

# How To Analyze SQL Run Times By Plan Using ASH Data

Craig Shallahamer  
craig@orapub.com



[www.orapub.com/ppts](http://www.orapub.com/ppts)



This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

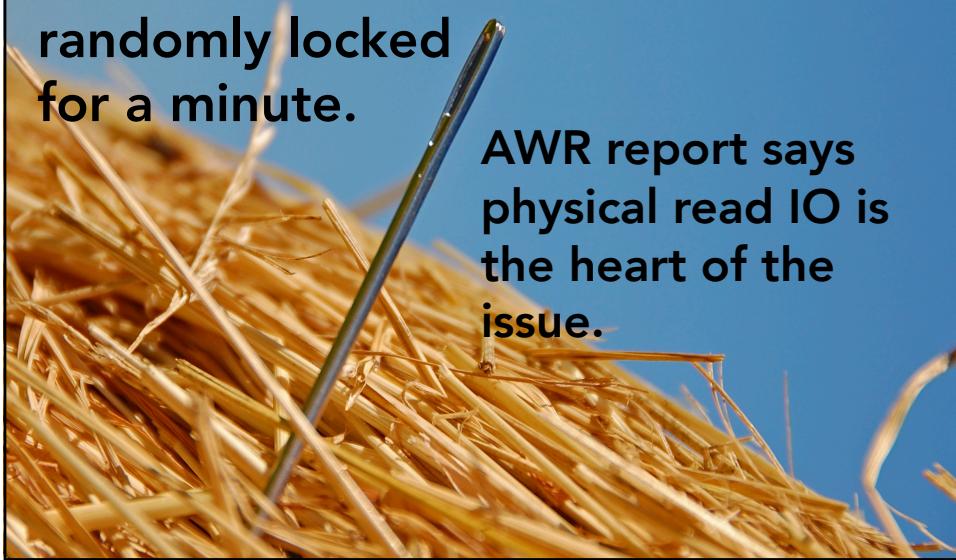


For the most updated version  
of this presentation, go to:

[www.orapub.com/ppts](http://www.orapub.com/ppts)

**A few users said  
their screen  
randomly locked  
for a minute.**

**AWR report says  
physical read IO is  
the heart of the  
issue.**



This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

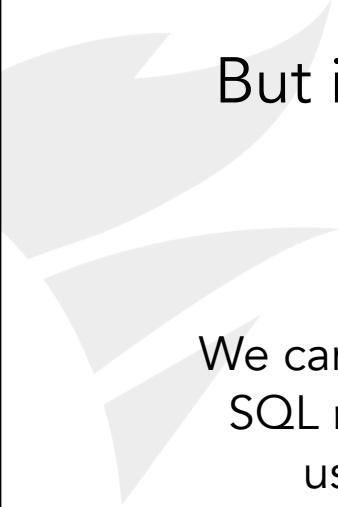
## Performance "incident" questions we may need to answer.

- When did the incident begin?
- When did the incident end?
- How many sessions where involved?
- How many people or processes were affected?
- Is there a genesis session?
- Is there a key SQL, module, program, action,...?
- Has it happened before? If so, how often?
- Will it happen again? If so, when? how often?



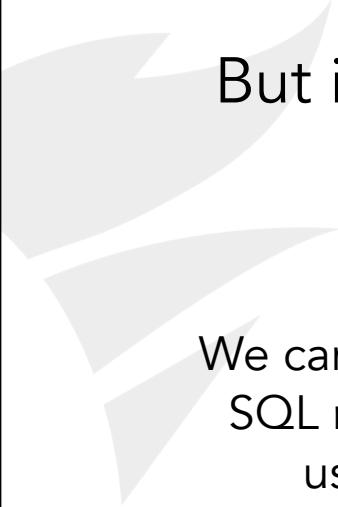
# But is there more?

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



# But is there more?

# Yes!



We can do some amazing  
SQL runtime analysis...  
using ASH data!

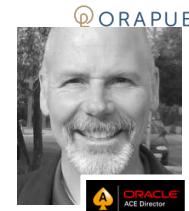
# Let's get down to the SQL!

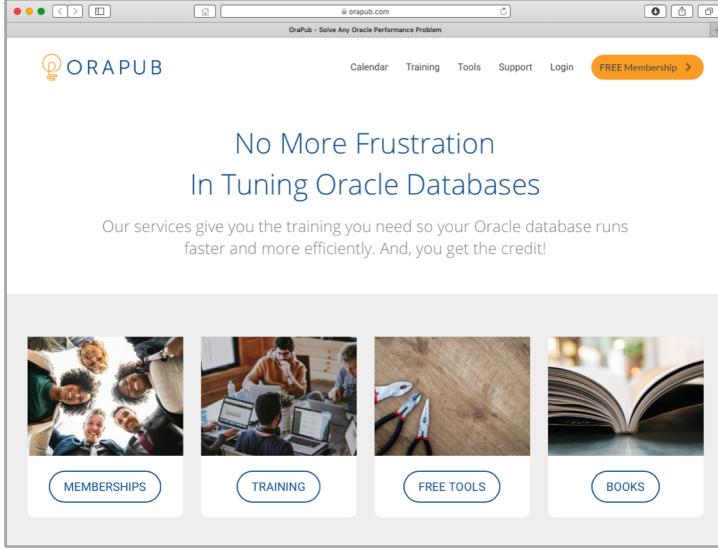
- What is the key incident SQL?
- How long did it run during the incident?
- Is this run time unusual?
- What is the typical run time, high, low, etc?
- How often is the SQL run?
- Are there multiple plans?
- Is this a slow plan?
- How often is the plan used?
- Is parallel query involved?

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## About Me...

- Long time Oracle DBA
- Specialize in Oracle Database performance and performance engineering
- Performance researcher
- Blogger: A Wider View About Oracle Performance Tuning
- Author: Oracle Performance Firefighting and Forecasting Oracle Performance.
- Conference speaker
- Teacher and mentor
- Oracle ACE Director
- IOUG DBA Track Manager





The screenshot shows the OraPub website homepage. The header features the OraPub logo and navigation links for Calendar, Training, Tools, Support, and Login, along with a 'FREE Membership' button. The main heading 'No More Frustration In Tuning Oracle Databases' is displayed, followed by a subtext: 'Our services give you the training you need so your Oracle database runs faster and more efficiently. And, you get the credit!'. Below this are four image cards: 'MEMBERSHIPS' (people), 'TRAINING' (people at a desk), 'FREE TOOLS' (tools), and 'BOOKS' (an open book). To the right of the main content, a vertical text box states: 'OraPub works with Oracle DBAs empowering them to beat bots, AI, machine learning and autonomous anything.'

9 (c)OraPub, Inc. ASH SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA. There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



## Our journey for today.

- The value of analyzing SQL run times
- The problem of getting SQL run times
- Manually inferring SQL run times
- Automatically inferring SQL run times
- SQL run time data analysis

10 (c)OraPub, Inc. ASH SQL Run Times

## Image the questions I can ask!



Each piece of paper is like an active Oracle session.  
Each stack contains all the active sessions captures during  
a single ASH sample, such as at 15:25:06.

11

(c)OraPub, Inc.

ASH SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the  
March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



## The Value Of Analyzing SQL Run Times

12

(c)OraPub, Inc.

ASH SQL Run Times

## A challenging incident...

- User: It took 90 seconds to query a customer order!
- DBA: Can you show me?
- User: No, it was like a random thing.
- DBA: Has this happened before?
- User: No, it was like a random thing.
- DBA: How long does the query usually take?
- User: Five seconds.
- DBA: When did the 90 second query occur?
- User: Around 10am
- DBA: Can you be more specific?
- User: No.

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## This is what I want to know!

- What is the key SQL?
- Verify how long the SQL ran during the incident
- Is this run time unusual?
- What is the typical run time, high, low, etc?
- Are there multiple plans?
- Is this a slow plan?
- How often is the SQL run?
- How often is this plan used?

# The Problem Of Getting SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## No Go: SQL run times using AWR report.

- The AWR keeps track of total SQL statement “elapsed time”, which is CPU plus non-idle wait time for all executions.
- So, we can easily get the average elapsed time over an AWR snapshot interval.
- But execution level details are not stored in AWR tables.
- Elapsed time may not be the run time. Why?
- But average elapsed time is only the average, lacking the diversity of data we need during an incident analysis.
- AWR report does not provide plan details.
- There is some details by mining the AWR tables.
- SQL Monitor can also provide useful details.
- But nothing provides the details and the ease of reporting, the depth of analysis and flexibility that ASH provides.

# Manually Inferring SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## The foundational columns.

```
SQL> desc dba_hist_active_sess_history
  Name          Null?    Type
  -----
SNAP_ID          NOT NULL NUMBER
DBID            NOT NULL NUMBER
INSTANCE_NUMBER  NOT NULL NUMBER
SAMPLE_ID        NOT NULL NUMBER
SAMPLE_TIME      NOT NULL TIMESTAMP(3)
SESSION_ID       NOT NULL NUMBER
SESSION_SERIAL#  NUMBER
SQL_ID           VARCHAR2(13)
SQL_CHILD_NUMBER NUMBER
SQL_PLAN_HASH_VALUE NUMBER
SQL_FULL_PLAN_HASH_VALUE NUMBER
SQL_EXEC_ID      NUMBER
SQL_EXEC_START    DATE
```

# The trick is the SQL start time.



```
SQL> desc dba_hist_active_sess_history
Name          Null?    Type
-----          -----
SNAP_ID        NOT NULL NUMBER
DBID          NOT NULL NUMBER
INSTANCE_NUMBER NOT NULL NUMBER
SAMPLE_ID      NOT NULL NUMBER
SAMPLE_TIME    NOT NULL TIMESTAMP(3)
SESSION_ID     NOT NULL NUMBER
SESSION_SERIAL# NUMBER
SQL_ID         VARCHAR2(13)
SQL_CHILD_NUMBER NUMBER
SQL_PLAN_HASH_VALUE NUMBER
SQL_FULL_PLAN_HASH_VALUE NUMBER
SQL_EXEC_ID    NUMBER
SQL_EXEC_START DATE
```

19

(c)OraPub, Inc.

ASH SQL Run Times

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

# The trick is the SQL start time.

```
SQL> desc dba_hist_active_sess_history
Name          Null?    Type
-----          -----
SNAP_ID        NOT NULL NUMBER
DBID          NOT NULL NUMBER
INSTANCE_NUMBER NOT NULL NUMBER
SAMPLE_ID      NOT NULL NUMBER
SAMPLE_TIME    NOT NULL TIMESTAMP(3)
SESSION_ID     NOT NULL NUMBER
SESSION_SERIAL# NUMBER
SQL_ID         VARCHAR2(13)
SQL_CHILD_NUMBER NUMBER
SQL_PLAN_HASH_VALUE NUMBER
SQL_FULL_PLAN_HASH_VALUE NUMBER
SQL_EXEC_ID    NUMBER
SQL_EXEC_START DATE
```

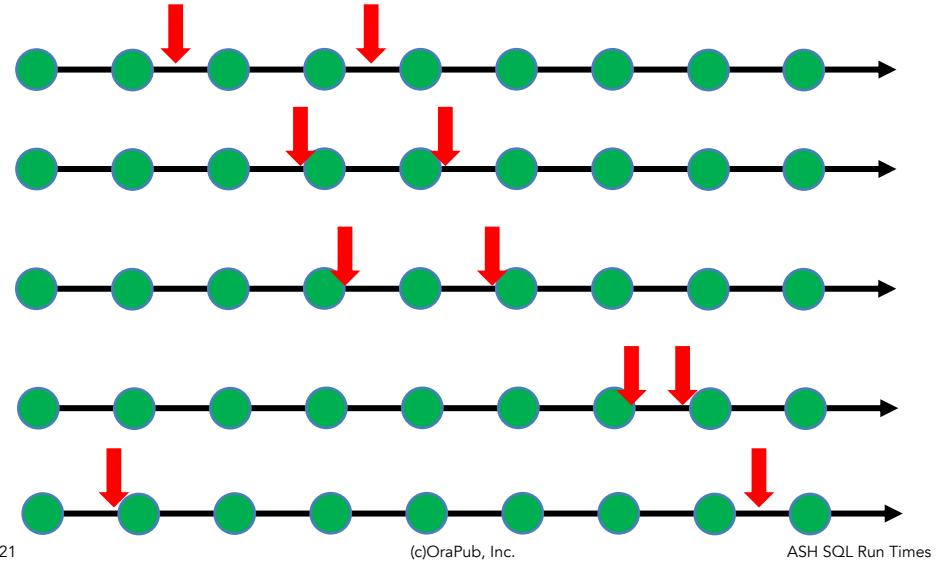
20

(c)OraPub, Inc.

ASH SQL Run Times

# Inferring Run Time By Counting Samples

ORAPUB



This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## To remember

- This is not about precision.
- How precise do you need to be?
- We are using samples, not instrumentation.
- Run times less the sample frequency may not be sampled.
- This method will **not** work for short duration SQL, that is, around the sample frequency.
- More samples per execution, the better the inference.
- The more executions, the better the inference... if we do the data analysis.

# A S H

## Scratch Pad

[www.orapub.com/tools](http://www.orapub.com/tools)

[www.orapub.com >](http://www.orapub.com)  
[Login >](#)  
[TOOLS & PRESENTATIONS >](#)  
[Tools](#)

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Inferring Run Time By Timelining A Session

```
def ashCols="sql_id,to_char(SQL_EXEC_START,'MI:SS') BT,
                      sql_exec_id,event"
def ashWhere='session_id=1377 and session_serial#=21665'

select sample_id, to_char(sample_time,'DD HH24:MI:SS') ST,
       session_id sid, session_state,
       &ashCols
  from &datasource
 where &timingdetails
       &dbahistdetails
       and &ashWhere
 order by 1,3
/
```

## Timeline An Execution #1

SAMPLE_ID	SAMPLE_TIME	SID	SES	STATE	SQL_ID
33806483	22 12:48:47	1377	ON CPU	078nfg1a3k2dt	
33806503	22 12:49:07	1377	ON CPU	5t51699jnz6vv	
33806513	22 12:49:17	1377	WAITING	asxdjf77byb88	
33806543	22 12:49:48	1377	ON CPU	b3898h5vbhn4j	
33806563	22 12:50:08	1377	ON CPU	9m9yhnsb2xan6	
33806573	22 12:50:18	1377	WAITING	8888smywmmr80	
33806583	22 12:50:28	1377	WAITING	8888smywmmr80	
33806603	22 12:50:48	1377	WAITING	8888smywmmr80	
33806613	22 12:50:58	1377	WAITING	0g1duccz2vmqg	
33806623	22 12:51:08	1377	WAITING	8888smywmmr80	
33806633	22 12:51:18	1377	WAITING	0g1duccz2vmqg	
33806883	22 12:55:28	1377	ON CPU	gxkuv8wrkvskg	
33806893	22 12:55:39	1377	WAITING	gxkuv8wrkvskg	
33806983	22 12:57:09	1377	WAITING	70s92z3tzbfg9	

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Timeline An Execution #1

SAMPLE_TIME	SID	SES	STATE	SQL_ID	START	SQL_EXEC_ID	EVENT
22 12:48:47	1377	ON CPU	078nfg1a3k2dt		48:46	50331648	
22 12:49:07	1377	ON CPU	5t51699jnz6vv		49:06	50331680	
22 12:49:17	1377	WAITING	asxdjf77byb88		49:16	50331648	db
22 12:49:48	1377	ON CPU	b3898h5vbhn4j		49:47	50331648	
22 12:50:08	1377	ON CPU	9m9yhnsb2xan6		50:06	50331661	
22 12:50:18	1377	WAITING	8888smywmmr80		50:13	50331716	db
22 12:50:28	1377	WAITING	8888smywmmr80		50:23	50331719	db
22 12:50:48	1377	WAITING	8888smywmmr80		50:43	50331723	db
22 12:50:58	1377	WAITING	0g1duccz2vmqg		50:50	50331730	db
22 12:51:08	1377	WAITING	8888smywmmr80		51:06	50331732	db
22 12:51:18	1377	WAITING	0g1duccz2vmqg		51:12	50331736	db
22 12:55:28	1377	ON CPU	gxkuv8wrkvskg		55:23	50331649	
22 12:55:39	1377	WAITING	gxkuv8wrkvskg		55:23	50331649	gc
22 12:57:09	1377	WAITING	70s92z3tzbfg9				lib

## Timeline An Execution #2

SAMPLE_ID	SAMPLE_TIME	SID	SES	STATE	SQL_ID
33808243	22 13:18:12	1377		WAITING	gxkuv8wrkvskg
33808253	22 13:18:23	1377		ON CPU	44x67s6n5j98s
33808263	22 13:18:33	1377		ON CPU	44x67s6n5j98s
33808273	22 13:18:43	1377		ON CPU	44x67s6n5j98s
33808283	22 13:18:53	1377		ON CPU	44x67s6n5j98s
33808293	22 13:19:03	1377		ON CPU	44x67s6n5j98s
33808303	22 13:19:13	1377		ON CPU	44x67s6n5j98s
33808313	22 13:19:23	1377		ON CPU	44x67s6n5j98s
33808323	22 13:19:33	1377		ON CPU	44x67s6n5j98s
33808333	22 13:19:43	1377		ON CPU	44x67s6n5j98s
33808343	22 13:19:53	1377		ON CPU	44x67s6n5j98s
33808353	22 13:20:03	1377		ON CPU	44x67s6n5j98s
33808423	22 13:21:13	1377		ON CPU	b4pwuvanmx0rh

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Timeline An Execution #2

SAMPLE_TIME	SID	SES	STATE	SQL_ID	START	SQL_EXEC_ID	EVE
22 13:18:12	1377		WAITING	gxkuv8wrkvskg	18:11	50331661	db
22 13:18:23	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:18:33	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:18:43	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:18:53	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:03	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:13	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:23	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:33	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:43	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:19:53	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:20:03	1377		ON CPU	44x67s6n5j98s	18:16	50331648	
22 13:21:13	1377		ON CPU	b4pwuvanmx0rh	21:12	50331650	

Is the incident run time the usual run time?

# Automatically Inferring SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

Let's use this knowledge  
to find the long run SQL.

How many samples for each SQL start time?

```
sql_id,  start_time,  count(ash_samples)

abc      2:00pm      10
abc      3:05pm      13
abc      3:45pm       9
abc      5:03pm      11
8uc      1:05pm      98
9t2 ...
```

Make sure to exclude PQ sessions!

## Q: What are the long running SQLs?

```

set tab off
set pagesize 60
set linesize 300
col the_count format 999999
col x format a30 trunc

select sql_id,
       to_char(SQL_EXEC_START, 'YYYY-Mon-DD HH24:MI:SS'), sql_exec_id,
       count(*)
  from BH2016JUNESPIKE3
 where sample_time between to_date('22-Jun-2010 12:00', 'DD-Mon-YYYY HH24:MI:SS')
                        and to_date('22-Jun-2020 17:00', 'DD-Mon-YYYY HH24:MI:SS')
   and dbid=3627646567 and instance_number=3
   and (program is null or program not like '%(P%)%')
 group by sql_id,
          to_char(SQL_EXEC_START, 'YYYY-Mon-DD HH24:MI:SS'), sql_exec_id
 order by count(*)

```

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Find the long running SQL.

SQL_ID	TO_CHAR(SQL_EXEC_STA	SQL_EXEC_ID	COUNT(*)
...			
2ym1fcubhalzh	2016-Jun-22 14:15:32	50332072	75
28xcchp701005	2016-Jun-22 11:57:47	50331649	85
504y5ryrj6vwv	2016-Jun-22 13:34:57	50331650	104
15fm71wtjpg88	2016-Jun-22 11:58:51	50331649	122
15fm71wtjpg88	2016-Jun-22 12:20:59	50331650	128
c7t2kg0dft03x	2016-Jun-22 14:30:41	50331648	139
dkwt3r5d0w90k	2016-Jun-22 13:42:00	50332172	150
dkwt3r5d0w90k	2016-Jun-22 13:40:40	50332170	158
345kqzqcmkv4z	2016-Jun-22 15:32:17	50331648	165
4c38ztzqsyb6d	2016-Jun-22 14:07:33	50331648	177
7ky1g5ucjq856	2016-Jun-22 12:35:57	50331648	187
gaqcyckx7mqws	2016-Jun-22 13:00:19	50331649	906
gaqcyckx7mqws	2016-Jun-22 12:49:15	50331648	982
			2028
6750 rows selected.			

## Q: What are the long running SQLs?

```
-- ScratchPad -- Top Anything Script
-- 

def ashCols="sql_id, to_char(SQL_EXEC_START,'YYYY-Mon-DD HH24:MI:SS'),
           sql_exec_id,'::'||program||'::'
def ashstate='%'
def ashCount="*"
def ashWhere="(program is null or program not like '%(P%)%' )"

select * from (
  select count(&ashCount) the_count, &ashCols
  from &datasource
  where &timingdetails
        &dbahistdetails
  and session_state like '&ashstate'
  and &ashWhere
  group by &ashCols
  order by 1 desc, 2 desc
)
where rownum < 100
/

```

34

(c)OraPub, Inc.

ORAPUB

ASH SQL Run Times

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Find the long running SQL.

THE_COUNT	SQL_ID	TO_CHAR(SQL_EXEC_STA	SQL_EXEC_ID	'::'  PROGRAM  '::'
982	gaqcyckx7mqws	2016-Jun-22 12:49:15	50331648	::::
906	gaqcyckx7mqws	2016-Jun-22 13:00:19	50331649	::::
632				::::
276				::oracle@03.acme.com (LGWR)
189				::oracle@03.acme.com (NSA3)
187	7ky1g5ucjq856	2016-Jun-22 12:35:57	50331648	::::
181	0z3vhzqabptfk	2016-Jun-22 13:37:20	50331648	::::
180	48ujm55s9b7uf	2016-Jun-22 12:05:53	50331648	::::
180	11p4wt11w1tc4	2016-Jun-22 13:07:19	50331648	::::
177	4c38ztzqsyb6d	2016-Jun-22 14:07:33	50331648	::::
165	345kqzcmkv4z	2016-Jun-22 15:32:17	50331648	::JDBC Thin Client::
158	dkwt3r5d0w90k	2016-Jun-22 13:40:40	50332170	::::
154				::JDBC Thin Client::
150	dkwt3r5d0w90k	2016-Jun-22 13:42:00	50332172	::::
139	c7t2kg0dft03x	2016-Jun-22 14:30:41	50331648	::WinAutomation.ServiceAgen
132	68bshh7jtj1lz			::sqlplus@03.acme.com (TNS)
128	15fm7lwtj jpg88	2016-Jun-22 12:20:59	50331650	::::
...				

35

(c)OraPub, Inc.

ASH SQL Run Times

# SQL Run Time Data Analysis

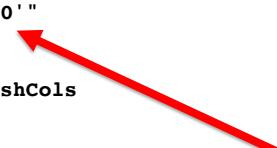
( Analyze A Specific SQL's Run Times )

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Q: What are "the" SQL's runtimes?

```
-- OraPub ScratchPad - Top Anything Script
-- 

def ashCols="sql_id, to_char(SQL_EXEC_START,'YYYY-Mon-DD HH24:MI:SS'),
           sql_exec_id,'::'||program||'::'
def ashstate='%'
def ashCount="*"
def ashWhere="(program is null or program not like '%(P%)%')
           and sql_id= '8888smywmmr80'"
```



```
select * from (
  select count(&ashCount) the_count, &ashCols
  from  &datasource
  where  &timingdetails
         &dbahistdetails
  and  session_state like '&ashstate'
  and  &ashWhere
  group by &ashCols
  order by 1 desc, 2 desc
)
where rownum < 100
/
```

## Run times for 8888smywmmr80.

```
def ashWhere="(program is null or program not like '%(P%)%')  
           and sql_id='8888smywmmr80'"
```

THE_COUNT	SQL_ID	TO_CHAR(SQL_EXEC_STA	SQL_EXEC_ID	' :::
5	8888smywmmr80	2016-Jun-22 14:07:49	50332059	:::w3w
2	8888smywmmr80	2016-Jun-22 12:45:23	50331659	:::w3w
2	8888smywmmr80	2016-Jun-22 14:51:48	50332107	:::w3w
2	8888smywmmr80	2016-Jun-22 14:45:02	50332079	:::w3w
2	8888smywmmr80	2016-Jun-22 15:06:17	50332114	:::w3w
2	8888smywmmr80	2016-Jun-22 13:51:48	50332030	:::w3w
2	8888smywmmr80	2016-Jun-22 14:49:17	50332087	:::w3w
1	8888smywmmr80	2016-Jun-22 13:59:18	50332046	:::w3w
1	8888smywmmr80	2016-Jun-22 12:38:52	50331648	:::w3w

...  
63 rows selected.

- What is the maximum run time?
- What is the average run time?
- Are there anomalous run times?

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Run times for gxkuv8wrkvskg.

```
def ashWhere="(program is null or program not like '%(P%)%')  
           and sql_id='gxkuv8wrkvskg'"
```

THE_COUNT	SQL_ID	TO_CHAR(SQL_EXEC_STA	SQL_EXEC_ID	' :::
7	gxkuv8wrkvskg	2016-Jun-22 13:48:11	50331665	:::w3w
4	gxkuv8wrkvskg			:::w3w
2	gxkuv8wrkvskg	2016-Jun-22 12:55:23	50331649	:::w3w
1	gxkuv8wrkvskg	2016-Jun-22 13:10:46	50331656	:::w3w
1	gxkuv8wrkvskg	2016-Jun-22 13:03:56	50331651	:::w3w
1	gxkuv8wrkvskg	2016-Jun-22 13:18:11	50331661	:::w3w
1	gxkuv8wrkvskg	2016-Jun-22 15:12:11	50331649	:::w3w

7 rows selected.

- What is the maximum run time?
- What is the average run time?
- Are there anomalous run times?

## Run times for 4b3rdu4yh4ztc.

```
def ashWhere="(program is null or program not like '%(P%)%')
and sql_id='4b3rdu4yh4ztc'"

THE_COUNT SQL_ID          TO_CHAR(SQL_EXEC_STA SQL_EXEC_ID ' :::' || PROGRAM
-----
 61 4b3rdu4yh4ztc 2016-Jun-22 14:29:36 50331648 ::::
 26 4b3rdu4yh4ztc 2016-Jun-22 15:10:21 50331648 ::::
 26 4b3rdu4yh4ztc 2016-Jun-22 12:16:38 50331648 ::::
 16 4b3rdu4yh4ztc 2016-Jun-22 13:23:38 50331648 ::::
 12 4b3rdu4yh4ztc 2016-Jun-22 13:56:54 50331648 ::::
  9 4b3rdu4yh4ztc 2016-Jun-22 12:21:30 50331649 ::::
  9 4b3rdu4yh4ztc 2016-Jun-22 15:45:03 50331648 ::::
  8 4b3rdu4yh4ztc 2016-Jun-22 14:40:07 50331649 ::::
  8 4b3rdu4yh4ztc 2016-Jun-22 12:51:19 50331648 ::::
  7 4b3rdu4yh4ztc          ::::
  5 4b3rdu4yh4ztc 2016-Jun-22 13:26:35 50331649 ::::
  4 4b3rdu4yh4ztc 2016-Jun-22 12:53:10 50331649 ::::
  4 4b3rdu4yh4ztc 2016-Jun-22 15:46:53 50331649 ::::
  4 4b3rdu4yh4ztc 2016-Jun-22 13:59:13 50331649 ::::
  4 4b3rdu4yh4ztc 2016-Jun-22 15:15:15 50331649 ::::
```

15 rows selected.

40

(c)OraPub, Inc.

ASH SQL Run Times

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

Q: Perhaps...  
the user experienced...  
a bad plan?

41

(c)OraPub, Inc.

ASH SQL Run Times

Q: What are the SQL's runtimes by execution plan?

```
-- OraPub ScratchPad - Top Anything Script
--
def ashCols="sql_id, sql_plan_hash_value,
           to_char(SQL_EXEC_START, 'YYYY-Mon-DD HH24:MI:SS'),
           sql_exec_id, '::'||program||'::'
def ashstate='%'
def ashCount="*"
def ashWhere="(program is null or program not like '%(P%)%')
             and sql_id= '8888smywmmr80'"

select * from (
  select count(&ashCount) the_count, &ashCols
  from  &datasource
  where &timingdetails
        &dbahistdetails
        and session_state like '&ashstate'
        and &ashWhere
  group by &ashCols
  order by 1 desc, 2 desc
)
where rownum < 100
/
42
```

(c)OraPub, Inc.

ORAPUB

ASH SQL Run Times

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Run times by plan for 8888smywmmr80.

ORAPUB

```
def ashWhere="(program is null or program not like '%(P%)%')
             and sql_id= '8888smywmmr80'"
```

THE_COUNT	SQL_ID	SQL_PLAN_HASH_VALUE	TO_CHAR(SQL_EXEC_ST
5	8888smywmmr80	1336928456	2016-Jun-22 14:07:4
2	8888smywmmr80	1336928456	2016-Jun-22 12:45:2
2	8888smywmmr80	1336928456	2016-Jun-22 14:49:1
2	8888smywmmr80	1336928456	2016-Jun-22 13:51:4
2	8888smywmmr80	1336928456	2016-Jun-22 15:06:1
2	8888smywmmr80	1336928456	2016-Jun-22 14:51:4
2	8888smywmmr80	1336928456	2016-Jun-22 14:45:0
1	8888smywmmr80	1336928456	2016-Jun-22 13:58:2
1	8888smywmmr80	1336928456	2016-Jun-22 12:52:0
...			

63 rows selected.

43

(c)OraPub, Inc.

ASH SQL Run Times

# Run times by plan for g<sub>x</sub>kuv8wrkvskg.



```
def ashWhere="(program is null or program not like '%(P%)%')  
           and sql_id='gxkuv8wrkvskg'"  
  
THE_COUNT SQL_ID          SQL_PLAN_HASH_VALUE TO_CHAR(SQL_EXEC_S  
-----  
7 gxkuv8wrkvskg      3382745219 2016-Jun-22 13:48:  
4 gxkuv8wrkvskg      3382745219  
2 gxkuv8wrkvskg      3382745219 2016-Jun-22 12:55:  
1 gxkuv8wrkvskg      3382745219 2016-Jun-22 13:18:  
1 gxkuv8wrkvskg      3382745219 2016-Jun-22 15:12:  
1 gxkuv8wrkvskg      3382745219 2016-Jun-22 13:03:  
1 gxkuv8wrkvskg      3382745219 2016-Jun-22 13:10:  
  
7 rows selected.
```

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

# Run times by plan for 4b3rd<sub>u</sub>4yh4ztc.



```
def ashWhere="(program is null or program not like '%(P%)%')  
           and sql_id='4b3rdu4yh4ztc'"  
  
THE_COUNT SQL_ID          SQL_PLAN_HASH_VALUE TO_CHAR(SQL_EXEC_STA SQL_EX  
-----  
61 4b3rdu4yh4ztc      790588729 2016-Jun-22 14:29:36 503  
26 4b3rdu4yh4ztc      480598186 2016-Jun-22 15:10:21 503  
26 4b3rdu4yh4ztc      480598186 2016-Jun-22 12:16:38 503  
16 4b3rdu4yh4ztc      480598186 2016-Jun-22 13:23:38 503  
12 4b3rdu4yh4ztc      480598186 2016-Jun-22 13:56:54 503  
9 4b3rdu4yh4ztc       480598186 2016-Jun-22 12:21:30 503  
9 4b3rdu4yh4ztc       480598186 2016-Jun-22 15:45:03 503  
8 4b3rdu4yh4ztc       480598186 2016-Jun-22 12:51:19 503  
8 4b3rdu4yh4ztc       480598186 2016-Jun-22 14:40:07 503  
7 4b3rdu4yh4ztc       480598186  
5 4b3rdu4yh4ztc       480598186 2016-Jun-22 13:26:35 503  
4 4b3rdu4yh4ztc       480598186 2016-Jun-22 15:46:53 503  
4 4b3rdu4yh4ztc       480598186 2016-Jun-22 15:15:15 503  
4 4b3rdu4yh4ztc       480598186 2016-Jun-22 13:59:13 503  
4 4b3rdu4yh4ztc       480598186 2016-Jun-22 12:53:10 503  
  
415 rows selected.
```

# Let's do some "real" data analysis!

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



# Demo

# Failure Slides

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## Simply change "Top Anything" for simple R analysis

```
def ashCols="sql_id, sql_plan_hash_value,
           to_char(SQL_EXEC_S -- select * from (
           sql_exec_id, '::::' select the_count||',' from (
def ashstate='%'                                     select count(&ashCount) the_
def ashCount="*"                                     count
def ashWhere="(program is null or program not like '%(P%)%')
           and sql_id= '4b3rdu4yh4ztc'"

-- select * from (
select the_count||',' from (
           select count(&ashCount) the_count, &ashCols
           from  &datasource
           where  &timingdetails
                  &dbahistdetails
           and  session_state like '&ashstate'
           and  &ashWhere
           group by &ashCols
           order by 1 desc, 2 desc
)
           where rownum < 100
/

```

ORAPUB

```
THE_COUNT||','
```

```
61,  
26,  
26,  
16,  
12,  
9,  
9,  
8,  
8,  
7,  
5,  
4,  
4,  
4,  
4,
```

Simple change to  
get data ready  
for simple R  
analysis

```
15 rows selected.
```

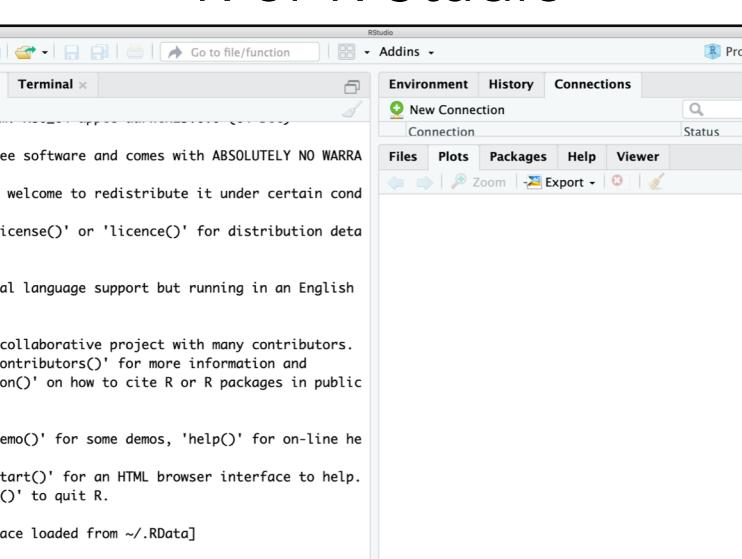
50

(c)OraPub, Inc.

ASH SQL Run Times

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

# R or R Studio



R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |

```

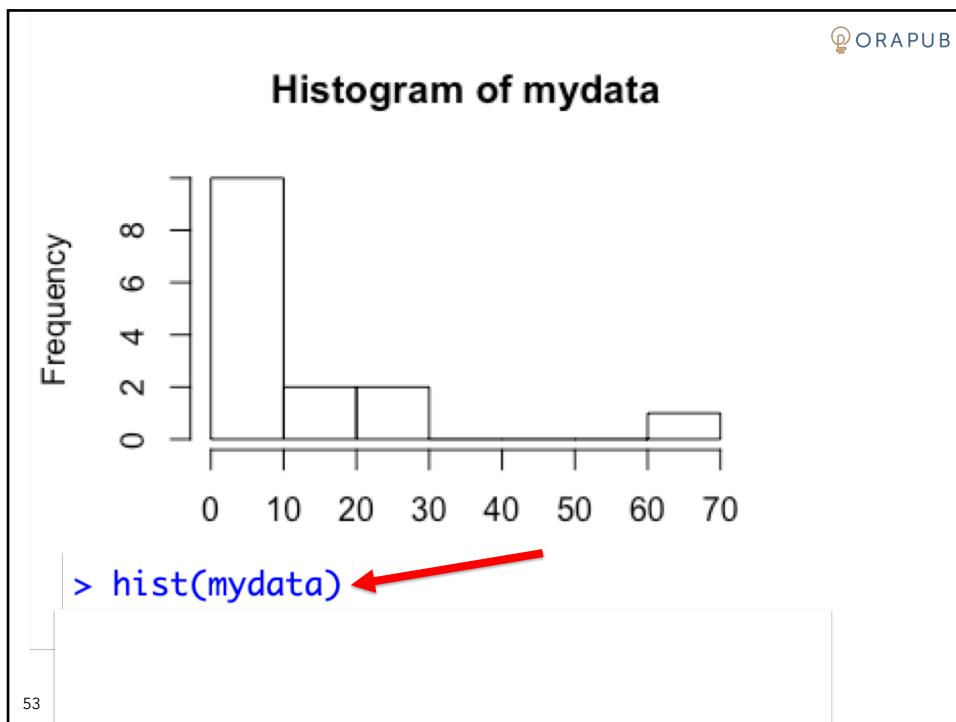
> mydata = c(61,
+           26,
+           26,
+           16,
+           12,
+           9,
+           9,
+           8,
+           8,
+           7,
+           5,
+           4,
+           4,
+           4)
> mydata
[1] 61 26 26 16 12 9 9 8 8 7 5 4 4 4 4
> summary(mydata)
  Min. 1st Qu. Median     Mean 3rd Qu.    Max.
  4.00   4.50   8.00  13.53  14.00  61.00
> quantile(mydata, c(.05, .50, .90))
  5% 50% 90%
  4   8   26
>
> mydata
[1] 61 26 26 16 12 9 9 8 8 7 5 4 4 4 4
> summary(mydata)
  Min. 1st Qu. Median     Mean 2nd Qu.    Max.
  4.00   4.50   8.00  13.53  14.00  61.00

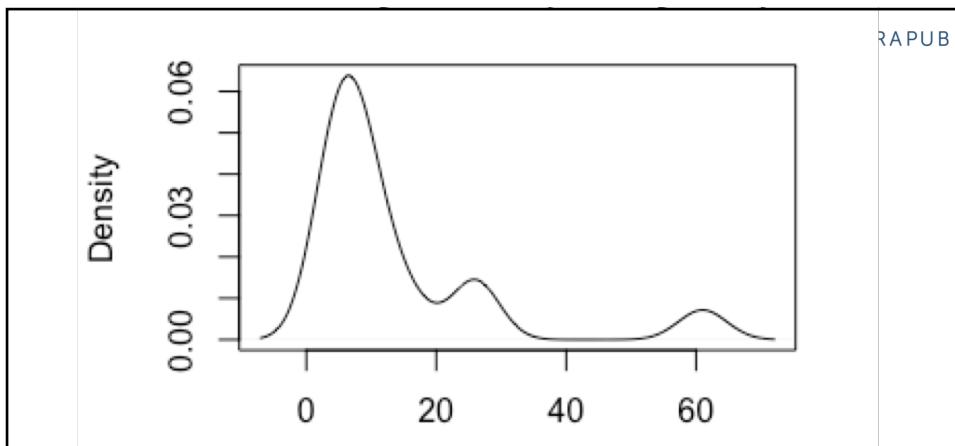
```

ASH SQL Run Times

52

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)





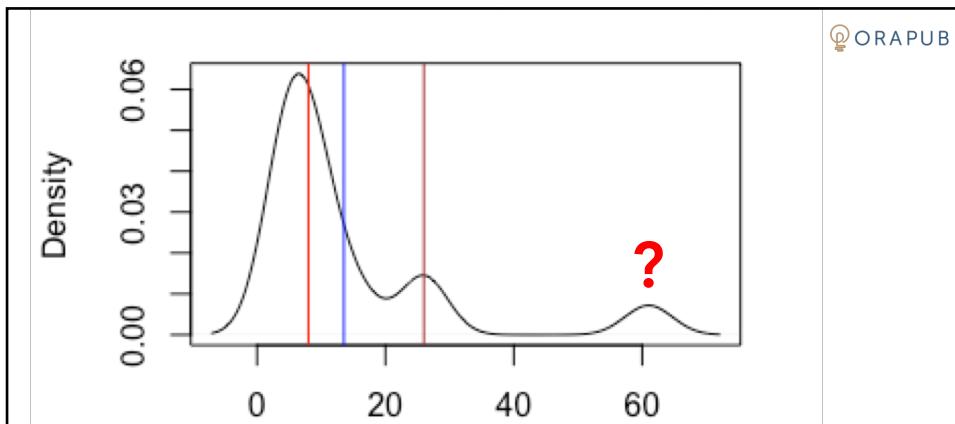
```

> hist(mydata)
> plot(density(mydata)) →
> abline(v=mean(mydata), col="blue")
> abline(v=median(mydata), col="red")
> abline(v=quantile(mydata,c(.90)), col="brown")

```

54

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



```

> hist(mvdata)
> plot(density(mydata))
> abline(v=mean(mydata), col="blue")
> abline(v=median(mydata), col="red")
> abline(v=quantile(mydata,c(.90)), col="brown")

```

5

So awesome! But...  
This is just the beginning.  
There is so much more when  
mastering ASH analysis.

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the  
March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)



[www.orapub.com/tools](http://www.orapub.com/tools)

[www.orapub.com](http://www.orapub.com) >  
Login >  
TOOLS & PRESENTATIONS >  
Tools

## Tuning Oracle Using ASH Strategies



### OraPub LVCs Are Different

You learn to master the topic at a deeper, higher confidence and more practical level compared to any other teaching method I offer.

This occurs because the class is spread out over multiple weeks, each session is 2 hours with a day in between, you do homework/activation on your real systems and I personally work with you through the entire class.

Details & Registration: [www.orapub.com/lvc-ash](http://www.orapub.com/lvc-ash)  
Event Calendar: <https://www.orapub.com/events>

This presentation was given by Craig Shallahamer (craig@orapub.com) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

Download ALL my presentations:  
[www.orapub.com/ppts](http://www.orapub.com/ppts)

FREE member webinars  
[www.orapub.com/free-webinars](http://www.orapub.com/free-webinars)

# Thank You!



This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)

## How To Analyze SQL Run Times By Plan Using ASH Data

Craig Shallahamer  
[craig@orapub.com](mailto:craig@orapub.com)



[www.orapub.com/ppts](http://www.orapub.com/ppts)



**utooug**  
utah oracle users group

training days  
march 13-14, 2019  
salt lake city, UT

For the most updated version  
of this presentation, go to:

[www.orapub.com/ppts](http://www.orapub.com/ppts)

This presentation was given by Craig Shallahamer ([craig@orapub.com](mailto:craig@orapub.com)) at the  
March 2019 UTOUG Conference in Salt Lake City, Utah USA.  
There is likely a more recent version at [www.orapub.com/ppts](http://www.orapub.com/ppts)