

BİL 212 Veri Yapıları

ÖDEV 1

Bu ödevde basitleştirilmiş bir RAM'de çalışan (in-memory) veritabanı (etuDB) oluşturacağız. Geliştirmeniz beklenen sınıf(lar)'ın arayüz bilgisi aşağıda tanımlanmıştır.

– etuDB sınıfının ***void createTable(String createTableCommand)*** şeklinde bir metodu bulunmalı. Bu metod sayesinde veritabanı nesnesi diskten dosya okuyarak, bir tablo oluşturur.

1) Dosya okuma komutları

CREATE TABLE FROM PathToFile

– etuDB sınıfının ***String[] query(String query)*** şeklinde bir metodu bulunmalı. Bu metod sayesinde okunan veritabanı tabloları üzerinde sorgular çalıştırılacak ve üretilen sonuçlar geri döndürülecek.

2) Veri Sorgulama

SELECT singleColumnName FROM tableName; //Tablodan sadece tek bir kolonu listeler

SELECT commaSeparatedColumnList FROM tableName; //istenilen kolonları listeler

*SELECT * FROM tableName; //Tablodaki tüm kolonları listeler*

3) Filtre Uygulayarak Veri sorgulama

SELECT columnOrColumnFamily FROM tableName WHERE columnName=SomeValue;

//ismi verilen tabloda WHERE ifadesinin sağındaki şartı sağlayan satır(lar)ı bul ve verilen kolon yada kolonlarının değerlerini döndür.

– etuDB sınıfının *string printSchema(String tableName)* isminde bir metodu olmalı ve ismi verilen tablonun kolon isimlerinin listesini ekrana basmalı.

Parser(Komut ve Sorgu Çözücü metodlar hakkında)

Bunun için bir query parser yazmanız gerekiyor. Yani bir CREATE yada SELECT ifadesi geldiğinde bu sorguyu tokenlerine ayırmanız gerekecek. Gelen sorguların yukarıda verilen formata tam uyacağını varsayabilirsiniz. Her bir sorguyu “ ” boşluk karakterine göre parçalara ayırıp, yukarıda BÜYÜK HARFLERLE verilen sorgulara özel anahtar sözcükleri ve columnName gibi tokenleri anlayıp çözümlemeli.

İstisnai Durumlar

– Herhangi bir durumda sorgu çözümlemede problem yaşanırsa bir exception atıp, gelen sorgu göz ardı edilip program çalışmaya devam etmeli.

– CREATE ifadesinde verilen yol (path)'de dosya bulunamıyorsa ekrana hata basılıp, program çalışmaya devam etmeli.

– CREATE ifadesinde verilen tablo ismi zaten yüklü ise eski tabloyu düşürmeli ve yeni tabloyu yüklemeli

- Okunan dosya formatı burada verilen kurallara uymuyorsa, ekrana bir exception mesajı basılarak CREATE ifadesi göz ardı edilir.
- SELECT ifadelerinde bahsedilen tablo henüz oluşturulmamış olabilir. Bu durumda tablo bulunamadı hatası basılıp, sorgu göz ardı edilmeli
- SELECT ifadelerinde bahsedilen kolon(lar) o tabloda bulunmayabilir. Bu durumda da ekrana bir exception mesajı basılıp, sorgu göz ardı edilmeli.

Kabuller

- Okuyacağınız dosyalar hep csv dosyası olacak, yani kolonlar birbiriyle virgül ile ayrılmış. Ayrıca okuduğunuz bütün csv dosyalarındaki ilk satırın header olduğunu varsayabilirsiniz. Örnek bir dosya ekte verilmiştir.
- Create metodu dosyayı diskten okuduktan sonra ilk satırda (header) kısmını okumalı. Yani csv dosyasının ilk satırında virgülle ayrılmış şekilde kolonların isimlerini bulacak.
- Kolon ve tablo isimleri tek kelime olmalı yada birden fazla ise camelCase şeklinde birleşik yazılmalı.
- Sorguda birden fazla kolon ismi listeleniyorsa, bunlar sadece virgül ile ayrılmalı aralarında boşluk olmamalı.
- Hem okuyacağınız dosyada hem de veritabanında bütün kolonların türü String olmalı.
- WHERE ifadelerinde eşitlik dışında bir operatör kullanılmayacak.

Gerçekleme (implementation)

etuDB sınıfı hakkında birçok detay yukarıda verildi. Bu sınıf herhangi bir CREATE ifadesi ile bir tablo yüklediğinde (her bir satır bir linkedlist halkası olacak şekilde) yüklemeli.

Bunun için row isminde ve table isminde iki sınıf oluşturacak şekilde organize etmeniz tavsiyemizdir. Bu sayede etuDB table'lara sahip; table'lar da row'lardan oluşan bir linkedlist'e sahip olacak bir kompozisyon kurulabilir.

Linkedlist ve tree veri yapılarının sadece kullanacağınız kısımlarını gerçekleyebilirsiniz. Diğer kısımları gerekli değil.

Index Oluşturma

Ödevin ikinci kısmında etuDB sınıfına ait her tablonuzun ilk kolonu için index oluşturmanız bekleniyor. Index olarak binary search tree veri yapısı kullanılmalı. etuDB sınıfına ait her bir tablo nesnesinin ağaç yapısında bir üye index'i olmalı. Bu ağacın düğümlerindeki BST

Bu sayede WHERE ifadesi bulunan bir sorguda eğer kullanılan kolon ilk kolon ise, linkedlist'in tamamını taramadan ağaç yapısı sayesinde sadece alakalı satır(lar)'a erişilebilir.



Örnek CSV Dosyası

```
public class VeriTabaniSurucu {

    public static void main(String[] args) {
```

```

        etuDB veriTabanim = new etuDB();
        veriTabanim.createTable("CREATE TABLE FROM /home/userLogs.csv");
        veriTabanim.printSchema("userLogs");
        veriTabanim.query("SELECT first_name FROM userLogs;" );
        veriTabanim.query("SELECT id,first_name,last_name,email FROM
userLogs;" );
        veriTabanim.query("SELECT * FROM userLogs;" );
        veriTabanim.query("SELECT email FROM userLogs WHERE id=10321;"
);
        veriTabanim.query("SELECT email FROM userLogs WHERE
first_name=Elfie;" );

//        veriTabanim.createTable("sample.txt");
//        veriTabanim.printSchema("sample");
//        veriTabanim.query("SELECT thirdColumn FROM sample;" );
//        veriTabanim.query("SELECT Columns FROM userLogs;" );
//        veriTabanim.query("SELECT * FROM sample;" );
//        veriTabanim.query("SELECT fourthColumn FROM sample WHERE
secondColumn=value;" );

    }

}

```

BONUS (10 puan)

- Bir command prompt geliştirmeniz bekleniyor.
- Program çalıştırıldığında ekrana “etuDB>>” basıp kullanıcıdan gelen sorguları beklemeli.
- CREATE yada SELECT sorguları geldiğinde gereken metodu arka planda çalıştırıp sonuçları ekrana basmalı
- Kullanıcı daha önceden yüklenmiş bir tablo ismi yazarsa, o tablonun kolonlarını listelemeli