

# EKLERİSTAN QMS - Teknik Dokümantasyon

## 1. Sistem Mimarisi

### 1.1 Genel Mimari

```
graph TB
    A[Streamlit Frontend] --> B[SQLAlchemy ORM]
    B --> C{Veritabanı}
    C -->|Yerel| D[SQLite]
    C -->|Cloud| E[Supabase PostgreSQL]
    A --> F[Graphviz]
    A --> G[Pandas]
    F --> H[PDF Export]
    G --> I[Excel Export]
```

### 1.2 Teknoloji Stack

Katman	Teknoloji	Versiyon	Amaç
Frontend	Streamlit	1.x	Web UI framework
Backend	Python	3.8+	Ana uygulama dili
ORM	SQLAlchemy	2.x	Veritabanı soyutlama
Veritabanı	PostgreSQL	14+	Cloud production DB
Veritabanı	SQLite	3.x	Yerel development DB
Görselleştirme	Graphviz	0.20+	Organizasyon şeması
Data Processing	Pandas	2.x	Veri manipülasyonu
PDF	FPDF	1.7+	PDF oluşturma
Excel	openpyxl	3.x	Excel export
Timezone	pytz	2023+	İstanbul saat dilimi

### 1.3 Proje Yapısı

```
EKLERİSTAN_QMS/
├── app.py                # Ana uygulama dosyası (3713 satır)
├── constants.py          # Pozisyon seviyeleri ve sabitler
├── requirements.txt      # Python bağımlılıkları
├── packages.txt          # Sistem bağımlılıkları (Graphviz)
├── ekleristan_local.db   # SQLite veritabanı (yerel)
├── .streamlit/
│   └── config.toml      # Streamlit konfigürasyonu
├── sql/
│   ├── supabase_personel_org_restructure.sql
│   ├── rbac_database_setup.sql
│   └── ... (25 migration dosyası)
└── docs/
```

		KULLANICI_KILAVUZU.md
		TEKNİK_DOKUMANTASYON.md
		TEST_SENARYOLARI.md
		README.md

## 2. Veritabanı Şeması

### 2.1 Ana Tablolar

#### personel - Kullanıcı ve Personel Bilgileri

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
ad_soyad	TEXT	Personel adı
kullanici_adi	TEXT	Sistem giriş adı (UNIQUE)
sifre	TEXT	Şifre (plain text)
rol	TEXT	Kullanıcı rolü
gorev	TEXT	İş tanımı
bolum	TEXT	Bölüm adı (legacy)
departman_id	INTEGER	FK → ayarlar_bolumler
yonetici_id	INTEGER	FK → personel (self-reference)
pozisyon_seviye	INTEGER	0-6 arası seviye
vardiya	TEXT	Vardiya bilgisi
durum	TEXT	AKTİF / PASİF
sorumlu_bolum	TEXT	Sorumlu olduğu bölüm adı
kat	TEXT	Çalıştığı kat
izin_gunu	TEXT	Haftalık izin günü
ise_giris_tarihi	TEXT	İşe giriş tarihi
is_cikis_tarihi	TEXT	İşten çıkış tarihi (varsa)
ayrilma_sebebi	TEXT	Ayrılma sebebi (varsa)

#### İndeksler:

- idx\_personel\_departman on departman\_id
- idx\_personel\_yonetici on yonetici\_id
- idx\_personel\_seviye on pozisyon\_seviye

#### ayarlar\_bolumler - Departman Hiyerarşisi

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
bolum_adi	TEXT	Departman adı
ana_departman_id	INTEGER	FK → ayarlar_bolumler (recursive)
aktif	BOOLEAN	Aktif/Pasif

sira_no	INTEGER	Sıralama
aciklama	TEXT	Açıklama

**Özellik:** Sınırsız derinlikte hiyerarşi desteği (recursive foreign key)

#### ayarlar\_roller - Rol Tanımları

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
rol_adi	TEXT	Rol adı (UNIQUE)
aciklama	TEXT	Rol açıklaması
aktif	BOOLEAN	Aktif/Pasif
olusturma_tarihi	TIMESTAMP	Oluşturulma zamanı

#### ayarlar\_yetkiler - Yetki Matrisi

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
rol_adi	TEXT	FK → ayarlar_roller
modul_adi	TEXT	Modül adı
erisim_turu	TEXT	Yok / Görüntüle / Düzenle

**Composite Key:** (rol\_adi, modul\_adi)

#### depo\_giris\_kayitlari - Üretim Kayıtları

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
tarih	DATE	Üretim tarihi
vardiya	TEXT	Vardiya
kullanici	TEXT	Kayıt eden
islem_tipi	TEXT	URETIM / DEPO
urun	TEXT	Ürün adı
lot_no	TEXT	Lot numarası
miktar	REAL	Üretim miktarı
fire	REAL	Fire miktarı
notlar	TEXT	Açıklama
zaman_damgasi	TIMESTAMP	Kayıt zamanı

#### urun\_kpi\_kontrol - Kalite Kontrol Kayıtları

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
tarih	DATE	Analiz tarihi

saat	TIME	Analiz saati
vardiya	TEXT	Vardiya
urun	TEXT	Ürün adı
lot_no	TEXT	Lot numarası
stt	DATE	Son Tüketim Tarihi
numune_no	TEXT	Numune sayısı
olcum1	REAL	Parametre 1 ortalama
olcum2	REAL	Parametre 2 ortalama
olcum3	REAL	Parametre 3 ortalama
karar	TEXT	ONAY / RED
kullanici	TEXT	Analist
tat	TEXT	Uygun / Uygun Değil
goruntu	TEXT	Uygun / Uygun Değil
notlar	TEXT	Detaylı veri (JSON)

#### **gmp\_soru\_havuzu - GMP Soru Bankası**

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
soru_metni	TEXT	Soru içeriği
kategori	TEXT	Hijyen / Altyapı / Personel
brc_ref	TEXT	BRC referans kodu
frekans	TEXT	GÜNLÜK / HAFTALIK / AYLIK
risk_puani	INTEGER	1-3 arası
lokasyon_ids	TEXT	Virgülle ayrılmış ID'ler
aktif	BOOLEAN	Aktif/Pasif

#### **gmp\_denetim\_kayitlari - GMP Denetim Sonuçları**

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
tarih	DATE	Denetim tarihi
saat	TIME	Denetim saati
kullanici	TEXT	Denetçi
lokasyon_id	INTEGER	FK → tanim_bolumler
soru_id	INTEGER	FK → gmp_soru_havuzu
durum	TEXT	UYGUN / UYGUN DEĞİL
fotograf_yolu	TEXT	Fotoğraf dosya adı
notlar	TEXT	Açıklama
brc_ref	TEXT	BRC referans
risk_puani	INTEGER	Risk seviyesi

## lokasyonlar - Hiyerarşik Lokasyon Yapısı (Fabrika Haritası)

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
ad	TEXT	Lokasyon adı
tip	TEXT	Kat / Bölüm / Hat / Ekipman
parent_id	INTEGER	FK → lokasyonlar (recursive)
sorumlu_id	INTEGER	FK → personel
sorumlu_departman	TEXT	Sorumlu departman adı
sira_no	INTEGER	Görüntüleme sırası
aktif	BOOLEAN	Aktif/Pasif
created_at	TIMESTAMP	Oluşturulma zamanı

**Özellik:** Sınırsız derinlikte hiyerarşi. Örnek: Kat 1 > Üretim > Fırın Hattı > Fırın 1

## ayarlar\_temizlik\_plani - Master Temizlik Planı

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
lokasyon_id	INTEGER	FK → lokasyonlar
ekipman_id	INTEGER	FK → lokasyonlar (tip = Ekipman)
yapisal_alan	TEXT	Zemin / Duvar / Tavan vb.
temizlik_turu	TEXT	Rutin / Arıza Sonrası / Özel
siklik	TEXT	Günlük / Haftalık / Aylık
uygulayici_personel	TEXT	Temizlik yapacak kişi adı
sorumlu_rol	TEXT	Eski (legacy)
kontrol_rol	TEXT	1. Kontrol sorumlusu rolü
kontrol2_rol	TEXT	2. Kontrol (Kalite) rolü
kimyasal_id	INTEGER	FK → kimyasal_envanter
metot_id	INTEGER	FK → tanim_metotlar
validasyon_siklik	TEXT	Validasyon sıklığı
verifikasyon_yontemi	TEXT	ATP / Swap / Görsel Kontrol
verifikasyon_siklik	TEXT	Doğrulama sıklığı
risk_seviyesi	TEXT	Düşük / Orta / Yüksek

## kimyasal\_envanter - Kimyasal Tanımları

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
kimyasal_adi	TEXT	Kimyasal adı
tedarikci	TEXT	Tedarikçi firma
msds_yolu	TEXT	MSDS dosya yolu

tds_yolu	TEXT	TDS dosya yolu
olusturma_tarihi	TIMESTAMP	Kayıt tarihi

#### **tanım\_metotlar - Temizlik Metotları**

Kolon	Tip	Açıklama
metot_adi	TEXT	Metot adı (Primary Key)
aciklama	TEXT	Detaylı açıklama

**Not:** Bu tabloda id sütunu yok, rowid kullanılır.

#### **temizlik\_kayitlari - Operasyonel Temizlik Kayıtları**

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
tarih	TEXT	Temizlik tarihi
saat	TEXT	Temizlik saati
kullanici	TEXT	Temizliği yapan
bolum	TEXT	Bölüm adı
islem	TEXT	Yapılan işlem
durum	TEXT	Tamamlandı / Beklemede
dogrulama_tipi	TEXT	Görsel / ATP / Swap
aciklama	TEXT	Notlar

#### **hiyyen\_kontrol\_kayitlari - Personel Hijyen Kayıtları**

Kolon	Tip	Açıklama
id	INTEGER	Primary Key
tarih	TEXT	Kontrol tarihi
saat	TEXT	Kontrol saati
kullanici	TEXT	Kontrolü yapan
vardiya	TEXT	Vardiya
bolum	TEXT	Bölüm
personel	TEXT	Kontrol edilen personel
durum	TEXT	UYGUN / UYGUN DEĞİL
sebepl	TEXT	Uygunluluk sebebi
aksiyon	TEXT	Alınan aksiyon
genel_karar	TEXT	Genel değerlendirme

## **2.2 View'lar**

#### **v\_organizasyon\_semasi - Organizasyon Şeması View**

```
CREATE OR REPLACE VIEW v_organizasyon_semasi AS
```

```
SELECT
    p.id,
    p.ad_soyad,
    p.gorev,
    p.rol,
    p.pozisyon_seviye,
    p.yonetici_id,
    y.ad_soyad as yonetici_adi,
    COALESCE(d.bolum_adi, p.bolum, 'Tanımsız') as departman,
    d.id as departman_id,
    p.kullanici_adi,
    p.durum,
    p.vardiya
FROM personel p
LEFT JOIN personel y ON p.yonetici_id = y.id
LEFT JOIN ayarlar_bolumler d ON p.departman_id = d.id
WHERE p.ad_soyad IS NOT NULL
ORDER BY p.pozisyon_seviye, p.ad_soyad;
```



**Amaç:** Organizasyon şeması için optimize edilmiş veri çekme

### 3. Tablo İlişkileri (ER)

#### 3.1 Foreign Key Özeti

Kaynak Tablo	Kaynak Kolon	Hedef Tablo	Hedef Kolon
personel	departman_id	ayarlar_bolumler	id
personel	yonetici_id	personel	id
ayarlar_bolumler	<b>ana_departman_id</b>	ayarlar_bolumler	id
lokasyonlar	parent_id	lokasyonlar	id
lokasyonlar	sorumlu_id	personel	id
ayarlar_temizlik_plani	lokasyon_id	lokasyonlar	id
ayarlar_temizlik_plani	ekipman_id	lokasyonlar	id
ayarlar_temizlik_plani	kimyasal_id	kimyasal_envanter	id
ayarlar_temizlik_plani	metot_id	tanim_metotlar	rowid
gmp_denetim_kayitlari	lokasyon_id	tanim_bolumler	id
gmp_denetim_kayitlari	soru_id	gmp_soru_havuzu	id

### 4. Modül-Tablo Eşlemesi

Modül	Kullanılan Tablolar	Açıklama
 <b>Üretim Girişi</b>	depo_giris_kayitlari, ayarlar_urunler, personel	Üretim ve fire kayıtları
 <b>KPI &amp; Kalite Kontrol</b>	urun_kpi_kontrol, urun_parametreleri, ayarlar_urunler	Ürün kalite ölçümleri

♥ <b>GMP Denetimi</b>	gmp_denetim_kayitlari, gmp_soru_havuzu, tanim_bolumler	Saha denetimleri
👤 <b>Personel Hijyen</b>	hijyen_kontrol_kayitlari, personel	Personel hijyen kontrolleri
🧼 <b>Temizlik Kontrol</b>	temizlik_kayitlari, ayarlar_temizlik_plani, lokasyonlar, kimyasal_envanter, tanim_metotlar	Master plan ve operasyonel kayıtlar
📊 <b>Kurumsal Raporlama</b>	Tüm tablolar (READ-ONLY)	Dashboard ve raporlar
⚙️ <b>Ayarlar</b>	personel, ayarlar_bolumler, ayarlar_roller, ayarlar_yetkiler, lokasyonlar, ayarlar_temizlik_plani, kimyasal_envanter, tanim_metotlar, gmp_soru_havuzu	Sistem tanımları

## 4.1 Modül Detayları (5N1K Formatı)

Bu bölüm yapay zekanın hatasız revizyon yapabilmesi için referans noktasıdır. Her modül için: NE, NİÇİN, NEREDE, NE ZAMAN, NASIL, KİM soruları cevaplanmıştır.

### MODÜL 1: 🏭 Üretim Girişi

#### 1.1 Üretim Kayıt Formu

5N1K

Soru	Cevap
NE?	Günlük üretim miktarlarını ve fire detaylarını kaydetme formu
NİÇİN?	Lot bazlı üretim takibi, fire analizi ve raporlama için veri girişi
NEREDE?	Ana Menü → 🏭 Üretim Girişi
NE ZAMAN?	Her vardiya sonunda veya üretim tamamlandığında
NASIL?	Form doldur → Kaydet → depo_giris_kayitlari tablosuna INSERT
KİM?	"Düzenle" yetkisi olan roller (Admin, Üretim Sorumlusu)

#### Teknik Detaylar

Bilgi	Değer
Tablo	depo_giris_kayitlari
Satırlar	app.py: 670-744
Form ID	uretim_giris_form
Fonksiyonlar	veri_getir("Ayarlar_Urunler"), bolum_bazli_urun_filtrele(), guvenli_kayit_ekle()

#### Form Alanları



UI Etiketi	DB Kolonu	Tip	Zorunlu	Varsayılan
Üretim Tarihi	tarih	DATE	✓	Bugün
Vardiya	vardiya	TEXT	✓	GÜNDÜZ VARDİYASI
Üretilen Ürün	urun	TEXT	✓	-
Lot No / Parti No	lot_no	TEXT	✓	-
Üretim Miktarı	miktar	REAL	✓	0.00
Fire Miktarı	fire	REAL	✗	0.00
Üretim Notu	notlar	TEXT	✗	-

#### SQL Sorguları

```
-- INSERT
INSERT INTO depo_giris_kayitlari
(tarih, vardiya, kullanıcı, islem_tipi, urun, lot_no, miktar, fire, not
VALUES (:t, :v, :k, 'URETIM', :u, :l, :m, :f, :n, :z)
```

#### Validasyon Kuralları

Kural	Koşul	Hata Mesajı
Lot No zorunlu	f_lot boş olamaz	"Lütfen Lot No ve Miktar alanlarını doldurun"
Miktar > 0	f_miktar > 0	"Lütfen Lot No ve Miktar alanlarını doldurun"

#### Hata Durumları

Hata	Sebebe	Çözüm
"Ürün tanımı bulunamadı"	ayarlar_urunler tablosu boş	Ayarlar > Ürün Yönetimi'nden ürün ekleyin
"Bu modüle erişim yetkiniz bulunmamaktadır"	Yetersiz yetki	Admin'den "Düzenle" yetkisi isteyin

#### Bağımlılıklar

Fonksiyon	Dosya	Satır	Amaç
veri_getir("Ayarlar_Urunler")	app.py	679	Ürün listesi çekme
bolum_bazli_urun_filtrele()	app.py	602-640	Bölüm bazlı ürün filtreleme
guvenli_kayit_ekle()	app.py	355-391	Güvenli INSERT işlemi
kullanici_yetkisi_var_mi()	app.py	582-600	Yetki kontrolü
get_istanbul_time()	app.py	349-350	Türkiye saat dilimi

## 1.2 Günlük Üretim İzleme

5N1K

Soru	Cevap
NE?	Seçilen tarihteki üretim kayıtlarını görüntüleme tablosu

NİÇİN?	Günlük üretim takibi ve özet raporlama
NEREDE?	🏭 Üretim Girişi → Aşağıdaki "📊 Günlük Üretim İzleme" bölümü
NE ZAMAN?	İstenildiği zaman
NASIL?	Tarih seç → Tablo görüntülenir → Toplam üretim/fire gösterilir
KİM?	Modüle erişimi olan herkes

#### Teknik Detaylar

Bilgi	Değer
Satırlar	app.py: 718-744
Fonksiyonlar	veri_getir("Depo_Giris_Kayitlari")

## MODÜL 2: 📊 KPI & Kalite Kontrol

### 2.1 Dinamik Kalite Kontrol Formu

5N1K

Soru	Cevap
NE?	Ürün bazlı kalite ölçümü ve ONAY/RED karar formu
NİÇİN?	Lot bazlı kalite kontrol, STT hesaplama, parametre takibi
NEREDE?	Ana Menü → 📊 KPI & Kalite Kontrol
NE ZAMAN?	Her üretim lotu için kalite kontrolü yapılacağında
NASIL?	Ürün seç → Parametreler yüklenir → Ölçüm gir → Otomatik ONAY/RED kararı
KİM?	"Görüntüle" veya "Düzenle" yetkisi olan roller

#### Teknik Detaylar

Bilgi	Değer
Tablo	urun_kpi_kontrol
Satırlar	app.py: 746-895
Form ID	kpi_form
Fonksiyonlar	veri_getir("Ayarlar_Urunler"), guvenli_kayit_ekle()

#### Form Alanları

UI Etiketi	DB Kolonu	Tip	Zorunlu
Ürün Seçin	urun	TEXT	✓
Lot No	lot_no	TEXT	✓
Vardiya	vardiya	TEXT	✓
STT (Otomatik)	stt	DATE	✓
Ölçüm Değerleri	olcum1, olcum2, olcum3	REAL	✓

Tat Kontrolü	tat	TEXT	×
Görüntü Kontrolü	goruntu	TEXT	×

#### Validasyon Kuralları

Kural	Koşul	Sonuç
Parametre limitleri	Min/Max aralığı kontrolü	ONAY veya RED
STT etiket kontrolü	Checkbox işaretli olmalı	Form submit engeli



#### Bağımlılıklar

Fonksiyon	Amaç
urun_parametreleri tablosu	Dinamik parametre yükleme
Raf ömrü hesaplama	$raf\_omru\_gun + bugün = STT$

## MODÜL 3: GMP Denetimi

### 3.1 Saha Denetim Formu

5N1K

Soru	Cevap
NE?	BRC V9 uyumlu GMP saha denetim formu
NİÇİN?	Lokasyon bazlı hijyen ve altyapı denetimi
NEREDE?	Ana Menü →   GMP Denetimi
NE ZAMAN?	Frekans algoritmasına göre (Günlük/Haftalık/Aylık)
NASIL?	Lokasyon seç → Aktif sorular listelenir → UYGUN/UYGUN DEĞİL seç → Kaydet
KİM?	Kalite Sorumlusu, Vardiya Amiri, Admin

#### Teknik Detaylar

Bilgi	Değer
Tablo	gmp_denetim_kayitlari
Satırlar	app.py: 897-1019
Fonksiyonlar	veri_getir("Tanim_Bolumler"), veri_getir("GMP_Soru_Havuzu")

#### Frekans Algoritması

```
# Satır 906-913
gun_index = simdi.weekday() # 0=Pazartesi
ay_gunu = simdi.day
```


```
aktif_frekanslar = ["GÜNLÜK"]
if gun_index == 0: aktif_frekanslar.append("HAFTALIK") # Pazartesi
if ay_gunu == 1: aktif_frekanslar.append("AYLIK") # Ayın 1'i
```

---

## MODÜL 4: Personel Hijyen

### 4.1 Akıllı Personel Kontrol Paneli

5N1K

Soru	Cevap
NE?	Vardiya ve bölüm bazlı personel hijyen kontrol formu
NİÇİN?	Personel sağlık ve hijyen takibi, uygunsuzluk kaydı
NEREDE?	Ana Menü →  Personel Hijyen
NE ZAMAN?	Her vardiya başında veya kontrol gerektiğinde
NASIL?	Bölüm/Vardiya filtrele → Personeli seç → Durum belirle → Kaydet
KİM?	Vardiya Amiri, Kalite Sorumlusu, Admin

#### Teknik Detaylar

Bilgi	Değer
Tablo	hijyen_kontrol_kayitlari
Satırlar	app.py: 1021-1148
Fonksiyonlar	guvenli_coklu_kayit_ekle()

#### Durum Seçenekleri



Durum	Açıklama
UYGUN	Kontrolden geçti
UYGUN DEĞİL	Hijyen uygunsuzluğu - sebep ve aksiyon gerekli
GELMEDİ	İzinli veya devamsız
SAĞLIK RİSKİ	Sağlık sorunu tespit edildi

---

## MODÜL 5: Temizlik Kontrol

### 5.1 Saha Uygulama Çizelgesi

5N1K

Soru	Cevap
NE?	Master plandan günlük temizlik işlerini takip etme
NİÇİN?	Lokasyon bazlı temizlik uygulaması ve doğrulama
NEREDE?	Ana Menü →  Temizlik Kontrol →  Saha Uygulama Çizelgesi


NE ZAMAN?	Günlük, vardiya bazlı
NASIL?	Kat/Bölüm seç → Temizlik işleri listelenir → TAMAMLANDI/YAPILMADI işaretle
KİM?	Temizlik personeli (kayıt), Vardiya Amiri (kontrol)

#### Teknik Detaylar

Bilgi	Değer
Tablo	temizlik_kayitlari
Satırlar	app.py: 1159-1262

## 5.2 Master Plan Görüntüleme

5N1K

Soru	Cevap
NE?	Master temizlik planını salt-okunur görüntüleme
NİÇİN?	Plan bilgisine erişim (düzenleme Ayarlar'dan yapılır)
NEREDE?	<input type="checkbox"/> Temizlik Kontrol →  <input type="checkbox"/> Master Plan Düzenleme (READ-ONLY)
NASIL?	Sadece görüntüleme - düzenleme için Ayarlar modülüne yönlendirir
KİM?	Modüle erişimi olan herkes

#### Teknik Detaylar

Bilgi	Değer
Satırlar	app.py: 1263-1312


⚠ **KRİTİK:** Master Plan **sadece**  **Ayarlar** > ☐ **Temizlik & Bölümler** bölümünden düzenlenir!

---

## MODÜL 6: Kurumsal Raporlama

### 6.1 Rapor Kategorileri

5N1K

Soru	Cevap
NE?	Tüm modüllerden veri çekerek dashboard ve raporlar oluşturma
NİÇİN?	Yönetim karar desteği, izleme ve analiz
NEREDE?	Ana Menü →  Kurumsal Raporlama
NE ZAMAN?	Periyodik raporlama veya anlık sorgu
NASIL?	Tarih aralığı seç → Rapor kategorisi seç → Görüntüle/Export
KİM?	"Görüntüle" yetkisi olan tüm roller

## Teknik Detaylar

Bilgi	Değer
Satırlar	app.py: 1313-2229
Mod	READ-ONLY (Veri değişikliği yapılmaz)

### Rapor Tipleri

Rapor	Tablo Kaynağı
Üretim ve Verimlilik	depo_giris_kayitlari
Kalite (KPI) Analizi	urun_kpi_kontrol
Personel Hijyen Özeti	hijyen_kontrol_kayitlari
Temizlik Takip Raporu	temizlik_kayitlari
Lokasyon & Proses Haritası	lokasyonlar, proses_tipleri
Personel Organizasyon Şeması	personel, v_organizasyon_semasi

## MODÜL 7: ⚙️ Ayarlar

### 7.1 Genel Yapı

5N1K






Soru	Cevap
NE?	Sistem yapılandırması ve tanım yönetimi
NİÇİN?	Temel verilerin (personel, ürün, lokasyon) tanımlanması
NEREDE?	Ana Menü → ⚙️ Ayarlar
NE ZAMAN?	Sistem kurulumu ve güncelleme gerektiğinde
NASIL?	Alt sekmelerden ilgili modüle git → Ekle/Düzenle/Sil
KİM?	Admin (tam yetki), Yönetim (kısıtlı)

## Teknik Detaylar

Bilgi	Değer
Satırlar	app.py: 2230-4284





### Alt Sekmeler

Sekme	Tablo	Satırlar
Personel	personel	2257-2358
Kullanıcılar	personel	2880-3000
Ürünler	ayarlar_urunler, urun_parametreleri	3001-3125
Roller	ayarlar_roller	3126-3202
Bölümler	ayarlar_bolumler	3203-3357
Yetkiler	ayarlar_yetkiler	3358-3428

 Lokasyonlar	lokasyonlar	3429-3617
 Prosesler	proses_tipleri, lokasyon_proses_atama	3618-3725
 Temizlik & Bölümler	ayarlar_temizlik_plani, tanim_metotlar, kimyasal_envanter	3726-4096
  GMP Sorular	gmp_soru_havuzu	4097-4284

## 7.2 Personel Ekle/Düzenle Formu


5N1K

Soru	Cevap
NE?	Personel bilgilerini ekleme ve güncelleme formu
NİÇİN?	Organizasyon şeması ve yetkilendirme için personel kaydı
NEREDE?	  Ayarlar →  Personel →  Personel Ekle/Düzenle
NE ZAMAN?	Yeni personel işe girişinde veya bilgi güncellemesinde
NASIL?	Mod seç (Ekle/Düzenle) → Form doldur → Kaydet
KİM?	Admin, Yönetim

### Teknik Detaylar

Bilgi	Değer
Tablo	personel
Satırlar	app.py: 2280-2355
Form ID	personel_detay_form

### Form Alanları

UI Etiketi	DB Kolonu	Tip	Zorunlu
Ad Soyad	ad_soyad	TEXT	✓
Görev / Unvan	gorev	TEXT	✗
Vardiya	vardiya	TEXT	✓
Durum	durum	TEXT	✓
Departman	departman_id	INTEGER	✗
Bağlı Olduğu Yönetici	yonetici_id	INTEGER	✗
 Hiyerarşi Seviyesi	pozisyon_seviye	INTEGER	✓
Çalıştığı Kat	kat	TEXT	✗
Haftalık İzin	izin_gunu	TEXT	✗

### Pozisyon Seviyeleri (Hiyerarşi)

Seviye	Açıklama
0	Yönetim Kurulu
1	Genel Müdür / CEO
2	Direktör

- 3 Müdür
- 4 Şef / Sorumlu / Koordinatör
- 5 Personel (Varsayılan)
- 6 Stajyer / Çırac

### SQL Sorguları

```
-- UPDATE
UPDATE personel SET
    ad_soyad=:a, gorev=:g, departman_id=:d, yonetici_id=:y,
    vardiya=:v, durum=:st, kat=:k, izin_gunu=:iz, pozisyon_seviye=:ps
WHERE id=:id

-- INSERT
INSERT INTO personel
    (ad_soyad, gorev, departman_id, yonetici_id, vardiya, durum, kat, i
VALUES (:a, :g, :d, :y, :v, :st, :k, :iz, :ps)
```

### Validasyon Kuralları

Kural	Koşul	Hata Mesajı
Ad Soyad zorunlu	Boş olamaz	"Ad Soyad zorunludur"
Pozisyon seviye	0-6 arası integer -	

### Bağımlılıklar

Fonksiyon	Amaç
veri_getir("personel")	Mevcut personel listesi (düzenleme için)
ayarlar_bolumler tablosu	Departman dropdown
personel tablosu (self-join)	Yönetici dropdown

## 5. Kritik Notlar ve Dikkat Edilecekler

### 5.1 Kolon Adı Uyarıları

Tablo	YANLIŞ Kullanım	DOĞRU Kullanım	Açıklama
ayarlar_bolumler	ust_bolum_id	ana_departman_id	Recursive parent FK
tanim_metotlar	id	rowid	Bu tabloda id sütunu YOK
personel	Sadece departman_id	departman_id VEYA bolum (text)	Legacy destek için ikisini de kontrol et

### 5.2 Veri Tipi Dönüşümleri

```
# parent_id NULL olabilir, Integer'a çevirirken dikkat:
df['parent_id'] = pd.to_numeric(df['parent_id'], errors='coerce').fillna
```



```
df['id'] = pd.to_numeric(df['id'], errors='coerce').fillna(0).astype(int)
```

## 5.3 Personel Filtreleme Kuralları

Temizlik personeli listelenirken şu sırayla kontrol et:

1. departman\_id → ayarlar\_bolumler tablosunda "Temizlik" veya "Bulaşık" içeren bölümler
2. bolum (TEXT) → "Temizlik" içeriyorsa dahil et (Legacy)
3. gorev (TEXT) → "Temizlik" veya "Meydancı" içeriyorsa dahil et

## 5.4 Lokasyon Hiyerarşisi

Tip Sıralaması:

1. Kat (parent\_id = 0 veya NULL)
2. Bölüm (parent\_id = Kat.id)
3. Hat (parent\_id = Bölüm.id)
4. Ekipman (parent\_id = Hat.id veya Bölüm.id)

**ÖNEMLİ:** Lokasyon filtrelemede parent\_id kullanılır, ana\_departman\_id DEĞİL!

- lokasyonlar tablosu → parent\_id
- ayarlar\_bolumler tablosu → ana\_departman\_id

## 5.5 UI-DB Eşitleme (Büyük Eşitleme)

Sistemde veri bütünlüğünü korumak ve son kullanıcı deneyimini iyileştirmek için **Büyük Eşitleme** stratejisi uygulanmıştır:

1. **Arka Plan (DB):** Veritabanı sütun isimleri her zaman Teknik Dokümandaki teknik terimleri (urun\_id, lot\_kpi, risk\_seviyesi) kullanır.
2. **Ön Plan (UI):** Kullanıcının gördüğü etiketler (Labels) en anlaşılır Türkçe terimleri (Analiz Edilecek Ürün, Lot / Parti No, Risk Seviyesi) kullanır.
3. **Mapping (Kod):** app.py içerisindeki formlar ve SQL sorguları, UI etiketlerini DB kolonlarına %100 uyumlu şekilde map eder. Bu sayede "İsim Karmaşası" (Naming Ambiguity) engellenmiştir.

---

## 6. İş Kuralları (Business Logic)

### 6.1 Master Temizlik Planı Oluşturma

AKIŞ:

1. Kat Seç → lokasyonlar WHERE tip='Kat'
2. Bölüm/Hat Seç → lokasyonlar WHERE parent\_id IN (Kat ve alt düğümleri)
3. Alan Tipi Seç:
  - Ekipman → lokasyonlar WHERE tip='Ekipman' AND parent\_id=seçilen\_bö
  - Yapısal → Statik liste (Zemin, Duvar, Tavan...)
4. Sorumlu Ata:
  - Uygulayıcı → personel WHERE departman\_id IN (Temizlik departmanlar
  - 1. Kontrol → Rol bazlı (Vardiya Amiri)
  - 2. Kontrol → Rol bazlı (Kalite Güvence)
5. Kaydet → ayarlar\_temizlik\_plani

## 6.2 Personel Departman Hiyerarşisi

```
# Temizlik departmanları ve alt departmanlarını bulma:
def get_cleaning_department_ids(engine):
    depts = pd.read_sql("SELECT id, bolum_adi, ana_departman_id FROM ay
    target_ids = set()

    # "Temizlik" veya "Bulaşık" içerenleri bul
    parents = depts[depts['bolum_adi'].str.contains("Temizlik|Bulaşık",
    target_ids.update(parents['id'].tolist())

    # Alt departmanları recursive olarak bul
    current = list(target_ids)
    for _ in range(3): # Max 3 seviye
        children = depts[depts['ana_departman_id'].isin(current)]
        if children.empty: break
        target_ids.update(children['id'].tolist())
        current = children['id'].tolist()

    return target_ids
```

## 6.3 Yetki Kontrol Akışı

1. Kullanıcı giriş yapar → session\_state['user\_rol'] = "Kalite Sorumlus
2. Modüle tıklar → kullanıcı\_yetkisi\_var\_mi(" Temizlik Kontrol", "Görü
3. Sistem kontrol eder:
  - Admin mi? → True
  - Değilse → ayarlar\_yetkiler tablosundan (rol\_adi, modul\_adi) çekip
4. "Yok" → st.error() ve st.stop()
5. "Görüntüle" veya "Düzenle" → İzin ver

## 6.4 Temizlik Kontrol Modülü Yapısı

□ Temizlik Kontrol modülü 2 sekmeden oluşur:

### Sekme 1: Saha Uygulama Çizelgesi (Operasyonel)

```
# Satır: 1186-1268 (app.py)
with tab_uygulama:
    # 1. Master Plandan günlük işleri çek
    plan_df = veri_getir("Ayarlar_Temizlik_Planı")

    # 2. Kat/Bölüm ve Vardiya seç
    secili_kat = st.selectbox("Denetlenecek Kat / Bölüm", kat_listesi)

    # 3. Yetki kontrolü
    is_controller = (user in CONTROLLER_ROLES) or (user in ADMIN_USERS)

    # 4. Form ile kayıt gir (temizlik_kayit_formu)
    with st.form("temizlik_kayit_formu"):
        # Ekipman/Alan bazında durum girişi (TAMAMLANDI / YAPILMADI)
        # Verifikasyon sonuçları (ATP, Swap vb.)
```

# → temizlik\_kayitlari tablosuna INSERT

### Özellikler:

- Sadece **yetkili roller** kayıt girebilir (Admin, Kalite, Vardiya Amiri)
- Her ekipman/alan için durum ve doğrulama sonucu girilir
- Kayıtlar temizlik\_kayitlari tablosuna yazılır

### Sekme 2: ⚙️ Master Plan Düzenleme (READ-ONLY)

```
# Satır: 1270-1322 (app.py)
with tab_master_plan:
    # 1. Ayarlar'daki planı çek
    master_df = pd.read_sql("SELECT * FROM ayarlar_temizlik_plani", eng

    # 2. READ-ONLY Dataframe göster
    st.dataframe(display_df, use_container_width=True, hide_index=True,

    # 3. Bilgilendirme mesajı
    st.info("Değişiklik yapmak için ⚙️ Ayarlar > Temizlik Yönetimi sayı
```

### Özellikler:

- **Salt-okunur** mod (kullanıcı değişiklik yapamaz)
- Ayarlar modülündeki ayarlar\_temizlik\_plani tablosunu gösterir
- Güncelleme butonu YOK (veri güvenliği için)

### ⚠️ KRİTİK KURAL:

Master Temizlik Planı **sadece Ayarlar > Temizlik Yönetimi** sayfasında düzenlenir.  
Temizlik Kontrol modülü bu planı sadece **görüntüler ve uygular**.  
Bu yaklaşım, planlama ile uygulama süreçlerini ayırır ve veri bütünlüğünü korur.

## 6.5 Veri Senkronizasyonu (Lokal ↔ Canlı)

Lokal → Canlı:

1. Lokal DB'den veri çek (SELECT \*)
2. Canlı DB'ye INSERT (UPSERT mantığı yok, dikkat!)
3. Cache temizle (st.cache\_data.clear())

Canlı → Lokal:

1. Canlı DB'den veri çek
2. Lokal DB'yi truncate et (DELETE FROM)
3. Lokal DB'ye INSERT

## 6.6 KPI & Kalite Kontrol Dinamik Parametre Yapısı

🧠 **KPI & Kalite Kontrol** modülü, statik sütunlar yerine **Dinamik Parametre** mimarisini kullanır:

1. **Parametre Yükleme:** Seçilen ürüne ait spesifikasyonlar urun\_parametreleri tablosundan çalışma anında (runtime) çekilir.

2. **Dinamik Form:** Çekilen parametre sayısı ve tipine göre Streamlit formunda otomatik giriş alanları oluşturulur.
3. **Limit Kontrolü (Decision Logic):** Girilen değerler, DB'den gelen min\_deger ve max\_deger aralığında mı diye otomatik kontrol edilir.
  - Tüm değerler uygunsa → karar = "UYGUN"
  - Bir değer dahi limit dışıysa → karar = "UYGUN DEĞİL"
4. **Kayıt:** Sonuçlar Urun\_KPI\_Kontrol tablosuna; ham ölçüm değerleri ise detayli\_veri sütununa JSON string formatında kaydedilir.

## 6.7 Üretim Kayıt Girişi ve Veri Akışı

 **Üretim Kayıt modülü,** fabrikanın günlük çıktılarını takip eder:

1. **Ürün Filtreleme:** Departman sorumluları sadece kendi bölümlerine ait ürünleri görebilir (bolum\_bazli\_urun\_filtrele fonksiyonu).
2. **Kayıt Doğrulama:** lot\_no ve miktar > 0 kontrolü zorunludur.
3. **Veri Tablosu:** Kayıtlar Depo\_Giris\_Kayitlari tablosuna URETIM tipi ile işlenir.
4. **Otomatik İstatistik:** Kayıt yapıldıktan sonra Dashboard'daki "Toplam Üretim" ve "Fire" metrikleri veri\_getir cache temizliği ile anında güncellenir.

---

## 7. RBAC Yetkilendirme Sistemi

### 7.1 Rol Hiyerarşisi

```
graph TD
  A[Admin] --> B[Yönetim]
  B --> C[Kalite Sorumlusu]
  C --> D[Vardiya Amiri]
  D --> E[Bölüm Sorumlusu]
  E --> F[Personel]
```

### 7.2 Modül Eşlemesi

**constants.py** içinde tanımlı:

```
MODUL_ESLEME = {
  "🏭 Üretim Girişi": "Üretim Girişi",
  "📊 KPI & Kalite Kontrol": "KPI Kontrol",
  "🛡️ GMP Denetimi": "GMP Denetimi",
  "👤 Personel Hijyen": "Personel Hijyen",
  "🧼 Temizlik Kontrol": "Temizlik Kontrol",
  "📋 Kurumsal Raporlama": "Raporlama",
  "⚙️ Ayarlar": "Ayarlar"
}
```

### 7.3 Yetki Kontrol Fonksiyonları

```
kullanici_yetkisi_getir(rol_adi, modul_adi)
```

**Amaç:** Veritabanından rol-modül yetki ilişkisini çeker

**Parametreler:**

- rol\_adi (str): Rol adı
- modul\_adi (str): Modül adı

**Dönüş:** "Yok" | "Görüntüle" | "Düzenle"

**Cache:** TTL = 300 saniye (5 dakika)

```
@st.cache_data(ttl=300)
def kullanıcı_yetkisi_getir(rol_adi, modul_adi):
    try:
        with engine.connect() as conn:
            sql = text("""
                SELECT erisim_turu FROM ayarlar_yetkiler
                WHERE rol_adi = :rol AND modul_adi = :modul
            """)
            result = conn.execute(sql, {"rol": rol_adi, "modul": modul_adi})
            return result[0] if result else "Yok"
    except:
        return "Yok"
```

**kullanici\_yetkisi\_var\_mi(menu\_adi, gereken\_yetki)**

**Amaç:** Kullanıcının modüle erişim yetkisini kontrol eder

**Parametreler:**

- menu\_adi (str): Menü adı (emoji ile)
- gereken\_yetki (str): "Görüntüle" veya "Düzenle"

**Dönüş:** bool

**Özel Durum:** Admin her zaman True döner

```
def kullanıcı_yetkisi_var_mi(menu_adi, gereken_yetki="Görüntüle"):
    user_rol = st.session_state.get('user_rol', 'Personel')

    # Admin bypass
    if user_rol == 'Admin':
        return True

    modul_adi = MODUL_ESLEME.get(menu_adi, menu_adi)
    erisim = kullanıcı_yetkisi_getir(user_rol, modul_adi)

    if gereken_yetki == "Görüntüle":
        return erisim in ["Görüntüle", "Düzenle"]
    elif gereken_yetki == "Düzenle":
        return erisim == "Düzenle"
    return False
```

## 7.4 Bölüm Bazlı Filtreleme

**bolum\_bazli\_urun\_filtrele(urun\_df)**

**Amaç:** Bölüm Sorumlusu rolündeki kullanıcılar için ürün listesini filtreler

**Mantık:**

1. Admin/Yönetim/Kalite → Tüm ürünler
2. Bölüm Sorumlusu → Sadece kendi bölümünün ürünleri
3. Hiyerarşik eşleşme: "Üretim > Pataşu" içinde "Pataşu" geçiyorsa göster

```
def bolum_bazli_urun_filtrele(urun_df):
    user_rol = st.session_state.get('user_rol', 'Personel')
    user_bolum = st.session_state.get('user_bolum', '')

    if user_rol in ['Admin', 'Yönetim', 'Kalite Sorumlusu', 'Vardiya Am
        return urun_df

    if 'sorumlu_departman' in urun_df.columns and user_bolum:
        mask_bos = urun_df['sorumlu_departman'].isna() | (urun_df['soru
        mask_eslesme = urun_df['sorumlu_departman'].astype(str).str.con
        return urun_df[mask_bos | mask_eslesme]

    return urun_df
```

---

## 8. Cache Sistemi

### 8.1 Cache Stratejisi

EKLERİSTAN QMS, performans optimizasyonu için Streamlit'in `@st.cache_data` ve `@st.cache_resource` dekoratörlerini kullanır.

### 8.2 Cache Türleri

#### Resource Cache - Veritabanı Bağlantısı

```
@st.cache_resource
def init_connection():
    if "DB_URL" in st.secrets:
        db_url = st.secrets["DB_URL"]
        return create_engine(
            db_url,
            pool_size=10,
            max_overflow=20,
            pool_pre_ping=True,
            pool_recycle=300
        )
    else:
        db_url = 'sqlite:///ekleristan_local.db'
        return create_engine(db_url, connect_args={'check_same_thread':
```

**Özellikler:**

- Connection pooling (10 + 20 overflow)
- Pre-ping: Bağlantı kopmalarını otomatik algıla
- Recycle: 5 dakikada bir bağlantıları yenile

## Data Cache - Sorgu Sonuçları

Fonksiyon	TTL	Amaç
run_query()	600s	Genel SQL sorguları
get_user_roles()	3600s	Rol listeleri
get_department_hierarchy()	600s	Departman hiyerarşisi
get_personnel_hierarchy()	5s	Organizasyon şeması (sık değişir)
cached_veri_getir()	60s	Tablo verileri
kullanici_yetkisi_getir()	300s	Yetki sorguları

## 8.3 Cache Invalidation

### Manuel Temizleme:

```
# Tüm cache'i temizle
st.cache_data.clear()

# Belirli fonksiyonu temizle
cached_veri_getir.clear()
```

### Otomatik Temizleme:

Veri değişikliği sonrası:

```
if guvenli_kayit_ekle("Depo_Giris_Kayitlari", yeni_kayit):
    cached_veri_getir.clear() # Cache'i temizle
    st.success("Kaydedildi!")
    st.rerun() # Sayfayı yenile
```

---

## 9. Fonksiyon Referansı

### 9.1 Veritabanı Fonksiyonları

**veri\_getir(tablo\_adi)**

**Amaç:** Belirli tablodan veri çeker (cached)

#### Parametreler:

- **tablo\_adi (str):** Tablo adı (örn: "Ayarlar\_Personel")

**Dönüş:** pd.DataFrame

#### Desteklenen Tablolar:

- Ayarlar\_Personel

- Ayarlar\_Urunler
- Depo\_Giris\_Kayitlari
- Ayarlar\_Fabrika\_Personel
- Ayarlar\_Temizlik\_Plani
- Tanim\_Bolumler
- Kimyasal\_Envanter
- GMP\_Soru\_Havuzu
- Ayarlar\_Bolumler

**guvenli\_kayit\_ekle(tablo\_adi, veri)**

**Amaç:** Veritabanına güvenli kayıt ekleme

**Parametreler:**

- `tablo_adi` (str): Hedef tablo
- `veri` (list): Sıralı veri listesi

**Dönüş:** bool (Başarı/Hata)

**Desteklenen Tablolar:**

- Depo\_Giris\_Kayitlari
- Urun\_KPI\_Kontrol

## 9.2 Yardımcı Fonksiyonlar

**get\_istanbul\_time()**

**Amaç:** İstanbul saat diliminde şu anki zamanı döndürür

**Dönüş:** datetime object

```
def get_istanbul_time():
    return datetime.now(pytz.timezone('Europe/Istanbul')) if 'Europe/Is
```

**get\_position\_name(level) - constants.py**

**Amaç:** Pozisyon seviyesinden isim döndürür

**Parametreler:**

- `level` (int): 0-6 arası seviye

**Dönüş:** str

**Eşleme:**

- 0 → "Yönetim Kurulu"
- 1 → "Genel Müdür"
- 2 → "Direktörler"
- 3 → "Müdürler"
- 4 → "Şef/Koordinatör"
- 5 → "Personel"



- 6 → "Stajyer/Geçici"
- 

## 10. Deployment

### 10.1 Yerel Deployment

#### Gereksinimler:

- Python 3.8+
- pip

#### Adımlar:

```
# 1. Bağımlılıkları yükle
pip install -r requirements.txt
```

```
# 2. Graphviz yükle (Windows)
choco install graphviz
```

```
# 3. Uygulamayı başlat
streamlit run app.py
```

### 10.2 Streamlit Cloud Deployment

#### Adım 1: GitHub'a Push

```
git add .
git commit -m "Deploy to Streamlit Cloud"
git push origin main
```

#### Adım 2: Streamlit Cloud Konfigürasyonu

1. [share.streamlit.io](https://share.streamlit.io) → New App
2. Repository seç: emrecavdar83/EKLER-STAN\_QMS
3. Branch: main
4. Main file: app.py

#### Adım 3: Secrets Ekle

```
# .streamlit/secrets.toml
DB_URL = "postgresql://user:password@host:5432/database"
```

#### Adım 4: Advanced Settings

```
# .streamlit/config.toml
[server]
maxUploadSize = 200
enableCORS = false

[theme]
primaryColor = "#8B0000"
backgroundColor = "#FFFFFF"
secondaryBackgroundColor = "#F0F2F6"
```

## 10.3 Supabase Kurulumu

### Adım 1: Proje Oluştur

1. [supabase.com](https://supabase.com) → New Project
2. Proje adı: ekleristan-qms
3. Database password belirle
4. Region: Europe (Frankfurt)

### Adım 2: Connection String Al

Settings → Database → Connection String → URI

```
postgresql://postgres:[PASSWORD]@db.[PROJECT-REF].supabase.co:5432/post
```

### Adım 3: SQL Migration Çalıştır

SQL Editor'de sırayla çalıştır:

1. sql/supabase\_personel\_org\_restructure.sql
2. sql/rbac\_database\_setup.sql
3. sql/database\_indexes.sql
4. Diğer migration dosyaları

### Adım 4: RLS (Row Level Security) Devre Dışı Bırak

⚠️ Production'da RLS kullanılması önerilir

```
ALTER TABLE personel DISABLE ROW LEVEL SECURITY;  
ALTER TABLE ayarlar_bolumler DISABLE ROW LEVEL SECURITY;  
-- Diğer tablolar için tekrarla
```

---

## 11. Performans Optimizasyonu

### 11.1 Veritabanı İndeksleri

```
-- Personel tablosu  
CREATE INDEX IF NOT EXISTS idx_personel_departman ON personel(departman)  
CREATE INDEX IF NOT EXISTS idx_personel_yoneticisi ON personel(yoneticisi)  
CREATE INDEX IF NOT EXISTS idx_personel_seviye ON personel(pozisyon_seviye)  
  
-- Üretim kayıtları  
CREATE INDEX IF NOT EXISTS idx_depo_tarih ON depo_giris_kayitlari(tarih)  
CREATE INDEX IF NOT EXISTS idx_depo_urun ON depo_giris_kayitlari(urun);  
  
-- GMP kayıtları  
CREATE INDEX IF NOT EXISTS idx_gmp_tarih ON gmp_denetim_kayitlari(tarih)  
CREATE INDEX IF NOT EXISTS idx_gmp_lokasyon ON gmp_denetim_kayitlari(lo
```

### 11.2 Query Optimizasyonu

Kötü:

```
# Her seferinde tüm tabloyu çek
df = pd.read_sql("SELECT * FROM depo_giris_kayitlari", engine)
df_filtered = df[df['tarih'] == today]
```

**İyi:**

```
# Filtreyi SQL seviyesinde uygula
sql = "SELECT * FROM depo_giris_kayitlari WHERE tarih = :t"
df = pd.read_sql(text(sql), engine, params={"t": today})
```

## 11.3 Streamlit Optimizasyonu

**Fragment Kullanımı:**

```
@st.fragment
def expensive_component():
    # Sadece bu bölüm yeniden render edilir
    data = fetch_data()
    st.dataframe(data)
```

**Session State:**

```
# Veriyi session state'de sakla
if 'data' not in st.session_state:
    st.session_state.data = fetch_expensive_data()

# Tekrar kullan
df = st.session_state.data
```

---

# 12. Güvenlik

## 12.1 Şifre Yönetimi

⚠️ **Kritik Güvenlik Açığı:** Şifreler plain text olarak saklanıyor!

**Mevcut Durum:**

```
# Güvensiz!
db_pass = str(u_data.iloc[0]['sifre']).strip()
if input_pass == db_pass:
    # Giriş başarılı
```

**Önerilen Çözüm:**

```
import hashlib

def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()

# Kayıt
hashed = hash_password("12345")
# DB'ye hashed değeri kaydet
```

```
# Giriş
if hash_password(input_pass) == db_hashed:
    # Giriş başarılı
```

## 12.2 SQL Injection Koruması

### Güvenli Parametre Kullanımı:

```
# Güvenli
sql = text("SELECT * FROM personel WHERE kullanıcı_adi = :user")
result = conn.execute(sql, {"user": username})

# Güvensiz (KULLANMAYIN!)
sql = f"SELECT * FROM personel WHERE kullanıcı_adi = '{username}'"
```

## 12.3 Session Yönetimi

### Session Timeout:

```
# config.toml
[server]
sessionTimeout = 1800 # 30 dakika
```

### Çıkış Yönetimi:

```
if st.button("Çıkış Yap"):
    st.session_state.logged_in = False
    st.session_state.clear() # Tüm session'ı temizle
    st.rerun()
```

---

## 🔗 Geliştirici Desteği

### Teknik Sorular:

- GitHub Issues: [github.com/emrecavdar83/EKLER-STAN QMS/issues](https://github.com/emrecavdar83/EKLER-STAN-QMS/issues)

### Katkıda Bulunma:

- Pull Request açın
- Kod standartlarına uyun (PEP 8)
- Test senaryoları ekleyin

---

**Son Güncelleme:** 27 Ocak 2026

**Versiyon:** 1.5 (5N1K Formatında Modül Detayları - AI Revizyon Referansı Eklendi)