BI

Group Project

Job Vacancy Analysis

(Group 4)

Rohan Chopra(46024286)

Table of Contents

[1. Introduction 3](#_Toc534726633)

[**1.1.** **Setup Checklist for Group Project** 3](#_Toc534726634)

[**1.2.** **Instructions** 3](#_Toc534726635)

[2. Problem Statement 4](#_Toc534726636)

[**2.1.** **Objective** 4](#_Toc534726637)

[**2.2.** **Technology used:** 4](#_Toc534726638)

[3. Implementation in BI LOT 5](#_Toc534726639)

[**3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:** 5](#_Toc534726640)

[**3.2 Guidelines on the functionality to be built:** 5](#_Toc534726641)

# **Introduction**

This document outlines a mini project for the BI LOT. The project is to analyse Job Vacancy Analysis. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the BI LOT.

## **Setup Checklist for Mini Project**

Minimum System Requirements

* Intel Pentium 90 or higher (P166 recommended)
* Microsoft Windows 2010 or above
* Memory: 8GB of RAM (4GB or more recommended)
* Internet Explorer 10.0 or higher
* MySQL/ HIVE/ Beeline
* SQOOP (ETL Tool)
* Apache Spark

## **Instructions**

* The code modules in the mini project should follow all the coding standards.
* Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **GroupProject**. Store your Project here.
* You can refer to your course material.
* You may also look up the help provided in the BI docs and documentation provided in respective tools.

# **Problem Statement**

## **Objective**

* To compare different Jobs in different Cities from the given data by performing ETL and generating reports.
* Helps the candidates with different choices of jobs in different cities based on their various criteria.
* Help students to identify industries in which there is higher chances to get a job.

## **Technology used:**

* MySql / HIVE / Microsoft EXCEL (Database) / Beeline
* SQOOP (ETL Tools)
* Apache Spark

# **Implementation in BI LOT**

## **3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:**

The participants need to develop the Job Vacancy Analysis by building the functionality incrementally in each of the course modules of BI LOT using one of the above ETL and reporting tool.

## **3.2 Guidelines on the functionality to be built:**

**Project flow:**

Heterogeneous Sources i.e. operational data (Flat files)

EXTRACT

TRANSFORM

LOAD

Data warehouse (Oracle)

Reports Based On Tool

R

E

P

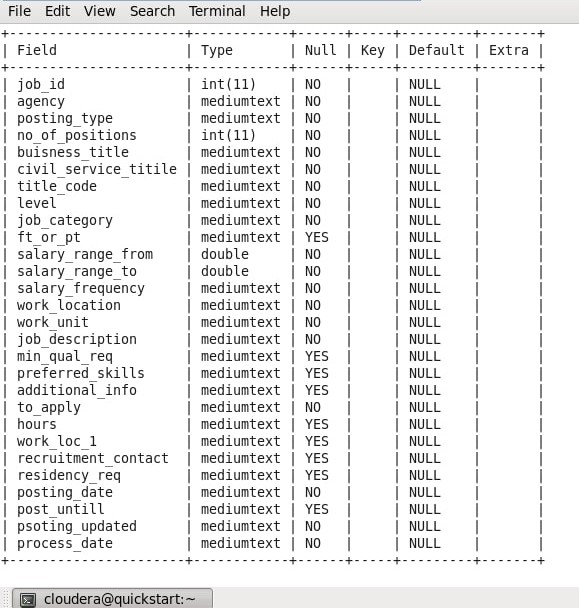
O

R

T

**Schema Design:**

* The project has a single table schema approach.
* It consists of 28 fields.
* The fields of tables are job\_id, agency, posting\_type, no\_of\_positions, buisness\_title, civil\_service\_titile, title\_code, level, job\_category, ft\_or\_pt, salary\_range\_from, salary\_range\_to, salary\_frequency, work\_location, work\_unit, job\_description, min\_qual\_req, preferred\_skills, additional\_info, to\_apply, hours, work\_loc\_1, recruitment\_contact, residency\_req, posting\_date, post\_untill, psoting\_updated, process\_date.



Dimension of table and it’s columns:

1. nyc\_job

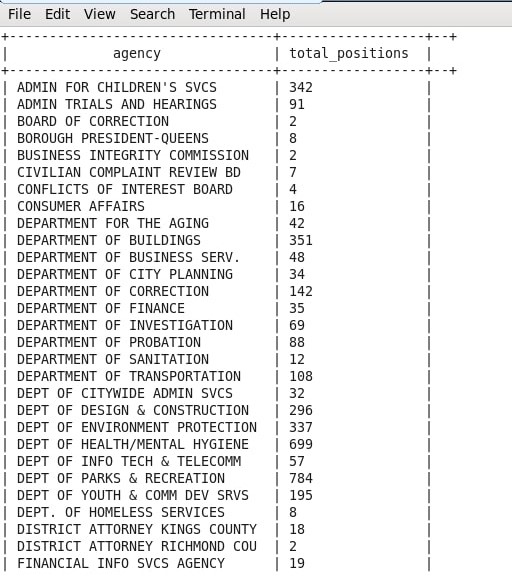
* job\_id int
* agency string
* posting\_type string
* no\_of\_positions int
* buisness\_title string
* civil\_service\_titile string
* title\_code string
* level string
* job\_category string
* ft\_or\_pt string
* salary\_range\_from double
* salary\_range\_to double
* salary\_frequency string
* work\_location string
* work\_unit string
* job\_description string
* min\_qual\_req string
* preferred\_skills string
* additional\_info string
* to\_apply string
* hours string
* work\_loc\_1 string
* recruitment\_contact string
* residency\_req string
* posting\_date string
* post\_untill string
* psoting\_updated string
* process\_date string

**SAMPLE REPORTS:**

Reports that can be generated:

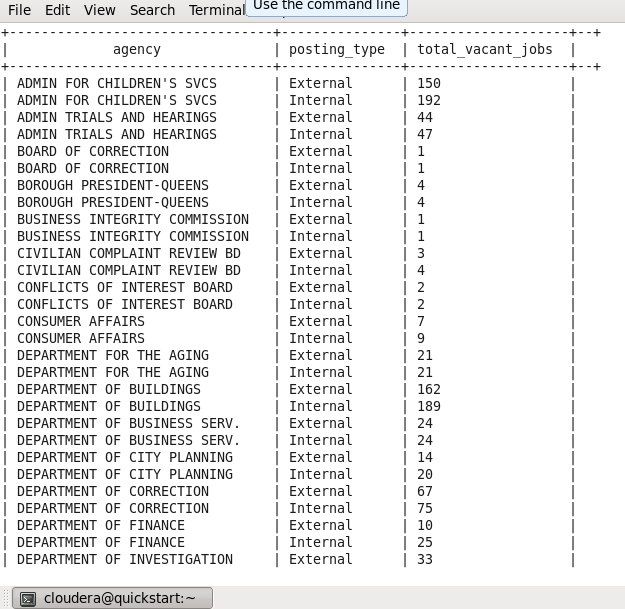
1. Most in demand job vacancies in each Agency.

select agency,sum(no\_of\_positions) as Total\_Positions from nyc\_job group by agency;



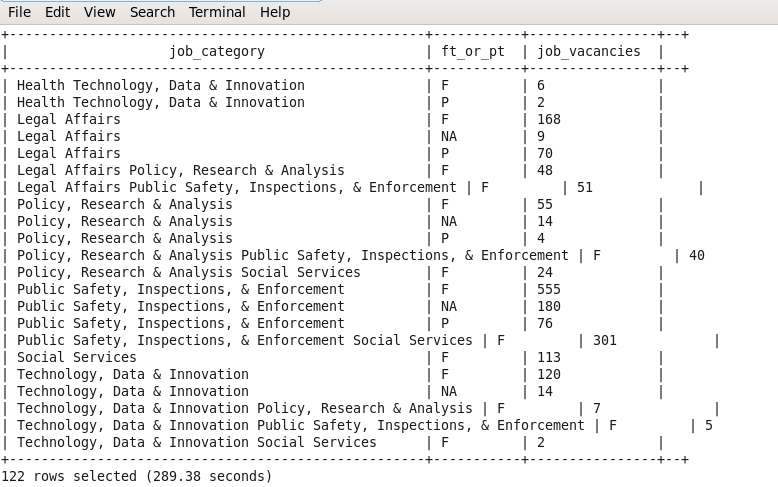
1. Number of external and internal jobs vacancy in each Agency.

select agency,posting\_type,sum(no\_of\_positions) as TOTAL\_VACANT\_JOBS from partition\_posting\_type where posting\_type='External' OR posting\_type='Internal' group by agency,posting\_type;



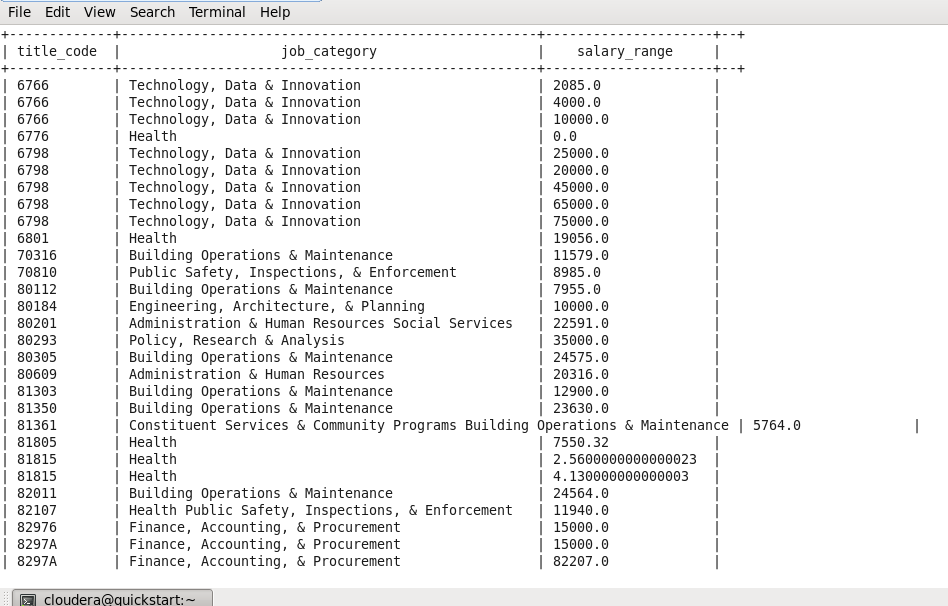
1. Determine number of part time and full time jobs vacancies in different job categories.

select job\_category,ft\_or\_pt,sum(no\_of\_positions) as JOB\_VACANCIES from nyc\_job group by job\_category,ft\_or\_pt;



1. Display different jobs with salary ranges (0 indicates that salary is fixed and there is no increment).

Select distinct title\_code,job\_category,(salary\_range\_to-salary\_range\_from) as salary\_range from nyc\_job order by title\_code;



1. Agency and Job Category having Highest paying jobs.

select z.agency,z.job\_category,z.wages from (select agency,job\_category,

case when salary\_frequency in ('Hourly') then salary\_range\_to\*9\*269

when salary\_frequency in ('Daily') then salary\_range\_to\*269

else salary\_range\_to

end as wages from nyc\_job)z where z.wages in

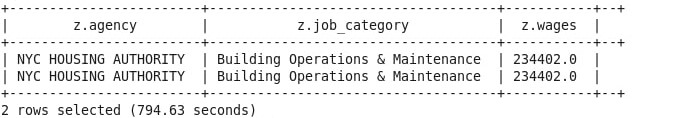
(select max(u.wages) from (select agency,job\_category,

case when salary\_frequency in ('Hourly') then salary\_range\_to\*9\*269

when salary\_frequency in ('Daily') then salary\_range\_to\*269

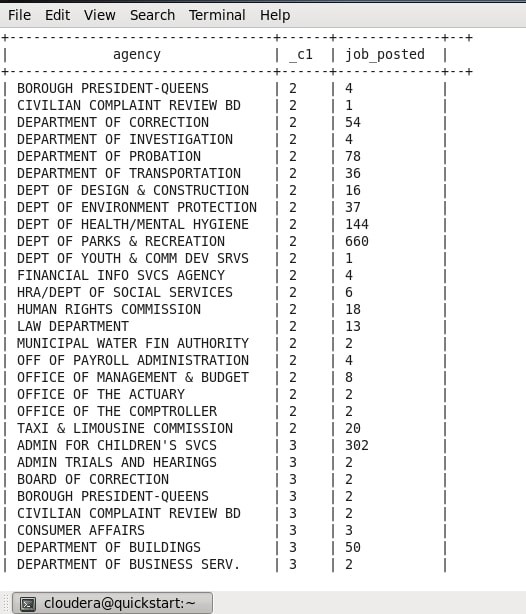
else salary\_range\_to

end as wages from nyc\_job)u);



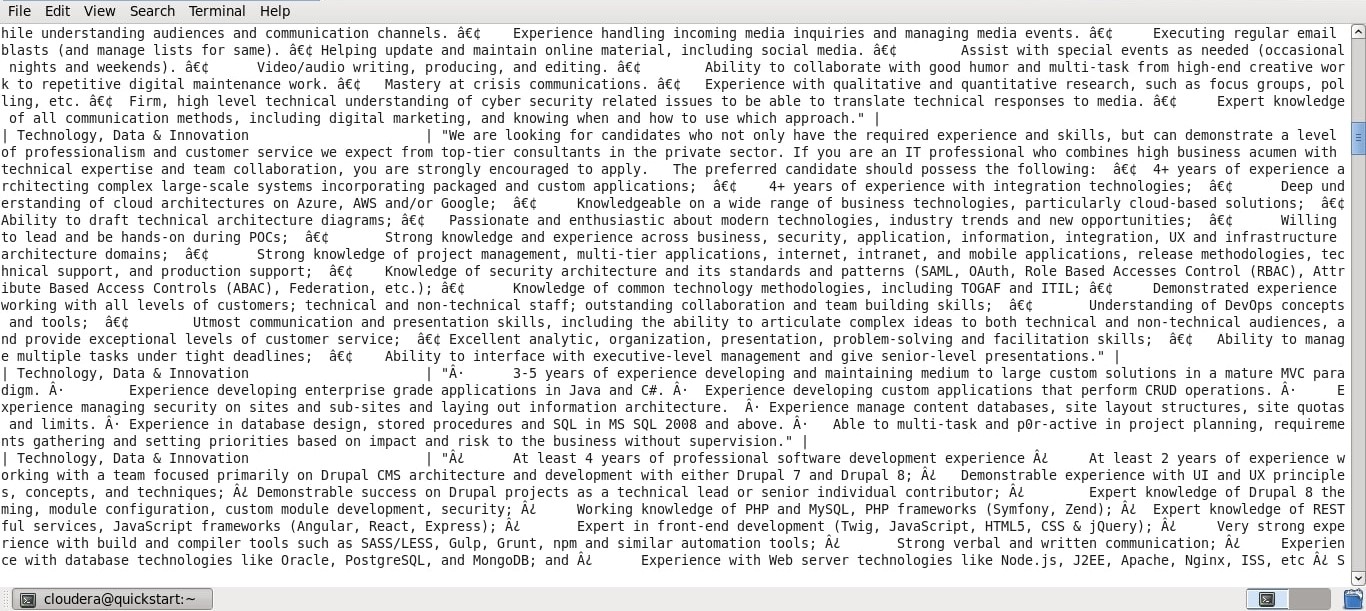
1. Month wise number of job posted in all agencies.

select agency,month(posting\_date),sum(no\_of\_positions) as job\_posted from nyc\_job group by month(posting\_date),agency;



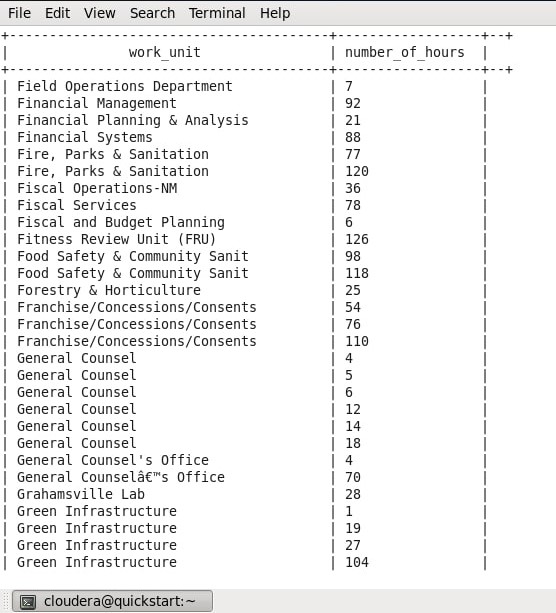
1. Most popular preferred skills per job category

Select distinct(job\_category) as JOB\_CATEGORY,preferred\_skills as SKILLS\_REQUIRED from nyc\_job;



1. Days taken in job application for each work unit

select distinct(work\_unit),datediff(to\_date(process\_date),to\_date(posting\_date))as NUMBER\_OF\_HOURS from nyc\_job;



1. Number of job positions in each work location

select work\_location,sum(no\_of\_positions) as TOTAL\_JOB\_POSITIONS from nyc\_job group by work\_location order by TOTAL\_JOB\_POSITIONS desc;

