HIVE QUERIES

create database indiaDB;

use indiaDB;

create table suicides(state string,year int,type\_code string,type string,gender string,age\_group string,case\_count int) row format delimited fields terminated by ',';

load data local inpath '/root/suicides.csv' into table suicides;

SPARK QUERIES

hc = spark.builder.enableHiveSupport().getOrCreate()

hc.sql("use indiaDB")

data = hc.sql("select \* from suicides")

data.show(20,False)

data.createOrReplaceTempView("suicides")

from pyspark.sql.functions import sum,col,min,max

spark.sql("select \* from suicides").show(30,False)

1. **Print the list of states in the dataset**

spark.sql("select distinct(state) as STATES from suicides order by STATES").show(40,False)

1. **Print the total cases according to the year**

spark.sql("select type\_code, year, sum(case\_count) as Total from suicides where group by type\_code,year order by year").filter(col("type\_code")=="Means\_adopted").drop("type\_code").show(40,False)

1. **Print the percentage of increase between 2001 and 2012**

query1 = spark.sql("select type\_code, year, sum(case\_count) as Total from suicides where group by type\_code,year order by year").filter(col("type\_code")=="Means\_adopted").drop("type\_code")

maxVal = query1.select(max("Total")).collect()[0][0]

minVal = query1.select(min("Total")).collect()[0][0]

print("Average increase: " + str((float(maxVal)-float(minVal))/float(minVal)\*100))

1. **Printing the Cases every year gender wise with %share of both gender**

maleData = data.filter((data.type\_code == "Means\_adopted") & (data.gender == "Male")).groupBy("year","gender").agg(sum("case\_count").alias("Male")).orderBy("year").drop("gender")

femaleData = data.filter((data.type\_code == "Means\_adopted") & (data.gender == "Female")).groupBy("year","gender").agg(sum("case\_count").alias("Female")).orderBy("year").drop("gender")

genderData = maleData.join(femaleData,"year")

genderData.withColumn("Male %", col("Male")/(col("Male")+col("Female"))\*100).withColumn("Female %" ,col("Female")/(col("Male")+col("Female"))\*100).show()

1. **List of Causes for suicide in descending order of total of cases**

data.filter(data.type\_code == "Causes").groupBy("type").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(50,False)

1. **Breaking down the causes for suicide based on age groups in descending order of total cases**

data.filter(data.type\_code == "Causes").groupBy("type","age\_group").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(200,False)

1. **Printing the total cases categorised under cause of suicide gender wise**

maleData = data.filter((data.type\_code == "Causes") & (data.gender == "Male")).groupBy("type").agg(sum("case\_count").alias("Male")).orderBy(col("Male").desc())

femaleData = data.filter((data.type\_code == "Causes") & (data.gender == "Female")).groupBy("type").agg(sum("case\_count").alias("Female")).orderBy(col("Female").desc())

genderData = maleData.join(femaleData,"type")

genderData.show(50,False)

1. **Printing the total cases categorised under cause of suicide of specific age group**

data.filter((data.type\_code == "Causes") & (data.age\_group == "15-29")).groupBy("type").agg(sum("case\_count").alias("Age 15-29")).orderBy(col("Age 15-29").desc()).show(20,False)

1. **Printing the total cases categorised under professional profile of the people who committed suicide**

data.filter(data.type\_code == "Professional\_Profile").groupBy("type").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(50,False)

1. **Printing the total cases categorised under method adopted for suicide**

data.filter(data.type\_code == "Means\_adopted").groupBy("type").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(50,False)

1. **Printing the total cases categorised under state**

data.filter(data.type\_code == "Means\_adopted").groupBy("state").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(50,False)

1. **Breaking down cases on the causes of suicide for a specific state**

data.filter((data.type\_code == "Causes") & (data.state == "Maharashtra")).groupBy("state","type").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).drop("state").show(50,False)

1. **Breaking down the cases on the education status of the people committed suicide**

data.filter(data.type\_code == "Education\_Status").groupBy("type").agg(sum("case\_count").alias("Total\_Cases")).orderBy(col("Total\_Cases").desc()).show(50,False)

1. **Partitioning the data based on state and type\_code of suicide**

data.write.option("header",True).partitionBy("state","type\_code").mode("overwrite").csv("/suicideData")

1. **Reading data from partition and comparing two states based on cause for suicide**

MHData = spark.read.options(header='True', inferSchema='True').csv("/suicideData/state=Maharashtra/type\_code=Causes/").withColumnRenamed("case\_count", "MH Cases")

WBData = spark.read.options(header='True', inferSchema='True').csv("/suicideData/state=West Bengal/type\_code=Causes/").withColumnRenamed("case\_count", "WB Cases")

outData = MHData.join(WBData,['year','type','gender','age\_group'])

outData.groupBy('type').agg(sum("MH Cases").alias("Cases in Maharashtra"), sum("WB Cases").alias("Cases in West Bengal")).orderBy(col("Cases in Maharashtra").desc(),col("Cases in West Bengal").desc()).withColumnRenamed("type","Causes for Suicide").show(50,False)

1. **Reading data from partition and comparing two states based on method adopted for suicide**

WBData = spark.read.options(header='True', inferSchema='True').csv("/suicideData/state=West Bengal/type\_code=Means\_adopted/").withColumnRenamed("case\_count", "WB Cases")

MHData = spark.read.options(header='True', inferSchema='True').csv("/suicideData/state=Maharashtra/type\_code=Means\_adopted/").withColumnRenamed("case\_count", "MH Cases")

outData = MHData.join(WBData,['year','type','gender','age\_group'])

outData.groupBy('type').agg(sum("MH Cases").alias("Cases in Maharashtra"), sum("WB Cases").alias("Cases in West Bengal")).orderBy(col("Cases in Maharashtra").desc(),col("Cases in West Bengal").desc()).withColumnRenamed("type","Means Adopted for Suicide").show(50,False)

1. **Printing the trend of a particular cause between the timeframe along with percentage contribution**

totalValue = data.filter(data.type == "Drug Abuse/Addiction").groupBy("year").agg(sum("case\_count").alias("TC")).agg(sum("TC")).collect()[0][0]

data.filter(data.type == "Drug Abuse/Addiction").groupBy("year").agg(sum("case\_count").alias("TC")).orderBy("year").withColumn("% Share of Total (" +str(totalValue)+ ")",(col("TC")/totalValue)\*100).withColumnRenamed("TC", "Drug Abuse/Addiction Cases").show(50,False)