ORACLE

Metadata Matters

Thomas Kyte
http://asktom.oracle.com/

Did you know…

- The optimizer uses constraints
  - Statistics plus
  - Extended statistics (profiles, virtual columns, etc) plus
  - System statistics plus
  - Dictionary data (object type information) plus
  - The Query plus
  - *All available bits of metadata*
- Constraints are not just about data integrity
  - Not that they would be less useful if they were not…
  - Constraints in a warehouse are crucial for performance (and getting the right answer)
  - Constraints in a transactional system are crucial for getting the right data (and for performance)

Datatypes are Constraints

ORACLE

“Wrong cardinality = Wrong Plan”

Datatypes are important

```
ops$tkyte%ORA11GR2> create table t ( str_date, date_date, number_date, data )
  2  as
  3  select to_char( dt+rownum,'yyyymmdd' ) str_date,
  4         dt+rownum date_date,
  5         to_number( to_char( dt+rownum,'yyyymmdd' ) ) number_date,
  6         rpad('*',45,'*') data
  7    from (select to_date('01-jan-1995','dd-mon-yyyy') dt
  8            from all_objects)
  9   order by dbms_random.random
 10  /
Table created.

ops$tkyte%ORA11GR2> create index t_str_date_idx on t(str_date);
ops$tkyte%ORA11GR2> create index t_date_date_idx on t(date_date);
ops$tkyte%ORA11GR2> create index t_number_date_idx on t(number_date);
```

Datatypes are important

```
ops$tkyte%ORA11GR2> begin
  2          dbms_stats.gather_table_stats
  3          ( user, 'T',
  4            method_opt=> 'for all indexed columns size 254',
  5            cascade=> true );
  6  end;
  7  /

PL/SQL procedure successfully completed.
```

Datatypes are important

```
ops$tkyte%ORA11GR2> select * from t
  2   where str_date between '20001231' and '20010101';

STR_DATE DATE_DATE NUMBER_DATE DATA
-------- --------- ----------- ----------------------------------------
20010101 01-JAN-01    20010101 ****************************************
20001231 31-DEC-00    20001231 ****************************************
-----------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |   254 | 11938 |   208   (1)| 00:00:03 |
|*  1 |  TABLE ACCESS FULL| T    |   254 | 11938 |   208   (1)| 00:00:03 |
-----------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("STR_DATE"<='20010101' AND "STR_DATE">='20001231')
```

Datatypes are important

```
ops$tkyte%ORA11GR2> select * from t
  2   where number_date between 20001231 and 20010101;

STR_DATE DATE_DATE NUMBER_DATE DATA
-------- --------- ----------- ----------------------------------------
20010101 01-JAN-01    20010101 ****************************************
20001231 31-DEC-00    20001231 ****************************************
-------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |   254 | 11938 |   208   (1)| 00:00:03 |
|*  1 |  TABLE ACCESS FULL| T    |   254 | 11938 |   208   (1)| 00:00:03 |
-------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("NUMBER_DATE"<=20010101 AND "NUMBER_DATE">=20001231)
```

Datatypes are important

```
ops$tkyte%ORA11GR2> select * from t where date_date
  2  between to_date('20001231','yyyymmdd') and to_date('20010101','yyyymmdd');

STR_DATE DATE_DATE NUMBER_DATE DATA
-------- --------- ----------- ------------------------------------------------
20001231 31-DEC-00    20001231 ************************************************
20010101 01-JAN-01    20010101 ************************************************
-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name            | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                 |     1 |    47 |     3   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID| T               |     1 |    47 |     3   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | T_DATE_DATE_IDX |     1 |       |     2   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("DATE_DATE">=TO_DATE(' 2000-12-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')
             AND "DATE_DATE"<=TO_DATE(' 2001-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

“Wrong Length = Inefficient Memory Use”

Datatypes are Important – Lengths Matter

Varchar2(n) where N is "right sized"	Varchar2(4000) for *everything* *(just in 'case')*
• Assume 10 columns, average width is 40 characters (some are 80, some are 10…)	• Assume 10 columns, average, minimum, maximum width is 4000
• 400 bytes per row on a fetch	• 40,000 bytes per row on a fetch
• Assume array fetch of 100 rows, so array fetch buffer of 40,000 bytes	• Assume array fetch of 100 rows, so array fetch buffer of 4,000,000 bytes
• Assume 25 open cursors, so 1,000,000 bytes of array fetch buffers	• Assume 25 open cursors, so 100,000,000 bytes
• Assume connection pool with 30 connections – **30mb**	• Assume connection pool with 30 connections – **3gb**

Constraints are *Important*

Why constraints matter

- Constraints are facts
- Constraints are more information
- Constraints convey information to the optimizer
- The presence of constraints open up access paths that would not be otherwise available.
- Examples…

“Check Constraints can rewrite queries”

Check constraints are important

```
ops$tkyte%ORA11GR2> create table t1
  2  as
  3  select * from stage
  4  where object_type in ( 'TABLE', 'VIEW' );
ops$tkyte%ORA11GR2> alter table t1 modify object_type not null;
ops$tkyte%ORA11GR2> alter table t1 add constraint t1_check_otype
  2  check (object_type in ('TABLE','VIEW'));

ops$tkyte%ORA11GR2> create table t2
  2  as
  3  select * from stage
  4  where object_type in ( 'SYNONYM', 'PROCEDURE' );
ops$tkyte%ORA11GR2> alter table t2 modify object_type not null;
ops$tkyte%ORA11GR2> alter table t2 add constraint t2_check_otype
  2  check (object_type in ('SYNONYM','PROCEDURE'));
```

Check constraints are important

```
ops$tkyte%ORA11GR2> create or replace view v
  2  as
  3  select * from t1
  4  union all
  5  select * from t2;

View created.

ops$tkyte%ORA11GR2> begin
  2      dbms_stats.gather_table_stats( user, 'T1' );
  3      dbms_stats.gather_table_stats( user, 'T2' );
  4  end;
  5  /

PL/SQL procedure successfully completed.
```

Check constraints are important

```
ops$tkyte%ORA11GR2> select * from v where object_type = 'TABLE';
-------------------------------------------------------------------------------
| Id  | Operation             | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |      |  8967 |  1164K|   147   (1)| 00:00:02 |
|   1 |  VIEW                 | V    |  8967 |  1164K|   147   (1)| 00:00:02 |
|   2 |   UNION-ALL           |      |       |       |            |          |
|*  3 |    TABLE ACCESS FULL  | T1   |  2851 |   242K|    33   (0)| 00:00:01 |
|*  4 |    FILTER             |      |       |       |            |          |
|*  5 |     TABLE ACCESS FULL | T2   |    39 |  3783 |   116   (0)| 00:00:02 |
-------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------
   3 - filter("OBJECT_TYPE"='TABLE')
   4 - filter(NULL IS NOT NULL)
   5 - filter("OBJECT_TYPE"='TABLE')
```

Check constraints are important

```
select * from v where object_type = 'TABLE'

call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total      194      0.00       0.01          0        295          0        2851

Rows     Row Source Operation
-------  ---------------------------------------------------
   2851  VIEW  V (cr=295 pr=0 pw=0 time=55056 us cost=147 size=1192611 card=8967)
   2851   UNION-ALL  (cr=295 pr=0 pw=0 time=24743 us)
   2851    TABLE ACCESS FULL T1 (cr=295 pr=0 pw=0 time=6088 us cost=33 …)
      0    FILTER  (cr=0 pr=0 pw=0 time=0 us)
      0     TABLE ACCESS FULL T2 (cr=0 pr=0 pw=0 time=0 us cost=116 ….)
```

Check constraints are important

```
ops$tkyte%ORA11GR2> alter table t1 drop constraint t1_check_otype;
ops$tkyte%ORA11GR2> alter table t2 drop constraint t2_check_otype;

select * from v where object_type = 'TABLE'

call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total      194      0.00       0.02          0        703          0        2851

Rows     Row Source Operation
-------  ---------------------------------------------------
   2851  VIEW  V (cr=703 pr=0 pw=0 time=27981 us cost=147 size=1192611 card=8967)
   2851   UNION-ALL  (cr=703 pr=0 pw=0 time=19690 us)
   2851    TABLE ACCESS FULL T1 (cr=295 pr=0 pw=0 time=5570 us cost=33 …)
      0    TABLE ACCESS FULL T2 (cr=408 pr=0 pw=0 time=0 us cost=116 …)
```

“Not Null = More Access Paths”

Not Null constraints are important

```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select * from all_objects;
Table created.

ops$tkyte%ORA11GR2> create index t_idx on t(object_type);
Index created.

ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'T' );
PL/SQL procedure successfully completed.
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> select count(*) from t;

Execution Plan
----------------------------------------------------------
Plan hash value: 2966233522

-------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |   286   (1)| 00:00:04 |
|   1 |  SORT AGGREGATE    |      |     1 |            |          |
|   2 |   TABLE ACCESS FULL| T    | 71482 |   286   (1)| 00:00:04 |
-------------------------------------------------------------------------------
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> alter table t modify object_type NOT NULL;
Table altered.

ops$tkyte%ORA11GR2> select count(*) from t;

Execution Plan
----------------------------------------------------------
Plan hash value: 1058879072

-------------------------------------------------------------------------------
| Id  | Operation             | Name  | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |       |     1 |    55   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE       |       |     1 |            |          |
|   2 |   INDEX FAST FULL SCAN| T_IDX | 71482 |    55   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> alter table t modify object_type NULL;
ops$tkyte%ORA11GR2> drop index t_idx;
ops$tkyte%ORA11GR2> create index t_idx on t(object_type,0);

ops$tkyte%ORA11GR2> select count(*) from t;


-----------------------------------------------------------------------------
| Id  | Operation             | Name  | Rows  | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |       |     1 |    61   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE       |       |     1 |            |          |
|   2 |   INDEX FAST FULL SCAN| T_IDX | 71482 |    61   (0)| 00:00:01 |
-----------------------------------------------------------------------------
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select case when mod(rownum,1000)=0 then null else object_type end otype,
  4         stage.*
  5    from stage
  6  /
ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'T' );
ops$tkyte%ORA11GR2> create index t_idx on t(otype);
ops$tkyte%ORA11GR2> analyze index t_idx validate structure;
ops$tkyte%ORA11GR2> select lf_rows, (select count(*) from t) ,
  2         lf_rows- (select count(*) from t) diff
  3    from index_stats;

   LF_ROWS (SELECTCOUNT(*)FROMT)       DIFF
---------- -------------------- ----------
     71411                71482        -71
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> select * from t where otype is null;

-------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |    71 |  7526 |   309   (1)| 00:00:04 |
|*  1 |  TABLE ACCESS FULL| T    |    71 |  7526 |   309   (1)| 00:00:04 |
-------------------------------------------------------------------------------

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

   1 - filter("OTYPE" IS NULL)
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> select /*+ index( t t_idx ) */ * from t where otype is null;

-------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |    71 |  7526 |   309   (1)| 00:00:04 |
|*  1 |  TABLE ACCESS FULL| T    |    71 |  7526 |   309   (1)| 00:00:04 |
-------------------------------------------------------------------------------

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

   1 - filter("OTYPE" IS NULL)
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> drop index t_idx;

Index dropped.

ops$tkyte%ORA11GR2> create index t_idx on t(otype,0);

Index created.
```

Not Null constraints are important

```
ops$tkyte%ORA11GR2> select * from t where otype is null;
-------------------------------------------------------------------------------
| Id  | Operation                    | Name  | Rows  | Bytes | Cost (%CPU)| Time
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |       |    71 |  7526 |     6   (0)| 00:0
|   1 |  TABLE ACCESS BY INDEX ROWID| T     |    71 |  7526 |     6   (0)| 00:0
|*  2 |   INDEX RANGE SCAN           | T_IDX |    71 |       |     2   (0)| 00:0
-------------------------------------------------------------------------------

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

   2 - access("OTYPE" IS NULL)
```

“Tell us how the tables relate and we can remove them from the plan…”

Referential / Entity Integrity Constrains are Important

```
ops$tkyte%ORA11GR2> create table emp
  2  as
  3  select *
  4    from scott.emp;
ops$tkyte%ORA11GR2> create table dept
  2  as
  3  select *
  4    from scott.dept;

ops$tkyte%ORA11GR2> begin
  2       dbms_stats.set_table_stats
  3           ( user, 'EMP', numrows=>1000000, numblks=>100000 );
  4       dbms_stats.set_table_stats
  5           ( user, 'DEPT', numrows=>100000, numblks=>10000 );
  6  end;
  7  /
```

Referential / Entity Integrity Constrains are Important

```
ops$tkyte%ORA11GR2> select ename
  2    from emp, dept
  3   where emp.deptno = dept.deptno;
-------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes |TempSpc| Cost (%CPU)| Time
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |  1000K|    31M|       | 31468   (1)| 00:06:
|*  1 |  HASH JOIN         |      |  1000K|    31M|  2448K| 31468   (1)| 00:06:
|   2 |   TABLE ACCESS FULL| DEPT |   100K|  1269K|       |  2713   (1)| 00:00:
|   3 |   TABLE ACCESS FULL| EMP  |  1000K|    19M|       | 27116   (1)| 00:05:
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   1 - access("EMP"."DEPTNO"="DEPT"."DEPTNO")
```

Referential / Entity Integrity Constrains are Important

```
ops$tkyte%ORA11GR2> alter table dept add constraint dept_pk primary key(deptno);

Table altered.

ops$tkyte%ORA11GR2> alter table emp add constraint emp_fk_dept foreign key(deptno)
  2  references dept(deptno);

Table altered.
```

Referential / Entity Integrity Constrains are Important

```
ops$tkyte%ORA11GR2> select ename
  2    from emp, dept
  3   where emp.deptno = dept.deptno;

-------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      | 50000 |   976K| 27117   (1)| 00:05:26 |
|*  1 |  TABLE ACCESS FULL| EMP  | 50000 |   976K| 27117   (1)| 00:05:26 |
-------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   1 - filter("EMP"."DEPTNO" IS NOT NULL)
```

“Tell us how the tables relate and we have more access paths available…”

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> create table emp
  2  as
  3  select *
  4    from scott.emp;
ops$tkyte%ORA11GR2> create table dept
  2  as
  3  select *
  4    from scott.dept;


ops$tkyte%ORA11GR2> begin
  2       dbms_stats.set_table_stats
  3           ( user, 'EMP', numrows=>1000000, numblks=>100000 );
  4       dbms_stats.set_table_stats
  5           ( user, 'DEPT', numrows=>100000, numblks=>10000 );
  6  end;
  7  /
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> create materialized view mv
  2  enable query rewrite
  3  as
  4  select dept.deptno, dept.dname, count (*)
  5    from emp, dept
  6   where emp.deptno = dept.deptno
  7   group by dept.deptno, dept.dname;

ops$tkyte%ORA11GR2> begin
  2    dbms_stats.set_table_stats
  3    ( user, 'MV', numrows=>100000, numblks=>10000 );
  4  end;
  5  /
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> select count(*) from emp;
  COUNT(*)
----------
        14
ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |       | 27112 (100)|          |
|   1 |  SORT AGGREGATE    |      |     1 |            |          |
|   2 |   TABLE ACCESS FULL| EMP  |  1000K| 27112   (1)| 00:05:26 |
-------------------------------------------------------------------------
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> alter table dept add constraint dept_pk primary key(deptno);
Table altered.

ops$tkyte%ORA11GR2> alter table emp add constraint emp_fk_dept foreign key(deptno)
  2  references dept(deptno);
Table altered.

ops$tkyte%ORA11GR2> alter table emp modify deptno NOT NULL;
Table altered.
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> select count(*) from emp;
  COUNT(*)
----------
        14

ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------------------
| Id  | Operation                     | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |      |       |       |  2713 (100)|          |
|   1 |  SORT AGGREGATE               |      |     1 |    13 |            |          |
|   2 |   MAT_VIEW REWRITE ACCESS FULL| MV   |   100K|  1269K|  2713   (1)| 00:00:33 |
-------------------------------------------------------------------------------------
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> alter table emp drop constraint emp_fk_dept ;
Table altered.

ops$tkyte%ORA11GR2> alter table dept drop constraint dept_pk;
Table altered.

ops$tkyte%ORA11GR2> alter table emp modify deptno NULL;
Table altered.

ops$tkyte%ORA11GR2> insert into emp (empno,deptno) values ( 1, 1 );
1 row created.

ops$tkyte%ORA11GR2> exec dbms_mview.refresh( 'MV' );
PL/SQL procedure successfully completed.
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> alter table dept
  2  add constraint dept_pk primary key(deptno)
  3  RELY disable NOVALIDATE
  4  /

ops$tkyte%ORA11GR2> alter table emp
  2  add constraint emp_fk_dept
  3  foreign key(deptno) references dept(deptno)
  4  RELY disable NOVALIDATE
  5  /

ops$tkyte%ORA11GR2> alter table emp modify deptno not null NOVALIDATE;
```

Referential / Entity Integrity Constrains are Important
part II

```
query_rewrite_integrity                string      enforced

ops$tkyte%ORA11GR2> select count(*) from emp;
  COUNT(*)
----------
        15

ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |       | 27112 (100)|          |
|   1 |  SORT AGGREGATE    |      |     1 |            |          |
|   2 |   TABLE ACCESS FULL| EMP  |  1000K| 27112   (1)| 00:05:26 |
-------------------------------------------------------------------------
```

Referential / Entity Integrity Constrains are Important
part II

```
ops$tkyte%ORA11GR2> alter session set query_rewrite_integrity = trusted;
Session altered.

ops$tkyte%ORA11GR2> select count(*) from emp;
  COUNT(*)
----------
        14

ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------------------
| Id  | Operation                     | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |      |       |       |  2713 (100)|          |
|   1 |  SORT AGGREGATE               |      |     1 |    13 |            |          |
|   2 |   MAT_VIEW REWRITE ACCESS FULL| MV   |   100K|  1269K|  2713   (1)| 00:00:33 |
-------------------------------------------------------------------------------------
```

Dimensions are important

Dimensions

- Dimensions are facts
- Dimensions are more information
- Dimensions convey information to the optimizer
- The presence of Dimensions open up access paths that would not be otherwise available.
- Look familiar?
- The more metadata we assert to the database, the better it can do…

Dimensions

- Describes to Oracle how to “roll up” data
  - You have a DATE column
    - Which implies Month-Year
      - Which implies FY-Quarter
        - Which implies FY
      - Which implies Calendar Year Quarter
        - Which implies Calendar Year
  - So, if you have a materialized view at the Month-Year level…

Dimensions are important

```
ops$tkyte%ORA11GR2> create table sales
  2  as
  3  select trunc(sysdate,'year')+mod(rownum,366) trans_date,
  4         mod(rownum,100) cust_id,
  5         round(dbms_random.value(1,1000),2) sales_amount
  6    from (select level l from dual connect by level <= 350000)
  7  /
Table created.
```

A years worth of sales data...

For 100 customers...

With random sales amounts...

```
ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'SALES' );
PL/SQL procedure successfully completed.
```

Dimensions are important

```
ops$tkyte%ORA11GR2> create table time_hierarchy
  2  as
  3  select trans_date day,
  4         month_year,
  5         cast(
  6             to_char(month_year,'yyyy')+
  7              case when to_char(month_year,'mm')>=6 then 1 else 0 end ||'-'||
  8              (mod(to_char(add_months(month_year,+1),'q')+1, 4 )+1) as varchar2(6)
  9         ) fy_qtr,
 10         to_char(month_year,'yyyy')+
 11              case when to_char(month_year,'mm')>=6 then 1 else 0 end fy,
 12         to_char(month_year,'yyyy-q') cy_qtr,
 13         to_char(month_year,'yyyy') cy
 14    from (select distinct trans_date, trunc(trans_date,'mm') month_year from sales);
```

Trans_date is primary key (the day)

Month_year is a mapping of date to month in a year

FY_QTR is a mapping of date to fiscal year/qtr

FY is a mapping of date to fiscal year

CY_QTR is a mapping of date to calendar year/qtr

CY is a mapping of date to calendar year

Dimensions are important

```
ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'TIME_HIERARCHY' );
PL/SQL procedure successfully completed.

ops$tkyte%ORA11GR2> alter table time_hierarchy add constraint th_pk primary
key(day);
Table altered.

ops$tkyte%ORA11GR2> alter table sales add constraint sales_fk
  2  foreign key(trans_date)
  3  references time_hierarchy(day);
Table altered.

ops$tkyte%ORA11GR2> alter table sales modify trans_date not null;
Table altered.
```

Dimensions are important

```
ops$tkyte%ORA11GR2> create materialized view mv
  2  build immediate
  3  refresh on demand
  4  enable query rewrite
  5  as
  6  select sales.cust_id, sum(sales.sales_amount) sales_amount,
             time_hierarchy.month_year
  7    from sales, time_hierarchy
  8   where sales.trans_date = time_hierarchy.day
  9   group by sales.cust_id, time_hierarchy.month_year
 10  /
Materialized view created.

ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'MV' );
PL/SQL procedure successfully completed.
```

Dimensions are important

```
ops$tkyte%ORA11GR2> select time_hierarchy.month_year, sum(sales_amount)
  2    from sales, time_hierarchy
  3   where sales.trans_date = time_hierarchy.day
  4   group by time_hierarchy.month_year
  5  /
…
13 rows selected.

ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------------------
| Id  | Operation                     | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |      |       |       |     5 (100)|          |
|   1 |  HASH GROUP BY                |      |    13 |   182 |     5  (20)| 00:00:01 |
|   2 |   MAT_VIEW REWRITE ACCESS FULL| MV   |  1250 | 17500 |     4   (0)| 00:00:01 |
-------------------------------------------------------------------------------------
```

Dimensions are important

```
ops$tkyte%ORA11GR2> select time_hierarchy.cy_qtr, sum(sales_amount)
  2    from sales, time_hierarchy
  3   where sales.trans_date = time_hierarchy.day
  4   group by time_hierarchy.cy_qtr;
…

ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------------------
| Id  | Operation             | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |                |       |       |   300 (100)|          |
|   1 |  HASH GROUP BY        |                |     5 |   180 |   300   (5)| 00:00:04 |
|*  2 |   HASH JOIN           |                |   366 | 13176 |   299   (5)| 00:00:04 |
|   3 |    VIEW               | VW_GBC_5       |   366 |  7686 |   295   (5)| 00:00:04 |
|   4 |     HASH GROUP BY     |                |   366 |  4758 |   295   (5)| 00:00:04 |
|   5 |      TABLE ACCESS FULL| SALES          |   350K|  4443K|   285   (1)| 00:00:04 |
|   6 |    TABLE ACCESS FULL  | TIME_HIERARCHY |   366 |  5490 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------
```

Dimensions are important

```
ops$tkyte%ORA11GR2> create dimension time_hierarchy_dim
  2       level day          is time_hierarchy.day
  3       level month_year   is time_hierarchy.month_year
  4       level cy_qtr       is time_hierarchy.cy_qtr
  5       level cy           is time_hierarchy.cy
  6       level fy_qtr       is time_hierarchy.fy_qtr
  7       level fy           is time_hierarchy.fy
  8  hierarchy cy_rollup
  9  ( day child of month_year child of cy_qtr child of cy)
 10  hierarchy fy_rollup
 11  ( day child of month_year child of fy_qtr child of fy)
 12  /

Dimension created.
```

Dimensions are important

```
ops$tkyte%ORA11GR2> alter session set query_rewrite_integrity=trusted;

ops$tkyte%ORA11GR2> select time_hierarchy.cy_qtr, sum(sales_amount)
  2    from sales, time_hierarchy
  3   where sales.trans_date = time_hierarchy.day
  4   group by time_hierarchy.cy_qtr;
…
ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-------------------------------------------------------------------------------------------
| Id  | Operation                      | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                |       |       |    10 (100)|          |
|   1 |  HASH GROUP BY                 |                |     5 |   145 |    10  (30)| 00:00:01 |
|*  2 |   HASH JOIN                    |                |  4423 |   125K|     9  (23)| 00:00:01 |
|   3 |    VIEW                        |                |    46 |   690 |     4  (25)| 00:00:01 |
|   4 |     HASH UNIQUE                |                |    46 |   690 |     4  (25)| 00:00:01 |
|   5 |      TABLE ACCESS FULL         | TIME_HIERARCHY |   366 |  5490 |     3   (0)| 00:00:01 |
|   6 |    MAT_VIEW REWRITE ACCESS FULL| MV             |  1250 | 17500 |     4   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

Dimensions are important

```
ops$tkyte%ORA11GR2> select time_hierarchy.fy, sum(sales_amount)
  2    from sales, time_hierarchy
  3   where sales.trans_date = time_hierarchy.day
  4   group by time_hierarchy.fy;
```

…

```
ops$tkyte%ORA11GR2> select * from table(dbms_xplan.display_cursor);
-----------------------------------------------------------------------------------------------
| Id  | Operation                      | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |                |       |       |    10 (100)|          |
|   1 |  HASH GROUP BY                 |                |     2 |    52 |    10  (30)| 00:00:01 |
|*  2 |   HASH JOIN                    |                |  1827 | 47502 |     9  (23)| 00:00:01 |
|   3 |    VIEW                        |                |    19 |   228 |     4  (25)| 00:00:01 |
|   4 |     HASH UNIQUE                |                |    19 |   228 |     4  (25)| 00:00:01 |
|   5 |      TABLE ACCESS FULL         | TIME_HIERARCHY |   366 |  4392 |     3   (0)| 00:00:01 |
|   6 |    MAT_VIEW REWRITE ACCESS FULL| MV             |  1250 | 17500 |     4   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------
```

Procrastination
Can be bad

Deferrable Constraints

- Constraint checking can be done
  - Immediately, at the statement level
  - Later, at commit time

Deferrable Constraints

- Deferred Unique/Primary key constraints
  - Use a non-unique index
  - No more index unique scan
  - Always an index range scan

Deferrable Constraints

- Deferring a constraint removes the ability of the optimizer to *use* that constraint
  - The constraint is not *necessarily true* anymore
  - That column *will be* NOT NULL when you commit, but it won't *necessarily* be NOT NULL in the middle of your transaction
  - That check constraint *will be* obeyed when you commit, but it won't *necessarily* be obeyed in the middle of your transaction

Deferred Constraints

```
ops$tkyte%ORA11GR2> create table t
  2  ( x int constraint x_not_null not null deferrable,
  3    y int constraint y_not_null not null,
  4    z varchar2(30)
  5  );
Table created.

ops$tkyte%ORA11GR2> insert into t(x,y,z)
  2  select rownum, rownum, rpad('x',30,'x')
  3    from all_users;
45 rows created.


ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'T' );
PL/SQL procedure successfully completed.
```

Deferred Constraints

```
ops$tkyte%ORA11GR2> create index t_idx on t(y);
Index created.

ops$tkyte%ORA11GR2> select count(*) from t;

Execution Plan
----------------------------------------------------------
Plan hash value: 995313729

-------------------------------------------------------------------------
| Id  | Operation         | Name  | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |       |     1 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE   |       |     1 |            |          |
|   2 |   INDEX FULL SCAN| T_IDX |    45 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------
```

Deferred Constraints

```
ops$tkyte%ORA11GR2> drop index t_idx;
Index dropped.

ops$tkyte%ORA11GR2> create index t_idx on t(x);
Index created.
```

Deferred Constraints

```
ops$tkyte%ORA11GR2> select count(*) from t;

Execution Plan
----------------------------------------------------------
Plan hash value: 2966233522

-------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |     3   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |      |     1 |            |          |
|   2 |   TABLE ACCESS FULL| T    |    45 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------
```

Deferred Constraints

```
ops$tkyte%ORA11GR2> alter table t drop constraint x_not_null;
Table altered.

ops$tkyte%ORA11GR2> alter table t modify x constraint x_not_null not null;
Table altered.
```

Deferred Constraints

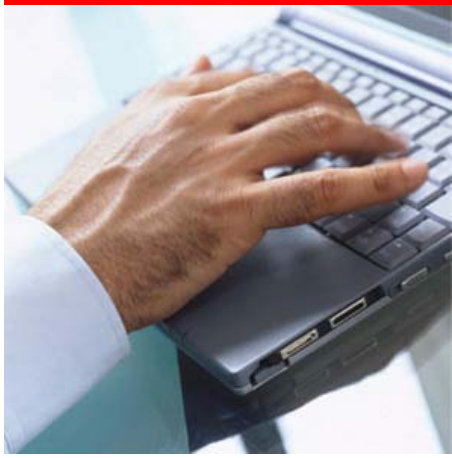
```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> select count(*) from t;
Execution Plan
----------------------------------------------------------
Plan hash value: 995313729
-------------------------------------------------------------------------
| Id  | Operation         | Name  | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |       |     1 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE   |       |     1 |            |          |
|   2 |   INDEX FULL SCAN| T_IDX |    45 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------
```

ORACLE OPEN WORLD

In Short

ORACLE

In Short…



- Datatypes Count
  - For more than just data integrity, not that data integrity wouldn't be *enough* by itself
- Proper Lengths matter
  - For more than just data integrity…
- Constraints are important
  - For OLTP
  - And *especially for performance* in a warehouse
- Metadata Matters
  - Dimensions, Constraints, NULLs, Datatypes – everything you can tell us about the data – helps us process the data

“Question Authority, Ask Questions”

Thomas.Kyte@oracle.com


ORACLE®