



**DENEY FÖY - 3**

**DERS: WEB LABORATUVAR**

**KONU: NODE.JS**

**ADI: AYŞEGÜL**

**SOYADI: AKMAN**

**NUMARA: 21060684**

**Madde 1:** Node.js ile Mysql bağlantısını gerçekleştiriniz. Employee isimli bir tablo oluşturunuz (Tabloyu Mysql üzerinde oluşturmayınız. Raporunuzda tablonun node.js komutlarıyla oluşturulması anlatılmalıdır).

- a) Node.js Mysql insert komutları ile sizlere verilen verileri oluşturduğunuz Employee tablosuna ekleyiniz.

```
js sqljs > connection.connect() callback > connection.query() callback
1  const mysql = require('mysql');
2
3  // Bağlantı kur
4  const connection = mysql.createConnection({
5    host: 'localhost',
6    user: 'root',
7    password: '1343',
8    database: 'madde1'
9  });
10
11 connection.connect(function(err) {
12   if (err) throw err;
13   console.log('MySQLe bağlandı. ');
14
15   // Employee tablosunu oluştur
16   const createTableQuery = `
17     CREATE TABLE IF NOT EXISTS Employee (
18       EmployeeId INT PRIMARY KEY,
19       FirstName VARCHAR(50),
20       LastName VARCHAR(50),
21       DepartmentName VARCHAR(50)
22     )
23   `;
24
25   // Employee tablosunu oluştur
26   const createTableQuery = `
27     CREATE TABLE IF NOT EXISTS Employee (
28       EmployeeId INT PRIMARY KEY,
29       FirstName VARCHAR(50),
30       LastName VARCHAR(50),
31       DepartmentName VARCHAR(50)
32     )
33   `;
34
35   connection.query(createTableQuery, function(err, result) {
36     if (err) throw err;
37     console.log("Tablo oluşturuldu.");
38
39     // Veri ekleme
40     const insertDataQuery = `
41       INSERT INTO Employee (EmployeeId, FirstName, LastName, DepartmentName) VALUES
42       (1, 'Ken', 'Sanchez', 'Executive'),
43       (2, 'Terri', 'Duffy', 'Engineering'),
44       (3, 'Roberto', 'Tamburello', 'Engineering'),
45       (4, 'Rob', 'Walters', 'Engineering'),
46       (5, 'Gail', 'Erickson', 'Engineering'),
47       (6, 'Josef', 'Goldberg', 'Engineering'),
48       (7, 'Dylan', 'Miller', 'Support'),
49       (8, 'Diane', 'Margheim', 'Support'),
50       (9, 'Gigi', 'Matthew', 'Support'),
51       (10, 'Mich', 'Raheem', 'Support')
52     `;
53   });
54 }
```

```
connection.query(insertDataQuery, function(err, result) {
  if (err) throw err;
  console.log("Veriler eklendi.");

  const selectAllQuery = 'SELECT * FROM Employee';
  connection.query(selectAllQuery, function(err, results) {
    if (err) throw err;
    console.log("Tablo:");
    results.forEach((employee) => {
      console.log(`EmployeeId: ${employee.EmployeeId}, FirstName: ${employee.FirstName},
        LastName: ${employee.LastName}, Department: ${employee.DepartmentName}`);
    });
  });
});
```

```
PS C:\Users\Aysegul\MySQL ve Node JS> node sql.js
Debugger attached.
MySQLe bağlandı.
Tablo oluşturuldu.
Veriler eklendi.
Tablo:
EmployeeId: 1, FirstName: Ken,
      LastName: Sanchez, Department: Executive
EmployeeId: 2, FirstName: Terri,
      LastName: Duffy, Department: Engineering
EmployeeId: 3, FirstName: Roberto,
      LastName: Tamburello, Department: Engineering
EmployeeId: 4, FirstName: Rob,
      LastName: Walters, Department: Engineering
EmployeeId: 5, FirstName: Gall,
      LastName: Erickson, Department: Engineering
EmployeeId: 6, FirstName: Josef,
      LastName: Goldberg, Department: Engineering
EmployeeId: 7, FirstName: Dylan,
      LastName: Miller, Department: Support
EmployeeId: 8, FirstName: Diane,
      LastName: Margheim, Department: Support
EmployeeId: 9, FirstName: Gigi,
      LastName: Matthew, Department: Support
EmployeeId: 10, FirstName: Mich,
      LastName: Raheem, Department: Support
```

Burada öncelikle bir mysql bağlantısı sağladık.

Daha sonra bir tablo oluşturarak içine eklenecek verileri belirttik.

Ben tablonun doğru oluştuğunu anlamak adına tabloyu ekrana yazdırdım.

- b) DepartmentName'i Engineering olan çalışanları listeleyen node.js komutlarını yazınız. Sonuçların ekran çıktılarını raporunuza ekleyiniz.

```
// Engineering departmanındaki çalışanları listele
const selectEngineeringQuery = `SELECT * FROM Employee WHERE DepartmentName = 'Engineering'`;
connection.query(selectEngineeringQuery, function(err, results) {
  if (err) throw err;
  console.log("Engineering departmanında çalışanlar:", results);
});
```

```
Engineering departmanında çalışanlar: [
  RowDataPacket {
    EmployeeId: 2,
    FirstName: 'Terri',
    LastName: 'Duffy',
    DepartmentName: 'Engineering'
  },
  RowDataPacket {
    EmployeeId: 3,
    FirstName: 'Roberto',
    LastName: 'Tamburello',
    DepartmentName: 'Engineering'
  },
  RowDataPacket {
    EmployeeId: 4,
    FirstName: 'Rob',
    LastName: 'Walters',
    DepartmentName: 'Engineering'
  },
  RowDataPacket {
    EmployeeId: 5,
    FirstName: 'Gail',
    LastName: 'Erickson',
    DepartmentName: 'Engineering'
  },
  RowDataPacket {
    EmployeeId: 6,
    FirstName: 'Josef',
    LastName: 'Goldberg',
    DepartmentName: 'Engineering'
  }
]
```

Burada da madde gereğini yerine getirdim ve tablodan sadece engineer olanları çekmeye çalıştım. Bunları da ekrana yazdırdım ve kontrol ettim.

c) İsmi Terri olan çalışanın DepartmentName'ini Executive olarak güncelleyecek kodu yazınız.

```
// Terri'nin departmanını güncelle
const updateDepartmentQuery = `UPDATE Employee SET DepartmentName = 'Executive' WHERE FirstName = 'Terri'`;
connection.query(updateDepartmentQuery, function(err, result) {
  if (err) throw err;
  console.log("Terri'nin departmanı güncellendi.");

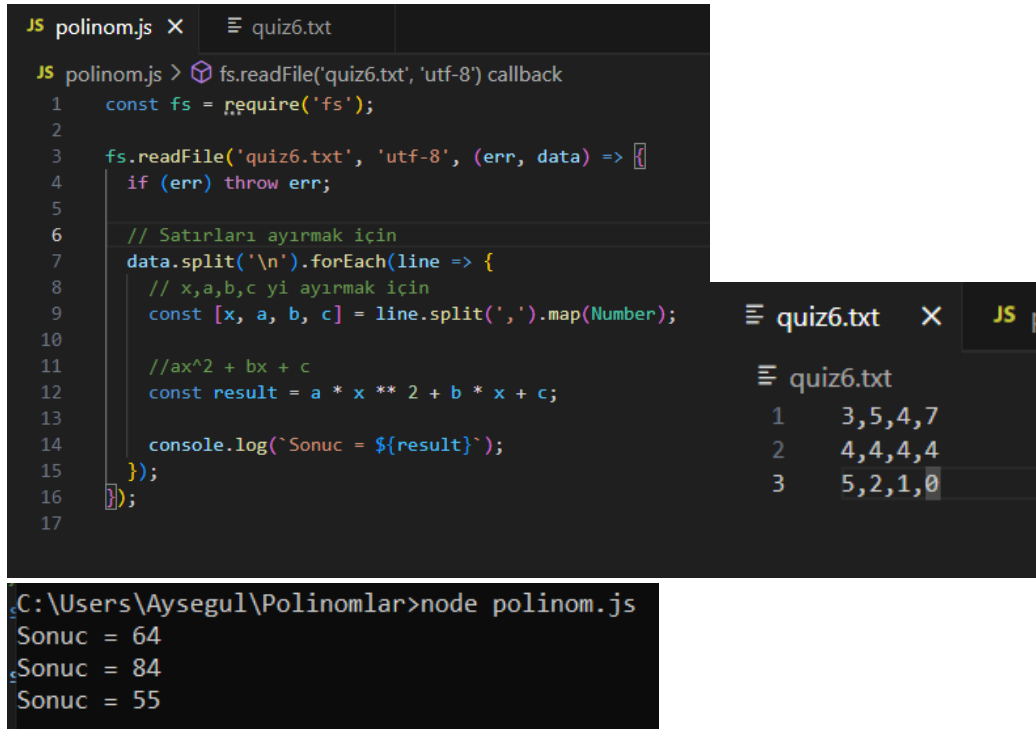
  // Bağlantıyı kapat
  connection.end(function(err) {
    if (err) throw err;
    console.log("MySQL kapatıldı.");
  });
});
```

```
Terri'nin departmanı güncellendi.
MySQL kapatıldı.
```

Burada ise Terri adlı kişinin bölümünü güncellemeye çalıştım ve sonrasında mysql bağlantısını kapattım.

Lakin tabloyu birçok kez çalıştırmak için tablonun her seferinde silinmesi gerekiyor.

**Madde 2:**  $A = ax^2 + bx + c$  polinomunun katsayılarını ve  $x$  değerlerini içeren quiz6.txt isimli bir dosya hazırlayınız. Dosyanın içeriği resimdeki gibi olmalıdır. Sırasıyla rakamlar  $x$ ,  $a$ ,  $b$  ve  $c$  değerlerini ifade etmektedir. Dosyadan okuduğunuz katsayılara göre  $A$  polinomunun değerlerini hesaplayıp ekrana yazan node.js kodunu yazınız.



```
JS polinom.js X quiz6.txt
JS polinom.js > fs.readFile('quiz6.txt', 'utf-8') callback
1  const fs = require('fs');
2
3  fs.readFile('quiz6.txt', 'utf-8', (err, data) => {
4    if (err) throw err;
5
6    // Satırları ayırmak için
7    data.split('\n').forEach(line => {
8      // x,a,b,c yi ayırmak için
9      const [x, a, b, c] = line.split(',').map(Number);
10
11      //ax^2 + bx + c
12      const result = a * x ** 2 + b * x + c;
13
14      console.log(`Sonuc = ${result}`);
15    });
16  });
17

quiz6.txt X JS p
quiz6.txt
1  3,5,4,7
2  4,4,4,4
3  5,2,1,0
```

```
C:\Users\Aysegul\Polinomlar>node polinom.js
Sonuc = 64
Sonuc = 84
Sonuc = 55
```

Burada basit bir polinom çözümü yaptırđım. Dosyayı okuyoruz ve sonrasında çözümleri ekrana yazdırıyoruz.

Spliti satırları ve değeri ayırmada kullanıyoruz.

**Madde 3:** 90'dan küçük asal sayıları asal\_sayi.txt dosyasına kaydediniz. Bu dosyayı okuyup eksik sayıları dolduran ve tum\_sayilar.txt dosyasına yazan node.js kodunu yazınız.

```
JS asal_sayim.js x  tum_sayilar.txt  asal_sayi.txt  {} launch.json
JS asal_sayim.js > fs.readFile('asal_sayi.txt', 'utf-8') callback > fs.writeFile('tum_s
1  const fs = require('fs');
2
3  //asal_sayi.txt dosyasını okuma
4  fs.readFile('asal_sayi.txt', 'utf-8', function(err, data) {
5    if (err)
6      throw err;
7
8    // Dosyadan okunan asal sayıları diziye çevirme
9    const asalSayilar = data.split(',').map(Number);
10
11    const tumSayilar = [];
12    for (let i = 0; i < 91; i++) {
13      tumSayilar.push(i);
14    }
15
16    const extra = tumSayilar.map(sayi =>
17      asalSayilar.includes(sayi) ? sayi : `${sayi}`
18    );
19
20    // Tüm sayılar tum_sayilar.txt'ye yazılır
21    fs.writeFile('tum_sayilar.txt', extra.join(','), function(err) {
22      if (err) throw err;
23      console.log('Kaydedildi.');
```

Öncelikle oluşturmuş olduğumuz asal\_sayi.txt dosyasını okuyoruz ve boş olan tum\_sayilar.txt dosyasına boş bir dizi ekliyoruz . for döngüsü ile tüm sayıları bir yığına itiyoruz. Eksik sayıları da 16. satırdaki kod ile bulmaya çalışıyoruz. En son dosyaya tüm sayıları yazdırıyoruz.

**Madde 4:** 1 ile 100 arasındaki asal sayıları bulan ve bulunan sayıları asal.txt dosyasına yazan node.js kodunu yazınız.

```
JS asal_sayilar.js •  ≡  asal.txt
JS asal_sayilar.js > ...
1  const fs = require('fs');
2
3  let asallar = [];
4
5  // 0 ile 100 arasındaki asal sayıları bulma
6  for (let i = 2; i <= 100; i++) {
7      let j;
8      for (j = 2; j < i; j++) {
9          if (i % j === 0) {
10             break;
11         }
12     }
13     if (j === i) {
14         asallar.push(i);
15     }
16 }
17
18 //asal.txt dosyasına asal sayıları yazma
19 fs.writeFile('asal.txt', asallar.join(','), function(err) {
20     if (err) throw err;
21     console.log('Kaydedildi.');
```

Öncelikle boş bir dizi oluşturuyoruz ve iç içe for döngüsü ile asal sayıları bulmaya çalışıyoruz. Bulduğumuz asalları bir yığına atıyoruz. En son asal.txt dosyasına yığındaki sayıları yazıyoruz.