GLOC
Great Lakes Oracle Conference

DBMS_XPLAN
We have some xplaining to do

Dallas Deeds

Nationwide

MAY 15 & 16, 2019

CLEVELAND PUBLIC AUDITORIUM, CLEVELAND, OHIO

WWW.NEOOUG.ORG/GLOC

Dallas Deeds

- Work for Nationwide
- Oracle DBA since 1997, v 7.3
- Primarily focused on Oracle & Solaris/Linux performance since 2001
- In IT since 1995
- 1,200+ databases
  - 100+ multi-node RAC databases
  - Whole bunch of RAC one-node
  - Small number of standalones
  - 10.2 -> 18.5

GLOC
Great Lakes Oracle Conference

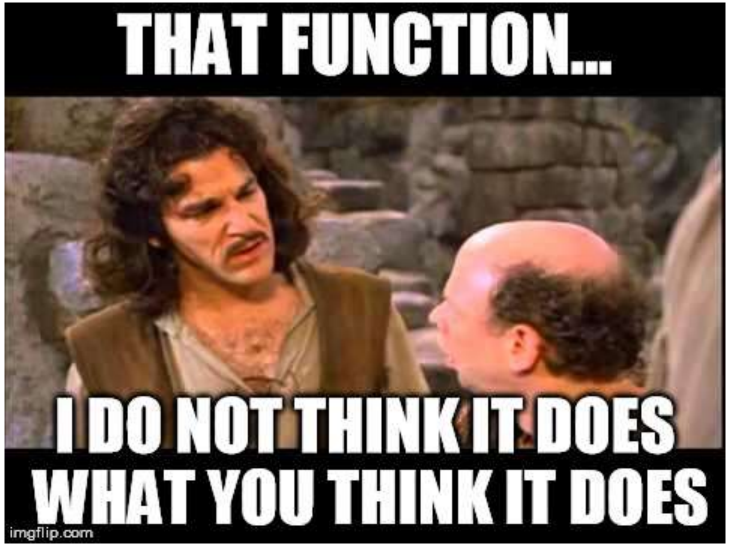
Why would I talk about dbms_xplan?

- Everyone uses it, right?
  - *most people use it, but most don't use it as well as they could*
- *Executions plans are fundamental*
- *Cloud!* *<the overlords require I say the word at least once>*
- Carol Dacko
- There are a lot of functions
  - *Do they do what you think they do?*
  - *What are the defaults giving you?*
  - *What are you giving to the functions?*
- There are a lot of formats
  - *What are they really displaying?*

GLOC
Great Lakes Oracle Conference

Some personal experience…

- Display()
- Cursor_child_no



GLOC
Great Lakes Oracle Conference

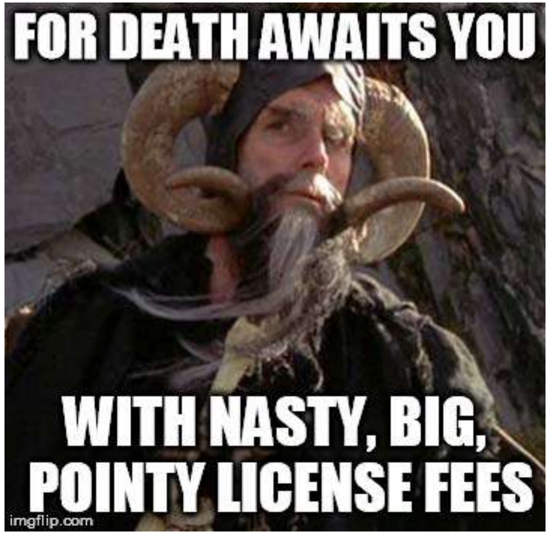
A bit of history

- First introduced in 9i
  - *Documentation only lists the display() function*
- Enhanced in each release since
  - *10g – 10 functions*
  - *11g – 21 functions*
  - *12c – 26 functions*
  - *18c – 30 functions*
  - *19c – 33 functions*

GLOC
Great Lakes Oracle Conference

Safety slide

- Some functions (display_awr, for example) require additional licensing



GLOC
Great Lakes Oracle Conference

Safety slide, cont'd

- NOTE: This slide represents my interpretation of documentation I have read. Oracle frequently changes their licensing and they are the final arbiter concerning what they license. Consult your Oracle representative for current licensing requirements.
- DISPLAY_AWR
  - *Requires the diagnostic pack*
- DISPLAY_SQLSET
  - *Requires Real Application Testing Option <or Tuning Pack>, depends on version*
  - *Some API subprograms to manage STS are included in 12.1 EE*
- DISPLAY_SQL_PLAN_BASELINE
  - *License docs indicate that SPM is included in 12.1 EE*
  - *Found no mention in the 11.1 docs*

GLOC
Great Lakes Oracle Conference

Useful functions

- DISPLAY
- DISPLAY_CURSOR
- DISPLAY_AWR
- DISPLAY_PLAN
- DIFF_PLAN(_AWR, _CURSOR, _OUTLINE, _BASELINE)
- DISPLAY_SQLSET
- DISPLAY_SQL_PLAN_BASELINE
- DISPLAY_SQL_P[ROFILE,ATCH]_PLAN
- COMPARE_PLANS
- There are also undocumented internal functions that dbms_xplan passes around

GLOC
Great Lakes Oracle Conference

DISPLAY

- Displays the contents of **a** plan table
  - May be ‘**the**’ plan_table or a table with the same column names
- Returns a DBMS_XPLAN_TYPE_TABLE
- Autotrace explain uses this function (can be seen in 10046 output)

```
orcl_ora_6497.trc:SELECT PLAN_TABLE_OUTPUT FROM
     TABLE(CAST(DBMS_XPLAN.PREPARE_RECORDS(:B1 , :B2 ) AS SYS.DBMS_XPLAN_TYPE_TABLE))
orcl_ora_6497.trc:SELECT PLAN_TABLE_OUTPUT FROM
          TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE', :1))
```

- Use caution, this is what the optimizer *thinks* will happen
- May use filter_preds option to restrict which rows are displayed
  - *Filter_preds is subject to SQL injection*

DISPLAY_PLAN

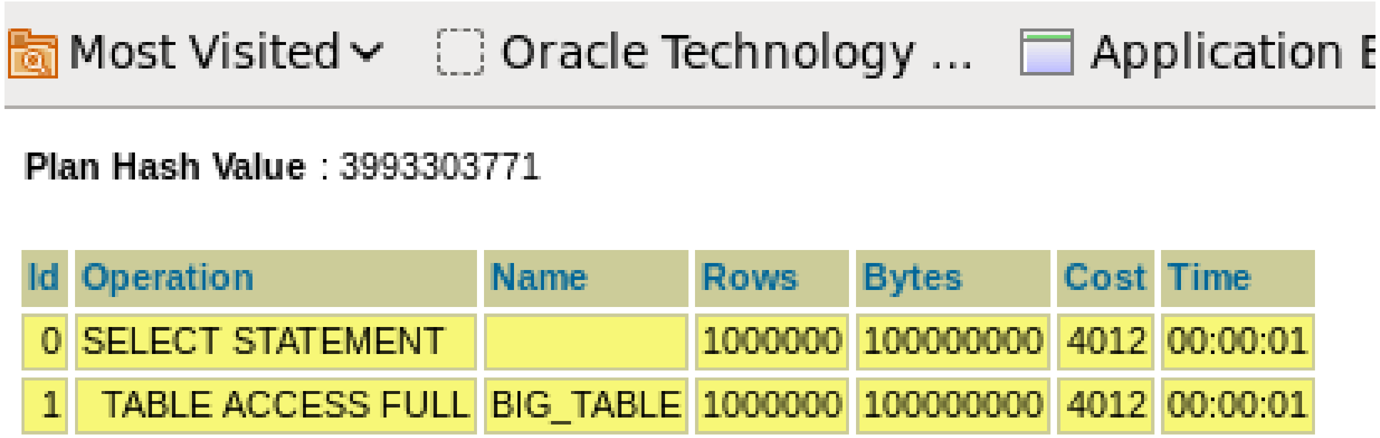
- Displays the contents of **a** plan_table with a CLOB return type
- Can produce multiple outputs (types)
  - TEXT
  - HTML
  - ACTIVE
  - XML
- ACTIVE is a rich EM-like interface
  - *Which I have been unable to get working properly*
  - *It requires outside connectivity as it downloads code to generate the ACTIVE display*
- XML doesn’t appear have a style with it, so my browser complains

DISPLAY_PLAN Syntax

```
DBMS_XPLAN.DISPLAY_PLAN (
   TABLE_NAME       IN    VARCHAR2   DEFAULT 'PLAN_TABLE',
   STATEMENT_ID     IN    VARCHAR2   DEFAULT NULL,
   FORMAT           IN    VARCHAR2   DEFAULT 'TYPICAL',
   FILTER_PREDS     IN    VARCHAR2   DEFAULT NULL,
   TYPE             IN    VARCHAR2   DEFAULT 'TEXT')
  RETURNS CLOB
```

DISPLAY_PLAN, cont'd

- Doesn't seem particularly useful, especially since it uses a plan_table



Plan Hash Value : 3993303771

Id	Operation	Name	Rows	Bytes	Cost	Time
0	SELECT STATEMENT		1000000	100000000	4012	00:00:01
1	TABLE ACCESS FULL	BIG_TABLE	1000000	100000000	4012	00:00:01

DISPLAY_CURSOR

- Displays an execution plan of a cursor from the cursor cache
- Shows what actually happened
- Plan statistics can be displayed with additional formats
- Must have select privileges to
  - V$sql_plan
  - V$sql
  - V$sql_plan_statistics_all

- Defaults to displaying the last executed statement
  - *If you don't set serveroutput off, you will get dbms_output data, not the last statement*

DISPLAY_CURSOR SYNTAX

```
DBMS_XPLAN.DISPLAY_CURSOR(
   SQL_ID            IN  VARCHAR2  DEFAULT  NULL,
   CURSOR_CHILD_NO   IN  NUMBER    DEFAULT  0,
   FORMAT            IN  VARCHAR2  DEFAULT  'TYPICAL');
RETURNS DBMS_XPLAN_TYPE_TABLE

(overloaded in recent releases to allow for sharding)
```

Great Lakes Oracle Conference

Displaying past plans

- How do you *find* plans from the past?
  - Dba_hist_sql_plan
  - Dbms_xplan.display_awr
- How do you display plans from the past?
  - Dbms_xplan.display_awr

GLOC
Great Lakes Oracle Conference

Finding past plans

- DBA_HIST_SQL_PLAN (also requires diagnostic pack license)

```
SQL> select distinct(plan_hash_value) from dba_hist_sql_plan
  2  where sql_id = '74cpnuu24wmx7';

PLAN_HASH_VALUE
---------------
     2135018976
       75148593
      908282281
      297866626
```

DISPLAY_AWR

- Display_awr
  - Default for plan_hash_value is null, so it will display **all** plans for a SQL_ID!
- Must have select privileges on
  - Dba_hist_sqltext
  - Dba_hist_sql_plan
  - V$database

Great Lakes Oracle Conference

DISPLAY_AWR Syntax

```
DBMS_XPLAN.DISPLAY_AWR(
   SQL_ID            IN     VARCHAR2,
   PLAN_HASH_VALUE   IN     NUMBER DEFAULT NULL,
   DB_ID             IN     NUMBER DEFAULT NULL,
   FORMAT            IN     VARCHAR2 DEFAULT TYPICAL,
   CON_ID            IN     NUMBER(38) DEFAULT NULL,
   AWR_LOCATION      IN     VARCHAR2 DEFAULT NULL);
RETURNS DBMS_XPLAN_TYPE_TABLE
```

GLOC
Great Lakes Oracle Conference

SPM, Tuning Sets, & Profiles

- STS – SQL statements, execution context/statistics, and plans
  - *Allows for grouping SQL and metadata into one object, for*
    - Input to advisors
    - Transport between databases
- SQL Profile – a database object containing supplemental stats for a particular statement
  - *Including cardinality adjustments*
  - *Doesn't tie the optimizer to a particular plan*
- SPM – 'baselines' a set of plans for a SQL handle that the optimizer is allowed to use
  - *Allows the optimizer to only use accepted plans*
  - *Prevents performance regression*

Finding plans for baselines

- DISPLAY_SQL_PLAN_BASELINE
  - *Requires select privilege on dba_sql_plan_baselines*
- Displays one or more plans for a SQL_HANDLE
- SQL_HANDLE and PLAN_NAME both default to null
  - *One might think that if both are null then all sql_handles and all plan_names would be displayed*
  - *Alas, that would be incorrect. One or the other must be supplied*
- If a PLAN_NAME is passed then that plan is displayed else all plans for a particular SQL_HANDLE are displayed

DISPLAY_SQL_PLAN_BASELINE Syntax

```
DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE (
   SQL_HANDLE      IN VARCHAR2 := NULL,
   PLAN_NAME       IN VARCHAR2 := NULL,
   FORMAT          IN VARCHAR2 := 'TYPICAL')
 RETURNS DBMS_XPLAN_TYPE_TABLE
```

GLOC
Great Lakes Oracle Conference

DISPLAY_SQLSET

- Displays plan(s) for a statement from a SQL Tuning Set
- If no plan_hash_value is provided, all plans for the statement are shown
- Need administer any sql tuning set to display SQL tuning sets you don’t own

GLOC
Great Lakes Oracle Conference

DISPLAY_SQLSET Syntax

```
DBMS_XPLAN.DISPLAY_SQLSET(
   SQLSET_NAME       IN  VARCHAR2,
   SQL_ID            IN  VARCHAR2,
   PLAN_HASH_VALUE   IN NUMBER := NULL,
   FORMAT            IN  VARCHAR2  := 'TYPICAL',
   SQLSET_OWNER      IN  VARCHAR2  := NULL)
  RETURN DBMS_XPLAN_TYPE_TABLE
```

DISPLAY_SQL_P[ROFILE,ATCH]_PLAN

- New in 12.1
- Not sure how terribly useful these are as yet
- Display
  - *SQL_TEXT*
    - For sys-generated profiles (SYS_SQLPROF*) it appears some hints are embedded
  - *Profile/Patch name*
  - *Status*
  - *Plan rows is listed, but not stored/displayed*

GLOC
Great Lakes Oracle Conference

DISPLAY_SQL_P[ROFILE,ATCH]_PLAN syntax

```
DBMS_XPLAN.DISPLAY_SQL_PROFILE_PLAN(
   NAME                 IN   VARCHAR2,
   FORMAT               IN   VARCHAR2   := 'TYPICAL')
  RETURN DBMS_XPLAN_TYPE_TABLE

DBMS_XPLAN.DISPLAY_SQL_PATCH_PLAN(
   NAME                 IN   VARCHAR2,
   FORMAT               IN   VARCHAR2   := 'TYPICAL')
  RETURN DBMS_XPLAN_TYPE_TABLE
```

DIFF_PLAN

- Has a whole bunch of different variants
- _CURSOR, _OUTLINE, _SQL_BASELINE, _AWR,
- All undocumented except DIFF_PLAN, and that is spotty/variable
- Sounds really cool
- I have been unable to get *any* of them to work, they don’t seem to be ready for prime time
- Morgan’s Library has an example of diff_plan_cursor, but I could not duplicate his success

For example...

- DIFF_PLAN_SQL_BASELINE
- Seems pretty nifty
- Requires the ADVISOR privilege – this gave me hope
- But only required it to error out properly:

```
select
dbms_xplan.diff_plan_sql_baseline('SQL_PLAN_0u5f7k97ywa5jcf314e9e','
SQL_PLAN_0u5f7k97ywa5jf67ee4d1') from dual
      *
ERROR at line 1:
ORA-14552: cannot perform a DDL, commit or rollback inside a query
or DML
ORA-06512: at "SYS.PRVT_ADVISOR", line 5026
ORA-14551: cannot perform a DML operation inside a query
```

COMPARE_PLANS

- Used to compare plans
- Takes a reference_plan (must be a single plan)
- and a plan_object_list (one or more plans)
- Plan sources for the list are described in the docs
- Docs indicate that plan sources for an object list can be

PLAN_TABLE_OBJECT	CURSOR_CACHE_OBJECT	AWR_OBJECT	SQLSET_OBJECT
SPM_OBJECT	SQL_PROFILE_OBJECT	ADVISOR_OBJECT	

- Parameters vary by object

Great Lakes Oracle Conference

COMPARE_PLANS, cont'd

- I have tried several of these
- And THEY WORK!!!!
- Some don't provide useful information
- But they don't barf!

GLOC
Great Lakes Oracle Conference

COMPARE_PLANS

```
DBMS_XPLAN.COMPARE_PLANS(
   reference_plan    IN generic_plan_object,
   compare_plan_list IN plan_object_list,
   type              IN VARCHAR2 := 'TEXT',
   level             IN VARCHAR2 := 'TYPICAL',
   section           IN VARCHAR2 := 'ALL')
 RETURN CLOB;
```

BASE Formats

- 4 simple formats
  - *Basic*
  - *Typical*
  - *Serial*
  - *All*

Basic format

- Displays *very* basic information
  - *Operation ID, Operation, Operation Options*
- Not terribly useful
  - *Unless you want an uncluttered view that shows *just* the plan steps*

```
SQL> select * from table(dbms_xplan.display_cursor(sql_id=>'akcdbbd6kxp09',format=>'BASIC'));

PLAN_TABLE_OUTPUT
----------------------------------------------------------------------------------------------
EXPLAINED SQL STATEMENT:
------------------------
SELECT REQUIREMENTDATAHISTORYID FROM REQUIREMENTDATAHISTORY WHERE
UPDATEDATE BETWEEN :B2 AND :B1

Plan hash value: 1638392505

-----------------------------------------------------
| Id  | Operation          | Name                   |
-----------------------------------------------------
|   0 | SELECT STATEMENT   |                        |
|   1 |  FILTER            |                        |
|   2 |   TABLE ACCESS FULL| REQUIREMENTDATAHISTORY |
-----------------------------------------------------
```

GLOC
Great Lakes Oracle Conference

Typical format

- The default
  - *if you don't specify, this is what you get*
- Columns included in BASIC format, plus
  - *Rows: number of rows the optimizer expects to return*
  - *Bytes: expected rows * avg_row_len*
  - *Cost*
  - *Time*
- If relevant, shows
  - *Partition pruning*
  - *Parallel operations (PCWP, Parallel to Serial)*
  - *Predicate information*
  - *Unused and/or errored hints*

Predicate information

- Why is it important?
- It shows which lines are access predicates and which are filter predicates
  - *Access – getting only data we need*
  - *Filter – we get more data than we need and throw some away*
- It shows datatype mismatches
  - *Where date_col = ‘string’*
  - *This is important!*
- Shows query transformations
  - *If a = b and c = b, then a = c*

GLOC
Great Lakes Oracle Conference

Serial format

- Typical minus parallel information
  - *Parallel is still apparent, but nothing about what it was doing*
    - No parallel operation information

ALL format

- TYPICAL + additional information
  - *Projection*
  - *Query block and object aliases*
  - *Remote*
  - *Hint_report*
- All does *not* mean everything

Projection

- Given inputs, here is the list of columns returned (and their datatypes)
- May be handy when someone has done a select *
- May be interesting to see the pile of columns you are carrying through a plan
- I’ve looked at thousands of plans and I haven’t used it once that I recall

Query Block and aliases

- Shows query block and object alias information (surprise!)
- Very useful in complicated queries to show where each plan step is occurring
- Shows up like `sel$`, etc if Oracle picks them
- User-defined via the `/*+ qb_name() */` hint

```
select /*+ qb_name(main_query) */ employee_id
from employees;
```

GLOC
Great Lakes Oracle Conference

Sauce for the goose, Mr. SaAvik

- Can add or subtract columns or entire sections
- May be used with ‘+’ or ‘-’ or nothing
- + means add specified format in addition to the specified (or default) base format
  - *Options separated by spaces are also legal, so ‘+’ is not required*
- - means the all of the base format *except* the specified format options
  - *The ‘-’ is NOT optional if you want to exclude formats*
- This can enhance the readability of plans by excluding the irrelevant bits

```
select * from table(dbms_xplan.display_cursor('3k4sh40hw6rum',0,'TYPICAL +ALIAS -BYTES’));
```

GLOC
Great Lakes Oracle Conference

Demo of base formats

- Base formats
- Additional adds and exclusions between base formats
- Like adding ‘alias’ to typical (producing qb/alias section), while removing bytes

Various add-on format options

- NOTE
- ADAPTIVE
- IOSTATS
- MEMSTATS
- ALLSTATS
- LAST
- PEEKED_BINDS
- OUTLINE
- ADVANCED
- HINT_REPORT

Great Lakes Oracle Conference

How do I know what I have?

- NOTE
- Displays the notes section
  - *It appears that in 11.2 and higher that this shows up when relevant in TYPICAL format even without requesting it*
- Shows
  - *whether a plan went adaptive, but not what was changed*
  - *Cardinality feedback (statistics feedback in 12.x) was used*
  - *Dynamic sampling was used*
  - *Plan Control was used*
    - SQL Plan Directives were used
    - SQL Plan Baselines were used
    - SQL Profiles were used

Adaptive plans

- The notes section tells you when something went adaptive, but nothing else
- This is where the ADAPTIVE format comes in!
- Displays the final plan
- ‘-’ denotes plan steps that were skipped
- Only available in 12.1 and above

GLOC
Great Lakes Oracle Conference

IOSTATS

- If plan statistics are gathered, this format option shows the I/O statistics for the plan
- Must have one of the following set
  - *`/*+ GATHER_PLAN_STATISTICS */` hint set in the query*
  - *Alter [session|system] set statistics_level = ALL;*
  - *STATISTICS_LEVEL parameter set to ALL (default is typical)*
- *NOTE* this causes additional overhead – please do not set to ALL for the entire database and leave it that way.
- If those are not set, the NOTE section will complain (shown in demo)
- IO means logical and physical I/O
- Have to adjust linesize to display properly (I use 132, but this changes with indents)
  - *If >=18c, use set linesize window*

IOSTATS, Cont'd

- IOSTATS gives us additional data
- Starts
- E-Rows and A-Rows
  - *This gives an optimizer sanity check – are e-rows and a-rows close?*
- A-Time
- Buffers
  - *This allows you to do a sanity check on efficiency (i/os per row/table?)*
    - LIOs 10x the number of rows is really good
    - 100x should stand out as probably can be improved
- Reads
- Writes

MEMSTATS

- Displays memory statistics used for an operation
- Requires auto PGA management
- Useful for memory-intensive operations (hash joins, sorts, etc)
- Shows whether things dumped to disk
- Adds 0Mem, 1Mem, Used-Mem, Used-Tmp columns
  - *These values come from v$sql_workarea (thanks, Rob van Wijk!)*

ALLSTATS

- Shortcut for IOSTATS + MEMSTATS
- Linesize 132 is too small, I use 180
  - *If >=18c, use set linesize window*
- This is what I typically use, so I can see everything

LAST

- Use with IOSTATS, MEMSTATS, ALLSTATS
- By default these show statistics for all executions (because we are greedy by default)
- LAST shows only statistics from the last execution

Deprecated formats

- Deprecated but still work for backward compatibility
- RUNSTATS_TOT
  - Same as IOSTATS
    - Displays IOSTATS across all executions
- RUNSTATS_LAST
  - Same as IOSTATS LAST
    - Displays IOSTATS for last execution

PEEKED_BINDS

- Displays the peeked binds the query used
- Appears to come from dba_hist_sqlbind (except for STS)
- Usable for DISPLAY_CURSOR(), DISPLAY_SQLSET(), and DISPLAY_AWR()

```
SELECT REQUIREMENTDATAHISTORYID FROM REQUIREMENTDATAHISTORY WHERE
UPDATEDATE BETWEEN :B2 AND :B1

Plan hash value: 1638392505

----------------------------------------------------------------------------------------------
| Id  | Operation          | Name                      | Rows  | Bytes | Cost (%CPU)| Time     |
----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                           |       |       |   160K(100)|          |
|*  1 |  FILTER            |                           |       |       |            |          |
|*  2 |   TABLE ACCESS FULL| REQUIREMENTDATAHISTORY    |     3 |    42 |   160K  (2)| 00:00:07 |
----------------------------------------------------------------------------------------------

Peeked Binds (identified by position):
--------------------------------------

   1 - :B2 (DATE): 01/02/2018 03:30:17
   2 - :B1 (DATE): 01/02/2018 03:53:46
```

GLOC Great Lakes Oracle Conference

OUTLINE

- Displays the full set of hints required to reproduce the plan
- This is particularly handy when trying to reproduce a past plan
  - *You have a query that performs poorly but used to perform well*
  - *Get the outline hints from display_awr()*
  - *Use those hints to get the good plan in cache and baseline it*
- Outline hints also shows join order (Thanks, Maria Colgan!)

ADVANCED

- Appears to give ALL + OUTLINE

HINT_REPORT[_USED|UNUSED]

- Hints can be hard to use, and may silently fail
- 19c has nifty stuff for seeing hints via DBMS_XPLAN
- _USED shows used hints
- _UNUSED shows hints that weren't used or had syntax errors
  - *Included in the TYPICAL base format*
- Status of hints
  - *No status – hint was used*
  - *U – ununsed*
  - *N – unresolved*
  - *E – syntax error*

GLOC
Great Lakes Oracle Conference

HINT_REPORT, cont'd

- Mutually exclusive hints, both are 'unused'
- Helpful comments are a 'syntax error'
- Query blocks that don't exist are 'not resolved'
  - don't have a status, but are associated with plan line # 0
- Multiple hints for the same thing in different places are 'unused' (one or more are overridden)
- Sometimes hints are used that cause others to be excluded, those are 'unused'

GLOC
Great Lakes Oracle Conference

Additional formats demo

Great Lakes Oracle Conference

Questions?

GLOC

Great Lakes Oracle Conference

Thank you!

- Thank you for attending this presentation!
- My contact information:
  - deedsd@nationwide.com
  - @dallasdeeds

GLOC
Great Lakes Oracle Conference

Acknowledgements & References

- Oracle Documentation
  - https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_XPLAN.html#GUID-ED750F75-CC30-4C4F-B5A5-94D7599AEB97
- Charles Hooper
  - https://hoopercharles.wordpress.com/2010/03/01/dbms_xplan-format-parameters/
- Dan Morgan
  - https://www.morganslibrary.org/reference/pkgs/dbms_xplan.html
- Maria Colgan
  - https://sqlmaria.com/2017/08/08/using-dbms_xplan-display_cursor-to-examine-execution-plans/
- Rob van Wijk
  - http://rwijk.blogspot.com/2008/03/dbmsxplandisplaycursor.html
- The Optimizer in Oracle Database 19c (Oracle whitepaper, Author Nigel Bayliss)
  - https://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-with-oracledb-19c-5324206.pdf
- Oracle docs for showing how hint_report works
  - https://docs.oracle.com/en/database/oracle/oracle-database/19/tgsql/influencing-the-optimizer.html#GUID-1697E7CA-9DD0-4C0D-9BC9-E4E17334C0AA

GLOC
Great Lakes Oracle Conference