**BIGDATA LAB**

**Priyanka Dattatreya Hegde-C3**

**1NT20IS407**

**Creating database in mongoDB**

use priya407

switched to db priya407

**CRUD OPERATIONS:**

**insert one in mongoDb:**

db.priyanka.insertOne({name:"archi",usn:"1NT20IS400", sec:"c",gender:"f",sub:"ise",age:"30

db.priyanka.find()

**INSERT MANY INTO MONGODB:**

> db.priyanka.insertMany([ { Name:"priya h",usn:"1NT20IS467", sec:"c",gender:"f",sub:"ise",age:"21" },{Text

Description automatically generated Name:"rohit",usn:"1NT20IS499", sec:"c",gender:"m",sub:"cse",age:"25"},{ Name:"riya",usn:"1NT20IS599", sec:"b",gender:"f",sub:"eee",age:"23"},{ Name:"priyuu",usn:"1NT20IS599", sec:"a",gender:"f",sub:"eee",age:"24"}])

Text

Description automatically generated

**UPDATEONE IN MONGODB:**

db.priyanka.updateOne({Name:"archi"},{$set:{Name:"ranjita"}}}){ \

**OUTPUT:**

**Text

Description automatically generated**

**UPDATEMANY IN MONGODB:**

db.priyanka.updateMany({age:"25"},{$set:{age:"26"}},{name:"riya"},{$set:{name:"rita"}})

Text

Description automatically generated

**DELETE MANY IN MONGODB:**

db.priyanka.deleteMany({name:'ranjita'})

A screenshot of a computer

Description automatically generated with medium confidence

**SET FUNCTION IN UPDATE:**

db.collection.updateOne({Name:"ranjita"},{$unset:{Name:""}})

**OUTPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**OR OPERATION:**

db.priyanka.find({$or:[{Name:'priya h'},{sec:'c'}]})

**OUTPUT:**

Text

Description automatically generated

**AND OPERATION:**

db.priyanka.find({$and:[{Name:'priya h'},{sec:'c'}]})

**OUTPUT:**

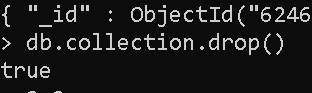
Text

Description automatically generated

**DROP OPERATION:**

db.collection.drop()

true



**MAPREDUCE IN MONGODB:**

> db.orders.insertMany([

... { \_id: 1, cust\_id: "Ant O. Knee", ord\_date: new Date("2020-03-01"), price: 25, items: [ { sku: "oranges", qty: 5, price: 2.5 }, { sku: "apples", qty: 5, price: 2.5 } ], status: "A" },

... { \_id: 2, cust\_id: "Ant O. Knee", ord\_date: new Date("2020-03-08"), price: 70, items: [ { sku: "oranges", qty: 8, price: 2.5 }, { sku: "chocolates", qty: 5, price: 10 } ], status: "A" },

... { \_id: 3, cust\_id: "Busby Bee", ord\_date: new Date("2020-03-08"), price: 50, items: [ { sku: "oranges", qty: 10, price: 2.5 }, { sku: "pears", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 4, cust\_id: "Busby Bee", ord\_date: new Date("2020-03-18"), price: 25, items: [ { sku: "oranges", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 5, cust\_id: "Busby Bee", ord\_date: new Date("2020-03-19"), price: 50, items: [ { sku: "chocolates", qty: 5, price: 10 } ], status: "A"},

... { \_id: 6, cust\_id: "Cam Elot", ord\_date: new Date("2020-03-19"), price: 35, items: [ { sku: "carrots", qty: 10, price: 1.0 }, { sku: "apples", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 7, cust\_id: "Cam Elot", ord\_date: new Date("2020-03-20"), price: 25, items: [ { sku: "oranges", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 8, cust\_id: "Don Quis", ord\_date: new Date("2020-03-20"), price: 75, items: [ { sku: "chocolates", qty: 5, price: 10 }, { sku: "apples", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 9, cust\_id: "Don Quis", ord\_date: new Date("2020-03-20"), price: 55, items: [ { sku: "carrots", qty: 5, price: 1.0 }, { sku: "apples", qty: 10, price: 2.5 }, { sku: "oranges", qty: 10, price: 2.5 } ], status: "A" },

... { \_id: 10, cust\_id: "Don Quis", ord\_date: new Date("2020-03-23"), price: 25, items: [ { sku: "oranges", qty: 10, price: 2.5 } ], status: "A" }

... ])

**1)query**

var mapFunction1 = function() {

... emit(this.cust\_id, this.price);

... };

> var reduceFunction1 = function(keyCustId, valuesPrices) {

... return Array.sum(valuesPrices);

... };

> db.priyankaa.mapReduce(

mapFunction1,

reduceFunction1,

{ out: "example" }

)

db.example.find().sort( { \_id: 1 } )

output:

Text

Description automatically generated

2) db.priyankaa.aggregate([

{ $group: { \_id: "$cust\_id", value: { $sum: "$price" } } },

{ $out: "agl" }])

>db.agl.find().sort( { \_id: 1 } )

Output:

Text

Description automatically generated

3)var mapFunction2 = function() {

for (var idx = 0; idx < this.items.length; idx++) {

var key = this.items[idx].sku;

var value = { count: 1, qty: this.items[idx].qty };

emit(key, value); }

};

> var reduceFunction2 = function(keySKU, countObjVals) {

reducedVal = { count: 0, qty: 0 };

for (var idx = 0; idx < countObjVals.length; idx++) {

reducedVal.count += countObjVals[idx].count;

reducedVal.qty += countObjVals[idx].qty;

}

return reducedVal;

};

> var finalizeFunction2 = function (key, reducedVal) {

reducedVal.avg = reducedVal.qty/reducedVal.count;

return reducedVal;

};

> db.priyankaa.mapReduce(

mapFunction2,

reduceFunction2,

{

out: { merge: "map" },

query: { ord\_date: { $gte: new Date("2020-03-01") } },

finalize: finalizeFunction2

} );

Text

Description automatically generated

**AGGREGATEPIPELINE IN MONGODB:**

**Inserting values:**

db.orders.insertMany( [{ \_id: 0, name: "Pepperoni", size: "small", price: 19,quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) },

{ \_id: 1, name: "Pepperoni", size: "medium", price: 20,quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) },

{ \_id: 2, name: "Pepperoni", size: "large", price: 21,quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) },

{ \_id: 3, name: "Cheese", size: "small", price: 12, quantity: 15, date : ISODate( "2021-03-13T11:21:39.736Z" ) } { \_id: 4, name: "Cheese", size: "medium", price: 13,quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) },

{ \_id: 5, name: "Cheese", size: "large", price: 14,quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) },

{ \_id: 6, name: "Vegan", size: "small", price: 17,quantity: 10, date : ISODate( "2021-01-13T05:08:13Z" ) },

{ \_id: 7, name: "Vegan", size: "medium", price: 18, quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) }] )

Text

Description automatically generated

1. db.orders.aggregate( [ {

$match: { size: "medium" }

}, {

$group: { \_id: "$name", totalQuantity: { $sum: "$quantity" } }

}] )

Output:

Text

Description automatically generated

**2)** db.orders.aggregate( [

{ $match: {

"date": { $gte: new ISODate( "2020-01-30" ), $lt: new ISODate( "2022-01-30" ) }

} }, {

$group: {

\_id: { $dateToString: { format: "%Y-%m-%d", date: "$date" } },

totalOrderValue: { $sum: { $multiply: [ "$price", "$quantity" ] } },

averageOrderQuantity: { $avg: "$quantity" }

}

}, {$sort: { totalOrderValue: -1 }} ] )

Output:

Text

Description automatically generated

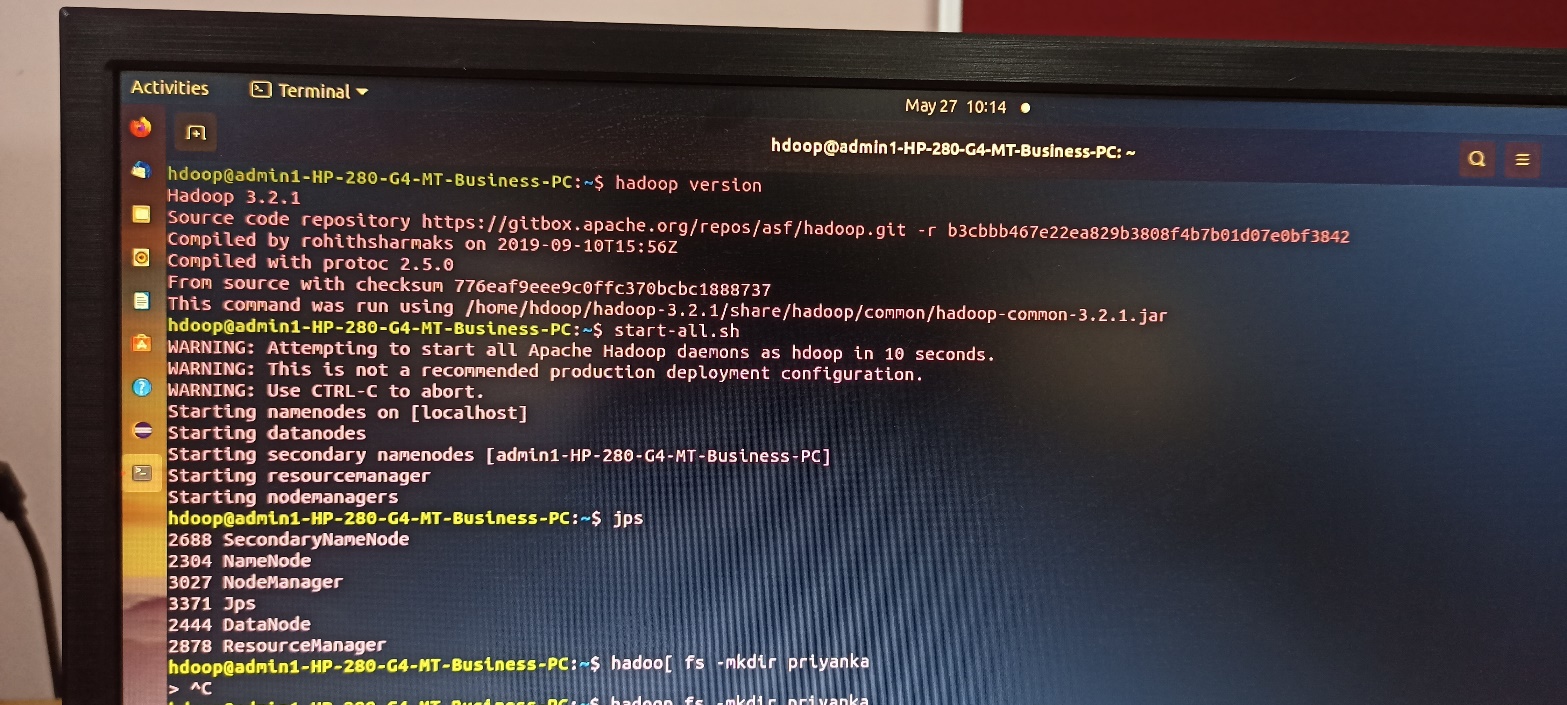
**HADOOP:**

**Entering in to hadoop:**

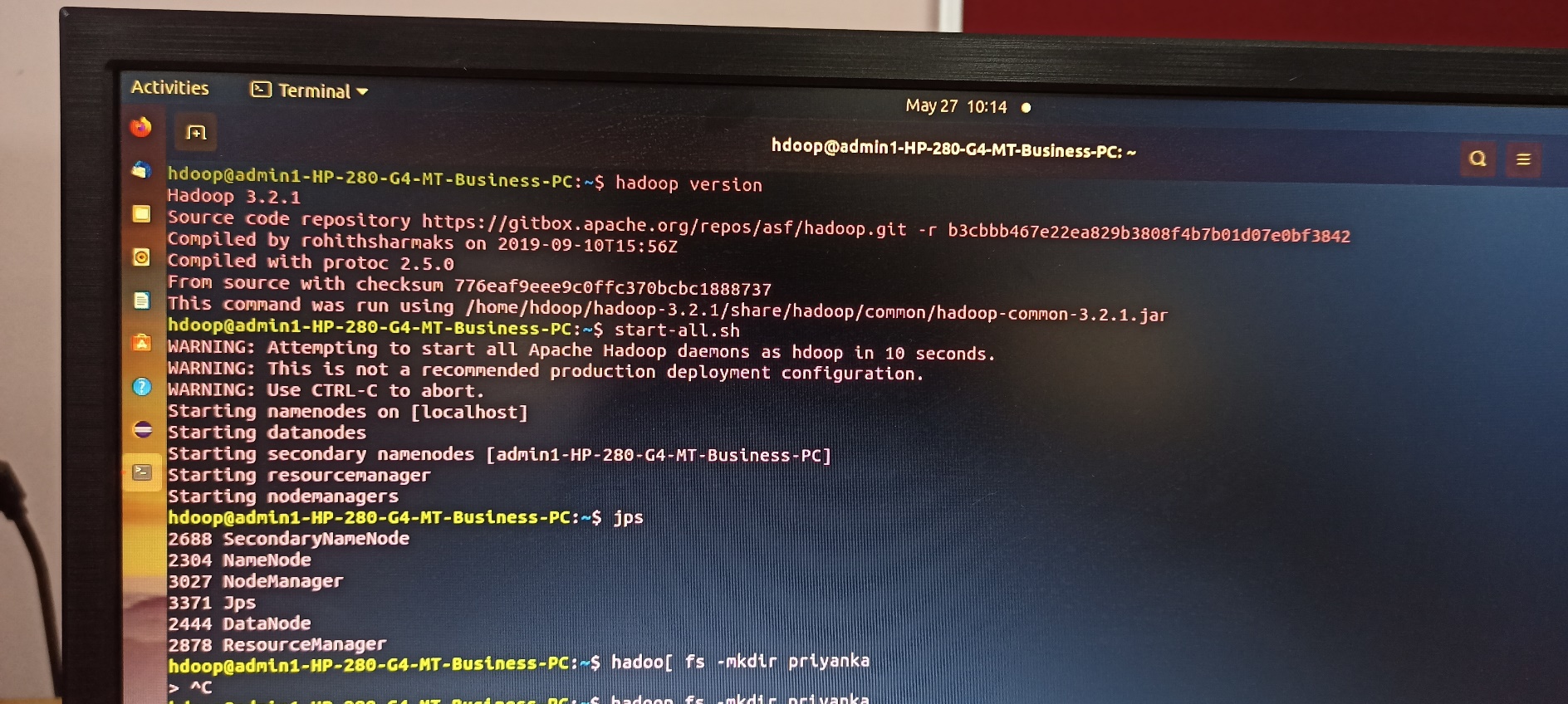
>Hadoop version

Hadoop 3. 2.1

>Start-all.sh



>jps



**CREATING DIRECTORY:**

1)>hadoop fs -mkdir priyanka

>hadoop fs -ls

Output:

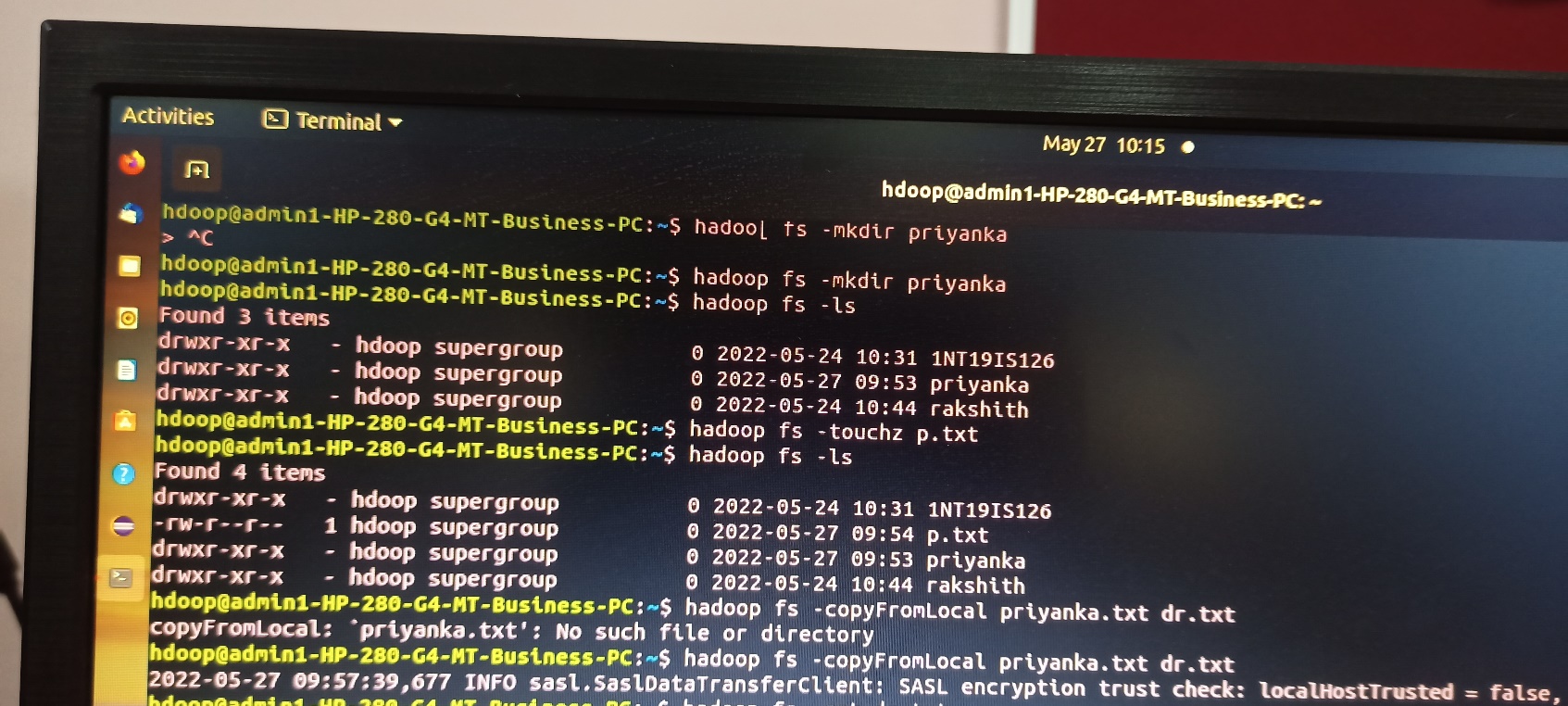
Text

Description automatically generated

2)>hadoop fs -touchz p.txt

>hadoop fs -ls

Output:



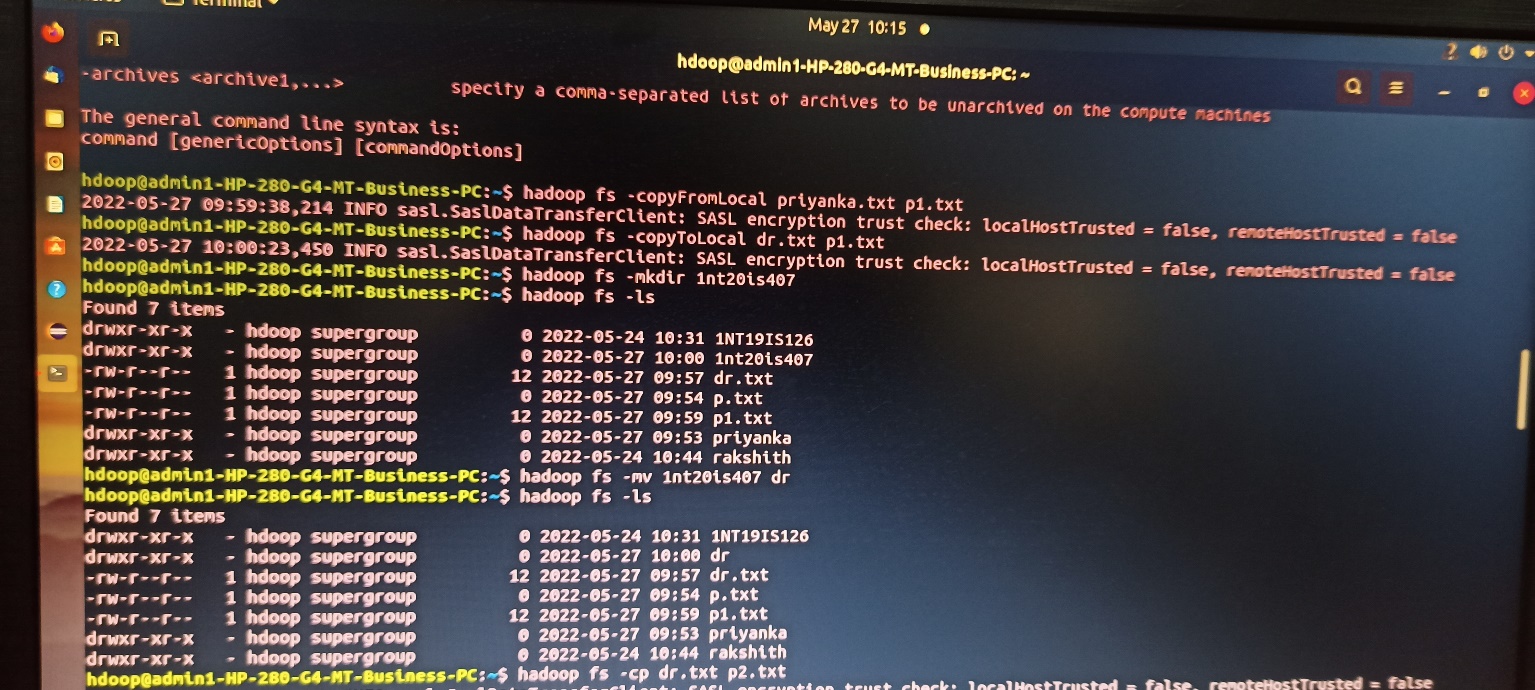
3)>hadoop fs -copyFromLocal priyanka.txt dr.txt

>hadoop fs -cat dr.txt

Text

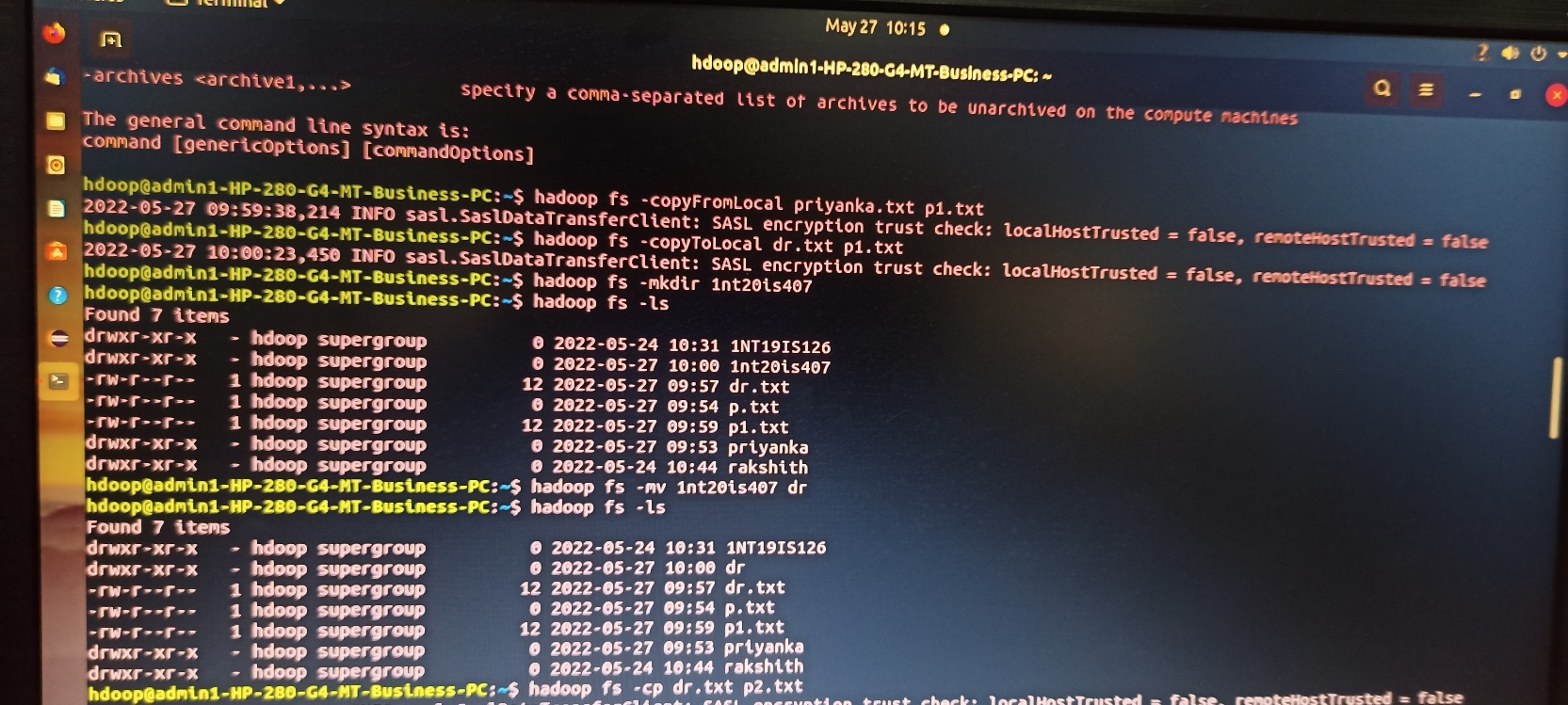
Description automatically generated

>hadoop fs -copyFromLocal priyanka txt p1.txt



>hadoop fs -mkdir 1nt20is407

>hadoop fs -ls



>hadoop fs -mv 1nt20is407 dr

>hadoop fs -ls

Text

Description automatically generated

>hadoop fs -cp dr.txt p2.txt

>hadoop fs -cat p2.txt

Text

Description automatically generated

>hadoop fs -du

A screenshot of a computer

Description automatically generated with medium confidence

>hadoop fs -rmr p.txt

>hadoop fs -ls

Text

Description automatically generated

>hadoop fs -stat priyanka

>hadoop fs -ls

Text

Description automatically generated

>hadoop fs -mv p2.txt priyanka

>hadoop fs -ls

