

# YMH339 Veri Tabanı Yönetim Sistemleri

## Laboratuvar Uygulaması: VTYS Mimarisi ve Bileşenleri

Bu laboratuvar çalışmasında, Veritabanı Yönetim Sistemlerinin (VTYS) temel mimari yapısını ve ana bileşenlerinin (DDL, DML, Sorgu İşlemcisi, Depolama Yöneticisi, İndeksler, Buffer Pool) MySQL veritabanı üzerinde nasıl çalıştığını pratik örneklerle deneyimleyeceksiniz.

### 1. Amaç

- VTYS mimarisinin temel bileşenlerini (DDL, DML, Sorgu İşlemcisi, Depolama Yöneticisi, İndeksler, Buffer Pool) anlamak.
- MySQL üzerinde bu bileşenlerin işleyişini gösteren SQL komutlarını uygulamak ve çıktılarını incelemek.
- Veritabanı performansını etkileyen faktörler hakkında bilgi edinmek.

### 2. Ön Hazırlık

- MySQL sunucunuzun çalıştığından emin olun.
- MySQL Workbench veya tercih ettiğiniz başka bir MySQL istemcisini açın.
- Yeni bir veritabanı oluşturun (örneğin `lab_db`) ve bu veritabanını kullanın:

```
CREATE DATABASE IF NOT EXISTS lab_db;
```

```
USE lab_db;
```

### 3. Veri Tanımlama Dili (DDL) ve Veri Manipülasyon Dili (DML)

**Tanım:** DDL (Data Definition Language), veritabanı şemasını tanımlamak ve düzenlemek için kullanılan komutları içerir (örneğin CREATE, ALTER, DROP). DML (Data Manipulation Language) ise veritabanındaki verileri manipüle etmek için kullanılır (örneğin SELECT, INSERT, UPDATE, DELETE).

#### Uygulama Adımları:

##### 1. Tablo Oluşturma (DDL - CREATE): Bir `Urunler` tablosu oluşturalım.

```
CREATE TABLE Urunler (
    urun_id INT PRIMARY KEY AUTO_INCREMENT,
    urun_adi VARCHAR(100) NOT NULL,
    fiyat DECIMAL(10, 2) NOT NULL,
```

```
    stok_adedi INT DEFAULT 0
);
```

- 2. Tablo Yapısını Görüntüleme (DESCRIBE):** Urunler tablosunun şema durumunu aşağıdaki SQL kodlarını tek tek çalıştırarak inceleyelim.

```
SHOW CREATE TABLE Urunler;
```

```
DESCRIBE Urunler;
```

```
SELECT COLUMN_NAME, COLUMN_TYPE, IS_NULLABLE, COLUMN_DEFAULT FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Urutler' AND TABLE_SCHEMA = DATABASE();
```

- 3. Tablo Yapısını Değiştirme (DDL - ALTER):** Urutler tablosuna eklenme\_tarihi sütunu ekleyelim. Ardından INFORMATION\_SCHEMA.COLUMNS ile kontrol ediniz.

```
ALTER TABLE Urutler
```

```
ADD COLUMN eklenme_tarihi DATETIME DEFAULT CURRENT_TIMESTAMP;
```

```
SELECT COLUMN_NAME, COLUMN_TYPE, IS_NULLABLE, COLUMN_DEFAULT FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'Urutler' AND TABLE_SCHEMA = DATABASE();
```

- 4. Veri Ekleme (DML - INSERT):** Urutler tablosuna yeni ürünler ekleyelim.

```
INSERT INTO Urutler (urut_adi, fiyat, stok_adedi)
VALUES (
    'Laptop', 12500.00, 50
),
(
    'Klavye', 450.00, 150
),
(
    'Mouse', 200.00, 200
);
```

- 5. Veri Sorulama (DML - SELECT):** Tüm ürünleri ve belirli ürünleri sırasıyla listeleyelim. EXPLAIN ile yürütüm planını inceleyiniz.

```
SELECT * FROM Urutler;
```

```
SELECT COUNT(*) AS n, SUM(stok_adedi) AS stok_toplami, AVG(fiyat) AS ort_fiyat FROM Urutler;
```

```
SELECT * FROM Urutler ORDER BY fiyat;
```

- 6. Veri Güncelleme (DML - UPDATE):** 'Laptop' ürününün fiyatını güncelleyelim.

```
UPDATE Urunler
SET fiyat = 13000.00
WHERE urun_adi = 'Laptop';
```

7. **Veri Silme (DML - DELETE):** Stok adedi 100'ün altında olan ürünleri silelim.

```
DELETE FROM Urunler
WHERE stok_adedi < 100;
```

8. **Tablo Silme (DDL - DROP):** Urunler tablosunu silelim (Bu komutu dikkatli kullanın, tüm veriyi ve tablo yapısını siler).

```
DROP TABLE Urunler;
```

#### DDL ve DML Soruları:

1. ALTER TABLE komutunun CREATE TABLE komutundan farkı nedir? Hangi durumlarda hangisi tercih edilmelidir?
2. INSERT, UPDATE, DELETE ve SELECT komutlarının veritabanı yönetimindeki temel görevlerini açıklayınız.
3. Bir DDL komutu çalıştırığınızda, MySQL'in içsel olarak hangi adımları izlediğini düşünüyorsunuz? (İpucu: Ders materyallerindeki DDL Compiler bölümünü hatırlayın).

#### 4. Soru İşlemcisi (Query Processor)

**Tanım:** Soru İşlemcisi, kullanıcı tarafından gönderilen SQL sorgularını analiz eden, optimize eden ve veritabanı motorunun anlayabileceği düşük seviyeli talimatlara dönüştüren VTYS bileşenidir. Ana bileşenleri arasında ayrıştırıcı (parser), soru iyileştirici (query optimizer) ve soru yürütücü (query executor) bulunur.

#### Uygulama Adımları:

1. **Test Verisi Oluşturma:** Soru işlemcisinin davranışını gözlemlemek için daha büyük bir tablo oluşturalım ve içine veri ekleyelim. Daha çok veri ekleyebilirsiniz.

```
CREATE TABLE Calisanlar (
    calisan_id INT PRIMARY KEY AUTO_INCREMENT,
    ad VARCHAR(50) NOT NULL,
    soyad VARCHAR(50) NOT NULL,
    departman VARCHAR(50),
    maas DECIMAL(10, 2),
    ise_giris_tarihi DATE
);

-- Örnek veri ekleme (Çok sayıda kayıt eklemek için döngü kullanılabilir
-- veya daha fazla INSERT ifadesi yazılabilir)
INSERT INTO Calisanlar (ad, soyad, departman, maas, ise_giris_tarihi) VALUES
```

```
('Ahmet', 'Yılmaz', 'IT', 75000.00, '2020-01-15'),
('Ayşe', 'Demir', 'HR', 60000.00, '2019-03-20'),
('Mehmet', 'Can', 'IT', 80000.00, '2021-07-01'),
('Zeynep', 'Kaya', 'Sales', 70000.00, '2020-11-10'),
('Ali', 'Veli', 'HR', 55000.00, '2022-05-01');
```

- 2. Sorgu Yürütme Planını İnceleme (EXPLAIN):** EXPLAIN komutu, MySQL'in bir sorguyu nasıl yürüteceğini gösteren bir yürütme planı sağlar. Bu, sorgu iyileştircisinin kararlarını anlamak için kritik bir araçtır.

```
EXPLAIN SELECT * FROM Calisanlar WHERE departman = 'IT' AND maas > 70000;
```

*Cıktıyı inceleyin ve type, rows, Extra gibi sütunların ne anlama geldiğini araştırın. Cıktıyı incelerken Result Grid, Form Editor, Field Types, Query Stats, Execution Plan seçeneklerini inceleyiniz. Query Stats kısmından sorgunun çalışma zamanını (Execution Time) not alınız.*

- 3. İndeks Etkisini Gözlemleme:** departman sütununa bir indeks ekleyelim ve sorgu planının nasıl değiştiğini gözlemleyelim.

```
CREATE INDEX idx_departman ON Calisanlar (departman);
EXPLAIN SELECT * FROM Calisanlar WHERE departman = 'IT' AND maas > 70000;
```

*Önceki EXPLAIN çıktısıyla karşılaştırın. İndeks eklemenin sorgu planını nasıl etkilediğini yorumlayın.*

### Sorgu İşlemcisi Soruları:

1. EXPLAIN çıktısında gördüğünüz type ve Extra sütunları ne anlama gelmektedir? Sorgu performansını değerlendirmede nasıl yardımcı olurlar?
2. Sorgu iyileştirici (Query Optimizer) neden önemlidir? Hangi faktörleri göz önünde bulundurarak bir sorgu için en iyi yürütme planını seçer?
3. İndeks eklemeden önce ve sonra EXPLAIN çıktılarındaki temel farklar nelerdir? Bu farklar performansı nasıl etkiler?
4. İndeksler veritabanı performansını nasıl artırır? İndekslemenin dezavantajları var mıdır?
5. CREATE INDEX komutu ile bir indeks oluşturduğunuzda, bu indeks fiziksel olarak nerede saklanır ve depolama yöneticisi bu indeksi nasıl kullanır?
6. MySQL'de B-Tree, Hash, Full-text gibi farklı indeks türleri bulunmaktadır. Her birinin kullanım senaryolarını araştırın.

### 5. Depolama Yöneticisi (Storage Manager)

**Tanım:** Depolama Yöneticisi, veritabanı sisteminin disk üzerindeki verileri nasıl depoladığını, eriştiğini ve yönettiğini kontrol eden bileşendir. MySQL'de bu işlevsellik depolama motorları (örneğin InnoDB, MyISAM) aracılığıyla sağlanır. İndeksler ise veritabanı tablolarındaki verilere daha hızlı erişim sağlamak için kullanılan özel veri yapılarıdır.

## Uygulama Adımları:

- InnoDB Depolama Motoru Kullanımı:** MySQL'de varsayılan depolama motoru InnoDB'dir. Bir tablo oluştururken veya mevcut bir tabloyu değiştirirken depolama motorunu belirtebiliriz. Mevcut bir tablonun depolama motorunu kontrol ediniz. Form Editor görünümü üzerinden çıktıyi inceleyiniz.

-- Varsayılan olarak InnoDB kullanılır, ancak açıkça belirtebiliriz

```
CREATE TABLE Musteriler (
    musteri_id INT PRIMARY KEY AUTO_INCREMENT,
    ad VARCHAR(50),
    soyad VARCHAR(50)
) ENGINE=InnoDB;
```

-- Mevcut bir tablonun depolama motorunu kontrol etme

```
SHOW CREATE TABLE Calisanlar;
```

## Depolama Yöneticisi Soruları:

- MySQL'de InnoDB ve MyISAM gibi farklı depolama motorları arasındaki temel farklar nelerdir? Hangi durumlarda hangi motor tercih edilmelidir?

## 6. Buffer Pool (Tampon Havuzu)

**Tanım:** Buffer Pool, InnoDB depolama motorunun sıkça erişilen tablo ve indeks verilerini önbelleğe aldığı ana bellek alanıdır. Bu sayede verilere diskten tekrar okunmak yerine doğrudan bellekten erişilerek performans artışı sağlanır.

## Uygulama Adımları:

- Buffer Pool Durumunu Görüntüleme:** SHOW ENGINE INNODB STATUS komutu, InnoDB motorunun dahili durumunu, özellikle de Buffer Pool kullanımını gösterir.

SHOW ENGINE INNODB STATUS; -- Çıktıyı inceleyin ve BUFFER POOL AND MEMORY bölümünü bulun.

Cıktıyı export seçeneği ile bilgisayarınıza kaydedin. Çıktıda Buffer pool size, Free buffers, Database pages, Buffer pool hit rate gibi değerleri not alın.

- Veri Okuma ve Buffer Pool Etkileşimi:** Calisanlar tablosundan büyük miktarda veri okuyarak Buffer Pool'un nasıl etkilendiğini gözlemleyelim.

-- Tüm Calisanlar tablosunu okuyun (Buffer Pool'a yüklenmesini sağlayın)  
SELECT \* FROM Calisanlar;

-- Tekrar Buffer Pool durumunu kontrol edin  
SHOW ENGINE INNODB STATUS;

Buffer pool hit rate ve Database pages değerlerindeki değişiklikleri yorumlayın.

**Buffer Pool Soruları:**

1. Cache bellek (önbellek) nedir ve veritabanı yönetim sistemlerinde neden kullanılır?
2. MySQL'in buffer pool (önbellek) mekanizmasını araştırın.
3. MySQL InnoDB depolama motorundaki buffer pool nedir ve hangi verileri saklar?
4. Önbellek boyutu performansı nasıl etkiler?
5. Buffer Pool nedir ve veritabanı yönetim sistemlerinde neden bu kadar önemlidir?
6. SHOW ENGINE INNODB STATUS çıktısındaki Buffer pool hit rate değeri neyi ifade eder? Yüksek veya düşük olması ne anlama gelir?
7. innodb\_buffer\_pool\_size parametresinin veritabanı performansı üzerindeki etkisini araştırın. innodb\_buffer\_pool\_size parametresini ayarlarken nelere dikkat edilmelidir? Bu parametrenin yanlış ayarlanması ne gibi sorunlara yol açabilir?

**7. Sonuç**

Bu laboratuvar çalışmasında, Veritabanı Yönetim Sistemlerinin temel mimari bileşenlerini ve MySQL üzerindeki pratik uygulamalarını deneyimlediniz. DDL ve DML komutlarının yanı sıra, sorgu işlemcisinin, depolama yöneticisinin, indekslerin ve Buffer Pool'un veritabanı performansındaki kritik rolünü gözlemlediniz. Bu kavramları anlamak, daha verimli ve performanslı veritabanı uygulamaları geliştirmek için temel oluşturacaktır.

**Çalışmalarınızı PDF formatında ilgili bölüme yükleyiniz.**