Bu bölümde Apache Spark ile MySql bağlantısından bahsedeceğiz

Eğer windows/mac üzerinde mysql kurulumunuz yoksa XAMPP gibi programlarla makinenize rahatlıkla MySql kurabilirsiniz

1 - MySQL kurulumu

Mac :

```
brew install mysql
```

2 - Start mysql

```
brew services restart mysql
```

3 - Connect mysql

```
mysql -u root
```

4 - Create database

```
create database books;
```

5 - Change database

```
use books;
```

5 - Create table

```
create table authors (id INT, name varchar(20),email VARCHAR(20))
```

6 - Insert data

```
insert into authors(id,name,email) values(1,'Ali','ali@gmail.com');
insert into authors(id,name,email) values(2,'Batu','batu@gmail.com');
```

7 - Add mysql connector to build

```
libraryDependencies += "mysql" % "mysql-connector-java" % "8.0.30"
```

Spark code (Read)

```
import org.apache.spark.sql.SparkSession


object SparkSQLMysql {
 def main(args: Array[String]): Unit = {

   val spark = SparkSession.builder.master("local").
     appName("SparkByExample")
     .getOrCreate()

   val database = "books"
   val table = "authors"
   val user = "root"
   val password = ""
   val connString = "jdbc:mysql://localhost:3306/" + database


   val jdbcDF = (spark.read.format("jdbc")
     .option("url", connString)
     .option("dbtable", table)
     .option("user", user)
     .option("password", password)
     .option("driver", "com.mysql.cj.jdbc.Driver")
     .load())

   jdbcDF.show()
```

```
 }
}
```

Spark code (Write)

```
import org.apache.spark.sql.{Encoders, SaveMode, SparkSession}

import java.util.Properties



case class Author(id:Int, name:String, email:String)


object SparkSQLMysqlWrite {
 def main(args: Array[String]): Unit = {

   val spark = SparkSession.builder.master("local").
     appName("SparkByExample")
     .getOrCreate()
   import spark.implicits._

   val database = "books"
   val table = "authors"
   val connString = "jdbc:mysql://localhost:3306/" + database

   val author1 = new Author(3,"Ayse", "ayse@gmail.com")
   val author2 = new Author(4,"Mehmet", "mehmet@gmail.com")

   //create a sequence
   val authorSequence = Seq(author1, author2)

   //create encoder
   val authorEncoder = Encoders.bean(Author.getClass)

   //create dataset
   val personDs = spark.createDataset(authorSequence).as(authorEncoder)
   personDs.show()

   val connectionProperties = new Properties()
   connectionProperties.put("user", "root")
   connectionProperties.put("password", "")
```

```
    personDs.write.mode(SaveMode.Append).jdbc(connString,table,connectionProperties)

  }
}
```