**实验五： Spark SQL 编程实践**

**实验基本信息：**

时间： : 2024年9月30日

实验类型： □验证性 □设计性 ☑综合性

班级： 计算机科学与技术专业（中外合作）4班 班级代码:

理论老师： 邱开金 实验指导： 邱开金

**实验报告提交说明：**

本次实验需要撰写实验报告，实验报告填写时以完成实验任务为目的，先简要回答完成实验任务需要的步骤或需要执行的命令代码，再配以结果截图予以说明，图片数量不宜过多，能说明问题即可。最后配上总结，并提交到教师指定位置。

**实验目的：**

1. 了解 SparkSession 创建 DataFrame。

2. 了解从 RDD 转化为 DataFrame 的方法。

3. 熟练使用 DataFrame 提供的算子完成具体任务。

4. 熟练使用 spark.sql 执行 SQL 语句，完成具体任务。

**实验任务：**

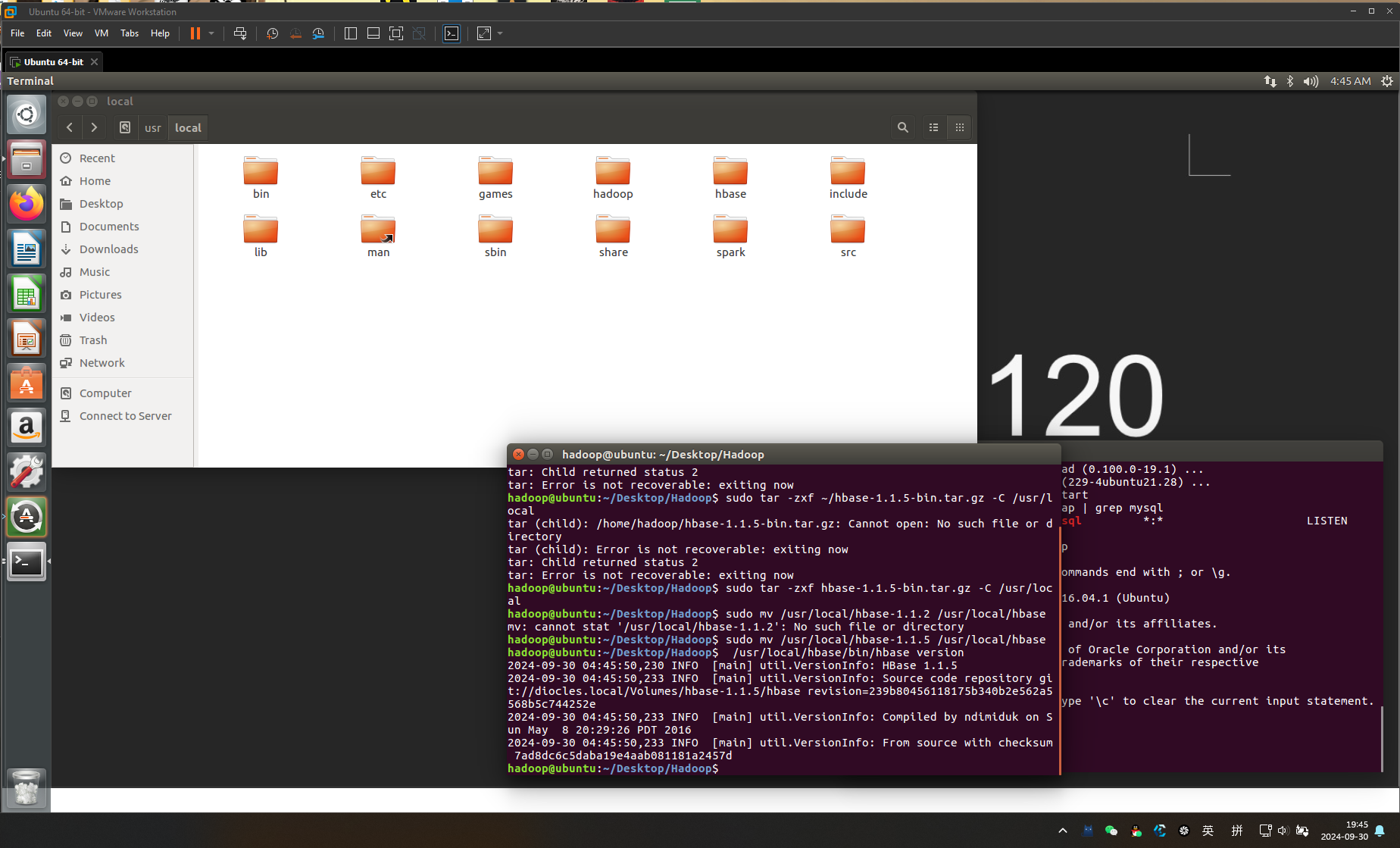
1. 安装mysql 和 HBase. 把安装的截图放上来。

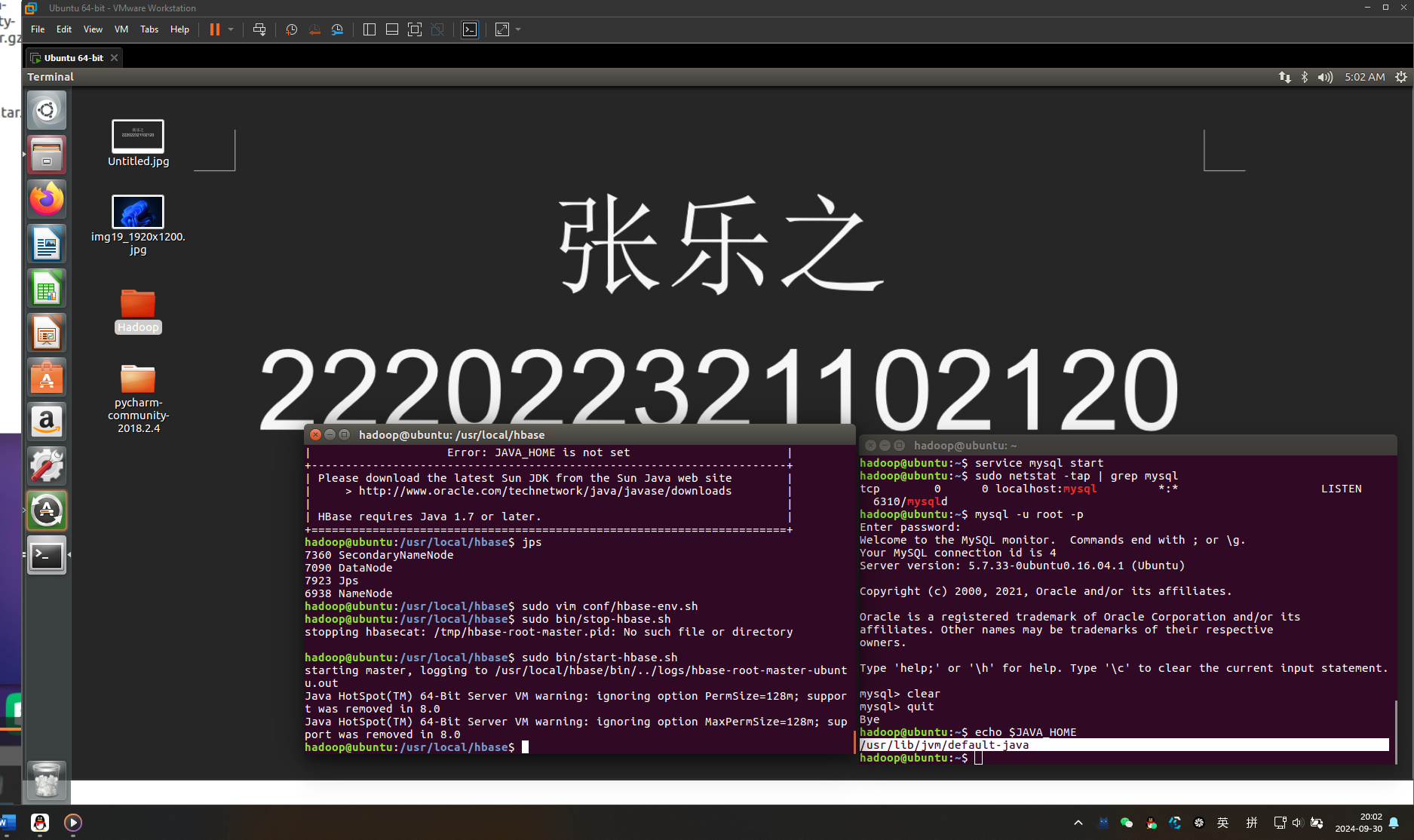
MySQL:





安装hbase:





2． 读取文件 people.json,转换为 DataFrame，完成以下实验任务

1. 文件位于： usr/local/spark/examples/src/main/resources/people.json

1）统计文件中记录总数。

2）查询年龄小于 20 岁的记录。

3）根据当前年份及年龄计算出生日期。

from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("People Analysis").getOrCreate()

df = spark.read.json("file:///usr/lodcal/spark/examples/src/main/resources/people.json")

df.createOrReplaceTempView("people")

total\_count = df.count()

print("Total record count: {}".format(total\_count))

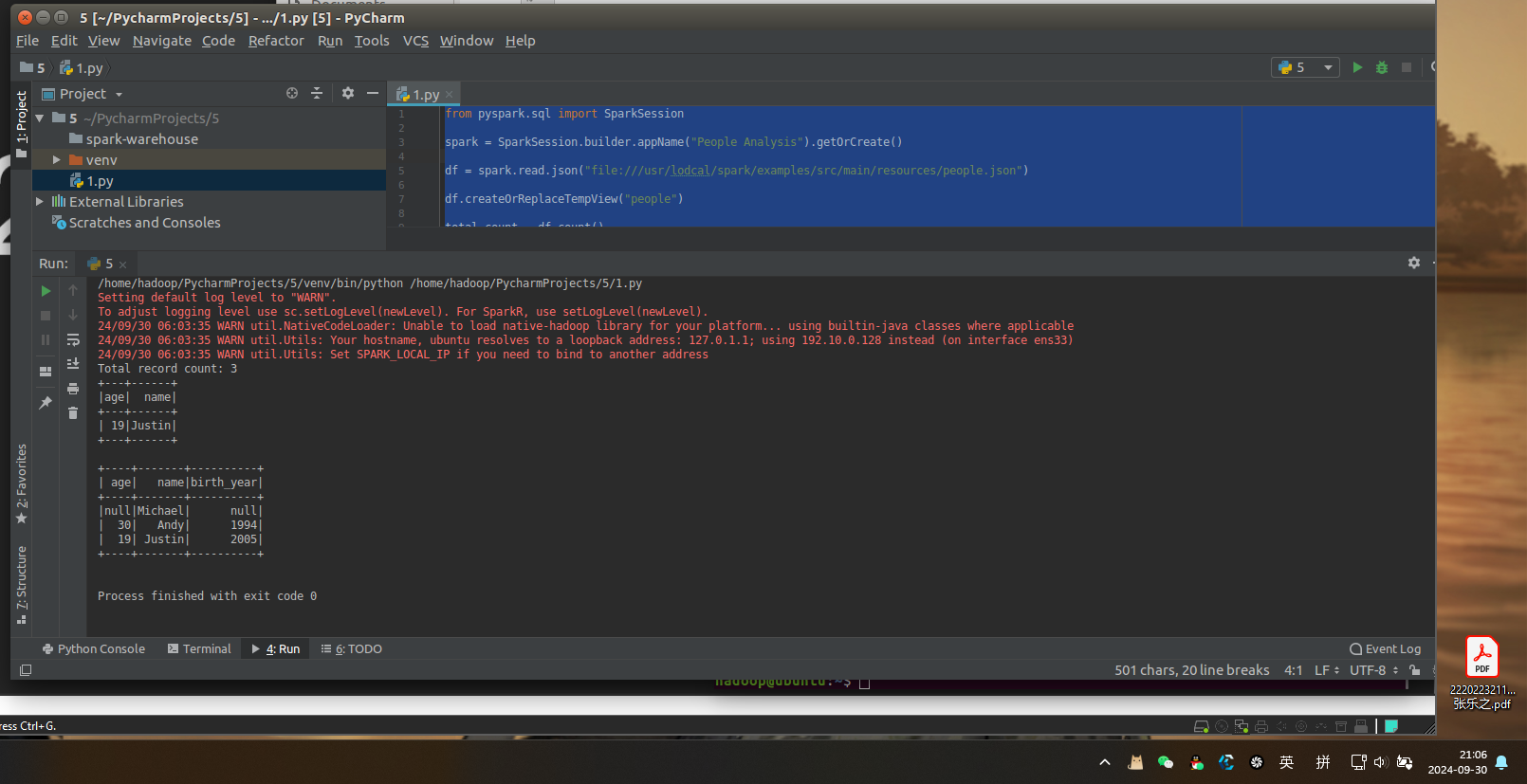
under\_20 = df.filter(df.age < 20)

under\_20.show()

from pyspark.sql.functions import expr

df\_with\_birthdate = df.withColumn("birth\_year", expr("2024 - age"))

df\_with\_birthdate.show()



4）利用 SQL 语句重新完成 1）-3)的任务。

total\_count\_sql = spark.sql("SELECT COUNT(\*) FROM people")

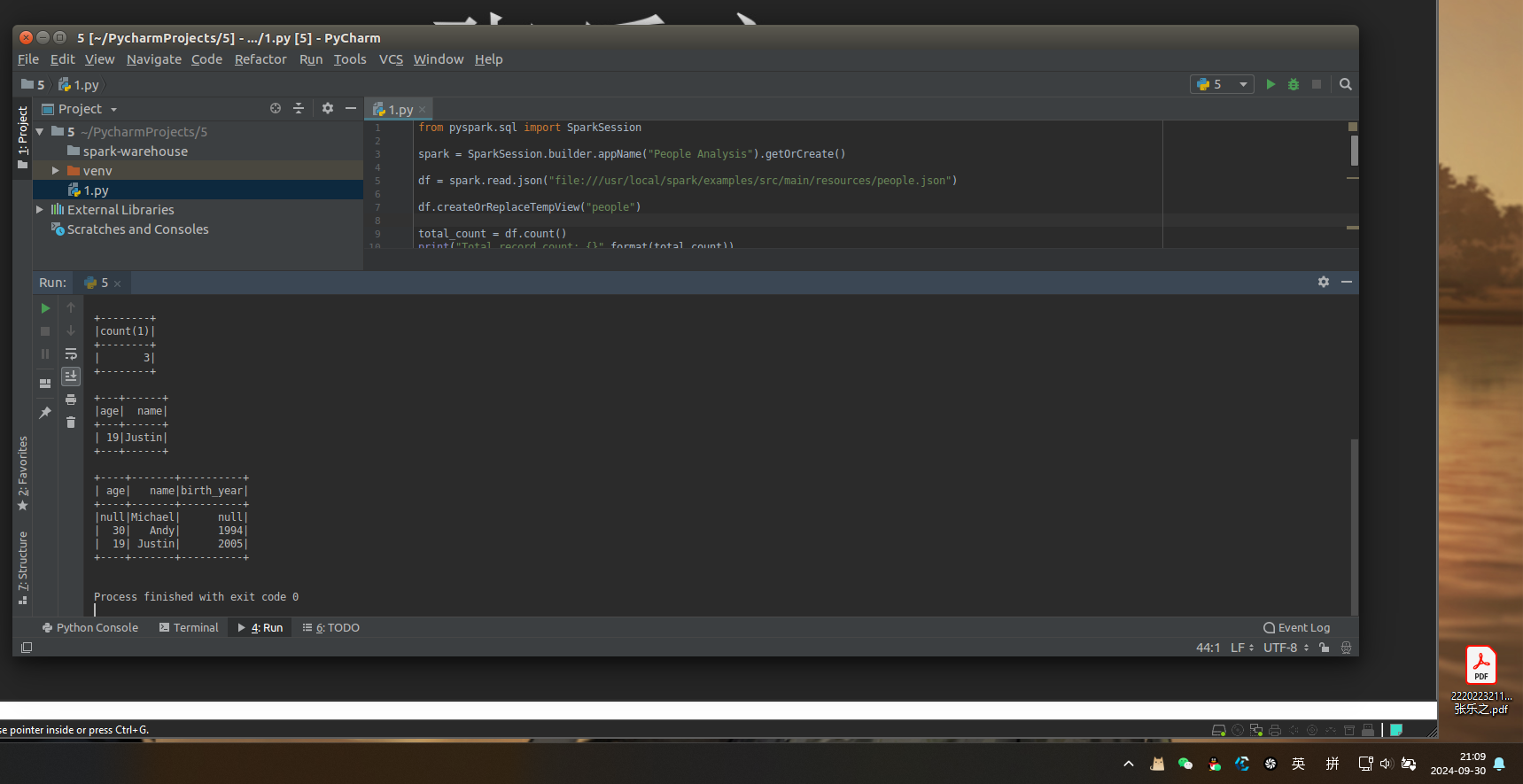
total\_count\_sql.show()

under\_20\_sql = spark.sql("SELECT \* FROM people WHERE age < 20")

under\_20\_sql.show()

birthdate\_sql = spark.sql("SELECT \*, (2024 - age) AS birth\_year FROM people")

birthdate\_sql.show()



2． 读取 iris.txt 并转换成 DataFrame,完成计算任务

1) iris.txt 与实验四内容一致。读入 iris.txt 到 RDD 中，显示文件行

数。

from pyspark.sql import SparkSession

from pyspark.sql import Row

spark = SparkSession.builder.appName("Iris Analysis").getOrCreate()

rdd = spark.sparkContext.textFile("file:///home/hadoop/Desktop/Hadoop/iris.txt")

line\_count = rdd.count()

print("Total number of lines: {}".format(line\_count))

2) 利用**任意一个种方法**将 RDD 转换为 DataFrame，其中涉及的模式为

SepalLength,SepalWidth,PetalLength,PetalWidth,TrainingClass 。

columns = ["SepalLength", "SepalWidth", "PetalLength", "PetalWidth", "TrainingClass"]

row\_rdd = rdd.map(lambda line: line.split(",")).map(lambda p: Row(SepalLength=float(p[0]),

SepalWidth=float(p[1]),

PetalLength=float(p[2]),

PetalWidth=float(p[3]),

TrainingClass=p[4]))

iris\_df = spark.createDataFrame(row\_rdd, schema=columns)

iris\_df.show()

3) 转换得到 DataFrame 后，利用xx.createOrReplaceTempView("**iris\_view**")代码创建 **iris\_view**。

iris\_df.createOrReplaceTempView("iris\_view")

4) 利用 spark.sql 执行 SQL 语句完成如下任务：

a.统计总行数;

b.统计每种花的数量;

c.统计 PetalWidth 小于 2 的记录有哪些。

d.计算 Iris-virginica 花的平均 PetalLength 值，并输出。

total\_count\_sql = spark.sql("SELECT COUNT(\*) FROM iris\_view")

total\_count\_sql.show()

flower\_count\_sql = spark.sql("SELECT TrainingClass, COUNT(\*) AS count FROM iris\_view GROUP BY TrainingClass")

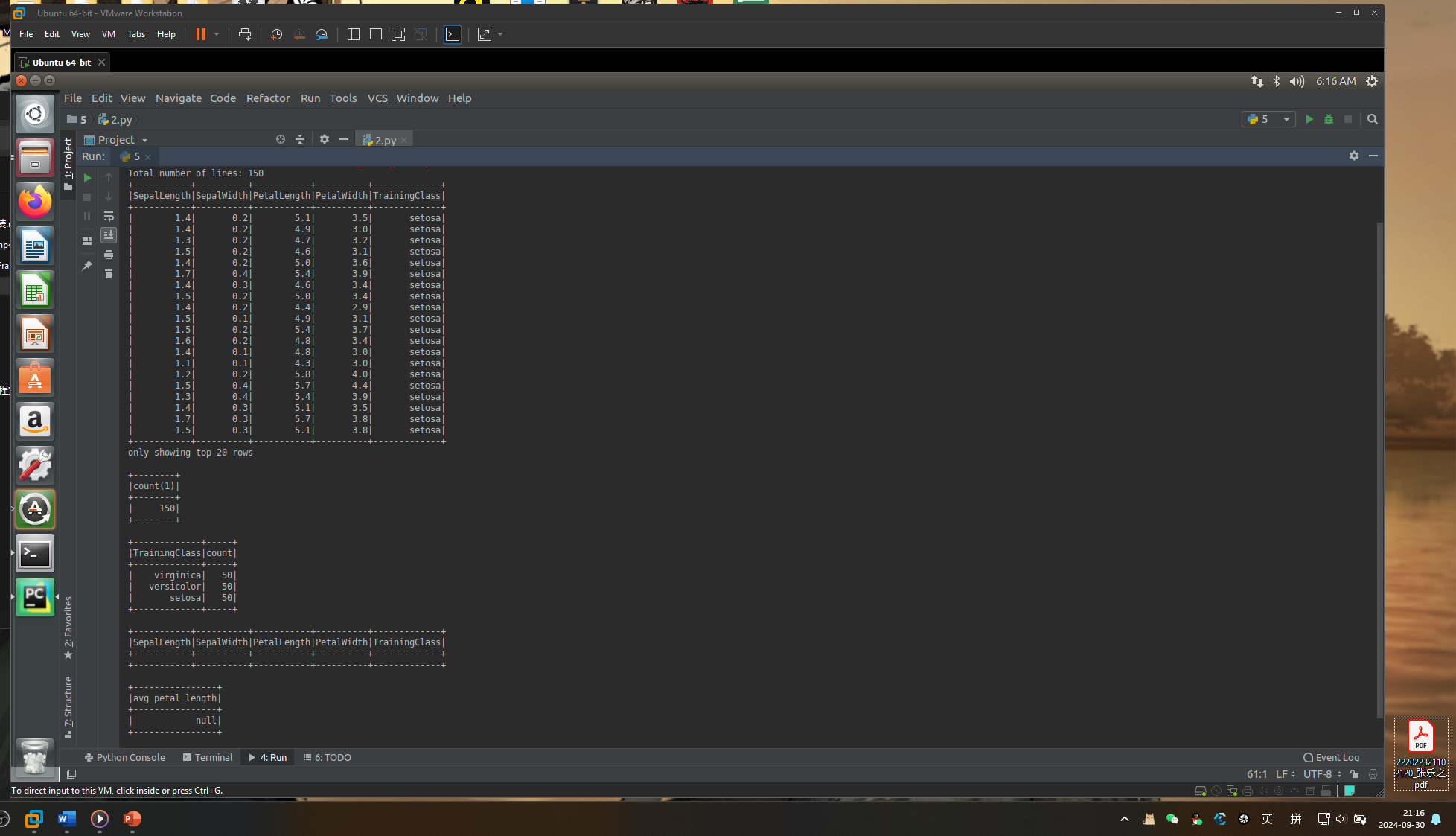
flower\_count\_sql.show()

petalwidth\_filter\_sql = spark.sql("SELECT \* FROM iris\_view WHERE PetalWidth < 2")

petalwidth\_filter\_sql.show()

avg\_petal\_length\_sql = spark.sql("SELECT AVG(PetalLength) AS avg\_petal\_length FROM iris\_view WHERE TrainingClass = 'Iris-virginica'")

avg\_petal\_length\_sql.show()



**实验总结：**

在本次实验中，我主要学习了 **SparkSQL** 技术来处理并分析数据。基于Mysql和hbase平台，并使用了RDD和DataFrame数据结构，我们编写了python程序来处理问题。首先将文本文件读取为 RDD，(Spark 中用于并行处理数据的底层结构)。然后通过 map 操作将 RDD 转换为 Row 格式，再使用 createDataFrame 将其转为 DataFrame，这样我们可以像操作数据库表一样对数据进行结构化查询。接着通过 createOrReplaceTempView 创建 SQL 视图，并利用 Spark SQL 进行总行数统计、分组统计、条件过滤等操作。Spark 的分布式计算模型和内存优化使其非常高效，适合大规模数据处理。