

DML İŞLEMLERİNİ OTOMATİK KAYIT ALTINA ALMA

Öğretim Görevlisi A. Berika VAROL MALKOÇOĞLU

İçindekiler

- Log nedir?
- Neden tutulur?
- Trigger ile İşlem Loglarını Tutma
- Change Data Capture nedir?
- Debezium ile Change Data Capture
- Apache Kafka Nedir?

DML İŞLEMLERİNİ OTOMATİK KAYIT ALTINA ALMA (TRIGGER)

Log Nedir?

- Uygulama bilgilerinin, sistem performansının veya kullanıcı etkinliklerinin ayrıntılı listesidir.
- Kuruluştaki sistem ve ağ performansını optimize etme, kullanıcıların eylemlerini kaydetme, kötü amaçlı etkinlikleri araştırma ve yararlı veriler sağlama gibi birçok işlevi yerine getiriyor.

Log

- Yedekleme sırasında olan olaylardan, uygulamanın çalışmasını durduran hatalardan veya kullanıcılar tarafından web sitesinden talep edilen dosyalara kadar her türlü olayları belgeleyebilmekte olan loglar,
 - denetim logları,
 - **işlem logları,**
 - mesaj logları,
 - olay logları gibi birçok farklı dosyadır.

Log Söz dizimi

- Log mesajlarının nasıl oluşturulduğu, taşındığı, saklandığı, incelendiği ve analiz edildiği bu söz dizimi ile tanımlanır.
- Genellikle kullanılan alanlar;
 - Tarih/zaman
 - Log girdisi türü
 - Üreten sistem
 - Üretildiği uygulama veya bileşen
 - Kullanıcı etkinlikleriyle ilgili loglar için kullanıcı adı

Neden Log Tutulur

- Kritik verilerin bulunduğu veritabanı uygulamalarında, veriler üzerindeki değişiklikleri veri güvenliği amacıyla kaydetmemiz gerekir.
- Bunun yapılması için;
 - Tetikleyiciler(triggers)
 - Change Data Capture (CDC) mimarisi kullanılır.








Trigger ile İşlem Loglarını Tutma

- Tetikleyiciler, belirli olaylardan önce veya sonra tetiklenen kancalardır
 - Örneğin; eklemekten sonra veya güncellemeden önce vs.
- Tetikleyicilerin en büyük avantajı, SQL seviyesinde yönetilebilmeleridir.
- Dezavantajı ise tetikleyicilerin veritabanı performansı üzerinde olumsuz bir etkiye sahip olmasıdır.
- Çünkü her tablo için ayrı ayrı tanımlanması gerekir.

Trigger ile İşlem Loglarını Tutma

```
1 • use beykoz;  
2  
3 • insert into bolumbilgisi values (1, 'bilgisayar mühendisliği');  
4 • insert into bolumbilgisi values (2, 'yazılım mühendisliği');  
5  
6 • select * from bolumbilgisi ;  
7
```

<

Result Grid |   Filter Rows: | Edit:    | Export/Import:  

	id	bolum_ad
▶	1	bilgisayar mühendisliği
	2	yazılım mühendisliği
*	NULL	NULL

Trigger ile İşlem Loglarını Tutma

- `create table akademisyen_log(
id INT auto_increment,
Islem_Tarihi datetime,
Islem char(10),
ad Varchar(50),
soyad varchar(50),
bolum int,
primary key (id));`

- `create table bolumbilgisi_log(
id INT auto_increment,
Islem_Tarihi datetime,
Islem char(10),
bolum_ad Varchar(50),
primary key (id));`

Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER akademisyen_log_insert AFTER INSERT  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(Islem_Tarihi,Islem,ad,soyad,bolum) values (CURRENT_TIMESTAMP(), 'Insert', new.ad, new.soyad, new.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
43 • insert into akademisyen values (5,'utku','kan',1);
44
45 • select * from akademisyen;
46 • select * from akademisyen_log;
47
```






Result Grid						
Filter Rows: <input type="text"/>						
	id	Islem_Tarihi	Islem	ad	soyad	bolum
▶	1	2020-12-20 16:43:41	Insert	utku	kan	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER akademisyen_log_update AFTER UPDATE  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(Islem_Tarihi,Islem,ad,soyad,bolum) values (CURRENT_TIMESTAMP(), 'Update', new.ad, new.soyad, new.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
60
61 • update akademisyen set ad='fatma' where id=4;
62
63
64 • select * from akademisyen;
65 • select * from akademisyen_log;
66
```

Result Grid |   Filter Rows: | Edit:   

	id	Islem_Tarihi	Islem	ad	soyad	bolum
▶	1	2020-12-20 16:43:41	Insert	utku	kan	1
	2	2020-12-20 16:51:11	Update	fatma	can	1
•	NULL	NULL	NULL	NULL	NULL	NULL






Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER akademisyen_log_delete AFTER DELETE  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(Islem_Tarihi,Islem,ad,soyad,bolum) values (CURRENT_TIMESTAMP(), 'Delete', old.ad, old.soyad, old.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
63 • delete from akademisyen where id=3;
64
65 • select * from akademisyen;
66 • select * from akademisyen_log;
67
```

<

Result Grid |   Filter Rows: | Edit:   

	id	Islem_Tarihi	Islem	ad	soyad	bolum
▶	1	2020-12-20 16:43:41	Insert	utku	kan	1
	2	2020-12-20 16:51:11	Update	fatma	can	1
	3	2020-12-20 16:51:47	Delete	utku	can	1
•	NULL	NULL	NULL	NULL	NULL	NULL

Trigger ile İşlem Loglarını Tutma

```
70 • ALTER TABLE akademisyen_log add column sorgu_sahibi varchar(100) after id;  
71  
72  
73 • select * from akademisyen_log;  
74
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Co

	id	sorgu_sahibi	Islem_Tarihi	Islem	ad	soyad	bolum
▶	1	NULL	2020-12-20 16:43:41	Insert	utku	kan	1
	2	NULL	2020-12-20 16:51:11	U Insert	fatma	can	1
	3	NULL	2020-12-20 16:51:47	Delete	utku	can	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER akademisyen_log_insert AFTER INSERT  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(sorgu_sahibi,Islem_Tarihi,Islem,ad,soyad,bolum)  
    values (CURRENT_USER() CURRENT_TIMESTAMP(), 'Insert', new.ad, new.soyad, new.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
|  
DELIMITER //  
CREATE TRIGGER akademisyen_log_update AFTER UPDATE  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(sorgu_sahibi,Islem_Tarihi,Islem,ad,soyad,bolum)  
    values (CURRENT_USER(),CURRENT_TIMESTAMP(), 'Update', new.ad, new.soyad, new.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER akademisyen_log_delete AFTER DELETE  
ON akademisyen  
FOR EACH ROW  
BEGIN  
    INSERT INTO akademisyen_log(sorgu_sahibi,Islem_Tarihi,Islem,ad,soyad,bolum)  
    values (CURRENT_USER(),CURRENT_TIMESTAMP(), 'Delete', old.ad, old.soyad, old.bolum);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

```
111 • insert into akademisyen values (6,'aylin','sarı',2);
112
113 • update akademisyen set ad='cansu' where id=5;
114
115 • delete from akademisyen where id=4;
116
117 • select * from akademisyen_log;
```

	id	sorgu_sahibi	Islem_Tarihi	Islem	ad	soyad	bolum
▶	1	NULL	2020-12-20 16:43:41	Insert	utku	kan	1
	2	NULL	2020-12-20 16:51:11	Update	fatma	can	1
	3	NULL	2020-12-20 16:51:47	Delete	utku	can	1
	4	root@localhost	2020-12-20 17:45:26	Insert	aylin	sarı	2
	5	root@localhost	2020-12-20 17:45:53	Update	cansu	kan	1
	6	root@localhost	2020-12-20 17:45:54	Delete	fatma	can	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
110
119 • select * from akademisyen;
120
```

	id	ad	soyad	bolum
▶	1	ali	can	1
	2	ayşe	öz	2
	5	cansu	kan	1
	6	aylin	sarı	2
*	NULL	NULL	NULL	NULL

Kullanılabilecek Fonskiyonlar

22 • `select now();`



<

Result Grid |   Filter Rows:

	now()
▶	2020-12-20 19:13:32

24 • `select current_time();`



25
<

Result Grid |   Filter Rows:

	current_time()
▶	19:16:00

23 • `select current_date();`

24
<

Result Grid |   Filter Rows:


	current_date()
▶	2020-12-20

Kullanılabilecek Fonskiyonlar

```
25 • select session_user();
```

26

<


Result Grid |  Filter Rows:

	session_user()
▶	root@localhost

```
26 • select system_user();
```

27

<


Result Grid |  Filter Rows:

	system_user()
▶	root@localhost

```
27 • select user();
```

28

<

Result Grid |  Filter Rows:

	user()
▶	root@localhost

Kullanılabilecek Fonskiyonlar

```
28 • select version();
```

```
29
```

Result Grid		Filter Rows:
	version()	
▶	8.0.21	

```
129 • select database();
```

Result Grid		Filter Rows:
	database()	
▶	beykoz	




Trigger ile İşlem Loglarını Tutma



```
127 • create table bolumbilgisi_log(  
128     id INT auto_increment,  
129     Islem_Tarihi datetime,  
130     Islem char(10),  
131     bolum_ad Varchar(50),  
132     primary key (id));  
133  
134 • alter table bolumbilgisi_log add column sorgu_sahibi varchar(100) after id;  
135 • alter table bolumbilgisi_log add column versiyon varchar(100) after sorgu_sahibi;  
136 • alter table bolumbilgisi_log add column veritabanı varchar(100) after versiyon;  
137 • select * from bolumbilgisi_log;  
138
```


<

Result Grid

Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

	id	sorgu_sahibi	versiyon	veritabanı	Islem_Tarihi	Islem	bolum_ad
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL








Trigger ile İşlem Loglarını Tutma

```
DELIMITER //  
CREATE TRIGGER bolumbilgisi_log_insert AFTER INSERT  
ON bolumbilgisi  
FOR EACH ROW  
BEGIN  
    INSERT INTO bolumbilgisi_log(sorgu_sahibi,versiyon,veritabanı,Islem_Tarihi,Islem,bolum_ad)  
    values (CURRENT_USER(),version(),database(),CURRENT_TIMESTAMP(), 'Insert', new.bolum_ad);  
END//  
DELIMITER ;
```

Trigger ile İşlem Loglarını Tutma

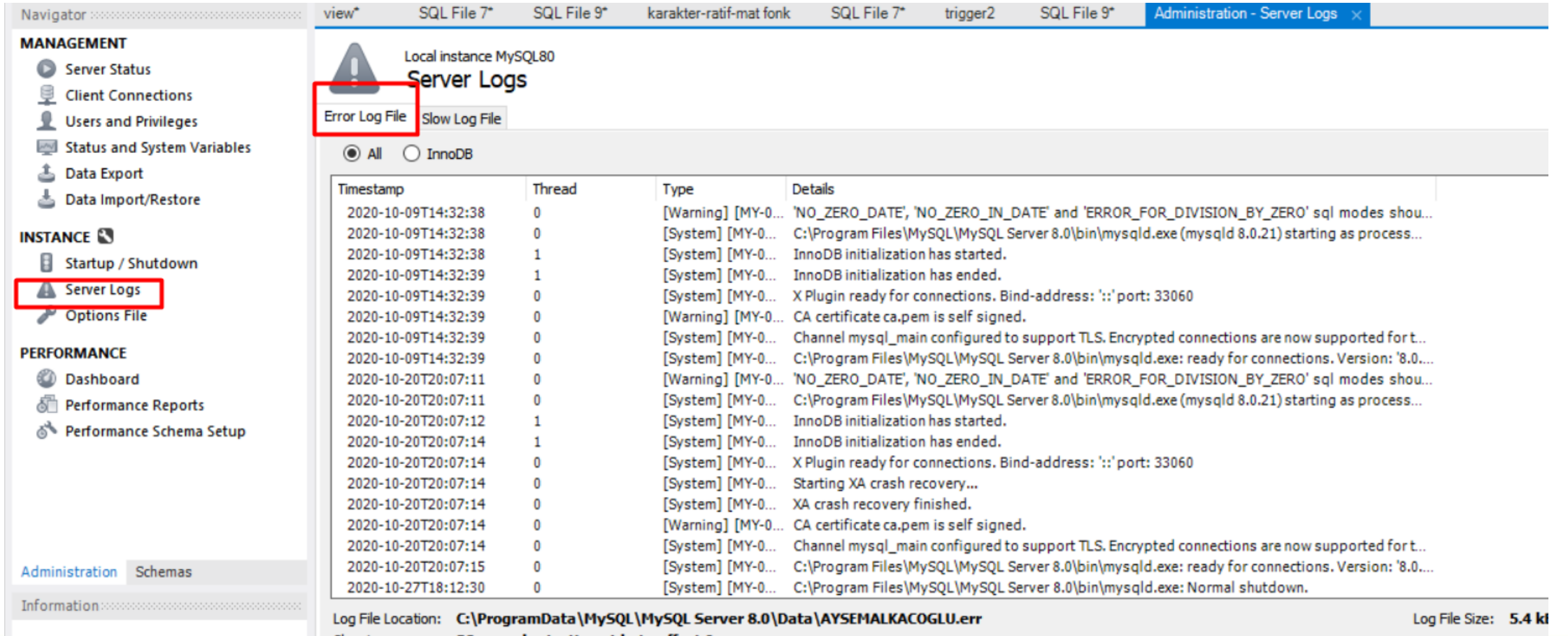
```
140
149 • insert into bolumbilgisi values (5, 'ekonomi');
150
151 • select * from bolumbilgisi_log;
```

<

Result Grid |   Filter Rows: | Edit:    | Export/Import:  

	id	sorgu_sahibi	versiyon	veritabani	Islem_Tarihi	Islem	bolum_ad
▶	1	root@localhost	8.0.21	beykoz	2020-12-20 19:33:53	Insert	ekonomi
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Sistem logları



The screenshot shows the MySQL Workbench interface with the 'Server Logs' tab selected in the left-hand 'MANAGEMENT' pane. The main window displays the 'Local instance MySQL80' logs. A red box highlights the 'Error Log File' button. The log table shows various system messages, including warnings and system events. The bottom status bar indicates the log file location and size.

Navigation Pane (Left):

- MANAGEMENT
 - Server Status
 - Client Connections
 - Users and Privileges
 - Status and System Variables
 - Data Export
 - Data Import/Restore
- INSTANCE
 - Startup / Shutdown
 - Server Logs**
 - Options File
- PERFORMANCE
 - Dashboard
 - Performance Reports
 - Performance Schema Setup

Main Window (Right):

Local instance MySQL80
Server Logs
Error Log File | Slow Log File

Radio buttons: ☒ All ☐ InnoDB

Timestamp	Thread	Type	Details
2020-10-09T14:32:38	0	[Warning] [MY-0...	'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ERROR_FOR_DIVISION_BY_ZERO' sql modes shou...
2020-10-09T14:32:38	0	[System] [MY-0...	C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe (mysqld 8.0.21) starting as process...
2020-10-09T14:32:38	1	[System] [MY-0...	InnoDB initialization has started.
2020-10-09T14:32:39	1	[System] [MY-0...	InnoDB initialization has ended.
2020-10-09T14:32:39	0	[System] [MY-0...	X Plugin ready for connections. Bind-address: '::' port: 33060
2020-10-09T14:32:39	0	[Warning] [MY-0...	CA certificate ca.pem is self signed.
2020-10-09T14:32:39	0	[System] [MY-0...	Channel mysql_main configured to support TLS. Encrypted connections are now supported for t...
2020-10-09T14:32:39	0	[System] [MY-0...	C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe: ready for connections. Version: '8.0...
2020-10-20T20:07:11	0	[Warning] [MY-0...	'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ERROR_FOR_DIVISION_BY_ZERO' sql modes shou...
2020-10-20T20:07:11	0	[System] [MY-0...	C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe (mysqld 8.0.21) starting as process...
2020-10-20T20:07:12	1	[System] [MY-0...	InnoDB initialization has started.
2020-10-20T20:07:14	1	[System] [MY-0...	InnoDB initialization has ended.
2020-10-20T20:07:14	0	[System] [MY-0...	X Plugin ready for connections. Bind-address: '::' port: 33060
2020-10-20T20:07:14	0	[System] [MY-0...	Starting XA crash recovery...
2020-10-20T20:07:14	0	[System] [MY-0...	XA crash recovery finished.
2020-10-20T20:07:14	0	[Warning] [MY-0...	CA certificate ca.pem is self signed.
2020-10-20T20:07:14	0	[System] [MY-0...	Channel mysql_main configured to support TLS. Encrypted connections are now supported for t...
2020-10-20T20:07:15	0	[System] [MY-0...	C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe: ready for connections. Version: '8.0...
2020-10-27T18:12:30	0	[System] [MY-0...	C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld.exe: Normal shutdown.

Log File Location: C:\ProgramData\MySQL\MySQL Server 8.0\Data\AYSEMALKACOGU.err
Log File Size: 5.4 kb

DML İŞLEMLERİNİ OTOMATİK KAYIT ALTINA ALMA (CHANGE DATA CAPTURE)

Change Data Capture (CDC)

- Değişim verilerini yakalamak olarak da tanımlanır.
- CDC, bir tabloda yapılan insert, update işlemlerini gerçek zamanlı yakalayan ve veri üzerindeki bu değişiklikleri istenilen hedefe aktarmamızı sağlayan bir design pattern'dir.
- Modern veritabanlarında bu işlem ilgili kaydın zaman, ilk durum ve son durum bilgilerinin log'lanmasıyla sağlanır.
- Triggerlara göre;
 - Daha performanslıdır, hızlıdır,
 - Sisteme daha az yük getirir

CDC Nasıl Çalışır?



CDC Nasıl Çalışır?

- CDC, ekleme, güncelleme ve silme gibi veritabanı işlevleri için değiştirilen kayıtlara ilişkin verileri sağlar.
- Bu değişikliğin kaydını bir veritabanının içinde veya verilere dayanan diğer uygulamalar için kullanılabilir hale getirir.
- CDC süreci, transaction log mekanizması üzerinden değişiklikleri alıp tablo üzerinde kayıt oluşturur.

CDC Nasıl Çalışır?

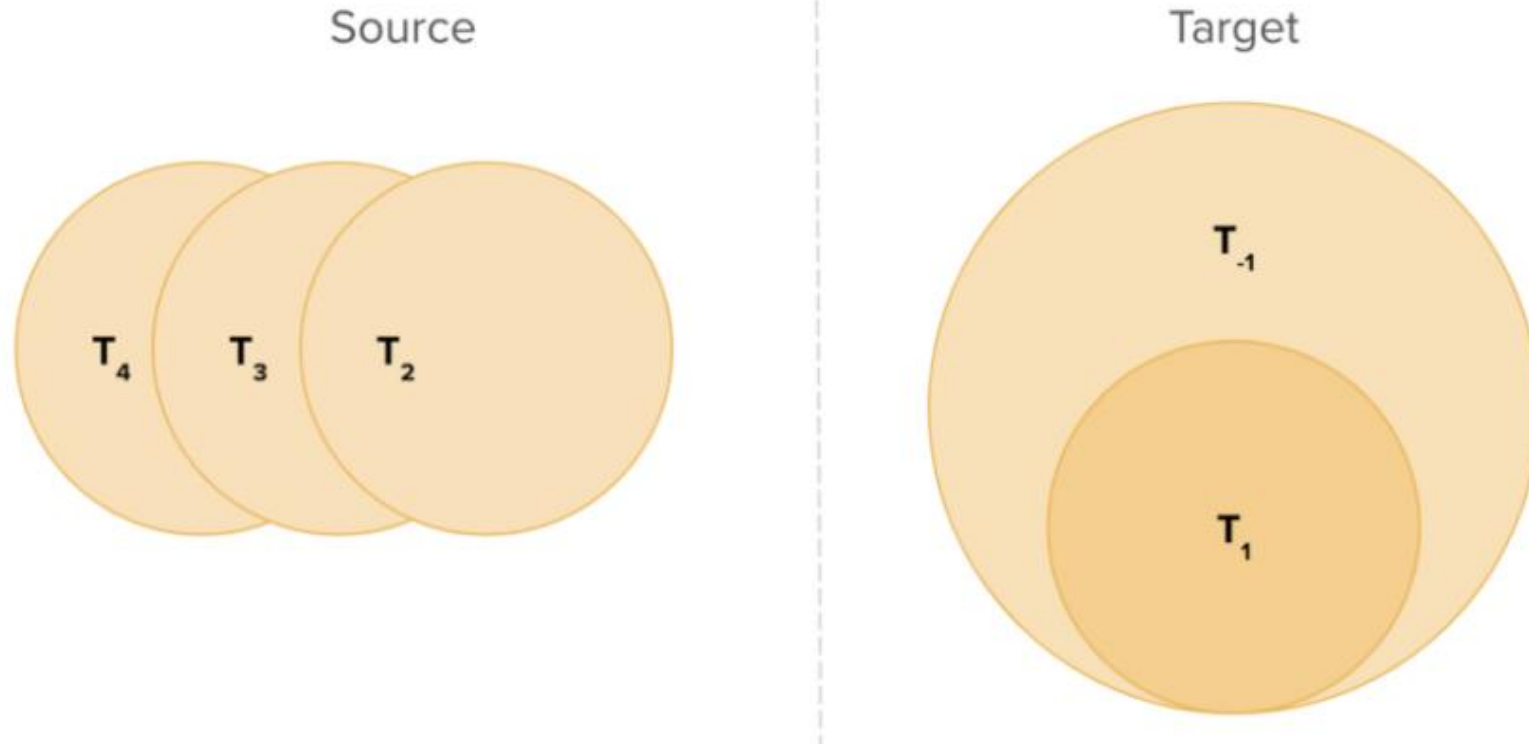
- Bu kayıt, izlenilmesi istenen tablonun aynı sütun yapısıyla ancak tablo üzerindeki değişikliğin ne olduğunu özetlemek için meta verileri içeren ek sütunlar bulundurur.
- CDC araçları tipik olarak, sistem kurtarma amacıyla kayıt değişikliklerini dahili olarak takip eden veritabanının işlem günlüğüne dayanır.

CDC Neden Kullanılır?

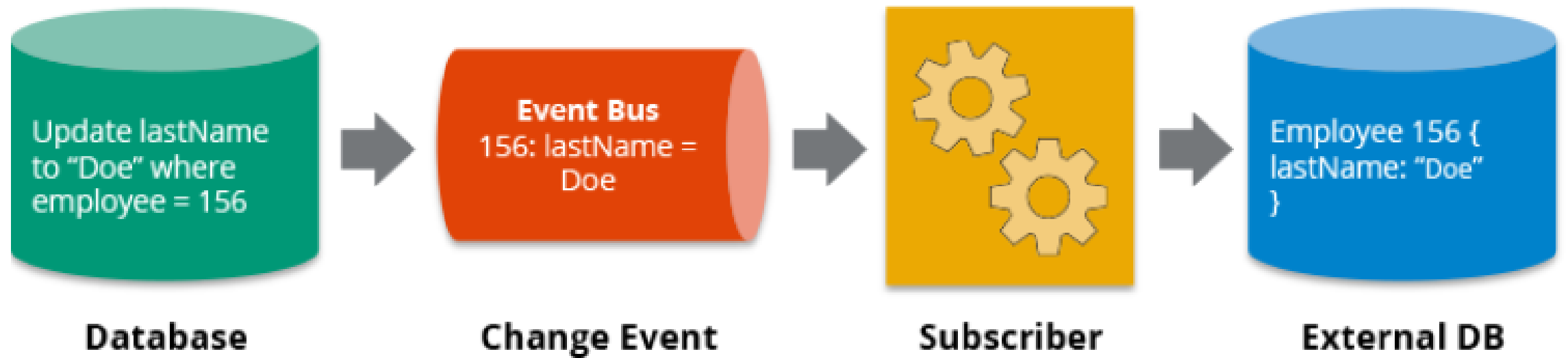
- CDC araçları, veritabanı değişikliklerini harici bir sisteme iletmek için kullanır.
- Sistemleri senkronize tutmak ve zaman içinde değişen veri kaydı geçmişini korumak için kullanır.



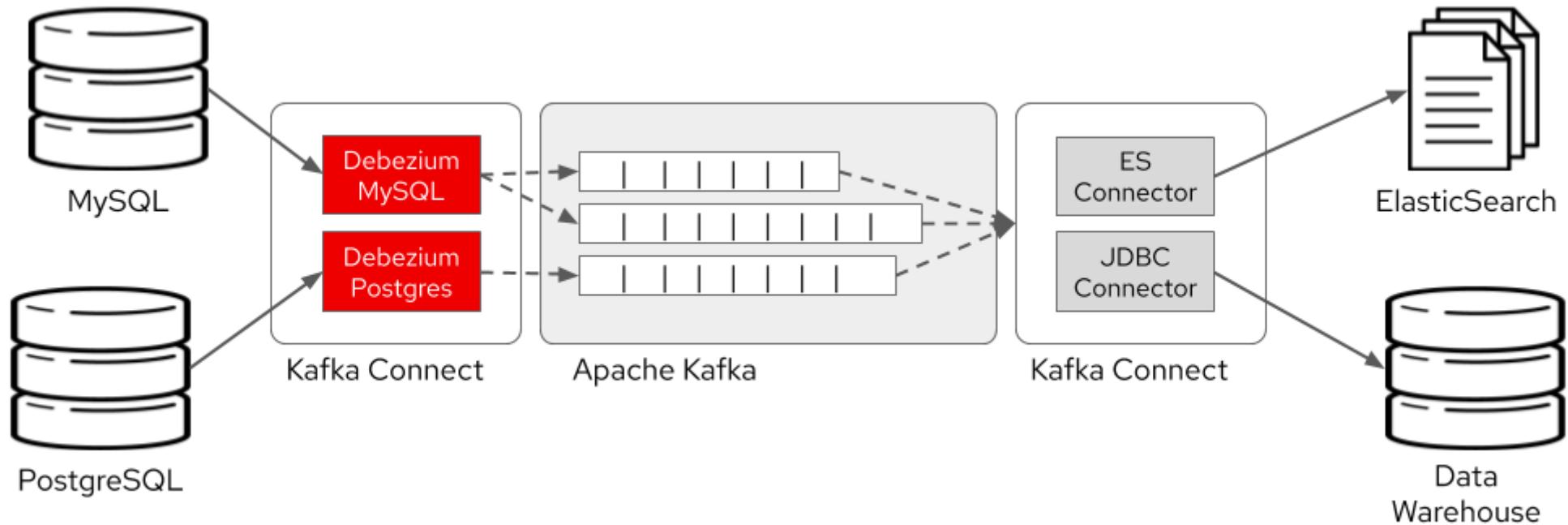
CDC Neden Kullanılır?



CDC

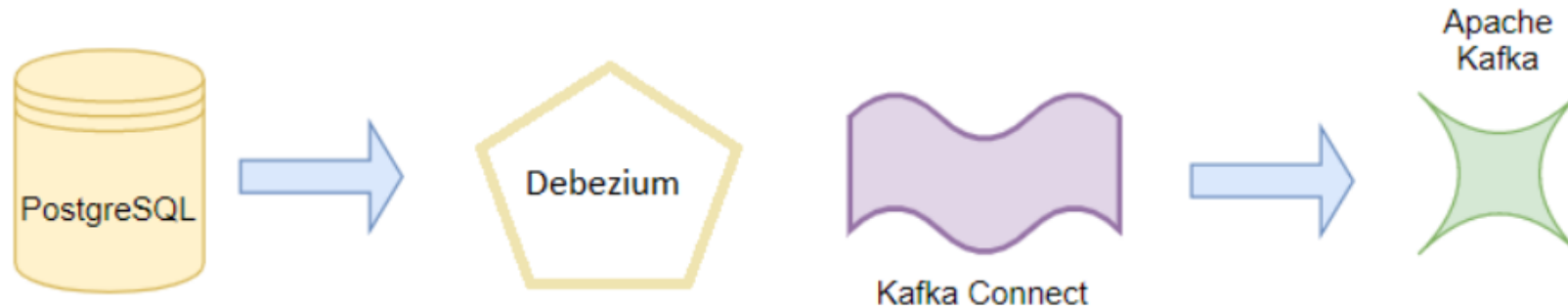


Debezium ile Change Data Capture



Debezium ile Change Data Capture

- Debezium, açık kaynak kodlu bir CDC framework'tur.
- Veritabanından izlenen verilerin Kafka Connect Api kullanılarak Kafka'da tanımlı topic'e aktarılmasını sağlar.



- Bilinen birçok veritabanı (MySQL, MongoDB, PostgreSQL, Oracle, SQL Server ve Cassandra vs.) Debezium ile çalışabilir.

Debezium ile Change Data Capture

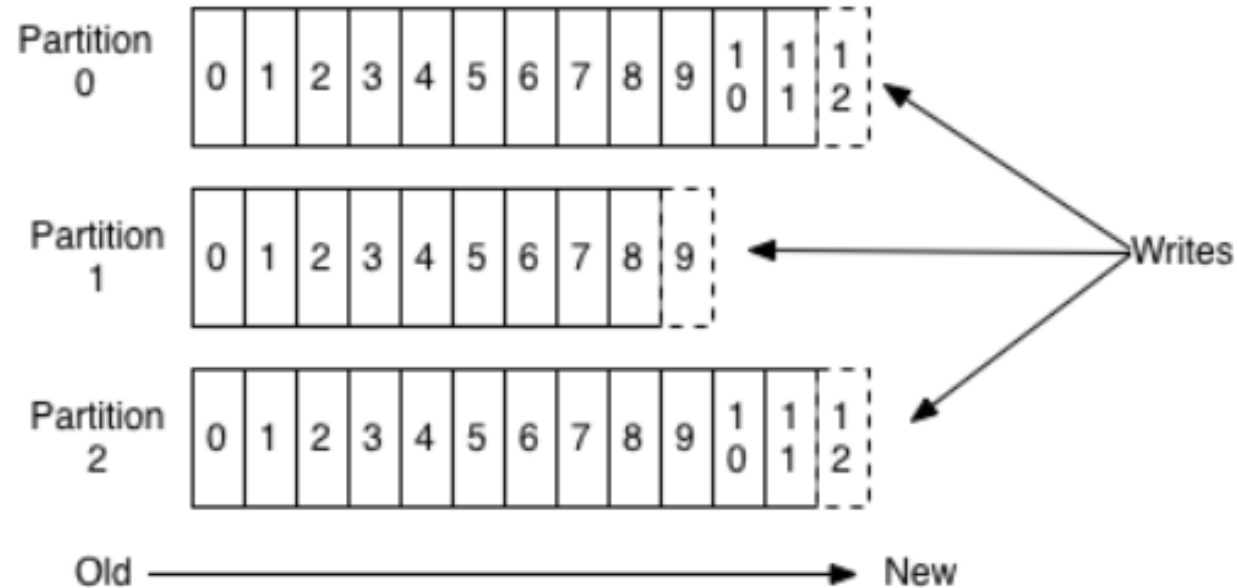
- Örnek:
 - Debezium MySQL bağlayıcısı, doğrudan MySQL ikili günlük dosyasından okur.
 - Debezium bağlayıcısı bir PostgreSQL sunucusuna veya kümesine ilk kez bağlandığında, tüm şemaların tutarlı bir anlık görüntüsünü alır. Bu anlık görüntü tamamlandıktan sonra, bağlayıcı, PostgreSQL veritabanına işlenen satır düzeyinde değişiklikleri sürekli olarak yakalar.
 - Debezium'un MongoDB bağlayıcısı, veritabanları ve koleksiyonlardaki belge değişiklikleri için bir MongoDB replika kümesini veya MongoDB parçalanmış kümeyi izleyerek bu değişiklikleri Kafka topic olarak kaydeder.

Apache Kafka Nedir?

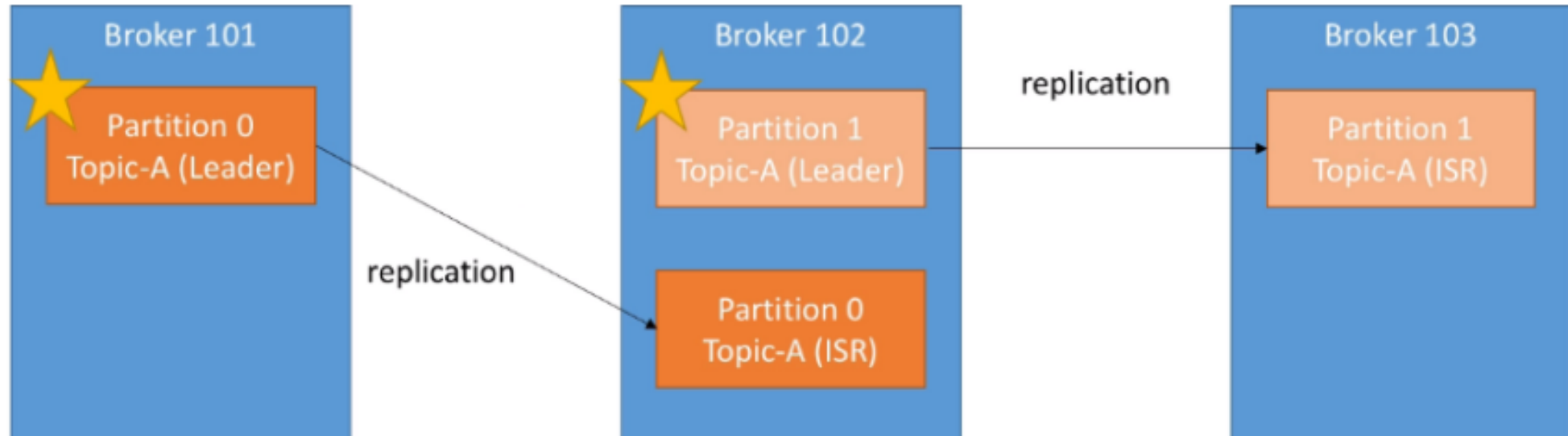
- Apache Kafka, LinkedIn tarafından geliştirilmiş, şu an Apache yönetiminde açık kaynak kodlu dağıtık veri akış platformudur.
- Sistemlerin birbirlerine bağımlılıklarını ortadan kaldırır.
- Hataya dayanıklı, yatay olarak ölçeklenebilen, esnek bir mimariye sahiptir.
- Yüksek performans ile bir sistemden diğer sisteme 10 ms'den az bir gecikme ile **neredeyse gerçek zamanlı (nearly real time)** olarak veri transferini mümkün kılmaktadır.

Apache Kafka (Topics, Partitions, Offsets)

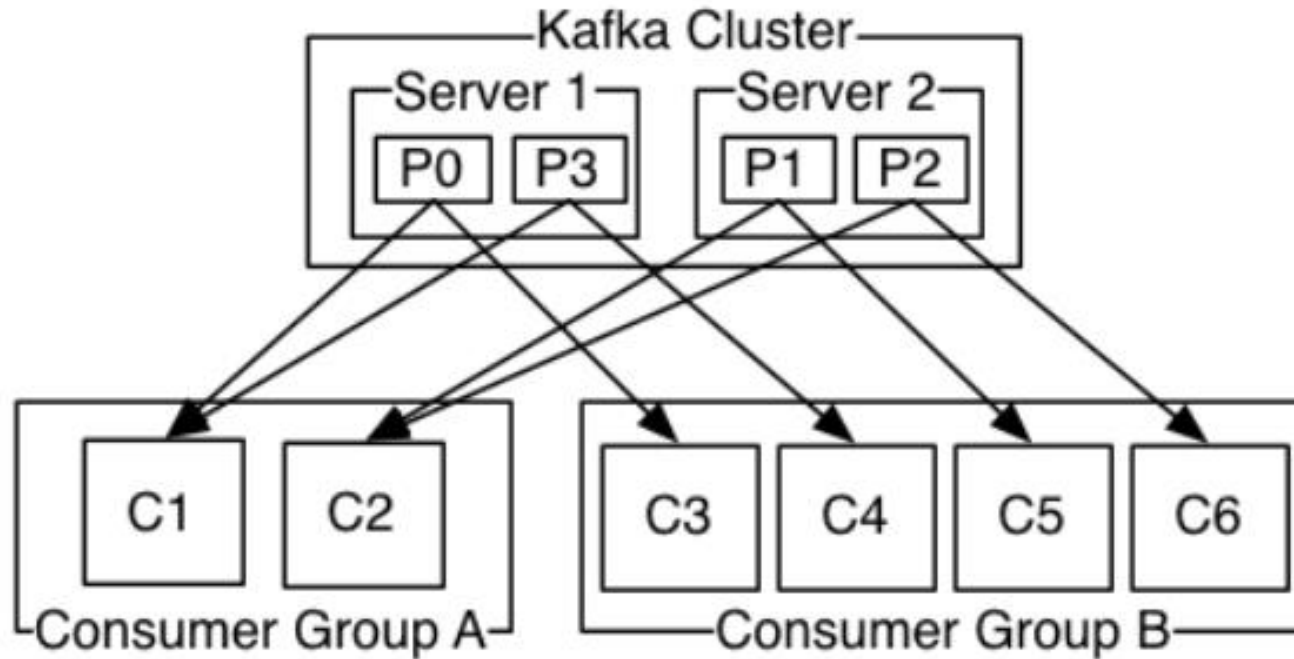
Anatomy of a Topic



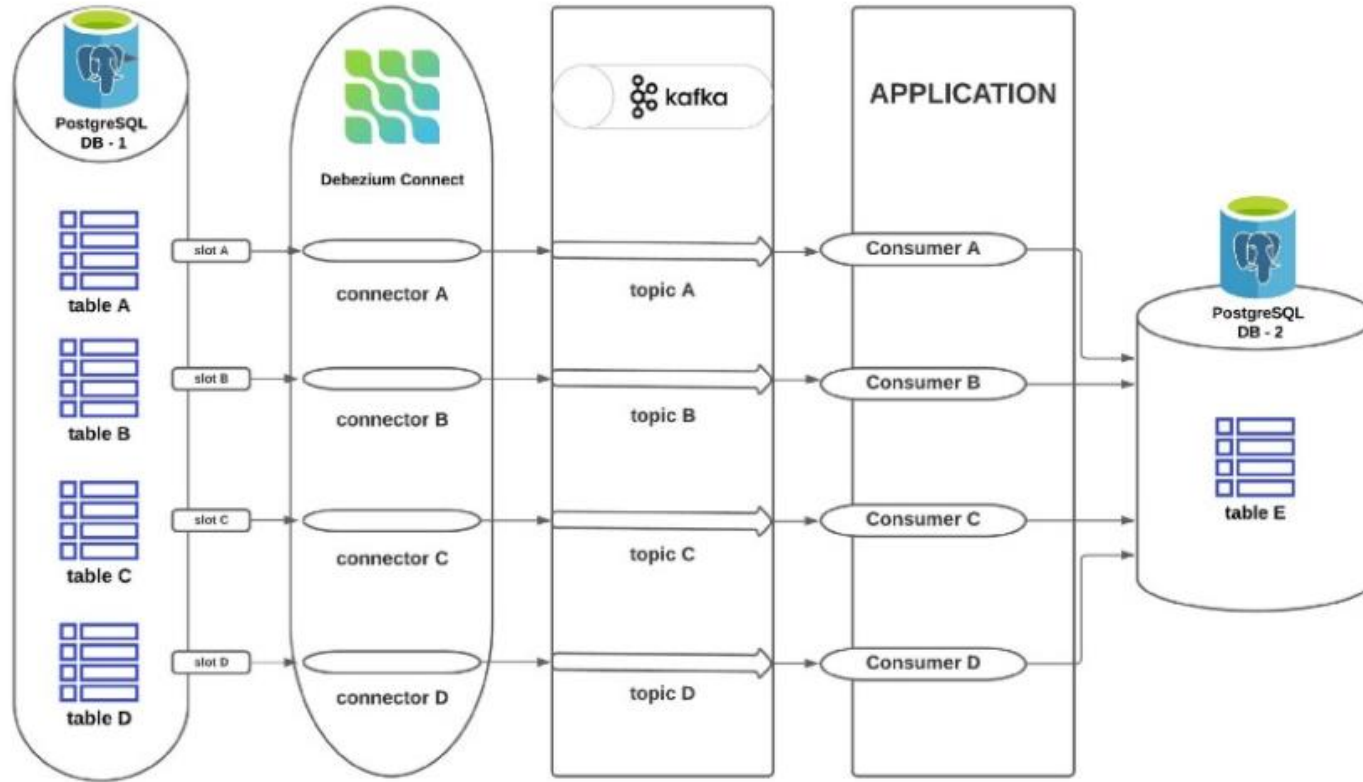
Apache Kafka (Broker, Topic replication)



Apache Kafka (Consumer & Consumer Groups)



Örnek: Sorunu



Sistem tasarımı

Örnek: Çözümü

- Gereksiz Sütunları Yoksaymak
 - Varsayılan olarak, bir satırda değişiklik olduğunda, Debezium mesajdaki tüm sütunları gönderir.
 - Ancak, mesajdaki her sütuna ihtiyaç duyulmadığında bu mesajları blacklist'e alabiliriz.

```
-----  
"column.blacklist":  
"public.settlements.created_date, public.settlements.status"  
-----
```

Örnek: Çözümü

- İlk Anlık Görüntüyü Filtreleme
 - Veritabanındaki geçmiş değerler için ilk anlık görüntüye ihtiyaç duyulur.
 - Ancak taşınacak kayıtların her seferinde tamamen alınması yerine yalnızca bir alt kümesi alınmalı.

```
-----  
"snapshot.select.statement.overrides": "public.settlements",  
"snapshot.select.statement.overrides.public.settlements":  
"SELECT * FROM public.settlements WHERE order_date> ' 2020-11-06 '"  
-----
```

CDC & Triggers

Change Data Capture	Triggers
Kurulumu zordur.	Kurulumu kolaydır.
Çok güvenilir.	Çok güvenilir.
Hızlıdır.	Yüksek işlem gereksinimlerinde yeterince hızlı değildir.
Sunucu kaynakları üzerinde küçük bir ek yük olsa da, tabloların kendisi üzerinde hiçbir etkisi yoktur.	Her yazma işlemine ek yük getirir.
İşlem günlüğünü kullandığı için, kullanıcılar 'hile yapamaz' fakat yine de kaydı kimin değiştirdiğini izleyemezsiniz.	Çoğu durumda bir kaydı kimin değiştirdiğini izleyemezsiniz.
Tablolarda yüksek bir işlem gereksinimi varsa (<i>tüm gün boyunca her saniye</i> eklenen / değiştirilen binlerce kayıt gibi) kullanılması daha avantajlıdır.	Tablolarda yüksek bir işlem gereksinimi varsa (<i>tüm gün boyunca her saniye</i> eklenen / değiştirilen binlerce kayıt gibi), tetikleyiciler çok fazla ek yük ekleyebilir.
Okuma işlemleri izlenemiyor.	Okuma işlemleri izlenemiyor.