# 概述

为什么要引入hint？

Hint是oracle数据库中很有特色的一个功能，是很多DBA优化中经常使用的一个手段。那么为什么oracle会考虑引入优化器呢？基于代价的优化器是很聪明的，在绝大多数情况下它会选择正确的优化器，减轻DBA的负担。

但有时它聪明反被聪明误，选择了很差的执行计划，使得某个语句的执行变得异常慢。此时就需要DBA进行人为地干预，告诉优化器使用指定的存储路径或连接类型生成执行计划，从而使语句高效地进行。Hint就是oracle提供的一种机制，用来告诉优化器按照告诉它的方式执行计划。

Hint与注释

提示是oracle为了不解决和其他数据库引擎之间对SQL语句的兼容性而提供的一种拓展功能。

Oracle决定把提示作为一种特殊的注释来添加，它的特殊性表现在提示必须紧跟着DELETE、INSERT、UPDATE或MERGE关键字。

换句话说，提示不能像普通注释那样在SQL语句中随处添加。且在注释分隔符之后的第一个字符必须是加号。

Hint存在一些弊端：

1. Hint是比较“暴力”的一种解决方式，不是很优雅。需要开发人员手动修改代码。
2. Hint不会去适应新的变化。比如数据结构、数据规模发生了重大变化，但使用hint的语句是不会感知变化并产生更优的执行计划。
3. Hint随着数据库版本的变化，可能会有一些差异，设置废弃的情况。此时，语句本身是无感知的，必须人工测试并修正。

# 功能

Hint提供的功能非常多，可以很灵活地调整语句的执行过程。通过hint，我们可以调整：

优化器类型、优化器优化目标、数据读取方式（访问路径）、查询转换类型、表间关联的顺序、表间关联的类型、并行特性、其他特性

# 使用

对于hint的使用注意一点：不要过分依赖hint

当遇到SQL执行计划不好的情况，应优先考虑统计信息等问题，而不是直接添加hint了事。如果统计信息无误，应该考虑物理结构是否合理，即没有合适的索引。只有在最后仍然不能SQL按优化的执行计划执行时，才考虑hint。

毕竟使用hint，需要应用系统修改代码，hint只能解决一条SQL的问题，并且由于数据分布的变化或其他原因（如索引更名）等，会导致SQL再次出现性能问题。