

SQL & Linux Öğrenme Notları (Learning SQL & Linux Notes)

1. Giriş

Bugünkü konularım SQL ve Linux oldu.

- SQL, yani veritabanı dili, test mühendisliğinde diğer yazılım dallarında da olduğu gibi önemli çünkü **veritabanından veri sorgulama**, test senaryolarında arka planda ne döndüğünü görme ve gerektiğinde **data manipülasyonu** yapma imkânı veriyor.
 - Linux tarafı ise test ortamlarında **logları inceleme, süreçleri yönetme** ve otomasyon scriptleri yazma açısından kritik.
 - Yani aslında her test mühendisi biraz SQL bilir, biraz da Linux'a hâkimdir. Ben de bu yüzden bugün bu iki konunun temellerine odaklandım.
-

2. SQL (Structured Query Language)

2.1 Temel SQL Komutları (Basic SQL Commands)

Araştırdığımda en sık kullanılan komutlar şunlar oldu:

- **SELECT:** Veritabanından veri çekmek.

```
SELECT * FROM Customers WHERE City='Ankara';
```

- **INSERT:** Yeni kayıt eklemek.

```
INSERT INTO Customers (Name, City) VALUES ('Ali', 'İstanbul');
```

- **UPDATE:** Var olan kaydı güncellemek.

```
UPDATE Customers SET City='İzmir' WHERE Name='Ali';
```

- **DELETE:** Veri silmek.

```
DELETE FROM Customers WHERE Name='Ali';
```

Ek olarak:

- **WHERE** (filtreleme),

- **ORDER BY** (sıralama),
- **GROUP BY** (gruplama),
- **JOIN** (tabloları birleştirme) → özellikle **INNER JOIN**, **LEFT JOIN** çok sık kullanılıyor.

JOIN Örneği

```
SELECT Orders.OrderID, Customers.Name
FROM Orders
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
```

Ararken “**basic SQL queries examples**” diye bakmam yeterli oldu.

2.2 Stored Procedures

- **Tanım:** Tekrar tekrar yazmak yerine, bir defa yazıp sakladığımız SQL blokları. Bir nevi fonksiyon gibi düşünebilirim.
- **Ne zaman kullanılır?** Eğer sürekli aynı sorguyu çalıştırmam gerekiyorsa (örneğin günlük satış raporu) bunu stored procedure haline getiririm.

Örnek Stored Procedure

```
CREATE PROCEDURE GetOrdersByCity @City NVARCHAR(50)
AS
BEGIN
    SELECT * FROM Orders
    WHERE CustomerCity = @City;
END;
```

- **Avantajları:**
 - Performansı artırır (önceden compile edilir).
 - Tekrarlayan işler için pratik.
 - **Dezavantajları:**
 - Bakımı zor olabilir.
 - Versiyon kontrolü zayıf (kod gibi Git'te kolay izlenemiyor).
-

2.3 ETL (Extract, Transform, Load)

- **Tanım:**
 - **Extract:** Veriyi bir kaynaktan almak.
 - **Transform:** İstediğim formata dönüştürmek (örn: tarih formatı değiştirmek).
 - **Load:** Hedef sisteme yüklemek.

- **Test süreçlerinde önemi:** Örneğin bir bankada müşteri hareketleri farklı tablolardan çekilip günlük rapor tablosuna yüklenir. Eğer ETL hatalıysa, müşteri bakiyesi yanlış görünebilir. Bir test mühendisi olarak ETL'i anlamak, "kaynak veri ↔ hedef veri" karşılaştırmasını test etmem açısından önemli.
-

2.4 MSSQL ve GaussDB Bağlantısı

MSSQL (Microsoft SQL Server)

- **Geliştirici & Tarihçe** Microsoft tarafından geliştirilen, kurumsal seviyede **en çok kullanılan ilişkisel veritabanı (RDBMS)** çözümlerinden biridir.
- **Özellikler**
 - **Stored Procedure, Trigger, View** gibi gelişmiş özellikler.
 - **Transaction desteği (ACID uyumluluğu)** sayesinde güvenli veri işlemleri.
 - **T-SQL (Transact-SQL)** isimli özel bir SQL dili kullanır.
 - **SSIS (Integration Services), SSRS (Reporting Services)** ve **SSAS (Analysis Services)** gibi ek modüllerle ETL, raporlama ve analiz yapılabilir.
- **Avantajları**
 - Güçlü **GUI aracı (SQL Server Management Studio – SSMS)** ile kullanım kolaylığı.
 - Kurumsal firmalarda çok yaygın → iş piyasasında avantaj.
 - Transaction yönetimi ve güvenlik özellikleri oldukça güçlü.
- **Dezavantajları**
 - Lisans maliyeti yüksek.
 - Genellikle Windows tabanlı çalışır (Linux desteği sonradan geldi).

Örnek Senaryo: Bir e-ticaret sitesinin sipariş tablosunu tutmak, raporlamak ve her gün **stored procedure** ile günlük satış raporları üretmek için MSSQL sıkça kullanılır.

GaussDB (Huawei Gauss Database)

- **Geliştirici & Tarihçe** Huawei tarafından geliştirilen, özellikle **Huawei Cloud** üzerinde çalışan yeni nesil veritabanı çözümüdür.
- **Mimari**
 - **Distributed Database (Dağıtık mimari)** → büyük veri kümeleri için ölçeklenebilir.
 - **HTAP (Hybrid Transactional and Analytical Processing)** desteği → hem OLTP (transactional işlemler) hem OLAP (analitik sorgular) yapılabilir.
- **Özellikler**
 - PostgreSQL tabanlıdır → yani PostgreSQL'in özelliklerini genişletir.
 - **AI-powered O&M (Operation & Maintenance):** Yapay zeka ile performans optimizasyonu ve hata tahmini.
 - **High Availability (Yüksek Erişilebilirlik):** Otomatik failover ve yedekleme.

- **Security by Design:** Veri şifreleme ve güvenlik standartlarına uyum.

- **Avantajları**

- Huawei Cloud ekosistemi ile sıkı entegrasyon.
- Açık kaynak PostgreSQL uyumlu → PostgreSQL bilenler için öğrenmesi kolay.
- Ölçeklenebilir, özellikle büyük veride performanslı.

- **Dezavantajları**

- MSSQL kadar yaygın değil, daha çok Huawei ekosisteminde karşılaşılır.
- Belgelendirme ve topluluk desteği MSSQL kadar büyük değil.

Örnek Senaryo: Bir telekom operatörünün müşteri hareketlerini milyarlarca satır halinde saklayıp, hem günlük faturalandırma (transactional) hem de aylık kullanım analizi (analytical) yapması gerektiğinde GaussDB tercih edilebilir.

MSSQL vs GaussDB Karşılaştırması

Özellik	MSSQL	GaussDB
Geliştirici	Microsoft	Huawei
Mimari	Tek sunucu (cluster opsiyonel)	Dağıtık, PostgreSQL tabanlı
Kullanım Alanı	Kurumsal (bankacılık, ERP)	Büyük veri, Huawei Cloud, telekom
Lisans	Ticari, ücretli	Açık kaynak uyumlu + Huawei sürümü
Popülerlik	Çok yüksek, dünya çapında	Huawei ekosisteminde yaygın
Öğrenme Kaynağı	Çok geniş topluluk, dökümantasyon	Daha sınırlı, ama gelişiyor
Güvenlik & Transaction	Çok güçlü (ACID uyumlu)	Güçlü, dağıtık mimari + AI optimizasyon

Özetle:

- **MSSQL** → Geleneksel, kurumsal dünyada çok güçlü ve yaygın.
- **GaussDB** → Huawei'nin modern, dağıtık ve AI destekli yaklaşımı.

3. Linux

3.1 Temel Linux Komutları

Test ortamlarında log okumak ve dosya yönetmek için şu komutları araştırdım:

- **ls, cd, pwd** → Dosya/dizin gezmek
- **cp, mv, rm** → Dosya kopyalama, taşıma, silme
- **cat, less, tail** → Dosya görüntüleme (özellikle **tail -f** log takip için çok önemli)
- **chmod, chown** → Dosya izinlerini değiştirme
- **ps, top, kill** → Process yönetimi

Araştırırken “**basic Linux commands for beginners**” diye baktım.

3.2 Shell Scripting

- **Tanım:** Linux üzerinde Bash ya da sh ile yazdığım küçük otomasyon scriptleri.
- **Kullanım Alanı:** Logları kontrol etmek, otomatik testleri başlatmak, sistem rutinlerini yönetmek.

Örneğin: Bir dosyanın satır sayısını bulan script

```
#!/bin/bash
wc -l myfile.txt
```

Ya da log'da “ERROR” kelimesini bulan script:

```
#!/bin/bash
grep "ERROR" /var/log/app.log
```

Bu tarz küçük scriptler test süreçlerinde büyük kolaylık sağlıyor. Aslında bu basit adımlar, ileride Jenkins veya CI/CD pipeline içine gömülen otomasyonların da ilk basamağı oluyor.

3.3 MobaXterm

- **Tanım:** Windows üzerinde Linux serverlarına bağlanmak için kullanılan bir terminal emülatörü.
- **Özellikleri:** SSH, SCP, SFTP, X11 desteği → yani sadece terminal değil, dosya transferi ve görsel uygulama desteği de var.
- **Kullanımı:** Kur → Server bilgilerini gir → SSH bağlantısı aç → Linux komutlarını yaz.

Ben **Ubuntu** kullandığım için, aslında MobaXterm'e ihtiyacım yok. Ubuntu'nun kendi terminali zaten SSH bağlantısını destekliyor. Yani pratikte ben direkt Ubuntu terminalinden Linux serverlarına bağlanabiliyorum. Bu yüzden MobaXterm'i sadece **teorik olarak öğrendim**, “Windows kullanıcıları için çözüm” olduğunu fark ettim.

4. Kapanış

Bugün SQL ve Linux üzerine çalıştım.

- **SQL tarafında:** SELECT, JOIN gibi temel sorgular, stored procedure mantığı ve ETL süreçlerini öğrenmek bana test senaryolarında **data validation** yaparken ciddi katkı sağlayacak.
- **Linux tarafında:** Komutlar ve shell script örnekleriyle, test ortamında logları analiz etmenin ne kadar kritik olduğunu gördüm.
- **MobaXterm tarafında:** Teorik olarak araştırdım ama Ubuntu kullandığım için kendi terminalimden doğrudan Linux serverlarına bağlanabiliyorum.

Genel olarak, bu bilgiler bana projelerde:

- Veritabanı testlerinde doğru sorgular yazmayı,

- Ortamlarda logları hızlıca kontrol etmeyi,
- Otomasyon için basit scriptler geliştirmeyi,
- ETL süreçlerinde “kaynak → hedef” veri doğruluğunu test etmeyi sağlayacak.