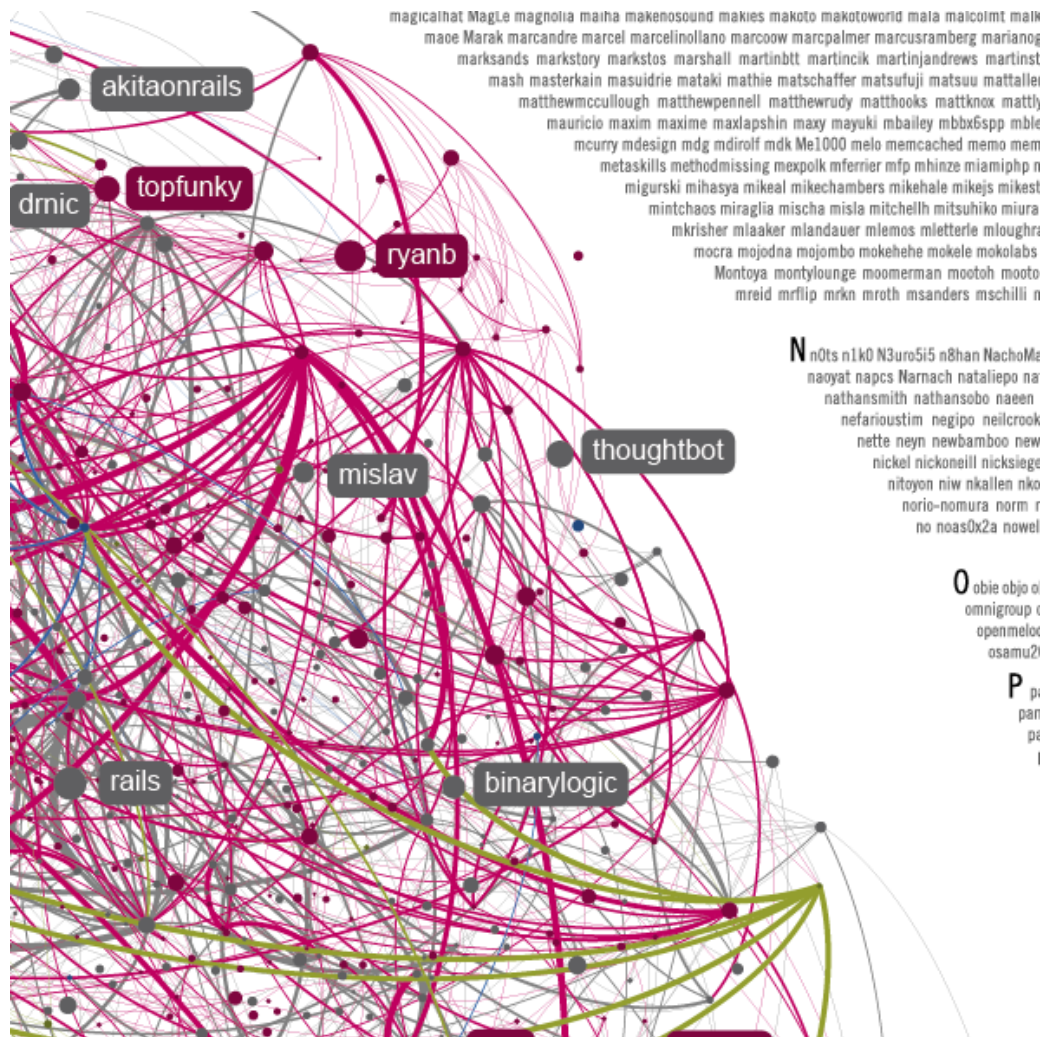


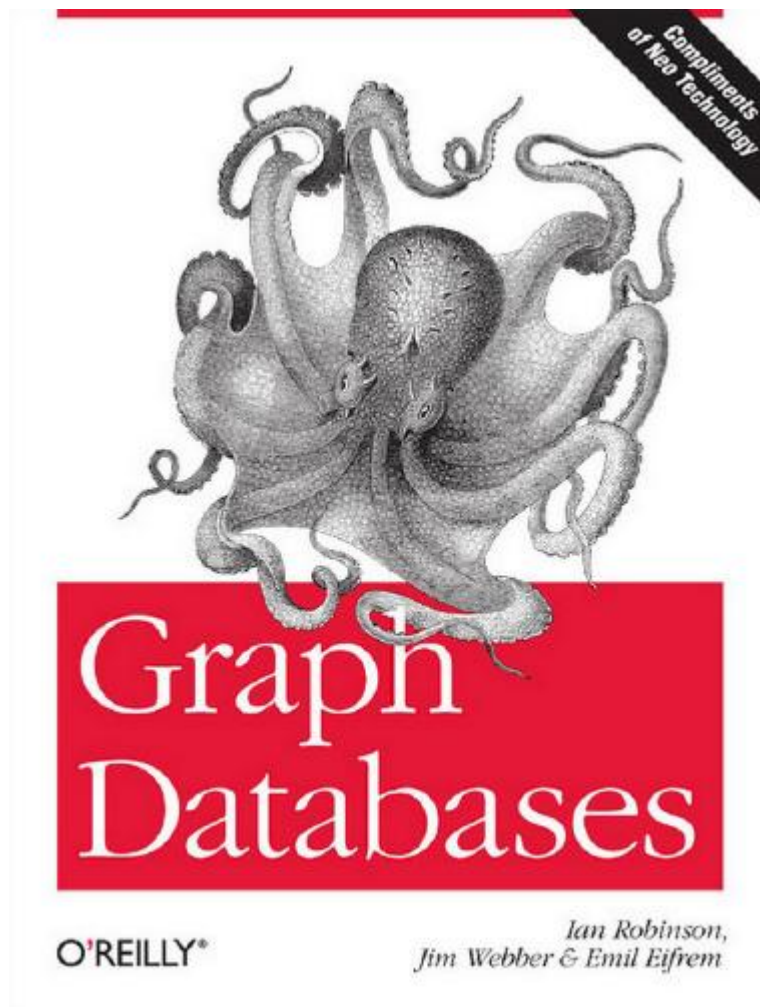
# DB2设计与 性能优化 第6周



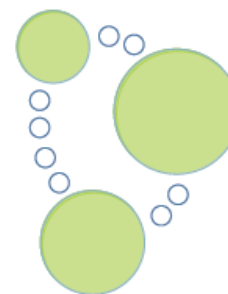
**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>



## The Neo4j Manual



- 
- Runstats
  - Reorg
  - Export
  - Import
  - Load
  - Backup & Restore

- 收集表和索引的统计信息
- 统计信息的作用
  - 优化器用来生成执行计划
  - 通过统计信息定位性能问题
- 何时需要做runstats
  - 当表的数据经过大量增删改后
  - 当表经过reorg后
  - 当创建新的索引后

- 收集表统计信息

```
runstats on table db2inst1.staff
```

- 收集索引统计信息

```
runstats on table db2inst1.staff for indexes all
```

- 收集表与索引统计信息

```
runstats on table db2inst1.staff and indexes all
```

- 收集表数据分布统计信息与详细索引统计信息

```
runstats on table db2inst1.staff with distribution and detailed  
indexes all
```

# 影响runstats工具性能的因素

- 表与索引的数据量大小
  - 数据量越大，runstats执行时间越长
- runstats中指定的列的数量
  - 指定的列越多，runstats执行时间越长
- runstats中指定的索引的数量
  - 指定的索引越多，runstats执行时间越长
- 是否指定收集数据分布信息
  - 收集数据分布信息将增加runstats的工作量
- 是否指定收集详细索引信息
  - 收集详细索引信息将增加runstats的工作量
- 是否指定使用采样方法收集统计信息
  - 采样能够减少runstats执行时间

# Runstats工具优化(1)

- 只对需要的列收集统计信息
  - SQL语句中出现在where子句中的列
  - SQL语句中出现在Join条件中的列
  - SQL语句中出现在Group by/Order by等子句中

```
runstats on table db2inst1.staff on columns (ID, NAME, DEPT);
```



## Runstats工具优化(2)

- 创建索引时，使用**COLLECT STATISTICS**参数
  - 创建索引的同时，收集索引的统计信息
  - 可以根据需要配合使用**DETAILED**或**SAMPLED**等参数

```
CREATE INDEX idx01 ON db2inst1.staff (name ASC) COLLECT STATISTICS;
```

- 使用**load**命令的**replace**选项向表中加载数据时，指定**STATISTICS USE PROFILE**参数
  - 加载数据的同时，收集统计信息
  - 需要在**load**命令执行之前，创建**STATISTICS PROFILE**

```
runstats on table db2inst1.staff and indexes all set profile only;
```

```
load from staff.ixf of ixf replace into db2inst1.staff STATISTICS USE PROFILE;
```

## Runstats工具优化(3)

- 收集“数据分布信息”将增加runstats执行时间
  - 当列被用在范围谓词(>,<,>=,<=,between等)时, 收集分布信息将会带来收益
  - NUM\_FREQVALUES(频率)参数越大, 信息越详细
  - NUM\_QUANTILES(分位数)参数越大, 信息越详细

```
runstats on table db2inst1.staff
WITH DISTRIBUTION on columns
(ID    NUM_FREQVALUES 100 NUM_QUANTILES 100,
 NAME NUM_FREQVALUES 50  NUM_QUANTILES 50,
 DEPT NUM_FREQVALUES 30  NUM_QUANTILES 30);
```

- 收集“索引详细信息”将增加runstats的执行时间
  - 指定SAMPLED参数, 可以使用采样技术

```
runstats on table db2inst2.staff for sampled detailed indexes all;
```

## Runstats工具优化(4)

- 使用采样，能够加快runstats的执行速度
  - 采用统计学方法，对部分采用数据收集统计信息
  - 适用于较大的表、非关键表
- 基于行的采样

```
runstats on table db2inst2.staff tablesample bernoulli(30);
```

- 基于页的采样

```
runstats on table db2inst2.staff tablesample system(30);
```

- 统计信息是优化器做优化的依据，因此十分重要
  - 统计信息不准确的情况下，SQL语句的性能分析将变得没有意义
- runstats后，不要忘记rebind静态SQL语句
- 表与索引的统计信息最好保持同步
- 对于某些短时间内变化量比较大而来不及做runstats的表(比如某些表在夜间会清空，第二天数据量又急剧增加)，可以将其volatile(易失表)属性打开

# 目录

- Runstats
- ➔ ■ Reorg
- Export
- Import
- Load
- Backup & Restore

# Reorg介绍

- Reorg是对表或索引进行重组的工具，用以提升系统性能
- Reorg动作以表或索引为单位
- 对表进行Reorg的作用
  - 整理存储碎片
  - 创建压缩字典
- 对索引进行Reorg的作用
  - 整理存储碎片
  - 创建压缩字典
  - 对索引二叉树的叶子节点进行整理
  - 对索引二叉树的高度进行整理

# Offline Reorg与Online Reorg

## ■ Offline Reorg

- 离线Reorg，重组表时，其他应用不能更新表
- Reorg的效果比较好
- 可以对表数据进行排序
- 自动重建索引，因此索引不需要单独Reorg
- 需要使用临时表空间

## ■ Online Reorg

- 在线Reorg，重组表时，其他应用程序不受影响
- Reorg的效果不如离线Reorg
- 不能对表数据进行排序
- 不会重建索引，因此索引需要单独Reorg
- 不需要使用临时表空间

- Offline Reorg的适用场景
  - 被重组的表有维护时间窗口
- Online Reorg的适用场景
  - 7×24系统，被重组的表没有维护时间窗口



- 离线重组表

```
reorg table db2inst1.staff use tempspace1
```

- 离线重组表，并按照索引idx01对表数据进行排序

```
reorg table db2inst1.staff index idx01 use tempspace1
```

- 开始在线重组表

```
reorg table db2inst1.staff INPLACE
```

- 停止在线重组表

```
reorg table db2inst1.staff INPLACE stop
```

- 暂停在线重组表

```
reorg table db2inst1.staff INPLACE pause
```

- 重启在线重组表

```
reorg table db2inst1.staff INPLACE resume
```

# reorgchk命令介绍

- 使用reorgchk命令判断表是否需要reorg
- reorgchk命令将统计信息作为计算依据，因此在reorgchk时应该确保统计信息的准确性
- 常用reorgchk命令

- 使用当前的统计信息，对某一张表运行reorgchk

```
reorgchk current statistics on table db2inst1.staff
```

- 使用当前的统计信息，对所有表运行reorgchk

```
reorgchk current statistics on table all
```

- 首先更新统计信息，然后对所有表运行reorgchk

```
reorgchk update statistics on table all
```

- 表相关的指标
  - F1:行溢出率, 应小于5%
  - F2:空间使用率, 应大于70%
  - F3:页面使用率, 应大于80%
- 索引相关的指标
  - F4:聚簇度, 应大于50%
  - F5:存放索引数据的叶子节点的空闲率应低于50%
  - F6:重建索引后, 是否能减小树的高度
  - F7:假删除的RID应在总数据量的20%以内
  - F8:假删除的叶子节点应在20%以内

# reorgchk命令结果分析(2)

## ■ 表相关的指标

- 当相关性指标被标记为\*时，表示需要reorg

Table statistics:

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

SCHEMA.NAME	CARD	OV	NP	FF	PCTBLK	TSIZE	F1	F2	F3	REORG
Table: DB2INST2.EMPMD*	10000	0	40	352	10	250000	0	9	90	---

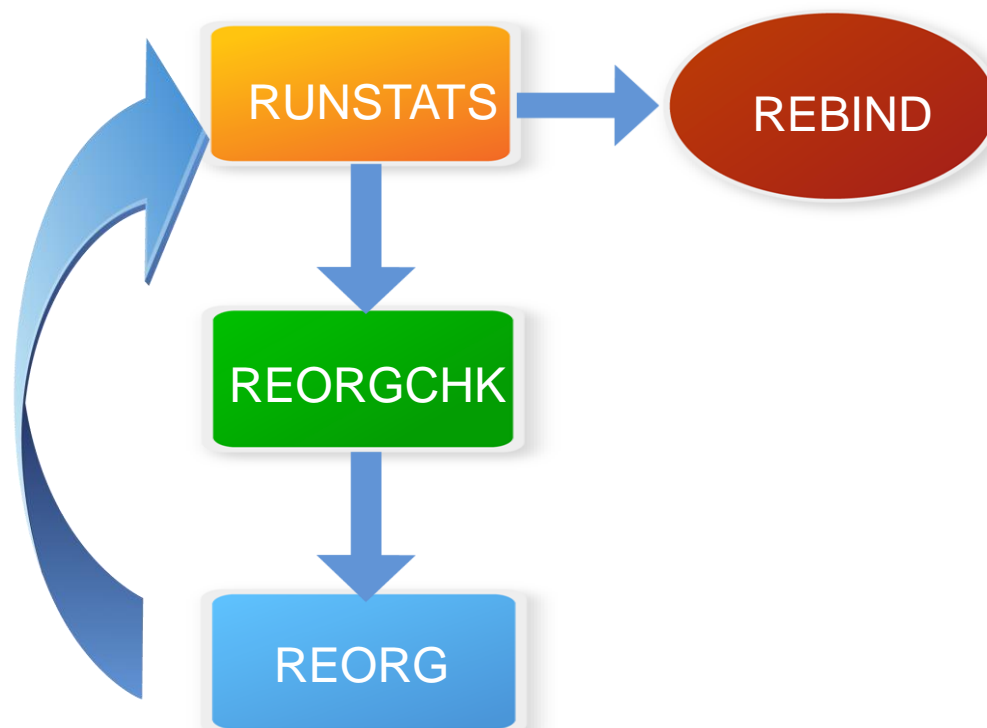
## ■ 索引相关的指标

Index statistics:

F4: CLUSTER RATIO or normalized CLUSTERFACTOR > 80  
F5: 100 \* (Space used on leaf pages / Space available on non-empty leaf pages) > MIN(50, (100 - PCTFREE))  
F6: (100 - PCTFREE) \* (Amount of space available in an index with one less level / Amount of space required for all keys) < 100  
F7: 100 \* (Number of pseudo-leaf pages / Total number of RIDs) < 20  
F8: 100 \* (Number of pseudo-empty leaf pages / Total number of leaf pages) < 20

SCHEMA.NAME	INDCARD	LEAF	ELEAF	LVLS	NDEL	KEYS	LEAF_RECSIZE	NLEAF_RECSIZE	LEAF_PAGE_OVERHEAD	NLEAF_PAGE_OVERHEAD	PCT_PAGES_SAVED	F4	F5	F6	F7	F8	REORG
Table: DB2INST2.STAFF																	
Index: DB2INST2.IDX01	35	1	0	1	0	35	9	9	1568	1568	0	100	-	-	0	0	---

1. 收集统计信息
2. 判断是否需要reorg
3. 执行reorg
4. 收集统计信息
5. 重新bind静态SQL



# Reorg工具的优化建议

- 通过分析reorgchk命令的结果，只对需要Reorg的表进行Reorg，尽量减少Reorg操作的次数
- 对于从来不使用的索引，应将其删除，以节省Reorg的成本
- 将临时表空间定义在性能较好的存储上
- 对于关键业务系统，可以使用online reorg，并实施监控，以便在出现性能问题的时候终止Reorg操作

# Reorg工具注意事项

- 对表进行offline reorg后，无需再对索引进行reorg
  - Offline table reorg后会重建索引
- Reorg后，不要忘记执行runstats以及bind静态SQL语句
- Reorg后，重新运行reorgchk，发现结果没有任何改变
  - reorgchk以runstats的结果作为运算依据，因此只要不重新运行runstats，reorgchk的结果就不会改变
- Offline Reorg中途失败，有可能导致索引的损坏，因此尽量不要强行终止正在运行的Offline Reorg

# 目录

- Runstats
- Reorg
-  ■ Export
- Import
- Load
- Backup & Restore



- 将**select**查询语句的结果存储为文件
- 支持的文件格式
  - **ixf**: DB2格式的二进制文件
  - **del**: 文本格式
  - **wsf**: Lotus文件格式
- 通常作为数据移动的工具使用

# 常用Export命令

- 导出普通表

```
export to staff.ixf of ixf select * from db2inst1.staff
```

- 导出包含LOB列的表

```
export to staff.ixf of ixf
  lobs to /db2exp1/, /db2exp2/ modified by lobsinfile
select * from db2inst1.staff
```

- 导出包含XML列的表

```
export to staff.ixf of ixf
  XML to /db2exp1/, /db2exp2/ modified by XMLINSEPFILS
select * from db2inst1.staff
```

# Export工具的性能

- 影响Export工具性能的因素
  - Export中指定的Select语句的执行效率
  - Export指定的文件目录的磁盘写速度
- 对Export工具的优化
  - 首先优化Select语句的性能
  - 指定写速度较快的磁盘作为文件存放目录
  - LOB或XML可以指定多个目录，实现并行写

- Runstats
- Reorg
- Export
- ➔ ■ Import
- Load
- Backup & Restore

- 将外部文件Insert到数据库表中
- 支持的文件格式
  - ixf: DB2格式的二进制文件
  - del: 文本格式
  - wsf: Lotus文件格式
- 支持的模式
  - INSERT
  - INSERT\_UPDATE
  - REPLACE
  - REPLACE\_CREATE
- 通常作为数据移动的工具使用

- 将文件中的数据插入到表中

```
import from staff.ixf of ixf insert into db2inst1.staff
```

- 导入过程中允许其他程序对表进行写操作

```
import from staff.ixf of ixf  
ALLOW WRITE ACCESS  
insert into db2inst1.staff
```

- 每10000条数据提交一次

```
import from staff.ixf of ixf  
COMMITCOUNT 10000  
insert into db2inst1.staff
```

# Import工具的性能

- 影响Import工具性能的因素
  - 指定的insert语句执行效率
  - 指定的文件目录的磁盘读速度
  - 提交频率
- 对Import工具的优化
  - 首先优化Insert语句的性能
  - 将录入文件存放在性能较好的磁盘目录上
  - 使用Online模式以避免对其他应用的影响
  - 在不产生日志满和锁升级等错误的前提下，尽量将COMMITCOUNT指定的大一些

# 目录

- Runstats
- Reorg
- Export
- Import
- ➔ ■ Load
- Backup & Restore



- 将外部文件装载到数据库表中
- Load与Import的区别
  - Load不记录日志
  - Load不触发TRIGGER
  - Load不检查外键约束和检查约束
  - Load过程中可以收集统计信息
  - 可以并发执行
- Load工具的4个阶段
  - Load数据阶段
  - 创建索引阶段
  - 删除非法数据阶段
  - 索引copy阶段

# 常用Load命令

- 将文件中的数据插入表中

```
load from staff.ixf of ixf insert into db2inst1.staff
```

- 清空表，并将文件中的数据插入表中

```
load from staff.ixf of ixf replace into db2inst1.staff
```

- 以在线方式将数据写入表中

```
load from staff.ixf of ixf
  insert into db2inst1.staff
  ALLOW READ ACCESS
```

- 对于归档模式的数据库，使用NONRECOVERABLE参数
  - 使用该参数，load后无需做表空间的backup

```
load from staff.ixf of ixf
  insert into db2inst1.staff
  NONRECOVERABLE
```

# Load工具优化(1)

- 使用CPU\_PARALLELISM参数指定CPU并行性
  - 将该参数指定为系统CPU个数(或核心数)

```
load from staff.ixf of ixf
insert into db2inst1.staff
CPU_PARALLELISM 5
ALLOW READ ACCESS
```

- 使用DISK\_PARALLELISM参数指定I/O并行性
  - 将该参数指定为表空间对应的物理磁盘数

```
load from staff.ixf of ixf
insert into db2inst1.staff
DISK_PARALLELISM 5
ALLOW READ ACCESS
```

- 使用DATA BUFFER参数指定LOAD使用的buffer大小
  - 建议每个CPU对应400-800之间
  - 比如5个CPU，可以设置： $5 \times 800 = 4000$

```
load from staff.ixf of ixf
insert into db2inst1.staff
DATA BUFFER 4000
ALLOW READ ACCESS
```

- 使用SORT BUFFER参数指定LOAD使用的排序堆大小
  - 该参数将覆盖SORTHEAP
  - 该参数定义的排序内存不能超过SORTHEAP

```
load from staff.ixf of ixf
insert into db2inst1.staff
SORT BUFFER 1024
ALLOW READ ACCESS
```

## Load工具优化(3)

- 指定**ANYORDER**参数提高并行性
  - 指定该参数后，数据写入的顺序将与文件内数据不一致
  - 指定该参数能够提升**36%**的性能(不包括索引维护)

```
load from staff.ixf of ixf
  modified by ANYORDER
insert into db2inst1.staff
```

- 指定**FASTPARSE**取消对文件内容的格式检查
  - 只能用在文本文件(**DEL**格式)的load中
  - 该参数假定文件的格式是正确的(由用户负责)
  - 可以提升**16%**的性能(不包括索引维护)
  - 有可能将错误的数据导入

```
load from staff.del of del
  modified by FASTPARSE
insert into db2inst1.staff
```

- 对于“不干净”的数据，指定NOROWWARNINGS参数
  - load默认会为每一条非法数据打印一个提示
  - 指定该参数后，对于非法数据不会提示

```
load from staff.del of del  
modified by NOROWWARNINGS  
insert into db2inst1.staff
```

- COPY YES参数对性能的影响
  - 指定COPY YES参数后，需要将数据进行copy，因此对性能有负面影响
  - COPY YES参数可以提高数据库的数据高可用性

```
load from staff.del of del  
modified by NOROWWARNINGS  
insert into db2inst1.staff  
COPY YES  
TO ./copy_yes_dir1/, ./copy_yes_dir2/
```

- 下列参数将对索引重建阶段具有优化作用
  - 调整**SORT**相关的系统参数
    - SORTHEAP**,
    - SHEAPTHRES\_SHR**
  - 调整**I/O**相关参数
    - NUM\_IOSERVERS**
    - NUM\_IOCLEANERS**

- 使用管道或Cursor进行load

- 使用管道进行load可以避免向操作系统写文件

```
$ mkfifo staff.fifo  
$ db2 "export to staff.fifo of del select * from staff"
```

```
$ db2 create table staff2 like staff  
$ db2 "load from staff.fifo of del insert into staff2"
```

- 使用Cursor进行load可以避免向操作系统写文件，同时避免通过FTP传输文件

```
declare mycurs cursor  
database DB1 user user1 using pwd1  
for select two,one,three from db2inst1.staff;  
load from mycurs of cursor insert into db2inst1.staff;
```



- 对生产系统进行load操作需要注意安全
  - 没有NONRECOVERABLE参数或COPY YES参数，可能导致表空间处于backup pending状态
- Load工具在Linux系统上需要足够的Message Queues支持
- 对于存在约束的表，Load完成后，表将处于check pending状态，需要运行SET INTEGRITY命令进行检查
- 不要终止正在运行的Load操作
- 对于重要系统，可以考虑使用Import代替Load

- Runstats
- Reorg
- Export
- Import
- Load
- ➔ ■ Backup & Restore

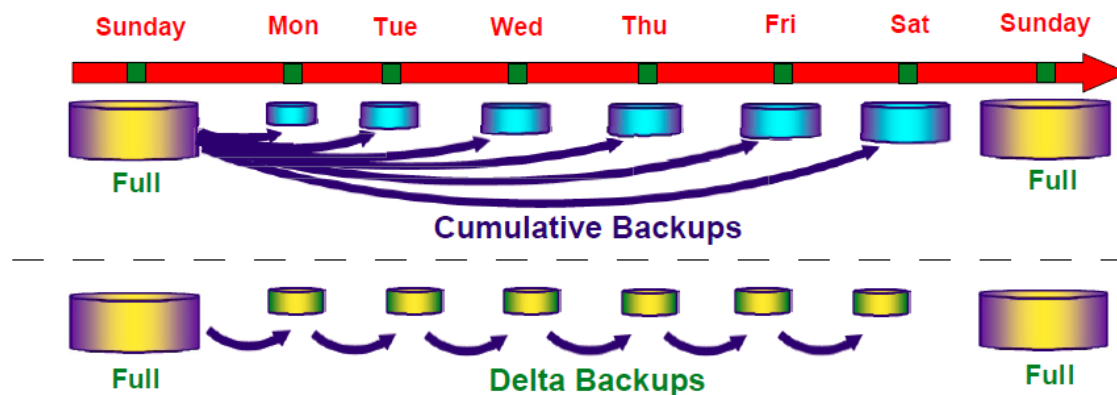
# 数据库备份恢复策略的优化

- 制定合理的数据库备份策略
  - 保证恢复时间尽量短
  - 保证备份对业务系统影响最小化，同时节省存储空间
- 数据库备份策略考虑因素
  - 全库备份周期
  - 增量备份周期与类型
  - 是否启用备份压缩
  - 离线与在线备份

# 备份周期的选择

## ■ 备份周期

- 备份周期较长，则需要更长的恢复时间
- 备份周期较短，则占用更多的存储空间

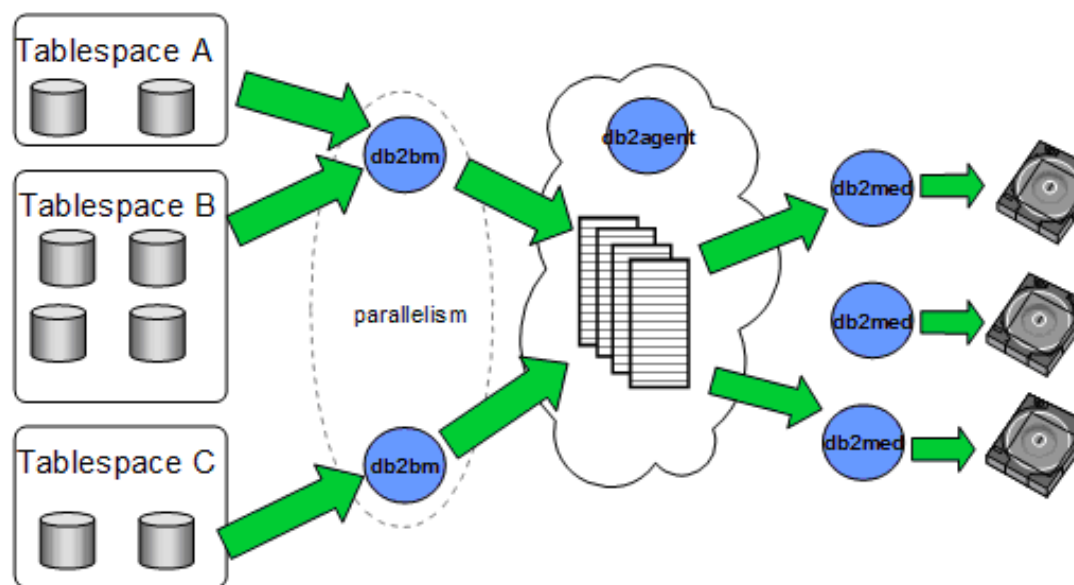


## ■ 备份周期的选择

- 考虑可接受的恢复时间
- 考虑当前存储系统的容量
- 可以通过增量备份，缩短恢复数据所需要的时间
- 通过压缩备份，节省存储空间

# 备份进程模型

- db2bm: 将数据从表空间读至缓冲区
- db2med: 将数据从缓冲区写入磁盘或磁带
- data buffer: 内存缓冲区，从UTIL HEAP中分配



# 备份工具参数的优化

- WITH num-buffers BUFFERS
  - 指定缓冲区的个数，建议配置为目标目录的2倍
- BUFFER buffer-size
  - 指定缓冲区的大小，建议设置为表空间Extend大小的倍数加一页
- PARALLELISM n
  - 指定并行度，建议配置为表空间的个数
- UTIL\_IMPACT\_PRIORITY
  - 指定Backup程序的优先级，1为最低，100为最高
- 指定COMPRESS压缩备份
  - CPU使用率将增高
  - 备份文件将明显变小，节省I/O成本
- 将上述参数调整后，需要确保UTIL\_HEAP\_SZ足够
  - num-buffers\* buffer-size不能超过该参数

```
db2 backup db sample \
to /backup1/, \
    /backup2/, \
    /backup3/ \
with 6 buffers \
buffer 1025 \
PARALLELISM 5 \
COMPRESS \
UTIL_IMPACT_PRIORITY 80
```

# 恢复工具参数的优化

- WITH num-buffers BUFFERS
  - 指定缓冲区的个数，建议配置为目标目录的2倍
- BUFFER buffer-size
  - 指定缓冲区的大小，建议设置为表空间Extend大小的倍数加一页
- PARALLELISM n
  - 指定并行度，建议配置为表空间的个数
- 将上述参数调整后，需要确保UTIL\_HEAP\_SZ足够
  - num-buffers\* buffer-size不能超过该参数

```
db2 restore db sample \
from /backup1/, \
    /backup2/, \
    /backup3/ \
taken at 20120104195556 \
with 6 buffers \
buffer 1025 \
PARALLELISM 5
```

- 在线备份，建议使用include log选项，防止归档日志丢失造成备份不可用
- 备份恢复策略不宜太复杂
- 定制备份恢复定期演练策略



# 数据库恢复的经典三步骤

1、数据库备份时必须加上online和include logs选项

```
db2 backup database sample online to /data/backup include logs
```

2、数据库恢复:

```
db2 "restore database sample from /data/backup taken at 20120331234149 logtarget /data/backup/logs"
```

3、恢复日志:

```
db2 "rollforward database sample to end of logs and complete overflow log path (/data/backup/logs)"
```

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 <http://edu.dataguru.cn>



# Thanks

## FAQ时间