DTCC

数，造，未，来

第十二届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2021

2021 年 10 月 18 日 - 20 日 | 北京国际会议中心

Spark
Hadoop
OLAP
HANA
MPP

IT168.com ChinaUnix.net ITPUB

inspur 浪潮 | 云溪数据库

inspur 浪潮 | 云溪数据库

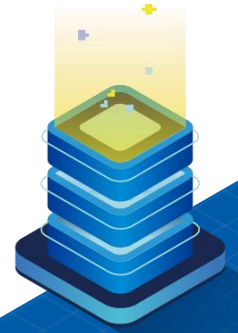
工业级SQL优化器技术蓝图

赵衍衍

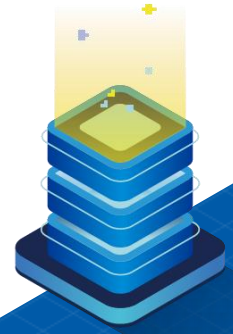
西电-浪潮数据库创新实验室

浪潮科学研究院分布式数据库研究所

目 录



目 录



传统查询优化

- **SQL是一种声明式语言，而非过程式语言！**
  - 用户指定数据访问和计算的目的 – what
  - 并非访问数据的算法 – not how
  - 将（最优）算法的选择权留给查询优化器 – 访问计划

```
select o.date, o.quantity, p.name
from customers c, orders o, products p
where c.contactNumber = ?
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
order by o.date
```

⇐ Projection

⇐ (Inner) Join

associative

commutative

⇐ Filtering (predicate)

⇐ Ordering (sort)

传统查询优化

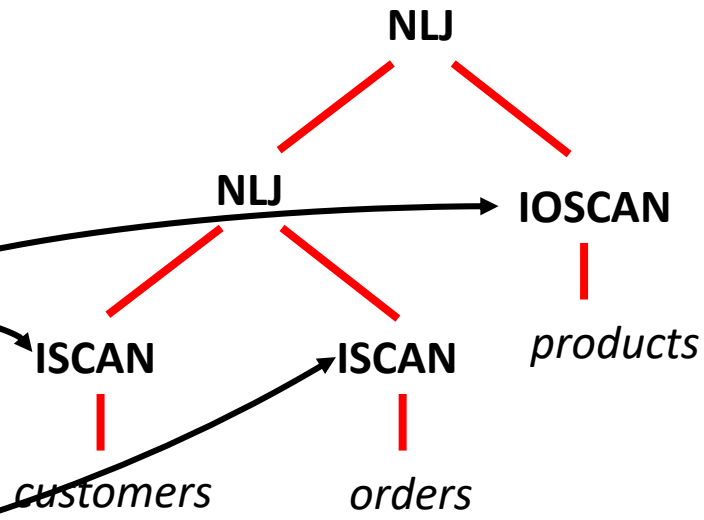
❑ 访问计划

- 访问方法: 一种数据访问的算法
  - Table Scan
  - Index Scan, Index-Only Scan等
- 连接方法: 一种连接两表的算法
  - Look-up Join
  - Merge Join
  - Hash Join
- 访问计划:为执行SQL查询语句而整合的一种访问方法和连接方法序列

一个简单的SQL查询:

```
select o.date, o.quantity, p.name
from customers c, orders o, products p
where c.contactNumber = ?
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
order by o.date
```

customers: index on contactNumber
orders: index on cid, date
products: index on id, name



传统查询优化

- **SQL是一种声明式语言，而非过程式语言！**
  - 连接操作的结合和交换特性 – 庞大的搜索空间
  - 数据访问可以是按序扫描和通过索引访问的 – 访问算法
  - 关系连接可以是对称和定向的– 连接算法

```
select o.date, o.quantity, p.name
from customers c, orders o, products p
where c.contactNumber = ?
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
order by o.date
```

⟹ 3! = 6 连接序列

假设 每张表有3个索引且底层RDBMS支持3种连接算法

⟹ 总的访问路径数：

3! X (4 x 4 x 4) x (3 x 3) = 3,456

传统查询优化

❑ SQL是一种声明式语言，而非过程式语言！

- ❑ 一般的，一个包含n表连接的查询可以有

  $n! \times ((i+1)^n) \times (j^{(n-1)})$ 种

  访问路径，假设DBMS支持*j* 种连接算法并且每张表平均有 *i* 个索引。

- ❑ 查询重写，查询并行化，其他优化技术以解锁更多的访问路径

- ❑ SQL语言查询优化本质上是一种 NP难问题！

传统查询优化

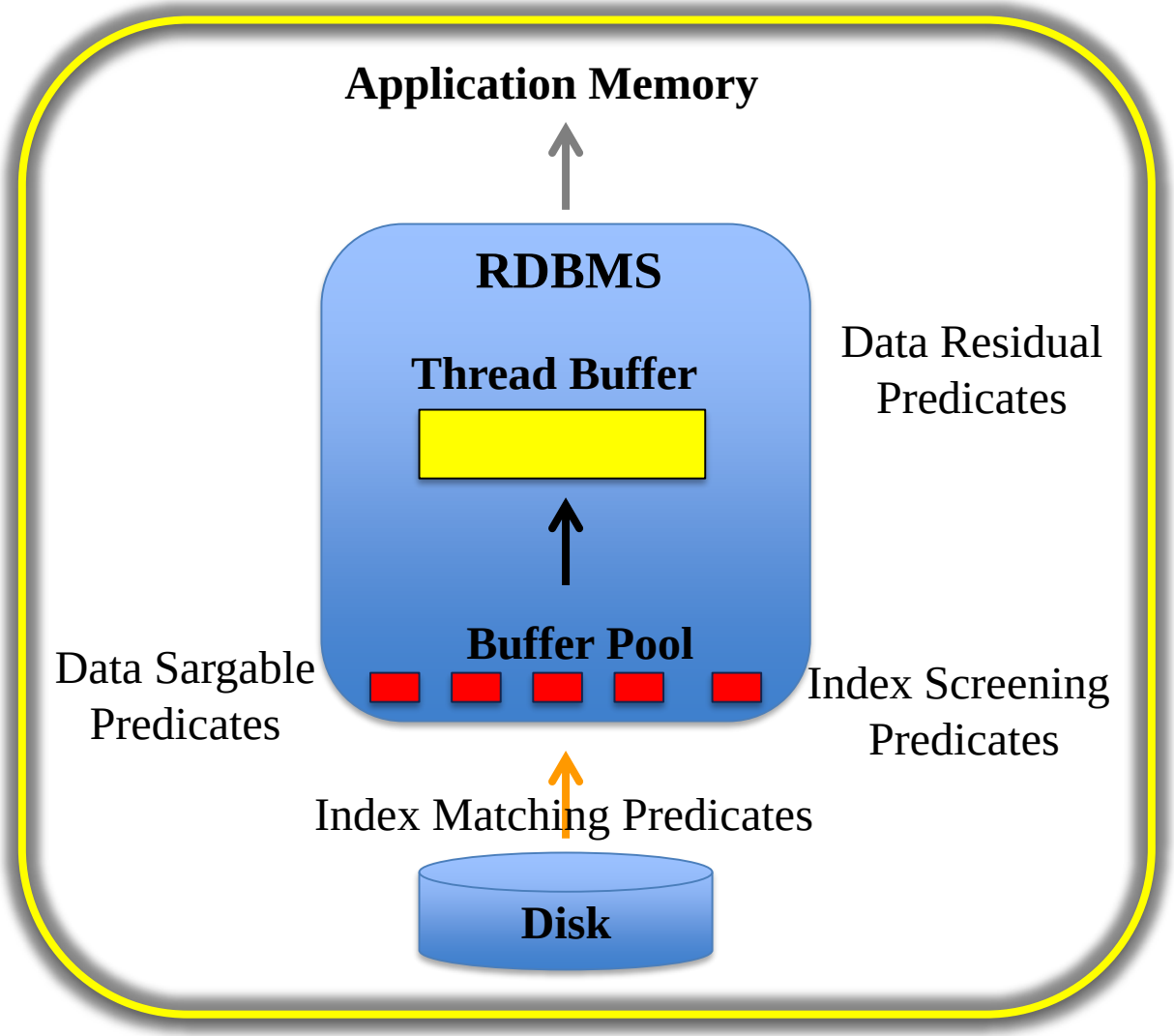
- **访问计划列举**
  - **挑战 – 庞大的搜索空间（NP难问题）**
    - 完全规范化的数据库范式
    - 工具生成的SQL语句
    - 分析查询（例如，星型模式）
  - **解决方案 – 减少搜索空间**
    - 剪枝算法
    - 遗传算法
    - 启发式算法（初始Join的选择）
    - 贪心算法
    - ……

传统查询优化

- **SQL语言查询优化本质上是一个NP难问题！**
- **一个智能查询优化器应当包含以下组件：**
  - 分词器，解析器，和语义检查
  - 查询重写
  - 访问路径枚举
  - 代价估算
  - 查询并行化 （SMP 多进程和 MPP多节点）
  - 解释器，hint机制，计划管理
  - 设计推荐 （统计，索引，分区，查询等）
  - 运行时优化
  - 可扩展优化
  - 查询调优向导

传统查询优化

- 代价估算 – 基于统计和成本公式
  - 选择率估计
    - Index matching predicate
    - Index screening predicate
    - Data sargable predicate
    - Data residual predicate
  - 基数估计
    - 索引页，索引键，记录数
    - 数据页，sargable predicates后数据记录
    - residual predicates后数据记录
  - 代价估算
    - 索引I/O和数据I/O
    - 索引CPU
    - 缓冲池中的数据CPU
    - 连接CPU



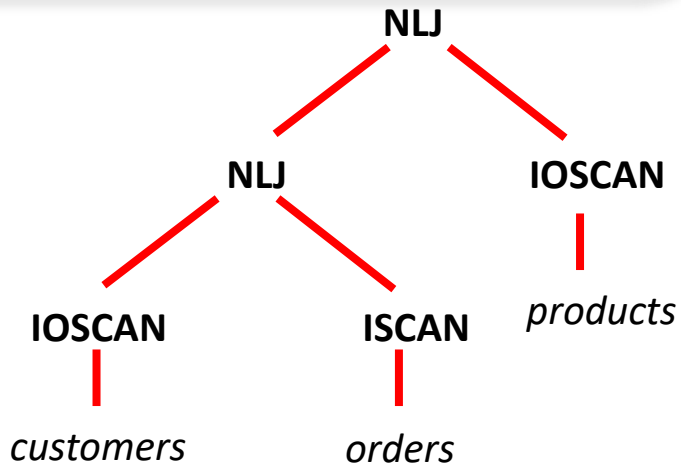
传统查询优化

❑ 代价估算– 基于统计和成本公式

```
select o.date, o.quantity, p.name
from customers c, orders o, products p
where c.contactNumber = ?
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
order by o.date
```

Column Cardinality	Filter Factor (selectivity)	cIndex Page Data Page	Index Keys Data Records
10^9	10^-9	2.3 / 1	1 / 1
10^9 / 10^8	10^-9	2.5 / 1	1 / 10
10^5 / 10^5	10^-5	1 / 1	1 / 1
2008/1 – 2018/8	???		

customers: index on contactNumber
orders: index on cid, date
products: index on id, name

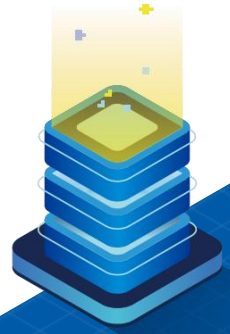


传统查询优化

❑ 代价估算 – 基于统计和代价公式

- ❑ 挑战
  - ❑ 数据倾斜：频率统计，直方图统计
  - ❑ 联合选择率：多列基数，频率，直方图
  - ❑ 变量：实时优化
  - ❑ Table表达式：统计视图
  - ❑ …
- ❑ DBA对统计数据收集了解多少?
- ❑ 成本公式能够在多大程度实现足够准确的成本估算?
- ❑ 解决方案?

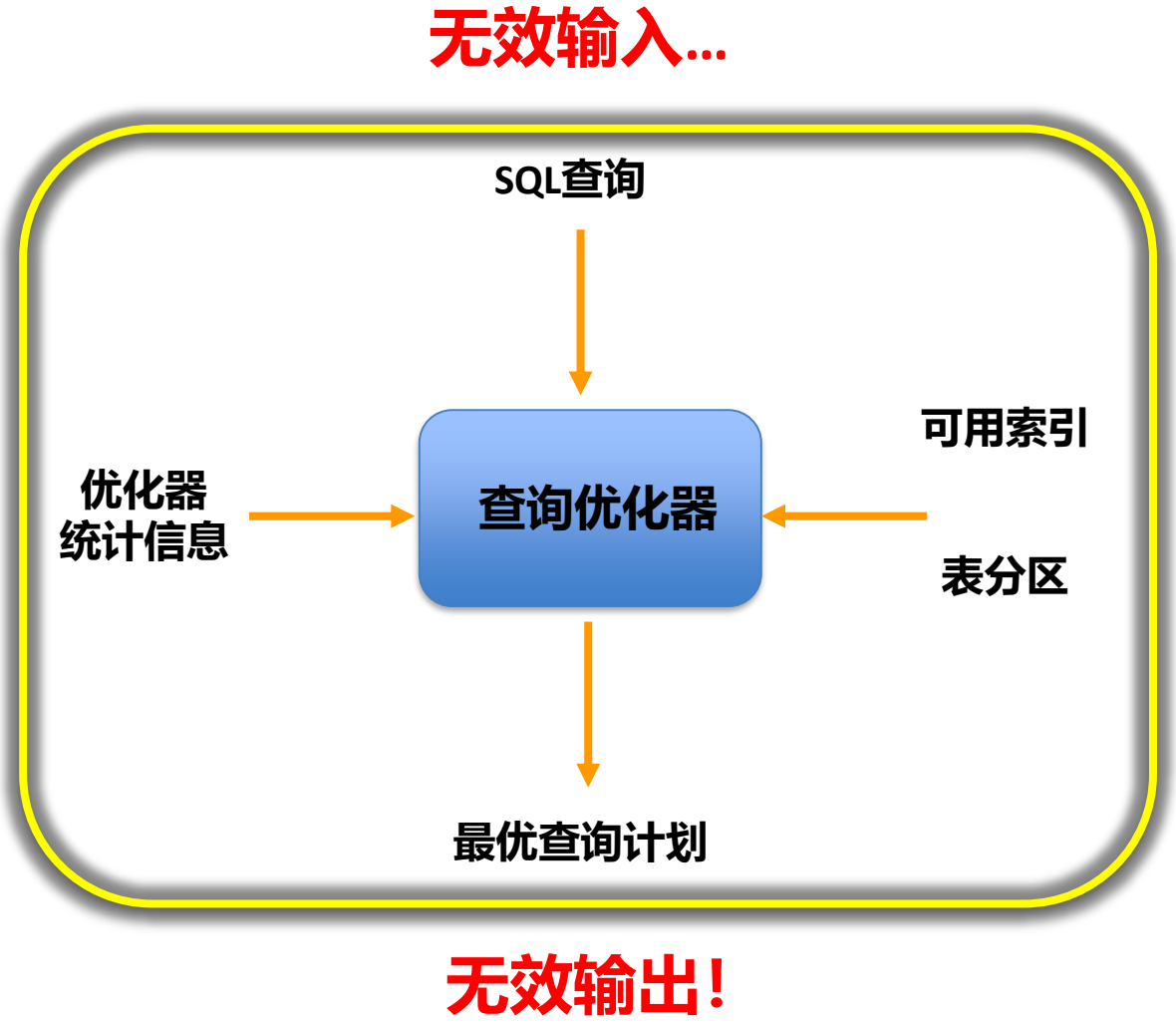
目 录



智能查询优化

❑ 管理依赖性 – 问题

- ❑ 优化器统计信息
  - ❑ 查询优化的基石
  - ❑ 收集什么?
    - ❑ 过度收集统计信息
    - ❑ 统计信息收集不足
  - ❑ 何时重新收集?
- ❑ 可用的索引
- ❑ 表分区
- ❑ 查询设计



智能查询优化

❑ 管理依赖性 - 解决方案

❑ 统计信息推荐

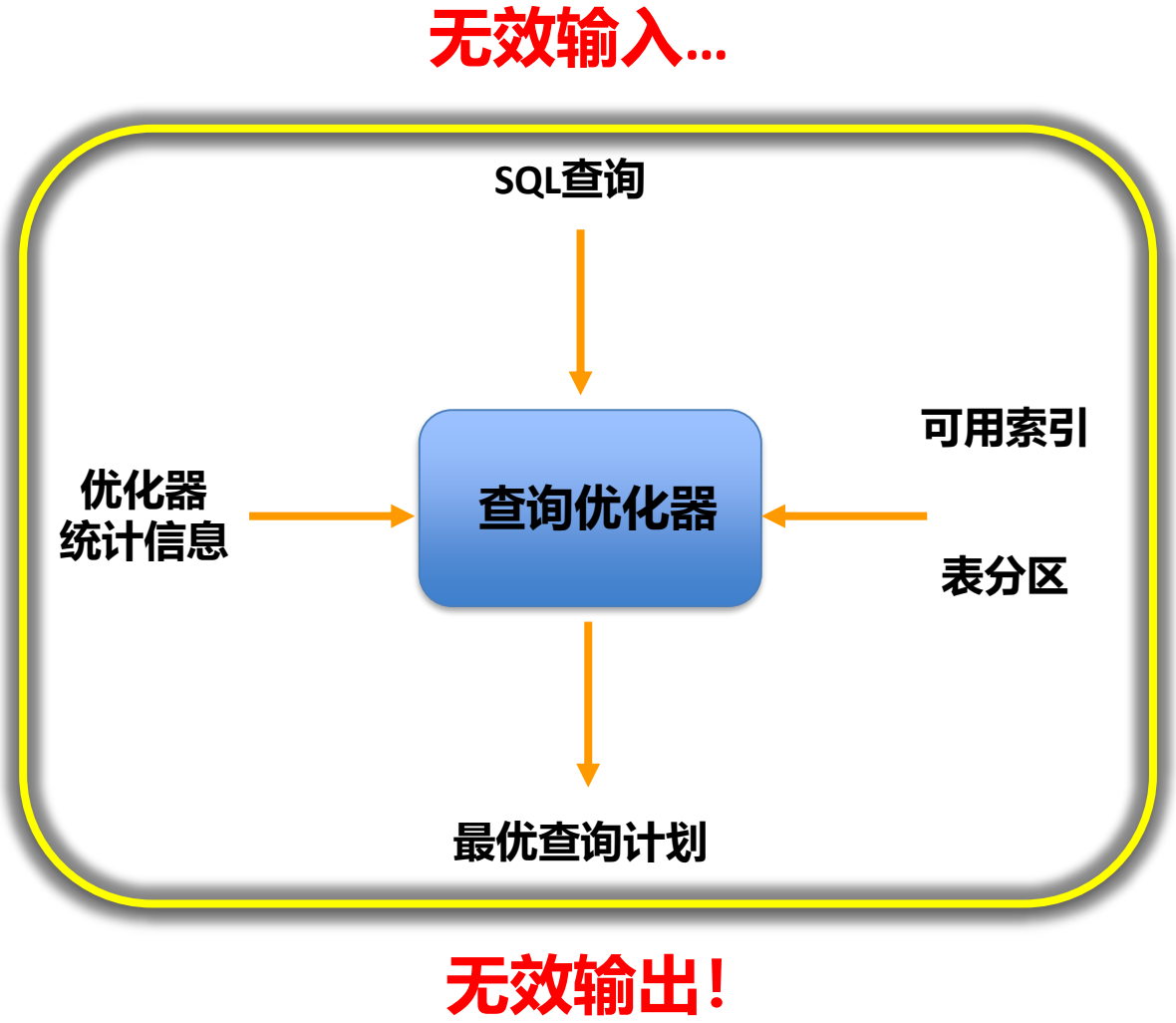
- ❑ 需要收集哪些关键统计数据?
- ❑ 时机：什么时候应该重新收集统计数据?

❑ 索引推荐

- ❑ 查询工作负载及单个查询
- ❑ 假设（what-if）分析

❑ 分区推荐

❑ 查询推荐



智能查询优化

❑ 管理依赖性 – 统计信息推荐

```
select o.date, o.quantity, p.name
from customers c, orders o, products p
where c.city in ('Wanzhou', 'Zhumadian')
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
    and p.type in ('Drink', 'Cosmetic')
order by o.date
```

customers: index on city
orders: index on cid, date
products: index on (1) id, name (2) type

Column Cardinality	Filter Factor (selectivity)	Index Page Data Page	Index Keys Data Records
500	2/500 ?	? / ?	? / ?
10^9 / 10^8	10^-9	2.5 / 1	1 / 10
10^5 / 10^5	10^-5	1 / 1	1 / 1
2008/1 – 2018/8	?	? / ?	? / ?
1000	2 / 1000 ?	? / ?	? / ?

customers join *orders*

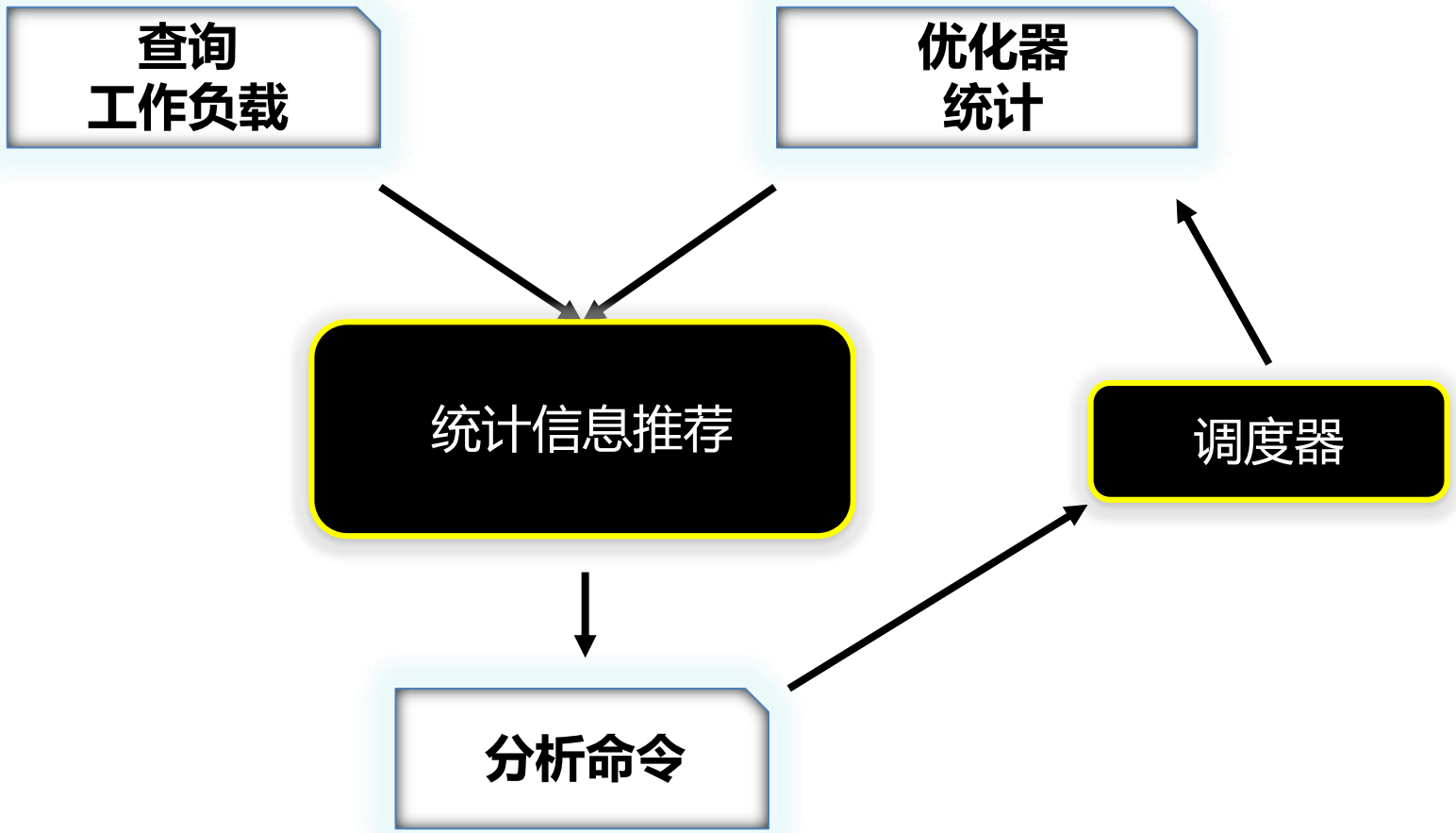
orders join *customers*

如果存在
数据倾斜怎么办?

如果存在数据相关性怎么办 ?

智能查询优化

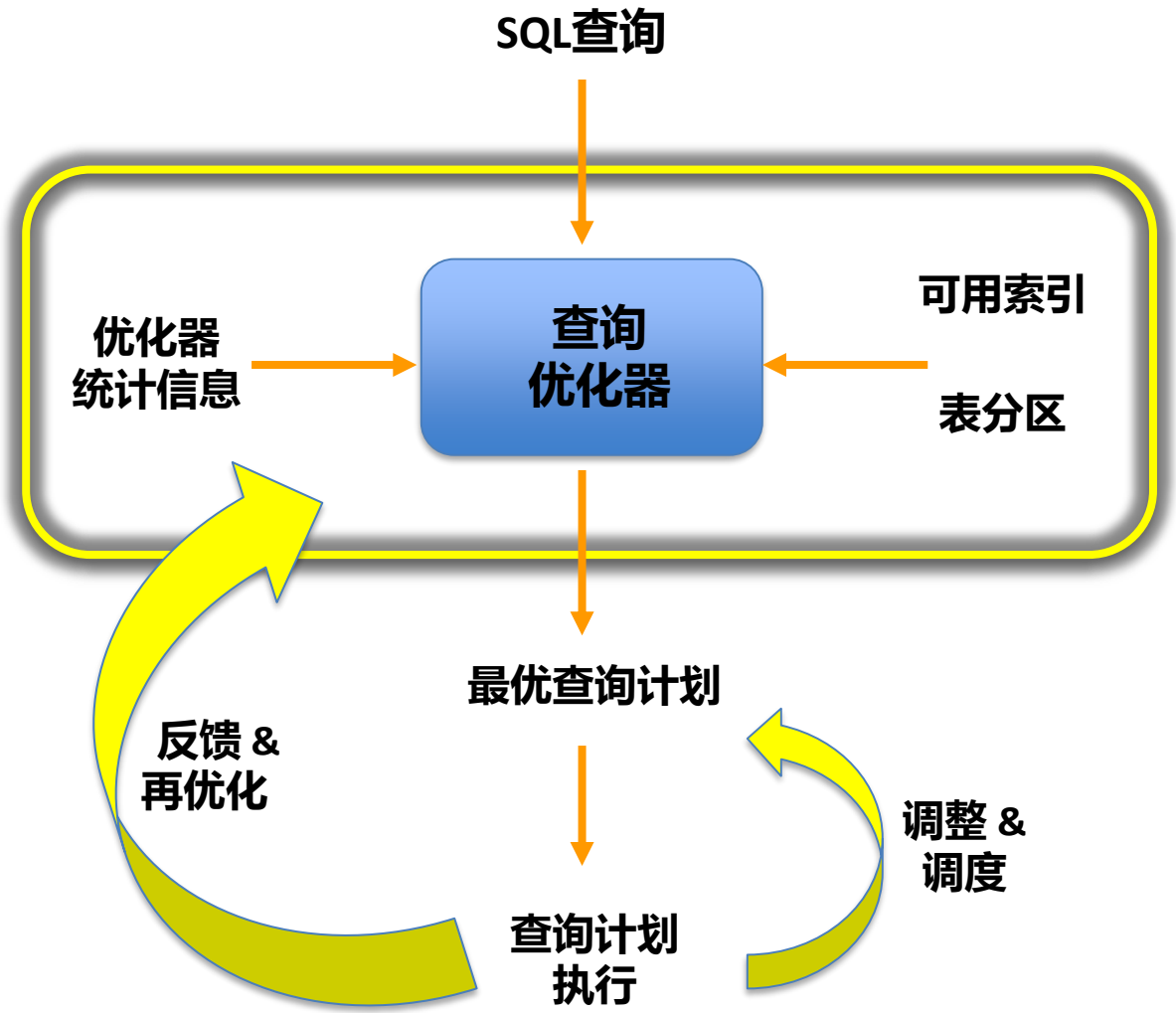
❑ 管理依赖性 – 统计信息推荐



智能查询优化

❑ 增强运行时

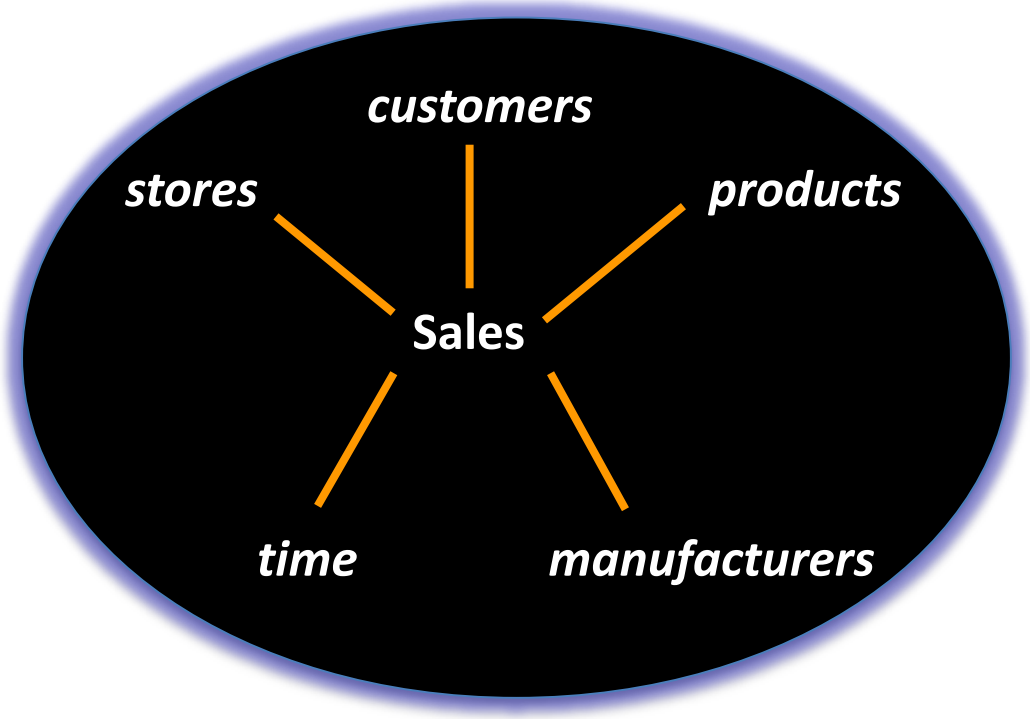
- ❑ **运行时基数反馈**
  - ❑ 收集实际基数信息
  - ❑ 触发“再优化”
- ❑ **运行时优化技术**
  - ❑ 带调度的多态访问计划（TP和AP）
  - ❑ 自适应访问计划和动态优化（AP）

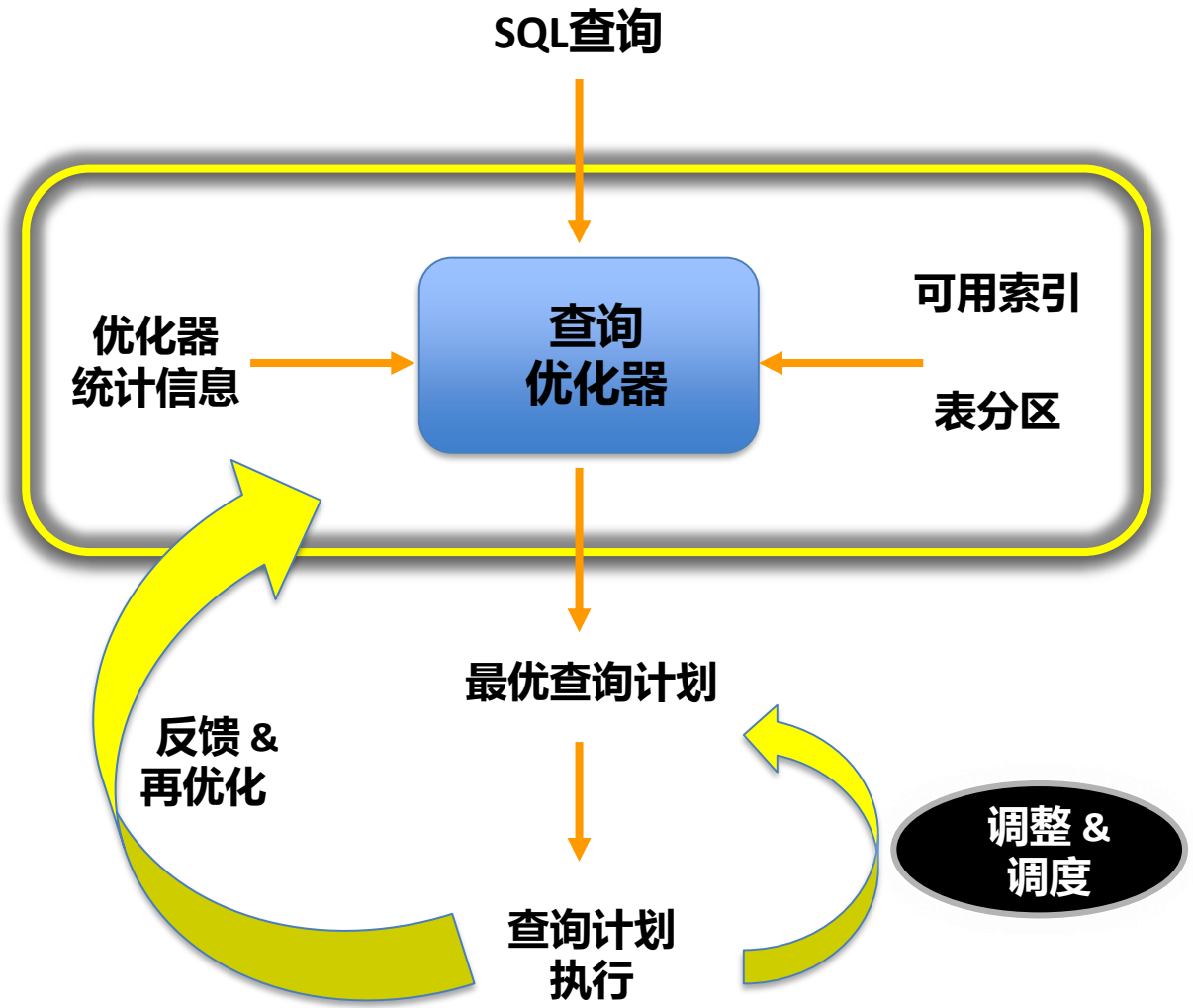


智能查询优化

❑ 增强运行时 – 以自适应优化为例

1. 评估维度表
2. 使用单例索引探查事实表
3. 使用多索引选择最具选择性（过滤性最高的）的维度扫描事实表

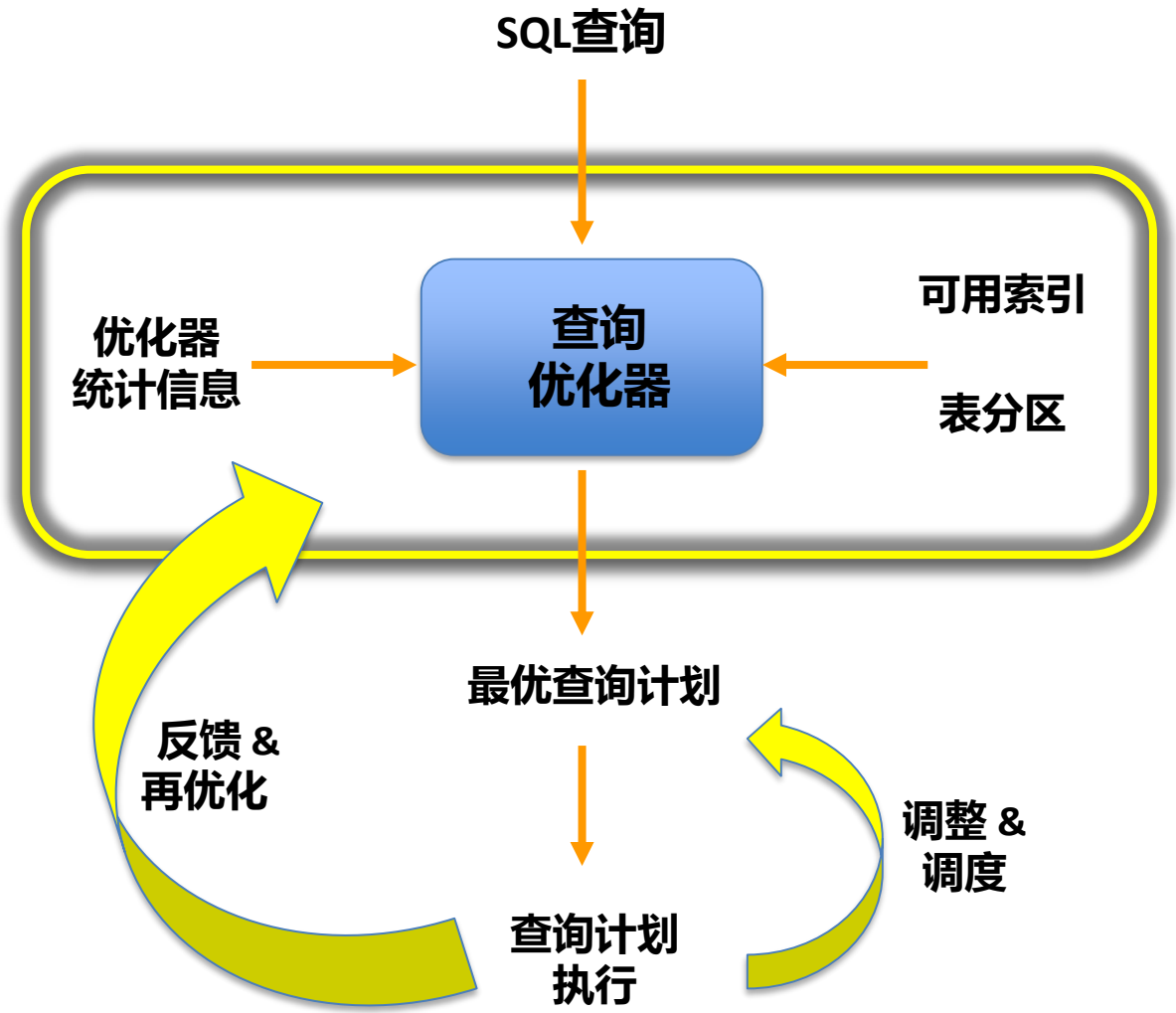




智能查询优化

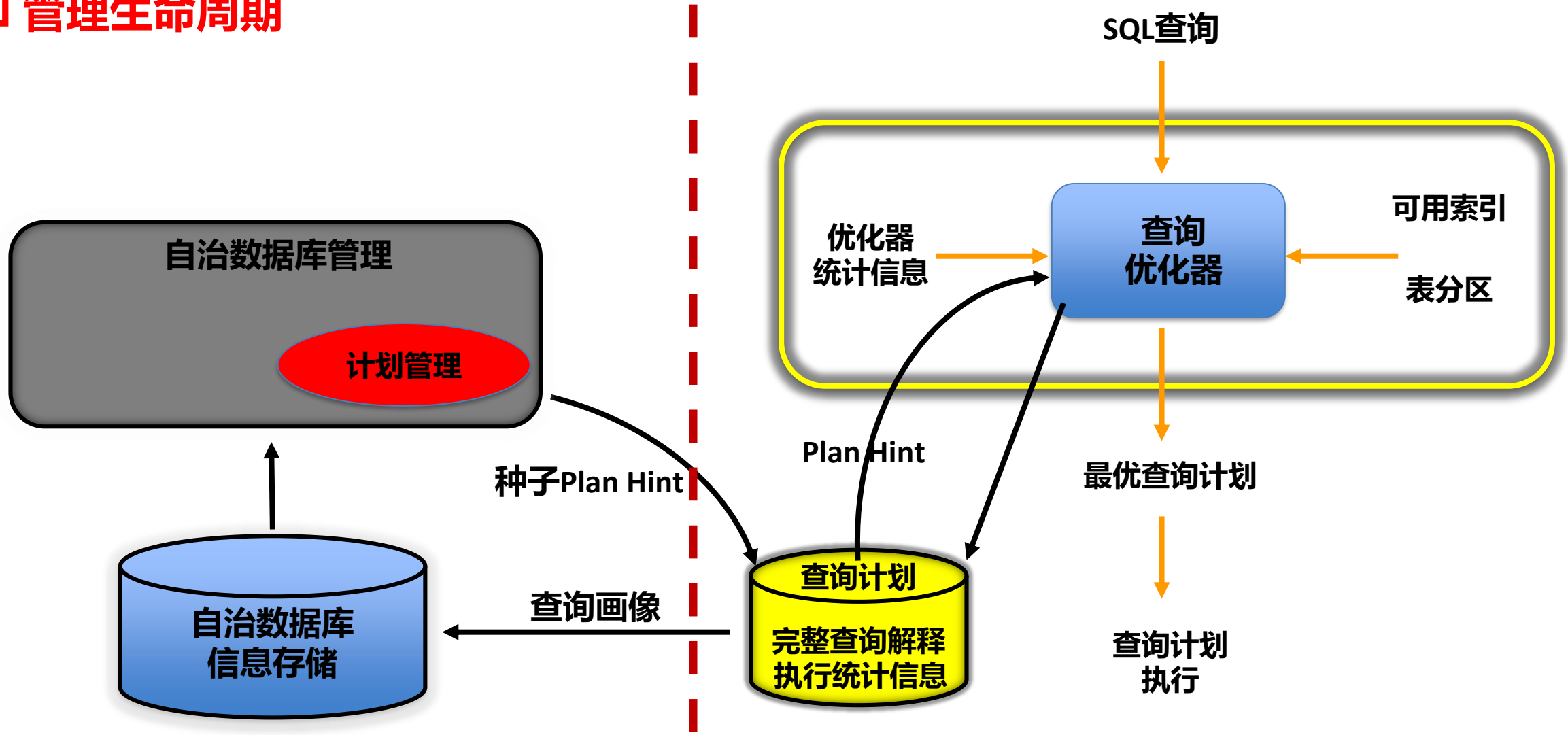
❑ 管理生命周期

- ❑ 每条查询完整的工作负载画像
  - ❑ 访问计划
  - ❑ 基数 – 实际值&估算值
  - ❑ 执行时间
  - ❑ 执行频率
- ❑ 基于丰富工作负载数据的推荐
- ❑ 全局的执行计划管理（计划稳定性）
  - ❑ 积极执行 – 计划变更的风险
  - ❑ 保守执行 – 冻结原来的计划
  - ❑ 自适应执行 – 回归后备



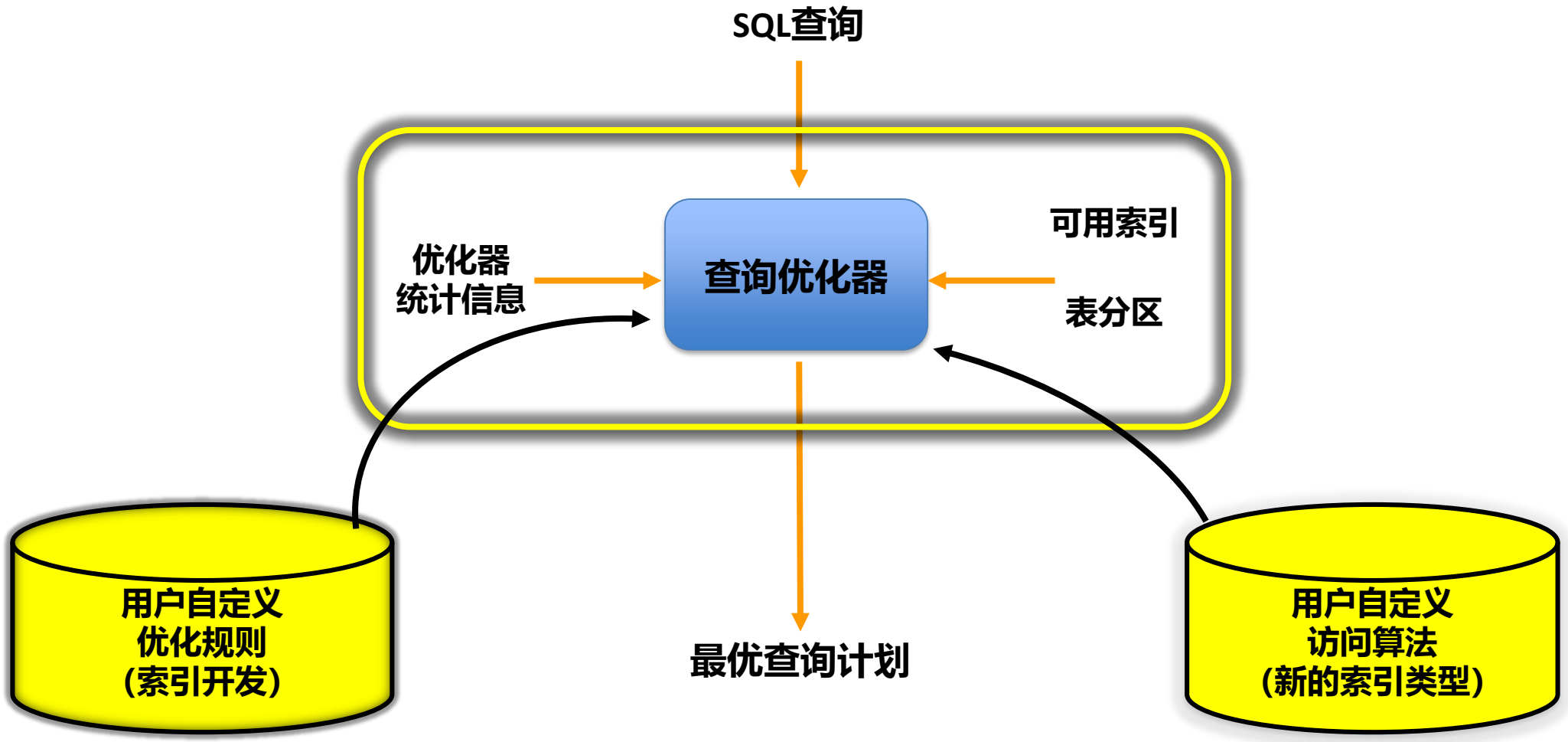
智能查询优化

❑ 管理生命周期



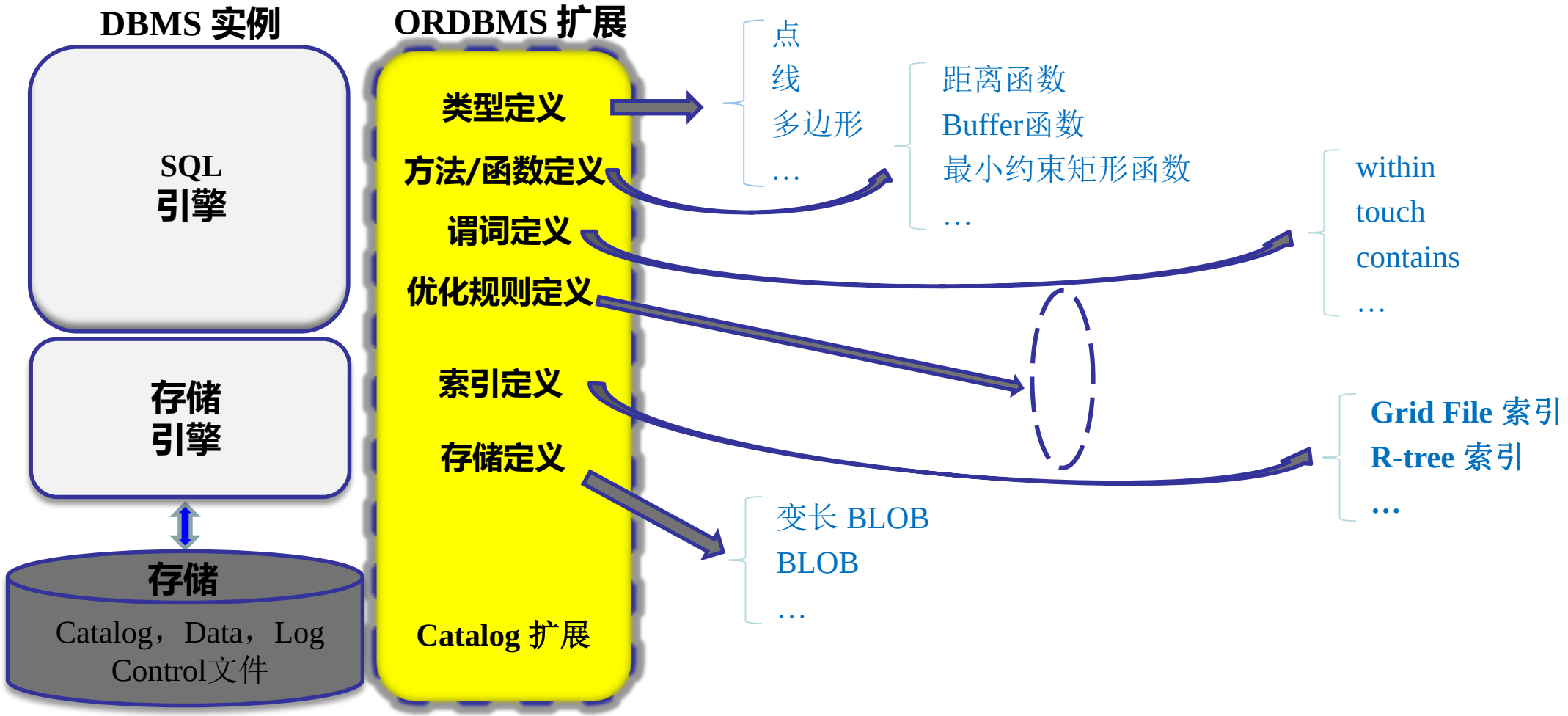
智能查询优化

❑ 可扩展查询优化——以空间数据库为例



智能查询优化

- 可扩展查询优化 – 面向对象的关系型数据库（ORDBMS）



智能查询优化

❑ 可扩展查询优化 – 面向对象的关系型数据库（ORDBMS）

blackList

id	dateofbirth	name	cellphone	address	gender

whereAbout

pid	type	s_time	e_time	s_loc	e_loc	cid
435567889	Hotel	8:41 pm 11/1/2017	8:45 am 11/4/2017	*****	*****	Shangrila Pudong
435567889	Flight	3:35 pm 11/4/2017	5:45 am 11/4/2017	*****	*****	CA1533
435567889	Hotel	10:32 pm 11/4/2017	9:55 am 11/10/2017	*****	*****	Shangrila Beijing
435567889	Bar	7:18 pm 11/4/2017	8:55 pm 11/4/2017	*****	*****	Fantatia

```
Select   w.pid, b.gender, b.dateofbirth, b.name, b.cellphone, b.address, …
from      whereAbout w, blackList b,
where   w.s_time < :crimeTime
   and   w.e_time > :crimeTime
   and   within(w.e_loc, buffer(:crimeLocation, 200)) = 1
   and   w.pid = b.id
```

查找曾在某个刑事案件案发现场出现过，并且列在黑名单上的人！

智能查询优化– SQL调优方法论与自动化调优

❑ 步骤一：查询注解

```
select o.date, o.quantity, p.name
from customers c,
        orders o,
        products p

where c.city in ('Wanzhou', 'Zhumadian')
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
    and p.type in ('Drink', 'Cosmetic')
order by o.date
```

Table Statistics	Available Indices	Partitioning Keys
10^9 / 10^7	I1: city, I2: id …	
10^10 / 10^8	I1: pid, I2: id, date …	
10^5 / 10^3	I1: id, I2: type …	

Column Statistics	Filter Factor (selectivity)	Index Page Data Page	Index Keys Data Records
500	2/500 ?	? / ?	? / ?
10^9 / 10^8	10^-9	2.5 / 1	1 / 10
10^5 / 10^5	10^-5	1 / 1	1 / 1
2008/1 – 2018/8	?	? / ?	? / ?
1000	2 / 1000 ?	? / ?	? / ?

❑ 步骤一：查询注解

- ❑ 重整查询 – 先将每一个 query block 清楚的分开, 然后针对每个 query block
  - ❑ 将 SELECT, FROM, WHERE, GROUP BY, ORDER BY 分开, 中间以一空行隔离
  - ❑ 将 SELECT list 中的 items 按表集中排序
  - ❑ 将 FROM 语句中的每一张表单独放一行 (以便加注索引与统计信息)
  - ❑ 将 WHERE 语句中的每一个 predicate 单独放在一行, 并按表或按 join/local 集中排序
  - ❑ 将 GROUP BY 的 items 按表排序
  - ❑ 将 ORDER BY 的 items 按表排序

❑ 步骤一：查询注解

- ❑ 信息加注 – 索引信息, 表的统计信息, 列的统计信息, 数据分布信息, 主键信息等
  - ❑ 每张表: 索引, 主键, 列, 分布, 统计信息, 当作首表的 join size 等
  - ❑ Join predicate: 所有列的统计信息, 估算的过滤率, 估算的双向 fanout 等
  - ❑ Local predicate: 列的统计信息, 估算的过滤率, 估算的准确性与稳定性等
  - ❑ GROUP BY list: 每一个 item 可能的基数, 有没有支持排序的索引
  - ❑ ORDER BY list: 有没有支持排序的索引

智能查询优化– SQL调优方法论与自动化调优

❑ 步骤二：首表的选择分析（过滤性与扫描开销）

```
select o.date, o.quantity, p.name
from customers c,   (0.4%, ISCAN)
     orders o,      (?, TSCAN)
     products p     (0.2%, ISCAN)

where c.city in ('Wanzhou', 'Zhumadian')
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
    and p.type in ('Drink', 'Cosmetic')
order by o.date
```

Table Statistics	Available Indices	Partitioning Keys
10^9 / 10^7	I1: city, I2: id …	
10^10 / 10^8	I1: pid, I2: id, date …	
10^5 / 10^3	I1: id, I2: type …	

Column Statistics	Filter Factor (selectivity)	Index Page Data Page	Index Keys Data Records
500	2/500 ?	? / ?	? / ?
10^9 / 10^8	10^-9	2.5 / 1	1 / 10
10^5 / 10^5	10^-5	1 / 1	1 / 1
2008/1 – 2018/8	?	? / ?	? / ?
1000	2 / 1000 ?	? / ?	? / ?

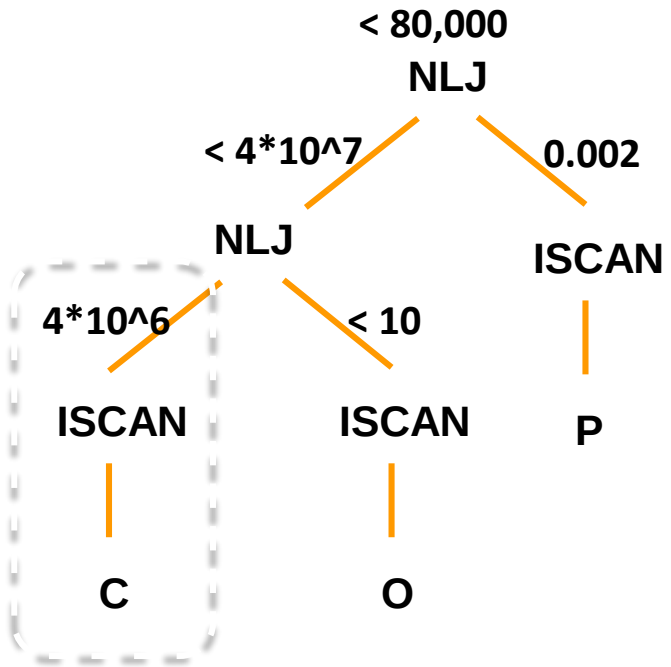
❑ 步骤二：首表的选择分析（过滤性与扫描开销）

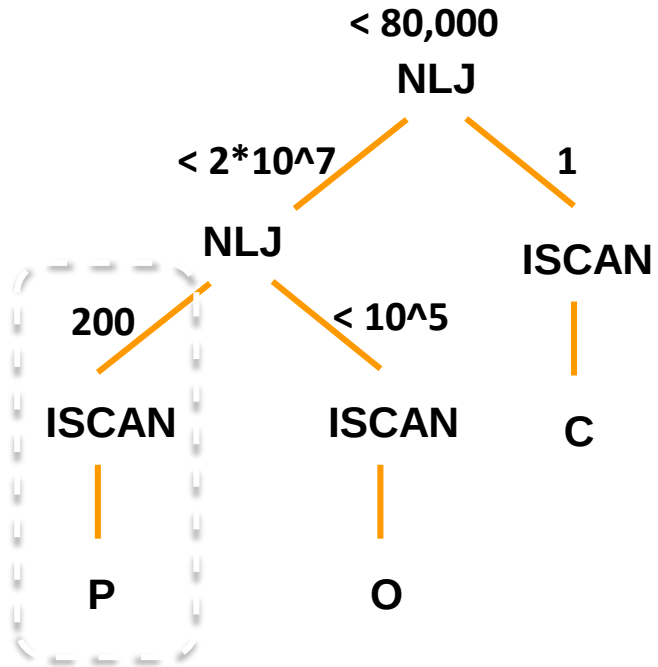
```
select o.date, o.quantity, p.name

from customers c,   (0.4%, ISCAN)
     orders o,      (?, TSCAN)
     products p     (0.2%, ISCAN)

where c.city in ('Wanzhou', 'Zhumadian')
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
    and p.type in ('Drink', 'Cosmetic')

order by o.date
```





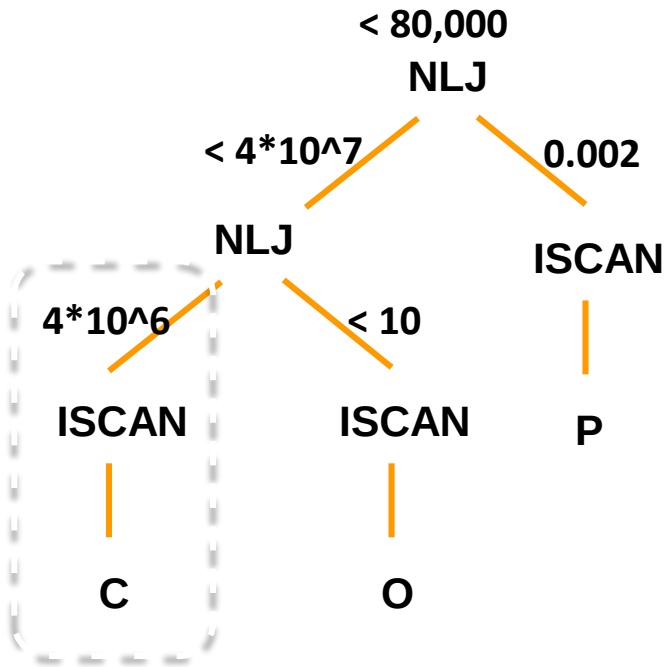
❑ 步骤二：新表的选择分析（过滤性与扫描开销）

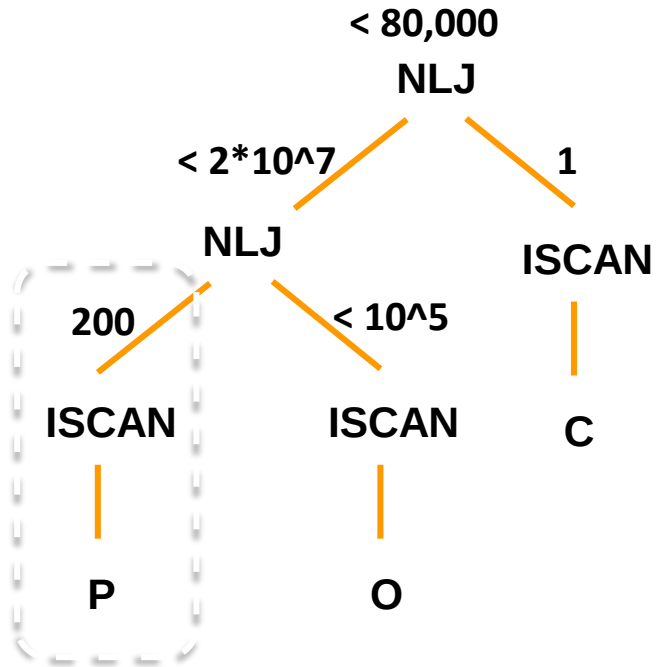
```
select o.date, o.quantity, p.name

from customers c,   (0.4%, ISCAN)
     orders o,      (?, TSCAN)
     products p     (0.2%, ISCAN)

where c.city in ('Wanzhou', 'Zhumadian')
    and c.id = o.cid
    and p.id = o.pid
    and o.date between ? and ?
    and p.type in ('Drink', 'Cosmetic')

order by o.date
```





THANKS

关注我们 / 了解更多