Full Table Scan- Rows are traversed sequentially

ROWID Scan- Rowid is mapped to actual row done majorly by index

Sample scan- some amount of sample data is going to be scaned to understand the pattern

Index Scan

Unique- happens generally for primary key or unique index

Range Scan- happens generally for non unique index

Index full scan- index traversal is done on left side of the index

Index fast full scan- the records are retrieved by means of index only

Index skip scan- probing the index only for distinct values

Index join scan- joining by means of indexed columns

Bitmap Index- utilized for OLAP or for querying low cardinality of the columns

Function based Index

SQL> create index funidx on emp(upper(empname));

Index created.

SQL> analyze table emp compute statistics;

Table analyzed.

SQL> select empname from emp;

Execution Plan

----------------------------------------------------------

Plan hash value: 3956160932

--------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

--------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 5 | 25 | 3 (0)| 00:00:01 |

| 1 | TABLE ACCESS FULL| EMP | 5 | 25 | 3 (0)| 00:00:01 |

--------------------------------------------------------------------------

SQL> select empname from emp where empname='John';

Execution Plan

----------------------------------------------------------

Plan hash value: 3956160932

--------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

--------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 1 | 5 | 3 (0)| 00:00:01 |

|\* 1 | TABLE ACCESS FULL| EMP | 1 | 5 | 3 (0)| 00:00:01 |

--------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

1 - filter("EMPNAME"='John')

SQL> select empname from emp where upper(empname)='John';

Execution Plan

----------------------------------------------------------

Plan hash value: 3765185667

--------------------------------------------------------------------------------

--------------

| Id | Operation | Name | Rows | Bytes | Cost (%CP

U)| Time |

--------------------------------------------------------------------------------

--------------

| 0 | SELECT STATEMENT | | 1 | 10 | 2 (

0)| 00:00:01 |

| 1 | TABLE ACCESS BY INDEX ROWID BATCHED| EMP | 1 | 10 | 2 (

0)| 00:00:01 |

|\* 2 | INDEX RANGE SCAN | FUNIDX | 1 | | 1 (

0)| 00:00:01 |

--------------------------------------------------------------------------------

--------------

Predicate Information (identified by operation id):

---------------------------------------------------

2 - access(UPPER("EMPNAME")='John')

SQL>

Join operations

Hash Join, Sort Merge JOIN, Nested Loop Join

Each distinct value is going to have single bucket

exec dbms\_stats.gather\_table\_stats(USER,'EMP',Options=>'GATHER AUTO');

exec dbms\_stats.gather\_table\_stats(USER,'EMP',Options=>'GATHER AUTO');

SQL> select table\_name, histogram from user\_tab\_col\_statistics;

SQL> exec dbms\_stats.gather\_table\_stats(ownname=>'HR',tabname=>'EMP',method\_opt

=>'For columns empname');

EMP DEPT

5000

5 rows

30000

10 rows

10000

10 rows

20000

10 rows