

A

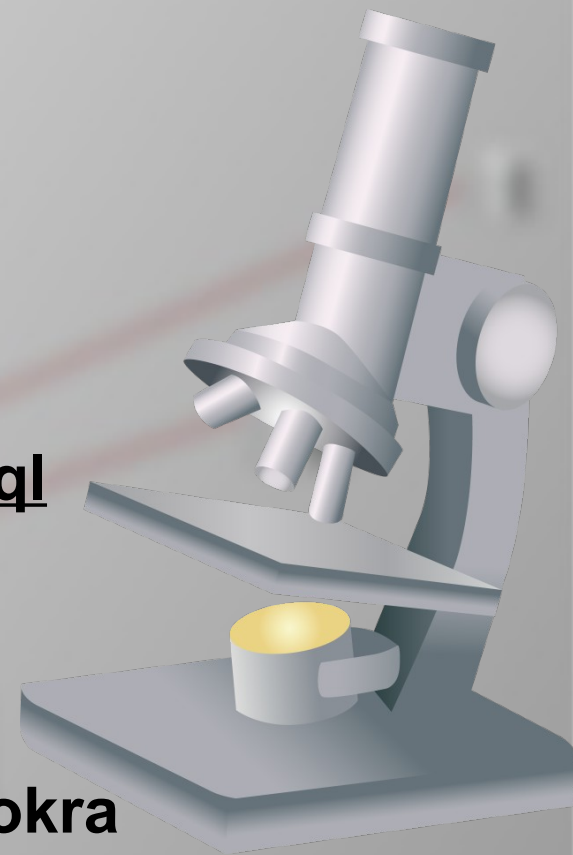
Production Database Reducer

9 tény amit tudni kell a prdare szoftverről



Mi is ez tulajdonképpen?

1. A Production Database Reducer (PDR) igazából egy **algoritmus**
2. **Éles** adatbázis méret **csökkentésére**
(éles adatkör részalmazát képz)
3. **Leginkább tesztelésre és adat archiválásra**
4. Implementáció: **Oracle, Mssql, Db2, Postgresql**
5. **Teljesen paraméterezhető**
6. Nem önálló rendszer, hanem csak **egy eszköz**
7. Javasolt **lépések a méretcsökkentő folyamatokra**
8. Időbeli és adatbázis darabszám-beli **korlátozás nélkül**
9. **További tervek** a funkcionalitás bővítésére



1. A Production Database Reducer (PDR) igazából egy **algoritmus**

Használatot támogató funkciók

Core PDR algoritmus

Van egy core PDR algoritmus, amely bejárja az adatbázist, miközben tesztadatokat válogat le.

Ezt körbeveszik azok a funkciók, amelyek a használatot, működést támogatják.

2. Éles adatbázis méret csökkentésére (éles adatkör részalmazát képzí)

- Nem tömörítés, hanem az éles adatkör egy érvényes részalmazának képzése.
- Az adatok a táblákra képzett kulcsok szerint validak maradnak.
- A PDR segítséget nyújt ahhoz is, hogy a teszt adatkör az alkalmazás logikájának is megfelelő maradjon.
- Tetszőlegesen kicsiny méretekben lehet gondolkodni!

3. Leginkább tesztelésre és adat archiválásra

- **a sokkal kisebb adatbázis méret miatt**
→ *jelentős a költségmegtakarítás (töredék diszk, RAM, CPU mag szám, hálózati forgalom, stb.)*
→ *sokkal több tesztkörnyezet építhető fel*
- **a kisebb méretű adatbázis gyorsabban működik**
→ *redukálható a tesztelési és a fejlesztési idő is*
- **a csökkentett adatbázisba releváns adatok önthetők**
→ *az adott tesztelés a lényeges, tipikus adatokon történhet*
- **a méret csökkentés szinte teljesen automatikus**
→ *jelentős üzemeltetési erőforrás szabadítható fel*
- **az új teszt adatbázis töredék idő alatt visszatölthető**
→ *az újabb tesztkörnyezetek sokkal hamarabb csatarendbe állíthatók*
- **a kevés adattal rendelkező adatbázist sokkal hamarabb lehet deperszonalizálni**
→ *a fejlesztő csapat rövidebb átfutással juthat hozzá a környezetéhez*
- **az adat archiválás automatikus folyamat**
→ *a szükséges adatkört tartalmazó adatbázis is könnyen előállítható*
- **... és még sok minden más, ami a töredéknyi adatbázis méretből fakad**

4. Implementáció: Oracle, Mssql, Db2, Postgresql

A scope:

Implementálni a PrDaRe-t a közép- és nagyvállalatok által leginkább használt adatbáziskezelőkbe.

A PDR algoritmus és minden egyéb PDR funkcionalitás implementálva lett a fentebbi adatbázisokba.

A következőket tartalmazza:

- egy adatbázis schema,
- tárolt eljárások,
- tárolt függvények,
- adattípusok,
- nézetek.

Minden implementáció ugyanazt a funkcionalitást valósítja meg!
(nem pontosan ugyanolyan módon, az adatbázis kezelők képességeitől függően)

5. Teljesen paraméterezhető

PDR változók

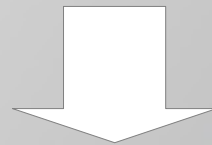
PDR

- A PDR viselkedése *módosítható a változóin keresztül*
- A tesztadatokra és archiválás után *maradó adatokra támasztott feltételek pontosan megadhatók*

A PDR információkat szolgáltat a paraméterezéshez
(*tábla információk, tábla kapcsolatok, foreign szigetek, stb.*)

6. Nem önálló rendszer, hanem csak **egy eszköz**

- Csak egy **eszköz!**
(új rendszer bevezetése egy cégnél probléma lehet)
- **Csak a legszükségesebb funkcionalitás került lefejlesztésre a nagy adatbázisok problémájának megoldására**



- Limitált használhatóság:
*Nincs „frontend”
Adatbázis szintű felhasználó és szerepkör kezelés
Főként üzemeltetők és DBA-k használják*

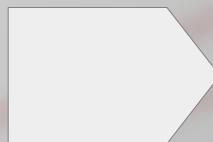
7. Javasolt lépések a méretcsökkentő folyamatokra

Ismerjük az implementációk működését...

...javaslatot tehetünk
a kis adatbázis képzésének konkrét lépéseire:



- **Mikor**
- **Kinek**
- **Mit**



kell tennie

A csatolt dokumentációban olvasható.

8. Időbeli és adatbázis darabszámbeli **korlátozás nélkül**

- ✓ Nincs időkorlát a licenszelt implementációkra!
- ✓ Tetszőleges számú adatbázist lehet csökkenteni vállalaton belül a licenszszerződéssel összhangban.
- ✓ A PDR a teljes életciklusa alatt támogatott!
Életciklusa után a PDR természetesen tovább használható.

9. További tervek a funkcionalitás bővítésére

Vannak terveink!

- új nyelvű implementációk
- univerzális PDR program
- központi tesztadat menedzsment
- adat deperszonalizálás
- rendszerszervezők általi teljes használhatóság



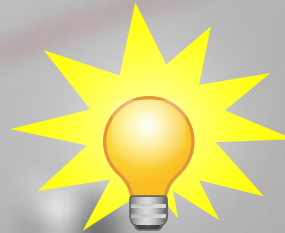
Reagálunk ügyfeleink visszajelzéseire!

A PDR lehető leghasznosabbá fejlesztése egyik legfontosabb célkitűzésünk.

Production Database Reducer



Kérdések?



Production Database Reducer

Köszönöm figyelmüket!

