Using EXPLAIN PLAN and DBMS_XPLAN.DISPLAY in Oracle SQL

1. Using EXPLAIN PLAN to Analyze a Query Execution Plan
The EXPLAIN PLAN command stores the execution plan in the PLAN_TABLE,
allowing you to inspect how Oracle processes a query.

Basic Syntax:
EXPLAIN PLAN FOR
<your SQL query>;

This does NOT execute the query but only generates the execution plan.

Example:
EXPLAIN PLAN FOR
SELECT * FROM EMPLOYEES WHERE DEPARTMENT_ID = 10;

---

2. Viewing the Execution Plan with DBMS_XPLAN.DISPLAY
After explaining the plan, retrieve the execution plan using:

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

This displays execution details like operation type, cost, and rows processed.

---

3. Interpreting the Execution Plan
Example Output:

Plan hash value: 1234567890

--

```
| Id  | Operation         | Name        | Rows  | Bytes | Cost |
-----------------------------------------------------------
|   0 | SELECT STATEMENT  |             |  1000 | 8000  |  10  |
|   1 |  TABLE ACCESS FULL| EMPLOYEES   |  1000 | 8000  |  10  |
-----------------------------------------------------------
```

Key Observations:
- **TABLE ACCESS FULL means a full table scan, which is expensive.**
- **If an index exists but is not used, check optimizer settings.**

---

4. Optimizing the Query
Using Indexes:
If a Full Table Scan occurs despite an index on DEPARTMENT_ID, force an index scan:

```
SELECT /*+ INDEX(EMPLOYEES EMP_DEPT_IDX) */ *
FROM EMPLOYEES
WHERE DEPARTMENT_ID = 10;
```

Using AUTOTRACE for Real Execution Plan:
Instead of EXPLAIN PLAN, use:

```
SET AUTOTRACE ON;
SELECT * FROM EMPLOYEES WHERE DEPARTMENT_ID = 10;
```

---

5. Common Optimization Techniques:
1. Use Indexes Efficiently - Ensure queries use the right indexes.
2. Avoid Full Table Scans - Use indexes or partitioning where applicable.
3. Rewrite Queries - Use EXISTS instead of IN, JOIN instead of SUBQUERY, etc.
4. Use GATHER_STATS - Ensure optimizer statistics are up-to-date:

```
EXEC DBMS_STATS.GATHER_TABLE_STATS('HR', 'EMPLOYEES');
```

5. Check for Implicit Conversions - Ensure indexed columns are used correctly:

SELECT * FROM EMPLOYEES WHERE TO_CHAR(DEPARTMENT_ID) = '10'; -- BAD
SELECT * FROM EMPLOYEES WHERE DEPARTMENT_ID = 10; -- GOOD

Summary:
- EXPLAIN PLAN FOR generates the execution plan.
- DBMS_XPLAN.DISPLAY displays the plan for analysis.
- Optimize queries by using indexes, analyzing statistics, and avoiding full table scans.