|  |
| --- |
|  |
| Spark Programming Assignment |
| 20-PBD-002 |
| Shraddha P Jain |
|  |
| **15-Nov-21** |

|  |
| --- |
|  |

**Set 1**

**ADDING DATA IN SERVER**

1. **FACULTY LIST**

val FacultyList = List(("F1","A.R.Patel","Lecturer","1970-05-12","1997-06-23","Male"),("F2","S.M. Bhaduria","Lecturer","1971-11-10","1997-08-02","Female"))

val FacultyListDF = FacultyList.toDF("FacultyListid","Facname","designation","birthdate","joiningdate","gender")

FacultyListDF.show()

1. **StationeryList**

val StationeryList = List(("S1","Ruled","C1"),("S2","Blank","C1"),("S3","Square sheet","C1"),("S4","CD","C2"),("S5","Pen Drive","C2"),("S6","Spring","C3"),("S7","Box","C3"),("S8","Spiral binding","C4"),("S9","Thick binding","C4"))

val StationeryListDF = StationeryList.toDF("Statid","Statname", "catid")

StationeryListDF.show()

1. **CategoryList**

val CategoryList = List(("C1","Paper"),("C2","Storage device"),("C3","Files"),("C4","Writing pads"))

val CategoryListDataFrame = CategoryList.toDF("catid","CategoryListname")

CategoryListDataFrame.show()

1. **Issuedto**

val Issuedto = List(("F1","S9","2001-09-12","2"),("F1","S3","2005-09-11","25"),("F1","S8","2001-09-12","3"),("F2","S9","2001-09-12","2"))

val IssuedtoDF = Issuedto.toDF("FacultyListid","Statid","Issuedate","Quantity")

IssuedtoDF.show()

**Questions and Answers**

**1.Display category names of which there is not a single stationery.**

val CatStat = CategoryListDataFrame.join(StationeryListDF, CategoryListDataFrame. (“categoryid”)== StationeryListDF (“catid”),”leftanti”)

CatStat.show

**2.Display faculty names and total no. of stationeries issued to the faculty.**

val FacultyListStat = FacultyListDF.join(IssuedtoDF, "FacultyListid")

val TotalFaculty = FacultyListStat.groupBy("Facname").agg(sum("Quantity").as("QuantitySum"))

TotalFaculty.show

**3. Display faculty name to whom maximum no. of stationeries issued.**

TotalFaculty.groupBy("Facname").agg(max("QuantitySum").as("QuantityMax")).show

val max\_val = TotalFaculty.agg(max("QuantitySum").as("Max\_Val"))

**4. Display names of female faculties.**

FacultyListStat.filter("gender = 'Female'").show

**5.Display names of the stationeries which are issued more than 5.**

IssuedtoDF.filter("Quantity>5").show

**6.Display names of the faculties to whom stationery ‘CDROM’ is issued.**

val FacultyName= FacultyListStat.join(StationeryListDF, "Statid")

FacultyName.filter("Statname = 'Thick binding'").show

**7. Display stationery names which are issued in the month of ‘January’.**

val Issued = IssuedtoDF.join(StationeryListDF, "Statid")

val Issued1 = Issued.withColumn("Issuedate", (col("Issuedate").cast("date")))

val Issued2 = Issued1.withColumn("Month", from\_unixtime(unix\_timestamp($"Issuedate", "yyyy-MM-dd"), "MMMMM"))

Issued2.filter("Month = 'January'").show

**8. Display names of faculties born between year 1970 and 1976.**

val FaDataFrame = FacultyListDF.withColumn("birthdate", (col("birthdate").cast("date")))

val FDF = FaDataFrame.withColumn("Year", from\_unixtime(unix\_timestamp($"birthdate", "yyyy-MM-dd"), "yyyy"))

val FDF = FaDataFrame.withColumn("Year", from\_unixtime(unix\_timestamp($"birthdate", "yyyy-MM-dd"), "yyyy").cast("int"))

FDF.filter("Year>=1970 and Year<=1976").show

**9. Display names of the stationeries starting with ‘P’.**

StationeryListDF.filter("Statname like 'P%'")

**10. Display pair of faculties whose designation is same.**

FacultyListDF.sort($"designation").show()

**11. Display total no. of male faculties.**

valAllFemaleFac = FacultyListDF.filter("gender = 'Female'")

AllFemaleFac.count()

**12.Display retirement date of each employee.(assuming 60 is the retirement age)**

val ListOfFacRetire = FacultyListDF.withColumn("Age", round(months\_between(current\_date(), col("birthdate")).divide(12)))

ListOfFacRetire.filter("Age>60").show

**13.Display faculty name, stationery name, category name and issue date.**

val Cat1 = CategoryListDataFrame.join(StationeryListDF,"catid")

val isCat1 = Cat1.join(IssuedtoDF,"Statid")

val Fac = isCat1.join(FacultyListDF,"FacultyListid")

Fac.show()

Fac.select("facname","Statname","CategoryListname","Issuedate").show()