Case Study 1 : Job Data Analysis

A)Jobs Revived Over Time

Query : select count(distinct job_id)/(30*24)

as jobs_per_day

from job_data

where monthname(cast(ds as date))='November';

output :

```
#A)no of jobs reviewd

select  count(distinct job_id)/(30*24)
as jobs_per_day
from job_data
where monthname(cast(ds as date))='November';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

jobs_per_day
0.0083

B) Throughput Analysis

Query : with query1 as

(select ds, count(distinct event) as total_events

from job_data

group by ds)

select ds,total_events,avg(total_events) over (order by ds rows between 6 preceding and current row) as 7_day_rolling_average

from query1;

Reason : Since 7 day rolling average provides better understanding of trends over time, as it provides a longer term perspective comperatively.

```
#Throughput Analysis

with query1 as
(select ds, count(distinct event) as total_events
from job_data
group by ds)
select ds,total_events,avg(total_events) over (order by ds rows between 6 preceding and current row ) as 7_day_rolling_avera
from query1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ds	total_events	7_day_rolling_average
2020-11-25 00:00:00	1	1.0000
2020-11-26 00:00:00	1	1.0000
2020-11-27 00:00:00	1	1.0000
2020-11-28 00:00:00	2	1.2500
2020-11-29 00:00:00	1	1.2000
2020-11-30 00:00:00	2	1.3333

Result 6 | Read Only

C)Language Share Analysis

Query : select language, count(language) as total_languages, count(*)*100/sum(count(*)) over() as percentage from job_data group by language;

```
#language share analysis
select language, count(language) as total_languages,
count(*)*100/sum(count(*)) over() as percentage from job_data group by language;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

language	total_languages	percentage
English	1	12.5000
Arabic	1	12.5000
Persian	3	37.5000
Hindi	1	12.5000
French	1	12.5000
Italian	1	12.5000

D)Duplicate rows detection

Query: with query1 as(select * , row_number() over (partition by job_id) as row_Num from job_data)

select * from query1 where row_num>1;

```
#duplicate row detection

with query1 as(select * , row_number() over (partition by job_id) as row_Num
from job_data)
select * from query1 where row_num>1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

job_id	actor_id	event	language	time_spent	org	ds	row_Num
23	1005	transfer	Persian	00:00:22	D	2020-11-28 00:00:00	2
23	1004	skip	Persian	00:00:56	A	2020-11-26 00:00:00	3

Project details:

Description : The purpose of this project is to identify no of jobs reviewed, to identify the duplicate records present in the data given and so on.

Approach: As a first step I went through the project to identify the topics that I need to implement in the project . later I learnt those concepts and implemented in the project.

Tech-stack used : MySQL workbench 8.0

Reason : I have used MySQL workbench. Reason why I have selected the mentioned tech stack is it is cross platform compatible open source RDBMS.

Insights : The insights we can draw from this project is to implement & day rolling average concept, number of jobs reviewed and so on

Result : I was able to extract the required data as per the project given. I gained knowledge in new concepts like calculating throughput, 7 day rolling average and so on.