# DS 2 Zapis

* Log
  + Ukladani informaci o všech operacich
  + Vyuziva sekvencni zapis, kdezto databaze nahodny pristup
  + Dopredny zapis
* Zotaveni
  + Vysledek je korektni stav DB (vse je ve spravnem stavu)
  + Vazano na cely DB systém
  + Provadi se v pripade globalnich chyb (chyba systemu, chyba media, …)
  + UNDO – presny stav transakce před chybou není znam, transakce je zrusena
  + REDO – transakce byla ukoncena COMMIT, ale zmeny nebyly preneseny z logu -> transakce musí byt prepracovana
* Transakce
  + Atomicka jednotka práce s DB
  + Cilem je prevest korektni stav DB na jiny korektni stav
  + Commit, rollback
* Konzistentni DB
  + V DB neexistuji vyjimky z danych integritnich omezeni
* Potvrzovaci bod (Commit point)
  + Operace COMMIT zavadi do logu stav databaze, na který se pripadne odkazuje ROLLBACK
* ACID
  + Kazda transakce musí splnovat ACID
  + Atomicnost – v transakci jsou provedene budto všechny operace nebo zadna
  + Korektnost – z korektniho do korektniho stavu
  + Izolovanost – transakce jsou navzajem izolovany (viditelne pro další transakce az po COMMITU)
  + Trvalost – zmeny jsou trvale i po pripadnem padu systemu
* Techniky zotaveni
  + Odlozena aktualizace (NO-UNDO / REDO)
    - Neprovadi aktualizaci logu a DB az do potvrzeni transakce (všechny aktualizace jsou zaznamenany do pameti)
    - REDO se provadi pokud systém zapsal aktualizace do logu
    - Není nutne provade UNDO (do DB nebyla aktualizovana)
    - V logu jsou nove hodnoty
    - Hrozi preteceni vyrovnavaci pameti
  + Okamzita aktualizace (UNDO / NO-REDO)
    - Aktualizace -> zapis log -> aktualiazce DB (po kazde aktualizaci transakce) a to před provedenim COMMIT
    - Pokud transakce selze před commit pointem, je nutne provest UNDO
    - Do logu se ukladaji puvodni hodnoty, coz umozni UNDO
    - Pomale, velky počet zapisu do DB, ale nedochazi k preteceni pameti
  + Kombinovana technika (UNDO / REDO)
  + Kontrolni body (Check points)
    - Vytvarene po urcitem poctu zaznamu zapsanych do logu
    - Zahrnuje všechny transakce vykonavane v době vytvoreni kontrolniho bodu + transakce, které byly ukonceny před kontrolnim bodem, ale nebyly ulozeny do DB v ramci predchoziho kontrolniho bodu
  + Algoritmus zotaveni
  + Zachranne body (save points)
    - Rozdeluje transakci na mensi casti (ROLLBACK TO <bod>, SAVEPOINT <jmeno>, RELEASE <jmeno>)

ORM

* ORM
  + Technika zpristupnujici relacni ci objektove relacni data pro objektove prostredi
* Parametrizovane dotazy
  + String querry = „SELECT \* FROM Users WHERE id = ?“
  + Stmt.setInt(1, 1)
* SQL Injection
  + Spusteni neosetreneho SQL dotazu, který muze best ke kompromitaci databaze
  + U dynamickeho SQL
  + Jak zabranit?
    - Sanizace vstupu, staticke sql, parametrizovane dotazy
* Domenovy objekt
* DAO (Data Acces Object)
  + Rozhrani k praci s DTO

OR Datovy Model

* Vlastnosti
  + Vlastní strukturovane datove typy (atributy i metody)
  + Funkce, procedury, triggery, které jsou ulozeny v SRBD
  + Dedicnost (CREATE TYPE student UNDER person)
  + Datovy typ ukazatele
  + Kolekce dat
* OID (Objektovy identifikator)
  + Identifikuje objekty

Fyzicka implementace, vykonavani dotazu, indexy

* Casova slozitost pro tabulky typu halda
  + Vkladani O(1), hledani, mazani a bodovy rozsah O(n)
* Casova slozitost pro tabulky typu b-strom
  + Vse O(log n)
* Jednoduchy index
* Slozeny index
* Kandidati na index
  + Primarni a cizi klice + atributy vyskytujici se často v klauzuli where
* Tabulka typu halda
* Shlukovana tabulka

Rizeni soubehu

* Seriovy plan
  + Transakce jsou provedeny za sebou (seriove)
  + Zapisujeme jako entici (A, B)
* Serializovatelny plan
  + Je korektni tehdy a jen tehdy pokud je serializovatelny -> plan je ekvivalentni s vysledkem libovolneho serioveho planu
  + Dvoufazove zamykani