Data Pump
Best Practices and Real World Scenarios
Virtual Classroom Series – Episode 15

Photo by Benjamin Davies on Unsplash






ROY SWONGER
Vice President
Database Upgrade, Utilities & Patching

royfswonger

@royfswonger






MIKE DIETRICH
Distinguished Product Manager
Database Upgrade and Migrations

mikedietrich

@mikedietrichde

https://mikedietrichde.com






DANIEL OVERBY HANSEN
Senior Principal Product Manager
Cloud Migrations

dohdatabase

@dohdatabase

https://dohdatabase.com






RODRIGO JORGE
Senior Principal Product Manager
Database Patching and Upgrade

rodrigoaraujorge

@rodrigojorgedba

https://dbarj.com.br/en






BILL BEAUREGARD
Senior Principal Product Manager
Data Pump and SQL Loader

william-beauregard-3053791






KLAUS GRONAU
Consulting Member
of Technical Staff

klaus-gronau-39a43aa9

Webinar | Get The Slides

https://MikeDietrichDE.com/slides

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

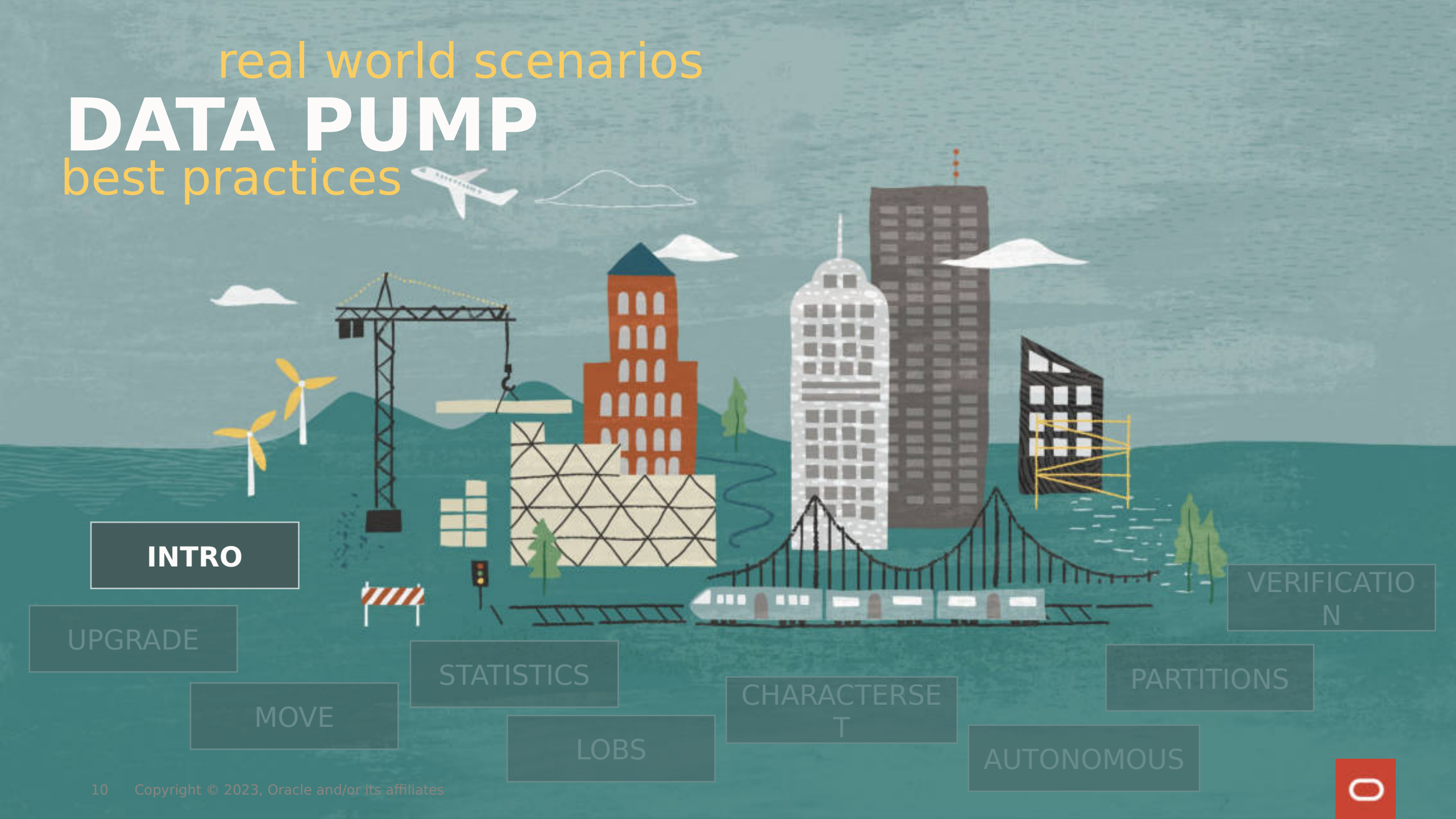
https://MikeDietrichDE.com/videos

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

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

PARTITIONS

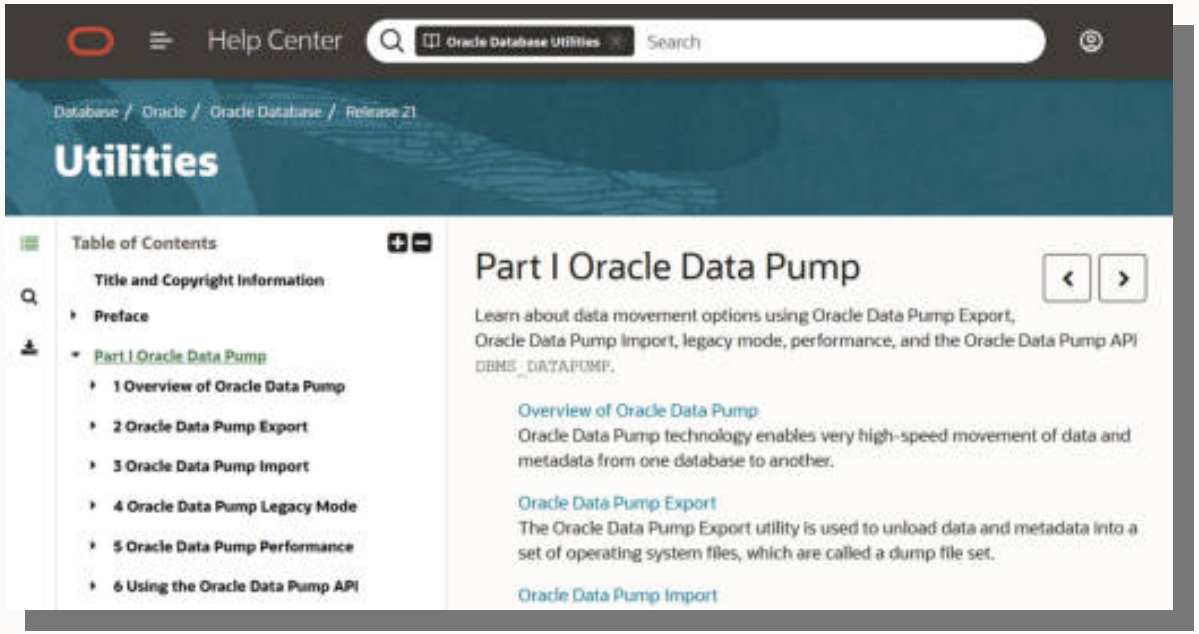
AUTONOMOUS

VERIFICATION

"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

Data Pump | Bundle Patch

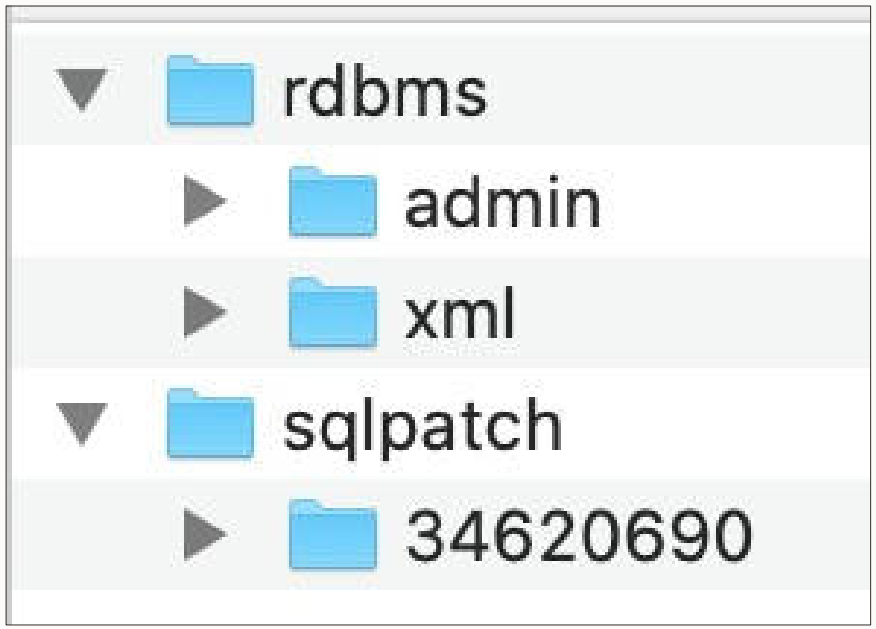
Fewer Bugs

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

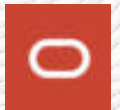
A global provider of financial services

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
```

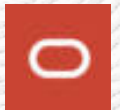
Specify parallelism
Use multiple dump files

Use PARALLEL parameter

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

Use DUMPFILE parameter

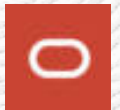
```
expdp ... dumpfile=mydump%L.dmp
expdp ... dumpfile=mydump%L.dmp filesize=5G
```

Include diagnostics in the logfile

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

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

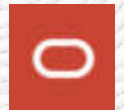



Use Interactive Command Mode

1. Press CTRL+C in Data Pump session

2. Attach from different Data Pump session

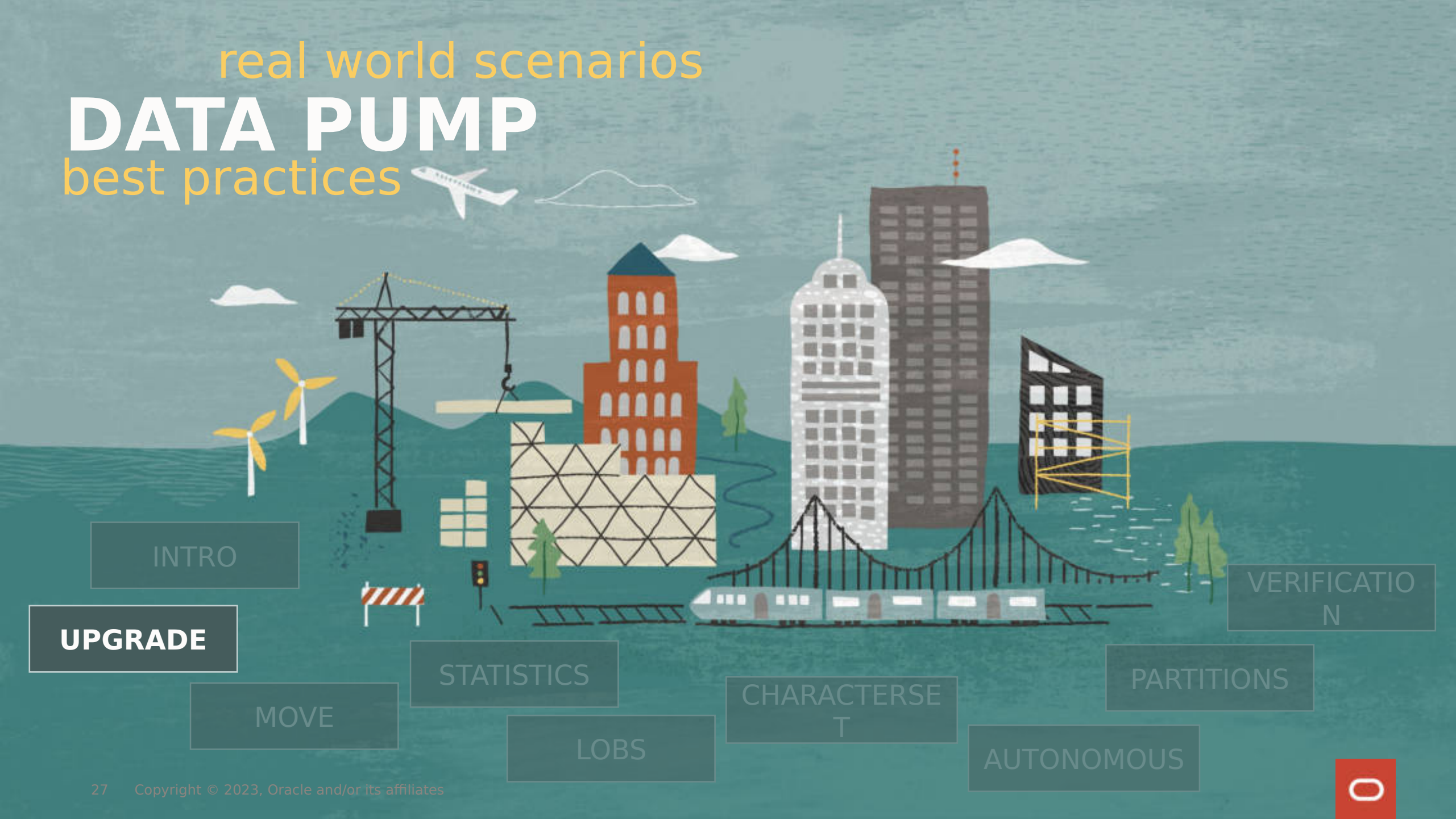
```
$ expdp .... attach=<job name>
$ impdp .... attach=<job name>
```

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

AUTONOMOUS

PARTITIONS

VERIFICATION

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

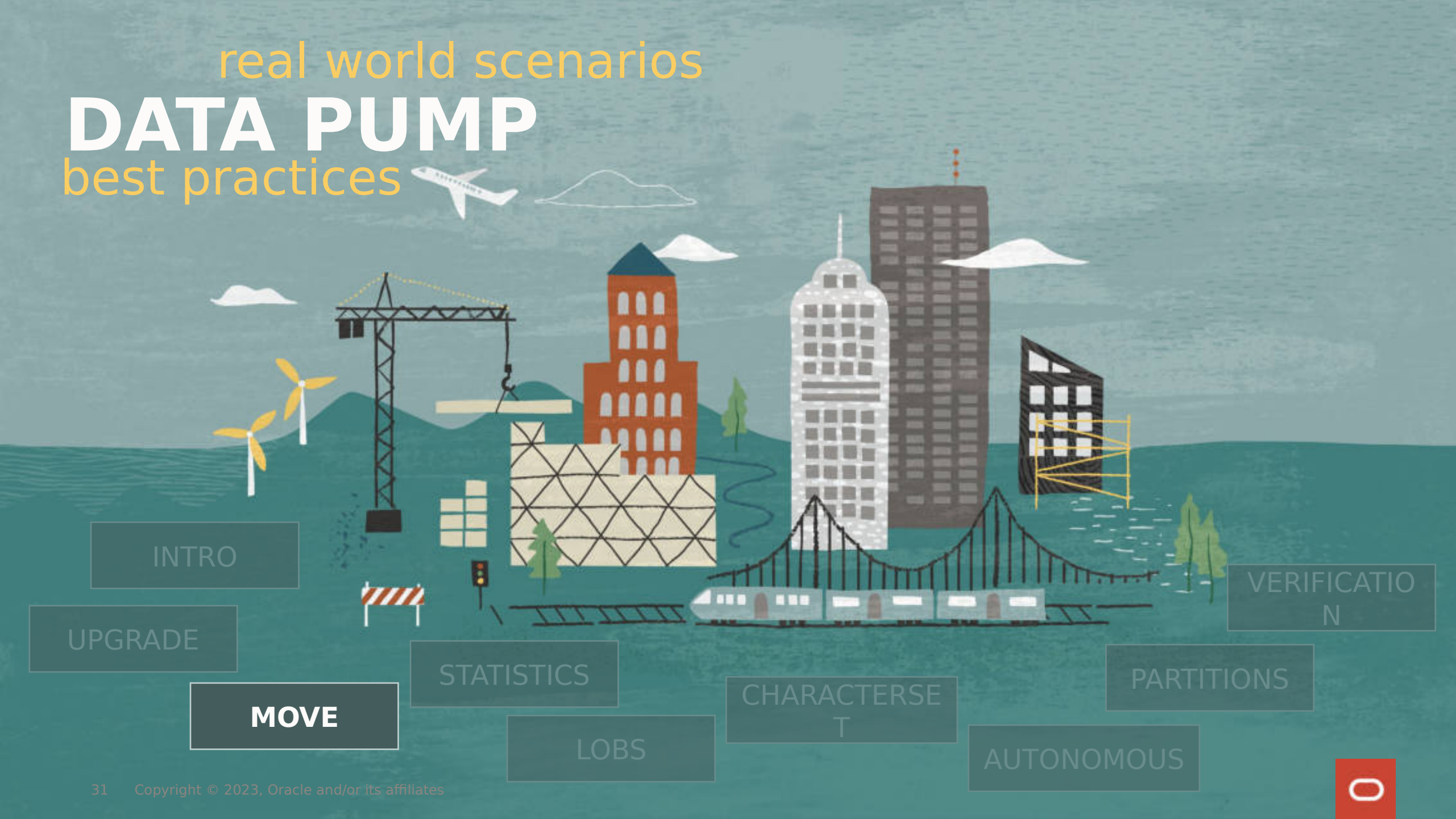
Considerations

- 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

CHARACTERSET

AUTONOMOUS

PARTITIONS

VERIFICATION

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

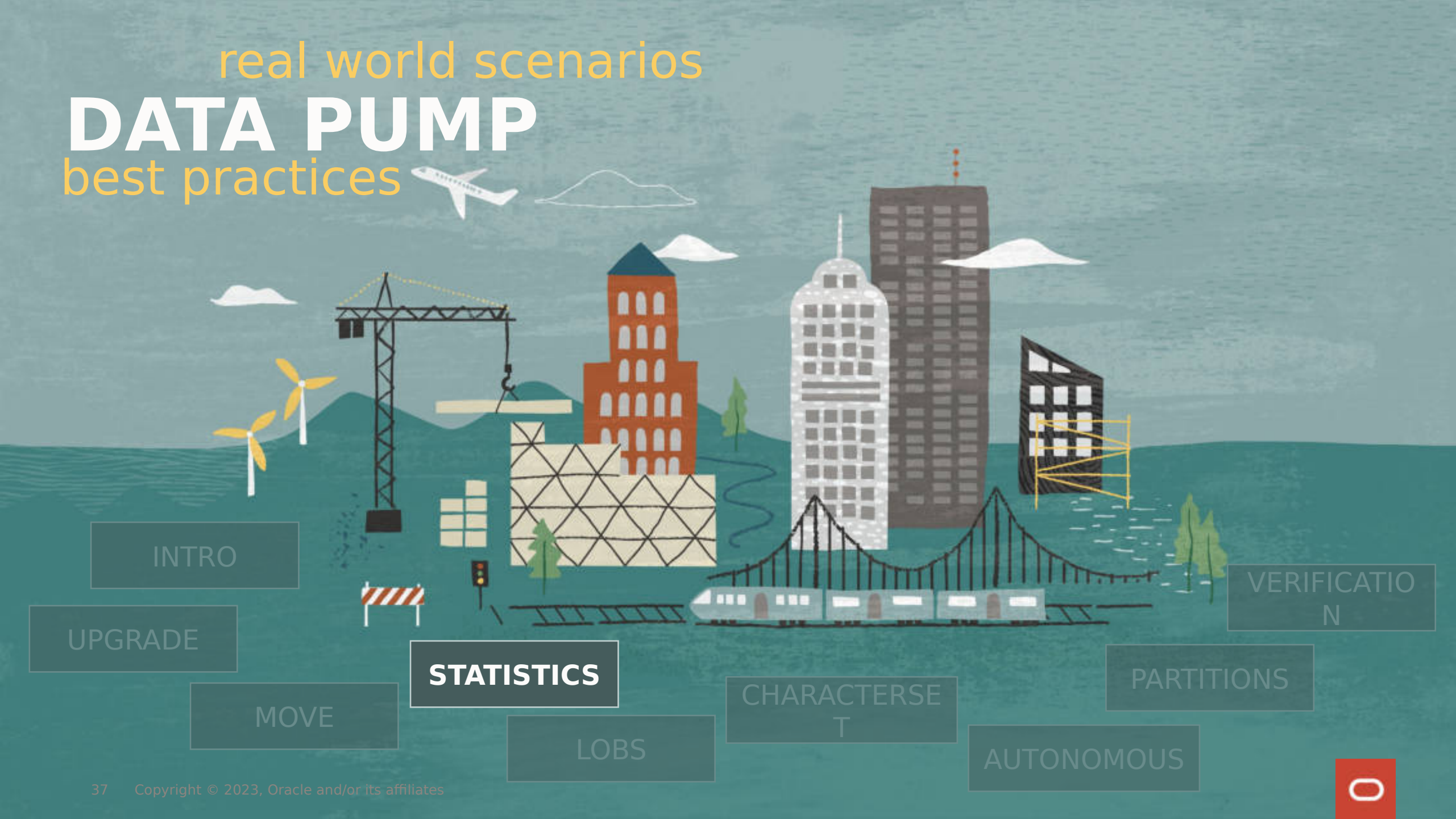
Data Pump never exports AWR

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

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

AUTONOMOUS

PARTITIONS

VERIFICATION

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`

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

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

Generally, we recommend excluding statistics from Data Pump export

- Use `EXCLUDE=STATISTICS`

`EXCLUDE=STATISTICS` ⇒

Table statistics

Index statistics

Statistics preferences

Column usage information

Table statistics

Index statistics

`EXCLUDE=STATISTICS` 

Statistics preferences

Column usage information

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

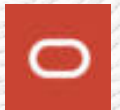



Table 171-131 SET_TABLE_PREFS Procedure Parameters

Parameter	Description
ownname	Owner name
tabname	Table name
pname	Preference name. You can set the default value for following preferences: • APPROXIMATE_NDV_ALGORITHM • AUTO_STAT_EXTENSIONS • CASCADE • DEGREE • ESTIMATE_PERCENT • GRANULARITY • INCREMENTAL • INCREMENTAL_LEVEL • INCREMENTAL_STALENESS • METHOD_OPT • NO_INVALIDATE • OPTIONS • PREFERENCE_OVERRIDES_PARAMETER • PUBLISH • STALE_PERCENT • 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

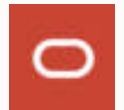
```
DBMS_STATS.GATHER_TABLE_STATS(...
```

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

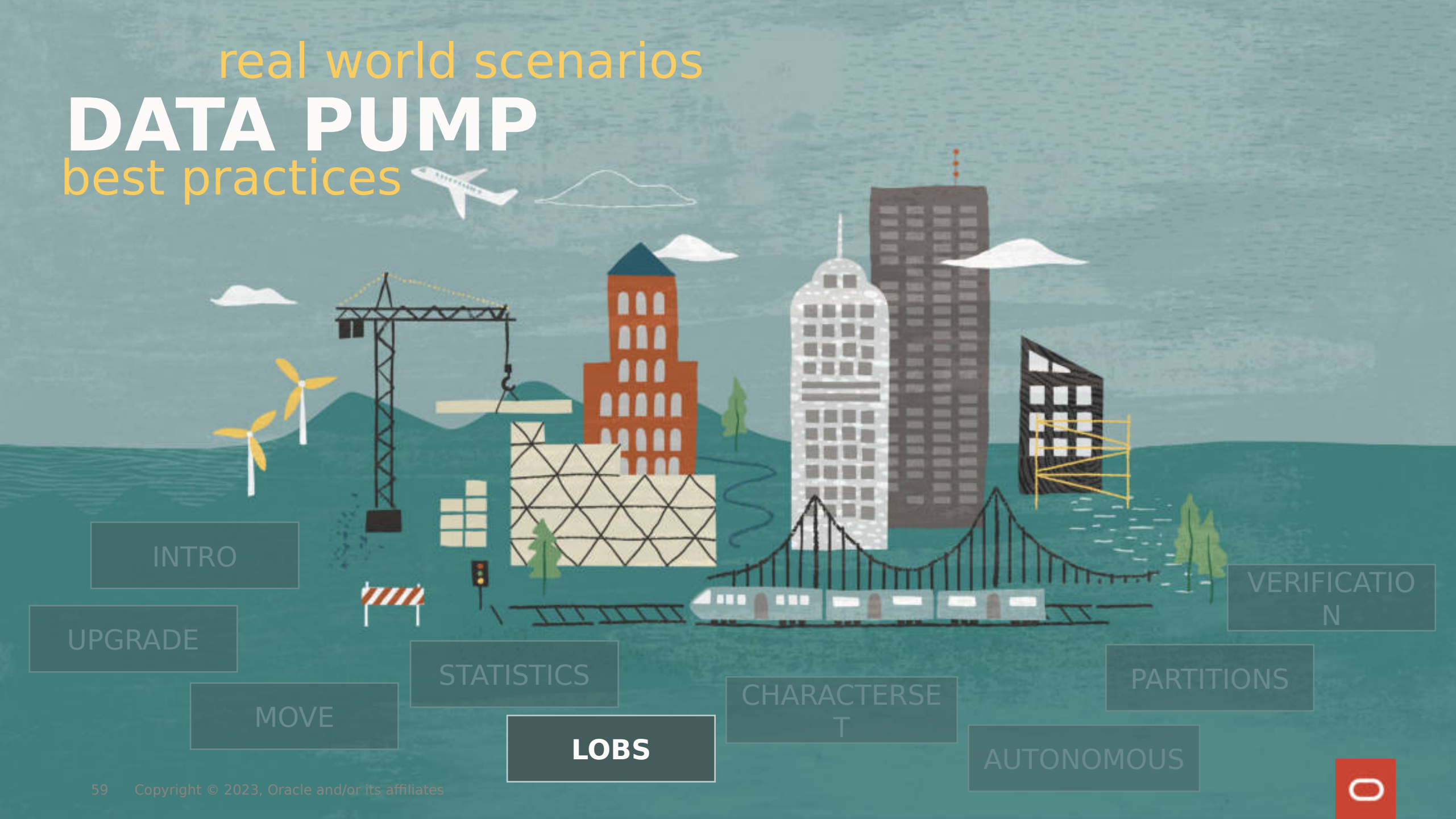
Comparing STATISTICS options

	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

CHARACTERSET

AUTONOMOUS

PARTITIONS

VERIFICATION

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

8*i* 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

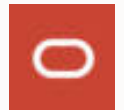
- BFILE

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

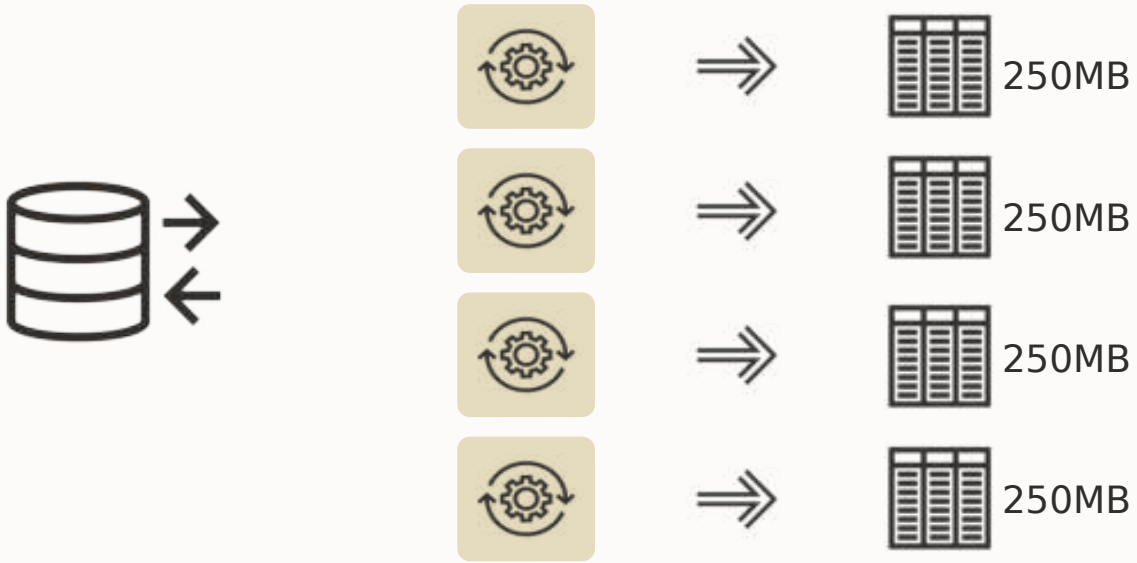
No parallelism with BasicFile LOBs

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

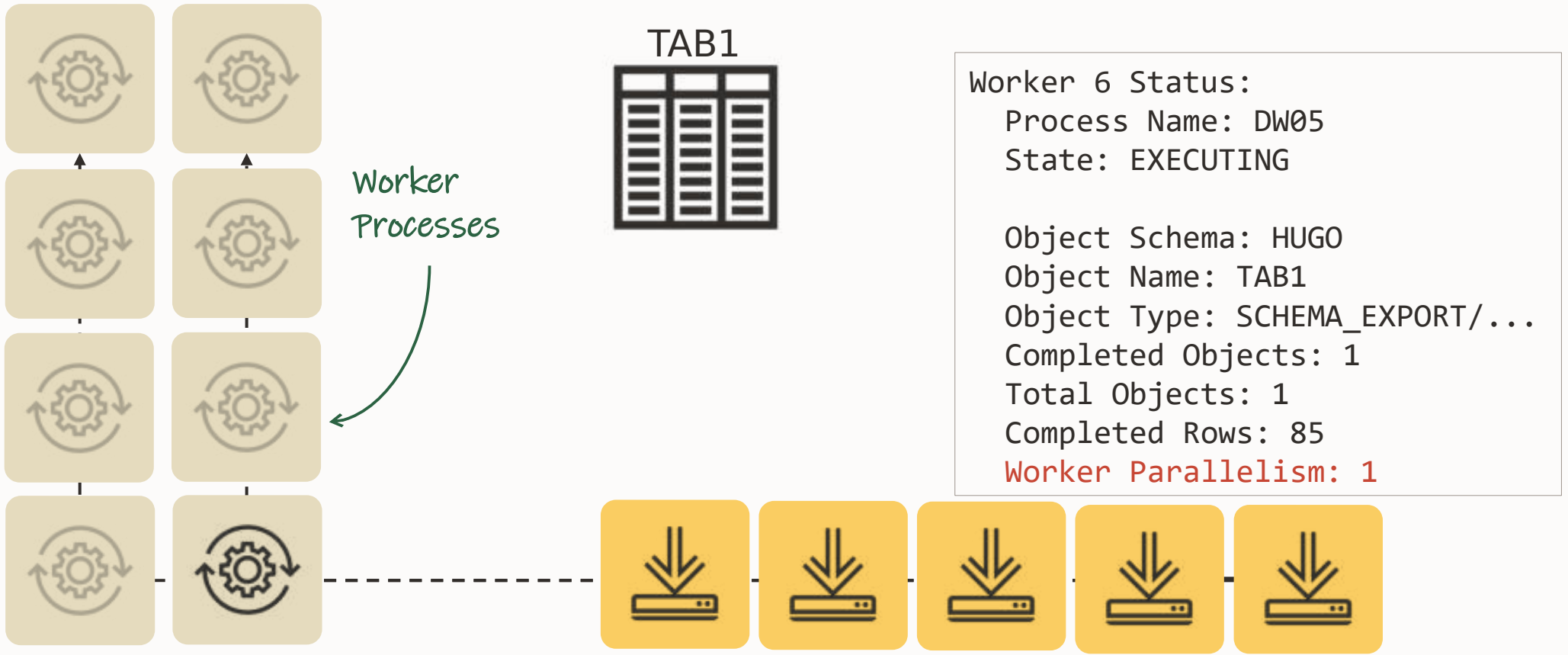
- Smaller LOBs less than 4000 bytes are stored in-row

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



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 ☹

TAB1

```
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
```

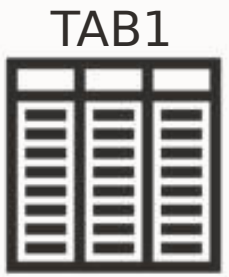
Is it possible to *analyze* a LOB segment?

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

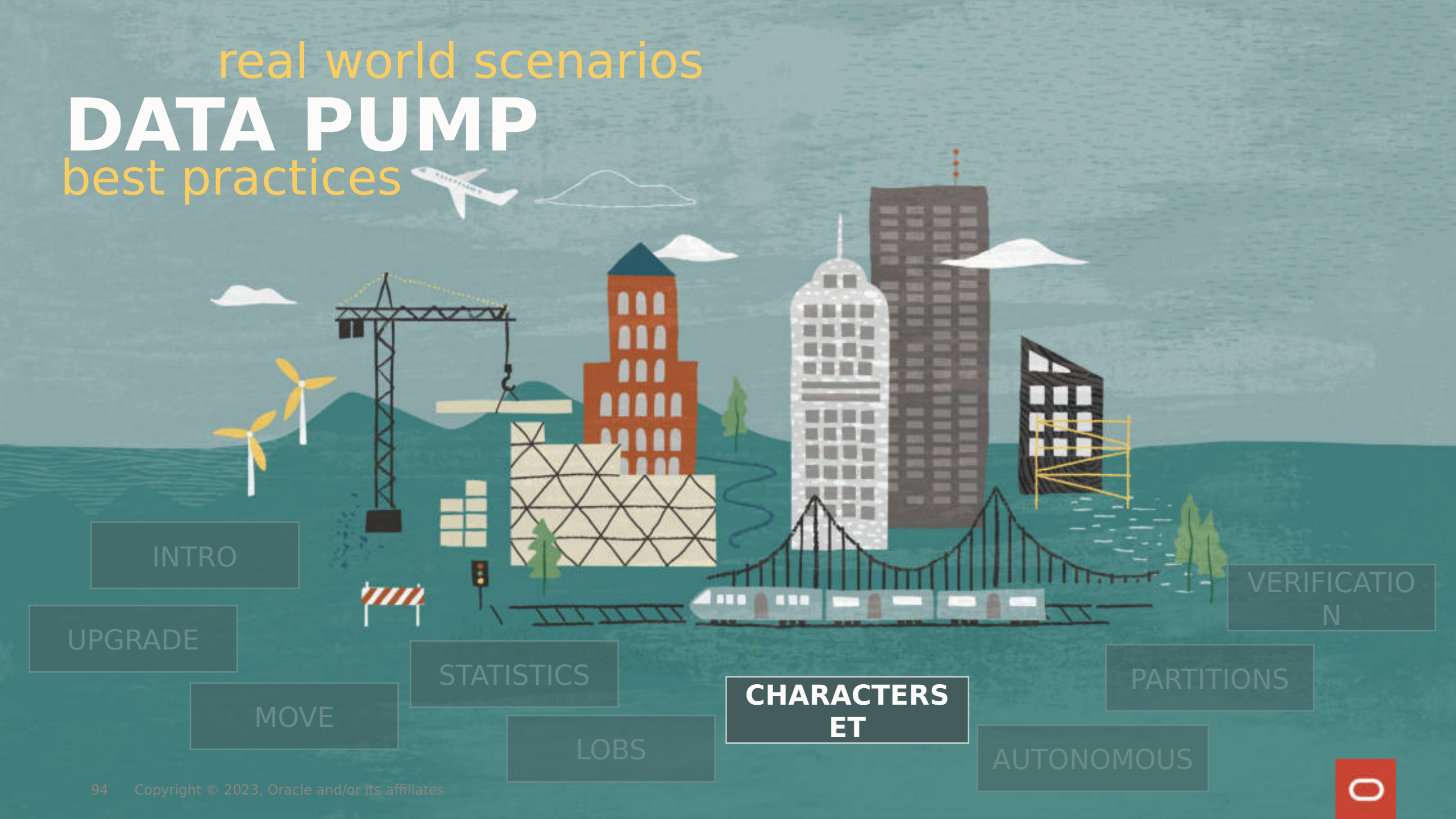
- BFILEs are always read-only

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

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERS ET

PARTITIONS

AUTONOMOUS

VERIFICATIO N

Character Sets | Brief Introduction

- Also known as a code page
- Each character is mapped to a numeric index, called a *codepoint*
- Codepoint stores character data in a computer system
- There are over hundred character sets:
  - International standards, maintained by the International Organization for Standardization (ISO)
  - Country-specific standards
  - Computer system vendor standard

Unicode is not just Unicode. It has evolved over time.

- Different encodings

Common ENCONDINGS of Unicode

	UTF-8	UTF-16	UTF-32
Bitness	8 bit	16 bit	32 bit
Width	Variable	Variable	Fixed
Minimum bytes per char	1	2	4
Maximum bytes per char	4	4	4
	Maximum US-ASCII compatibility		

Character Sets | Unicode Encoding Example

Unicode Encoding
UTF-32
UTF-16
UTF-8

Character Sets | Unicode Encoding Example

Unicode Encoding	Latin Small Letter a (U+0061)
UTF-32	0x0000006
UTF-16	0x0061
UTF-8	0x61

Character Sets | Unicode Encoding Example

Unicode Encoding	Latin Small Letter a (U+0061)	Latin Small Letter ñ (U+00F1)
UTF-32	0x0000006	0x000000F1
UTF-16	0x0061	0x00F1
UTF-8	0x61	0xC3B1

Character Sets | Unicode Encoding Example

Unicode Encoding	Latin Small Letter a (U+0061)	Latin Small Letter ñ (U+00F1)	€ Symbol (U+20AC)
UTF-32	0x0000006	0x000000F1	0x000020AC
UTF-16	0x0061	0x00F1	0x20AC
UTF-8	0x61	0xC3B1	0xE282AC

Character Sets | Superset and Subset

US7ASCII to AL32UTF8
WE8ISO8859P15 to AL32UTF8
Migrating to superset
No data loss

AL32UTF8 to WE8ISO8859P15
AL32UTF8 to US7ASCII
Migrating to subset
Potential data loss

ORA-39346:
"data loss in character set conversion for object %s"

Cause: Oracle Data Pump import converted a metadata object from the export database character set into the target database character set prior to processing the object. Some characters could not be converted to the target database character set and so the default replacement character was used.

Fix: No specific user action is required. This type of data loss can occur if the target database character set is not a superset of the export database character set.

Character Sets | Recommendations

Use AL32UTF8

- National character set AL32UTF16

Multitenant can mix different character sets

- Available since Oracle Database 12.2.0.1
- CDB$ROOT must be created with AL32UTF8
- New PDBs will be provisioned with AL32UTF8

So will my data move without issues to AL32UTF8?

It depends ...

Character Sets | Demo

Database Migration Assistant for Unicode

1
SCAN

Non-intrusive scan of the database

2
REPORT

Types of findings:

- Need no conversion
- Needs conversion
- Invalid binary representation
- Exceeds column limit
- Exceeds data type limit

3
FIX

Before migration or as part of the migration

Use Database Migration Assistant for Unicode
before you export data to a different character set

Exercise caution if your partitioning key is a CHAR or VARCHAR2 column

- Comparison using binary collation might change in which partition data is stored

Character Sets | LOBs

CLOB data is stored in a format that is compatible with UCS-2 if the database character set is multibyte

Character Sets | Demo

Character Sets | LOBs

LOB data takes up at least twice as much space

When doing a database sizing estimate, take it into consideration

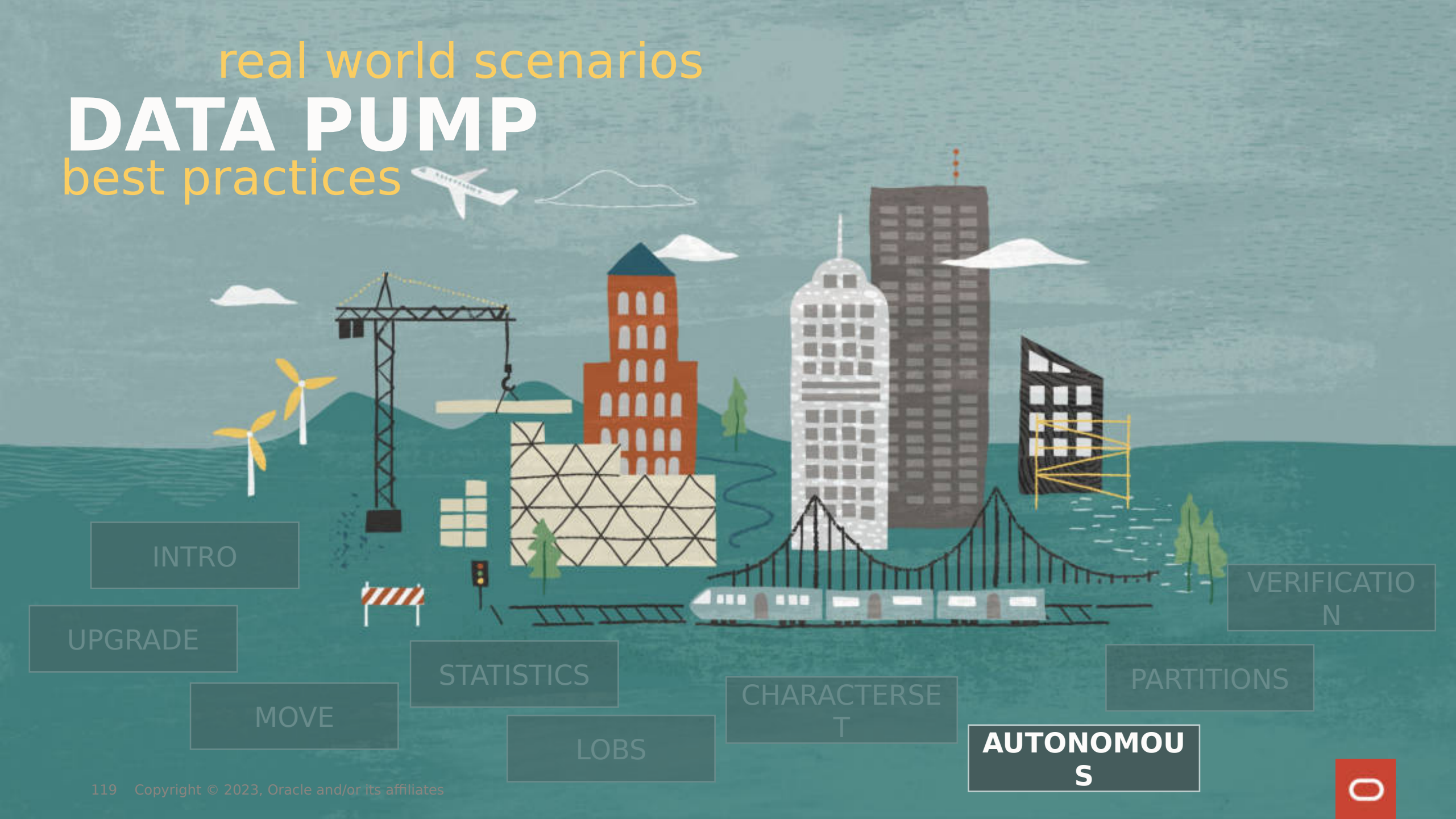
Example: CLOB uses 128 MB in WE8ISO8859P15

- Estimate 256 MB in AL32UTF8

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

PARTITIONS

VERIFICATION

AUTONOMOUS

Data Pump is the ideal tool to move to Autonomous Database

Data Pump | Migration to ADB

Data Pump is:

- Universal
- Platform independent
- Release independent

Integrated with:

- Zero Downtime Migration
- SQL Developer
- Database Migration Service

Data Pump | Migration to ADB

With Object Storage

Without Object Storage

1 Perform a Data Pump schema mode export

2 Create an Object Storage Credential on target

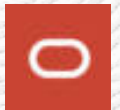
3 Copy dump files to Object Storage

4 Import to Autonomous Database

1 Perform a Data Pump schema mode export

```
expdp sh/sh@orcl \
schemas=sh \
exclude=cluster,indextype,db_link \
parallel=16 \
dumpfile=export%L.dmp \
encryption_pwd_prompt=yes
```

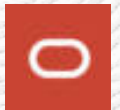
Import Data Using Oracle Data Pump on Autonomous Database



2 Create Object Storage Credential on target

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'obj_store_cred',
    username => 'adb_user@example.com',
    password => 'password'
  );
END;
/
```

Import Data Using Oracle Data Pump on Autonomous Database



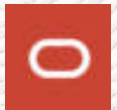
3 Copy dump files to Object Storage

Use a tool that supports Swift such as curl

```
curl -v -X PUT -u '<user>:<SWIFT token>' --upload-file <local
file location> https://objectstorage.us-ashburn-
1.oraclecloud.com/n/namespace-string/b/bucketname/o/
export1.dmp
```

Note:
curl does not support wildcards or substitution characters. Use multiple curl commands or a script that supports substitution characters

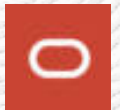
Copy Files to the Object Storage



4 Import to Autonomous Database

```
impdp admin/password@db2022adb_high \
      directory=data_pump_dir \
      credential=obj_store_cred \
      dumpfile=https://objectstorage..../bucketname/o/export%L.dmp \
      parallel=16 \
      encryption_pwd_prompt=yes
```

Import Data Using Oracle Data Pump on Autonomous Database



As of Oracle Database 21c, you can export directly into OCI Object Storage

- Eliminates the step of copying dump files

As of Oracle Database 19c, network mode imports are supported to Autonomous Database


```
impdp admin/password@db2022adb_high \
      schema=schema_name \
      network_link=<link_name> \
      parallel=n \
      transform=segment_attributes:n \
      exclude=cluster \
      nologfile=yes
```


Data Pump | Migration to ADB

With Object Storage

Without Object Storage

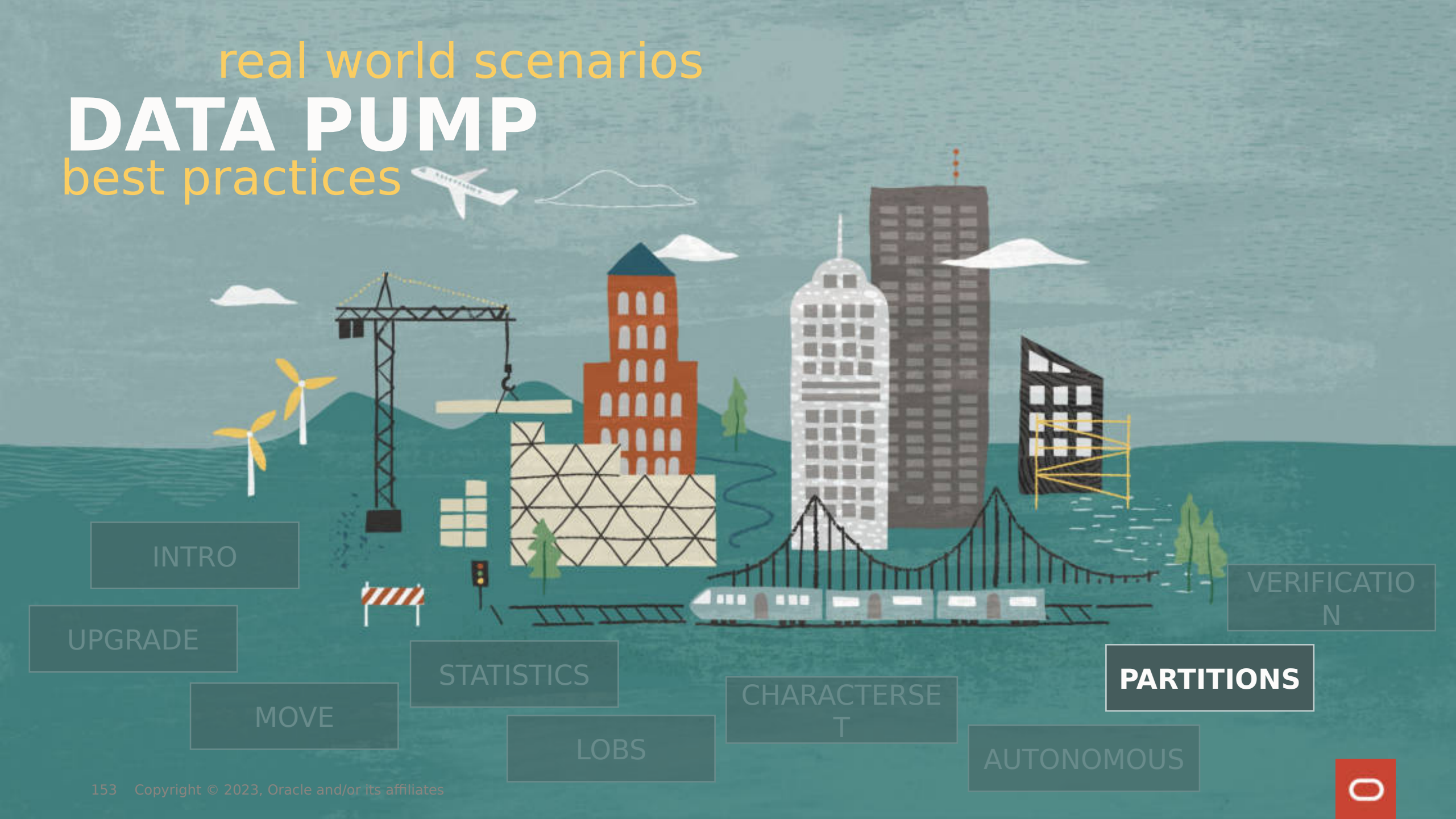
Using this cool hack you can bypass object storage when you import into Autonomous Database

- Applies to export as well

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

AUTONOMOUS

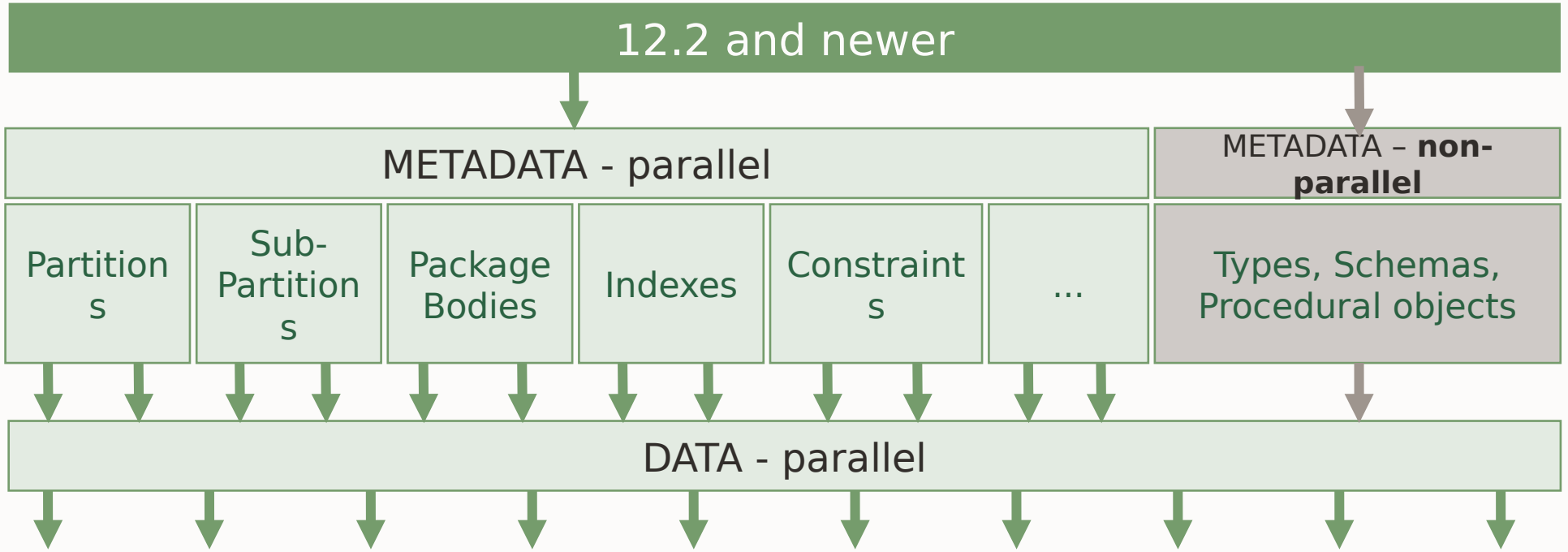
PARTITIONS

VERIFICATION

Parallel | Metadata Import - Recap

Since 12.2: Metadata import happens concurrently

- Most metadata & data objects are imported in parallel when `PARALLEL=2` or greater

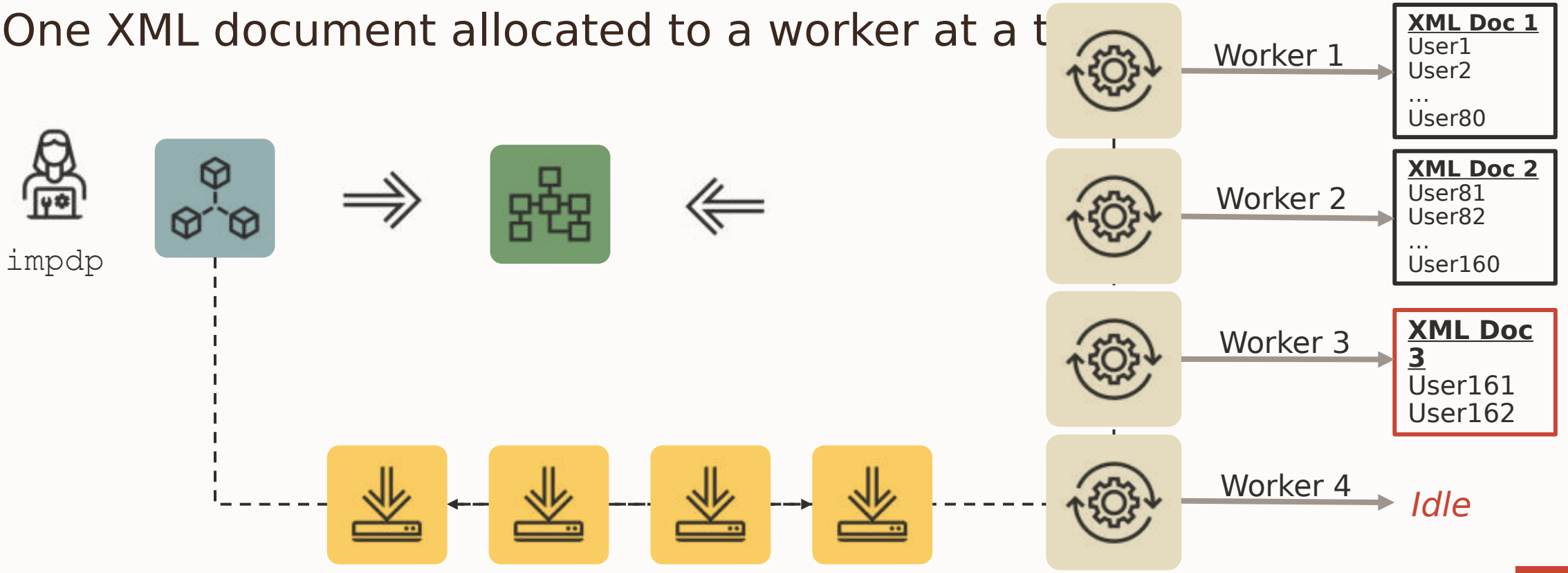


Parallel | Metadata Import - Recap

Metadata is exported in XML documents into dumpfile

- Each XML document contains N objects of a given type

One XML document allocated to a worker at a t



Limitations on parallelism

A recap

Network import does not support loading metadata in parallel

No parallelism on BasicFile LOBs

Convert *old* BasicLOBs to SecureFile LOBs

No parallelism in 19c for metadata export and import with Transportable Tablespaces

As of Oracle Database 21c, full support for parallel metadata export and import in all modes

- Applies to Transportable Tablespaces as well

Parallel Import | Existing Tables

Table
Partition
Subpartition

Known or trusted — Full parallel

Unknown — One worker only

Parallel Import | Existing Tables

Table
Partition
Subpartition

Known or trusted — Full parallel

Unknown — One worker only

`table_exists_action=truncate`

Parallel Import | Existing Tables

TRUNCATE

```
20-MAR-23 22:44:42.720: Starting "SYSTEM"."SYS_IMPORT_TABLE_01":  system/******** tables=departments_hash parallel=16
dumpfile=departments_hash%u.dmp directory=mydir metrics=y logtime=all table_exists_action=truncate logfile=imp_truncate.log
...
20-MAR-23 22:44:45.292: W-12 . . imported "SYSTEM".”DEPT":"SYS_P1352"          0 KB       0 rows in 0 seconds using automatic
20-MAR-23 22:44:54.077: W-1 . . imported "SYSTEM"."DEPT":"SYS_P1347"     413.1 MB 28246016 rows in 10 seconds using external_table
20-MAR-23 22:45:01.369: W-9 . . imported "SYSTEM".”DEPT":"SYS_P1343"     304.8 MB 18808832 rows in 7 seconds using external_table
20-MAR-23 22:45:07.833: W-8 . . imported "SYSTEM".”DEPT":"SYS_P1348"     270.0 MB 18874368 rows in 6 seconds using external_table
20-MAR-23 22:45:11.339: W-3 . . imported "SYSTEM".”DEPT":"SYS_P1345"     162.0 MB 9437184 rows in 4 seconds using external_table
20-MAR-23 22:45:15.468: W-6 . . imported "SYSTEM".”DEPT":"SYS_P1351"     153.0 MB 9437184 rows in 4 seconds using external_table
…
20-MAR-23 22:45:33.856: W-5      Completed 16 TABLE_EXPORT/TABLE/TABLE_DATA objects in 49 seconds
20-MAR-23 22:45:34.058: Job "SYSTEM"."SYS_IMPORT_TABLE_01" successfully completed at Mon Mar 20 22:45:34 2023 elapsed 0 00:00:52
```

From the timestamps we can see that each partition is imported serially

Parallel Import | Existing Tables

Table
Partition
Subpartition

Known or trusted
`table_exists_action=replace`

Full parallel

Unknown

One worker only

Parallel Import | Existing Tables

REPLACE

```
20-MAR-23 22:45:57.265: Starting "SYSTEM"."SYS_IMPORT_TABLE_01":  system/******** tables=departments_hash parallel=16
dumpfile=departments_hash%u.dmp directory=mydir metrics=y logtime=all table_exists_action=replace logfile=imp_replace.log
...
20-MAR-23 22:46:00.140: W-14 . . imported "SYSTEM"."DEPT":"SYS_P1352"         0 KB       0 rows in 0 seconds using automatic
20-MAR-23 22:46:06.293: W-3 . . imported "SYSTEM"."DEPT":"SYS_P1344"     135.0 MB 9437184 rows in 7 seconds using direct_path
20-MAR-23 22:46:06.309: W-2 . . imported "SYSTEM"."DEPT":"SYS_P1345"     162.0 MB 9437184 rows in 7 seconds using direct_path
20-MAR-23 22:46:06.311: W-11 . . imported "SYSTEM"."DEPT":"SYS_P1349"     135.0 MB 9437184 rows in 7 seconds using direct_path
20-MAR-23 22:46:06.811: W-5 . . imported "SYSTEM"."DEPT":"SYS_P1355"     117.0 MB 9437184 rows in 7 seconds using direct_path
20-MAR-23 22:46:06.818: W-8 . . imported "SYSTEM"."DEPT":"SYS_P1351"     153.0 MB 9437184 rows in 7 seconds using direct_path
...
20-MAR-23 22:46:11.107: W-6      Completed 16 TABLE_EXPORT/TABLE/TABLE_DATA objects in 12 seconds
20-MAR-23 22:46:11.292: Job "SYSTEM"."SYS_IMPORT_TABLE_01" successfully completed at Mon Mar 20 22:46:11 2023 elapsed 0 00:00:15
```

- This time we see partitions imported in parallel
- But, what if you don’t want Data Pump to create the table?

Parallel Import | Existing Tables

Table
Partition
Subpartition

Known or trusted — Full parallel

```
table_exists_action=truncate
data_options=trust_existing_table_partitions
```

Unknown — One worker only

Parallel Import | Existing Tables

TRUST_EXISTING_TABLE_PARTITIONS (1)

```
20-MAR-23 22:46:41.572: Starting "SYSTEM"."SYS_IMPORT_TABLE_01":  system/******** tables=departments_hash parallel=16
dumpfile=departments_hash%u.dmp directory=mydir metrics=y logtime=all table_exists_action=truncate logfile=imp_trust.log
data_options=TRUST_EXISTING_TABLE_PARTITIONS
...
20-MAR-23 22:46:45.354: W-13 . . imported "SYSTEM"."DEPT":"SYS_P1352"         0 KB       0 rows in 0 seconds using automatic
20-MAR-23 22:46:55.085: W-10 . . imported "SYSTEM"."DEPT":"SYS_P1345"     162.0 MB 9437184 rows in 10 seconds using external_table
20-MAR-23 22:46:55.437: W-3 . . imported "SYSTEM"."DEPT":"SYS_P1344"     135.0 MB 9437184 rows in 11 seconds using external_table
20-MAR-23 22:46:55.439: W-11 . . imported "SYSTEM"."DEPT":"SYS_P1349"     135.0 MB 9437184 rows in 11 seconds using external_table
20-MAR-23 22:46:55.676: W-5 . . imported "SYSTEM"."DEPT":"SYS_P1351"     153.0 MB 9437184 rows in 11 seconds using external_table
20-MAR-23 22:46:55.820: W-4 . . imported "SYSTEM"."DEPT":"SYS_P1354"     144.0 MB 9437184 rows in 11 seconds using external_table
...
20-MAR-23 22:46:11.107: W-6      Completed 16 TABLE_EXPORT/TABLE/TABLE_DATA objects in 12 seconds
20-MAR-23 22:46:11.292: Job "SYSTEM"."SYS_IMPORT_TABLE_01" successfully completed at Mon Mar 20 22:46:11 2023 elapsed 0 00:00:15
```

Again we see partitions imported in parallel, but the existing table was re-used

Parallel Import | Existing Tables

TRUST_EXISTING_TABLE_PARTITIONS (2)

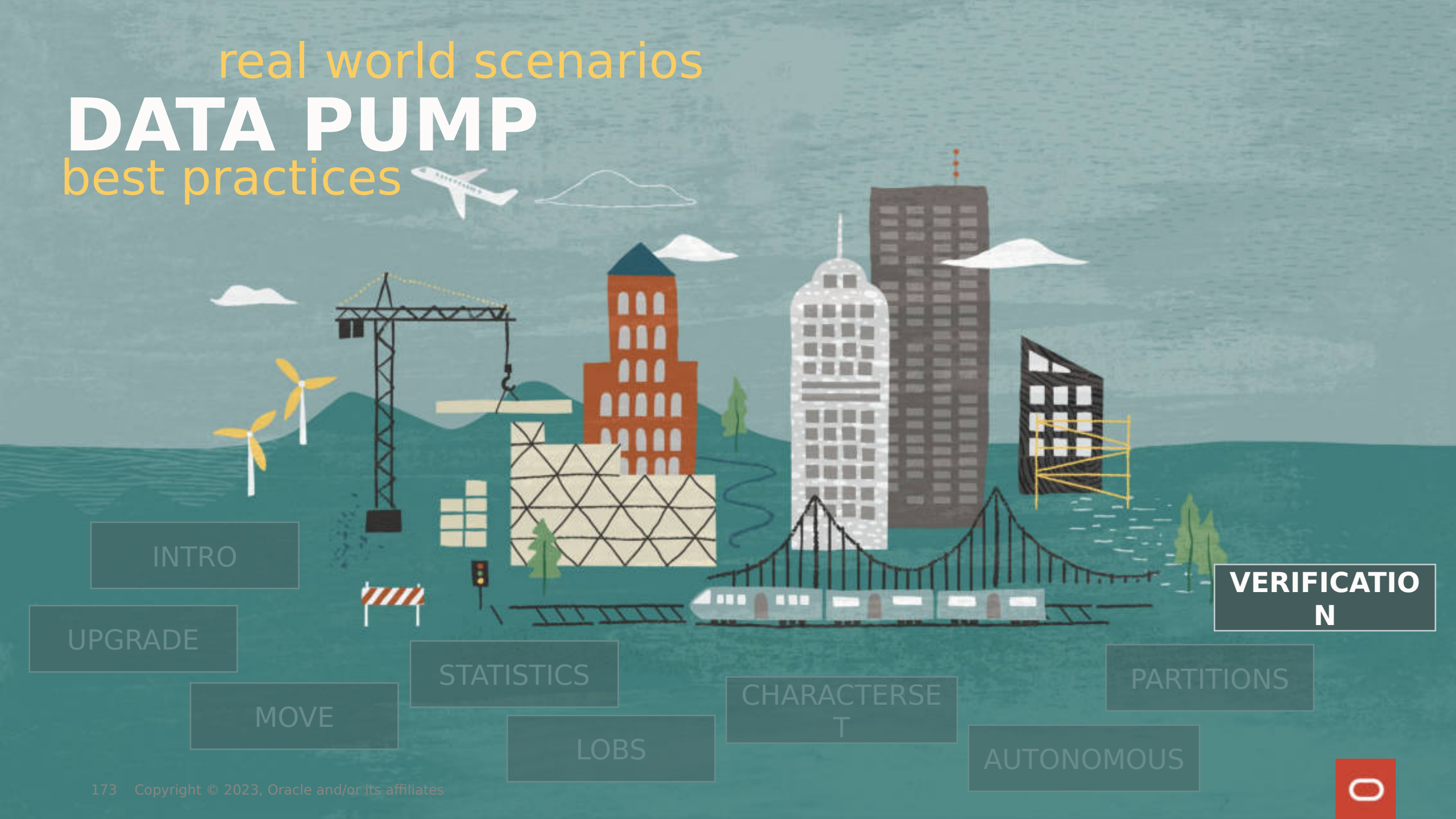
```
20-MAR-23 23:18:34.980: Starting "SYSTEM"."SYS_IMPORT_TABLE_01":  system/******** tables=departments_hash parallel=16
dumpfile=departments_hash%u.dmp directory=mydir metrics=y logtime=all table_exists_action=truncate logfile=imp_trust.log
data_options=TRUST_EXISTING_TABLE_PARTITIONS
...
20-MAR-23 23:18:39.225: ORA-31693: Table data object "SYSTEM"."DEPARTMENTS_HASH":"SYS_P1345" failed to load/unload and is being
skipped due to error:
ORA-02149: Specified partition does not exist
20-MAR-23 23:18:39.236: ORA-31693: Table data object "SYSTEM"."DEPARTMENTS_HASH":"SYS_P1347" failed to load/unload and is being
skipped due to error:
ORA-02149: Specified partition does not exist
...
```

The difference in partitioning scheme is detected and data is not imported!

real world scenarios

DATA PUMP

best practices



INTRO

UPGRADE

MOVE

STATISTICS

LOBS

CHARACTERSET

AUTONOMOUS

PARTITIONS

VERIFICATION

Data Pump Error Messages

Are there error messages we can ignore?

Was this job successful?

```
ORA-31634: job already exists
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 79
ORA-06512: at "SYS.KUPV$FT", line 1159
ORA-31637: cannot create job Q1_AGG_TRAF_ROAMER_0 for user KGRONAU
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 95
ORA-06512: at "SYS.KUPV$FT", line 1152
```

What about this one?

```
ORA-31626: job does not exist
ORA-31633: unable to create master table "KGRONAU.Q2_AGG_IND_MO_27"
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 95
ORA-06512: at "SYS.KUPV$FT", line 1163
ORA-00955: name is already used by an existing object
ORA-06512: at "SYS.KUPV$FT", line 1056
ORA-06512: at "SYS.KUPV$FT", line 1044
```

Or this one?

```
ORA-31693: Table data object "APPLSYS"."FND_LOG_MESSAGES" failed
           to load/unload and is being skipped due to error:
ORA-02354: error in exporting/importing data
ORA-01555: snapshot too old: rollback segment number 25
           with name "_SYSSMU25_1608416701$" too small
```

Data Pump finished successfully

But can you trust the expdp/impdp log files?

Log file example | Fatal error

```
ORA-39126: Worker unexpected fatal error in
KUPW$WORKER.LOCATE_DATA_FILTERS
[TABLE_DATA:“KGRONAU"."REACT":"REACT_20220501_1"]
SELECT COUNT(*) FROM DUAL WHERE :1  IN ('REACT_20220605_2758',
'REACT_20220605_2757','REACT_20220605_2756','REACT_20220605_2633',
...
ORA-06512: at "SYS.KUPW$WORKER", line 6415
----- PL/SQL Call Stack -----
  object      line  object
  handle    number  name
5e4ffba30     15370  package body SYS.KUPW$WORKER
…
…
Job "KGRONAU"."Q2_FACT_MOV_SRV_RE_175" stopped due to fatal error at
06:27:31
```

Log file example | More errors

```
…
ORA-31684: Object type USER:"KGRONAU" already exists
…
ORA-39111: Dependent object type
ALTER_FUNCTION:"KGRONAU"."GETDDL_F$" skipped, base object type
FUNCTION:"KGRONAU"."TPGETDDL_F$" already exists
…
ORA-39082: Object type VIEW:"KGRONAU"."MyCaseSensitiveView" created with compilation
warnings
Processing object type SCHEMA_EXPORT/PACKAGE/PACKAGE_BODY
ORA-39346: data loss in character set conversion for object
...
ORA-01653: unable to extend table KGRONAU.MYTABLE by 8192 in tablespace KGRONAU
ORA-39171: Job is experiencing a resumable wait.
...
ORA-12899: value too large for column COD_PAIS_A2 (actual: 3, maximum: 2)
...
ORA-39083: Object type REF_CONSTRAINT:“KGRONAU"."R_CALCEVIK" failed to create with error:
ORA-02298: cannot validate (SCDAT.R_GLTRANS_CALCEVIK) - parent keys not found

Job "KGRONAU"."EDU12_SCHEMA" completed with 42 error(s)
```

And even when it completed successfully ...

Log file example | Successful completion

```
Job "KGRONAU"."Q2_AGG_MYTMN_SENTM_10"
successfully completed at Thu Aug 11 00:23:34 2022 elapsed 0 00:03:59
```

Are you sure? Really??

Log file example | Look closer ...

```
...
W-1      Completed 1 TABLE objects in 17 seconds
W-1      Completed by worker 1 1 TABLE objects in 17 seconds
W-1 Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
…
W-1 . . imported "KGRONAU"."AGG_MYTMN_SENTMSG_D1":"AGG_MYTMN_SENTMSG_D1_20071024"  13.34 KB
   231 rows in 2 seconds using external_table
W-1 . . imported "KGRONAU"."AGG_MYTMN_SENTMSG_D1":"AGG_MYTMN_SENTMSG_D1_20071029"  13.34 KB
     0 out of 231 rows in 0 seconds using external_table
W-1 . . imported "KGRONAU"."AGG_MYTMN_SENTMSG_D1":"AGG_MYTMN_SENTMSG_D1_20071008"  13.26 KB
     0 out of 228 rows in 0 seconds using external_table
...
W-1 Processing object type TABLE_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
W-1      Completed 30 OBJECT_GRANT objects in 0 seconds
W-1      Completed by worker 1 30 OBJECT_GRANT objects in 0 seconds
W-1      Completed 98 TABLE_EXPORT/TABLE/TABLE_DATA objects in 2 seconds
…
Job "KGRONAU"."Q2_AGG_MYTMN_SENTM_10" successfully completed at Thu Aug 11 00:23:34 2022
elapsed 0 00:03:59
```

Challenges

How do you validate expdp/impdp results?

Comparison possibilities

Comparison possibilities

Rows Objects Content

Comparison| Row counts

```
cat MyImpDp.dplog  | grep -w imported | grep -w rows | awk '{print $5,$8}' >myfile


…
"KGRONAU"."AGG_MYTMN_SENTMSG_D1":"AGG_MYTMN_SENTMSG_D1_20071024" 231
"KGRONAU"."AGG_MYTMN_SENTMSG_D1":"AGG_MYTMN_SENTMSG_D1_20071029" 0
…
```

Comparison| Row counts

```
Export Datapump:
cat expdp.dplog |grep -w exported | grep -w rows |awk ...  >>rowcount.txt
cat rowcount.txt |sort -k 1 >>rowcount_src.txt

Import Datapump:
cat impdp.dplog |grep -w imported | grep -w rows |awk ...  >>rowcount.txt
cat rowcount.txt |sort -k 1 >>rowcount_trg.txt

Differences:
diff rowcount_src.txt rowcount_trg.txt
```

How can you validate the amount of rows in a GoldenGate based scenario?

- Important for ZDM Logical Migrations too

Comparison| Row counts

```
SQL> select 'SELECT /*+ PARALLEL(16) */ '|| chr(39)||owner ||'.'||table_name ||'
number of rows: '|| chr(39) || '|| count(1) from 'as rowcount_stmts from
dba_tables where owner='KGRONAU';
||owner ||'."'||table_name||'";'

ROWCOUNT_STMTS
--------------------------------------------------------------------
SELECT /*+ PARALLEL(16) */ 'KGRONAU.DUMMY number of rows: '|| count(1) from
KGRONAU."DUMMY";

SELECT /*+ PARALLEL(16) */ 'KGRONAU.HASH_TEST number of rows: '|| count(1) from
KGRONAU."HASH_TEST";
```

Comparison| Row counts

```
SQL> set heading off
SQL> spool rowcount_src.log
SQL> SELECT /*+ PARALLEL(16) */ 'KGRONAU.DUMMY number of rows: '|| count(1) from
KGRONAU."DUMMY";

KGRONAU.DUMMY number of rows: 6

SQL> SELECT /*+ PARALLEL(16) */ 'KGRONAU.HASH_TEST number of rows: '|| count(1)
from KGRONAU."HASH_TEST";

KGRONAU.HASH_TEST number of rows: 5
```

Comparison possibilities

Rows **Objects** Content

How can you validate objects?

Recompile invalid objects

- `@?/rdbms/admin/utlrp`
- `@?/rdbms/admin/utlprp` *n*

Comparison| Object validation with MINUS query

```
select owner c1, object_type c3, object_name c2 from
dba_objects where status != 'VALID’

minus

select owner c1, object_type c3, object_name c2 from
dba_objects@sourcedb where status != 'VALID';
```

Comparison| Object validation with subquery

```
select 'alter VIEW '||co||'.'||cn||' compile; '
                    ||CHR(13)||chr(10)||
       'select line, text from dba_errors where owner=
                  '||''''||co||''''||' and name='||''''||
cn||''''||'
       order by line;'
from (
      select owner co, object_type ct, object_name cn from
      dba_objects where status = 'INVALID' and object_type='VIEW'
      minus
      select owner co, object_type ct, object_name cn from
      dba_objects@sourcedb where status = 'INVALID'  and
      object_type='VIEW' ) ;
```

Comparison| Object validation with a view

```
alter VIEW KGRONAU.MIRROR compile;

select line, text from dba_errors where owner= 'KGRONAU'
and name='MIRROR' order by line;
```

Comparison| Objects - constraints

```
select table_name,count(table_name) from dba_constraints@sourcedb

   where owner='KGRONAU'
   and constraint_name not like 'BIN%'
   group by table_name
minus
select table_name,count(table_name) from dba_constraints
   where owner='KGRONAU'
   and constraint_name not like 'BIN%'
   group by table_name;


TABLE_NAME                     COUNT(TABLE_NAME)
------------------------------ -----------------
MYTABLE                                        1
```

Comparison possibilities

Rows

Objects

Content

How can you guarantee data on source matches data on target exactly?

Comparison| Data validation

DBMS_COMPARISON

```
    DBMS_COMPARISON.CREATE_COMPARISON

    DBMS_COMPARISON.COMPARE

   (DBMS_COMPARISON.CONVERGE)
```

Data validation | Demo

Data validation | Oracle GoldenGate Veridata

GoldenGate Veridata

https://www.oracle.com/at/a/ocom/docs/middleware/veridata-datasheet.pdf

Comparison| Alternative options

DBMS_CRYPTO package

STANDARD_HASH function

Data Validation | DMBS_CRYPTO Package

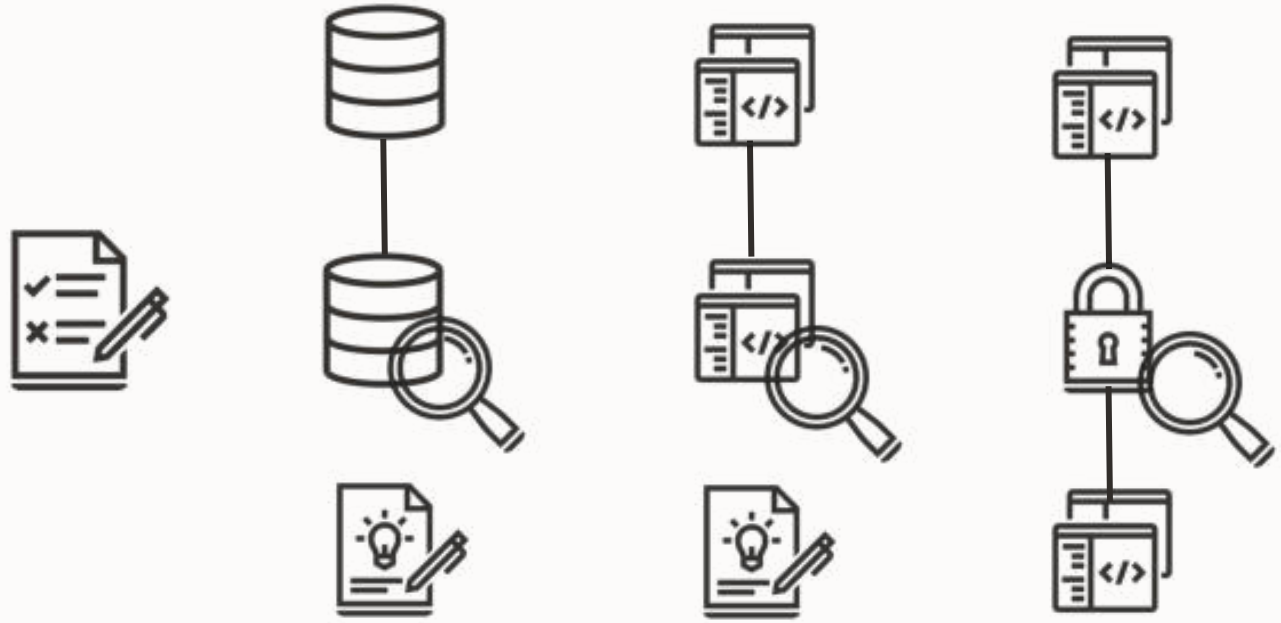
```
select col1,col2, rawtohex(DBMS_CRYPTO.Hash (UTL_I18N.STRING_TO_RAW
(col2, 'AL32UTF8'), 2)) as hash2 from HASH_TEST;

      COL1 COL2                 HASH2
---------- -------------------- --------------------------------
         0 Hello world!!        1D94DD7DFD050410185A535B9575E184
         1 Hello world!         86FB269D190D2C85F6E0468CECA42A20
         2 Hello world!!        1D94DD7DFD050410185A535B9575E184
         3 Hello world!!        1D94DD7DFD050410185A535B9575E184
         4 Hello world!!        1D94DD7DFD050410185A535B9575E184
```

Data Validation | STANDARD_HASH Function

```
select col1, col2 ,STANDARD_HASH (col2 , 'MD5' ) HASH2 FROM
kgronau.hash_test;
      COL1 COL2                 HASH2
---------- -------------------- --------------------------------
         0 Hello world!!        1D94DD7DFD050410185A535B9575E184
         1 Hello world!         86FB269D190D2C85F6E0468CECA42A20
         2 Hello world!!        1D94DD7DFD050410185A535B9575E184
         3 Hello world!!        1D94DD7DFD050410185A535B9575E184
         4 Hello world!!        1D94DD7DFD050410185A535B9575E184
```

Summary | Comparison and Validation



Validation is time consuming

Often it can’t be done for all rows in all tables

Photo by Zbynek Burival on Unsplash

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

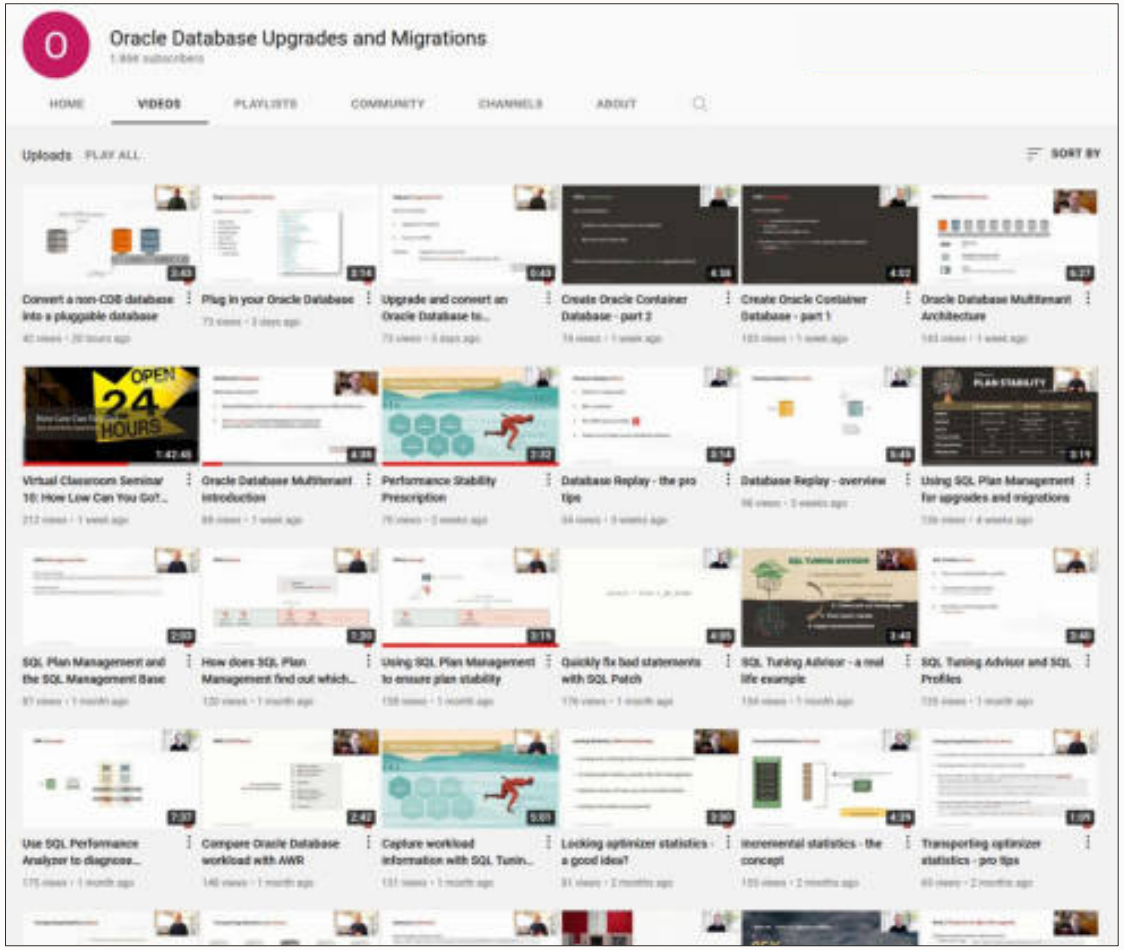
115 minutes – Apr 8, 2021

Recorded Web Seminars

https://MikeDietrichDE.com/videos

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

YouTube | Oracle Database Upgrades and Migrations



Link

- 200+ videos
- New videos every week
- No marketing
- No buzzword
- 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:

OracleDatabaseUpgradesandMigrations

THANK YOU

Release and Patching Strategy
for Oracle Database 23c

May 10, 2023 – 16:00h CET

THANK YOU