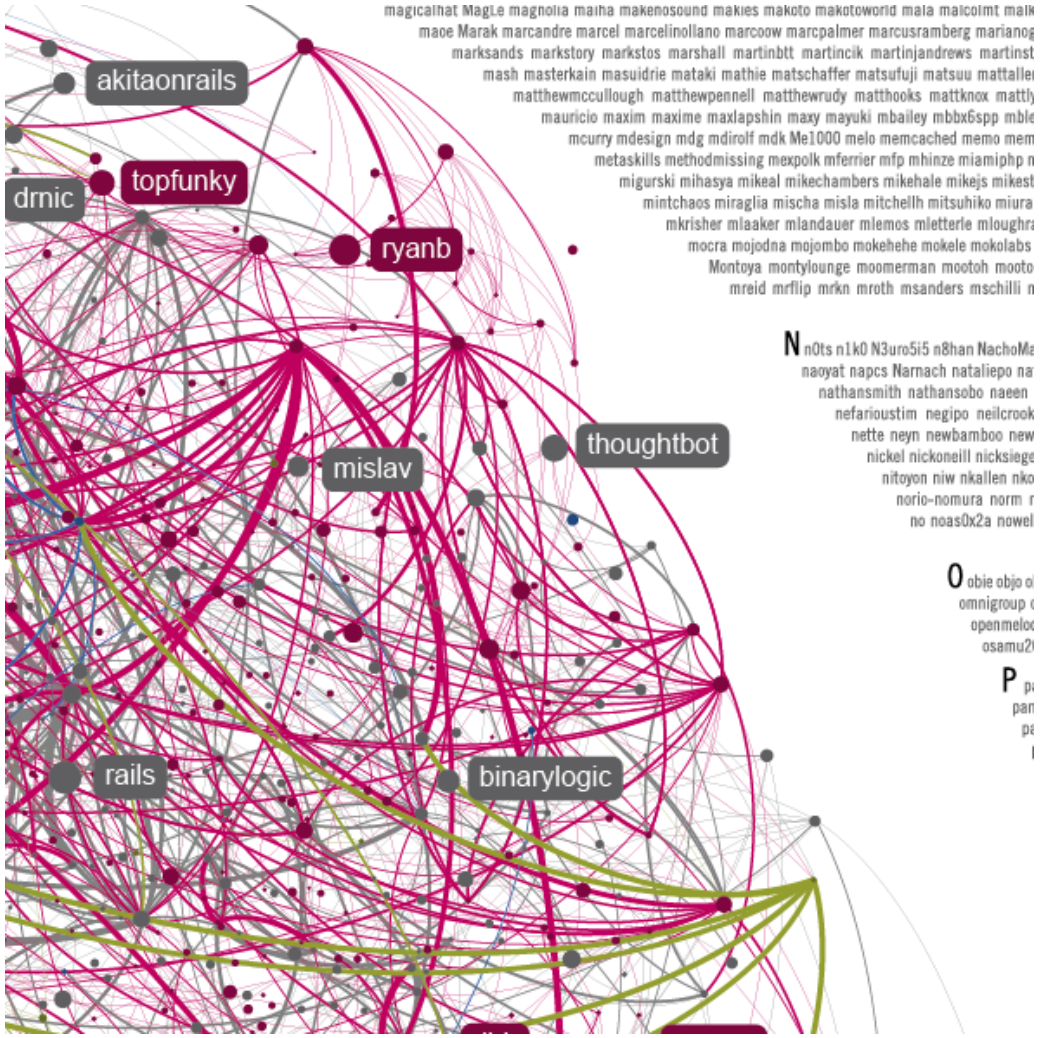
DATAGURU 炼数成金

# DB2设计与性能优化

# 第8周



# 法律声明

【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

# 提纲

- 索引的优势
- 索引在DB2内核的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

# DB2何时使用全表扫描

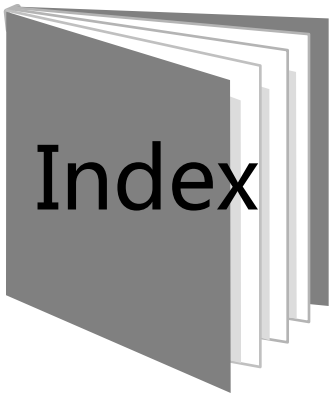
- 全表扫描检索行时会从数据页直接读取所有行。
- DB2优化器选择表扫描，而不选择索引扫描的情况：
  - 表的大部分数据行都必须被访问才能获得查询结果
  - 表很小
  - 索引不存在

```
Original Statement:
------------------
select acct_id, name
from acct
where balance > 25000


Access Plan:
-----------
        Total Cost:       11421.2
        Query Degree:     1

      Rows
     RETURN
     (   1)
      Cost
       I/O
       |
     5263.16
     TBSCAN
     (   2)
     11421.2
      2859
       |
     100000
  TABLE: ADMIN
 ACCT
```

# 如何避免全表扫描?

- 全表扫描的代价：
  - 更多的I/O操作
  - 更多的缓冲池空间
  - 扫描全部行需要更多的CPU
- 如何避免全表扫描：
  - 合适的索引
    - 范围定界谓词（Range-delimiting）
    - Index-SARGable的谓词
  - 完全索引扫描(Index-Only Access )
  - 支持双向扫描的索引
  - MQT（物化视图）

# 利用索引提高性能（1/2）




- 创建索引，可以:
  - 避免不必要的全表扫描
  - 保证列上数据唯一性
  - 避免排序
  - 加速频繁执行的查询
  - 内表上创建的索引能加速多表连接
  - 减少锁等待和死锁
- 为表创建有意义的主键
- 维护聚集索引
- 灵活运用组合索引

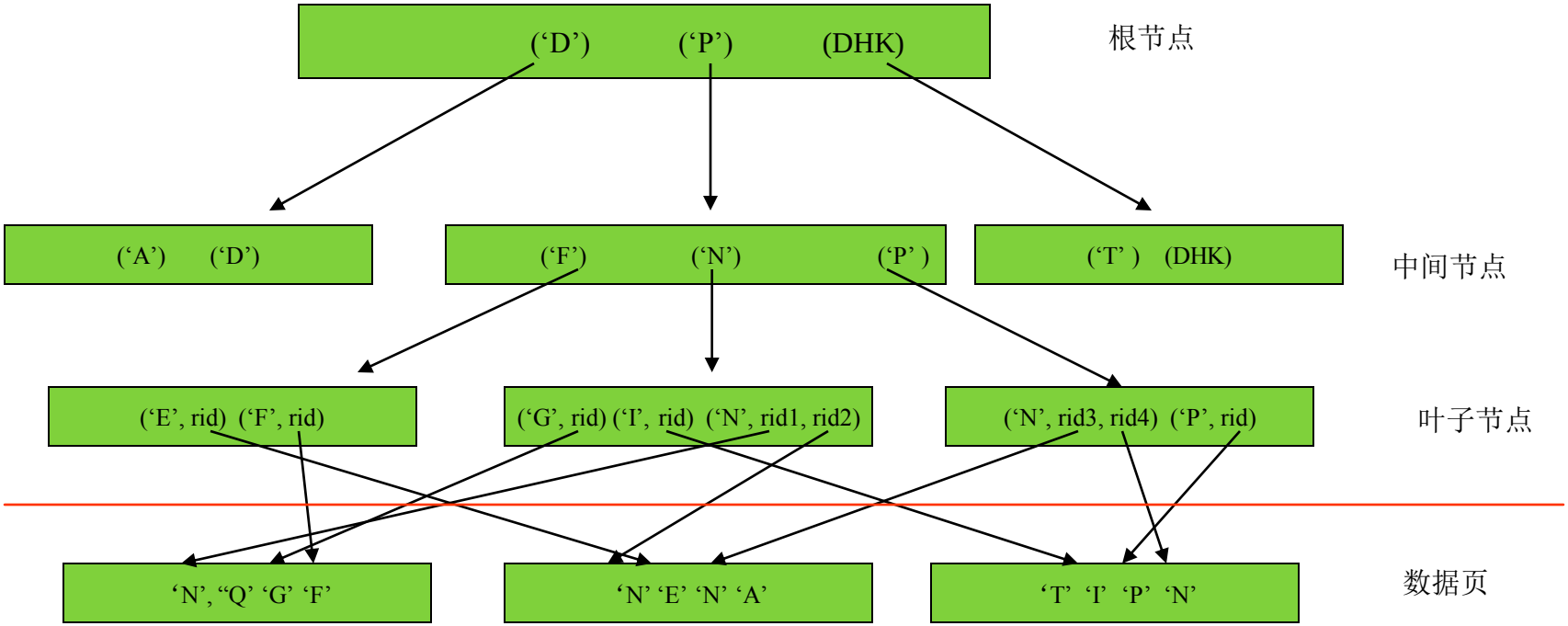
# 使用索引提高性能（2/2）

- 不要创建冗余的索引
- 保证你的索引正被使用
  - 使用EXPLAIN工具分析SQL的执行计划
  - 在SYSCAT.PACKAGEDEP中检查静态SQL的索引使用
- 维护索引的代价
  - IUD（Insert，Update，Delete）操作的额外开销
  - 更多的磁盘空间
  - 备份和恢复的时间
  - 增加编译或者准备SQL语句的代价
  - 实用工具的代价(RUNSTATS, REORG, LOAD)
- 在某些情况下让完全索引扫描成为可能

# 提纲

- 索引的优势
- 索引在DB2内核的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

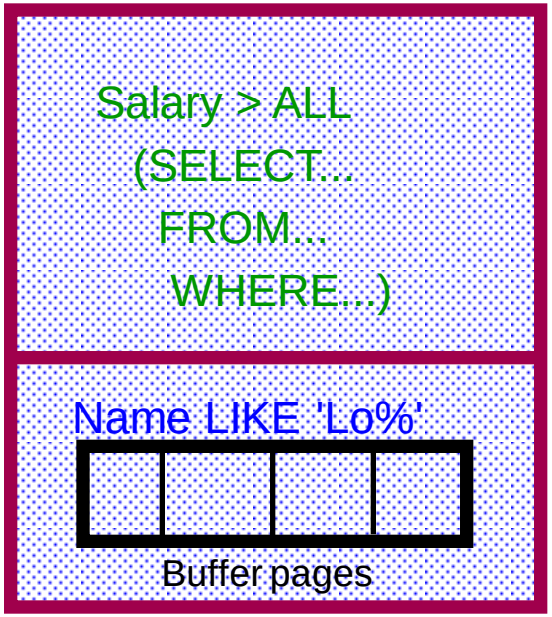
# 索引的内部结构- B+树




Note 1: DHK=Dummy High Key

# 谓词处理的方式

Residual 谓词



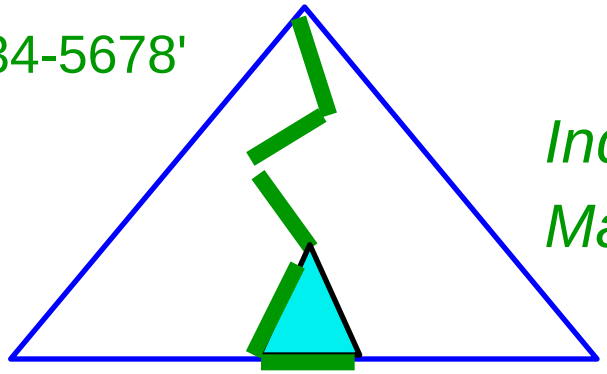

*RDS*

*Data Manager*

Search Arguments (SARGs)谓词

SSN= '012-34-5678'

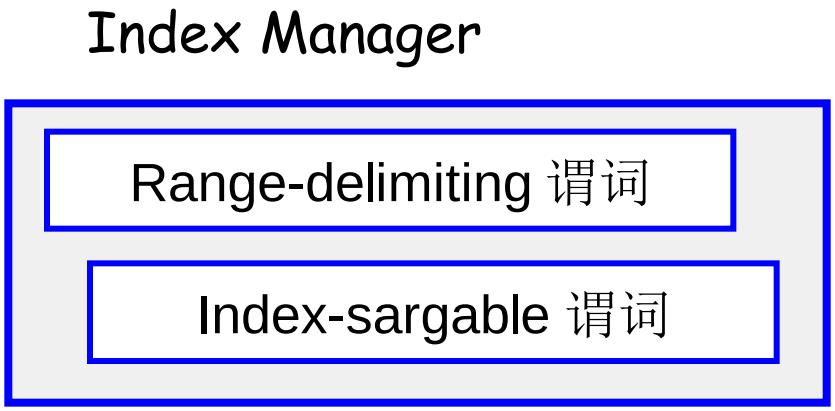
Rang-delimiting 谓词：
Index Start/Stop Key

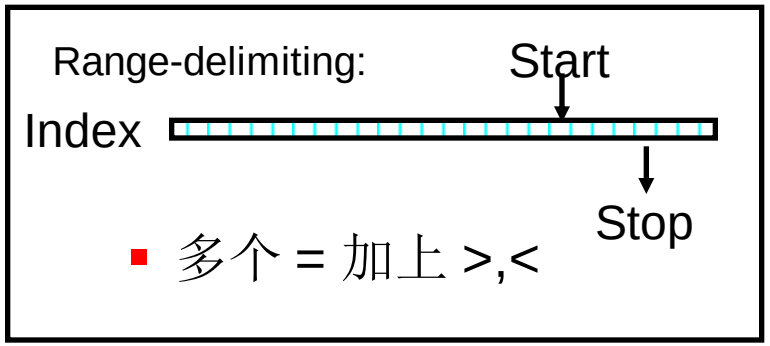


*Index Manager*

Index Sargable 谓词：
可用叶子节点中的数据计算

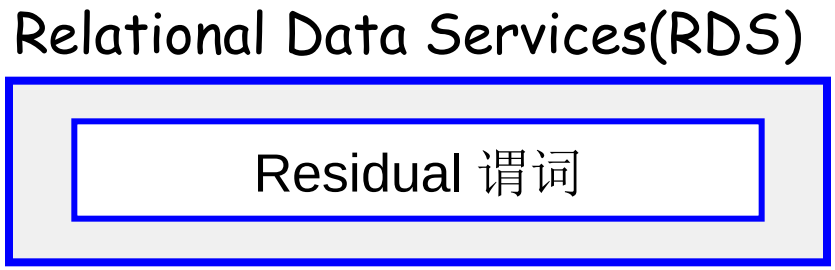
# WHERE子句中谓词（Predicate）处理效率



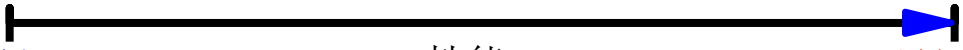





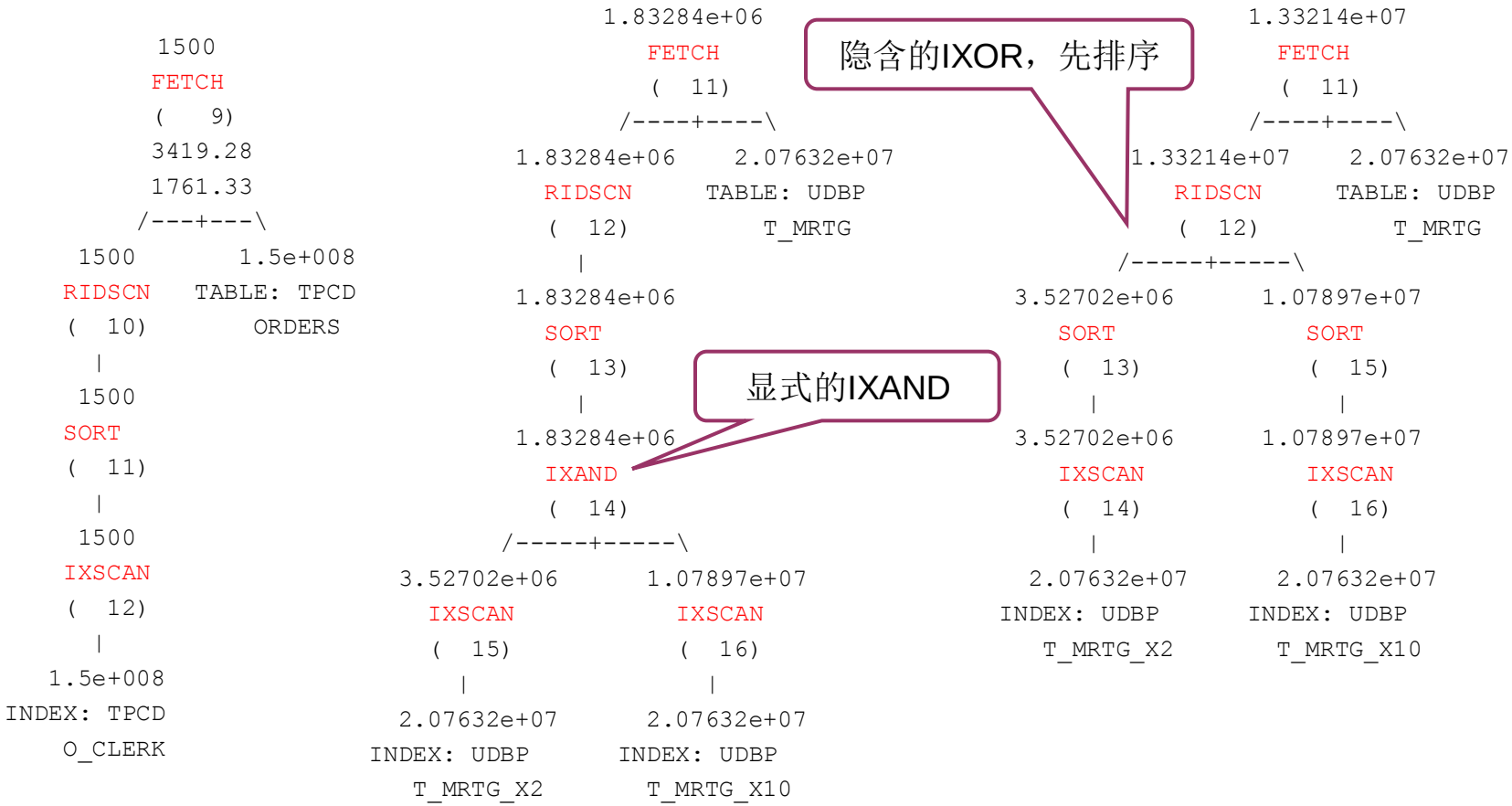
- <>
- Like '%H'



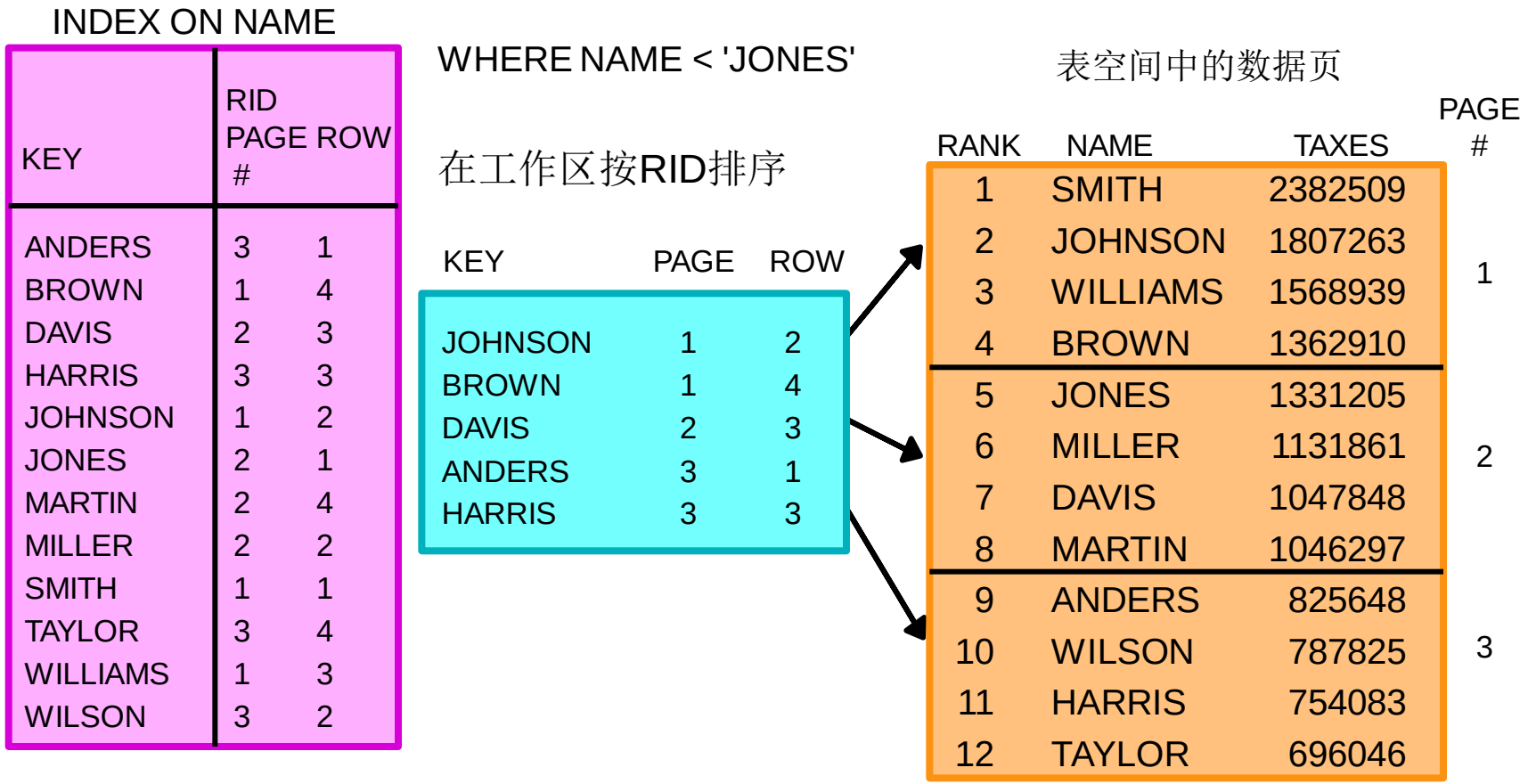

- 关联子查询
- 量化子查询
  (any, all, some, in)
- Long varchar等数据类型



# List Prefetch, Index ANDing和Index ORing



# 列表预取(List Prefetch)是如何工作的



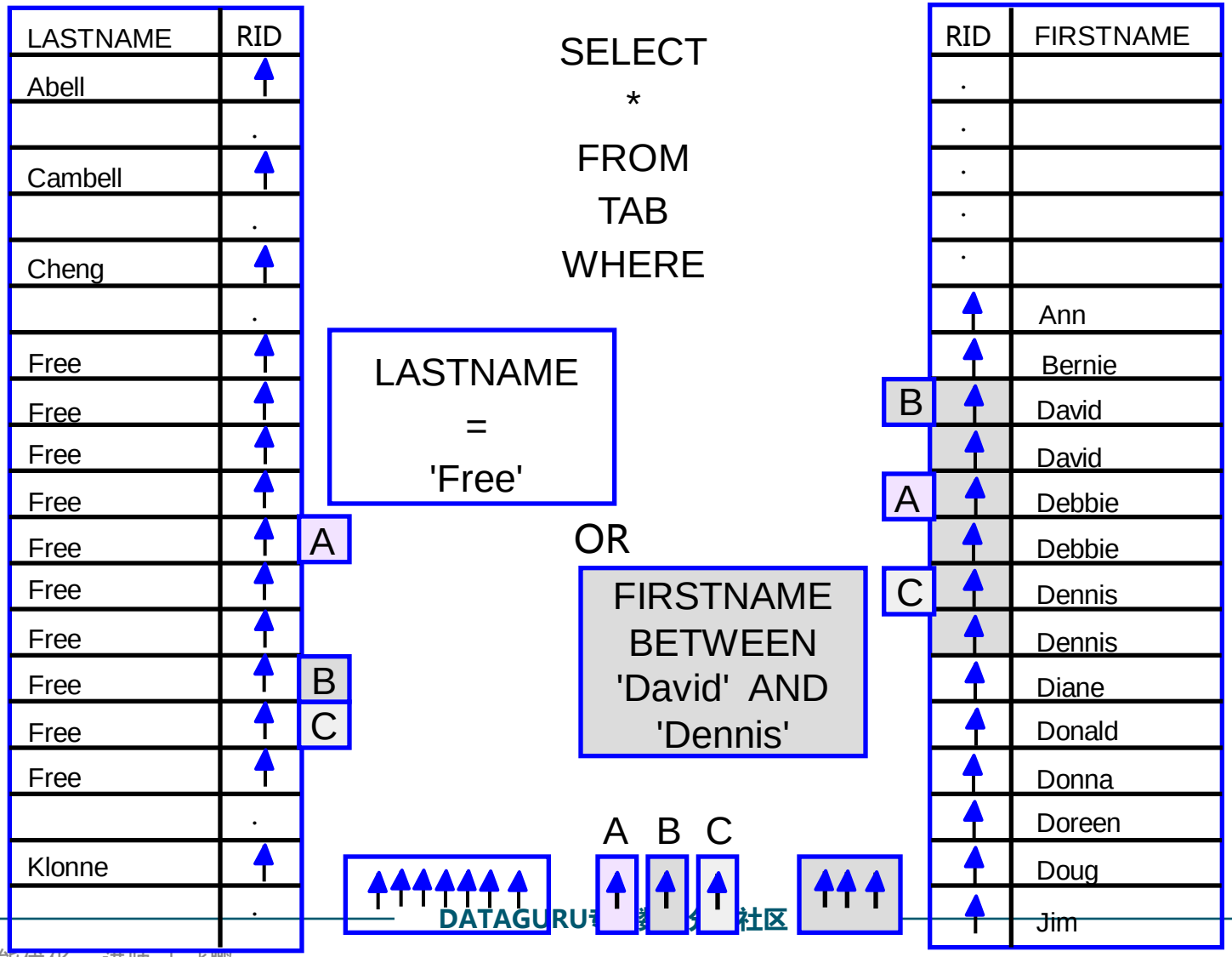

- 能更多的挖掘非聚集（nonclustered）索引的好处
- 避免对同一个数据页多次I/O

# Explain详细信息中的顺序预取

```
IXSCAN: (Index Scan)
        Arguments:
        ---------
        MAXPAGES: (Maximum pages for
                   prefetch)
            83
        PREFETCH: (Type of Prefetch)
            SEQUENTIAL
        ROWLOCK  : (Row Lock intent)
            NONE
        SCANDIR  : (Scan Direction)
            FORWARD
        TABLOCK  : (Table Lock intent)
            INTENT NONE
```

```
TBSCAN: (Table Scan)
        Arguments:
        ---------
        MAXPAGES: (Maximum pages for
                   prefetch)
            ALL
        PREFETCH: (Type of Prefetch)
            SEQUENTIAL
        ROWLOCK  : (Row Lock intent)
            NEXT KEY SHARE
        SCANDIR  : (Scan Direction)
            FORWARD
        TABLOCK  : (Table Lock intent)
            INTENT SHARE
```

# 索引或操作(Index ORing)



# 提纲

- 索引的优势
- 索引在DB2内核中的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

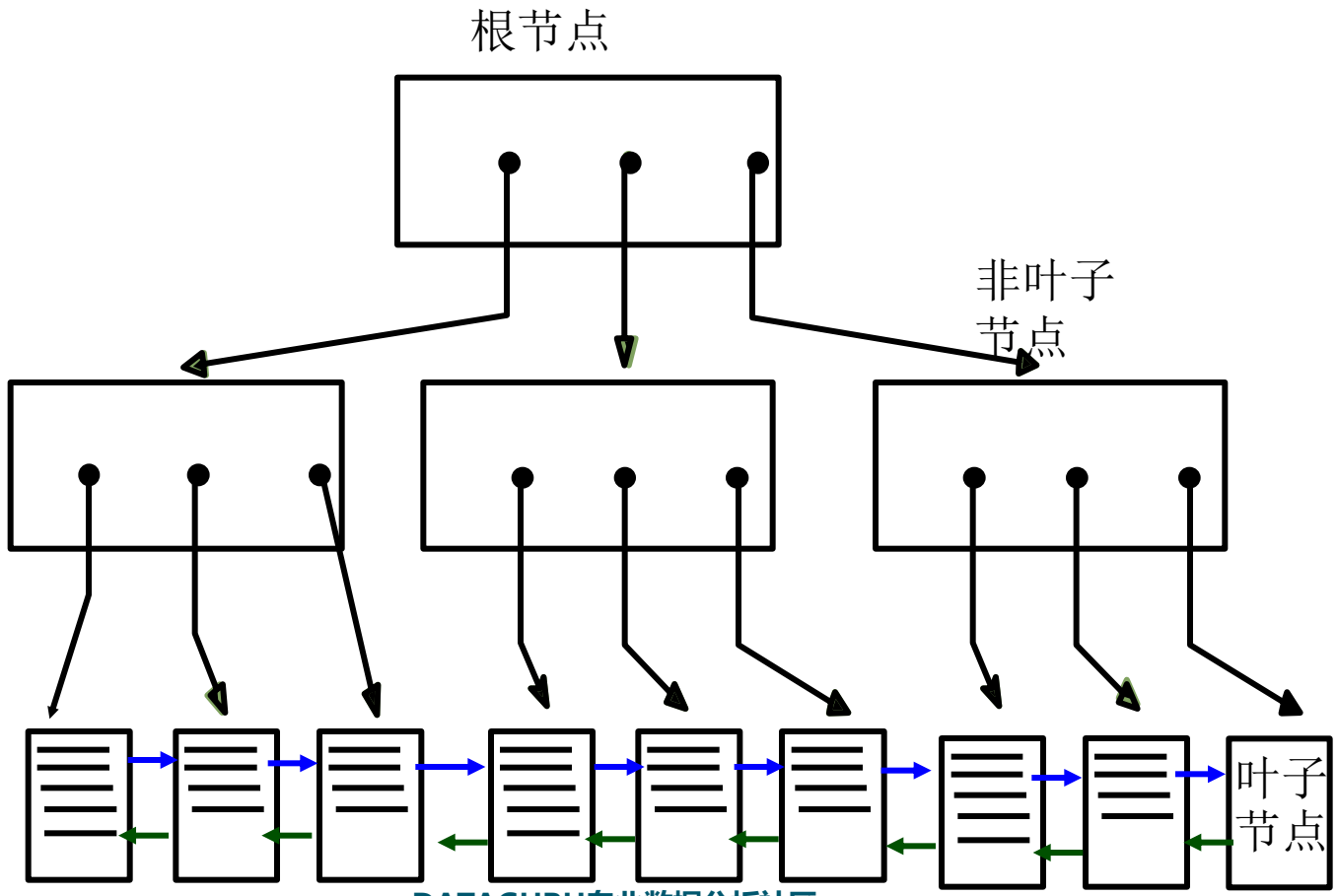
# 完全索引扫描(Index Only Access)和包含列(INCLUDE Columns)

```
CREATE UNIQUE INDEX XACCT ON ACCT
(ACCT_GRP ASC, ACCT_ID DESC)
INCLUDE (ACCT_NAME, ADDRESS)
```

- CREATE INDEX 命令允许在唯一索引的键后面加上额外的数据列
- 这些列作为索引的一部分，但不是键的组成
- 一些查询只需访问这样的索引而避免访问基本表，从而获得更好的性能
- 比如：SELECT acct_name, address FROM acct WHERE acct_grp = ? AND acct_id = ?
- 注意：当需要更新这些包含列时，会给索引的维护带来额外的开销

# 双向索引

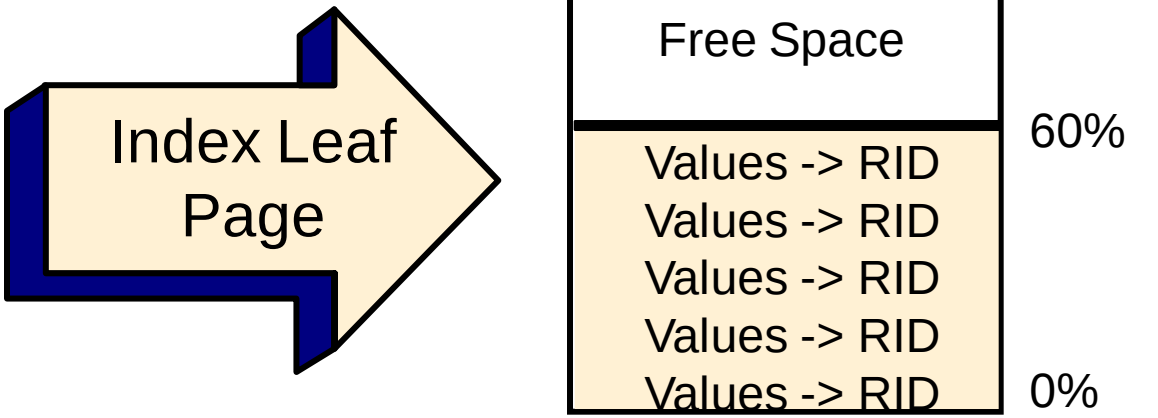
```
CREATE INDEX IX1
...ALLOW REVERSE SCANS
```



# 索引的空闲空间(Free Space)

```
CREATE INDEX L_PK ON LINEITEM (L_PARTKEY)
       PCTFREE 40
```

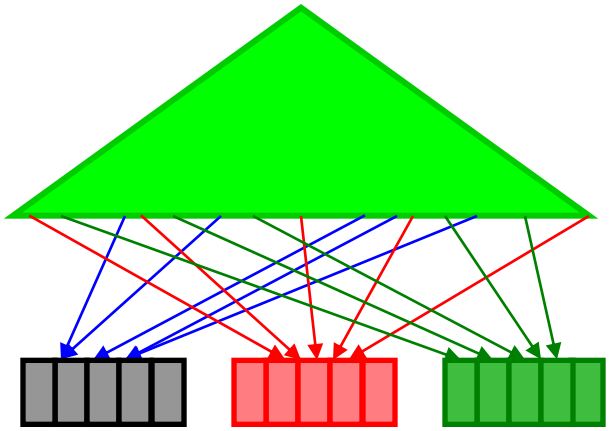
PCTFREE 40
*(40 = 40% 空闲)*



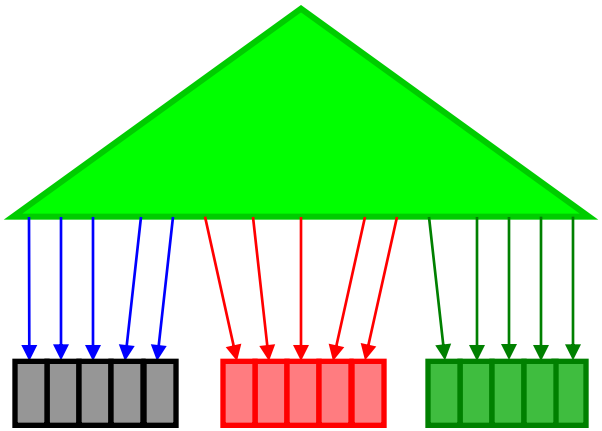

- 避免索引页的分裂，可在索引维护时获取更好的性能
- 这些空间在维护索引或者REORG时使用
- 默认设置为10%
- 对只读表，将PCTFREE设置为0

# 非聚集索引 vs 聚集索引

CREATE INDEX IX1 (product_id)



CREATE INDEX IX2 (product_id) CLUSTER



# 聚集索引

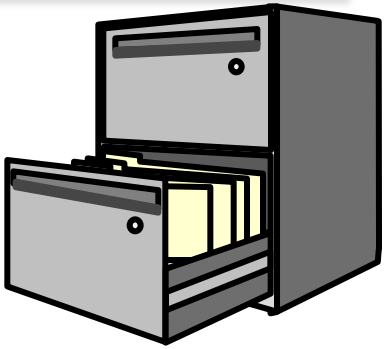
- 让数据记录在物理页上的存储顺序基于聚集索引的顺序
- DB2 在输入新的记录时尽量维护这种聚集
- 利用表的PCTFREE设置，可以为建有聚集索引的表的数据插入预留空间
- 这可以在CREATE TABLE指定或ALTER TABLE命令实现
- 例如:
  - `CREATE INDEX CL_INX ON LINEITEM (L_SHIPDATE) CLUSTER`
  - `ALTER TABLE LINEITEM PCTFREE 30`

# 索引的利用率 - 组合索引

| LASTNAME | FIRSTNAME | AGE | RID |
|---|---|---|---|
| Abell | Jim | 40 | ↑ |
| ... | ... | ... | ... |
| Cambell | Debbie | 32 | ↑ |
| ... | ... | ... | ... |
| Chang | David | 45 | ↑ |
| ... | ... | ... | ... |
| Free | Ann | - - | ↑ |
| Free | Bernie | | ↑ |
| Free | David | 35 | ↑ |
| Free | Debbie | 39 | ↑ |
| Free | Dennis | 37 | ↑ |
| Free | Diane | 41 | ↑ |
| Free | Donald | 33 | ↑ |
| Free | Donna | 42 | ↑ |
| Free | Doreen | 36 | ↑ |
| Free | Doug | - - | ↑ |
| ... | ... | ... | ... |
| Klonne | Dennis | 42 | ↑ |
| ... | ... | ... | ... |

1. SELECT *
   WHERE
   LASTNAME = 'Free'
2. SELECT LASTNAME
   WHERE
   FIRSTNAME = 'David'
3. SELECT *
   WHERE
   LASTNAME = 'Free' AND
   FIRSTNAME > 'Diane'
4. SELECT *
   WHERE
   LASTNAME = 'Free' AND
   AGE > 38
5. SELECT ...
   WHERE ...
   GROUP BY LASTNAME
   -or-
   ORDER BY LASTNAME

# 索引的统计信息维护



- 索引的统计信息(SYSCAT.INDEXES)
  - CLUSTERRATIO

    或
  - CLUSTERFACTOR
  - PAGE_FETCH_PAIRS
- 执行RUNSTATS来搜集索引的详细统计信息

```
db2 RUNSTATS ON TABLE USERID.ACCT
       AND DETAILED INDEXES ALL
```

  - 或者

```
db2 RUNSTATS ON TABLE USERID.ACCT
       AND SAMPLED DETAILED INDEXES ALL
```

# 索引的额外开销

- 对INSERT，UPDATE，DELETE，LOAD等操作增加了额外的CPU和IO开销
- 由于索引给优化器带来更多选择，会增加查询的编译时间
- 更多的磁盘空间
- 更多的备份/恢复时间

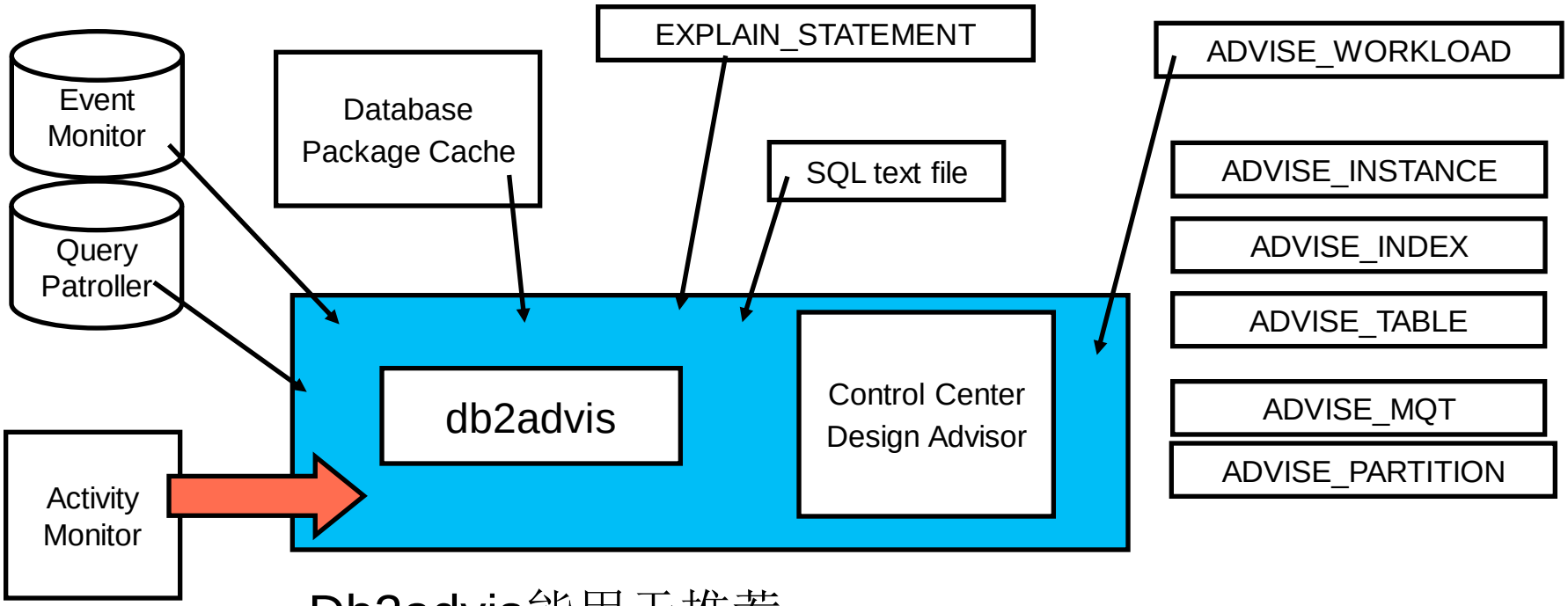
# 索引的最佳实践

- 对所有的主键PK和绝大多少外键FK建立索引。因为大部分JOIN操作都发生在主键和外键之间，所以对PK和FK建索引对性能很关键
- FK外键上的索引也能提高引用完整性（RI）约束的检查
- 为WHERE条件中常引用的列建立索引，但是<>这种谓词除外。也就是说范围谓词和等于谓词引用的列是更好的选择
- 组合索引的列数目一般不大于5
- 避免建重复的索引

# 提纲

- 索引的优势
- 索引在DB2内核中的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

# 设计向导概要




Db2advis能用于推荐：

新的索引

新的物化视图(MQTs)

多维集群表(MDC)

表的重新分区 of tables

根据特定的工作负载删除未使用的索引和MQT

# 使用Design Advisor

命令行使用方法

```
db2advis -d sample -m MICP -i da.sql
```

Create Explain Tables first

-d database name

-m M-MQT I-Indexes C-MDC tables P-Partitioning

Workload type keyword: (choose one)

-s Single SQL statement

-i SQL from input file

-qp SQL from Query Patroller table

-w SQL from ADVISE_WORKLOAD table by workload name

-g Get workload from dynamic SQL snapshot

Other keywords:

-l number of MB available for indexes and MQTs (-1 for unlimited)

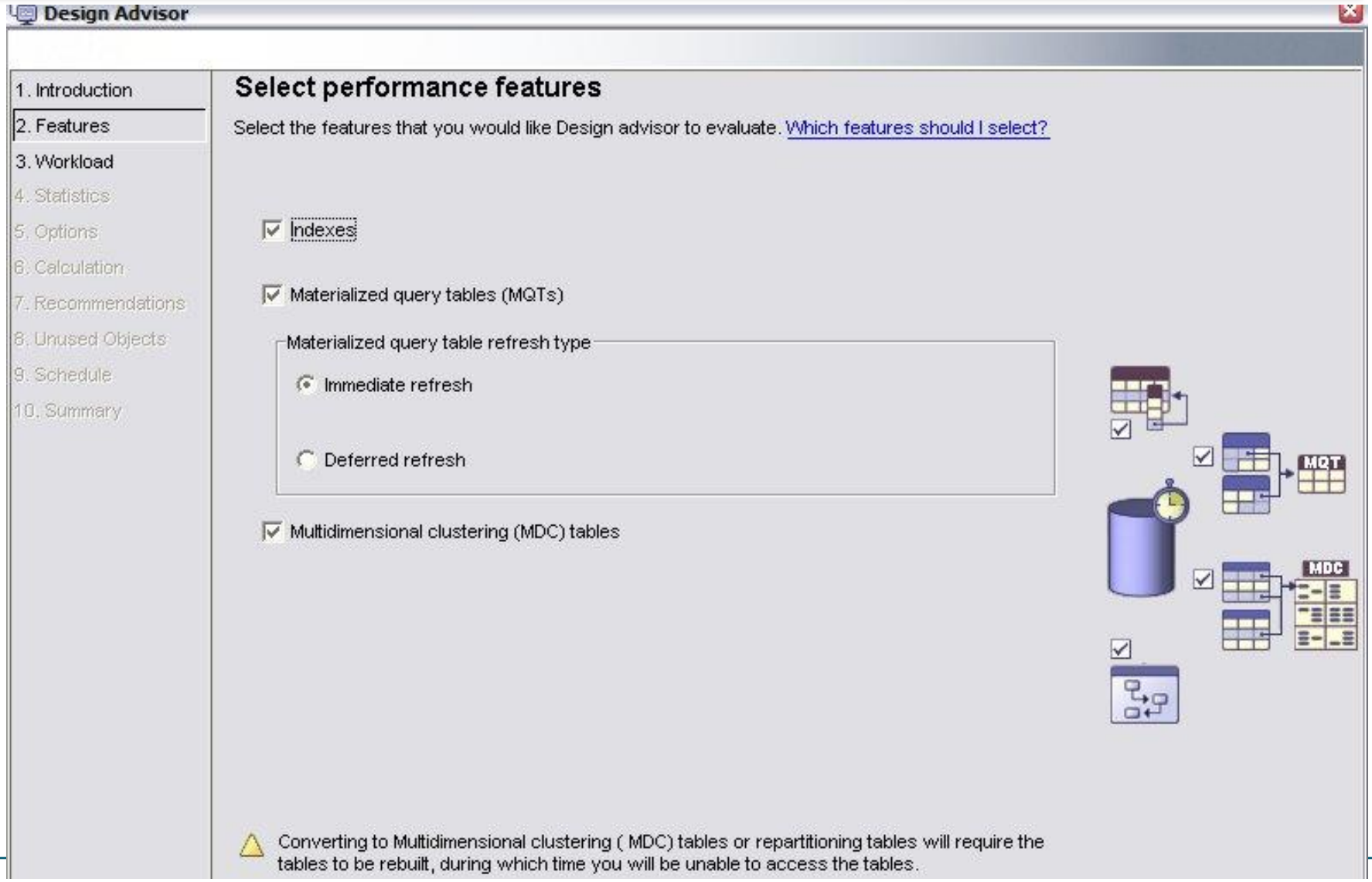
-t specifies the maximum time, in minutes, to complete the operation

# 利用db2advis推荐索引

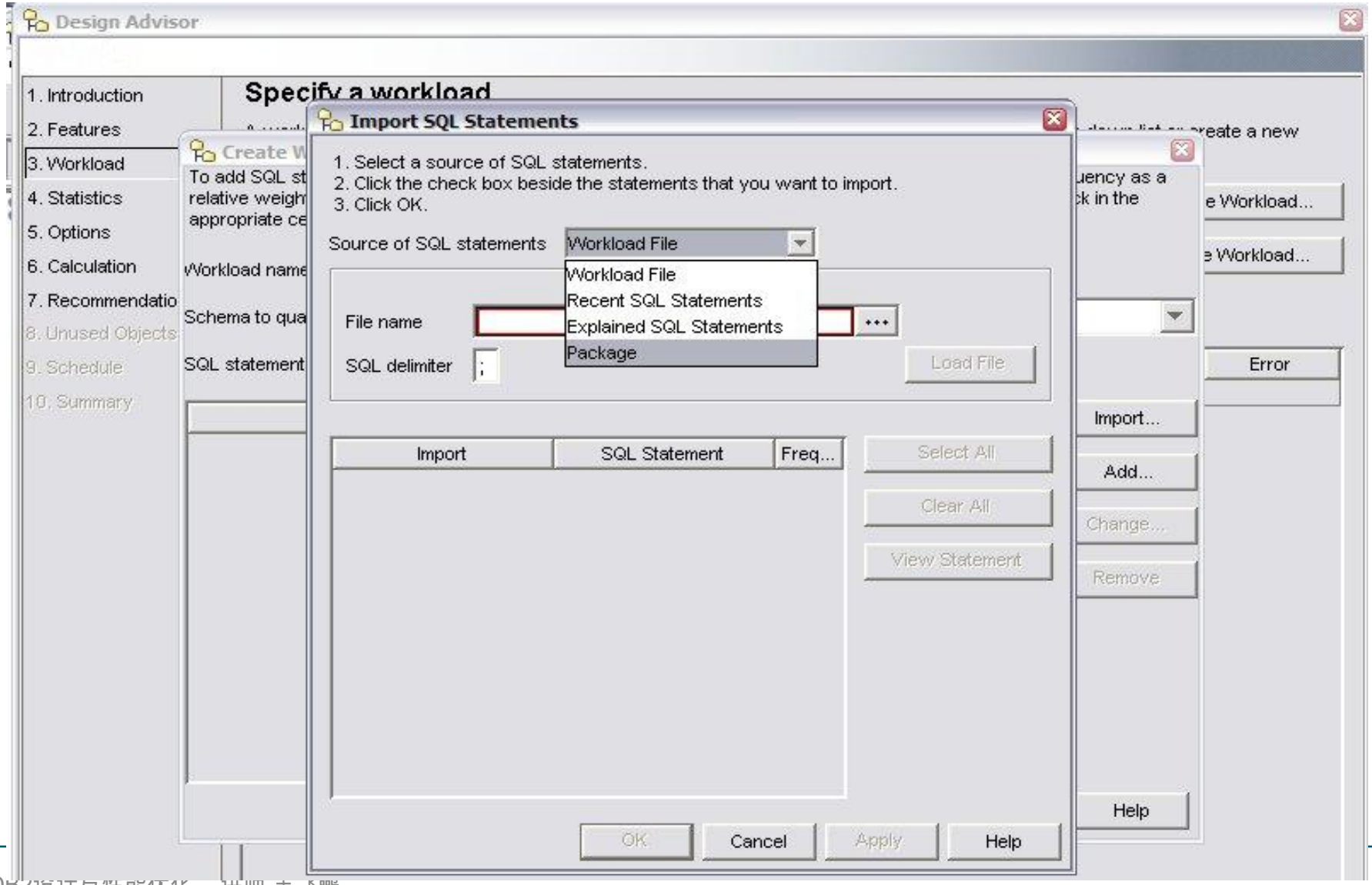
```
db2advis -d tpcds -i cost.sql -disklimit 2 -o newindex.ddl > adviseout.txt
```

```
execution started at timestamp 2012-05-17-11.54.08.236000
found [3] SQL statements from the input file
Recommending indexes...
total disk space needed for initial set [   0.817] MB
total disk space constrained to         [   2.000] MB
Trying variations of the solution set.
Optimization finished.
  2  indexes in current solution
 [1118.4197] timerons  (without recommendations)
 [362.6014] timerons  (with current solution)
 [67.58%] improvement

-- LIST OF RECOMMENDED INDEXES

CREATE INDEX "DB2ADMIN"."IDX509062043470000"
ON "TPCD    "."LINEITEM"
("L_RETURNFLAG" ASC, "L_DISCOUNT" ASC,
 "L_EXTENDEDPRICE" ASC, "L_ORDERKEY" ASC)
ALLOW REVERSE SCANS ;

RUNSTATS ON TABLE "TPCD    "."LINEITEM"
FOR INDEX "DB2ADMIN"."IDX509062043470000" ;

CREATE UNIQUE INDEX "DB2ADMIN"."IDX509062044160000"
ON "TPCD    "."ORDERS"
("O_ORDERDATE" ASC, "O_ORDERKEY" ASC, "O_CUSTKEY" ASC)
ALLOW REVERSE SCANS ;

RUNSTATS ON TABLE "TPCD    "."ORDERS"
FOR INDEX "DB2ADMIN"."IDX509062044160000" ;
```

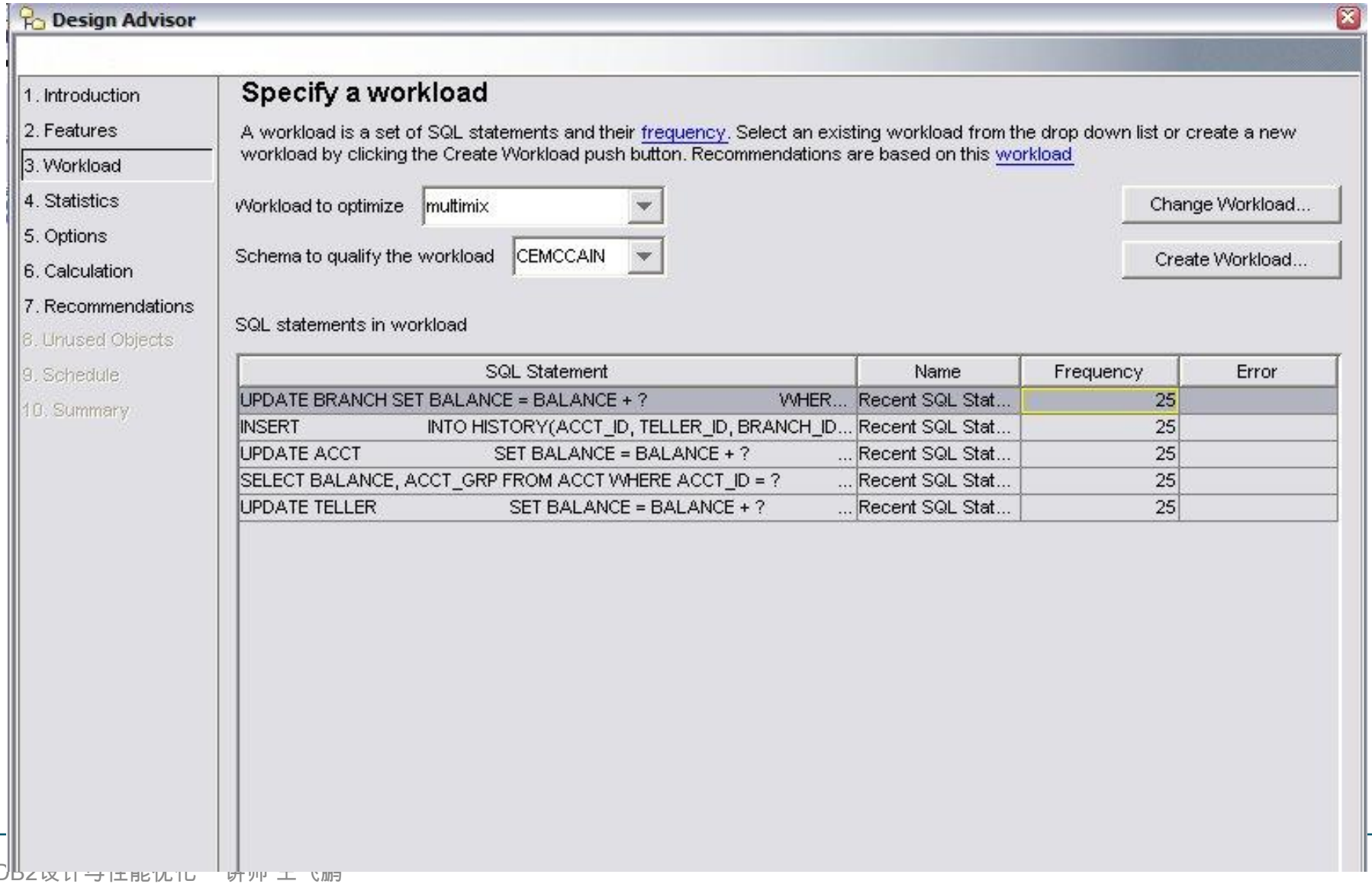
# 设计向导的图形化界面


Design Advisor
1. Introduction
2. Features
3. Workload
4. Statistics
5. Options
6. Calculation
7. Recommendations
8. Unused Objects
9. Schedule
10. Summary
Select performance features
Select the features that you would like Design advisor to evaluate. Which features should I select?
Indexes
Materialized query tables (MQTs)
Materialized query table refresh type
Immediate refresh
Deferred refresh
Multidimensional clustering (MDC) tables
MQT
MDC
Converting to Multidimensional clustering ( MDC) tables or repartitioning tables will require the tables to be rebuilt, during which time you will be unable to access the tables.

# 设计向导 – 工作负载定义


Design Advisor
1. Introduction
2. Features
3. Workload
4. Statistics
5. Options
6. Calculation
7. Recommendatio
8. Unused Objects
9. Schedule
10. Summary
Specify a workload
Import SQL Statements
1. Select a source of SQL statements.
2. Click the check box beside the statements that you want to import.
3. Click OK.
Source of SQL statements
Workload File
Workload File
Recent SQL Statements
Explained SQL Statements
Package
File name
SQL delimiter
;
Load File
Import
SQL Statement
Freq...
Select All
Clear All
View Statement
OK
Cancel
Apply
Help
Import...
Add...
Change...
Remove
Error
Help

# 设计向导 – 工作负载配置


Design Advisor
1. Introduction
2. Features
3. Workload
4. Statistics
5. Options
6. Calculation
7. Recommendations
8. Unused Objects
9. Schedule
10. Summary
Specify a workload
A workload is a set of SQL statements and their frequency. Select an existing workload from the drop down list or create a new workload by clicking the Create Workload push button. Recommendations are based on this workload
Workload to optimize
multimix
Change Workload...
Schema to qualify the workload
CEMCCAIN
Create Workload...
SQL statements in workload
SQL Statement
Name
Frequency
Error
UPDATE BRANCH SET BALANCE = BALANCE + ? WHER... Recent SQL Stat... 25
INSERT INTO HISTORY(ACCT_ID, TELLER_ID, BRANCH_ID... Recent SQL Stat... 25
UPDATE ACCT SET BALANCE = BALANCE + ? ... Recent SQL Stat... 25
SELECT BALANCE, ACCT_GRP FROM ACCT WHERE ACCT_ID = ? ... Recent SQL Stat... 25
UPDATE TELLER SET BALANCE = BALANCE + ? ... Recent SQL Stat... 25

# 设计向导 – 选择限制和选项

Design Advisor

1. Introduction
2. Features
3. Workload
4. Statistics
5. Options
6. Calculation
7. Recommendations
8. Unused Objects
9. Schedule
10. Summary

## Specify recommendation options

You can choose to set and specify maximum disk space for recommended objects. For materialized query tables, you can select the table space and schema, and whether to derive statistics by sampling.

☑ Set maximum disk space for the recommended objects 50 MB

Hint: Specifying a disk limit might reduce the number of recommended objects to fit in the given disk space.

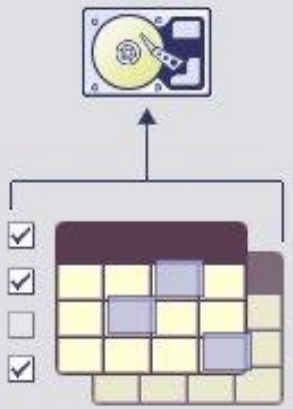
Materialized query table options

Table space: AUTOMATIC SELECT

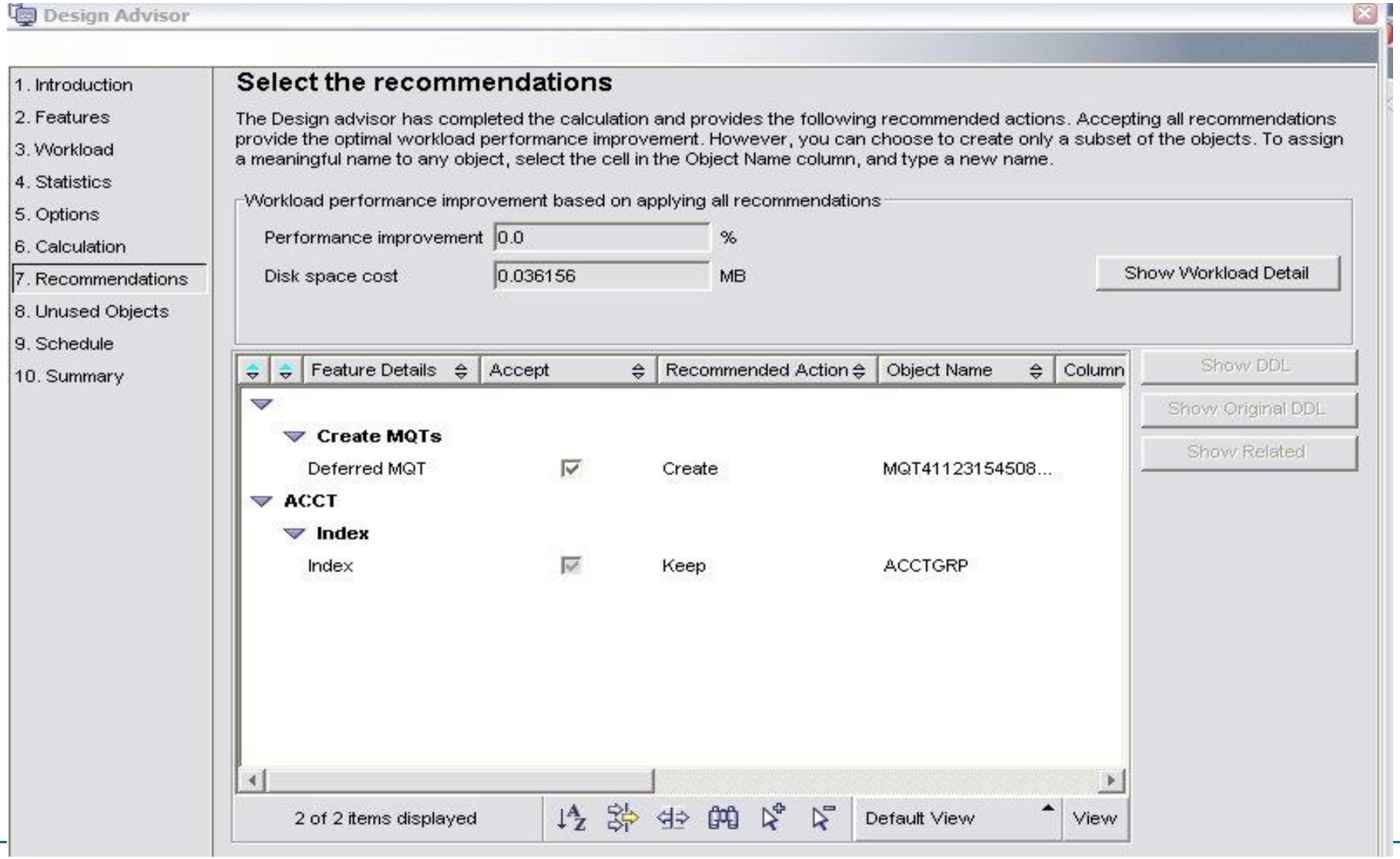
Schema name: AUTOMATIC SELECT

☐ Derive statistics by sampling

Hint: If you select to derive statistics by sampling, the Design advisor can produce better MQT recommendations, but the calculations will take longer to complete.



# 设计向导 – 建议

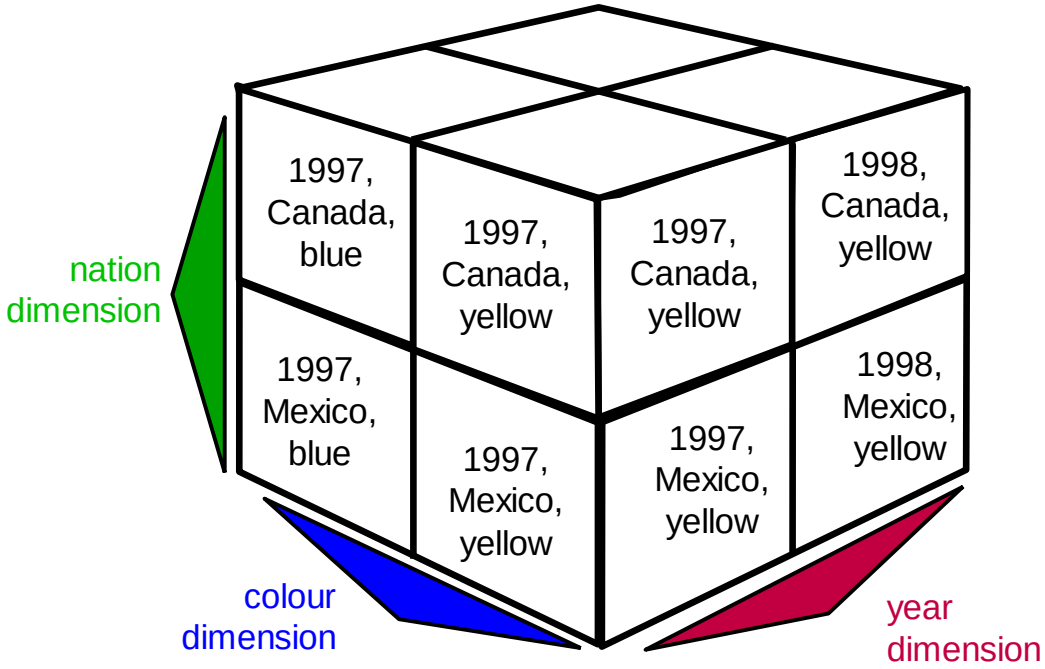

Design Advisor
1. Introduction
2. Features
3. Workload
4. Statistics
5. Options
6. Calculation
7. Recommendations
8. Unused Objects
9. Schedule
10. Summary
Select the recommendations
The Design advisor has completed the calculation and provides the following recommended actions. Accepting all recommendations provide the optimal workload performance improvement. However, you can choose to create only a subset of the objects. To assign a meaningful name to any object, select the cell in the Object Name column, and type a new name.
Workload performance improvement based on applying all recommendations
Performance improvement
0.0
%
Disk space cost
0.036156
MB
Show Workload Detail
Feature Details
Accept
Recommended Action
Object Name
Column
Create MQTs
Deferred MQT
Create
MQT41123154508...
ACCT
Index
Index
Keep
ACCTGRP
Show DDL
Show Original DDL
Show Related
2 of 2 items displayed
Default View
View

# 提纲

- 索引的优势
- 索引在DB2内核中的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

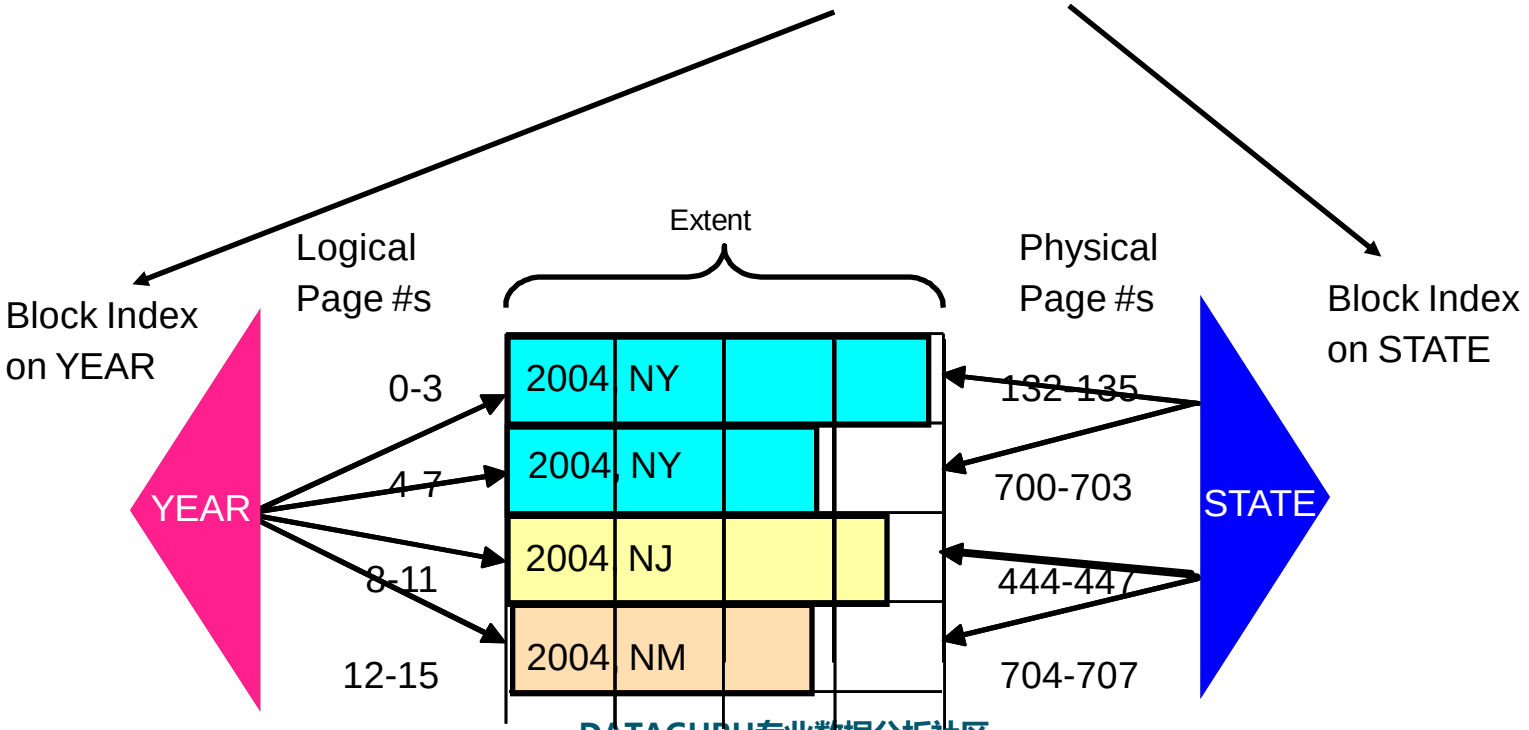
# 多维聚集表

- 每一个维度就是一个坐标轴， MDC的数据按照这个方式在磁盘上物理存放
- 示例: 3维MDC – nation, colour 和year

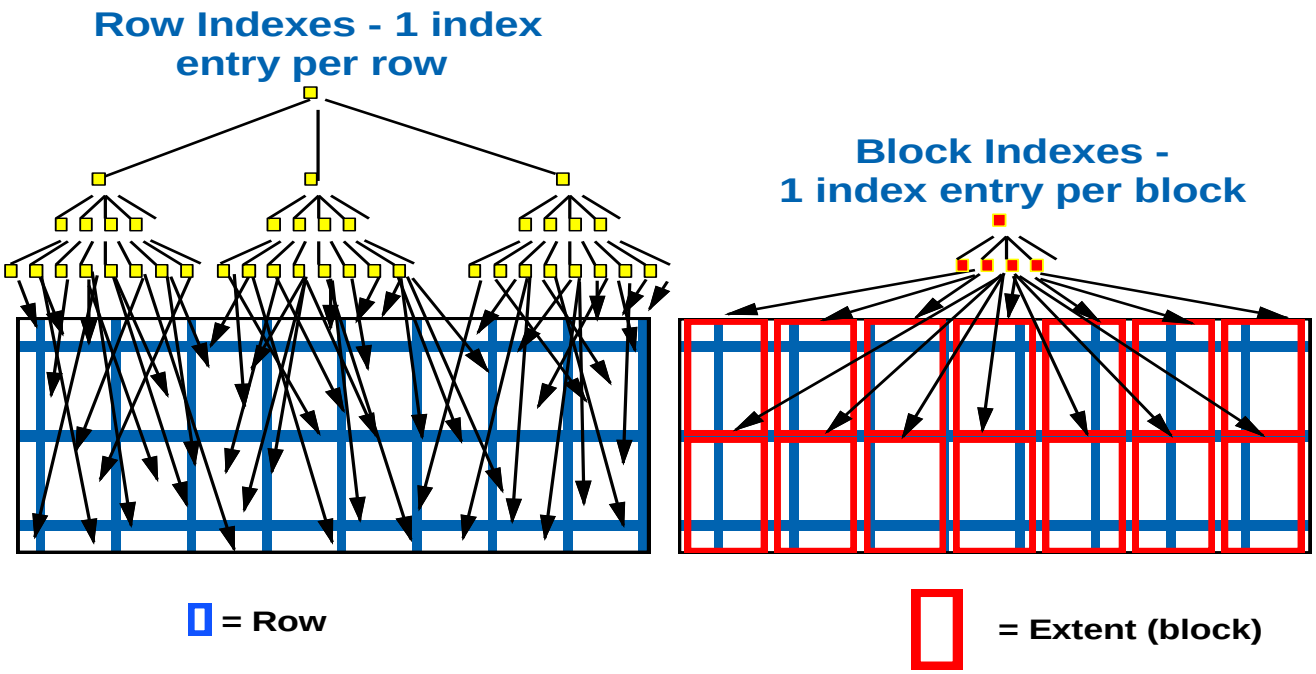


# MDC与块索引

```
CREATE TABLE MDCTABLE (
       YEAR INT,
       STATE CHAR (2),
       SALES INT,
...)
ORGANIZE BY (YEAR, STATE)
```



# 行索引 vs 块索引


Row Indexes - 1 index entry per row
Block Indexes - 1 index entry per block
= Row
= Extent (block)

# MDC与聚集索引的区别

| MDC | 聚集索引 |
| --- | --- |
| 表上的数据在数据块中总是按MDC的维度列组织(强制聚集) | 减少但不能消除reorg的需求 (非强制的聚集)。 良好的PCTFREE设置可更好的维护聚集度。 |
| 数据在多个维度上聚集 | 数据在一个维度聚集 |
| 块索引(每维度一个块索引) | 行索引(每个表只能有一个聚集索引) |
| Reorg能释放不再使用的块，并合并部分填充的块 | Reorg基于聚集索引来排序表中数据行的物理存储 |
| 为避免大量不必要的空间分配，维度的粒度应该适度粗（维度列的基数较低） | 聚集列的粒度比MDC细 |
| 如果选择了不合适的维度，可能增加表在磁盘上的占用空间 | 不影响表在磁盘上的大小，除非你为了维护聚集而增大PCTFREE |

# 维度选择的考虑

- 在设计MDC表选择维度列，需要考虑以下几点:
  - 那些查询能受益于块级聚集
    - 查询中等于或范围谓词的列
    - 粗粒度的列
    - 事实表（fact table）上的外键列
  - 块单元格的密度
    - 可能的单元格数据 = 维度列的笛卡尔乘积
    - 稀疏单元格：某些块可能只存储很少的几行数据
  - 可以操作的三个方面：
    - 区间大小(extent size) – 当稀疏单元格太多，减小extent大小
    - 维度列的数目
    - 利用生成列将某个维度提升到更高的粒度，如month(s_date)

# 基于生成列的MDC维度

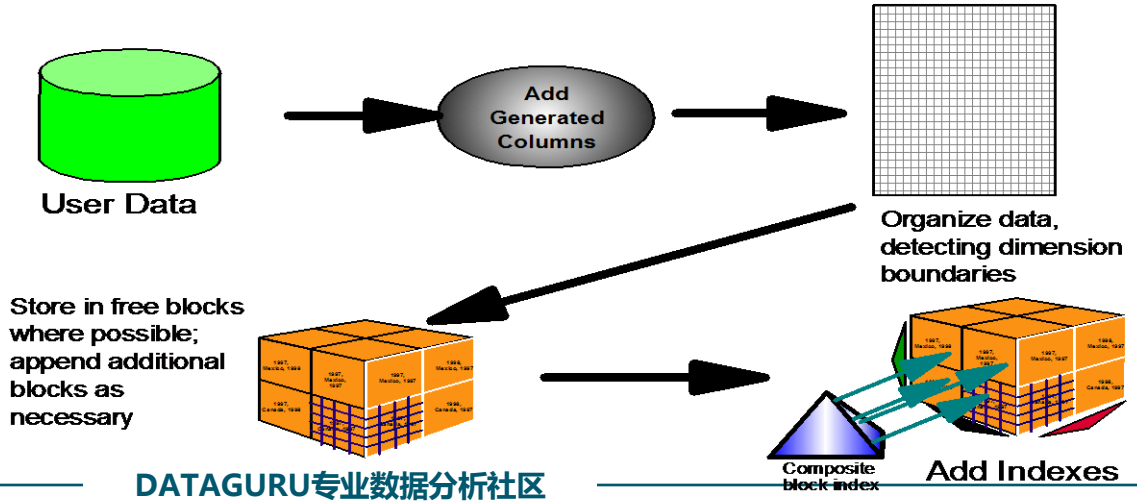
- 一个维度能基于生成列(GENERATED)创建
- 例如:

```
CREATE TABLE MDCTABLE
            ( Date DATE,
              Nation CHAR(25),
              Color VARCHAR(10),
              Month generated always as (Month(Date)),
              ... )
          ORGANIZE BY( Month, Color )
```

- 这样，MDC表将拥有两个粒度适中的维度

# MDC表的数据转入

- 利用DB2 LOAD做快速数据转入
  - 高效算法来根据各维度组织数据
  - 更少的日志和更少的块索引更新
- MDC Load展示更好的空间管理
  - 转入新的数据片能利用块中原有的空闲页
  - Load能利用块映射迅速找到空闲块
- 在线或者离线



# MDC设计最佳实践

- 设计MDC考虑的因素：
  - 调整维度的数目
  - 调整某维度的粒度(利用生成列)
  - 调整块(区间extent)的大小以及页大小
- 利用设计向导来推荐可能的MDC设计，从而降低应用负载的查询代价
  - 设计向导的图形化界面（控制中心-db2cc）
  - **db2advis -d mdb -i workload.sql -m C -o advise1.out**

# MDC设计向导(1 of 2)

```
execution started at timestamp 2005-09-26-16.55.00.513000
found [4] SQL statements from the input file
Recommending Multi-Dimensional Clusterings...
total disk space needed for initial set [   0.035] MB
total disk space constrained to         [   5.855] MB

Note: MDC selection in the DB2 Design Advisor requires the target database
      to be populated with a data sample. This sample is used for estimating
      the number and density of MDC cells in any MDC solution that the
      Design Advisor will recommend. If your database is empty the
      Design Advisor will not recommend MDC.

Prioritizing Multi-dimensional Clustering candidate tables...
Multi-dimensional Clustering candidate tables, in priority sequence:

Table 0: HIST1,
   number of pages 2016,
   block size 16
There are 1 candidate tables considered for Multi-dimensional Clustering conversion

Searching the multidimensional space for solutions for HIST1...

Percentage of search points visited...
□□□0% □□□100

2 clustering dimensions in current solution
[8665.0000] timerons (without any recommendations)
[5108.7624] timerons (with current solution)
[41.04%] improvement
```

# MDC设计向导(2/2)

```
--
--
-- LIST OF MODIFIED CREATE-TABLE STATEMENTS WITH RECOMMENDED
   PARTITIONING KEYS AND TABLESPACES AND/OR RECOMMENDED
   MULTI-DIMENSIONAL CLUSTERINGS
-- ============================
-- CREATE TABLE "MDC     "."HIST1" ( "ACCT_ID" INTEGER NOT NULL ,
--     "TELLER_ID" SMALLINT NOT NULL ,
--     "BRANCH_ID" SMALLINT NOT NULL ,
--     "BALANCE" DECIMAL(15,2) NOT NULL ,
--     "DELTA" DECIMAL(9,2) NOT NULL ,
--     "PID" INTEGER NOT NULL ,
--     "TSTMP" TIMESTAMP NOT NULL WITH DEFAULT  ,
--     "ACCTNAME" CHAR(20) NOT NULL ,
--     "TEMP" CHAR(6) NOT NULL   ,
--    MDC509262055030000 GENERATED ALWAYS AS ( (SMALLINT(TELLER_ID-(2))/(32))) )
--    IN "MDCTSP1"
-- ORGANIZE BY (
--    MDC509262055030000,
--    BRANCH_ID )
-- ;
-- COMMIT WORK ;
```

# 提纲

- 索引的优势
- 索引在DB2内核的工作方式
- 索引设计与性能
- 索引设计向导
- MDC与块索引
- 小结与练习

# 小结

- 索引对性能至关重要，可以避免全表扫描，减少IO，避免排序等。
- 索引在DB2查询计划中有多种存在形式：
  - IXSCAN – FETCH
  - INDEX - ONLY
  - LIST PREFETCH
  - Index ANDing
  - Index ORing
- 索引设计在DB2中技术含量高
  - 唯一索引的包含列可以带来INDEX-ONLY扫描
  - 聚集索引给性能带来好处，能减少IO
  - 索引空闲空间的设置对索引维护意义重大
  - 索引的统计信息应用runstats命令定期维护
- 索引设计向导是个不错的工具，针对某些SQL语句推荐有用的索引
- 对于MDC和块索引，维度列的选择非常关键。

# 炼数成金逆向收费式网络课程

- 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