

```
SQL>
SQL> create table t
  2 as
  3 select
  4     rownum as id
  5     , case
  6       when rownum <= 910000
  7         then 1
  8         else mod(rownum, 999) + 2
  9       end as fk
 10     , rpad('x', 100) as filler
 11 from
 12     dual
 13 connect by
 14     level <= 1000000
 15 ;
```

Table created.

```
SQL>
SQL> exec dbms_stats.gather_table_stats(null, 't', method_opt => 'for all columns size 1')
```

PL/SQL procedure successfully completed.

```
SQL>
SQL> create index t_idx on t (fk);
```

Index created.

```
SQL>
SQL> create table d
  2 as
  3 select
  4     rownum as id
  5     , 'DESC' || rownum as some_attr
  6     , rpad('x', 100) as filler
  7 from
  8     dual
  9 connect by
 10     level <= 1000
 11 ;
```

Table created.

```
SQL>
SQL> exec dbms_stats.gather_table_stats(null, 'd', method_opt => 'for all columns size 1')
```

PL/SQL procedure successfully completed.

```
SQL>
SQL> alter table d add constraint d_pk primary key (id);
Table altered.
SQL>
SQL> alter table t add constraint t_fk_d foreign key (fk) references d(id);
Table altered.
```

```
SQL>
SQL> select
  2      *
  3 from
  4      (
  5          select
  6              fk
  7              , count(*) as cnt
  8          from
  9              t
 10          group by
 11              fk
 12          order by
 13              cnt desc
 14              , fk
 15      )
 16 where
 17      rownum <= 10
 18 ;
```

FK	CNT
1	910000
2	91
3	91
913	91
914	91
915	91
916	91
917	91
918	91
919	91

10 rows selected.

```
SQL>
SQL> select
  2      *
```

```
SQL>
SQL> select
  2
  3 from
  4   (
  5     select
  6         id
  7         , some_attr
  8     from
  9         d
 10     order by
 11         id
 12   )
13 where
14     rownum <= 10
15 ;
```

ID SOME\_ATTR

```
-----
  1 DESC1
  2 DESC2
  3 DESC3
  4 DESC4
  5 DESC5
  6 DESC6
  7 DESC7
  8 DESC8
  9 DESC9
 10 DESC10
```

10 rows selected.

```
SQL>
SQL> select
  2     num_distinct
  3     , utl_raw.cast_to_number(low_value) as low_value
  4     , utl_raw.cast_to_number(high_value) as high_value
  5     , num_buckets
  6     , sample_size
  7     , histogram
  8     , last_analyzed
  9 from
 10     user_tab_col_statistics
 11 where
 12     table_name = 'T'
 13 and  column_name = 'FK'
 14 ;
```

```
13 and column_name = 'FK'  
14 ;
```

NUM_DISTINCT	LOW_VALUE	HIGH_VALUE	NUM_BUCKETS	SAMPLE_SIZE	HISTOGRAM	LAST_ANALYZED
1000	1	1000	1	1000000	NONE	20120408 21:13:08

```
SQL>  
SQL> explain plan for  
2 select  
3     count(t.filler)  
4 from  
5     t  
6     , d  
7 where  
8     d.some_attr = 'DESC1'  
9 and  t.fk = d.id  
10 ;
```

Explained.

```
SQL>  
SQL> select * from table(dbms_xplan.display);
```

PLAN\_TABLE\_OUTPUT

Plan hash value: 1302414233

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	117	113 (0)	00:00:02
1	SORT AGGREGATE		1	117		
2	NESTED LOOPS					
3	NESTED LOOPS		1000	114K	113 (0)	00:00:02
* 4	TABLE ACCESS FULL	D	1	12	6 (0)	00:00:01
* 5	INDEX RANGE SCAN	T_IDX	1000		2 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	T	1000	102K	107 (0)	00:00:02

Predicate Information (identified by operation id):

- 4 - filter("D"."SOME\_ATTR"='DESC1')
- 5 - access("T"."FK"="D"."ID")

19 rows selected.

```
SQL>
```

```
SQL> explain plan for
2 select
3     count(*)
4 from   t
5 where  t.fk = 1
6 ;
```

Explained.

```
SQL>
SQL> select * from table(dbms_xplan.display);
```

PLAN\_TABLE\_OUTPUT

Plan hash value: 293504097

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	4	4 (0)	00:00:01
1	SORT AGGREGATE		1	4		
* 2	INDEX RANGE SCAN	T_IDX	1000	4000	4 (0)	00:00:01

Predicate Information (identified by operation id):

2 - access("T"."FK"=1)

14 rows selected.

```
SQL>
SQL> exec dbms_stats.gather_table_stats(null, 't', method_opt => 'for all columns size 1 for columns size 254 fk')
```

PL/SQL procedure successfully completed.

```
SQL>
SQL> select
2     num_distinct
3     , utl_raw.cast_to_number(low_value) as low_value
4     , utl_raw.cast_to_number(high_value) as high_value
5     , num_buckets
6     , sample_size
7     , histogram
8     , last_analyzed
9 from
10    user_tab_col_statistics
```

14 rows selected.

SQL>

SQL> exec dbms\_stats.gather\_table\_stats(null, 't', method\_opt => 'for all columns size 1 for columns size 254 fk')

PL/SQL procedure successfully completed.

SQL>

```
SQL> select
 2  num_distinct
 3  , utl_raw.cast_to_number(low_value) as low_value
 4  , utl_raw.cast_to_number(high_value) as high_value
 5  , num_buckets
 6  , sample_size
 7  , histogram
 8  , last_analyzed
 9  from
10  user_tab_col_statistics
11 where
12  table_name = 'T'
13 and  column_name = 'FK'
14 ;
```

NUM_DISTINCT	LOW_VALUE	HIGH_VALUE	NUM_BUCKETS	SAMPLE_SIZE	HISTOGRAM	LAST_ANALYZED
1000	1	1000	254	5463	HEIGHT BALANCED	20120408 21:13:14

SQL>

```
SQL> select
 2  cum_cnt - lag(cum_cnt, 1, 0) over (order by endpoint_value) as cnt
 3  , endpoint_value
 4  from
 5  (
 6  select
 7  round(h.endpoint_number/c.num_buckets*t.sample_size) as cum_cnt
 8  , endpoint_value
 9  from
10  user_tab_histograms h
11  , user_tables t
12  , user_tab_columns c
13  where
14  h.table_name = 'T'
15  and  h.column_name = 'FK'
16  and  t.table_name = h.table_name
17  and  c.table_name = h.table_name
18  and  c.column_name = h.column_name
19  )
20 ;
```

```
SQL> select
2   cum_cnt - lag(cum_cnt, 1, 0) over (order by endpoint_value) as cnt
3   , endpoint_value
4 from
5   (
6     select
7       round(h.endpoint_number/c.num_buckets*t.sample_size) as cum_cnt
8     , endpoint_value
9     from
10      user_tab_histograms h
11     , user_tables t
12     , user_tab_columns c
13     where
14       h.table_name = 'T'
15      and h.column_name = 'FK'
16      and t.table_name = h.table_name
17      and c.table_name = h.table_name
18      and c.column_name = h.column_name
19   )
20 ;
```

CNT	ENDPOINT_VALUE
913386	1
3937	13
3937	88
3937	122
3937	169
3937	199
3937	250
3937	290
3937	337
3937	381
3937	425
3937	487
3937	532
3937	575
3937	608
3937	657
3937	710
3937	753
3937	802
3937	854
3937	906
3937	958
3937	1000

23 rows selected.

```
SQL> explain plan for
SQL> 2 select
3     count(*)
4   from
5     t
6  where
7     t.fk = 1
8  ;
```

Explained.

```
SQL>
SQL> select * from table(dbms_xplan.display);
```

PLAN\_TABLE\_OUTPUT

Plan hash value: 1058879072

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	4	547 (3)	00:00:07
1	SORT AGGREGATE		1	4		
* 2	INDEX FAST FULL SCAN	T_IDX	913K	3567K	547 (3)	00:00:07

Predicate Information (identified by operation id):

2 - filter("T"."FK"=1)

14 rows selected.

```
SQL>
SQL> explain plan for
SQL> 2 select
3     count(t.filler)
4   from
5     t
6     , d
7  where
8     d.some_attr = 'DESC1'
9  and
10    t.fk = d.id
10 ;
```

Explained.



-----  
Predicate Information (identified by operation id):

2 - filter("T"."FK"=1)

14 rows selected.

```
SQL>
SQL> explain plan for
2  select
3     count(t.filler)
4  from
5     t
6     , d
7  where
8     d.some_attr = 'DESC1'
9  and
10    t.fk = d.id
10 ;
```

Explained.

```
SQL>
SQL> select * from table(dbms_xplan.display);
```

PLAN\_TABLE\_OUTPUT

-----  
Plan hash value: 1302414233

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	117	113 (0)	00:00:02
1	SORT AGGREGATE		1	117		
2	NESTED LOOPS					
3	NESTED LOOPS		1000	114K	113 (0)	00:00:02
* 4	TABLE ACCESS FULL	D	1	12	6 (0)	00:00:01
* 5	INDEX RANGE SCAN	T_IDX	1000		2 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	T	1000	102K	107 (0)	00:00:02

-----  
Predicate Information (identified by operation id):

4 - filter("D"."SOME\_ATTR"='DESC1')  
5 - access("T"."FK"="D"."ID")

19 rows selected.

```
SQL>  
SQL> alter session set statistics_level = all;  
  
Session altered.
```

```
SQL>  
SQL> select  
2     count(t.filler)  
3   from  
4     t  
5     , d  
6   where  
7     d.some_attr = 'DESC1'  
8   and   t.fk = d.id  
9   ;
```

```
COUNT(T.FILLER)  
-----  
          910000
```

```
SQL>  
SQL> @xplan_extended_display_cursor  
SQL> set echo off verify off termout off  
SQL_ID dmdw54zsmvpa, child number 0
```

```
select      count(t.filler) from      t      , d where  
d.some_attr = 'DESC1' and      t.fk = d.id
```

Plan hash value: 1302414233

Id	Pid	Ord	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers	Reads	A-Time Self	Bufs Self	Reads Self	A-Ti S-Graph
0		7	SELECT STATEMENT		1		1	00:00:04.57	16236	14475	00:00:00.00	0	0	
1	0	6	SORT AGGREGATE		1	1	1	00:00:04.57	16236	14475	00:00:00.08	0	0	
2	1	5	NESTED LOOPS		1		910K	00:00:04.49	16236	14475	00:00:00.38	0	0	@
3	2	3	NESTED LOOPS		1	1000	910K	00:00:00.43	1794	33	00:00:00.15	0	0	
* 4	3	1	TABLE ACCESS FULL	D	1	1	1	00:00:00.01	18	33	00:00:00.01	18	33	
* 5	3	2	INDEX RANGE SCAN	T_IDX	1	1000	910K	00:00:00.27	1776	0	00:00:00.27	1776	0	@
6	2	4	TABLE ACCESS BY INDEX ROWID	T	910K	1000	910K	00:00:03.68	14442	14442	00:00:03.68	14442	14442	@@@@@@@@@@

Predicate Information (identified by operation id):

- 4 - filter("D"."SOME\_ATTR"='DESC1')
- 5 - access("T"."FK"="D"."ID")