Data Pump: export i import

Poziom: operacyjny **Dotyczy: Oracle Database 12c–19c, Data Pump (expdp/impdp)**

Czas: ~25 min

Data Pump to logiczny eksport i import danych — inny mechanizm niż backup fizyczny RMAN, przydatny wszędzie tam, gdzie chcemy przenieść tylko wybrany schemat, tabelę albo wersję danych między bazami różnych wersji.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
RMAN	Recovery Manager: narzędzie Oracle do kopii zapasowych
SQL	Structured Query Language: język zapytań do relacyjnych baz danych
WMS	Warehouse Management System: system zarządzania magazynem

Fiszka

ZMIENIA SYSTEM

Utworzenie katalogu na pliki Data Pump: CREATE DIRECTORY

```
CREATE DIRECTORY NazwaKatalogu AS 'SciezkaDoKatalogu';
```

> **Pojęcie: directory (obiekt bazy).** Data Pump nie przyjmuje dowolnej ścieżki systemowej wprost w poleceniu `expdp`/`impdp` — najpierw trzeba zarejestrować katalog systemowy jako obiekt bazy danych, a dopiero potem odwoływać się do niego po nazwie. Katalog fizyczny pod wskazaną ścieżką musi istnieć na dysku i być zapisywalny przez konto systemowe `oracle`, zanim cokolwiek do niego trafi.

Przykład:

```
CREATE DIRECTORY EXPDIR AS '/oracle/db/wmsprd/datapump';
```

```
Directory created.
```

Nadanie uprawnień do zapisu i odczytu w tym katalogu dla użytkownika wykonującego export lub import:

```
GRANT READ, WRITE ON DIRECTORY NazwaKatalogu TO NazwaUsera;
```

□□ TYLKO ODCZYT

Sprawdzenie uprawnień do directories

```
SELECT grantee, table_name AS directory_name, privilege
FROM dba_tab_privs
WHERE table_name = 'NazwaKatalogu';
```

Sprawdzamy, kto ma prawo czytać i pisać do danego katalogu Data Pump — directory jest w słowniku danych traktowany jak obiekt, więc jego uprawnienia widać w tym samym miejscu co uprawnienia do tabel.

Przykład:

```
GRANTEE     DIRECTORY_NAME  PRIVILEGE
----------- --------------- ----------
WMS_APP      EXPDIR          READ
WMS_APP      EXPDIR          WRITE
WMS_ADMIN    EXPDIR          READ
```

□□ TYLKO ODCZYT

Export pełnej bazy: expdp FULL

```
expdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log FULL=Y
```

Eksportujemy całą zawartość bazy do jednego zrzutu — używane rzadko, głównie przy pełnej migracji instancji na inny serwer albo inną wersję silnika.

Przykład:

```
Export: Release 19.0.0.0.0 - Production on Fri Sep 25 10:02:11 2026
Starting "SYS"."SYS_EXPORT_FULL_01":
Estimate in progress using BLOCKS method...
Total estimation using BLOCKS method: 12.4 GB
Master table "SYS"."SYS_EXPORT_FULL_01" successfully loaded/unloaded
Dump file set for SYS.SYS_EXPORT_FULL_01 is:
  /oracle/db/wmsprd/datapump/full_export.dmp
Job "SYS"."SYS_EXPORT_FULL_01" successfully completed
```

□□ **expdp** = export Data Pump: eksport danych Oracle

□□ TYLKO ODCZYT

Export wybranych schematów: expdp SCHEMAS

```
expdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
SCHEMAS=NazwaSchematu1,NazwaSchematu2
```

Eksportujemy tylko wskazane schematy razem z ich obiektami — najczęstszy wariant, kiedy przenosimy konkretną aplikację, a nie całą bazę.

Przykład:

```
expdp directory=EXPDIR dumpfile=appusr_export.dmp logfile=appusr_export.log
schemas=WMS_APP,WMS_ADMIN
```

```
Estimate in progress using BLOCKS method...
Total estimation using BLOCKS method: 3.1 GB
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . exported "WMS_APP"."ZAMOWIENIA"                      412.3 MB 1834021 rows
Master table "SYS"."SYS_EXPORT_SCHEMA_01" successfully loaded/unloaded
Job "SYS"."SYS_EXPORT_SCHEMA_01" successfully completed
```

□□ **expdp** = export Data Pump: eksport danych Oracle

□□ TYLKO ODCZYT

Export wskazanych tablespace: expdp TABLESPACES

```
expdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
TABLESPACES=NazwaTablespace
```

Eksportujemy wszystkie obiekty leżące w konkretnym tablespace, niezależnie od tego, do jakich schematów należą — przydaje się, kiedy dzielimy dane po tablespace, a nie po właścicielu.

Przykład:

```
expdp directory=EXPDIR dumpfile=intg_export.dmp logfile=intg_export.log tablespaces=INTG
```

```
Job "SYS"."SYS_EXPORT_TABLESPACE_01" successfully completed
```

expdp = export Data Pump: eksport danych Oracle

ZMIENIA SYSTEM

Export z filtrowaniem obiektów i wierszy: EXCLUDE, INCLUDE, QUERY

```
expdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
SCHEMAS=NazwaSchematu EXCLUDE=STATISTICS QUERY=NazwaSchematu.NazwaTabeli:"WHERE
data_utworzenia >= SYSDATE-30"
```

Precyzyjnie zawężamy, co trafia do zrzutu: `EXCLUDE` / `INCLUDE` filtruje typy obiektów (na przykład pomijając statystyki albo indeksy), a `QUERY` filtruje same wiersze konkretnej tabeli tak jak klauzula `WHERE` w zwykłym `SELECT`.

Przykład:

```
expdp directory=EXPDIR dumpfile=appusr_recent.dmp logfile=appusr_recent.log
schemas=WMS_APP exclude=statistics query=WMS_APP.ZAMOWIENIA:"WHERE data_utworzenia >=
SYSDATE-30"
```

```
. . exported "WMS_APP"."ZAMOWIENIA":"WHERE data_utworzenia >= SYSDATE-30"   38.2 MB
91442 rows
Job "SYS"."SYS_EXPORT_SCHEMA_01" successfully completed
```

expdp = export Data Pump: eksport danych Oracle

TYLKO ODCZYT

Export równoległy i skompresowany: PARALLEL, COMPRESSION

```
expdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku_%U.dmp LOGFILE=NazwaPliku.log
SCHEMAS=NazwaSchematu PARALLEL=4 COMPRESSION=ALL
```

Przyspieszamy duży export przez rozbicie go na kilka równoległych procesów (`PARALLEL`) i zmniejszamy rozmiar zrzutu kompresją (`COMPRESSION`) — przy `PARALLEL` większym niż 1 nazwa pliku musi zawierać `%U`, żeby każdy proces mógł zapisywać do osobnego pliku.

Przykład:

```
expdp directory=EXPDIR dumpfile=wmsprd_%U.dmp logfile=wmsprd_export.log schemas=WMS_APP
parallel=4 compression=all
```

```
Starting "SYS"."SYS_EXPORT_SCHEMA_01" with 4 workers
Job "SYS"."SYS_EXPORT_SCHEMA_01" successfully completed
```

expdp = export Data Pump: eksport danych Oracle

TYLKO ODCZYT

Import ze zrzutu do wskazanego schematu: impdp SCHEMAS

```
impdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
SCHEMAS=NazwaSchematu1,NazwaSchematu2
```

Importujemy zawartość zrzutu z powrotem do bazy — podstawowy wariant, kiedy docelowy schemat i tablespace mają te same nazwy co przy eksporcie.

Przykład:

```
impdp directory=EXPDIR dumpfile=appusr_export.dmp logfile=appusr_import.log
schemas=WMS_APP,WMS_ADMIN
```

```
Master table "SYS"."SYS_IMPORT_SCHEMA_01" successfully loaded/unloaded
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "WMS_APP"."ZAMOWIENIA"                          412.3 MB 1834021 rows
Job "SYS"."SYS_IMPORT_SCHEMA_01" successfully completed
```

impdp = import Data Pump: import danych Oracle

TYLKO ODCZYT

Import z zachowaniem istniejących tabel: TABLE_EXISTS_ACTION

```
impdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
TABLESPACES=NazwaTablespace TABLE_EXISTS_ACTION=REPLACE
```

Sterujemy tym, co ma się stać, jeśli tabela z importowanego zrzutu już istnieje w docelowej bazie — `REPLACE` dropuje istniejącą tabelę i tworzy ją od nowa z danymi ze zrzutu, `APPEND` dokłada wiersze do istniejącej, `SKIP` (domyślne) pomija tabelę całkowicie, `TRUNCATE` czyści tabelę przed wczytaniem nowych danych.

Przykład:

```
impdp directory=EXPDIR dumpfile=intg_export.dmp logfile=intg_import.log tablespaces=INTG
table_exists_action=replace
```

```
Job "SYS"."SYS_IMPORT_TABLESPACE_01" successfully completed
```

impdp = import Data Pump: import danych Oracle

TYLKO ODCZYT

Import do innego schematu i tablespace: REMAP_SCHEMA, REMAP_TABLESPACE

```
impdp DIRECTORY=NazwaKatalogu DUMPFILE=NazwaPliku.dmp LOGFILE=NazwaPliku.log
REMAP_SCHEMA=NazwaSchematuZrodlowego:NazwaSchematuDocelowego
REMAP_TABLESPACE=NazwaTablespaceZrodlowego:NazwaTablespaceDocelowego
```

Przenosimy dane pod innym właścicielem i do innego tablespace niż w bazie źródłowej — typowy przypadek przy odtwarzaniu produkcyjnych danych na środowisku testowym, gdzie schemat i tablespace nazywają się inaczej albo celowo chcemy oddzielić dane testowe od produkcyjnych nazw.

Przykład:

```
impdp directory=EXPDIR dumpfile=appusr_export.dmp logfile=appusr_import_test.log
remap_schema=WMS_APP:WMS_APP_TEST remap_tablespace=INTG:TESTDATA
```

```
Processing object type SCHEMA_EXPORT/USER
. . imported "WMS_APP_TEST"."ZAMOWIENIA"                  412.3 MB 1834021 rows
Job "SYS"."SYS_IMPORT_SCHEMA_01" successfully completed
```

impdp = import Data Pump: import danych Oracle

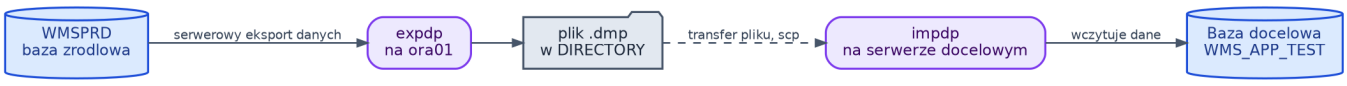
Z życia wzięte

Z ŻYCIA WZIĘTE

Ola prosi o kopię schematu WMS_APP z produkcji, żeby przetestować nową wersję raportu bez ryzyka dla WMSPRD. Tomek robi `expdp` ze `SCHEMAS=WMS_APP` na ora01, przenosi zrzut i importuje go do bazy testowej poleceniem `impdp` z `REMAP_SCHEMA` na WMS_APP_TEST. Ola dostaje świeże dane produkcyjne do testów, a Marek w Swarzędzu nie widzi żadnej zmiany na produkcyjnym WMS.

Czym jest Data Pump

Pojęcie: Data Pump. Mechanizm eksportu i importu danych działający po stronie serwera bazy danych — polecenia `expdp` (export) i `impdp` (import) łączą się z instancją i proszą ją samą o wygenerowanie albo wczytanie pliku zrzutu (`.dmp`), zamiast przesyłać dane przez klienta SQL wiersz po wierszu. Dzięki działaniu po stronie serwera Data Pump jest wielokrotnie szybszy od starszych narzędzi `exp` / `imp`, które są dziś przestarzałe, i pozwala na równoległość (`PARALLEL`), kompresję w locie oraz precyzyjne filtrowanie eksportowanych obiektów.



Export bierze dane bezpośrednio z bazy źródłowej do pliku dmp w katalogu DIRECTORY, plik wędruje na inny serwer, a import wczytuje go do bazy docelowej (kliknięcie otwiera pełny rozmiar)

W przeciwieństwie do RMAN, który kopiuje pliki fizyczne bazy, Data Pump eksportuje dane logicznie — sam zestaw wierszy i definicji obiektów, niezależnie od wersji silnika i fizycznej struktury

tablespace'ów. To czyni go właściwym narzędziem do przenoszenia pojedynczego schematu między środowiskami, migracji między różnymi wersjami Oracle albo do zrobienia kopii danych bez odtwarzania całej bazy — role, których RMAN (opisany na stronie RMAN: backup, walidacja, czyszczenie archiwów) nie pełni.

Monitorowanie zadania Data Pump

Sprawdzenie aktywnych zadań: DBA_DATAPUMP_JOBS

```
SELECT owner_name, job_name, operation, job_mode, state, degree
FROM dba_datapump_jobs;
```

Sprawdzamy, jakie zadania Data Pump aktualnie działają w bazie — przydaje się, kiedy sesja z terminala, która uruchomiła `expdp`, została rozłączona, a chcemy sprawdzić, czy zadanie mimo to kontynuuje pracę w tle.

Przykład:

```
OWNER_NAME  JOB_NAME                 OPERATION  JOB_MODE  STATE      DEGREE
----------- ------------------------ ---------- --------- ---------- -------
SYS         SYS_EXPORT_SCHEMA_01     EXPORT     SCHEMA    EXECUTING       1
```

Podłączenie się do działającego zadania: ATTACH

```
expdp NazwaUsera/hasło ATTACH=NazwaZadania
```

Podłączamy się z powrotem do zadania działającego w tle bazy, nawet jeśli sesja terminala, w której je uruchomiliśmy, już dawno się zakończyła — po podłączeniu dostajemy interaktywną konsolę Data Pump z możliwością wydania kolejnych poleceń.

Przykład:

```
Job: SYS_EXPORT_SCHEMA_01
  Owner: SYS
  Operation: EXPORT
  Mode: SCHEMA
  State: EXECUTING

Export>
```

Sprawdzenie statusu z poziomu konsoli zadania: STATUS

```
Export> STATUS
```

Po podłączeniu się do zadania sprawdzamy jego bieżący postęp — ile obiektów przetworzono, ile jeszcze zostało, i czy nie utknęło na którymś kroku.

Przykład:

```
Job: SYS_EXPORT_SCHEMA_01
  Operation: EXPORT
  Mode: SCHEMA
  State: EXECUTING
  Bytes Processed: 3,221,004,288
  Percent Done: 62

Worker 1 Status:
  Object Schema: WMS_APP
  Object Name: ZAMOWIENIA
  Object Type: SCHEMA_EXPORT/TABLE/TABLE_DATA
  Completed Objects: 14
  Total Objects: 22
```

Częste błędy

Brak uprawnień READ/WRITE do directory przed pierwszym uruchomieniem. `expdp`/`impdp` kończy się błędem o braku dostępu, jeśli obiekt directory istnieje, ale konto wykonujące export czy import nie ma do niego nadanych uprawnień fiszką `GRANT READ, WRITE ON DIRECTORY`.

Import bez `TABLE_EXISTS_ACTION` na środowisku, gdzie tabele już istnieją. Domyślne zachowanie po prostu pomija istniejące tabele bez ostrzeżenia wyglądającego na błąd krytyczny — warto zawsze świadomie wybrać `REPLACE`, `APPEND` albo `TRUNCATE`, zamiast polegać na domyślnym `SKIP`.

Pominięcie `REMAP_TABLESPACE` przy imporcie na środowisko z inną strukturą dysków. Jeśli docelowa baza nie ma tablespace o takiej samej nazwie jak źródłowa, import zakończy się błędem, zamiast automatycznie założyć dane gdziekolwiek indziej.

Powiązane: RMAN: backup, walidacja, czyszczenie archiwów · Migracja bazy danych na inny serwer · Tablespace: sprawdzenie, powiększenie, utworzenie, usunięcie

Źródła: notatki własne, dokumentacja Oracle Database (Utilities — Data Pump Export, Data Pump Import)

Revision #43
Created 2026-09-24 23:34:36 UTC by Tomasz Konieczny
Updated 2026-10-26 14:09:06 UTC by Tomasz Konieczny