

trainity

# PROJECT 3

BY ASHISH KUMAR SAMANTARAY



# PROJECT DESCRIPTION:

In this project; we will take on the role of a **Lead Data Analyst** at a company like **Microsoft**. You'll be provided with various datasets and tables, and we will have to derive insights from this data to answer questions posed by different **departments** within the company. Our task is to use our advanced **SQL** skills to analyze the data and provide valuable insights that can help improve the **company's operations** and understand sudden changes in key metrics.



## TECH STACK USED:

- **XAAMP CONTROL PANEL v3.3.0** SUPPORTING MYSQL
- **CANVA** FOR CREATING PPT

I chose **XAAMP CONTROL PANEL** because previously I had already worked on Xaamp Admin supporting mySQL where I had performed some queries for a PG room finding site database.

I chose **Canva** so as to make my PPT look more visually appealing.

MADE IN  
*Canva*



# insights AHEAD

WITH DETAILED APPROACH AND OUTPUT AND QUERY BOX

Case Study 1

# JOB DATA ANALYSIS



# TASK A

We are required to calculate the number of jobs reviewed per hour for each day in **November 2020** for which we need to write SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

## QUERY

```
1 select ds, count(job_id) as no_of_jobs,  
2 sum(time_spent)/3600 as total_time_in_hours,  
3 count(job_id)/(sum(time_spent)/3600) as avg_jobreviewedperhour  
4 from job_data  
5 group by ds;
```

## OUTPUT


ds	no_of_jobs	total_time_in_hours	avg_jobreviewedperhour
11/25/2020	1	0.0125	80.0000
11/26/2020	1	0.0156	64.2857
11/27/2020	1	0.0289	34.6154
11/28/2020	2	0.0092	218.1818
11/29/2020	1	0.0056	180.0000
11/30/2020	2	0.0111	180.0000

# TASK A

---

## APPROACH

As we need insight of each day of the month, we need to count the total no of jobs, **sum of total time spent** and we divide it by **3600** so as to change it to hours and then divide total no of jobs divided by total hours spend so as to get the **total no of jobs reviewed per hour** and group it by dates.

A decorative graphic in the background consisting of two large, light blue arrows pointing in opposite directions (one right, one left) and a large, light blue gear-like shape at the bottom.

# TASK B

We are required to calculate the 7-day rolling average of throughput (number of events per second). We need to write an SQL query to calculate the 7-day rolling average of throughput.

## QUERY

```
1 SELECT ds,  
2 SUM(event_persecond) OVER(ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW)  
3 as 7dayRolling_average  
4 FROM (SELECT ds, COUNT(event)/SUM(time_spent) as event_persecond  
5 FROM job_data  
6 GROUP BY ds  
7 ORDER BY ds) through_put;
```

## OUTPUT

ds	7dayRolling_average
11/25/2020	0.0222
11/26/2020	0.0401
11/27/2020	0.0497
11/28/2020	0.1103
11/29/2020	0.1603
11/30/2020	0.2103

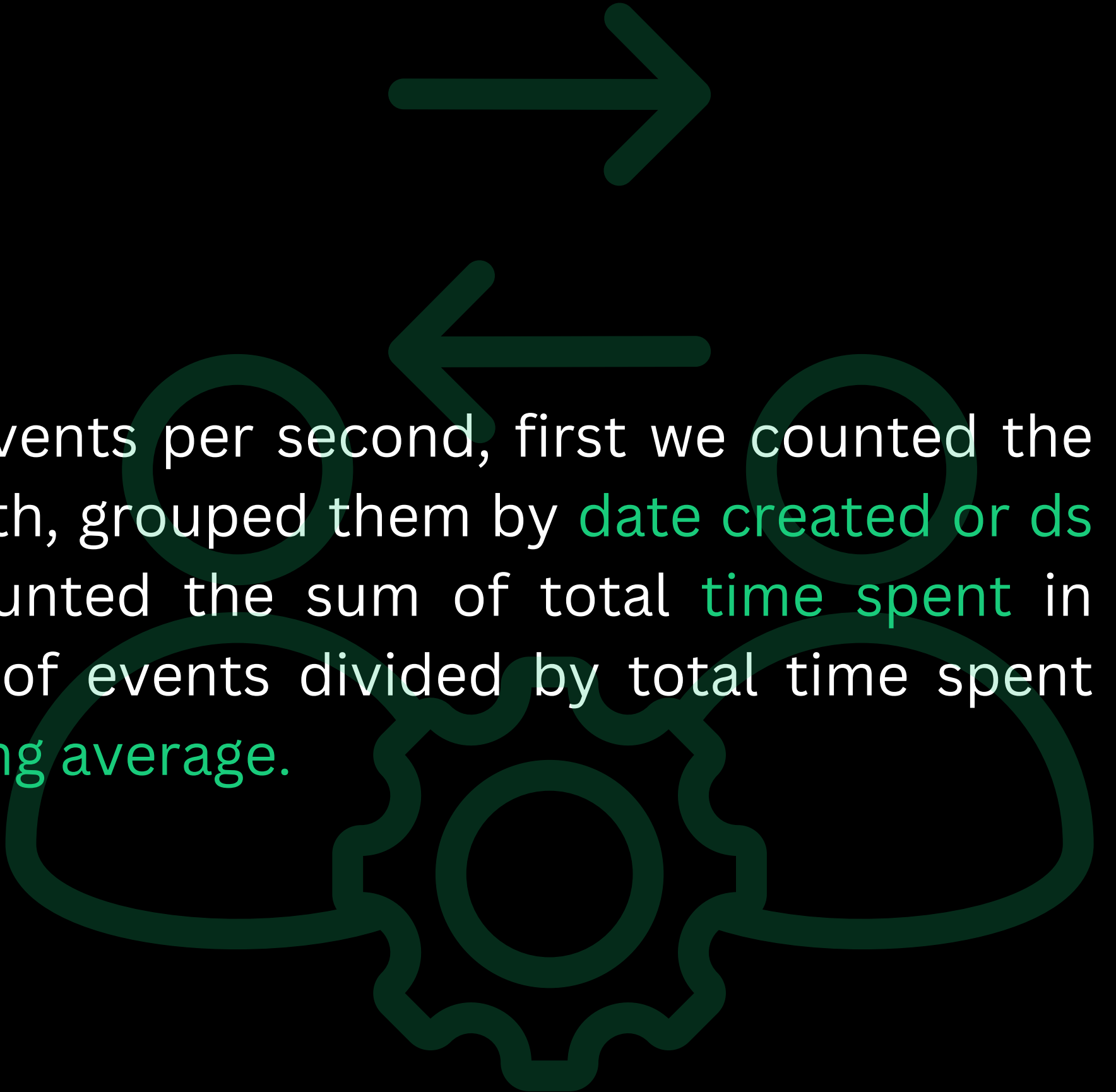


# TASK B

---

## APPROACH

As we were needed to count no of events per second, first we counted the **number of events** per day in the month, grouped them by **date created or ds** as given in the table and finally counted the sum of total **time spent** in seconds and then evaluated the no of events divided by total time spent grouped by date to get the **7-day rolling average**.

A decorative background graphic on the right side of the slide. It features a large, dark green gear with a circular center. Two arrows, also in dark green, are positioned above the gear: one pointing to the right and one pointing to the left, as if they are interacting with or moving the gear.

# TASK C

We are required to calculate the **percentage share** of each language in the last 30 days.

## QUERY

```
1 SELECT
2 language,
3 (COUNT(language) * 100) / (SELECT COUNT(job_id) FROM job_data) AS language_percentage
4 FROM
5 job_data
6 GROUP BY
7 language;
```

## OUTPUT

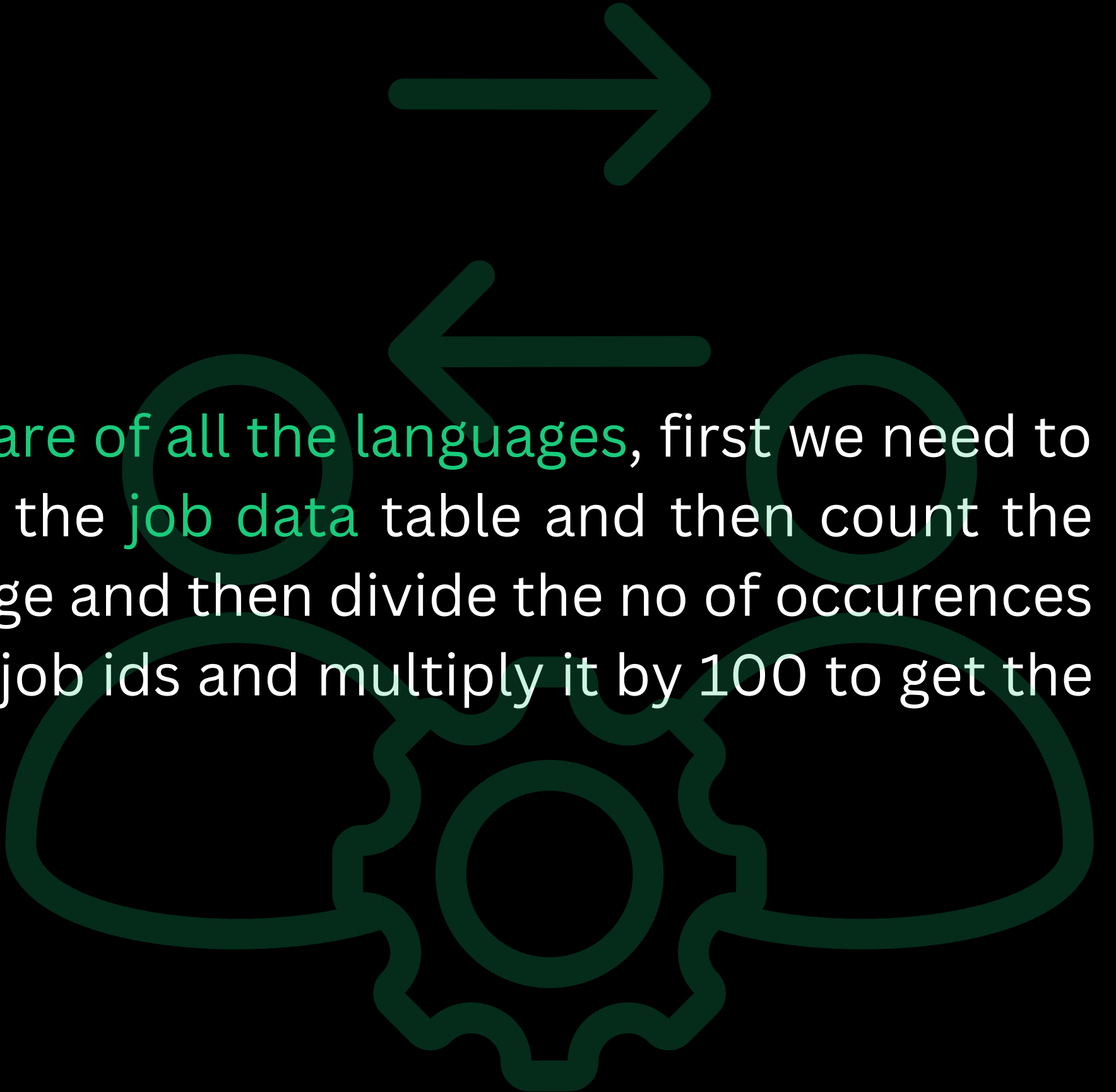
language	language_percentage
Arabic	12.5000
English	12.5000
French	12.5000
Hindi	12.5000
Italian	12.5000
Persian	37.5000

# TASK C

---

## APPROACH

As we were needed the **percentage share of all the languages**, first we need to count the total number of rows from the **job data** table and then count the number of **entries** taking which language and then divide the no of occurrences of a particular language by total no of job ids and multiply it by 100 to get the **percentage share**.

A decorative graphic in the background consisting of two large, light blue arrows pointing in opposite directions (one right, one left) and a large, light blue gear-like shape at the bottom.

# TASK D

---

We are required to Identify **duplicate rows** in the data.

## QUERY

```
1 select job_id, count(job_id) as job_id_count
2 from job_data
3 group by job_id
4 HAVING job_id_count > 1 ;
```

## OUTPUT

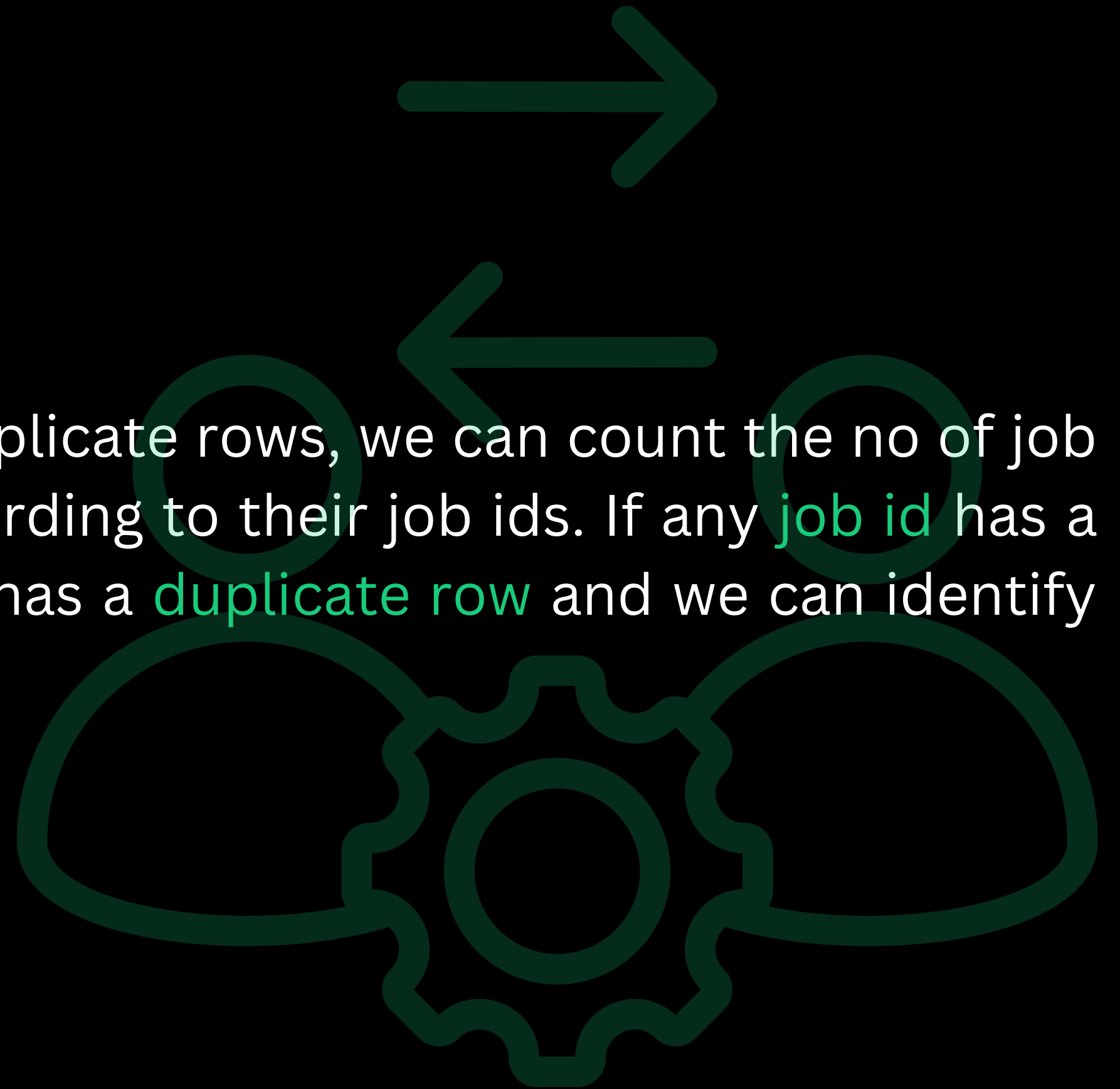
job_id	job_id_count
23	3

# TASK D

---

## APPROACH

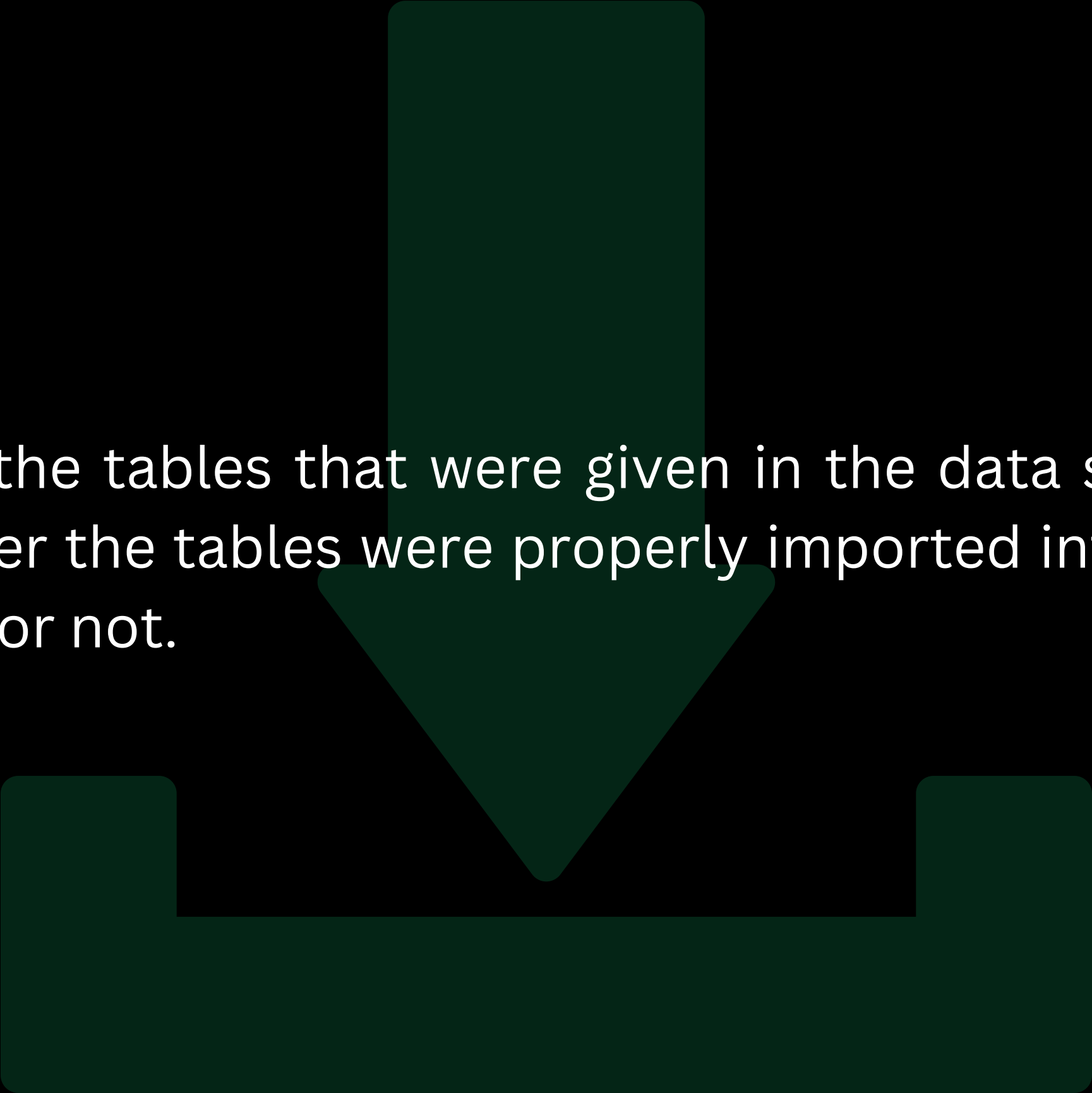
As we were needed to identify the duplicate rows, we can count the no of job id rows present and group them according to their job ids. If any **job id** has a count greater than 1, thus the job id has a **duplicate row** and we can identify the duplicate rows easily.



Case Study 2

# INVESTIGATING METRIC SPIKE





First we **imported** the tables that were given in the data set of the question and verified whether the tables were properly imported into the database we desired to work on or not.

# TASK A

We are required to measure the **activeness** of users on a weekly basis.

## QUERY

```
1 SELECT
2 (SELECT COUNT(DISTINCT(user_id)) AS active_users
3 FROM users
4 WHERE state = 'active') / (SELECT COUNT(DISTINCT(user_id)) AS active_users
5 FROM users) * 100 as user_engagement_rate
6 FROM users;
```

## OUTPUT

user_engagement_rate
49.2

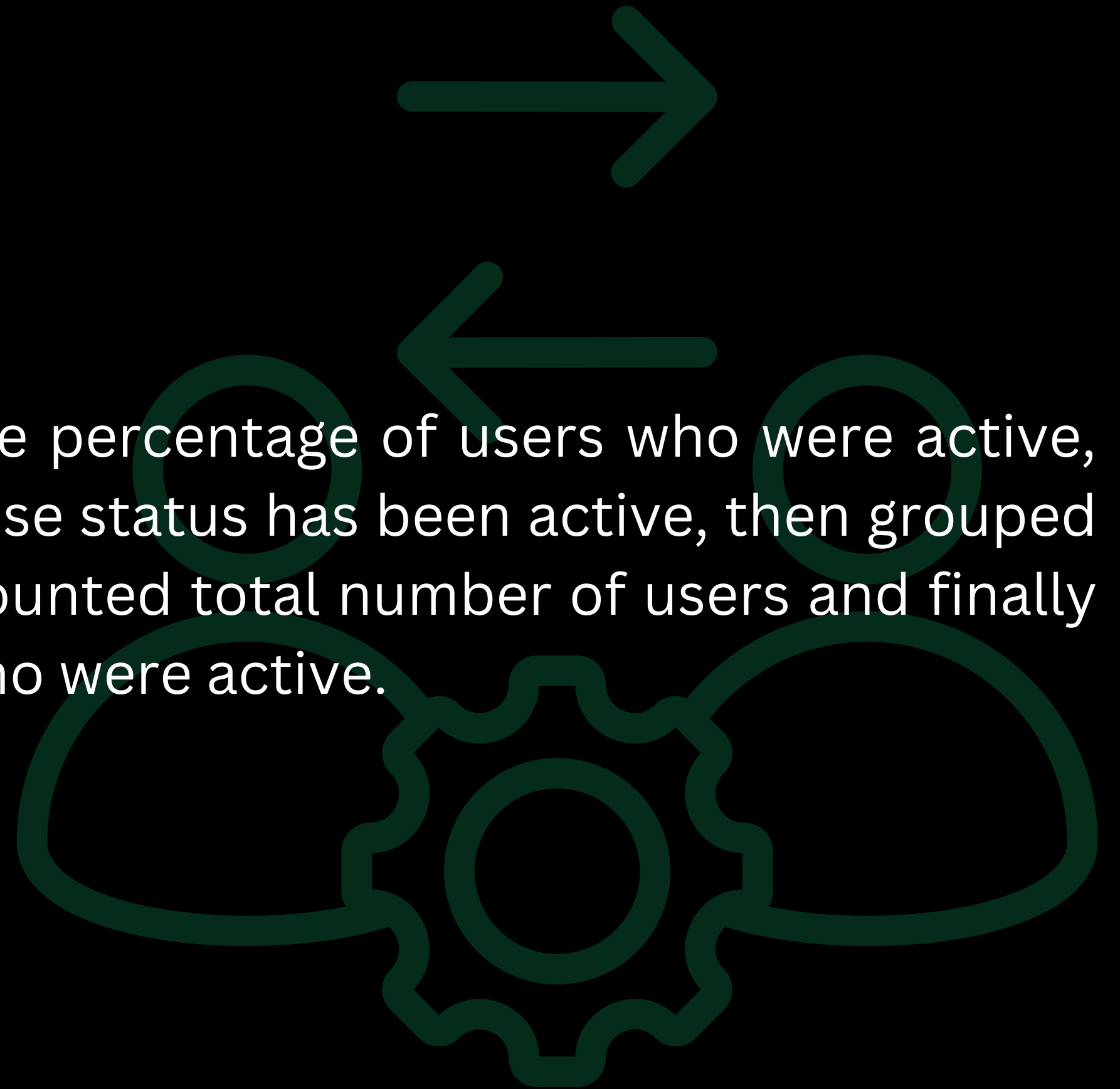


# TASK A

---

## APPROACH

As we were required to calculate the percentage of users who were active, first we extracted the no of users whose status has been active, then grouped them by selecting distinct user ids, counted total number of users and finally extracted the percentage of users who were active.



# TASK B

We are required to analyze the growth of users over time for a product.

## QUERY

```
1 SELECT
2 new_users.device,
3 avg(new_users.new_users / first_month.total_users*100) AS growth_rate
4 FROM (SELECT device,
5 month(occurred_at) AS signup_month,
6 COUNT(DISTINCT user_id) AS new_users FROM events
7 GROUP BY device, signup_month ) AS new_users
8 JOIN (SELECT device,
9 COUNT(DISTINCT user_id) AS total_users
10 FROM events
11 GROUP BY device) AS first_month
12 ON new_users.device = first_month.device
13 group by new_users.device;
```

## OUTPUT

device	growth_rate
acer aspire desktop	36.49
acer aspire notebook	35.58
amazon fire phone	32.2
asus chromebook	34.89
dell inspiron desktop	35.56
dell inspiron notebook	36.38
hp pavilion desktop	35.3
htc one	33.37
ipad air	33.14
ipad mini	32.34
iphone 4s	34.7
iphone 5	35.11
iphone 5s	33.99

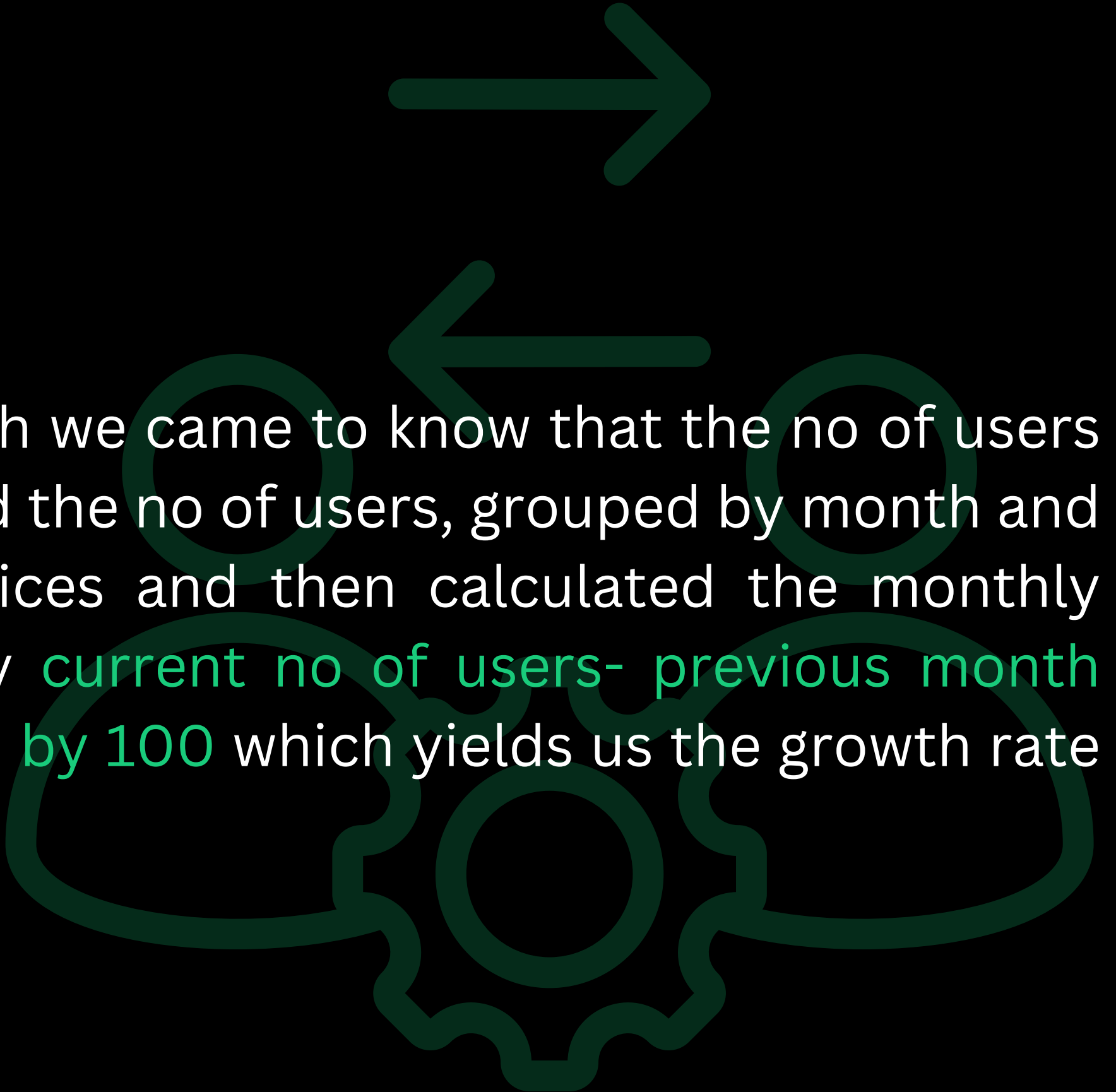
device	growth_rate
kindle fire	32.64
lenovo thinkpad	36.15
mac mini	39.41
macbook air	35.63
macbook pro	35.86
nexus 10	33
nexus 5	34.28
nexus 7	32.81
nokia lumia 635	36.25
samsung galaxy tablet	31.85
samsung galaxy note	33.67
samsung galaxy s4	33.64
windows surface	32.35

# TASK B

---

## APPROACH

First we analysed the table after which we came to know that the no of users changed over time. Thus, we extracted the no of users, grouped by month and year and grouped them by the devices and then calculated the monthly growth rate of users for a device by  $\frac{\text{current no of users} - \text{previous month users}}{\text{current month users}} \times 100$  which yields us the growth rate of users for a particular device.



# TASK C

We are required to analyze the retention of users on a weekly basis after signing up for a product.

## QUERY

```
1 SELECT
2 WEEK(occurred_at) AS week,
3 COUNT(DISTINCT user_id) AS signups,
4 COUNT(DISTINCT CASE WHEN event_type = 'engagement' THEN user_id END) AS engagements,
5 COUNT(DISTINCT CASE WHEN event_type = 'engagement' THEN user_id END) / COUNT(DISTINCT user_id) *100 AS retention_rate
6 FROM events
7 GROUP BY WEEK(occurred_at)
8 ORDER BY week;
```

## OUTPUT

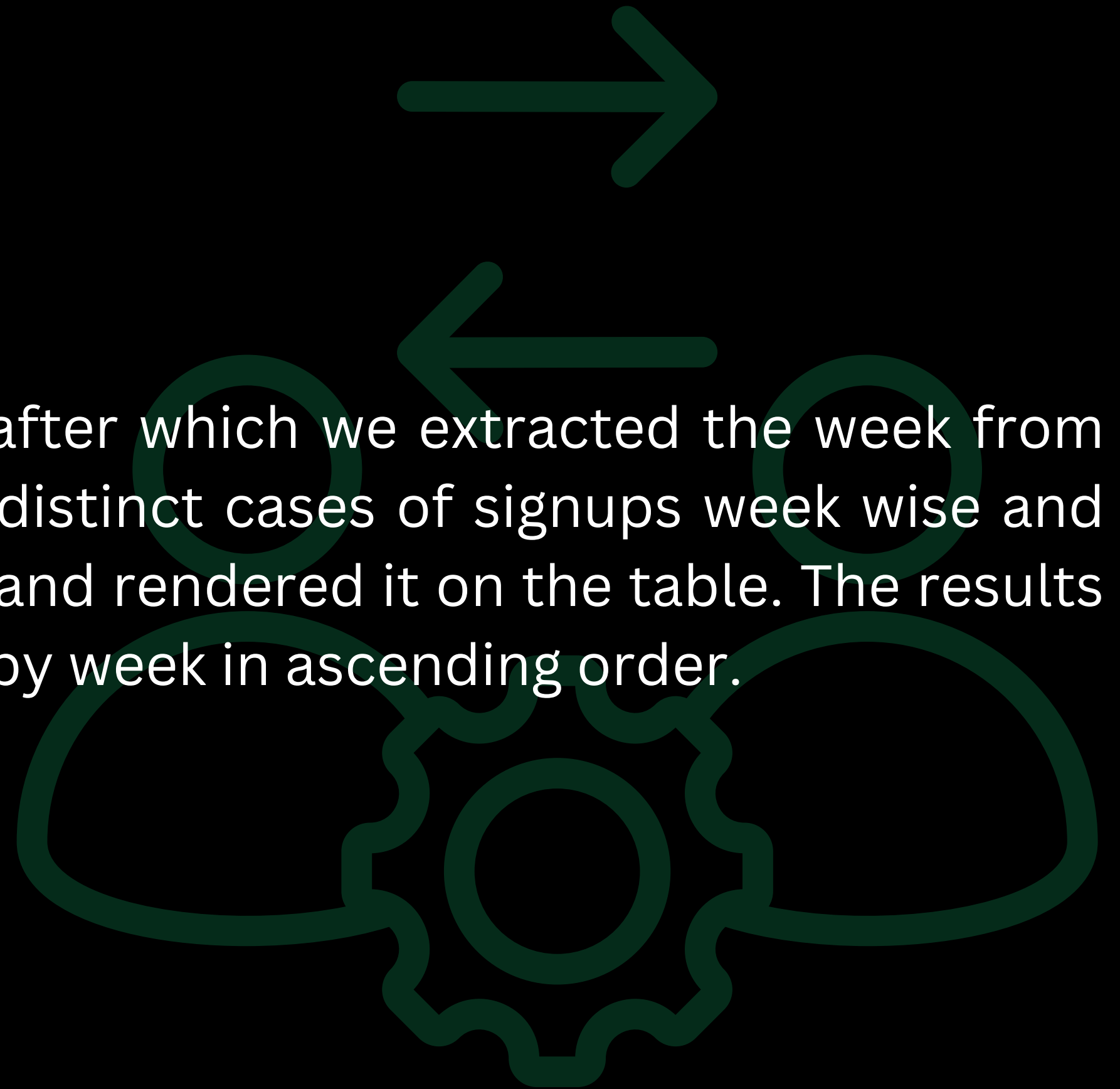
week	signups	engagements	retention_rate
17	740	663	89.5946
18	1260	1068	84.7619
19	1287	1113	86.4802
20	1351	1154	85.4182
21	1299	1121	86.2972
22	1381	1186	85.8798
23	1446	1232	85.2006
24	1471	1275	86.6757
25	1459	1264	86.6347
26	1509	1302	86.2823
27	1573	1372	87.2219
28	1577	1365	86.5568
29	1607	1376	85.6254
30	1706	1467	85.9906
31	1514	1299	85.7992
32	1454	1225	84.2503
33	1438	1225	85.1878
34	1443	1204	83.4373
35	118	104	88.1356

# TASK C

---

## APPROACH

First we analysed the events table , after which we extracted the week from the date field and then counted the distinct cases of signups week wise and finally calculated the retention rate and rendered it on the table. The results were grouped weekwise and ordered by week in ascending order.



# TASK D

We are required to measure the **activeness of users** on a weekly basis per device.

## QUERY

```
1 SELECT
2 WEEK(occurred_at) AS week,
3 COUNT(DISTINCT CASE
4 WHEN event_type = 'engagement' THEN user_id END) /
5 COUNT(DISTINCT user_id)/(select count(distinct(device)) from events) * 100
6 AS engagement_rate_per_device
7 FROM events
8 GROUP BY WEEK(occurred_at)
9 ORDER BY week;
```

## OUTPUT

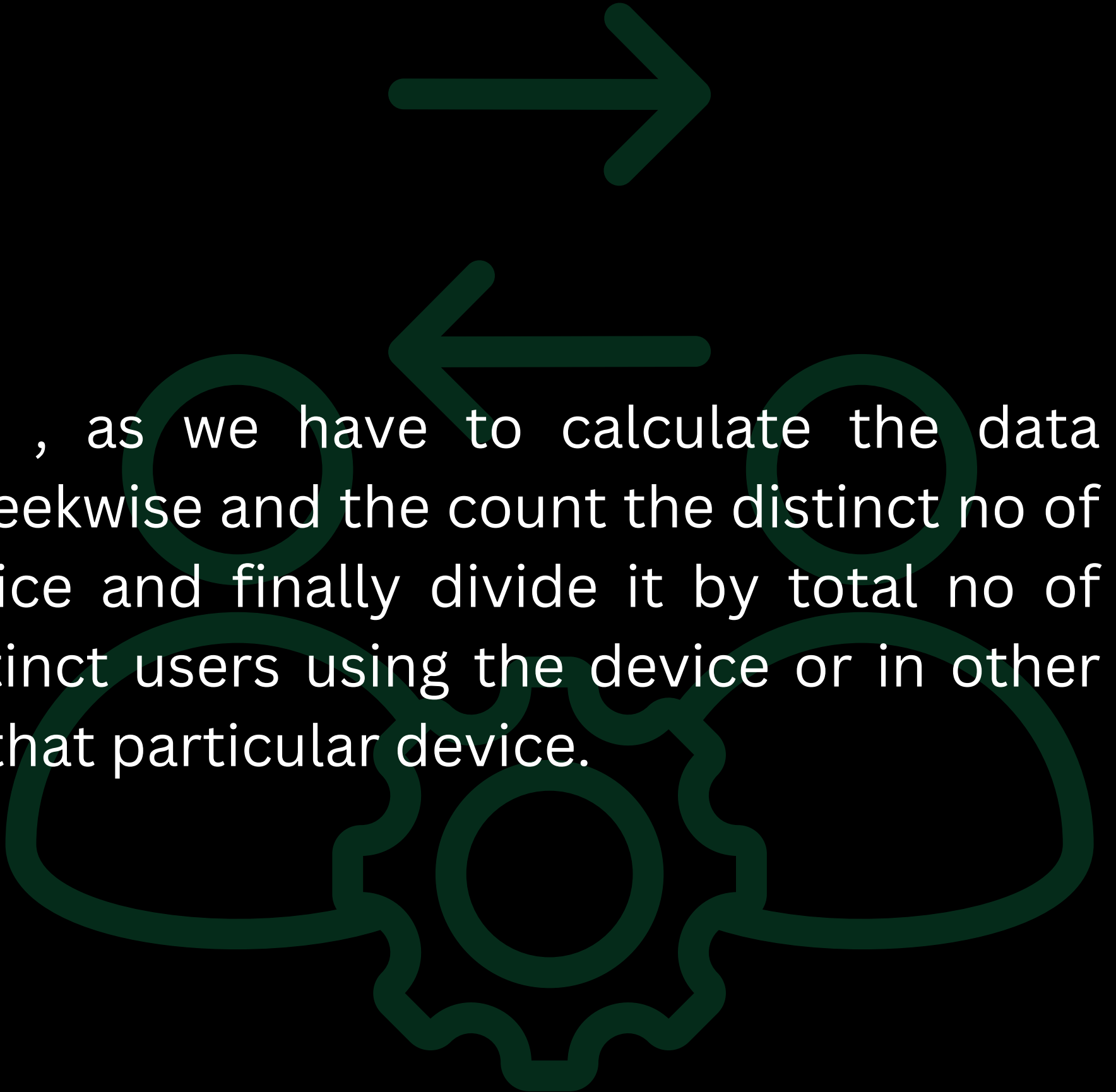
week	engagement_rate_per_device
17	3.44594594
18	3.26007326
19	3.32616102
20	3.28531572
21	3.31912122
22	3.30306912
23	3.27694435
24	3.33368195
25	3.33210312
26	3.31855023
27	3.35468727
28	3.3291059
29	3.29328419
30	3.30733159
31	3.29996951
32	3.24039784
33	3.27645233
34	3.20912628
35	3.38983051

# TASK D

---

## APPROACH

First we analysed the events table , as we have to calculate the data weewekwise, we have to extract date weekwise and the count the distinct no of users using a particular type of device and finally divide it by total no of distinct devices to get the no of distinct users using the device or in other words we are measuring the utility of that particular device.

A decorative graphic in the bottom right corner featuring two interlocking gears. A large arrow points from the top right towards the center, and another arrow points from the center towards the left. The entire graphic is rendered in a dark teal color.

# TASK E

We are required to analyze how users are engaging with the **email service**.

## QUERY

```
1 SELECT
2 engagements,
3 total_users,
4 engagements/total_users*100 AS engagement_rate
5 from ( SELECT
6 count(distinct(user_id)) AS total_users,
7 COUNT(DISTINCT CASE
8 WHEN action = 'email_open' THEN user_id
9 WHEN action = 'email_clickthrough' THEN user_id END) AS engagements
10 FROM email_events) as counts;
```

## OUTPUT

engagements	total_users	engagement_rate
5927	6179	95.9217



# TASK E

---

## APPROACH

First we analysed the email events table and then counted the total users and counted no of users according to their mail action specifically when mail action is email open or email clickthrough. Thus we stored this data into a temporary field engagements and then divided it by total users and multiplied the result with 100 to get the engagement rate of users with email.

# RESULT:

I successfully handled a practical situation and learned to run queries and provide insights for market analysis or investors metrics. Also, I was able to learn a lot of new queries and also learned to relate tables, join them and get desired work to be done. I also took help from my seniors and some of the colleagues who have already taken this course as I am always in the mode of learning from each and every source I get in.

Overall, I feel like I have mastered the basics which feels not more than an actual data scientist.

