

## USE CASE STUDY REPORT : Milestone 3

**Group No.:** Group 15

**Student Names:** Vraj Diyora and Amruta Hombali

This report follows from the previous report which contained the relational database. The tables were created in SQL from this schema and populated with realistic sample data. Here are some of the scenario questions asked regarding the job search tool cross platform database and their respective SQL queries

SQL Queries :

- 1) The Analytics team at Job Search tool cross platform wants to know jobs derived from a particular website, For ex: the list of jobs which includes the JobId, Job Title, location, Salary that were posted on a particular website called 'My works'. The SQL query for the same is:

**Select** j.jobID, j.Job\_Title, j.Company\_Name, j.Location, j.Salary  
**From** job\_listing j  
**Inner Join** website w on j.JobID=w.JobID  
**Where** websitename = 'myworks';

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

|   | jobID      | Job_Title                    | Company_Name | Location  | Salary |
|---|------------|------------------------------|--------------|-----------|--------|
| ▶ | 59-6716525 | Civil Engineer               | Tagfeed      | Rokytnice | 340579 |
|   | 40-0162691 | Computer Systems Analyst III | Rhynoodle    | Irاندوبا  | 393649 |

- 2) The Analytics team at Job Search tool cross platform is assigned to understand the popularity and demand of the job title 'Paralegal'. Hence they want to know the total number of applicants that applied for a job title named 'Paralegal'. The SQL query for the same is:

**Select** Sum(numberofapplicants) as TotalApplicants  
**From** applicants a  
**Inner Join** job\_listing j on a.jobid=j.jobid  
**Where** j.job\_title='paralegal';

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

|  |                 |
|--|-----------------|
|  | TotalApplicants |
|  | 993             |

- 3) The Analytics team at Job Search tool cross platform wants to know of all the companies that are actively hiring and posting multiple projects on the platform, particularly companies that posted more than 5 jobs in descending order. The SQL query for the same is:

```
Select company_name, count(*) as total_jobs_posted
From job_listing
Group by company_name
Having count(*)>5
Order by count(*) desc
Limit 10;
```

Output:

| Result Grid |              | Filter Rows:      | Export: | Wrap Cell Content: | Fetch rows: |
|-------------|--------------|-------------------|---------|--------------------|-------------|
|             | company_name | total_jobs_posted |         |                    |             |
|             | Skimia       | 9                 |         |                    |             |
|             | Quatz        | 9                 |         |                    |             |
|             | Bubblemix    | 8                 |         |                    |             |
|             | Jabbersphere | 8                 |         |                    |             |
|             | Cogilith     | 8                 |         |                    |             |
|             | Linkbuzz     | 8                 |         |                    |             |
|             | Wordify      | 8                 |         |                    |             |
|             | Mydo         | 7                 |         |                    |             |
|             | Mybuzz       | 7                 |         |                    |             |

- 4) The Analytics team at Job Search tool cross platform wants to analyse the job market and the average salaries offered in various locations, this can help also the job seekers to make further decisions. They want to find average salary for jobs posted in each location, but only for locations where the average salary is greater than \$120,000. The SQL query for the same is:

```
Select location, avg(salary) as mean_salary
From job_listing
Group by location
Having avg(salary)> 120000
Limit 10;
```

Output:

| Result Grid |            | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows: |
|-------------|------------|--------------|---------|--------------------|-------------|
|             | location   | mean_salary  |         |                    |             |
| ▶           | Qiaodi     | 142494.0000  |         |                    |             |
|             | MulchÃ©n   | 263488.0000  |         |                    |             |
|             | ÃtiÃ#evac  | 185430.0000  |         |                    |             |
|             | Melipilla  | 140002.0000  |         |                    |             |
|             | SibatÃ©    | 164018.0000  |         |                    |             |
|             | Tupesy     | 165381.0000  |         |                    |             |
|             | Forquilha  | 385014.0000  |         |                    |             |
|             | Cincinnati | 388221.0000  |         |                    |             |
|             | Anjiang    | 279597.0000  |         |                    |             |

- 5) The Analytics team at Job Search tool cross platform wants to the top 10 most popular job listings (by number of applicants) to create a report for all the stakeholders involved. The sql query for the same is:

```
SELECT job_listing.JobID, Company_Name, Job_Title, Location, Salary, COUNT(*)
AS NumApplicants
FROM Job_Listing
INNER JOIN Applicants ON Job_Listing.JobID = Applicants.JobID
GROUP BY Job_Listing.JobID
ORDER BY NumApplicants DESC
LIMIT 10;
```

Output:

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

|   | JobID      | Company_Name | Job_Title               | Location  | Salary | NumApplicants |
|---|------------|--------------|-------------------------|-----------|--------|---------------|
| ▶ | 83-5296491 | Avaveo       | Programmer IV           | Baqiu     | 246600 | 2             |
|   | 52-6528690 | Yozio        | Director of Sales       | Tianchang | 172564 | 2             |
|   | 00-0571656 | Miboo        | VP Accounting           | Qiaodi    | 142494 | 1             |
|   | 00-1033402 | Skinix       | Food Chemist            | MulchÃ©n  | 263488 | 1             |
|   | 00-2404738 | Avamm        | Web Developer I         | Melipilla | 140002 | 1             |
|   | 00-3438022 | Zoovu        | Programmer II           | SibatÃ©   | 164018 | 1             |
|   | 00-6969012 | Babbleset    | Human Resources Manager | Oropesa   | 108938 | 1             |
|   | 00-7103579 | Youbridge    | VP Sales                | Forquilha | 385014 | 1             |
|   | 99-9682022 | Kaymbo       | Account Executive       | Yidian    | 118135 | 1             |
|   | 01-4777451 | Tanoodle     | Environmental Tech      | Ginebra   | 117261 | 1             |

Result 1



- 6) The Job search tool cross platform analytics team wants to analyze which employers are actively posting jobs and how many job listings they are posting, in order to check the employer trends. Hence they want to get the total number of job listings posted by each employer. The SQL query for the same is:

```

SELECT Employer.EmployerID, EmployerName, COUNT(*) AS TotalJobListings
FROM JobPosts
INNER JOIN Employer ON JobPosts.EmployerID = Employer.EmployerID
GROUP BY Employer.EmployerID
LIMIT 10;

```

Output:

| Result Grid    Filter Rows: <input type="text"/>   Export:  Wra |            |              |                  |
|---|------------|--------------|------------------|
|   | EmployerID | EmployerName | TotalJobListings |
| ▶   | 00-7380753 | InnoZ        | 1                |
|   | 00-9338979 | Npath        | 1                |
|   | 01-1644313 | Riffpedia    | 1                |
|   | 01-2580029 | Babbleopia   | 1                |
|   | 01-5375538 | Izio         | 1                |
|   | 01-9768578 | Miboo        | 1                |
|   | 02-0140098 | Centizu      | 1                |
|   | 02-0838045 | Skimia       | 1                |
|   | 02-2582715 | Taofeed      | 1                |



- 7) The Analytics team at Job Search tool cross platform wants to analyze the the job titles that are most in demand(number of applicants) in different locations and the compensation offered respectively in order to check the supply and demand for each job. Hence they will be Comparing the number of job postings and the number of applicants for each job title across different locations. The SQL query for the same is :

```

Select j.job_title, j.location, Count(distinct j.Jobid) As No_of_Postings,
Sum(a.NumberofApplicants) As No_of_Applicants
From job_listing j
Left join jobposts jp on j.jobID=jp.jobID
Left join applicants a on j.jobID=a.jobID
Group by j.job_title, j.location
Limit 10;

```

Output:

| Result Grid |                     |              |                |  |
|-------------|---------------------|--------------|----------------|--|
|             |                     | Filter Rows: |                | Export:  Wrap Cell Content:  Fetch rows: |
|             | job_title           | location     | No_of_Postings | No_of_Applicants   |
| ▶           | Account Coordinator | Kang-neung   | 1              | NULL   |
|             | Account Coordinator | Lagarto      | 1              | 155  |
|             | Account Coordinator | Lens         | 1              | 85   |
|             | Account Coordinator | Pszczyna     | 1              | 110  |
|             | Account Coordinator | Raszowa      | 1              | NULL   |
|             | Account Coordinator | Rinc n       | 1              | 178  |
|             | Account Coordinator | Villa Ocampo | 1              | NULL   |
|             | Account Executive   | Billdal      | 1              | NULL   |
|             | Account Executive   | Camilaca     | 1              | NULL   |
|             | Account Executive   | Caucag ito   | 1              | NULL   |



- 8) The Analytics team at Job Search tool cross platform is given the task to estimate salary ranges in particular highlight the companies offering higher salaries than their avg salary for that job title. The SQL Query for the same is :

```

SELECT job_title, salary
FROM job_listing j
WHERE salary > (
SELECT AVG(salary)
FROM job_listing
WHERE job_title = j.job_title
);

```

Output:

| Result Grid |                         |  |
|-------------|-------------------------|--|
|             |                         | Filter Rows: <input type="text"/>  |
|             |                         | Export:  Wrap Cell Content:  |
|             | job_title               | salary   |
| ▶           | VP Sales                | 385014   |
|             | Paralegal               | 359751   |
|             | Senior Quality Engineer | 253906   |
|             | Junior Executive        | 323277   |
|             | Research Nurse          | 358390   |
|             | Paralegal               | 271811   |
|             | Chief Design Engineer   | 311933   |
|             | Web Developer III       | 234815   |
|             | Senior Sales Associate  | 242668   |
|             | Accountant III          | 376907   |
|             | Office Assistant II     | 316382   |
|             | Geological Engineer     | 381569   |

- 9) The Analytics team at Job search tool cross platform wants to provide more targeted job posts, and showcase the location having the highest demand for jobs. Hence they want to find the location with the highest job post. The sql query for the same is:

```
WITH temp_table AS  
(SELECT location  
,COUNT(*) as num_postings  
FROM job_listing  
GROUP BY location  
)  
SELECT location  
,COUNT(*) as num_postings  
FROM job_listing  
GROUP BY location  
HAVING COUNT(*) = (SELECT MAX(num_postings) FROM temp_table);
```

Output:

| Result Grid |                 |              | Filter Rows: | Export: | Wrap Cell Content: |
|-------------|-----------------|--------------|--------------|---------|--------------------|
|             | location        | num_postings |              |         |                    |
| ▶           | Lazaro Cardenas | 3            |              |         |                    |