

Kerepes Tamás – Czinkóczki László

Adatbázisok

**Tranzakcionálisan konzisztens működés, záruk
A működés naplózása**



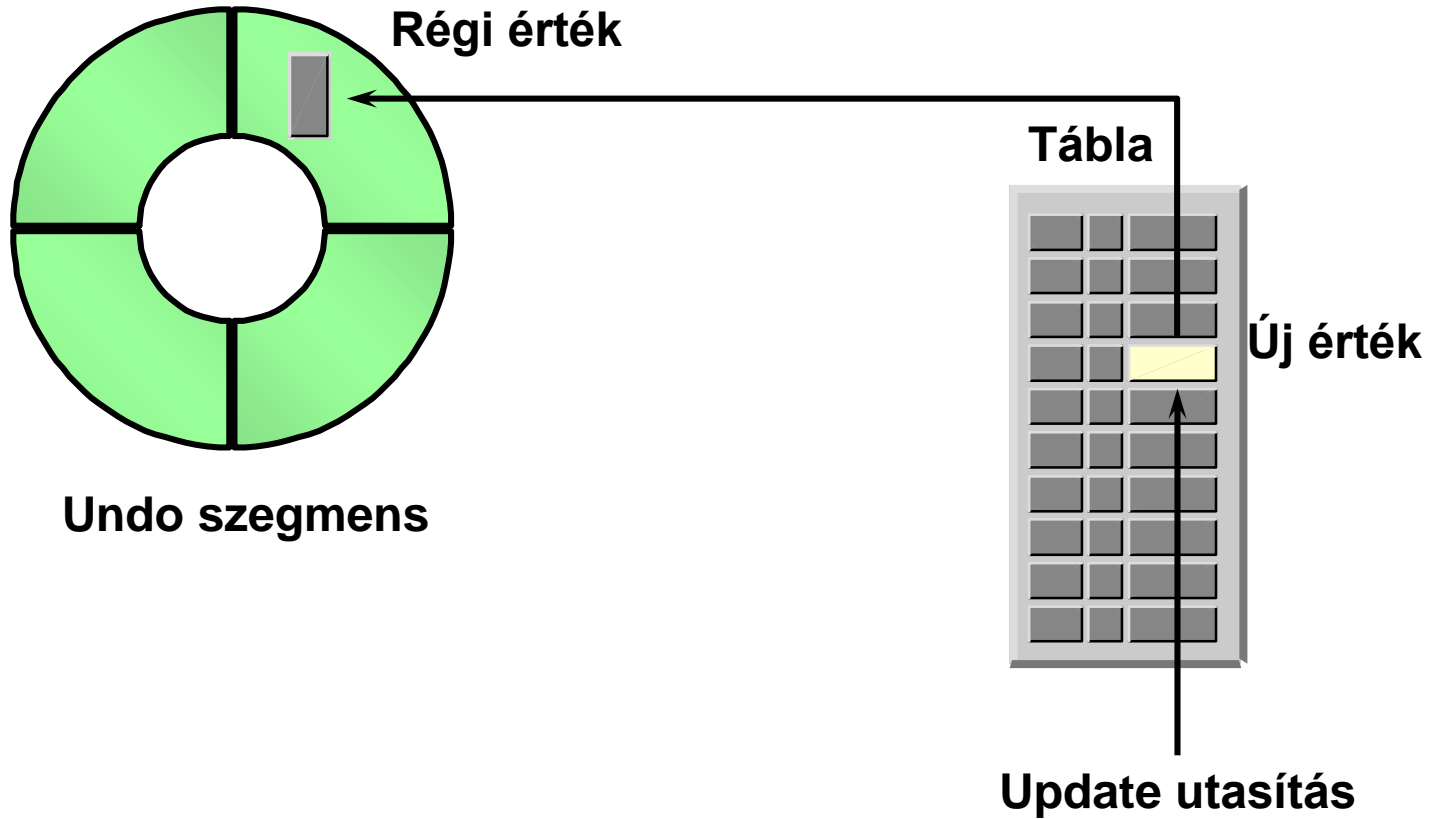
Tranzakciók és adatkonzisztencia

- A felhasználók nemcsak olvassák, hanem módosítják is az adatbázisban lévő adatokat
- A tranzakciók során a változtatásokat bármikor jóvá tudják hagyni (COMMIT), vagy akár visszavonni (ROLLBACK)
- Emiatt a tranzakció során az adatbáziskezelő rendszernek tárolnia kell mind a módosítás eredményeit („After Image”), mind a módosítás előtti – régi – értékeket („Before Image”)
- Amíg a tranzakció nem ér véget, mások nem látják az új értékeket. Ezért nekik az adatbáziskezelő a „Before Image”-et mutatja lekérdezéskor.

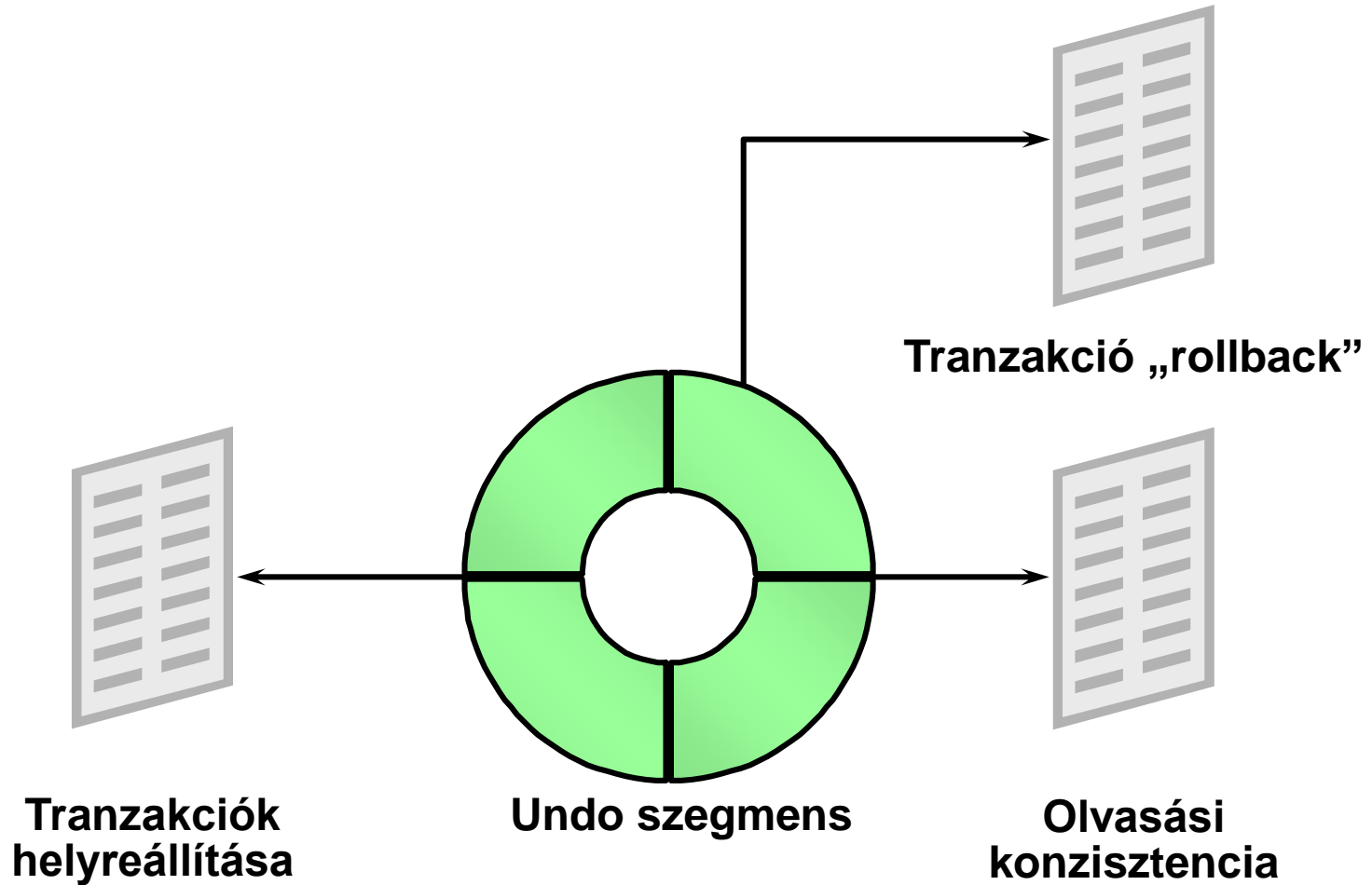
A „régi” és az „új” tárolása

- Vagy az új értéket tároljuk ideiglenes helyen, vagy a régít
- Mivel a tranzakciók nagy többsége COMMIT-tal ér véget, hatékonyabb a régít tárolni „máshol”, míg az újat a táblában – az eredeti helyen
- A régi értékek tárolását az adatbázison belül, egy „speciális” szegmensben tárolhatjuk. Oracle esetén ezt Undo Szegmensnek nevezik. Ez az adatfájlok része.

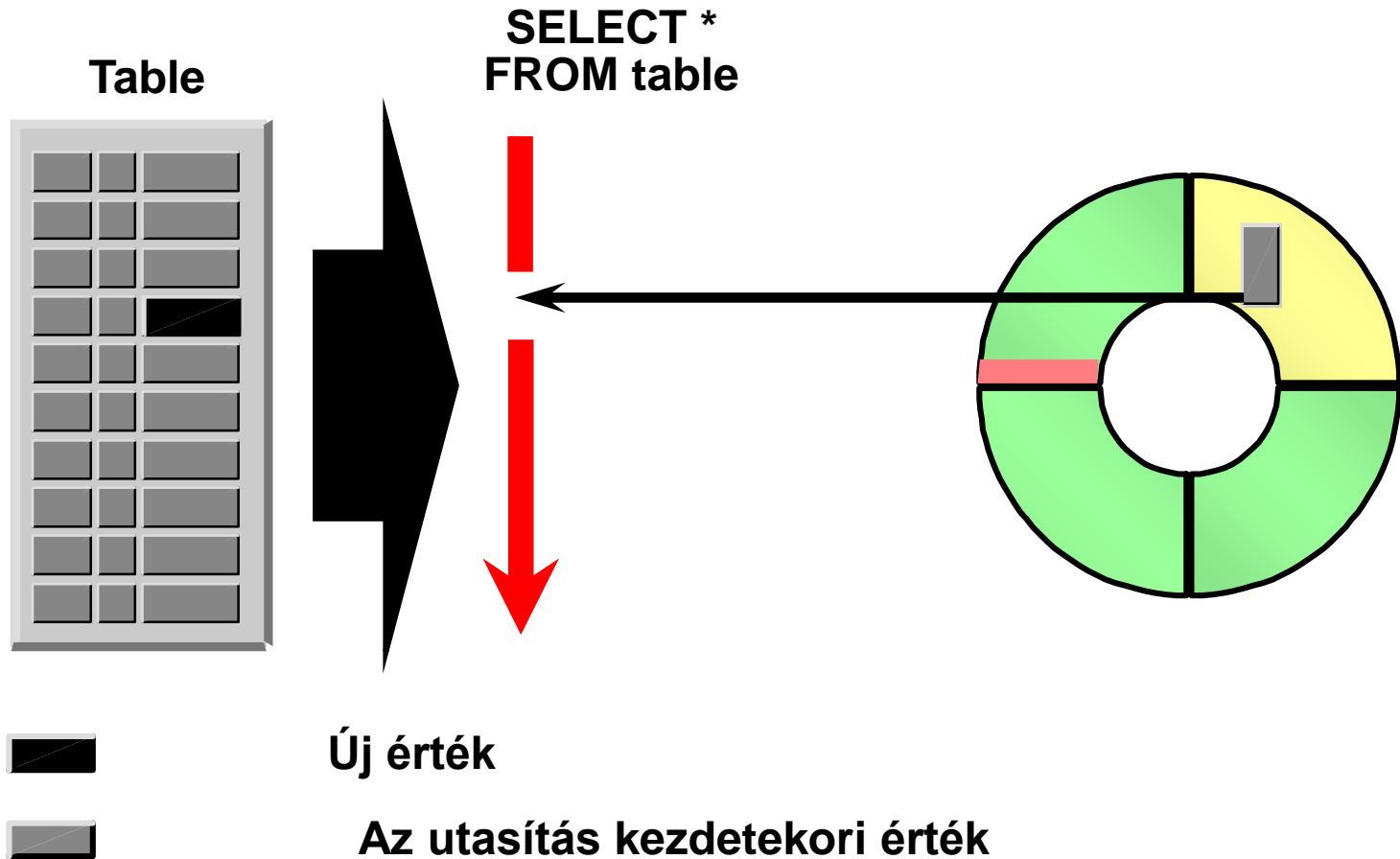
Undo szegmens



Miért kell az Undo szegmens?



Olvasási konzisztencia



A „before image” és az olvasási konzisztencia

- A régi értéket mindenképpen megőrzi a rendszer a tranzakció végéig
- Ezért azok a lekérdezések, amelyek ebben az időszakban futnak, garantáltan sikerülnek
- Nem garantált, hogy egy hosszan tartó SELECT sikerrel végződik. A sikertelen olvasás mégis jobb megoldás, mint a téves eredmény (azaz a „dirty read”)
- A „before image”-et még a COMMIT után is egy ideig igyekszik a rendszer megőrizni, mert „kellhet még”
- Ha az adatok többszörösen módosulnak, akkor az olvasási konzisztencia is többszörös „pointer”-ek mentén találja meg az eredményeit

A zárac („lock”) és a „Before Image”

- A módosításkor a sorokat lezárja „lelock-olja” az adatbáziskezelő rendszer
- Sorszintű és egyéb lockolások
- A Lock megakadályozza, hogy mások módosítsanak, de alapesetben nem akadályoz meg másokat abban, hogy olvassanak. (Lásd még mélyebben: „Repeatable reads”, „Read committed” illetve „Dirty read” működési módok). Viszont nem is a lezárt új értékeket olvassák mások, hanem a „Before Image”-et
- Ha a Lock megakadályozott valaki mást a módosításban, akkor ez a másik tranzakció várakozni fog – lockra vár
- A várakozás alapértelmezetten addig tart, amíg az első tranzakció be nem fejeződik (COMMIT vagy ROLLBACK)

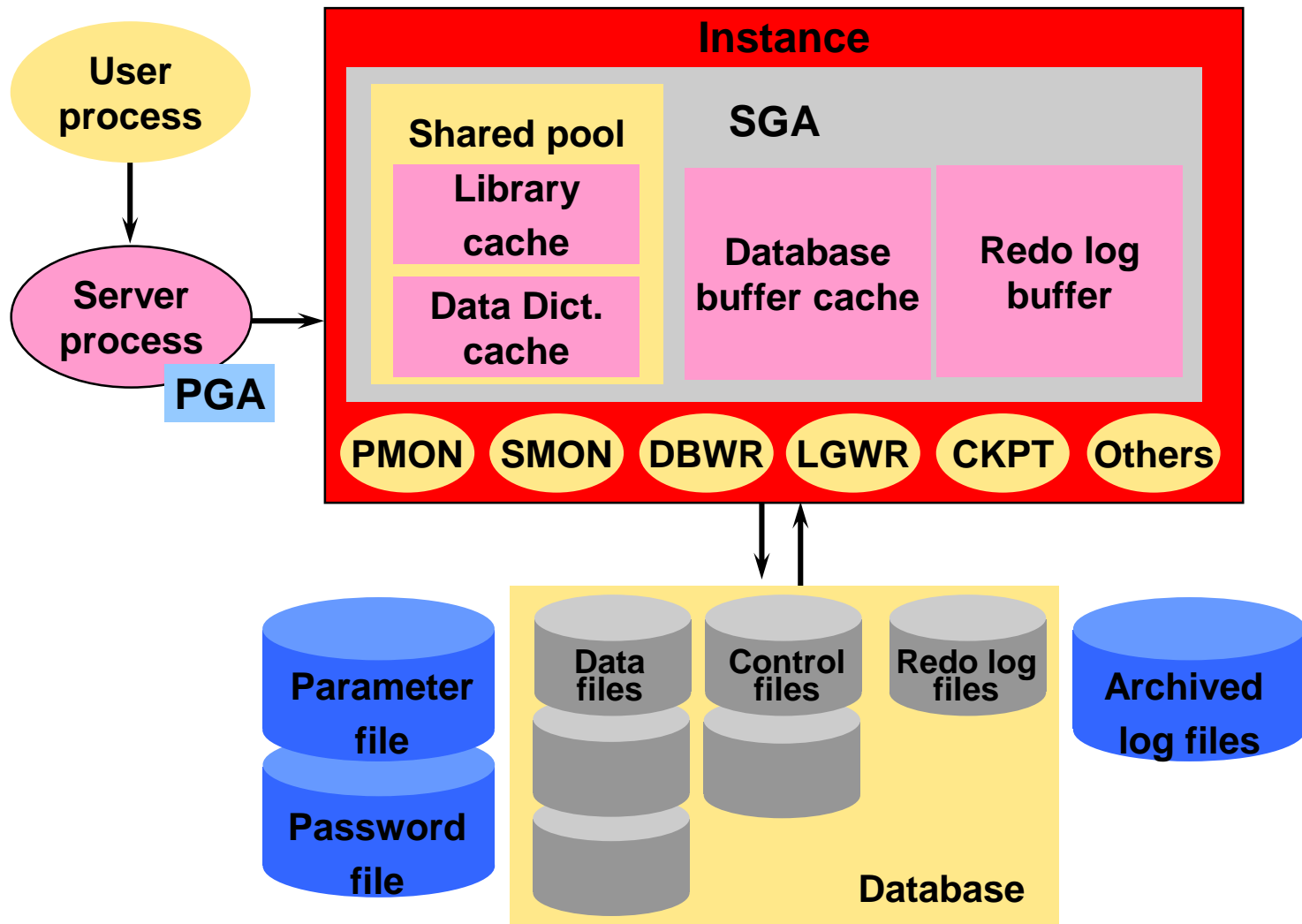
A sikertelen SELECT

- Ha a SELECT sokáig tart, lehet hogy időközben eltűnnek a „Before Image” értékek, amelyekre szüksége van
- Lehet a kudarc oka más természetű is: túl nagy munkaterület kellhet az eredmények vagy a közbülső eredmények előállításához – ez egy másik téma
- SELECT FOR UPDATE – nemcsak olvasni akarunk, hanem egyúttal le is akarjuk foglalni az illető sorokat és várakoznunk kell. Ötlet egy kis kutatáshoz: nézzenek utána az Oracle SQL-je esetén a SELECT FOR UPDATE utasítás NOWAIT klauzulájának, majd próbálják is ezt ki a gyakorlatban. (Figyelem, csak úgy van értelme, ha nem egy, hanem két külön munkamenetben és két külön tranzakcióban folyik a kísérlet)

Az „Undo” és a gyorsítótár viszonya Oracle adatbázis esetén

- Az adatbázis adatállományainak a tartalma a Database Buffer Cache nevű gyorsítótáron keresztül kerül „használatba”. Mivel az Undo is az adatállomány része, erre is vonatkozik ez a cache-elés.
- Az Undo-t felfoghajtuk úgy is, mint egy „speciális” táblát

Emlékeztető: az Oracle adatbázis „architektúrája”



A tranzakciókezelés és a gyorsítótár viszonya

- A tranzakciókezelés (tehát az „undo” kezelése is), és a cache-elés egymásra ortogonálisak. Egyidőben zajlanak, de nem befolyásolják egymást.
- Ez a két esemény egymástól való függetlensége rendkívül fontos
- A cache-elés során azért nem veszítünk adatot, mert zajlik egy másik tevékenység is: a naplózás (redo logging)
- A naplózás a lefagyások és a szabálytalan leállások elleni védelem
- A naplózás teszi lehetővé, hogy az adatok a gyorsítótárból csak jóval később íródjanak ki a végleges adatok az adatállományokba

A naplózás jellemzői

- Szinte minden adatbáziskezelőnél bináris adatok ezek
- Esetleg visszafejthetők belőle a felhasználók DML műveletei, de nem ez az elsődleges funkciójuk
- A naplózás is kezdetben a memóriába íródik egy ciklikus pufferbe – Oracle esetén ez a Redo Log Buffer
- Legkésőbb COMMIT-kor ki kell a naplóbejegyzésnek kerülnie a lemezre
- Oracle esetén ez egy külön fájl típus a lemezen – a „Redo Log File”, pontosabban „Online Redo Log File”
- Ily módon az áramszünet és lefagyás ellen az adatbáziskezelő eleve „meg van védve”

Amire csak a naplóállomány nem elég

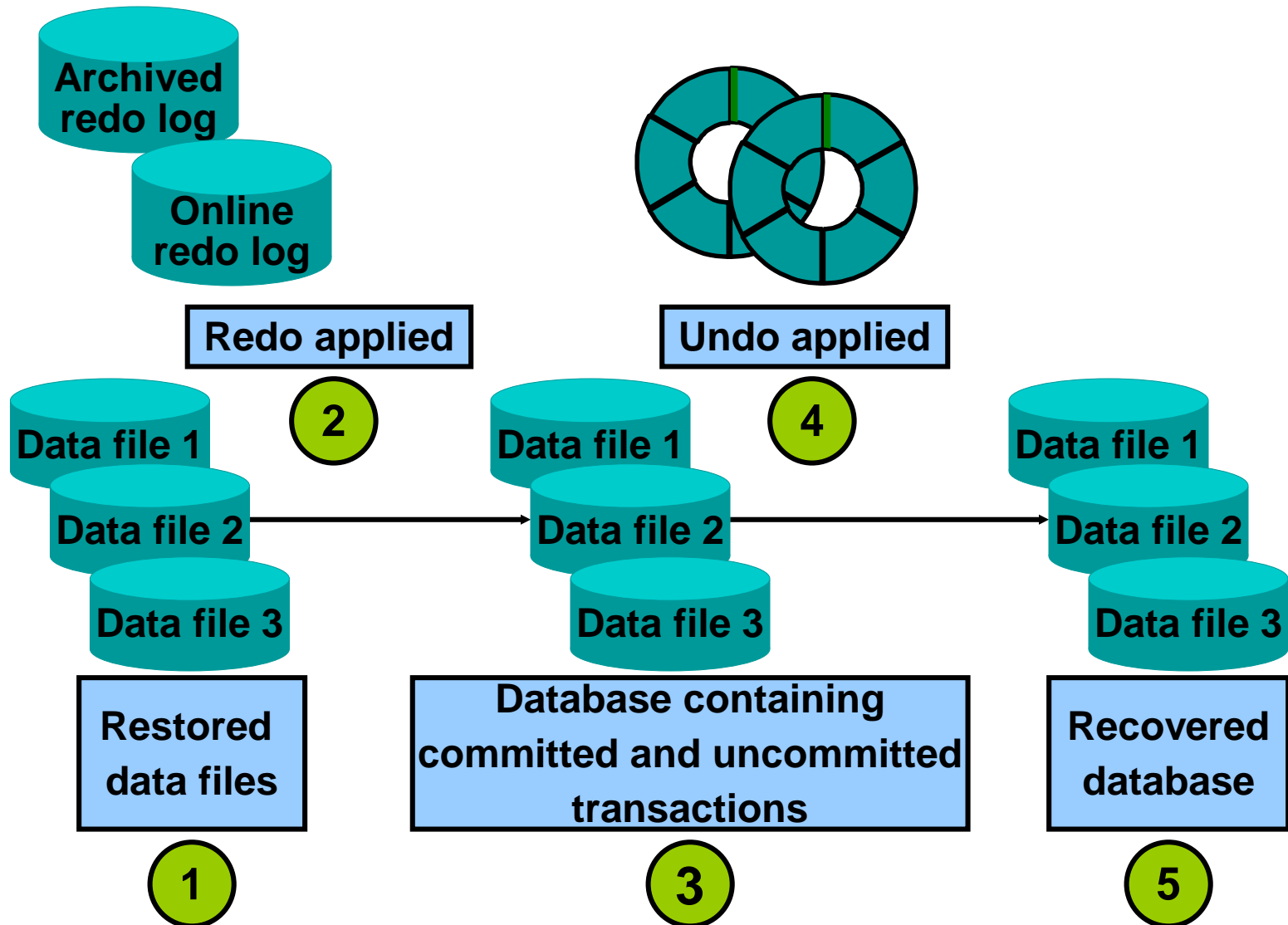
- A naplóállományok nem az idők kezdete óta őrzik az adatokat. Jellemzőn az utolsó 10-20-30 percének a munkája van bennük. Utána jellemzően felülíródnak.
- Ennyi naplóadat bőven elég a lefagyás utáni helyreállításhoz
- Csupán ez azonban nem véd meg minket az adatfájlok elvesztésétől
- Ha lemezhiba ellen is védett adatbázisokat akarunk, akkor még két dolog kell:
 - Időnként menteni az adatbázis fájljait
 - Magakadályozni az online naplóállományok tartalmának az „elvesztését”. Oracle esetén ezt nevezzük archiválásnak

Védettség adatállományok elvesztése esetén

Oracle esetén

- Rendelkezünk egy régebbi mentéssel (backup)
- Be van kapcsolva az archiváló üzemmód és rendelkezünk az archív redo log fájlok szériájával a mentéstől mostanáig
- Ha elveszítünk egy adatállományt, akkor:
 - Visszamásoljuk az illető fájl régi mentését
 - Időrendi sorban rájátszuk a mentés óta az összes naplóbejegyzést az archivált redo log fájlokból, majd az online redo log fájlokból („Rollforward recovery”)
 - Megnyitjuk az így módon helyreállított adatfájlt a „friss” tartalommal
 - Nagyon érdekes automatikus esemény a folyamat utolsó lépése, a „rollback recovery”

Az adatbázis helyreállításának a fázisai



Luxus lehetőségek

- Egyes adatbáziskezelő rendszerek képesek nemcsak előre görgetni az eseményeket az időben, hanem visszafelé is
- Oracle esetén ezt „Flashback Database”-nek nevezik
- Oracle esetén ez opcionális lehetőség. Külön összekonfigurálандó, és külön parancsokkal tudja a rendszergazda elvégezni az „időutazást”.
- Nem azonos ez azzal, amit az Oracle „Flashback Query”-nek nevez, és ami az Undo adatait használja
- Javaslat: keressenek utána ennek a Flashback Query lehetőségnek, és akár próbálják is ki a gyakorlatban