|  |  |  |  |
| --- | --- | --- | --- |
|  | RDD(userRDD) | DataFrame(df) | Spark SQL(sqlContext.sql) |
| 建立 | global Path  If sc.master[0:5]==”local”:  Path=…  Else:  Path=…  userRDD = sc. textFile(Path+”path u want”) | From pyspark.sql Import Row  sqlContext = SparkSession.builder.getOrCreate()  user\_Rows = userRDD.map):lambda p:Row(user=int(p[0]),age=int(p[1])))  #建立Row資料型態  User\_df= sqlContext.createDataFrame(user\_Rows)  #建立DataFrame  df=user\_df.alias(“df”)  #建立別名 | User\_df.registerTempTable(“user\_table”) |
| 查看 | .take(number) | .Schema | sqlContext.sql(”SELECT \* counts FROM user\_table”).show(number) |
| 計算筆數 | .count() |  | sqlContext.sql(”SELECT count(\*) counts FROM user\_table”).show() |
| 顯示部分欄位 | .map(lambda x:x[index]) | .select(“userid”,”occupation”,”gender”,”age”).show()  .select([dataframe名稱].[欄位名稱]).show(number)  .select([dataframe別名].[欄位名稱]).show(number)  df[別名[‘欄位名稱’]].show(number) | sqlContext.sql(”SELECT 欄位名稱, FROM user\_table”).show(number) |
|  | .map(lambda x:f(x[i]))  #直接增加欄位 | .select(“userid”,”occupation”,”gender”,”age”,(2016-df.age).alias(“欄位別稱”)).show(number)  #直接增加欄位 | sqlContext.sql(”SELECT 欄位名稱,f(欄位名稱) 別名 FROM user\_table”).show(number) |
| 篩選 | .filter(lambda r:f(r)).take(number)  #f(r)為篩選條件 | .filter(“欄位1=?”).filter(“欄位2=?”).show()  #多個filter相當於and的意思  .filter(“條件1” and “條件2”).show()  別名.filter(別名.欄位1名稱==條件1) & (別名.欄位2名稱==條件2) &(別名.欄位3==條件3名稱).show()  別名.filter(別名[欄位1]==條件1) & (別名[欄位2]==條件2) &(別名[欄位3]==條件3).show() | sqlContext.sql(”SELECT \* FROM user\_table where 條件1 and 條件2”).show(number) |
| 排序 | .takeOrdered(numbers, key=lambda x:f(x)) | .orderBy(欄位, ascending=0).show()  .orderBy(別名.欄位名稱.desc()).show() | sqlContext.sql(“”SELECT \* FROM user\_table ORDER BY 欄位名稱 DESC””).show(number) |
| 多欄位排序 | .takeOrdered(numbers, key=lambda x: (-int(x[1]),x[2])) | .orderBy([欄位1,欄位2]), ascending=[0,1]).show()  .orderBy(別名.欄位名稱1.desc(),別名.欄位名稱2).show() | sqlContext.sql(“”SELECT \* FROM user\_table ORDER BY 欄位名稱1 DESC, 欄位名稱2””).show(number) |
| 不重複資料 | .map(lambda x:f(x)).distinct().collect() | .select(欄位1,欄位2).distinct().show() | sqlContext.sql(”SELECT distinct 欄位1,欄位2 FROM user\_table”).show(number) |
| 統計資料 | A = map()產生的key-value  A.reduceByKey(func).collect()  #reduceByKey是利用func將目標key的values做merge | .select(欄位).groupby(欄位).count().show()  #依照欄位進行統計，count()進行群組統計 | SqlContext.sql(“SELECT 欄位名稱, count(\*) counts FROM user\_table GROUP BY 欄位名稱””).show() |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |