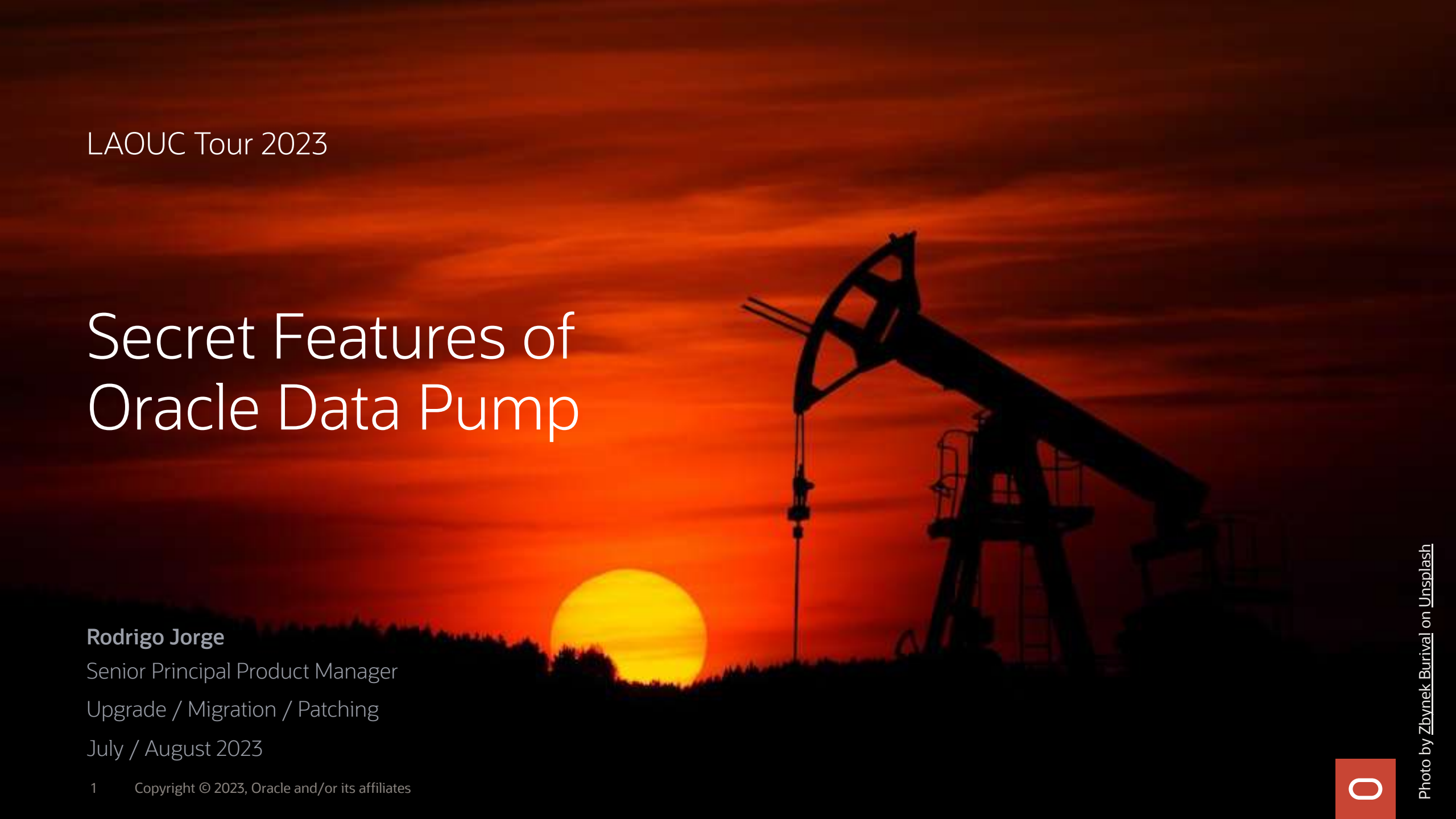
LAOUC Tour 2023

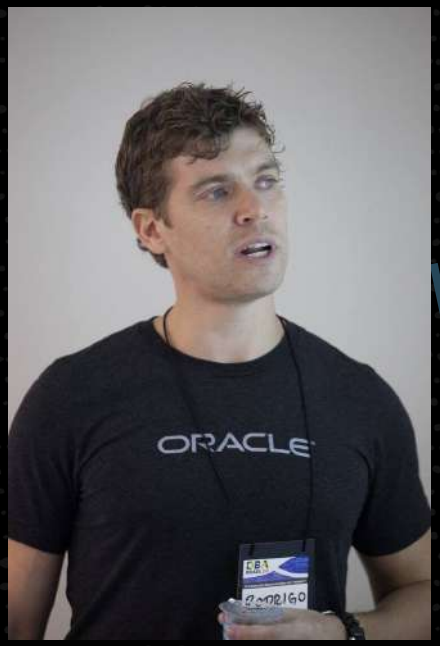
Secret Features of Oracle Data Pump



Rodrigo Jorge
Senior Principal Product Manager
Upgrade / Migration / Patching

July / August 2023

Photo by Zbynek Burival on Unsplash



- OCMs 11g / 12c / MAA / Cloud
- OCEs 11g / 12c
- (…)

www.dbarj.com.br
@rodrigojorgedba
/rodrigoaraujorge

Oracle ACE Program

500+ technical experts & community leaders helping peers globally

The **Oracle ACE Program** recognizes & rewards individuals for their technical & community contributions to the Oracle community

3 membership tiers

Oracle ACE Director | Oracle ACE Pro | Oracle ACE Associate

Learn more - ace.oracle.com

Oracle ACE

Nominate
yourself or a candidate:

ace.oracle.com/nominate

Safe Harbor Statement

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

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://dbarj.com.br/webinars

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

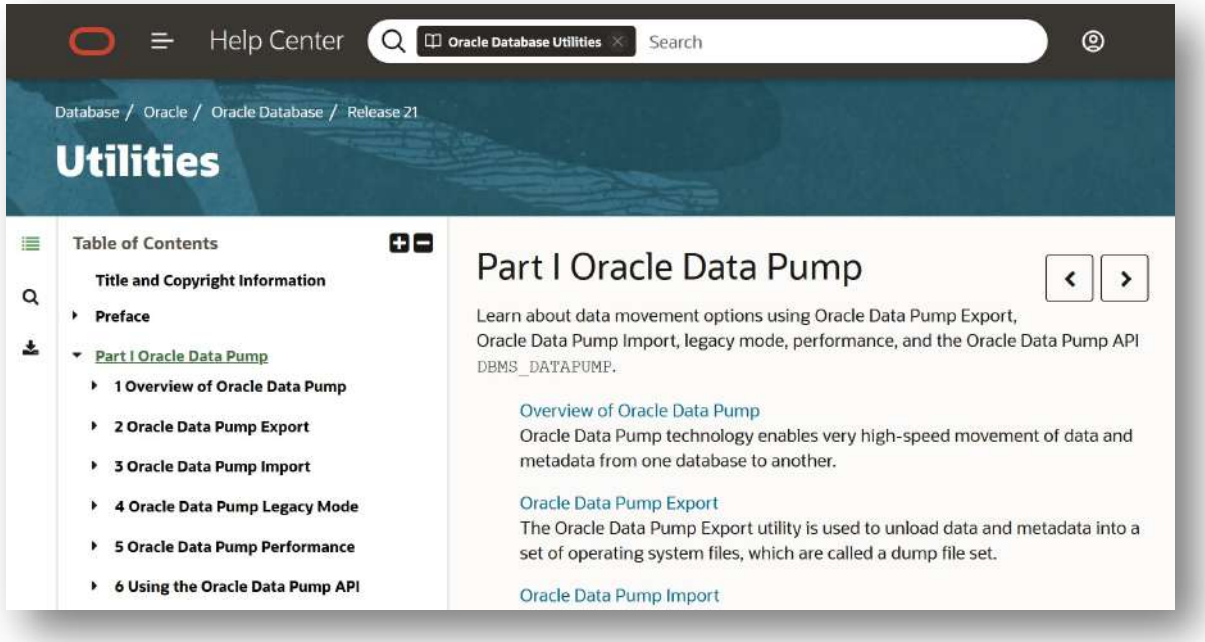
Quiz 1

When was Data Pump first released?

1. 8i
2. 9i
3. 10g
4. 11g
5. What is Data Pump?

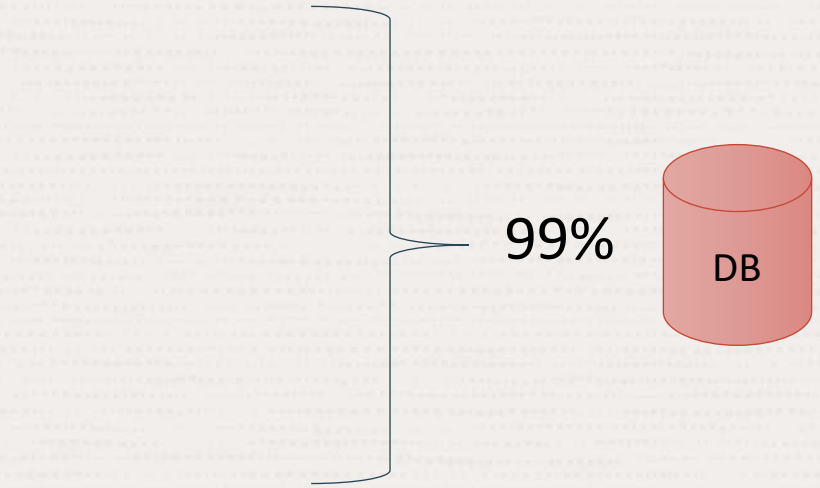
Data Pump | Documentation

- Oracle Database 19c – Utilities Guide
- Oracle Database 21c – Utilities Guide



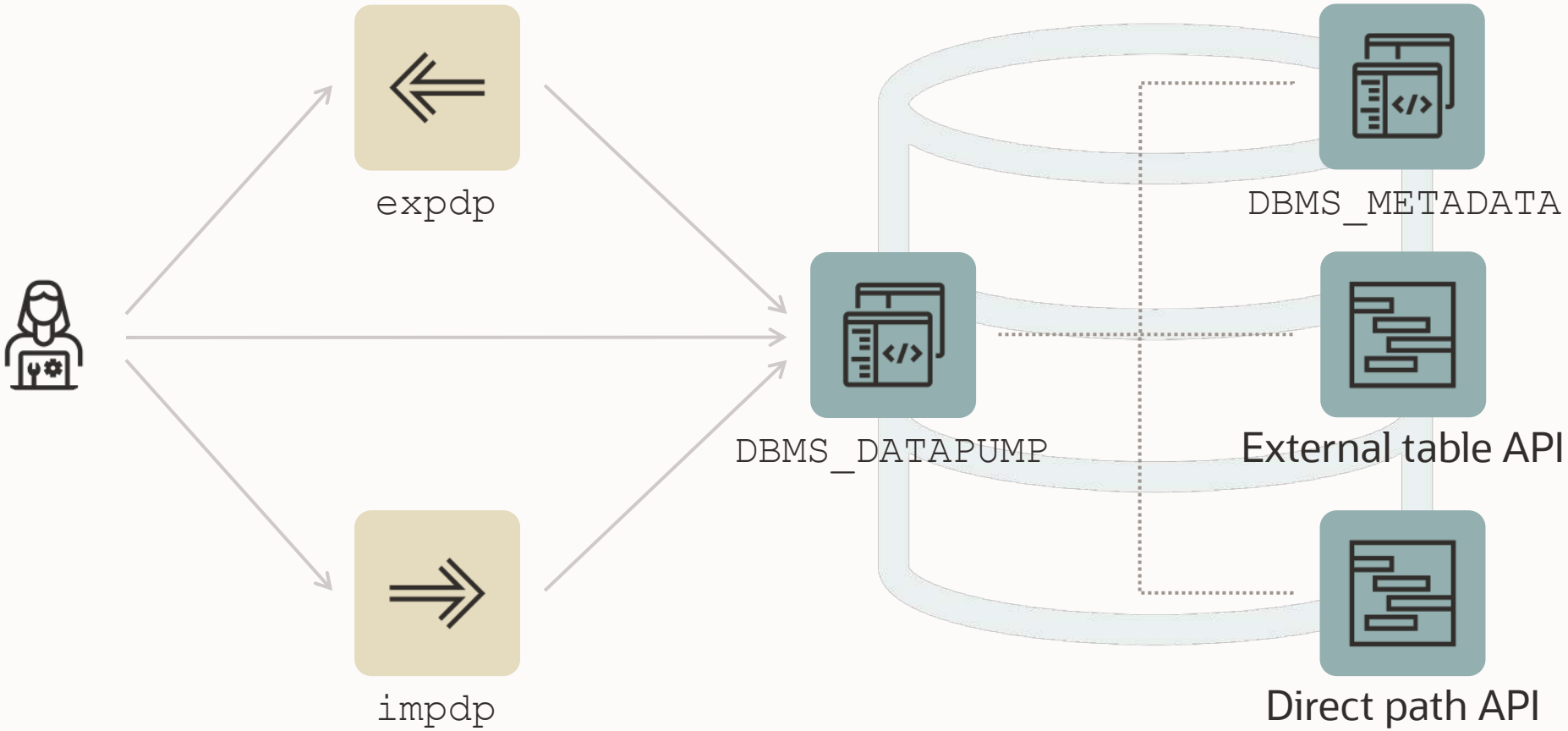
Where is Datapump code?

- DBMS_DATAPUMP
- DBMS_METADATA
- expdp / impdp binaries



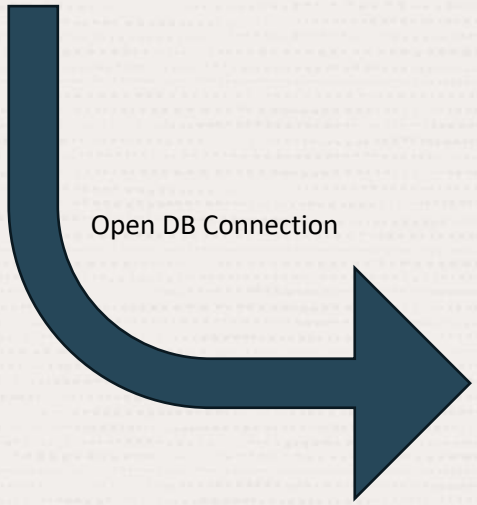
expdp / impdp are shortcuts!

Data Pump | Architecture



expdp

```
$ expdp scott/tiger@orcl directory=exp_schema
dumpfile=scott.dmp logfile=scott.log schemas=scott
```



```
BEGIN
  DBMS_METADATA.XXX (
  ...
);
END;
```


DBMS_DATAPUMP | API

The Data Pump API (`DBMS_DATAPUMP`) is used many places:

- Zero Downtime Migration
- Enterprise Manager
- SQL Developer
- SQLcl

...

You can use it as well,
it is documented and supported

DBMS_DATAPUMP | API

Ideas:

- Use Data Pump functionality without installing a client
- Schedule export or imports using `DBMS_SCHEDULER`
- Dynamically build Data Pump jobs
- Integrate into automation tools (Ansible, Puppet)
- Accessible via ORDS / REST API as well
- Rename schema using a loopback database link
- Take a snapshot of a schema during application development

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
h1 := DBMS_DATAPUMP.OPEN(
        operation => 'EXPORT',
        job_mode => 'SCHEMA',
        remote_link => null,
        job_name => 'MY_JOB',
        version => null);

-- Create a Data Pump job to do a schema
-- export. Give it a meaningful name
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.METADATA_FILTER(
   handle => h1,
   name => 'SCHEMA_EXPR',
   value => 'IN (''APP'')');

-- Specify the schema to be exported. We let
-- the object_path parameter default in this
-- call, so this applies to all objects in
-- the job
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.ADD_FILE(
   handle => h1,
   filename => 'exp%u.dmp',
   directory => 'MYDIR',
   filetype=>DBMS_DATAPUMP.KU$_FILE_TYPE_DUMP_FILE);

-- Specify the dumpfile for the job using a
-- wildcard. The directory object must be
-- supplied for each file added to the job
-- FILETYPE defaults to dumpfile but we
-- specify it anyway to be clear
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.ADD_FILE(
   handle => h1,
   filename => 'exp.log',
   directory => 'MYDIR',
   filetype=>DBMS_DATAPUMP.KU$_FILE_TYPE_LOG_FILE);

-- Specify the log file for the job. The directory
-- object must be supplied for each file added to
-- the job.
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.SET_PARALLEL(
   handle => h1,
   degree => 4 );

-- Set the parallelism for the job
-- Or get a little creative

select value into parallel_degree
from   v$parameter
where  name='cpu_count';
DBMS_DATAPUMP.SET_PARALLEL(
   handle => h1,
   degree => parallel_degree);
```

DBMS_DATAPUMP | Comparison

Client

```
expdp directory=mydir \
   logfile=exp.log \
   dumpfile=exp%u.dmp \
   schemas=app \
   parallel=4 \
   metrics=y \
   logtime=all
```

API

```
DBMS_DATAPUMP.SET_PARAMETER(
   handle => h1,
   name => 'METRICS',
   value => 1);

DBMS_DATAPUMP.SET_PARAMETER(
   handle => h1,
   name => 'LOGTIME',
   value => 'ALL');

-- set other job parameters
```

DBMS_DATAPUMP | Comparison

Client

API

```
DBMS_DATAPUMP.START_JOB (
   handle => h1);

-- now start the job
-- wait for it to complete

DBMS_DATAPUMP.WAIT_FOR_JOB (
   handle => h1,
   job_state);
```

Use 10046 trace to generate DBMS_DATAPUMP calls

Data Pump | Generate PL/SQL

1. Enable SQL trace on a test database

```
SQL> alter system
     set event='10046 trace name context forever, level 4';
```

2. Execute your Data Pump command

```
$ impdp system ... parfile=import.par
```

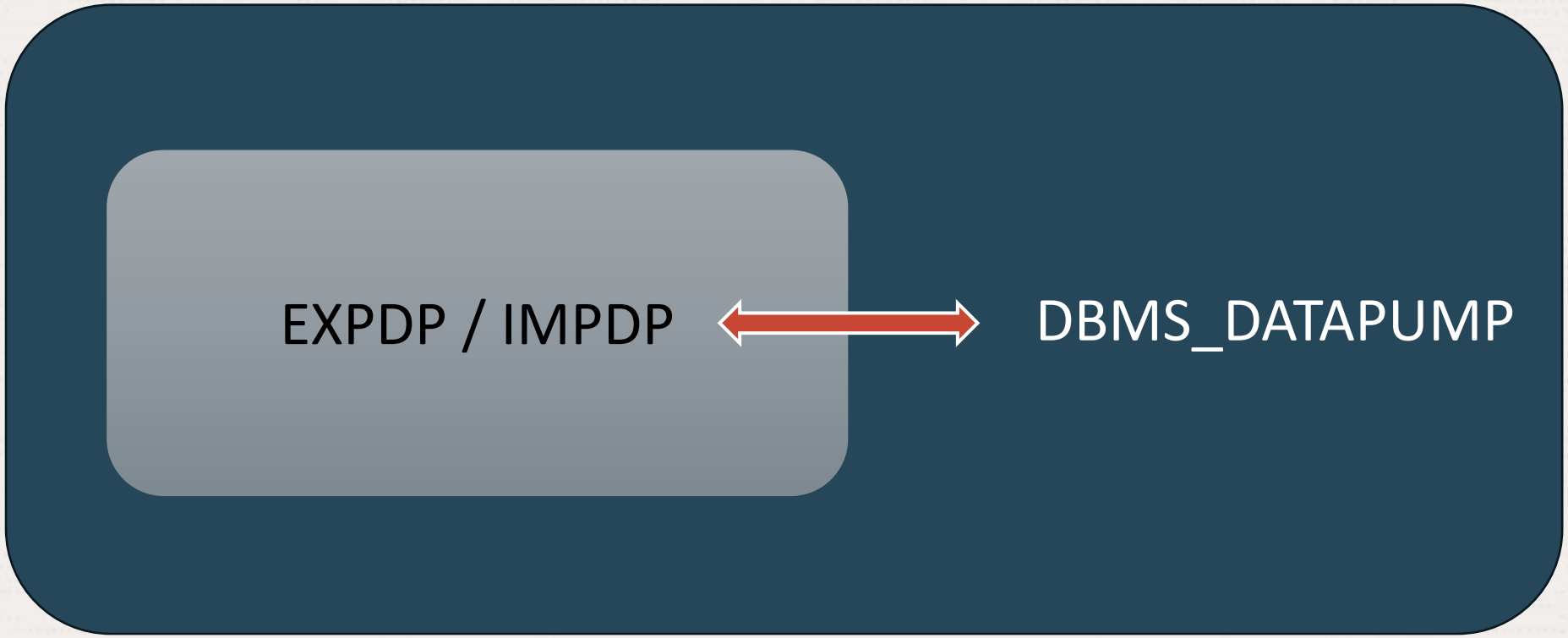
3. Examine the trace file

```
$ vi ORCL_ora_12345.trc
```

Pro tip: Grep for *DBMS_DATAPUMP* to find the right trace file

The documentation has many good examples on using `DBMS_DATAPUMP`

expdp/impdp vs DBMS_DATAPUMP



Apply the Data Pump Bundle Patch

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

Data Pump Bundle Patch

Fewer Bugs

Important patches are included.

Monitor for bugs that affect many customers.

Faster Patching

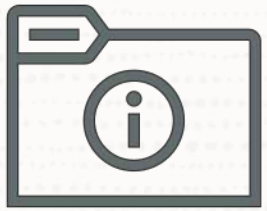
The bundle patch changes the way Data Pump is patched. Subsequent patches apply faster.

125 fixes

Data Pump Bundle Patch for 19.19.0

MOS Note: 2819284.1

Bug	Description
Bug 28318139	ORA-31003 ERROR WHEN IMPORTING FULL DATABASE IN PARALLEL
Bug 28357349	SCHEMA LEVEL EXPORT/IMPORT CHANGES VIRTUAL COLUMN DEFINITION
Bug 28555193	DBMS_METADATA.GET_DDL CAPTURE INCORRECT STORAGE OPTIONS OF THE XML COLUMN ON GTT
Bug 28771564	DATAPUMP EXPORT INVOKED BY A PRIVILEGE USER EXECUTES A QUERY FOR V$OPEN_CURSOR
Bug 28990738	12.2 DBMS_METADATA.GET_DDL IS SLOW DUE TO SLOW ACCESS ON DICTIONARY VIEWS
Bug 29276889	ATP-D: DATA PUMP IMPORT FROM ATP-D INSTANCE TO A LOCAL DB INSTANCE FAILS
Bug 29543605	18.4 ADWC - ORA-39242: UNABLE TO EXPORT/IMPORT "LONG RAW" DATA TYPE
Bug 29613245	ORA-31684 ORA-39112 WITH FIX 28539085 AND VERSION=11.2
Bug 29959025	EXPDP RUNNING LONG TIME QUERYING KU$_SUBPARTITION_EST_VIEW WHEN PROCESSING TABLE_DATA
Bug 30155338	POSSIBLE DEADLOCK/TIMEOUT ERRORS DURING PARALLEL IMPORT WITH TABLE_EXISTS_ACTION=REPLACE
Bug 30157766	ORA-21560 DBMS_METADATA.FETCH_DDL IN 19C NOT IN 12.2
Bug 30430932	DBMS_METADATA NOT DISPLAYING THE SEMICOLON AND SLASH FOR TYPE SPECIFICATIONS
Bug 30582819	REMAP TABLESPACE IS NOT CONSIDERED FOR LOCAL TEMPORARY TABLESPACE DURING IMPDP
Bug 30662417	IMPDP WORKER TERMINATED WITH ORA-39029 AFTER MULTIPLE ORA-01775
Bug 30763851	IMPDP 11.2 TO 18C OR HIGHER HITS ORA-904 WHEN TABLES HAVE EXTENDED STATISTICS
Bug 30822078	IMPDP VERY SLOW DUE TO PROCESS REORDERING
Bug 30858671	18C DBMS_METADATA.GET_DDL FAILED WITH ORA-16000 IN READ ONLY MODE
Bug 30928455	DATA PUMP EXPORT | HITTING ORA-31637 WHILE RUNNING DATA PUMP-DPLOAD CONCURRENCY TEST IN SAME PDB
Bug 30944402	SELECT FROM MASTER TABLE RUNS SLOW DURING TABLE_DATA EXPORT WHEN THERE ARE MANY SUBPARTITIONS
Bug 30978304	ORA-20000 DURING IMPDP WITH STATS AND THE UNIQUE INDEX FOR THE PK IS NOT CREATED
Bug 31050896	PARALLEL DATAPUMP SLOW ON CONSTRAINTS
Bug 31174337	DBMS_METADATA.GET_DDL GENERATES NO KEYWORDS FOR NOT COMPRESSED INDEXES
Bug 31191614	TTS EXPDP QUERIES V$ENCRYPTED_TABLESPACES FOR EVERY TBS SLOWING DOWN PERFORMANCE
Bug 31200854	ADB-D: IMPORT PERFORMANCE OF PACKAGE_BODY
Bug 31393386	SPIN-OFF OF BUG# 31317961 FOR PARTIAL BACKOUT OF BUG# 27403988 FROM MAIN LABEL
Bug 31402031	DBMS_METADATA_UTIL THROWS AN INVALID CURSOR EXCEPTION.
Bug 31412130	ADBD:: COMPLETE FIX FOR 29543605 WHICH INCLUDES ALL THE MISSING FILES
Bug 31424070	APPSST19C: XTTS PDB - TABLE IMPORT/CREATION FAILED WITH ORA-39083 ORA-14334
Bug 31711479	ADB-S: ORA39126 AND ORA01031 WHILE IMPORT USING FA FULL DUMP INTO ADB-S
Bug 31725941	TOTAL ESTIMATION USING BLOCKS METHOD IS MISSING STARTING WITH 12.2
Bug 31830685	ZDM : IMPORT ADW-S DB LINK MIGRATION THROWS INTERNAL ERROR
Bug 32096059	IMPDP TO 19C USING EXPORT DUMP OF 11.2.0.4 HANGS WITH ENQ: TM - CONTENTION
Bug 32370367	EXPDP IN 19.7 THREE TIMES SLOWER THAN IT WAS IN 11.2.0.4
Bug 32452792	DBMS_METADATA.GET_DDL GETS WRONG OUTPUT FROM 12.2.0.1. TESTED TILL 19.3.0.0
Bug 32512780	PROCOBJ PLSQL SCRIPTS ARE NOT EXCLUDED ON IMPORT WITH EXCLUDE=TAG
Bug 32647307	ADB-D:: PACKAGE BODIES IMPORT SLOWER AFTER AUTONOMOUS REFRESH TO 19.10DBRU
Bug 32731035	ATPD MIGRATION:ORA-04021: TIMEOUT OCCURRED WHILE WAITING TO LOCK OBJECT
Bug 33163877	ATPD MIGRATION:IMPDP HITS TABLE OR VIEW DOES NOT EXIST ON SOME DATAPUMP RELATED TABLES
Bug 33204663	TCH19C :: ORA-39139: DATA PUMP DOES NOT SUPPORT XMLTYPE OBJECTS WHEN DOING XTTS WITH BINARY XML STORAGE
Bug 33297599	UNUSED XMLTYPE/CLOB COLUMNS CAUSE IMPORT FAILURE
Bug 33346378	REWRITE DATA PUMP PATCH LOCKING TEST: TKDPATCHRAC.TSC
Bug 33448450	TCH19C :: ORA-01647: TABLESPACE 'APPS_TS_TX_DATA' IS READ-ONLY, CANNOT ALLOCATE SPACE
Bug 33470563	METADATA API FAILS TO RECOGNIZE TAB CHARACTER AS DELIMITER WHEN PARSING SOURCE LINES OF TYPE OBJECT
Bug 33498804	DATAPUMP IMPORT IGNORES EXCLUDE AND INCLUDE VALUES FOR TAGS FOR IMPORT CALLOUTS
Bug 33660169	CONSOLIDATED BUG FOR DATA PUMP AQ FIXES 31338354, 31844376, 31868443 FOR 19.10 AND LATER
Bug 33720650	TCH19C :: OCI-21500: INTERNAL ERROR CODE [QMCXDGETQNAMEINFO2], [14003] IN XMLTYPE CLOUMN TYPE
Bug 33735435	TRACKING BUG FOR COMBO OF 32759991 32878145 32919937 32984678 (REPLACEMENT FOR MINI MLR 33407604)
Bug 34052641	END_PLUGTS_BLK OBJECT TYPE MISSING FROM FULL TTS EXPORT WHEN INCLUDE SPECIFIED
Bug 34525626	TRACKING BUG TO MERGE 33599275 AND 33498804 SO CAN BE BACKPORTED TOGETHER TO 19.16



Why aren't these fixes included in an RU?

Data Pump Bundle Patch is not RAC Rolling and Standby-First Installable

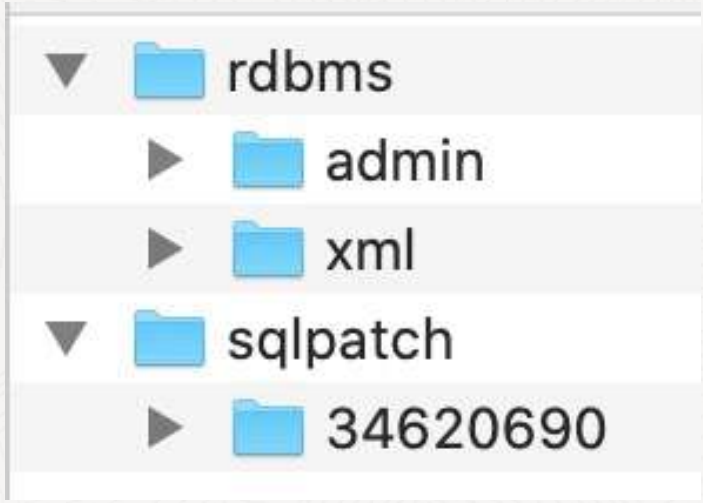
But … it's much easier than it looks like

Data Pump Bundle Patch Contents

Bundle Patch contains only:

- SQL
- PL/SQL
- XML

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



It can be applied online

```
OPatch continues with these patches:   34620690

Do you want to proceed? [y|n]
y
User Responded with: Y
All checks passed.a
Backing up files...
Applying interim patch '34620690' to OH '/u01/app/oracle/product/19'

Patching component oracle.rdbms, 19.0.0.0.0...

Patching component oracle.rdbms.dbscripts, 19.0.0.0.0...
Patch 34620690 successfully applied.
```


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

Non-Binary Online Patching Safeguards

Installing the Data Pump Bundle Patch when Data Pump is in use:

Built-in 3-minute timeout before signaling an error

```
BEGIN ku$_dpload.initial_phase; END;
*
ERROR at line 1:
ORA-20000: Retry dpload.sql script later when
Data Pump and Metadata API are not in use; current users are:
pid:11720, user:SYS, machine:<Machine>, sid:263,
module:sqlplus@<ConnectString> (TNS V1-
ORA-06512: at "SYS.KU$_DPLOAD", line 1042
ORA-06512: at line 1
```

Non-Binary Online Patching Safeguards

Attempting to run Data Pump while patching is in progress:

```
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
ORA-31626: job does not exist
ORA-31637: cannot create job SYS_EXPORT_FULL_01 for user SYSTEM
ORA-06512: at "SYS.KUPV$FT", line 1142
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 95
ORA-06512: at "SYS.KUPV$FT", line 1751
ORA-39062: error creating master process DM00
ORA-39107: Master process DM00 violated startup protocol. Master error:
…
```

Note:
With the 19.14 (or later) Data Pump Bundle Patch installed you will see a much better error message:

```
ORA-39442: Data Pump software update in progress
```

Once applied, Data Pump Bundle Patch speeds up future patching significantly

Importing a complete application with data goes from almost 2,5 hours to 48 minutes – by just applying the Data Pump Bundle Patch

Global provider of financial services

Quiz 2

What does a Data Pump full database export include?

1. AWR Reports
2. Grants to SYS objects. Eg: EXECUTE ON DBMS_STATS
3. Audit Trail and Policies
4. Global Stats Preferences
5. BFILE LOBs

Use Case – Analyse dump contents before import

```
-- Only generate master and stop.
...
H1 := DBMS_DATAPUMP.OPEN(
  OPERATION => 'IMPORT',
  JOB_MODE => 'SCHEMA',
  REMOTE_LINK => NULL,
  JOB_NAME => 'MY_JOB_1',
  VERSION => 'LATEST');
...
DBMS_DATAPUMP.SET_PARAMETER(HANDLE => H1, NAME => 'MASTER_ONLY', VALUE => 1);
...

DBMS_DATAPUMP.START_JOB(HANDLE => H1);
```

```
SELECT DISTINCT OBJECT_SCHEMA, OBJECT_NAME, OBJECT_TYPE
FROM MY_JOB_1
WHERE OBJECT_SCHEMA IS NOT NULL AND OBJECT_TYPE IS NOT NULL AND OBJECT_NAME IS NOT NULL;
```


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
```

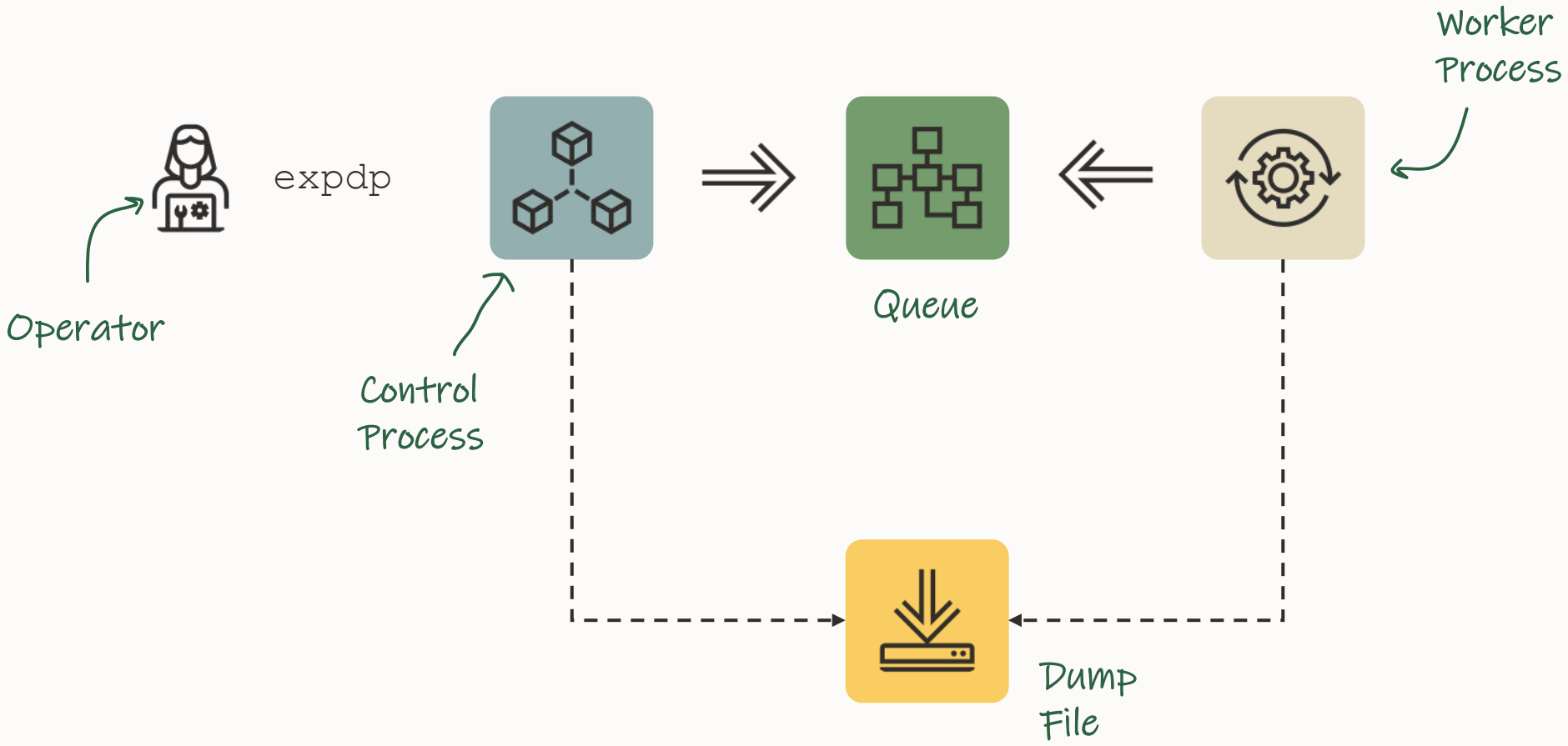
Use DUMPFILE parameter

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


Parallel | Control and Worker process

If you use the default or `PARALLEL=1`

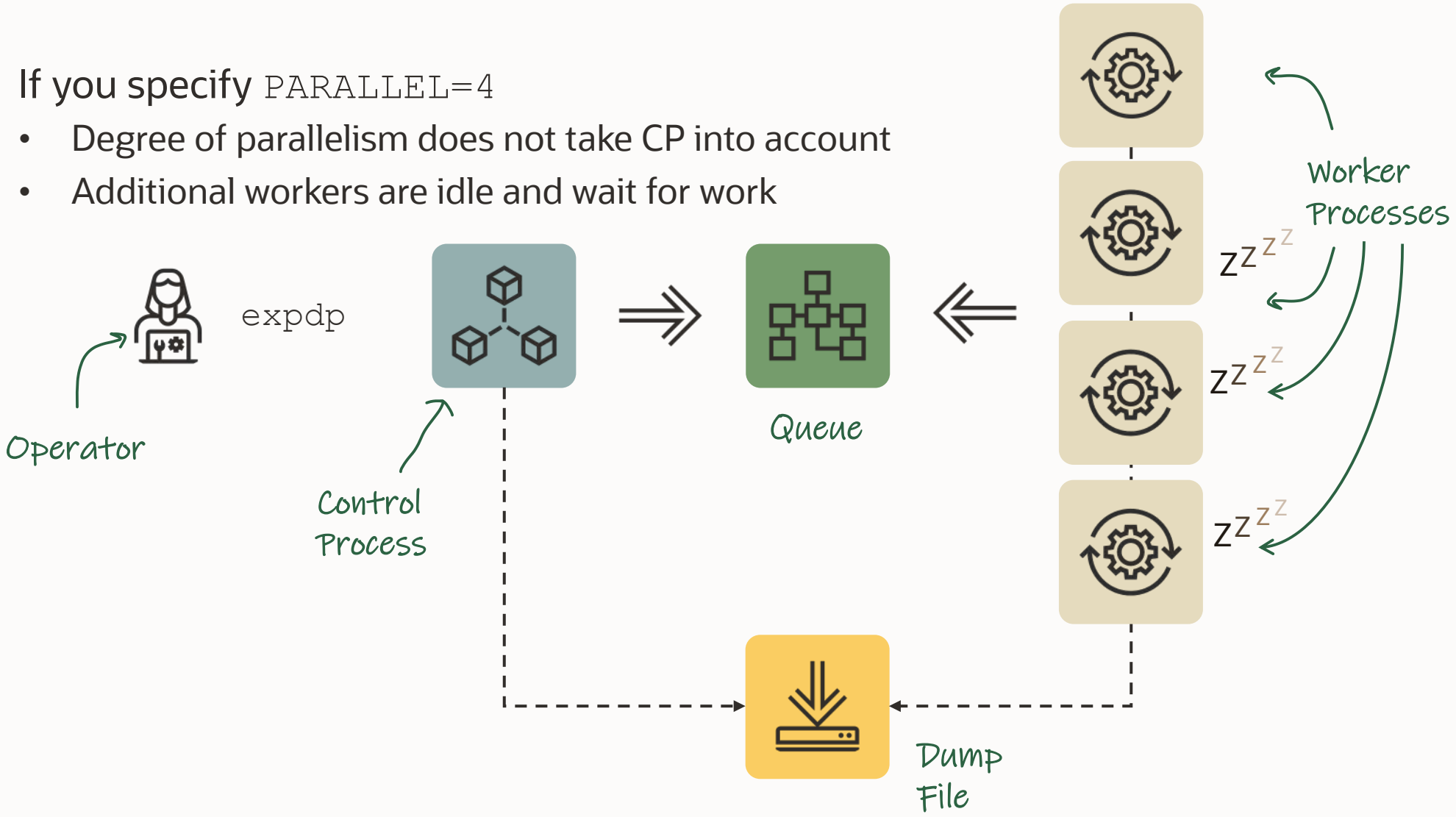
- 2 processes, 1 control process and 1 worker



Parallel | Degree of Parallelism

If you specify `PARALLEL=4`

- Degree of parallelism does not take CP into account
- Additional workers are idle and wait for work



Include diagnostics in the logfile

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

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


```
21-JUL-23 19:57:20.829: W-1 Startup on instance 6 took 0 seconds
21-JUL-23 19:57:21.089: W-1 Master table "HASH"."Z_TABLE_IMPORT_652" successfully loaded/unloaded
21-JUL-23 19:57:22.132: Starting "HASH"."Z_TABLE_IMPORT_652":
21-JUL-23 19:57:22.159: W-1 Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
*** Job percent done = 7
21-JUL-23 19:57:46.439: W-1 . . imported "HASH"."Z_DM_CONTENTS_652"          111.8 MB   12861 rows in 23 seconds using external_table
*** Job percent done = 9
21-JUL-23 19:57:53.168: W-1 . . imported "HASH"."Z_DM_CODES_652"            30.65 MB   16080 rows in 7 seconds using external_table
21-JUL-23 19:57:55.551: W-1 . . imported "HASH"."Z_T_SYMBOLS_652"           3.772 MB  556882 rows in 2 seconds using external_table
*** Job percent done = 10
21-JUL-23 19:57:58.385: W-1 . . imported "HASH"."Z_T_TAB_COLUMNS_652"       5.139 MB  328647 rows in 3 seconds using external_table
21-JUL-23 19:58:00.191: W-1 . . imported "HASH"."Z_T_FILES_652"             3.409 MB   88391 rows in 2 seconds using external_table
21-JUL-23 19:58:02.406: W-1 . . imported "HASH"."Z_T_OBJECTS_652"           1.914 MB  275442 rows in 2 seconds using external_table
21-JUL-23 19:58:04.234: W-1 . . imported "HASH"."Z_T_TAB_PRIVS_652"         1.380 MB  132981 rows in 2 seconds using external_table
21-JUL-23 19:58:06.222: W-1 . . imported "HASH"."Z_T_PROCEDURES_652"        996.7 KB   87076 rows in 2 seconds using external_table
21-JUL-23 19:58:08.000: W-1 . . imported "HASH"."Z_T_FIXED_VIEW_DEFINITION_652" 169.7 KB  1326 rows in 1 seconds using external_table
21-JUL-23 19:58:09.961: W-1 . . imported "HASH"."Z_T_SYNONYMS_652"          517.1 KB   37510 rows in 1 seconds using external_table
21-JUL-23 19:58:11.451: W-1 . . imported "HASH"."Z_T_TXTCOLLECTION_652"     347.9 KB   15761 rows in 2 seconds using external_table
21-JUL-23 19:58:13.003: W-1 . . imported "HASH"."Z_T_HASH_652"              645.6 KB   16139 rows in 1 seconds using external_table
21-JUL-23 19:58:14.474: W-1 . . imported "HASH"."Z_T_XTABCOLS_652"          108.4 KB   18949 rows in 2 seconds using external_table
21-JUL-23 19:58:16.046: W-1 . . imported "HASH"."Z_T_PARAMETER_652"         115.4 KB    3978 rows in 2 seconds using external_table
21-JUL-23 19:58:16.050: W-1 . . imported "HASH"."Z_T_REGISTRY_ERROR_652"        0 KB       0 rows in 0 seconds using external_table
21-JUL-23 19:58:17.677: W-1 . . imported "HASH"."Z_T_TRIGGERS_652"          58.09 KB    1779 rows in 1 seconds using external_table
21-JUL-23 19:58:19.276: W-1 . . imported "HASH"."Z_T_AUDIT_UNIFIED_POLICIES_652" 41.16 KB  5700 rows in 2 seconds using external_table
21-JUL-23 19:58:20.714: W-1 . . imported "HASH"."Z_T_BUGSFIXED_652"         56.18 KB    1685 rows in 1 seconds using external_table
```

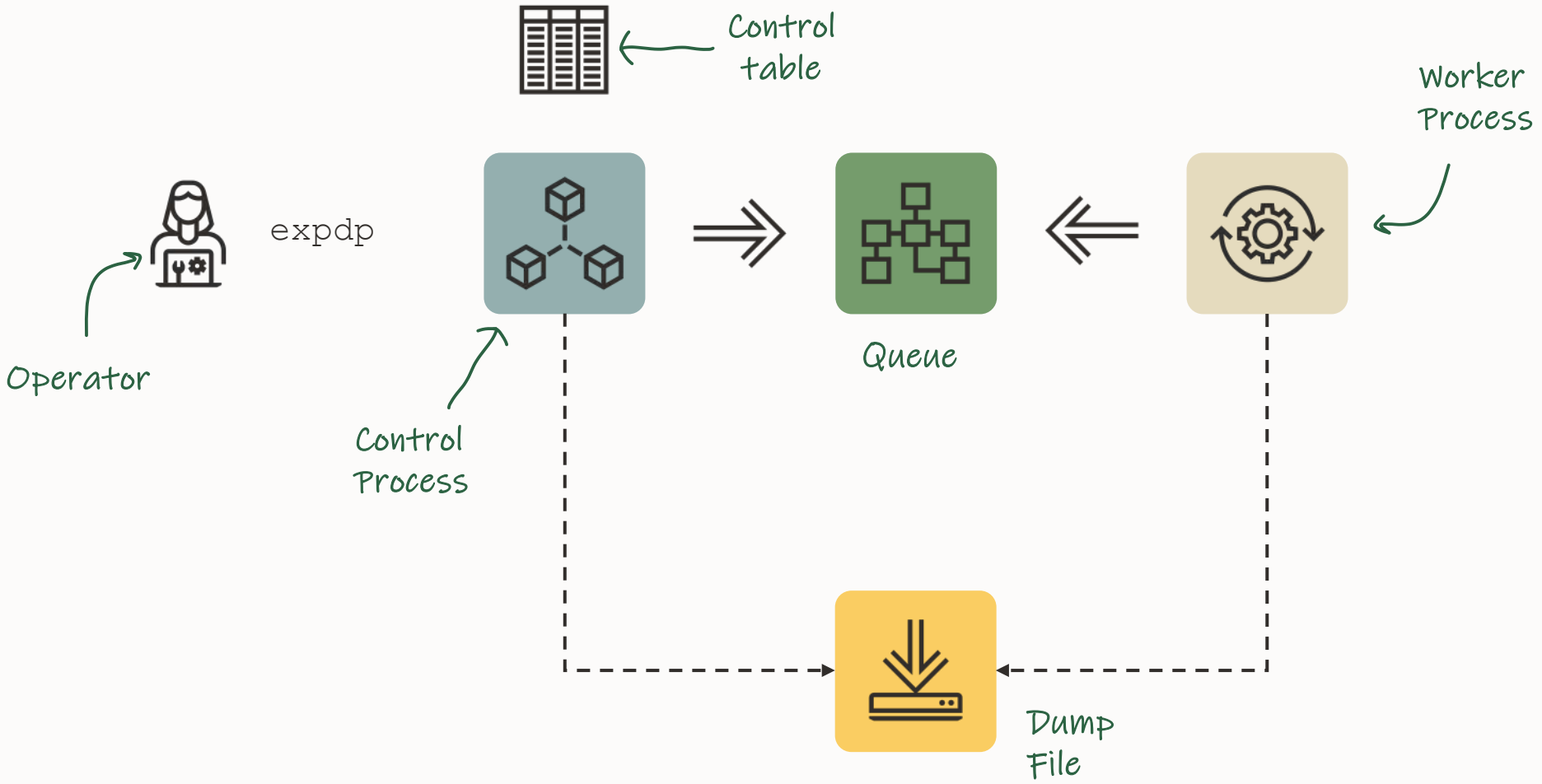
Quiz 3

What happens when you do CTRL+C while expdp/impdp is running?

1. Oh no, this will kill the Data Pump process!

2. It will get you into the Data Pump iteractive mode.

Data Pump | Architecture



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>
```

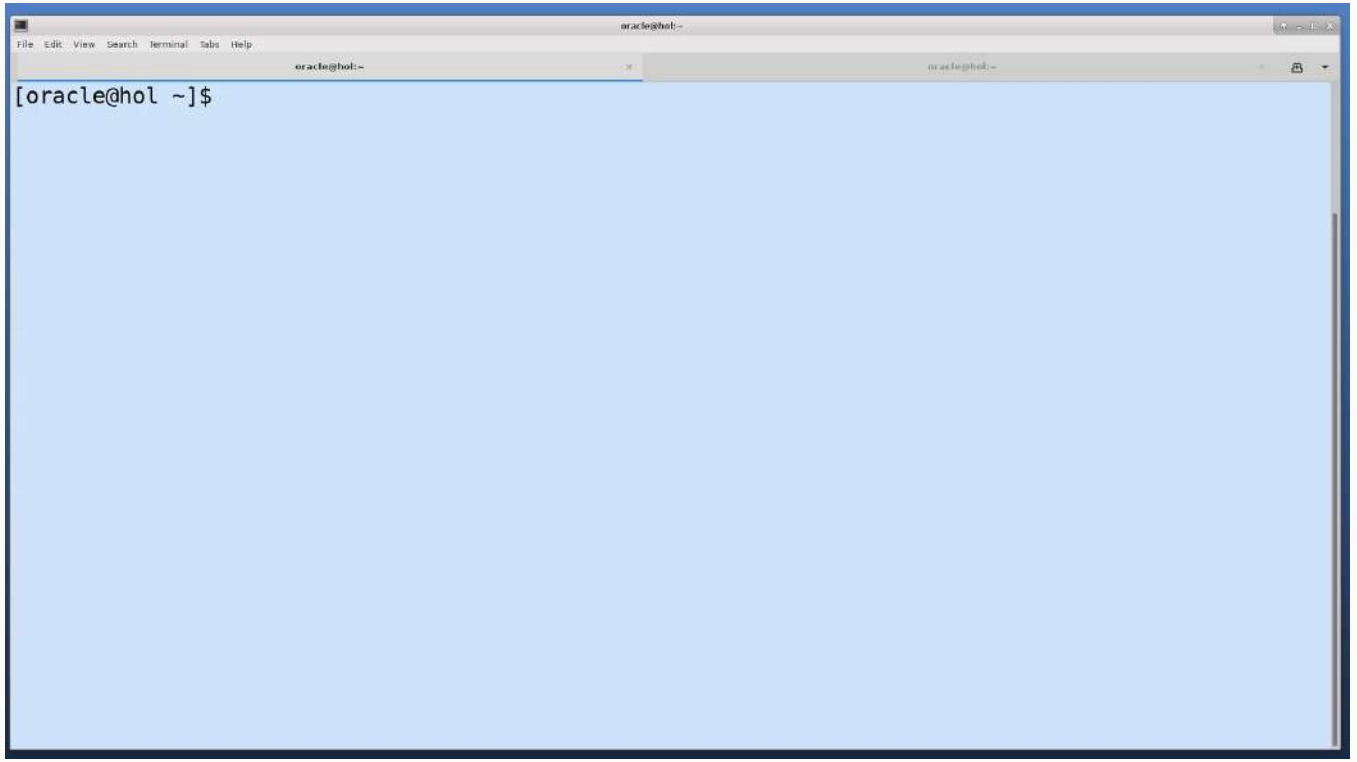

Interactive Command Mode | Overview

Different commands are available for exports and imports.

Command	Mode	Description
`PARALLEL=n`	Both	Change the parallelism for current job. Increases almost immediately.
`STATUS`	Both	Get the job and worker status. Includes Operation, Mode, State, Percent Done, and Current Parallelism .
`STATUS=120`	Both	As above but refreshes every 120 second
`FILESIZE=n`	Export	Changes the file size (in bytes) of the dump files. Optionally specify denominator, e.g., FILESIZE=5G
`ADD_FILE=name`	Export	Adds an additional dump file. Or a dump file pattern, e.g., ADD_FILE=more_files%L.dmp
`TRACE=nnn`	Both	Adds tracing, see MOS ID 286496.1 for details

More commands are found in the documentation

Interactive Command Mode | Demo



Watch on YouTube

Photo by Kai Pilger on Unsplash

I have no server access.

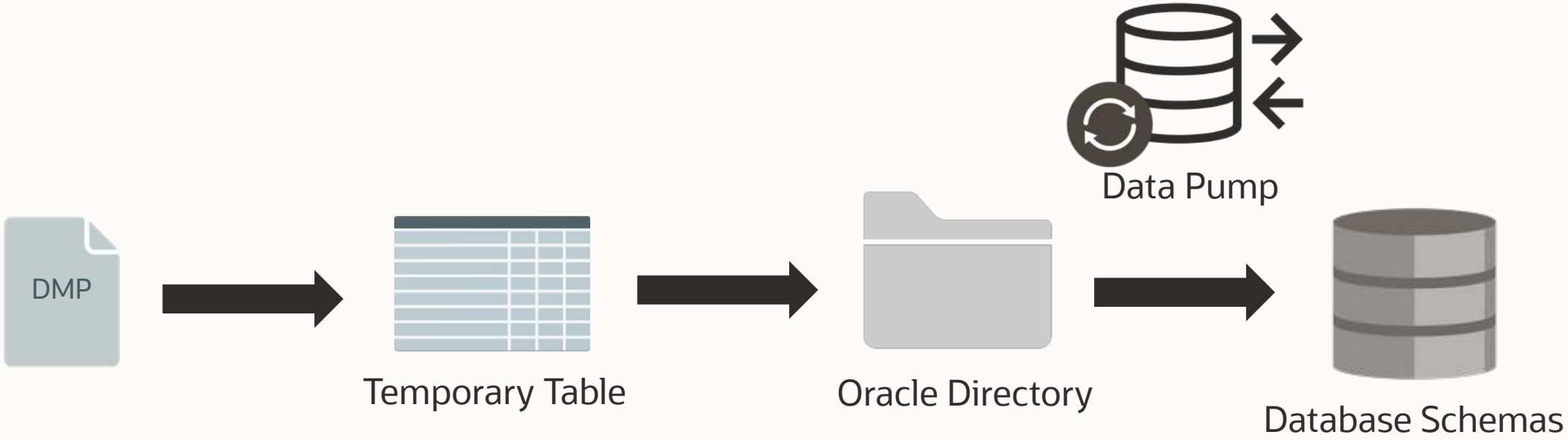
How do I load the data?

No server access | Hacks

You will need at least:

- CREATE SESSION
- CREATE TABLE
- READ and WRITE in directory (eg: DATA_PUMP_DIR)

No server access | Import Strategy

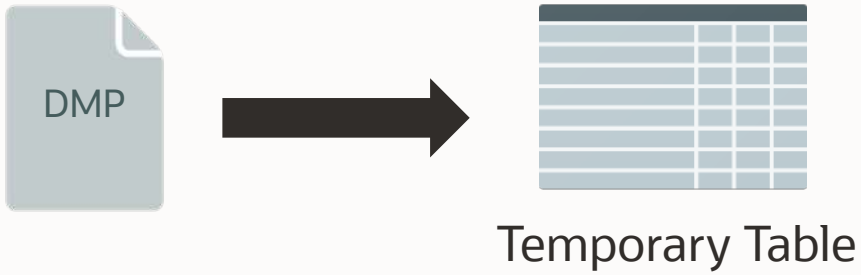



```
DBMS_LOB.READ + UTL_FILE.PUT_RAW
```

```
CREATE TABLE TEMP_LOB (
        file_name varchar2(1000) primary key,
        blob_content BLOB
);
```

No server access | Moving file to BLOB

- Missing step:



No server access | Moving file to BLOB

Strategies:

1. sqlldr
2. sqlcl
3. Base64 decode / encode

No server access | Moving file to BLOB | sqlldr

lob_test.ctl

```
LOAD DATA
INFILE 'lob_test_data.txt'
append
INTO TABLE lob_tab
FIELDS TERMINATED BY ','
(file_name CHAR(100),
blob_content LOBFILE(file_name) TERMINATED BY EOF)
```

No server access | Moving file to BLOB | sqlldr

```
$ echo 'mydump_meta_backup_20220606_175153.dmp' > lob_test_data.txt
$ sqlldr /@adb_tp control=lob_test.ctl log=lob_test.log bad=lob_test.bad
```

```
SQL> select file_name from lob_tab;
FILE_NAME
----------------------------------------
mydump_meta_backup_20220606_175153.dmp
```

Blog Post: https://www.dbarj.com.br/en/2022/06/how-to-run-impdp-in-adb-when-you-dont-have-access-to-object-storage-or-db-links/

No server access | Moving file to BLOB

Strategies:

1. sqlldr
2. sqlcl
3. Base64 decode / encode

No server access | Moving file to BLOB | sqlcl

- sqlcl can run JavaScript
- Create a javascript code that writes a blob into a table
- Example: **upload_file.js**

```
SQL> script
2 ctx.write('My first script\n');
3 /
My first script
SQL>
```

No server access | Moving file to BLOB | sqlcl

```
function putFile(filename) {

var blob = conn.createBlob();
var stream = blob.setBinaryStream(0);
var path = java.nio.file.FileSystems.getDefault().getPath(filename);
java.nio.file.Files.copy(path, stream);
stream.flush();

var ret=util.execute(
'insert into lob_tab(file_name, blob_content) values (:file_name , :blob_content)',
{ file_name : filename,
blob_content : blob }
);

if (!ret) {
print('Something unintended happened.');
}

}

putFile('mydump_meta_backup_20220606_175153.dmp');
conn.commit();
```

No server access | Moving file to BLOB | sqlcl

- sqlcl can run JavaScript
- Create a javascript code that writes a blob into a table
- Example: **upload_file.js**
- Call the created *js* from sqlcl

No server access | Moving file to BLOB | sqlcl

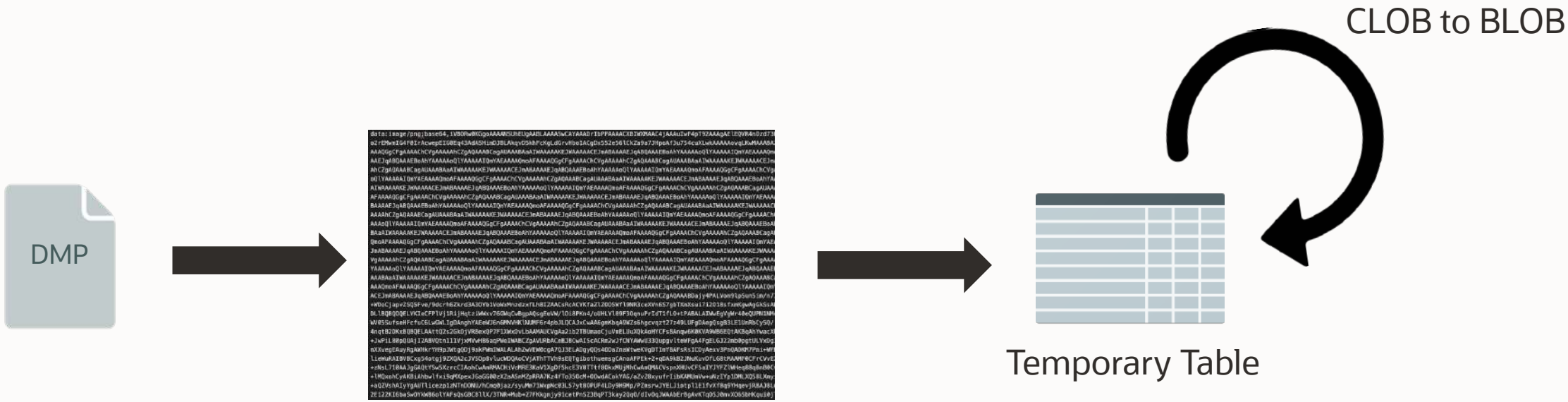
```
SQL> script upload_file.js
SQL>
SQL> select file_name from lob_tab;
FILE_NAME
--------------------------------------------------
mydump_meta_backup_20220606_175153.dmp
```


No server access | Moving file to BLOB

Strategies:

1. sqlldr
2. sqlcl
3. Base64 decode / encode

No server access | Moving file to BLOB | Base64




```
$ base64 -i mydump_meta_backup_20220606_175153.dmp
```

```
CREATE TABLE TEMP_LOB (
                                                    rimary key,
                blob_content BLOB
);
```

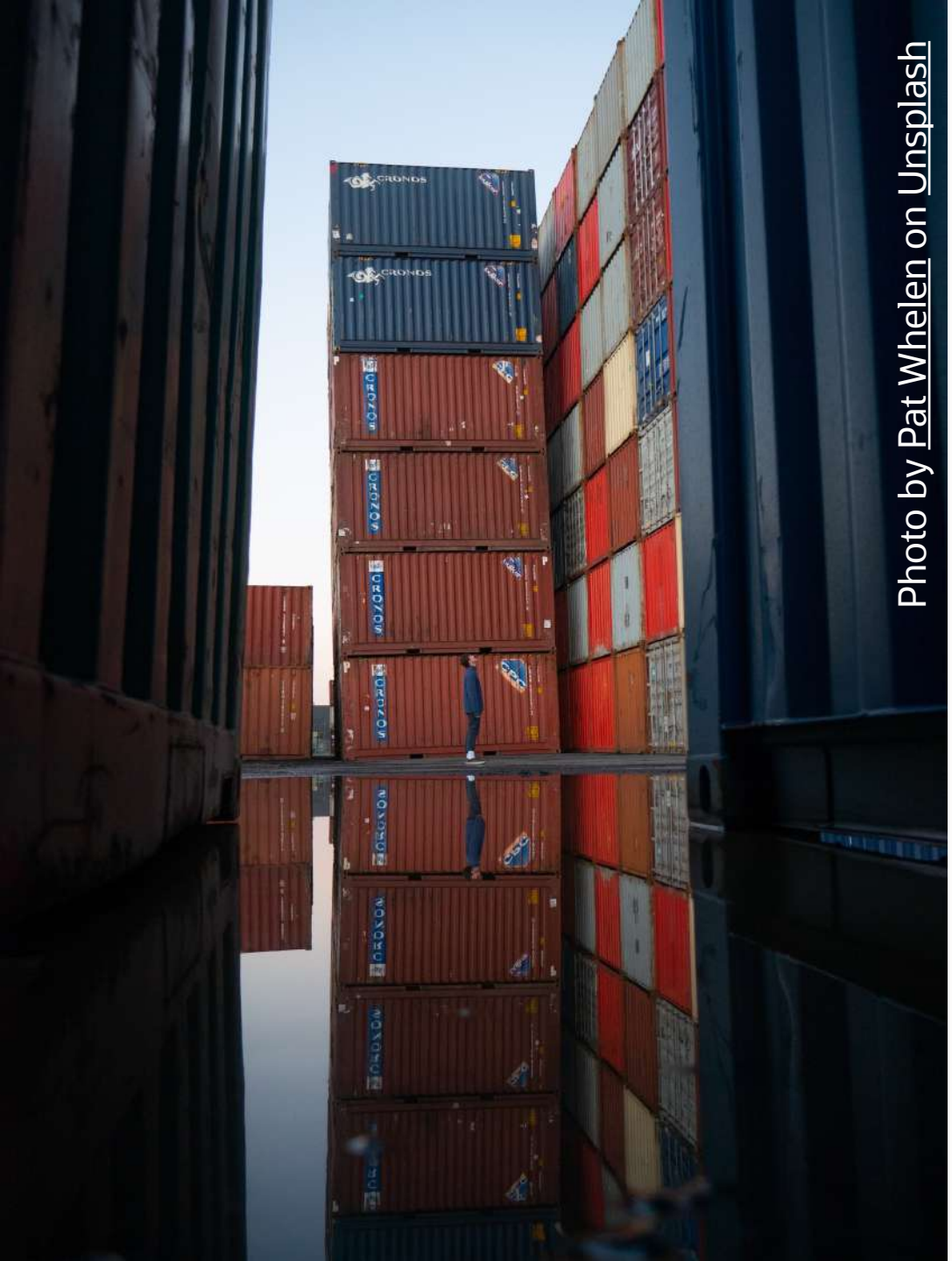
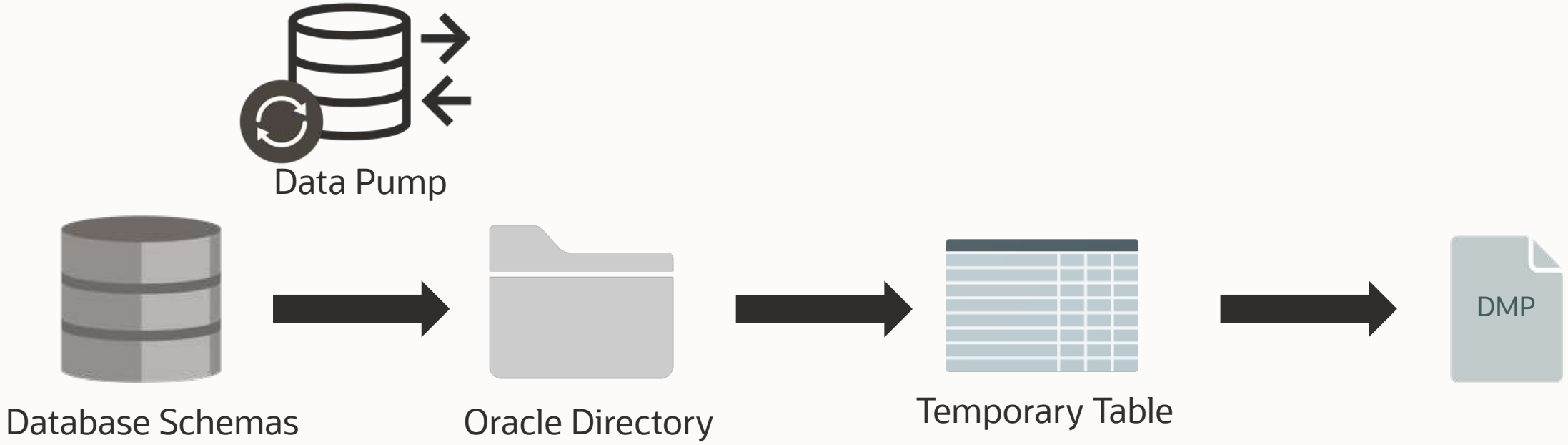


Photo by Pat Whelen on Unsplash

And how to export ?

No server access | Export Strategy

Same strategy but on the oposite direction!



Want to know more about

DATAPUMP

Virtual Classroom 13:

Data Pump Extreme - Deep Dive with Development

Thank you


www.dbarj.com.br


@rodrigojorgedba


/rodrigoaraujorge