Oracle Hints

Hints for Access Paths

- Each of the following hints instructs the optimizer to use a specific access path for a table.
- Specifying one of these hints causes the optimizer to choose the specified access path only if the access path is available.
  - existence of an index
  - synthetic constructs of the SQL statement
- You must specify the table to be accessed exactly as it appears in the statement.
  - if the statement uses an alias for the table, then use the alias rather than the table name

- FULL (table)
- INDEX (table indexNames)
- NO_INDEX (table indexNames)
- INDEX_COMBINE (table indexNames)
- INDEX_FFS (table indexNames)
- NO_INDEX_FFS (table indexNames)
Hints for Access Paths

- **FULL(table)**
  - Full table scan on the specified table
  - If a table alias is defined, the table must be referenced with its alias
- **INDEX(table indexName1 indexName2 ...)**
  - Index scan using one or more specified indexes for the specified table
  - Does not consider a full table scan or a scan on an index not listed
- **NO_INDEX(table indexName1 indexName2 ...)**
  - Avoid using one or more specified indexes for the specified table
- **INDEX.Combine(table indexName1 indexName2 ...)**
  - Uses a bitmap access path (Boolean combination) of the specified indexes for the table
- **INDEX.FFS(table indexName1 indexName2 ...)**
  - instructs Oracle to perform a fast full index scan rather than a full table scan
- **NO_INDEX.FFS(table indexName1 indexName2 ...)**
  - Excludes a fast full index scan of the specified indexes on the specified table

---

Join Operations

- Each of the following hints instructs the optimizer to use a specific join operation for the specified tables
  - USE_NL(table1, table2, ...)
  - USE_NL(e j)
  - USE_MERGE(e j)
  - USE_HASH(e j)
  - USE_HASH(...)
  - USE_INDEX(e j)
  - USE_HASH(...)

Oracle uses these hints when the referenced table is forced to be the inner table of a join; the hints are ignored if the referenced table is the outer table.

---

Join Orders - Example

- **SELECT /*+ ORDERED */ */**
  - FROM emp e, dept d
  - WHERE d.deptno = e.deptno
  - GROUP BY e.empID, e.Name
  - SUM(j.salary)

- **INDEX_COMBINE(e j)**
  - Index scan combining the specified indexes for the join

---

Join Orders

- The following hints suggest join orders
  - **ORDERED**
  - **LEADING( table1 table2 ...)**

The ORDERED hint instructs Oracle to join tables in the order in which they appear in the FROM clause.

The LEADING hint instructs the optimizer to use the specified set of tables as the hint parameters.

These hints let you choose an inner and outer table
- The first table is the outer table
- The second table is the inner table

---

Example

- **SELECT /*+**
  - **LEADING(e j)**
  - **USE_NL(e j)**
  - **USE_INDEX(empID_index)**
  - **FULL(e)/**
  - **e.empID, e.Name, sum(j.salary)**
  - **GROUP BY e.empID, e.Name**

The LEADING hint specifies the exact join order to be used.
- The index empID_index is suggested to be used.
- The join method USE_NL to be used on the join tables is also specified.
- The FULL table access path to table jobs is suggested.