

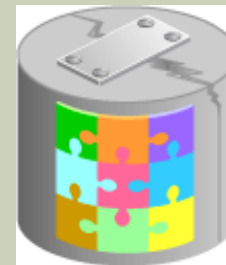
# ZÁLOHOVÁNÍ

# CO JE MŮJ JOB

- DBA musí:
  - Uchránit databázi proti všem poruchám ... pokud možno ...
  - Zvětšovat the mean time mezi poruchami (MTBF)
  - Zajistit redundanci kritických komponent
  - Zmenšovat the mean time pro obnovení (MTTR)
  - Minimalizovat jakoukoliv ztrátu dat

# KATEGORIE PORUCH

- Poruchy mohou být generalizovány zhruba do následujících kategorií:
  - Chyba příkazu
  - Selhání uživatelského procesu
  - Síťová porucha
  - Uživatelská chyba
  - Porucha instance
  - Porucha média



# CHYBA PŘÍKAZU

Typický problém	Možné řešení
Vkládání neplatných dat	Korekce na straně uživatele
Pokus o operaci, na kterou nemá uživatel právo	Získání patřičných práv
Neúspěšný pokus o alokaci místa	<ul style="list-style-type: none"><li>• Povolit resumable space allocation.</li><li>• Zvětšit kvótu.</li><li>• Zvětšit tablespace.</li></ul>
Logické chyby v aplikaci	Korekce na straně vývojářů

# SELHÁNÍ UŽIVATELSKÉHO PROCESU

Typický problém	Možné řešení
Odhlášení uživatele nestandardním způsobem	Zásah administrátora není většinou nutný, databáze automaticky odroluje nepotvrzené změny. Může jít o jednorázový problém nebo problém může mít hlubší příčinu. Je nutné dlouhodobější sledování
Nestandardní ukončení připojení do databáze	
Programová chyba, která připojení ukončí	



# SÍŤOVÁ PORUCHA

Typický problém	Možné řešení
Porucha listeneru	Konfigurace záložního listeneru
Network Interface Card (NIC) porucha	Konfigurace pro více adres
Porucha síťového připojení	Záložní síťové připojení

# UŽIVATELSKÁ CHYBA

Typické příčiny	Možné řešení
Chyba při změně dat nebo jejich smazání	Odrolování změn nebo použití flashbacku
Zrušení tabulky	Obnova z odpadkového koše

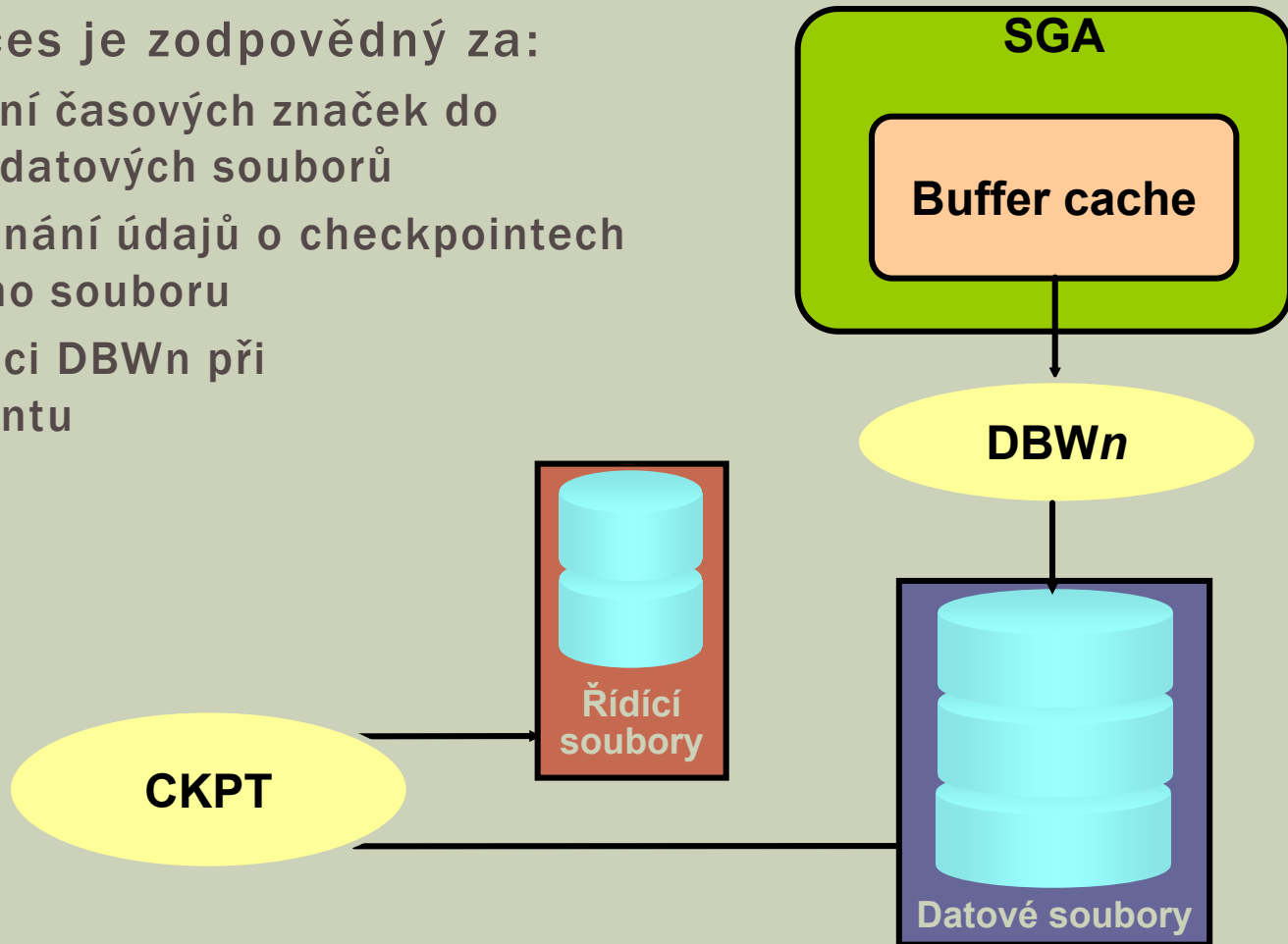
# PORUCHA INSTANCE

Typické příčiny	Možné řešení
Výpadek proudu	Restartujte instanci. Obnova instance je automatická, zahrnuje provedení všech transakcí podle transakčního logu a následné odrolování nepotvrzených transakcí  Zjistěte příčiny pomocí alert logu a trace souborů
HW porucha	
Porucha jednoho z kritických background procesů	
Náhlý shutdown	

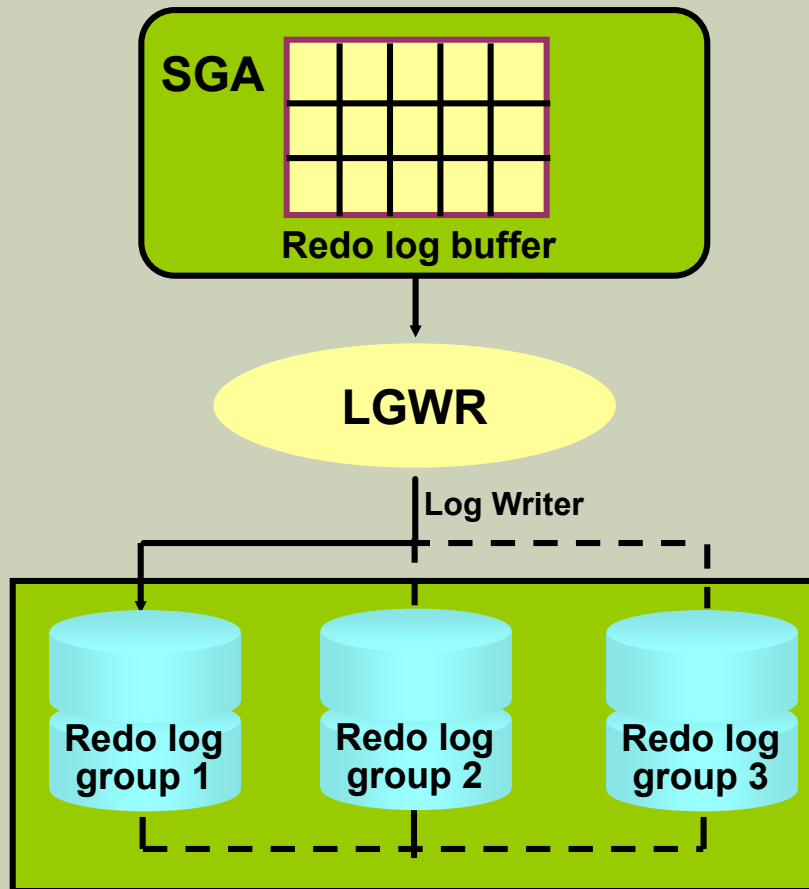


# CKPT

- CKPT proces je zodpovědný za:
  - Zapisování časových značek do hlaviček datových souborů
  - Zaznamenání údajů o checkpointech do řídicího souboru
  - Signalizaci DBWn při checkpointu



# LGWR



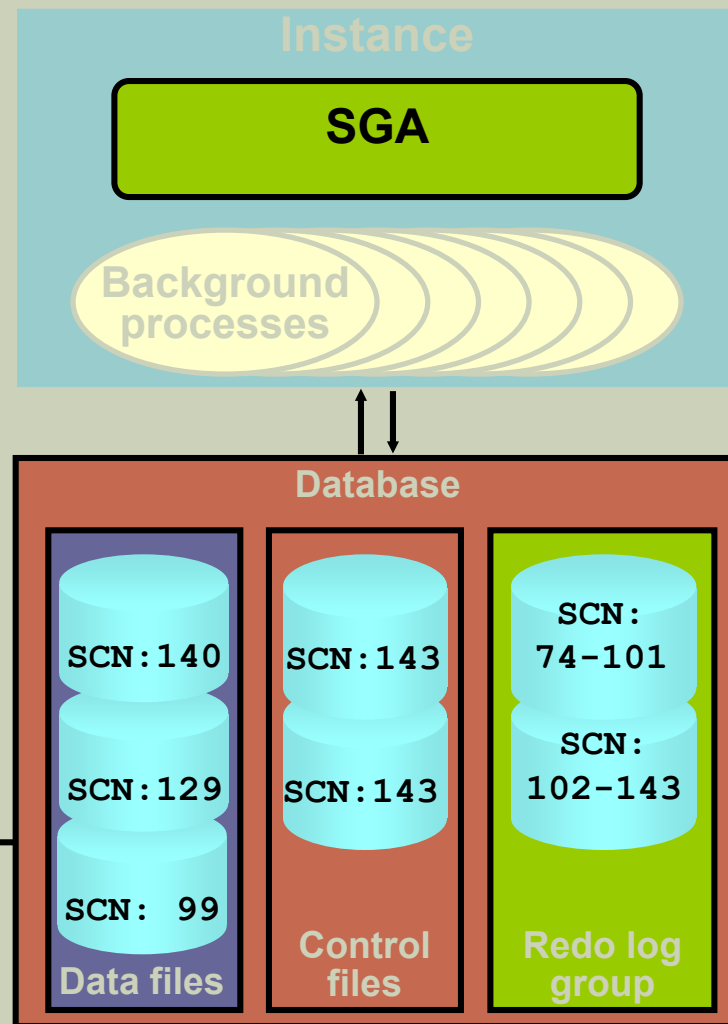
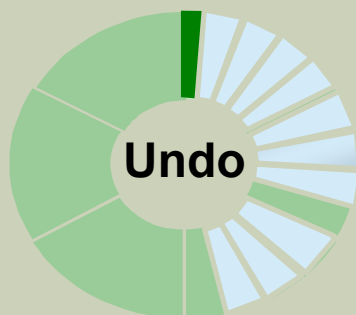
- Redo log soubory:
  - Zaznamenávají všechny změny v databázi
  - ~~Měly by být~~/Musí být multiplexovány
- Log Writer zapisuje:
  - Při commitu
  - Při zaplnění jedné třetiny
  - Každé tři vteřiny
  - Před zápisem DBWn
  - Před standardním shutdownem

# OBNOVA INSTANCE

- Automatická obnova instance:
  - Nastane, pokud se pokoušíme otevřít databázi, jejíž soubory nebyly při shutdownu synchronizovány
  - Použijí se informace z redo logů pro synchronizaci datových souborů
  - Zahrnuje dvě rozdílné operace:
    - Rolling forward: Změny z redo logů (jak potvrzené, tak nepotvrzené) jsou aplikovány na datové soubory.
    - Rolling back: Aplikované změny, které ale nejsou potvrzené jsou odrolovány zpět do stavu před začátkem příslušné transakce.

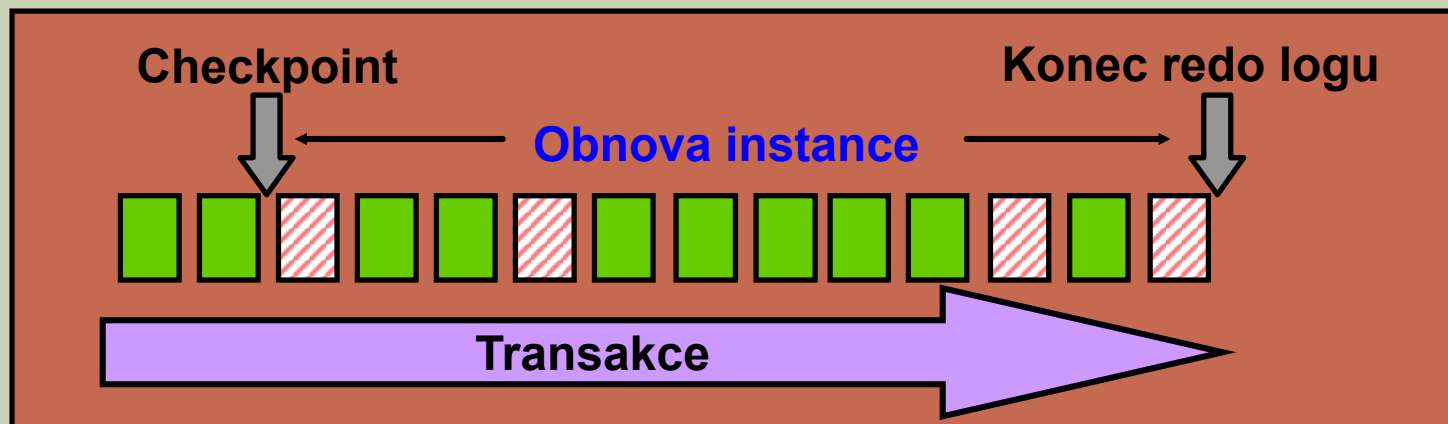
# FÁZE OBNOVY DATABÁZE

- 1. Startup instance (datové soubory nejsou aktuální)
- 2. Roll forward (redo)
- 3. V datových souborech jsou potvrzená i nepotvrzená data
- 4. otevření databáze
- 5. Roll back (undo)
- 6. Zůstala jen potvrzená data



# LADĚNÍ OBNOVY INSTANCE

- Během obnovy instance musí být všechny změny po posledním checkpointu až do konce redo logu aplikovány na datové soubory.
- V tomto případě můžeme ladit, jak dlouhá může být maximální doba od posledního checkpointu do konce redo logu (mezi dvěma checkpointy).
- Inicializační parametr `fast_start_mttr_target` – udává, za kolik sekund se maximálně obnoví DB v případě poruchy



# PORUCHA MÉDIA

Typické příčiny	Možné řešení
Porucha disku	<ol style="list-style-type: none"><li>1. Obnova (restore) postiženého souboru ze zálohy.</li><li>2. Zadat nové umístění souboru do řídicího souboru (pokud je to nutné).</li><li>3. Obnova (recover) souboru pomocí redo logu (pokud je to nutné).</li></ol>
Porucha diskového řadiče	
Ztráta souboru potřebného pro běh databáze	

# KONFIGURACE FAST RECOVERY AREA

- Fast recovery area:
  - Doporučené umístění záloh
  - Místo na disku (separované od datových souborů)
  - Nastavuje se parametrem `DB_RECOVERY_FILE_DEST`
  - Velikost pak parametrem `DB_RECOVERY_FILE_DEST_SIZE`
    - *POZOR – není žádný vztah mezi tímto parametrem a fyzickou velikostí volného místa*
  - Musí mít dostatečnou velikost pro všechny zálohy, archivní logy, flashback logy, multiplexované řídící soubory a multiplexované redo logy
  - Automaticky organizovaná podle tzv. retention policy
- Konfigurací fast recovery area máme na mysli nastavení umístění, velikosti a retention policy.

# MULTIPLEXING ŘÍDÍCÍCH SOUBORŮ

- Databázi **bez řídicího souboru nenastartujeme**, proto je nutná redundance – více řídicích souborů - multiplexing.

	Souborový systém
Best Practice	Alespoň dvě kopie, každá na zvláštním disku (nebo alespoň na zvláštním řadiči)
Jednotlivé kroky	<ol style="list-style-type: none"><li>1. Modifikace SPFILE příkazem ALTER SYSTEM SET control_files</li><li>2. Shutdown.</li><li>3. Vytvoření kopie(i) řídicího souboru.</li><li>4. Startup.</li></ol>

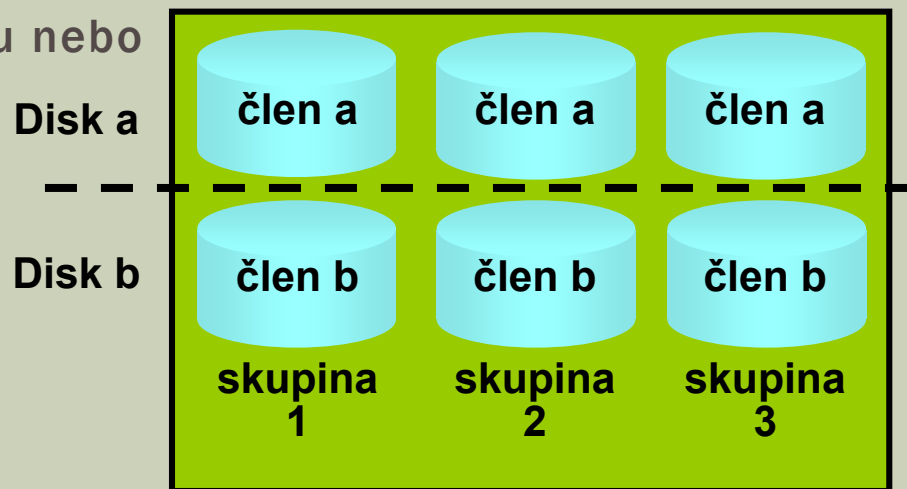
- Textová záloha řídicího souboru:

```
SQL>alter database backup controlfile to trace
```



# REDO LOGY

- Multiplexování redo logů je nutné pro prevenci ztráty dat, redo logy jsou potřebné pro obnovu databáze, bez nich to nejde.
- ~~Doporučuje se mít~~/Musíte mít:
  - Minimálně dva soubory v každé skupině
  - Každý člen na zvláštním disku nebo alespoň na zvláštním řadiči



**Poznámka: Multiplexování redo logů významně ovlivňuje celkový výkon databáze – nárůst I/O operací**

# MULTIPLEXING REDO LOGŮ

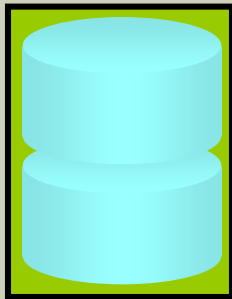
- Přidání redo logů do skupiny:

```
SQL>ALTER DATABASE ADD LOGFILE MEMBER  
'/oracle/data/log2b.rdo' TO GROUP 2;
```

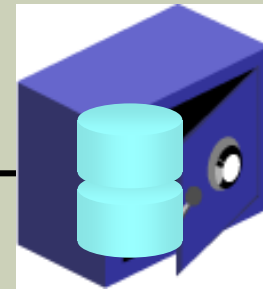
```
SQL>ALTER DATABASE ADD LOGFILE MEMBER  
'/oracle/data/log2c.rdo' TO ('/oracle/data/log2a.rdo',  
'/oracle/data/log2b.rdo');
```

# ARCHIVNÍ LOGY

- Pro zachování údajů z transakčního logu kopírujeme redo logy postupně do archivních logů. Postupujeme dle následujících kroků:
  1. Zadáme konvenci pro pojmenovávání archivních logů.
  2. Zadáme jednu nebo více umístění archivních logů.
  3. Přepneme databázi do **ARCHIVELOG** módu .



**Online redo log files**

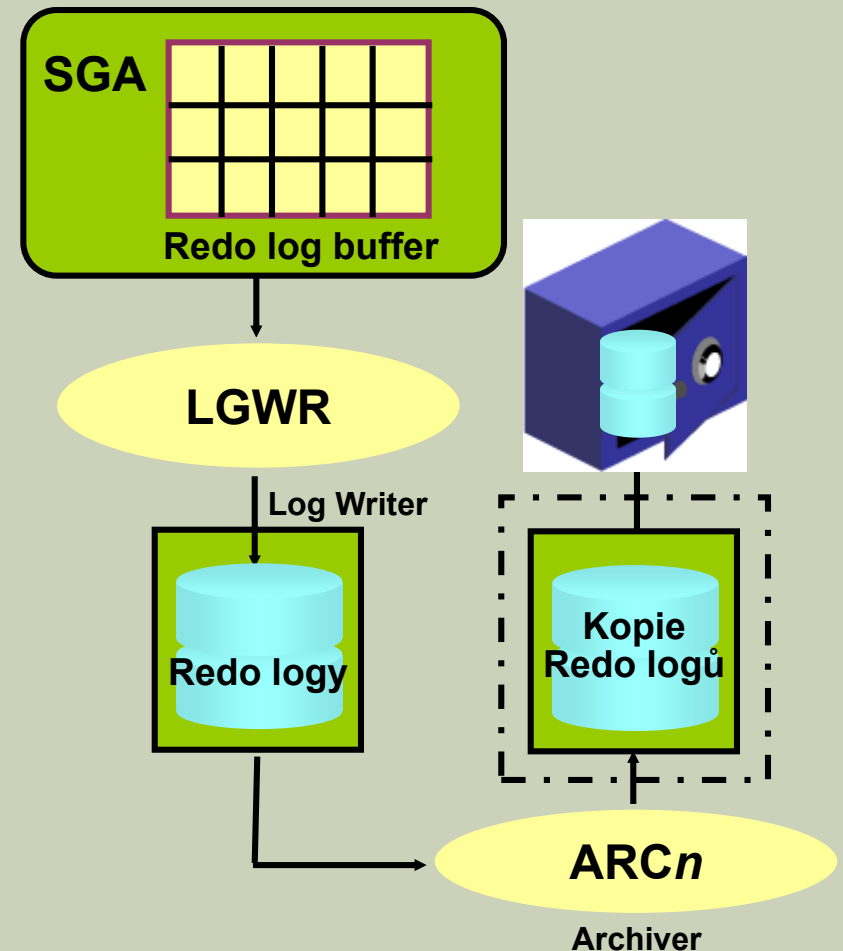


**Archive log files**

# ARCn

## ■ Archiver (ARCn):

- Je nepovinný databázový proces
- Automaticky archivuje redo logy pokud je databáze v `ARCHIVELOG` módu
- Uchovává úplně všechny změny v databázi
- U provozních databází je to nutnost



# ARCHIVNÍ LOGY: POJMENOVÁNÍ A UMÍSTĚNÍ

- Jsou dány inicializačními parametry:
  - **Maska** `LOG_ARCHIVE_FORMAT = arch_%t_%s_%r.arc`
  - **Pevné umístění**
    - `LOG_ARCHIVE_DEST_n = 'LOCATION=/disk1/arc,`
  - **Umístění dle nastavení flash recovery area**
    - `LOG_ARCHIVE_DEST_n =  
'LOCATION=USE_DB_RECOVERY_FILE_DEST'`
  - **n je od 1 do 31**

# ZAPNUTÍ ARCHIVELOG MÓDU

- V jakém modu běží databáze zjistíme následujícím příkazem

```
$ sqlplus / as sysdba
...
SQL> select log_mode from v$database;

LOG_MODE
-----
NOARCHIVELOG

SQL>
```

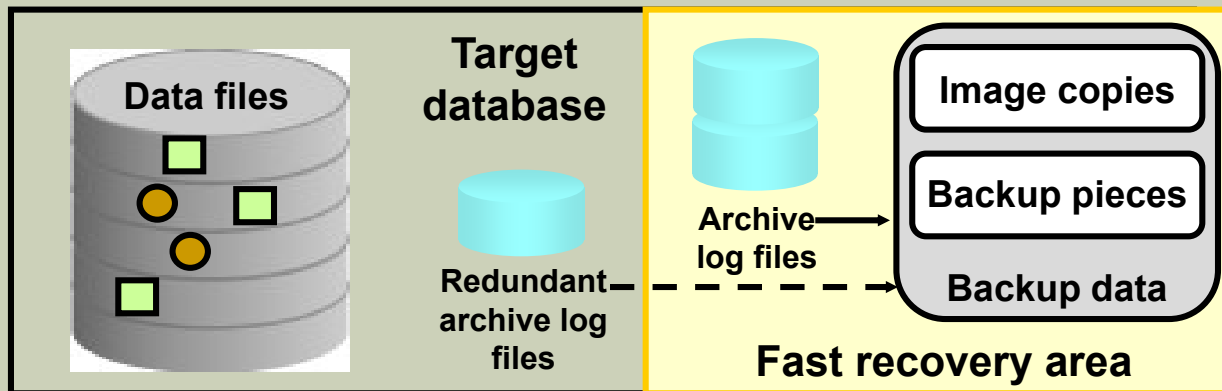
# ZAPNUTÍ ARCHIVELOG MÓDU

- Pro přepnutí databáze do ARCHIVELOG módu je třeba postupovat podle následujících kroků:

```
$ sqlplus / as sysdba
...
SQL> shutdown immediate
SQL> startup mount
SQL> alter database archivelog;
SQL> alter database open;
```

# JAK ZÁLOHOVAT

- Zálohovat můžeme:
  - Pomocí Recovery Manageru (RMAN)
  - Pomocí Oracle Secure Backupu
  - Uživatelsky





# ORACLE SECURE BACKUP A RMAN

- Oracle Secure Backup a RMAN jsou finálními řešeními pro zálohování v prostředí Oracle, oba umožňují:
  - Centralizovanou správu úložiště záloh (disk, páska)
  - Zálohovat všechny typy úložišť data databáze Oracle
  - Provádět zálohování přes síť



# UŽIVATELSKÉ ZÁLOHY

- Uživatelsky řízené zálohování:
  - Zálohování na míru
  - Typicky pomocí vlastních skriptů
  - Ruční změna statusu databázových souborů
  - Zálohy se dělají přímo příkazy OS

```
SQL> alter tablespace data offline;  
...  
$ cp /home/oracle/oradata/data.dbf /backup_location  
SQL> alter tablespace data online;
```

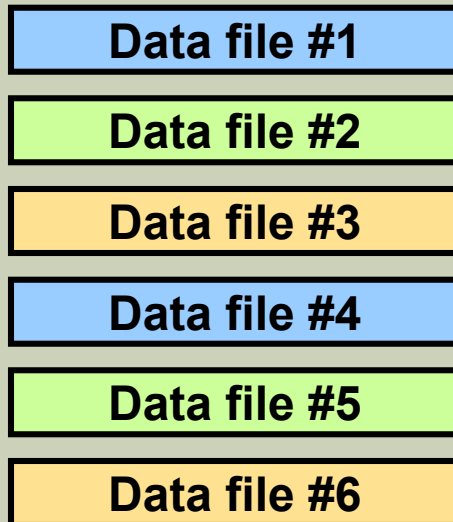
- Dnes se prakticky nepoužívá

# TERMINOLOGIE

- Strategie zálohování:
  - Celá databáze
  - Jen část databáze
- Typy záloh:
  - Všechny datové bloky (full backup – level 0)
  - Jen bloky změněné od poslední zálohy (incremental backup – level 1)
    - Kumulativní – od poslední full zálohy
    - Diferenciální – od poslední inkrementální zálohy
- Backup mód:
  - Offline (konzistentní, cold)
  - Online (nekonzistentní, hot)

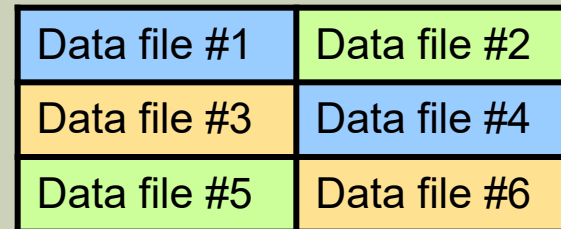
# TERMINOLOGIE

- Zálohy mohou být uloženy jako:
  - Kopie souborů (image copies)
  - Zálohovací sety (backup sets resp. compressed backup sets)



**Image copies**

**(Duplikovaná data a log soubory v OS formátu)**



**Backup set**

**(Binární komprimované  
soubory v proprietárním  
formátu Oracle)**

# RECOVERY MANAGER (RMAN)

- Skriptovací jazyk pro řízení zálohování
- Integrovatelný s Cloud Control Enterprise Managerem
- Veřejné API, které umožňuje navázat na RMAN nejrůznější zálohovací sw třetí strany
- Zálohuje datové soubory, řídicí soubory, archivní logy a inicializační soubor
- Zálohuje na disk nebo pásku

# RMAN - KONFIGURACE

- Konfiguruje se příkazem `configure`
- Výpis konfigurace získáme příkazem `show all`

```
$ rman target /
```

```
RMAN>show all;
```

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
```

```
CONFIGURE BACKUP OPTIMIZATION OFF; # default
```

```
CONFIGURE ...
```

```
...
```

# RMAN - KONFIGURACE

```
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; #
default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/home/oracle/app/oracle/product/12.1.0.2/dbs/db_home1/snapcf_cdb.f'; #
default
```

# RMAN – PŘÍKAZOVÁ ŘÁDKA

```
$ rman target /
```

```
RMAN>
```

## ■ Konfigurace

```
RMAN> configure ...
```

```
RMAN> configure retention policy to recovery window of 30  
days;
```

```
RMAN> configure retention policy to redundancy 3;
```

```
RMAN> configure controlfile autobackup on;
```

```
RMAN> configure device type disk parallelism 1 backup  
type compressed backupset;
```



# RMAN – PŘÍKAZOVÁ ŘÁDKA

- Zálohy dalších souborů
  - Archivní logy – můžeme je po zazálohování rovnou smazat
  - Spfile a Controlfile – pokud nastavíme pomocí configure, pak se automaticky zálohuje při každé záloze databáze
  - *nelze init soubor, ten „není součástí Oracle“*

- Základní příkaz je:

```
RMAN> backup database plus archivelog delete input;
```

- Zazálohuje datové soubory, pokud je nastaveno **controlfile autobackup** na **on** tak i controlfile a spfile, zazálohuje všechny archivní logy, co najde a ty následně smaže z původního umístění

# RMAN – PŘÍKAZOVÁ ŘÁDKA

- Modifikace základního příkazu:

- Nezálahuje vůbec archivní logy

```
RMAN> backup database;
```

- Archivní logy zazálahuje ale nesmaže je

```
RMAN> backup database plus archivelog;
```

- Zazálahuje pouze spfile

```
RMAN> backup spfile;
```

- Zazálahuje pouze controlfile

```
RMAN> backup current controlfile;
```

# RMAN – PŘÍKAZOVÁ ŘÁDKA

## ■ Základní typy zálohování:

- Level 0 = full
- Level 1 = inkrementální (oproti poslední záloze – jakékoliv)
- Level 1 cumulative = inkrementální oproti poslední full záloze

```
RMAN> backup [incremental level=0] database;
```

```
RMAN> backup incremental level=1 database;
```

```
RMAN> backup incremental level=1 cumulative database;
```

# RMAN – CONTROLFILE X KATALOG

- Pro uložení stavu záloh RMAN používá
  - Controlfile
    - Používáme, pokud máme málo databází
    - Komplet seznam záloh plus nastavení je přímo v souboru
    - Pokud zálohujeme „často“, může to vést k nárůstu velikosti controlfile
  - Katalog
    - Používáme, pokud máme hodně databází
    - Komplet seznam záloh plus nastavení všech databází je ve zvláštním schématu
    - Doporučuje se mít jednu centrální zálohovací databázi s pouze s katalogem záloh bez jakýchkoliv dalších schémat a systémů
    - Databáze s katalogem musí být „nerozbitná“
    - Šetří se velikost a místo v controlfile

# RMAN – PŘÍKAZOVÁ ŘÁDKA

- Výpis všeho, co je zazálohováno

```
RMAN> list backup;
```

- Crosscheck provede porovnání toho, co je zazálohováno podle řídicího souboru (katalogu) a toho, co je fyzicky na disku
  - Používá se pro vlastní zálohy nebo pro archivní logy

```
RMAN> crosscheck backup;
```

```
RMAN> crosscheck archivelog all;
```

# RMAN – PŘÍKAZOVÁ ŘÁDKA

- Expired = neexistující
  - Fyzicky neexistuje, na serveru bylo smazáno nebo přemístěno !!!
- Obsolete = nevyhovující retention policy
  - Fyzicky existuje, ale podle nastavení retention policy už není potřeba
  - Stará záloha (recovery window) nebo nadbytečná záloha (redundancy)

```
RMAN> delete expired backup;
```

```
RMAN> delete expired archivelog all;
```

```
RMAN> delete obsolete;
```

**DOTAZY?**