# SQL Change Guard Kullanım Kılavuzu

Sürüm: 1.0

Kapsam: Kurulum, yapılandırma, günlük kullanım, denetim

Hedef okuyucu: Ürünü ilk kez kullanan herkes

## Bölüm 0. Giriş ve Terimler

### 0.1 SQL Change Guard nedir

SQL Change Guard, veritabanlarına yapılan her müdahaleyi kayıt altına alan, onaya bağlayan ve kanıtını üreten bir yönetişim (governance) uygulamasıdır.

Kurumlarda tipik durum şudur: bir geliştirici bir SQL betiği yazar, bunu bir yöneticiye e-posta ile gönderir, yönetici "tamam" der, bir veritabanı yöneticisi betiği canlı sunucuda çalıştırır. Bu akışta üç şey eksiktir:

1. Betiğin ne yaptığı sistematik olarak incelenmemiştir.

2. Onayın kim tarafından, ne zaman ve hangi kurala göre verildiği kayıtlı değildir.

3. Altı ay sonra denetçi "bu değişikliği kim onayladı" diye sorduğunda cevap e-posta kutularında aranır.

SQL Change Guard bu üç boşluğu kapatır. Betik uygulamaya girer, otomatik olarak çözümlenir, riski hesaplanır, kural setine göre gereken onay adımları oluşur, onaylandıktan sonra uygulama tarafından çalıştırılır ve her adım değiştirilemez bir denetim kaydına yazılır.

### 0.2 Üç odak alanı

Ürün üç başlıkta çalışır. Kılavuz boyunca hangi bölümün hangi başlığa hizmet ettiği belirtilir.

| **Odak alanı** | **Anlamı** | **Kılavuzdaki bölümler** |
|---|---|---|
| Database Change Governance | Veritabanı yapısını değiştiren işlerin yönetimi. Kim ne değiştirdi, kim onayladı, ne zaman çalıştı. | 10, 11, 12, 14, 15, 16 |
| Production Query Governance | Canlı ortamdan veri okuma işlerinin yönetimi. Kim hangi sorguyu çalıştırdı, sonuç kime gitti. | 13 |
| Data Governance & Compliance | Kişisel ve hassas verinin korunması, denetim kanıtı üretimi. | 9, 13, 17, 18, 20 |

Ürünün genel konumlandırması Database Operations Governance başlığıdır: veritabanı operasyonlarının yönetişimi.

### 0.3 Kimler kullanır

| **Rol** | **Kılavuzda öncelikli bölümler** |
|---|---|
| Geliştirici | 11, 12, 14, 22, 26 |
| Veritabanı yöneticisi | 6, 11, 12, 15, 25, 26 |
| Değişiklik yöneticisi | 10, 11, 19, 20 |

| Rol | Kılavuzda öncelikli bölümler |
|---|---|
| Bilgi işlem yöneticisi | 2, 3, 7, 8, 25 |
| Denetçi | 17, 18, 20, 23 |
| Sistem yöneticisi | 2, 3, 4, 5, 6, 7, 8, 9, 25 |

## 0.4 Terimler sözlüğü

Kılavuz boyunca kullanılan terimler. Her biri ilgili bölümde ayrıntılı anlatılır.

| Terim | Açıklama |
|---|---|
| Talep (Change Request) | Bir veya birden çok SQL betiğinin, bir veya birden çok hedef sunucuda çalıştırılması için açılan kayıt. Kısaltması CR. |
| Sorgu Sonucu (Query Result) | Canlı ortamdan veri okumak için açılan talep türü. Sonuç şifreli bir dosya olarak teslim edilir. Kısaltması QR. |
| Rollback talebi | Bir değişiklik talebini geri almak için ona bağlı olarak açılan talep. |
| Betik (Script) | Talebin içindeki tek bir SQL metni. Bir talep birden çok betik taşıyabilir. |
| Ayrıştırma (Parse) | Betiğin metin olarak değil, yapı olarak çözümlenmesi. Hangi nesneye hangi işlemin yapıldığı buradan çıkar. |
| SQL standardı | Betikte aranan bir kural. Örnek: "xp_cmdshell çağrısı var mı". |
| Kritik nesne | Kurumun özel olarak işaretlediği veritabanı nesnesi. Bu nesneye dokunan talep farklı davranır. |
| Risk bandı (Band) | Talebin risk seviyesi. Beş kademe: Info, Low, Medium, High, Critical. |
| Risk skoru | Bandın 0-100 aralığında okunabilir karşılığı. Yüksek sayı daha güvenli demektir. |
| Onay politikası | Hangi durumda kaç kişinin, hangi rolle onay vereceğini tanımlayan kural seti. |
| Onay adımı | Politikanın ürettiği tek bir onay isteği. |
| Dört göz (Four eyes) | Bir işlemin en az iki farklı kişi tarafından tamamlanması ilkesi. |
| Görevler ayrılığı (SoD) | Talebi açan, onaylayan ve çalıştıran kişilerin ayrı olması ilkesi. |
| Acil durum talebi (Emergency) | Normal onay yolunun kısaltıldığı, gerekçesi zorunlu talep türü. |

| Terim | Açıklama |
|---|---|
| Sandbox | Betiğin canlıya girmeden önce bir test sunucusunda denenmesi. |
| Sürüm paketi (Release Package) | Birden çok talebin sıralı biçimde birlikte çalıştırılması. |
| Denetim kaydı (Audit log) | Sistemde olan her olayın değiştirilemez kaydı. |
| Denetim dosyası (Evidence Dossier) | Tek bir talebin tüm yaşam döngüsü kanıtını içeren ZIP paketi. |
| Maskeleme | Sorgu sonucundaki hassas değerlerin okunamaz hale getirilmesi. |
| Worker | Zamanlanmış işleri arka planda yürüten Windows servisi. |

## Bölüm 1. Mimari ve Bileşenler

### 1.1 Genel görünüm

Sistem dört ana parçadan oluşur. Bunlar birbirinden bağımsız kurulabilir.

## 1.2 Bileşenler ve görevleri

| Bileşen | Ne yapar | Çalışmazsa ne olur |
|---|---|---|
| API | Tüm iş kuralları burada çalışır. Yetki kontrolü, onay akışı, risk hesabı, denetim yazımı. | Hiçbir ekran çalışmaz. |
| Web arayüzü | Kullanıcının gördüğü ekranlar. İş kuralı içermez, her isteği API'ye sorar. | Kullanıcı ekranlara erişemez, ancak arka plan işleri devam eder. |
| Worker servisi | Zamanlanmış çalıştırma, sunucu bağlantı kontrolü, bildirim gönderimi, denetim bütünlük kontrolü, bakım işleri. | Zamanlanmış talepler çalışmaz, e-posta gitmez, bağlantı durumu güncellenmez. Elle çalıştırma yine mümkündür. |
| Uygulama veritabanı | Talepler, kurallar, kullanıcılar, denetim kaydı. | Sistem tümüyle durur. |
| Immutable veritabanı | Denetim kaydının ayrı bir veritabanına yazılmış ikinci kopyası. İsteğe bağlıdır. | Denetim kaydı yine ana veritabanında tutulur, ancak ikinci kopya karşılaştırması yapılamaz. |
| Hedef sunucular | Değişikliklerin ve sorguların gerçekten çalıştığı veritabanları. | Sadece o sunucuya ait talepler çalışmaz. |

## 1.3 Desteklenen veritabanı türleri

DatabaseType değerleri:

| Değer | Tür |
|---|---|
| 1 | SQL Server |
| 2 | Oracle |
| 3 | PostgreSQL |

İki ayrı yerde kullanılır ve karıştırılmamalıdır:

1. Uygulamanın kendi veritabanı: DatabaseProvider ayarı ile seçilir. Bu kılavuzun anlattığı kurulum SQL Server üzerinedir.

2. Hedef sunucular: Her sunucu tanımının kendi türü vardır. Aynı kurulum aynı anda SQL Server, Oracle ve PostgreSQL sunucularını yönetebilir.

Hangi hedef türlerinin kullanılabileceği lisanstan gelir. Ayrıntı: Bölüm 3.

### 1.4 Worker işleri

Worker servisi Quartz zamanlayıcısı ile çalışır. Her iş kendi periyoduna sahiptir ve periyotlar API tarafındaki appsettings.json içindeki Workers bölümünden okunur.

| İş | Ne yapar | Ayar anahtarı | Varsayılan |
|---|---|---|---|
| MainWorkJob | Sırayla dört şey yapar: (1) zamanı gelmiş sürüm paketlerini çalıştırır, (2) kuyruğa alınmış talepleri çalıştırır, (3) bekleyen sandbox işlerini çalıştırır, (4) zamanı gelmiş talepleri çalıştırır. | Workers:MainWorkSeconds | 30 saniye |
| ConnectionCheckJob | Tanımlı sunuculara bağlanmayı dener, durum bilgisini günceller. | Workers:ConnectionCheckSeconds | 60 saniye |
| NotificationJob | Kuyruktaki e-posta ve bildirimleri gönderir. | Workers:NotificationFlushSeconds | 60 saniye |
| MaintenanceJob | Bakım görevlerini (MaintenanceTasks tablosu) zamanına göre çalıştırır. | Workers:MaintenanceSeconds | 30 saniye |
| AuditIntegrityJob | Denetim kaydı zincirini doğrular, bozulma varsa uyarı üretir. | Kod içinde sabit periyot | - |
| HeartbeatJob | Worker servisinin ayakta olduğunu API'ye bildirir. | Kod içinde sabit periyot | - |

Açma ve kapama bayrakları:

| Bayrak | Etkisi |
|---|---|
| Workers:MainWorkEnabled | false ise otomatik çalıştırma tümüyle durur. |
| Workers:ConnectionCheckEnabled | false ise sunucu bağlantı durumu güncellenmez. |
| Workers:MaintenanceEnabled | false ise bakım görevleri çalışmaz. |

MainWorkJob aynı anda iki kez çalışmaz. Bir tur bitmeden yeni tur başlamaz. Bu, aynı talebin iki kez çalıştırılmasını engelleyen ilk katmandır. İkinci katman veritabanı seviyesindedir: bir talebi çalıştırmaya başlamadan önce worker o talebi "sahiplenir"; başka bir worker aynı talebi sahiplenemez.

### 1.5 Ölçekleme ve dağıtım

| **Bileşen** | **Kaç kopya çalıştırılabilir** | **Dikkat edilmesi gereken** |
|---|---|---|
| API | Birden çok kopya çalıştırılabilir. Oturum bilgisi jetonun içinde taşınır, sunucu tarafında oturum durumu tutulmaz. | İstek sınırlama sayaçları kopya bazındadır. İki kopya varsa kullanıcı başına etkin sınır iki katına çıkar. Kopya sayısı kaydetme hızını artırmaz; sebebi bölüm 1.6'da açıklanmıştır. |
| Web arayüzü | Birden çok kopya çalıştırılabilir. | Her kopyanın adresi AllowedOrigins listesinde bulunmalıdır. |
| Worker | Tek kopya çalıştırılmalıdır. | Bir talebin sahiplenilmesi veritabanı seviyesinde atomiktir, yani iki worker aynı talebi aynı anda alamaz. Ancak worker'lar arasında kiralama (lease) mekanizması yoktur: yarım kalmış çalıştırmaların toparlanması yalnızca zaman eşiğine dayanır. Bkz. 25.7. |
| Uygulama veritabanı | Tek veritabanı. | Yüksek erişilebilirlik gerekiyorsa veritabanı sunucusu seviyesinde çözülür (yük devretme kümesi gibi). |

Kapasite ile ilgili pratik notlar:

- Bir talebin çalışma süresi hedef sunucudaki betiğin süresine bağlıdır; uygulama tarafında bir üst sınır MaxQueryTimeout parametresidir.

- Worker turları üst üste binmediği için, çok uzun süren bir talep aynı turdaki diğer işleri bekletir. Uzun süren bakım işlerini yoğun olmayan saatlere zamanlamanız önerilir.

- Rapor sorguları ayrı bir zaman aşımına tabidir (Reports:QueryTimeoutSeconds) ve uygulama veritabanı üzerinde çalışır. Çok geniş tarih aralıklarında rapor almak uygulama veritabanını yorar.

### 1.6 Ölçülmüş kapasite ve boyutlandırma

Aşağıdaki değerler tahmin değildir; gerçek talepler uçtan uca akıştan geçirilerek ölçülmüştür ve ölçüm beş kez tekrarlanmıştır. Tablonun nasıl okunacağı ve önerilen sunucu, hemen altındaki başlıkta anlatılmıştır.

| Ölçü | Değer |
|---|---|
| Talep kaydetme süresi, ortanca | 0,1 ile 0,6 saniye |
| Talep kaydetme süresi, p95 | 0,2 ile 0,7 saniye |
| Tek kullanıcı için saniyede kaydedilen talep | 2 ile 9 |
| Talep başına veritabanı büyümesi | yaklaşık 57 KB |

Aralık, ölçüm makinesinin o anda başka ne yaptığına bağlıdır. Makine boştayken kaydetme onda bir saniye, makine yoğunken yarım saniye civarındadır. Kullanıcı açısından ikisi de anlık kabul edilir.

### Bu tablo nasıl okunmalı ve önerilen sunucu

Yayınlanan değerler ölçülen değerlerdir ve alt sınırdır. Ölçüm, uygulama ile veritabanının aynı makinede çalıştığı dört çekirdekli, 8 GB bellekli bir dizüstü bilgisayarda yapılmıştır. Ayrılmış bir sunucuda değerlerin bundan kötü olması beklenmez. Sunucu üzerinde ayrıca ölçüm yapılmadığı için bu belgede tahmini bir sunucu değeri yayınlanmaz: yazan her rakam gerçekten ölçülmüştür.

Önerilen asgari kurulum:

| Bileşen | Asgari | Not |
|---|---|---|
| Uygulama sunucusu (API ve Worker) | 4 çekirdek, 8 GB bellek, 20 GB disk | Web arayüzü de aynı sunucuda durabilir. |
| Uygulama veritabanı sunucusu | 4 çekirdek, 16 GB bellek, SSD | Beş yıllık saklama için 10 GB mertebesinde veri alanı yeterlidir. |
| Ağ | Uygulama sunucusu ile veritabanı sunucusu aynı yerel ağda olmalıdır | Sebebi aşağıda. |

Uygulama ile veritabanı arasındaki ağ mesafesi, işlemci seçiminden daha çok fark yaratır. Tek bir talep kaydı yüz civarında veritabanı gidiş dönüşü yapar. Aradaki gecikme 0,3 milisaniye ise bunların toplamı otuz milisaniyedir ve fark edilmez. Aynı gecikme 5 milisaniye olursa (farklı veri merkezi veya VPN arkası) yalnızca bu yüzden kaydetme süresine yarım saniye eklenir. Veritabanı sunucusu uygulama sunucusuna ağ olarak yakın tutulmalıdır.

### Depolama planlaması

Talep başına 57 KB üzerinden:

| Günlük talep | Yıllık büyüme |
|---|---|
| 20 | yaklaşık 400 MB |

| Günlük talep | Yıllık büyüme |
|---|---|
| 50 | yaklaşık 1 GB |
| 100 | yaklaşık 2 GB |

Bu rakam ham büyümedir. Bakım işi eski kayıtları arşivlediğinde büyüme yavaşlar.

Talep başına yazılan satır sayısı ilk bakışta beklenenden fazladır, çünkü her değişiklik talebi kendi geri alma talebini de doğurur ve veritabanı denetim izi her yazma işlemini ayrı satırda saklar.

### Eş zamanlı kullanım ve ekip büyüklüğü

Denetim izi zinciri, kayıtların birbirine bağlı olmasını garanti etmek için kaydetme işlemlerini sıraya sokar. Bu yüzden aynı anda kaydetme yapan kullanıcı sayısı arttıkça kaydetme süresi uzar.

Ölçülen davranış, beş turun toplamı:

| Tam olarak aynı anda kaydeden kişi | Denenen | Kullanıcıya hata dönen |
|---|---|---|
| 1 | 40 | 0 |
| 2 | 80 | 0 |
| 4 | 160 | 3 |
| 8 | 320 | 4 |

İki kişiye kadar hiç hata yoktur. Dört ve sekiz kişi tam olarak aynı anda kaydettiğinde her yüz istekten bir ile ikisi kullanıcıya hata döner. Böyle bir durumda kaydet düğmesine yeniden basmak yeterlidir; kayıp veri olmaz, yarım kayıt oluşmaz.

Sistem genelinde ölçülen kaydetme tavanı saniyede yaklaşık 20 taleptir. Bu tavan işlemciden değil denetim izi zincirinden gelir. Buradan çıkan pratik bir sonuç vardır: API'yi birden çok kopya çalıştırmak kaydetme hızını artırmaz, çünkü sıraya sokma noktası uygulamada değil veritabanındadır. Birden çok kopya okuma tarafındaki yükü paylaştırır.

Kaç kişilik ekibi kaldırır. Ekip büyüklüğü pratikte sınır değildir. Birkaç yüz kişilik bir ekip desteklenir. Sınırı belirleyen şey ekip mevcudu değil, ekibin gerçekte kaç değişiklik yazdığıdır. Saniyede 20 kaydetme, sekiz saatlik bir iş gününde yüz binlerce kaydetme demektir; kurumsal veritabanı değişiklik hacmi bu mertebede değildir. Dört kişinin tam olarak aynı saniyede kaydete basması, ancak günlük talep sayısı binleri bulduğunda sıradan hale gelir. Günde elli talep üreten bir ekipte aynı anda kaydetme durumu haftada bir mertebesindedir. Yoğun sürüm saatlerinde taleplerin gün içine yayılması bu ihtimali daha da azaltır.

### Sunucu sayısı

Yönetilen sunucu sayısı kaydetme hızını etkilemez. Sınırı belirleyen şey bağlantı sağlık kontrolüdür: worker altmış saniyede bir tanımlı sunucuların tamamını sırayla yoklar ve erişilemeyen bir sunucu için otuz saniye bekler.

| Ortam | Önerilen sunucu sayısı |
|---|---|
| Yerel ağdaki sunucular | 100 ve üzeri |
| Uzak veya bulut sunucular | 30 ile 40 |

Erişilemeyen sunucular turu uzatır. Kullanımdan kalkan sunucuları pasif duruma almak, sağlık kontrolünün zamanında tamamlanmasını sağlar.

### Sınırların özeti

| Sınır | Değer | Sınırı belirleyen |
|---|---|---|
| Ekip büyüklüğü | birkaç yüz kişi | Ekip mevcudu değil, günlük talep sayısı |
| Aynı anda kaydeden kişi | 2 kişiye kadar hatasız; 4 ve 8 kişide yüz istekte 1 ile 2 tekrar | Denetim izi zinciri |
| Sistem geneli kaydetme tavanı | saniyede yaklaşık 20 talep | Denetim izi zinciri |
| Günlük talep | 5.000 talepte sistemin meşgul olduğu süre oranı yüzde 2 | Yukarıdaki tavan |
| Yönetilen sunucu | yerel ağda 100 ve üzeri, uzak veya bulutta 30 ile 40 | Bağlantı sağlık kontrolü turu |
| Depolama | talep başına yaklaşık 57 KB | Ölçüldü |

Bu değerler talep kaydetme yolu için, yeni kurulmuş bir veritabanında ölçülmüştür. Uzun süre biriken veride liste ve rapor sorguları veri hacmine bağlı hale gelir; bakım görevleriyle arşivleme bu nedenle önerilir.

## Bölüm 2. Kurulum

Bu bölüm sıfırdan kurulum içindir. Sıra önemlidir.

### 2.1 Ön koşullar

| Gereksinim | Açıklama |
|---|---|
| Uygulama sunucusu donanımı | En az 4 çekirdek, 8 GB bellek, 20 GB disk. API, Worker ve web arayüzü aynı sunucuda çalışabilir. |
| Veritabanı sunucusu donanımı | En az 4 çekirdek, 16 GB bellek, SSD. Beş yıllık saklama için 10 GB mertebesinde veri alanı yeterlidir. |
| Ağ yakınlığı | Uygulama sunucusu ile uygulama veritabanı sunucusu aynı yerel ağda olmalıdır. Sebebi bölüm 1.6'da açıklanmıştır. |
| .NET 10 çalışma zamanı | API ve Worker bu sürümle derlenir (net10.0). |
| SQL Server 2016 veya üzeri | Uygulama veritabanı için. Daha yeni sürüm gerektiren özellik kullanılmaz. |
| Windows sunucu | Worker Windows servisi olarak çalışır. Denetim dosyası PDF üretimi de Windows yazı tiplerini kullanır. |
| Ağ erişimi | API sunucusundan hedef veritabanı sunucularına erişim gerekir. SQL Server için 1433, Oracle için 1521, PostgreSQL için 5432 varsayılan portlardır. |
| SMTP sunucusu | Bildirim e-postaları için. Zorunlu değildir, kapatılabilir. |
| Servis hesabı | Worker servisinin çalışacağı hesap. Uygulama veritabanına ve QR dosya klasörüne yazma yetkisi olmalıdır. |

### 2.2 Kurulum adımları

Kurulum şu sırayla yapılır:

```
1. Veritabanini olustur
       |
       v
2. API appsettings.json dosyasini duzenle
       |
       v
3. API'yi baslat  (sema ve varsayilan veri otomatik olusur)
       |
       v
```

```
 4. Lisans dosyasini yukle
        |
        v
 5. Ilk yonetici ile giris yap, sifreyi degistir
        |
        v
 6. Sunucuları tanimla
        |
        v
 7. Kullanicilari ekle
        |
        v
 8. Parametreleri ve onay politikalarini ayarla
        |
        v
 9. Worker servisini baslat
        |
        v
10. Dogrulama kontrol listesini uygula
```

### 2.3 Veritabanının oluşturulması

Boş bir veritabanı oluşturmanız yeterlidir. Tablolar, saklı yordamlar ve varsayılan veriler API ilk açıldığında otomatik kurulur.

Bunu sağlayan ayar:

```
"Database": {
  "AutoMigrate": true
}
```

AutoMigrate true iken API açılışta şu dosyaları sırayla uygular:

| **Dosya** | **İçerik** |
|---|---|
| 001_Tables.sql | Tüm tablolar ve indeksler |
| 002_Migrations.sql | Şema sürüm takibi |
| 003_SeedData.sql | Roller, varsayılan yönetici, parametreler, SQL standartları, rapor tanımları, bakım görevleri |

| Dosya | İçerik |
|---|---|
| 004_PostSeedFixes.sql | Varsayılan veri düzeltmeleri |
| 005_StoredProcedures.sql | Rapor saklı yordamları |
| 006_TableWriteLogs.sql | Veritabanı seviyesi yazma kaydı ve tetikleyicileri |

Betikler tekrar çalıştırılabilir biçimde yazılmıştır. Var olan kayıt tekrar eklenmez, sizin yaptığınız değişiklikler ezilmez.

AutoMigrate false yapılırsa bu dosyaları veritabanı yöneticisinin elle uygulaması gerekir. Sıkı değişiklik kontrolü olan kurumlarda bu tercih edilir.

### 2.4 appsettings.json alan alan

Aşağıdaki tablo API tarafındaki appsettings.json dosyasının tüm bölümlerini açıklar. Üretim ortamında REPLACE_IN_PROD yazan her değer mutlaka değiştirilmelidir.

#### 2.4.1 ConnectionStrings

| Anahtar | Ne işe yarar | Örnek |
|---|---|---|
| ConnectionStrings:Default | Uygulamanın kendi kayıt veritabanının bağlantı metni. | Server=SQLSRV01;Database=SQLChangeGuard;User Id=scg;Password=...;TrustServerCertificate=True |
| DatabaseProvider | Uygulama veritabanı sağlayıcısı. | SqlServer |

Uygulamanın kendi kayıt veritabanı SQL Server üzerinde çalışır. PostgreSQL ve Oracle yönetilen hedef veritabanlarıdır; onların bağlantı bilgileri burada değil, Sunucular ekranındaki kayıtlarda tutulur.

Yanlış verilirse: API açılışta bağlanamaz ve günlük dosyasına bağlantı hatası yazar. Hiçbir ekran açılmaz.

#### 2.4.2 ImmutableDb

Denetim kaydının ikinci, değiştirilemez kopyasının yazılacağı ayrı veritabanı.

| Anahtar | Ne işe yarar |
|---|---|
| ImmutableDb:Enabled | true ise her denetim kaydı ikinci veritabanına da yazılır. |
| ImmutableDb:ConnectionStrings:Default | İkinci veritabanının bağlantı metni. |

Bu veritabanının kullanıcısına yalnızca ekleme (INSERT) yetkisi verilmesi önerilir. Amaç, ana veritabanına tam yetkiyle erişen birinin denetim izini geriye dönük düzeltememesidir. Ayrıntı: Bölüm 17.

### 2.4.3 Auth

| Anahtar | Ne işe yarar | Varsayılan | Örnek etki |
|---|---|---|---|
| Auth:Mode | Kimlik doğrulama biçimi. Local yerel kullanıcı, LDAP kullanımı için LDAP bölümü doldurulur. | Local | |
| Auth:AccountLockoutMaxAttempts | Kaç hatalı şifre denemesinden sonra hesap kilitlenir. | 5 | 5 iken altıncı yanlış denemede hesap kilitlenir. |
| Auth:AccountLockoutMinutes | Kilit süresi (dakika). | 15 | 15 iken kilitlenen hesap 15 dakika sonra kendiliğinden açılır. |
| Auth:Password:MinLength | En az şifre uzunluğu. | 6 | |
| Auth:Password:RequireUppercase | Büyük harf zorunlu mu. | false | |
| Auth:Password:RequireDigit | Rakam zorunlu mu. | true | true iken "sifre" reddedilir, "sifre1" kabul edilir. |
| Auth:Password:HistoryCount | Son kaç şifre tekrar kullanılamaz. 0 = kontrol yok. | 0 | 3 iken kullanıcı son üç şifresinden birini tekrar seçemez. |
| Auth:Password:ExpiryDays | Şifre kaç günde bir değişmeli. 0 = süresiz. | 0 | 90 iken 90 gün sonra kullanıcı şifre değiştirmeye zorlanır. |
| Auth:Password:ResetTokenExpiryMinutes | Şifre sıfırlama bağlantısının geçerlilik süresi. | 30 | |
| Auth:Password:ResetBaseUrl | Şifre sıfırlama e-postasındaki bağlantının kök adresi. Web arayüzünün adresi yazılır. | http://localhost:5173 | Yanlış yazılırsa kullanıcı e-postadaki bağlantıya tıkladığında hatalı sayfaya gider. |

### 2.4.4 Jwt

Kullanıcı giriş yaptığında verilen erişim jetonunun ayarları.

| Anahtar | Ne işe yarar | Varsayılan |
|---|---|---|
| Jwt:Issuer | Jetonu üreten tarafın adı. | SCG |
| Jwt:Audience | Jetonu kullanacak tarafın adı. | SCG-Clients |
| Jwt:AccessTokenMinutes | Erişim jetonunun ömrü (dakika). Süre dolunca yenileme jetonu ile yenilenir. | 120 |
| Jwt:RefreshTokenHours | Yenileme jetonunun ömrü (saat). Bu da dolarsa kullanıcı yeniden giriş yapar. | 48 |
| Jwt:SigningKey | Jetonun imza anahtarı. Üretimde mutlaka değiştirin. | REPLACE_IN_PROD |

SigningKey neden önemlidir: jeton, bu anahtarla imzalanır. Anahtarı bilen biri istediği kullanıcı adına geçerli jeton üretebilir, yani parola bilmeden sisteme girebilir. En az 32 karakter, rastgele üretilmiş bir değer kullanın.

Anahtar değiştirilirse: o anda açık olan tüm oturumlar geçersiz olur ve herkes yeniden giriş yapar. Veri kaybı olmaz.

### 2.4.5 Security (şifreleme anahtarları)

| Anahtar | Ne işe yarar | Kaybolursa ne olur |
|---|---|---|
| Security:ServerPasswordKey | Sunucu tanımlarındaki veritabanı şifrelerinin diskte şifrelenmesi için kullanılan AES-256 anahtarı. 32 baytlık, base64 kodlu değer. | Kayıtlı sunucu şifreleri çözülemez. Her sunucu tanımına şifre yeniden girilmelidir. |
| Security:QrZipPasswordKey | Sorgu sonucu ZIP paketlerinin parolalarının şifrelenmesi için kullanılan AES-256 anahtarı. | Eski sonuç dosyalarının parolaları görüntülenemez. |
| Audit:ActiveKeyId ve Audit:HmacKeys | Denetim kaydı imzası için kullanılan HMAC anahtar halkası. | Eski kayıtların imzası doğrulanamaz ve "doğrulanamadı" olarak işaretlenir. |

Bu anahtarları nasıl üretirsiniz: 32 baytlık rastgele bir değeri base64 olarak kodlayın. PowerShell ile:

```
$bytes = New-Object byte[] 32
[System.Security.Cryptography.RandomNumberGenerator]::Create().GetBytes($bytes)
[Convert]::ToBase64String($bytes)
```

Bu anahtarların appsettings.json içinde düz metin durması yerine ortam değişkeni veya kullanıcı sırları (user secrets) ile verilmesi önerilir.

Anahtar rotasyonu tuzağı: ServerPasswordKey değiştirilirse eski şifrelenmiş değerler yeni anahtarla çözülemez. Anahtar değiştirmeden önce tüm sunucu şifrelerinin yeniden girileceğini planlayın. Denetim HMAC anahtarında ise durum farklıdır: anahtar halkası birden çok anahtar tutabilir, eski kayıtlar kendi anahtar kimliğiyle doğrulanmaya devam eder.

### 2.4.6 License

| Anahtar | Ne işe yarar | Varsayılan |
|---|---|---|
| License:LicensePath | Lisans dosyasının yolu. | license.scglicense |
| License:PublicKeyPem | Lisans imzasının doğrulandığı açık anahtar. Değiştirilmez. | Ürünle gelir |
| License:ExpiryWarningDays | Lisans bitişine kaç gün kala uyarı verilsin. | 30 |

ExpiryWarningDays 30 iken, lisansın bitmesine 29 gün kaldığında lisans durumu "Expiring" olur ve arayüzde uyarı görünür. Sistem çalışmaya devam eder.

### 2.4.7 Mail

| Anahtar | Ne işe yarar | Not |
|---|---|---|
| Mail:Enabled | E-posta gönderimi açık mı. | false ise hiç e-posta gitmez, kayıt yine tutulur. |
| Mail:SmtpHost | SMTP sunucu adresi. | |
| Mail:SmtpPort | SMTP portu. | 587 yaygın kullanımdır. |
| Mail:Username, Mail:Password | SMTP kimlik bilgileri. | |
| Mail:FromAddress, Mail:FromName | Gönderen adresi ve görünen adı. | |
| Mail:EnableSsl | Şifreli bağlantı kullanılsın mı. | Üretimde true olmalıdır. |
| Mail:DevMode | true ise tüm e-postalar gerçek alıcı yerine tek bir test adresine gider. | Üretimde mutlaka false yapın. |
| Mail:DevOverrideEmail | DevMode açıkken kullanılacak test adresi. | |

Üretim kurulumunda en sık yapılan hata: DevMode true bırakmak. Bu durumda talep sahiplerine hiç e-posta ulaşmaz, hepsi test kutusuna düşer.

### 2.4.8 Ldap

Kurum dizini üzerinden giriş için kullanılır.

| Anahtar | Ne işe yarar |
|---|---|
| Ldap:Host | Dizin sunucusu adresi. |
| Ldap:Port | Port. Şifresiz 389, şifreli 636. |
| Ldap:UseSsl | Şifreli bağlantı kullanılsın mı. |
| Ldap:BaseDn | Aramanın başlayacağı kök. |
| Ldap:BindDn, Ldap:BindPassword | Dizinde arama yapacak servis hesabı. |
| Ldap:Domain | Etki alanı kısa adı. |
| Ldap:TimeoutSeconds | Bağlantı zaman aşımı. |
| Ldap:AllowUntrustedCertificate | Güvenilmeyen sertifikaya izin verilsin mi. Üretimde false olmalıdır. |

LDAP kullanıcıları ile yerel kullanıcıların farkı Bölüm 4.8'de anlatılır.

### 2.4.9 Diğer ayarlar

| Anahtar | Ne işe yarar | Varsayılan |
|---|---|---|
| Display:TimeZone | Belgelerde ve sunucu tarafında üretilen çıktılarda saatlerin gösterileceği saat dilimi. | Europe/Istanbul |
| AllowedOrigins | Web arayüzünün çalıştığı adresler. Buraya yazılmayan adresten gelen istek reddedilir. | localhost adresleri |
| Mfa:Enabled | İki adımlı doğrulamanın ana anahtarı. Kapalıysa hiçbir kullanıcı iki adımlı doğrulama kuramaz ve girişte kod istenmez. Açık olması tek başına kod zorunlu kılmaz; ayrıntı aşağıda. | true |
| Mfa:Method | Yöntem. Totp mobil doğrulayıcı uygulaması demektir. | Totp |
| Mfa:Issuer | Doğrulayıcı uygulamada görünecek ad. | SQLChangeGuard |

İki adımlı doğrulama nasıl devreye girer: Ana anahtar açık olsa bile kod, yalnızca kendi hesabında iki adımlı doğrulamayı kurmuş kullanıcılardan istenir. Kurulum kullanıcı bazındadır ve kullanıcının kendisi tarafından yapılır. Ana anahtar kapatılırsa mevcut kurulumlar da devre dışı kalır.

Dizin (LDAP) hesapları için iki adımlı doğrulama kurulamaz; bu hesaplarda kimlik doğrulaması kurum dizinine aittir.

| **QueryResult:OutputPath** | **Sorgu sonucu ZIP dosyalarının yazılacağı klasör.** | **C:\SCG\QrFiles** |
|---|---|---|
| QueryResult:MaxMailAttachmentMB | E-posta ekinin en büyük boyutu. Bunu aşan dosya eklenmez, kullanıcı arayüzden indirir. | 10 |
| Reports:QueryTimeoutSeconds | Rapor sorgularının zaman aşımı. | 500 |
| RateLimiting:PermitPerMinute | Dakikada izin verilen istek sayısı. | 60 |
| RateLimiting:QueueLimit | Sıraya alınacak istek sayısı. | 10 |
| Workers:ApiKey | Worker servisinin API'ye kimliğini kanıtladığı anahtar. Worker tarafındaki WorkerApi:Key ile aynı olmalıdır. | REPLACE_IN_PROD |
| Workers:WorkerErrorMailTo | Worker beklenmedik hata verirse bildirimin gideceği adres. | |
| Workers:StaleExecutionRecoveryMinutes | Çökme kurtarma eşiği (dakika). Bu süreden daha uzun süredir devam eden çalıştırmalar ölü kabul edilip talep Approved durumuna döndürülür. Ayrıntı: Bölüm 25.7. | 120 |

QueryResult:OutputPath klasörüne worker servis hesabının yazma yetkisi olmalıdır. Yetki yoksa sorgu çalışır ama dosya oluşturulamaz ve talep detayında paketleme hatası görünür.

### 2.5 Worker tarafı appsettings.json

Worker'ın kendi dosyası kısadır:

| **Anahtar** | **Ne işe yarar** |
|---|---|
| WorkerApi:BaseUrl | API adresi. Worker durum bilgisini buraya bildirir. |
| WorkerApi:Key | API tarafındaki Workers:ApiKey ile aynı olmalıdır. |
| Serilog:MinimumLevel | Günlük ayrıntı seviyesi. |

İki anahtar birbirini tutmazsa: worker çalışır ama API'ye durum bildiremez. Arayüzde worker "çevrimdışı" görünür.

### 2.6 İlk yönetici ve ilk giriş

Kurulumda hiç kullanıcı yoksa tek bir yönetici hesabı oluşturulur:

| Alan | Değer |
|---|---|
| Kullanıcı adı | admin |
| Başlangıç şifresi | admin123 |
| Rol | SystemManager (35) |

İlk giriş sonrasında bu şifreyi mutlaka değiştirin. Bu hesap üretim kopyasında tek başına gelir; başka hiçbir örnek kullanıcı oluşturulmaz.

### 2.7 Kurulum doğrulama kontrol listesi

Aşağıdakilerin hepsi sağlanıyorsa kurulum başarılıdır:

[ ] API adresine tarayıcıdan erişiliyor ve giriş ekranı açılıyor.

[ ] admin hesabıyla giriş yapılabiliyor ve şifre değiştirildi.

[ ] Ayarlar > Lisans ekranında lisans durumu geçerli görünüyor.

[ ] Ayarlar > Sunucular ekranında en az bir sunucu tanımlı ve bağlantı testi başarılı.

[ ] Ayarlar > Tanılama ekranındaki kontroller yeşil.

[ ] Worker servisi çalışıyor ve arayüzde çevrimiçi görünüyor.

[ ] Test amaçlı bir talep açılıp onaylanabiliyor.

[ ] Denetim Kaydı ekranında bu işlemlerin kaydı görünüyor.

### 2.8 Sık karşılaşılan kurulum hataları

| Belirti | Olası sebep | Çözüm |
|---|---|---|
| API açılmıyor, günlükte bağlantı hatası var | ConnectionStrings:Default yanlış veya veritabanı yok | Bağlantı metnini ve veritabanı erişimini kontrol edin. |
| Ekranlar açılıyor ama her istek yetkisiz dönüyor | Jwt:SigningKey değiştirilmiş, eski oturum açık | Çıkış yapıp yeniden giriş yapın. |
| Web arayüzü API'ye erişemiyor | AllowedOrigins içinde arayüz adresi yok | Arayüz adresini listeye ekleyin ve API'yi yeniden başlatın. |

| Belirti | Olası sebep | Çözüm |
|---|---|---|
| Yazma işlemleri 402 hatası veriyor | Lisans süresi dolmuş veya geçersiz | Geçerli lisans dosyası yükleyin. Ayrıntı: Bölüm 3.6. |
| E-posta gitmiyor | Mail:Enabled false veya DevMode true | Ayarları düzeltin, Bildirim Kayıtları ekranından sonucu izleyin. |
| Zamanlanmış talep çalışmıyor | Worker durmuş veya MainWorkEnabled false | Servisi başlatın, bayrağı true yapın. |
| Sorgu sonucu dosyası oluşmuyor | QueryResult:OutputPath yok veya yazma yetkisi yok | Klasörü oluşturun, servis hesabına yazma yetkisi verin. |
| Sunucu şifresi çözülemiyor hatası | Security:ServerPasswordKey değişmiş | Eski anahtarı geri koyun veya sunucu şifrelerini yeniden girin. |

## Bölüm 3. Lisans

### 3.1 Lisans dosyası nedir

Lisans, .scglicense uzantılı imzalı bir dosyadır. İçinde müşteri adı, geçerlilik tarihleri, açık modüller, izin verilen sunucu sayısı ve izin verilen veritabanı türleri bulunur.

Dosya dijital imzalıdır. İmza, License:PublicKeyPem ayarındaki açık anahtar ile doğrulanır. İçeriği elle değiştiren bir dosya imza doğrulamasından geçmez ve geçersiz sayılır.

### 3.2 Lisans içeriği

| **Alan** | **Ne işe yarar** | **Boş bırakılırsa** |
|---|---|---|
| CustomerName | Müşteri adı. Arayüzde gösterilir. | - |
| Edition | Sürüm adı. Bilgi amaçlıdır. | Core |
| MaxInstances | İzin verilen en fazla yönetilen sunucu sayısı. Sunucular tablosundaki tüm satırlar sayılır, pasif olanlar dahil. | 0 yazılırsa sınırsız |
| DatabaseTypes | İzin verilen hedef veritabanı türleri. | Boş liste = hepsi açık |
| Modules | Açık modüller. | Boş liste = hepsi açık |
| GracePeriodDays | Lisans bittikten sonra kaç gün daha çalışılabilir. | 7 |
| ValidFrom, ValidTo | Geçerlilik aralığı. | - |
| IssuedAt | Üretim tarihi. | - |

Dikkat edilmesi gereken kural: Modules listesi boş ise "hepsi açık" demektir. Hiçbir opsiyonel modülün açık olmadığını belirtmek için listeye tek başına None değeri yazılır. Bu iki durum karıştırılırsa çekirdek satışta bütün modüller açılmış olur.

### 3.3 Modüller

Altı opsiyonel modül vardır. Çekirdek işlevler (talep açma, onay akışı, çalıştırma, denetim kaydı, kullanıcı ve rol yönetimi, LDAP girişi) modülden bağımsız her zaman açıktır.

| **Modül** | **Açıkken gelen özellikler** | **Kapalıyken ne olur** |
|---|---|---|
| QueryGovernance | Sorgu sonucu talebi açma, hassas veri maskeleme, şifreli sonuç teslimi, sonuç dosyası indirme. | Sorgu sonucu talebi kaydedilemez ve çalıştırılamaz, indirme düğmesi gizlenir. |
| Sandbox | Talebi test sunucusunda deneme. | Sandbox düğmesi gizlenir, sandbox komutu reddedilir. |

| Modül | Açıkken gelen özellikler | Kapalıyken ne olur |
|---|---|---|
| RollbackRecovery | Otomatik geri alma betiği üretimi, sunucu bazında geri alma zorunluluğu, Oracle otomatik geri alma. | Geri alma üret düğmesi gizlenir. Ayrıca "Rollback Zorunlu" kuralı etkisizleşir: otomatik üretim yokken talep çalıştırmanın kilitlenmemesi için zorunluluk uygulanmaz. Sunucu tanımındaki bayrak görünmeye devam eder ama çalıştırmayı engellemez. |
| ScheduledExecution | Talebi ileri bir tarihe zamanlama, sunucu bazında otomatik çalıştırma. | Zamanla düğmesi gizlenir, onaydan sonra otomatik zamanlama yapılmaz. |
| ReleasePackages | Sürüm paketi oluşturma ve çalıştırma. | Sürüm paketi ekranındaki işlemler reddedilir. |
| InsightsReporting | Yönetim raporları ve Nesne Geçmişi ekranı. | Raporlar ve nesne geçmişi açılmaz. |

Önemli ilke: Kapalı modülün geçmiş verisi okunabilir kalır. Örneğin sürüm paketi modülü kapatılırsa eski paketler listede görünmeye devam eder, yalnızca yeni işlem yapılamaz. Veri rehin alınmaz.

Modül kapalıyken ilgili düğme arayüzde gizlenir, gri gösterilmez. Bunun sebebi, kullanıcının yapamayacağı bir işlemi denemeye çalışıp hata mesajı almasının anlamsız olmasıdır.

### 3.4 Sunucu limiti

MaxInstances değeri Sunucular tablosundaki satır sayısı ile karşılaştırılır. Pasif sunucular da sayılır. Limit dolduğunda yeni sunucu eklenemez, mevcut sunucular çalışmaya devam eder.

Örnek: MaxInstances 3 iken üç sunucu tanımlıysa dördüncüyü ekleme denemesi hata verir. Bir sunucuyu pasife almak yer açmaz; silmek gerekir.

### 3.5 Veritabanı türlerinin lisanstan gelmesi

Sunucu tanımı ekranındaki veritabanı türü listesi lisanstan doldurulur. Lisansta DatabaseTypes boşsa üç tür de seçilebilir. Örneğin lisansta yalnızca SqlServer varsa:

- Yeni sunucu eklerken listede yalnızca SQL Server görünür.

- Daha önce eklenmiş bir Oracle sunucusu varsa o sunucuda çalıştırma, bağlantı testi ve sandbox işlemleri engellenir. Kayıt silinmez, yalnızca kullanılamaz.

### 3.6 Lisans durumları ve sistem davranışı

| Durum | Anlamı | Sistem davranışı |
|---|---|---|
| Valid | Geçerli. | Her şey normal çalışır. |

| Durum | Anlamı | Sistem davranışı |
|---|---|---|
| Trial | Deneme lisansı. | Her şey çalışır, tüm modüller açık kabul edilir. |
| Expiring | Bitişe ExpiryWarningDays günden az kaldı. | Her şey çalışır, arayüzde uyarı görünür. |
| GracePeriod | Süre doldu ama ek süre içinde. | Her şey çalışır, uyarı görünür. |
| Expired | Süre ve ek süre doldu. | Kısıtlı mod. |
| Invalid | İmza doğrulanamadı veya dosya bozuk. | Kısıtlı mod. |
| Missing / NoLicense | Lisans dosyası yok. | Kısıtlı mod. |

Kısıtlı mod ne demektir: Tüm yazma işlemleri (POST, PUT, PATCH, DELETE) reddedilir ve 402 kodu ile "lisans geçersiz veya süresi dolmuş" hatası döner. İstisna üç uçtur: giriş işlemleri, lisans işlemleri ve API dokümantasyonu. Yani sisteme girip yeni lisans yükleyebilirsiniz, ama başka hiçbir kayıt oluşturamaz veya değiştiremezsiniz. Okuma işlemleri çalışmaya devam eder, mevcut verilerinizi görebilirsiniz.

### 3.7 Lisans yükleme adımları

1. Ayarlar ekranını açın.

2. Lisans sekmesine geçin. Bu sekmeyi görmek için rolünüzde lisans yönetimi yetkisi olmalıdır.

3. Mevcut durum kartında müşteri adı, bitiş tarihi ve kalan gün sayısı görünür.

4. Lisans dosyasını seçin veya içeriğini yapıştırın.

5. Kaydedin. Uygulama imzayı doğrular ve lisansı yeniden yükler.

Yükleme başarılıysa durum kartı hemen güncellenir. Uygulamayı yeniden başlatmak gerekmez.

### 3.8 Lisans hata mesajları

| Mesaj | Sebep | Çözüm |
|---|---|---|
| Lisans geçersiz veya süresi dolmuş | Kısıtlı modda yazma denemesi yapıldı | Geçerli lisans yükleyin. |
| İmza doğrulanamadı | Dosya değiştirilmiş veya başka bir kuruma ait | Doğru dosyayı tedarikçinizden isteyin. |
| Sunucu limiti aşıldı | MaxInstances doldu | Kullanılmayan sunucu tanımını silin veya lisansı yükseltin. |
| Bu modül lisans kapsamında değil | Kapalı bir modülün işlemi çağrıldı | Modülü lisansınıza ekletin. |

## Bölüm 4. Roller, Yetkiler ve Hiyerarşi

### 4.1 Roller ve sayısal değerleri

Sistemde sekiz rol vardır. Rol kimlikleri beşer artar.

| Kimlik (Id) | Rol adı | Hiyerarşi seviyesi | Kısa açıklama |
|---|---|---|---|
| 0 | NoAccess | 0 | Erişimi yok. Hesap açık ama hiçbir şey yapamaz. |
| 5 | Viewer | 5 | Yalnızca kontrol panelini görür. |
| 10 | Auditor | 10 | Denetçi. İzler, kanıt üretir ve doğrular. Değişiklik akışına giremez. |
| 15 | Requester | 15 | Talep açar. |
| 20 | RequestManager | 20 | Talep açar ve onaylar. |
| 25 | DbAdmin | 25 | Veritabanı yöneticisi. Çalıştırır ve tüm ayarları yönetir. |
| 30 | ChangeManager | 30 | Değişiklik yöneticisi. DbAdmin ile aynı yetkilere sahiptir, hiyerarşide üsttedir. |
| 35 | SystemManager | 35 | Sistem yöneticisi. En üst seviye. Rol tanımı düzenlenemez. |

Neden beşer artıyor: İleride iki rolün arasına yeni bir rol gerekirse aradaki boş sayılardan biri seçilir ve hiçbir kullanıcı kaydı kaydırılmaz. Ardışık numaralarda (4 ile 5 arasında) bu mümkün olmaz.

### 4.2 Kimlik ile hiyerarşi ayrı kavramlardır

Bu kılavuzun teknik ekiplerce en dikkatli okunması gereken maddesidir.

Her rolün iki sayısı vardır:

- Kimlik (Id): Kullanıcı kayıtlarında saklanan numara.

- Seviye (Level): Hiyerarşide nerede durduğunu belirten numara.

Bu ikisi bugün aynı değerlerdedir, ancak aynı kavram değildir. Bir rol kimlik olarak listenin sonuna eklenip hiyerarşide ortaya yerleştirilebilir. Denetçi rolü tam olarak bu sebeple ayrı yönetilir: izleme yetkisi Viewer'dan geniştir, ancak değişiklik akışında hiçbir yetkisi yoktur.

Kural: Rol karşılaştırması hiçbir zaman enum sayısına doğrudan bakılarak yapılmaz. Karşılaştırma her zaman seviye üzerinden yapılır. Bu kural bozulursa yeni eklenen bir rol sessizce yetki kazanır.

Tanımsız bir rol değeri gelirse seviye 0 kabul edilir. Yani tanımsız rol yetki kazanmaz.

### 4.3 Denetçi rolünün özel durumu

Denetçi onay akışında onaylayan olarak seçilemez. Denetleyen kişi onaylayan konumuna geçerse görevler ayrılığı bozulur, çünkü kendi onayladığı işi denetlemiş olur.

Denetçinin yapabildikleri: raporları görmek, kontrol panelini görmek, denetim kayıtlarını görmek, tüm talepleri görmek, bildirim kayıtlarını görmek, denetim dosyası dışa aktarmak.

Denetçinin yapamadıkları: talep açmak, onaylamak, çalıştırmak, sandbox çalıştırmak, herhangi bir ayarı değiştirmek, sürüm paketi yönetmek.

### 4.4 Rol yetki matrisi

Aşağıdaki tablo varsayılan rol tanımlarını gösterir. Roller ekranından değiştirilebilir; SystemManager rolü düzenlenemez.

| Yetki | NoAccess | Viewer | Auditor | Requester | RequestManager | DbAdmin | ChangeManager | SystemManager |
|---|---|---|---|---|---|---|---|---|
| Talep ekleme | - | - | - | Evet | Evet | Evet | Evet | Evet |
| Çalıştırma | - | - | - | - | - | Evet | Evet | Evet |
| Onaylama | - | - | - | - | Evet | Evet | Evet | Evet |
| Sandbox | - | - | - | - | - | Evet | Evet | Evet |
| Rapor görme | - | - | Evet | - | Evet | Evet | Evet | Evet |
| Kontrol paneli | - | Evet | Evet | Evet | Evet | Evet | Evet | Evet |
| Denetim kaydı görme | - | - | Evet | - | - | Evet | Evet | Evet |
| Sürüm paketi yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Tüm talepleri görme | - | - | Evet | Evet | Evet | Evet | Evet | Evet |
| Sunucu yönetimi | - | - | - | - | - | Evet | Evet | Evet |

| Yetki | NoAccess | Viewer | Auditor | Requester | RequestManager | DbAdmin | ChangeManager | SystemManager |
|---|---|---|---|---|---|---|---|---|
| Kullanıcı yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Rol yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Parametre yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Onay politikası yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| SQL standardı yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Kritik nesne yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| ITSM bilet yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Hassas kolon yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Hassas desen yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| E-posta kaynağı yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Bakım yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Bildirim kayıtları görme | - | - | Evet | - | - | Evet | Evet | Evet |
| Lisans yönetimi | - | - | - | - | - | Evet | Evet | Evet |
| Tanılama kullanımı | - | - | - | - | - | Evet | Evet | Evet |

| Yetki | NoAccess | Viewer | Auditor | Requester | RequestManager | DbAdmin | ChangeManager | SystemManager |
|---|---|---|---|---|---|---|---|---|
| Denetim dosyası dışa aktarma | - | - | Evet | - | - | Evet | Evet | Evet |

Talep ekleme yetkisi olan roller aynı zamanda sorgu sonucu talebi de açabilir.

### 4.5 Yetki anahtarları

Her yetki bir anahtar ile temsil edilir ve giriş jetonuna yazılır. Ekranların ve düğmelerin görünürlüğü bu anahtarlara bakar.

| Anahtar | Açtığı işlem |
|---|---|
| cr.add | Talep ve sorgu sonucu talebi açma |
| qr.add | Sorgu sonucu talebi açma |
| cr.execute | Talep çalıştırma |
| cr.sandbox | Sandbox çalıştırma |
| cr.approve | Onaylama, reddetme, zamanlama, manuel işaretleme |
| cr.seeall | Başkalarının taleplerini görme |
| report.view | Raporlar ekranı |
| dashboard.view | Kontrol paneli |
| audit.view | Denetim kaydı ekranı |
| audit.evidenceexport | Denetim dosyası dışa aktarma |
| rp.manage | Sürüm paketi yönetimi |
| settings.servers ve diğer settings.* anahtarları | İlgili ayar sekmesi |

### 4.6 Arayüz gizlemesi güvenlik değildir

Bir düğmenin gizlenmesi kullanıcının o işlemi yapmasını engellemez. Web arayüzü tarayıcıda çalışır ve kullanıcı isterse doğrudan API'ye istek gönderebilir.

Bu yüzden her işlem iki kere kontrol edilir:

1. Arayüz, düğmeyi gösterip göstermeyeceğine karar verir.

2. API, isteği aldığında aynı kuralı bağımsız olarak tekrar uygular.

İkinci kontrol asıl güvenliktir. Birinci kontrol yalnızca kullanıcı deneyimi içindir.

### 4.7 Kullanıcı yaşam döngüsü

| **İşlem** | **Davranış** |
|---|---|
| Kullanıcı ekleme | Kullanıcı adı ve e-posta benzersiz olmalıdır. |
| Kullanıcı silme | Kayıt fiziksel olarak silinmez, silindi olarak işaretlenir (yumuşak silme). Geçmiş taleplerdeki adı korunur. |
| Silinmiş kullanıcı ile aynı adda ekleme | Yeni kayıt oluşturulmaz, eski kayıt canlandırılır. Böylece aynı kişi aynı geçmişe geri döner. |
| Son sistem yöneticisini silme | Engellenir. Sistemde en az bir sistem yöneticisi kalmalıdır. |
| Kendini silme | Engellenir. |
| Dört göz kilitlenme kapısı | Ayar onayı açıkken, bir ekranın yetkisine sahip son ikinci kişiyi silmek engellenir. Aksi halde o ekrandaki değişiklikler hiç onaylanamaz hale gelirdi. |
| Hesap kilidi | Ardışık hatalı giriş sonrası hesap geçici kilitlenir. Yönetici elle de açabilir. |

### Yetki değişikliği ne zaman etkili olur

Bir kullanıcının rolü değiştiğinde, hesabı kapatıldığında ya da rolün yetki matrisi güncellendiğinde değişiklik beklemeden etkili olur. Bunu iki mekanizma birlikte sağlar.

1. Yüksek etkili işlemlerde yetki veritabanından okunur. Aşağıdaki işlemlerde yetki, kullanıcının elindeki oturum anahtarından değil doğrudan veritabanından doğrulanır:

- Talebi çalıştırma

- Çalıştırmayı zamanlama

- Talep onaylama

- Ayar değişikliğini onaylama ve reddetme

Yetkisi çekilmiş ya da hesabı kapatılmış bir kullanıcı, oturumu hâlâ açık görünse bile bu işlemleri yapamaz.

2. Oturum yenileme kapatılır. Rol değiştiğinde, hesap kapatıldığında veya rolün yetki matrisi güncellendiğinde etkilenen kullanıcıların oturum yenileme kayıtları iptal edilir. Elindeki oturum anahtarının süresi dolduğunda yenileme yapılamaz ve kullanıcı yeniden giriş yapmak zorunda kalır.

İptal denetim izine de yazılır: kullanıcı bazlı iptal User.SessionsRevoked, rol bazlı iptal Role.SessionsRevoked olayıdır. Kayıt, iptal sebebini ve kaç oturumun kapatıldığını taşır.

Yeni verilen bir yetkinin görünmesi için kullanıcının yeniden giriş yapması gerekir. Bu bilinçli bir tercihtir: yetki çekilmesi anında, yetki verilmesi ise oturum yenilendiğinde geçerli olur.

### 4.8 LDAP kullanıcıları ile yerel kullanıcıların farkı

| Konu | Yerel kullanıcı | LDAP kullanıcısı |
|---|---|---|
| Şifre nerede | SQL Change Guard veritabanında, şifrelenmiş özet olarak. | Kurum dizininde. Uygulama şifreyi saklamaz. |
| Şifre değiştirme | Uygulama içinden. | Kurum dizininden. |
| Rol | Uygulama içinden atanır. | Uygulama içinden atanır. Dizindeki grup bilgisi rolü belirlemez. |
| Hesap kilidi | Uygulama yönetir. | Dizin de kilitleyebilir; uygulama kilidi ayrıdır. |

### 4.9 Şifre politikası ve BCrypt

#### Hash nedir

Bir metni geri döndürülemez biçimde sabit uzunlukta bir değere çeviren işleme özet (hash) denir. Şifreler veritabanında düz metin olarak tutulmaz, özetleri tutulur. Kullanıcı giriş yaptığında girdiği şifrenin özeti hesaplanır ve kayıttaki özetle karşılaştırılır.

Veritabanını ele geçiren biri özetleri görür, şifreleri göremez.

#### Salt nedir

Aynı şifreyi kullanan iki kullanıcının özeti aynı çıkarsa, saldırgan bir tanesini çözdüğünde diğerini de çözer. Bunu engellemek için her şifreye rastgele bir ek değer (salt) karıştırılır. Böylece aynı şifre farklı kullanıcılarda farklı özet üretir.

#### Neden BCrypt seçildi

SQL Change Guard kullanıcı şifreleri için BCrypt kullanır.

| Özellik | Düz SHA-256 | BCrypt |
|---|---|---|
| Hız | Çok hızlı. Saniyede milyarlarca deneme yapılabilir. | Kasıtlı olarak yavaş. |
| Salt | Ayrıca eklemek gerekir. | Kendi içinde üretir ve saklar. |
| Ayarlanabilir maliyet | Yok. | Var (iş faktörü). |
| Şifre saklamaya uygunluk | Uygun değil. | Uygun. |

İş faktörü nedir: BCrypt'in kaç tur döneceğini belirleyen sayıdır. Bir artırmak süreyi iki katına çıkarır. Donanım hızlandıkça bu sayı artırılarak aynı koruma seviyesi korunur.

SHA-256'nın hızlı olması normal şartlarda iyi bir özelliktir, ama şifre saklamada zayıflıktır: saldırgan da o hızda deneme yapabilir. BCrypt kasıtlı olarak yavaştır; kullanıcı girişinde fark edilmeyen bir gecikme, milyarlarca deneme yapmaya çalışan saldırgan için aşılamaz bir maliyete dönüşür.

SHA-256 üründe başka bir amaçla kullanılır: denetim kaydı zinciri. Orada amaç şifre saklamak değil, içeriğin değişmediğini kanıtlamaktır ve hız bir zayıflık değildir. Ayrıntı: Bölüm 5.4.

### 4.10 Kullanıcılar ekranı

Ekran: Ayarlar > Kullanıcılar

Yetki: settings.users

| Alan | Ne işe yarar | Zorunlu | Notlar |
|---|---|---|---|
| Domain Kullanıcı Adı | Kullanıcının giriş adı. Talep, onay ve denetim kayıtlarında bu ad görünür. | Evet | Benzersizdir. Silinmiş bir kullanıcıyla aynı ad girilirse eski kayıt canlandırılır. |
| E-posta | Bildirimlerin gideceği adres. | Evet | Benzersizdir. Sorgu sonucu e-posta onayında da kullanılır. |
| Rol | Kullanıcının rolü. | Evet | Bölüm 4.1. |
| Yönetici | Kullanıcının doğrudan yöneticisi. | Hayır | Onay adımının rolünü karşılıyorsa adım bu kişiye atanır ve bildirim önce ona gider. |
| Kimlik Doğrulama Tipi | Yerel hesap mı, dizin hesabı mı. | Evet | Bölüm 4.8. |
| Şifre / Yeni Şifre | Yerel hesaplarda şifre. | Yerel hesapta evet | Dizin hesaplarında görünmez. Şifre politikası Bölüm 2.4.3'teki ayarlardan gelir. |
| Kilitli | Hesabın kilitli olup olmadığı. | - | Hatalı giriş denemeleri sonucu otomatik oluşur. |

Ekrandaki işlemler:

| İşlem | Ne yapar |
|---|---|
| Kilidi aç | Kilitlenmiş hesabı süre dolmadan açar. |
| Şifre sıfırla | Yerel hesaba yeni şifre atar. |

| İşlem | Ne yapar |
|---|---|
| CSV içe aktar | Toplu kullanıcı ekler. Dosyadaki her satır tek tek doğrulanır; hatalı satırlar sonuç özetinde bildirilir. |
| Sil | Yumuşak silme yapar. Bölüm 4.7'deki koruma kuralları uygulanır. |

Dört göz kuralı açıkken bu ekrandaki değişiklikler de onaya düşer. Bölüm 8.

### 4.11 Roller ekranı

Ekran: Ayarlar > Roller

Yetki: settings.roles

Her rol için Bölüm 4.4'teki yetkiler tek tek açılıp kapatılabilir. Üç kural vardır:

1. SystemManager rolü düzenlenemez. Aksi halde biri en üst rolün yetkisini kısıtlayıp sistemi yönetilemez hale getirebilirdi.

2. Denetçi rolüne onaylama veya çalıştırma yetkisi verilemez. Görevler ayrılığı gereğidir.

3. Rolün hiyerarşi seviyesi ekrandan değiştirilmez. Seviye ürün tanımının parçasıdır.

## Bölüm 5. Güvenlik ve Şifreleme Modeli

### 5.1 Üç farklı ihtiyaç, üç farklı yöntem

| İhtiyaç | Kullanılan yöntem | Neden |
|---|---|---|
| Kullanıcı şifresi saklamak | BCrypt | Geri döndürülmemeli, kırılması pahalı olmalı. |
| Sunucu şifresi ve sonuç paketi parolası saklamak | AES-256-GCM | Uygulamanın bu değerleri geri okuyabilmesi gerekir. |
| Denetim kaydının değişmediğini kanıtlamak | SHA-256 ve HMAC-SHA256 | Geri okumak gerekmez, yalnızca doğrulamak gerekir. |

Temel ayrım şudur: bir değeri geri okumanız gerekiyorsa şifreleme, yalnızca doğrulamanız gerekiyorsa özet kullanılır.

### 5.2 Sunucu şifreleri: AES-256-GCM

#### Simetrik şifreleme nedir

Aynı anahtarla hem şifreleyip hem çözebildiğiniz yönteme simetrik şifreleme denir. AES bunun en yaygın örneğidir. 256, anahtar uzunluğunun bit cinsinden değeridir.

#### GCM neden seçildi

AES'in birden çok çalışma kipi vardır. GCM kipi şifrelemeye ek olarak bir doğrulama etiketi üretir. Şifreli metin sonradan değiştirilirse çözme işlemi hata verir, sessizce bozuk veri dönmez.

Klasik kiplerde (örneğin CBC) saldırgan şifreli metni bozabilir ve uygulama bunu fark etmeden anlamsız bir sonuç üretebilir. GCM bu riski ortadan kaldırır.

#### Nerede kullanılır

| Veri | Nerede tutulur | Anahtar |
|---|---|---|
| Sunucu tanımındaki veritabanı şifresi | Servers.SqlPassword sütunu | Security:ServerPasswordKey |
| Sorgu sonucu ZIP parolası | ChangeRequests.QrZipPassword sütunu | Security:QrZipPasswordKey |

Anahtarlar veritabanında değil, yapılandırmada durur. Böylece yalnızca veritabanına erişen biri bu değerleri çözemez.

Anahtar değişirse: Eski değerler çözülemez. Sunucu şifrelerinin yeniden girilmesi gerekir. Bu yüzden anahtar değişimi planlı yapılmalıdır.

## 5.3 Neyin nerede korunduğu

| Veri | Durum |
|---|---|
| Kullanıcı şifresi | BCrypt özeti olarak saklanır. Geri okunamaz. |
| Sunucu veritabanı şifresi | AES-256-GCM ile şifreli saklanır. Arayüzde hiçbir zaman gösterilmez. |
| Sorgu sonucu ZIP parolası | AES-256-GCM ile şifreli saklanır. Yalnızca yetkili kullanıcıya gösterilir. |
| Sorgu sonucu verisi | Şifreli ZIP içinde diskte tutulur, saklama süresi dolunca silinir. |
| Denetim kaydı | Açık metin olarak tutulur, ancak zincirle imzalanır. |
| Günlük dosyaları | Şifre ve parola alanları maskelenir. |

## 5.4 Denetim kaydı: SHA-256 ve HMAC

### SHA-256 nedir

Bir metinden 64 karakterlik sabit uzunlukta bir özet üreten algoritmadır. Aynı metin her zaman aynı özeti verir; metnin tek bir karakteri değişse özet tümüyle değişir. Özetten metni geri üretmek pratikte mümkün değildir.

### Hash zinciri neden değiştirilemezlik sağlar

Her denetim kaydının özeti hesaplanırken bir önceki kaydın özeti de hesaba katılır. Kayıtlar böylece birbirine bağlanır:

```
Kayit 1: icerik1 + "GENESIS"  --> ozet1
Kayit 2: icerik2 + ozet1      --> ozet2
Kayit 3: icerik3 + ozet2      --> ozet3
Kayit 4: icerik4 + ozet3      --> ozet4
```

Biri 2 numaralı kaydı değiştirirse ozet2 artık tutmaz. ozet2 değişince ozet3 de tutmaz, ozet3 değişince ozet4 de tutmaz. Yani tek bir kaydı gizlice düzeltmek için ondan sonraki bütün kayıtları yeniden hesaplamak gerekir.

### HMAC ile düz özetin farkı

Düz SHA-256'da formül herkese açıktır. Veritabanına yazma yetkisi olan biri kayıtları değiştirip zinciri baştan sona yeniden hesaplayabilir, çünkü hesaplamak için gizli bir bilgiye ihtiyacı yoktur.

HMAC-SHA256'da hesaba gizli bir anahtar girer. Anahtar veritabanında değildir, yapılandırmadan gelir. Anahtarı bilmeyen biri geçerli bir özet üretemez; zinciri yeniden hesaplayamaz.

### Sürümler

| Sürüm | Algoritma | Kapsam |
|---|---|---|
| 1 | SHA-256 | Eski değer ve IP adresi kapsam dışı. Yalnızca eski kayıtlarda görülür. |
| 2 | SHA-256 | Eski değer ve IP adresi dahil. HMAC anahtarı tanımlı değilse bu sürüm yazılır. |
| 3 | HMAC-SHA256 | Sürüm 2 ile aynı kapsam, ek olarak gizli anahtar kullanılır. |

Anahtar tanımlıysa yeni kayıtlar sürüm 3 olarak yazılır. Her kayıt hangi anahtarla imzalandığını da tutar (KeyId), böylece anahtar değiştirildiğinde eski kayıtlar hâlâ doğrulanabilir.

Anahtar halkasında karşılığı bulunmayan bir sürüm 3 kaydı "doğrulanamadı" sayılır. Güvenli taraf her zaman budur.

## 5.5 Veritabanı seviyesi yazma kaydı

Denetim kaydının yanında ikinci bir iz daha tutulur: TableWriteLogs. İzlenen her tabloda ekleme, güncelleme ve silme olduğunda satırın eski ve yeni hali buraya yazılır. Kaydı veritabanı tetikleyicileri yazar.

İkisi ayrı işlere bakar:

| | Denetim kaydı (AuditLogs) | Yazma kaydı (TableWriteLogs) |
|---|---|---|
| Ne saklar | Uygulamadan geçen yönetişim olayları | Veritabanındaki ham yazma hareketi |
| Kim yazar | Uygulama | Veritabanı tetikleyicisi |
| Uygulamayı atlayan değişikliği görür mü | Hayır | Evet |
| Mühür | HMAC, anahtar veritabanının dışında | Yok, aşağıya bakınız |

Yazma kaydında değişikliği yapan veritabanı girişi de saklanır. Biri doğrudan SQL ile kayıt üzerinde değişiklik yaparsa, hangi girişin yaptığı buradan görülür.

### Yazma kaydında neden mühür yok

Denetim kaydı HMAC ile mühürlüdür ve anahtar veritabanının dışında durur. Bu yüzden yalnız veritabanı erişimi olan biri onu yeniden hesaplayamaz. Denetim kanıtı orada durur.

Yazma kaydı için aynı yöntem uygulanabilir değildir: mühür veritabanının içinde hesaplanmak zorunda kalır ve anahtar da orada bulunur. Bu durumda veritabanına yazma yetkisi olan biri, yani mührün

yakalaması gereken kişi, kaydı silip mührü yeniden hesaplayabilir. Koruma sağlamayan bir mühür yerine iki gerçek kontrol kullanılır.

### Kurulumda yapılması gereken

1. Yazma kaydına yalnız ekleme yetkisi verin. Uygulama hesabının bu tabloyu değiştirmesi veya silmesi gerekmez:

```
GRANT INSERT ON dbo.TableWriteLogs TO [uygulama_hesabi];

DENY UPDATE, DELETE ON dbo.TableWriteLogs TO [uygulama_hesabi];
```

Bu adım kuruluma bırakılmıştır, çünkü hesap adı kuruma göre değişir. Uygulandığında, uygulama üzerinden yazma kaydını bozmak imkansız hale gelir.

2. Günlük çapayı açık tutun. Worker her gün yazma kaydının kapsam özetini (satır sayısı ve numara aralığı) ayrı bir veritabanına mühürler. Toplu silme bu sayede ana veritabanının dışından tespit edilir. Bu, worker çalıştığı sürece kendiliğinden olur.

Geriye kalan tek durum, veritabanında değiştirme yetkisi olan birinin tek bir satırı yerinde düzenlemesidir. Bunu veritabanı içinde hiçbir yöntem engelleyemez; çözümü birinci maddedeki yetki ayarıdır.

## 5.6 Sır maskeleme

Denetim kaydına ve günlüklere yazılan bilgilerde şifre, parola ve bağlantı metni gibi alanlar maskelenir. Amaç, denetim izini okuma yetkisi olan bir kişinin bu izden gizli bilgi toplayamamasıdır.

## Bölüm 6. Sunucu Tanımları

Ekran: Ayarlar > Sunucular

Yetki: settings.servers

### 6.1 Bu ekran ne işe yarar

Talep ve sorguların çalıştırılacağı hedef veritabanı sunucuları burada tanımlanır. Bir sunucu tanımı yoksa hiçbir talep açılamaz, çünkü talep en az bir hedef sunucu seçmek zorundadır.

### 6.2 Alan alan açıklama

| Alan | Ne işe yarar | Zorunlu | Geçerli değerler | Varsayılan | Boş bırakılırsa | Değiştirince ne olur |
|---|---|---|---|---|---|---|
| Sunucu Adı | Sistem içinde bu sunucuyu tanımlayan ad. Taleplerde ve raporlarda bu ad görünür. | Evet | Serbest metin | - | Kaydedilmez, hata verir | Eski taleplerdeki sunucu adı metin olarak saklandığı için geçmiş kayıtlar eski adı gösterir. |
| Host | Sunucunun ağ adresi veya adı. | Evet | Ad veya IP | - | Kaydedilmez | Yeni bağlantılar yeni adrese gider. |
| Port | Bağlantı portu. Veritabanı tipini değiştirdiğinizde otomatik güncellenir. | Hayır | Sayı | SQL Server 1433, Oracle 1521, PostgreSQL 5432 | Varsayılan kullanılır | |
| Veritabanı Tipi | Hedefin türü. Liste lisansa göre daralır. | Evet | SQL Server, Oracle, PostgreSQL | SQL Server | - | Betik ayrıştırma ve çalıştırma davranışı tümüyle değişir. |
| Veritabanı Adı | Bağlanılacak veritabanı. PostgreSQL için gereklidir. Oracle için servis adıdır ve genelde host içinde verilir. | Türe bağlı | Serbest metin | Boş | SQL Server'da varsayılan veritabanı kullanılır | |
| Ortam | Sunucunun hangi ortama | Evet | Production, Staging, | Production | - | Politika eşleşmesi |

| Alan | Ne işe yarar | Zorunlu | Geçerli değerler | Varsayılan | Boş bırakılırsa | Değiştirince ne olur |
|---|---|---|---|---|---|---|
| | ait olduğu. Onay politikası seçimini doğrudan etkiler. | | Development, Test | | | değişir, yeni talepler farklı onay adımları alabilir. |
| SQL Kullanıcı Adı | Hedef sunucuya bağlanılacak hesap. | Evet | Serbest metin | - | Bağlantı kurulamaz | |
| SQL Şifre | Hesabın şifresi. Şifreli saklanır, hiçbir ekranda geri gösterilmez. | Yeni kayıtta evet | Serbest metin | - | Düzenlemede boş bırakılırsa mevcut şifre korunur | |
| Aktif | Sunucunun kullanımda olup olmadığı. | - | Açık / Kapalı | Açık | - | Kapatılırsa yeni taleplerde seçilemez. |
| CR Otomatik Çalıştırma | Onaylanan değişiklik talepleri bu sunucuda otomatik çalışsın mı. Zamanlama modülü gerektirir. | - | Açık / Kapalı | Kapalı | - | Bkz. 6.4 |
| QR Otomatik Çalıştırma | Onaylanan sorgu sonucu talepleri bu sunucuda otomatik çalışsın mı. | - | Açık / Kapalı | Kapalı | - | Bkz. 6.4 |
| Rollback Zorunlu | Bu sunucuda çalıştırma yapılabilmesi için geri alma betiğinin oluşturulmuş olması şartı. | - | Açık / Kapalı | Kapalı | - | Açıldığında geri alma betiği olmayan talepler çalıştırılamaz. |
| Otomatik Rollback | Talep onaylandığında geri alma betiği | - | Açık / Kapalı | Kapalı | - | |

| Alan | Ne işe yarar | Zorunlu | Geçerli değerler | Varsayılan | Boş bırakılırsa | Değiştirince ne olur |
|---|---|---|---|---|---|---|
| | otomatik üretilsin mi. | | | | | |
| SSL Zorunlu | Yalnızca PostgreSQL için görünür. Bulut PostgreSQL servislerinde gereklidir. | - | Açık / Kapalı | Kapalı | - | |
| Maskeleme | Bu sunucudan dönen sorgu sonuçlarında hassas veri tespitinin nasıl yapılacağı. | Evet | Tam (katalog + desen), Yalnız içerik | Tam | - | Bkz. 6.5 |
| Muafiyet gerekçesi | Maskeleme "Tam" dışında seçildiyse yazılı gerekçe. | Tam dışında evet | Serbest metin | Boş | Kaydedilmez, hata verir | Denetim kaydına yazılır. |

### 6.3 Bağlantı testi

Sunucu düzenleme ekranında bağlantı testi düğmesi vardır. Test, girilen bilgilerle sunucuya bağlanmayı dener ve sonucu hemen gösterir.

Test sonuçlarının anlamı:

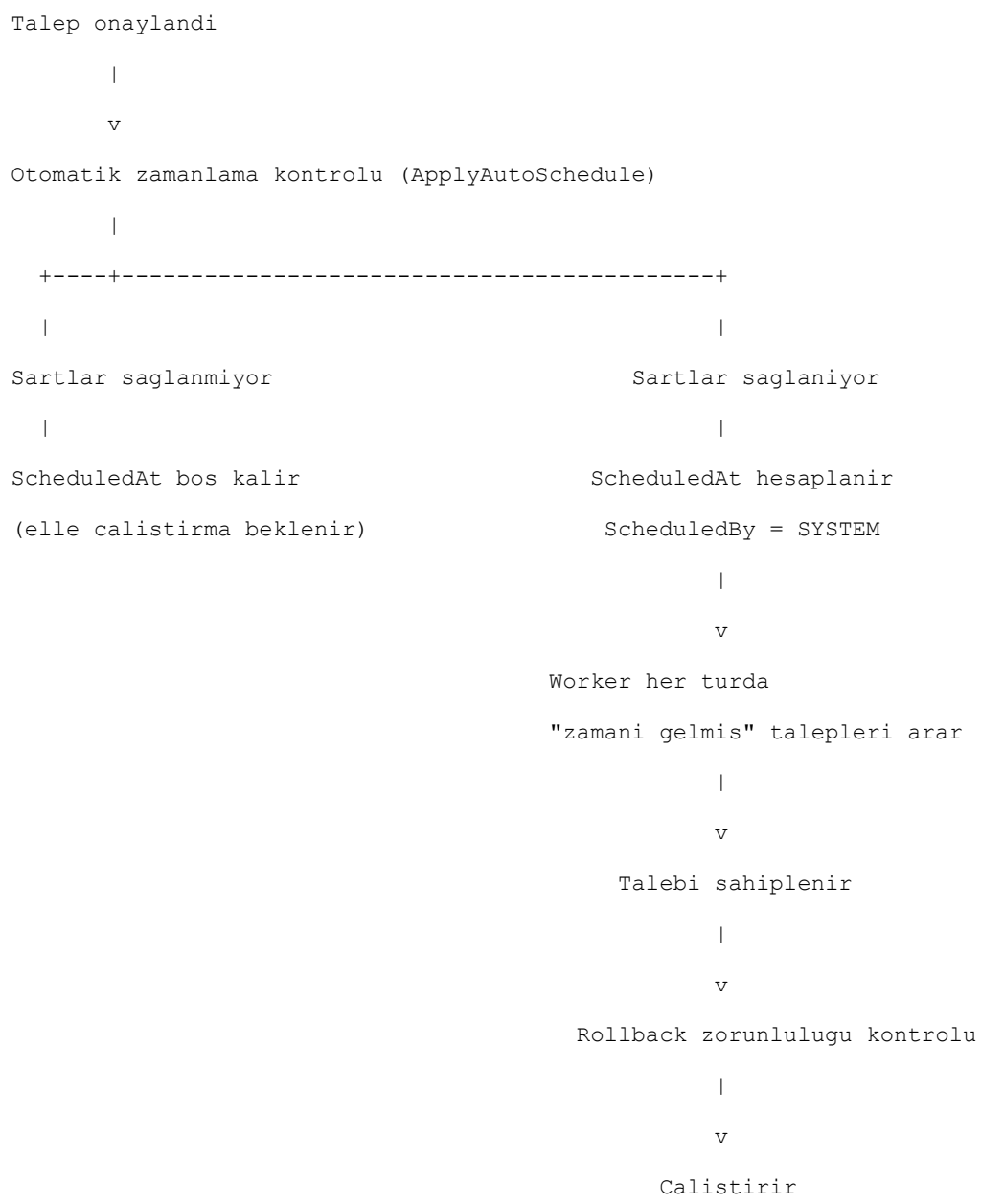
| Sonuç | Anlamı | Ne yapmalı |
|---|---|---|
| Başarılı | Bağlantı kuruldu. | - |
| Kimlik doğrulama hatası | Kullanıcı adı veya şifre yanlış. | Bilgileri kontrol edin. |
| Sunucuya ulaşılamıyor | Ağ, host adı veya port hatalı ya da güvenlik duvarı engelliyor. | Ağ erişimini kontrol edin. |
| Veritabanı bulunamadı | Veritabanı adı yanlış. | Adı kontrol edin. |
| Bu veritabanı tipi lisans kapsamında değil | Lisans bu türe izin vermiyor. | Bölüm 3.5. |

Worker servisi ayrıca ConnectionCheckJob ile düzenli aralıklarla tüm aktif sunuculara bağlanmayı dener ve durumu günceller. Bu sayede bir sunucu erişilemez hale geldiğinde talep çalıştırılmadan önce fark edilir.

### 6.4 Otomatik çalıştırma: arka planda ne oluyor

Bu, kılavuzun en çok soru alan konularından biridir. Adım adım anlatalım.

#### Zincir




#### Şartlar

Otomatik zamanlama yalnızca aşağıdakilerin hepsi doğruysa yapılır:

1. Zamanlama modülü lisanslı.

2. Talebin risk bulgularından hiçbiri "asla atlanamaz" (NeverSkip) işaretli değil.

3. Talep bir rollback talebi değil.

4. Talep aktif bir sürüm paketi içinde değil.

5. Talebin tüm hedef sunucularında ilgili otomatik çalıştırma bayrağı açık. Değişiklik talepleri için CR bayrağı, sorgu sonucu talepleri için QR bayrağı bakılır.

6. Değişiklik talebi ise ve sunuculardan herhangi birinde "Rollback Zorunlu" açıksa, geri alma betiği oluşturulmuş olmalı.

Şartlardan biri sağlanmazsa talep onaylı durumda bekler ve birinin elle çalıştırmasını bekler. Daha önce sistem tarafından konmuş bir zamanlama varsa temizlenir.

### Zaman nasıl hesaplanır

Sıralama şudur:

1. Kullanıcı talepte belirli bir çalıştırma tarihi verdiyse (RequestedExecutionDate) o kullanılır.

2. Vermediyse ve gecikme dakikası verdiyse (ScheduleDelayInMinutes), onay anı + o dakika kullanılır.

3. Hiçbiri yoksa onay anının kendisi kullanılır.

Sonra ExecutionDelaySeconds parametresi devreye girer: hesaplanan zaman, "şu an + ExecutionDelaySeconds" değerinden erken ise o değere çekilir.

Somut örnek: ExecutionDelaySeconds 300 (5 dakika). Bir talebi saat 14:00'te onayladınız ve talepte özel bir tarih yok.

- Hesaplanan zaman: 14:00

- En erken izin verilen zaman: 14:05

- Sonuç: çalıştırma zamanı 14:05 olur.

Worker 14:05'ten önce bu talebi almaz. Bu bekleme süresi bilinçlidir: yanlışlıkla verilen bir onaydan sonra talebi iptal etmek için insan zamanı bırakır.

Aynı parametre elle zamanlamada da alt sınır olarak çalışır, ancak orada en az 30 saniye uygulanır: ExecutionDelaySeconds 10 girilse bile elle zamanlamada en erken 30 saniye sonrası seçilebilir.

### Kayıt kime yazılır

Otomatik zamanlamada düğmeye basan bir kişi yoktur. Bu yüzden:

- ScheduledBy = SYSTEM

- ExecutionRequestedBy = SYSTEM

Alan boş bırakılsaydı raporu okuyan kişi "yazılmayı mı unuttular" diye düşünürdü. Kararın sahibi açıkça SYSTEM olarak yazılır.

### 6.5 Maskeleme rejimi

| Rejim | Ne yapar | Ne zaman kullanılır |
|---|---|---|
| Tam (Strict) | Hem sınıflandırma kataloğuna (kolon adı) hem veri desenlerine bakar. | Gerçek veri barındıran her ortam. Varsayılan budur. |
| Yalnız içerik (ContentOnly) | Kolon adına bakmaz, yalnızca değerlere bakar. | Sentetik (üretilmiş) veri barındıran test ortamları. |

Sunucu seviyesinde "kapalı" seçeneği bilinçli olarak yoktur. Maskelemeyi tümüyle kapatmak yalnızca genel QrMaskingEnabled parametresi ile mümkündür ve bu bir kontrol değişikliği sayılır.

Yalnız içerik rejimi kendini doğrular: Bu rejimde kolon adına bakılmaz, ama değer bazlı kontrol çalışmaya devam eder. Sunucuya gerçek veri yüklenirse kimlik numarası, kart numarası ve IBAN gibi doğrulanabilir alanlar yine yakalanır ve maskelenir. Yani "test ortamı" diye açılan muafiyet, ortama gerçek veri girdiği anda kendiliğinden dar bir korumaya döner.

Bu rejimde açık kalan alanlar: ad, adres, tanı gibi belirli bir deseni olmayan bilgiler. Ekranda bu uyarı açıkça gösterilir.

Rejim "Tam" dışında seçildiğinde yazılı gerekçe zorunludur ve teslim anında kayda dondurulur. Sunucu sonradan "Tam" moduna çevrilse bile o teslimin hangi rejimle yapıldığı kanıtta kalır.

### 6.6 Çoklu sunucu kuralı

Bir talep birden çok sunucuya gidebilir. Sunucu adları tek alanda, noktalı virgül ile ayrılarak saklanır:

```
PROD-SQL01;PROD-SQL02;PROD-SQL03
```

Bunun sonuçları:

- Talep listelerinde ve raporlarda sunucu sütunu birden çok ad içerebilir.

- Onay politikası çözümlemesi her sunucu için ayrı yapılır ve en katı sonuç uygulanır.

- Otomatik çalıştırma için tüm sunucularda bayrağın açık olması gerekir.

- Sorgu sonucu taleplerinde her sunucu kendi maskeleme rejimiyle değerlendirilir; canlı sunucu maskelenirken test sunucusu açık kalabilir.

- Sunucu adına göre filtre yapan raporlarda tam eşleşme değil, liste içinde arama yapılır.

## 6.7 Sunucu silme ve pasife alma

| **İşlem** | **Etkisi** |
|---|---|
| Pasife alma | Yeni taleplerde seçilemez. Mevcut talepler etkilenmez. Lisans sayımında yer kaplamaya devam eder. |
| Silme | Kayıt tablodan çıkar, lisans sayımında yer açılır. Geçmiş taleplerde sunucu adı metin olarak saklandığı için kayıtlar bozulmaz. |

## Bölüm 7. Parametreler

Ekran: Ayarlar > Parametreler

Yetki: settings.params

### 7.1 Bu ekran ne işe yarar

Sistem davranışını belirleyen tekil ayarlar burada tutulur. Her parametrenin bir anahtarı, bir değeri ve bir grubu vardır. Ekranda parametreler gruplara göre listelenir.

### 7.2 Ekrandaki alanlar

| **Alan** | **Anlamı** |
|---|---|
| Anahtar (Key) | Parametrenin sistem içindeki adı. Değiştirilmez. |
| Değer (Value) | Etkin değer. |
| Açıklama | Parametrenin ne yaptığı. |
| Kontrol tipi | Değerin ekranda nasıl girileceği: Numeric (sayı), Boolean (açık/kapalı), Text (metin), Combo (listeden seçim). |
| Seçenekler | Combo tipindeki parametrenin geçerli değerleri. |
| Grup | Ekrandaki başlık. |
| Kontrol ayarı rozeti | Bu parametre bir kontrolün sertliğini doğrudan belirliyorsa gösterilir. Bkz. Bölüm 8. |

### 7.3 Genel grubu

#### DefaultQueryTimeout

- Ne işe yarar: Yeni açılan taleplerde sorgu zaman aşımı alanının başlangıç değeri ve alt sınırı.

- Değer: Saniye cinsinden sayı.

- Varsayılan: 30

- Nerede okunur: Talep editörü ekranı ve talep kaydetme komutu.

- Örnek: 30 iken yeni bir talep açtığınızda zaman aşımı alanı 30 gelir. Elle 10 yazarsanız kayıt sırasında 30'a yükseltilir.

- Boş veya 0 ise: 30 kabul edilir.

- Geçmişi etkiler mi: Hayır, yalnızca yeni taleplerde geçerlidir.

#### MaxQueryTimeout

- Ne işe yarar: Bir talebe verilebilecek en yüksek zaman aşımı.

- Değer: Saniye cinsinden sayı.

- Varsayılan: 3600 (1 saat)

- Nerede okunur: Talep kaydetme ve sürüm paketi kaydetme komutları.

- Örnek: 3600 iken kullanıcı 7200 girerse değer 3600'e düşürülür.

- Neden vardır: Çok uzun süren bir betik hedef sunucuda kilit tutar ve diğer işleri bekletir. Üst sınır bu riski sınırlar.

#### ExecutionDelaySeconds

- Ne işe yarar: Onaydan sonra çalıştırmaya kadar geçecek en az süre.

- Değer: Saniye cinsinden sayı.

- Varsayılan: 300 (5 dakika)

- Nerede okunur: Onay komutu, onay akışı servisi, elle zamanlama komutu.

- Örnek: Bölüm 6.4'te ayrıntılı anlatıldı. 300 iken 14:00'te onaylanan talep en erken 14:05'te çalışır.

- 0 ise: Bekleme uygulanmaz, otomatik çalıştırma onay anında devreye girer. Elle zamanlamada yine 30 saniye alt sınır geçerlidir.

- Neden vardır: Yanlış verilen onaydan dönmek için zaman bırakır.

#### RejectOnError

- Ne işe yarar: Çalıştırma hata verdiğinde talep otomatik reddedilsin mi.

- Değer: true / false

- Varsayılan: true

- Nerede okunur: Zamanlanmış çalıştırma servisi.

- Örnek: true iken gece çalışan bir talep hata alırsa durumu Rejected olur ve bir daha denenmez. false iken talep Approved durumuna geri döner ve tekrar denenebilir.

- Kontrol ayarı mı: Evet.

- Hangisini seçmeli: Sıkı kontrol isteyen kurumlarda true tercih edilir; başarısız bir işin sessizce tekrar denenmesi istenmez.

### 7.4 Görevler Ayrılığı grubu

Bu gruptaki parametreler kurum genelindeki varsayılanlardır. Bir onay politikası bunları sunucu veya ortam bazında değiştirebilir. Etkin değerler, onay akışı kurulurken talebe dondurulur; sonradan yapılan değişiklikler karara bağlanmış talepleri yeniden yazmaz.

### RequesterCanSelfApproveExecute

- Ne işe yarar: Talebi açan kişi kendi talebini onaylayıp çalıştırabilir mi.

- Değer: true / false

- Varsayılan: true

- Nerede okunur: Onaylama, çalıştırma, zamanlama ve manuel işaretleme kontrolleri.

- Örnek: false iken bir geliştirici kendi açtığı talebi onaylamak isterse "talep sahibi kendi talebini onaylayamaz" mesajını alır ve düğme pasif görünür.

- Kontrol ayarı mı: Evet.

### ApproverCanExecute

- Ne işe yarar: Bir talebi onaylayan kişi aynı talebi çalıştırabilir mi.

- Değer: true / false

- Varsayılan: true

- Nerede okunur: Çalıştırma ve zamanlama kontrolleri.

- Örnek: false iken talebi onaylayan yönetici çalıştır düğmesini pasif görür; başka bir yetkilinin çalıştırması gerekir.

- Neden vardır: Onaylayan ile uygulayanın ayrılması yaygın bir denetim beklentisidir.

- Kontrol ayarı mı: Evet.

### RequireDistinctApproverPerStep

- Ne işe yarar: Aynı kişi bir talepte birden çok onay adımını kapatabilir mi.

- Değer: true / false

- Varsayılan: false

- Örnek: İki adımlı bir politikada, false iken yeterli role sahip tek kişi iki adımı da kapatabilir. true iken ikinci adım için farklı bir kişi gerekir. Gerçek dört göz kuralı budur.

- Kontrol ayarı mı: Evet.

### EmergencyMinimumRole

- Ne işe yarar: Acil durum talebi açabilmek için gereken en düşük rol.

- Değer: Rol adı.

- Varsayılan: Requester

- Örnek: ChangeManager yapılırsa geliştiriciler acil durum talebi açamaz; yalnızca değişiklik yöneticisi ve üstü açabilir.

- Neden vardır: Acil durum ilan etmek bir yetkidir. Herkeste olursa normal onay yolu kolayca aşılır.

- Kontrol ayarı mı: Evet.

### 7.5 Ayar Onayı grubu

#### SettingsApprovalPolicy

- Ne işe yarar: Ayar ekranlarında dört göz kuralının kapsamı.

- Değerler:

  - Disabled: Değişiklikler anında uygulanır.

  - ControlSettingsOnly: Yalnızca bir kontrolü zayıflatan veya güçlendiren değişiklikler onay bekler.

  - All: Bu ekranlardaki her değişiklik onay bekler.

- Varsayılan: Disabled

- Kontrol ayarı mı: Evet. Kapatmak da onay ister; açmak hemen uygulanır, çünkü politika kapalıyken yakalama zaten çalışmaz.

- Ayrıntı: Bölüm 8.

### 7.6 ITSM Bilet grubu

#### ChangeTicketRequired

- Ne işe yarar: Talep açarken bilet numarası zorunlu mu.

- Değer: true / false

- Varsayılan: true

- Örnek: true iken bilet numarası girilmeden talep kaydedilemez.

- Kontrol ayarı mı: Evet.

#### ChangeTicketRegex

- Ne işe yarar: Bilet numarasının biçim kontrolü.

- Değer: Düzenli ifade (regular expression). Boş bırakılabilir.

- Varsayılan: ^CR-\d{5}$

- Örnek: Varsayılan değerde CR-12345 kabul edilir, 12345 ve CR-123 reddedilir.

- Boş bırakılırsa: Biçim kontrolü yapılmaz, herhangi bir metin kabul edilir.

- Kendi kurumunuz için: Bilet numaralarınız INC0001234 biçimindeyse ^INC\d{7}$ yazabilirsiniz.

### 7.7 Sorgu Sonucu grubu

#### QrFileRetentionDays

- Ne işe yarar: Sonuç ZIP dosyalarının diskte kaç gün tutulacağı.

- Değer: Gün sayısı.

- Varsayılan: 1

- Nerede uygulanır: Bakım görevi (QR Dosya Temizleme).

- Örnek: 1 iken bugün üretilen dosya yarın silinir. Silme işlemi denetim kaydına yazılır ve Veri Erişim raporunda görünür.

- Kontrol ayarı mı: Evet.

- Neden kısa tutulur: Canlı veri içeren dosyanın diskte kalma süresi ne kadar kısaysa risk o kadar azdır.

#### QrRequesterCanDownload

- Ne işe yarar: Talebi açan kişi kendi sonuç dosyasını indirebilir mi.

- Değer: true / false

- Varsayılan: true

- Örnek: false yapılırsa sorgu sonucunu isteyen kişi dosyayı indiremez; yalnızca aşağıdaki rol eşiğini geçen kişiler indirebilir. Bu, veriyi isteyen ile veriye erişen kişiyi ayırmak isteyen kurumlar içindir.

- Kontrol ayarı mı: Evet.

#### QrDownloadMinimumRole

- Ne işe yarar: Talebi açan kişi dışındakiler için indirme yetkisinin alt sınırı.

- Değer: Rol adı.

- Varsayılan: ChangeManager

- Örnek: ChangeManager iken bir DbAdmin başkasının sonuç dosyasını indiremez, çünkü hiyerarşide alttadır.

- Kontrol ayarı mı: Evet.

### 7.8 Hassas Veri grubu

#### QrMaskingEnabled

- Ne işe yarar: Maskelemenin ana anahtarı.

- Değer: true / false

- Varsayılan: true

- Kapatılırsa: Hiç tespit ve maskeleme yapılmaz. Ancak her teslim yine kayda yazılır ve uygulanan politika MaskingOff görünür. Denetçinin "maskeleme kapalıyken gönderilmiş sonuç" durumunu görebildiği tek yer burasıdır.

- Kontrol ayarı mı: Evet.

### QrSensitiveDataPolicy

- Ne işe yarar: Maskeleme açıkken davranış.

- Değerler:

  - AutoMask: Tespit edilen hassas kolonlar maskelenir. Onaylı istisnalar açık kalır.

  - WarnOnly: Hiçbir şey maskelenmez, tespit sonucu yalnızca kayda yazılır.

- Varsayılan: AutoMask

- Örnek: WarnOnly, yeni kurulumda desenlerin doğru çalışıp çalışmadığını görmek için kısa süreli kullanılır. Kalıcı kullanımı korumayı ortadan kaldırır.

- Bilinmeyen bir değer yazılırsa: Güvenli tarafta kalınır, maskeleme uygulanır.

- Kontrol ayarı mı: Evet.

### QrSensitiveDetectionSampleSize

- Ne işe yarar: Her kolondan kaç satır örnek alınıp desen taraması yapılacağı.

- Varsayılan: 1000

- Örnek: 1000 iken 500 bin satırlık bir sonuçta her kolondan 1000 hücre incelenir. Sayıyı büyütmek doğruluğu artırır, süreyi uzatır.

- Kontrol ayarı mı: Evet.

### QrSensitiveDetectionThresholdPct

- Ne işe yarar: Örneklenen hücrelerin yüzde kaçı eşleşirse kolon hassas sayılır.

- Değer: 0-100 arası sayı.

- Varsayılan: 20

- Örnek: 1000 satır örneklenip eşik 20 iken, 200 hücrede kimlik numarası deseni bulunursa kolon hassas kabul edilir ve maskelenir. 150 hücrede bulunursa kabul edilmez.

- Düşürülürse: Daha çok kolon maskelenir, yanlış pozitif artar.

- Yükseltilirse: Daha az kolon maskelenir, kaçırma riski artar.

- Kontrol ayarı mı: Evet.

### QrSensitiveDetectionSampleMode

- Ne işe yarar: Örnek satırların nasıl seçileceği.

- Değerler: Random, Sequential

- Varsayılan: Random

- Örnek: Sıralı veride ilk 1000 satır aynı bölgeye ait olabilir ve temsil etmeyebilir. Random tüm sonuca yayılmış seçim yapar.

- Kontrol ayarı mı: Evet.

### QrSensitiveDetectionTimeBudgetSeconds

- Ne işe yarar: Tüm tespit turunun toplam süre bütçesi.

- Varsayılan: 10

- Bütçe aşılırsa: Kalan kolonlar "taranmadı" olarak işaretlenir ve maskelenir. Güvenli taraf her zaman kazanır.

- 0 ise: Sınır yoktur.

- Kontrol ayarı mı: Evet.

### QrColumnPreviewEnabled

- Ne işe yarar: Sorgu sonucu talebi kaydedilirken hedef sunucudan kolon listesi alınsın mı.

- Varsayılan: true

- Ne yapar: Sorguyu çalıştırmaz. Yalnızca "bu sorgu hangi kolonları döndürür" diye sorar, tek satır veri okumaz.

- Neden önemlidir: Önizleme, takma ad (alias) kullanılan kolonların gerçek kaynağını öğrenir. SELECT TCKN AS Kod yazıldığında kolon adı Kod görünür ama kaynak kolon TCKN'dir. Bu bilgi olmadan maskeleme atlatılabilirdi.

- Sunucu cevap veremezse: Talep yine kaydedilir.

### UnmaskedColumnRequestPolicy

- Ne işe yarar: Kullanıcı belirli kolonların maskelenmeden gelmesini isteyebilir mi.

- Değerler:

  - RequireApproval: İstenen kolonlar listelenir, ayrı bir onay adımı eklenir. Onaylanan kolonlar maskelenmez.

  - AllowWithReason: Ek onay adımı yoktur, gerekçe yazılır ve otomatik onaylı sayılır.

  - Disabled: Özellik kapalıdır, editörde ilgili alanlar gizlenir ve varsa liste temizlenir.

- Varsayılan: RequireApproval

- Kontrol ayarı mı: Evet.

- Bağımlılık: QrMaskingEnabled kapalıysa bu ayar Disabled gibi davranır. Maskelenecek bir şey yokken maskesiz isteme onayı istemek, onay mekanizmasını lastik damgaya çevirirdi.

### UnmaskedColumnApproverRole

- Ne işe yarar: Maskesiz kolon isteğini kim onaylar.

- Varsayılan: SystemManager

- Kontrol ayarı mı: Evet.

## 7.9 E-posta Onayı grubu

### QrEmailApprovalPolicy

- Ne işe yarar: Sorgu sonucunun gönderileceği e-posta adresleri onaydan geçsin mi.

- Değerler: RequireApproval, Disabled

- Varsayılan: RequireApproval

- Örnek: RequireApproval iken daha önce onaylanmamış bir adrese sonuç göndermek istendiğinde talebe ek bir onay adımı eklenir. Amaç, canlı verinin kurum dışı bir adrese sessizce gitmesini engellemektir.

- Kontrol ayarı mı: Evet.

### QrEmailApproverRole

- Ne işe yarar: E-posta adresi onayını kim verir.

- Varsayılan: ChangeManager

- Kontrol ayarı mı: Evet.

### QrEmailAutoApproveUsers

- Ne işe yarar: Sistemde kayıtlı kullanıcılara ait adresler otomatik onaylı sayılsın mı.

- Değerler: True / False

- Varsayılan: True

- Örnek: True iken bir kullanıcının kendi kurumsal adresine sonuç göndermesi ek onay istemez; kurum dışı bir adres eklerse onay adımı devreye girer.

- Kontrol ayarı mı: Evet.

## 7.10 Parametre değişikliği ne zaman devreye girer

| **Durum** | **Davranış** |
|---|---|
| Yeni açılan talepler | Yeni değer hemen geçerlidir. |
| Onay akışı henüz kurulmamış talepler | Yeni değer geçerlidir. |
| Onay akışı kurulmuş talepler | Görevler ayrılığı ayarları talebe dondurulmuştur, eski değerlerle devam eder. |
| Tamamlanmış talepler | Etkilenmez. |
| Maskeleme tespit ayarları | Her teslimde o anki değerler kullanılır ve kayda dondurulur. |

## Bölüm 8. Ayar Değişikliği Onayı (Dört Göz)

### 8.1 Ne işe yarar

Ayar ekranlarındaki değişikliklerin tek kişinin elinden çıkmamasını sağlar. Bir kullanıcı ayarı değiştirdiğinde değişiklik hemen uygulanmaz, bekleyen kayıt olarak saklanır ve başka bir kullanıcının onayını bekler.

Bu neden gereklidir: onay politikalarını, kritik nesne listesini veya maskeleme ayarını tek kişi kapatabiliyorsa, bütün kontrol sistemi o kişinin insafına kalmış demektir.

### 8.2 Kapsam

SettingsApprovalPolicy parametresi kapsamı belirler:

| Değer | Kapsam |
|---|---|
| Disabled | Yakalama yoktur, her değişiklik anında uygulanır. |
| ControlSettingsOnly | Yalnızca bir kontrolü zayıflatan veya güçlendiren değişiklikler beklemeye düşer. |
| All | Kapsamdaki ekranlardaki her değişiklik beklemeye düşer. |

Kapsamdaki ekranlar: sunucular, kullanıcılar, roller, parametreler, onay politikaları, SQL standartları, kritik nesneler, hassas kolonlar, hassas veri desenleri, aday kolonlar, e-posta kaynakları, bakım görevleri, ITSM biletleri.

#### Kontrol ayarı ne demektir

ControlSettingsOnly kapsamında beklemeye düşecek parametreler kod içinde sabit bir liste ile belirlenir. Bu liste veritabanında tutulmaz. Sebebi şudur: sınıflandırmanın kendisi düzenlenebilir olsaydı, biri önce parametreyi "kontrol değil" olarak işaretleyip sonra korumayı onaysız kapatabilirdi.

Kontrol ayarı sayılan parametreler:

SettingsApprovalPolicy, QrMaskingEnabled, QrSensitiveDataPolicy, QrSensitiveDetectionThresholdPct, QrSensitiveDetectionSampleSize, QrSensitiveDetectionSampleMode, QrSensitiveDetectionTimeBudgetSeconds, UnmaskedColumnRequestPolicy, UnmaskedColumnApproverRole, QrRequesterCanDownload, QrDownloadMinimumRole, QrFileRetentionDays, QrEmailApprovalPolicy, QrEmailApproverRole, QrEmailAutoApproveUsers, RequesterCanSelfApproveExecute, ApproverCanExecute, RequireDistinctApproverPerStep, EmergencyMinimumRole, RejectOnError, ChangeTicketRequired.

Ayrıca onay politikaları, SQL standartları, kritik nesneler, hassas kolonlar ve desenler ile sunucu kontrol bayrakları da her zaman kontrol değişikliği sayılır.

### 8.3 Akış

```
Kullanici ayari degistirir ve kaydeder
```

```
       |
       v
Politika Disabled mi?  ---evet---> Degisiklik ANINDA uygulanir
       | hayir
       v
ControlSettingsOnly ve degisiklik kontrol degil mi? ---evet---> ANINDA uygulanir
       | hayir
       v
Kullanicinin o ekran yetkisi var mi? ---hayir---> Handler kendi hatasini verir
       | evet
       v
Onaylayacak baska kimse var mi? ---hayir---> ANINDA uygulanir (uyari gunluge yazilir)
       | evet
       v
Ayni kayit icin bekleyen talep var mi? ---evet---> HATA: zaten bekleyen bir talep var
       | hayir
       v
Bekleyen kayit olusturulur
Canli kayda DOKUNULMAZ, eski deger yururlukte kalir
Denetim kaydina "SettingChange.Requested" yazilir
       |
       v
Baska bir kullanici onaylar veya reddeder
       |
  +----+-----------------------------+
  |                                  |
ONAY                                RED
  |                                  |
Komut aynen tekrar oynatilir     Hicbir sey uygulanmaz
(dogrulamalar, lisans limiti,    Istenen degerler denetim
 eszamanlilik kontrolu dahil)     kaydinda kalir
  |                                  |
Basarili -> Approved             Rejected
```

```
Basarisiz -> Failed
```

### 8.4 Onayın komutu tekrar oynatması ne demektir

Onay verildiğinde sistem, kullanıcının başlangıçta gönderdiği komutu saklandığı yerden çıkarır ve aynen yeniden çalıştırır.

Bunun faydası: doğrulamalar, lisans limitleri, eşzamanlılık kontrolü ve denetim kaydı, ilk gönderimde nasıl çalışıyorsa onay anında da aynen çalışır. Bu mantık ikinci kez yazılmaz, dolayısıyla iki yol arasında fark oluşamaz.

Bunun bir sonucu vardır: onay anında koşullar değişmişse komut başarısız olabilir. Örneğin sunucu limiti bu arada dolmuşsa onay "Failed" olarak kapanır ve sebebi kayda yazılır.

Kayıt üzerindeki oluşturan ve değiştiren alanları talep edene yazılır, çünkü değişiklik onun eseridir. Onaylayanın izi ayrı alanda ve denetim kaydında tutulur.

### 8.5 Kim onaylar

Ayrı bir "ayar onaylayıcısı" rolü yoktur. Onaylayan kişi o ekranın kendi yetkisine sahip olmalıdır. Sunucu değişikliğini sunucu yetkisi olan biri onaylar.

Değişmez kural: Kimse kendi talebini onaylayamaz veya reddedemez. Bu kural politikadan bağımsız olarak her zaman geçerlidir.

### 8.6 Kilitlenme kapısı

Bir ekranın yetkisine yalnızca tek kişi sahipse, o kişinin yaptığı değişikliği onaylayacak kimse yoktur ve değişiklik sonsuza kadar bekler. Bu durumda sistem değişikliği doğrudan uygular ve günlüğe uyarı yazar.

Bu bilinçli bir tercihtir: kural zaten uygulanamıyorsa sistemi kilitlemek yerine durumu görünür kılmak daha doğrudur. Kalıcı çözüm, her ekran yetkisine en az iki kullanıcı atamaktır.

Aynı sebeple kullanıcı silme işlemi de bu durumu kontrol eder ve son ikinci yetkiliyi silmeyi engeller.

### 8.7 Bekleyen Değişiklikler ekranı

Ekran: Ayarlar > Bekleyen Değişiklikler

| **Kolon** | **Ne gösterir** |
|---|---|
| Kayıt tipi | Hangi ekrana ait değişiklik (sunucu, parametre, politika vb.). |
| Kayıt anahtarı | Hangi kaydın değiştirileceği. |
| İşlem | Ekleme, güncelleme veya silme. |
| Özet | Değişikliğin insan tarafından okunabilir özeti. |
| Talep eden | Değişikliği isteyen kullanıcı. |
| Talep zamanı | İstek zamanı. |
| Durum | Bekliyor, Onaylandı, Reddedildi, Başarısız. |

| Kolon | Ne gösterir |
|---|---|
| Karar veren | Onaylayan veya reddeden kullanıcı. |
| Gerekçe | Red gerekçesi veya başarısızlık sebebi. |

Kaydın üzerine tıklandığında mevcut değer ile istenen değer yan yana gösterilir. Şifre gibi sır taşıyan alanlar maskeli görünür; onay ekranı şifreli komut yükünü hiçbir zaman açmaz.

Reddetme işleminde gerekçe zorunludur.

### 8.8 Bu ekrandaki mesajlar

| Mesaj | Sebep | Ne yapmalı |
|---|---|---|
| Değişiklik onaya gönderildi | Kayıt beklemeye alındı. | Yetkili bir meslektaşınızdan onay isteyin. |
| Bu kayıt için zaten bekleyen bir değişiklik var | Aynı kayda ikinci bir istek gönderildi. | Önce mevcut isteği sonuçlandırın. |
| Bu talep zaten karara bağlanmış | Başka biri sizden önce onayladı veya reddetti. | Listeyi yenileyin. |
| Kendi talebinizi onaylayamazsınız | Dört göz kuralı. | Başka bir yetkiliye onaylatın. |
| Yetkiniz yok | O ekranın yetkisine sahip değilsiniz. | Yetki talep edin. |
| Değişiklik uygulanamadı | Onay anında komut hata verdi. | Gerekçeyi okuyun, sorunu giderip yeniden isteyin. |

## Bölüm 9. Kritik Nesneler, SQL Standartları ve Hassas Veri

### 9.1 Kritik Nesneler

Ekran: Ayarlar > Kritik Nesneler

Yetki: settings.critical

#### Ne işe yarar

Kurumun "bu nesneye dokunulursa özel dikkat gerekir" dediği veritabanı nesnelerinin listesidir. Örnek: müşteri tablosu, bakiye tablosu, yetkilendirme prosedürü.

#### Alanlar

| Alan | Ne işe yarar | Zorunlu | Örnek |
|---|---|---|---|
| Veritabanı Adı | Nesnenin bulunduğu veritabanı. | Evet | MusteriDB |
| Nesne Tipi | Tablo, prosedür, görünüm, fonksiyon, tetikleyici. | Evet | Table |
| Şema Adı | Nesnenin şeması. | Evet | dbo |
| Nesne Adı | Nesnenin adı. | Evet | Musteri |
| Erişim Kontrolü | Bu nesne üzerindeki yetki değişiklikleri izlensin mi. | - | Açık |
| Şema Değişikliği | Bu nesnenin yapısını değiştiren işlemler izlensin mi. | - | Açık |

#### Eşleşme mantığı

Eşleşme dört alanın birlikte karşılaştırılmasıyla yapılır: veritabanı, nesne tipi, şema, nesne adı. Aynı dörtlü iki kez tanımlanamaz.

Karşılaştırma büyük-küçük harf duyarsızdır. dbo.Musteri ile DBO.MUSTERI aynı nesnedir.

#### Kritik nesne tespit edilince ne değişir

1. Talep ayrıştırıldığında ilgili ifade "kritik nesne" olarak işaretlenir. Talep detayındaki nesne tablosunda kritik sütunu işaretli görünür.

2. CriticalObjectDrop standardı devreye girebilir. Bu standardın önem derecesi Critical'dır ve "asla atlanamaz" işaretlidir.

3. Risk bandı Critical'a çıkar.

4. Onay politikasında bu standarda bağlı bir adım varsa o adım kilitlenir ve hiçbir kural ile atlanamaz.

5. Otomatik çalıştırma devre dışı kalır; talep onaylansa bile bir insanın çalıştır düğmesine basması gerekir.

Önemli kural: Kritiklik kararı talep ayrıştırıldığı anda kaydedilir. Kataloğu sonradan düzenlemeniz geçmiş talepleri yeniden yazmaz. Denetimde "o gün ne biliyorduk" sorusunun cevabı bu şekilde korunur.

### 9.2 SQL Standartları

Ekran: Ayarlar > SQL Standartları

Yetki: settings.standards

### Ne işe yarar

Betiklerde aranan kuralların listesidir. Kurulumla birlikte 87 standart gelir. Her biri açılıp kapatılabilir, önem derecesi değiştirilebilir.

### Alanlar

| **Alan** | **Ne işe yarar** | **Değerler** |
|---|---|---|
| Aktif | Kural çalışsın mı. | Açık / Kapalı |
| Standart Anahtarı | Kuralın sistem içindeki adı. Onay politikası adımları bu anahtara bağlanır. | Örnek: XpCmdShellUsage |
| Başlangıç metni | Bazı kurallarda aranan metin. | |
| Hata Mesajı | Kural tetiklendiğinde kullanıcıya gösterilen açıklama. | |
| Önem Derecesi (Tier) | Kuralın risk seviyesi. | 0=Info, 1=Low, 2=Medium, 3=High, 4=Critical |
| Asla Atlanamaz (NeverSkip) | Tetiklenirse otomatik çalıştırma yapılmaz ve buna bağlı onay adımı atlanamaz. | Açık / Kapalı |
| Kaydı Engeller | Tetiklenirse talep hiç kaydedilemez. | Açık / Kapalı |
| Engellenen Talep Tipleri | Kuralın hangi talep tiplerinde kaydı engelleyeceği. | |
| Üst Sınır (MaxValue) | Sayısal kurallarda eşik değeri. | |
| Veritabanı Tipi | Kuralın hangi veritabanlarında geçerli olduğu. Noktalı virgülle ayrılır. Boş ise hepsi. | 1;3 = SQL Server ve PostgreSQL |

### İhlalde ne olur

Üç farklı sonuç vardır:

| Durum | Sonuç |
|---|---|
| Kural tetiklendi, "Kaydı Engeller" kapalı | Talep kaydedilir. Bulgu risk hesabına girer, bandı yükseltebilir. |
| Kural tetiklendi, "Kaydı Engeller" açık | Talep kaydedilmez. Kullanıcı hata mesajını görür ve betiği düzeltmek zorundadır. |
| Kural tetiklendi, "Asla Atlanamaz" açık | Talep kaydedilir, ancak otomatik çalıştırma yapılmaz ve ilgili onay adımı kilitlenir. |

## Örnek standartlar

| Anahtar | Ne kontrol eder | Önem | Asla atlanamaz |
|---|---|---|---|
| CriticalObjectDrop | Kritik nesnenin silinmesi (DROP). | Critical | Evet |
| XpCmdShellUsage | xp_cmdshell çağrısı. İşletim sistemi komutu çalıştırır. | Critical | Evet |
| LinkedServerUsage | Dış veri kaynağı erişimi (bağlı sunucu, OPENROWSET, veritabanı bağlantısı). | High | Evet |
| DynamicSqlUsage | Dinamik SQL kullanımı. Enjeksiyon riski taşır ve çalışan sorgu denetimde görünmez. | High | Hayır |
| TempTableUsage | Geçici tablo kullanımı. | Medium | Hayır |
| WaitForDelayUsage | WAITFOR DELAY kullanımı. Bağlantıyı bloke eder. | Medium | Hayır |
| DeprecatedTypeUsage | Kullanımdan kaldırılmış veri tipleri. | Medium | Hayır |
| IdentityFunctionUsage | @@IDENTITY kullanımı. Tetikleyici zincirinde yanlış değer döndürebilir. | Medium | Hayır |
| RequireTryCatch | Prosedür gövdesinde hata yakalama bloğu yok. | Low | Hayır |
| PrintStatementInObject | Nesne gövdesinde PRINT ifadesi. | Low | Hayır |
| OrderByPosition | ORDER BY 1, 2 gibi konum bazlı sıralama. | Info | Hayır |

Tam liste Ayarlar > SQL Standartları ekranındadır.

### 9.3 Hassas Veri Desenleri

Ekran: Ayarlar > Hassas Veri Desenleri

Yetki: settings.patterns

#### Ne işe yarar

Sorgu sonucundaki değerlerin içinde hassas veri arayan kurallardır. Kolon adına değil, değerin kendisine bakar.

#### Alanlar

| **Alan** | **Ne işe yarar** | **Örnek** |
|---|---|---|
| Ad | Desenin adı. | TCKN |
| Regex | Aranan biçim. | 11 haneli sayı |
| Maskeleme Biçimi | Tam maskeleme veya kısmi maskeleme. | PartialMask |
| Sağda Bırakılacak Karakter | Kısmi maskelemede açık kalacak karakter sayısı. | 4 |
| Doğrulayıcı | Biçim eşleşmesinden sonra çalışan algoritma. | TcknChecksum, Luhn, IbanMod97 |
| Sayısal Kolonlara Uygulanır | Sayı ve tarih kolonları da metne çevrilip taransın mı. | Kapalı |
| Kategori | Sektör veya konu etiketi. | Finance, Health |
| Yerleşik | Ürünle gelen desen mi. | |
| Etkin | Desen çalışsın mı. | |

#### Doğrulayıcılar neden vardır

Yalnızca biçime bakmak yanlış pozitif üretir. 11 haneli her sayı kimlik numarası değildir; sipariş numarası da 11 hane olabilir.

Bu yüzden desen eşleşmesi tek başına yeterli değildir. Doğrulayıcı tanımlıysa algoritmanın da geçmesi gerekir:

| **Doğrulayıcı** | **Ne yapar** |
|---|---|
| TcknChecksum | Kimlik numarasının kontrol basamağı hesabını doğrular. |
| Luhn | Kredi kartı numaralarında kullanılan kontrol algoritmasıdır. |

| Doğrulayıcı | Ne yapar |
|---|---|
| IbanMod97 | IBAN'ın 97'ye bölüm kontrolünü yapar. |

Sonuç: 11 haneli bir sipariş numarası kimlik kontrolünden geçemez ve maskelenmez. Böylece hem gereksiz maskeleme hem de "bu kolonu açın" talepleri azalır.

### Kısmi maskeleme örneği

PartialMask ve sağda 4 karakter ayarıyla 12345678901 değeri *8901 olarak görünür. Tam maskelemede değer tümüyle yıldızla değiştirilir.

## 9.4 Hassas Kolon Kataloğu

Ekran: Ayarlar > Hassas Kolonlar

Yetki: settings.qrsensitive

### Ne işe yarar

Kolon adına göre çalışan sınıflandırma kataloğudur. Veri değerine bakılmaz; bu yüzden sınıflandırma yapmak için kimsenin ham veriyi görmesi gerekmez.

### Alanlar

| Alan | Ne işe yarar | Değerler |
|---|---|---|
| Kolon Adı | Aranan ad. | |
| Eşleşme Biçimi | Exact tam eşleşme, Contains içinde geçme, Regex düzenli ifade. | |
| Şema Adı | Kural yalnızca bu şemada geçerli olsun. Boş ise her şema. | |
| Tablo Adı | Kural yalnızca bu tabloda geçerli olsun. Boş ise her tablo. | |
| Veritabanı Tipi | Kural yalnızca bu türde geçerli olsun. Boş ise hepsi. | |
| Sınıflandırma | Denetçi etiketi. | KVKK-Kimlik, PCI-Kart |
| Maskeleme Biçimi | Tam veya kısmi. Boş ise tam maskeleme. | |
| Sağda Bırakılacak Karakter | Kısmi maskelemede. | |
| Aktif | Kural çalışsın mı. | |

Aynı kolon adı farklı kapsamlarda tanımlanabilir. Tekillik kapsamla birlikte sağlanır.

Sunucunun maskeleme rejimi "Yalnız içerik" ise bu katalog o sunucu için devre dışı kalır; yalnızca desenler çalışır.

### 9.5 Aday Kolonlar (öğrenme döngüsü)

Ekran: Ayarlar > Aday Kolonlar

Yetki: settings.qrsensitive

#### Aday nasıl oluşur

Bir sorgu sonucunda desenle yakalanan ama katalogda olmayan bir kolon bulunursa aday kaydı oluşur veya var olan adayın görülme sayısı artar.

Bu tabloda hiçbir veri değeri tutulmaz. Yalnızca kolon adı, tablo adı, şema adı, sunucu adı ve eşleşen desenin adı saklanır.

#### Kolonlar

| **Kolon** | **Ne gösterir** |
|---|---|
| Kolon Adı | Sonuçtaki kolon adı. |
| Kaynak Şema / Tablo / Kolon | Kolonun gerçek kaynağı. Takma ad kullanılmışsa asıl ad burada görünür. |
| Sunucu | Hangi sunucuda görüldü. |
| Tespit Edilen Desen | Hangi desen eşleşti. |
| Görülme Sayısı | Kaç teslimde karşılaşıldı. Sık görülenler üstte listelenir. |
| İlk / Son Görülme | Tarihler. |
| Durum | New karar bekliyor, Accepted kataloğa alındı, Ignored yok sayıldı. |
| Karar veren / Karar zamanı | |

#### Onaylanırsa ne olur

Aday kabul edildiğinde hassas kolon kataloğuna eklenir. Bundan sonra o kolon adına göre de yakalanır, yani değer desenine uymayan satırlarda bile maskelenir.

Bu döngü sayesinde katalog kullanıldıkça zenginleşir; kurumun kendi veri yapısını elle taramasına gerek kalmaz.

## Bölüm 10. Onay Politikaları

Ekran: Ayarlar > Onay Politikaları

Yetki: settings.approvals

### 10.1 Politika nedir

Politika, "hangi durumda kimin onayı gerekir" sorusunun cevabını tutan kural setidir. Bir politikanın içinde sıralı adımlar vardır ve her adım bir rol ister.

Birden çok politika bulunmasının sebebi, farklı ortamların farklı sertlik gerektirmesidir. Canlı ortamda üç onay isterken test ortamında hiç onay istemeyebilirsiniz.

### 10.2 Politika alanları

| Alan | Ne işe yarar | Boş bırakılırsa |
|---|---|---|
| Ad | Politikanın adı. Talebe kalıcı iz olarak yazılır. | Zorunlu |
| Açıklama | Serbest açıklama. | - |
| Aktif | Politika kullanımda mı. | Pasif politika hiç eşleşmez. |
| Acil Durum Politikası | Bu politika yalnızca acil durum talepleri için mi. | Kapalı ise yalnızca normal talepler eşleşir. |
| Talep sahibi kendi talebini onaylayabilir | Kurum varsayılanını bu politika kapsamında değiştirir. | Boş ise kurum parametresi kullanılır. |
| Onaylayan çalıştırabilir | Aynı şekilde. | Boş ise true kabul edilir. |
| Her adım farklı onaylayan | Aynı şekilde. | Boş ise false kabul edilir. |

Önemli: Acil durum talebi yalnızca acil durum politikasına, normal talep yalnızca normal politikaya eşleşir. Bu iki küme birbirine karışmaz.

### 10.3 Politika eşleştirme (kapsam)

Politikalar doğrudan talebe bağlanmaz. Arada bir eşleştirme kaydı vardır ve bu kayıt kapsamı belirler:

| Alan | Anlamı |
|---|---|
| Sunucu | Belirli bir sunucu. Boş ise sunucu bazlı değil. |
| Ortam | Production, Staging, Test, Development. Boş ise ortam bazlı değil. |
| Talep Tipi | Change, QueryResult, Rollback. Boş ise her tip. |

| Alan | Anlamı |
|---|---|
| Öncelik (Priority) | Aynı kapsamda birden çok politika varsa hangisinin önce deneneceği. Büyük sayı önce gelir. |

### Priority ne işe yarar

Aynı kapsam katmanında birden çok politika tanımlıysa, önce yüksek öncelikli olan denenir. Eşitlik durumunda politika kimliği küçük olan önce gelir; bu, sonucun her zaman aynı çıkmasını sağlar.

Öncelik yalnızca aynı katman içinde karşılaştırılır. Farklı katmanlar arasında öncelik değil, özgüllük belirleyicidir.

### 10.4 Politika seçim akışı

Bu akış, talep kaydedildiğinde onay adımları oluşturulurken çalışır. Talebin her sunucusu için ayrı ayrı yürütülür.

```
Talebin her sunucusu icin:
  |
  v
1. Bu sunucuya OZEL ve tam bu talep tipine ait esleme var mi?
       evet -> adaylar bunlar, oncelige gore sirala
       hayir -> devam
  v
2. Bu sunucuya OZEL ve tip belirtilmemis esleme var mi?
       evet -> adaylar bunlar
       hayir -> devam
  v
3. Sunucunun ORTAMINA ait ve tam bu talep tipine ait esleme var mi?
       evet -> adaylar bunlar
       hayir -> devam
  v
4. Sunucunun ORTAMINA ait ve tip belirtilmemis esleme var mi?
       evet -> adaylar bunlar
       hayir -> devam
  v
5. GENEL (sunucu ve ortam belirtilmemis) ve tam bu tipe ait esleme var mi?
```

```
        evet -> adaylar bunlar
        hayir -> devam
   v
 6. GENEL ve tip belirtilmemis esleme
   |
   v
 Adaylar oncelik sirasiyla denenir.
 Ilk "en az bir aktif adim ureten" aday kazanir.
   |
   v
 Sunucular arasi karsilastirma:
   - Once en kritik ORTAM kazanir (Production > Staging > Test > Development >
tanimsiz)
   - Ayni ortam seviyesinde daha COK aktif adim iceren kazanir
```

### Birden çok politika eşleşirse ne olur

Yukarıdaki sıra tek bir katman seçer. O katmandaki adaylar öncelik sırasıyla denenir ve en az bir aktif adım üreten ilk aday kazanır.

Bu davranışın sebebi şudur: yüksek öncelikli bir politikanın adımları koşullara bağlı olabilir ve hiçbiri tetiklenmeyebilir. O politika kazansaydı, aslında onay isteyecek olan düşük öncelikli politika sessizce devre dışı kalırdı.

### Hiçbiri eşleşmezse ne olur

Hiç politika eşleşmezse onay adımı oluşmaz ve talep otomatik olarak onaylanır. Bu durum denetim kaydına ChangeRequest.AutoApproved olayı olarak yazılır ve talep üzerindeki uygulanan politika alanı boş kalır.

Kontrol İstisnaları raporunda "eşleşen politika yok" başlığı bu durumu listeler. Denetçilerin ilk baktığı yerlerden biridir.

Kurulum önerisi: En az bir genel politika tanımlayın. Böylece hiçbir talep politikasız kalmaz.

### Tüm adımlar atlanırsa ne olur

Politika eşleşti ama koşullar nedeniyle tüm adımlar atlandıysa talep yine otomatik onaylanır. Ancak bu durumda atlanan adımlar gerekçeleriyle birlikte kaydedilir. Talep detayında ve denetim dosyasında "hangi adım neden atlandı" görünür.

Bu iki durumu ayırt etmenin yolu: uygulanan politika adı doluysa politika eşleşmiş ama adımlar atlanmıştır, boşsa hiç politika eşleşmemiştir.

### 10.5 Adım alanları ve davranışları

Her politika adımı için aşağıdaki alanlar tanımlanır.

| Alan | Ne işe yarar | Ne set edersen nasıl davranır |
|---|---|---|
| Sıra (StepOrder) | Adımın kaçıncı sırada olduğu. | Adımlar bu sıraya göre oluşturulur. Aynı politikada aynı sıra iki kez kullanılamaz. |
| Adım Adı | Ekranda görünen ad. | Örnek: "Yönetici Onayı". |
| Onaylayan Rol | Adımı kapatabilecek en düşük rol. | DbAdmin yazarsanız DbAdmin ve üstü onaylayabilir. |
| Band En Az (TriggerIfBandAtLeast) | Risk bandı bu değere eşit veya üstündeyse adım devreye girer. | 3 (High) yazarsanız adım yalnızca High ve Critical taleplerde çalışır. |
| Band En Fazla (SkipIfBandAtMost) | Risk bandı bu değere eşit veya altındaysa adım atlanır. | 1 (Low) yazarsanız Info ve Low taleplerde adım atlanır. |
| Tetikleyen Standartlar (TriggerIfStandardKeys) | Bu standartlardan biri tetiklenmişse adım devreye girer. Noktalı virgülle ayrılır. | XpCmdShellUsage;LinkedServerUsage yazarsanız adım yalnızca bu bulgular varsa çalışır. |
| Talep Sahibi Rolü En Az (SkipIfRequesterRoleAtLeast) | Talebi açan kişinin rolü bu seviyede veya üstündeyse adım atlanır. | ChangeManager yazarsanız değişiklik yöneticisinin açtığı taleplerde bu adım atlanır. |
| Kilitli Adım (IsNonSkippable) | Adım hiçbir kural ile atlanamaz. | Açıksa band, rol ve diğer tüm atlama kuralları geçersizdir. |

### Adım nasıl karar verir

Bir adım için sıralı olarak şu kontroller yapılır:

```
1. Adim KILITLI mi (IsNonSkippable)?          evet -> ATLANMAZ, karar biter
2. Adimi tetikleyen bir standart NeverSkip
   isaretli ve tetiklendi mi?                 evet -> ATLANMAZ, karar biter
3. Adim bir TETIKLEME adimi mi
   (standart anahtari veya band esigi var)?
      evet ve hicbiri eslesmedi               -> ATLANIR ("kosul olusmadi")
      evet ve eslesti                         -> asagidaki 5 ve 6 UYGULANMAZ
4. Talep sahibinin rolu esigi asiyor mu?      evet -> ATLANIR ("talep sahibi rolu")
5. Talep sahibinin rolu, adimin onaylayan
   rolune esit veya ustunde mi?               evet -> ATLANIR ("sahip rolu")
6. Band, atlama esiginin altinda mi?          evet -> ATLANIR ("dusuk band")
```

```
7. Hicbiri degilse                          -> ADIM AKTIF, onay bekler
```

Dikkat edilmesi gereken nokta: 3. adımda koşulu eşleşen bir tetikleme adımı, 5 ve 6 numaralı kurallarla atlanamaz. Yani "yüksek riskte devreye gir" diye tanımlanmış bir adım, talebi açan kişinin rolü yüksek diye atlanmaz. Aksi halde riskli işi yapan kişinin kıdemi kontrolü kaldırırdı.

### Örnek politika

Bir üretim ortamı politikası şöyle kurulabilir:

| Sıra | Adım Adı | Onaylayan Rol | Koşul | Kilitli |
|---|---|---|---|---|
| 1 | Takım Lideri Onayı | RequestManager | Band en fazla Info ise atla | Hayır |
| 2 | Veritabanı Yöneticisi Onayı | DbAdmin | Band en az Medium | Hayır |
| 3 | Değişiklik Yöneticisi Onayı | ChangeManager | Band en az High | Hayır |
| 4 | Güvenlik Onayı | SystemManager | XpCmdShellUsage;LinkedServerUsage tetiklenirse | Evet |

Bu politikada:

- Info bandındaki bir talep hiç onay istemez, otomatik onaylanır.

- Medium bandındaki bir talep 1. ve 2. adımı ister.

- High bandındaki bir talep 1, 2 ve 3. adımı ister.

- İçinde xp_cmdshell geçen bir talep ayrıca 4. adımı ister ve bu adım hiçbir şekilde atlanamaz.

### 10.6 Görevler ayrılığı (SoD) kuralları

Üç ayar üç farklı soruyu cevaplar:

| Ayar | Soru |
|---|---|
| RequesterCanSelfApproveExecute | Talebi açan kendi talebini onaylayabilir ve çalıştırabilir mi? |
| ApproverCanExecute | Onaylayan kişi aynı talebi çalıştırabilir mi? |
| RequireDistinctApproverPerStep | Aynı kişi birden çok adımı kapatabilir mi? |

Bunlar önce kurum parametresi olarak tanımlanır (Bölüm 7.4), sonra politika bazında değiştirilebilir. Politikada boş bırakılan alan kurum varsayılanını kullanır.

Dondurma kuralı: Onay akışı kurulurken etkin değerler talebe yazılır. Politika veya parametre sonradan değişse bile o talep girdiği kural setiyle değerlendirilir. Denetim dosyası o günün kuralını gösterir.

### Rol atlama izi

Bir adım rol kuralı nedeniyle atlandığında adım kaydı silinmez; "atlandı" durumuyla ve gerekçesiyle saklanır. Karar veren olarak "System" yazılır. Böylece denetimde "bu adım neden hiç sorulmadı" sorusu cevaplanabilir.

## 10.7 Kilitlenme senaryoları

| Senaryo | Sonuç | Çözüm |
|---|---|---|
| Adım DbAdmin istiyor, sistemde DbAdmin yok | Talep sonsuza kadar bekler. | Rolü olan kullanıcı ekleyin veya adımın rolünü düşürün. |
| RequireDistinctApproverPerStep açık, iki adım var, yalnızca bir yetkili kişi var | İkinci adım kapatılamaz. | İkinci yetkili ekleyin. |
| RequesterCanSelfApproveExecute kapalı, tek yetkili kişi hem talebi açıyor hem onaylıyor | Kendi talebini onaylayamaz. | Başka bir onaylayıcı ekleyin. |
| Acil durum politikası tanımlı değil | Acil durum seçeneği kullanılamaz. | Bkz. 10.8. |

## 10.8 Acil durum uygunluğu

Bir talepte acil durum seçeneğinin görünmesi için:

1. Seçili sunucuların hepsi için eşleşen bir acil durum politikası bulunmalı.

2. Bu politikalarda en az bir kilitli adım olmalı.

İkinci şart bilinçlidir. Acil durum, onayı tamamen kaldırmak değil, kısaltmaktır. Hiç kilitli adım yoksa acil durum ilan etmek onay mekanizmasını tümüyle devre dışı bırakırdı ve bu bir yönetişim boşluğu olurdu.

Ayrıca acil durum talebi açabilmek için kullanıcının rolü EmergencyMinimumRole eşiğini geçmelidir.

## Bölüm 11. Değişiklik Talebi Uçtan Uca

### 11.1 Durum makinesi

| Değer | Durum | Anlamı | Ölü mü |
|---|---|---|---|
| 1 | Pending | Onay bekliyor. | Hayır |
| 2 | Approved | Onaylandı, çalıştırılmayı bekliyor. | Hayır |
| 3 | Executed | Çalıştırıldı. | Evet |
| 4 | Rejected | Reddedildi. | Evet |
| 5 | Manual | Elle uygulandığı beyan edildi. | Evet |
| 6 | NoLongerNeeded | Artık gerekli değil. | Evet |
| 7 | Executing | Şu anda çalışıyor. | Hayır |

Ölü durum ne demektir: Talep aktif akıştan çıkmıştır. Worker onu almaz, üzerinde yeni işlem yapılamaz, düzenlenemez.

### Geçiş matrisi

| Mevcut durum | Yapılabilecek işlem | Yeni durum |
|---|---|---|
| Pending | Onayla (son adım kapanınca) | Approved |
| Pending | Reddet | Rejected |
| Pending | Manuel işaretle | Manual |
| Pending | Artık gerekli değil | NoLongerNeeded |
| Pending | Düzenle (yalnızca sahibi) | Pending |
| Approved | Çalıştır | Executing, sonra Executed veya Rejected |
| Approved | Zamanla | Approved (zamanlama bilgisi eklenir) |
| Approved | Zamanlamayı iptal et | Approved |
| Approved | Reddet | Rejected |
| Approved | Manuel işaretle | Manual |
| Approved | Artık gerekli değil | NoLongerNeeded |
| Executing | Çalıştırmayı iptal et | Approved |
| Executed | Tamamlandı işaretle | Executed (bilgi güncellenir) |

| Mevcut durum | Yapılabilecek işlem | Yeni durum |
|---|---|---|
| Rejected, Manual, NoLongerNeeded | Hiçbiri | - |

Çalıştırma hata verdiğinde davranış RejectOnError parametresine bağlıdır: true ise Rejected, false ise Approved durumuna döner.

### 11.2 Talep oluşturma ekranı

Ekran: Talepler > Yeni Talep

Yetki: cr.add

| Alan | Ne işe yarar | Zorunlu | Notlar |
|---|---|---|---|
| Açıklama | Talebin ne yaptığının insan diliyle özeti. | Evet | Listelerde ve raporlarda görünür. |
| Talep Tipi | Change (değişiklik) veya QueryResult (sorgu sonucu). | Evet | Rollback tipi elle seçilmez, üretilir. |
| Sunucular | Hedef sunucular. Birden çok seçilebilir. | Evet | Bkz. Bölüm 6.6. |
| Bilet Numarası | ITSM bilet numarası. | ChangeTicketRequired parametresine bağlı | Biçim kontrolü ChangeTicketRegex ile yapılır. |
| Sorgu Zaman Aşımı | Betiğin en fazla ne kadar çalışabileceği. | - | Alt sınır DefaultQueryTimeout, üst sınır MaxQueryTimeout. |
| Betikler | Bir veya daha çok SQL metni. | Evet | Her betiğin adı ve sırası vardır. |
| Acil Durum | Acil durum talebi mi. | - | Görünmesi için Bölüm 10.8 şartları gerekir. |
| Acil Durum Gerekçesi | Acil durum seçildiyse zorunlu yazılı gerekçe. | Koşullu | |
| Acil Durum Bileti | Acil durum kaydının bilet numarası. | Koşullu | |
| İstenen Çalıştırma Tarihi | Talebin ne zaman çalışmasını istediğiniz. | - | Otomatik zamanlamada kullanılır. |
| Gecikme (dakika) | Onaydan sonra kaç dakika beklensin. | - | Tarih verilmediyse kullanılır. |

Sorgu sonucu talepleri için ek alanlar Bölüm 13'te anlatılır.

## 11.3 Betik editörü ve ayrıştırma

### Ayrıştırma ne zaman çalışır

Betik yazıldıktan sonra ayrıştırma tetiklenir ve talep kaydedilirken tekrar çalışır. Kaydetme anındaki sonuç kalıcı olarak saklanır.

### Ayrıştırma neyi üretir

| **Üretilen bilgi** | **Açıklama** |
|---|---|
| Nesne listesi | Betiğin dokunduğu her nesne: veritabanı, şema, ad, tip. |
| İşlem türü | CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, GRANT gibi. |
| Bulgu listesi | Tetiklenen SQL standartları, önem dereceleriyle. |
| Kritik nesne işareti | Dokunulan nesne katalogda var mı. |
| Risk bandı | Tüm bulguların değerlendirilmesiyle. |

### Ayrıştırma hatası

Betik çözümlenemezse (sözdizimi hatası veya desteklenmeyen yapı) bulgu listesinde ayrıştırma hatası olarak görünür. Bu durumda:

- Talep yine kaydedilebilir.

- Nesne listesi boş kalabilir.

- Onay adımlarında standart anahtarına bağlı tetikleyiciler çalışmaz, çünkü hangi kuralın tetiklendiği bilinemez. Bu durumda daha temkinli davranmak için band bazlı adımlar kullanmanız önerilir.

## 11.4 Talep detay ekranı

Ekran: Talepler > (bir talep seçilir)

### Üst bilgi alanları

| **Alan** | **Ne gösterir** |
|---|---|
| Talep numarası | Kaydın kimliği. |
| Açıklama | Talep açıklaması. |
| Durum | Bölüm 11.1'deki durumlardan biri. |
| Talep tipi | Change, QueryResult veya Rollback. |

| Alan | Ne gösterir |
|---|---|
| Risk bandı ve skoru | Bölüm 12. |
| Sunucular | Noktalı virgülle ayrılmış hedef listesi. |
| Bilet numarası | ITSM bileti. |
| Acil durum | Acil durum talebi ise işaret ve gerekçe. |
| Uygulanan politika | Onay akışı kurulurken eşleşen politika adı. Boş ise politika eşleşmemiştir. |
| Oluşturan / Oluşturma zamanı | |
| Çalıştırmayı başlatan / Zamanı | Düğmeye basan kişi. Otomatik zamanlamada SYSTEM. |
| Çalıştıran / Çalıştırma zamanı | İşi yürüten servis veya kullanıcı. |
| Etkilenen satır sayısı | Çalıştırma sonucunda değişen satır sayısı. |

### Betikler ve nesneler tablosu

Bu tablo kılavuzun en çok sorulan kolonlarını içerir.

| Kolon | Ne gösterir | Nasıl hesaplanır | Örnek |
|---|---|---|---|
| Betik | Hangi betiğe ait. | Betiğin adı. | 01_tablo_ekle.sql |
| Şema | Nesnenin şeması. | Ayrıştırmadan. | dbo |
| Nesne Adı | Dokunulan nesne. | Ayrıştırmadan. | Musteri |
| Nesne Tipi (Object Type) | Nesnenin türü. | Ayrıştırmadan. | Table, Procedure, View |
| İşlem (Operation) | Nesneye yapılan işlem. | Ayrıştırmadan. | ALTER, DROP, INSERT |
| İfade (Statement) | Betikteki ilgili SQL parçası veya satır bilgisi. | Ayrıştırmadan, satır ve sütun numarasıyla. | satır 12 |
| Kritik Nesne (Critical Object) | Bu nesne kritik nesne kataloğunda mı. | Talep ayrıştırıldığı anda katalogla karşılaştırılır ve karar kaydedilir. Katalog sonradan değişse bile bu işaret değişmez. | İşaretli / Boş |
| Riskli İşlem (Risky Operation) | Bu ifade bir SQL standardını tetikledi mi. | Tetiklenen standardın önem derecesi Medium ve üstündeyse riskli sayılır. | İşaretli / Boş |
| Mesaj | Tetiklenen standardın açıklaması. | Standart tanımındaki hata mesajı. | "Dinamik SQL kullanımı tespit edildi." |

| Kolon | Ne gösterir | Nasıl hesaplanır | Örnek |
|---|---|---|---|
| Önem | Bulgunun risk seviyesi. | Standart tanımındaki Tier. | High |

Critical Object ve Risky Operation farkı:

- Critical Object, *neye* dokunulduğu ile ilgilidir. Kurum o nesneyi önemli ilan etmiştir.

- Risky Operation, *ne yapıldığı* ile ilgilidir. İşlemin kendisi bir kuralı ihlal etmiştir.

Bir talep sıradan bir tabloya tehlikeli bir işlem yapıyorsa Risky Operation işaretli, Critical Object boş olur. Kritik bir tabloya zararsız bir SELECT yapıyorsa tersi olur. İkisi birden işaretliyse talep en yüksek dikkati hak eder.

### Onay adımları tablosu

| Kolon | Ne gösterir |
|---|---|
| Sıra | Adımın sırası. |
| Adım Adı | Politikadan gelen ad. |
| Gereken Rol | Adımı kapatabilecek en düşük rol. |
| Durum | Bekliyor, Onaylandı, Reddedildi, Atlandı. |
| Karar Veren | Adımı kapatan kullanıcı. Atlanan adımlarda "System". |
| Karar Zamanı | |
| Açıklama / Gerekçe | Onay yorumu veya atlama gerekçesi. |
| Geçersiz Kılma | Adım acil durum yetkisiyle geçildiyse işaretlidir. |
| Atanan Kişi | Talep sahibinin doğrudan yöneticisi adım rolünü karşılıyorsa adım ona atanır. |

### Çalıştırma sonucu

| Alan | Ne gösterir |
|---|---|
| Çalıştırma günlüğü | Hangi betiğin hangi bloğu çalıştı, kaç satır etkilendi, hata varsa hangi blokta oluştu. |
| Etkilenen satır | Toplam. |
| Hata mesajı | Varsa. |

| Alan | Ne gösterir |
|---|---|
| Kısmi çalıştırma uyarısı | Oracle hedeflerde, hata öncesi çalışan blokların geri alınmamış olabileceği uyarısı. |

### Denetim / olay listesi

Talebe ait denetim kayıtları burada listelenir: oluşturma, düzenleme, onay, red, zamanlama, çalıştırma, iptal.

## 11.5 Düğmeler ve aktiflik koşulları

Her düğme için iki farklı davranış vardır:

- Gizli: Kullanıcı bu işlemi hiçbir zaman yapamaz (rol yetkisi yok, talep tipi uygun değil, sahibi değil, modül lisanslı değil).

- Pasif (gri): Kullanıcı yapabilir ama şu anda yapılamaz (yanlış durum, aktif paket, kendi talebi, kaynak henüz uygulanmadı). Fare üzerine gelince sebep görünür.

| Düğme | Görünme koşulu | Aktif olma koşulu |
|---|---|---|
| Düzenle | Talebin sahibi olmak | Talep ölü durumda değil, aktif pakette değil |
| Onayla | Onay yetkisi | Durum Pending, aktif pakette değil, kendi talebi değilse veya izin varsa, uygun bekleyen adım var |
| Reddet | Onay yetkisi | Durum Pending veya Approved |
| Çalıştır | Çalıştırma yetkisi, QueryGovernance kapalıysa QR için gizli | Durum Approved, aktif pakette değil, gerekliyse geri alma var, görevler ayrılığı kuralları engellemiyor |
| Zamanla | Onay yetkisi ve zamanlama modülü | Durum Approved, aynı görevler ayrılığı kuralları |
| Zamanlamayı iptal et | Onay yetkisi | Durum Approved ve zamanlama var |
| Tamamlandı işaretle | Onay yetkisi | Durum Executed |
| Manuel işaretle | Onay yetkisi | Durum Pending veya Approved |
| Artık gerekli değil | Sahibi veya onay yetkisi | Durum Pending veya Approved |
| Geri alma üret | Talep tipi Change, sahibi, ekleme yetkisi, geri alma modülü | Ölü durumda değil (çalıştırma sırasında düşmüş talepler istisnadır), aktif geri alma yok |
| Kopyala | Ekleme yetkisi, talep tipi Rollback değil | Her zaman |

| Düğme | Görünme koşulu | Aktif olma koşulu |
|---|---|---|
| Sandbox | Talep tipi Change, sandbox yetkisi ve modülü | Ölü durumda değil |
| Çalıştırmayı iptal et | Çalıştırma yetkisi | Durum Executing |
| Sonucu indir | QueryGovernance modülü, indirme kuralları | Sonuç dosyası var |

Düğme ve arka uç tutarlılığı: Düğmenin aktif olup olmadığına karar veren kural ile arka ucun isteği kabul edip etmediğine karar veren kural aynı koddur. Bu, "düğme aktifti ama işlem hata verdi" durumunun oluşmamasını sağlar.

### 11.6 Çalıştırma akışı

```
Kullanici "Calistir" der
       |
       v
Yetki ve durum kontrolleri (CrPolicy)
       |
       v
Talep "Executing" durumuna alinir, kuyruga girer
ExecutionRequestedBy = dugmeye basan kisi
       |
       v
Worker kuyruktaki talebi sahiplenir
       |
       v
Hedef sunucuda transaction acilir
       |
       v
Betikler sirayla, blok blok calistirilir
       |
  +----+------------------------+
  |                             |
Hepsi basarili              Hata olustu
```

```
  |                              |
Hedef commit                 Hedef rollback
Uygulama kaydi commit        Talep Rejected veya Approved
Durum Executed               (RejectOnError parametresine gore)
  |                              |
  v                              v
Bildirim gonderilir        Bildirim gonderilir
```

### Zamanlanmış çalıştırma

Elle zamanlamada kullanıcı bir tarih seçer. Bu tarih en az ExecutionDelaySeconds (en az 30 saniye) sonrası olmalıdır. Zamanı gelince worker talebi alır.

Otomatik zamanlamada tarih sistem tarafından hesaplanır. Bkz. Bölüm 6.4.

### Çalıştırmayı iptal etme

Talep çalışırken iptal istenebilir. Sistem iki saniyede bir iptal isteğini kontrol eder. İptal geldiğinde çalışan işlem durdurulur, hedefteki transaction geri alınır ve talep Approved durumuna döner.

### Çökme kurtarma

Worker servisi çalışma ortasında durursa, talep Executing durumunda kalır. Bu talepler bir zaman eşiğine göre tespit edilip Approved durumuna geri alınır, böylece kilitli kalmazlar. Eşik Workers:StaleExecutionRecoveryMinutes ayarındadır ve varsayılanı 120 dakikadır. Ayrıntı ve eşiğin nasıl seçileceği: Bölüm 25.7.

## 11.7 Kısmi başarı ve geri alma

Bir talep birden çok blok içerir. Hata oluşursa:

| **Veritabanı** | **Davranış** |
|---|---|
| SQL Server, PostgreSQL | Transaction geri alınır. Hiçbir değişiklik kalmaz. |
| Oracle | DDL ifadeleri örtük olarak kaydedilir (implicit commit). Hatadan önce çalışan DDL blokları geri alınamaz. Çalıştırma günlüğüne bu durum açıkça yazılır. |

Bu yüzden Oracle hedeflerde geri alma betiği daha önemlidir ve DDL içeren taleplerde zorunlu tutulabilir.

Hata mesajında kaçıncı bloğa kadar çalışıldığı ve hangi blokta durulduğu yazılır.

## 11.8 Geri alma (rollback) talepleri

- Geri alma talebi, bir değişiklik talebine bağlı olarak açılır.

- Yalnızca talebin sahibi üretebilir.

- Bir talebin aynı anda birden çok aktif geri alma talebi olamaz.

- Geri alma talebi normal onay akışına girer. Politikadan adım çıkmazsa kaynak talebin onay gereksinimi yansıtılır: değişikliği onaylamaya yetkili rol, geri almayı da onaylar.

- Geri alma yalnızca kaynak talep uygulandıktan sonra çalıştırılabilir. İstisna: kaynak talep çalıştırma sırasında hata verdiyse, kısmi değişikliği temizlemek için geri alma çalıştırılabilir.

- Geri alma talepleri otomatik zamanlanmaz. Her zaman bir insanın çalıştırması gerekir.

### 11.9 Çalıştırmayı başlatan kişi (attribution)

Worker çalıştırdığında "çalıştıran" alanına servis adı yazılır. Bu, denetim için yeterli değildir: asıl soru düğmeye kimin bastığıdır.

Bu yüzden ayrı bir alan tutulur:

| **Durum** | **ExecutionRequestedBy** |
|---|---|
| Kullanıcı elle çalıştırdı | Kullanıcı adı |
| Kullanıcı elle zamanladı | Kullanıcı adı |
| Sistem otomatik zamanladı | SYSTEM |
| Sürüm paketi içinde çalıştı | Paketi çalıştıran kişi |

Zamanlama iptal edilirse bu alan temizlenir. Aksi halde iptal edilmiş bir işlem için "şu kişi başlattı" yazılı kalırdı.

### 11.10 Acil durum geçersiz kılma (Emergency Override)

Acil durum talebinde, kilitli olmayan onay adımları yetkili kişi tarafından geçilebilir. Bu işlem:

- Yazılı gerekçe ister.

- Adım kaydına "geçersiz kılındı" işareti ve gerekçe yazar.

- Denetim kaydına ayrı bir olay olarak geçer.

- Kontrol İstisnaları raporunda listelenir.

- Kontrol panelindeki istisna sayacına yansır.

Kilitli adımlar acil durumda bile geçilemez. Acil durum onayı kaldırmaz, kısaltır.

Acil durum ilan edebilmek için kullanıcının rolü EmergencyMinimumRole eşiğini geçmelidir.

### 11.11 Talep listesi ekranı

| Kolon | Ne gösterir |
|---|---|
| Numara | Talep kimliği. |
| Açıklama | Talep açıklaması. |
| Tip | Change, QueryResult, Rollback. |
| Durum | Bölüm 11.1. |
| Risk | Band ve skor. |
| Sunucular | Noktalı virgülle ayrılmış. |
| Bilet | ITSM bileti. |
| Oluşturan | |
| Oluşturma zamanı | Tarayıcınızın yerel saatiyle. |
| Onay durumu | Bekleyen adım varsa hangi rolün beklendiği. |
| Zamanlama | Varsa çalıştırma zamanı. |

Filtreler: durum, tip, sunucu, tarih aralığı, oluşturan, bilet numarası, metin arama.

cr.seeall yetkisi olmayan kullanıcı yalnızca kendi taleplerini görür.

## Bölüm 12. Risk Skoru

### 12.1 Amaç ve ilke

Risk modeli tek bir soruya cevap verir: "Bu talep ne kadar dikkat gerektiriyor?"

Modelin üç temel ilkesi vardır:

1. En kötü durum belirleyicidir. Ortalama, toplam veya çarpan yoktur. Bulguların en yükseği bandı belirler.

2. Karar bandla verilir, sayıyla değil. Onay adımlarının açılması, otomatik çalıştırmanın kapanması gibi kararlar banda bakar. Sayısal skor yalnızca gösterim içindir.

3. Sonuç deterministiktir. Aynı bulgular her zaman aynı bandı ve aynı skoru üretir.

### 12.2 Band nedir

Band, talebin risk seviyesidir. Beş kademe vardır:

| **Band** | **Sayısal değer** | **Anlamı** |
|---|---|---|
| Info | 0 | Bilgilendirme. Risk sayılmaz. |
| Low | 1 | Düşük risk. |
| Medium | 2 | Orta risk. |
| High | 3 | Yüksek risk. |
| Critical | 4 | Kritik risk. |

Her SQL standardının bir önem derecesi (Tier) vardır ve bu derece yukarıdaki beş kademeden biridir.

### 12.3 Band nasıl belirlenir

```
Betikler ayristirilir
        |
        v
Tetiklenen standart bulgulari toplanir
        |
        v
Ayni standart anahtarina ait bulgular TEK GRUP olur
(grubun derecesi = gruptaki en yuksek derece)
        |
        v
BAND = tum gruplarin EN YUKSEK derecesi
```

Hiç bulgu yoksa band Info olur.

Kritik ayrıntı: Gruplama standart anahtarına göre yapılır. Aynı standardın betik içinde on kez tetiklenmesi tek bir grup sayılır.

### 12.4 Skor nasıl hesaplanır

Skor 0-100 arasında bir sayıdır ve yüksek sayı daha güvenli demektir. Her bandın kendi aralığı vardır:

| Band | Skor aralığı |
|---|---|
| Info | 90 - 100 |
| Low | 70 - 89 |
| Medium | 50 - 69 |
| High | 25 - 49 |
| Critical | 0 - 24 |

Hesap şu şekildedir:

```
aralik_ust = bandin ust siniri
aralik_alt = bandin alt siniri

ust_kademe_sayisi = bandi belirleyen derecedeki FARKLI standart sayisi

eger ust_kademe_sayisi <= 0 ise
    skor = aralik_ust
degilse
    dusus = min(ust_kademe_sayisi - 1, aralik_ust - aralik_alt)
    skor  = aralik_ust - dusus
```

Yani üst kademede ne kadar çok farklı bulgu varsa skor aralık içinde o kadar aşağı iner, ancak asla bandın dışına çıkmaz.

### 12.5 Somut örnek: band High ve üç High bulgu

Bir talepte şu bulgular tetiklendi:

| Standart | Derece | Kaç kez |
|---|---|---|
| DynamicSqlUsage | High | 2 |
| NoUpdateWithoutWhere | High | 1 |

| Standart | Derece | Kaç kez |
|---|---|---|
| NoTruncateTable | High | 5 |
| TempTableUsage | Medium | 3 |
| PrintStatementInObject | Low | 1 |

Adım adım:

1. Gruplama: Beş farklı standart anahtarı var, yani beş grup. NoTruncateTable beş kez tetiklendi ama tek grup sayılır.

2. Band: Gruplardaki en yüksek derece High'dır. Band = High.

3. Üst kademe sayısı: High derecesindeki grup sayısı = 3 (DynamicSqlUsage, NoUpdateWithoutWhere, NoTruncateTable).

4. Aralık: High için 25 - 49.

5. Düşüş: min(3 - 1, 49 - 25) = min(2, 24) = 2.

6. Skor: 49 - 2 = 47.

Sonuç: Band High, skor 47.

### Aynı örneğin varyasyonları

| Durum | Üst kademe grup sayısı | Skor |
|---|---|---|
| Tek High bulgu | 1 | 49 - 0 = 49 |
| İki farklı High bulgu | 2 | 49 - 1 = 48 |
| Üç farklı High bulgu | 3 | 49 - 2 = 47 |
| Aynı High standardı 10 kez tetiklendi | 1 | 49 - 0 = 49 |
| 30 farklı High bulgu | 30 | 49 - min(29, 24) = 49 - 24 = 25 |
| Hiç bulgu yok | 0 | Band Info, skor 100 |
| Tek Critical bulgu | 1 | Band Critical, 24 - 0 = 24 |

Son satır önemlidir: tek bir Critical bulgu, otuz High bulgudan daha düşük (daha riskli) bir skor üretir. Bu bilinçlidir; en kötü durum belirleyicidir.

### 12.6 Skor açıklaması (ScoreExplain)

Kullanıcıya yalnızca bir sayı göstermek yeterli değildir. Sistem her talep için açıklama üretir:

| Alan | Ne gösterir |
|---|---|
| Band | Hesaplanan risk bandı. |
| Skor | 0-100 arası sayı. |
| Belirleyici bulgu | Bandı belirleyen bulgu. Listede ilk sırada gösterilir. |
| Bulgu listesi | Her bulgu için: standart anahtarı, derecesi, mesajı, kaç kez tetiklendiği, asla atlanamaz işareti. |
| Etki | Her bulgunun bandı belirleyip belirlemediği. Determining bandı belirledi, Neutral belirlemedi. |
| Asla atlanamaz var mı | Bulgular arasında NeverSkip işaretli olan var mı. |

Bulgular önce derecesine göre azalan, sonra anahtarına göre alfabetik sıralanır. Böylece en önemli bulgu her zaman üstte durur.

### 12.7 Asla atlanamaz (NeverSkip) kuralları

Bir SQL standardı "asla atlanamaz" olarak işaretlenebilir. Böyle bir standart tetiklendiğinde:

1. Otomatik çalıştırma yapılmaz. Talep onaylansa bile sistem onu kendiliğinden çalıştırmaz. Daha önce sistem tarafından konmuş bir zamanlama varsa temizlenir.

2. Buna bağlı onay adımı kilitlenir. Adımın atlama koşulları ne olursa olsun adım çalışır.

3. Talep detayında ve denetim dosyasında bu bulgu ayrıca işaretlenir.

Varsayılan olarak asla atlanamaz işaretli standartlar: CriticalObjectDrop, XpCmdShellUsage, LinkedServerUsage ve benzeri en yüksek riskli kurallar. Tam liste Ayarlar > SQL Standartları ekranında görülebilir.

Atlanmaya çalışılırsa ne olur: Kilitli bir adımı atlamanın hiçbir yolu yoktur. Acil durum geçersiz kılma bile kilitli adımı geçemez.

### 12.8 Skor nerede görünür

| Yer | Nasıl görünür |
|---|---|
| Talep listesi | Band rozeti ve skor. |
| Talep detayı | Band, skor, bulgu listesi, açıklama. |
| Kontrol paneli | Risk dağılımı grafiği. |

| Yer | Nasıl görünür |
|---|---|
| Yüksek Riskli Değişiklikler raporu | Band ve skor sütunları. |
| Denetim dosyası | Risk özeti ve bulgu listesi. |

### 12.9 Skorun kararlara etkisi

| Karar | Neye bakar |
|---|---|
| Onay adımının devreye girmesi | Banda (TriggerIfBandAtLeast). |
| Onay adımının atlanması | Banda (SkipIfBandAtMost). |
| Otomatik çalıştırma | NeverSkip bulgusu varlığına. |
| Talebin kaydedilebilmesi | "Kaydı engeller" işaretli standartlara. |

Sayısal skor hiçbir kararda kullanılmaz. Yalnızca insan okuması içindir.

## Bölüm 13. Sorgu Sonucu (Query Result) Akışı

### 13.1 Ne işe yarar

Canlı ortamdan veri okuma ihtiyacı her kurumda vardır: bir müşteri şikayetinin araştırılması, bir hatanın kök nedeninin bulunması, bir raporun doğrulanması. Bu ihtiyaç genellikle kontrolsüz karşılanır; birine "şu sorguyu çalıştır, sonucu bana at" denir.

Sorgu Sonucu özelliği bu işi kayda bağlar:

- Sorgu bir talep olarak açılır ve onaydan geçer.

- Sonuç doğrudan ekranda gösterilmez; şifreli bir dosya olarak paketlenir.

- Hassas kolonlar otomatik maskelenir.

- Kimin hangi veriyi hangi gerekçeyle aldığı kayıt altına alınır.

Bu özellik QueryGovernance modülüne bağlıdır.

### 13.2 Sorgu tipleri

| Değer | Tip | Anlamı |
|---|---|---|
| 0 | None | Tanımsız. |
| 1 | Change | Standart değişiklik talebi. |
| 2 | QueryResult | Veri okuma talebi. Geri alınamaz, sonuç dosya olarak teslim edilir. |
| 3 | Rollback | Bir değişikliğe bağlı geri alma talebi. |

### 13.3 Sorgu sonucu talebi oluşturma

Bölüm 11.2'deki alanlara ek olarak:

| Alan | Ne işe yarar | Zorunlu |
|---|---|---|
| Alıcı (To) | Sonucun gönderileceği e-posta adresleri. Noktalı virgülle ayrılır. | Hayır |
| Bilgi (CC) | Kopya alıcılar. | Hayır |
| Maskesiz istenen kolonlar | Maskelenmeden gelmesi istenen kolonlar. UnmaskedColumnRequestPolicy Disabled ise gizlenir. | Hayır |
| Maskesiz isteme gerekçesi | Yukarıdaki liste doluysa gerekçe. | Koşullu |

Betik sayısı kuralı: Sorgu sonucu talebine yalnızca tek bir betik eklenebilir. Birden çok sorgu çalıştırmanız gerekiyorsa hepsini tek betikte birleştirin; her sonuç kümesi çıktı dosyasında ayrı bir Excel sayfası olarak yer alır. Sayfa adı sunucu adı ve sıra numarasından oluşur, örnek: PROD-SQL01_01.

### Kolon önizlemesi

Talep kaydedilirken, QrColumnPreviewEnabled açıksa sistem hedef sunucuya bağlanır ve sorgunun hangi kolonları döndüreceğini sorar.

Bu işlem sorguyu çalıştırmaz. Yalnızca şema bilgisi istenir, tek satır veri okunmaz, hedefte hiçbir yan etki oluşmaz.

Neden önemlidir: Önizleme, takma ad kullanılan kolonların gerçek kaynağını öğrenir.

```
SELECT TCKN AS MusteriKodu FROM Musteri
```

Bu sorguda dönen kolonun adı MusteriKodu'dur. Kolon adına bakan bir katalog bu kolonu yakalayamaz. Önizleme kaynak kolonun TCKN olduğunu öğrenir ve bu bilgi çalıştırma anında geri bağlanır. Böylece takma adla maskeleme atlatılamaz.

Önizleme sonucu kalıcı olarak saklanır. Böylece onaylayan kişi, talep sahibinin gördüğü listenin aynısını görür ve denetim dosyasında "beklenen kolonlar" ile "gerçekleşen maskeleme" karşılaştırılabilir.

Sunucu cevap veremezse talep yine kaydedilir; önizleme zorunlu değildir.

### E-posta onay adımı

QrEmailApprovalPolicy RequireApproval ise, alıcı listesindeki adresler kontrol edilir. Daha önce onaylanmamış bir adres varsa talebe ek bir onay adımı eklenir ve bu adımı QrEmailApproverRole rolündeki bir kullanıcı kapatır.

QrEmailAutoApproveUsers True ise sistemde kayıtlı kullanıcılara ait adresler onaylı sayılır.

### Maskesiz kolon onay adımı

UnmaskedColumnRequestPolicy değerine göre:

| **Politika** | **Davranış** |
|---|---|
| RequireApproval | Ek onay adımı eklenir. UnmaskedColumnApproverRole rolündeki kişi onaylar. |
| AllowWithReason | Ek adım yoktur, talep sahibi kendi adına otomatik onaylanmış sayılır. |
| Disabled | Özellik kapalıdır. Kaydedilmiş liste temizlenir. |

QrMaskingEnabled kapalıysa politika Disabled gibi davranır.

### 13.4 Çalıştırma anında arka planda ne oluyor

Bu, kılavuzun en çok merak edilen akışıdır. Adım adım:

```
 1. Talep calistirilir (elle veya zamanlanmis)
 2. Talebin TUM hedef sunuculari sirayla dolasilir
 3. Her sunucuda:
      a. Baglanti acilir, transaction baslatilir
      b. Betikler blok blok calistirilir, sonuc kumeleri okunur
      c. Kolon meta bilgisi eklenir
      d. Kayit anindaki KOLON ONIZLEMESI geri baglanir
         (takma adin kaynak kolonu buradan gelir)
      e. Satirlar bellege alinir
 4. Tum sunuculardan veri toplandiktan sonra:
      MASKELEME KARARI verilir
 5. Karar her sunucu icin AYRI verilir
      (prod maskelenirken test acik kalabilir)
 6. Maskeleme kaydi yazilir (her sunucu icin bir satir)
      Politika kapali olsa bile kayit YAZILIR
 7. Sonuclar Excel dosyasina donusturulur
 8. Dosya sifreli ZIP icine konur, parola uretilir
 9. Parola sifrelenerek talep kaydina yazilir
10. E-posta alicilari varsa gonderim kuyruguna eklenir
11. Calistirma gunlugu yazilir:
      kac satir dondu, hangi kolonlar maskelendi,
      hangi kolonlar acik gitti, sure butcesi asildi mi
12. Talep Executed durumuna gecer
```

#### Maskeleme kararı nasıl verilir

```
 QrMaskingEnabled kapali mi?
      evet -> Hicbir tespit ve maskeleme yapilmaz.
              Kayit YINE yazilir, uygulanan politika = MaskingOff.
              Denetcinin bu durumu gorebildigi tek yer burasidir.
      hayir -> devam
```

```
        |
        v
QrSensitiveDataPolicy = WarnOnly mi?
     evet -> Tespit calisir, hicbir sey maskelenmez.
             Bulunan kolonlar "acik gitti" olarak kayda yazilir.
     hayir -> AutoMask (bilinmeyen deger de AutoMask sayilir)
        |
        v
Her sunucu icin, her kolon icin:
        |
        v
Sunucunun rejimi "Yalniz icerik" mi?
     evet -> kolon adi katalogu ATLANIR, yalniz desen kontrolu yapilir
     hayir -> hem katalog hem desen kontrolu
        |
        v
Kolon hassas mi?
     hayir -> dokunulmaz
     evet -> devam
        |
        v
Kullanici bu kolonu maskesiz istedi ve ONAYLANDI mi?
     evet -> acik birakilir, gerekce "UnmaskApproved" olarak yazilir
     hayir -> MASKELENIR, gerekce "Catalog" veya "Pattern" olarak yazilir
```

Süre bütçesi aşılırsa taranamayan kolonlar "taranmadı" olarak işaretlenir ve maskelenir. Güvenli taraf her zaman kazanır.

### Kayıt ne tutar

İki tablo yazılır:

Özet kayıt (sunucu başına bir satır): talep numarası, sunucu adı, gönderim zamanı, gönderen, maskelenen kolonlar, maskesiz kolonlar, eşleşen desenler, uygulanan politika, teslim anındaki sunucu rejimi, muafiyet gerekçesi, tespit ayarları, istisna onayı var mı.

Ayrıntı kayıt (kolon başına bir satır): kolon adı, kaynak kolon adı, karar (maskelendi / maskelenmedi), gerekçe tipi, gerekçe ayrıntısı, uygulanan maskeleme biçimi.

Gerekçe tipleri:

| Tip | Anlamı |
|---|---|
| Catalog | Kolon adı hassas kolon kataloğuyla eşleşti. |
| Pattern | Kolon değerleri bir hassas veri desenine uydu. |
| UnmaskApproved | Kullanıcı istedi ve onaylandı, açık bırakıldı. |
| NotScanned | Süre bütçesi nedeniyle taranamadı, güvenli tarafta maskelendi. |
| PolicyWarnOnly | Politika WarnOnly olduğu için maskelenmedi. |
| MaskingOff | Ana anahtar kapalı olduğu için hiç tespit yapılmadı. |

Tespit ayarlarının kayda dondurulması önemlidir. Eşik veya örneklem sonradan değiştirilirse "bu teslimde eşik neydi" sorusu başka hiçbir yerden cevaplanamaz.

### 13.5 Sonuç dosyası ve teslim

| Konu | Davranış |
|---|---|
| Dosya biçimi | Şifreli ZIP içinde Excel dosyası. |
| Parola | Her teslim için ayrı üretilir. |
| Parola nerede | Talep kaydında şifreli saklanır, yetkili kullanıcıya gösterilir. |
| E-posta | Alıcı varsa gönderilir. Dosya MaxMailAttachmentMB sınırını aşarsa eklenmez. |
| İndirme | Talep detayından. |
| Saklama süresi | QrFileRetentionDays gün. Süre dolunca bakım görevi dosyayı siler ve silme kayda geçer. |

### İndirme yetkisi

```
Indirmek isteyen kisi talebin sahibi mi?
     evet -> QrRequesterCanDownload true mu?
               evet -> indirebilir
               hayir -> indiremez
     hayir -> Rolu QrDownloadMinimumRole esigini geciyor mu?
               evet -> indirebilir
```

```
hayir -> indiremez
```

Her indirme denetim kaydına yazılır ve Veri Erişim ve Dışa Aktarım raporunda görünür.

### 13.6 Bu akıştaki özel kurallar

| **Kural** | **Sebep** |
|---|---|
| Sorgu sonucu talepleri sürüm paketi içinde çalıştırılamaz. | Paket akışında maskeleme ve kayıt üretilmez; maskelenmemiş veri kayıtsız dışarı çıkardı. |
| Hedefteki transaction sorgu sonucu taleplerinde commit edilmez. | Okuma işlemidir, hedefte değişiklik bırakmaz. |
| Bir sunucuda hata olursa tüm teslim başarısız sayılır. | Eksik sonuç teslim etmek yanıltıcı olurdu. |
| Aday kolon yazımı teslimi engellemez. | Öğrenme döngüsünün hatası veri teslimini durdurmamalıdır; hata günlüğe yazılır. |

## Bölüm 14. Sandbox

### 14.1 Ne işe yarar

Sandbox, bir değişiklik talebini gerçek bir veritabanında denemek ama kalıcı hale getirmemektir. Betik bir transaction içinde çalıştırılır, sonuç kaydedilir ve transaction geri alınır.

Amaç, betiğin canlıya girmeden önce gerçekten çalışıp çalışmadığını görmektir. Sözdizimi hataları, eksik nesneler ve izin sorunları burada yakalanır.

Bu özellik Sandbox modülüne bağlıdır.

### 14.2 Nasıl çalışır

```
Kullanici "Sandbox" der ve bir sunucu secer
       |
       v
Sandbox kaydi olusturulur (isleniyor durumunda)
       |
       v
Worker bekleyen sandbox kayitlarini alir
       |
       v
Hedefte transaction acilir
       |
       v
Betikler blok blok calistirilir, gunluk toplanir
       |
       v
Transaction HER DURUMDA geri alinir
       |
       v
Sonuc kaydedilir: basarili / basarisiz + gunluk
```

### 14.3 Sonucun akışa etkisi

| Durum | Etki |
|---|---|
| Sandbox başarılı | Talep normal akışına devam eder. Sandbox Test Sonuçları raporunda kanıt olarak görünür. |

| Durum | Etki |
|---|---|
| Sandbox başarısız | Talep engellenmez, ancak sonuç detayda ve raporda görünür. Onaylayanın bunu görmesi beklenir. |
| Sandbox sonrası talep düzenlendi | Sandbox sonucu "eskimiş" olarak işaretlenir. Eskimiş bir test güncel betikleri kapsamaz. |

### 14.4 Bilinen sınırlar

Oracle hedeflerde DDL içeren talepler sandbox'lanamaz. Sebebi şudur: Oracle'da CREATE, ALTER, DROP, GRANT, REVOKE ve TRUNCATE gibi ifadeler örtük olarak kaydedilir (implicit commit). Sandbox'ın sonundaki geri alma bunları geri almaz, yani "deneme" kalıcı bir değişiklik bırakırdı.

Bu durumda sandbox denemesi başlatıldığında engellenir ve hangi ifade tiplerinin engele yol açtığı mesajda listelenir.

Yalnızca veri işleyen (INSERT, UPDATE, DELETE, MERGE, SELECT) Oracle betikleri güvenle sandbox'lanabilir.

SQL Server ve PostgreSQL hedeflerde DDL de dahil her şey geri alınabilir.

## Bölüm 15. Sürüm Paketi

### 15.1 Ne işe yarar

Birbirine bağlı birden çok talebin doğru sırayla ve birlikte çalıştırılmasını sağlar. Örnek: önce tablo eklenmeli, sonra prosedür güncellenmeli, sonra veri taşınmalı.

Bu özellik ReleasePackages modülüne bağlıdır.

### 15.2 Paket alanları

| **Alan** | **Ne işe yarar** |
|---|---|
| Ad | Paketin adı. |
| Açıklama | Serbest açıklama. |
| Sunucu | Paketin çalışacağı sunucu. |
| Çalıştırma Modu | Bkz. 15.3. |
| Zamanlama | Paketin çalışacağı tarih. |
| Sorgu Zaman Aşımı | Üst sınır MaxQueryTimeout parametresidir. |
| Durum | Hazırlanıyor, kuyrukta, çalıştırıldı. |
| Başarı | Çalıştırma başarılı mı. |

### 15.3 Çalıştırma modları

| **Değer** | **Mod** | **Davranış** |
|---|---|---|
| 1 | AllOrNothing | Tüm talepler tek bir transaction içinde çalışır. Biri başarısız olursa hepsi geri alınır. |
| 2 | StopOnError | Her talep kendi transaction'ında çalışır. Hata olduğunda paket durur, o ana kadar başarılı olanlar kalır. |
| 3 | ContinueOnError | Her talep kendi transaction'ında çalışır. Hata olduğunda o talep ve ona bağımlı olanlar atlanır, kalanlar devam eder. |

Hangi modu seçmeli:

- Değişiklikler birbirinden ayrılamaz bir bütün oluşturuyorsa AllOrNothing.

- Sıra önemliyse ve ilk hatada durup incelemek istiyorsanız StopOnError.

- Talepler büyük ölçüde bağımsızsa ve bir hatanın tüm sürümü durdurmasını istemiyorsanız ContinueOnError.

### 15.4 Paket içeriği ve sıralama

| **Alan** | **Ne işe yarar** |
|---|---|
| Sıra No | Taleplerin çalıştırılma sırası. |
| Bağımlı Olduğu Talep | Bu talep hangi talebin başarılı olmasına bağlı. ContinueOnError modunda atlama zinciri bu alandan çıkar. |
| Çalıştırıldı | Bu kalem çalıştı mı. |
| Çalıştırma Açıklaması | Kalem bazında sonuç. |

### 15.5 Tekil talep akışından farkları

| **Konu** | **Tekil talep** | **Sürüm paketi** |
|---|---|---|
| Transaction | Talep başına bir transaction. | Moda göre tek veya talep başına. |
| Onay | Talep bazında. | Talepler ayrı ayrı onaylanmış olmalıdır. |
| Talep üzerinde işlem | Serbest. | Aktif pakete alınmış talepte düzenleme, onay, çalıştırma ve zamanlama düğmeleri pasiftir. Talep artık paketin denetimindedir. |
| Otomatik zamanlama | Yapılabilir. | Aktif pakete alınmış talep için yapılmaz. |

### 15.6 Kısıtlar

- Sorgu sonucu talepleri pakete alınamaz. Maskeleme ve teslim kaydı yalnızca kendi akışında üretilir.

- Bir talep aynı anda yalnızca bir aktif pakette bulunabilir.

- Paket çalıştırma zamanı da MainWorkJob tarafından denetlenir; zamanı gelen paketler önce çalıştırılır, sonra tekil talepler işlenir.

## Bölüm 16. Nesne Geçmişi (DDL History)

### 16.1 Ne işe yarar

Hedef sunuculardaki nesnelerin ne zaman ve nasıl değiştiğini gösterir. Veri, SQL Change Guard kayıtlarından değil, hedef veritabanının kendisinden okunur.

Bu özellik InsightsReporting modülüne bağlıdır.

### 16.2 Ekran akışı

```
Sunucu sec
    |
    v
Veritabani sec
    |
    v
Nesne ara / listele
    |
    v
Nesne sec -> gecmisini gor
```

| Adım | Ne gösterir |
|---|---|
| Sunucu listesi | Tanımlı ve aktif sunucular. |
| Veritabanı listesi | Seçilen sunucudaki veritabanları. |
| Nesne özeti | Nesne adı, şema, tip, son değişiklik tarihi. Arama kutusu ile filtrelenir. |
| Nesne geçmişi | Seçilen nesnenin sürümleri ve değişiklik zamanları. |

### 16.3 Denetim değeri

Bu ekranın asıl faydası, talep dışı değişiklikleri görünür kılmasıdır.

Bir nesne SQL Change Guard üzerinden değiştirildiyse, ilgili talep kaydı vardır. Nesne geçmişinde bir değişiklik görünüyor ama bu tarihte hiçbir talep yoksa, o değişiklik sistemin dışından yapılmış demektir.

Denetimde sorulacak soru: "Bu değişikliği kim, hangi yetkiyle, kayıt dışı yaptı?"

Bu karşılaştırmayı yaparken Nesne Değişiklik Sıklığı ve Yapılandırma Değişiklikleri raporları da kullanılır.

## Bölüm 17. Denetim Kaydı ve Değiştirilemezlik

Ekran: Denetim Kaydı

Yetki: audit.view

### 17.1 Denetim kaydında ne tutulur

Her denetim satırı şu alanları taşır:

| **Alan** | **Ne gösterir** |
|---|---|
| Olay (Action) | Ne oldu. Örnek: ChangeRequest.Approved. |
| Kayıt Tipi (EntityType) | Hangi tür kayıt etkilendi. Örnek: ChangeRequest, Server, SettingChange. |
| Kayıt Kimliği (EntityId) | Etkilenen kaydın numarası. |
| Eski Değer | Değişiklikten önceki durum. |
| Yeni Değer | Değişiklikten sonraki durum. |
| Yapan (PerformedBy) | Kullanıcı adı veya sistem aktörü. |
| IP Adresi | İsteğin geldiği adres. |
| Zaman (OccurredAt) | Olay zamanı, UTC olarak saklanır. |
| Özet (Hash) | Kaydın imzası. |
| Önceki Özet (PreviousHash) | Bir önceki kaydın imzası. |
| Sürüm (HashVersion) | 1, 2 veya 3. Bkz. Bölüm 5.4. |
| Anahtar Kimliği (KeyId) | Sürüm 3 kayıtlarda hangi anahtarın kullanıldığı. |

### 17.2 Ne tutulmaz

- Sorgu sonucu verisinin kendisi hiçbir zaman denetim kaydına girmez.

- Şifreler ve parolalar maskelenmiş olarak yazılır.

- Kolon içerikleri ve satır değerleri tutulmaz.

### 17.3 Olay adlandırma kuralı

Olay adları KayıtTipi.Eylem biçimindedir:

| **Örnek olay** | **Anlamı** |
|---|---|
| ChangeRequest.Create | Talep oluşturuldu. |
| ChangeRequest.Approve | Talep onaylandı. |

| Örnek olay | Anlamı |
|---|---|
| ChangeRequest.AutoApproved | Onay adımı oluşmadığı için sistem otomatik onayladı. |
| ChangeRequest.Reject | Talep reddedildi. |
| ChangeRequest.Executed | Talep çalıştırıldı. |
| SettingChange.Requested | Ayar değişikliği onaya gönderildi. |
| SettingChange.Approved | Ayar değişikliği onaylandı ve uygulandı. |
| SettingChange.Rejected | Ayar değişikliği reddedildi. |
| SettingChange.ApplyFailed | Onaylandı ama uygulanamadı. |
| User.SessionsRevoked | Kullanıcının oturum yenileme kayıtları iptal edildi. |
| Role.SessionsRevoked | Rolü taşıyan kullanıcıların oturum yenileme kayıtları iptal edildi. |
| EvidencePackageExported | Denetim dosyası dışa aktarıldı. |

Kontrol zayıflatma olayları ayrı adla kaydedilir. Sıradan bir ayar güncellemesi ile bir kontrolün gevşetilmesi aynı olay adıyla yazılmaz. Denetçi, kontrolün ne zaman zayıflatıldığını tek bir olay adıyla arayabilir.

### 17.4 Eski ve yeni değer sözleşmesi

İki kural vardır:

1. Eski değer ve yeni değer aynı biçimde yazılır. Farklı biçimlerde yazılsalardı yan yana karşılaştırılamazlardı.

2. Yazma işleminden sonra değer veritabanından geri okunur. Böylece kayda giren değer, uygulamanın niyeti değil, veritabanının gerçeği olur.

Reddedilen ayar değişikliklerinde hiçbir işlem uygulanmaz. Bu durumda istenen değerler denetim satırında saklanır; başka hiçbir yerde iz kalmaz.

### 17.5 Kural seti anlık görüntüsü ("o gün neydi")

Bir talep oluştuğunda, o anda geçerli olan kural seti talebe dondurulur:

- Uygulanan onay politikasının kimliği ve adı.

- Etkin görevler ayrılığı ayarları.

- Kritik nesne kararları.

- Tetiklenen standartlar ve dereceleri.

Politika sonradan silinse veya değiştirilse bile talebin denetim izi bozulmaz. Denetçi "bu talep hangi kurallarla değerlendirildi" sorusunu tek bir kayıttan cevaplayabilir.

### 17.6 Hash zinciri ve doğrulama

Zincirin nasıl çalıştığı Bölüm 5.4'te anlatıldı. Bu bölüm doğrulama araçlarını anlatır.

#### Zincir Doğrula

Denetim Kaydı ekranındaki bu işlem, tüm zinciri baştan sona kontrol eder:

| **Kontrol** | **Ne bakar** |
|---|---|
| Kayıt içi tutarlılık | Her kaydın özeti kendi içeriğiyle uyuşuyor mu. |
| Zincir bağlantısı | Her kaydın "önceki özet" alanı gerçekten bir öncekinin özeti mi. |
| Anahtar bulunabilirliği | Sürüm 3 kayıtların anahtarı halkada var mı. |

Sonuçta bozuk kayıtların numaraları listelenir.

#### Immutable Doğrula

Bu işlem, ana veritabanındaki kayıtları ayrı veritabanındaki mühürlü kopya ile karşılaştırır.

Neden gereklidir: HMAC anahtarını ele geçiren bir saldırgan ana veritabanındaki zinciri baştan sona yeniden yazabilir ve "Zincir Doğrula" temiz sonuç verebilir. Ayrı veritabanındaki kopya onun erişimi dışındadır.

Dönen sonuçlar:

| **Alan** | **Anlamı** |
|---|---|
| Etkin mi | Immutable katman açık mı. |
| Toplam denetim kaydı | Ana veritabanındaki satır sayısı. |
| Koruma başlangıcı | Kopyalamanın başladığı ilk kayıt numarası. Bu numaradan önceki kayıtlar korumasızdır. |
| Korunan sayısı | Koruma kapsamındaki kayıt sayısı. |
| Korumasız sayısı | Koruma başlamadan önceki kayıt sayısı. |
| Eşleşen | Kopya ile birebir aynı olan kayıt sayısı. |
| Eşleşmeyen | Kopyadan farklı özete sahip kayıtlar. |
| Kopyası eksik | Ana veritabanında var, kopyada yok. |
| Sahipsiz kopya | Kopyada var, ana veritabanında yok. |

Sonuçların yorumu:

| Bulgu | Anlamı | Aciliyet |
|---|---|---|
| Eşleşmeyen sıfır | Kriptografik bütünlük sağlam. | - |
| Eşleşmeyen sıfırdan büyük | Manipülasyon kanıtı. Ekleme-yalnız bir denetim satırının özeti meşru olarak asla değişmez. | Yüksek |
| Kopyası eksik | Meşru sebebi olabilir: o anda ikinci veritabanı erişilemezdi ve kopya yazılamadı. Anomali olarak incelenir. | Orta |
| Sahipsiz kopya | Meşru sebebi olabilir: denetim yazan işlem geri alındı, kopya ayrı bağlantıda kaydedildiği için kaldı. | Düşük |

Eksik ve sahipsiz kayıtlar bütünlük bayrağını etkilemez, çünkü yanlış alarm üretirlerdi. Yalnızca eşleşmeyen kayıt tek başına yeterli kanıttır.

Mevcut sınır: Immutable doğrulayıcı şu an yalnızca SQL Server için uygulanmıştır. Uygulama veritabanı PostgreSQL veya Oracle üzerinde çalışıyorsa bu işlem "devre dışı" sonucu döndürür. Bu bilinçli bir durum bildirimi olup hata değildir.

### Betik Bütünlüğü Doğrula

Belirli bir talebin betiklerinin kaydedildikten sonra değişmediğini kontrol eder. Talep detayından çalıştırılır.

## 17.7 Denetim bütünlüğü işi

AuditIntegrityJob worker içinde düzenli aralıklarla çalışır ve zinciri kontrol eder. Bozulma bulursa kritik seviyede bildirim üretir. Böylece manipülasyon bir insanın ekranı açmasını beklemeden fark edilir.

## 17.8 Denetim Kaydı ekranı

| Kolon | Ne gösterir |
|---|---|
| Zaman | Olay zamanı, tarayıcınızın yerel saatiyle. |
| Olay | Olay adı. |
| Kayıt Tipi | Etkilenen kayıt türü. |
| Kayıt No | Etkilenen kaydın numarası. |
| Yapan | Kullanıcı veya sistem aktörü. |

| Kolon | Ne gösterir |
|---|---|
| IP | İstek adresi. |
| Eski Değer / Yeni Değer | Satır açıldığında yan yana gösterilir. |
| Doğrulama | Kaydın imzası geçerli mi. |

Filtreler: tarih aralığı, olay adı, kayıt tipi, kayıt numarası, kullanıcı.

## Bölüm 18. Denetim Dosyası (Evidence Dossier)

### 18.1 Ne işe yarar

Tek bir talebin tüm yaşam döngüsü kanıtını, denetçiye verilebilecek biçimde tek bir ZIP dosyasında toplar.

Kullanım senaryosu: Denetçi "şu değişikliğin kanıtını göster" der. Talep detayından dosya dışa aktarılır ve denetçiye verilir. Denetçi dosyayı sisteme girmeden inceleyebilir ve doğrulayabilir.

### 18.2 Kim üretir

Dışa aktarma için iki katman yetki gerekir:

1. audit.evidenceexport yetkisi. Bu yetki rol tanımından gelir. Varsayılan olarak Auditor, DbAdmin, ChangeManager ve SystemManager rollerinde açıktır.

2. Talebi görme yetkisi. cr.seeall yetkisi yoksa kullanıcı yalnızca kendi taleplerinin dosyasını üretebilir.

İki katmanın sebebi: dışa aktarma yetkisi, görmediğiniz bir talebin verisine erişim sağlamamalıdır.

### 18.3 Dosya içeriği

| Dosya | İçerik |
|---|---|
| Dossier.json | Kanonik veri. Talebin künyesi, onay adımları, betikler, risk özeti, sandbox sonuçları, denetim izi. Paket özeti bu dosya üzerinden hesaplanır. |
| Manifest.json | Paket özeti (packageHash) ve denetim kaydı numarası. Doğrulama bu dosyadaki iki değerle yapılır. |
| Dossier.pdf | Aynı bilginin okunabilir, yazdırılabilir hali. |

Denetim izi en fazla 500 kayıtla sınırlıdır; çok uzun geçmişi olan talepler için en ilgili kayıtlar alınır.

### 18.4 Sorgu sonucu verisi neden girmez

Denetim dosyasına sorgu sonucu verisi hiçbir zaman konmaz.

Sebebi şudur: denetim dosyası, kontrol mekanizmasının çalıştığını kanıtlamak içindir, verinin kendisini taşımak için değil. Sonuç verisini dosyaya koymak, maskeleme ve indirme yetkisi kontrollerini tümüyle atlatan bir yan kapı açardı. Denetim dosyası indirme yetkisi olan biri, sonuç dosyası indirme yetkisi olmadan canlı veriye ulaşırdı.

Dosyada bulunan: sorgunun metni, kaç satır döndüğü, hangi kolonların maskelendiği, hangi kolonların açık gittiği ve neden.

Dosyada bulunmayan: satır içeriklerinin kendisi.

### 18.5 Dışa aktarmanın kendisi denetlenir

Dosya üretildiğinde denetim kaydına EvidencePackageExported olayı yazılır. Bu kayıt:

- Kimin, ne zaman, hangi talep için dosya ürettiğini tutar.

- Paket özetini (packageHash) taşır.

- Veri Erişim ve Dışa Aktarım raporunda görünür.

Yani "kanıt üretme" işleminin kendisi de kanıtlanır.

### 18.6 Dosya nasıl doğrulanır

```
1. Denetci elindeki ZIP dosyasini acar
2. Manifest.json icinden iki degeri okur:
     - packageHash
     - auditEntryId
3. Denetim Kaydi ekranindaki "Kanit Dogrula" islemini acar
4. Iki degeri girer
5. Sistem, o numaradaki denetim kaydini bulur
6. Kaydin icindeki packageHash ile girilen degeri karsilastirir
```

Dönen sonuç:

| **Alan** | **Anlamı** |
|---|---|
| Geçerli mi | İki özet aynı mı. |
| Denetim kaydı numarası | Karşılaştırılan kayıt. |
| Dışa aktarma zamanı | Dosyanın üretildiği an. |
| Dışa aktaran | Dosyayı üreten kullanıcı. |
| Beklenen özet | Denetim kaydındaki değer. |
| Verilen özet | Denetçinin girdiği değer. |

### Doğrulama başarısız olursa ne anlama gelir

| **Durum** | **Anlamı** |
|---|---|
| Kayıt bulunamadı | Verilen numarada bir dışa aktarma kaydı yok. Dosya bu sistemden çıkmamış olabilir veya numara yanlış girilmiştir. |
| Özetler farklı | Dosya üretildikten sonra değiştirilmiş. İçeriğe güvenilemez. |

| Durum | Anlamı |
|---|---|
| Özetler aynı | Dosya üretildiği andaki halindedir. |

Doğrulama, dosyanın içeriğinin doğru olduğunu değil, değişmediğini kanıtlar. İçeriğin doğruluğu, denetim kaydı zincirinin bütünlüğünden gelir (Bölüm 17.6).

## 18.7 Denetçi bu dosyayla neyi ispatlayabilir

| Soru | Dosyadaki cevap |
|---|---|
| Bu değişikliği kim istedi? | Talep künyesi. |
| Ne değiştirdi? | Betikler ve ayrıştırılmış nesne listesi. |
| Riski neydi? | Risk bandı, skor, bulgu listesi. |
| Hangi kurala göre değerlendirildi? | Uygulanan politika adı ve dondurulmuş görevler ayrılığı ayarları. |
| Kim onayladı? | Onay adımları, karar verenler, zamanlar. |
| Atlanan adım var mı, neden? | Atlanan adımlar gerekçeleriyle. |
| Test edildi mi? | Sandbox sonuçları. |
| Ne zaman ve kim tarafından çalıştırıldı? | Çalıştırmayı başlatan kişi, çalıştıran servis, zamanlar, sonuç. |
| Bu belge sonradan değiştirildi mi? | Paket özeti doğrulaması. |

## Bölüm 19. Kontrol Paneli (Dashboard)

Ekran: Kontrol Paneli

Yetki: dashboard.view

### 19.1 Zaman ayrımı

Paneldeki sayılar iki gruba ayrılır ve ekranda da ayrı gösterilir:

| Grup | Anlamı |
|---|---|
| Aralık (Range) | Seçilen tarih aralığında olan işlerin sayısı. Geçmişe bakar. |
| Şimdi (Now) | Şu an bekleyen işlerin sayısı. Tarih aralığından bağımsızdır. |

Bu ayrım önemlidir. İkisi tek bir kart dizisinde karıştırılırsa "toplam 500, bekleyen 12" gibi birbiriyle ilgisiz sayılar yan yana gelir ve panel yanlış okunur.

### 19.2 Aralık kartları

| Kart | Ne gösterir | Nasıl hesaplanır |
|---|---|---|
| Oluşturulan | Aralıkta açılan talep sayısı. | Oluşturma tarihine göre. |
| Çalıştırılan | Aralıkta başarıyla çalıştırılan talep sayısı. | Çalıştırma tarihine göre. |
| Başarısız | Aralıkta çalıştırma denemesi başarısız olan talep sayısı. | |
| Reddedilen | Aralıkta reddedilen talep sayısı. | |

### 19.3 Şimdi kartları

| Kart | Ne gösterir |
|---|---|
| Onay bekleyen | Şu an Pending durumundaki talepler. |
| Çalıştırma bekleyen | Onaylanmış ama henüz çalıştırılmamış talepler. |
| Çalışan | Şu an Executing durumundaki talepler. |
| Önümüzdeki 24 saatte zamanlanmış | Zamanlaması bugün ve yarın arasında olan talepler. |
| Bekleyen ayar onayı | Karar bekleyen ayar değişiklikleri. |

### 19.4 Çalıştırma başarı oranı

Formül: başarılı / (başarılı + başarısız).

Reddedilen talepler bu hesaba dahil değildir, çünkü reddedilen talep hiç çalıştırılmamıştır ve başarısızlık sayılmaz.

Aralıkta hiç çalıştırma denenmediyse kart boş görünür. Sıfır yazmak "hepsi başarısız oldu" gibi okunurdu.

### 19.5 Grafikler ve listeler

| **Bölüm** | **Ne gösterir** | **Nasıl okunur** |
|---|---|---|
| Eğilim (Trend) | Zaman içinde oluşturulan ve çalıştırılan talep sayısı. | İki çizgi arasındaki fark birikmiş iş yükünü gösterir. Oluşturma sürekli çalıştırmanın üstündeyse kuyruk büyüyor demektir. |
| En aktif kullanıcılar | Aralıkta en çok talep açan kullanıcılar. | Tek bir kişide aşırı yoğunlaşma, bilgi ve yetki yoğunlaşmasına işaret edebilir. |
| Sunucu yükü | Sunucu başına talep sayısı. | Hangi ortamın en çok değişiklik aldığını gösterir. |
| En uzun bekleyenler | En eski bekleyen talepler. | Talep numarası, açıklama, bilet, açan kişi, açılış zamanı, durum, hangi adımda ve hangi rolde beklediği. |
| Risk dağılımı | Bandlara göre talep sayısı ve yüzdesi. | Band boş (hesaplanmamış) kayıtlar ayrı görünür. Critical dilimin büyümesi dikkat gerektirir. |
| Onay kuyruğu | Rol bazında bekleyen adım sayısı ve ortalama bekleme süresi. | Hangi rolün darboğaz olduğunu gösterir. Ortalama bekleme saatinin yükselmesi o rolde yeterli kişi olmadığına işaret eder. |

### 19.6 İstisnalar kartı

Normal akışın dışına çıkan taleplerin sayısıdır. Denetçi ve yönetim için "kontrol çalışıyor mu" sorusunun panel karşılığıdır.

| **Sayaç** | **Ne sayar** |
|---|---|
| Acil durum | Aralıkta açılan acil durum talepleri. |
| Geçersiz kılınan adımlar | Acil durum yetkisiyle geçilen onay adımları. |
| Kendi kendine onaylanan | Talep sahibinin kendi talebini onayladığı durumlar. |
| Eşleşen politika yok | Hiçbir onay politikasının eşleşmediği talepler. |

| Sayaç | Ne sayar |
|---|---|
| Onaysız maskesiz veri | Maskesiz kolon istenmiş ama onay alınmamış talepler. |

Bu sayılardan herhangi biri beklenenden yüksekse ayrıntısı Kontrol İstisnaları raporundadır.

### 19.7 Rol bazında farklar

Panel, kullanıcının yetkisine göre daralır. cr.seeall yetkisi olmayan kullanıcı yalnızca kendi taleplerine ait sayıları görür. Ayar onayı sayacı, ilgili yetkisi olmayan kullanıcıda görünmez.

### 19.8 Boş veri durumu

Seçilen aralıkta hiç kayıt yoksa kartlar sıfır, grafikler boş durum mesajıyla görünür. Bu bir hata değildir.

## Bölüm 20. Raporlar

Ekran: Raporlar

Yetki: report.view

Modül: InsightsReporting

### 20.1 Genel kurallar

- Raporlar beş grupta toplanmıştır ve toplam 30 rapor vardır.

- Rapor çıktıları İngilizce üretilir. Sebebi, raporların denetim ve uyum çalışmalarında kurum dışına verilebilmesi ve uluslararası denetçilerin de okuyabilmesidir.

- Her raporun kendi parametreleri vardır. Tarih aralığı parametreleri varsayılan olarak son 30 günü kapsar.

- Bazı raporlar rol kısıtlıdır. Kısıt hem listede hem çalıştırmada arka uçta uygulanır.

- Rapor sorguları zaman aşımına tabidir (Reports:QueryTimeoutSeconds, varsayılan 500 saniye).

### 20.2 Ortak parametreler

| **Parametre** | **Tip** | **Anlamı** |
|---|---|---|
| Start Date | Tarih | Aralığın başlangıcı. Varsayılan: 30 gün önce. |
| End Date | Tarih | Aralığın bitişi. Varsayılan: bugün. |
| Server | Liste | Sunucu filtresi. Çoklu sunucu taleplerinde liste içinde arama yapılır. |
| Status | Liste | Talep durumu. |
| Object Name | Metin | Nesne adı filtresi. |
| ITSM Ticket No | Metin | Bilet numarası. |
| Request No | Sayı | Talep numarası. |
| Minimum Risk Band | Liste | Risk bandı eşiği. Varsayılan: 3 (High). |
| Statement Category | Liste | DDL, DML, DCL gibi. |
| Statement Type | Liste | CREATE, ALTER, DROP gibi. |
| Environment | Liste | Ortam. |
| Database Type | Liste | Veritabanı türü. |
| Principal / Account | Metin | Veritabanı hesabı. |
| User | Metin | Uygulama kullanıcısı. |
| Record Type | Metin | Denetim kaydı tipi. |

| Parametre | Tip | Anlamı |
|---|---|---|
| Include deleted accounts | Onay kutusu | Silinmiş hesaplar dahil edilsin mi. |

Tarih filtreleri yerel gün sınırıyla çalışır. Ayrıntı: Bölüm 23.4.

### 20.3 Change Governance grubu

| Rapor | Neyi gösterir | Hangi soruyu cevaplar | Kısıt |
|---|---|---|---|
| Change Request Summary | Talep başına tek satır: durum, risk bandı, uygulanan politika, acil durum işareti, çalıştırmayı başlatan kişi, etkilenen satır sayısı. | "Bu dönemde hangi değişiklikler yapıldı?" | - |
| Change Request by Object | Hangi talep hangi nesneye dokundu. | "Şu tabloyu son altı ayda kim değiştirdi?" | - |
| Work in Progress | Henüz çalıştırılmamış talepler ve kaç gündür bekledikleri. | "Kuyrukta ne var, ne kadar bekliyor?" | - |
| Rejected and Cancelled Requests | Reddedilen veya iptal edilen talepler, kararı veren kişi ve gerekçe. Karar veren bilgisi onay adımından okunur, sonraki otomatik güncellemeler bunu ezemez. | "Neden reddedildi, kim reddetti?" | - |
| High Risk Changes | Seçilen bandın üstündeki talepler, acil durum işareti ve uygulanan politika. Skor ikincil değer olarak gösterilir. | "En riskli değişiklikler hangileriydi?" | - |
| Rollback Changes | Geri alma talepleri ve kaynak talepleri. | "Hangi değişiklikler geri alındı?" | - |
| Approval and Execution Duration | Talep, son onay ve çalıştırma arasındaki saat farkları. | "Süreç nerede tıkanıyor?" | - |
| Execution Activity | Hedef sunucularda gerçekte ne çalıştı: başlatan kişi, çalıştıran servis, süre, etkilenen satır, sonuç, hata metni. Otomatik zamanlamada başlatan SYSTEM görünür. Sürüm | "Canlıda ne çalıştı?" | - |

| Rapor | Neyi gösterir | Hangi soruyu cevaplar | Kısıt |
|---|---|---|---|
| | paketi içinde çalıştıysa paket de gösterilir. | | |
| Approval Step Detail | Her onay adımı: gereken rol, karar veren, karar zamanı, geçersiz kılma işareti ve gerekçe. | "Çift kontrol gerçekten uygulandı mı?" | - |
| Sandbox Test Results | Test sunucusu, sonuç, çalıştıran, testten sonra talep düzenlendi mi. Eskimiş test güncel betikleri kapsamaz. | "Canlıya çıkmadan önce denendi mi?" | - |
| Control Exceptions | Normal akıştan çıkan talepler: acil durum kayıtları, politikasız talepler, geçersiz kılınan adımlar, kendi kendine onaylananlar, onaysız maskesiz veri istekleri. | "Kontroller nerede devre dışı kaldı?" | Auditor, DbAdmin, ChangeManager, SystemManager |

Denetçiler genellikle Control Exceptions raporundan başlar. Bu rapor boşsa kontrol mekanizması istisnasız çalışmış demektir.

### 20.4 Object & Structure grubu

| Rapor | Neyi gösterir | Nasıl yorumlanır |
|---|---|---|
| Critical Object Access | Kritik işaretli nesneye dokunan talepler. Kritiklik, talep ayrıştırıldığında kaydedilen karardır; sonraki katalog düzenlemeleri geçmişi yeniden yazmaz. | Bu listedeki her satır ayrıca incelenmelidir. |
| Object Change Frequency | Prosedür, fonksiyon, tetikleyici, görünüm ve tabloların ne sıklıkta değiştiği. En sık değişenler üstte. | Üstteki nesneler mimari borç veya istikrarsızlık işaretidir. |
| Table Structure Changes | ALTER TABLE ADD ve DROP COLUMN işlemleri, etkilenen kolon adları ve sayıları. | Veri kaybı riski en yüksek işlemler buradadır. |
| Data Modification Frequency | Tablo bazında INSERT, UPDATE, DELETE, MERGE ve prosedür çağrılarının sıklığı, son çalıştırma tarihi. | Sık elle veri düzeltmesi alan tablolar süreç sorununa işaret eder. |
| Multi Release Objects | Birden çok bilet altında değişen nesneler. | Birkaç takımın aynı nesneyi düzenlediği sıcak noktaları gösterir. |

| Rapor | Neyi gösterir | Nasıl yorumlanır |
|---|---|---|
| Active Conflicts | Aynı ortamda, farklı biletlerle, aynı nesneyi değiştirmek üzere olan açık talepler. | Çakışmayı canlıya ulaşmadan yakalamak için kullanılır. |

### 20.5 Access & Authorization grubu

| Rapor | Neyi gösterir | Kısıt |
|---|---|---|
| Privileged Account Operations | Talepler içindeki tüm yetkilendirme ifadeleri: GRANT, REVOKE, DENY, hesap ve rol yönetimi, hedef hesap ve verilen yetkiler. | Auditor, DbAdmin, ChangeManager, SystemManager |
| User DDL Activity | Kullanıcı başına CREATE, ALTER, DROP ve yetkilendirme işlem sayıları, o kullanıcının kaç talebinin yüksek riskli veya acil durum olduğu. | - |
| User Permission Matrix | Her hesap: rolü, hiyerarşi seviyesi, hesap durumu, son giriş ve rolün tüm yetki seti. Erişim kontrolü kanıtı tek tabloda. | Auditor, DbAdmin, ChangeManager, SystemManager |
| User Access Changes | Hesap ve rol yaşam döngüsü: kim hesap açtı, kim yetki değiştirdi, kim sildi, canlandırdı veya kilidi açtı ve hangi adresten. | Auditor, DbAdmin, ChangeManager, SystemManager |

### 20.6 Data & Privacy grubu

| Rapor | Neyi gösterir | Kısıt |
|---|---|---|
| Query Result Activity | Canlı veri okuyan her talep: dönen satır sayısı, alıcı adresi, istenen maskesiz kolonlar ve teslimde uygulanan maskeleme kararı. | Auditor, DbAdmin, ChangeManager, SystemManager |
| Sensitive Column Touches | Hassas veri kataloğundaki bir kolonu ekleyen veya silen yapısal değişiklikler, kolonun sınıflandırması ve maskeleme biçimi. Tam ve içeren kuralları maskeleme motoruyla aynı biçimde uygulanır; şema veya tabloyla sınırlı bir kural yalnızca o kapsamda eşleşir. | - |

| Rapor | Neyi gösterir | Kısıt |
|---|---|---|
| | Düzenli ifade kuralları burada tam kolon adıyla eşleştirilir. | |
| Masking Activity | Her teslimde maskelemenin ne yaptığı: uygulanan politika ve rejim, kaç kolon maskelendi, kaç kolon açık gitti ve bu onaylı mıydı. Maskeleme kapalıyken yapılan teslimler MaskingOff politikasıyla görünür. | Auditor, DbAdmin, ChangeManager, SystemManager |
| Sensitive Column Catalog | Sınıflandırma kataloğunun ve öğrenme döngüsünün durumu: aktif kurallar, kapsamları, maskeleme biçimleri ve karar bekleyen aday kolonlar. Hiçbir veri değeri içermez. | - |
| Masking Exempt Servers | Maskeleme rejimi gevşetilmiş sunucular: hangi mod, yazılı gerekçe, kim ne zaman ayarladı ve o sunucudan kaç teslim yapıldı. | Auditor, DbAdmin, ChangeManager, SystemManager |
| Manual Data Modifications | Veriyi doğrudan değiştiren talepler (DML veya prosedür çağrısı). | - |
| Data Access and Export Log | Kim sonuç dosyası indirdi, hangi gerekçeyle ve hangi adresten, kim denetim dosyası dışa aktardı, hangi sonuç dosyası saklama süresi dolduğu için silindi, kim kısıtlı bir raporu açtı. Kişisel veri erişim izidir. | Auditor, DbAdmin, ChangeManager, SystemManager |

## 20.7 Configuration & Compliance grubu

| Rapor | Neyi gösterir | Kısıt |
|---|---|---|
| Configuration Changes | Kontrollerin kendisine yapılan her değişiklik: sunucular, parametreler, onay politikaları, roller, SQL standartları, kritik nesneler, hassas kolonlar, desenler, mail kaynakları ve lisans olayları, eski ve yeni değerleriyle. | Auditor, DbAdmin, ChangeManager, SystemManager |
| Settings Approval Trail | Ayarlarda çift kontrol kanıtı: kim değişiklik istedi, kim onayladı veya reddetti, karar ne kadar sürdü ve aynı kişi ikisini birden yaptı mı. | Auditor, DbAdmin, ChangeManager, SystemManager |

## 20.8 Rapor okuma önerileri

| Amaç | Bakılacak raporlar |
|---|---|
| Denetime hazırlık | Control Exceptions, Approval Step Detail, Settings Approval Trail, User Permission Matrix |
| Kişisel veri uyumu | Query Result Activity, Masking Activity, Data Access and Export Log, Masking Exempt Servers |
| Süreç verimliliği | Approval and Execution Duration, Work in Progress, Execution Activity |
| Teknik borç | Object Change Frequency, Multi Release Objects, Active Conflicts |
| Yetki gözden geçirme | User Permission Matrix, User Access Changes, Privileged Account Operations |

## 20.9 Dışa aktarma

Rapor sonuçları ekranda tablo olarak gösterilir ve dosya olarak dışa aktarılabilir. Kısıtlı raporların açılması denetim kaydına yazılır ve Data Access and Export Log raporunda görünür.

## Bölüm 21. Bildirimler ve E-posta

### 21.1 Hangi olayda kime bildirim gider

| Olay | Alıcı |
|---|---|
| Talep oluşturuldu | İlgili onay adımının rolündeki kullanıcılar. |
| Onay adımı sıraya geldi | O adımı kapatabilecek roldeki kullanıcılar. Adım bir kişiye atanmışsa o kişi. |
| Talep onaylandı | Talep sahibi. |
| Talep zamanlandı | Talep sahibi. |
| Talep çalıştırıldı | Talep sahibi. Sorgu sonucu taleplerinde ayrıca alıcı listesi. |
| Talep reddedildi | Talep sahibi, gerekçe ile. |
| Onay adımı reddedildi | Talep sahibi, gerekçe ile. |
| Talep artık gerekli değil işaretlendi | Talep sahibi. |
| Otomatik çalıştırma başarısız | Bildirim kuyruğuna hata bildirimi. |
| Sürüm paketi başarısız | Bildirim kuyruğuna hata bildirimi. |
| Worker beklenmedik hata | Workers:WorkerErrorMailTo adresine kritik bildirim. |

Adım bir kullanıcıya atanmışsa (talep sahibinin doğrudan yöneticisi adım rolünü karşılıyorsa) bildirim öncelikle ona gider.

### 21.2 Transaction içinde e-posta gönderilmez

E-posta gönderimi hiçbir zaman veritabanı işleminin içinde yapılmaz. Bunun yerine bildirim kuyruğa yazılır, işlem tamamlandıktan sonra worker kuyruğu boşaltır.

Sebebi: SMTP sunucusu yavaşsa veya erişilemezse veritabanı işlemi kilitli kalırdı. Ayrıca işlem geri alınsa bile e-posta gönderilmiş olurdu ve "gerçekleşmemiş bir olayın bildirimi" ortaya çıkardı.

Kullanıcıya yansıması: ekranda işlem tamamlandı mesajını gördükten sonra e-posta birkaç saniye içinde gelir, anında gelmeyebilir. Gönderim aralığı Workers:NotificationFlushSeconds ayarındadır.

### 21.3 E-posta Kaynakları ekranı

Ekran: Ayarlar > E-posta Kaynakları

Yetki: settings.email

Sorgu sonucu teslimlerinde kullanılabilecek adreslerin havuzudur.

| Alan | Ne gösterir |
|---|---|
| E-posta | Adres. |
| Aktif | Kullanımda mı. |
| Onaylı | Adres onaylanmış mı. Onaysız adrese teslim ek onay adımı gerektirir. |
| Onaylayan / Onay zamanı | |
| Kaynak | Elle mi eklendi, sistem mi ekledi. |

### 21.4 Bildirim Kayıtları ekranı

Ekran: Ayarlar > Bildirim Kayıtları

Yetki: settings.notificationlogs

| Kolon | Ne gösterir |
|---|---|
| Kanal | Gönderim kanalı (e-posta). |
| Alıcı | Hedef adres. |
| Konu | E-posta konusu. |
| Durum | Gönderildi, başarısız, kuyrukta. |
| Hata | Başarısızsa sebebi. |
| Deneme sayısı | Kaç kez denendi. |
| Zaman | |

Bir bildirim gitmediyse ilk bakılacak yer burasıdır. Kayıtlar 90 gün sonra bakım göreviyle silinir.

# Bölüm 22. ITSM Bilet Numarası

## 22.1 Ne işe yarar

Kurumun mevcut değişiklik yönetim sistemindeki bilet numarasının talebe bağlanmasını sağlar. Böylece SQL Change Guard kaydı ile kurumsal süreç kaydı eşleşir.

Ekran: Ayarlar > ITSM Biletleri

Yetki: settings.itsm

## 22.2 Alanlar

| **Alan** | **Ne işe yarar** |
|---|---|
| Bilet No | Bilet numarası. Benzersizdir. |
| Başlık | Biletin konusu. |
| Bağlantı | Bilete doğrudan gitmek için adres. |
| Kaynak | Biletin geldiği sistem. |

## 22.3 Zorunluluk ve biçim kontrolü

İki parametre ile yönetilir:

| **Parametre** | **Etkisi** |
|---|---|
| ChangeTicketRequired | true ise bilet numarası olmadan talep kaydedilemez. |
| ChangeTicketRegex | Girilen numaranın biçimini kontrol eder. Boş ise kontrol yapılmaz. |

Örnek: ChangeTicketRequired true ve ChangeTicketRegex ^CR-\d{5}$ iken:

| **Girilen** | **Sonuç** |
|---|---|
| CR-12345 | Kabul edilir. |
| CR-1234 | Reddedilir, beş hane gerekir. |
| 12345 | Reddedilir, önek eksik. |
| (boş) | Reddedilir, zorunlu. |

Bilet numarası talep listesinde, raporlarda ve denetim dosyasında görünür. Çakışma raporları da bilet numarasını kullanarak aynı nesneyi farklı biletlerle değiştirmeye çalışan talepleri bulur.

## Bölüm 23. Tarih ve Saat Modeli

### 23.1 Saklama: her zaman UTC

Sistemdeki tüm tarih alanları DATETIMEOFFSET tipindedir ve UTC olarak saklanır. İstisna yoktur.

Etkilenen tablolar: talepler, denetim kayıtları, onay adımları, sunucular, kullanıcılar, parametreler, bakım kayıtları, bildirim kayıtları, maskeleme kayıtları ve diğerleri. Her tablodaki CreatedAt, ModifiedAt, OccurredAt, ExecutedAt, ScheduledAt gibi alanların hepsi UTC'dir.

### 23.2 Neden UTC saklanır

Üç sebep vardır:

1. Yaz saati uygulaması. Yerel saatte saklanan bir kayıt, saat geri alındığında aynı saatin iki kez yaşanmasına sebep olur. "02:30'da ne oldu" sorusunun iki farklı cevabı olur. UTC'de bu belirsizlik yoktur.

2. Farklı şehirlerdeki kullanıcılar. İstanbul'daki kullanıcı ile Londra'daki kullanıcı aynı olayı farklı saatlerde görür, ama kayıt tek ve tutarlıdır.

3. Sıralama. Denetim zincirinin doğru sıralanabilmesi için tek bir zaman ekseni gerekir.

### 23.3 Gösterim

| **Yer** | **Hangi saat gösterilir** |
|---|---|
| Web arayüzü | Tarayıcının yerel saati. Kullanıcı ayar yapmaz, otomatik dönüşür. |
| Sunucu tarafında üretilen belgeler (denetim dosyası PDF, e-posta metinleri) | Display:TimeZone ayarındaki saat dilimi. Varsayılan Europe/Istanbul. |
| Raporlar | Rapor çıktısındaki tarihler UTC'den gösterim saat dilimine çevrilir. |
| Denetim kaydı | Ekranda yerel saat, veritabanında UTC. |

Sunucu tarafında üretilen belgelerde tarayıcı saati bilinemez; bu yüzden merkezi bir saat dilimi ayarı kullanılır.

Yanlış saat dilimi girilirse: Uygulama belge üretimini durdurmaz, UTC'ye düşer ve günlüğe bir uyarı yazar. Ayrıca belgelerde kullanılan saat dilimi etiketi görünür, böylece okuyan kişi hangi dilimin uygulandığını görebilir. Kurulumdan sonra bir belge üretip bu etiketi kontrol etmeniz önerilir.

### 23.4 Somut örnek

Bir talep 2 Ağustos 2026 saat 12:00 UTC'de çalıştırıldı.

| Kim bakıyor | Ne görür |
|---|---|
| İstanbul'daki kullanıcı (UTC+3) | 2 Ağustos 2026, 15:00 |
| Londra'daki kullanıcı (yaz saati, UTC+1) | 2 Ağustos 2026, 13:00 |
| Denetim dosyası PDF (Display:TimeZone = Europe/Istanbul) | 2 Ağustos 2026, 15:00 |
| Veritabanı kaydı | 2026-08-02 12:00:00 +00:00 |

Üç görüntü de aynı anı gösterir. Kayıt tektir.

### 23.5 Filtrelerde yerel gün sınırı

Bir kullanıcı "2 Ağustos" için filtre uyguladığında, kastettiği kendi yerel gününün başlangıcı ve sonudur.

İstanbul'daki bir kullanıcı için 2 Ağustos:

- Başlangıç: 1 Ağustos 21:00 UTC

- Bitiş: 2 Ağustos 21:00 UTC

Sistem bu dönüşümü otomatik yapar. Kullanıcı UTC hesabı yapmak zorunda kalmaz.

Bu ayrıntı raporlarda önemlidir: aynı raporu farklı saat dilimlerinden çalıştıran iki kişi, aynı tarihi seçse bile biraz farklı kayıt kümesi görebilir. Bu bir hata değildir, her ikisi de kendi gününü doğru görmektedir.

# Bölüm 24. Dil ve Yerelleştirme

## 24.1 Arayüz dili

Web arayüzü Türkçe ve İngilizce çalışır. Dil seçimi arayüz tarafındadır ve kullanıcı istediği zaman değiştirebilir. Tüm etiketler, mesajlar ve yardım metinleri kaynak dosyalarında tutulur.

## 24.2 API mesajlarının dili

API mesajları her zaman İngilizce döner. API'nin kanonik dili sabittir; istemcinin dil tercihi API cevabını değiştirmez.

Sebep: API cevapları günlüklere, denetim kayıtlarına ve entegrasyonlara girer. Bunların dilinin isteği yapan kişiye göre değişmesi, aynı olayın iki farklı dilde kaydedilmesine yol açardı ve arama ile karşılaştırmayı imkansız kılardı.

Arayüz, aldığı cevabı kendi dil kaynaklarıyla eşleştirerek kullanıcının seçtiği dilde gösterir. Yani kullanıcı Türkçe mesaj görür, ancak arka planda kayda giren metin İngilizcedir.

## 24.3 Rapor çıktıları

Rapor adları, açıklamaları ve kolon başlıkları İngilizcedir. Sebep Bölüm 20.1'de anlatıldı.

## 24.4 Genel kabul görmüş terimler

Rol adları gibi genel kabul görmüş teknik terimler çevrilmez. DbAdmin, ChangeManager, SystemManager her iki dilde de aynı yazılır. Bu terimlerin çevrilmesi ekipler arası iletişimi zorlaştırırdı.

# Bölüm 25. Bakım ve İşletme

## 25.1 Bakım Görevleri ekranı

Ekran: Ayarlar > Bakım

Yetki: settings.maintenance

| Alan | Ne işe yarar | Varsayılan |
|---|---|---|
| Ad | Görevin adı. | |
| Açıklama | Ne yaptığı. | |
| Yordam Adı | Çalıştırılacak saklı yordam. INTERNAL: ile başlıyorsa uygulama içi görevdir. | |
| Sıklık (dakika) | Kaç dakikada bir çalışacağı. | |
| Zaman Aşımı (saniye) | En fazla ne kadar sürebileceği. | |
| Deneme Sayısı | Hata durumunda kaç kez yeniden denenecek. | |
| Deneme Aralığı (saniye) | Denemeler arası bekleme. | |
| Etkin | Görev çalışsın mı. | Açık |
| İç Görev | Ürünle gelen, silinemeyen görev mi. | |

## 25.2 Varsayılan bakım görevleri

| Görev | Ne yapar | Sıklık | Zaman aşımı |
|---|---|---|---|
| Expired Token Cleanup | Süresi dolmuş oturum yenileme kayıtlarını siler. | 60 dakika | 60 saniye |
| Old Notification Log Cleanup | 90 günden eski bildirim kayıtlarını siler. | 1440 dakika (günde bir) | 120 saniye |
| QR File Cleanup | Saklama süresi dolan sorgu sonucu dosyalarını diskten siler ve veritabanındaki yolu temizler. QrFileRetentionDays parametresine bağlıdır. | Uygulama içi | - |

### 25.3 Bakım kayıtları

Her çalıştırma kaydedilir: başlangıç, bitiş, süre, durum, hata mesajı, deneme numarası. Bir görev sürekli hata veriyorsa bu listeden görülür.

### 25.4 Tanılama ekranı

Ekran: Ayarlar > Tanılama

Yetki: settings.diagnostics

| Kontrol | Ne doğrular | Sorun çıkarsa |
|---|---|---|
| Mail yapılandırması | Uygulamanın fiilen kullandığı SMTP ayarları. Şifre gösterilmez. | Ayarların beklediğinizle aynı olduğunu doğrulayın. |
| Mail testi | Verilen ayarlarla veya mevcut ayarlarla deneme e-postası gönderir. | Hata mesajı SMTP sunucusundan gelir; sunucu adı, port, kimlik bilgileri ve SSL ayarını kontrol edin. |

### 25.5 Worker izleme

Worker servisi düzenli olarak API'ye kendi durumunu bildirir. Arayüzde worker'ın çevrimiçi olup olmadığı ve son bildirim zamanı görünür.

Worker çevrimdışı görünüyorsa:

1. Windows servisinin çalıştığını kontrol edin.

2. WorkerApi:BaseUrl doğru mu bakın.

3. WorkerApi:Key ile API tarafındaki Workers:ApiKey aynı mı bakın.

4. Worker günlük dosyalarına bakın.

### 25.6 Yedekleme önerileri

| Yedeklenecek | Neden | Sıklık |
|---|---|---|
| Uygulama veritabanı | Tüm talepler, kurallar ve denetim kayıtları. | Kurumsal politikanıza göre. |
| Immutable veritabanı | Denetim kaydının ikinci kopyası. Ana veritabanıyla birlikte yedeklenmelidir; yoksa karşılaştırma yapılamaz. | Ana veritabanı ile aynı. |
| Şifreleme anahtarları | Security:ServerPasswordKey, Security:QrZipPasswordKey, Audit:HmacKeys, Jwt:SigningKey. | Değiştikçe. |
| Lisans dosyası | Yeniden kurulumda gerekir. | Değiştikçe. |

| Yedeklenecek | Neden | Sıklık |
|---|---|---|
| appsettings dosyaları | Yapılandırmanın tamamı. | Değiştikçe. |

En kritik uyarı: Şifreleme anahtarları veritabanı yedeğinin içinde değildir. Yalnızca veritabanını yedekleyip anahtarları kaybederseniz, sunucu şifrelerini geri getiremez ve eski denetim kayıtlarını doğrulayamazsınız. Anahtarları veritabanı yedeğinden ayrı ve güvenli bir yerde saklayın.

### 25.7 Çökme kurtarma eşiği

Worker bir talebi çalıştırmaya başladığında o talebi sahiplenir ve başlangıç anını kaydeder. Servis çalışma ortasında durursa (çökme, yeniden başlatma, sunucu kapanması) talep Executing durumunda kalır ve kimse onu bitirmez.

Worker her turun başında bu takılı kalmış talepleri arar ve Approved durumuna geri döndürür. Aramanın ölçütü tek bir sorudur: bu talep ne kadar süredir sahiplenilmiş durumda?

| Ayar | Anlamı |
|---|---|
| Workers:StaleExecutionRecoveryMinutes | Bu süreden daha uzun süredir devam eden çalıştırmalar ölü kabul edilir. |
| Varsayılan | 120 dakika |
| Alt sınır | 5 dakika. Daha küçük bir değer yazılırsa yok sayılır ve 120 dakika kullanılır. |
| Üst sınır | 365 gün. Daha büyük bir değer yazılırsa 365 güne çekilir. |

Sınır dışı bir değer yazıldığında worker açılışında günlüğe bir uyarı yazılır.

#### Eşik neden var

Eşik olmasaydı, her turun başında sahiplenilmiş her talep ölü sayılırdı. Tek worker kopyasında bu sorun çıkarmaz, çünkü turlar üst üste binmez. Ancak ikinci bir worker kopyası çalıştırıldığında, o kopyanın turu birinci kopyanın o an yürütmekte olduğu talebi ölü sayıp Approved durumuna çevirebilirdi. Talep yeniden alınabilir ve aynı betik iki kez çalışabilirdi.

#### Değeri nasıl seçmelisiniz

Eşik, kurumunuzdaki en uzun meşru çalıştırmadan büyük olmalıdır.

| Durum | Sonuç |
|---|---|
| Eşik çok küçük | Hâlâ çalışmakta olan bir talep ölü sayılabilir. |

| Durum | Sonuç |
| --- | --- |
| Eşik çok büyük | Gerçekten çökmüş bir talebin kurtarılması gecikir; talep o süre boyunca Executing görünür. |

Örnek: en uzun betiğiniz yaklaşık 20 dakika sürüyorsa 45 dakika makul bir değerdir. Kurtarma 45 dakika içinde gerçekleşir ve çalışan hiçbir iş etkilenmez.

Kurtarmayı beklemek istemiyorsanız talebi elle sonlandırabilirsiniz: takılı kalmış talepte çalıştırmayı iptal edin veya talebi manuel işaretleyin.

# Bölüm 26. Sorun Giderme

## 26.1 Uyarı ve hata mesajları sözlüğü

Aşağıdaki tablo, kullanıcının karşılaşabileceği mesajları, nerede çıktığını, sebebini ve çözümünü verir.

### Yetki mesajları

| Mesaj (TR) | Mesaj (EN) | Nerede | Sebep | Çözüm |
|---|---|---|---|---|
| Bu işlem için yetkiniz yok. | You don't have permission for this action. | Her yerde | Rolünüzde gerekli yetki yok. | Yöneticinizden yetki isteyin. |
| Yeni talep oluşturma yetkiniz yok. | You don't have permission to create requests. | Talep ekranı | cr.add yetkisi yok. | Rolünüzü kontrol ettirin. |
| Bu isteği düzenleme yetkiniz yok. | You don't have permission to edit this request. | Talep detayı | Talebin sahibi değilsiniz. | Talep sahibiyle iletişime geçin. |
| Script çalıştırma yetkiniz yok. | You don't have execute permission. | Talep detayı | cr.execute yetkisi yok. | Yetkili bir kullanıcıdan çalıştırmasını isteyin. |
| Bu işlem için onay yetkisi gereklidir. | Approval permission is required. | Talep detayı | cr.approve yetkisi yok. | Yetkili onaylayıcıya yönlendirin. |
| QR dosyasını indirme yetkiniz yok. | You don't have permission to download this QR file. | Talep detayı | QrRequesterCanDownload kapalı veya rolünüz QrDownloadMinimumRole eşiğinin altında. | Bölüm 7.7. |
| Denetçi rolüne onaylama veya çalıştırma yetkisi verilemez. | The auditor role cannot be granted approve or execute permissions. | Roller ekranı | Denetçi görevler ayrılığı gereği akışa giremez. | Farklı bir rol kullanın. |
| Bu hesap kapatılmıştır. | This account has been closed. | Giriş ekranı | Kullanıcı silinmiş. | Yöneticinize başvurun. |

### Durum mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Sadece Pending durumundaki istekler onaylanabilir. | Only Pending requests can be approved. | Talep zaten onaylanmış veya kapanmış. | Sayfayı yenileyin. |

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Sadece Approved durumundaki istekler çalıştırılabilir. | Only Approved requests can be executed. | Talep henüz onaylanmamış. | Onay sürecinin tamamlanmasını bekleyin. |
| Sadece Approved durumundaki istekler zamanlanabilir. | Only Approved requests can be scheduled. | Aynı sebep. | |
| Tamamlanmış veya iptal edilmiş istekler düzenlenemez. | Completed or cancelled requests cannot be edited. | Talep ölü durumda. | Yeni talep açın veya mevcut talebi kopyalayın. |
| Bu durumda ({0}) {1} yapılamaz. | Cannot perform '{1}' in status: {0}. | İşlem mevcut durumla uyumsuz. | Bölüm 11.1'deki geçiş matrisine bakın. |
| Sadece zamanlanmış Approved isteklerin zamanlaması iptal edilebilir. | Only scheduled Approved requests can have their schedule cancelled. | Talepte zamanlama yok. | |

## Görevler ayrılığı mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| İstek sahibi kendi isteğini onaylayamaz. | The requester cannot approve their own request. | RequesterCanSelfApproveExecute kapalı. | Başka bir onaylayıcıya yönlendirin. |
| İstek sahibi kendi scriptini çalıştıramaz. | The requester cannot execute their own script. | Aynı ayar. | Yetkili başka kişi çalıştırmalı. |
| Bu talebi onaylayan kişi aynı talebi çalıştıramaz. | The person who approved this request cannot execute it. | ApproverCanExecute kapalı. | Farklı bir yetkili çalıştırmalı. |

## Geri alma mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Bu sunucu için çalıştırmadan önce rollback kaydı zorunludur. | A rollback record is required before execution on this server. | Sunucuda "Rollback Zorunlu" açık ve geri alma yok. | Geri alma talebi oluşturun. |
| Bu talep için zaten aktif bir rollback kaydı var. | An active rollback already exists for this request. | İkinci geri alma denemesi. | Mevcut geri almayı kullanın. |

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Rollback yalnızca talebin sahibi tarafından oluşturulabilir. | Rollback can only be created by the owner of the request. | Sahibi değilsiniz. | Talep sahibinden isteyin. |
| Kaynak talep henüz çalıştırılmadı. | The source request has not been executed yet. | Geri alma, kaynak uygulanmadan çalıştırılamaz. | Önce kaynak talebi çalıştırın. |
| Rollback sadece Change tipindeki talepler için oluşturulabilir. | Rollback can only be generated for Change type requests. | Sorgu sonucu talebinde geri alma yoktur. | - |
| Rollback tipi talepler kopyalanamaz. | Rollback type requests cannot be copied. | | Kaynak talebi kopyalayın. |
| Rollback talebi kaynak talebin sunucusuyla eşleşmelidir. | The rollback request server must match the source request server. | Sunucu değiştirilmiş. | Aynı sunucuyu seçin. |
| Kaynak talebin scriptleri düzenlendiği için mevcut rollback geçersiz kılındı. | The existing rollback was superseded because the source request scripts were edited. | Kaynak değişti, eski geri alma artık uymuyor. | Talep yeniden onaylandığında güncel geri alma oluşur; otomatik kapalıysa elle oluşturun. |

## Sürüm paketi ve sandbox mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Bu talep '{0}' yayın paketi altında. Önce paketten çıkarın. | This request is under release package '{0}'. Remove it from the package first. | Talep aktif pakette. | Paketten çıkarın veya paket üzerinden ilerleyin. |
| Sürüm paketine yalnızca değişiklik talepleri eklenebilir. | Only change requests can be added to a release package. | Sorgu sonucu talebi eklenmeye çalışıldı. | Bölüm 15.6. |
| Sorgu sonucu talepleri release paketi içinde çalıştırılamaz. | Query result requests cannot be executed inside a release package. | Maskeleme ve kayıt yalnızca kendi akışında üretilir. | Talebi tek başına çalıştırın. |
| Oracle sunucuları şu an release paketlerinde desteklenmemektedir. | Oracle servers are not currently supported in release packages. | Bilinen sınır. | Oracle taleplerini tek tek çalıştırın. |
| Sandbox yalnızca 'Change' tipindeki talepler için geçerlidir. | Sandbox is only available for 'Change' type requests. | | - |

## Ayar onayı mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Değişiklik onaya gönderildi. Onaylanana kadar mevcut değer geçerlidir. | The change was sent for approval. | Dört göz kuralı devrede. | Yetkili bir meslektaşınızdan onay isteyin. |
| Bu kayıt için onay bekleyen bir değişiklik zaten var. | There is already a pending change for this record. | İkinci istek. | Önce mevcut isteği sonuçlandırın. |
| Bu değişiklik için karar zaten verilmiş. | A decision has already been made for this change. | Başkası sizden önce karar verdi. | Listeyi yenileyin. |
| Kendi talep ettiğiniz ayar değişikliğini onaylayamaz veya reddedemezsiniz. | You cannot approve or reject a settings change you requested yourself. | Dört göz kuralı. | Aynı yetkiye sahip başka bir kullanıcı karar vermeli. |
| Reddetme gerekçesi zorunludur. | A rejection reason is required. | Gerekçe boş. | Gerekçe yazın. |
| Değişiklik onaylandı ancak uygulanamadı: {0} | The change was approved but could not be applied: {0} | Onay anında komut hata verdi. | Gerekçeyi okuyun, sorunu giderip yeniden isteyin. |
| Dört göz kuralı için parametre yetkisine sahip en az iki kullanıcı gerekir. | The four eyes rule needs at least two users holding the parameters permission. | Tek yetkili kullanıcı var. | İkinci yetkili kullanıcı tanımlayın. |

## Sunucu ve lisans mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Sunucu aktif değil: {0} | Server is not active: {0} | Sunucu pasife alınmış. | Sunucuyu aktif edin veya başka sunucu seçin. |
| Bu sunucunun veritabanı tipi ({0}) lisansınız tarafından desteklenmiyor. | This server's database type ({0}) is not covered by your license. | Lisans kapsamı dışı. | Lisansınızı güncelleyin. |
| Seçilen sunucunun veritabanı tipi ({0}) talep sunucusu tipi ({1}) ile uyuşmuyor. | The selected server type ({0}) does not match the request server type ({1}). | Farklı türde sunucular karıştırılmış. | Aynı türde sunucular seçin. |

### Talep içeriği mesajları

| Mesaj (TR) | Mesaj (EN) | Sebep | Çözüm |
|---|---|---|---|
| Talep en az 1 script içermelidir. | A request must have at least 1 script. | Boş talep. | Betik ekleyin. |
| QueryResult tipi talepler için yalnızca 1 script eklenebilir. | QueryResult requests can only have 1 script. | Çoklu betik denemesi. | Sorguyu tek betikte birleştirin. |
| En erken zamanlama tarihi: {0} (minimum {1} saniyelik gecikme). | Earliest schedule date: {0} (minimum delay: {1} seconds). | Seçilen tarih çok yakın. | Daha ileri bir tarih seçin. Bölüm 6.4. |
| Tam maskeleme dışında bir rejim seçtiğinizde gerekçe zorunludur. | A reason is required when a mode other than full masking is selected. | Muafiyet gerekçesi boş. | Yazılı gerekçe girin. |

### Çalıştırma günlüğü notları

| Not | Anlamı |
|---|---|
| -- {0} blok başarıyla çalıştı; {1}. blokta HATA alındı ve yürütme durdu. | Kaça kadar çalıştığını ve nerede durduğunu gösterir. |
| -- ORACLE ATOMİKLİK NOTU: ... | Oracle hedefte, hata öncesi çalışan DDL ifadeleri kalıcı olabilir. Kalıntı durumu inceleyin ve gerekirse geri alma talebini elle çalıştırın. |
| -- Detection time budget exceeded; unscanned columns were masked. | Hassas veri tespiti süre bütçesini aştı, taranamayan kolonlar güvenli tarafta maskelendi. |

## 26.2 Sık karşılaşılan senaryolar

### Giriş yapılamıyor

| Kontrol | Ne yapmalı |
|---|---|
| Hesap kilitli mi | Ardışık hatalı denemeler hesabı kilitler. Auth:AccountLockoutMinutes kadar bekleyin veya yöneticiye kilidi açtırın. |
| Hesap silinmiş mi | "Bu hesap kapatılmıştır" mesajı geliyorsa yönetici hesabı canlandırmalıdır. |
| LDAP mı yerel mi | LDAP hesabında şifre kurum dizininden değişir. |

| Kontrol | Ne yapmalı |
|---|---|
| İki adımlı doğrulama | Hesabınızda iki adımlı doğrulama kuruluysa doğrulayıcı uygulamadan kod girmeniz gerekir. Kurulu değilse kod istenmez. Bkz. Bölüm 2.4.9. |

## Sunucuya bağlanılamıyor

| Kontrol | Ne yapmalı |
|---|---|
| Bağlantı testi | Ayarlar > Sunucular ekranından test edin, mesajı okuyun. |
| Ağ ve port | Güvenlik duvarını ve port numarasını kontrol edin. |
| Kimlik bilgileri | Şifre değişmiş olabilir. |
| Şifreleme anahtarı | Security:ServerPasswordKey değiştiyse eski şifreler çözülemez. Şifreyi yeniden girin. |
| Lisans | Veritabanı türü lisans kapsamında mı. |

## Talep çalışmıyor

| Kontrol | Ne yapmalı |
|---|---|
| Durum | Talep Approved mı. |
| Bekleyen onay adımı | Talep detayında hangi adımda beklediğine bakın. |
| Zamanlama | Zamanlanmışsa zamanı gelmemiş olabilir. |
| Geri alma zorunluluğu | Sunucuda "Rollback Zorunlu" açıksa geri alma oluşturun. |
| Asla atlanamaz bulgu | Böyle bir bulgu varsa otomatik çalıştırma yapılmaz, elle çalıştırın. |
| Worker | Worker çalışıyor mu. |
| Aktif paket | Talep bir sürüm paketinde olabilir. |

## Worker çalışmıyor

| Kontrol | Ne yapmalı |
|---|---|
| Servis durumu | Windows servisini kontrol edin. |

| Kontrol | Ne yapmalı |
|---|---|
| MainWorkEnabled | Bayrak false olabilir. |
| API anahtarı | WorkerApi:Key ile Workers:ApiKey aynı mı. |
| Günlükler | Worker günlük dosyalarına bakın. |

## Rapor boş geliyor

| Kontrol | Ne yapmalı |
|---|---|
| Tarih aralığı | Varsayılan son 30 gündür. Aralığı genişletin. |
| Sunucu filtresi | Yanlış sunucu seçilmiş olabilir. |
| Rol kısıtı | Kısıtlı raporlarda yetkiniz olmayabilir. |
| Modül | InsightsReporting modülü lisanslı mı. |
| Veri | Gerçekten o dönemde kayıt olmayabilir. Bu iyi bir sonuç da olabilir. |

## E-posta gitmiyor

| Kontrol | Ne yapmalı |
|---|---|
| Mail:Enabled | Kapalı olabilir. |
| Mail:DevMode | Açıksa tüm e-postalar test adresine gider. |
| Bildirim kayıtları | Ayarlar > Bildirim Kayıtları ekranından hata mesajını okuyun. |
| Mail testi | Ayarlar > Tanılama ekranından deneme e-postası gönderin. |
| Worker | Bildirimleri worker gönderir; çalışıyor mu. |

# Bölüm 27. Ekler

## 27.1 Enum sözlüğü

### Roller (enmRole)

| Değer | Ad | Hiyerarşi seviyesi |
|---|---|---|
| 0 | NoAccess | 0 |
| 5 | Viewer | 5 |
| 10 | Auditor | 10 |
| 15 | Requester | 15 |
| 20 | RequestManager | 20 |
| 25 | DbAdmin | 25 |
| 30 | ChangeManager | 30 |
| 35 | SystemManager | 35 |

### Talep durumu (ChangeRequestStatus)

| Değer | Ad | Ölü durum |
|---|---|---|
| 1 | Pending | Hayır |
| 2 | Approved | Hayır |
| 3 | Executed | Evet |
| 4 | Rejected | Evet |
| 5 | Manual | Evet |
| 6 | NoLongerNeeded | Evet |
| 7 | Executing | Hayır |

### Talep tipi (QueryType)

| Değer | Ad |
|---|---|
| 0 | None |
| 1 | Change |
| 2 | QueryResult |

| Değer | Ad |
|---|---|
| 3 | Rollback |

### Veritabanı tipi (DatabaseType)

| Değer | Ad |
|---|---|
| 1 | SqlServer |
| 2 | Oracle |
| 3 | PostgreSQL |

### Risk bandı (RiskTier)

| Değer | Ad | Skor aralığı |
|---|---|---|
| 0 | Info | 90 - 100 |
| 1 | Low | 70 - 89 |
| 2 | Medium | 50 - 69 |
| 3 | High | 25 - 49 |
| 4 | Critical | 0 - 24 |

### Sürüm paketi çalıştırma modu (ReleaseExecutionMode)

| Değer | Ad |
|---|---|
| 1 | AllOrNothing |
| 2 | StopOnError |
| 3 | ContinueOnError |

### Onay adımı durumu

| Değer | Anlamı |
|---|---|
| 0 | Bekliyor |

| Değer | Anlamı |
|---|---|
| 1 | Onaylandı |
| 2 | Reddedildi |
| 3 | Atlandı |

### Sunucu ortamı

| Değer |
|---|
| Production |
| Staging |
| Development |
| Test |

Onay politikası çözümlemesinde önem sırası: Production > Staging > Test > Development > tanımsız.

### Maskeleme rejimi

| Değer | Anlamı |
|---|---|
| Strict | Katalog ve desen birlikte. |
| ContentOnly | Yalnızca desen. |

### Maskeleme politikası kaydı

| Değer | Anlamı |
|---|---|
| AutoMask | Tespit edilen kolonlar maskelendi. |
| WarnOnly | Tespit yapıldı, maskelenmedi. |
| MaskingOff | Ana anahtar kapalıydı, hiç tespit yapılmadı. |

## Maskeleme gerekçe tipleri

| Değer | Anlamı |
|---|---|
| Catalog | Kolon adı kataloğa uydu. |
| Pattern | Değer desene uydu. |
| UnmaskApproved | Onaylı istisna. |
| NotScanned | Süre bütçesi nedeniyle taranamadı, güvenli tarafta maskelendi. |
| PolicyWarnOnly | Politika WarnOnly. |
| MaskingOff | Ana anahtar kapalı. |

## Aday kolon durumu

| Değer | Anlamı |
|---|---|
| New | Karar bekliyor. |
| Accepted | Kataloğa alındı. |
| Ignored | Yok sayıldı. |

## Ayar değişikliği durumu

| Değer | Anlamı |
|---|---|
| Pending | Karar bekliyor. |
| Approved | Onaylandı ve uygulandı. |
| Rejected | Reddedildi. |
| Failed | Onaylandı ama uygulanamadı. |

## Lisans durumu

| Değer | Anlamı | Kısıtlı mod |
|---|---|---|
| Valid | Geçerli. | Hayır |
| Trial | Deneme. | Hayır |

| Değer | Anlamı | Kısıtlı mod |
|---|---|---|
| Expiring | Bitişe az kaldı. | Hayır |
| GracePeriod | Ek süre içinde. | Hayır |
| Expired | Süresi dolmuş. | Evet |
| Invalid | Geçersiz. | Evet |
| Missing / NoLicense | Yok. | Evet |

### Denetim özet sürümü

| Değer | Algoritma | Kapsam |
|---|---|---|
| 1 | SHA-256 | Eski değer ve IP hariç. |
| 2 | SHA-256 | Eski değer ve IP dahil. |
| 3 | HMAC-SHA256 | Sürüm 2 kapsamı, gizli anahtarla. |

## 27.2 Parametre hızlı referansı

| Anahtar | Grup | Varsayılan | Kontrol ayarı |
|---|---|---|---|
| DefaultQueryTimeout | Genel | 30 | Hayır |
| MaxQueryTimeout | Genel | 3600 | Hayır |
| ExecutionDelaySeconds | Genel | 300 | Hayır |
| RejectOnError | Genel | true | Evet |
| RequesterCanSelfApproveExecute | Görevler Ayrılığı | true | Evet |
| ApproverCanExecute | Görevler Ayrılığı | true | Evet |
| RequireDistinctApproverPerStep | Görevler Ayrılığı | false | Evet |
| EmergencyMinimumRole | Görevler Ayrılığı | Requester | Evet |
| SettingsApprovalPolicy | Ayar Onayı | Disabled | Evet |
| ChangeTicketRequired | ITSM Bilet | true | Evet |
| ChangeTicketRegex | ITSM Bilet | ^CR-\d{5}$ | Hayır |
| QrFileRetentionDays | Sorgu Sonucu | 1 | Evet |
| QrRequesterCanDownload | Sorgu Sonucu | true | Evet |

| Anahtar | Grup | Varsayılan | Kontrol ayarı |
|---|---|---|---|
| QrDownloadMinimumRole | Sorgu Sonucu | ChangeManager | Evet |
| QrMaskingEnabled | Hassas Veri | true | Evet |
| QrSensitiveDataPolicy | Hassas Veri | AutoMask | Evet |
| QrSensitiveDetectionSampleSize | Hassas Veri | 1000 | Evet |
| QrSensitiveDetectionThresholdPct | Hassas Veri | 20 | Evet |
| QrSensitiveDetectionSampleMode | Hassas Veri | Random | Evet |
| QrSensitiveDetectionTimeBudgetSeconds | Hassas Veri | 10 | Evet |
| QrColumnPreviewEnabled | Hassas Veri | true | Hayır |
| UnmaskedColumnRequestPolicy | Hassas Veri | RequireApproval | Evet |
| UnmaskedColumnApproverRole | Hassas Veri | SystemManager | Evet |
| QrEmailApprovalPolicy | E-posta Onayı | RequireApproval | Evet |
| QrEmailApproverRole | E-posta Onayı | ChangeManager | Evet |
| QrEmailAutoApproveUsers | E-posta Onayı | True | Evet |

## 27.3 Yetki anahtarları hızlı referansı

| Anahtar | Açtığı ekran veya işlem |
|---|---|
| cr.add | Talep oluşturma |
| qr.add | Sorgu sonucu talebi oluşturma |
| cr.execute | Çalıştırma |
| cr.sandbox | Sandbox |
| cr.approve | Onaylama, reddetme, zamanlama, manuel işaretleme |
| cr.seeall | Tüm talepleri görme |
| report.view | Raporlar |
| dashboard.view | Kontrol paneli |
| audit.view | Denetim kaydı ve doğrulama işlemleri |
| audit.evidenceexport | Denetim dosyası dışa aktarma |
| rp.manage | Sürüm paketleri |
| settings.servers | Sunucular |

| Anahtar | Açtığı ekran veya işlem |
|---|---|
| settings.users | Kullanıcılar |
| settings.roles | Roller |
| settings.params | Parametreler |
| settings.approvals | Onay politikaları |
| settings.standards | SQL standartları |
| settings.critical | Kritik nesneler |
| settings.itsm | ITSM biletleri |
| settings.qrsensitive | Hassas kolonlar ve aday kolonlar |
| settings.patterns | Hassas veri desenleri |
| settings.email | E-posta kaynakları |
| settings.maintenance | Bakım görevleri |
| settings.notificationlogs | Bildirim kayıtları |
| settings.license | Lisans |
| settings.diagnostics | Tanılama |

### 27.4 UTC saklama notu

Aşağıdaki tabloların tarih alanlarının tamamı DATETIMEOFFSET tipindedir ve UTC saklanır:

Users, ChangeRequests, ChangeRequestApprovalSteps, ChangeRequestResultColumns, AuditLogs, RefreshTokens, Servers, SqlStandards, CriticalObjects, EmailSources, ReleasePackages, Parameters, Scripts, ScriptObjects, ItsmTickets, QrSensitiveColumns, ReportDefinitions, ReportParameters, MaintenanceTasks, MaintenanceTaskLogs, NotificationLogs, ApprovalPolicies, ApprovalPolicySteps, ServerApprovalPolicies, SensitiveDataPatterns, QrMaskingAuditLog, QrMaskingAuditDetail, SensitiveColumnCandidates, SettingChangeRequests, SandboxResults, RiskCalc_Result.

Gösterim kuralları Bölüm 23'tedir.

### 27.5 Kurulum kontrol listesi

[ ] Boş veritabanı oluşturuldu.

[ ] ConnectionStrings:Default dolduruldu.

[ ] Jwt:SigningKey üretildi ve yazıldı.

[ ] Security:ServerPasswordKey üretildi ve yazıldı.

[ ] Security:QrZipPasswordKey üretildi ve yazıldı.

[ ] Audit:HmacKeys ve Audit:ActiveKeyId tanımlandı.

[ ] Mail ayarları yapıldı, DevMode false.

[ ] Display:TimeZone kuruma göre ayarlandı.

[ ] AllowedOrigins içine web arayüzü adresi eklendi.

[ ] QueryResult:OutputPath klasörü oluşturuldu ve yetki verildi.

[ ] Workers:ApiKey ile worker tarafındaki WorkerApi:Key eşleştirildi.

[ ] API başlatıldı, şema ve varsayılan veri oluştu.

[ ] Lisans yüklendi.

[ ] admin şifresi değiştirildi.

[ ] Worker servisi kuruldu ve başlatıldı.

### 27.6 İlk yapılandırma kontrol listesi

[ ] Sunucular tanımlandı ve bağlantı testleri başarılı.

[ ] Kritik nesneler listesi girildi.

[ ] SQL standartları gözden geçirildi, gerekmeyenler kapatıldı.

[ ] Hassas kolon kataloğu ve desenler gözden geçirildi.

[ ] En az bir genel onay politikası tanımlandı.

[ ] Ortam bazlı politikalar tanımlandı.

[ ] Acil durum politikası tanımlandı (kullanılacaksa).

[ ] Kullanıcılar eklendi ve rolleri atandı.

[ ] Her ayar yetkisine en az iki kullanıcı atandı.

[ ] SettingsApprovalPolicy kuruma uygun değere getirildi.

[ ] ChangeTicketRegex kurumun bilet biçimine göre ayarlandı.

[ ] Görevler ayrılığı parametreleri kararlaştırıldı.

[ ] Test talebi açılıp uçtan uca yürütüldü.

### 27.7 Denetim hazırlığı kontrol listesi

[ ] Kontrol İstisnaları raporu çalıştırıldı, her satır açıklanabiliyor.

[ ] Onay Adımı Detayı raporu ile çift kontrol kanıtı alındı.

[ ] Ayar Onay İzi raporu ile kontrol değişiklikleri gözden geçirildi.

[ ] Kullanıcı Yetki Matrisi raporu alındı.

[ ] Veri Erişim ve Dışa Aktarım Kaydı raporu incelendi.

[ ] Maskeleme Etkinliği raporunda MaskingOff kaydı olup olmadığı kontrol edildi.

[ ] Maskeleme Muafiyetli Sunucular raporundaki gerekçeler güncel.

[ ] Zincir Doğrula işlemi çalıştırıldı ve temiz.

[ ] Immutable Doğrula işlemi çalıştırıldı, eşleşmeyen kayıt yok.

[ ] Örnek talepler için denetim dosyası üretildi ve doğrulandı.