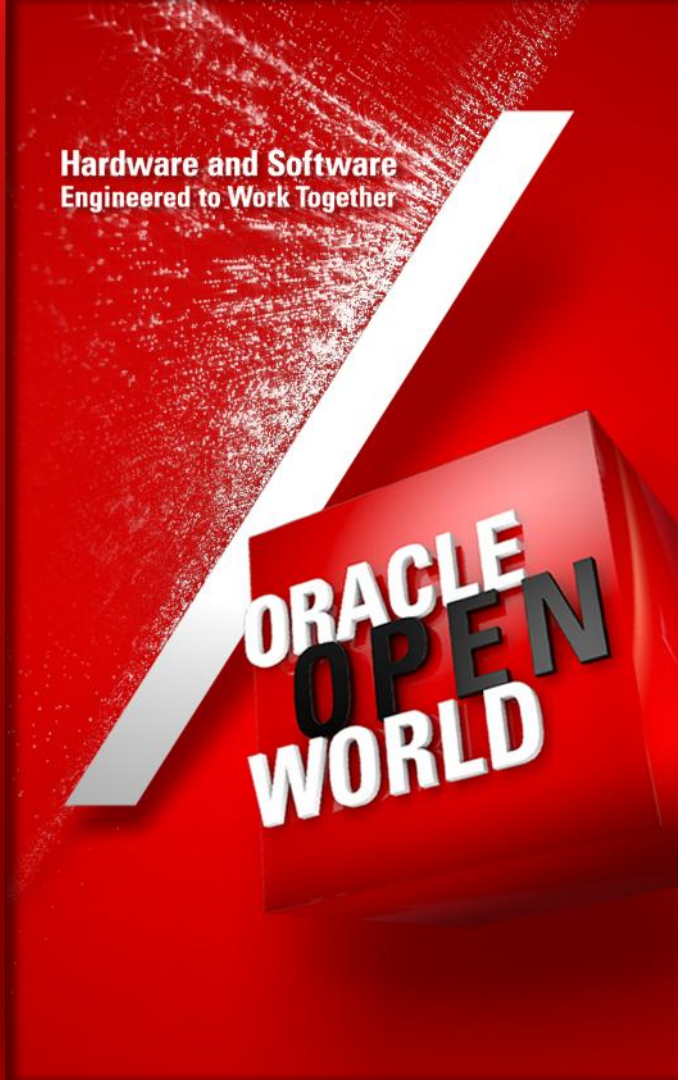

ORACLE®

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE

# Five Things about SQL and PL/SQL you might not have known about

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



# Program Agenda

- Warnings
- Better Statistics
- Optimizer Optimizations
- SQL*Net Compression
- Implicit Conversions Are *Evil*

# Warnings

# Warnings

- PL/SQL Compiler has been warning us since 10.1 (2004!)
- Not widely used
- Can be warnings or compile errors

# Warnings

- Severe: code might cause unexpected action or wrong results
- Performance: condition might cause performance issues
- Informational: code as written won't be wrong or slow – just bad code

# Severe

```
c##tkyte%CDB1> alter session set plsql_warnings='enable:severe';
Session altered.

c##tkyte%CDB1> create or replace procedure p
  2  as
  3          procedure substr
  4          is
  5          begin
  6                  null;
  7          end;
  8  begin
  9          null;
 10  end;
 11  /
SP2-0804: Procedure created with compilation warnings

c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
1/1      PLW-05018: unit P omitted optional AUTHID clause; default value
         DEFINER used

3/12     PLW-05004: identifier SUBSTR is also declared in STANDARD or is a
         SQL builtin
```

# Performance

```
c##tkyte%CDB1> alter session set plsql_warnings='enable:performance';
Session altered.

c##tkyte%CDB1> create or replace procedure p
  2  as
  3          l_string varchar2(5);
  4  begin
  5          for x in (select * from emp where empno = l_string)
  6          loop
  7                  null;
  8          end loop;
  9  end;
 10  /
SP2-0804: Procedure created with compilation warnings

c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
5/44     PLW-07204: conversion away from column type may result in
         sub-optimal query plan
```

# Informational

```
c##tkyte%CDB1> alter session set plsql_warnings='enable:informational';
Session altered.

c##tkyte%CDB1> create or replace procedure p
  2  as
  3  begin
  4          if (null is not null)
  5          then
  6                  dbms_output.put_line( 'hello world' );
  7          end if;
  8  end;
  9  /
SP2-0804: Procedure created with compilation warnings

c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
6/3      PLW-06002: Unreachable code
```

# My Current Favorite…

```
c##tkyte%CDB1> alter session set
                     plsql_warnings='enable:all,disable:5018,error:6009,error:7204';
Session altered.
c##tkyte%CDB1> create or replace procedure p
  2  as
  3  begin
  4          dbms_output.put_line( 'hello world' );
  5  exception
  6  when others
  7          then null;
  8  end;
  9  /
Warning: Procedure created with compilation errors.
c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
6/6      PLS-06009: procedure "P" OTHERS handler does not end in RAISE or
         RAISE_APPLICATION_ERROR
```

# Warnings

- Can be set at
  - The system level ALTER SYSTEM
  - The session level ALTER SESSION
  - Unit by unit
    ALTER PROCEDURE *P* COMPILE PLSQL_WARNINGS='…' REUSE SETTINGS;

# Better Statistics

# Better Statistics

- Wrong Plan => Wrong Cardinality
- Pipelined Functions => Wrong Cardinality by default
- We can do better – five ways to better statistics for pipelined functions
- http://www.oracle-developer.net/display.php?id=427
  Nice write up of some of them by **Adrian Billington**

# #1 Cardinality Hint (9i and above, undocumented)

```
c##tkyte%CDB1> create or replace type str2tblType as table of varchar2(30)
  2  /
Type created.
c##tkyte%CDB1> create or replace
  2  function str2tbl( p_str in varchar2, p_delim in varchar2 default ',' )
  3  return str2tblType
  4  PIPELINED
  5  as
  6      l_str      long default p_str || p_delim;
  7      l_n        number;
  8  begin
  9      loop
 10          l_n := instr( l_str, p_delim );
 11          exit when (nvl(l_n,0) = 0);
 12          pipe row( ltrim(rtrim(substr(l_str,1,l_n-1))) );
 13          l_str := substr( l_str, l_n+1 );
 14      end loop;
 15  end;
 16  /
Function created.
```

# #1 Cardinality Hint (9i and above, undocumented)

```
c##tkyte%CDB1> variable x varchar2(15)
c##tkyte%CDB1> exec :x := '1,2,3,a,b,c'

PL/SQL procedure successfully completed.

c##tkyte%CDB1> select * from table(str2tbl(:x));

COLUMN_VALUE
------------------------------
1
2
3
a
b
c


6 rows selected.
```

# #1 Cardinality Hint (9i and above, undocumented)

```
c##tkyte%CDB1> select * from table(dbms_xplan.display_cursor);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------
SQL_ID  ddk1tv9s5pzq5, child number 0
-------------------------------------
select * from table(str2tbl(:x))

Plan hash value: 2407808827

-----------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    29 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |  8168 | 16336 |    29   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------
```

# #1 Cardinality Hint (9i and above, undocumented)

```
c##tkyte%CDB1> select * from table(dbms_xplan.display_cursor);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------
-------
SQL_ID  bd2f8rh30z3ww, child number 0
-------------------------------------
select /*+ cardinality(sq 10) */ * from table(str2tbl(:x)) sq

Plan hash value: 2407808827

-----------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    29 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |    10 |    20 |    29   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------


13 rows selected.
```

# #1 Cardinality Hint (9i and above, undocumented)

```
select * from t where object_name in (select * from table(str2tbl(:x)))

Plan hash value: 1957688699

-------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |   431 (100)|          |
|*  1 |  HASH JOIN RIGHT SEMI               |         |     2 |   232 |   431   (1)| 00:00:01 |
|   2 |   COLLECTION ITERATOR PICKLER FETCH| STR2TBL |  8168 | 16336 |    29   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL                 | T       | 87322 |  9721K|   401   (1)| 00:00:01 |
-------------------------------------------------------------------------------------------------

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

   1 - access("OBJECT_NAME"=VALUE(KOKBF$))
```

# #1 Cardinality Hint (9i and above, undocumented)

```
select * from t where object_name in (select /*+ cardinality(sq 10) */
* from table(str2tbl(:x)) sq)

Plan hash value: 3519658119

-----------------------------------------------------------------------------------------------
| Id  | Operation                           | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |          |       |       |    44 (100)|          |
|   1 |  NESTED LOOPS                       |          |       |       |            |          |
|   2 |   NESTED LOOPS                      |          |    17 |  1972 |    44   (0)| 00:00:01 |
|   3 |    SORT UNIQUE                      |          |    10 |    20 |    29   (0)| 00:00:01 |
|   4 |     COLLECTION ITERATOR PICKLER FETCH| STR2TBL |    10 |    20 |    29   (0)| 00:00:01 |
|*  5 |    INDEX RANGE SCAN                 | T_IDX    |     2 |       |     2   (0)| 00:00:01 |
|   6 |   TABLE ACCESS BY INDEX ROWID       | T        |     2 |   228 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------

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

   5 - access("OBJECT_NAME"=VALUE(KOKBF$))
```

# #2 OPT_ESTIMATE Hint (10g and above, undocumented, used by SQL Profiles)

```
c##tkyte%CDB1> select 10/8168 from dual;

   10/8168
----------
 .00122429

select /*+ opt_estimate(table, sq, scale_rows=0.00122429) */ *
  from table(str2tbl(:x)) sq

Plan hash value: 2407808827

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    29 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |    10 |    20 |    29   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

# #3 Extensible Optimizer (10g and above)

```
c##tkyte%CDB1> CREATE OR REPLACE TYPE str2tbl_stats
  2  AS OBJECT
  3  (
  4     x NUMBER,
  5
  6     STATIC FUNCTION ODCIGetInterfaces
  7     ( p_interfaces OUT SYS.ODCIObjectList
  8     ) RETURN NUMBER,
  9
 10     STATIC FUNCTION ODCIStatsTableFunction
 11     ( p_function IN  SYS.ODCIFuncInfo,
 12       p_stats    OUT SYS.ODCITabFuncStats,
 13       p_args     IN  SYS.ODCIArgDescList,
 14       p_str      IN  varchar2 default NULL,
 15       p_delim    IN  varchar2  default ','
 16     ) RETURN NUMBER
 17  );
 18  /

Type created.
```

# #3 Extensible Optimizer (10g and above)

```
c##tkyte%CDB1> CREATE or replace TYPE BODY str2tbl_stats
  2  AS
  3
  4     STATIC FUNCTION ODCIGetInterfaces (
  5                     p_interfaces OUT SYS.ODCIObjectList
  6                     ) RETURN NUMBER IS
  7     BEGIN
  8        p_interfaces :=
  9                  SYS.ODCIObjectList( SYS.ODCIObject ('SYS', 'ODCISTATS2'));
 10        RETURN ODCIConst.success;
 11     END ODCIGetInterfaces;
 12
 13     STATIC FUNCTION ODCIStatsTableFunction (
 14                     p_function IN  SYS.ODCIFuncInfo,
 15                     p_stats    OUT SYS.ODCITabFuncStats,
 16                     p_args     IN  SYS.ODCIArgDescList,
 17                     p_str      IN  varchar2 default NULL,
 18                     p_delim    IN  varchar2  default ','
 19                     ) RETURN NUMBER IS
 20     BEGIN
 21        p_stats := SYS.ODCITabFuncStats
 22            ( nvl( length(p_str)-length(replace(p_str,p_delim,''))+1, 10) );
 23        RETURN ODCIConst.success;
 24     END ODCIStatsTableFunction;
 25  END;
 26  /
```

# #3 Extensible Optimizer (10g and above)

```
c##tkyte%CDB1> associate statistics with functions str2tbl using str2tbl_stats;

Statistics associated.
```

# #3 Extensible Optimizer (10g and above)

```
Select * from table ( str2tbl( :x||'',',') )

Plan hash value: 2407808827

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    29 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |    10 |    20 |    29   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

# #3 Extensible Optimizer (10g and above)

```
Select * from table ( str2tbl( 'a,b,c',',') )

Plan hash value: 2407808827

-----------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    29 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |     3 |     6 |    29   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------
```

# #4 Dynamic Sampling (11gR1 and above)

```
select /*+ dynamic_sampling( sq, 2 ) */ * from table( str2tbl(:x,',') ) sq

Plan hash value: 2407808827

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |         |       |       |    11 (100)|          |
|   1 |  COLLECTION ITERATOR PICKLER FETCH| STR2TBL |     6 |    12 |    11   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Note
-----
   - dynamic sampling used for this statement (level=2)


(must be hinted)
```

# #5 Cardinality Feedback (11gR2 and above)

```
with sq as (select /*+ materialize */ *    from table( str2tbl( :x ) )
) select * from sq

Plan hash value: 630596523

-------------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                       | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                            |       |       |    32 (100)|          |
|   1 |  TEMP TABLE TRANSFORMATION         |                            |       |       |            |          |
|   2 |   LOAD AS SELECT                   |                            |       |       |            |          |
|   3 |    COLLECTION ITERATOR PICKLER FETCH| STR2TBL                    |  8168 | 16336 |    29   (0)| 00:00:01 |
|   4 |   VIEW                             |                            |  8168 |   135K|     3   (0)| 00:00:01 |
|   5 |    TABLE ACCESS FULL               | SYS_TEMP_0FD9D6652_20F4CB  |  8168 | 16336 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------


18 rows selected.
```

# #5 Cardinality Feedback (11gR2 and above)

```
with sq as (select /*+ materialize */ *    from table( str2tbl( :x ) )
) select * from sq

Plan hash value: 630596523

-----------------------------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name                       | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                            |       |       |    32 (100)|          |
|   1 |  TEMP TABLE TRANSFORMATION          |                            |       |       |            |          |
|   2 |   LOAD AS SELECT                    |                            |       |       |            |          |
|   3 |    COLLECTION ITERATOR PICKLER FETCH| STR2TBL                    |  8168 | 16336 |    29   (0)| 00:00:01 |
|   4 |   VIEW                              |                            |     6 |   102 |     3   (0)| 00:00:01 |
|   5 |    TABLE ACCESS FULL                | SYS_TEMP_0FD9D6653_20F4CB  |     6 |    12 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------------

Note
-----
   - cardinality feedback used for this statement


22 rows selected.
```

# Optimizer Optimizations

# Optimizer Optimizations

- You thought this would be about SQL, it isn't
- Since 10.1, PL/SQL has used an optimizing compiler
  - *"In even rarer cases, PL/SQL **might raise an exception earlier than expected** or **not at all**." (PL/SQL Language Reference)*

# Optimizer Optimizations

- Three levels – PLSQL_OPTIMIZE_LEVEL
  - *1: no code rearranging, code is “as is”*
  - *2: code rearranging possible, many optimizations such as implicit array fetch added*
  - *3: aggressive code rearranging*

# Optimizer Optimizations

```
c##tkyte%CDB1> alter session set plsql_optimize_level=1;
Session altered.

c##tkyte%CDB1> create or replace procedure p
  2  as
  3  begin
  4          for x in ( select * from t )
  5          loop
  6                  null;
  7          end loop;
  8  end;
  9  /
Procedure created.

c##tkyte%CDB1> exec dbms_monitor.session_trace_enable;
PL/SQL procedure successfully completed.

c##tkyte%CDB1> exec p
PL/SQL procedure successfully completed.
```

# Optimizer Optimizations

```
SELECT * FROM T


call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch    87323      0.81       0.85       1438      87325          0       87322
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total    87325      0.81       0.85       1438      87325          0       87322
```

# Optimizer Optimizations

```
c##tkyte%CDB1> alter procedure p compile plsql_optimize_level=2;
Procedure altered.

c##tkyte%CDB1> exec dbms_monitor.session_trace_enable;
PL/SQL procedure successfully completed.

c##tkyte%CDB1> exec p
PL/SQL procedure successfully completed.

SELECT * FROM T

call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch      874      0.25       0.28       1438       2303          0       87322
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total      876      0.25       0.28       1438       2303          0       87322
```

# Optimizer Optimizations, code rearranging

```
c##tkyte%CDB1> alter session set Plsql_Warnings = 'Enable:All'
  2  /
Session altered.

c##tkyte%CDB1> alter session set Plsql_Optimize_Level = 2
  2  /
Session altered.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> create or replace procedure p
  2  authid definer
  3  as
  4    function Is_Number(x in varchar2) return varchar2
  5    is
  6      n number;
  7    begin
  8      n := To_Number(x);
  9      return 'Y';
 10    exception
 11      when value_error then
 12        return 'N';
 13    end Is_Number;
 14
 15  begin
 16    DBMS_Output.Put_Line(Is_Number('1'));
 17    DBMS_Output.Put_Line(Is_Number('a'));
 18  end p;
 19  /
Procedure created.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> SHOW ERRORS
No errors.
c##tkyte%CDB1> exec p
Y
N

PL/SQL procedure successfully completed.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> alter procedure p compile plsql_optimize_level=3;

SP2-0805: Procedure altered with compilation warnings

c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
4/3      PLW-06006: uncalled procedure "IS_NUMBER" is removed.
16/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
17/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
c##tkyte%CDB1> exec p;
Y
Y

PL/SQL procedure successfully completed.
```

# Optimizer Optimizations

- TO_NUMBER is “pure”
- It has no side effects
- It might raise an exception, but that isn’t a side effect.
- N is an unused variable
- The code becomes simply return ‘Y’

```
c##tkyte%CDB1> create or replace procedure p
  2  authid definer
  3  as
  4    function Is_Number(x in varchar2)
                 return varchar2
  5    is
  6      n number;
  7    begin
  8      n := To_Number(x);
  9      return 'Y';
 10    exception
 11      when value_error then
 12        return 'N';
 13    end Is_Number;
 14
 15  begin
 16    DBMS_Output.Put_Line(Is_Number('1'));
 17    DBMS_Output.Put_Line(Is_Number('a'));
 18  end p;
 19  /
Procedure created.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> create or replace procedure p
  2  authid definer
  3  as
  4    function Is_Number(x in varchar2) return varchar2
  5    is
  6      n number;
  7    begin
  8      n := To_Number(x);
  9      if ( n = -1 )
 10      then
 11          return 'y'; -- what about 'Y'??
 12      else
 13          return 'Y';
 14      end if;
 15    exception
 16      when value_error then
 17        return 'N';
 18    end Is_Number;
 19  begin
 20    DBMS_Output.Put_Line(Is_Number('1'));
 21    DBMS_Output.Put_Line(Is_Number('a'));
 22  end p;
 23  /
```

# Optimizer Optimizations

```
c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
4/3      PLW-06006: uncalled procedure "IS_NUMBER" is removed.
20/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
21/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
c##tkyte%CDB1> exec p;
Y
N

PL/SQL procedure successfully completed.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> create or replace procedure p
  2  authid definer
  3  as
  4    function Is_Number(x in varchar2) return varchar2
  5    is
  6      n number;
  7    begin
  8      n := To_Number(x);
  9      if ( n = -1 )
 10      then
 11          return 'Y';
 12      else
 13          return 'Y';
 14      end if;
 15    exception
 16      when value_error then
 17        return 'N';
 18    end Is_Number;
 19  begin
 20    DBMS_Output.Put_Line(Is_Number('1'));
 21    DBMS_Output.Put_Line(Is_Number('a'));
 22  end p;
 23  /
```

# Optimizer Optimizations

```
c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
4/3      PLW-06006: uncalled procedure "IS_NUMBER" is removed.
20/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
21/3     PLW-06005: inlining of call of procedure 'IS_NUMBER' was done
c##tkyte%CDB1> exec p;
Y
Y

PL/SQL procedure successfully completed.
```

# Optimizer Optimizations

```
c##tkyte%CDB1> create or replace procedure p
  2  authid definer
  3  as
  4      function Is_Number(X in varchar2) return varchar2
  5      is
  6          procedure Check_Number(Y in number)
  7          is
  8          begin
  9              null;
 10          end;
 11      begin
 12              pragma Inline(Check_Number, 'No');
 13              Check_Number(To_Number(X));
 14              return 'Y';
 15      exception
 16         when value_error
 17      then
 18          return 'N';
 19      end Is_Number;
 20  begin
 21    DBMS_Output.Put_Line(Is_Number('1'));
 22    DBMS_Output.Put_Line(Is_Number('a'));
 23  end p;
 24  /
```

# Optimizer Optimizations

```
c##tkyte%CDB1> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
13/12    PLW-06008: call of procedure 'CHECK_NUMBER' will not be inlined
c##tkyte%CDB1> exec p
Y
N

PL/SQL procedure successfully completed.
```

# SQLNet Compression

ORACLE

# SQLNet Compression

- How you retrieve the data matters
- Not all result sets are the same – even if they have the same data

# SQLNet Compression

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

ops$tkyte%ORA11GR2> begin
  2          dbms_stats.gather_table_stats( user, 'T' );
  3  end;
  4  /
PL/SQL procedure successfully completed.
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> set arraysize 15

ops$tkyte%ORA11GR2> set autotrace traceonly statistics
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t;
72228 rows selected.


Statistics
----------------------------------------------------------
       5794  consistent gets
    8015033  bytes sent via SQL*Net to client
      53385  bytes received via SQL*Net from client
       4817  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t order by timestamp;
72228 rows selected.


Statistics
----------------------------------------------------------
       1031  consistent gets
    3427630  bytes sent via SQL*Net to client
      53385  bytes received via SQL*Net from client
       4817  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t order by timestamp,
object_type, owner;
72228 rows selected.


Statistics
----------------------------------------------------------
       1031  consistent gets
    3280011  bytes sent via SQL*Net to client
      53385  bytes received via SQL*Net from client
       4817  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> set arraysize 100
ops$tkyte%ORA11GR2> set autotrace traceonly statistics
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t;
72228 rows selected.


Statistics
----------------------------------------------------------
       1842  consistent gets
    7482943  bytes sent via SQL*Net to client
       8362  bytes received via SQL*Net from client
        724  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t order by timestamp;
72228 rows selected.


Statistics
----------------------------------------------------------
       1031  consistent gets
    2907819  bytes sent via SQL*Net to client
       8362  bytes received via SQL*Net from client
        724  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

```
ops$tkyte%ORA11GR2> select * from t order by timestamp,
object_type, owner;
72228 rows selected.


Statistics
----------------------------------------------------------
       1031  consistent gets
    2760200  bytes sent via SQL*Net to client
       8362  bytes received via SQL*Net from client
        724  SQL*Net roundtrips to/from client
      72228  rows processed
```

# SQLNet Compression

| | No Order 15 | Some Order 15 | Very Ordered 15 | No Order 100 | Some Order 100 | Very Ordered 100 |
|---|---|---|---|---|---|---|
| **Bytes Sent** | 8 m | 3.4 m | 3.2 m | 7.4 m | 2.9 m | 2.7 m |
| **% of original** | 100% | 43% | 41% | 93% | 36% | 34% |
| **Consistent Gets** | 5794 | 1031 | 1031 | 1842 | 1031 | 1031 |

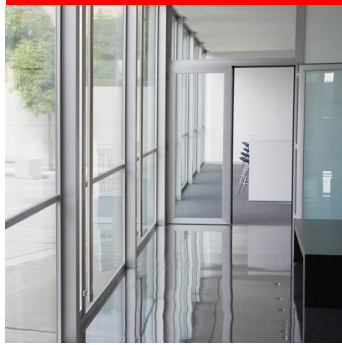
```
ops$tkyte%ORA11GR2> select round(1031*8/1024,2) from dual;

ROUND(1031*8/1024,2)
--------------------
                8.05
```

# SQLNet Compression

| | No Order 1000 | Some Order 1000 | Very Ordered 1000 | No Order 100 | Some Order 100 | Very Ordered 100 |
|---|---|---|---|---|---|---|
| **Bytes Sent** | 7.3 m | 2.8 m | 2.6 m | 7.4 m | 2.9 m | 2.7 m |
| **% of original** | 92% | 35% | 33% | 93% | 36% | 34% |
| **Consistent Gets** | 1105 | 1031 | 1031 | 1842 | 1031 | 1031 |

# Implicit conversions are evil

# Implicit conversions are evil



- SQL/PLSQL are too ‘user friendly’ – unlike C which complained about everything
- Implicit conversions of strings to numbers, strings to dates, raw to string, etc are probably the #2 cause of bugs I see time and time again
- Even worse are the implicit conversions that rely on default NLS settings!

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure inj( p_date in date )
  2  as
  3          l_rec   all_users%rowtype;
  4          c       sys_refcursor;
  5          l_query long;
  6  begin
  7          l_query := '
  8          select *
  9            from all_users
 10           where created = ''' ||p_date ||''''';   -- DOUBLE implicit conversion!
 11
 12          dbms_output.put_line( l_query );
 13          open c for l_query;
 14
 15          for i in 1 .. 5
 16          loop
 17                  fetch c into l_rec;
 18                  exit when c%notfound;
 19                  dbms_output.put_line( l_rec.username || '.....' );
 20          end loop;
...
```

# Implicit conversions are evil

```
  7          l_query := '
  8          select *
  9            from all_users
 10           where created = ''' ||p_date ||''''';
```

***Creates a query that is semantically equivalent to:***

```
Where created = to_date( to_char( date-field ) );
```

***Two implicit conversions – both of which rely on the NLS_DATE settings!***

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> exec inj( sysdate )

          select *
            from all_users
           where created = '04-OCT-11'

PL/SQL procedure successfully completed.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> alter session set
  2  nls_date_format = 'dd-mon-yyyy''' or ''a'' = ''a''';

Session altered.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> exec inj( sysdate )

         select *
           from all_users
          where created = '04-oct-2011' or 'a' = 'a'
A.....
EBRAPP.....
EBRTBLS.....
UTIL.....
USER2.....

PL/SQL procedure successfully completed.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> alter session set
  2  nls_date_format = '"''union select tname,0,null from tab--"';
Session altered.


Select *
  from all_users
 where created = ''union select tname,0,null from tab--'
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create user a identified by a;

User created.

ops$tkyte%ORA11GR2> grant create session, create procedure to a;

Grant succeeded.

ops$tkyte%ORA11GR2> grant execute on inj to a;

Grant succeeded.

ops$tkyte%ORA11GR2> desc t
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 X                                                  NUMBER(38)
```

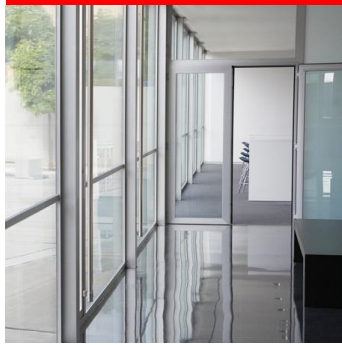
# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> connect a/a
Connected.

a%ORA11GR2> create or replace function f return varchar2
  2  authid current_user
  3  as
  4          pragma autonomous_transaction;
  5  begin
  6          execute immediate 'drop table t';
  7          return null;
  8  end;
  9  /

Function created.

a%ORA11GR2> grant execute on f to ops$tkyte;

Grant succeeded.
```

# Implicit conversions are evil

```
a%ORA11GR2> alter session set
  2  nls_date_format = '"'' or ops$tkyte.f() = ''a"';

Session altered.

a%ORA11GR2> exec ops$tkyte.inj( sysdate )

        select *
          from all_users
         where created = '' or ops$tkyte.f() = 'a'

PL/SQL procedure successfully completed.

a%ORA11GR2> connect /
Connected.
ops$tkyte%ORA11GR2> desc t
ERROR:
ORA-04043: object t does not exist
```

# Implicit conversions are evil



- What about performance?
  - Repeated conversions
  - Access path reductions
  - Partition elimination eliminated

```
SQL> alter session set Plsql_Warnings = 'enable:all';
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3      l_date  varchar2(30) := '01-jan-2011';
  4      l_start number := dbms_utility.get_cpu_time;
  5  begin
  6      for i in 1 .. 10
  7      loop
  8         for x in ( select owner, object_name
  9                      from big_table.big_table
 10                     where created = l_date )
 11         loop
 12             null;
 13         end loop;
 14      end loop;
 15      dbms_output.put_line( 'CPU: ' ||
 16      to_char( dbms_utility.get_cpu_time-l_start ) );
 17  end;
 18  /
SP2-0804: Procedure created with compilation warnings
ops$tkyte%ORA11GR2> exec p
CPU: 132
```

# Implicit conversions are evil

```
…
  7      loop
  8         for x in ( select owner, object_name
  9                      from big_table.big_table
 10                     where created = l_date )
 11         loop
 12             null;
 13         end loop;
…
```

```
ops$tkyte%ORA11GR2> show errors procedure p
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
10/36    PLW-07204: conversion away from column type may result in
         sub-optimal query plan
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3      l_date  date := to_date('01-jan-2011','dd-mon-yyyy');
  4      l_start number := dbms_utility.get_cpu_time;
  5  begin
  6      for i in 1 .. 10
  7      loop
  8         for x in ( select owner, object_name
  9                      from big_table.big_table
 10                     where created = l_date )
 11         loop
 12             null;
 13         end loop;
 14      end loop;
 15      dbms_output.put_line( 'CPU: ' ||
 16      to_char( dbms_utility.get_cpu_time-l_start ) );
 17  end;
 18  /
Procedure created.
ops$tkyte%ORA11GR2> exec p
CPU: 94      30% less CPU in this case
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create table t
  2  ( x varchar2(20) constraint t_pk primary key,
  3    y varchar2(30)
  4  );
Table created.

ops$tkyte%ORA11GR2> insert into t
  2  select user_id, username
  3    from all_users;
47 rows created.

ops$tkyte%ORA11GR2> commit;
Commit complete.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3          l_rec t%rowtype;
  4          l_key number := 5;
  5  begin
  6          select * into l_rec from t where x = l_key;
  7          for x in (select plan_table_output
  8                      from TABLE( dbms_xplan.display_cursor() ) )
  9          loop
 10                  dbms_output.put_line( x.plan_table_output );
 11          end loop;
 12  end;
 13  /

SP2-0804: Procedure created with compilation warnings
```

# Implicit conversions are evil

```
…
  5  begin
  6          select * into l_rec from t where x = l_key;
  7          for x in (select plan_table_output
…

ops$tkyte%ORA11GR2> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
6/42     PLW-07204: conversion away from column type may result in
         sub-optimal query plan
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> exec p

SQL_ID  18796jgha0hwz, child number 0
-------------------------------------
SELECT * FROM T WHERE X = :B1

Plan hash value: 1601196873

-----------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |       |       |     3 (100)|          |
|*  1 |  TABLE ACCESS FULL| T    |     1 |    29 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
1 - filter(TO_NUMBER("X")=:B1)
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3          l_rec t%rowtype;
  4          l_key varchar2(5) := '5';
  5  begin
  6          select * into l_rec from t where x = l_key;
  7          for x in (select plan_table_output
  8                      from TABLE( dbms_xplan.display_cursor() ) )
  9          loop
 10                  dbms_output.put_line( x.plan_table_output );
 11          end loop;
 12  end;
 13  /
Procedure created.

ops$tkyte%ORA11GR2> show errors
No errors.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> exec p

SQL_ID  18796jgha0hwz, child number 1
-------------------------------------
SELECT * FROM T WHERE X = :B1

Plan hash value: 1303508680
-------------------------------------------------------------------------------
| Id  | Operation                   | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |      |       |       |     1 (100)|          |
|   1 |  TABLE ACCESS BY INDEX ROWID| T    |     1 |    29 |     1   (0)| 00:00:01 |
|*  2 |   INDEX UNIQUE SCAN         | T_PK |     1 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
2 - access("X"=:B1)
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> CREATE TABLE t
  2  (
  3    dt  date,
  4    x   int,
  5    y   varchar2(30)
  6  )
  7  PARTITION BY RANGE (dt)
  8  (
  9    PARTITION part1 VALUES LESS THAN(to_date('31-jan-2011', 'dd-mon-yyyy')),
 10    PARTITION part2 VALUES LESS THAN(to_date('28-feb-2011', 'dd-mon-yyyy'))
 11  )
 12  /

Table created.
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3      l_date timestamp := timestamp'2011-01-15 00:00:00.000';
  4      l_count number;
  5  begin
  6      select count(*) into l_count from t where dt = l_date;
  7
  8      for x in (select plan_table_output
  9                  from TABLE( dbms_xplan.display_cursor() ) )
 10      loop
 11          dbms_output.put_line( '.'||x.plan_table_output );
 12      end loop;
 13  end;
 14  /

SP2-0804: Procedure created with compilation warnings
```

# Implicit conversions are evil

```
…
  5  begin
  6      select count(*) into l_count from t where dt = l_date;
  7
…

SP2-0804: Procedure created with compilation warnings

ops$tkyte%ORA11GR2> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
6/47     PLW-07204: conversion away from column type may result in
         sub-optimal query plan
```

# Implicit conversions are evil

```
SQL_ID  0t5m83d3m67q7, child number 0
-------------------------------------
SELECT COUNT(*) FROM T WHERE DT = :B1

Plan hash value: 3225603066
-----------------------------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      |       |       |     2 (100)|          |       |       |
|   1 |  SORT AGGREGATE      |      |     1 |     9 |            |          |       |       |
|   2 |   PARTITION RANGE ALL|      |     1 |     9 |     2   (0)| 00:00:01 |     1 |     2 |
|*  3 |    TABLE ACCESS FULL | T    |     1 |     9 |     2   (0)| 00:00:01 |     1 |     2 |
-----------------------------------------------------------------------------------------------

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

   3 - filter(INTERNAL_FUNCTION("DT")=:B1)
```

# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3      l_date date := to_date( '2011-01-15', 'yyyy-mm-dd' );
  4      l_count number;
  5  begin
  6      select count(*) into l_count from t where dt = l_date;
  7
  8      for x in (select plan_table_output
  9                  from TABLE( dbms_xplan.display_cursor() ) )
 10      loop
 11          dbms_output.put_line( '.'||x.plan_table_output );
 12      end loop;
 13  end;
 14  /
Procedure created.
ops$tkyte%ORA11GR2> show errors
No errors.
```

# Implicit conversions are evil

```
.SQL_ID  0t5m83d3m67q7, child number 1
.-------------------------------------
.SELECT COUNT(*) FROM T WHERE DT = :B1
.
.Plan hash value: 3660200434
.
.-----------------------------------------------------------------------------------------------
.| Id  | Operation                | Name | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
.-----------------------------------------------------------------------------------------------
.|   0 | SELECT STATEMENT         |      |       |       |     2 (100)|          |       |       |
.|   1 |  SORT AGGREGATE          |      |     1 |     9 |            |          |       |       |
.|   2 |   PARTITION RANGE SINGLE|      |     1 |     9 |     2   (0)| 00:00:01 |   KEY |   KEY |
.|*  3 |    TABLE ACCESS FULL     | T    |     1 |     9 |     2   (0)| 00:00:01 |   KEY |   KEY |
.-----------------------------------------------------------------------------------------------
.
.Predicate Information (identified by operation id):
.---------------------------------------------------
.
.   3 - filter("DT"=:B1)
```

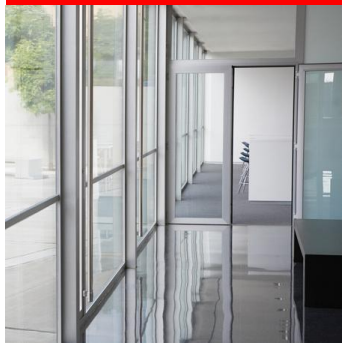
# Implicit conversions are evil

```
ops$tkyte%ORA11GR2> alter session set Plsql_Warnings = 'error:all';

ops$tkyte%ORA11GR2> create or replace procedure p authid definer
  2  as
  3      l_date timestamp := timestamp'2011-01-15 00:00:00.000';
  4      l_count number;
  5  begin
  6      select count(*) into l_count from t where dt = l_date;
  7
  8      for x in (select plan_table_output
  9                  from TABLE( dbms_xplan.display_cursor() ) )
 10      loop
 11          dbms_output.put_line( '.'||x.plan_table_output );
 12      end loop;
 13  end;
 14  /
```

# Implicit conversions are evil

```
Warning: Procedure created with compilation errors.

ops$tkyte%ORA11GR2> show errors
Errors for PROCEDURE P:

LINE/COL ERROR
-------- -----------------------------------------------------------------
6/47     PLS-07204: conversion away from column type may result in
         sub-optimal query plan

ops$tkyte%ORA11GR2> exec p
BEGIN p; END;

      *
ERROR at line 1:
ORA-06550: line 1, column 7:
PLS-00905: object OPS$TKYTE.P is invalid
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
```

# Implicit conversions are evil



- Lots and lots of bugs
  - What is ‘01/02/03’?
  - where column = to_char(sysdate) – what does that mean?

# Program Agenda

- Warnings
- Better Statistics
- Optimizer Optimizations
- SQL*Net Compression
- Implicit Conversions Are *Evil*


# Hardware and Software
## Engineered to Work Together

ORACLE