**9. Transakce, zotavení**

**Otázky:**

* Rozdíl mezi okamžitou a odloženou aktualizací
* Co je Commit Point
* Co znamená forma ACID transakcí
* Popište význam jednotlivých vlastností ACID
* Popište operace UNDO a REDO vykonávané při zotavení, uveďte i typy transakcí, které jsou těmito operacemi zpracovány
* K čemu v DBS slouží log soubor? Popište také proč je efektivnější zápis do logu než zápis do DB
* Popište techniku odložené aktualizace. Co a kam se ukládá? Která z operací UNDO a REDO se provádí?

**K čemu v DBS slouží log soubor? Popište také proč je efektivnější zápis do logu než zápis do DB**

* Slouží k ukládání informací o všech provedených operacích
* Systém je na základě logu schopen provést zotavení databáze z chyby (UNDO, REDO operace)
* Při operaci ROLLBACK - log umožňuje vrátit hodnoty záznamu na původní hodnoty před začátkem transakce (do posledního Commit pointu)
* Pravidlo dopředného zápisu do logu – všechny změny musí být zapsány do logu před zápisem do DB
* Je efektivnější, jelikož využívá sekvenčního zápisu, zatímco zápis do databáze využívá náhodný přístup na disk

**Zotavení** znamená zotavení databáze z nějaké chyby. Výsledkem zotavení musí být korektní stav DB.

**Transakce** je logická (atomická) jednotka práce s DB, začíná BEGIN TRANSACTION a končí COMMIT nebo ROLLBACKem. Úkolem transakce je převést korektní stav DB na jiný korektní stav, přičemž nemusí zanechat DB v korektním stavu po jednotlivých aktualizacích transakce.

**COMMIT –** úspěšné dokončení transakce. DB je v korektním stavu, změny můžou být uloženy trvale v DB

**ROLLBACK –** neúspěšné provedení transakce. DB může být v nekorektním stavu, všechny změny provedené v rámci transakce musí být zrušeny

Lokální chyby

* Chyba v dotazu
* Přetečení hodnoty atributu

Chyby globální

* Systémové – výpadek proudu, pád systému či SŘBD (soft crash)
* Chyby média (hard crash)

**Konzistentní DB –** v DB neexistují žádné výjimky z daných integritních omezení

**Korektní –** transakce je posloupnost operací převádějící DB v korektním stavu do jiného korektního stavu. Respektujeme výsledek operací, které jsou vykonány v reálném světě

**Potvrzovací bod**

* Transakce začíná operací BEGIN TRANSACTION a končí COMMIT nebo ROLLBACK
* Operace COMMIT zavádí tzv. Commit point (do logu je zapsán Commit záznam)
* ROLLBACK vrací DB do stavu, ve kterém byla před BEGIN TRANSACTION – vrací DB k předchozímu COMMIT POINTU

Všechny změny musí být zapsány do logu před samotným zápisem změn do DB. Před ukončením vykonávání operace COMMIT je do logu zapsán tzv. COMMIT záznam.

Toto pravidlo nazýváme pravidlo **dopředného zápisu do logu**. Systém je pak schopen na základě logu provést zotavení DB.

**ACID**

* Každá transakce musí splňovat vlastnost ACID
* **Atomičnost –** transakce musí být atomická. Jsou provedeny všech operace transakce nebo žádná
* **Korektnost –** transakce převádí korektní stav DB do jiného korektního stavu, mezi začátkem a koncem transakce nemusí být DB v korektním stavu
* **Izolovanost –** transakce jsou navzájem izolovány: změny provedené jednou transakcí jsou pro ostatní transakce viditelné až po provedení COMMIT
* **Trvalost –** jakmile je transakce potvrzena, změny v DB se stávají trvalými i po případném pádu systému

**Zotavení systému**

* Zotavení je vázáno na celý DB systém. Je nutné ho provést v případě globálních chyb (chyba systému, chyba média..)
* Problémem je ztráta obsahu hlavní paměti (vyrovnávací paměti), log nicméně aktualizace obsahuje
* **UNDO –** přesný stav transakce přerušené chybou není znám, transakce je zrušena
* **REDO –** transakce byla úspěšně ukončena COMMITem, změny ale nebyly přeneseny z logu do DB. Transakce musí být přepracována (redo)
* **Techniky zotavení:**
  + **Odložená aktualizace** (NO-UNDO / REDO)
    - Neprovádí aktualizaci logu a DB až do potvrzení transakce: všechny aktualizace jsou zaznamenány do paměti
    - REDO bude provedeno v případě, kdy systém zapsal aktualizace do logu, ale k zapsání změn do DB nedošlo
    - Pokud transakce selže, není nutné provést UNDO (DB nebyla aktualizovaná)
    - Do logu jsou zapsány nové hodnoty (kvůli REDO)
    - Výkonná, ale hrozí přetečení vyrovnávací paměti
  + **Okamžitá aktualizace** (UNDO / NO-REDO)
    - Provádí aktualizace logu a DB po každé aktualizaci transakce
    - Aktualizace jsou zapsány do logu a poté je aktualizována DB (před provedením COMMIT)
    - Pokud transakce selže před dosažením potvrzovacího bodu, pak je nutné provést UNDO (na disk byly zapsány aktualizace, které musí být zrušeny)
    - Do logu se ukládají původní hodnoty, což umožní operaci UNDO
    - Dochází k velkému počtu zápisu do DB, ale nedochází k přetečení vyrovnávací paměti
  + **Kombinovaná technika** (UNDO / REDO)
    - Aktualizace jsou zapisovány do logu po COMMIT
    - K aktualizaci dochází v určitých intervalech – kontrolních bodech

**Kontrolní body (check points)**

* Vytvářeny po určitém počtu záznamů zapsaných do logu
* Záznam o kontrolním bodu zahrnuje všechny transakce vykonávané v době vytvoření kontrolního bodu a také transakce, které byly ukončeny před kontrolním bodem, ale nebyly uloženy do DB v rámci předchozího kontrolního bodu.

**Algoritmus zotavení**

* Po restartu DBS se spustí algo:
* Vytvoří se 2 seznamy transakcí: UNDO a REDO
* Do UNDO vlož všechny transakce, které nebyly dokončeny před posledním kontrolním bodem. REDO je prázdný.
* Začni procházet záznamy v logu od posledního kontrolního bodu. Pokud je pro transakci T nalezen v logu záznam COMMIT, přesuň T z UNDO do REDO.
* Systém projde log zpětně a ruší aktualizace transakcí z UNDO
* Systém projde logem dopředu a přepracovává aktualizace transakcí ze seznamu REDO
* DB je v korektním stavu

**Záchranné body (savepoints)**

* Rozdělují transakci na menší části. ROLLBACK TO <jmenobodu> vrátí transakci do ZB. SAVEPOINT <jmeno>, RELEASE <jmeno> zrušení