

# Moderné databázové systémy

Architektúra databázových systémov

Ján Jozaf

[Jan.Jozaf@profinit.eu](mailto:Jan.Jozaf@profinit.eu)



IT řešení, které přináší ovoce

<http://www.profinit.eu/cz/podpora-univerzit/univerzitetni-vyuka>



# Obsah

- Architektúra databázových systémov
- Transakčný mechanizmus
- Databázové procesy
- Buffery
- Dátové súbory
- Súbory pre podporu transakčného mechanizmu
- Zálohovanie a recovery



# Význam databázových systémov

- Databázy

- obsahujú dáta, ktoré sú pre firmy kritické
  - Údaje o zákazníkoch, obchodoch, trhu, prevádzke, infraštruktúre atď.
- umožňujú operácie nad uloženými dátami
  - Vyhľadávanie, aktualizovanie, vymazávanie atď.
- obsahujú firemné **know-how**

Data. It's organization's most proprietary asset.  
The competitors can copy your processes.  
They can buy the same equipment.  
They can even steal your customers.  
But the data is unique to your business and distinctly yours.  
*Sohaib Abbasi, Chairman and CEO, Informatica company*



# Architektúra databázového systému

- Databázy sú kľúčovým elementom každej firmy a architektúra ich systému musí byť vysoko sofistikovaná
- Požiadavky na architektúru:
  - Neobmedzený počet používateľov
  - Vzdialený prístup
  - Stabilita
  - Podpora transakčného mechanizmu
  - Centralizovaná administrácia
  - Vysoká úložná kapacita dát
  - Výkonnosť
  - Monitoring



# Transakcie

- Architektúra databázy je vo veľkej miere ovplyvnená používaním transakčného mechanizmu
- Operácie v databázach sa vykonávajú v **transakciách**
- Transakcia je celok, ktorý sa vždy uskutoční ako celok alebo žiadna časť z neho t.j. dáta sa vrátia do stavu ako pred začatím transakcie
- Pokiaľ nie je transakcia ukončená, ostatní užívatelia nevidia zmeny

## Význam:

- Zamedziť vzniku nekonzistentného stavu, napríklad: polovica údajov sa aktualizovala a druhá polovica nie
- Zabezpečiť jednoznačnosť prístupu k dátam v prípade súbežného prístupu viacerých používateľov

## Princíp:

Krok 1, Začiatok transakcie  
Krok 2, Spustenie aktualizácie dát  
Krok 3, V prípade že nastane chyba tak návrat k pôvodným hodnotám  
Krok 4, V prípade že nenastane chyba tak potvrdenie aktualizovaných údajov



# Zámky

- Mechanizmus, ktorý podporuje podporujúci transakcie
- Transakcia pri modifikácii dát zamkne modifikované záznamy a uvoľní ich až po skončení transakcie
- Ostatné transakcie ktoré chcú modifikovať záznamy, ktoré majú zámok musia čakať na ich sprístupnenie
- Existujú rôzne úrovne a typy zámkov
  - Zámok nad tabuľkou, zámok nad jednotlivými riadkami
  - Exkluzívny zámok, sharovaný zámok



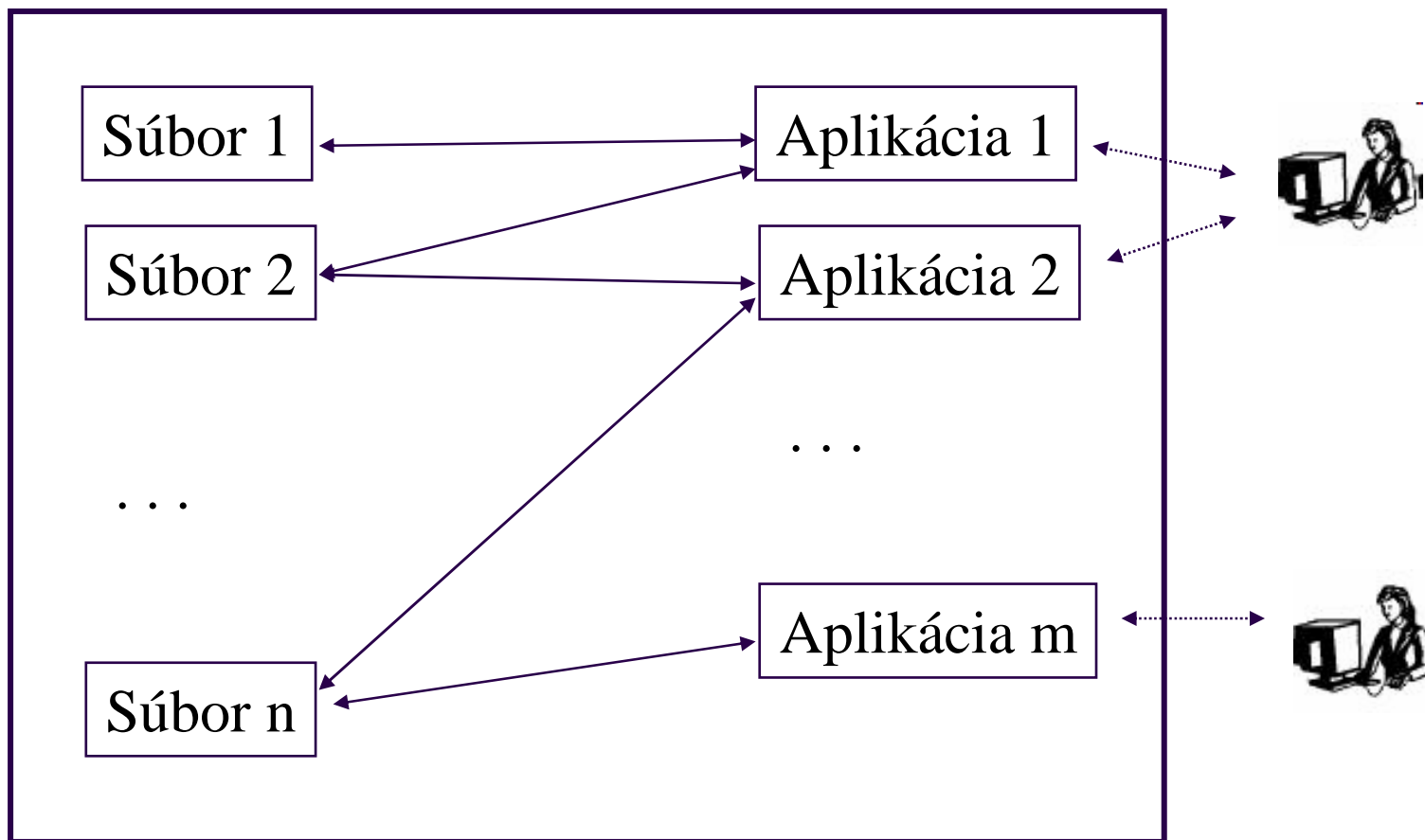
# ACID

- (A)tomicity – atomizácia
  - Transakcia sa buď vykoná ako celok, alebo sa vôbec nevykoná
- (C)onsistency - konzistencia
  - Databáza ostáva v konzistentnom stave pred každou transakciou a po každej transakcii
- (I)solation - izolácia
  - Transakcia je izolovaná od ostatných transakcií
- (D)urability – trvácnosť
  - Dáta sú bezpečne a trvalo uložené



# Návrh architektúry I.

Jednoduchá architektúra ukladania dát:



Problémy:

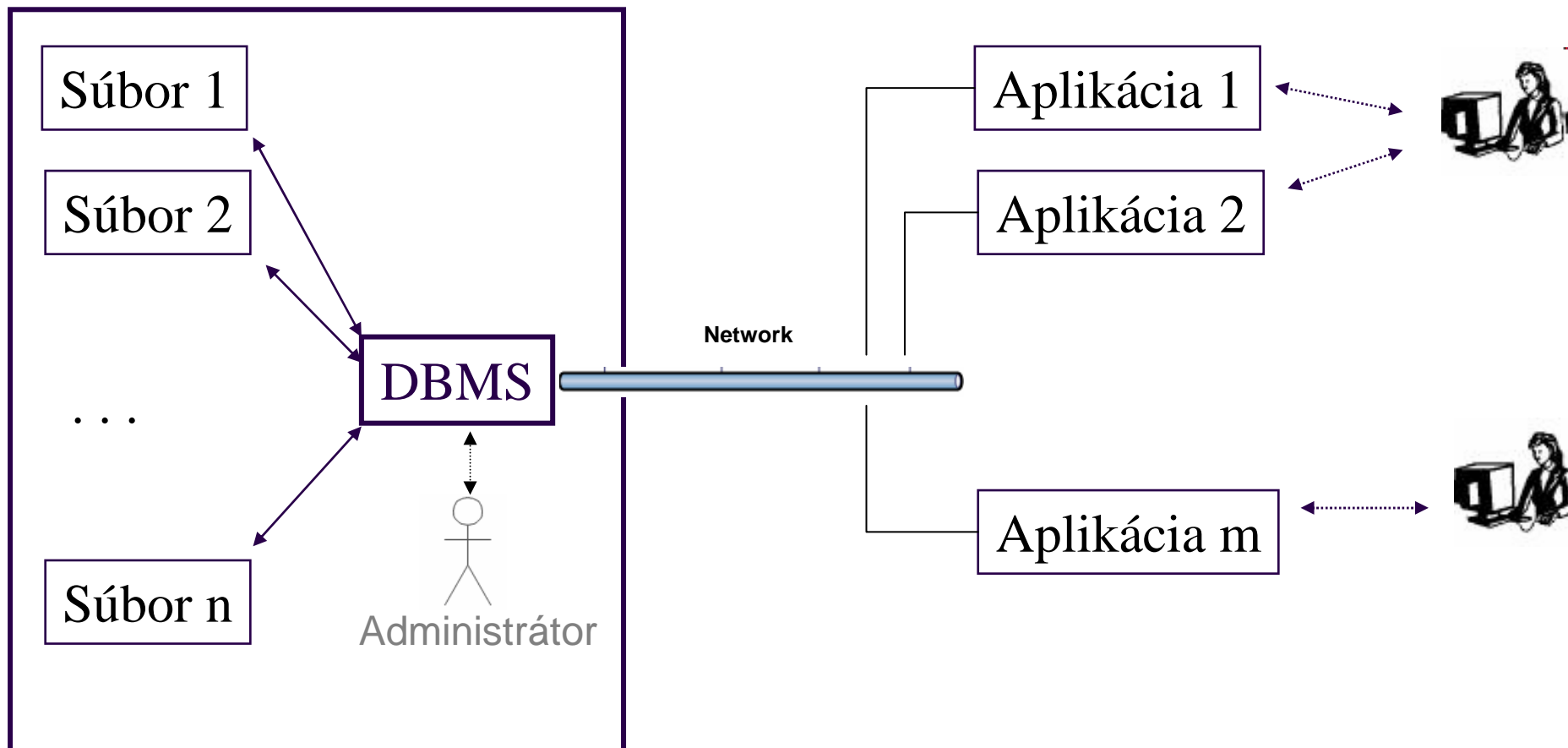
- ACID
- Stabilita
- Nezávislosť
- Integrita
- Bezpečnosť





# Návrh architektúry II.

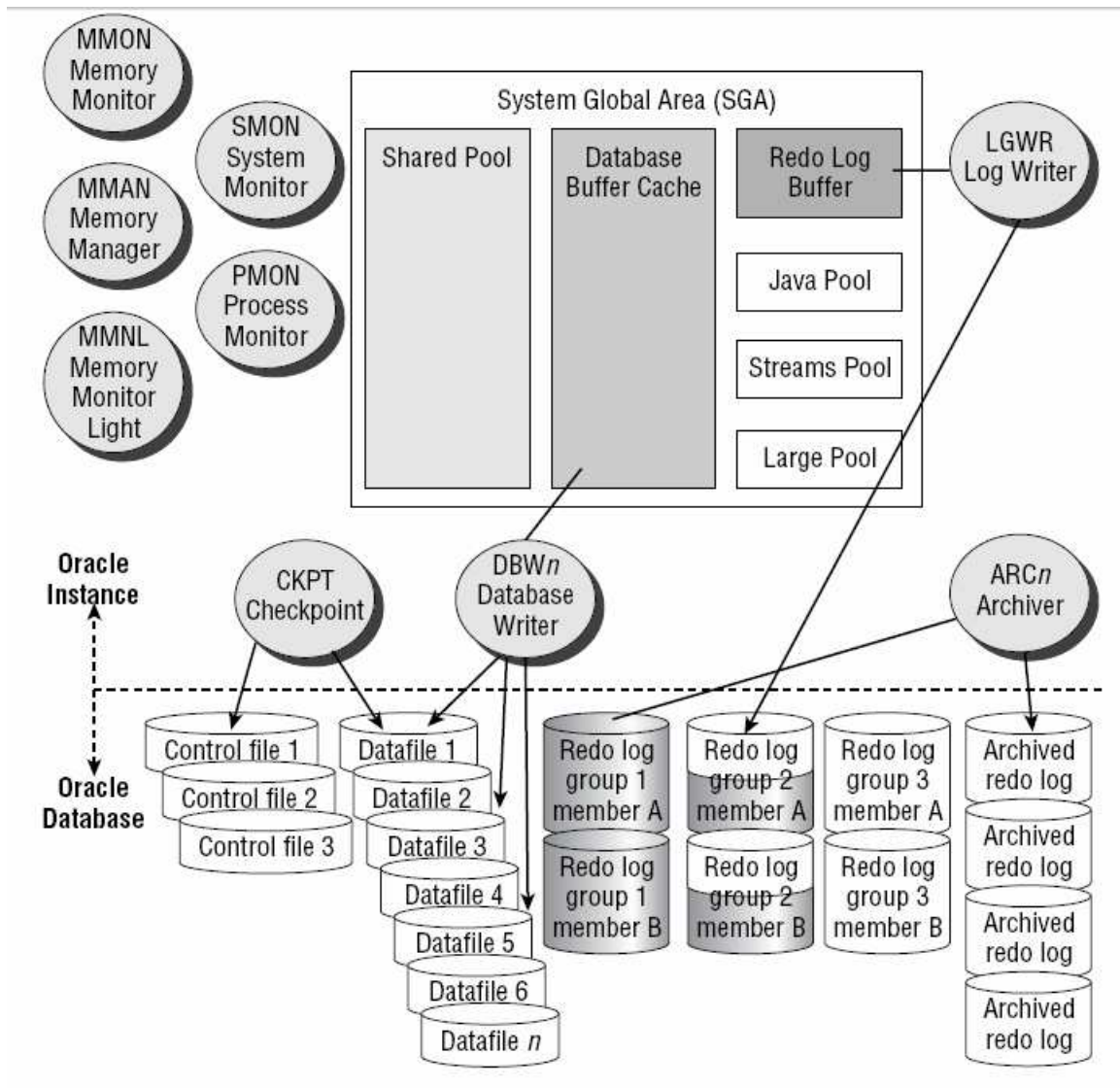
Komplexnejšia architektúra ukladania dát:





# Návrh architektúry III.

## Príklad architektúry z praxe (Oracle):



- Procesy
- Buffery
- Dátové súbory
- Súbory podporujúce transakčný mechanizmus



# Procesy

- Podpora používateľov
  - Vytvorenie a udržiavanie aktívneho spojenia medzi clientom a serverom, zabezpečenie prenosu dát
- Správa pamäte
  - Alokácia a dealokácia pamäte, koordinácia pri pridelovaní pamäte, bezpečnosť prístupu k pamäti
- Správa transakcií
  - Podpora pre transakčný mechanizmus, ukladanie informácií z aktívnych transakcií, rollback dát pri neúspešných transakciách
- Zabezpečenie stability a obnovy databázy
  - Monitorovanie systémových súborov
  - Zaznamenávanie kontrolných checkpointov a zabezpečenie obnovy databázy v prípade chyby



# Procesy pre kontrolu a monitorovanie

- Proaktívne monitorovanie a zaznamenávanie možných problémov
  - Príklad: blížiaci sa nedostatok miesta, málo pamäte
  - Možnosť definovania vlastných alertov
- Monitoring výkonnosti a rýchlosti prístupu k dátam
  - Pomalé transakcie, často používané transakcie
- Audit prístupov a manipulácie s dátami
  - Monitorovanie prístupu k dátam, môže byť viacúrovňová

The screenshot shows the Oracle Applications Manager interface for 'System Alerts and Metrics'. It includes a summary table for Alerts and Occurrences, a 'New Alerts' section with a table of recent alerts, and a 'Quick Searches' section.

Severity	New	Open
Critical	2	0
Error	1	0
Warning	0	0

Severity	New	Open
Critical	2	0
Error	2	0
Warning	0	0

Select Message	Severity	Category	Time	Occurrences	Notes
<input type="checkbox"/> User has reported a...	Critical	User	30-11-2004 15:22:31	1	
<input type="checkbox"/> Failed to display Sa...	Critical	Product	30-11-2004 15:22:20	2	
<input type="checkbox"/> An Unexpected Error...	Error	System	30-11-2004 15:17:43	2	

Quick Searches: An Unexpected Error -942 has occurred - a log file has been created. ORA-01003: no statement parsed



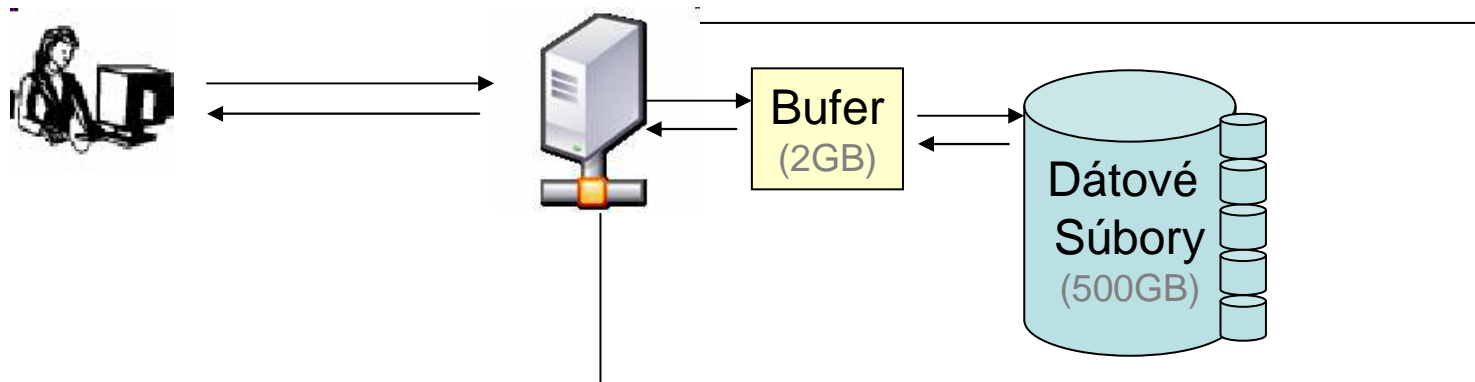
# Buffers I.

- Rýchlosť prístupu k dátam je kriticky dôležitá, veľké databázy obsahujú desiatky TB údajov, ktoré sa čítajú z dátových súborov a musia byť rýchlo prístupné
- Dostupnosť dát z pamäte - bufferov je niekoľko násobne rýchlejšia ako pri prístupe k informáciám uloženým v dátových súboroch
- Dáta ku ktorým sa v databáze pristupuje sú v istej miere predvídateľné a táto vlastnosť sa dá využiť
- Často a naposledy pristupované dáta sa v odmedzenej kapacite ukladajú do pamäte do **bufferov** a následne odtiaľ čítajú
- Pri ukladaní dát do bufferov sa používa sa LRU (least recently used) algoritmus



# Buffery II.

- Dátový buffer
  - Naposledy čítané údaje, často používané údaje
  - Menené údaje – kvôli výkonnosti sa nezapisujú do dátových súborov hneď ale v definovaných intervaloch
- Systémový buffer
  - Systémové nastavenia, používateľské nastavenia, databázové operácie – napr. triedenie dát
- Aplikačný buffer
  - Podpora aplikácií napr. JAVA





# Dátové súbory I.

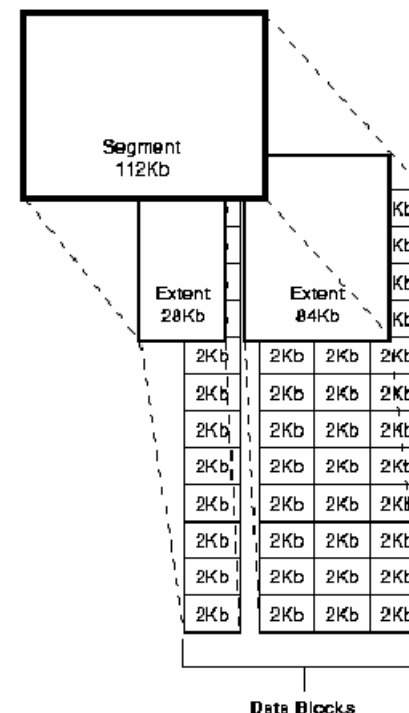
- Sú to fyzické súbory, ktoré obsahujú dáta uložené v databáze
- Ich vnútorná štruktúra závisí od konkrétnej implementácie databázového systému
- Veľkosť súborov môže byť od pár MB až po desiatky GB
- Jeden dátový súbor obsahuje väčšinou dáta, ktoré tvoria logický celok
  - V prípade straty dátového súboru, sa stratí iba časť dát, ostatné logické a dátové celky ostanú nedotknuté
- Pre rýchlejší prístup k dátam sa môžu použiť rozličné techniky
  - Paralelizmus, kompresia dát



# Dátové súbory II.

- Dátový blok = základná jednotka pri operáciách s dátami
- Dátové bloky sa ďalej zoskupujú do vyšších celkov, príklad: extent, segment
- Operácie s dátovými súbormi sa robia priamo príkazmi v databáze, ako príklad definícia miesta v dátovom súbore:

tablespace lecture  
storage (initial 6144,  
next 6144,  
minextents 1,  
maxextents 5,  
pctincrease 5,  
pctfree 20);







# Súbory pre transakčný mechanizmus I.

- Slúžia na podporu transakčného mechanizmu
  - Rollback, commit, recovery
- Súbory na disku, do ktorých sa zapisujú informácie o všetkých transakciách v databáze aj s ich dátami
  - DDL operácie: CREATE, DROP, ALTER atď.
  - DML operácie: INSERT, UPDATE, DELETE, MERGRE atď.
- Informácie sa zapisujú ešte predtým ako sú zapísané do dátových súborov



# Súbory pre transakčný mechanizmus II.

- Príklad architektúry - **Redo log a Undo log**:
  - Redo log - obsahuje informácie, ktoré sú potrebné pre korektné obnovenie databázy ak príde k strate dátového súboru – t.j. je tam uložená celá história dát od poslednej zálohy
  - Undo Log – obsahuje dáta, ktoré sú potrebné pre aktívne transakcie v prípade rollbacku, ďalej môže obsahovať dáta z už ukončených transakcií napríklad (vrátenie sa v čase dozadu, long running queries)

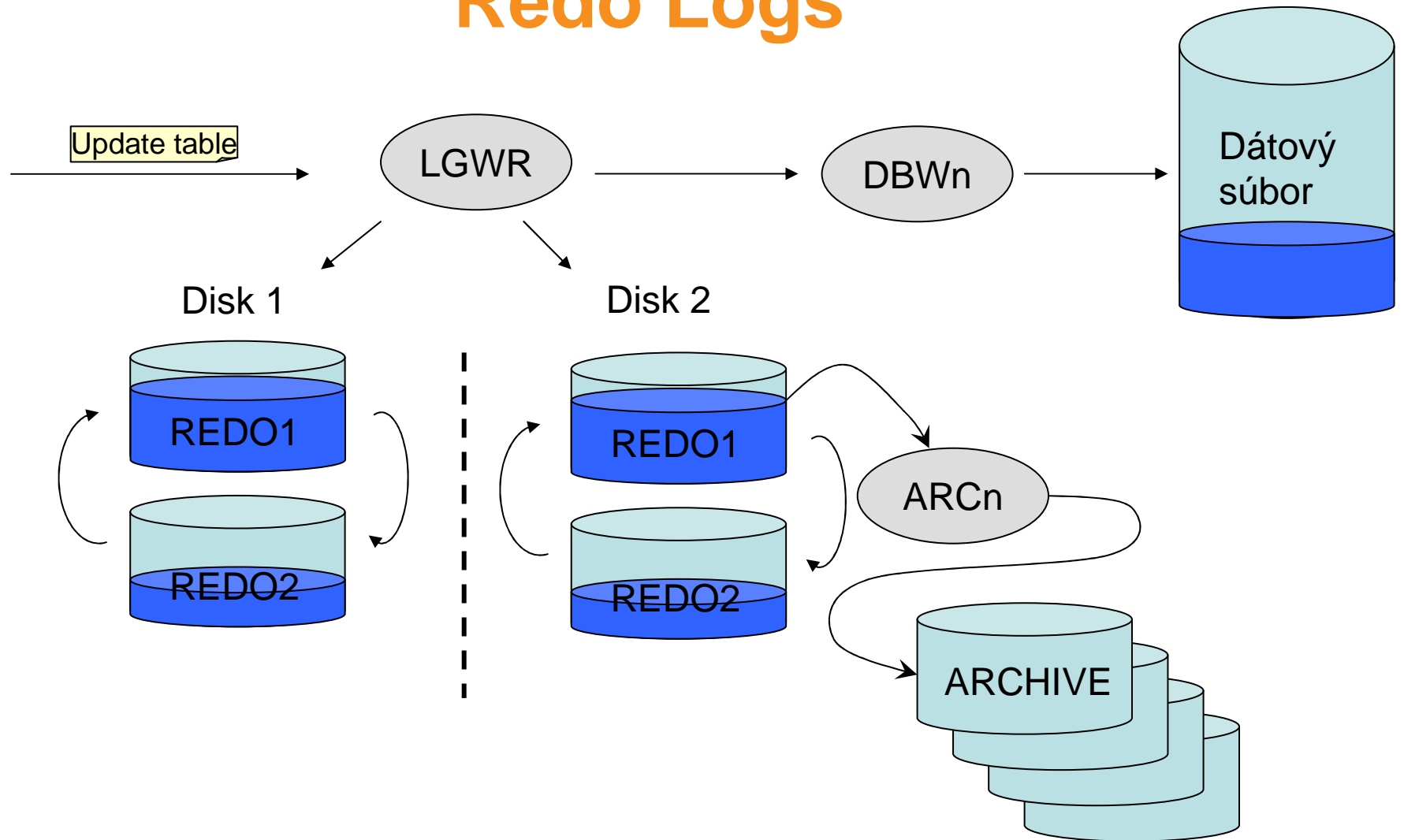


# Súbory pre transakčný mechanizmus III.

- Nakoľko Redo Log má iba obmedzenú kapacitu a obrovský význam je potrebné ho archivovať
- Redo Log obsahuje vždy minimálne 2 súbory, pričom sa striedajú v zapisovaní do nich – jeden je online a druhý offline, keď sa prvý zaplní tak sa následne vymenia
- Po tomto procese sa offline Redo Log archivuje a až následne môže byť prepísaný novými dátami

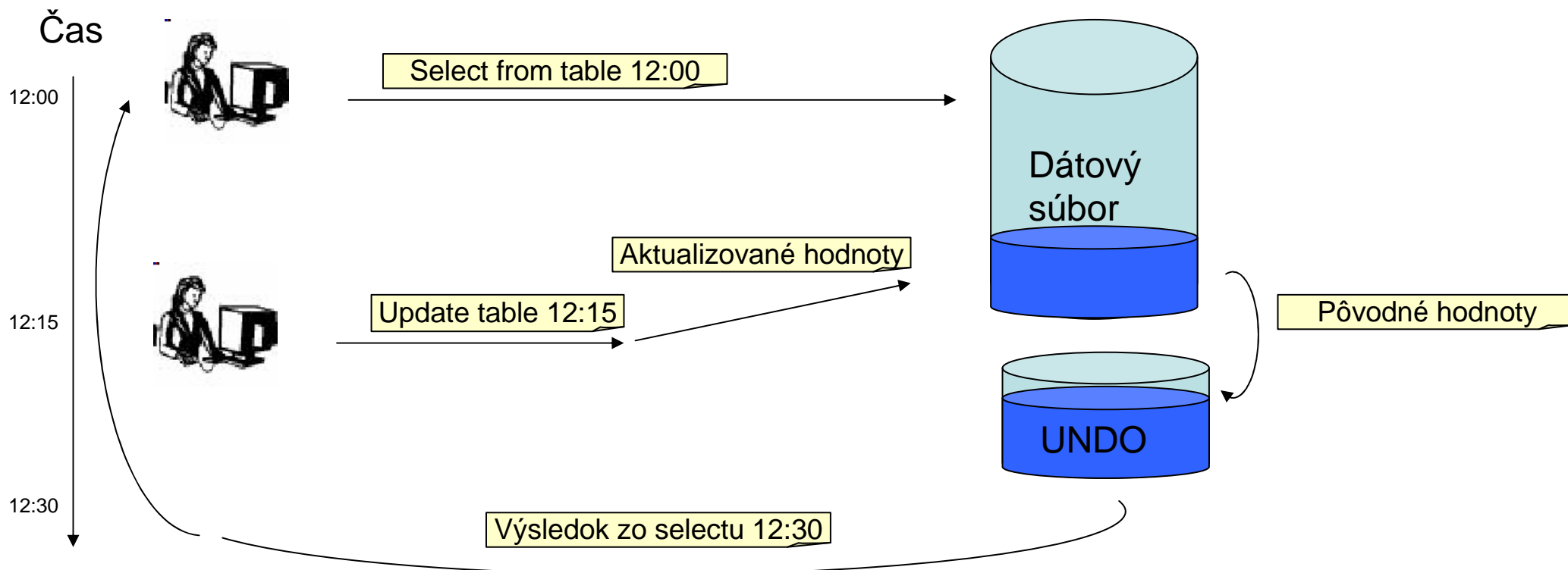


# Redo Logs





# Undo log





# Zálohovanie

- Pre prípad poškodenia dát v dátových súborov
  - Chýbajúci dátový súbor, poškodený dátový súbor, užívateľská chyba
- Rôzne zálohové média
  - Diskové polia, pásy
  - Vytvára sa viacero kópií zo zálohy
- Kompletná záloha
  - Kompletná záloha systému – kópia 1 k 1
  - Nevýhodou sú vysoké nároky na voľné miesto na zálohovom médiu
- Inkrementálna záloha
  - Čiastočná záloha – záloha zmenených častí od posledného backupu
  - Existujú rôzne mechanizmy zisťovania zmenených dát

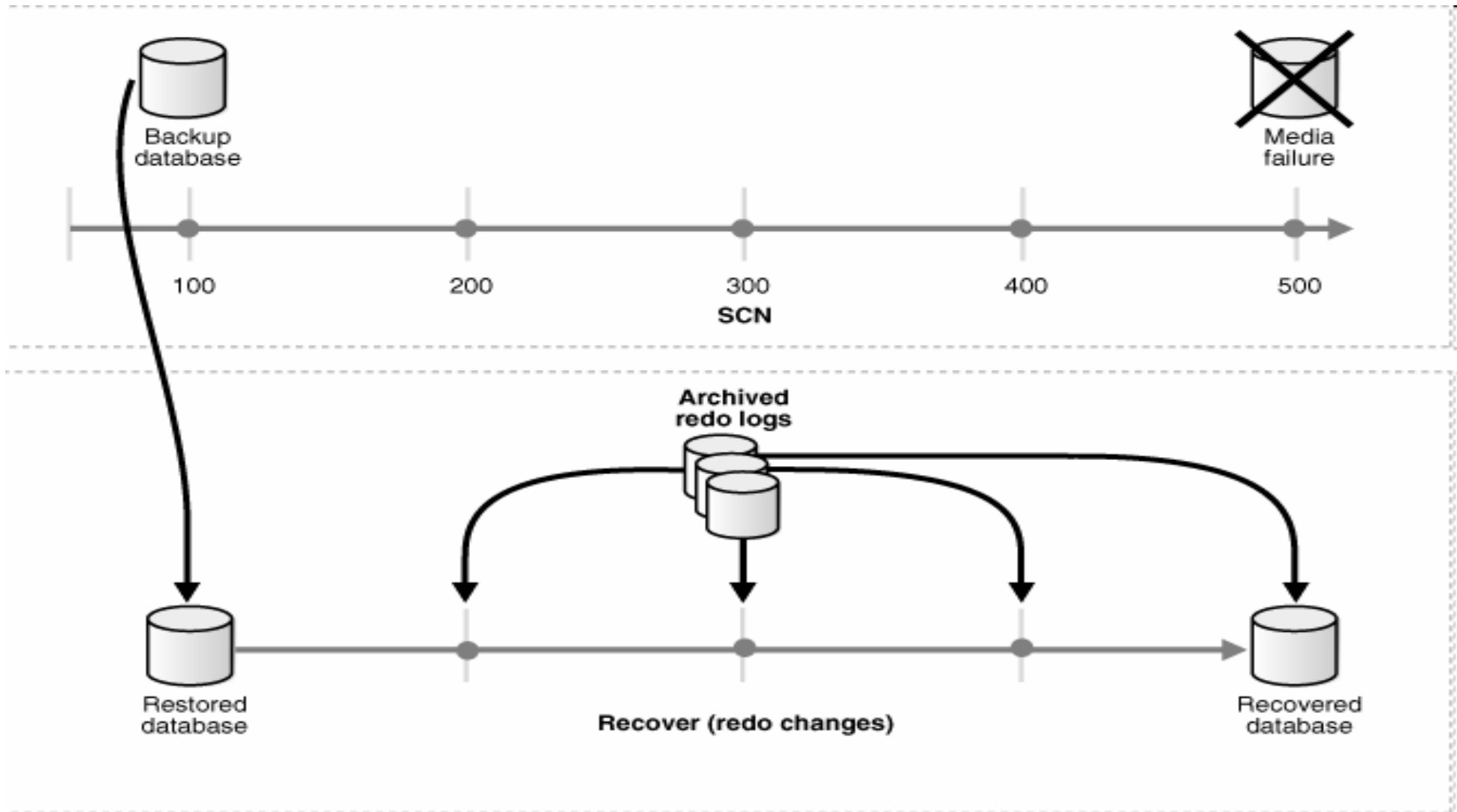


# Recovery I.

- Proces, ktorý je vo veľkej miere riadený samotnou databázou
- Rozlišuje sa podľa typu problému ktorý nastal
  - Strata dátového súboru, nekonzistentné ukončenie databázy, užívateľská chyba, nedostatok miesta atď.
- Recovery prebieha pomocou archivovaných a aktuálnych redo logov ako aj za pomoci zálohovaných údajov
- Recovery po užívateľských chybách
  - Recycle bin, flashback, log miner
- Dĺžka recovery závisí od nastavenia parametrov v databáze a dá sa vo veľkej miere ovplyvniť



# Recovery II.







# Diskusia

- Komentáre
- Otázky
- Pripomienky
- Upresnenie
- Poznámky
- ...

