**1. 3rd highest salary holder name**

select \* from(

select ename, sal, dense\_rank() over(partition by dept order by sal desc)r from Employee)

where r=&n;

To find to the 2nd highest sal set n = 2

To find 3rd highest sal set n = 3 and so on.

SELECT name, salary

FROM #Employee e1

WHERE N-1 = (SELECT COUNT(DISTINCT salary) FROM #Employee e2

WHERE e2.salary > e1.salary)

SELECT TOP 1 salary FROM (SELECT DISTINCT TOP N salary FROM #Employee ORDER BY salary DESC) AS temp ORDER BY salary  
  
for the 2nd maximum you can replace N with 2, and for 3rd maximum replace N with 3, here is the output:

import org.apache.spark.\_  
import org.apache.log4j.{Level, Logger}  
  
object MaxSalByCity extends App{  
 Logger.*getLogger*("org").setLevel(Level.*ERROR*)  
  
 // Create a SparkContext using every core of the local machine  
 val *sc* = new SparkContext("local[\*]", "MaxSalByCity")  
 val *rdd2*=*sc*.textFile("C:\\Users\\singh\\Desktop\\ScalaProjectFiles\\EmpInfo.txt").map{ x => x.split(" ") }  
 .map{ x =>((x(2)),(x(1),x(4).toDouble))}.groupByKey  
 *rdd2*.foreach(*println*)  
 for(i<-*rdd2*)  
 {  
 *println*(i.\_1,i.\_2.filter(x=>x.\_2==i.\_2.map(x=>x.\_2).max))  
 }  
}

**2. Partioning and Bucketing**

<https://www.guru99.com/hive-partitions-buckets-example.html>

Partition divides large amount of data into multiple slices based on value of a table column(s).  
  
Assume that you are storing information of people in entire world spread across 196+ countries spanning around 500 crores of entries. If you want to query people from a particular country (Vatican City), in absence of partitioning, you have to scan all 500 crores of entries even to fetch thousand entries of a country. If you partition the table based on country, you can fine tune querying process by just checking the data for only one country partition. Hive partition creates a separate directory for a column(s) value.  
  
Bucketing decomposes data into more manageable or equal parts.  
  
With partitioning, there is a possibility that you can create multiple small partitions based on column values. If you go for bucketing, you are restricting number of buckets to store the data. This number is defined during table creation scripts.

**3. Performance optimization in Hive**

<https://acadgild.com/blog/hive-optimization-techniques-with-examples>

<https://www.qubole.com/blog/hive-best-practices/>

<https://data-flair.training/blogs/hive-optimization-techniques/>

**4. Skew data**

<https://data-flair.training/blogs/skew-join-in-hive/>

<https://medium.com/expedia-group-tech/skew-join-optimization-in-hive-b66a1f4cc6ba>

<https://cwiki.apache.org/confluence/display/Hive/Skewed+Join+Optimization>

**5. Joins in spark**

<https://acadgild.com/blog/what-join-apache-spark>

<https://medium.com/@achilleus/https-medium-com-joins-in-apache-spark-part-1-dabbf3475690>

<https://databricks.com/session/optimizing-apache-spark-sql-joins>

<https://towardsdatascience.com/the-art-of-joining-in-spark-dcbd33d693c>

**6. How can we achieve high availability**

<https://data-flair.training/blogs/hadoop-high-availability-tutorial/>

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSHighAvailabilityWithNFS.html>

<https://www.dezyre.com/article/what-is-hadoop-2-0-high-availability/87>

**7. If we have processes running in spark and some process are stuck how to debug them**

you can put Log4j statements to do logging during spark execution. Logs can then be collected from cluster.

Since your execution is stuck, you need to check the Spark Web UI and drill down from

Job > Stages > Tasks and try and figure out what is causing things to stuck.

Some of the generic questions asked are: a. How many executors are running b. Is there a stage/task that is getting re-created after failure? Is there a memory contention? d. Does Garbage collection taking just too long to finish. e. How much time it is expected to take. f. Is server having enough CPU and memory.

*A stage is nothing but a step in a physical execution plan*. It is a physical unit of the execution plan. It is a set of parallel tasks i.e. one task per partition. In other words, each job which gets divided into smaller sets of tasks is a stage. Although, it totally depends on each other. However, we can say it is as same as the map and reduce stages in [MapReduce](https://data-flair.training/blogs/hadoop-mapreduce-tutorial/)**.**

*Executors in Spark are worker nodes*. Those help to process in charge of running individual tasks in a given Spark job. Moreover, we launch them at the start of a Spark application. Then it typically runs for the entire lifetime of an application. As soon as they have run the task, sends results to the driver. Executors also provide in-memory storage for [Spark RDDs](https://data-flair.training/blogs/apache-spark-rdd-tutorial/) that are cached by user programs through Block Manager

<https://tech.channable.com/posts/2018-04-10-debugging-a-long-running-apache-spark-application.html>

<https://stackoverflow.com/questions/53150630/how-to-debug-the-apache-spark-when-it-stuck-at-the-certain-line>

[**https://mapr.com/support/s/article/Spark-Troubleshooting-Guide-Master-list-with-hyperlinks-to-detailed-articles?language=en\_US**](https://mapr.com/support/s/article/Spark-Troubleshooting-Guide-Master-list-with-hyperlinks-to-detailed-articles?language=en_US)

<https://docs.databricks.com/spark/latest/rdd-streaming/debugging-streaming-applications.html>

**8. How to get no. of object instances initiated for a class**

<https://stackoverflow.com/questions/8628123/counting-instances-of-a-class>

<https://docs.scala-lang.org/tour/singleton-objects.html>

**9. AWS technologies**

**S3 :** <https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html>

<https://www.javatpoint.com/aws-s3-concepts>

<https://www.edureka.co/blog/understanding-amazon-s3/>

**EC2 :** <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

<https://www.edureka.co/blog/ec2-aws-tutorial-elastic-compute-cloud/>

<https://searchaws.techtarget.com/definition/Amazon-Elastic-Compute-Cloud-Amazon-EC2>

<https://www.datacamp.com/community/tutorials/aws-ec2-beginner-tutorial>

**Lambda :** <https://www.guru99.com/aws-lambda-function.html>

<https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-concepts.html>

<https://docs.aws.amazon.com/lambda/latest/dg/with-s3.html>

**RedShift :** <https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html>

<https://www.sqlbot.co/blog/aws-redshift-what-is-it-15-minute-tutorial-explanation>

<https://www.tutorialspoint.com/amazon_web_services/amazon_web_services_redshift.htm>

**Athena :** <https://docs.aws.amazon.com/athena/latest/ug/what-is.html>

<https://cloudacademy.com/blog/what-is-amazon-athena/>

<https://mindmajix.com/aws-athena> <https://tutorialsdojo.com/amazon-athena/>

<https://www.edureka.co/blog/amazon-athena-tutorial>

<https://towardsdatascience.com/query-data-from-s3-files-using-aws-athena-686a5b28e943>

**IQ :** <https://www.onlineinterviewquestions.com/aws-s3-interview-questions/>

<https://www.wisdomjobs.com/e-university/aws-ec2-interview-questions.html>

<https://mindmajix.com/aws-lambda-interview-questions>

<https://www.whizlabs.com/blog/aws-database-interview-questions/>

<https://career.guru99.com/top-15-aws-interview-questions/>

<https://mindmajix.com/aws-interview-questions>

**10. NoSQL basic concepts**

<https://www.guru99.com/nosql-tutorial.html>

<https://www.educba.com/nosql-interview-questions/>

<https://intellipaat.com/blog/interview-question/no-sql-interview-questions/>

**11. How to merge files in a hdfs directory in a single job**

<https://www.edureka.co/community/53340/what-the-best-way-merge-multi-part-hdfs-files-into-single-file>

<https://acadgild.com/blog/merging-files-hdfs>

<https://stackoverflow.com/questions/3548259/merging-multiple-files-into-one-within-hadoop>

**12. How to fix if we want to write data to a single file itself in spark**

[**https://www.dataneb.com/post/how-to-write-single-csv-file-using-spark**](https://www.dataneb.com/post/how-to-write-single-csv-file-using-spark)

<https://stackoverflow.com/questions/31674530/write-single-csv-file-using-spark-csv>

**13. Count no. of occurrences of a letter in a string and print letter with highest occurrence**

<https://www.javatpoint.com/apache-spark-char-count-example>

<https://www.edureka.co/community/35107/python-program-to-print-occurrence-of-character-in-string>

**14. Python what is list and tuples**

<https://www.afternerd.com/blog/difference-between-list-tuple/>

<https://www.geeksforgeeks.org/python-difference-between-list-and-tuple/>

<https://www.tutorialspoint.com/difference-between-list-and-tuples-in-python>

**15. How to add an item in a dictionary in Python**

<https://www.journaldev.com/23232/python-add-to-dictionary>

<https://www.geeksforgeeks.org/python-add-new-keys-to-a-dictionary/>

**16. Closure example in Python**

<http://zetcode.com/python/python-closures/>

<https://www.geeksforgeeks.org/python-closures/>

<https://www.programiz.com/python-programming/closure>