ANALİTİK PANEL · 2026

BiSoft® Bilgi Teknolojileri

PostgreSQL

PostgreSQL'in Modern İş Analitiğindeki Yeri

Serdar Güler
Berna Ağababaoğlu

Mimarideki konumu, alternatiflerle kıyası, extension ekosistemi ve sektörlerden gerçek senaryolar.

BiSoft Bilgi Teknolojileri A.Ş. · Ankara

GÜNDEM

Bu sunumda neyi konuşacağız?

Beş başlıkta PostgreSQL: neden, nerede, nasıl — ve sektörden somut kanıtlar

01 **Neden PostgreSQL, neden şimdi?**
Olgunluk · ekosistem · ekonomi

02 **PostgreSQL vs alternatifler**
Oracle, SQL Server, MySQL ve ClickHouse karşısında konum

03 **Analitik mimaride referans tasarım**
Kaynaktan KDS'ye tek PostgreSQL omurgası

04 **Extension ekosistemi**
Citus · PostGIS · TimescaleDB · AGE · pgvector

05 **Sektörden gerçek senaryolar ve kaynaklar**
Kamu · Bankacılık · Enerji · Fintech

Neden PostgreSQL? Neden şimdi?

Ticari DBMS'nin yerine değil — çoğu analitik projenin doğal başlangıç noktası

4 kez

DB-Engines 'Yılın Veritabanı' — 2017, 2018, 2020, 2023 · 2024'te Snowflake'in ardından ikinci

30+ yıl

1996'dan beri kurumsal sistemlerde · kökeni 1986 Berkeley araştırmalarına uzanır

Ücretsiz

Lisans bedeli yok, vendor kilidi yok · OSI (Open Source Initiative) onaylı PostgreSQL/BSD (Berkeley Software Distribution) lisansı — özgürce dağıtılır

Kurumsal olgunluk

Banka işlemi güvenilirliği, eş zamanlı milyonlarca sorgu, canlı kopya ve otomatik devir — operasyonel ve analitik yükü aynı motorda taşıyan 30+ yıllık mühendislik.

Tek motor, çok yetenek

Aynı motor üzerinde **dağıtık ölçek (Citus)**, **coğrafi analitik (PostGIS)**, **zaman serisi (TimescaleDB)**, **ilişki ağı (Apache AGE)**, **AI vektör araması (pgvector)**, **metin benzerliği (pg_trgm)** ve **Data Lake köprüsü (pg_duckdb, parquet_fdw)** — yeni iş yükleri için ek ürün veya lisans gerekmez.

Ekonomi & egemenlik

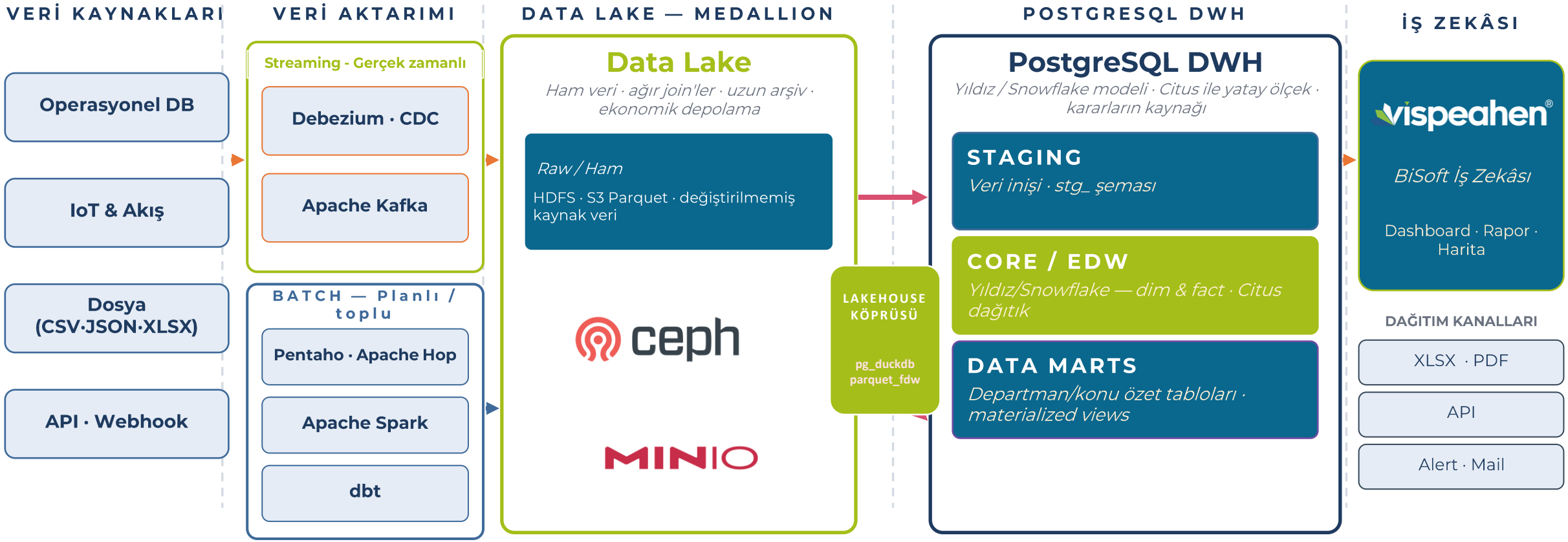
Lisans ücreti yok · bulut, şirket içi ve hibrit ortamlarda aynı motor · veriniz sizin · geniş Türkiye ekosistemi ve 7/24 uzmanlık erişimi.

PostgreSQL alternatifler karşısında ne konumda?

Kriter	PostgreSQL	Oracle DB	SQL Server	MySQL	ClickHouse
Lisans maliyeti	**Yok (açık kaynak)**	Çok yüksek (kullanıcı/CPU)	Yüksek (Core/CAL)	Topluluk: yok · Enterprise: ücretli	Yok (açık kaynak)
Analitik extension'lar	**Citus · PostGIS · Timescale · AGE · pgvector · pg_duckdb · parquet_fdw**	Oracle Spatial, Partitioning, Advanced Analytics — ek lisans	Bazı analitik özellikler dahili (CDC, Columnstore)	Sınırlı; analytics için MariaDB	Native columnar (MergeTree); MPP; vektörize sorgu motoru
Coğrafi / CBS	**PostGIS — açık kaynak referans uygulama**	Oracle Spatial (ek lisans)	SQL Server Spatial (sınırlı)	Sınırlı	Sınırlı
Zaman serisi / IoT	**TimescaleDB (native extension)**	Sınırlı (manuel partitioning)	Sınırlı (manuel partitioning)	Yok	Sınıfının en iyilerinden
Vektör / AI	**pgvector (native)**	Oracle 23ai Vector	Vector type (preview)	Yok	Native vector search
Vendor kilidi	**Yok**	Yüksek	Yüksek	Çok Düşük	Yok

Data Lake + PostgreSQL DWH — BiSoft Referans Mimarisi

Streaming + Batch ile beslenen iki katmanlı mimari · Lake ile DWH arasında çift yönlü Lakehouse köprüsü



NE ZAMAN DATA LAKE?
Çok büyük dosyalar (Parquet/ORC) · uzun arşiv · ağır join'ler

NE ZAMAN POSTGRESQL DWH?
Modellenmiş gerçek kaynağı · BI sorguları · hızlı yanıt

NE ZAMAN LAKEHOUSE (HİBRİT)?
Karar verisi DWH'da · ham/keşif Lake'te · pg_duckdb / parquet_fdw köprüsü

PostgreSQL'i analitiğe taşıyan kaldıraç: extension'lar

Tek motor, çok yetenek - ayrı ürün satın almadan yeni iş yüklerine uzanın

DAĞITIK SQL

Citus

PostgreSQL'i çoklu sunucuya dağıtır · analitik sorgularda 100x+ hızlanma · 5–10x kolon bazlı sıkıştırma.

COĞRAFİ

PostGIS

Coğrafi olarak sorgulanan her veri için 24 yıldır endüstri standardı · QGIS (Quantum Geographic Information System), ArcGIS, GeoServer doğal konuşur.

ZAMAN SERİSİ

TimescaleDB

Sensör, sayaç, işlem verisi · %90+ sıkıştırma · sürekli özet tabloları · retention policy ile otomatik temizlik.

İLİŞKİ AĞI

Apache AGE

Neo4j muadili · ilişkisel + graph aynı motorda · openCypher destekli · Apache Software Foundation Top-Level projesi.

VEKTÖR AI

pgvector

AI uygulamaları için vektör saklama ve benzerlik arama · AWS, Azure, GCP'nin yönetilen PostgreSQL'lerinde hazır.

METİN ARAMA

pg_trgm + FTS

Yazım hatalı eşleşme + Türkçe tam metin araması · müşteri birleştirme ve kurumsal aramanın sessiz kahramanı.

LAKEHOUSE

pg_duckdb

DuckDB motorunu PostgreSQL'e taşır · Parquet, CSV, Iceberg dosyalarını S3/GCS/Azure'dan SQL ile sorgulayın.

DOSYA SORGUSU

parquet_fdw

Lake'teki Parquet dosyalarını PostgreSQL'de yabancı tablo olarak gösterir · veri taşımadan analiz.

Citus — Aynı PostgreSQL, dağıtık güç

Coordinator + Worker mimarisi · 2019'dan beri Microsoft destekli açık kaynak

NE İŞE YARAR?

Citus, PostgreSQL'i **çoklu sunucuya dağıtık bir analitik motora** dönüştürür. Tablolarınız dağıtım kolonu üzerinden shard'lara bölünür; sorgular **Coordinator** tarafından planlanır, **Worker** sunuculara paralel gönderilir, sonuç birleşmiş olarak istemciye döner.
Tüm sunucularda standart PostgreSQL + Citus extension'u çalışır. Uygulama tek bir PostgreSQL adresine bağlanır; değişiklik gerektirmez.

NEREDE FARK YARATIR?

- Yatay ölçek — Yeni Worker sunucu eklenir, kapasite genişler; uygulama kodunda tek satır değişmez.
- Paralel sorgu — Büyük fact tablolarında sorgu Worker'lar arasında eş zamanlı koşar; analitik raporlar 100 kat ve üzeri hızlanabilir.
- Kolon bazlı sıkıştırma — 5–10 katı sıkıştırma ile disk ve I/O maliyeti belirgin düşer; aynı donanımla daha çok veri saklanır.
- Dağıtık tablolar (distributed) — Tabloyu hangi kolona göre ve kaç parçaya (shard) böleceğinizi tablo oluşturulurken siz belirlersiniz — büyük fact tabloları için ideal.
- Referans + lokal tablolar — Küçük sözlük/parametre tabloları tüm Worker'lara kopyalanır (reference table → join'ler hızlı), çok küçük yardımcı tablolar yalnız Coordinator'da kalır (local table); ağ trafiği azalır.

NE ZAMAN SEÇİN?

Üç sinyalden biri tetiklendiğinde Citus devreye girmeli:

- Veri hacmi — fact tablolar milyar satıra yaklaştığında
- Sorgu süresi — dashboard'lar saniye altı beklerken raporlar dakikalara çıktığında
- Sorgu karmaşıklığı — büyük GROUP BY ve JOIN'ler tek motorda paralelleşemediğinde

100×+

Microsoft / Citus Data resmi benchmark'larında analitik sorgularda **100 kat ve üzeri hızlanma** ölçüldü. Pratikte sonuç: gece beklenen rapor sabah masada, saatlerce süren analiz dakikalarda; yönetim canlı veriyle karar verir.

PostGIS — PostgreSQL'in coğrafi analitik gücü

Kamu CBS sistemlerinin açık kaynak motoru · 2001'den beri kurumsal sahnede

NE İŞE YARAR?

PostGIS, PostgreSQL'i tam donanımlı bir **coğrafi veritabanına** dönüştürür. Parsel, hat, tesis, rota gibi **vektör nesneler** ve uydu görüntüsü, yükseklik modeli, yağış gridi gibi **raster veriler** aynı motorda durur; uzaklık, kesişim, en yakın komşu ve alan hesabı gibi mekânsal sorgular tek SQL satırı ile yazılır.
QGIS, ArcGIS, GeoServer dâhil tüm büyük CBS araçları PostGIS'i doğal olarak destekler. Uluslararası **OGC (Open Geospatial Consortium)** standartlarına tam uyumlu olduğu için verileriniz hangi araca taşınırsa taşınsın okunabilir — **vendor kilidi yoktur**.

NEREDE FARK YARATIR?

- Mekânsal indeksleme — milyonlarca geometri (parsel, hat, tesis) üzerinde GiST / SP-GiST indeksleri sayesinde milisaniye seviyesinde sorgu
- Vektör ve raster tek motorda — nokta/çizgi/poligon nesneler ile uydu görüntüsü, yükseklik modeli, yağış gridi yan yana saklanır
- 300+ mekânsal fonksiyon — en yakın şube, en kısa rota, servis bölgesi, kesişim alanı: hepsi tek SQL sorgusu ile
- QGIS, ArcGIS Pro, GeoServer ile native entegrasyon — mevcut CBS ekipleri ve araç yatırımları olduğu gibi kullanılır
- Açık kaynak CBS dünyasının fiili referans motoru — kamu, su/enerji altyapısı, lojistik ve perakende lokasyon analitiğinde 24+ yıllık üretim deneyimi

NE ZAMAN SEÇİN?

Üç sinyalden biri tetiklendiğinde PostGIS devreye girmeli:

- Lokasyon filtresi — "en yakın", "içinde", "kesişen" sorularına SQL ile cevap aranıyorsa
- Mekânsal agregasyon — bölge / poligon / mesafe bandı başına KPI, talep, gelir hesabı yapılıyorsa
- Çok katmanlı analiz — vektör (parsel, rota) ve raster (uydu, yağış, yükseklik) verisi tek sorguda buluşturulacaksa

25 yıl

2001'den beri kurumsal CBS sistemlerinin arkasında. Uluslararası OGC standardıyla tam uyumlu; tüm büyük CBS ürünlerinde destekleniyor.

TimescaleDB - sensör, sayaç ve işlem verisinin doğal yuvası

Tek motorda dört avantaj: retention · sürekli özetler · zaman partition · space dimension

NE İŞE YARAR?

TimescaleDB, PostgreSQL'i zaman-eksenli verilere özel olarak güçlendirir. Sensör ölçümleri, akıllı sayaç kayıtları, finansal işlemler — saniyede binlerce satır üreten her veri için disk maliyetini onda bire indirir, sürekli özet tabloları ile tarihsel sorguları 1000 kata kadar hızlandırır. Çekirdek hypertable yapısı veriyi otomatik olarak zamana göre parçalara böler.

NEREDE FARK YARATIR?

- Retention policy: belirli süre öncesi veriler otomatik silinir — manuel temizlik joblarına gerek yok
- Continuous aggregates: sürekli güncellenen özet tabloları — dashboard'lar milisaniyede açılır
- Hypertable + time partition: zaman bazlı bölünme kendinden gelir, planlama ve indeks bakımı kolaylaşır
- Space dimension: zamanın yanına ek bir dimension (cihaz, tesis, müşteri) tanımlayarak yatayda da bölme
- %90+ kolon bazlı sıkıştırma · Arka planda otomatik · Eski veri ekonomik tutulur
- Not: Multi-node hypertable community sürümünde deprecated · büyük dağıtım için Citus tarafı önerilir

NE ZAMAN SEÇİN?

Üç sinyalden biri tetiklendiğinde TimescaleDB devreye girmeli:

- Zaman penceresi — "son 24 saat", "geçen çeyrek", rolling window gibi zaman-filtreli sorgular ağır basıyorsa
- Zaman bucket'ı — saatlik / günlük / aylık agregasyon, downsampling, gap-fill gerekiyorsa
- Sürekli akış — yüksek hacimli append-only insert, retention ve sıkıştırma birlikte yönetilecekse

%90+ **1000×**

Resmi dokümantasyondaki tipik kompresyon oranı ve sürekli özet tablolarında ölçülen sorgu hızlanması. Hem disk maliyetini hem dashboard yanıt süresini belirgin düşürür.

Aynı motorda ilişki ağı, AI vektörü ve metin eşleştirme

Ayrı bir graf veritabanı, ayrı bir vektör veritabanı, ayrı bir arama motoru yerine tek PostgreSQL

İLİŞKİ AĞI · GRAPH

Apache AGE

Neo4j muadili — ama belirgin farkla: Neo4j sadece graph veritabanıdır; Apache AGE ile aynı motorda hem ilişkisel (SQL) hem graph (openCypher) çalışır. Aynı tablonun üzerinde SQL JOIN ile Cypher pattern matching yan yana yazılır. Müşteri ağları, ortak cihaz/IP izleri, AML, müşteri 360 — ayrı graf veritabanı kurma yükü yok.

SQL + Cypher

İlişkisel + graph tek motorda · openCypher destekli · Apache Software Foundation Top-Level projesi (2022)

VEKTÖR AI

pgvector

AI asistanlar için kurumsal bilgi kaynağı, semantik arama, anomali ve sahtecilik tespiti. AWS, Azure ve GCP'nin yönetilen PostgreSQL servislerinde hazır — ayrı AI altyapısı kurmak gerekmez.

3 büyük bulutta

AWS, Azure, GCP yönetilen PostgreSQL'lerinde hazır — kurulum ve entegrasyon yükü yok

METİN ARAMA

pg_trgm + FTS

'Mehmet Yılmaz' = 'Mehmet Yilmaz' = 'M.Yılmaz' eşleşmesini yakalar; Türkçe metin aramasını destekler. Müşteri birleştirme ve kurumsal arama senaryolarının omurgası.

Milisaniye

Milyonlarca kayıt içinde yazım hatalı eşleştirme ve tam metin araması milisaniye seviyesinde

Hangi durumda PostgreSQL+Citus, hangi durumda ClickHouse?

Temel ilke: PostgreSQL ekosistemi ihtiyacınızı karşılıyorsa onunla devam edin — ClickHouse'a erken geçiş karmaşa yaratabilir.

POSTGRESQL + CITUS - TERCİH EDİN, EĞER:

İşlem ve analiz aynı veri üzerinde çalışıyorsa: Uygulama veriyi yazıp güncellerken raporlama da aynı veriyi okuyor — OLTP ve OLAP'ı ayırmaya gerek yok, tek motor yeter.
Fact tablo ölçeği büyük ama dev değilse: Yüz milyonlardan birkaç milyar satıra kadar — Citus'un dağıtık mimarisi ve kolonsal sıkıştırması bu aralığı rahat taşır.
Karmaşık ilişkisel SQL zorunluysa: Çoklu tablo JOIN'leri, CTE'ler, window function'lar, foreign key, transaction ve constraint'ler iş mantığının merkezinde.
Güncelleme ve silme sık yapılıyorsa: KVKK silme talepleri, satır düzeltmeleri, statü değişiklikleri — ClickHouse gibi motorlarda mutation maliyeti pahalıdır; PostgreSQL bu senaryoda doğal davranır.
Tek motorda farklı analitik yetenekler gerekiyorsa: Coğrafi sorgular (PostGIS), zaman serisi (TimescaleDB) ve vektör araması (pgvector) aynı veritabanında yan yana çalışır.
Ekip ve araçlar PostgreSQL'e hazırsa: DBA bilgisi, ORM kütüphaneleri, BI araçları, migration ve izleme çözümleri — olgun ve geniş bir ekosistem hazır bekliyor.
Operasyonel sadelik öncelikse: Tek motor, tek yedekleme stratejisi, tek HA mimarisi — yönetilecek bileşen sayısı minimumda kalır.

CLICKHOUSE 26.x - TERCİH EDİN, EĞER:

İş yükü saf okuma-toplama ve veri sürekli ekleniyorsa: Log kayıtları, kullanıcı tıklama akışı, metrik/trace toplama, IoT telemetrisi — append-only karakterde, neredeyse hiç güncelleme yok.
Veri hacmi gerçekten dev seviyedeyse: Günde milyarlarca yeni satır, toplamda onlarca–yüzlerce TB — bu ölçekte Citus dahi zorlanır; ClickHouse'un kolonsal mimarisi devreye girer.
Saniye altı yanıt + yüksek eşzamanlılık zorunluysa: Yönetim panelleri sürekli açık, yüzlerce kullanıcı aynı anda dev veri üzerinde toplama sorguları çalıştırıyor — düşük gecikme bir tercih değil, gereklilik.
Disk ve bellek maliyeti karar verici hale geldiyse: ClickBench testlerinde aynı 100 milyon satır PostgreSQL'de ~100 GiB yer kaplarken ClickHouse'da ~9.3 GiB'a iniyor — yaklaşık 10× depolama tasarrufu.
Zaman serisi trilyon satır eşiğine ulaşmışsa: Daha küçük ölçeklerde TimescaleDB iyi bir tercihtir; ancak bu boyutta ClickHouse'un projection ve materialized view esnekliği belirgin biçimde öne çıkar.
Veri zaten düzleştirilmiş tek tabloya akıyorsa: Kafka'dan tek satırlık event'ler geliyor, JOIN ihtiyacı minimum — denormalize edilmiş wide-table modeli ClickHouse mimarisiyle birebir örtüşür.

Olgun mimarilerde iki sistem birlikte yaşayabilir: operasyonel veri PostgreSQL+Citus'ta tutulur, CDC (Debezium / Kafka) ile ClickHouse'a akıtılır ve en ağır analitik yük orada çalışır — birbirinin rakibi değil, birbirini tamamlayan katmanlar.

JOIN motorları - algoritmalar, varsayılan davranış ve dağıtık model

JOIN motorları — algoritmalar, varsayılan davranış ve dağıtık model *ClickHouse 26.x varsayılan sırası: direct → parallel_hash → hash · Citus stratejisi: co-located, reference ve repartition JOIN üçlüsü*

POSTGRESQL + CITUS — JOIN MODELİ

Algoritmalar: Nested Loop, Hash Join, Merge Join — PostgreSQL'in klasik üçlüsü, planlayıcı maliyete göre seçer.

Co-located JOIN: İki tablo aynı dağıtım kolonu üzerinden bölünmüşse JOIN her worker'da lokal çalışır, sunucular arası ağ trafiği oluşmaz.

Reference tablo: küçük lookup tabloları tüm worker'lara replike edilir, JOIN saniye altında tamamlanır.

Local tablo: yalnız Coordinator'da kalan çok küçük yardımcı tablolar, ağ trafiği oluşturmaz.

Repartition JOIN: Farklı kolonlara bölünmüş büyük×büyük tablolarda veri worker'lar arası yeniden dağıtılır; maliyetlidir ama gerektiğinde otomatik devreye girer.

Olgun planlayıcı + disk spill: 3+ tablolu sorgularda JOIN sırasını otomatik optimize eder; bellek yetersiz kalırsa diske geçer, OOM yaşanmaz.

CLICKHOUSE 26.x — JOIN MODELİ

Varsayılan algoritma sırası: direct → parallel_hash → hash (24.12 sürümünden itibaren bu sırayla denenir).

Direct JOIN: sözlük (dictionary) üzerinden tek adımda O(1) arama — yalnız LEFT ANY destekler, yıldız şemada en hızlı seçenek.

Parallel hash: varsayılan 16 bucket, çoklu thread paralel hash table kurar — hızlıdır ama RAM yoğun çalışır.

Hash (klasik): tek thread çalışan jenerik algoritma — tüm JOIN tiplerini destekler, kısıtsızdır ama yavaştır.

Grace hash: Bellek güvenli mod; RAM yetersiz kaldığında diske spill yapar, bucket sayısı manuel ayarlanabilir.

Sorting / merge JOIN: önceden sıralı veri için merge tabanlı çalışır, büyük sağ tablolarda RAM tasarrufu sağlar.

Senaryo	PostgreSQL + Citus	ClickHouse 26.x
Yıldız şema (büyük fact + küçük dim tabloları)	Hızlı — reference tablo ile lokal JOIN	Çok hızlı — direct JOIN (sözlük, O(1))
Aynı kolonla bölünmüş iki büyük tablo	Çok hızlı — lokal hash, ağ trafiği yok, doğrusal ölçek	Hızlı — parallel_hash, RAM'e bağlı
Farklı kolonla bölünmüş iki büyük tablo	Repartition devreye girer; veri taşındığı için orta hız	GLOBAL JOIN — sağ tablo tüm sunuculara kopyalanır, ağırdır ve OOM riski taşır (grace_hash önerilir)
3+ tablolu karmaşık rapor	Olgun planlayıcı, JOIN reorder ve paralel yürütme ile öne geçer	Planlayıcı iyileşti; karmaşık senaryolarda manuel sıralama gerekebilir
Tek satır arayan (point-lookup) JOIN	B-tree indeks ile milisaniyede sonuç — PostgreSQL'in doğal alanı	Tasarımı için uygun değil (OLAP odaklı motor)
ANTI / SEMI JOIN (NOT EXISTS, IN, NOT IN)	Tam destek — sorgular doğal sözdizimle yazılır	26.3'te ANTI/SEMI/FULL JOIN'lerde tabloları optimizer otomatik takas eder

Kolon sıkıştırma - codec çeşitliliği ve disk maliyetindeki belirleyici fark

ClickHouse: 2 genel + 6 özel amaçlı codec · Citus Columnar: 4 codec · PostgreSQL TOAST: 2 codec

10–30×

ClickHouse'un kolonsal sıkıştırma oranı — varsayılan LZ4, daha yüksek oran için ZSTD; düşük çeşitlilikli kolonlarda 30× ve üstü ölçülür.

3–10×

Citus Columnar (PostgreSQL 12+ Table Access Method) — varsayılan ZSTD codec; cstore_fdw'nin yenilenmiş halefi.

1.5–3×

PostgreSQL satır tablosu + TOAST — yalnızca ~2 KB üzeri değerleri sıkıştırır; PG 18'den itibaren varsayılan LZ4.

Kıyaslama Boyutu	PostgreSQL TOAST	Citus Columnar	ClickHouse 26.x
Tipik sıkıştırma oranı	1.5–3× (analitik veride)	3–10× (ZSTD ile)	10–30× (kolon homojenliğine göre)
Genel amaçlı codec'ler	pglz, lz4 (PG 14+)	none, pglz, zstd, lz4	LZ4 (varsayılan), ZSTD
Özel amaçlı codec'ler	—	—	Delta, DoubleDelta, Gorilla, T64, GCD, FPC
Varsayılan codec	lz4 (PG 18+), pglz (öncesi)	zstd	LZ4
Çalışma modeli	Yerinde güncelleme, TOAST yeniden yazılır	Yalnız ekleme, UPDATE/DELETE sınırlı	Yalnız ekleme, arka plan birleşmede yeniden sıkıştırma
Ne zaman devreye girer	Yalnızca ~2 KB üzeri değerlerde otomatik	Tablo bazında seçilir	Kolon bazında SETTINGS ile, MergeTree kuralıyla
ClickBench (100M satır)	~100 GiB	(testte yer almıyor)	~9.3 GiB (~10× tasarruf)
En uygun kullanım	Karma OLTP+OLAP, , JSONB	fact tablolar / arşiv bölümleri	Saf OLAP, sürekli akan log / metrik / event verisi

PostgreSQL · Pratik İpuçları

Operasyonu hafifleten, performansı belirgin artıran küçük ama kuvvetli pratikler

PARTITIONING

ATTACH PARTITION ile partition exchange

Mevcut bir tablo, declarative partitioning'e tek komutla katılır: ALTER TABLE parent ATTACH PARTITION child FOR VALUES … — büyük tabloyu yeniden yazmadan partition'lı modele geçilir.

postgresql.org/docs · ddl-partitioning

BULK YÜKLEME

INSERT yerine COPY

Toplu veri yükleme için COPY, satır bazlı INSERT'e göre büyüklük mertebesinde daha hızlıdır: COPY tablo FROM STDIN (FORMAT csv) — ETL ve toplu içe aktarma adımlarının standardı.

postgresql.org/docs · sql-copy

DEPOLAMA

TOAST — otomatik büyük değer sıkıştırması

PostgreSQL, ~2 KB üzerindeki değerleri (metin, JSONB, byte dizileri) otomatik olarak sıkıştırıp ayrı saklar. Geliştirici tarafında ek bir ayar gerekmez — büyük metinler ve JSONB için doğrudan kazanç.

postgresql.org/docs · storage-toast

İNDEKS

Partial index — küçük, odaklı, hızlı

CREATE INDEX … WHERE durum = 'AKTİF' gibi koşullu indeks; yalnızca alakalı satırları indeksler. Boyut ve bakım maliyeti düşer, sorgu hızı artar.

postgresql.org/docs · indexes-partial

İŞLETİM

CREATE INDEX CONCURRENTLY

Canlı sistemde tabloyu kilitlemeden indeks yaratır. Uzun süren büyük indeksler için zorunlu desen — kullanıcılar etkilenmeden bakım penceresi olmadan çalışır.

postgresql.org/docs · sql-createindex

PARALEL SORGU

Otomatik paralel sorgu (PG 9.6+)

Büyük tablo taramaları, agregasyonlar ve hash join'ler PostgreSQL tarafından otomatik paraleltir. max_parallel_workers_per_gather parametresi ile worker sayısı ayarlanır.

postgresql.org/docs · parallel-query

JSONB + GIN

JSONB üzerinde indeksli arama

JSONB kolonuna GIN indeksi ile @>, ? operatörleri saniyenin altında yanıt verir. Şema esnekliği + relational performans aynı motorda — MongoDB ihtiyacının büyük bölümünü karşılar.

postgresql.org/docs · datatype-json

BAKIM

REINDEX CONCURRENTLY (PG 12+)

Şişmiş indeksleri canlı sistemde, tablo kilidi olmadan yeniden inşa eder. Operasyonel bakım pencerelerini neredeyse sıfıra indirir.

postgresql.org/docs · sql-reindex

Büyük Ölçekli Su/Altyapı Kurumu - Coğrafi Karar Destek Sistemi

Dağınık dosya arşivlerinden tek coğrafi analitik platformuna

Milyonlarca parsel

Geometri ve ilişkili öznitelikleriyle harita üzerinde milisaniye sorgu

On binlerce tesis

Sulama, içme suyu ve hidrolojik gözlem altyapısı

Bölgesel birimler

Rol bazlı erişim — tek harita + dashboard üzerinden

BAĞLAM

Geniş coğrafyaya yayılmış binlerce hat, onbinlerce tesis ve hidrolojik ölçüm ağı. Veri farklı dosya biçimlerinde ve farklı veritabanlarında dağınık; aynı varlık için farklı kurumlarda farklı koordinat ve öznitelik gözleniyor.

ZORLUK

- Dağınık coğrafi veriyi tek ve sorgulanabilir bir 'gerçek' altında toplamak
- Farklı kaynaklardan gelen dosyaların kontrol altında, izlenebilir aktarımı
- Saha ölçümleriyle zenginleşen, harita üzerinde canlı bir analitik katman
- Saha teşkilatına yetkiyle sınırlı harita ve dashboard dağıtımı

POSTGRESQL KOMPOZİSYONU

PostgreSQL ODS	PostGIS DWH
Pentaho · Debezium	Airflow · Git
OpenMetadata · Katalog	Vispeahen Harita+Dashboard

SONUÇ

Tek coğrafi gerçek kaynağı: parsel, hat ve tesis için 'aynı kayıt, aynı koordinat'. Saha kararları dakikalar içinde harita üzerinde görünür oldu; birimler arası veri tutarsızlığı raporlamadan kalktı.

Bankacılık Projesi - Yüksek Hacimli Analitik ve Kesintisizlik

Operasyonel yükten kopmadan analitik hacmi Citus ile ölçeklendirmek

Milyonlarca işlem/gün

Operasyonel sistemlerden analitik katmana gerçek zamanlı akış

Milyarlarca satır

Çoklu sunucuya dağıtılmış analitik tablo — paralel, hızlı sorgu

Belirgin TCO avantajı

Ticari DWH'lere kıyasla lisans + operasyon maliyetinde belirgin azalma (sektör kıyası)

BAĞLAM

Çok bölgeli operasyon, yoğun günlük işlem hacmi, denetim ve uyum raporlamasında gecikme tolerans yok. Ticari DWH lisans maliyetleri bütçeyi yoruyor; 7/24 kullanım ve kesintisizlik zorunlu.

ZORLUK

- Operasyonel sistemlerden analitik veri ambarına gerçek zamanlı besleme
- Yüksek hacimli tablolarda saniyelik risk, uyum ve karlılık sorguları
- Kesintisiz çalışma garantisi — otomatik failover ve 7/24 DBA desteği
- Bölge ve ürün bazında veri izolasyonu, yetki yönetimi, denetim izi

POSTGRESQL KOMPOZİSYONU

- PostgreSQL Operasyon
- Citus DWH
- Debezium (CDC)
- Patroni + BFM (HA)
- Airflow
- Vispeahen

SONUÇ

Analitik sorgular Citus ölçeğinde belirgin biçimde hızlandı. Denetim ve karlılık raporları akşamı beklemeden dakikalar içinde hazır. Olası sunucu sorunlarında sistem otomatik devrediyor, operasyon kesintisiz sürüyor.

IoT ve Üretim Verimliliği - Saha Telemetrisinden Yönetim Dashboard'una

Sensörden yönetim dashboard'una saniyelik karar zinciri · benzerlik temelli anomali tespiti

On binlerce sensör

Basınç, sıcaklık, akış, titreşim, elektrik tüketimi

Saniyede yüksek hacim

Saha ekipmanlarından gelen yoğun telemetri akışı

Yüksek kompresyon

TimescaleDB resmi dokümantasyonundaki tipik disk tasarrufu (yaklaşık %90+)

BAĞLAM

Üretim sahalarında çok sayıda sensör yoğun bir telemetri akışı üretiyor. Hem geçmişe dönük çok yıllık analitik, hem anlık operasyonel görünürlük, hem de kestirimci bakım ihtiyacı var. Anomali ve benzerlik tespiti için pgvector (vektör benzerliği) ile pg_trgm (metin/log benzerliği) aynı motorda birlikte kullanılıyor.

ZORLUK

- Saniye çözünürlüklü ve yıllara yayılan ham veriyi ekonomik olarak saklamak
- Operatöre saniyelik canlı dashboard, yönetime aylık/yıllık KPI sunmak
- Benzer çalışma örüntülerini bulmak: pgvector ile sensör imzaları, pg_trgm ile bakım log/alarm metinleri
- Saha sistemleri ile kurumsal BT arasında kontrollü veri akışı ve tek yetki modeli

POSTGRESQL KOMPOZİSYONU

- PostgreSQL + TimescaleDB
- Kafka → Debezium
- pgvector (sensör benzerliği)
- pg_trgm (log/alarm benzerliği)
- Sürekli Özet Tabloları
- Vispeahen Operasyon Paneli

SONUÇ

Ham veri yüksek sıkıştırmayla ekonomik tutuluyor; tarihsel aralık sorguları belirgin hızlanmayla canlı dashboard'u besliyor. pgvector + pg_trgm benzerlik aramaları aynı motorda — bakım ekibi benzer arıza örüntülerini ve alarm mesajlarını öncesinden yakalıyor.

Kart & Ödeme Operatörü - Sahtecilik ve Müşteri 360

Yazım hatası toleranslı eşleştirme + ilişki ağı analizi + AI benzerliği tek motorda

Milyonlarca kullanıcı

Farklı kanallarda (mobil, web, POS) aynı kişi farklı yazımla

3. derece ağ

Apache AGE ile ilişki ağı sorgusu — saniyeler cinsinden

ms-altı eşleşme

pg_trgm ile yazım hatalı kayıtların milisaniye seviyesinde eşleşmesi

BAĞLAM

Çok sayıda üye işyeri, çok kanallı işlem akışı. Aynı kişi farklı adlarla, yazım hatalarıyla, ortak cihaz/IP'lerle sistemde birden fazla hesap olarak görünebiliyor. Sahtecilik ekibi ilişki ağını saniyeler içinde görmek istiyor.

ZORLUK

- 'Mehmet Yılmaz' ≈ 'Mehmet Yilmaz' ≈ 'M.Yılmaz' — farklı yazımları tek kişiye indirgemek
- Ortak cihaz, IP ve kart paternlerini ilişki ağı olarak sorgulamak
- Gerçek zamanlı risk skoru ve analistin tek ekranda görebildiği bütünleşik görüntü
- Tüm bunları tek veri yığınında — ayrı graf / arama / AI altyapısı kurmadan

POSTGRESQL KOMPOZİSYONU

- PostgreSQL ODS
- pg_trgm (yazım toleransı)
- Apache AGE (ilişki ağı)
- Metin + Vektör Arama
- Airflow Senaryoları
- Vispeahen + Uyarılar

SONUÇ

Sahtecilik şüphesinde 3. derece ilişki ağı saniyelerde görünür oldu. Müşteri 360 tek kaynaktan beslendiği için operasyon ve pazarlama aynı gerçekle konuşuyor. Tek PostgreSQL motoru, üç ayrı veritabanının (graf + arama + AI) yerine geçiyor.

Tek cümlede:

PostgreSQL, modern analitik mimarinin doğal başlangıç ve birleştirici motorudur.

Kilit Çıkarımlar

Ekonomi

Lisans bedeli yok · vendor kilidi yok · ticari DWH'lere kıyasla belirgin TCO avantajı

Ekosistem

Tek motor, çok yetenek — Citus, PostGIS, TimescaleDB, AGE ve pgvector aynı platformda

Olgunluk

4 kez DB-Engines 'Yılın Veritabanı' · otomatik devir ve 7/24 uzmanlık ile kurumsal kalite

Değer

Vispeahen ile son mili kapatan BiSoft çözümü — veriden karara aynı omurgada

Kaynaklar & Referanslar

Tüm sayısal ifadeler aşağıdaki birincil ve kurumsal kaynaklara dayanmaktadır

PostgreSQL · DBMS of the Year

DB-Engines Blog — 2017, 2018, 2020, 2023 ödülleri · 2024'te ikinci (Snowflake)

db-engines.com/en/blog_post

PostgreSQL Lisansı & Resmi Doküman

postgresql.org/docs · OSI onaylı PostgreSQL/BSD lisansı

postgresql.org · opensource.org/license/postgresql

Citus · resmi kaynak

Microsoft / Citus Data benchmark'ları · 20×–300× analitik hızlanma · 5–10× kolon sıkıştırma

citusdata.com · techcommunity.microsoft.com

TimescaleDB · resmi doküman

%90+ kompresyon, 1000×'e varan aralık sorgu hızlanması · multi-node hypertable deprecated notu

docs.timescale.com

PostGIS · olgunluk

postgis.net · OGC SFSQL uyumlu · 2001'den beri aktif kurumsal kullanım

postgis.net

Apache AGE · openCypher + SQL

Apache Software Foundation Top-Level projesi · github.com/apache/age

age.apache.org · github.com/apache/age

pgvector · vektör arama

github.com/pgvector/pgvector · AWS / Azure / GCP yönetilen PostgreSQL'lerde hazır

github.com/pgvector/pgvector

pg_duckdb · Lakehouse köprüsü

DuckDB Labs + MotherDuck ortak projesi · S3 / GCS / Azure üzerindeki Parquet, CSV, Iceberg dosyalarını PostgreSQL'den SQL ile sorgular

github.com/duckdb/pg_duckdb

parquet_fdw · Parquet foreign data wrapper

Adjust GmbH tarafından açık kaynak · Lake'teki Parquet dosyalarını PostgreSQL'de yabancı tablo olarak gösterir

github.com/adjust/parquet_fdw

Dicle Elektrik DWH · BiSoft saha referansı

Pentaho ETL + Spark + Hive + PostgreSQL Citus cluster · Coordinator + Worker mimarisi

BiSoft Bilgi Teknolojileri proje dokümantasyonu

Oracle vs PostgreSQL TCO

Bytebase, ScaleGrid, EnterpriseDB kıyas analizleri · 3 yılda belirgin TCO farkı

bytebase.com · scalegrid.io · enterprisedb.com

PostgreSQL Hap Bilgiler

Resmi dokümantasyondan: ddl-partitioning, sql-copy, storage-toast, indexes-partial, parallel-query

postgresql.org/docs/current

Kaynaklar - ClickHouse 26.x Karşılaştırma Slaytları

Slayt 7, 8 ve 9'daki tum sayisal ifadeler ve teknik iddialar asagidaki birincil kaynaklara dayanmaktadir

ClickHouse · Karar cercevesi

Genel karsilastirma · OLAP vs OLTP konumlandirmasi · ClickHouse'un tasarim felsefesi

clickhouse.com/comparison/postgresql

ClickBench Benchmark

100M satir gercek dunya analitik verisi · 50+ veritabani karsilastirmasi · acik kaynak benchmark

benchmark.clickhouse.com

ClickHouse 26.x Changelog

Lightweight UPDATE · 26.3 LTS surumu · ANTI/SEMI/FULL JOIN otomatik taraf degisimi

clickhouse.com/docs/whats-new/changelog

ClickHouse JOIN algoritmalari (Hash, Parallel Hash, Grace Hash)

Resmi blog serisi: hash join calismasi, parallel_hash 16 bucket varsayilan, grace_hash disk spill

clickhouse.com/blog/clickhouse-fully-supports-joins-hash-joins-part2

ClickHouse Direct JOIN (Sözlük tabanli)

Dictionary uzerinden O(1) lookup · yalnizca LEFT ANY destekler · star schema icin en hizli

clickhouse.com/blog/clickhouse-fully-supports-joins-direct-join-part4

ClickHouse JOIN algoritma secimi

join_algorithm = direct, parallel_hash, hash · 24.12'den itibaren varsayilan degisti

clickhouse.com/blog/clickhouse-fully-supports-joins-how-to-choose-the-right-algorithm-part5

ClickHouse Sikistirma · Encodings ve Codec'ler

Delta, DoubleDelta, Gorilla, T64, GCD, FPC · 10-30x tipik oran · LZ4 varsayilan

clickhouse.com/resources/engineering/database-compression

Citus Columnar (PG 12+)

Table Access Method API · ZSTD varsayilan codec · 3-10x sikistirma · cstore_fdw halefi

citusdata.com/blog/2021/03/06/citus-10-columnar-compression-for-postgres

Citus dagitik tablolar (Co-located, Reference, Local)

Distribution column · co-located JOIN · reference tablo replikasyonu · repartition mantigi

docs.citusdata.com

PostgreSQL TOAST sikistirma

pglz, lz4 · ~2 KB ustu deger · PG 14+ lz4 destegi · PG 18'den itibaren varsayilan lz4

postgresql.org/docs/current/storage-toast.html

PostgreSQL Mutations (UPDATE/DELETE)

Lightweight delete · ALTER TABLE UPDATE asenkron mutation · row-level vs column rewrite

clickhouse.com/docs/guides/developer/mutations

Citus dagitik PostgreSQL extension'i

Microsoft tarafindan gelistirilir · Azure Cosmos DB for PostgreSQL · acik kaynak AGPLv3

github.com/citusdata/citus