Backup i odtwarzanie

Tryb ARCHIVELOG, RMAN, Data Pump i odtworzenie z backupu Networkera.

- Tryb ARCHIVELOG i logowanie bazy
- RMAN: backup, walidacja, czyszczenie archiwów
- Data Pump: export i import
- Odtworzenie z backupu Networkera na innym hoście

Tryb ARCHIVELOG i logowanie bazy

Poziom: operacyjny **Dotyczy: Oracle Database 12c–19c** **Czas: ~20 min**

Bez trybu ARCHIVELOG nie ma backupu na gorąco ani odtwarzania do punktu w czasie — instancję da się wtedy odtworzyć wyłącznie do stanu z ostatniego backupu zimnego. To jest przełącznik, który dzieli bazy „produkcyjne" od „takich, które można stracić".

Skróty użyte na tej stronie (5)

Skrót	Co znaczy
FRA	Fast Recovery Area: katalog Oracle na archiwalne logi i kopie RMAN
PITR	Point-In-Time Recovery: odtworzenie bazy do dowolnej chwili, np. 14:31
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

TYLKO ODCZYT

Sprawdzenie bieżącego trybu bazy: v$database

```
SELECT name, log_mode FROM v$database;
```

Sprawdzamy, w jakim trybie działa baza, zanim podejmiemy decyzję o przełączeniu.

Przykład:

```
NAME      LOG_MODE
--------- ------------
```

```
WMSPRD    NOARCHIVELOG
```

□□ TYLKO ODCZYT

Sprawdzenie bieżącego trybu bazy, wersja szczegółowa: ARCHIVE LOG LIST

```
ARCHIVE LOG LIST
```

To samo, w wersji bardziej szczegółowej, poleceniem SQL*Plus.

Przykład:

```
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     412
Current log sequence           414
```

□□ **SQL** = Structured Query Language: język zapytań do relacyjnych baz danych

□□ ZMIENIA SYSTEM

Zamknięcie bazy przed zmianą trybu: SHUTDOWN IMMEDIATE

```
SHUTDOWN IMMEDIATE;
```

Zamykamy bazę.

□□ **shutdown** = Shutdown: wyłącza lub restartuje system po zadanym czasie

□□ ZMIENIA SYSTEM

Zamontowanie bazy bez otwierania: STARTUP MOUNT

```
STARTUP MOUNT;
```

Montujemy ją bez otwierania.

⚠️ ZMIENIA SYSTEM

Przełączenie trybu na ARCHIVELOG

```
ALTER DATABASE ARCHIVELOG;
```

Przełączamy tryb.

⚠️ ZMIENIA SYSTEM

Otwarcie bazy po przełączeniu trybu: ALTER DATABASE OPEN

```
ALTER DATABASE OPEN;
```

Otwieramy bazę do pracy.

Przykład:

```
Database closed.
Database dismounted.
ORACLE instance shut down.
ORACLE instance started.
Database mounted.
Database altered.
Database altered.
```

✅ TYLKO ODCZYT

Weryfikacja trybu po zmianie: v$database

```
SELECT name, log_mode FROM v$database;
```

Przykład:

```
NAME      LOG_MODE
--------- ------------
WMSPRD    ARCHIVELOG
```

□□ TYLKO ODCZYT

Sprawdzenie, czy archiwizacja faktycznie wystartowała: ARCHIVE LOG LIST

```
ARCHIVE LOG LIST
```

Sprawdzamy też, czy archiwizacja faktycznie startuje po pierwszym przełączeniu grupy logów.

□□ ZMIENIA SYSTEM

Powrót do NOARCHIVELOG

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE NOARCHIVELOG;
ALTER DATABASE OPEN;
```

Tą samą sekwencją, tylko z odwrotnym poleceniem trybu.

□□ **shutdown** = Shutdown: wyłącza lub restartuje system po zadanym czasie

□□ ZMIENIA SYSTEM

Uzgodnienie rejestru RMAN po usunięciu plików archiwum: CROSSCHECK i DELETE EXPIRED

```
RMAN> CROSSCHECK ARCHIVELOG ALL;
RMAN> DELETE EXPIRED ARCHIVELOG ALL;
```

Jeżeli przy wyłączaniu archivelog usunęliśmy ręcznie z systemu plików pliki z katalogu archiwum, usuwamy je również z rejestru RMAN — inaczej narzędzie backupu będzie próbowało odwoływać się do plików, których już nie ma, i backup przestanie działać. Pełny opis tych dwóch poleceń RMAN, razem z ich realnym outputem, pokazuje strona RMAN: backup, walidacja, czyszczenie archiwów.

□□ **RMAN** = Recovery Manager: narzędzie Oracle do kopii zapasowych

□□ ZMIENIA SYSTEM

Uruchomienie pełnego logowania: ALTER DATABASE FORCE LOGGING

```
ALTER DATABASE FORCE LOGGING;
```

> **Pojęcie: FORCE LOGGING.** Tryb, w którym baza ignoruje klauzulę `NOLOGGING` ustawioną na poziomie pojedynczych obiektów czy operacji i zawsze zapisuje pełne redo, niezależnie od tego, co poprosił deweloper czy skrypt ładujący dane. Włączamy go na bazach, na których zależy nam na tym, żeby żadna operacja nie ominęła archiwizacji — inaczej fragment danych wstawiony w trybie `NOLOGGING` (opisanym w częstych błędach na stronie Redo logi: grupy, przełączanie, rozmiar) staje się nieodtwarzalny z backupu i archiwum, mimo włączonego ARCHIVELOG.

Przykład:

```
Database altered.
```

□□ TYLKO ODCZYT

Sprawdzenie parametrów FRA: SHOW PARAMETER

```
SHOW PARAMETER db_recovery_file_dest
```

Sprawdzenie bieżących wartości.

Przykład:

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest                string      /oracle/db/wmsprd/fra
db_recovery_file_dest_size           big integer 500G
```

□□ **FRA** = Fast Recovery Area: katalog Oracle na archiwalne logi i kopie RMAN · **show** = FortiOS: wypisuje konfigurację, w podstawowej postaci tylko ustawienia różniące się od domyślnych, a z wariantem full-configuration także wartości domyślne

TYLKO ODCZYT

Sprawdzenie zajętości FRA: v$recovery_file_dest

```
SELECT * FROM v$recovery_file_dest;
```

Sprawdzenie bieżącej zajętości FRA.

Przykład:

```
NAME                    SPACE_LIMIT SPACE_USED SPACE_RECLAIMABLE NUMBER_OF_FILES
----------------------- ----------- ---------- ----------------- ---------------
/oracle/db/wmsprd/fra    536870912000 412047820800        38654705664             214
```

Kiedy FRA zapełnia się archiwum, którego RMAN już nie potrzebuje (bo zostało już zbackupowane i minęła retencja), zwalniamy miejsce poleceniami czyszczącymi opisanymi na stronie RMAN: backup, walidacja, czyszczenie archiwów, zamiast ręcznie kasować pliki z dysku.

FRA = Fast Recovery Area: katalog Oracle na archiwalne logi i kopie RMAN · **RMAN** = Recovery Manager: narzędzie Oracle do kopii zapasowych

Z życia wzięte

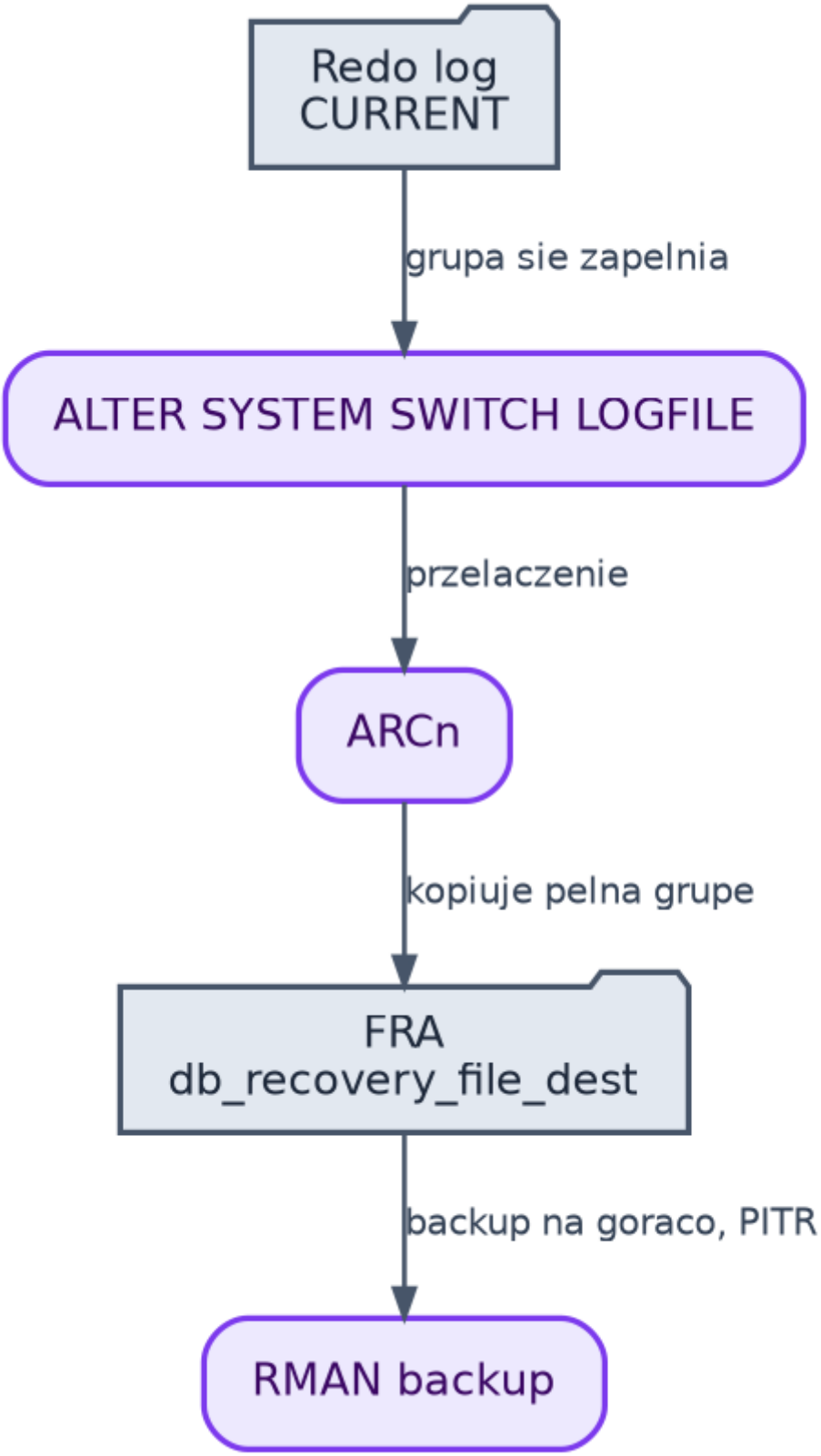
Z ŻYCIA WZIĘTE

Miesiąc po uruchomieniu WMS na ora01 nikt nie zauważył, że baza wciąż pracuje w trybie NOARCHIVELOG. Tomek odkrywa to dopiero, gdy Marek pyta, czy da się odtworzyć zamówienia sprzed wczorajszej awarii dysku, a jedyny backup okazuje się zimny i sprzed tygodnia. Po przełączeniu WMSPRD na ARCHIVELOG i skonfigurowaniu FRA kolejna taka awaria kończy się odtworzeniem do minuty przed problemem, a nie utratą całego tygodnia danych.

Czym jest tryb ARCHIVELOG

> **Pojęcie: tryb ARCHIVELOG.** Ustawienie bazy, w którym Oracle, zanim nadpisze zawartość redo loga podczas kolejnego przełączenia grupy, kopiuje jego pełną zawartość do osobnego

pliku archiwalnego. Bez tego trybu (`NOARCHIVELOG`) redo log jest po prostu nadpisywany w kółko, a historia zmian starsza niż to, co mieszczą aktualne grupy, bezpowrotnie znika.



Redo log zapełnia się, baza przełącza grupę, ARCn kopiuje pełną grupę do FRA, a stamtąd RMAN robi z niej backup na gorąco (kliknięcie otwiera pełny rozmiar)

To właśnie ciągły łańcuch zarchiwizowanych logów pozwala na dwie rzeczy, których bez archivelog zrobić się nie da: backup bazy na gorąco, czyli bez zatrzymywania instancji (opisany ogólnie na stronie Backup pełny, różnicowy, przyrostowy, logi), oraz odtwarzanie do dowolnego punktu w czasie między dwoma backupami pełnymi, czyli PITR (temat strony Odtwarzanie do punktu w czasie (PITR)). Bez archivelog jedyną drogą odtworzenia jest cofnięcie się do stanu z ostatniego backupu zimnego, wykonanego przy zatrzymanej bazie — a wszystko, co wydarzyło się od tamtej pory, przepada.

Przebieg: włączenie trybu ARCHIVELOG

Zanim zaczniemy: przełączenie trybu wymaga zamknięcia bazy i ponownego jej zamontowania bez otwierania — instancja jest przez chwilę niedostępna dla aplikacji. Warto tę zmianę robić w oknie serwisowym. Parametr `log_mode` nie jest ustawieniem trzymanym w SPFILE do ręcznej edycji — przełączamy go wyłącznie poleceniem `ALTER DATABASE`, edycja pliku parametrów tu nic nie da.

Flash Recovery Area (FRA)

> **Pojęcie: FRA (Flash Recovery Area).** Domyślny, scentralizowany katalog, w którym Oracle trzyma zarchiwizowane redo logi, kopie kontrolne i inne pliki związane z odzyskiwaniem, jeśli nie wskazaliśmy jawnie innej lokalizacji archiwum. Rozmiar i lokalizację FRA kontrolują dwa parametry:

Parametr	Znaczenie
`db_recovery_file_dest`	katalog, w którym leży FRA
`db_recovery_file_dest_size`	twardy limit miejsca, jakie FRA może zająć

Częste błędy

Próba przełączenia trybu bez zamknięcia bazy. `ALTER DATABASE ARCHIVELOG` wymaga stanu `MOUNT` bez otwarcia — wykonane na bazie otwartej kończy się błędem.

Ręczne usuwanie plików archiwum z dysku bez aktualizacji RMAN. Prowadzi do niespójności między tym, co RMAN sądzi, że istnieje, a rzeczywistością na dysku — kolejny backup

albo `CROSSCHECK` zwróci błędy o brakujących plikach.

Zapełnienie FRA do granicy `db_recovery_file_dest_size`. Kiedy FRA się zapełni, baza przestaje móc archiwizować kolejne redo logi i w skrajnym przypadku wstrzymuje pracę do czasu zwolnienia miejsca — monitoring zajętości FRA (przez `v$recovery_file_dest`, jak wyżej) warto mieć ustawiony na stałe, nie tylko sprawdzać doraźnie.

Powiązane: Redo logi: grupy, przełączanie, rozmiar · RMAN: backup, walidacja, czyszczenie archiwów · Backup pełny, różnicowy, przyrostowy, logi · Odtwarzanie do punktu w czasie (PITR)

Źródła: notatki własne, dokumentacja Oracle Database (Administrator's Guide — Managing Archived Redo Log Files)

RMAN: backup, walidacja, czyszczenie archiwów

Poziom: operacyjny **Dotyczy: Oracle Database 12c–19c, RMAN** **Czas: ~30 min**

RMAN jest natywnym narzędziem backupu Oracle — zna strukturę bazy od środka, więc potrafi robić kopie na gorąco, przyrostowe co do bloku i samodzielnie pilnować, co jest jeszcze potrzebne, a co można bezpiecznie skasować.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
FRA	Fast Recovery Area: katalog Oracle na archiwalne logi i kopie RMAN
OS	Operating System: system operacyjny
RMAN	Recovery Manager: narzędzie Oracle do kopii zapasowych

Fiszka

TYLKO ODCZYT

Konfiguracja polityki retencji: CONFIGURE RETENTION POLICY

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF LiczbaDni DAYS;
```

Ustawiamy, jak długo RMAN ma uznawać backupy za potrzebne — retencja liczona w oknie odzyskiwania (liczbie dni wstecz, do których musimy móc się odtworzyć) zamiast prostej liczby backupów. Backupy starsze niż to okno, o ile nie są jeszcze potrzebne łańcuchowi przyrostowemu, RMAN oznaczy jako `OBSOLETE`.

Przykład:

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 14 DAYS;
```

```
new RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 14 DAYS;
new RMAN configuration parameters are successfully stored
```

RMAN = Recovery Manager: narzędzie Oracle do kopii zapasowych

TYLKO ODCZYT

Konfiguracja automatycznego backupu pliku kontrolnego: CONFIGURE CONTROLFILE AUTOBACKUP

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
```

Włączamy automatyczny backup pliku kontrolnego i SPFILE po każdej operacji backupu oraz po każdej zmianie strukturalnej bazy (dodanie tablespace, datafile) — bez tego ustawienia odtworzenie pliku kontrolnego z backupu na innym serwerze byłoby dużo trudniejsze, bo trzeba by znać dokładny backup piece, w którym ostatnio wylądował.

Przykład:

```
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

Format nazwy pliku autobackupu na urządzeniu taśmowym ustawiamy osobno — przydaje się to przy odtwarzaniu na innym hoście, opisanym na stronie Odtworzenie z backupu Networkera na innym hoście:

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE 'SBT_TAPE' TO
'NazwaBazy_CONTROLFILE_%F';
```

SBT_TAPE = Sequential Backup Tape: interfejs, przez który RMAN zapisuje backup na bibliotece taśmowej innego producenta zamiast na dysk

TYLKO ODCZYT

Konfiguracja urządzenia backupu: CONFIGURE DEFAULT DEVICE TYPE

```
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
```

Ustawiamy, gdzie RMAN domyślnie zapisuje backupy — `DISK` przy kopiach lokalnych na dysk lub udział sieciowy, `SBT_TAPE` przy korzystaniu z biblioteki backupowej innego producenta (na przykład Networkera, jak w procedurze na stronie Odtworzenie z backupu Networkera na innym hoście).

Przykład:

```
new RMAN configuration parameters:
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
new RMAN configuration parameters are successfully stored
```

RMAN = Recovery Manager: narzędzie Oracle do kopii zapasowych · **SBT_TAPE** = Sequential Backup Tape: interfejs, przez który RMAN zapisuje backup na bibliotece taśmowej innego producenta zamiast na dysk

ZMIENIA SYSTEM

Backup pełnej bazy razem z archiwum: BACKUP DATABASE PLUS ARCHIVELOG

```
BACKUP DATABASE PLUS ARCHIVELOG;
```

Robimy pełny backup wszystkich plików danych razem z bieżącym łańcuchem zarchiwizowanych redo logów w jednym poleceniu — najprostszy sposób na kompletną kopię bazy gotową do odtworzenia od zera na dowolnym punkcie w czasie objętym backupem.

Przykład:

```
Starting backup at 25-SEP-26
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: finished piece 1 at 25-SEP-26, elapsed time: 00:14:22
Finished backup at 25-SEP-26

Starting backup at 25-SEP-26
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: finished piece 1 at 25-SEP-26, elapsed time: 00:01:08
Finished backup at 25-SEP-26
```

TYLKO ODCZYT

Backup przyrostowy poziomu: BACKUP INCREMENTAL LEVEL

```
BACKUP INCREMENTAL LEVEL LiczbaPoziomu DATABASE;
```

Robimy backup tylko zmienionych bloków względem ostatniego backupu na poziomie 0 (pełnego) — poziom 0 zakłada kopię bazową, każdy kolejny backup poziomu 1 kopiuje wyłącznie bloki zmienione od ostatniego backupu tego samego lub niższego poziomu.

Przykład:

```
BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

```
Starting backup at 25-SEP-26
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: finished piece 1 at 25-SEP-26, elapsed time: 00:03:40
Finished backup at 25-SEP-26
```

TYLKO ODCZYT

Walidacja poprawności backupu bazy: VALIDATE DATABASE

```
VALIDATE DATABASE;
```

Sprawdzamy, czy pliki danych i backupy są fizycznie czytelne i nie mają uszkodzonych bloków — bez faktycznego odtwarzania, tylko weryfikacja spójności. Uruchamiamy okresowo, żeby wykryć uszkodzenie backupu, zanim będzie naprawdę potrzebny.

Przykład:

```
Starting validate at 25-SEP-26
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: containing 1187442792 blocks
List of Datafiles
=================
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
---- ------ -------------- ------------ --------------- ----------
```

```
1    OK     0              14231         1184890880      168534717
Finished validate at 25-SEP-26
```

TYLKO ODCZYT

Walidacja i wyczyszczenie rejestru archive logów: CROSSCHECK ARCHIVELOG ALL

```
CROSSCHECK ARCHIVELOG ALL;
```

Sprawdzamy zgodność tego, co RMAN sądzi, że ma na dysku czy taśmie, z tym, co tam faktycznie jest — pliki, których fizycznie brakuje (na przykład usunięte ręcznie po wyłączeniu archivelog), oznaczane są jako `EXPIRED`. Wykonujemy to zawsze przed właściwym czyszczeniem, żeby rejestr RMAN odzwierciedlał rzeczywistość.

Przykład:

```
allocated channel: ORA_DISK_1
crosschecked archived log
archived log file name=/oracle/db/wmsprd/oraarch/1_412_1187442792.arc RECID=890
STAMP=1187512033
validation succeeded for specified archived log
crosschecked archived log
archived log file name=/oracle/db/wmsprd/oraarch/1_398_1187442792.arc RECID=876
STAMP=1187498211
validation failed for specified archived log
Crosschecked 214 objects
```

RMAN = Recovery Manager: narzędzie Oracle do kopii zapasowych

TYLKO ODCZYT

Usunięcie backupów i archiwów oznaczonych jako EXPIRED: DELETE EXPIRED

```
DELETE EXPIRED ARCHIVELOG ALL;
```

Kasujemy z rejestru RMAN te wpisy, które `CROSSCHECK` oznaczył jako brakujące fizycznie — samych plików to polecenie nie usuwa (nie ma ich już na dysku), tylko porządkuje katalog

RMAN, żeby przestał odwoływać się do nieistniejących obiektów.

Przykład:

```
Deleting the following obsolete archived logs
archived log file name=/oracle/db/wmsprd/oraarch/1_398_1187442792.arc RECID=876
STAMP=1187498211
Deleted 1 EXPIRED objects
```

RMAN = Recovery Manager: narzędzie Oracle do kopii zapasowych

TYLKO ODCZYT

Usunięcie zarchiwizowanych logów starszych niż podana data: DELETE NOPROMPT ARCHIVELOG ALL COMPLETED BEFORE

```
DELETE NOPROMPT ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-LiczbaDni';
```

Kasujemy fizycznie i z rejestru archiwa starsze niż podana liczba dni, bez pytania o potwierdzenie każdego pliku — RMAN sam pilnuje, żeby nie skasować logu, który jest jeszcze potrzebny nieukończonemu backupowi przyrostowemu.

Przykład:

```
DELETE NOPROMPT ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7';
```

```
deleted archived log
archived log file name=/oracle/db/wmsprd/oraarch/1_401_1187442792.arc RECID=879
STAMP=1187499001
Deleted 38 EXPIRED objects
```

RMAN = Recovery Manager: narzędzie Oracle do kopii zapasowych

TYLKO ODCZYT

Wylistowanie podsumowania backupów: LIST BACKUP SUMMARY

```
LIST BACKUP SUMMARY;
```

Sprawdzamy skróconą historię wszystkich backupów widocznych RMAN-owi — typ, rozmiar, urządzenie, datę zakończenia — pierwsze polecenie, kiedy chcemy się zorientować, co w

ogóle jest dostępne do odtworzenia.

Przykład:

```
List of Backups
===============
Key     TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------- -- -- - ----------- --------------- ------- ------- ---------- ---
1024    B  F  A DISK        24-SEP-26       1       1       NO
TAG20260924T030000
1031    B  1  A DISK        25-SEP-26       1       1       NO
TAG20260925T030000
1032    B  A  A DISK        25-SEP-26       1       1       NO
TAG20260925T030512
```

□□ TYLKO ODCZYT

Wyszukanie backupów przeterminowanych: REPORT OBSOLETE

```
REPORT OBSOLETE;
```

Sprawdzamy, które backupy nie mieszczą się już w skonfigurowanej polityce retencji i można je bezpiecznie usunąć — samo `REPORT` tylko pokazuje listę, nic nie kasuje.

Przykład:

```
RMAN retention policy is set to recovery window of 14 days
Report of obsolete backups and copies
Type     Key    Completion Time Filename/Handle
-------- ------ --------------- --------------------------------------------
Backup Set    1002   05-SEP-26       /backup/wmsprd/1002_1_full.bck
Backup Set    1009   12-SEP-26       /backup/wmsprd/1009_1_incr.bck
```

□□ TYLKO ODCZYT

Usunięcie backupów przeterminowanych: DELETE OBSOLETE

```
DELETE OBSOLETE;
```

Fizycznie kasujemy z dysku (albo z biblioteki taśmowej) i z rejestru wszystkie backupy zgłoszone przez `REPORT OBSOLETE` jako niepotrzebne wobec aktualnej polityki retencji.

Przykład:

```
Deleting the following obsolete backups and copies:
Type     Key    Completion Time Filename/Handle
-------- ------ --------------- ----------------------------------------
Backup Set    1002   05-SEP-26       /backup/wmsprd/1002_1_full.bck
Backup Set    1009   12-SEP-26       /backup/wmsprd/1009_1_incr.bck
deleted backup
Deleted 2 objects
```

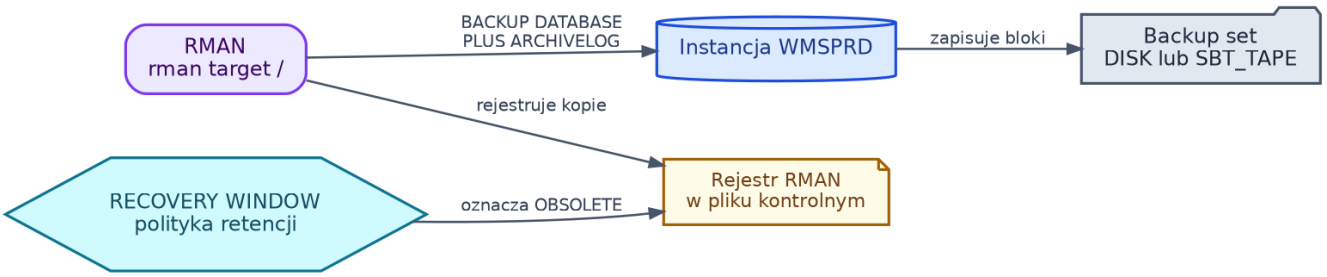
Z życia wzięte

□□ Z ŻYCIA WZIĘTE

W nocy zaplanowany backup RMAN bazy WMSPRD kończy się błędem, bo katalog FRA na ora01 jest pełny starymi kopiami przeterminowanymi wg polityki retencji. Tomek uruchamia `CROSSCHECK` i `DELETE OBSOLETE`, żeby uporządkować rejestr, zanim ponowi `BACKUP DATABASE PLUS ARCHIVELOG`. Rano Marek nawet nie wie, że coś się działo, bo backup domyka się przed startem zmiany o 6:00.

Czym jest RMAN

Pojęcie: RMAN (Recovery Manager). Narzędzie wbudowane w Oracle Database, które zarządza całym cyklem backupu i odtwarzania — zna wewnętrzną strukturę bazy (pliki danych, redo logi, plik kontrolny) i komunikuje się z instancją bezpośrednio, zamiast kopiować pliki na ślepo z poziomu systemu operacyjnego. Dzięki temu RMAN potrafi robić backup przy otwartej bazie bez ryzyka niespójności, robić kopie przyrostowe na poziomie pojedynczych zmienionych bloków, a nie całych plików, oraz sam pilnować retencji — wie, które backupy są jeszcze potrzebne, a które można skasować.



RMAN łączy się z instancją WMSPRD, zapisuje backup set na dysku albo taśmie i rejestruje kopię w swoim rejestrze, gdzie polityka retencji oznacza stare backupy jako OBSOLETE (kliknięcie otwiera pełny rozmiar)

Różnica względem zwykłego kopiowania plików systemowym `cp` czy `rsync` jest zasadnicza: kopia plikowa zrobiona na żywej bazie bez odpowiedniej sekwencji poleceń `ALTER TABLESPACE BEGIN/END BACKUP` daje niespójny obraz danych, natomiast RMAN sam dba o tę spójność i o rejestrowanie backupu w katalogu, z którego później korzysta przy odtwarzaniu.

Połączenie do RMAN

```
rman target /
```

Łączymy się do RMAN lokalnie, jako użytkownik systemowy z uprawnieniami do instancji (zwykle `oracle`), autoryzując się przez OS zamiast podawać hasło.

Przykład:

```
Recovery Manager: Release 19.0.0.0.0 - Production on Fri Sep 25 09:12:03 2026
Copyright (c) 1982, 2019, Oracle and/or its affiliates.  All rights reserved.

connected to target database: WMSPRD (DBID=3032243418)

RMAN>
```

Sprawdzenie historii i postępu zadań RMAN

Sprawdzenie historii zakończonych zadań: v$rman_backup_job_details

```
SELECT session_key, input_type, status,
       TO_CHAR(start_time,'mm/dd/yy hh24:mi') start_time,
       TO_CHAR(end_time,'mm/dd/yy hh24:mi') end_time,
       ROUND(elapsed_seconds/3600, 2) hrs
FROM v$rman_backup_job_details
ORDER BY session_key;
```

Sprawdzamy, kiedy i jak długo trwały poprzednie zadania backupu, oraz czy zakończyły się powodzeniem — pierwsze miejsce, kiedy podejrzewamy, że zaplanowany backup w ogóle się nie odpalił albo skończył się błędem.

Przykład:

```
SESSION_KEY INPUT_TYPE  STATUS      START_TIME       END_TIME         HRS
----------- ----------- ----------- ---------------- ---------------- -----
        118 DB FULL     COMPLETED   09/24/26 03:00   09/24/26 03:41    .69
        126 DB INCR     COMPLETED   09/25/26 03:00   09/25/26 03:04    .07
```

Sprawdzenie postępu zadania aktualnie w trakcie wykonywania: v$session_longops

```
SELECT sid, serial#, context, sofar, totalwork,
       ROUND(sofar/totalwork*100, 2) "% COMPLETE"
FROM v$session_longops
WHERE opname LIKE 'RMAN%'
  AND opname NOT LIKE '%aggregate%'
  AND totalwork != 0
  AND sofar <> totalwork;
```

Sprawdzamy procentowy postęp backupu albo odtworzenia, które właśnie trwa — przydaje się przy długich operacjach na dużych bazach, kiedy chcemy wiedzieć, ile jeszcze zostało, zamiast czekać w ciemno.

Przykład:

```
   SID  SERIAL#    CONTEXT      SOFAR   TOTALWORK  % COMPLETE
------- --------- --------- ---------- ---------- ----------
    412      8821         1     184320      921600       20.0
```

Sprawdzenie logów bieżących operacji: gv$rman_output

```
SELECT output FROM gv$rman_output
WHERE session_recid = &SESSION_RECID
  AND session_stamp = &SESSION_STAMP
ORDER BY recid;
```

Podglądamy surowy log RMAN wybranej sesji bezpośrednio z bazy, bez dostępu do konsoli, na której backup był uruchomiony — `session_recid` i `session_stamp` bierzemy z poprzedniego zapytania po `v$rman_backup_job_details`.

Przykład:

```
OUTPUT
------------------------------------------------------------------
Starting backup at 25-SEP-26
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: finished piece 1, elapsed time: 00:14:22
Finished backup at 25-SEP-26
```

Częste błędy

Uruchomienie DELETE bez wcześniejszego CROSSCHECK. Rejestr RMAN i rzeczywistość na dysku rozjeżdżają się z czasem (ręczne usunięcia, awarie storage'u) — `CROSSCHECK` zawsze poprzedza właściwe czyszczenie, inaczej `DELETE EXPIRED` nie znajdzie nic do usunięcia, mimo że w rejestrze wisi mnóstwo martwych wpisów.

Traktowanie `REPORT OBSOLETE` i `DELETE OBSOLETE` jako czyszczenia archive logów. Te polecenia dotyczą backupów, nie samych zarchiwizowanych redo logów — do archiwum służy osobna para `CROSSCHECK ARCHIVELOG` / `DELETE ARCHIVELOG`.

Zbyt agresywna retencja przy aktywnym łańcuchu przyrostowym. Skrócenie okna retencji nie usunie backupu, który wciąż jest podstawą dla nowszego backupu przyrostowego — RMAN sam to pilnuje, ale warto o tym pamiętać, planując politykę retencji, żeby nie liczyć na miejsce, które i tak się nie zwolni.

Powiązane: Tryb ARCHIVELOG i logowanie bazy · Odtworzenie z backupu Networkera na innym hoście · Backup pełny, różnicowy, przyrostowy, logi · Migracja bazy danych na inny serwer

Źródła: notatki własne, dokumentacja Oracle Database (Backup and Recovery User's Guide)

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

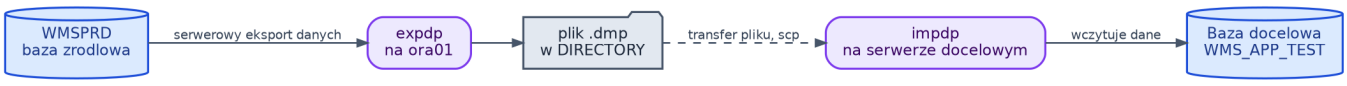
Z życia wzięte

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)

Odtworzenie z backupu Networkera na innym hoście

Typ: procedura **Dotyczy: Oracle Database 19c, RMAN, Dell EMC NetWorker (NMDA)**

Czas: ~2–4 godz.

Efekt: baza `WMSPRD` odtworzona z backupu zarejestrowanego w systemie NetWorker, uruchomiona na nowym hoście (w przykładach `NazwaNowegoSerwera`) pod tym samym DBID co baza źródłowa, z poprawnymi ścieżkami do plików danych i otwarta poleceniem `RESETLOGS`.

Skróty użyte na tej stronie (4)

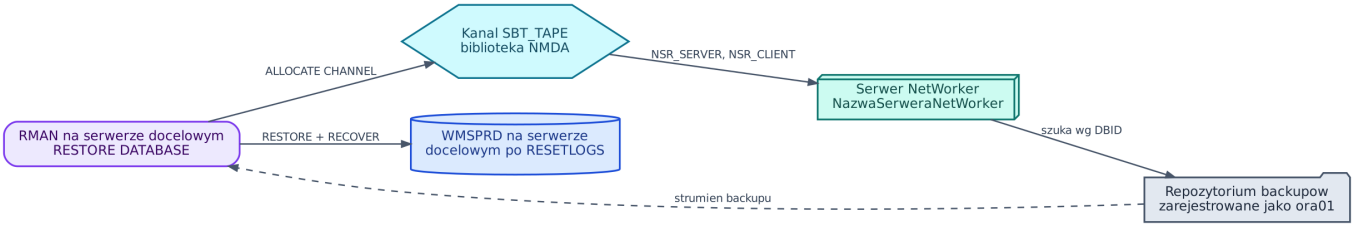
Skrót	Co znaczy
DBID	Database Identifier: unikalny numer bazy Oracle, przydatny przy rozróżnianiu dwóch baz o tej samej nazwie
RMAN	Recovery Manager: narzędzie Oracle do kopii zapasowych
SBT_TAPE	Sequential Backup Tape: interfejs, przez który RMAN zapisuje backup na bibliotece taśmowej innego producenta zamiast na dysk
WMS	Warehouse Management System: system zarządzania magazynem

Zanim zaczniemy

Z ŻYCIA WZIĘTE

Po awarii dysku na ora01 zespół musi odtworzyć WMSPRD na zapasowym serwerze, a jedyny backup leży w systemie NetWorker, nie na lokalnym dysku. Tomek konfiguruje kanał SBT_TAPE z biblioteką NMDA i ustawia `NSR_CLIENT` na nazwę starego serwera, bo backup był rejestrowany pod nią, nie pod nazwą nowego serwera. Po `RESTORE CONTROLFILE`, `RESTORE DATABASE` i otwarciu bazy `RESETLOGS` Marek widzi WMS znowu online, tym razem na nowym adresie.

Backup bazy `WMSPRD` leży nie na dysku lokalnym, tylko w systemie NetWorker — RMAN nie widzi go bezpośrednio, tylko przez kanał typu `SBT_TAPE`, obsługiwany przez bibliotekę mediów (NMDA), którą trzeba mieć zainstalowaną i widoczną w `LD_LIBRARY_PATH` na docelowym serwerze.



RMAN na serwerze docelowym alokuje kanał SBT_TAPE przez bibliotekę NMDA, łączy się z serwerem NetWorker i stamtąd odtwarza backup zarejestrowany pod nazwą serwera źródłowego (kliknięcie otwiera pełny rozmiar)

Wymaganie	Uwaga
Baza o docelowej nazwie istnieje na `NazwaNowegoSerwera`	wystarczy pusta instancja z tym samym `ORACLE_SID` co źródłowa — restore i tak nadpisze jej zawartość
Biblioteka NMDA (NetWorker Module for Database Applications) zainstalowana na `NazwaNowegoSerwera`	RMAN łączy się z NetWorkerem właśnie przez nią, udając urządzenie taśmowe `SBT_TAPE`
Znany DBID bazy źródłowej	RMAN musi wiedzieć, którego backupu w repozytorium NetWorkera szukać, zanim jeszcze ma dostęp do lokalnego pliku kontrolnego
Ścieżki plików danych na `ora01` i `NazwaNowegoSerwera`	jeśli różnią się (inny layout dysków), trzeba przygotować mapowanie `SET NEWNAME` dla każdego datafile
Dostęp sieciowy z `NazwaNowegoSerwera` do serwera NetWorkera (`NazwaSerweraNetWorker`)	bez niego kanał `SBT_TAPE` nie połączy się z repozytorium

> (gdzie `NazwaNowegoSerwera` to serwer docelowy, na który odtwarzamy bazę, bo Zielona ma dziś jeden serwer Oracle, `ora01`, a `NazwaSerweraNetWorker` to pełna nazwa serwera NetWorkera; w Zielonej `bkp01` nie jest serwerem NetWorker, tylko serwerem kopii restic)

Uważaj: `ALTER DATABASE OPEN RESETLOGS` na końcu procedury zeruje sekwencję redo logów bezpowrotnie — od tego momentu stara historia archiwum sprzed odtworzenia przestaje pasować do nowej gałęzi bazy. Backup wykonany po tej operacji trzeba traktować jako nową linię czasu.

Przebieg

1. Przygotowanie środowiska na serwerze docelowym

Logujemy się na `NazwaNowegoSerwera` na konto systemowe `oracle`, wybieramy poprawny `ORACLE_SID` (baza o docelowej nazwie musi już istnieć na serwerze docelowym — wystarczy pusta) i dopinamy bibliotekę mediów NetWorkera do ścieżki wyszukiwania bibliotek:

```
sudo su - oracle
. oraenv
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/lib64/:${LD_LIBRARY_PATH}
export NLS_DATE_FORMAT="DD-MM-YYYY HH24:MI:SS"
```

Ustawienie `NLS_DATE_FORMAT` jest istotne, bo klauzula `SET UNTIL TIME` w dalszych krokach parsuje datę zgodnie z tym formatem — bez tego RMAN odczyta datę błędnie albo w ogóle jej nie przyjmie.

2. Uruchomienie RMAN i wstępne uruchomienie instancji

```
rman target /
```

Instancja na serwerze docelowym najczęściej nie ma jeszcze poprawnego pliku parametrów — startujemy mimo to, RMAN sam poradzi sobie z brakiem `pfile`/`spfile` na tym etapie:

```
RMAN> STARTUP NOMOUNT;
```

Przykład:

```
startup failed: ORA-01078: failure in processing system parameters
LRM-00109: could not open parameter file '/oracle/db/21c/dbs/initwmsprd.ora'
starting Oracle instance without parameter file for retrieval of spfile
ORACLE instance started

Total System Global Area    7180646608 bytes
Fixed Size                     8956112 bytes
Variable Size               1191182336 bytes
Database Buffers            5972688896 bytes
```

```
Redo Buffers                    7819264 bytes
```

3. Ustawienie DBID bazy źródłowej

Podmieniamy DBID instancji na ten z bazy, którą odtwarzamy — RMAN bez pliku kontrolnego nie wie jeszcze, jakiej bazy szuka w repozytorium, więc musimy mu to powiedzieć jawnie:

```
RMAN> SET DBID NumerDBID;
```

Przykład:

```
RMAN> set dbid 3032243418;
```

```
executing command: SET DBID
```

4. Konfiguracja kanału SBT_TAPE do NetWorkera

Ustawiamy domyślne urządzenie backupu na taśmowe (obsługiwane przez NetWorker) i konfigurujemy kanał, wskazując serwer i klienta NetWorkera:

```
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO 'SBT_TAPE';
RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' SEND
'NSR_ENV=(NSR_SERVER=NazwaSerwera,NSR_CLIENT=NazwaSerwera)';
```

`NSR_SERVER` to serwer NetWorkera, na którym leży repozytorium backupów, `NSR_CLIENT` to nazwa klienta, pod jaką backup został zarejestrowany — zwykle nazwa serwera źródłowego, z którego backup pierwotnie powstał, a nie serwera, na którym teraz odtwarzamy.

Przykład:

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' SEND
'NSR_ENV=(NSR_SERVER=NazwaSerweraNetWorker,NSR_CLIENT=ora01.zielona.example)';
```

```
new RMAN configuration parameters are successfully stored
```

Ustawiamy też format nazwy pliku autobackupu pliku kontrolnego dla tego samego typu urządzenia — potrzebny w kolejnym kroku do znalezienia backupu pliku kontrolnego:

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE 'SBT_TAPE' TO
'NazwaBazy_CONTROLFILE_%F';
```

5. Odtworzenie pliku kontrolnego

Bez pliku kontrolnego RMAN nie zna struktury bazy, więc odtwarzamy go jako pierwszy, alokując kanał ręcznie w bloku `RUN`:

```
RMAN> RUN {
2>   ALLOCATE CHANNEL ch1 TYPE 'SBT_TAPE';
3>   SEND 'NSR_ENV=(NSR_SERVER=NazwaSerwera,NSR_CLIENT=NazwaSerwera)';
4>   SET CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE 'SBT_TAPE' TO
'NazwaBazy_CONTROLFILE_%F';
5>   RESTORE CONTROLFILE FROM AUTOBACKUP;
6> }
```

Przykład:

```
allocated channel: ch1
channel ch1: SID=257 device type=SBT_TAPE
channel ch1: NMDA Oracle v19.10.0.3
sent command to channel: ch1
executing command: SET CONTROLFILE AUTOBACKUP FORMAT

Starting restore at 25-SEP-26
channel ch1: restoring control file
channel ch1: restore complete, elapsed time: 00:00:07
output file name=/oracle/db/wmsprd/oradata/WMSPRD/control01.ctl
output file name=/oracle/db/wmsprd/oraarch/WMSPRD/control02.ctl
Finished restore at 25-SEP-26
```

Jeśli znamy dokładny identyfikator backup piece pliku kontrolnego (na przykład z logów poprzedniego backupu), możemy wskazać go wprost zamiast `FROM AUTOBACKUP`:

```
RESTORE CONTROLFILE FROM 'NazwaBazy_xxxxxxxx_1_1';
```

Montujemy bazę na świeżo odtworzonym pliku kontrolnym:

```
RMAN> STARTUP MOUNT;
```

6. Porównanie ścieżek plików danych między serwerami

Jeśli układ dysków na `NazwaNowegoSerwera` różni się od `ora01`, sprawdzamy na serwerze źródłowym rzeczywiste ścieżki, żeby wiedzieć, co i na co zmieniać:

```
SELECT file_id, tablespace_name, file_name
FROM dba_data_files
ORDER BY file_id;
```

Przykład (na serwerze źródłowym):

```
FILE_ID TABLESPACE_NAME FILE_NAME
------- --------------- ----------------------------------------------------
      1 SYSTEM          /oracle/db/wmsprd/oradata/WMSPRD/system01.dbf
      2 SYSAUX          /oracle/db/wmsprd/oradata/WMSPRD/sysaux01.dbf
      3 UNDOTBS1        /oracle/db/wmsprd/oradata/WMSPRD/undotbs01.dbf
      4 USERS           /oracle/db/wmsprd/oradata/WMSPRD/users01.dbf
      5 KOLEKTOR        /oracle/db/wmsprd/oradata/WMSPRD/kolektor.dbf
```

Porównujemy z docelowym układem katalogów na `NazwaNowegoSerwera` i, jeśli ścieżki mają się różnić, przygotowujemy dla każdego pliku polecenie `SET NEWNAME`.

7. Odtworzenie bazy z mapowaniem nowych ścieżek i punktem w czasie

Jeżeli chcemy odtworzyć bazę nie do ostatniego backupu, tylko do konkretnego momentu sprzed jakiegoś zdarzenia, dodajemy `SET UNTIL TIME`. Cały blok wykonujemy jako jedno polecenie `RUN`:

```
RMAN> RUN {
2>   ALLOCATE CHANNEL ch1 TYPE 'SBT_TAPE';
3>   SEND 'NSR_ENV=(NSR_SERVER=NazwaSerwera,NSR_CLIENT=NazwaSerwera)';
4>   SET UNTIL TIME "TO_DATE('DataICzas', 'yyyy-mm-dd hh24:mi:ss')";
5>   SET NEWNAME FOR DATAFILE 1 TO 'SciezkaDoPliku';
6>   SET NEWNAME FOR DATAFILE 2 TO 'SciezkaDoPliku';
7>   SET NEWNAME FOR DATAFILE 3 TO 'SciezkaDoPliku';
8>   -- analogicznie SET NEWNAME dla każdego kolejnego datafile z listy z kroku 6
9>   RESTORE DATABASE;
```

```
10>  SWITCH DATAFILE ALL;
11>  RECOVER DATABASE;
12> }
```

Przykład (fragment, dla trzech pierwszych plików bazy `WMSPRD`):

```
SET UNTIL TIME "TO_DATE('2026-09-24 06:00:00', 'yyyy-mm-dd hh24:mi:ss')";
SET NEWNAME FOR DATAFILE 1 TO '/oracle/db/wmsprd/oradata/WMSPRD/SYSTEM.DBF';
SET NEWNAME FOR DATAFILE 2 TO '/oracle/db/wmsprd/oradata/WMSPRD/SYSAUX.DBF';
SET NEWNAME FOR DATAFILE 3 TO '/oracle/db/wmsprd/oradata/WMSPRD/UNDOTBS1.DBF';
```

```
executing command: SET until clause
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=250 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: NMDA oracle v19.10.0.3
allocated channel: ORA_DISK_1

Starting restore at 25-SEP-26
starting datafile backup set restore
restoring datafile 00001 to /oracle/db/wmsprd/oradata/WMSPRD/SYSTEM.DBF
restoring datafile 00002 to /oracle/db/wmsprd/oradata/WMSPRD/SYSAUX.DBF
restoring datafile 00003 to /oracle/db/wmsprd/oradata/WMSPRD/UNDOTBS1.DBF
reading from backup piece WMSPRD_933bq6ir_1_1
Finished restore at 25-SEP-26

datafile 1 switched to datafile copy
input datafile copy RECID=10 STAMP=1187442792 file
name=/oracle/db/wmsprd/oradata/WMSPRD/SYSTEM.DBF
datafile 2 switched to datafile copy
datafile 3 switched to datafile copy

Starting recover at 25-SEP-26
media recovery complete, elapsed time: 00:02:14
Finished recover at 25-SEP-26
```

`SWITCH DATAFILE ALL` przełącza plik kontrolny na wskazywanie odtworzonych kopii jako właściwych plików bazy, zamiast robić pełną, dodatkową kopię fizyczną — szybszy wariant niż klasyczny `RESTORE` + fizyczne kopiowanie.

8. Odtworzenie plików tymczasowych

Tempfile nie jest częścią backupu — mapujemy jego docelową ścieżkę i przełączamy osobno, już po otwarciu bazy:

```
SET NEWNAME FOR TEMPFILE 'SciezkaDoPliku' TO 'SciezkaDoNowegoPliku';
SWITCH TEMPFILE ALL;
```

Przykład:

```
SET NEWNAME FOR TEMPFILE '/oracle/db/wmsprd/oradata/WMSPRD/datafile/o1_mf_temp_l4hn1ygv_.tmp'
TO '/oracle/db/wmsprd/oradata/WMSPRD/TEMP01.dbf';
SWITCH TEMPFILE ALL;
```

```
executing command: SET NEWNAME
renamed tempfile 1 to /oracle/db/wmsprd/oradata/WMSPRD/TEMP01.dbf in control file
```

9. Otwarcie bazy z RESETLOGS

```
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

Otwieramy bazę, zerując sekwencję redo logów — konieczne po odtworzeniu przy pomocy `SET UNTIL TIME`, bo baza jest teraz w innym punkcie historii niż to, co pamiętał plik kontrolny sprzed odtworzenia.

Przykład:

```
database opened
```

Weryfikacja

Sprawdzamy, że instancja działa i że jest to właściwa baza pod właściwym DBID:

```
SELECT name, dbid, open_mode, log_mode FROM v$database;
```

Przykład:

```
NAME          DBID       OPEN_MODE            LOG_MODE
--------- ---------- -------------------- ------------
WMSPRD    3032243418 READ WRITE           ARCHIVELOG
```

Sprawdzamy, że wszystkie tablespace są dostępne i online:

```
SELECT tablespace_name, status FROM dba_tablespaces;
```

Sprawdzamy w aplikacji korzystającej z bazy, że dane po odtworzeniu wyglądają na kompletne i spójne w punkcie czasowym, do którego celowaliśmy.

Wycofanie zmian

Jeżeli po otwarciu bazy okaże się, że odtworzenie trafiło w zły punkt w czasie albo dane są niekompletne, najbezpieczniej jest zacząć procedurę od nowa: zamykamy instancję (`SHUTDOWN IMMEDIATE`), czyścimy katalog z plikami odtworzonymi w poprzedniej próbie i wracamy do kroku 2 z poprawioną wartością `SET UNTIL TIME`. Nie próbujemy korygować częściowo otwartej bazy po `RESETLOGS` — to nowa gałąź bazy i nie da się jej cofnąć bez ponownego pełnego odtworzenia.

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`ORA-01078` / `LRM-00109` przy `STARTUP NOMOUNT`	brak pliku parametrów na serwerze docelowym	oczekiwane przy pierwszym uruchomieniu — RMAN i tak wystartuje bez pliku i odtworzy `spfile` z backupu w dalszych krokach
RMAN nie widzi backupu mimo poprawnego `SET DBID`	zły `NSR_CLIENT` w konfiguracji kanału	`NSR_CLIENT` musi wskazywać nazwę serwera źródłowego, pod jaką backup był rejestrowany w NetWorkerze, nie serwera docelowego
Kanał `SBT_TAPE` nie łączy się z NetWorkerem	biblioteka NMDA nie jest widoczna w `LD_LIBRARY_PATH`	sprawdzamy, czy pakiet NMDA jest zainstalowany, i poprawiamy `LD_LIBRARY_PATH` przed ponownym uruchomieniem `rman`
`RESTORE DATABASE` zgłasza brak miejsca na docelowym dysku	ścieżki z `SET NEWNAME` wskazują wolumen mniejszy niż źródłowy	sprawdzamy `df -h` na serwerze docelowym przed startem restore, nie w trakcie
Baza nie otwiera się po `RESETLOGS`, zgłasza niespójność datafile	pominięty którykolwiek plik w `SET NEWNAME` przy różnych ścieżkach między serwerami	wracamy do kroku 6, porównujemy pełną listę `dba_data_files` ze źródła z listą faktycznie odtworzonych plików

Powiązane: RMAN: backup, walidacja, czyszczenie archiwów · Pliki kontrolne · Migracja bazy danych na inny serwer · Odtwarzanie do punktu w czasie (PITR)

Źródła: notatki własne, dokumentacja Oracle RMAN, dokumentacja Dell EMC NetWorker Module for Database Applications (NMDA)