

# PostgreSQL

Veritabanı Sunucusu

## HOT, VACUUM ve BGWRITER

Devrim GÜNDÜZ

PostgreSQL Geliştiricisi

Command Prompt, Inc.

[devrim@CommandPrompt.com](mailto:devrim@CommandPrompt.com)

[devrim@postgresql.org](mailto:devrim@postgresql.org)[.tr]

20.06.2008  
Ankara

2. Özgür Yazılım Konferansı  
[devrim@CommandPrompt.com](mailto:devrim@CommandPrompt.com)



# Giriş

---

- Bu sunuda aşağıdaki konular anlatılacaktır:
  1. PostgreSQL'de MVCC kavramı
  2. En önemli 8.3 özelliği: HOT
  3. VACUUM nedir?
  4. Bgwriter nasıl çalışır?

# PostgreSQL'de MVCC

- MVCC : Multi Version Concurrency Control
  - SELECT işlemleri UPDATE, DELETE VE INSERT işlemlerini beklemez.
  - Tersisi de geçerlidir.
  - Bu sayede lock işlemleri sunucuya yük getirmez.
- MVCC, bir transaction'ın kendisinden sonra başlayan tx'lerin görülmesini engeller.

# PostgreSQL'de MVCC

- Bir satırın birden fazla sürümü oluşur, bu da diskte yer kaplar ve başarımla düşmesine neden olabilir.
- Bakım gereksinmesini artırır.
- Bir ölü tuple, eğer bir index tarafından hala gösteriliyorsa, o zaman kaldırılamaz.

# MVCC nasıl çalışır?

- Her satır için 2 txid:
  - O satırı oluşturan tx'in yaratma id'si
  - O satırın süresini dolduran tx'in id'si
- Update işlemi: Yeni satır yaratılır ve eskisi expire olur.
  - Yani, aynı satırın iki farklı sürümü oluşturulur.
- Satır seviyesinde kilitleme (row-level locking) kullanan veritabanları, veri bütünlüğünü sağlamak için kilitleme mekanizmasına (locking) gereksinim duyar.

# MVCC nasıl çalışır?

- Bir tx başladığında, o transaction'a aşağıdaki satır sürümleri gösterilebilir:
  - Satırın yaratılma txid'si commit edilmişse ve de mevcut tx sayacından düşükse
  - Satırın süresi dolmamışsa (expiration txid) ya da bu txid sorgu başladığında henüz commit edilmemişse (=hala çalışıyorsa)

# VACUUM

---

- Vacuum işte burada devreye girer.
- Free Space Map: Kullanılmayan alanların saklandığı yer.
  - Bu değer düşük olursa, diskte olan ama bu map içinde olmayan alan kullanılamaz.
  - Bu da diskte kullanılmayan alanların birikmesine neden olur.
    - Başarım sorunu!

# VACUUM

- VACUUM VERBOSE çıktısı:
- INFO: free space map contains 10020 pages in 931 relations  
DETAIL: A total of 23520 page slots are in use (including overhead).  
23520 page slots are required to track all free space.  
Current limits are: 2048000 page slots, 2000 relations, using 12127 kB.

# VACUUM

- Normal VACUUM kirli satırları tekrar kullanılmak üzere PostgreSQL'e verir.
- VACUUM FULL ise bu satırların olduğu disk alanını işletim sistemine verir - DİKKAT: Tabloları kilitler!
- I/O gereksinimi fazladır; bu nedenle düzgün yapılandırmak gereklidir.

# HOT

- Heap Only Tuple,
  - Sıkça güncellenen tablolardaki **bazı** başarım sorunlarını çözüyor.
  - Şu ana kadarki en büyük yama
- UPDATE sorgularının indexlenmiş herhangi bir kolonu ya da index ifadeleri tarafından çağırılan kolonlarını güncellemediği durumlarda HOT devreye girer ve sık VACUUM gereksinimini ölü alanın üzerine bilgileri yeniden yazarak azaltır.

# HOT için bir örnek

- T1 tablosunda c1 ve c2 kolonları, ve c1 kolonunda da index olsun.
  - \* UPDATE t1 SET c2='Güneş Deniz';
  - \* UPDATE t2 SET c1 = '2007';
- İlk sorgu HOT için uygundur; çünkü index alanını değiştirmiyor.
- İkinci sorgu ise HOT için uygun değildir.

# BGWRITER

- postmaster diskten veri okurken veriler öncelikle shared buffer (SB) içine atılır. SB dolduğu zaman postmaster süreci eski bilgileri SB dışına atar ve yenilerine yer açar.
- Eğer SB içindeki bir sayfa değişmişse bunların diske yazılması gerekir.

20.06.2008  
Ankara

2. Özgür Yazılım Konferansı  
devrim@CommandPrompt.com



# BGWRITER

- BGWRITER (Background Writer)
- Değişmiş sayfaları diske yazar
- Genelde “uyku” durumunda bekler.
- 8.0 itibariyle ayrı bir süreç olarak çalışır:

20.06.2008  
Ankara

2. Özgür Yazılım Konferansı  
devrim@CommandPrompt.com



# BGWRITER

- Eğer SB daha fazla sayfa tutabilirse bgwriter daha az çalışacak ve diske yazma işlemleri azalacaktır.
- CHECKPOINT işlemlerindeki yük bgwriter sayesinde azalır - CP, SB içinde artık daha az değişmiş sayfa bulacaktır.

20.06.2008  
Ankara

2. Özgür Yazılım Konferansı  
devrim@CommandPrompt.com



## 8.3 ve CHECKPOINT ve bgwriter iyileştirmeleri

- Bu sürümle CHECKPOINT işlemleri daha uzun sürelerle yayılabilir.
- Writer sürecindeki tampon havuzunu (buffer pool) tarayan ve yazan kısım artık bu sürümde yok.
- “Just-in-time” background writer
  - Daha az tarama, daha az I/O gereksinimi
  - Ön tanımlı olarak 4 MB/s yazabilme (ayarlarla arttırılabilir)

# Kaynaklar

---

- <http://www.postgresql.org/>
- Vacuum ve bgwriter: Alvaro Herrera - Command Prompt, Inc.
- <http://pgcon.org/2008> - Pavan Deolasse - HOT
- <http://www.postgresql.org/docs>
- <http://wiki.PostgreSQL.org>
- <http://www.pgfoundry.org/>

**20.06.2008**  
**Ankara**

**2. Özgür Yazılım Konferansı**  
**devrim@CommandPrompt.com**



# Seminerin Güncel Hali

---

- <http://www.gunduz.org/seminer/pg>
- <http://www.postgresql.org.tr/>

**20.06.2008**  
**Ankara**

**2. Özgür Yazılım Konferansı**  
**devrim@CommandPrompt.com**



# PostgreSQL

Veritabanı Sunucusu

## HOT, VACUUM ve BGWRITER

Devrim GÜNDÜZ

PostgreSQL Geliştiricisi

Command Prompt, Inc.

[devrim@CommandPrompt.com](mailto:devrim@CommandPrompt.com)

[devrim@postgresql.org](mailto:devrim@postgresql.org)[.tr]

20.06.2008  
Ankara

2. Özgür Yazılım Konferansı  
[devrim@CommandPrompt.com](mailto:devrim@CommandPrompt.com)

