
# Data Pump
## Best Practices and Real World Scenarios
### Migration Tips and Tricks



Photo by Benjamin Davies on Unsplash

# Daniel Overby Hansen

## Senior Principal Product Manager

## Cloud Migration


dohdatabase

@dohdatabase

https://dohdatabase.com

Episode 1

Release and Patching Strategy

*105 minutes – Feb 4, 2021*


Episode 2

AutoUpgrade to Oracle Database 19c

*115 minutes – Feb 20, 2021*


Episode 3

Performance Stability, Tips and Tricks and Underscores

*120 minutes – Mar 4, 2021*


Episode 4

Migration to Oracle Multitenant

*120 minutes – Mar 16, 2021*


Episode 5

Migration Strategies – Insights, Tips and Secrets

*120 minutes – Mar 25, 2021*


Episode 6

Move to the Cloud – Not only for techies

*115 minutes – Apr 8, 2021*


# Recorded Web Seminars

https://MikeDietrichDE.com/videos

More than 30 hours of technical content, on-demand, anytime, anywhere

real world scenarios

# DATA PUMP

best practices

**INTRO**

UPGRADE

MOVE

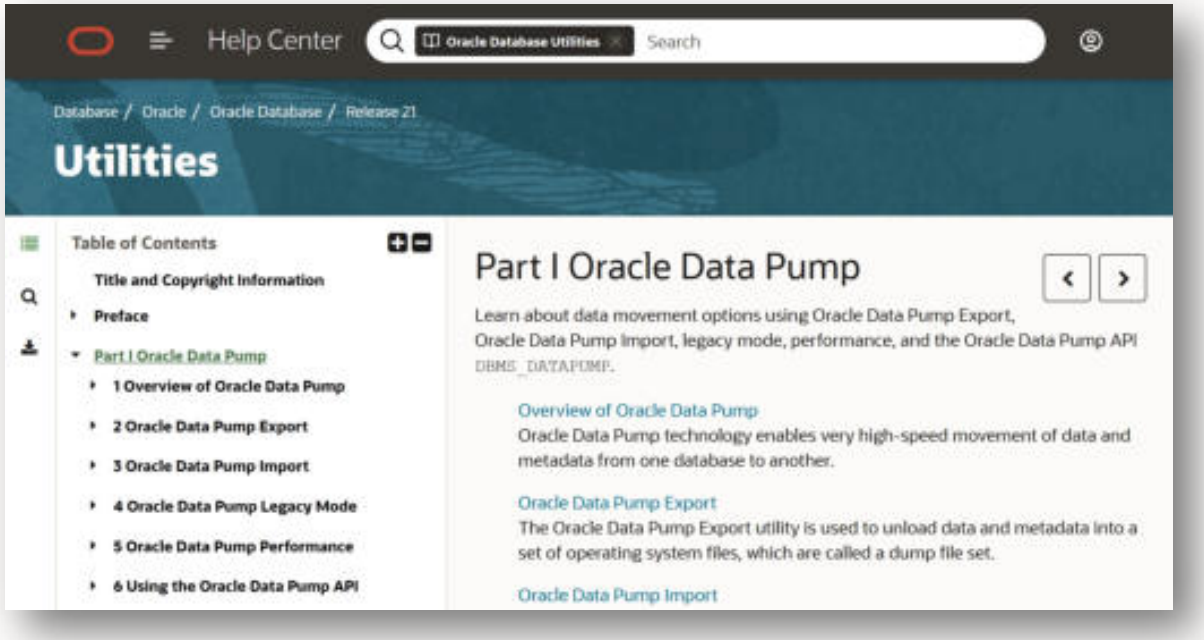
STATISTICS

LOBS

"Oracle Data Pump technology enables very high-speed movement of data and metadata from one database to another."

**Oracle Database Utilities 19c**

# Data Pump | Documentation




[Oracle Database 19c – Utilities Guide]

[Oracle Database 21c – Utilities Guide]

# Use a Data Pump parameter (.par) file

- Avoid errors typing long commands

```
$ cat export.par
schemas=app
directory=dp_dir
```

```
$ expdp dpuser parfile=export.par
```




Specify parallelism
Use multiple dump files

## Use PARALLEL parameter

```
expdp ... parallel=n
impdp ... parallel=n
```

## Use DUMPFILE parameter

```
expdp ... dumpfile=mydump%L.dmp
expdp ... dumpfile=mydump%L.dmp filesize=5G
```




# Include diagnostics in the logfile

```
expdp ... logtime=all metrics=yes
```

```
impdp ... logtime=all metrics=yes
```




# Use Interactive Command Mode

1. Press CTRL+C in Data Pump session

2. Attach from different Data Pump session

```
$ expdp .... attach=<job name>
$ impdp .... attach=<job name>
```

# Data Pump | Bundle Patch

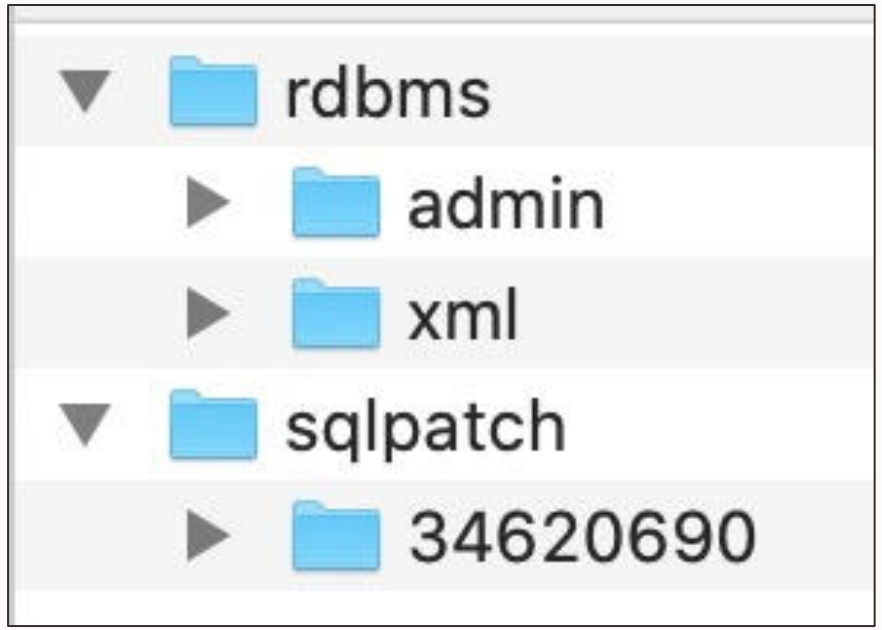
## Fewer Bugs

Important patches are included.
Monitor for bugs that affects many customers.

## Faster Patching

The bundle patch changes the way Data Pump is patched. Subsequent patches apply faster.

# Data Pump | Bundle Patch Contents




Bundle Patch contains only:

- `sql`
- `plsql`
- `xml`

But it does not contain any files which require a compilation/make of `rdbms`

➔ It can be applied ONLINE

# Update to the latest Release Update and then apply the Data Pump bundle patch

Data Pump Recommended Proactive Patches
For 19.10 and Above (Doc ID 2819284.1)

# The Data Pump bundle patch is not in the Oracle Database Release Update

It is not RAC Rolling and Standby-first Installable

When you run `datapatch`, ensure that there are no active Data Pump jobs

# Importing a complete application with data drops from almost 2.5 hours to 48 minutes – by just applying the Data Pump bundle patch

A global provider of financial services

real world scenarios

# DATA PUMP

best practices

INTRO

**UPGRADE**

MOVE

STATISTICS

LOBS

# You can use Data Pump to move data into a newer release of Oracle Database

- Oracle recommends upgrading the database using AutoUpgrade

# Upgrade via Data Pump

Suitable when

- Small amount of data
- Less complex database
- Going to multitenant
- Re-organization is required

# Upgrade via Data Pump

Considerations

- Longer downtime
- AutoUpgrade made upgrades much easier
- A full export might be the best option

real world scenarios

# DATA PUMP

best practices

INTRO

UPGRADE

**MOVE**

STATISTICS

LOBS

To migrate your data, you typically use Data Pump in schema or full mode

**SCHEMA**

Individual schemas and what they own

**FULL**

All schemas plus
more or less everything in the database

# Move | Full Export

## Objects exported only in full export:

- Audit trail and policies
- Database Vault
- Directories
- Profiles and password verify function
- Public database links
- Public synonyms
- Roles
- SQL Management Objects (plan histories, SQL plan baselines, SQL profiles, etc.)
- Tablespaces
- Users (other than those specified in `SCHEMAS` parameter)
- Workspace manager (for schema export you need to use `DBMS_WM.Export_Schemas`)

...

# Data Pump never exports grants on SYS objects

- Not even in a full export
- Add them manually following the import

# Data Pump never exports AWR

- Not even in a full export
- Use `rdbms/admin/awrextr.sql`

real world scenarios

# DATA PUMP

best practices

INTRO

UPGRADE

MOVE

**STATISTICS**

LOBS

1 Include statistics in Data Pump

2 Exclude statistics in Data Pump
Regather statistics after import

3 Exclude statistics in Data Pump
Import statistics using `DBMS_STATS`

# Generally, we recommend excluding statistics from Data Pump export

- Use `EXCLUDE=STATISTICS`

# Transporting Statistics | Customer Feedback

> *We have adopted this method for stats. We migrated 60 TB database from AIX to Exadata using cross-platform transportable tablespace without stats.*
>
> *Gathering stats from scratch took more than 36 hours.*
> *We transported the statistics in less then 2 hours.*

Taoqir Hassan, comment on YouTube channel

On YouTube we have videos on DBMS_STATS, including a demo and pro tips

**EXCLUDE=STATISTICS** ⇒

Table statistics

Index statistics

Statistics preferences

Column usage information

```
EXCLUDE=STATISTICS
```

Table statistics

Index statistics

**Statistics preferences**

Column usage information

```
BEGIN
    DBMS_STATS.SET_TABLE_PREFS (
        OWNNAME => '...',
        TABNAME => '...',
        PNAME    => 'TABLE_CACHED_BLOCKS',
        PVALUE   => '42'
    );
END;
```




***Table 171-131 SET_TABLE_PREFS Procedure Parameters***

| Parameter | Description |
|---|---|
| ownname | Owner name |
| tabname | Table name |
| pname | Preference name. You can set the default value for following preferences:<br>• APPROXIMATE_NDV_ALGORITHM<br>• AUTO_STAT_EXTENSIONS<br>• CASCADE<br>• DEGREE<br>• ESTIMATE_PERCENT<br>• GRANULARITY<br>• INCREMENTAL<br>• INCREMENTAL_LEVEL<br>• INCREMENTAL_STALENESS<br>• METHOD_OPT<br>• NO_INVALIDATE<br>• OPTIONS<br>• PREFERENCE_OVERRIDES_PARAMETER<br>• PUBLISH<br>• STALE_PERCENT<br>• TABLE_CACHED_BLOCKS |
| pvalue | Preference value. If NULL is specified, it will set the Oracle default value. |

PL/SQL Packages and Types Reference, Oracle Database 19c

# Data Pump exports table-level statistics preferences together with table statistics

- In full, schema and table mode
- In transportable, it is controlled by `USER_PREF_STATISTICS`

# Data Pump never exports global statistics preferences

- Not even in a full export
- Define manually using `DBMS_STATS.SET_GLOBAL_PREFS`

DBMS_STATS package has dedicated procedures for transporting table-level statistics preferences

## You often use statistics preferences to solve a particular problem

- Evaluate whether that problem exists in the target environment

```
EXCLUDE=STATISTICS
```

Table statistics

Index statistics

Statistics preferences

**Column usage information**

# Statistics | Column Usage Information

- Information on how you join tables
- Used by the optimizer to determine when to create histograms
  `METHOD_OPT => ... SIZE AUTO`
- When missing, statistics gathering creates no or few histograms
- Stored internally in `SYS.COL_USAGE$`

When Data Pump transfers statistics,
it also transfers column usage information

## EXCLUDE

`EXCLUDE=STATISTICS`

`COL_USAGE$` empty

## REGATHER

First time only

```
METHOD_OPT =>
SIZE SKEWONLY
```

## GO LIVE

Column usage information is updated

## REGATHER

Use default

```
METHOD_OPT =>
SIZE AUTO
```

# Importing statistics might be a bad idea

When source and target database do not match




# Statistics | When Importing Stats Is Bad

Fragmented table

| | |
|---|---|
| Blocks | 12000 |
| Leaf blocks | 11000 |
| B-level | 4 |
| Clustering factor | 10000 |

Compacted table

| | |
|---|---|
| Blocks | 12000 |
| Leaf blocks | 11000 |
| B-level | 4 |
| Clustering factor | 10000 |

```
DBMS_STATS.GATHER_TABLE_STATS(...
```

| | |
|---|---|
| Blocks | 5000 |
| Leaf blocks | 4000 |
| B-level | 2 |
| Clustering factor | 20000 |

# Statistics | When Importing Stats Is Bad

- Potentially a problem
  - Fragmented tables
  - Changing block size
  - Changing character set
  - Compress or decompress

    …
- Only a problem for table and index base statistics, column statistics remain accurate

Accurate statistics is the starting point for good performance

# Comparing STATISTICS options

| | Import with Data Pump | Regather | Import with DBMS_STATS |
|---|---|---|---|
| **Time** | Significant | Significant | Short |
| **Column usage information** | Included | Missing | Missing |
| **Accuracy** | Potentially inaccurate | Accurate | Potentially inaccurate |
| **Statistics preferences** | Included | Missing | Optional |

real world scenarios

# DATA PUMP

best practices

INTRO

UPGRADE

MOVE

STATISTICS

**LOBS**

# A short history of **binary data types**




| | |
|---|---|
| v4 | LONG and LONG RAW |
| 8*i* | CLOB and BLOB |
| 10g | SecureFile LOBs |

v4 LONG and LONG RAW

8i BasicFile LOBs

10g SecureFile LOBs




v4

## LONG and LONG RAW

- Only 1 column per table
- Max size: 2GB - 1

8*i*

## BasicFile LOBs

- Performance constraints
- Data Pump can act with one worker only
- Max size: (4GB - 1) * `DB_BLOCK_SIZE`

10g

## SecureFile LOBs

- Improved performance
- Data Pump can act with multiple workers
- Deduplication, encryption and more
- Max size: same as with CLOB/BLOB

As of today, all legacy binary data types should have been migrated to SecureFile LOBs

```
impdp ... transform=lob_storage:securefile
```




# Different LOB types

Internal LOBs stored inside the database

- CLOB
- NCLOB
- BLOB

External LOBs stored outside the database

- BFILE




# Initialization Parameter

DB_SECUREFILE

- NEVER
- PERMITTED
- PREFERRED ← LOBs are created as SecureFile LOBs unless explicitly stated
- ALWAYS
- IGNORE

Tablespace must use Automatic Segment Space Management (ASSM)




# Data Pump & LOBs
# **Things to know and consider**

# No parallelism with BasicFile LOBs

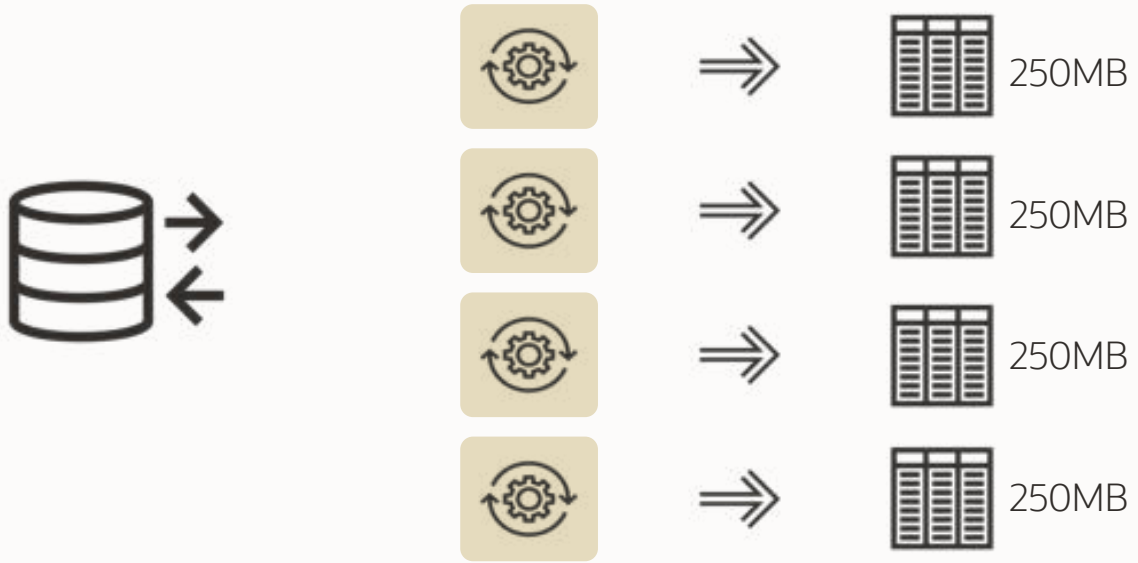
# Always use SecureFile LOBs

*"But why is there only one worker?"*




# Data Pump | Parallel Worker Activity

Normally, Data Pump *employs* one worker per 250MB table segment



# LOB Export | Example Table

```
CREATE OR REPLACE DIRECTORY BLOB_DIR AS '/tmp/mydir';
```

10GB

```
CREATE TABLE tab1 ( id NUMBER, blob_data BLOB );
```

```
BEGIN ... DBMS_LOB.LOADBLOBFROMFILE ...
```

```
exec DBMS_STATS.GATHER_TABLE_STATS('HUGO','TAB1');
```

For a complete example, please visit oracle-base.com

# LOB data is stored out-of-row in a separate LOB segment

- Smaller LOBs less than 4000 bytes are stored in-row

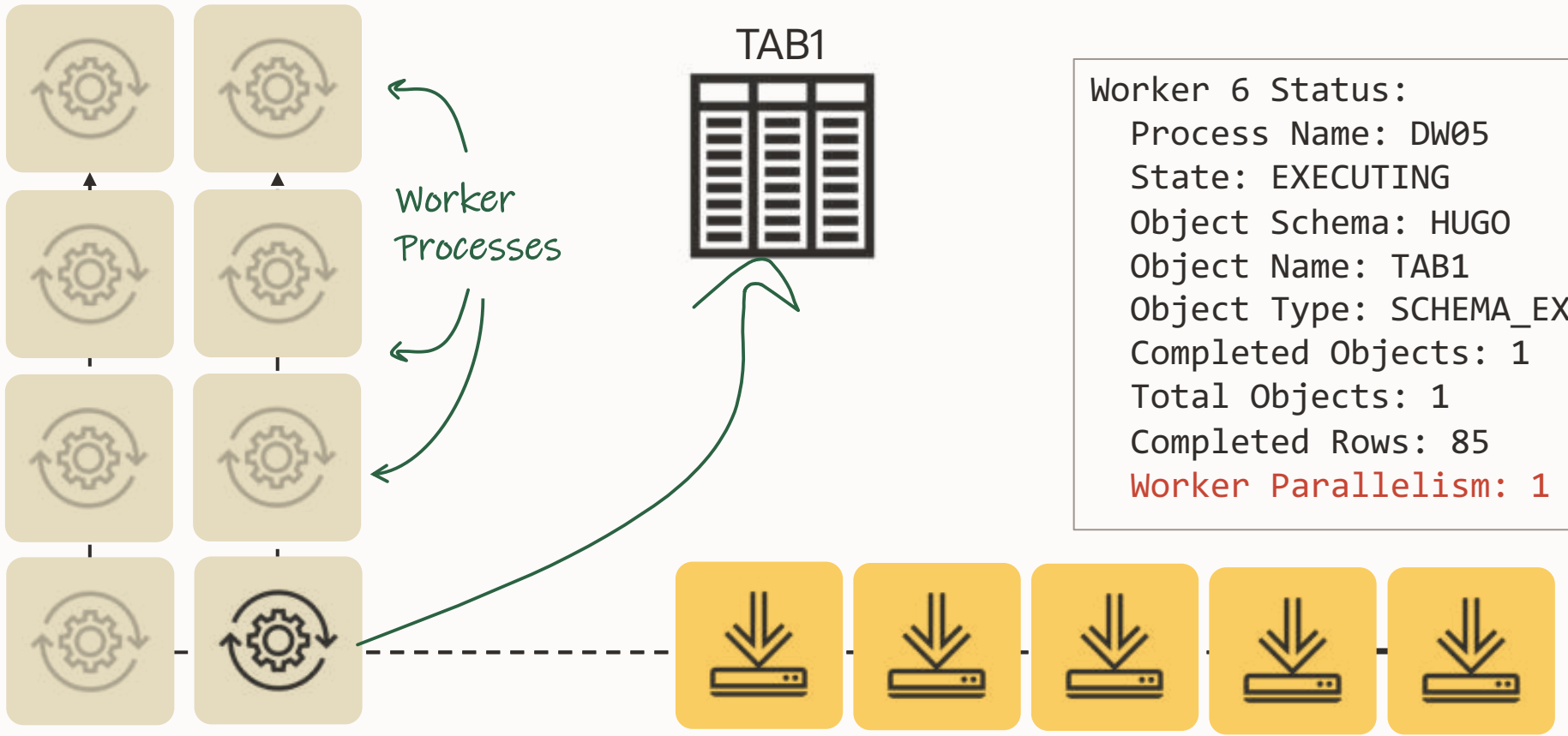
# Starting Data Pump – Test:

```
DIRECTORY=DATA_PUMP_DIR
DUMPFILE=MYDUMP%L.DMP
LOGFILE=MYDUMP01.LOG
SCHEMAS=HUGO
LOGTIME=ALL
METRICS=YES
PARALLEL=8
```




# LOB Export | Lazy Workers?

## 8 workers, 5 dump files – and only 1 worker exports TAB1




```
Worker 6 Status:
  Process Name: DW05
  State: EXECUTING
  Object Schema: HUGO
  Object Name: TAB1
  Object Type: SCHEMA_EXPORT/...
  Completed Objects: 1
  Total Objects: 1
  Completed Rows: 85
  Worker Parallelism: 1
```

Maybe the table's PARALLEL DEGREE is too low?

# LOB Export | Parallel Degree

TAB1

```
ALTER TABLE tab1 PARALLEL 8;
```

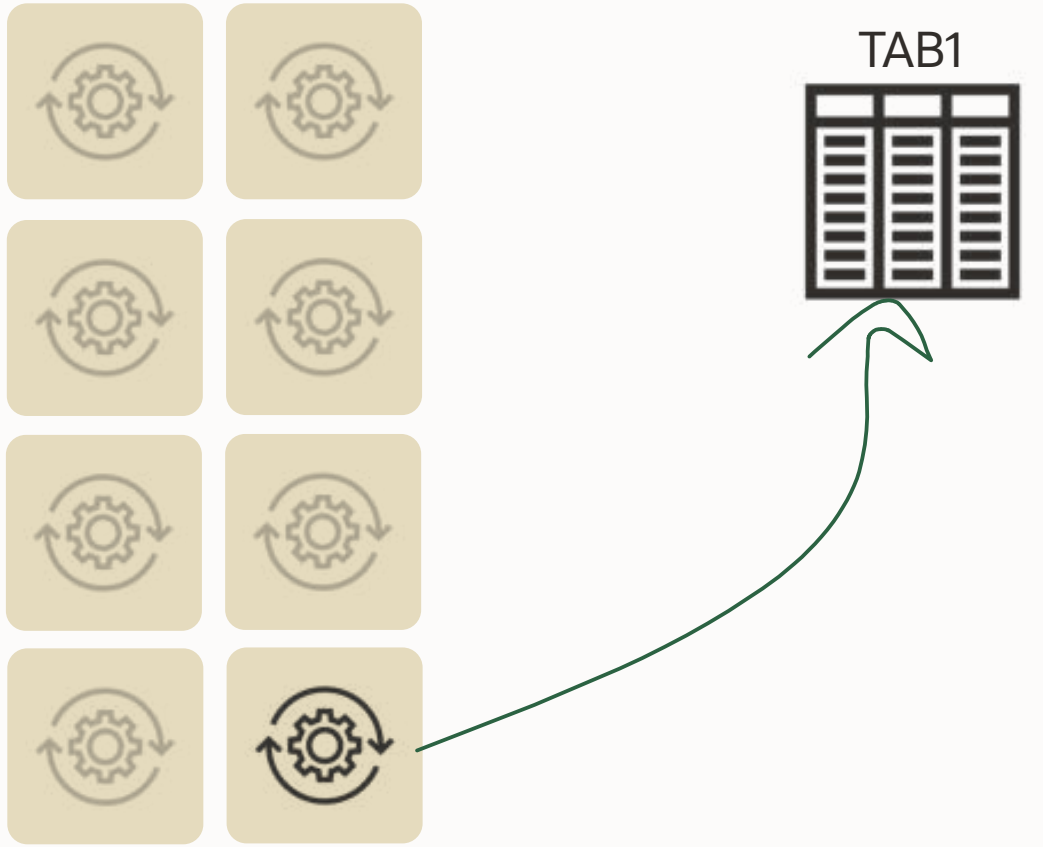
TAB1

```
select degree
from DBA_TABLES
where table_name='TAB1';

 DEGREE
________________
          1
```

```
select degree
from DBA_TABLES
where table_name='TAB1';

 DEGREE
________________
          8
```

# LOB Export | Parallel Degree

## No relief ☹




```
Worker 1 Status:
Process Name: DW08
  State: EXECUTING
  Object Schema: HUGO
  Object Name: TAB1
  Object Type: SCHEMA_EXPORT/...
  Completed Objects: 1
  Total Objects: 1
  Completed Rows: 85
  Worker Parallelism: 1
```

# LOB Export | Table Segments and Extents

TAB1

Segments

Extents

```
select BYTES, BLOCKS, EXTENTS
from   DBA_SEGMENTS
where  SEGMENT_NAME = 'TAB1'
and    OWNER = 'HUGO';
```

| BYTES | BLOCKS | EXTENTS |
|---|---|---|
| 131072 | 16 | 2 |

```
select ROUND(SUM(BYTES)/1024/1024/1024,2) "GB"
from   DBA_EXTENTS
where  SEGMENT_NAME IN
           (select SEGMENT_NAME
            from   DBA_LOBS
            where  TABLE_NAME = 'TAB1'
            and    OWNER = 'HUGO');
```

| GB |
|---|
| 10.31 |

# LOB Export | Table Statistics

TAB1

Table  Columns

```
select NUM_ROWS, BLOCKS, AVG_ROW_LEN
from   DBA_TAB_STATISTICS
where  TABLE_NAME = 'TAB1';
```

| NUM_ROWS | BLOCKS | AVG_ROW_LEN |
|---|---|---|
| 85 | 13 | 720 |

```
select COLUMN_NAME, NUM_DISTINCT,
       SAMPLE_SIZE, AVG_COL_LEN
from   DBA_TAB_COL_STATISTICS
where  TABLE_NAME='TAB1';
```

| COLUMN_N | NUM_DIST | SAMPLE_SIZE | AVG_COL_LEN |
|---|---|---|---|
| ID | 1 | 85 | 3 |
| BLOB_DATA | 0 | 85 | 717 |

It looks like Data Pump doesn't know anything about the dimensions of the LOB segment

# LOB Export | User Objects

```
select OBJECT_NAME, OBJECT_TYPE from DBA_OBJECTS
where OWNER = 'HUGO';

                    OBJECT_NAME     OBJECT_TYPE
_____________________________________ ___________________
TAB1                                  TABLE
SYS_IL0000070285C00002$$              INDEX
SYS_LOB0000070285C00002$$             LOB
```

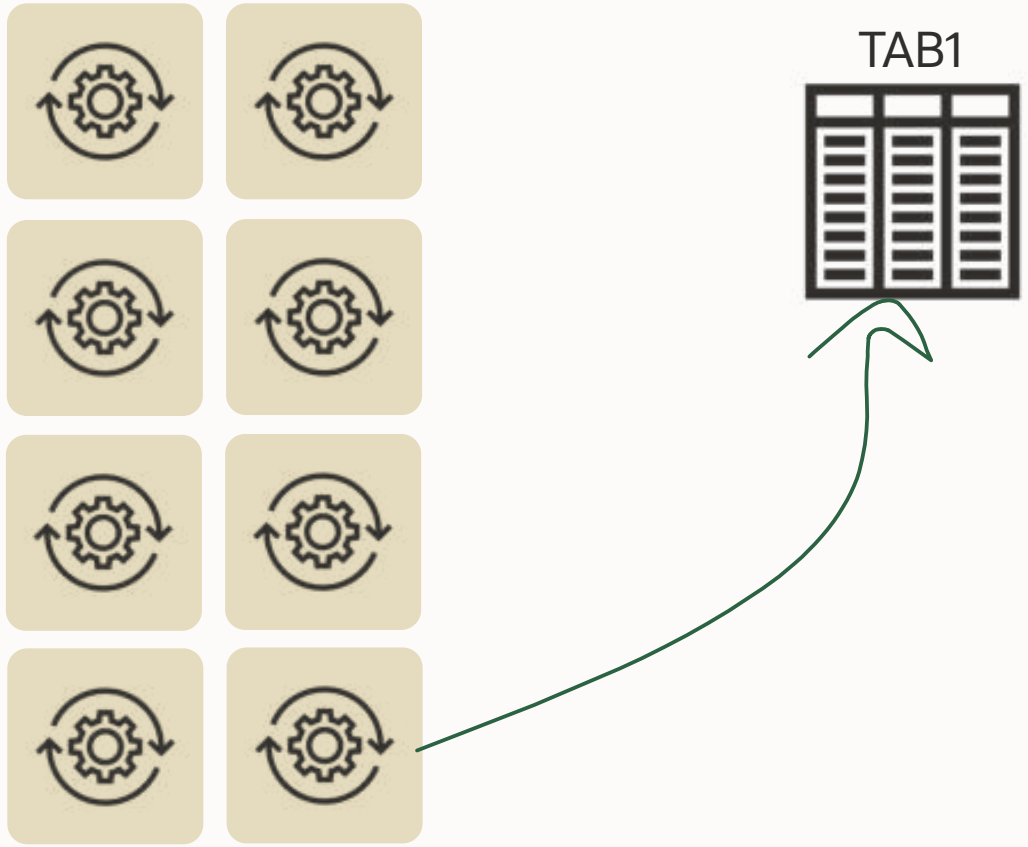
Is it possible to *analyze* a LOB segment?

# LOB Export | Manipulating Statistics

```
begin
 DBMS_STATS.SET_TABLE_STATS (
   ownname => 'HUGO',
   tabname => 'TAB1',
   numrows => 10000000,
   numblks => 1000000);
 end;
 /
```

# LOB Export | Parallel Degree

## Relief ☺ Workers do PQ now!




```
Worker 2 Status:
Process Name: DW01
  State: EXECUTING
  Object Schema: HUGO
  Object Name: TAB1
  Object Type: SCHEMA_EXPORT/...
  Completed Objects: 1
  Total Objects: 1
  Completed Rows: 85
  Completed Bytes: 1,474,081,152
  Worker Parallelism: 7
```

Why only one worker with PQ?
Why not multiple workers?

You can boost parallelism
by using partitioned tables

*"And BFILE LOBs?"*




# BFILE LOBs

## External LOBs stored outside the database

Full export:

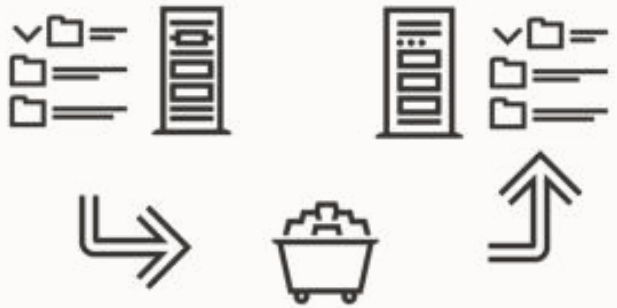
- Directory definition gets exported/imported
- You must copy the files

Schema export:

- You must create the directory within the database
- You must copy the files

Table export:

- You must create the directory within the database
- You must copy the files



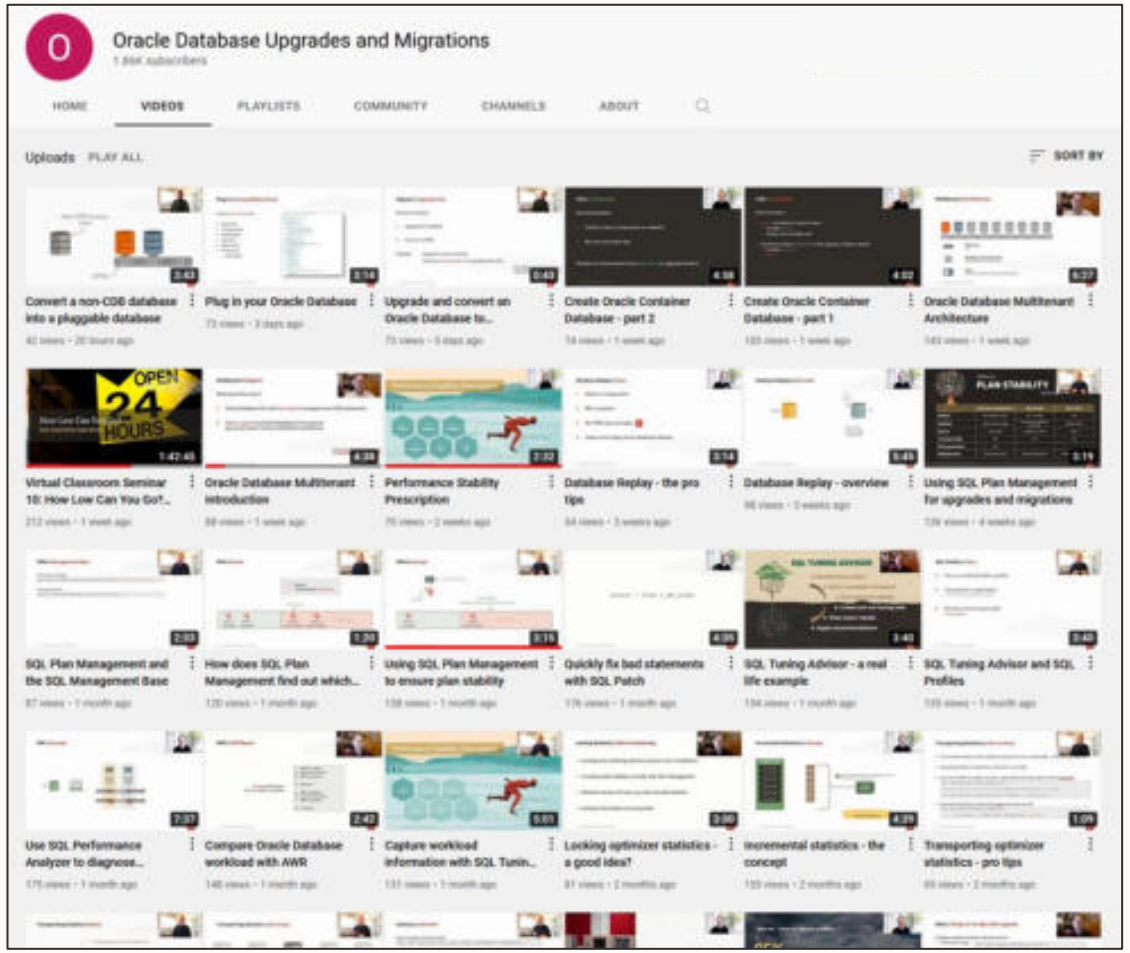
# Save downtime by copying the external files in advance

- BFILEs are always read-only

If the directory path changes,
make sure to update the directory object

Photo by Zbynek Burival on Unsplash

# YouTube | Oracle Database Upgrades and Migrations




Link

- 300+ videos
- New videos every week
- No marketing
- No buzzwords
- All tech

SCAN ME

# THANK YOU

**Visit our blogs:**

https://MikeDietrichDE.com

https://DOHdatabase.com

https://www.dbarj.com.br/en

# THANK YOU

**Webinars:**

https://MikeDietrichDE.com/videos

**YouTube channel:**

@UpgradeNow

# THANK YOU