



CQRS AND EVENT SOURCING



Metehan Gültekin

Java Developer at
@DefineX

Co-organizer at @Folksdev

Metehan Gültekin graduated in Computer Engineering and has been working as a Java Developer for 3 years. He is currently working at Definex. He enjoys learning new things and sharing his knowledge with the community. Driven by this passion, he shares his experiences through presentation videos and technical presentations on YouTube.

 @metehan_gltkn

 @mgmetehan

 @mgmetehan



C Q R S

Command
(Komut)

Query
(Sorgu)

Responsibility
(Sorumluluk)

Segregation
(Ayrıştırma)



CQRS == CQS

??



- ✘ Bir fonksiyon ya veriyi değiştirmeli (command) ya da veriyi döndürmeli (query), ancak her ikisini aynı anda yapmamalıdır.
- ✘ 1988 yılında ortaya atılmıştır.



CQRS vs CRUD



read

Query == /GET

write

Command == /PUT

/POST

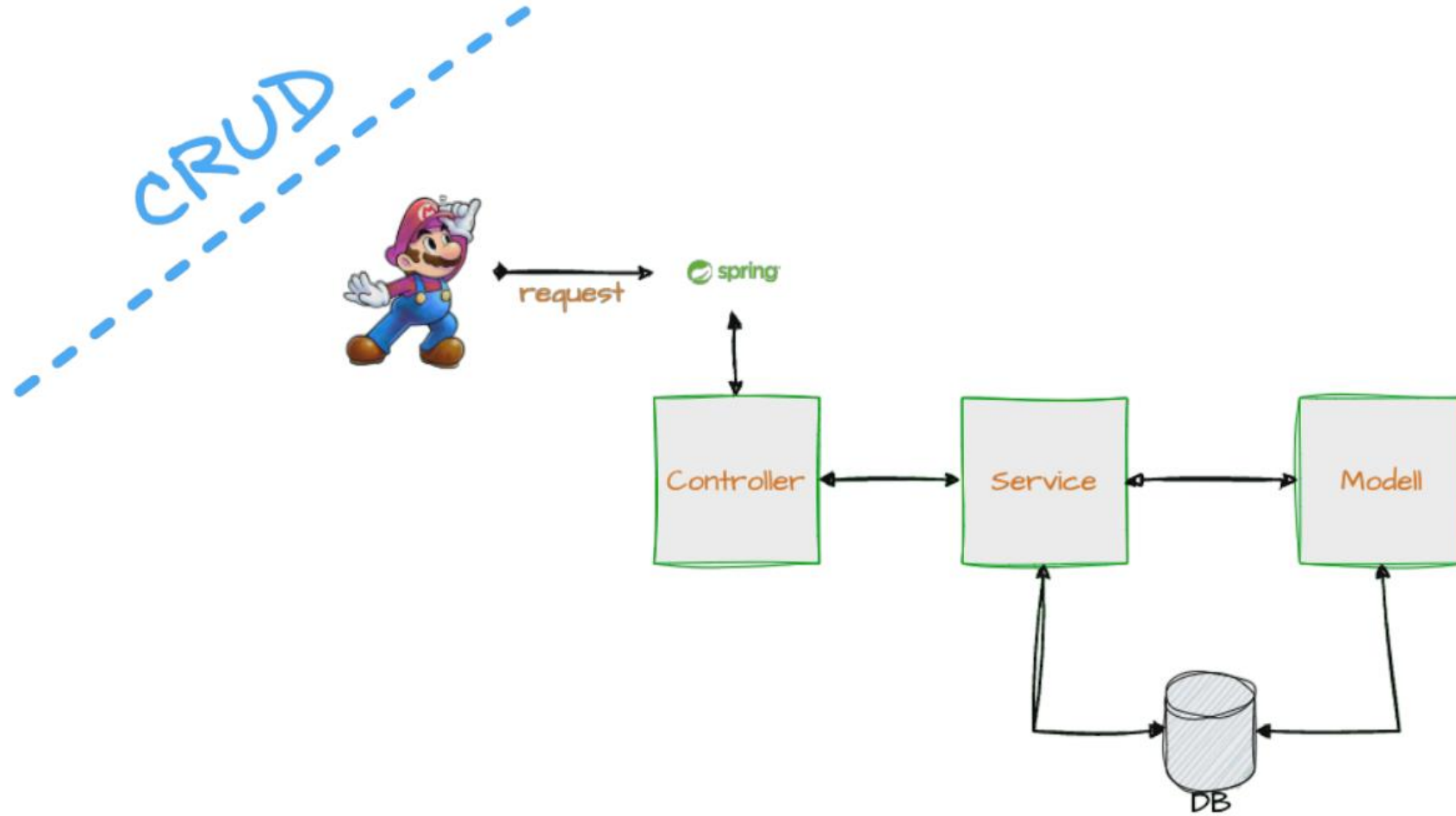
/DELETE

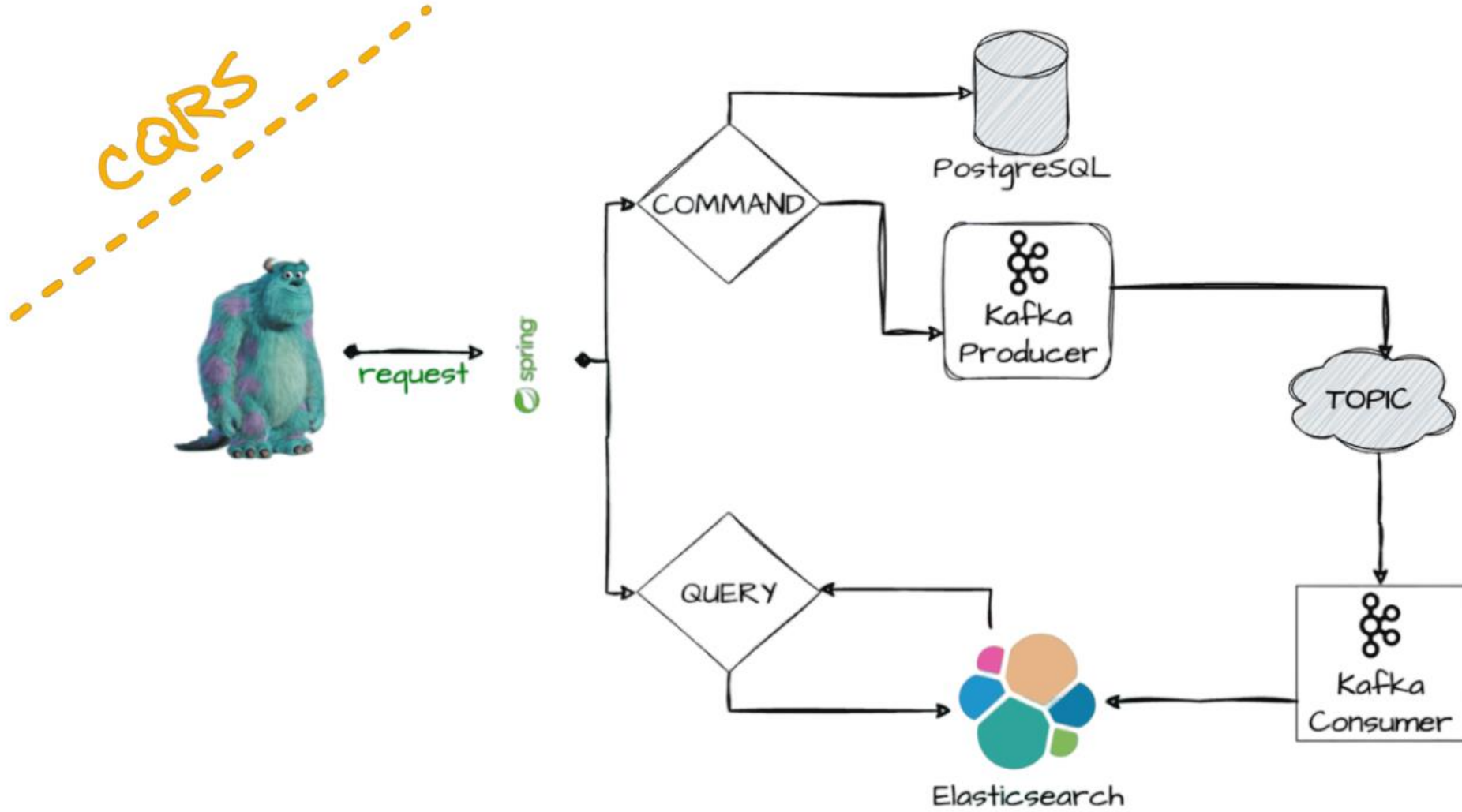
Create == /POST

Read == /GET

Update == /PUT

Delete == /DELETE





CQRS'nin avantajları:



✖ Çeşitli Çözümler Sunma Esnekliği

✖ Bakım Kolaylığı

✖ Hata ve Kesinti İzolasyonu

✖ Geliştirme Hızı

✖ Daha İyi Güvenlik Kontrolü

✖ Bekleme Süresi Olmadan İşlemlerin Yürütülmesi

✖ Read Model Optimizasyonu

✖ Bağımsız Ölçekleme

✖ Farklı Domain Logic Üzerinde Çalışma

CQRS'nin dezavantajları

✘ Karmaşıklık ve Öğrenme Eğrisi

✘ Eğitim ve Adaptasyon Zorlukları

✘ Artan Geliştirme Zamanı

✘ İş Süreçleri Arasındaki Kopukluk

✘ Veri Senkronizasyonu Zorlukları

✘ Artan İletişim İhtiyacı

✘ İki Veritabanı Kullanımı & Maliyet Artışı



CQRS'i

Ne Zaman Kullanmalıyım
ve Kullanmamalıyım?







CQRS Ne Zaman Kullanılmalı ?



✖ Karmaşık İş Mantığı

✖ Bağımsız Geliştirme ve Dağıtım

✖ Farklı İhtiyaçlara Sahip Modeller

✖ Hata ve Kesinti İzolasyonu || Uygulama Güvenilirliği

✖ Yüksek Trafikli Sistemler

✖ Event Sourcing Kullanımı

✖ Farklı Veritabanı ve Teknoloji Kullanımı

CQRS'i Ne Zaman Kullanmamalıyız?

✘ Proje Karmaşıklığı ve Ölçeği Küçükse

✘ Hızlı Geliştirme ve Prototip İhtiyacı

✘ Ekip Deneyimi ve Eğitimi Yetersizse

✘ Veri Tutarlılığı Mutlak Öncelik Değilse

✘ Ekstra Karmaşıklık İstenmiyorsa

✘ Event Sourcing Gereksinimi Yoksa

✘ Maliyet ve Kaynak Sınırlamaları

✘ Performans Kritik Değilse

CQRS'in Kullanılabileceği Senaryolar

 Netflix

 E-Ticaret Platformları

 Uber

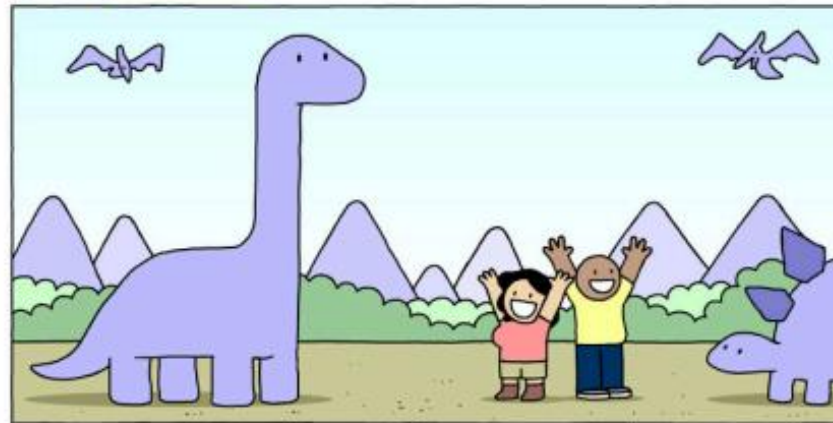
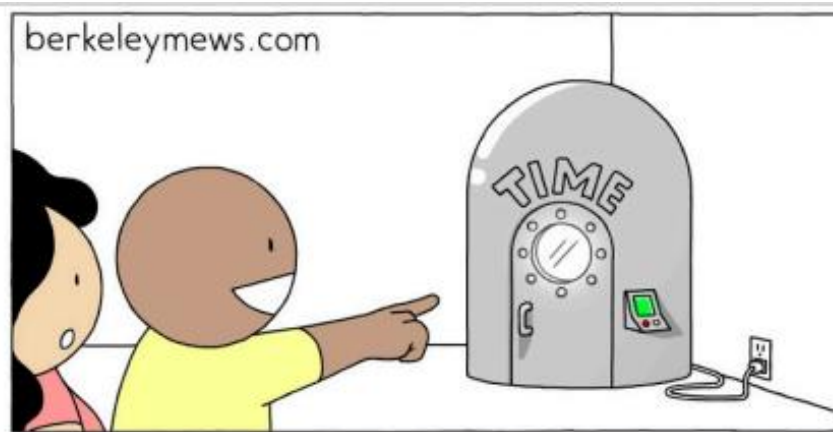
 Twitter



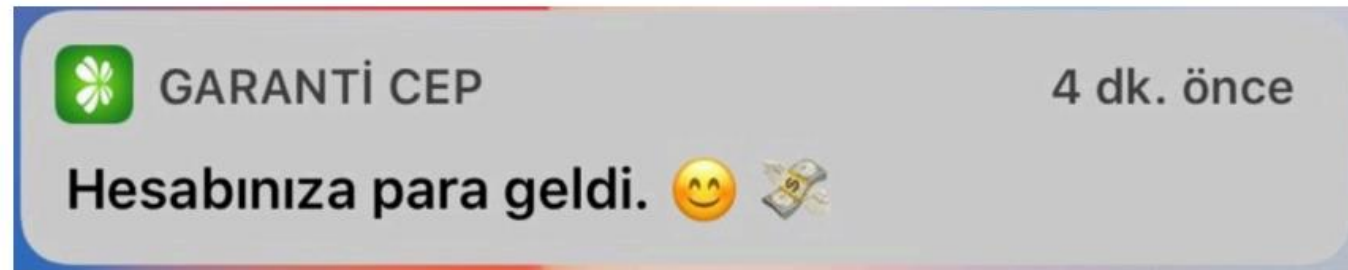
Event Sourcing

(Olay Tabanlı Mimari)



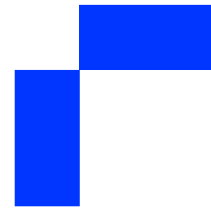


Event?



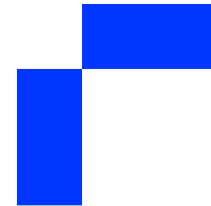


State?





Aggregate?





t1



Total Item
Stock
7

Items Created
8

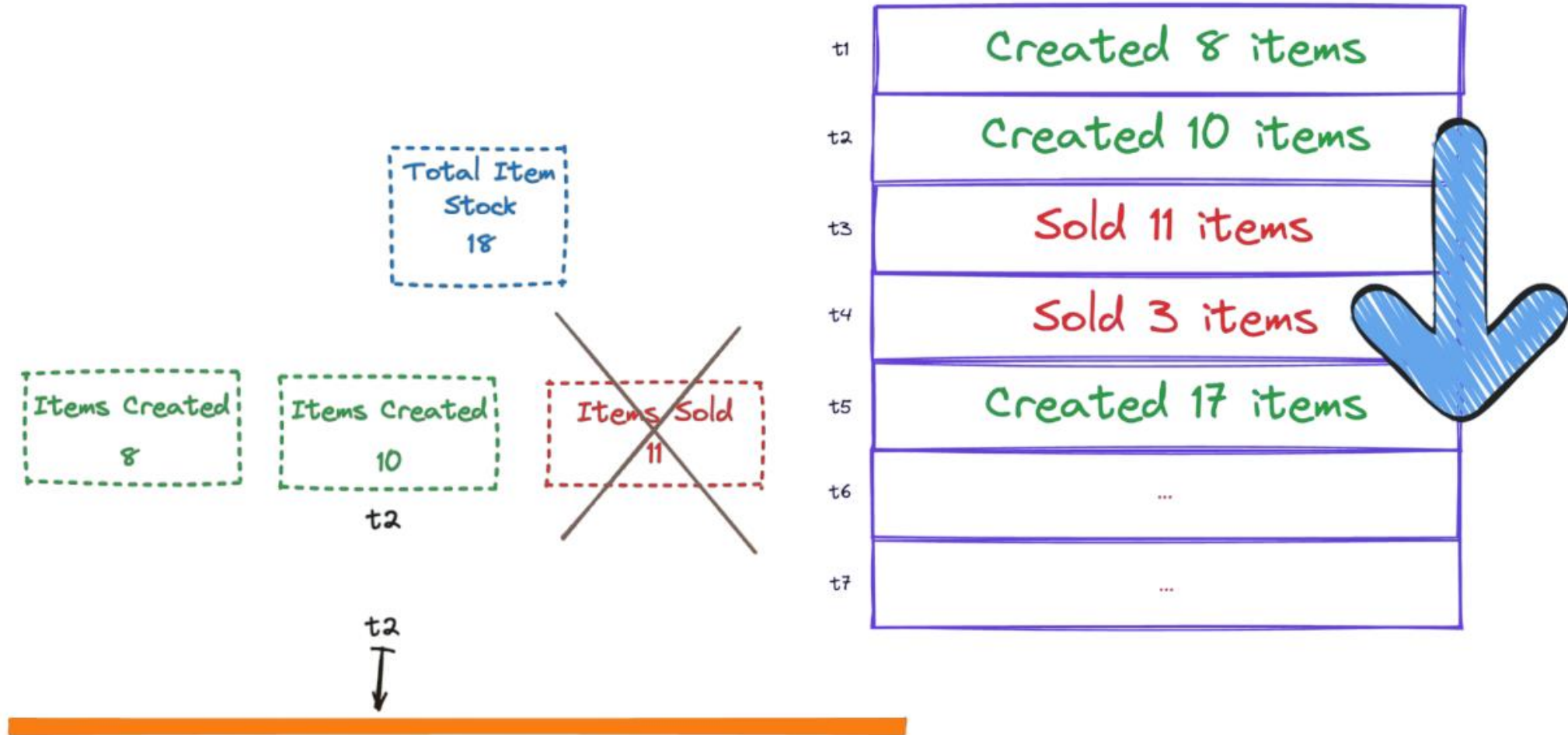
Items Created
10

Items Sold
11

t3

t3
↓





Event Sourcing == Event-Driven Architecture ??

- ✘ Event Sourcing, uygulamanın durumunu her durum değişikliğini temsil eden event aracılığıyla kaydeden bir mimari desen
- ✘ Event-Driven Architecture ise bileşenler arasında event'lerin yayınlanması ve dinlenmesine dayanan bir iletişim prensibidir

Entities vs Events

id	order_name	order_state	create_date	update_date
1	Mac M2	Created	t1	-

Order Created

t1

event_id	event_type	entity_type	entity_id	event_data	event_date
80	Order Created	Order	101	{...}	t1

id	order_name	order_state	create_date	update_date
1	Mac M2	Approved	t1	t2

Order Approved

t2

event_id	event_type	entity_type	entity_id	event_data	event_date
80	Order Created	Order	101	{...}	t1
81	Order Approved	Order	101	{...}	t2

Order Shipped

t3

id	order_name	order_state	create_date	update_date
1	Mac M2	Shipped	t1	t3

event_id	event_type	entity_type	entity_id	event_data	event_date
80	Order Created	Order	101	{...}	t1
81	Order Approved	Order	101	{...}	t2
82	Order Shipped	Order	101	{...}	t3
...
...

Event Sourcing'in avantajları



✕ Geçmiş İzleme ve Denetleme

✕ Durum Geri Yükleme Yeteneği

✕ Hata Ayıklama Kolaylığı

✕ İş Zekası ve Raporlama

✕ Paralel İşleme Yeteneği

✕ İleriye Dönük Uyumluluk

Event Sourcing'in dezavantajları

✖ Karmaşıklık

✖ Veritabanı Boyutu

✖ Öğrenme Eğrisi

✖ Analitik Zorluklar

✖ Performans

✖ Sistem Yeniden İnşası

Event Sourcing'in Kullanılabileceği Senaryolar



✖ Bankacılık ve Finansal İşlemler

✖ E-Ticaret Sistemleri

✖ Lojistik ve Tedarik Zinciri Yönetimi



That's all folks!

Define 

That's all folks!