

1906003022015

Veritabanı Yönetim Sistemleri

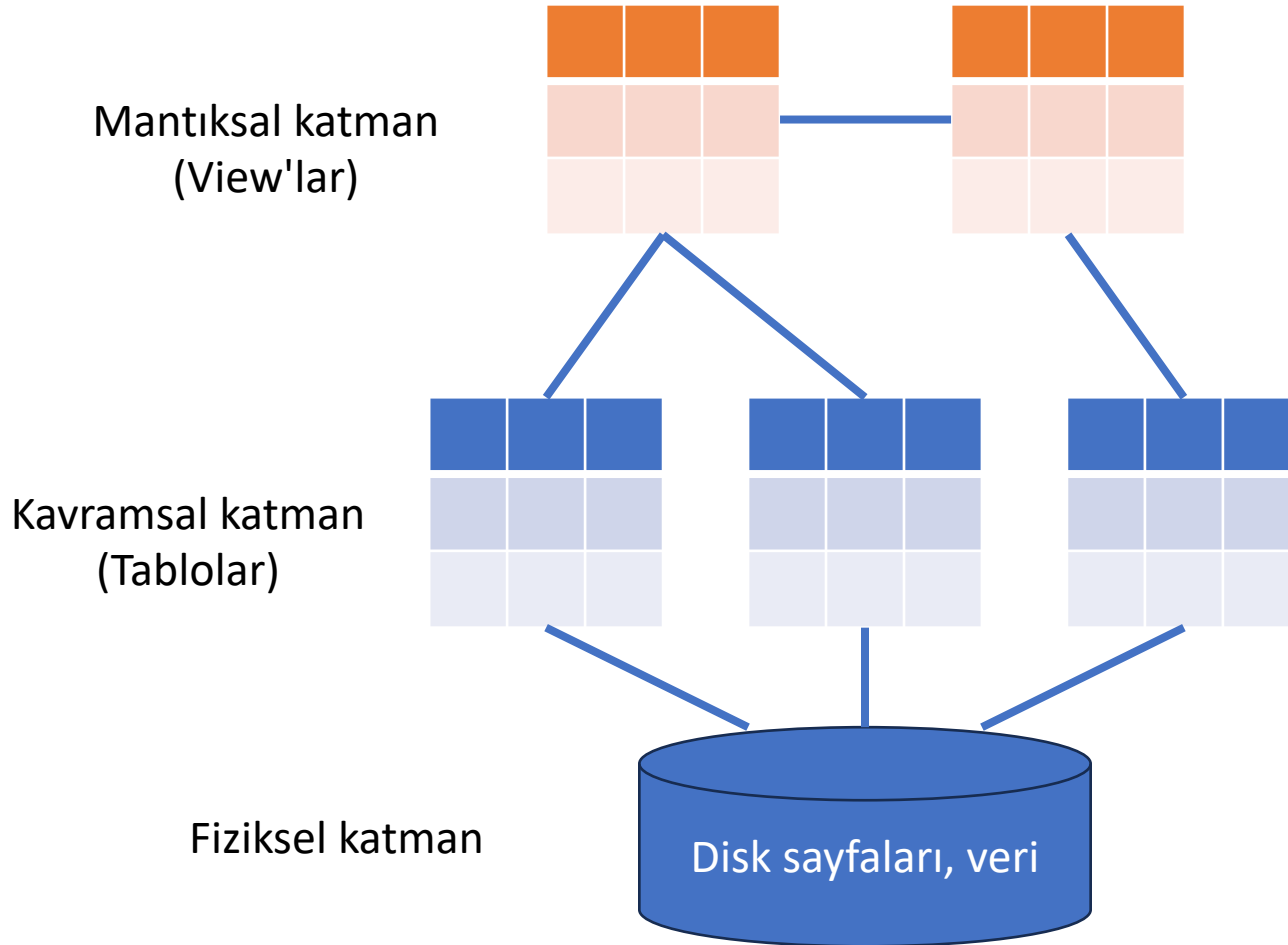
BAİBÜ Bilgisayar Müh.

Dr. Öğr. Üyesi İsmail Hakkı Parlak

ismail.parlak@ibu.edu.tr

Oda: 329

Görünümler (Views)



View'lar Niçin Kullanılır?

- Kullanıcılardan veri gizleme
- Bazı sorguları yazmayı daha kolay/doğal hale getirme
- Veritabanına erişimi modülerleştirme

View oluşturma ve Kullanma

- View $V = \text{ViewSorgusu}(R_1, R_2, \dots, R_n)$
- V'nin şeması view sorgusunun sonucudur.
- V'yi barındıran bir sorgu (Q) çalıştırılırken DBMS arka planda Q'yu R_1, R_2, \dots, R_n 'i kullanacak şekilde tekrar yazar.
- View somut (fiziksel) bir yapı değildir.
- R_i 'ler tablo ya da view olabilir.

Create View `ViewAdı (A1, A2, ..., An)` As
`SQL Sorgusu`

View Uygulaması

SQLite

View'larda Değişiklik

- View'larda seçim (select) işlemleri aynı bir tablo üzerinde seçim yapıyormuş gibi çalışabilir.
- View'larda değişiklik yapmak istediğimizde (insert, update, delete) tablolardaki gibi değişiklik yapabilir miyiz?
- View'lar sadece mantıksal görüntüler olduğu için bu işlemler mantıklı görünmeyebilir.
- Ancak view'lar bazı kullanıcıların DB'ye açılan tek penceresi olduğu için değişiklik işlemleri zorunlu hale gelebilir.

View'larda Değişiklik

1. Otomatik view değişiklikleri

- Hatasız gerçekleştirilebilir ancak çok kısıtlıdır.
- View V oluşturulurken **select** işlemi (**distinct** kullanılmadan) tek bir tablo T üzerinde çağrılmalıdır.
- T 'de bulunup da V 'de bulunmayan niteliklere değer olarak **NULL** girilebilmeli veya **default** değerleri olmalıdır.
- V oluşturulurken **group by** kullanılamaz.
- Tüm DBMS'ler otomatik view değişikliğini sağlamamaktadır.

2. Trigger'lar ile view değişiklikleri

- Mantıklı tüm değişiklikler gerçekleştirilebilir ancak hataya açıktır.
- **INSTEAD OF** trigger'ı kullanılır.

Uygulama

SQLite