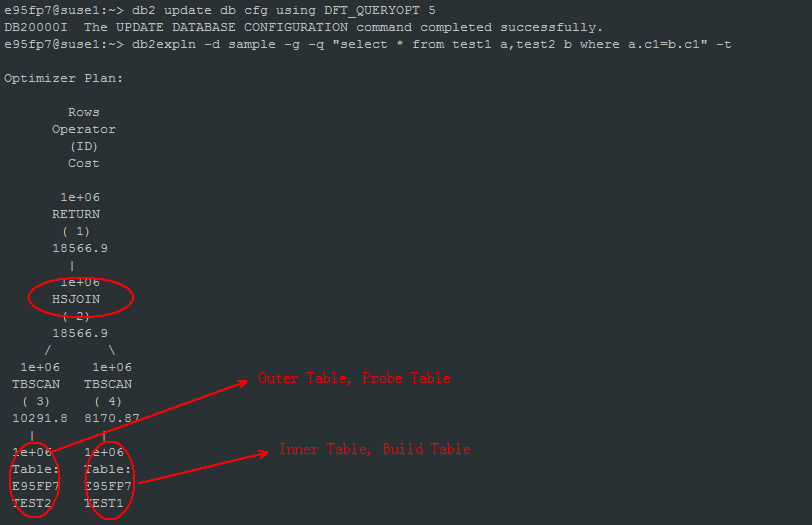




1. **Hash join**



db2expln -d sample -g -q "select \* from test1 a,test2 b where a.c1=b.c1" –t



* 1. **统计信息**
     1. **Catalog statistics**

When you execute the RUNSTATS utility for a table, statistical view, or for a table and its associated indexes, the following kinds of

statistical information are stored in the system catalog tables:

For a table and index:

1. The number of pages in use
2. The number of pages that contain rows
3. The number of rows that overflow
4. The number of rows in the table (cardinality)
5. For MDC tables, the number of blocks that contain data
6. For partitioned tables, the degree of data clustering within a single data partition

For each column in the table or statistical view and the first column in the index key:

1. The cardinality of the column
2. The average length of the column
3. The second highest value in the columns
4. The second lowest value in the column
5. The number of NULLs in the column

For indexes only:

1. The number of index entries (index cardinality)
2. The number of leaf pages
3. The number of index levels
4. The degree of clustering of the table data to this index
5. The degree of clustering of the index keys with regard to data partitions
6. The ratio of leaf pages on disk in index key order to the number of pages in the range of pages occupied by the index
7. The number of distinct values in the first column of the index
8. The number of distinct values in the first two, three, and four columns of the index
9. The number of distinct values in all columns of the index
10. The number of leaf pages located on disk in index key order, with few or no large gaps between them
11. The number of pages on which all RIDs are marked deleted
12. The number of RIDs marked deleted on pages on which not all RIDs are marked deleted
    * 1. **Distribution statistics**

The optimizer uses data distribution statistics to estimate efficient access plans for tables and statistical views in which data

is not evenly distributed and columns have a significant number of duplicate values.

Command:

db2 runstats on table e95fp7.test1 with distribution

* + 1. **Detailed index statistics**

The optimizer uses detailed index statistics to determine how efficient it is to access a table through an index.

If you execute RUNSTATS for indexes with the DETAILED clause, you collect statistical information about indexes that allows the optimizer to estimate how many data page fetches will be required, based on various buffer-pool sizes. This additional information helps the optimizer make better estimates of the cost of accessing a table through an index.

When you collect detailed index statistics, RUNSTATS takes longer and requires more memory and CPU processing.

The DETAILED statistics PAGE\_FETCH\_PAIRS and CLUSTERFACTOR will be collected only if the table is of a sufficient size: around 25 pages. In this case, CLUSTERFACTOR will be a value between 0 and 1; and CLUSTERRATIO will be -1 (not collected). For tables smaller than 25 pages, CLUSTERFACTOR will be -1 (not collected), and CLUSTERRATIO will be a value between 0 and 100; even if the DETAILED clause is specified for an index on that table.

The statistical information also provides finer estimates of the degree of clustering of the table rows to the index order. The less the table rows are clustered in relation to the index, the more I/Os are required to access table rows through the index. The optimizer considers both the buffer size and the degree of clustering when it estimates the cost of accessing a table through an index.

You should collect DETAILED index statistics when queries reference columns that are not included in the index. In addition, DETAILED index statistics should be used in the following circumstances:

1. The table has multiple unclustered indexes with varying degrees of clustering
2. The degree of clustering is non-uniform among the key values
3. The values in the index are updated non-uniformly

Command:

db2 runstats on table e95fp7.test1 and detailed indexes all

* + 1. **Declared temporary table**

You can perform a RUNSTATS on a declared temporary table, but the resulting statistics are not stored in the system catalogs

because declared temporary tables do not have catalog entries. However, the statistics are stored in memory structures that

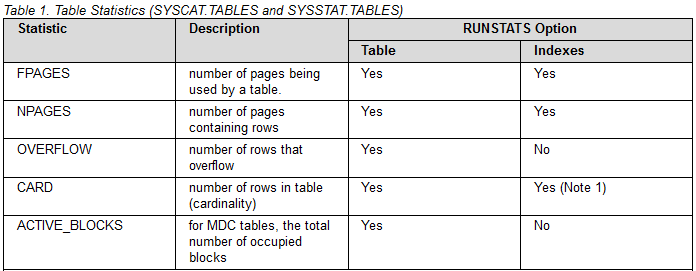
represent the catalog information for declared temporary tables. In some cases, therefore, it might be useful to perform a RUNSTATS on these tables.

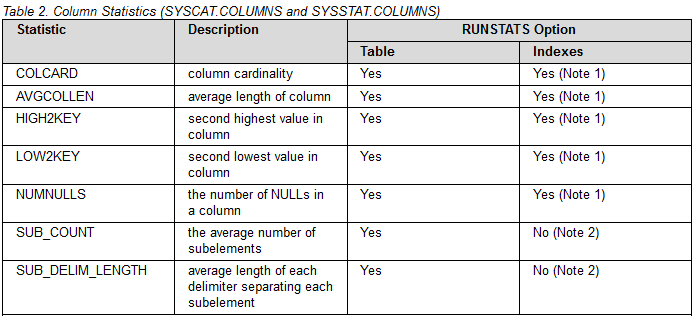
* + 1. **Cached dynamic statements**

Statistics collection invalidates cached dynamic statements that reference tables for which statistics have been collected.

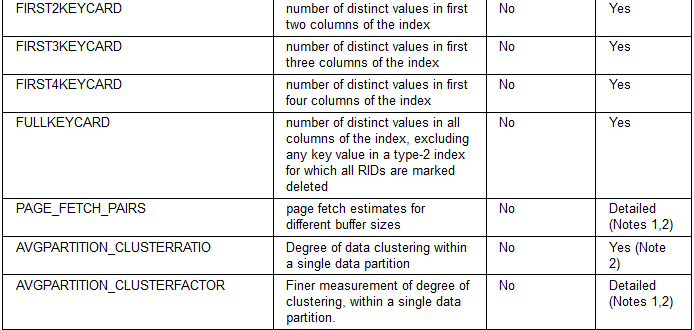
This is done so that cached dynamic statements can be re-optimized with the latest statistics.

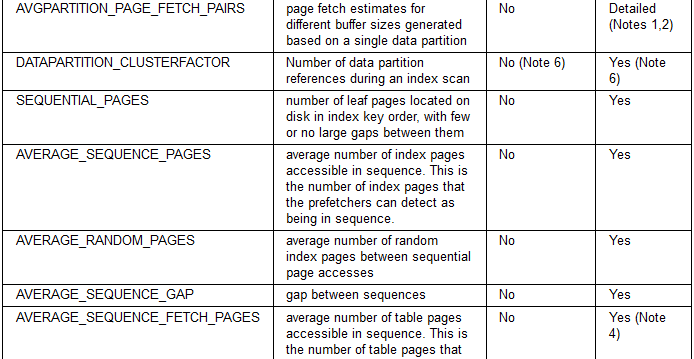
* + 1. **Catalog statistics tables**

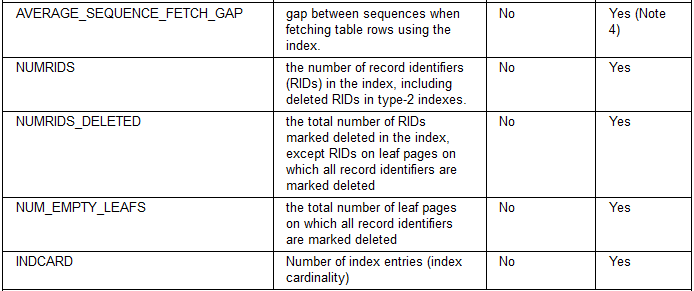


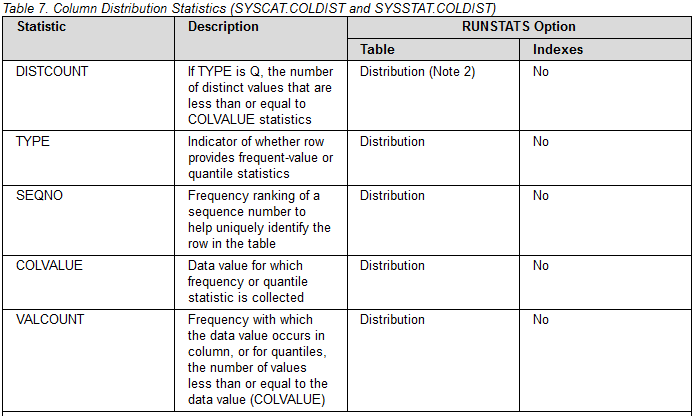






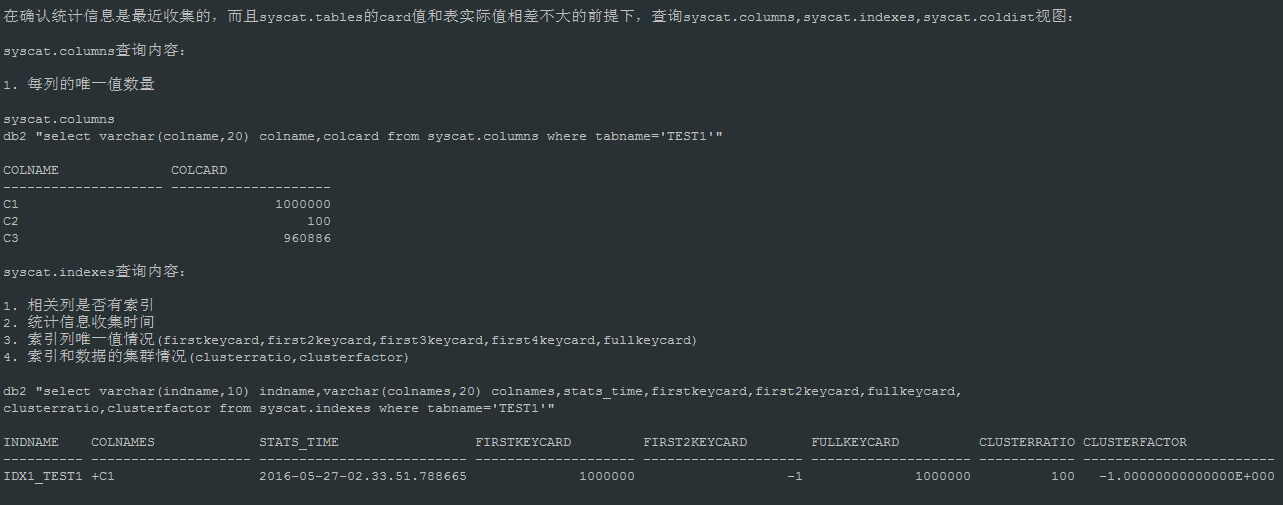


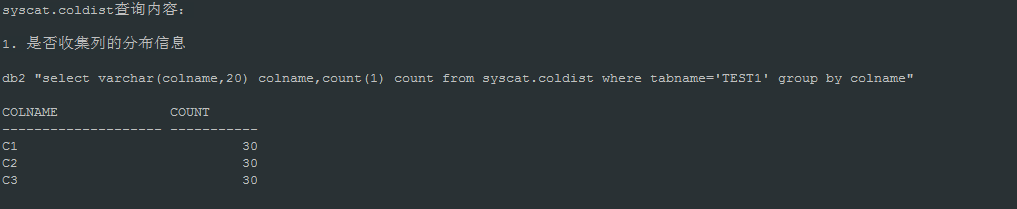




* + 1. **统计信息查询内容**







* 1. **执行计划解读**
  2. **执行计划案例分析**

测试数据：

create table test1\_bak(c1 integer, c2 integer, c3 varchar(128))

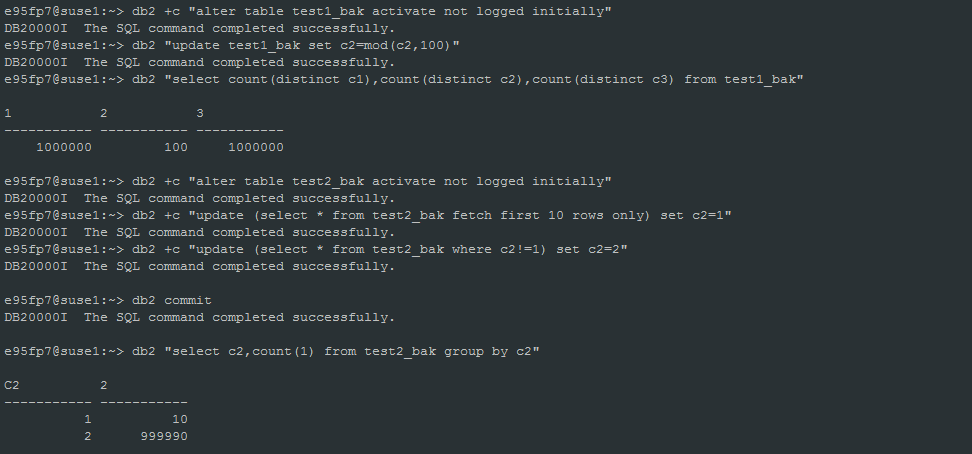
create table test2\_bak(c1 integer, c2 integer, c3 varchar(128))

create table test3\_bak(c1 integer, c2 integer, c3 varchar(128))

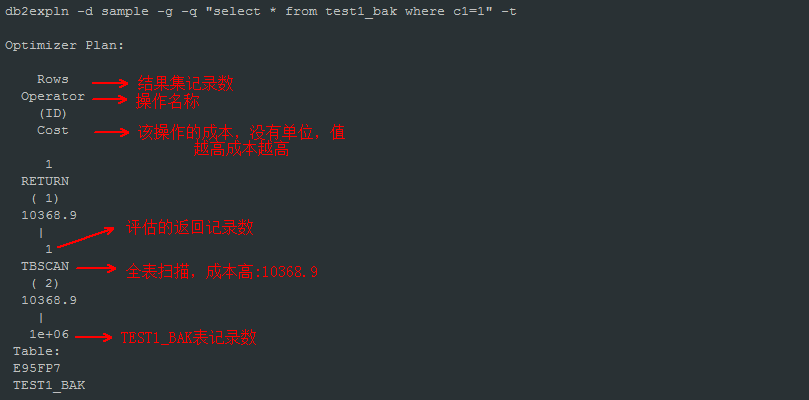
db2 "call sysproc.admin\_cmd('load from(with tmp(c1,c2,c3) as (select 1,1,''done1'' from sysibm.sysdummy1 union all select a.c1 + 1,a.c2+1,''done''||char(a.c1+1) from tmp a where a.c1<1000000) select \* from tmp) of cursor replace into test1\_bak nonrecoverable')"

db2 "call sysproc.admin\_cmd('load from (select \* from test1\_bak) of cursor replace into test2\_bak nonrecoverable')"

db2 "call sysproc.admin\_cmd('load from (select \* from test1\_bak) of cursor replace into test3\_bak nonrecoverable')"



* + 1. 案例1



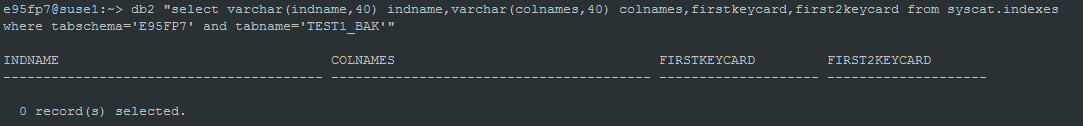
分析优化思路：

查询条件只有c1=1的简单查询，用了全表扫描，表记录数是1000000

优化器通过syscat.tables视图的card字段获得表记录数，该值是runstats时统计的表记录数，并不是表当前真实的记录数。

为什么使用全表扫？是C1列上没有建索引，还是建了索引，但C1列不在索引的第一列上。

或者已经建了索引，但没有用到，没有收集表和索引的统计信息？

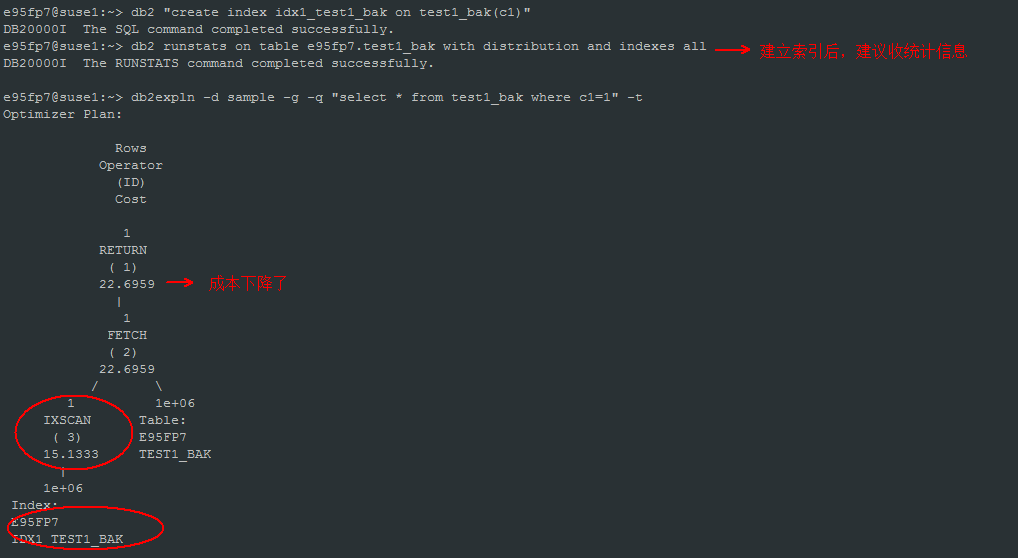


通过查询syscat.indexes视图，发现test1\_bak表的c1列上没有建索引。评估是否需要建索引，C1列的唯一值多不多？如果c1=1的记录数有90万条，表记录数100万，

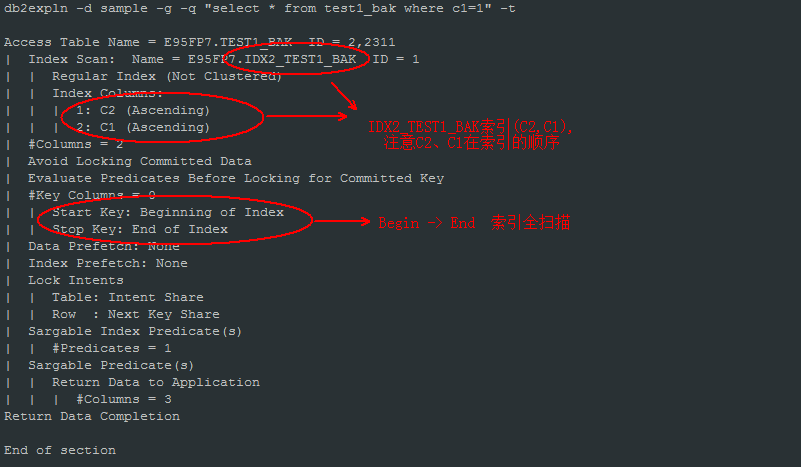
即使c1列上建了索引，最终也使用了索引，那性能也是不好的。

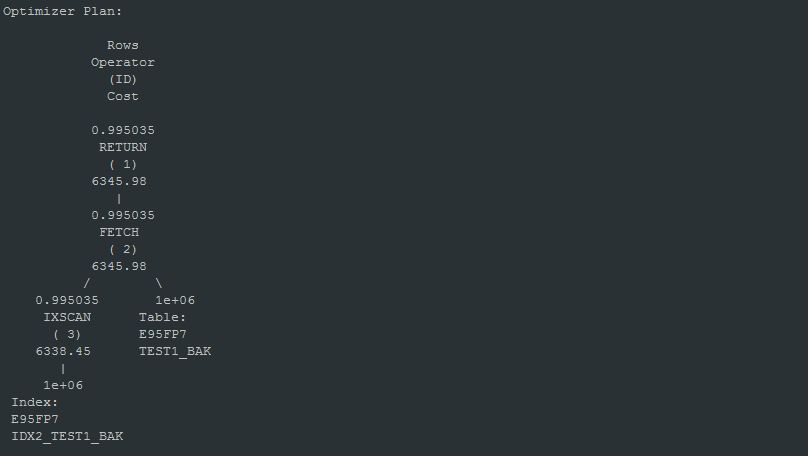


C1=1的记录数只有1条，而且唯一值有100万，C1列是适合建索引的。



* + 1. 案例2





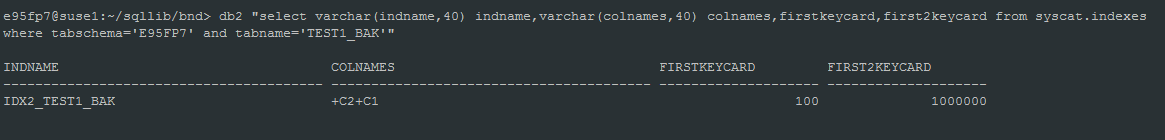
分析优化思路：

还是案例1的SQL，执行计划使用的也是索引扫描，没有问题了么？

这里我们不仅要看执行计划树，还要看上面的详细数据，那里包含了索引如何使用的信息。需要关注索引信息的Start key, Stop key。

SQL在这里使用了索引的全扫描(beginning of index -> end of index)， 如果索引页很多，必然IO开销会很大，性能也不会很好。

为什么会使用索引全扫描呢，我们需要检查索引结构



发现问题了，虽然C1列上有索引，但是并不是索引的第一列，所以需要遍历整棵索引树，找到C1=1的索引页，最终定位具体的行地址。

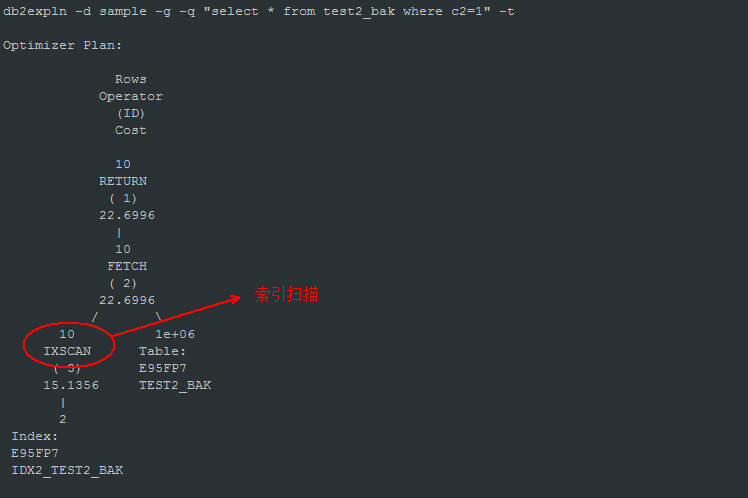
如何优化索引呢？

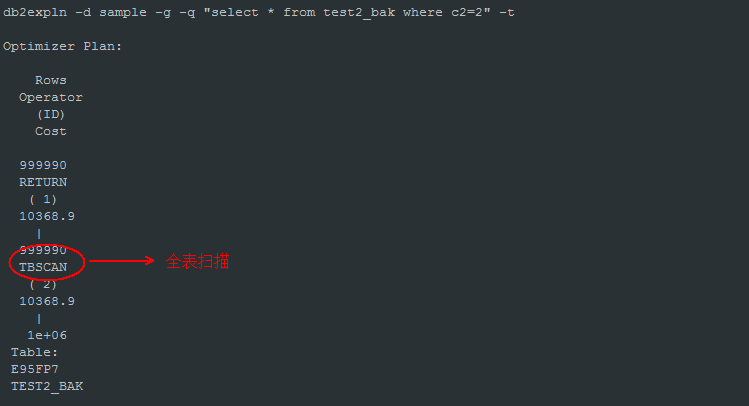
有几种选择： 1. 重建原来的索引，把C1列放前，C2列放后，但是要考虑有没有C2列的查询，否则它们的查询性能也不会很好。

2. 把原来的索引拆分成C1, C2列单独索引

3. 原来索引不变，新建一个C2列的索引

* + 1. 案例3

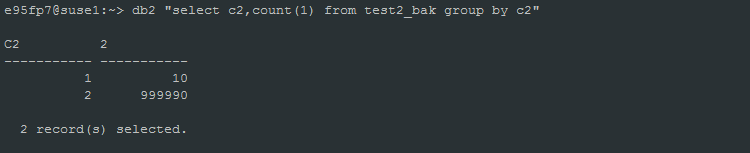




分析优化思路：

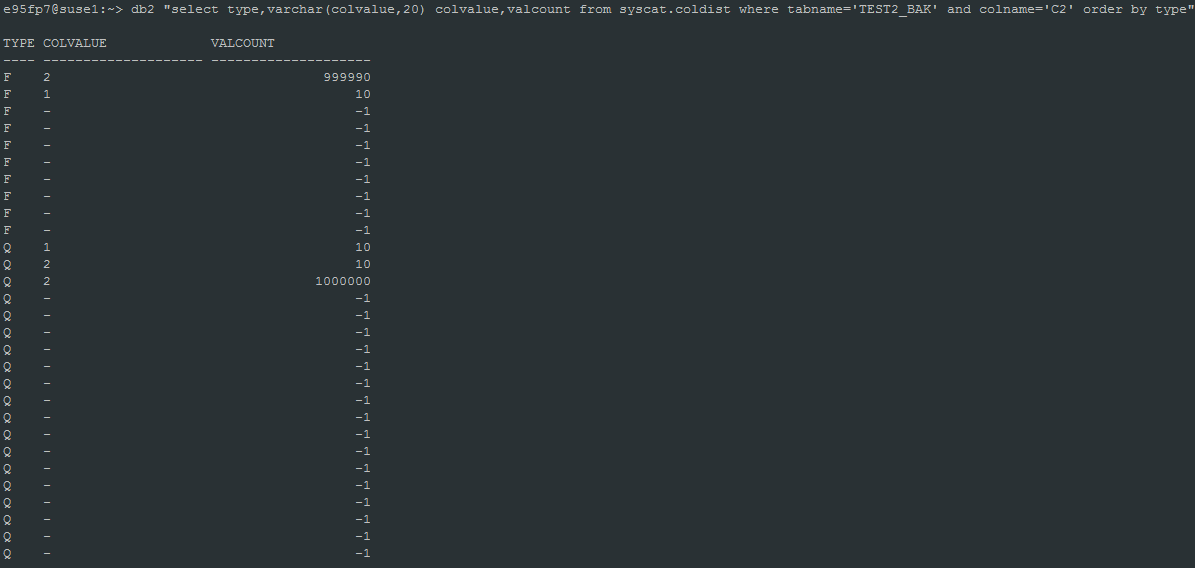
C1列上建了索引，C1=1, C1=2两条SQL使用了完全不同的执行计划，C1=1是索引扫描，C1=2是全表扫描，为什么？

检查值的分布情况



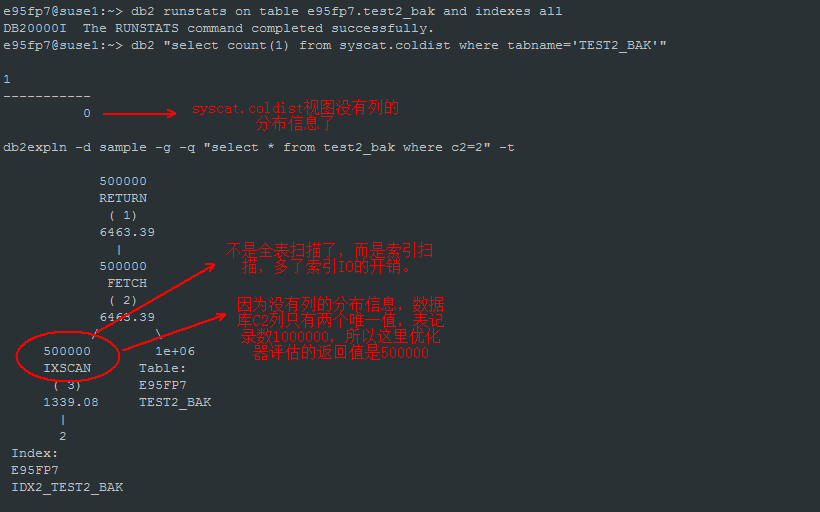
C1=1的值有10条，C1=2值有999990条，值的分布非常不均匀。这里明显是数据库优化器按值的分布情况选择了不同的执行计划。

数据库是如何知道列值的分布情况的？如何收集列的分布信息？列的分布信息存放在SYSIBM. SYSCOLDIST表，视图：SYSCAT.COLDIST。

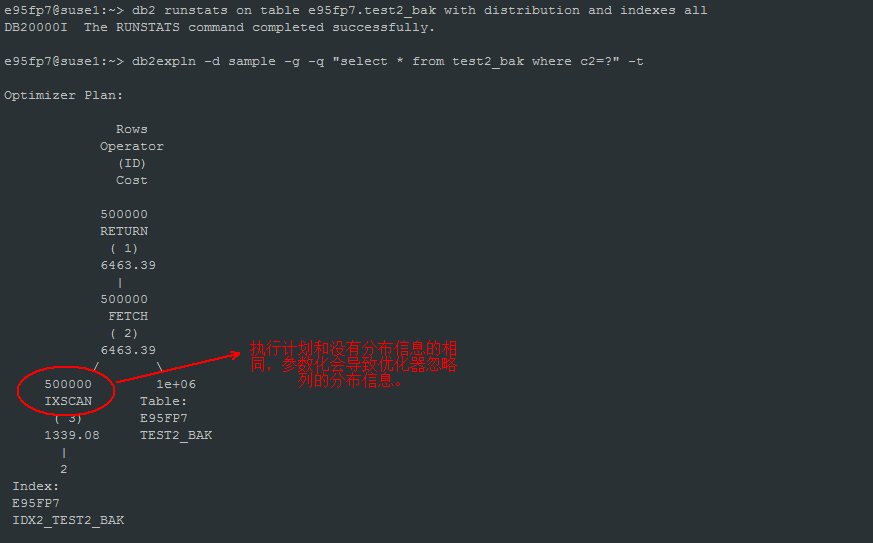


收集列的分布信息: db2 runstats on table e95fp7.test2\_bak with distribution

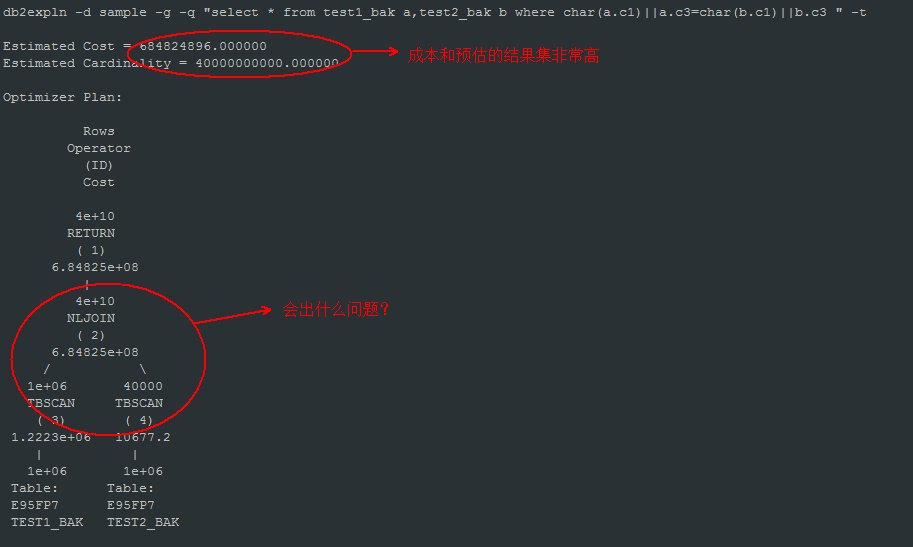
如果不收集列的分布信息，执行计划是怎样的？



有分布信息的情况下，查询条件参数化时的执行计划？



* + 1. 案例4



分析优化思路：

执行计划采用了NLJOIN，左边是驱动表，通过全表扫描扫一次表，每条记录都会扫描右边的Inner表一次，而且每次都是全表扫，算法是N\*M，笛卡尔积，非常坏的算法，性能非常糟糕。

查询条件char(a.c1)||a.c3=char(b.c1)||b.c3，没办法用到HSJOIN或MSJOIN，只能是NLJOIN，而且索引也用不上。

写代码的时候，不能只考虑完成功能就算，要考虑SQL执行的算法也就是执行计划，上面的SQL可能要跑N个小时也跑不出结果。

改成a.c1=b.c1 and a.c3=b.c3，执行计划有什么变化？

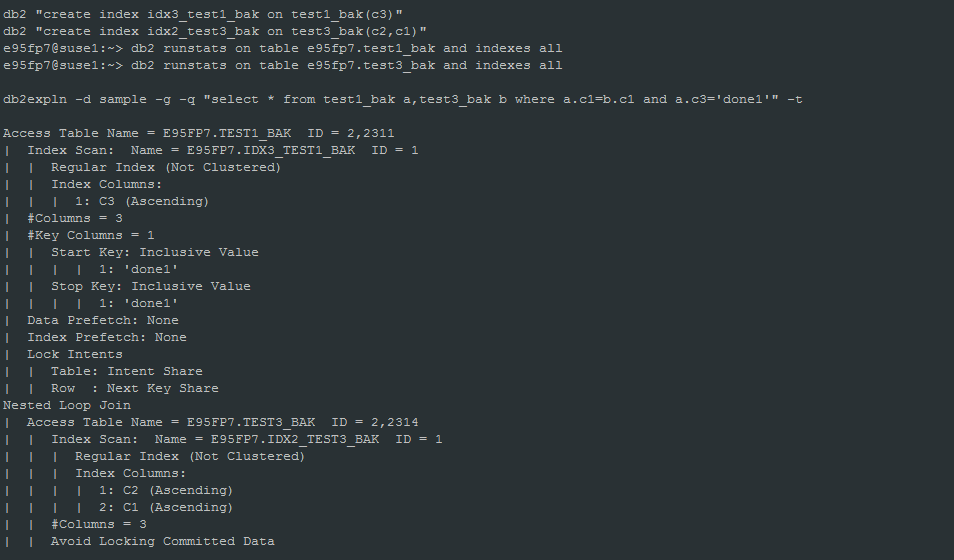


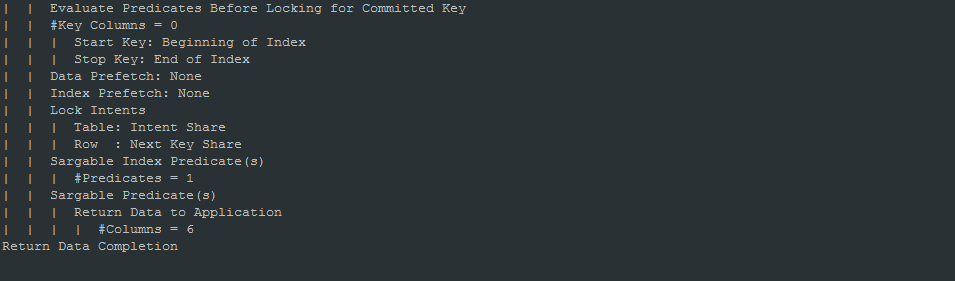
执行计划使用了HSJOIN，虽然要扫描两张大表，但也只是分别扫描一次，算法是N，比之前的NLJOIN算法好多了。

表之间的关联字段不要用||表达式

如果是NLJOIN，要避免N\*M的笛卡尔积算法。执行计划树右边的Inner表必须保证使用高效的索引查询。

* + 1. 案例5

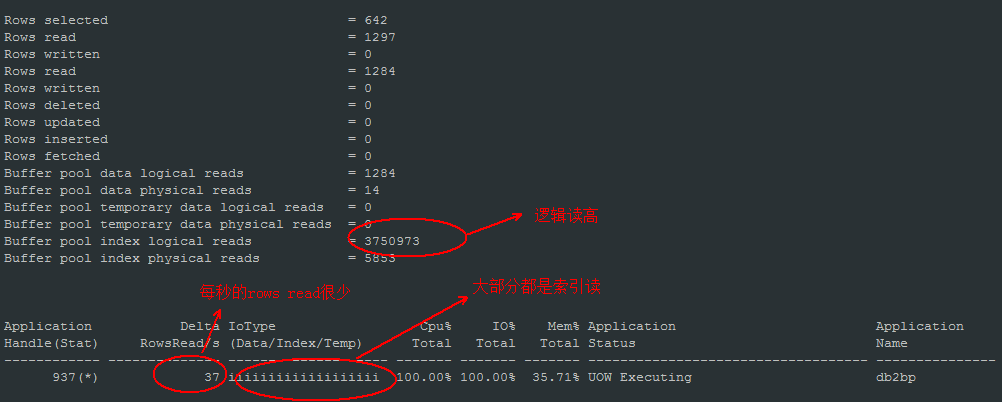




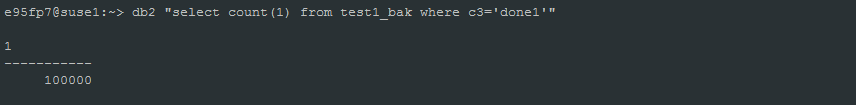


分析优化思路：

以上SQL执行了很长的时间，以下是SQL执行时的快照

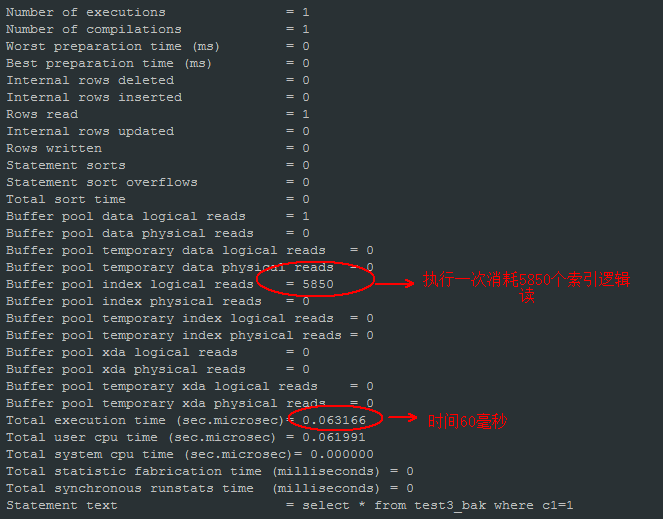


执行计划使用了 NLJOIN，C3=’done1’使用了IDX3\_TEST1\_BAK索引扫描，评估的结果集记录数是1.11111，实际C3=’done1’的记录数是多少？



实际的记录数与评估的记录数相差很大，这导致了优化器采用了不好的执行计划：驱动表逐条取出100000条记录(C3=’done1’)，每条记录再全索引扫描Inner表(TEST3\_BAK)，

索引全扫描的时间大约是60毫秒，1秒的时间驱动表TEST1\_BAK大约只能读出1000/60=16条记录，从而导致了每秒rows read很低，都消耗在了索引扫描上。



两个问题：

1. 为什么C3=’done1’评估的记录数和真实记录数相差那么大？

因为没有收集列的分布信息。

1. 为什么Inner表的索引效率不高？

Inner表使用的索引列不是该索引的第一列，所以需要用到索引全扫描。

我们重新收集TEST1\_BAK表的列分布信息：

db2 runstats on table e95fp7.test1\_bak with distribution and indexes all



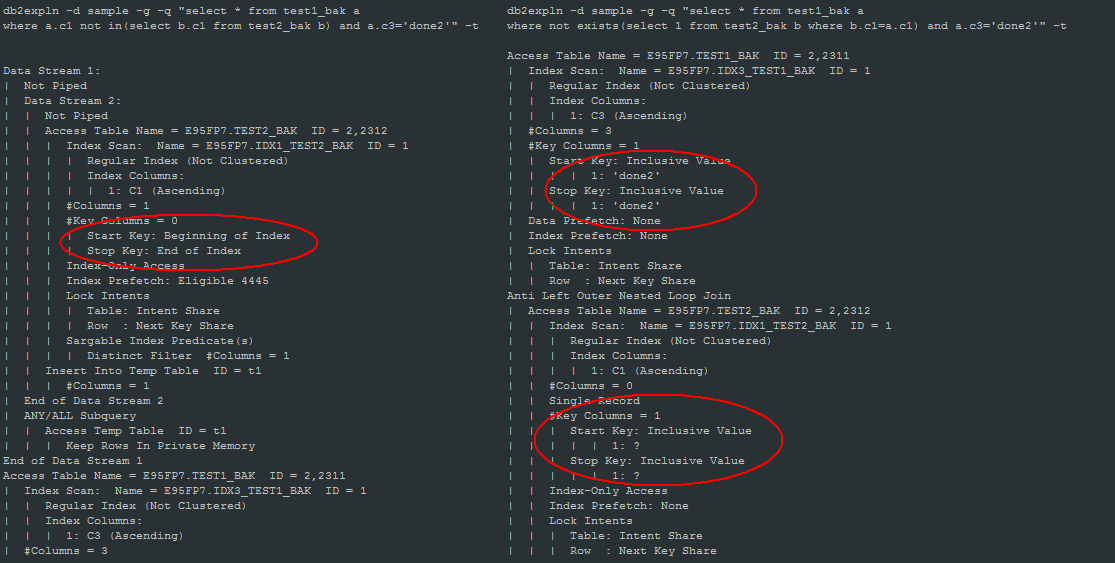
执行计划使用了HSJOIN，SQL很快就执行完成！

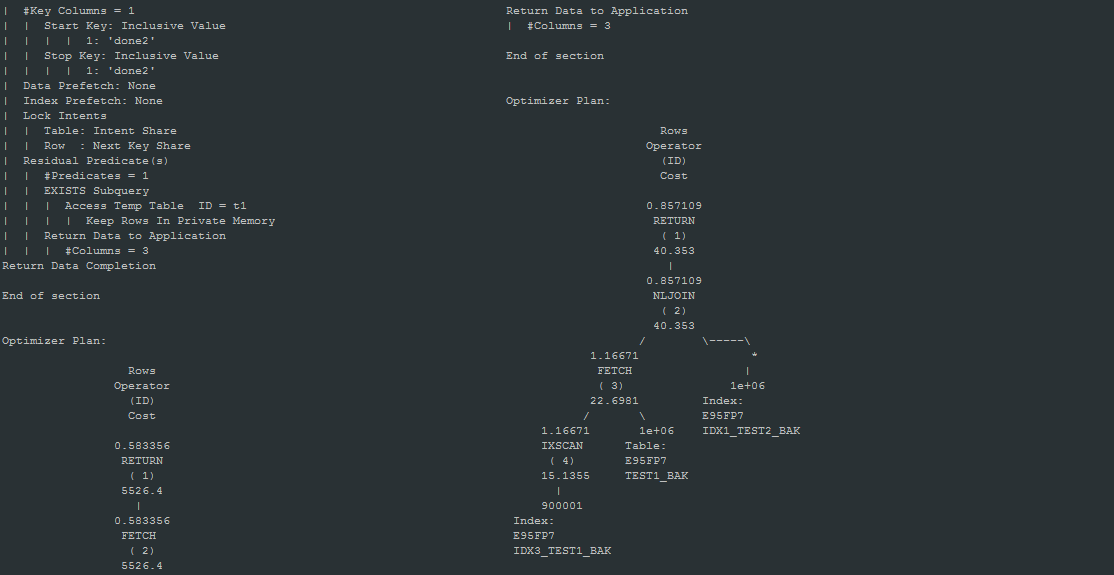
* + 1. 案例6

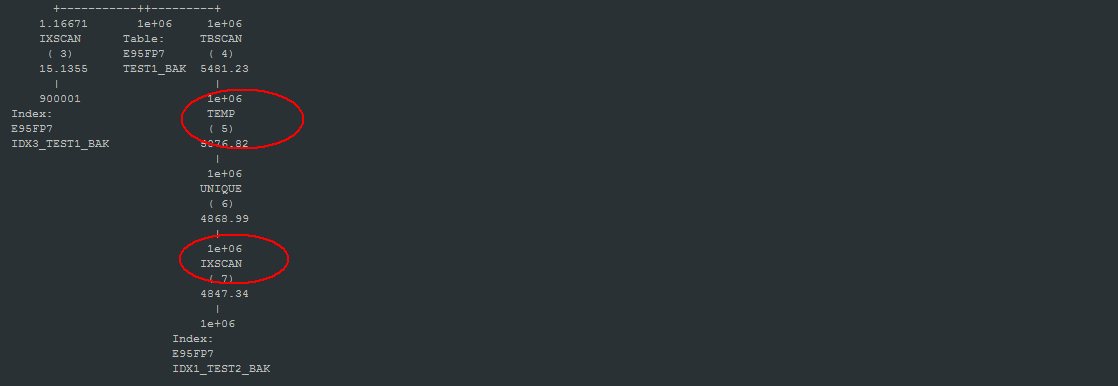
下面两条SQL哪个的性能好？

select \* from test1\_bak a where a.c1 not in(select b.c1 from test2\_bak b) and a.c3='done2'

select \* from test1\_bak a where not exists(select 1 from test2\_bak b where b.c1=a.c1) and a.c3='done2'





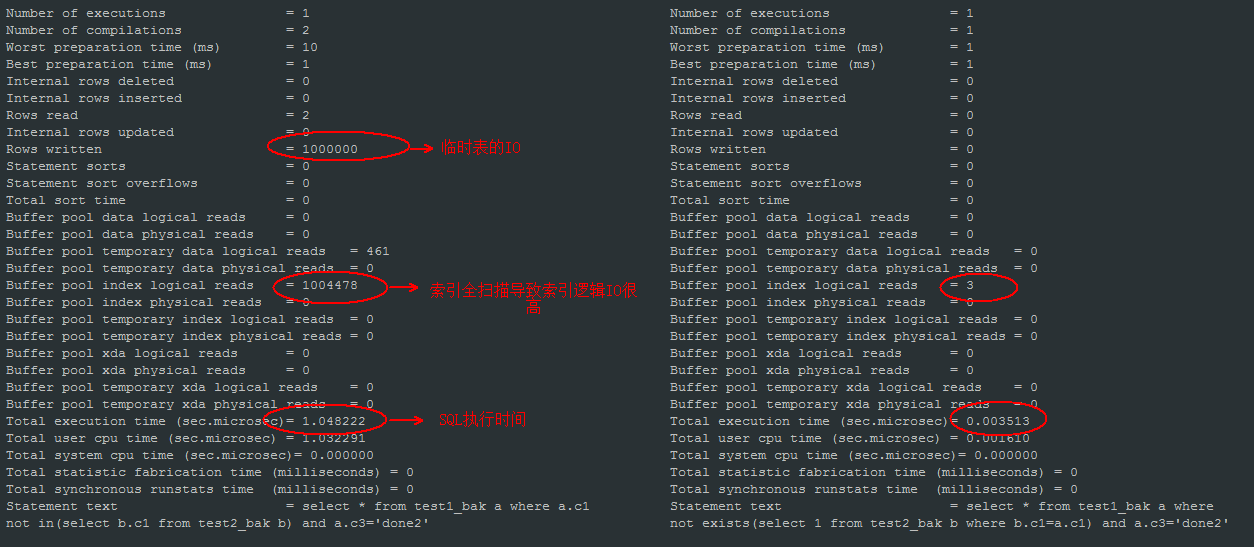


分析优化思路：

NOT IN查询：先是IDX1\_TEST2\_BAK的索引全扫表，把索引列插入临时表里，然后再索引扫描TEST1表，取得数据与临时表里的数据比较，返回数据。开销在临时表的IO上。

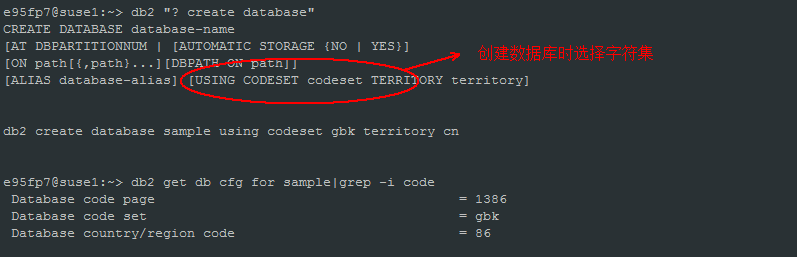
NOT EXISTS查询：先是IDX3\_TEST1\_BAK的索引扫描(C3=’done2’)，’done2’的值不多，索引查询很高效，取得的记录逐条通过IDX1\_TEST2\_BAK索引扫描TEST2\_BAK表，索引效率依然很高效。

再看一下dynamic sql快照的对比：



1. **参数**
   1. Database code set

数据库字符集， 只能在数据库创建的时候指定， 创建数据库后不能修改。



* 1. DFT\_QUERYOPT

优化级别，默认值5

取值范围：

0 - minimal query optimization.

1 - roughly comparable to DB2 Version 1.

2 - slight optimization.

3 - moderate query optimization.

5 - significant query optimization with heuristics to limit the effort expended on selecting an access plan. This is the default.

7 - significant query optimization.

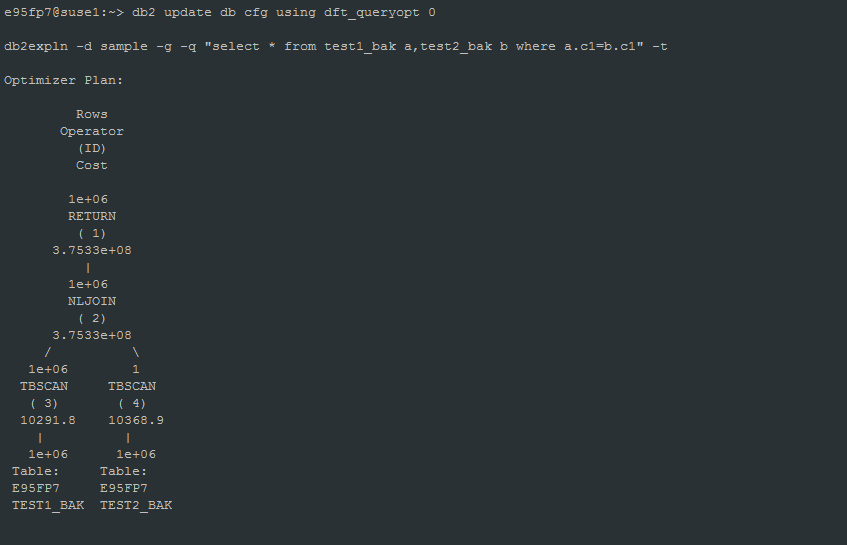
9 - maximal query optimization

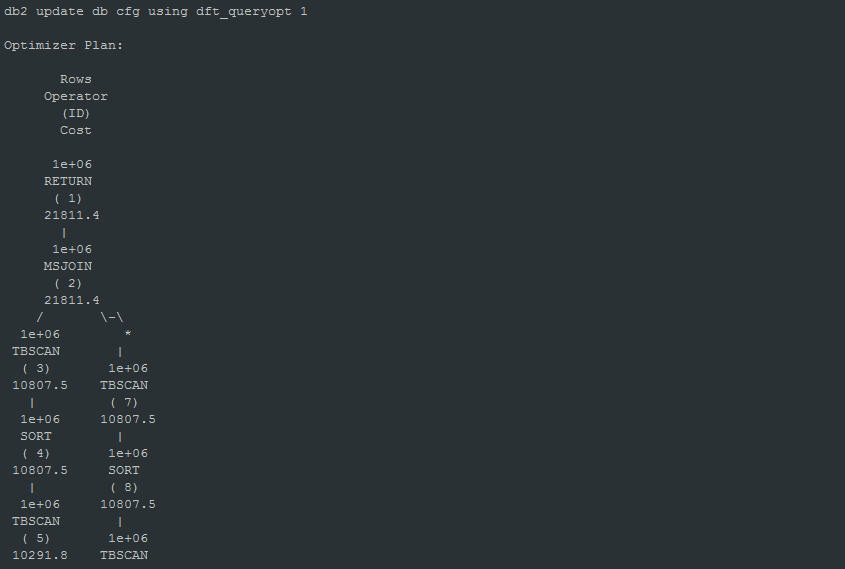
不同的取值对表连接方式的支持：

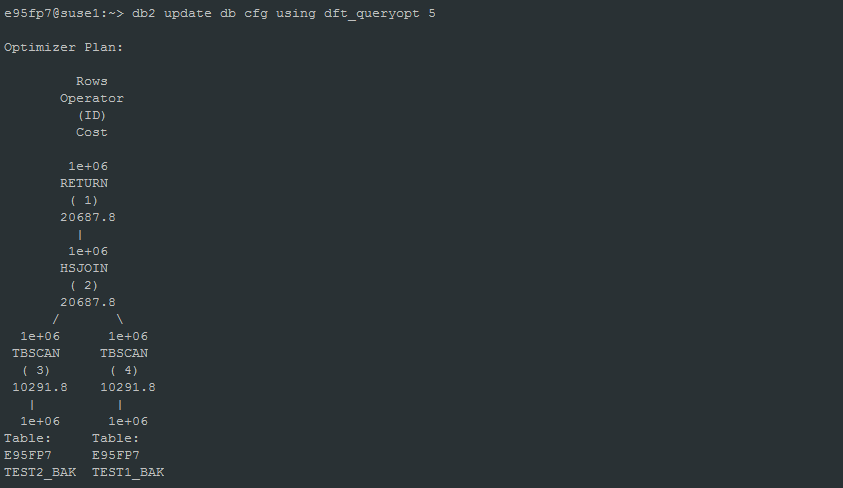
0 – 只支持Nested-loop Join

1及以后 – 支持Merge Join

5及以后 – 支持Hash Join







* 1. DFT\_DEGREE、MAX\_QUERYDEGREE

DFT\_DEGREE:

Specifies the default value for the CURRENT DEGREE special register and the DEGREE bind option.

A value of 1 means no intrapartition parallelism.

A value of -1 (or ANY) means the optimizer determines the degree of intrapartition parallelism based on the number of processors and the type of query.

MAX\_QUERYDEGREE:

Specifies the maximum degree of intrapartition parallelism that is used for any SQL statement executing on this instance of the database manager.

分区内并行特性必须打开,intra\_parallel参数控制

e95fp7@suse1:~> db2 update dbm cfg using intra\_parallel YES

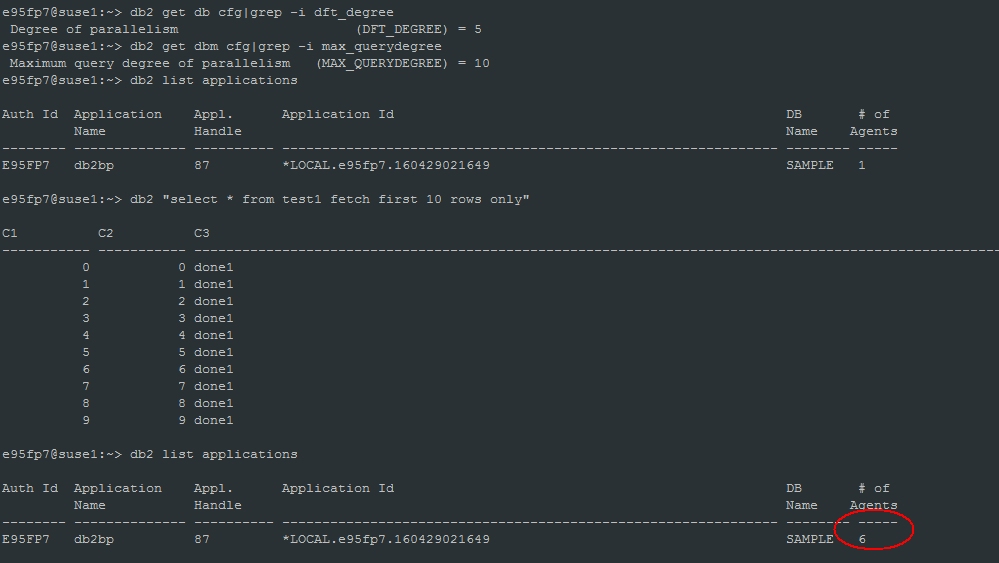
需要重启实例才生效。

分区内并行的运行时并行度由以下三个值的最小值决定：

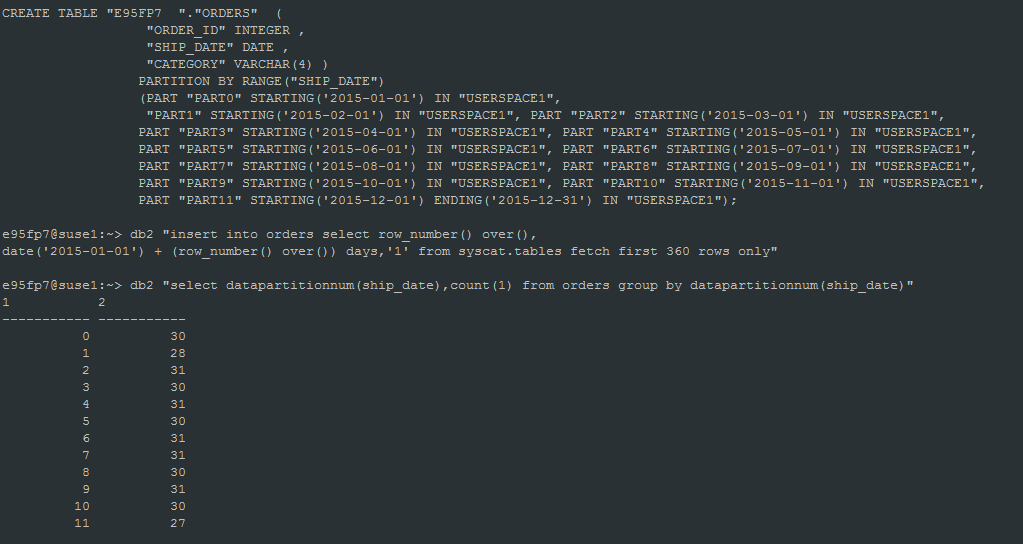
Maximum query degree (max\_querydegree) configuration parameter

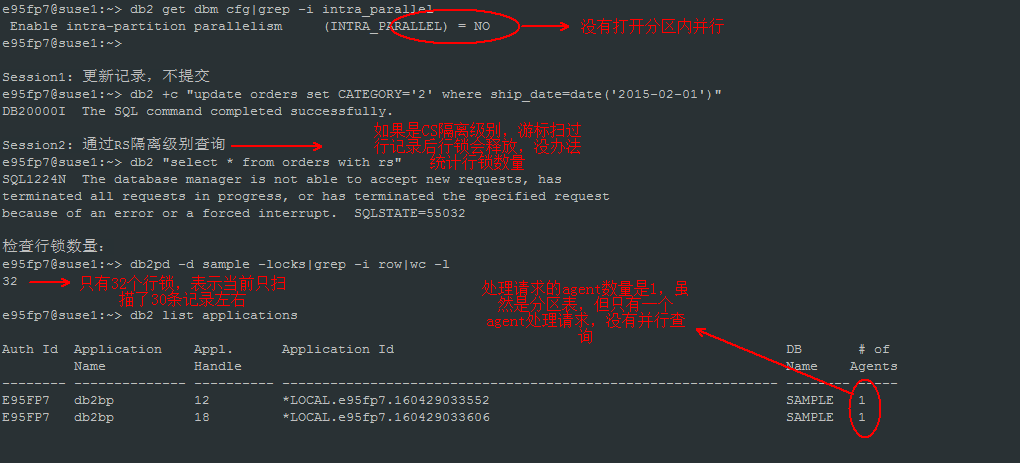
Application runtime degree (set runtime degree ...)

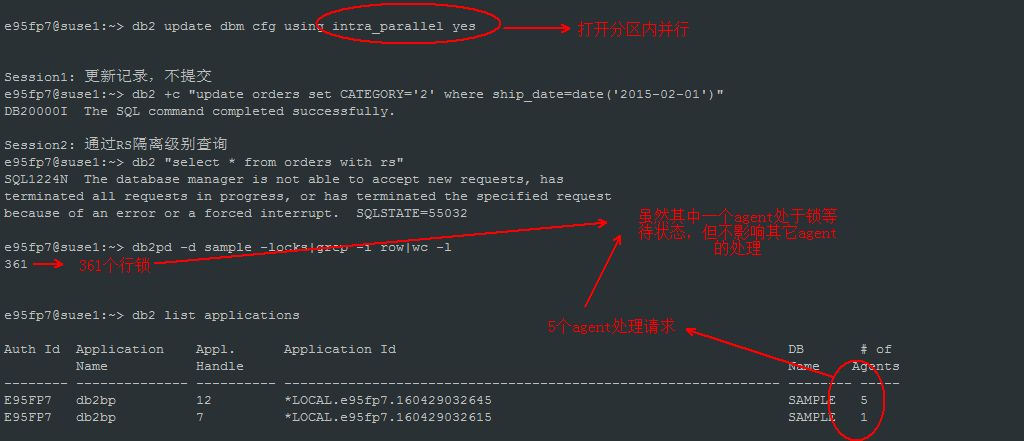
SQL statement compilation degree(dft\_degree)



分区表并行查询实验：







* 1. NUM\_FREQVALUES、NUM\_QUANTILES

控制列分布信息收集的参数。

NUM\_FREQVALUES：

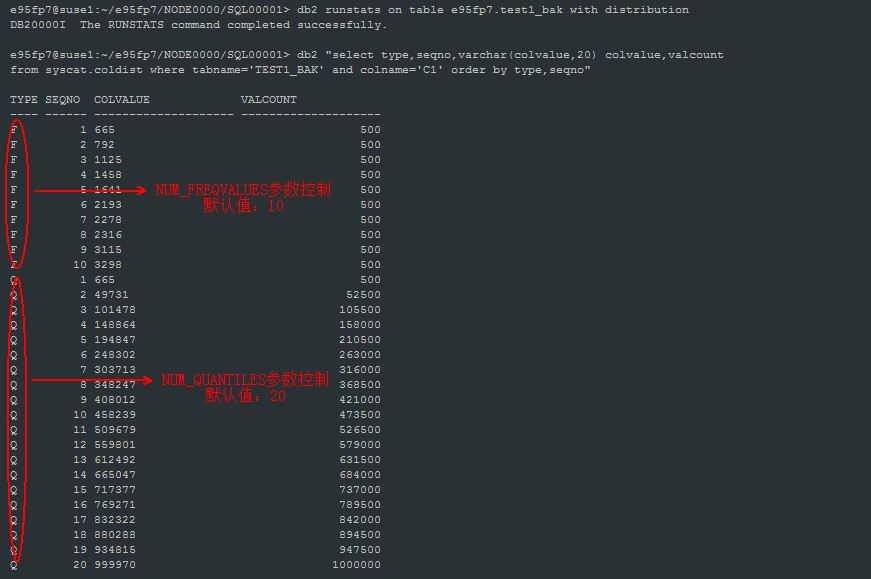
This parameter allows you to specify the number of "most frequent values" that will be collected when the WITH DISTRIBUTION option is specified on the RUNSTATS command.

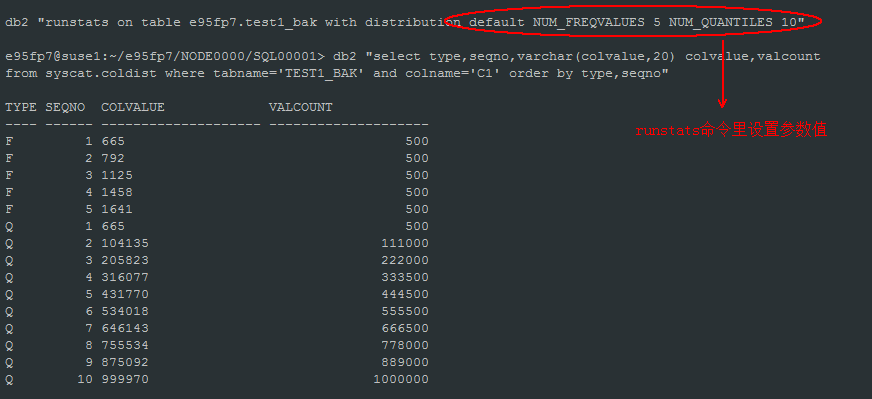
默认值：10

NUM\_QUANTILES：

This parameter controls the number of quantiles that will be collected when the WITH DISTRIBUTION option is specified on the RUNSTATS command.

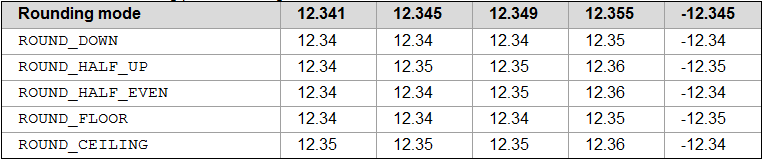
默认值：20



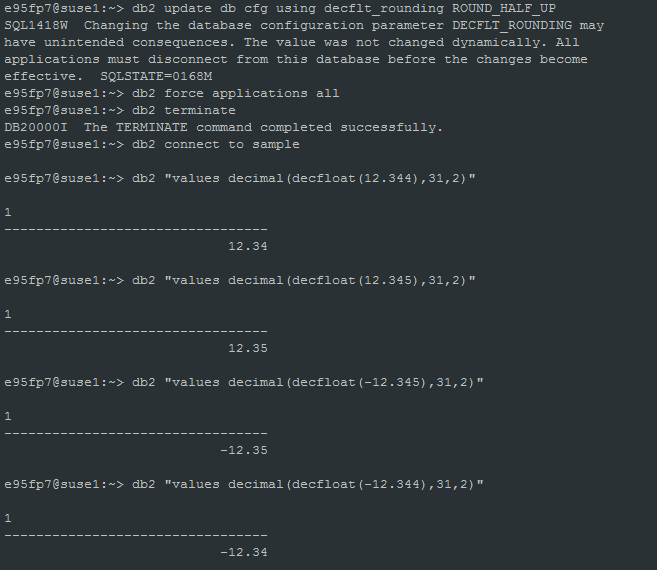


* 1. DECFLT\_ROUNDING

控制小数点取舍的方法。默认值：ROUND\_HALF\_EVEN



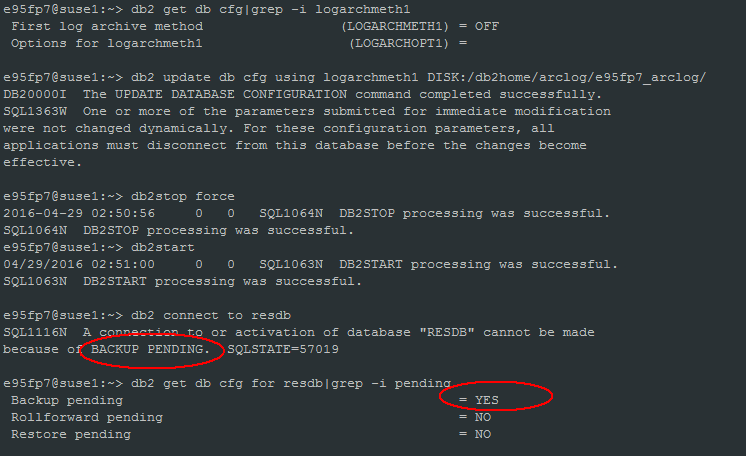
修改为四舍五入方式：



* 1. Backup pending(backup\_pending)

The backup\_pending parameter indicates whether you need to do a full backup of the database before accessing it. 该参数是内部参数，不可手工修改。

This parameter is set to ON only if the database configuration is changed so that the database moves from being unrecoverable to recoverable. That is, initially both the logarchmeth1 and logarchmeth2 parameters were set to OFF, then either one or both of these parameters is set, and the update to the database configuration is accepted.

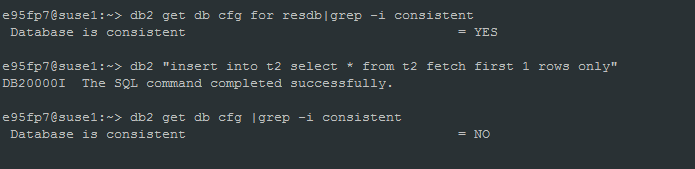


* 1. Database is consistent(database\_consistent)

This parameter indicates whether the database is in a consistent state.

YES indicates that all transactions have been committed or rolled back so that the data is consistent. If the system "crashes" while the database is consistent, you do not need to take any special action to make the database usable.

NO indicates that a transaction is pending or some other task is pending on the database and the data is not consistent at this point. If the system "crashes" while the database is not consistent, you will need to restart the database using the RESTART DATABASE command to make the database usable.



* 1. Multi-page file allocation enabled(multipage\_alloc)

Enables the use of multi-page file allocation for a database. With multi-page file allocation enabled for SMS table spaces, disk space is allocated one extent at a time rather than one page at a time.

Note:

Multipage file allocation is not applicable to temporary table spaces.

Multi-page file allocation only affects the data and index portions of a table. This means that the .LF, .LB, and .LBA files are not extended one extent at a time.

* 1. SELF\_TUNING\_MEM

This parameter determines whether the memory tuner will dynamically distribute available memory resources as required between memory consumers that are enabled for self-tuning.

Because memory is being traded between memory consumers, there must be at least two memory consumers enabled for self-tuning in order for the memory tuner to be active. When self\_tuning\_mem is set to ON, but there are less than two memory consumers enabled for self-tuning, the memory tuner is inactive. (The exception to this is the sort heap memory area, which can be tuned regardless of whether other memory consumers are enabled for self-tuning or not.)

The memory consumers that can be enabled for self-tuning include:

1. Buffer pools (controlled by the size parameter of the ALTER BUFFERPOOL and CREATE BUFFERPOOL statements)
2. Package cache (controlled by the pckcachesz configuration parameter)
3. Lock List ( controlled by the locklist and maxlocks configuration parameters)
4. Sort heap (controlled by the sheapthres\_shr and sortheap configuration parameters)
5. Database shared memory (controlled by the database\_memory configuration parameter)
   1. DATABASE\_MEMORY

This parameter specifies the amount of memory that is reserved for the database shared memory region. If this amount is less than the amount calculated from the individual memory parameters (for example, locklist, utility heap, bufferpools, and so on), the larger amount will be used.

* 1. DB\_MEM\_THRESH

This parameter represents the maximum percentage of committed, but currently unused, database shared memory that the database manager will allow before starting to release committed pages of memory back to the operating system. 默认值：10

This database configuration parameter relates to how the database manager handles excess unused database shared memory. Typically, as pages of memory are touched by a process, they are committed, meaning that a page of memory has been allocated by the operating system and occupies space either in physical memory or in a page file on disk.

* 1. LOCKLIST、MAXLOCKS

LOCKLIST:

This parameter indicates the amount of storage that is allocated to the lock list.

MAXLOCKS:

This parameter defines a percentage of the lock list held by an application that must be filled before the database manager performs lock escalation.

The database manager determines which locks to escalate by looking through the lock list for the application and finding the table with the most row locks.

锁升级条件：

1. 应用持有的锁数量超过了MAXLOCKS的限制
2. 锁列表内存用尽

锁升级的日志信息记录在db2diag.log里。

* 1. PCKCACHESZ

This parameter is allocated out of the database shared memory, and is used for caching of sections for static and dynamic SQL and XQuery statements on a database.

Caching packages allows the database manager to reduce its internal processing time by eliminating the need to access the system catalogs when reloading a package; or, in the case of dynamic SQL or XQuery statements, eliminating the need for compilation. Sections are kept in the package cache until one of the following events occurs:

1. The database is shut down
2. The package or dynamic SQL or XQuery statement is invalidated
3. The cache runs out of space.

**The limit specified by the pckcachesz parameter is a soft limit.**

**PCKCACHESZ是软限制，实际值可以超过参数的设置值。在V10.5 FP4有个PCKCACHESZ导致内存耗尽的APAR:**

IBM IT04326: PACKAGE CACHE MEMORY USE CAN EXCEED CONFIGURED SIZE RESULTING IN MEMORY EXHAUSTION ON DB2 Cancun Release 10.5.0.4.

APAR描述：

When the size of the package cache reaches its configured limit and new entries in the package cache are needed, DB2 will evict

older entries in order to maintain the size at the configured limit. With this APAR, under certain timing conditions, if the

size limit is reached, rather than evicting older entries, the package cache may exceed its configured limit and continue to

grow. This can result in the package cache unexpectedly consuming additional database memory or instance memory.

* 1. SHEAPTHRES\_SHR、SORTHEAP

SHEAPTHRS\_SHR:

This parameter represents a soft limit on the total amount of database shared memory that can be used by sort memory consumers at any one time.

SORTHEAP:

This parameter defines the maximum number of private memory pages to be used for private sorts, or the maximum number of shared memory pages to be used for shared sorts.

* 1. DBHEAP

This parameter determines the maximum memory used by the database heap.

With Version 9.5, this database configuration parameter has a default value of AUTOMATIC, meaning that the database heap can increase as needed until either the database\_memory limit is reached, or the instance\_memory limit is reached.

There is one database heap per database, and the database manager uses it on behalf of all applications connected to the database. It contains control block information for tables, indexes, table spaces, and buffer pools. It also contains space for the log buffer (logbufsz) and temporary memory used by utilities.

* 1. CATALOGCACHE\_SZ

This parameter specifies the maximum space in pages that the catalog cache can use from the database heap.

This parameter is allocated out of the database shared memory, and is used to cache system catalog information.

* 1. LOGBUFSZ

This parameter allows you to specify the amount of the database heap (defined by the dbheap parameter) to use as a buffer for log records before writing these records to disk.

Log records are written to disk when one of the following occurs:

1. A transaction commits or a group of transactions commit, as defined by the mincommit configuration parameter
2. The log buffer is full
3. As a result of some other internal database manager event.

V9.5该参数默认值是8(32K)，默认值太小，需要修改，一般为1~4M即可。

通过以下指标观察LOGBUFSZ设置是否合适：

e95fp7@suse1:~/sqllib/db2dump> db2 get snapshot for db on resdb|grep -i full

Number log buffer full = 0

如果log buffer full的次数很多，则需要适当增加LOGBUFSZ的大小。

* 1. UTIL\_HEAP\_SZ

This parameter indicates the maximum amount of memory that can be used simultaneously by the backup, restore, and load (including load recovery) utilities.

* 1. STMTHEAP

This parameter specifies the size of the statement heap, which is used as a work space for the SQL or XQuery compiler during compilation of an SQL or XQuery statement.

With Version 9.5, this database configuration parameter has a default value of AUTOMATIC, meaning that it increases as needed until either the appl\_memory limit is reached, or the instance\_memory limit is reached.

* 1. APPLHEAPSZ

In previous releases, the applheapsz database configuration parameter referred to the amount of application memory each individual database agent working for that application could consume. With Version 9.5, applheapsz refers to the total amount of application memory that can be consumed by the entire application.

With Version 9.5, this database configuration parameter has a default value of AUTOMATIC, meaning that it increases as needed until either the appl\_memory limit is reached, or the instance\_memory limit is reached.

* 1. APPL\_MEMORY

This parameter allows to control the maximum amount of application memory that is allocated by DB2 database agents to service application requests. By default, its value is set to AUTOMATIC, meaning that all application memory requests will be allowed as long as the total amount of memory allocated by the database partition is within the instance\_memory limits.

* 1. STAT\_HEAP\_SZ

This parameter indicates the maximum size of the heap used in collecting statistics using the RUNSTATS command.

With Version 9.5, this database configuration parameter has a default value of AUTOMATIC, meaning that it increases as needed until either the appl\_memory limit is reached, or the instance\_memory limit is reached.

* 1. DLCHKTIME、LOCKTIMEOUT

DLCHKTIME: 死锁检测间隔，默认10秒。

LOCKTIMEOUT: 锁超时时间，默认-1，没有锁超时。

建议：

In a transaction processing (OLTP) environment, you can use an initial starting value of 30 seconds.

* 1. NUM\_IOCLEANERS、NUM\_IOSERVERS

NUM\_IOCLEANERS:

This parameter allows you to specify the number of asynchronous page cleaners for a database.

NUM\_IOSERVERS:

This parameter specifies the number of I/O servers for a database.

I/O servers, also called prefetchers, are used on behalf of the database agents to perform prefetch I/O and asynchronous I/O by utilities such as backup and restore.

* 1. TRACKMOD

设置该值支持增量备份，设置完后需要做个全量备份。

After setting this parameter to "Yes", you must take a full database backup in order to have a baseline against which incremental backups can be taken. Also, if this parameter is enabled and if a table space is created, then a backup must be taken which contains that table space. This backup could be either a database backup or a table space backup. Following the backup, incremental backups will be permitted to contain this table space.

* 1. MAXFILOP

This parameter specifies the maximum number of file handles that can be open per database.(start with v9.5)

If opening a file causes this value to be exceeded, some files in use by this database are closed. If maxfilop is too small, the additional processing time of opening and closing files will become excessive and might degrade performance.

Both SMS table spaces and DMS table space file containers are treated as files in the database manager's interaction with the operating system, and file handles are required. More files are generally used by SMS table spaces compared to the number of containers used for a DMS file table space. Therefore, if you are using SMS table spaces, you will need a larger value for this parameter compared to what you would require for DMS file table spaces.

V9.1, MAXFILOP是每个应用能打开的句柄最大数量。设置过大，内存消耗会很多(AIX环境)。

* 1. LOGFILSIZ、LOGPRIMARY、LOGSECOND

LOGFILSIZ:

This parameter defines the size of each primary and secondary log file.

单个日志文件大小：默认值:1000，单位4KB

LOGPRIMARY:

This parameter allows you to specify the number of primary log files to be preallocated.

主日志文件个数，预分配空间。默认值:3

LOGSECOND:

This parameter specifies the number of secondary log files that are created and used for recovery log files (only as needed).

辅助日志文件个数，只在需要的时候分配空间。默认值:2

If logsecond has a value of -1, logprimary <= 256.

If logsecond does not have a value of -1, (logprimary + logsecond) <= 256.

对于生产系统来说，LOGFILSIZ、LOGPRIMARY的值设置都过小，需要按实际情况调整。

* 1. NEWLOGPATH、MIRRORLOGPATH

NEWLOGPATH: 活动日志文件目录

MIRRORLOGPATH: 镜像日志文件目录

为了避免IO争用，建议NEWLOGPATH不要和数据、索引文件放同一个磁盘。

* 1. BLK\_LOG\_DSK\_FUL

This parameter can be set to prevent disk full errors from being generated when DB2 cannot create a new log file in the active log path.

在日志文件空间满导致无法创建新的日志文件的时候，是否阻止disk full的错误抛出。默认值是NO。

Setting blk\_log\_dsk\_ful to yes causes applications to hang when DB2 encounters a log disk full error.

If blk\_log\_dsk\_ful is set to no, then a transaction that receives a log disk full error will fail and will be rolled back.

In some situations, the database will come down if a transaction causes a log disk full error.

生产系统建议设置为NO

* 1. BLOCKNONLOGGED

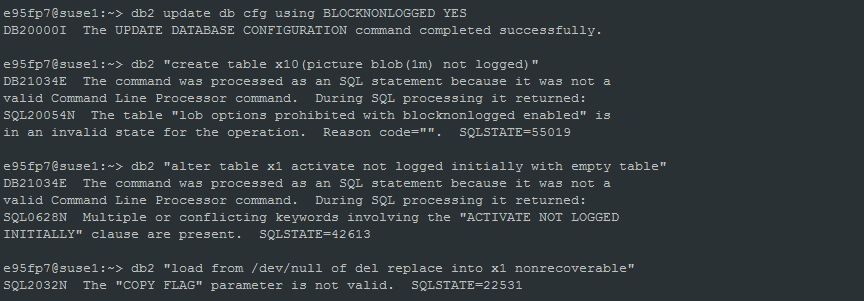
This parameter specifies whether the database manager will allow tables to have the NOT LOGGED or NOT LOGGED INITIALLY attributes activated.

If blocknonlogged is set to YES, then CREATE TABLE and ALTER TABLE statements will fail if one of the following conditions is true:

1. The NOT LOGGED INITIALLY parameter is specified.
2. The NOT LOGGED parameter is specified for a LOB column.
3. A CLOB, DBCLOB, or BLOB column is defined as not logged.

If blocknonlogged is set to YES, then the LOAD command fails if the following situations exist:

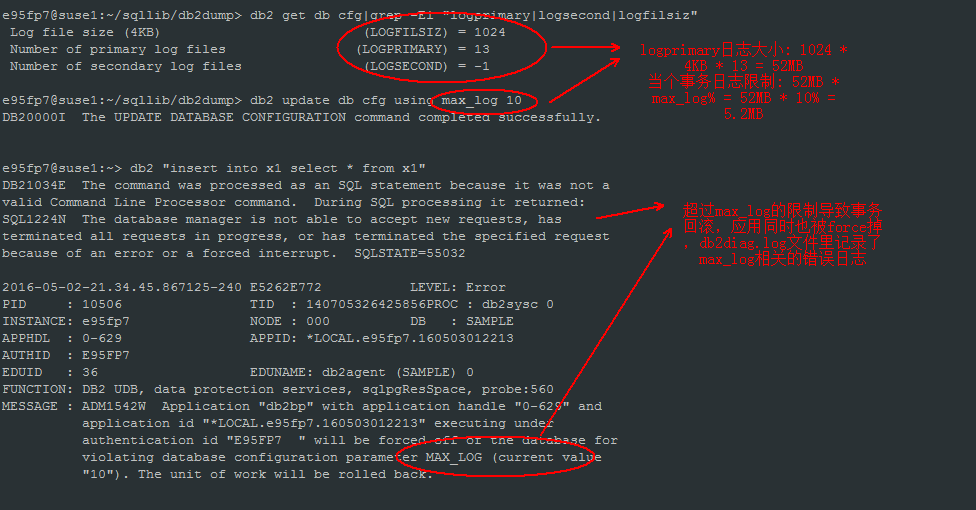
1. You specify the NONRECOVERABLE option.
2. You specify the COPY NO option.



* 1. MAX\_LOG、NUM\_LOG\_SPAN

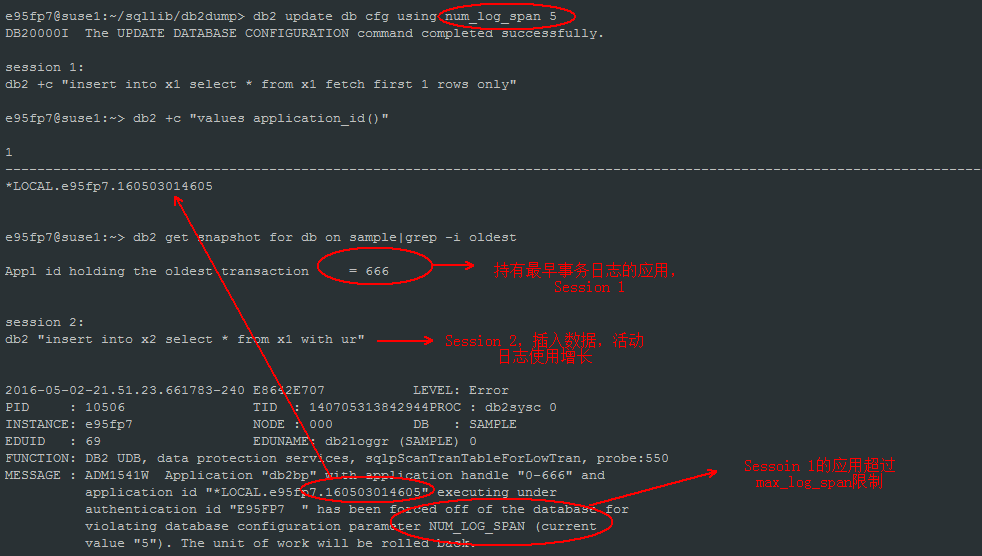
MAX\_LOG:

This parameter specifies if there is a limit to the percentage of the primary log space that a transaction can consume, and what that limit is.



NUM\_LOG\_SPAN:

This parameter specifies whether there is a limit to how many log files one transaction can span, and what that limit is.



以下操作不受max\_log, num\_log\_span参数限制：

ARCHIVE LOG, BACKUP DATABASE, **LOAD, REORG**, RESTORE DATABASE, and ROLLFORWARD DATABASE.

* 1. MINCOMMIT

The MINCOMMIT parameter allows you to delay the writing of log records to disk until a minimum number of commits have been performed by the applications.

MINCOMMIT is a void parameter when set for AIX 64 bit running AIX 5.2 and above.

MINCOMMIT is ignored in the following versions and fix packs of DB2:

- DB2 9.1 - all DB2 9.1 fix packs on AIX

- DB2 9.5 - all DB2 9.5 fix packs on AIX

- DB2 9.7 - GA, FP1 and FP2 on AIX. It's re-enabled starting with FP3.

- DB2 9.8 - GA, FP1 and FP2 on AIX. It's re-enabled starting with FP3.

* 1. CHNGPGS\_THRESH、SOFTMAX

CHNGPGS\_THRESH：

This parameter specifies the level (percentage) of changed pages at which the asynchronous page cleaners will be started, if they are not currently active.

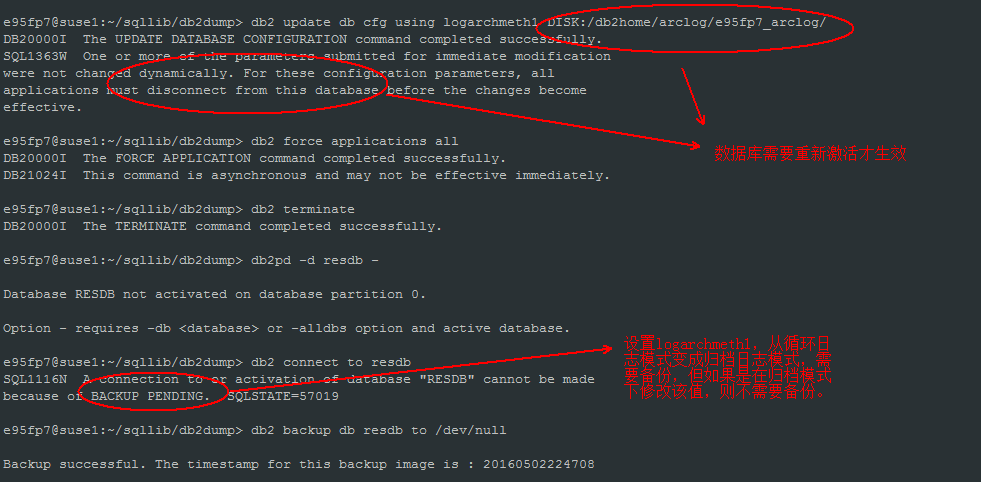
SOFTMAX:

This parameter determines the frequency of soft checkpoints and the recovery range, which help out in the crash recovery process.

* 1. LOGARCHMETH1

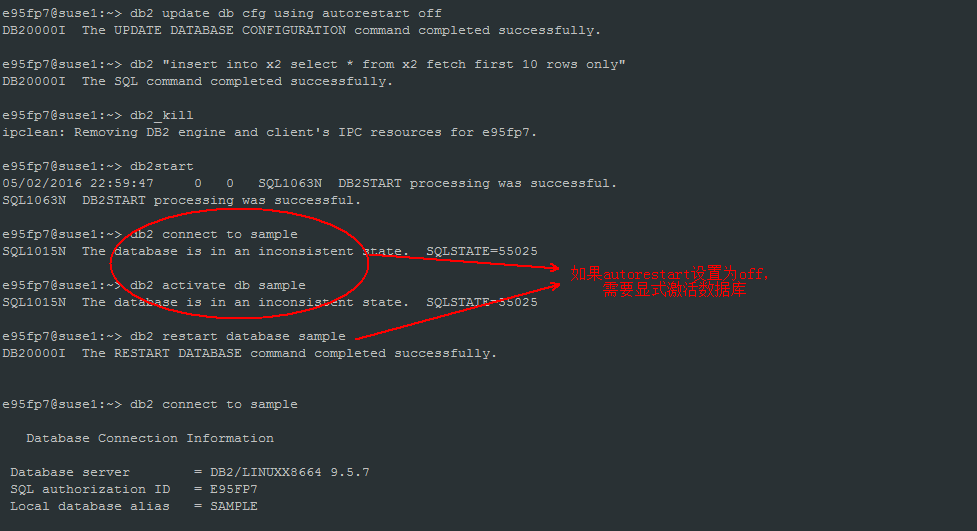
日志归档路径

This parameter specifies the media type of the primary destination for logs that are archived from the current log path.



* 1. AUTORESTART

The autorestart configuration parameter determines if the database manager automatically initiates crash recovery when a user connects to a database that had previously terminated abnormally. If the autorestart configuration parameter is not set, the user must issue an explicit restart database command before they can connect to the database.



* 1. INDEXREC

This parameter indicates when the database manager will attempt to rebuild invalid indexes, and whether or not any index build will be redone during DB2 rollforward

or HADR log replay on the standby database.

* 1. LOGINDEXBUILD

This parameter specifies whether index creation, recreation, or reorganization operations are to be logged so that indexes can be reconstructed during DB2 rollforward operations or high availability disaster recovery (HADR) log replay procedures.

默认值是OFF，设置OFF可以减少索引创建、重组时生成的日志量，提高性能。

HADR环境一定要设置为ON。



* 1. REC\_HIS\_RETENTN、NUM\_DB\_BACKUPS、AUTO\_DEL\_REC\_OBJ

历史文件相关的参数.

A recovery history file is created with each database and is automatically updated whenever:

A database or table spaces are backed up

A database or table spaces are restored

A database or table spaces are rolled forward

A database is automatically rebuilt and more than one image is restored

A table space is created

A table space is altered

A table space is quiesced

A table space is renamed

A table space is dropped

A table is loaded

A table is dropped (when dropped table recovery is enabled)

A table is reorganized

On-demand log archiving is invoked

A new log file is written to (when using recoverable logging)

A log file is archived (when using recoverable logging)

A database is recovered

REC\_HIS\_RETENTN:

This parameter specifies the number of days that historical information on backups will be retained.

设置历史文件(db2rhist.asc、 db2rhist.bak)信息保存的天数，默认值是366。历史文件过大，会影响load、drop table、commit的性能，建议历史文件不要超过10MB。

No matter how small the retention period, the most recent full database backup plus its restore set will always be kept, unless you use the PRUNE utility with the FORCE option.

不管REC\_HIS\_RETENTN值设置多小，最近的数据库全备份和恢复集都会被保留。

NUM\_DB\_BACKUPS:

This parameter specifies the number of database backups to retain for a database. 默认值: 12

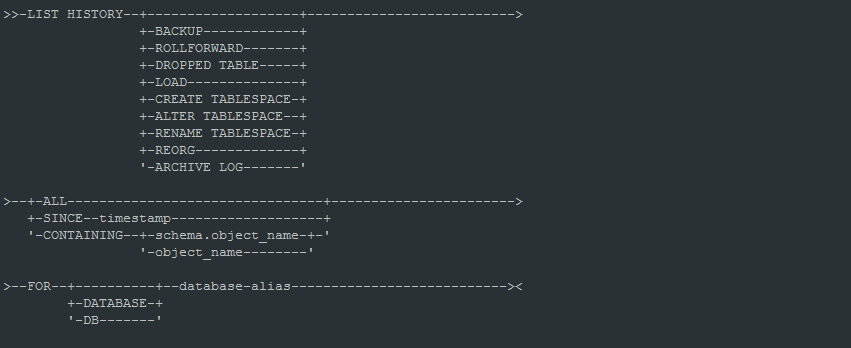
After the specified number of backups is reached, old backups are marked as expired in the recovery history file. Recovery history file entries for the table space backups and load copy backups that are related to the expired database backup are also marked as expired. When a backup is marked as expired, the physical backups can be removed from where they are stored (for example, disk, tape, TSM). The next database backup will prune the expired entries from the recovery history file.

AUTO\_DEL\_REC\_OBJ:

This parameter specifies whether database log files, backup images, and load copy images should be deleted when their associated recovery history file entry is pruned.

默认值:OFF

查看历史文件命令：



1. 查看sample数据库的所有备份历史记录

db2 list history backup all for sample

1. 查看sample数据库20160501以来的备份历史记录

db2 list history backup since 20160501 for sample

* 1. AUTO\_RUNSTATS、AUTO\_REORG

设置数据库自动收集统计信息、表自动重组。默认ON。建议关闭这些自动特性。

* 1. FEDERATED

This parameter enables or disables support for applications submitting distributed requests for data managed by data sources (such as the DB2 Family and Oracle).

默认值：NO

联邦数据库需要设置该值为YES

* 1. JDK\_PATH

This parameter specifies the directory under which the Software Developer's Kit (SDK) for Java, to be used for running Java stored procedures and user-defined functions, is installed.

* 1. DIAGLEVEL、DIAGPATH

DIAGLEVEL:

This parameter specifies the type of diagnostic errors that will be recorded in the db2diag.log file.

Valid values for this parameter are:

0 – No diagnostic data captured

1 – Severe errors only

2 – All errors

3 – All errors and warnings

4 – All errors, warnings and informational messages

默认值：3，生产上不要设置为4。

DIAGPATH:

This parameter allows you to specify the fully qualified path for DB2 diagnostic information.

* 1. DFT\_MON\_BUFPOOL(DFT\_MON\_XXX)
  2. SYSADM\_GROUP、SYSCTRL\_GROUP、SYSMAINT\_GROUP、SYSMON\_GROUP

SYSADM\_GROUP:

This parameter defines the group name with SYSADM authority for the database manager instance.

SYSCTRL\_GROUP:

This parameter defines the group name with system control (SYSCTRL) authority.

SYSCTRL has privileges allowing operations affecting system resources, but does not allow direct access to data.

SYSMAINT\_GROUP:

This parameter defines the group name with system maintenance (SYSMAINT) authority.

SYSMON\_GROUP:

This parameter defines the group name with system monitor (SYSMON) authority.

* 1. AUTHENTICATION

This parameter specifies and determines how and where authentication of a user takes place.

* 1. MON\_HEAP\_SZ

This parameter determines the amount of the memory, in pages, to allocate for database system monitor data. Memory is allocated from the monitor heap when you perform database monitoring activities such as taking a snapshot, turning on a monitor switch, resetting a monitor, or activating an event monitor.

With Version 9.5, this database configuration parameter has a default value of AUTOMATIC, meaning that the monitor heap can increase as needed until the instance\_memory limit is reached.

* 1. JAVA\_HEAP\_SZ

This parameter determines the maximum size of the heap that is used by the Java interpreter started to service Java DB2 stored procedures and UDFs.

* 1. AUDIT\_BUF\_SZ

This parameter specifies the size of the buffer used when auditing the database.

* 1. INSTANCE\_MEMORY

This parameter specifies the maximum amount of memory that can be allocated for a database partition.

V9.5以后，该参数控制实例使用的内存大小(包括数据库共享内存)。

* 1. SHEAPTHRES

This parameter is an instance-wide soft limit on the total amount of memory that can be consumed by private sorts at any given time.

When the total private sort memory consumption for an instance reaches this limit, the memory allocated for additional incoming private sort requests is considerably reduced.

控制私有排序内存大小，建议设置为0，排序内存由共享排序内存控制(SHEAPTHRES\_SHR)

* 1. DIR\_CACHE

This parameter determines whether the database, node and DCS directory files will be cached in memory.

默认值：YES

* 1. ASLHEAPSZ

The application support layer heap represents a communication buffer between the local application and its associated agent.

This buffer is allocated as shared memory by each database manager agent that is started.

* 1. RQRIOBLK

This parameter specifies the size of the communication buffer between remote applications and their database agents on the database server.

It is also used to determine the I/O block size at the Data Server Runtime Client when a blocking cursor is opened.

* 1. NUM\_POOLAGENTS、NUM\_INITAGENTS

NUM\_POOLAGENTS:

This parameter sets the maximum size of the idle agent pool.

默认值：100

NUM\_INITAGENTS:

This parameter determines the initial number of idle agents that are created in the agent pool at DB2START time.

默认值：0

* 1. MAX\_COORDAGENTS、MAX\_CONNECTIONS

MAX\_COORDAGENTS:

This parameter is used to limit the number of coordinating agents.

MAX\_CONNECTIONS:

This parameter indicates the maximum number of client connections allowed per database partition.

* 1. KEEPFENCED、FENCED\_POOL

KEEPFENCED:

This parameter indicates whether or not a fenced mode process is kept after a fenced mode routine call is complete.

Fenced mode processes are created as separate system entities in order to isolate user-written fenced mode code from the database manager agent process.

FENCED\_POOL:

This parameter represents the number of threads cached in each db2fmp process for threaded db2fmp processes (processes serving threadsafe stored procedures and UDFs).

For nonthreaded db2fmp processes, this parameter represents the number of processes cached.

* 1. SVCENAME

This parameter contains the name of the TCP⁄IP port which a database server will use to await communications from remote client nodes.

* 1. INTRA\_PARALLEL

控制是否打开分区内并行。默认不打开。

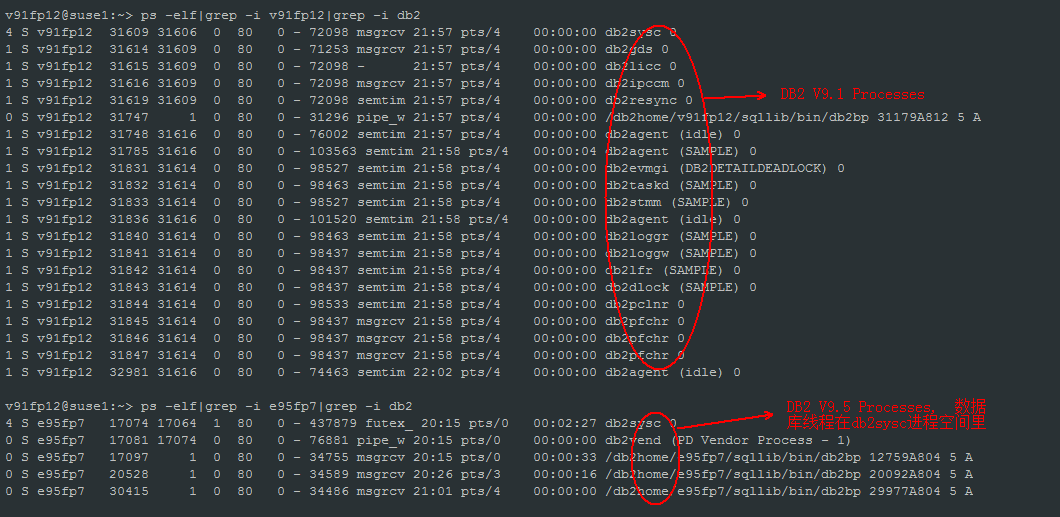
注意事项：

1. Parallel index creation does not use this configuration parameter.
2. If you change this parameter value, packages might be rebound to the database, and some performance degradation might occur.
   1. START\_STOP\_TIME

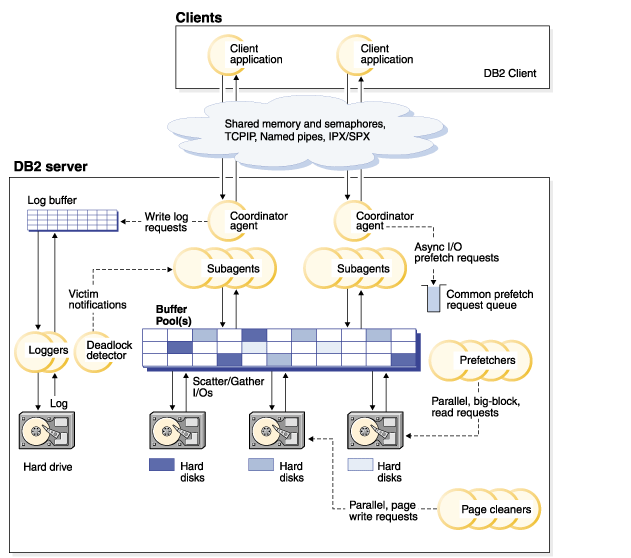
This parameter specifies the time, in minutes, within which all database partition servers must respond to a DB2START or a DB2STOP command.

1. **进程分析**
   1. 进程/线程

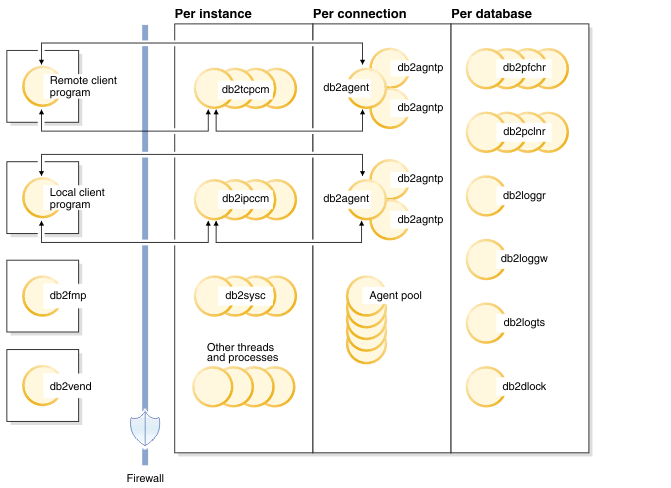
V9.5以前，DB2采用的是进程模型，V9.5开始，采用线程模型。



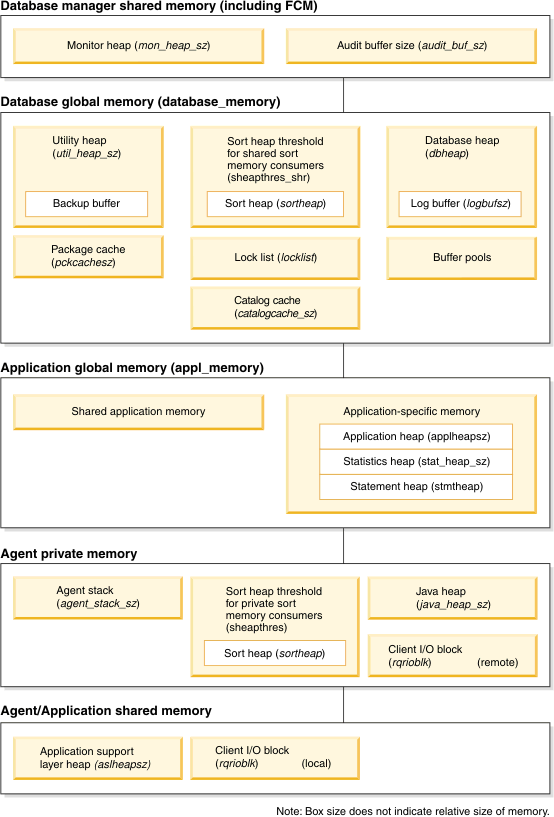
* 1. DB2 architecture and process overview



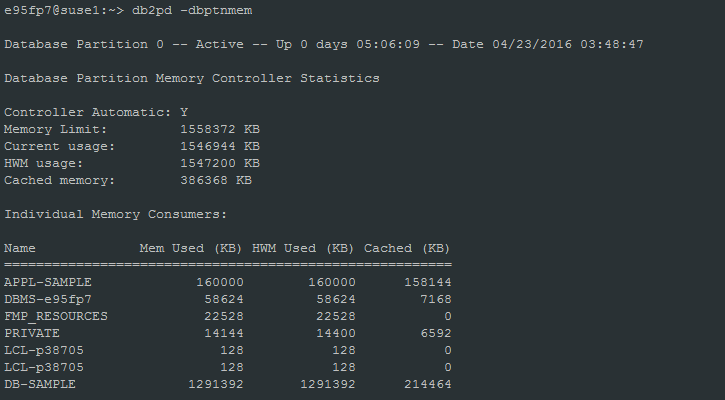
* 1. Process Model



1. **内存分析**
   1. **数据库内存模型**



* 1. **内存使用**





DB2V9.5以后, 数据库实例能使用的内存总和由instance\_memory参数控制。

Controller Automatic: Y 表示instance\_memory设置为automatic

Memory Limit: 1558372 KB -> instance\_memory 389593 \* 4KB

Current usage: 1546944 KB -> Individual Memory Consumers的SUM(Mem Used)

DB-SAMPLE: 1291392 -> database\_memory 322848 \* 4KB

以上内存表示已经分配给数据库的内存，并不代表内存已经使用。



Memory Sets: SAMPLE

Size(Kb) : 分配给database的共享内存(database\_memory)

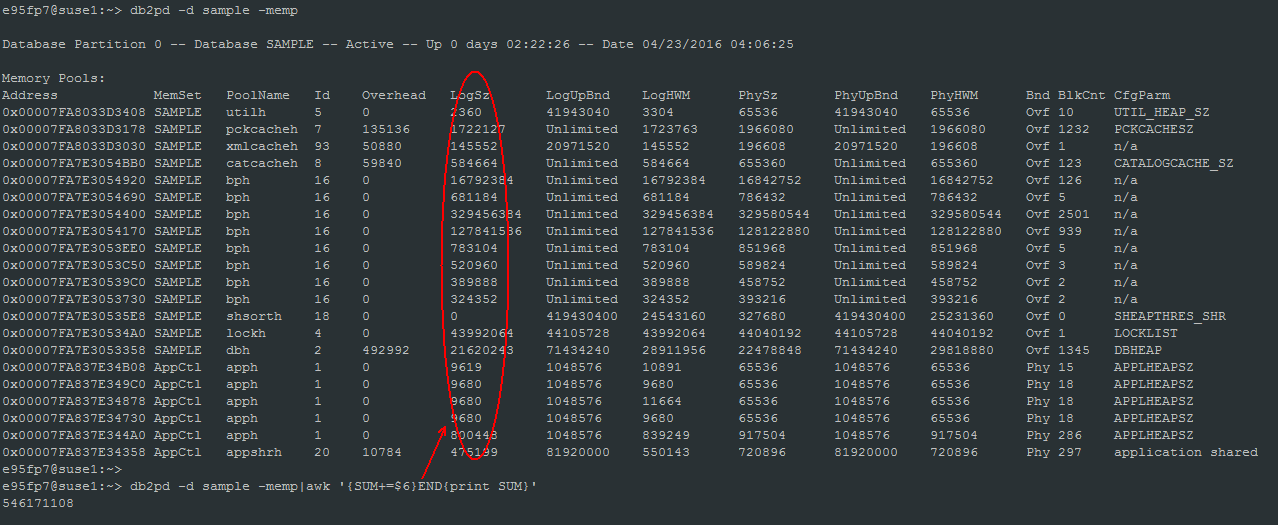
Used(Kb) : 正在使用的内存

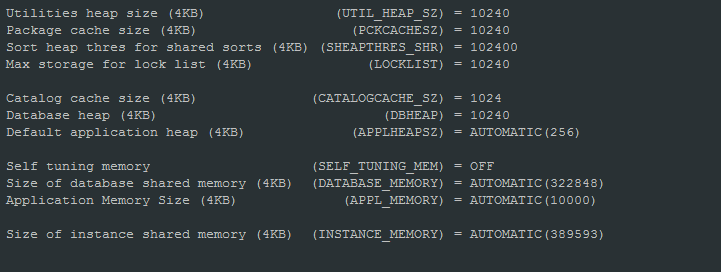
Cmt(Kb): 已经提交的内存(commited)

db\_mem\_thresh: This parameter represents the maximum percentage of committed, but currently unused, database shared memory that the database manager will allow before starting to release committed pages of memory back to the operating system.

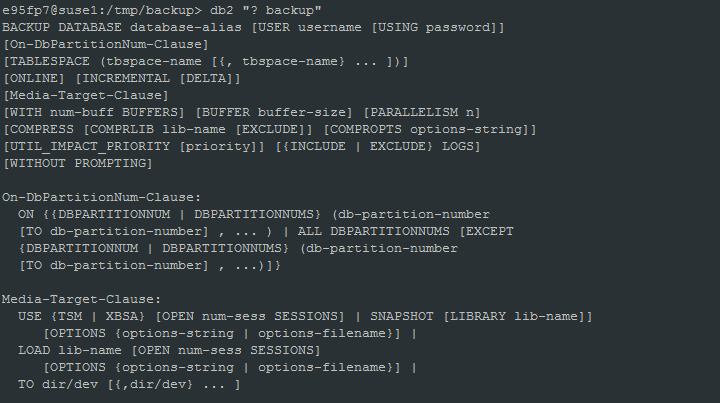


以下是各个内存池使用的内存大小：



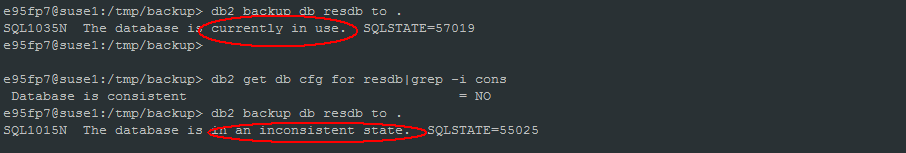


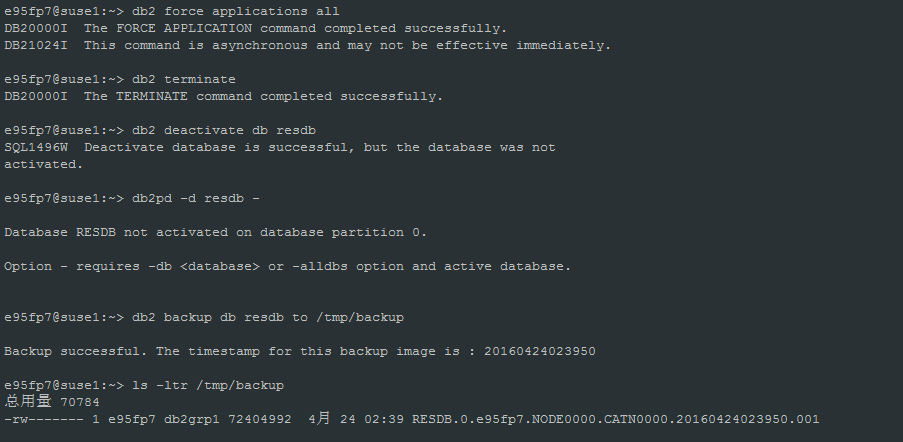
1. **备份恢复**
   1. **备份**
      1. **备份命令**



* + 1. **脱机备份**

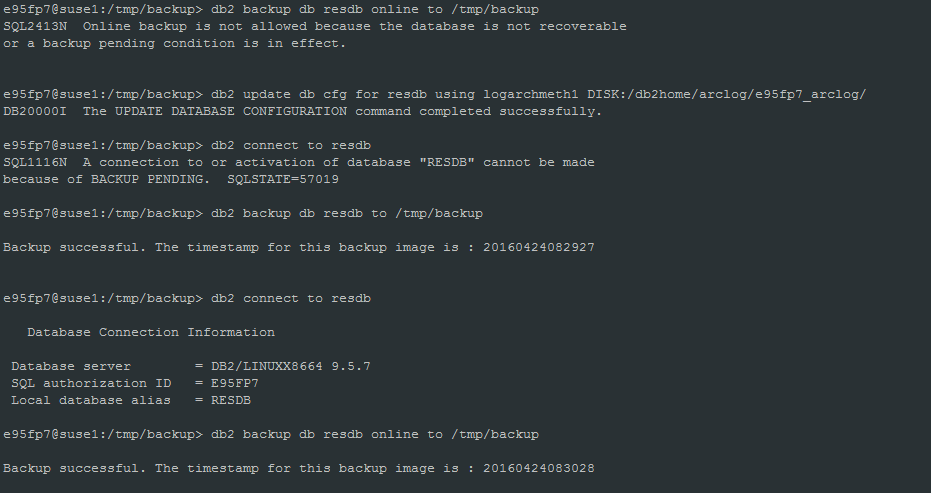
脱机备份需要数据库处于不激活、一致性的状态。



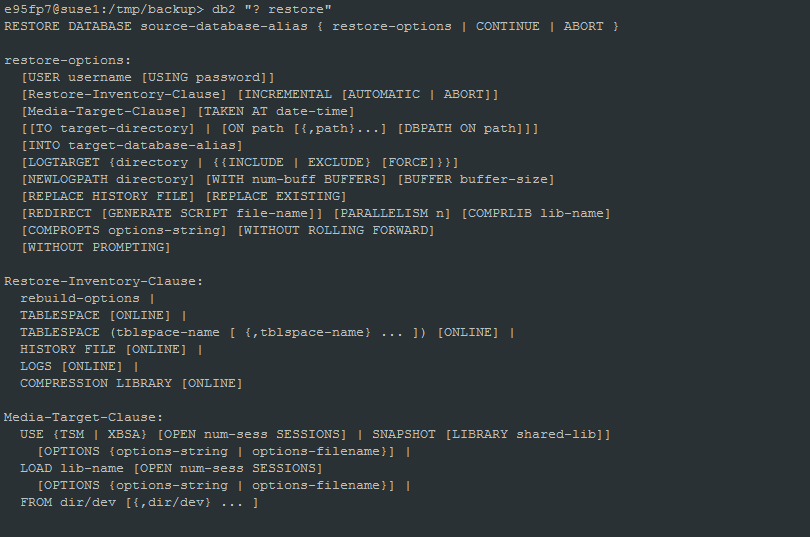


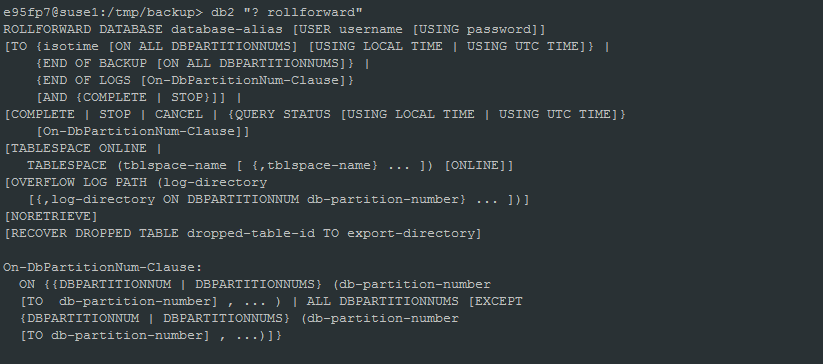
* + 1. **在线备份**

归档日志模式才支持在线备份，设置归档路径logarchmeth1，设置后，需要脱机备份一次数据库。



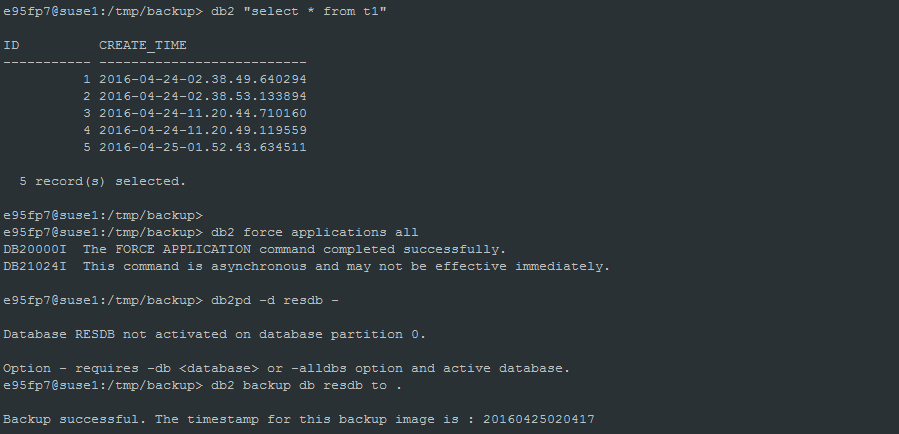
* 1. **恢复**
     1. **恢复命令**

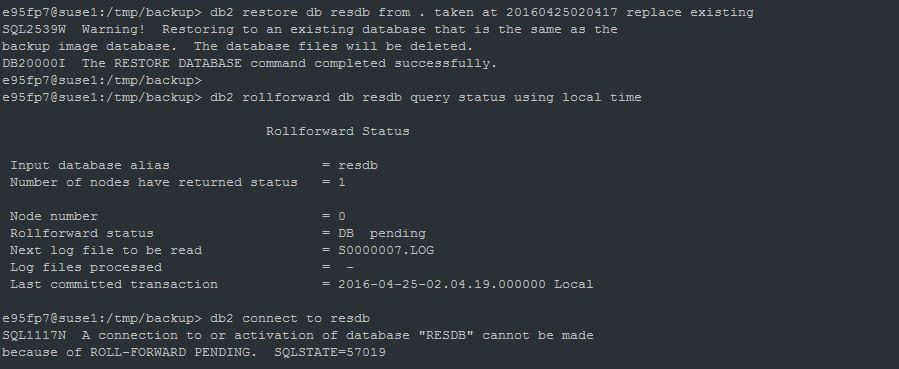


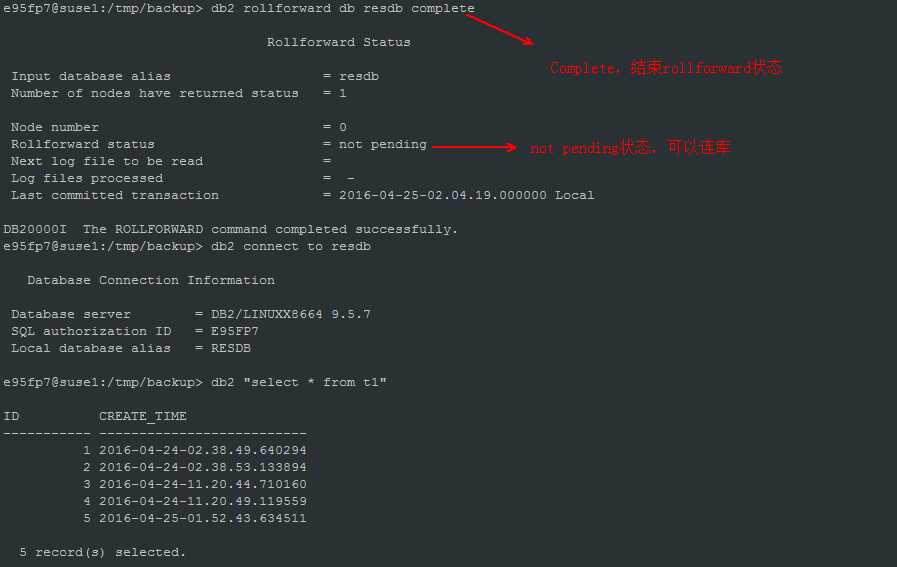


* + 1. **恢复**

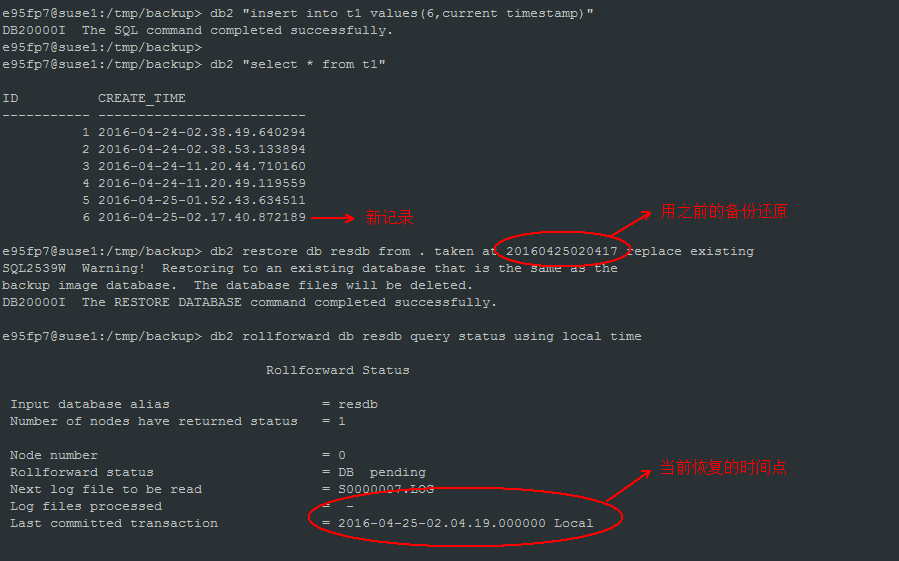
1. **一致性恢复(通过脱机备份恢复)**

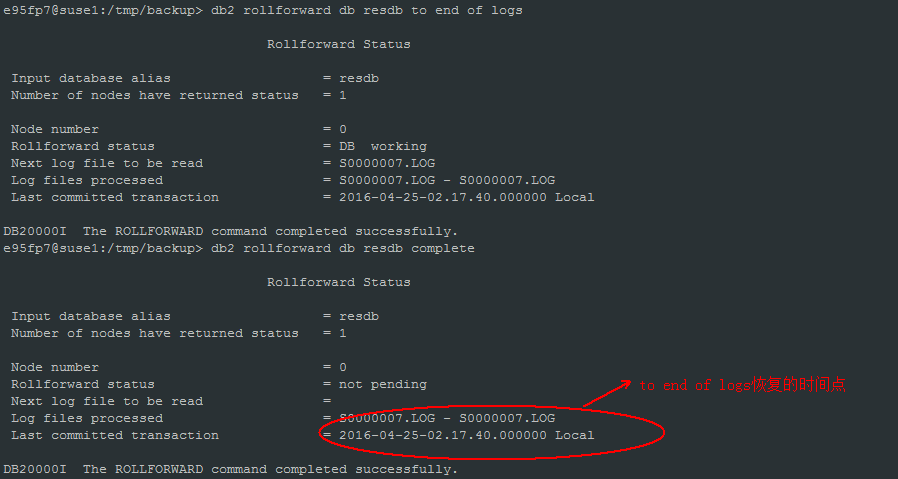


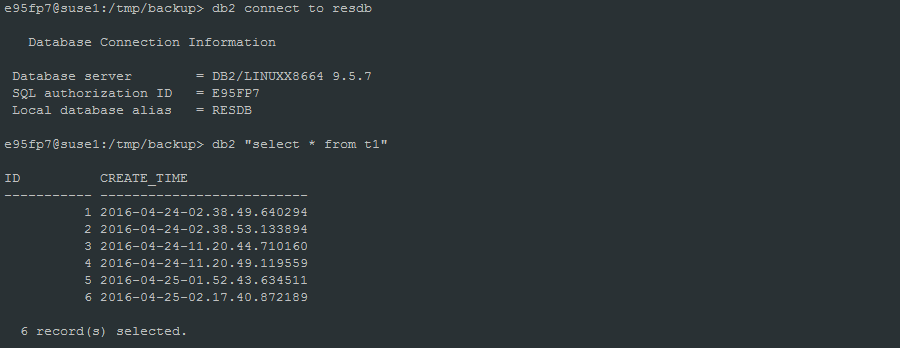




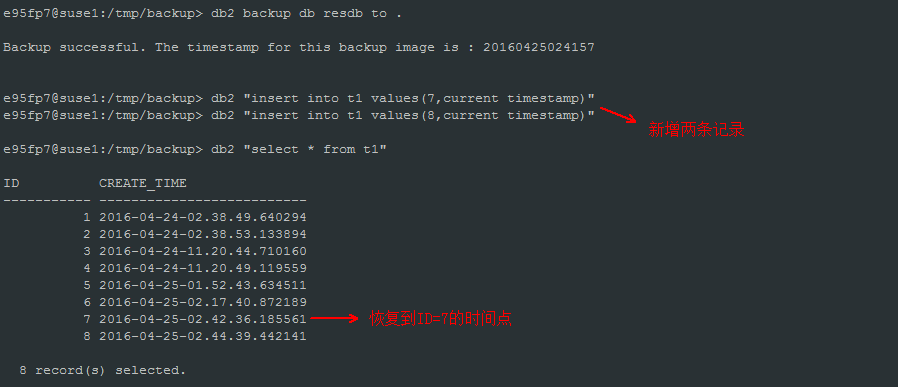
1. **恢复到end of logs(恢复到数据库当前时间点)**

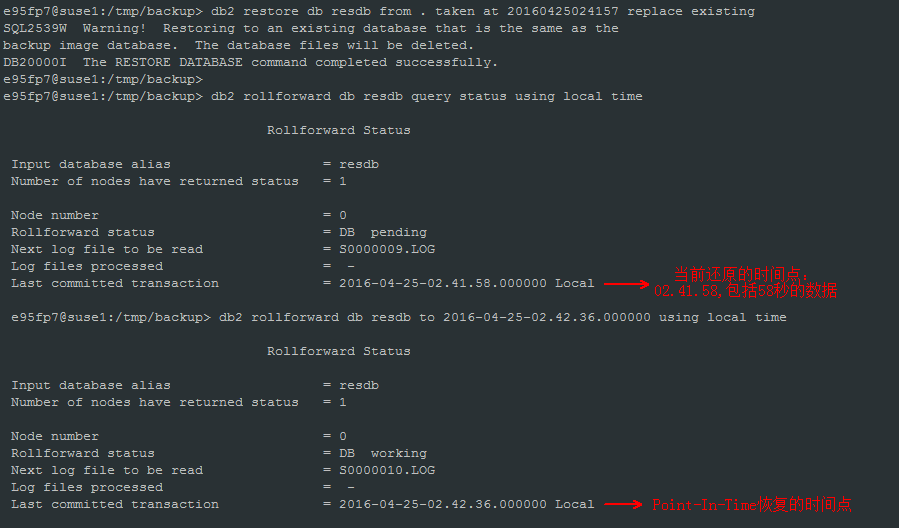


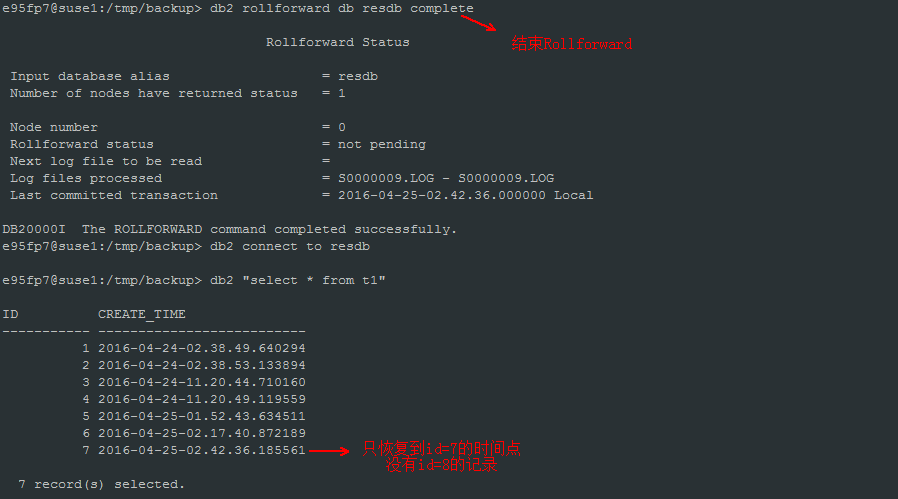




1. **恢复到某个时间点**



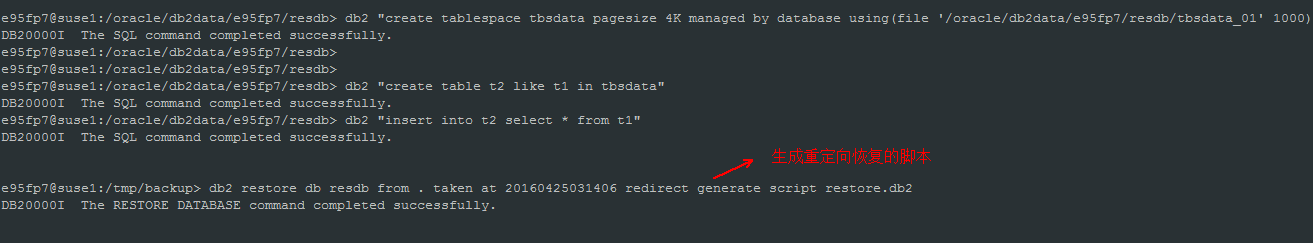


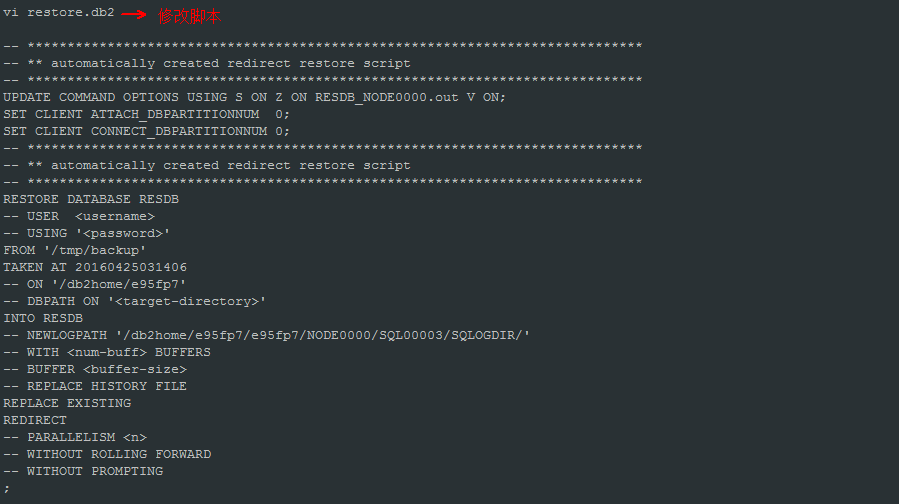


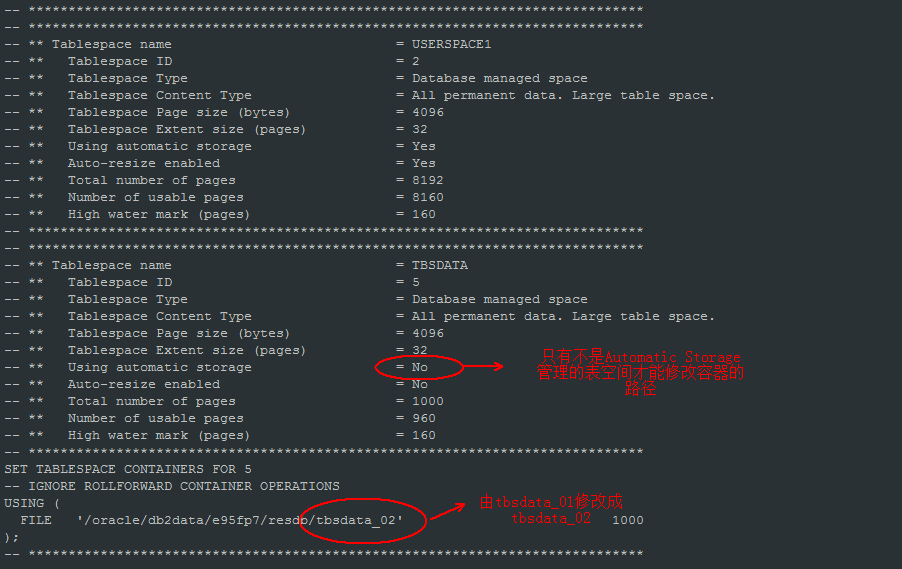
1. **重定向恢复**

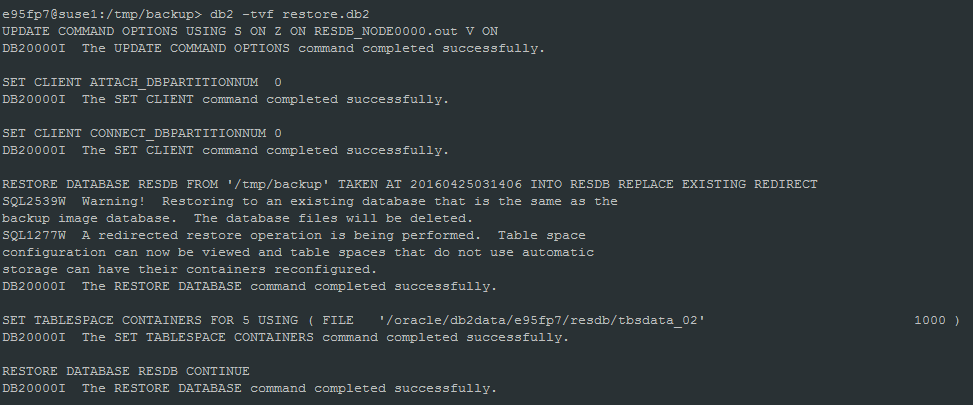
A redirected restore operation is performed when one of the following situations occur:

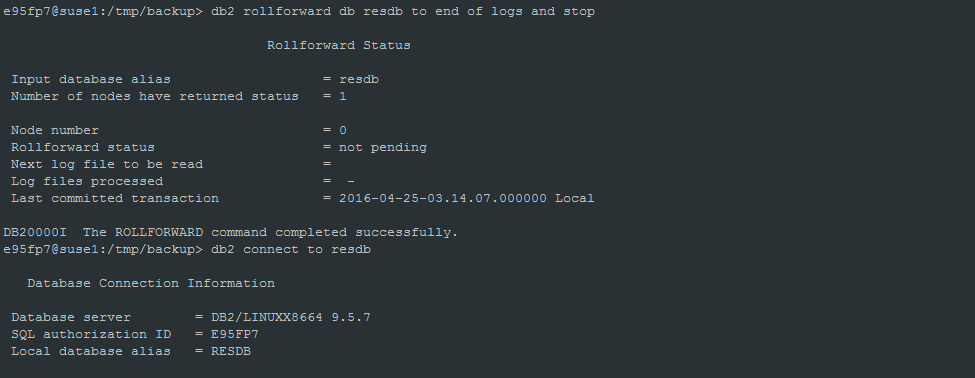
1. You want to restore a backup image to a target machine that is different than the source machine
2. You want to restore your table space containers into a different physical location
3. Your restore operation has failed because one or more containers is inaccessible











* 1. **备份恢复限制**

1. 跨平台恢复限制(分三种平台: Big-endian Linux and UNIX、Little-endian Linux and UNIX、Windows), 不能跨平台恢复
2. 在线备份不能跨版本恢复(比如V9.5的在线备份不能恢复到V9.7版本上)
3. 脱机备份只能向高版本恢复，不能向低版本恢复。(如V9.5可以恢复到V9.7, 但是V9.7不能恢复到V9.5)
4. 脱机备份最多跨两个版本恢复(V9.5 -> V10.1, 中间两个版本V9.7、V9.8 如果V9.5要升级到V10.5, 需先升级到V9.7再升级到V10.5)
5. 备份恢复不能降低表空间高水位
6. SMS表空间不能恢复到DMS表空间
7. Automatic Storage管理的表空间不能通过set tablespace containers修改容器路径
8. **补丁版本升级**

介质下载地址： <http://www-01.ibm.com/support/docview.wss?uid=swg27007053>

* 1. **补丁升级**

补丁升级指Fixpack升级，比如V9.5 FP7 升级到V9.5 FP9。

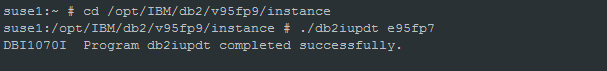
有两种升级方式：

1. installFixPak直接在原来的安装目录上升级产品
2. 在不同的路径安装fixpack，通过db2iupdt升级。优点是需要回退的时候，不需要再安装旧的fixpack产品(因为没有被覆盖)。

下面例子采用第二种方式，一般生产系统都采用这种方式。

例子： V9.5FP7 升级到V9.5 FP9

db2iupdt <instname> ->需要升级的实例名称



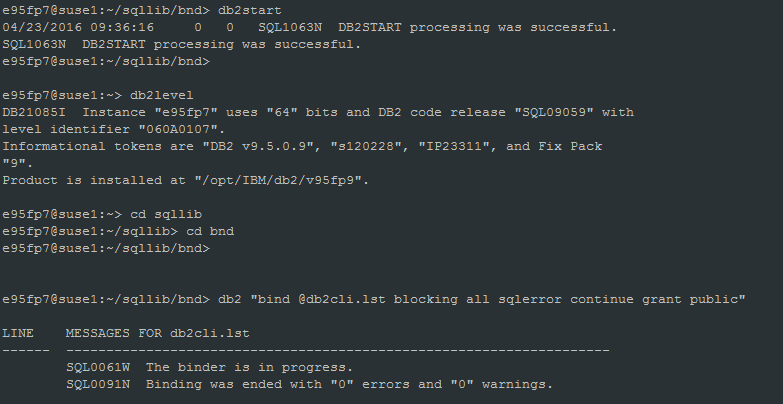
db2iupdt后，需要绑包：

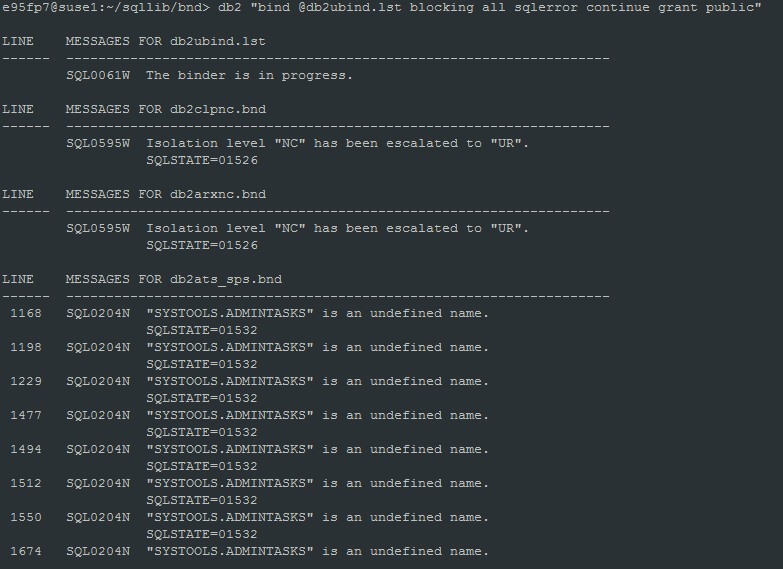
cd INSTHOME\sqllib\bnd

db2 "bind @db2cli.lst blocking all sqlerror continue grant public"

db2 "bind @db2ubind.lst blocking all sqlerror continue grant public"

db2 "bind db2schema.bnd blocking all sqlerror continue grant public"







可选操作：

After you install a fix pack, some packages are marked as invalid. Packages marked as invalid are implicitly rebound the first time an application uses them. To eliminate this overhead and to ensure that the rebind is successful, manually rebind all packages. For example, issue the db2rbind command:

db2rbind <dbname> -l logfile all

补丁升级可以回退：通过 –D选项

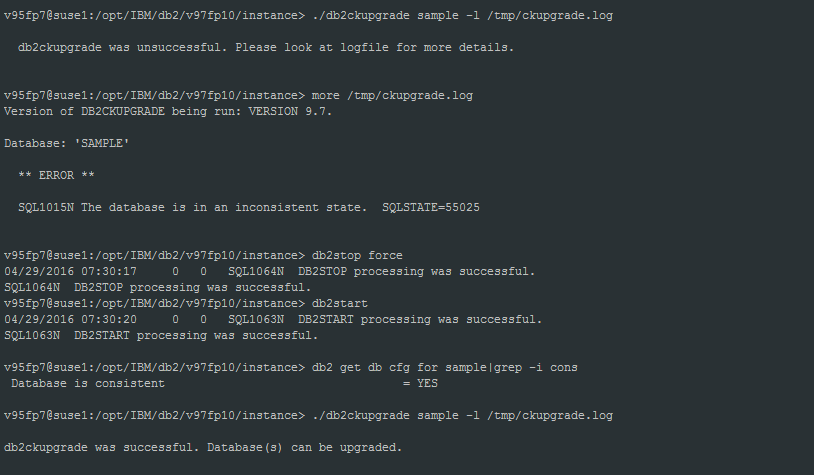


* 1. **版本升级**

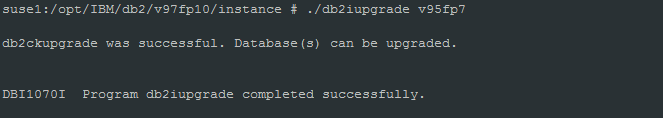
例子：V9.5 升级到 V9.7

* + 1. 先升级实例

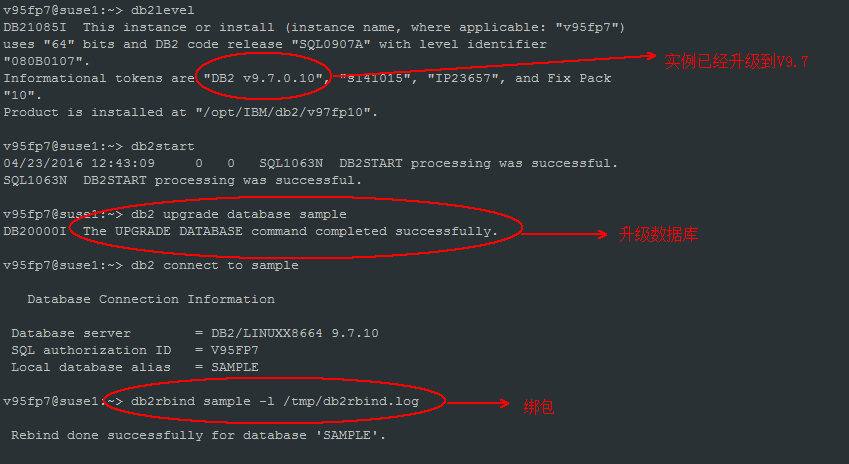
**建议先通过db2ckupgrade检查数据库是否能升级：**



如果数据库能升级，再升级实例。

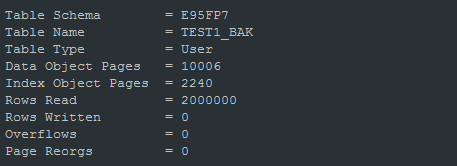


* + 1. 再升级数据库



1. **监控**
   1. **快照及指标分析**
2. **Table snapshot**

**db2 get snapshot for tables on <dbname>**



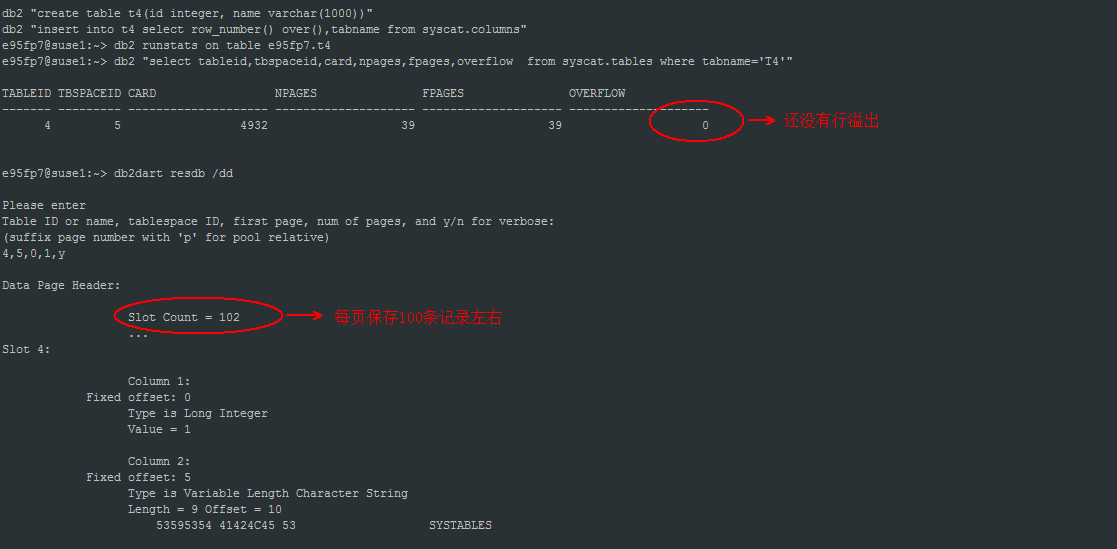
**Data Object Pages: 数据页数量**

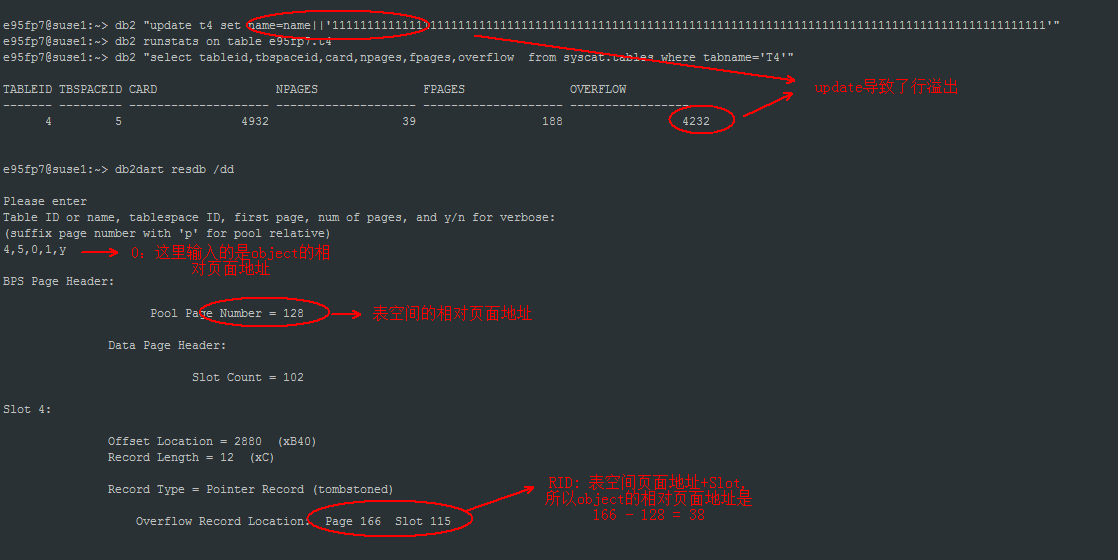
**Index Object Pages: 索引页数量**

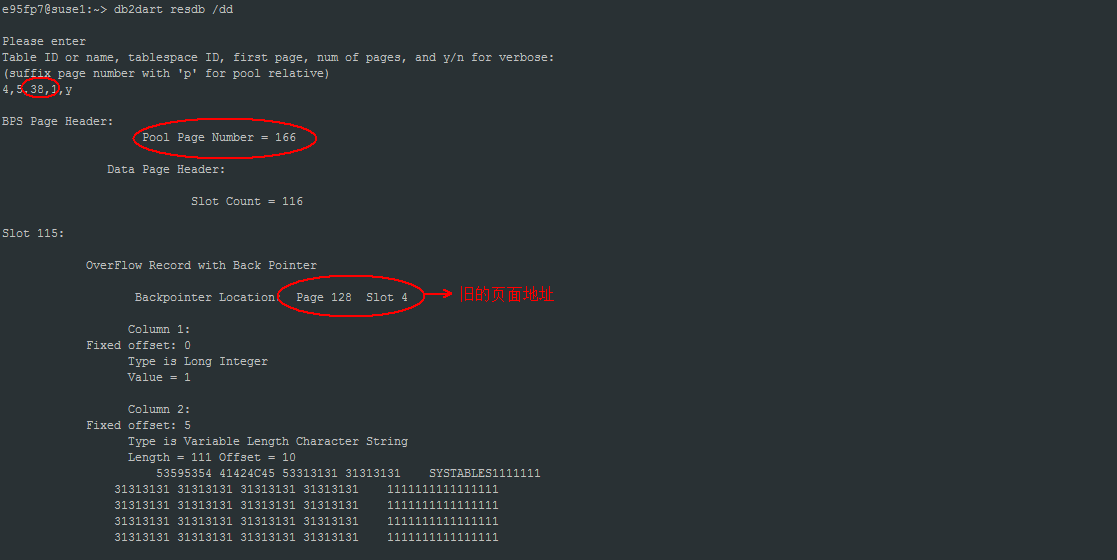
**Rows Read: 查询的行记录数，值越大，表示表扫描IO很多，需要检查是否没有建合适的索引。**

**Rows Written: 写表的行记录数，包括增删改。**

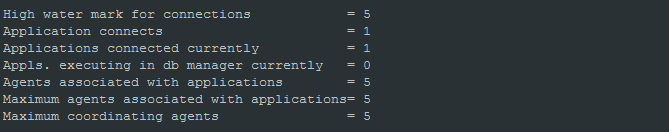
**Overflows:表示访问overflowed 的行记录数。如果该值很大，表示表的行溢出比较严重，需要通过reorg重组表。A row overflows if it is updated and no longer fits in the data page where it was originally written. This usually happens as a result of an update of a VARCHAR or an ALTER TABLE statement.**







1. **Database snapshot**



**High water mark for connections:** The highest number of simultaneous connections to the database since the database was activated.

**Applicatoin connects:** Indicates the number of connections to the database since the first connect, activate, or last reset.

**Applicatoins connected currently:** Indicates the number of applications that are currently connected to the database.

**Appls. Executing in db manager currently:** Indicates the number of applications that are currently connected to the database, and for which the database manager is currently processing a request.

**Agents associated with applications:** At the application level, this is the number of subagents associated with an application. At the database level, it is the number of subagents for all applications.

**Maximum agents associated with applications:** At the activity level, this is the maximum number of agents that were used when executing the statement. At the database level, it is the maximum number of agents for all applications.

**Maximum coordinating agents:**  The maximum number of coordinating agents working at one time.



**Locks held currently:** The number of locks currently held.

**Lock waits:** The total number of times that applications or connections waited for locks.

**Time database waited on locks(ms):** The total elapsed time spent waiting for locks. The value is given in milliseconds.

**Lock list memory in use (Bytes):** The total amount of lock list memory (in bytes) that is in use.

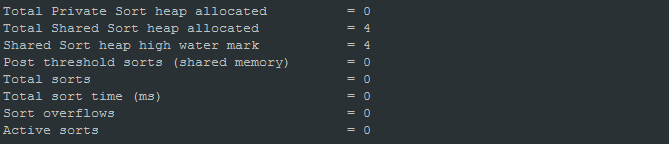
**Deadlocks detected:** The total number of deadlocks that have occurred.

**Lock escalations:** The number of times that locks have been escalated from several row locks to a table lock.

**Agents currently waiting on locks:** Indicates the number of agents waiting on a lock.

**Lock Timeouts:** The number of times that a request to lock an object timed out instead of being granted.

**Number of indoubt transactions:** The number of outstanding indoubt transactions in the database.



**Total Private Sort heap allocated:** The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.

**Total Shared Sort heap allocated:** Total amount of shared sort memory allocated in the database.

**Shared sort heap high water mark:** Database-wide shared sort memory high watermark in 4 KB pages.

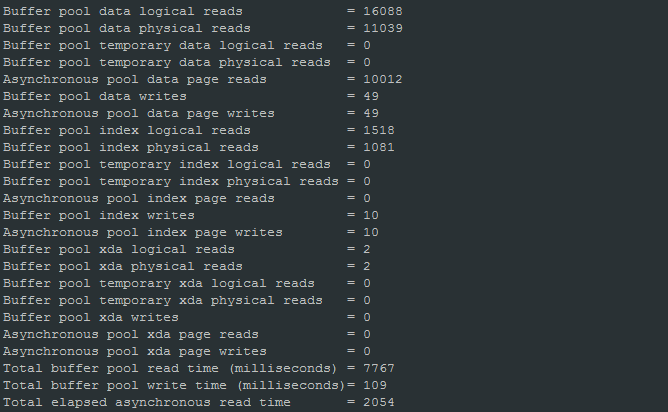
**Post threshold sorts(shared memory):** The total number of sorts that were throttled back by the sort memory throttling algorithm. A throttled sort is a sort that was granted less memory than requested by the sort memory manager.

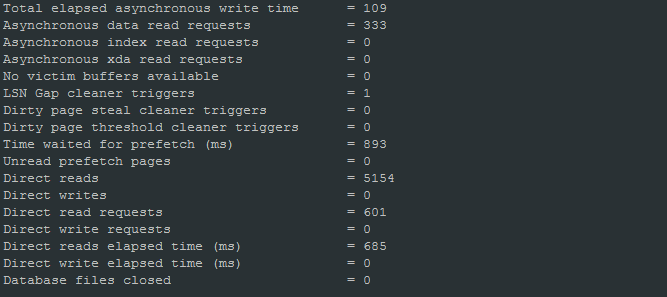
**Total sorts:** The total number of sorts that have been executed.

**Total sort time(ms):** The total elapsed time for all sorts that have been executed. This value is reported in milliseconds.

**Sort overflows:** The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.

**Active sorts:** The number of sorts in the database that currently have a sort heap allocated.





**Buffer pool data logical reads:** The number of data pages which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool data physical reads:** Indicates the number of data pages read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary data logical reads:** Indicates the number of data pages which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary data physical reads:** Indicates the number of data pages read in from the table space containers (physical) for temporary table spaces.

**Asynchronous pool data page reads:** Indicates the number of data pages read in from the table space containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of table spaces.

**Buffer pool data writes:** The number of times a buffer pool data page was physically written to disk.

**Asynchronous pool data page writes:** The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner, or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.

**Buffer pool index logical reads:** Indicates the number of index pages which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool index physical reads:** Indicates the number of index pages read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary index logical reads:** Indicates the number of index pages which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary index physical reads:** Indicates the number of index pages read in from the table space containers (physical) for temporary table spaces.

**Asynchronous pool index page reads:** Indicates the number of index pages read in from the table space containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of table spaces.

**Buffer pool index writes:** Indicates the number of times a buffer pool index page was physically written to disk.

**Asynchronous pool index page writes:** The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner, or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.

**Buffer pool xda logical reads:** Indicates the number of data pages for XML storage objects (XDAs) which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool xda physical reads:** Indicates the number of data pages for XML storage objects (XDAs) read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary xda logical reads:** Indicates the number of pages for XML storage object (XDA) data which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary xda physical reads:** Indicates the number of pages for XML storage object (XDA) data read in from the table space containers (physical) for temporary table spaces.

**Buffer pool xda writes:** Indicates the number of times a buffer pool data page for an XML storage object (XDA) was physically written to disk.

**Asynchronous pool xda page reads:** Indicates the number of XML storage object (XDA) data pages read in from the table space containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of table spaces.

**Asynchronous pool xda page writes:** The number of times a buffer pool data page for an XML storage object (XDA) was physically written to disk by either an asynchronous page cleaner, or a prefetcher. A prefetcher may have written dirty pages to disk to make space for the pages being prefetched.

**Total buffer pool read time:** Indicates the total amount of time spent reading in data and index pages from the table space containers (physical) for all types of table spaces. This value is given in milliseconds.

**Total buffer pool write time:** Provides the total amount of time spent physically writing data or index pages from the buffer pool to disk. Elapsed time is given in milliseconds.

**Total elapsed asynchronous read time:** Indicates the total amount of time spent reading in data and index pages from the table space containers (physical) by asynchronous engine dispatchable units (EDUs) for all types of table spaces. This value is given in milliseconds.

**Total elapsed asynchronous write time:** The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners. This value is reported in milliseconds.

**Asynchronous data read requests:** The number of asynchronous read requests by the prefetcher to the operating system. These requests are typically large block I/Os of multiple pages.

**Asynchronous index read requests:** The number of asynchronous read requests for index pages.

**Asynchronous xda read requests:** The number of asynchronous read requests for XML storage object (XDA) data.

**No victim buffers available:** Number of times an agent did not have a preselected victim buffer available.

**LSN Gap cleaner triggers:** The number of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database.

**Dirty page steal cleaner triggers:** The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.

**Dirty page threshold cleaner triggers:** The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.

**Time waited for prefetch (ms):** The time an application spent waiting for an I/O server (prefetcher) to finish loading pages into the buffer pool. The value is given in milliseconds.

**Unread prefetch pages:** Indicates the number of pages that the prefetcher read into the bufferpool that were never used.

**Direct reads:** The number of read operations that do not use the buffer pool.

**Direct writes:** The number of write operations that do not use the buffer pool.

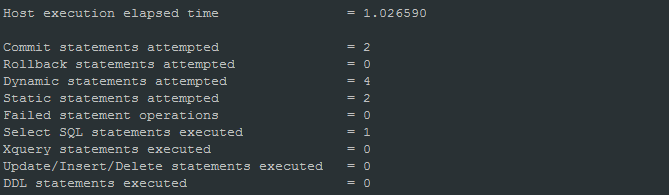
**Direct read requests:** The number of requests to perform a direct read of one or more sectors of data.

**Direct write requests:** The number of requests to perform a direct write of one or more sectors of data.

**Direct reads elapsed time (ms):** The elapsed time required to perform the direct reads. This value is given in milliseconds.

**Direct write elapsed time (ms):** The elapsed time required to perform the direct writes. This value is reported in milliseconds.

**Database files closed:** The total number of database files closed.



**Host execution elapsed time:** This value represents the sum of the host execution times for all the statements that were executed for a particular database.

**Commit statements attempted:** The total number of SQL COMMIT statements that have been attempted.

**Rollback statements attempted:** The total number of SQL ROLLBACK statements that have been attempted.

**Dynamic statements attempted:** The number of dynamic SQL statements that were attempted.

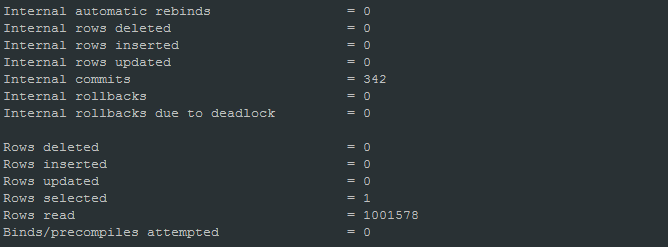
**Static statements attempted:** The number of static SQL statements that were attempted.

**Failed statement operations:** The number of SQL statements that were attempted, but failed.

**Select SQL statements executed:** The number of SQL SELECT statements that were executed.

**Update/Insert/Delete statements executed:** The number of UPDATE, INSERT, MERGE and DELETE statements that were executed.

**DDL statements executed:** This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.



**Internal automatic rebinds:** The number of automatic rebinds (or recompiles) that have been attempted.

**Internal rows deleted:** This is the number of rows deleted from the database as a result of internal activity.

**Internal rows inserted:** The number of rows inserted into the database as a result of internal activity caused by triggers.

**Internal rows updated:** This is the number of rows updated from the database as a result of internal activity.

**Internal commits:** The total number of commits initiated internally by the database manager.

**Internal rollbacks:** The total number of rollbacks initiated internally by the database manager.

**Internal rollbacks due to deadlock:** The total number of forced rollbacks initiated by the database manager due to a deadlock.

**Rows deleted:** This is the number of row deletions attempted.

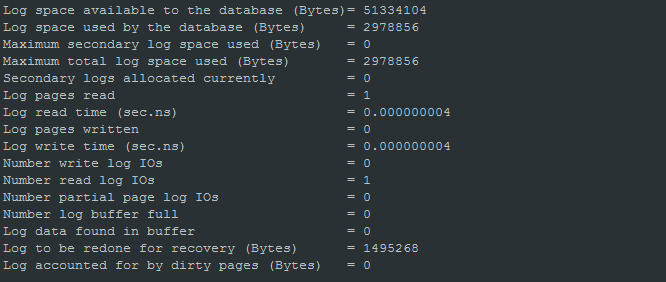
**Rows inserted:** The number of row insertions attempted.

**Rows updated:** This is the number of row updates attempted.

**Rows selected:** This is the number of rows that have been selected and returned to the application.

**Rows read:** The number of rows read from the table.

**Binds/precompiles attempted:** The number of binds and pre-compiles attempted.



**Log space available to the database (Bytes):** The amount of active log space in the database that is not being used by uncommitted transactions (in bytes).

**Log space used by the database (Bytes):** The total amount of active log space currently used (in bytes) in the database.

**Maximum secondary log space used (Bytes):** The maximum amount of secondary log space used (in bytes).

**Maximum total log space used (Bytes):** The maximum amount of total log space used (in bytes).

**Secondary logs allocated currently:** The total number of secondary log files that are currently being used for the database.

**Log pages read:** The number of log pages read from disk by the logger.

**Log read time (sec.ns):** The total elapsed time spent by the logger reading log data from the disk.

**Log pages written:** The number of log pages written to disk by the logger.

**Log write time (sec.ns):** The total elapsed time spent by the logger writing log data to the disk.

**Number write log IOs:** The number of I/O requests issued by the logger for writing log data to the disk.

**Number read log IOs:** The number of I/O requests issued by the logger for reading log data from the disk.

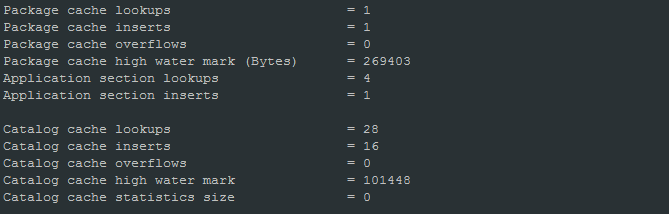
**Number partial page log IOs:** The number of I/O requests issued by the logger for writing partial log data to the disk.

**Number log buffer full:** The num\_log\_buffer\_full element stores the number of times agents have to wait for log data to write to disk while copying log records into the log buffer. This value is incremented per agent per incident. For example, if two agents attempt to copy log data while the buffer is full, then this value is incremented by two.

**Log data found in buffer:** The number of times an agent reads log data from the buffer. Reading log data from the buffer is preferable to reading from the disk because the latter is slower.

**Log to be redone for recovery (Bytes):** The amount of log (in bytes) that will have to be redone for crash recovery.

**Log accounted for by dirty pages (Bytes):** The amount of log (in bytes) corresponding to the difference between the oldest dirty page in the database and the top of the active log.



**Package cache lookups:** The number of times that an application looked for a section or package in the package cache.

**Package cache inserts:** The total number of times that a requested section was not available for use and had to be loaded into the package cache.

**Package cache overflows:** The number of times that the package cache overflowed the bounds of its allocated memory.

**Package cache high water mark (Bytes):** The largest size reached by the package cache. Note: The monitor element is deprecated starting with DB2 Version 9.5. Using this monitor element will not generate an error. However, it does not return a valid value. This monitor element is no longer recommended and might be removed in a future release.

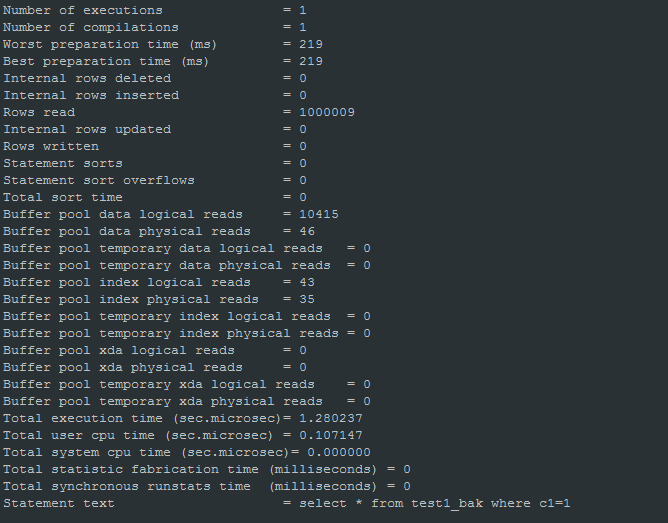
**Catalog cache lookups:** The number of times that the catalog cache was referenced to obtain table descriptor information or authorization information.

**Catalog cache inserts:** The number of times that the system tried to insert table descriptor or authorization information into the catalog cache.

**Catalog cache overflows:** The number of times that the catalog cache overflowed the bounds of its allocated memory.

**Catalog cache high water mark:** The largest logical size reached by the catalog cache.

1. **Dynamic sql snapshot**



**Number of executions:** The number of times that an SQL statement has been executed.

**Number of compilations:** The number of different compilations for a specific SQL statement.

**Worst preparation time (ms):** The longest amount of time in milliseconds that was required to prepare a specific SQL statement.

**Best preparation time (ms):** The shortest amount of time in milliseconds that was required to prepare a specific SQL statement.

**Internal rows deleted:** This is the number of rows deleted from the database as a result of internal activity.

**Internal rows inserted:** The number of rows inserted into the database as a result of internal activity caused by triggers.

**Rows read:** The number of rows read from the table.

**Internal rows updated:** This is the number of rows updated from the database as a result of internal activity.

**Rows written:** This is the number of rows changed (inserted, deleted or updated) in the table.

**Statement sorts:** The total number of times that a set of data was sorted in order to process the stmt\_operation.

**Statement sort overflows:** The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.

**Total sort time:** The total elapsed time for all sorts that have been executed. This value is reported in milliseconds.

**Buffer pool data logical reads:** The number of data pages which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool data physical reads:** Indicates the number of data pages read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary data logical reads:** Indicates the number of data pages which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary data physical reads:** Indicates the number of data pages read in from the table space containers (physical) for temporary table spaces.

**Buffer pool index logical reads:** Indicates the number of index pages which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool index physical reads:** Indicates the number of index pages read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary index logical reads:** Indicates the number of index pages which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary index physical reads:** Indicates the number of index pages read in from the table space containers (physical) for temporary table spaces.

**Buffer pool xda logical reads:** Indicates the number of data pages for XML storage objects (XDAs) which have been requested from the buffer pool (logical) for regular and large table spaces.

**Buffer pool xda physical reads:** Indicates the number of data pages for XML storage objects (XDAs) read in from the table space containers (physical) for regular and large table spaces.

**Buffer pool temporary xda logical reads:** Indicates the number of pages for XML storage object (XDA) data which have been requested from the buffer pool (logical) for temporary table spaces.

**Buffer pool temporary xda physical reads:** Indicates the number of pages for XML storage object (XDA) data read in from the table space containers (physical) for temporary table spaces.

**Total execution time:** The total time in seconds and microseconds that was spent executing a particular statement in the SQL cache.

**Total user cpu time:** The total user CPU time for an SQL statement.

**Total system cpu time:** The total system CPU time for an SQL statement.

1. **性能及其它开发问题**
   1. **JDBC驱动**
      1. **JDBC 驱动类型**

Type 1 ：

Drivers that implement the JDBC API as a mapping to another data access API, such as Open Database Connectivity (ODBC). Drivers of this type are generally dependent on a native library that limits their portability. The DB2 database system does not provide a type 1 driver.

Type 2 :

Drivers that are written partly in the Java™ programming language and partly in native code. The drivers use a native client library specific to the data source to which they connect. Because of the native code, their portability is limited.

驱动类: COM.ibm.db2.jdbc.app.DB2Driver, 位于包 db2java.zip中。

JDK必须能访问db2的native library(如Linux: libdb2jdbc.so.1)。

Type 3 :

Drivers that use a pure Java client and communicate with a data server using a data-server-independent protocol. The data server then communicates the client's requests to the data source. The DB2 database system does not provide a type 3 driver.

驱动:COM.ibm.db2.jdbc.net.DB2Driver, 位于包 db2java.zip中。 通过独立于数据库的协议和中间件服务器通信(Jd), 中间件服务器再将客户端请求传给数据源。

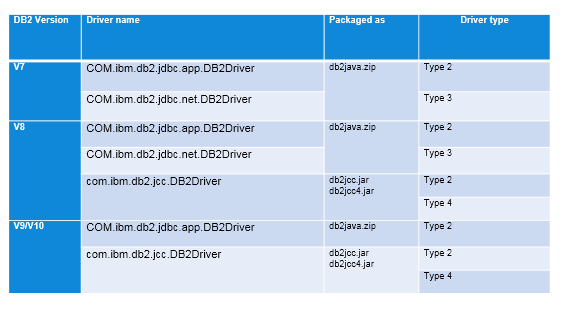
对 DB2 JDBC type 2 driver 的调用被转换成 Java native method。 需要安装DB2 客户端, JDBC 请求通过客户端转到 DB2 服务器。

Type 4 :

Drivers that are pure Java and implement the network protocol for a specific data source. The client connects directly to the data source.

纯JAVA驱动:com.ibm.db2.jcc.DB2Driver, 位于包db2jcc.jar中。

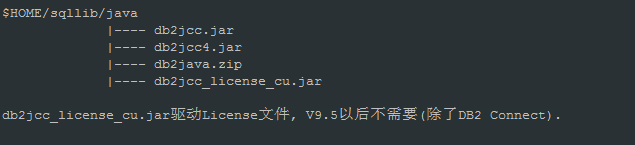
* + 1. **DB2支持的驱动类型**



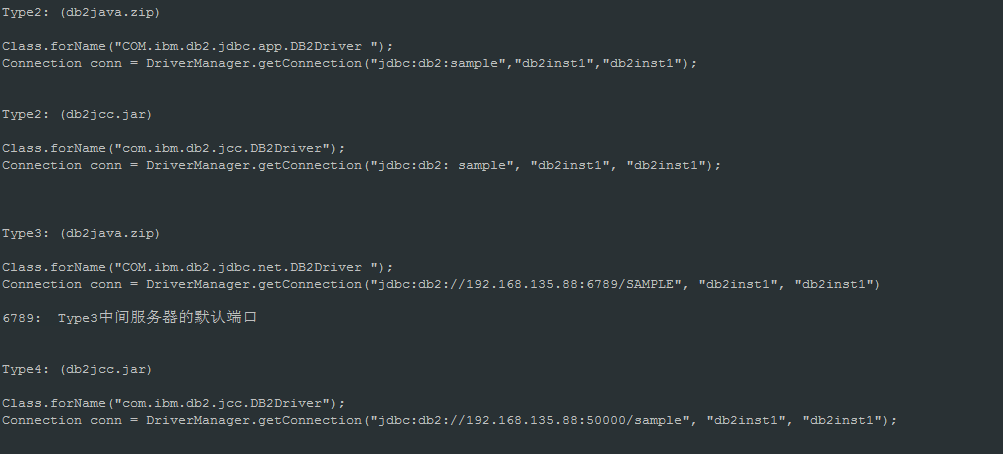
*db2jcc.jar for JDBC 3.0 support，db2jcc4.jar for JDBC 4.0 support*

*IBM Data Server Driver for JDBC and SQLJ (type 2 and type 4): db2jcc.jar, db2jcc4.jar, sqlj.zip, sqlj4.zip*

*Legancy JDBC Driver: db2java.zip, Only supports Type 2、Type 3*



* + 1. **驱动URL**

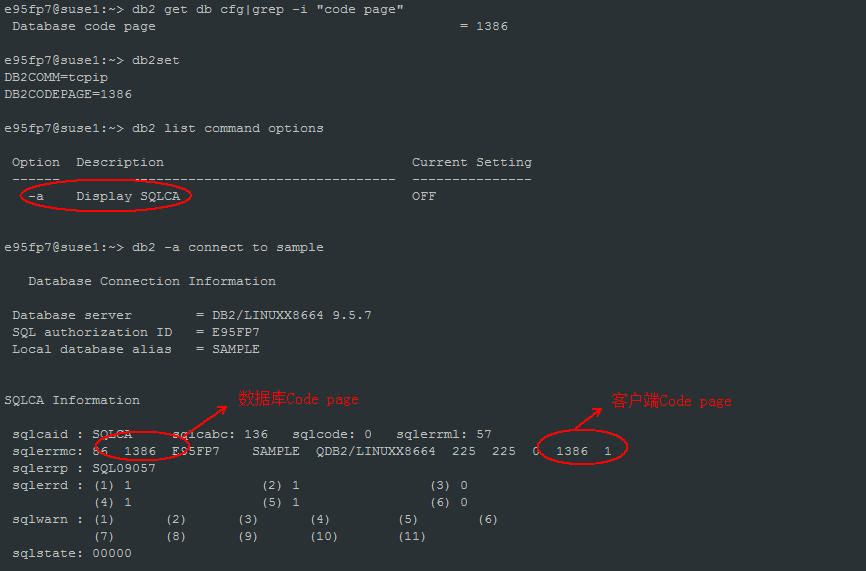


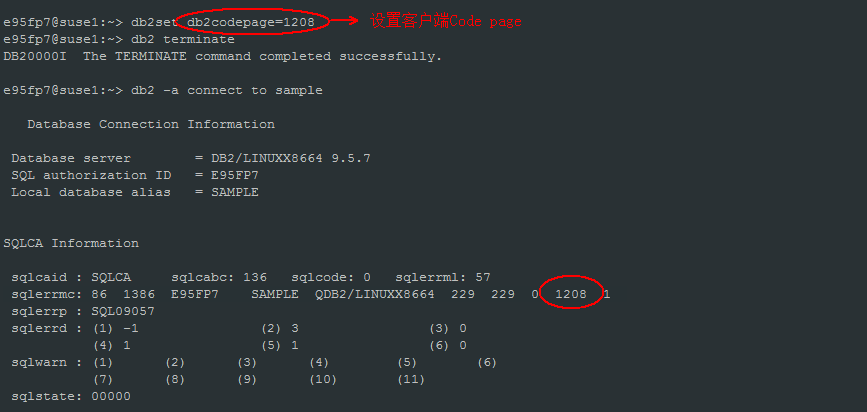
* 1. **数据库代码页问题(codepage)**
     1. **Code page**

1. **Data code page**
2. **Applicatoin code page**
3. **Database code page**



* + 1. **查看Application代码页**



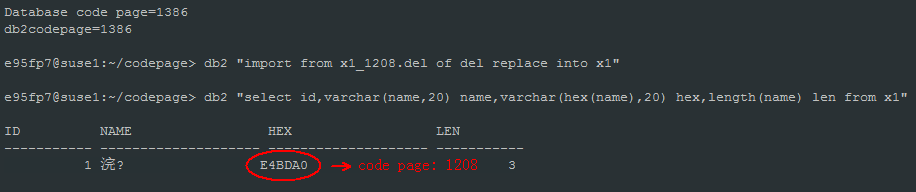


* + 1. **LOAD与Import不同的转换方式(del格式)**

缺 省情况下，LOAD 认为输入文件是用数据库代码页编码的，直接将文件转化为数据库 codepage 编码。如果输入文件不是以数据库 codepage 编码的，可以通过 codepage 修饰符来导入正确文件。而缺省情况下 DB2 IMPORT 实用程序认为输入文件中的数据是用当前系统的代码页编码的。当将数据文件导入到数据库时，DB2 会自动将数据文件从当前系统代码页转换成数据库代码页。如果输入文件不是当前系统的代码页编码的。也可以通过 codepage 修饰符来轻松导入正确文件。

所以一般来说，LOAD 直接把源数据的 codepage 转化为数据库的 codepage，而 IMPORT 则先将源数据的 codepage 转换为应用程序的 codepage，然后将应用程序的 codepage 再转换为数据库的 codepage。

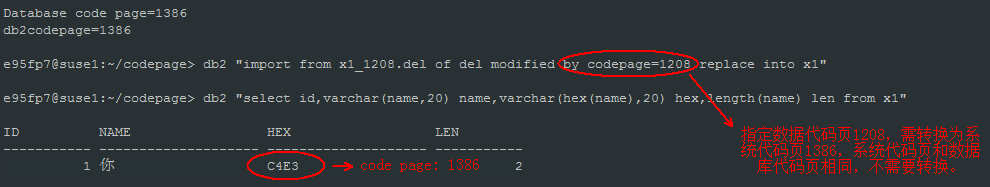
1. Import 案例1 (无转码)



缺省IMPORT 导入，认为输入文件中的数据是用当前系统的代码页编码的。

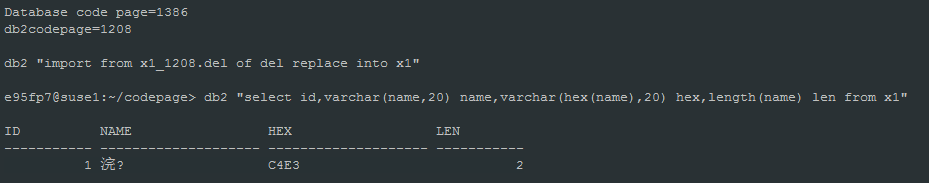
系统的代码页1386和数据库代码页1386相同，不进行代码页转换，文件中的数据(代码页1208)直接保存到1386的数据库，不正确的Import。

1. Import 案例2 (正确的Import: 数据代码页->系统代码页)



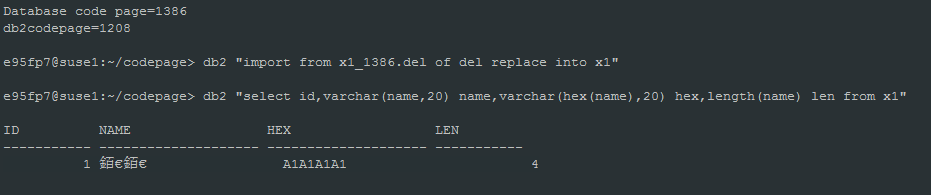
如果数据代码页和系统代码页不相同，可以通过codepage修饰符转换。 数据代码页->系统代码页->数据库代码页

1. Import 案例3 (正确的Import: 系统代码页->数据库代码页)



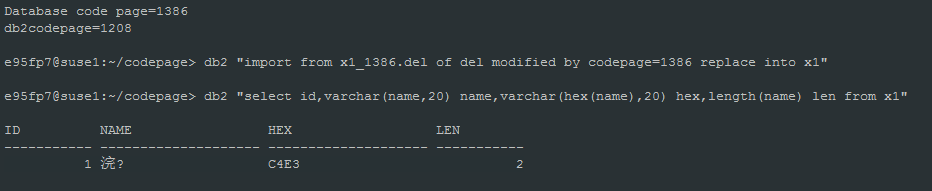
缺省的import导入，认为输入文件和系统代码页是相同的(都是1208，输入文件的确也是1208的代码页)，系统代码页(1208)和数据库代码页(1386)不相同，需要转码(1208->1386)。

1. Import 案例4 (不正确的Import: 系统代码页->数据库代码页)



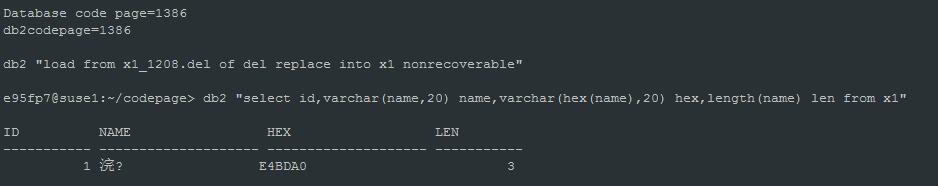
缺省的import导入，认为输入文件和系统代码页是相同的(都是1208，但输入文件不是1208的代码页)，把1386当做1208的代码页转码成1386的数据库代码页，转码错误。

1. Import 案例5 (正确的Import: 数据代码页->系统代码页->数据库代码页)



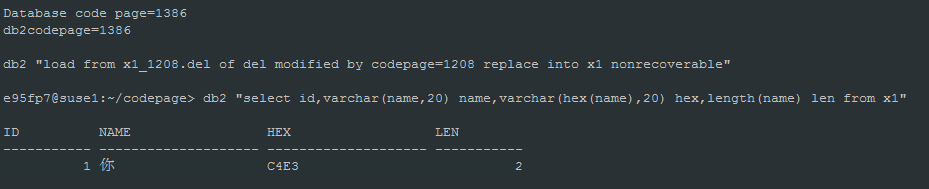
指定数据代码页为1386， 先转换为1208的系统代码页，再转换为1386数据库代码页，转码正确。

1. Load 案例1



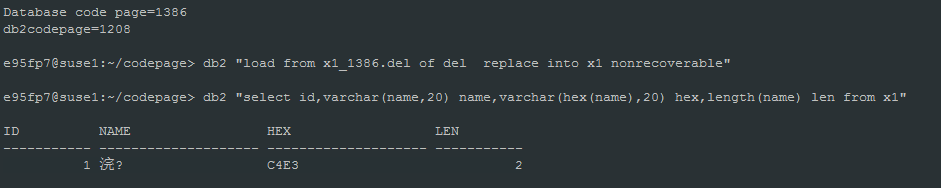
缺省load，1208代码页的数据直接按数据库的code page 1386保存，不正确的load

1. Load 案例2



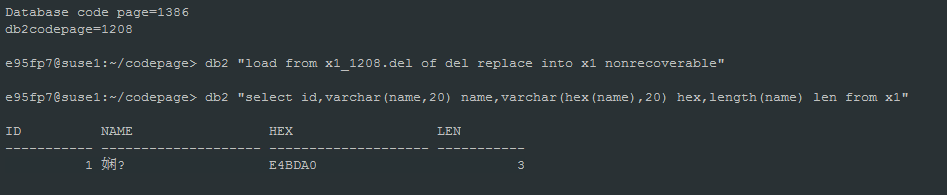
指定输入文件code page为1208，和数据库的code page不相同，需要转码(1208 -> 1386)，正确。

1. Load 案例3



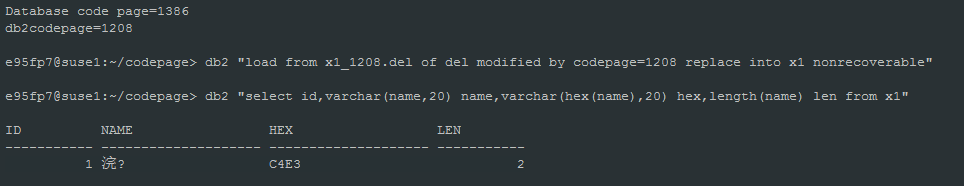
缺省Load，代码页为1386的数据直接保存到代码页为1386的数据库，正确。

1. Load 案例4



缺省Load，代码页为1208的数据直接保存到代码页为1386的数据库，不正确。

1. Load 案例5

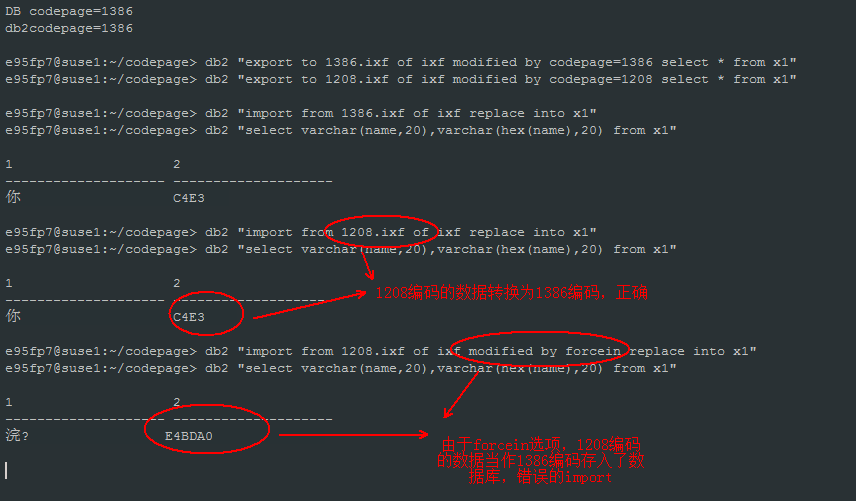


指定输入文件code page为1208，和数据库的code page不相同，需要转码(1208 -> 1386)，正确。

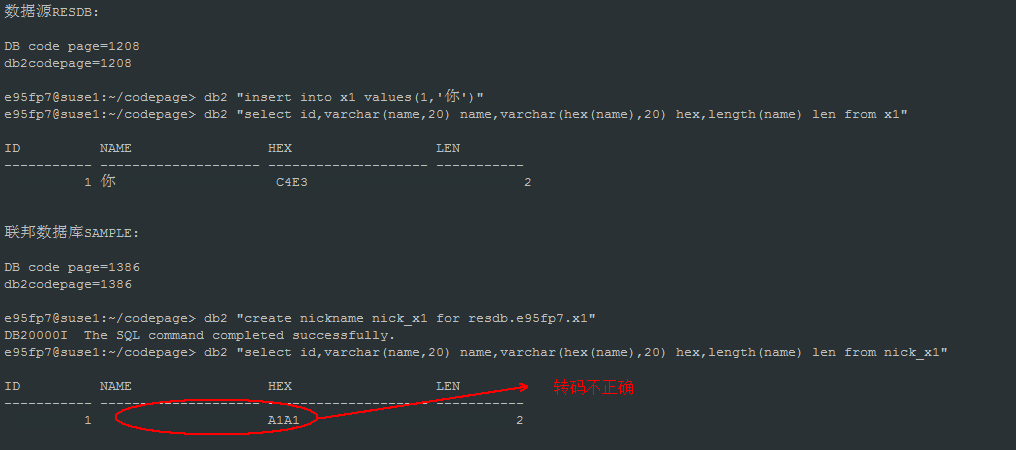
* + 1. **IXF格式code page转换(import/load)**

**IXF文件导入不支持modified by codepage=x选项。**

1. **If the IXF file and the import utility are in the same code page, processing occurs as for a regular application.**
2. **If the two differ, the FORCEIN option is not specified, and there is a conversion table, all data in the IXF file will be converted from the file code page to the application code page.**
3. **If the two differ, and the FORCEIN option is specified, the import utility assumes that data in the IXF file has the same code page as the application performing the import. (modified by forcein)**

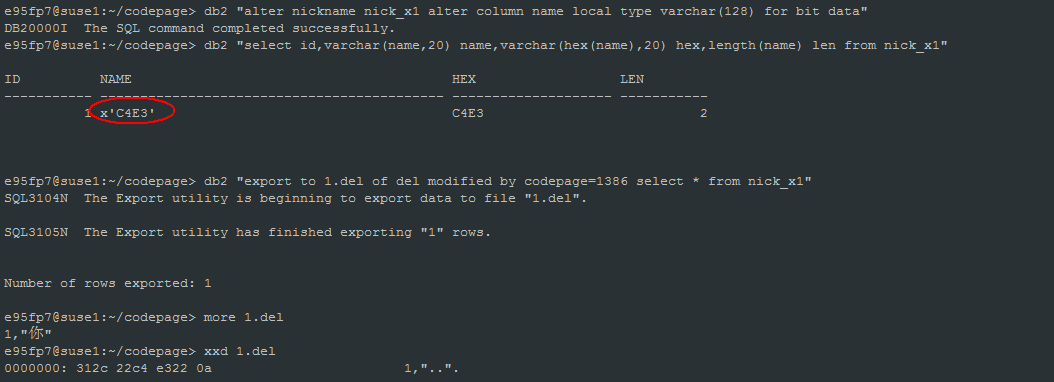


* + 1. **不同编码的数据库联邦中文问题**

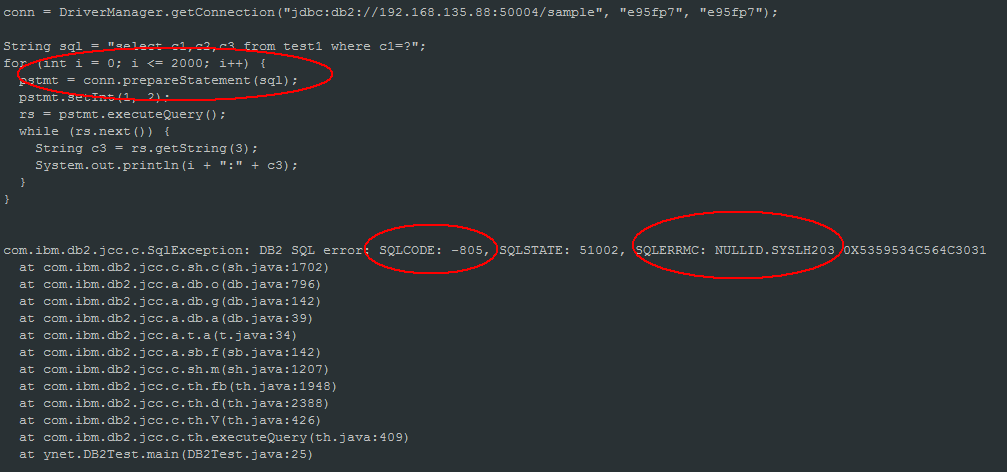


由于数据源(RESDB)code page是1208，联邦数据库(SAMPLE)code page是1386，需要转码，但是数据源x1表保存的数据code page是1386，导致转码错误。

如何解决？用for bit data, 设置该列不转码。



* 1. **SQL0805 messages for "SYSS[HN]xyy" and "SYSL[HN]xyy" packages**



**原因：**

The SQL0805 error for package NULLID.SYSL[HN]xyy can be reported when the CLI packages have not been bound to the database or the application has exhausted the available statement handles for the packages that have been bound to the database.

By default, there are 3 small and 3 large packages. Each small package allows a maximum of 64 statement handles per connection, and each large package allows a maximum of 384 statement handles per connection. The number of available statement handles by default is therefore (3 \* 64) + (3 \* 384) = 1344.

Definitions for the package name SYSL[HN]xyy:

[HN] - 'H' represents WITH HOLD, and 'N' represents NOT WITH HOLD.

'x' - is the isolation level: 0=NC, 1=UR, 2=CS, 3=RS, 4=RR

'yy' - is the package iteration 00 through FF

If the package name referenced matches SYSL[HN]x03 then this indicates the application had exhausted all available statement handles for the default packages bound to the database. Additional packages may be required.

**问题解决：**

Review the application logic and ensure statement objects are efficiently being closed and that the application is not excessively

opening more statement handles than what the imposed database limits allow.

If additional statement handles are required then from the application client take the following actions to bind additional packages.

db2 connect to <dbalias>

db2 "bind @db2cli.lst blocking all sqlerror continue grant public CLIPKG X"

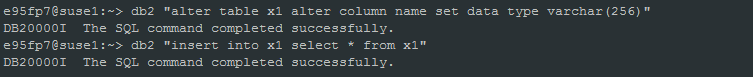
Where 'X' is the number of packages, up to 30 max, to be bound.

* 1. **什么情况下表结构修改需要reorg**

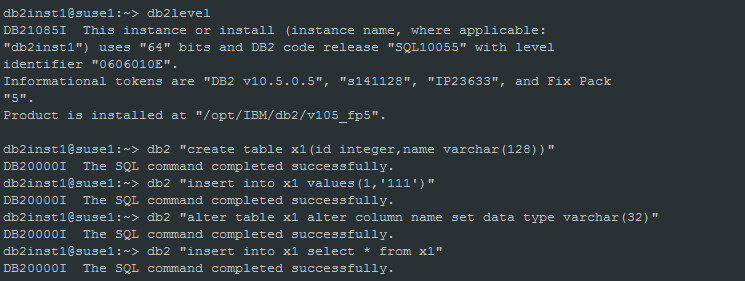
**以下操作不需要reorg表：**

1. **Set data type只有以下两种情况不需要reorg:**

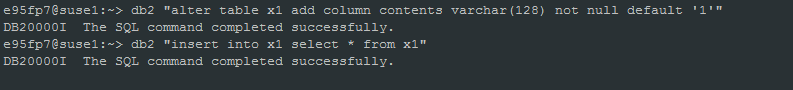
**Increasing the length of a VARCHAR or VARGRAPHIC column**



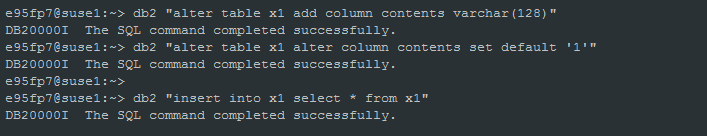
**Decreasing the length of a VARCHAR or VARGRAPHIC column without truncating trailing blanks from existing data (v9.7以后可以缩小varchar类型的长度)**



1. **Add column不需要reorg**

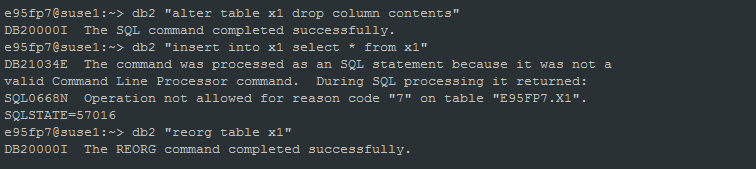


1. **Set default不需要reorg**

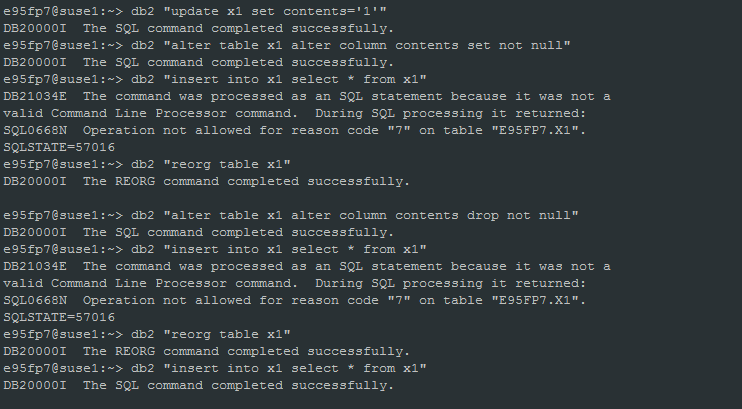


**以下操作需要reorg表：**

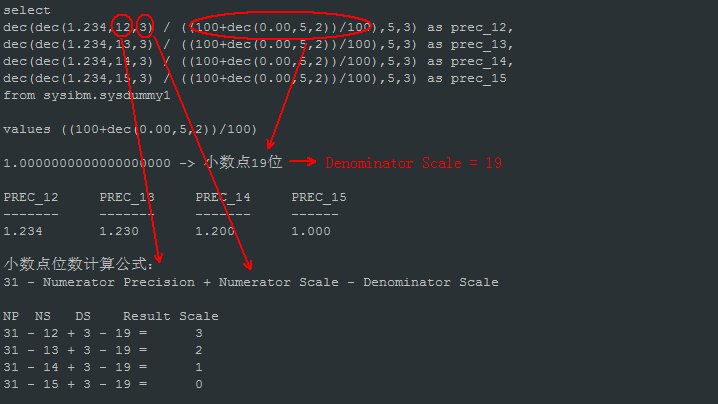
1. **drop column**



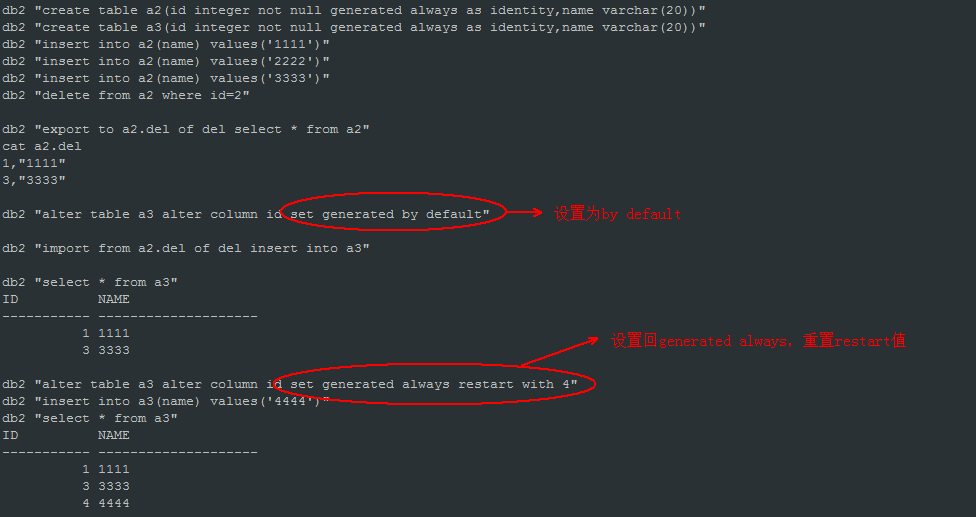
**2) set/drop not null**



* 1. **decimals类型除法，小数点的计算**



* 1. **数据迁移中identity字段的处理**



* 1. **Offline reorg分析**

步骤：

There are four phases in a CLASSIC or offline table reorganization operation:

1. SORT - During this phase, if an index was specified on the REORG TABLE command, or a clustering index was defined on the table, the rows of the table are first sorted according to that index. If the INDEXSCAN option is specified, an index scan is used to sort the table; otherwise, a table scan sort is used. This phase applies only to a clustering table REORG operation. Space reclaiming REORG operations begin at the build phase.
2. BUILD - During this phase, a reorganized copy of the entire table is built, either in its table space or in a temporary table space that was specified on the REORG TABLE command.
3. REPLACE - During this phase, the original table object is replaced by a copy from the temporary table space, or a pointer is created to the newly built object within the table space of the table that is being reorganized.
4. RECREATE ALL INDEXES - During this phase, all indexes that were defined on the table are re-created.

锁相关：

During the build phase, a U lock is acquired and held on the table. A U lock allows the lock owner to update the data in the table. Although no other application can update the data, read access is available. The U lock is upgraded to a Z lock after the replace phase starts. During this phase, no other applications can access the data. This lock is held until the table REORG operation completes.

在sort, build阶段，表可读；replace阶段开始，因为表加了Z锁，表不可读(也不允许脏读).

性能：

1. If there is enough space to do so, use the same table space for both the original table and the reorganized copy of the table, instead of using a temporary table space. This saves the time that is needed to copy the reorganized table from the temporary table space.
2. Consider dropping unnecessary indexes before reorganizing a table so that fewer indexes need to be maintained during the reorg operation.
3. Tune the sortheap database configuration parameters to control the space that is available for sorts. Tune the bufferpool size for temporary tablespace.
4. Adjust the number of page cleaners to ensure that dirty index pages in the buffer pool are cleaned as soon as possible.
5. If there is no HADR, consider turning off LOGINDEXBUILD.