1. /*+ALL_ROWS*/

表明对语句块选择基于开销的优化方法,并获得最佳吞吐量,使资源消耗最小化.

例如:

SELECT /*+ALL+_ROWS*/ * EMP_NO,EMP_NAM,DAT_IN FROM BSEMPMS WHERE EMP_NO='SCOTT';

2. /*+FIRST_ROWS*/

表明对语句块选择基于开销的优化方法,并获得最佳响应时间,使资源消耗最小化.

例如:

SELECT /*+FIRST_ROWS*/ EMP_NO,EMP_NAM,DAT_IN FROM BSEMPMS WHERE EMP_NO='SCOTT';

3. /*+CHOOSE*/

表明如果数据字典中有访问表的统计信息,将基于开销的优化方法,并获得最佳的吞吐量;

表明如果数据字典中没有访问表的统计信息,将基于规则开销的优化方法;

例如:

SELECT /*+CHOOSE*/ EMP_NO,EMP_NAM,DAT_IN FROM BSEMPMS WHERE EMP_NO='SCOTT';

4. /*+RULE*/

表明对语句块选择基于规则的优化方法.

例如:

SELECT /*+ RULE */ EMP_NO,EMP_NAM,DAT_IN FROM BSEMPMS WHERE EMP_NO='SCOTT';

5. /*+FULL(TABLE)*/

表明对表选择全局扫描的方法.

例如:

SELECT /*+FULL(A)*/ EMP_NO,EMP_NAM FROM BSEMPMS A WHERE EMP_NO='SCOTT';

6. /*+ROWID(TABLE)*/

提示明确表明对指定表根据 ROWID 进行访问.

例如:

SELECT /*+ROWID(BSEMPMS)*/ * FROM BSEMPMS WHERE ROWID>='AAAAAAAAAAAAAA'
AND EMP_NO='SCOTT';

7. /*+CLUSTER(TABLE)*/

提示明确表明对指定表选择簇扫描的访问方法,它只对簇对象有效.

例如:

SELECT /*+CLUSTER */ BSEMPMS.EMP_NO,DPT_NO FROM BSEMPMS,BSDPTMS
WHERE DPT_NO='TEC304' AND BSEMPMS.DPT_NO=BSDPTMS.DPT_NO;

8. /*+INDEX(TABLE INDEX_NAME)*/ 强制索引

表明对表选择索引的扫描方法.

例如:

SELECT /*+INDEX(BSEMPMS SEX_INDEX) USE SEX_INDEX BECAUSE THERE ARE FEWMALE BSEMPMS */ FROM BSEMPMS WHERE SEX='M';

9. /*+INDEX_ASC(TABLE INDEX_NAME)*/

表明对表选择索引升序的扫描方法.

例如:

SELECT /*+INDEX_ASC(BSEMPMS PK_BSEMPMS) */ FROM BSEMPMS WHERE DPT_NO='SCOTT';

10. /*+INDEX_COMBINE*/

为指定表选择位图访问路经,如果 INDEX_COMBINE 中没有提供作为参数的索引,将选择出位图索引的布尔组合方式.

例如:

SELECT /*+INDEX_COMBINE(BSEMPMS SAL_BMI HIREDATE_BMI)*/ * FROM BSEMPMS
WHERE SAL<5000000 AND HIREDATE

11. /*+INDEX_JOIN(TABLE INDEX_NAME)*/

提示明确命令优化器使用索引作为访问路径.

例如:

SELECT /*+INDEX_JOIN(BSEMPMS SAL_HMI HIREDATE_BMI)*/ SAL,HIREDATE
FROM BSEMPMS WHERE SAL<60000;

12. /*+INDEX_DESC(TABLE INDEX_NAME)*/

表明对表选择索引降序的扫描方法.

例如:

SELECT /*+INDEX_DESC(BSEMPMS PK_BSEMPMS) */ FROM BSEMPMS WHERE DPT_NO='SCOTT';

13. /*+INDEX_FFS(TABLE INDEX_NAME)*/

对指定的表执行快速全索引扫描,而不是全表扫描的办法.

例如:

SELECT /*+INDEX_FFS(BSEMPMS IN_EMPNAM)*/ * FROM BSEMPMS WHERE DPT_NO='TEC305';

14. /*+ADD_EQUAL TABLE INDEX_NAM1,INDEX_NAM2,...*/

提示明确进行执行规划的选择,将几个单列索引的扫描合起来.

例如:

SELECT /*+INDEX_FFS(BSEMPMS IN_DPTNO,IN_EMPNO,IN_SEX)*/ * FROM BSEMPMS WHERE EMP_NO='SCOTT' AND DPT_NO='TDC306';

15. /*+USE_CONCAT*/

对查询中的 WHERE 后面的 OR 条件进行转换为 UNION ALL 的组合查询.

例如:

SELECT /*+USE_CONCAT*/ * FROM BSEMPMS WHERE DPT_NO='TDC506' AND SEX='M';

16. /*+NO_EXPAND*/

对于 WHERE 后面的 OR 或者 IN-LIST 的查询语句,NO_EXPAND 将阻止其基于优化器对其进行扩展.

例如:

SELECT /*+NO_EXPAND*/ * FROM BSEMPMS WHERE DPT_NO='TDC506' AND SEX='M';

17. /*+NOWRITE*/

禁止对查询块的查询重写操作.

18. /*+REWRITE*/

可以将视图作为参数.

19. /*+MERGE(TABLE)*/

能够对视图的各个查询进行相应的合并.

例如:

SELECT /*+MERGE(V) */ A.EMP_NO,A.EMP_NAM,B.DPT_NO FROM BSEMPMS A (SELET DPT_NO ,AVG(SAL) AS AVG_SAL FROM BSEMPMS B GROUP BY DPT_NO) V WHERE A.DPT_NO=V.DPT_NO AND A.SAL>V.AVG_SAL;

20. /*+NO_MERGE(TABLE)*/

对于有可合并的视图不再合并.

例如:

SELECT /*+NO_MERGE(V) */ A.EMP_NO,A.EMP_NAM,B.DPT_NO FROM BSEMPMS A (SELECT DPT_NO,AVG(SAL) AS AVG_SAL FROM BSEMPMS B GROUP BY DPT_NO) V WHERE A.DPT_NO=V.DPT_NO AND A.SAL>V.AVG_SAL;

21. /*+ORDERED*/

根据表出现在 FROM 中的顺序,ORDERED 使 ORACLE 依此顺序对其连接.

例如:

SELECT /*+ORDERED*/ A.COL1,B.COL2,C.COL3 FROM TABLE1 A,TABLE2 B,TABLE3 C WHERE A.COL1=B.COL1 AND B.COL1=C.COL1;

22. /*+USE_NL(TABLE)*/

将指定表与嵌套的连接的行源进行连接,并把指定表作为内部表.

例如:

SELECT /*+ORDERED USE_NL(BSEMPMS)*/ BSDPTMS.DPT_NO,BSEMPMS.EMP_NO,BSEMPMS.EMP_NAM FROM BSEMPMS,BSDPTMS WHERE BSEMPMS.DPT_NO=BSDPTMS.DPT_NO;

23. /*+USE_MERGE(TABLE)*/

将指定的表与其他行源通过合并排序连接方式连接起来.

例如:

SELECT /*+USE_MERGE(BSEMPMS,BSDPTMS)*/ * FROM BSEMPMS,BSDPTMS WHERE BSEMPMS.DPT_NO=BSDPTMS.DPT_NO;

24. /*+USE_HASH(TABLE)*/

将指定的表与其他行源通过哈希连接方式连接起来.

例如:

SELECT /*+USE_HASH(BSEMPMS,BSDPTMS)*/ * FROM BSEMPMS,BSDPTMS WHERE BSEMPMS.DPT_NO=BSDPTMS.DPT_NO;

25. /*+DRIVING_SITE(TABLE)*/

强制与 ORACLE 所选择的位置不同的表进行查询执行.

例如:

SELECT /*+DRIVING_SITE(DEPT)*/ * FROM BSEMPMS,DEPT@BSDPTMS WHERE BSEMPMS.DPT_NO=DEPT.DPT_NO;

26. /*+LEADING(TABLE)*/

将指定的表作为连接次序中的首表.

27. /*+CACHE(TABLE)*/

当进行全表扫描时,CACHE 提示能够将表的检索块放置在缓冲区缓存中最近最少列表 LRU 的最近使用端

例如:

SELECT /*+FULL(BSEMPMS) CAHE(BSEMPMS) */ EMP_NAM FROM BSEMPMS;

28. /*+NOCACHE(TABLE)*/

当进行全表扫描时,CACHE 提示能够将表的检索块放置在缓冲区缓存中最近最少列表 LRU 的最近使用端

例如:

SELECT /*+FULL(BSEMPMS) NOCAHE(BSEMPMS) */ EMP_NAM FROM BSEMPMS;

29. /*+APPEND*/

直接插入到表的最后,可以提高速度.

insert /*+append*/ into test1 select * from test4 ;

30. /*+NOAPPEND*/

通过在插入语句生存期内停止并行模式来启动常规插入.

insert /*+noappend*/ into test1 select * from test4 ;