Operation Analytics and Investigating Metric Spike

Project No. 2 for
Trainity DA Training

Prepared By:
Nilesh Kulkarni

Agenda

- ➢ Project Description
- ➢ Project Approach
- ➢ Tech Stack Used
- ➢ Insights
- ➢ Results

By: Nilesh Kulkarni

Project Description

This project is about operational analytics where we will analyse provided operational data and derive valuable business Insights for various business teams like operations, support, and marketing. The project consists of 2 case studies – each consists of few insights to be derived as listed below.

1. **Case Study-1 Job Data Analysis:** In this case study we have operational jobs data like job id, review time, language, actor, status/action, date, organisation of actor etc. We need to derive following insights:
   A. Jobs Reviewed Over Time: Number of jobs reviewed per hour per day
   B. Throughput Analysis: Daily/weekly Throughput & 7-day rolling average of throughput
   C. Language Share Analysis: Percentage share of each language over last 30 days
   D. Duplicate Rows Detection: Identify if there is any duplicate row in given jobs data
2. **Case Study-2 Investigating Metric Spike:** in this case study we have data about users and related events. We need to derive following insights:
   A. Weekly User Engagement: Derive weekly stats on user engagement
   B. User Growth Analysis: Calculate user growth over the period of time
   C. Weekly Retention Analysis:Calculate the weekly retention of users based on their sign-up cohort.
   D. Weekly Engagement Per Device: Device wise weekly engagement for all devices used
   E. Email Engagement Analysis: Calculate Email engagement metrics

Project Approach

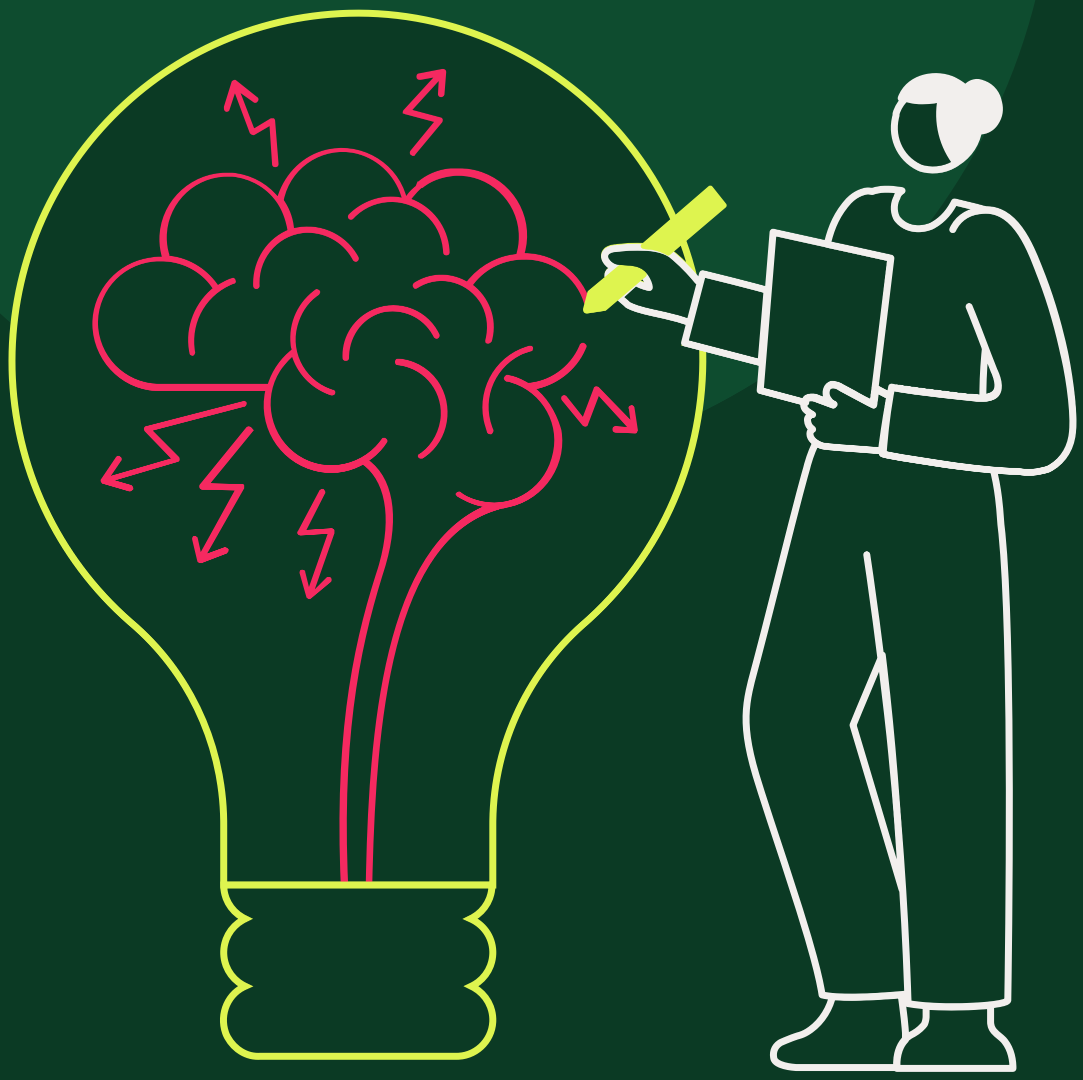
- High level steps for the Project approach are as outlined below:
  - **Database & Tables creation:** Create new database in MySQL. Write & execute DDL statement to create new tables using provided table specs. Verify tables match the specs.
  - **Data Loading:** Load the provided data (CSV) into database using DML statements (SQL) via MySQL Command Line interface.
  - **Analysis:** Analyse each insights requirement in detail and prepare SQL queries. Select optimal and efficient SQL queries/approach.
  - **Extract insights:** Run the SQL queries to extract new insights as required
  - **Review:** Review and cross check SQL output to verify it matches with the requirements
  - **Document:** Document the insights and results to be shared across business teams

By: Nilesh Kulkarni

Tech Stack Used

- Operating System: **Microsoft Windows** 11 Version 22H2
- **MS Excel -** The input data is provided in CSV files (excel) that is to be loaded in tables.
- **MySQL Workbench** – This is user friendly interface to administrate, manage and query MySQL database, This is used in analysis to run SQL queries.
- **MySQL Command Line interface** - This is very good option used to load data in MySQL tables quickly when data is large. Used for loading data in tables.
- **MS PowerPoint** (MS Office 365) – documentation of insights and results.
- **Acrobat Reader (PDF) -** documentation and sharing results.

Insights



Case Study 1:
Insights A. Job review Rate

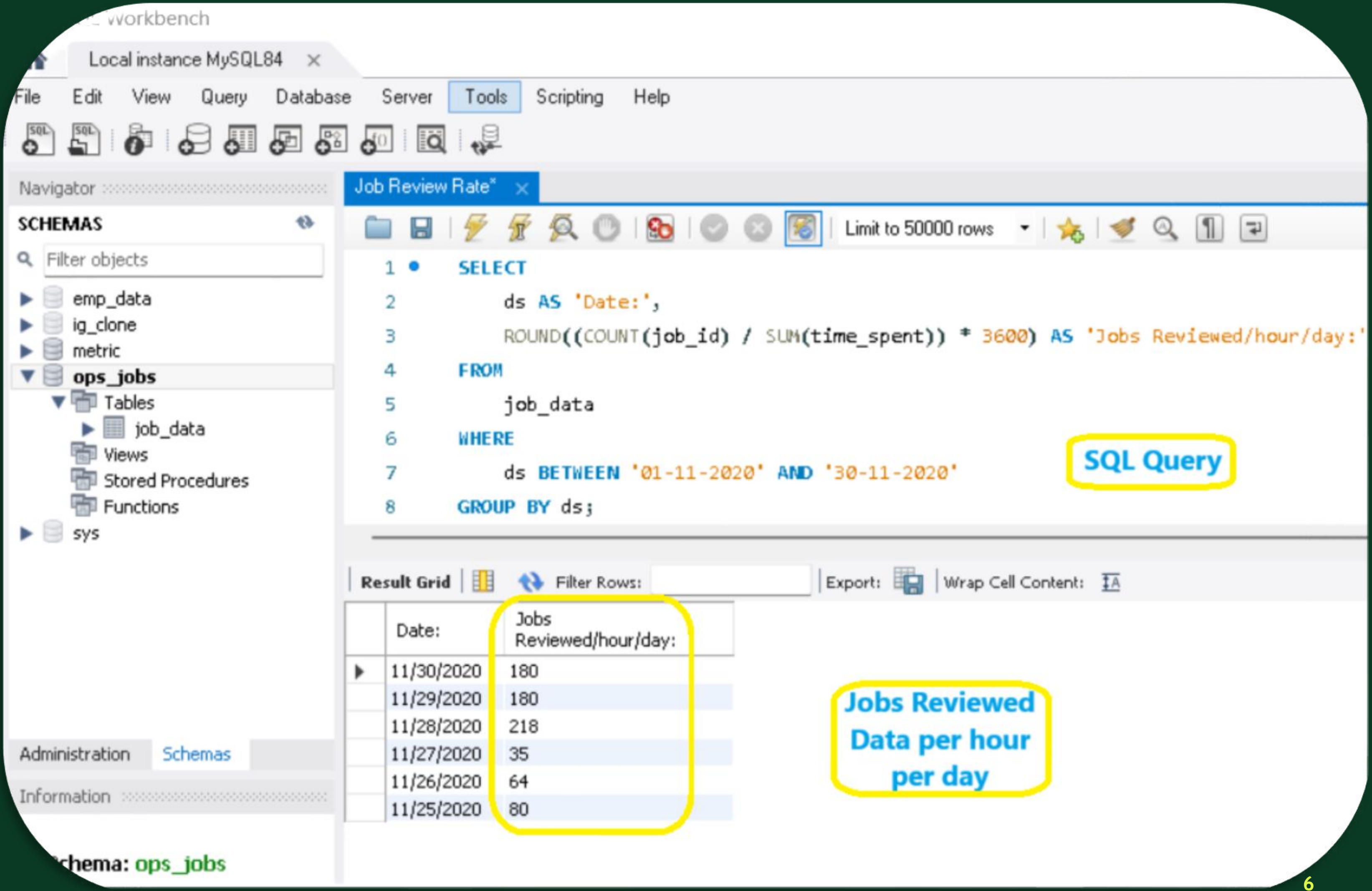
Jobs Reviewed Over Time:

- Objective: Calculate the number of jobs reviewed per hour for each day in November 2020.
- Task: Write an SQL query to calculate the number of jobs reviewed per hour for each day in November 2020.

Insights:

- Jobs review rate per hour per day vary from lowest as 35 to highest as 218.
- Investigate for root cause for low jobs review rate on given dates to see any technical or process related issues.

SQL Query and Output showing Job Review Rate



```
SELECT
    ds AS 'Date:',
    ROUND((COUNT(job_id) / SUM(time_spent)) * 3600) AS 'Jobs Reviewed/hour/day:'
FROM
    job_data
WHERE
    ds BETWEEN '01-11-2020' AND '30-11-2020'
GROUP BY ds;
```

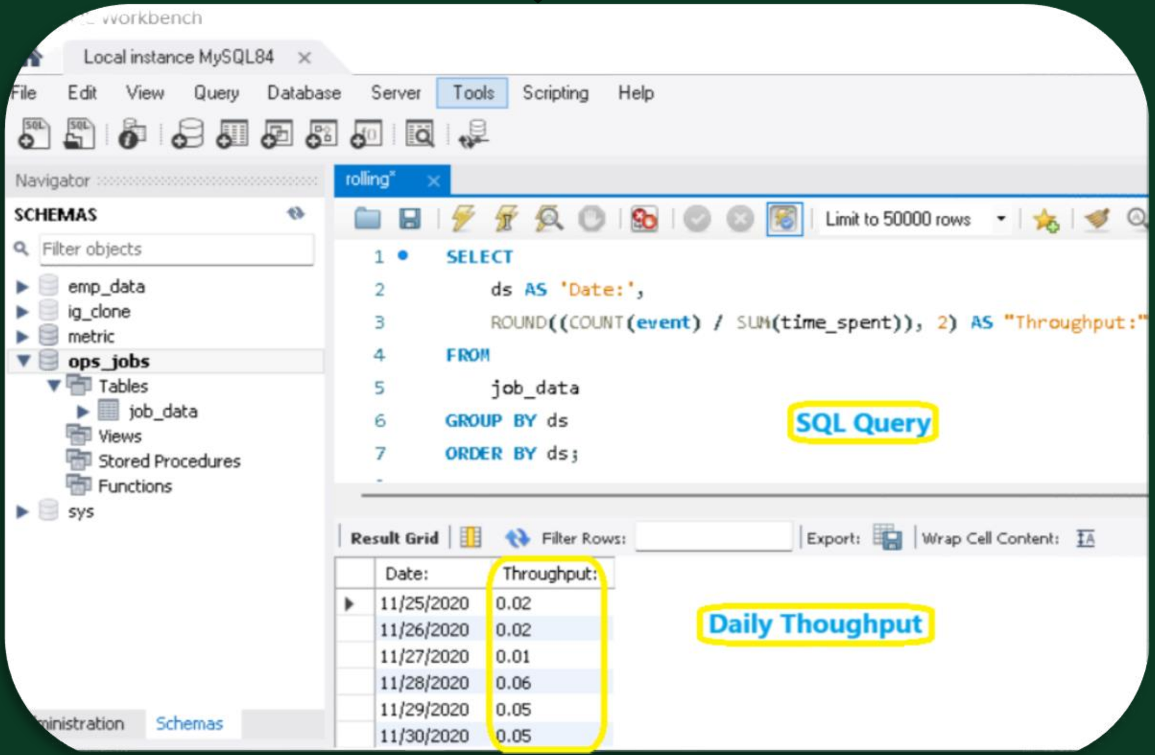
Date:	Jobs Reviewed/hour/day:
11/30/2020	180
11/29/2020	180
11/28/2020	218
11/27/2020	35
11/26/2020	64
11/25/2020	80

By: Nilesh Kulkarni

Case Study 1: Insights B. Throughput Analysis

- Objective: Calculate the 7-day rolling average of throughput (number of events per second).
- Task: Write an SQL query to calculate the 7-day rolling average of throughput. Additionally, explain whether you prefer using the daily metric or the 7-day rolling average for throughput, and why.

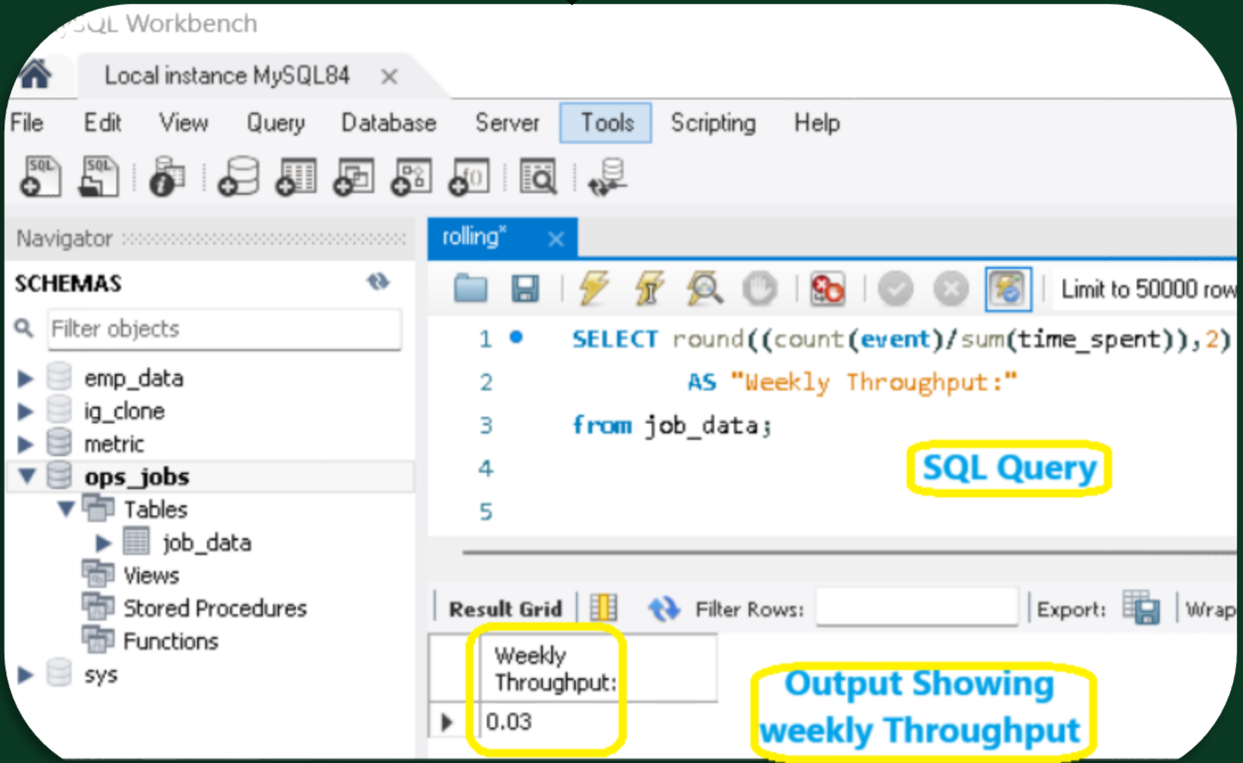
SQL Query and Output showing Daily Throughput



```
SELECT
    ds AS 'Date:',
    ROUND((COUNT(event) / SUM(time_spent)), 2) AS "Throughput:"
FROM
    job_data
GROUP BY ds
ORDER BY ds;
```

Date:	Throughput:
11/25/2020	0.02
11/26/2020	0.02
11/27/2020	0.01
11/28/2020	0.06
11/29/2020	0.05
11/30/2020	0.05

SQL Query and Output showing Weekly Throughput



```
SELECT round((count(event)/sum(time_spent)),2)
        AS "Weekly Throughput:"
from job_data;
```

Weekly Throughput:
0.03

By: Nilesh Kulkarni

(Insight B Continued..)

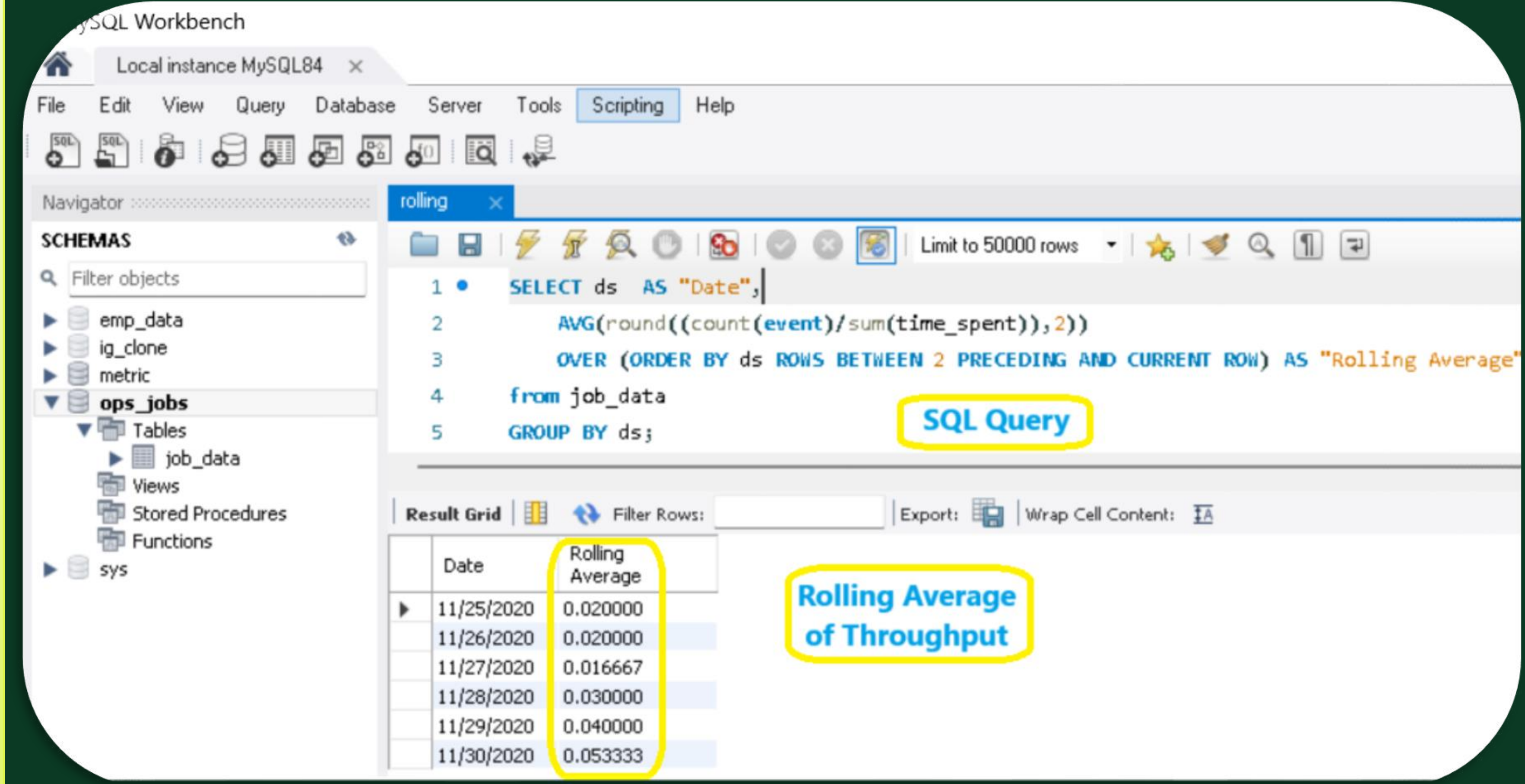
Case Study 1:
Insights B. Throughput Analysis
(..continued)

SQL Query and Output showing Rolling Average of Throughput

Insight
Daily Throughput vary largely between 0.01 to 0.06 over the week. Investigate root cause for such variation.

Daily Metric & Rolling Average:

Owing to variations, daily metrics can vary with ups and downs, so rolling average gives more clear & indicative picture of long term patterns/trends. so it is preferred to use rolling average compared to daily metrics to spot trends and it helps with informed decision making as well as correct representation of process performance over the period of time.



```sql
SELECT ds  AS "Date",
    AVG(round((count(event)/sum(time_spent)),2))
    OVER (ORDER BY ds ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS "Rolling Average"
from job_data
GROUP BY ds;
```

Date	Rolling Average
11/25/2020	0.020000
11/26/2020	0.020000
11/27/2020	0.016667
11/28/2020	0.030000
11/29/2020	0.040000
11/30/2020	0.053333

By: Nilesh Kulkarni

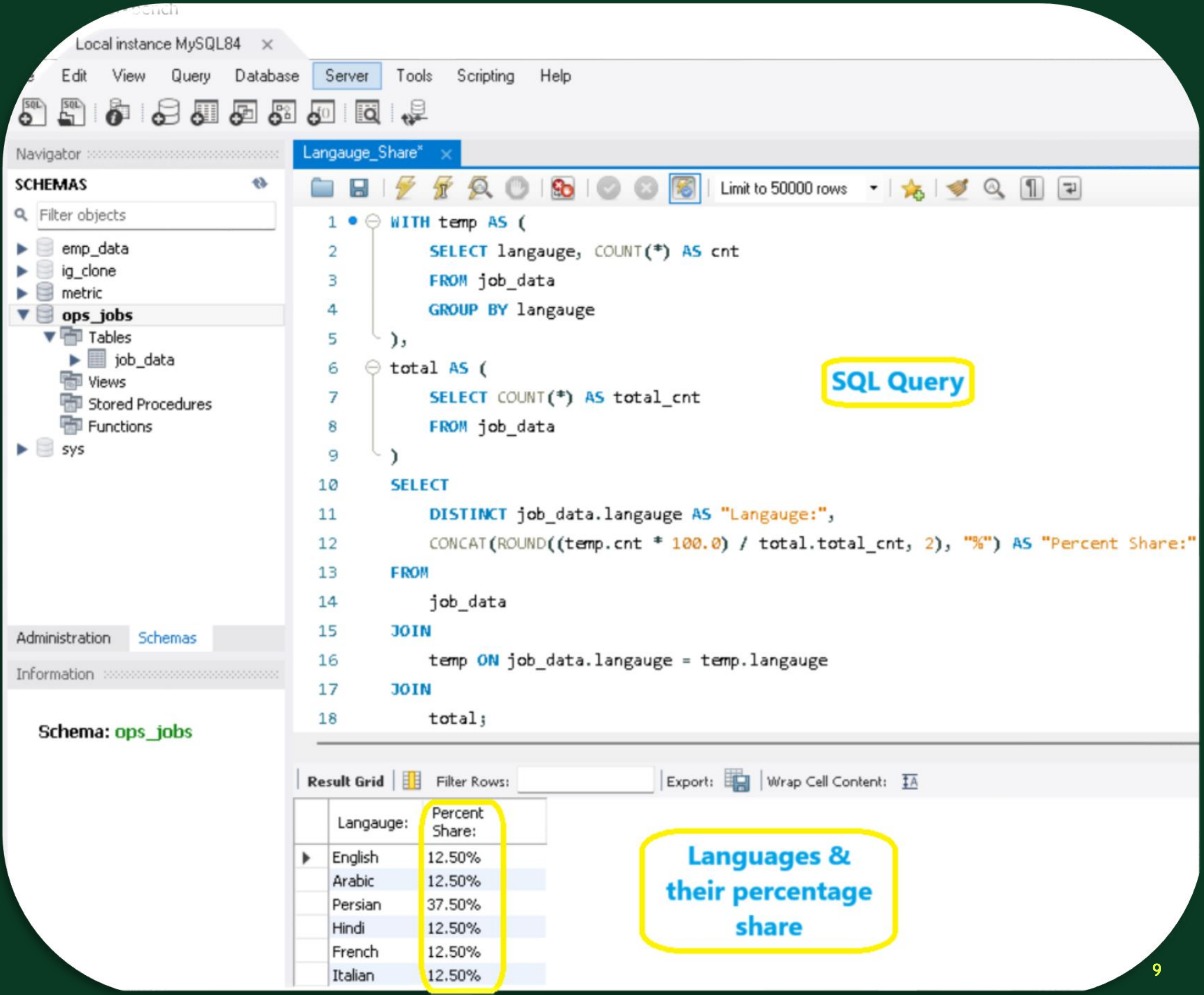
Case Study 1: Insights C. Language Share Analysis

- ➢ Objective: Calculate the percentage share of each language in the last 30 days.
- ➢ Task: Write an SQL query to calculate the percentage share of each language over the last 30 days.

SQL Query and Output showing percentage share of all languages

Insights:

- ➢ Language % share is evenly distributed across except 'Persian' language which has highest share of 37.5%
- ➢ Target to increase share for languages with low share (12.5%) using specific content and user preferences



```sql
WITH temp AS (
    SELECT langauge, COUNT(*) AS cnt
    FROM job_data
    GROUP BY langauge
),
total AS (
    SELECT COUNT(*) AS total_cnt
    FROM job_data
)
SELECT
    DISTINCT job_data.langauge AS "Langauge:",
    CONCAT(ROUND((temp.cnt * 100.0) / total.total_cnt, 2), "%") AS "Percent Share:"
FROM
    job_data
JOIN
    temp ON job_data.langauge = temp.langauge
JOIN
    total;
```

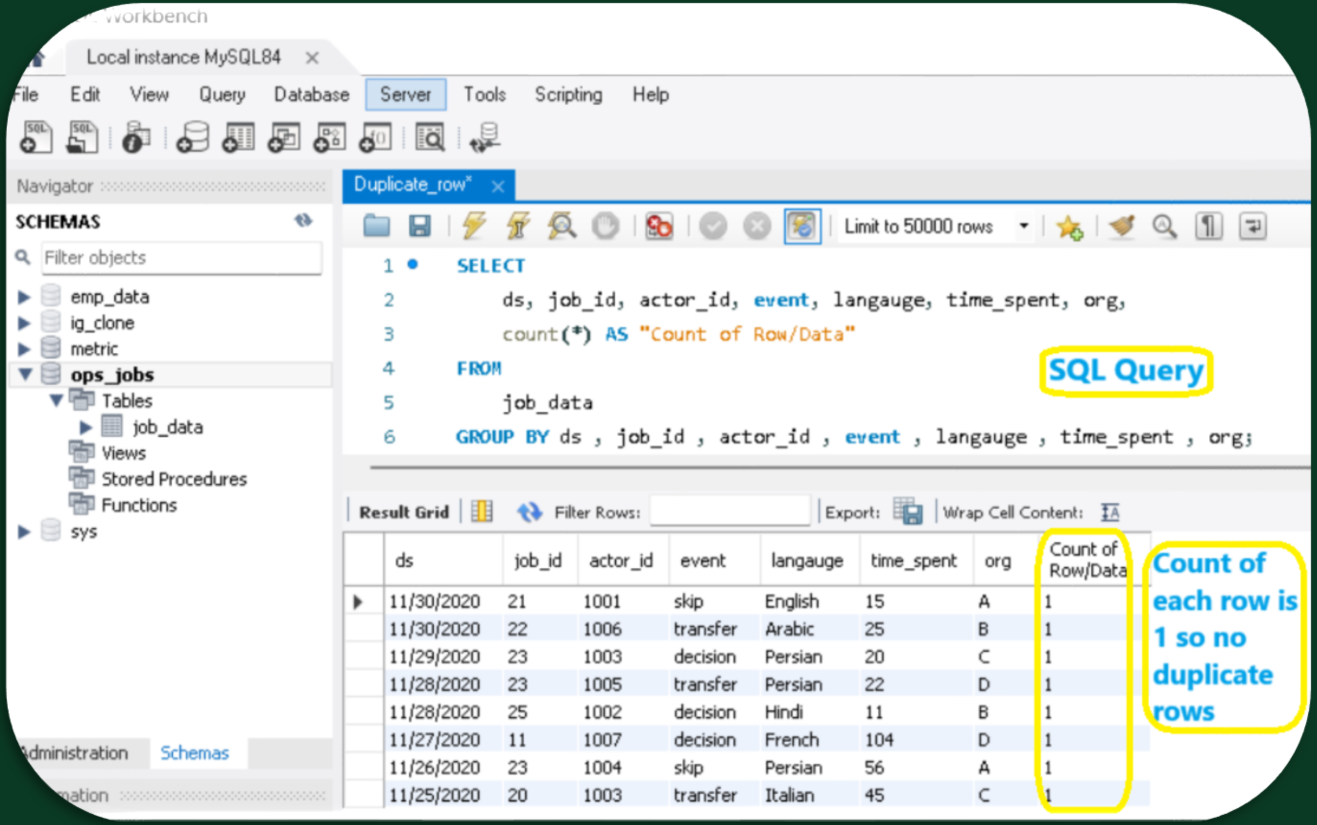
Langauge:	Percent Share:
English	12.50%
Arabic	12.50%
Persian	37.50%
Hindi	12.50%
French	12.50%
Italian	12.50%

By: Nilesh Kulkarni

Case Study 1: Insights D. Duplicate Rows Detection

- Objective: Identify duplicate rows in the data.
- Task: Write an SQL query to display duplicate rows from the job_data table.

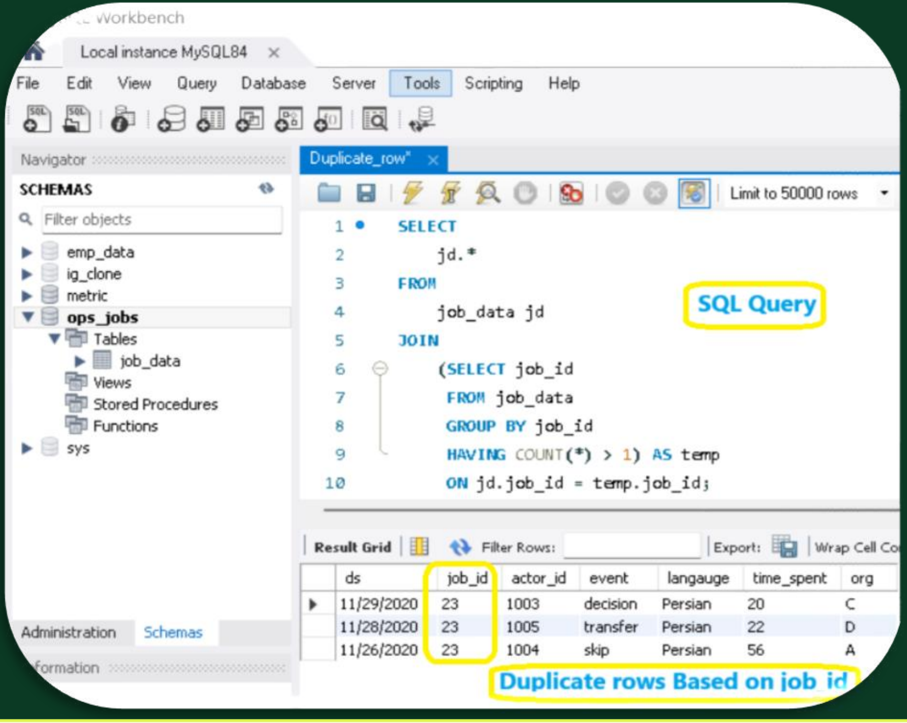
SQL Query and Output showing NO Duplicate rows based on all columns



```
SELECT
    ds, job_id, actor_id, event, langauge, time_spent, org,
    count(*) AS "Count of Row/Data"
FROM
    job_data
GROUP BY ds , job_id , actor_id , event , langauge , time_spent , org;
```

ds	job_id	actor_id	event	langauge	time_spent	org	Count of Row/Data
11/30/2020	21	1001	skip	English	15	A	1
11/30/2020	22	1006	transfer	Arabic	25	B	1
11/29/2020	23	1003	decision	Persian	20	C	1
11/28/2020	23	1005	transfer	Persian	22	D	1
11/28/2020	25	1002	decision	Hindi	11	B	1
11/27/2020	11	1007	decision	French	104	D	1
11/26/2020	23	1004	skip	Persian	56	A	1
11/25/2020	20	1003	transfer	Italian	45	C	1

SQL Query and Output showing 3 Duplicate rows based on the "job_id" column



```
SELECT
    jd.*
FROM
    job_data jd
JOIN
    (SELECT job_id
     FROM job_data
     GROUP BY job_id
     HAVING COUNT(*) > 1) AS temp
     ON jd.job_id = temp.job_id;
```

ds	job_id	actor_id	event	langauge	time_spent	org
11/29/2020	23	1003	decision	Persian	20	C
11/28/2020	23	1005	transfer	Persian	22	D
11/26/2020	23	1004	skip	Persian	56	A

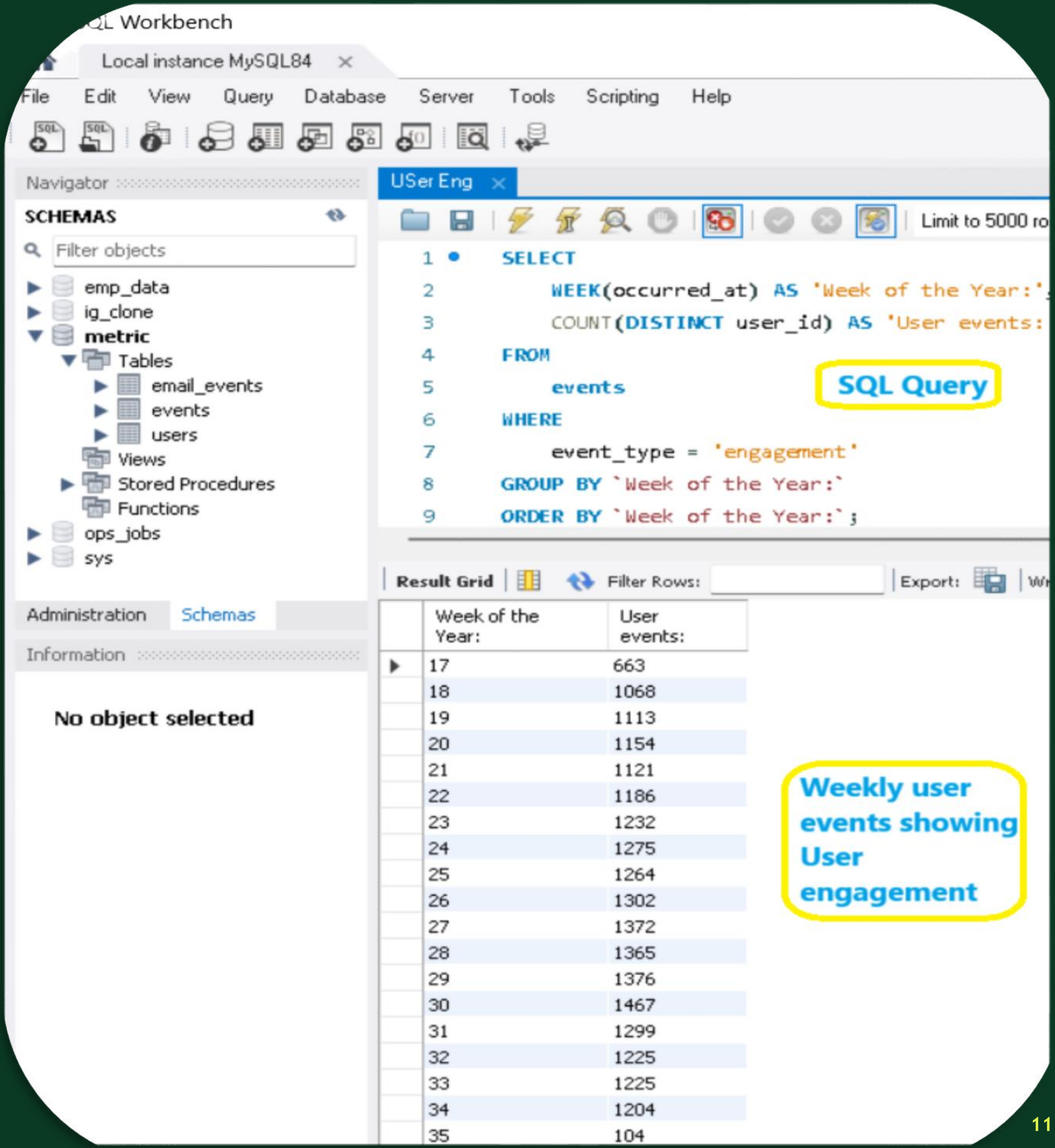
Insights:

- Duplicate row exists with same job_id. If it is invalidating business rules, ensure to add data validations to prevent duplicate rows

Case Study 2: Insights A. Weekly User Engagement

- ➢ Objective: Measure the activeness of users on a weekly basis.
- ➢ Task: Write an SQL query to calculate the weekly user engagement.

SQL Query and Output showing User Engagement on the weekly basis



```
SELECT
    WEEK(occurred_at) AS 'Week of the Year:'
    COUNT(DISTINCT user_id) AS 'User events:
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY `Week of the Year:`
ORDER BY `Week of the Year:`;
```

Week of the Year:	User events:
17	663
18	1068
19	1113
20	1154
21	1121
22	1186
23	1232
24	1275
25	1264
26	1302
27	1372
28	1365
29	1376
30	1467
31	1299
32	1225
33	1225
34	1204
35	104

Insights:

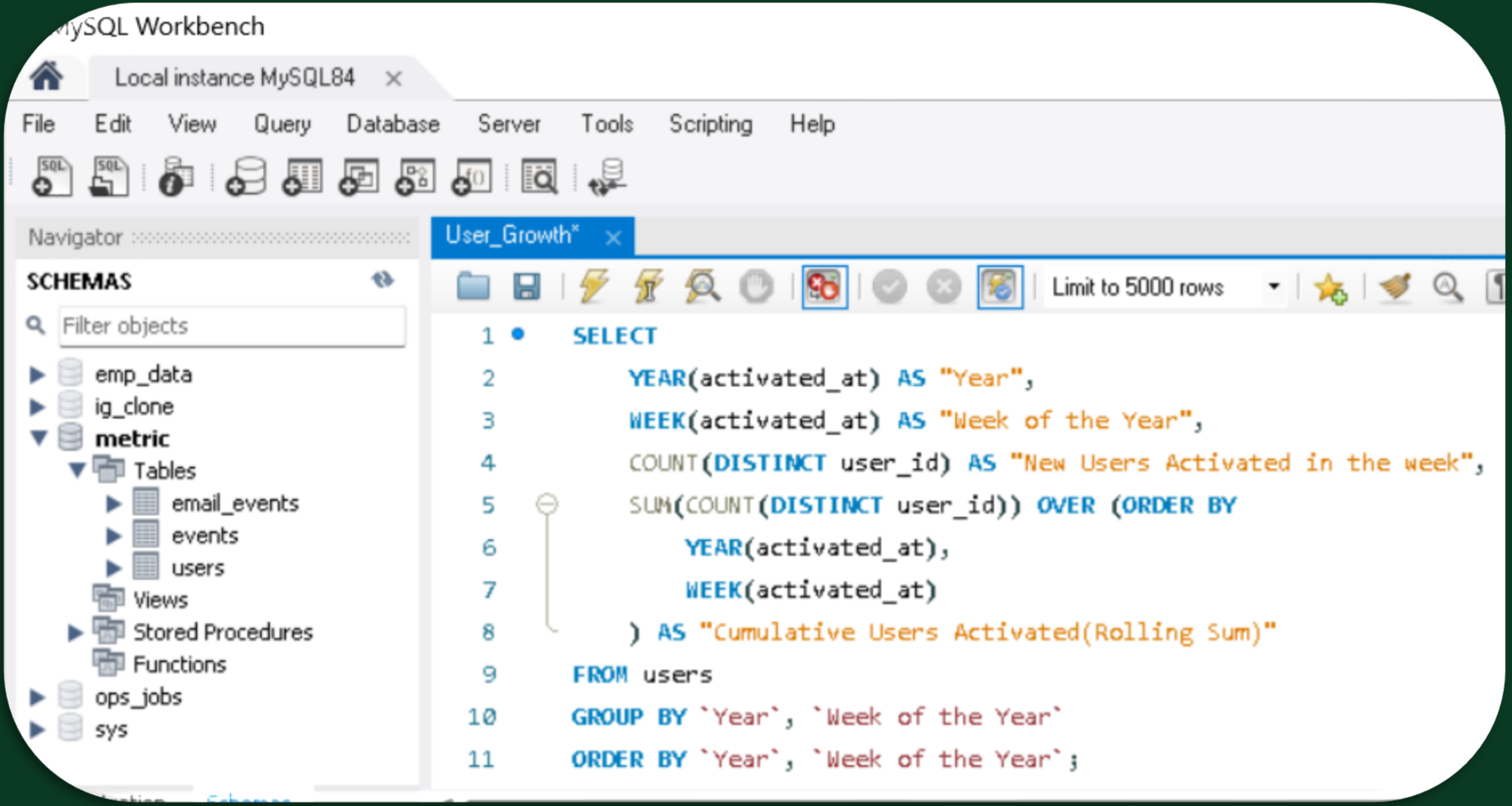
- ➢ User engagement peaked in week 30 (1467 events), relate this with specific marketing campaign or external events to find corelation. This can be used to increase user engagement in future.
- ➢ Relate marketing campaign week wise to find effectiveness and impact on weekly user engagement and use it as feedback on marketing campaigns.

By: Nilesh Kulkarni

Case Study 2:
Insights B. User Growth Analysis

SQL Query to extract weekly User growth data

- Objective: Analyze the growth of users over time for a product.
- Your Task: Write an SQL query to calculate the user growth for the product.



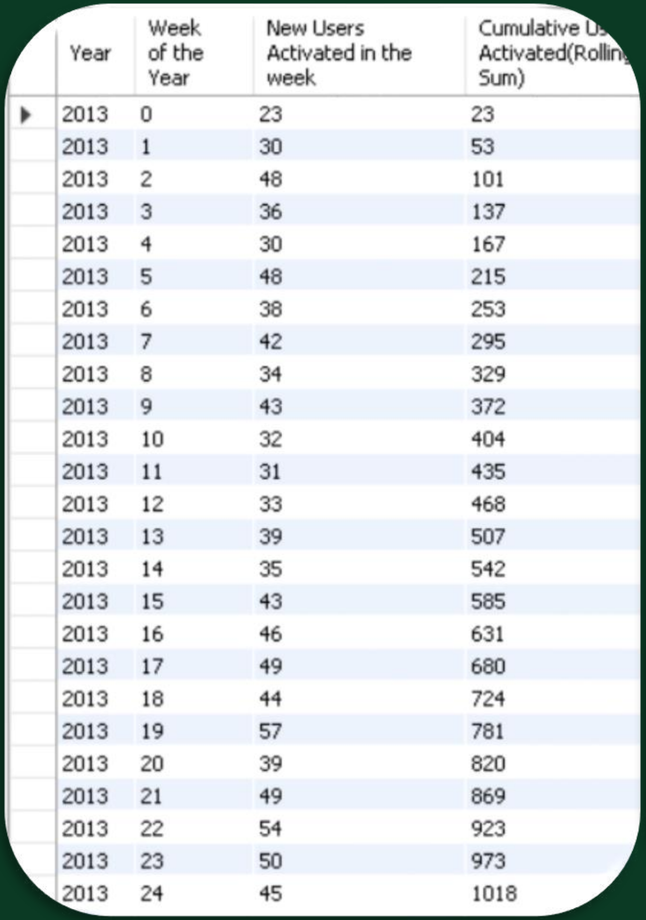
```sql
SELECT
    YEAR(activated_at) AS "Year",
    WEEK(activated_at) AS "Week of the Year",
    COUNT(DISTINCT user_id) AS "New Users Activated in the week",
    SUM(COUNT(DISTINCT user_id)) OVER (ORDER BY
        YEAR(activated_at),
        WEEK(activated_at)
    ) AS "Cumulative Users Activated(Rolling Sum)"
FROM users
GROUP BY `Year`, `Week of the Year`
ORDER BY `Year`, `Week of the Year`;
```

By: Nilesh Kulkarni

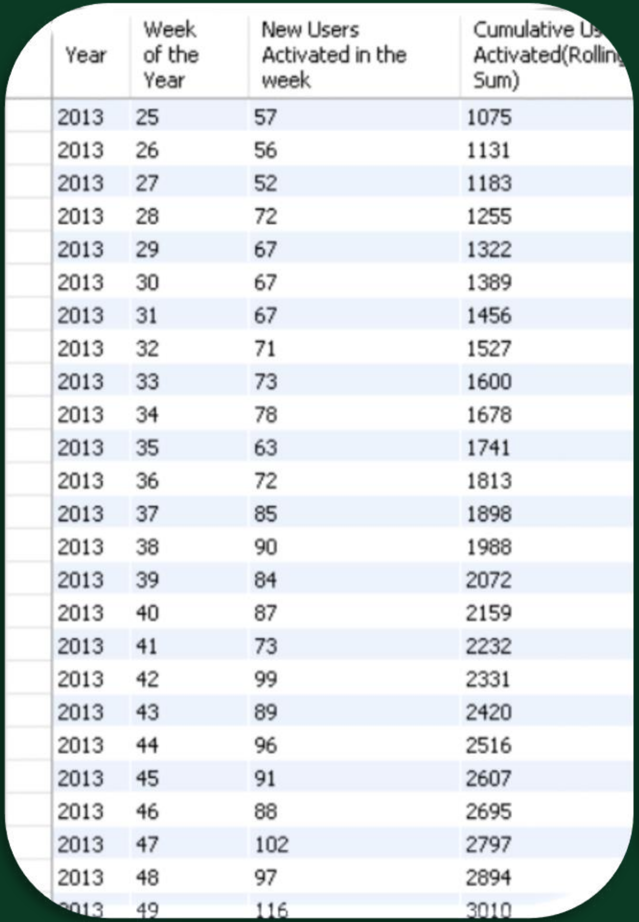
(Insight B Continued..)

Case Study 2: Insights B. User Growth Analysis (..continued)

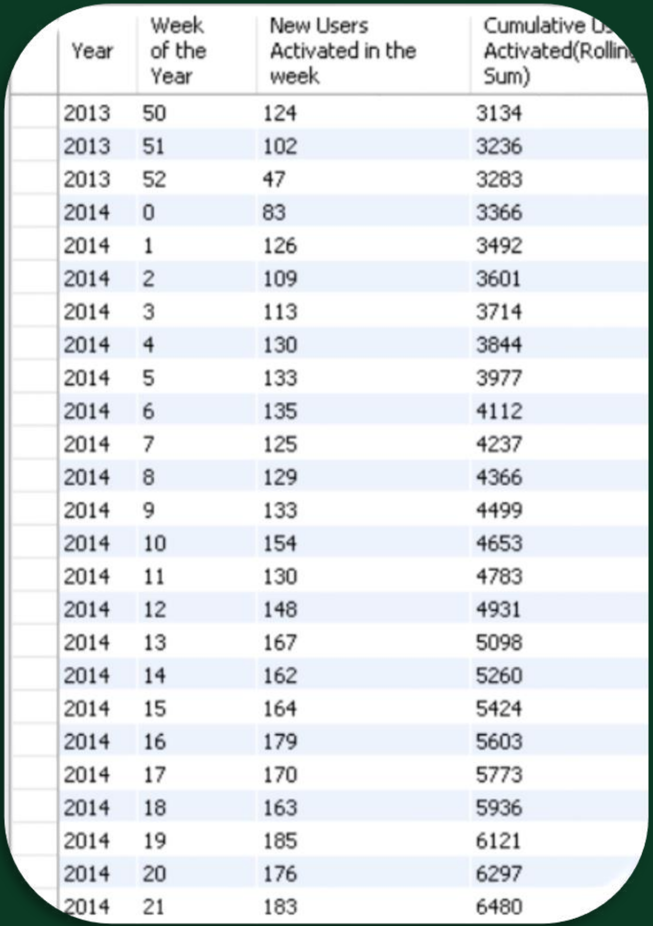
SQL Query and Output showing New Users activated week-wise and cumulative basis



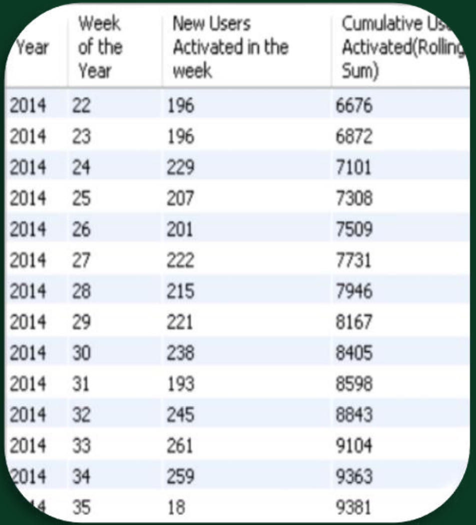
Year	Week of the Year	New Users Activated in the week	Cumulative U Activated(Rollin Sum)
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781
2013	20	39	820
2013	21	49	869
2013	22	54	923
2013	23	50	973
2013	24	45	1018



Year	Week of the Year	New Users Activated in the week	Cumulative U Activated(Rollin Sum)
2013	25	57	1075
2013	26	56	1131
2013	27	52	1183
2013	28	72	1255
2013	29	67	1322
2013	30	67	1389
2013	31	67	1456
2013	32	71	1527
2013	33	73	1600
2013	34	78	1678
2013	35	63	1741
2013	36	72	1813
2013	37	85	1898
2013	38	90	1988
2013	39	84	2072
2013	40	87	2159
2013	41	73	2232
2013	42	99	2331
2013	43	89	2420
2013	44	96	2516
2013	45	91	2607
2013	46	88	2695
2013	47	102	2797
2013	48	97	2894
2013	49	116	3010



Year	Week of the Year	New Users Activated in the week	Cumulative U Activated(Rollin Sum)
2013	50	124	3134
2013	51	102	3236
2013	52	47	3283
2014	0	83	3366
2014	1	126	3492
2014	2	109	3601
2014	3	113	3714
2014	4	130	3844
2014	5	133	3977
2014	6	135	4112
2014	7	125	4237
2014	8	129	4366
2014	9	133	4499
2014	10	154	4653
2014	11	130	4783
2014	12	148	4931
2014	13	167	5098
2014	14	162	5260
2014	15	164	5424
2014	16	179	5603
2014	17	170	5773
2014	18	163	5936
2014	19	185	6121
2014	20	176	6297
2014	21	183	6480



Year	Week of the Year	New Users Activated in the week	Cumulative Us Activated(Rolling Sum)
2014	22	196	6676
2014	23	196	6872
2014	24	229	7101
2014	25	207	7308
2014	26	201	7509
2014	27	222	7731
2014	28	215	7946
2014	29	221	8167
2014	30	238	8405
2014	31	193	8598
2014	32	245	8843
2014	33	261	9104
2014	34	259	9363
14	35	18	9381

Insights:

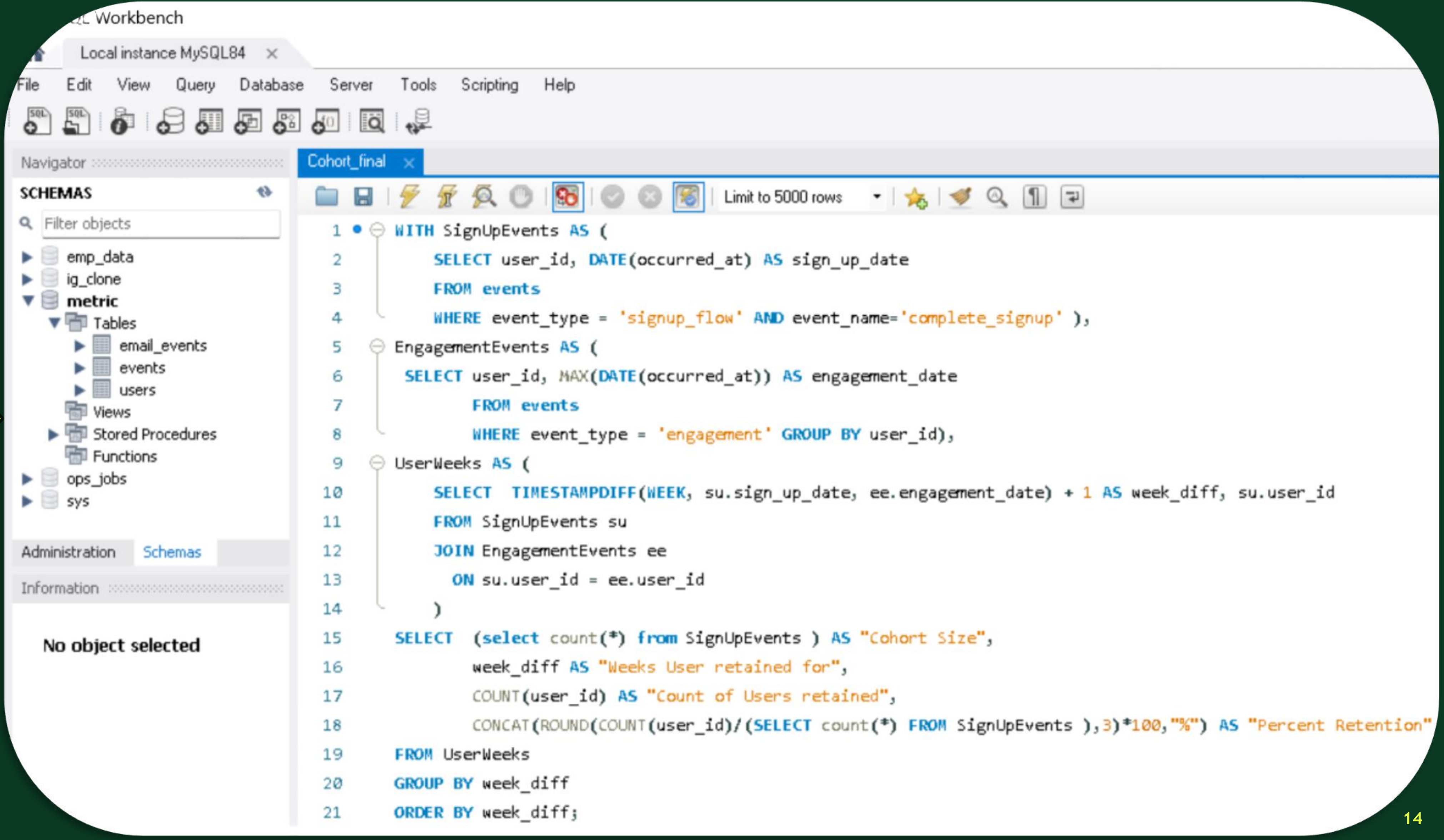
- User Growth has improved in 2014 compared to 2013, specifically post week 20 of 2014. Relate this with marketing campaigns/events.
- Peak user growth is in week 33 of 2014(261)
- Relate weekly user growth with ongoing campaigns/external events as feedback

By: Nilesh Kulkarni

Case Study 2: Insights C. Weekly Retention Analysis (SQL Query)

- Objective: Analyze the retention of users on a weekly basis after signing up for a product.
- Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort

SQL Query for weekly User retention data for sign-up cohort



```sql
WITH SignUpEvents AS (
    SELECT user_id, DATE(occurred_at) AS sign_up_date
    FROM events
    WHERE event_type = 'signup_flow' AND event_name='complete_signup' ),
EngagementEvents AS (
 SELECT user_id, MAX(DATE(occurred_at)) AS engagement_date
       FROM events
       WHERE event_type = 'engagement' GROUP BY user_id),
UserWeeks AS (
    SELECT  TIMESTAMPDIFF(WEEK, su.sign_up_date, ee.engagement_date) + 1 AS week_diff, su.user_id
    FROM SignUpEvents su
    JOIN EngagementEvents ee
      ON su.user_id = ee.user_id
    )
SELECT  (select count(*) from SignUpEvents ) AS "Cohort Size",
        week_diff AS "Weeks User retained for",
        COUNT(user_id) AS "Count of Users retained",
        CONCAT(ROUND(COUNT(user_id)/(SELECT count(*) FROM SignUpEvents ),3)*100,"%") AS "Percent Retention"
FROM UserWeeks
GROUP BY week_diff
ORDER BY week_diff;
```

By: Nilesh Kulkarni

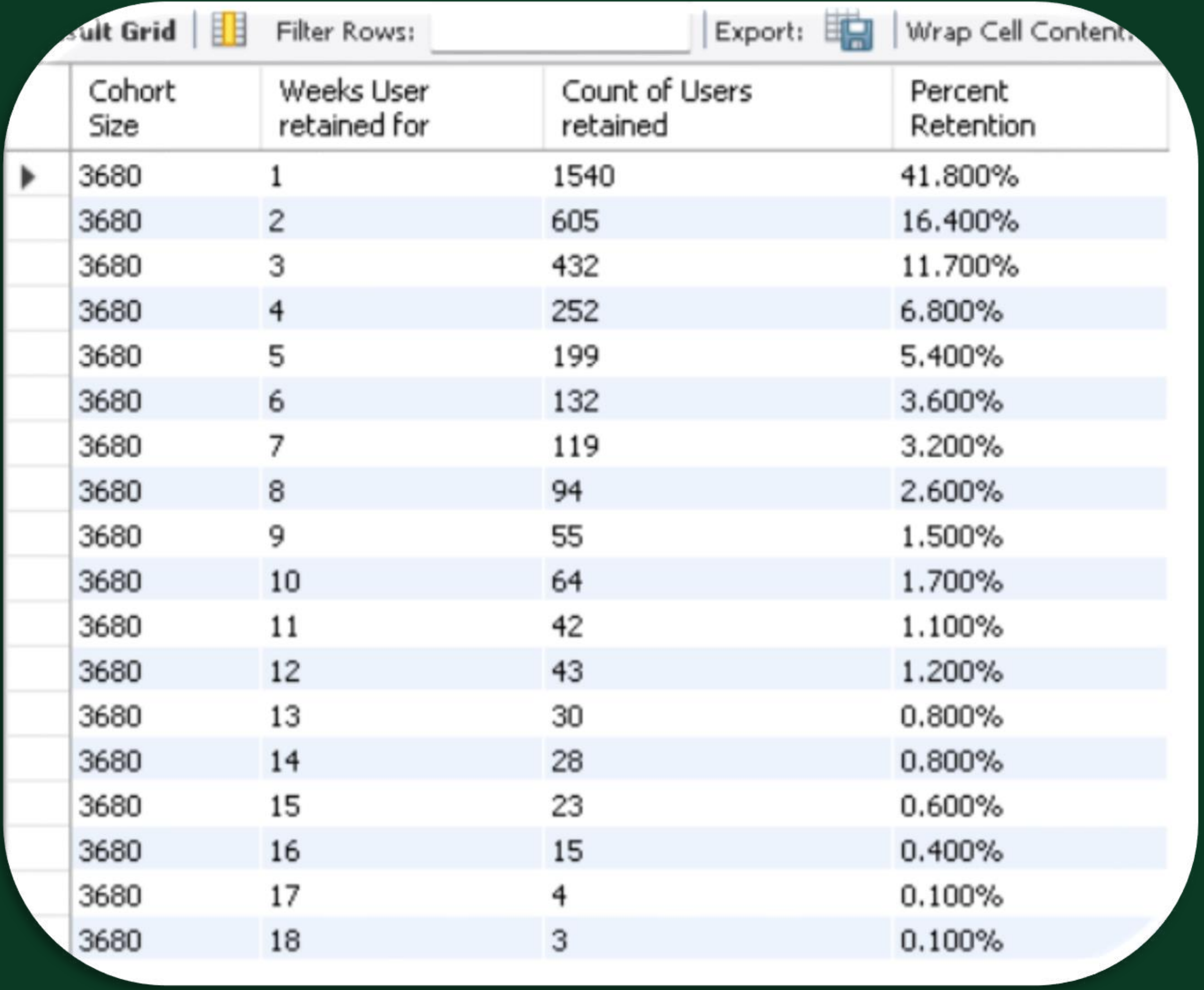
Case Study 2: Insights C. Weekly Retention Analysis (SQL Output)

User retention is calculated based on weeks between user sign-up date and user's last engagement activity/event date. The cohort is considered as collection of users who sign-up (row with event type 'signup_flow' & event name 'complete_signup' in events table).

Insights:

- User retention is at peak in first week post sign-up at 41.8%
- User retention is dropping below 10% after 3 weeks of sign-up and below 5% after 6 weeks of sign-up
- Need to target users with more engaging content to improve user retention post first 3-4 weeks. Need RCA to find causes for dropping users retention post initial few weeks.

SQL Output for weekly User retention data for sign-up cohort



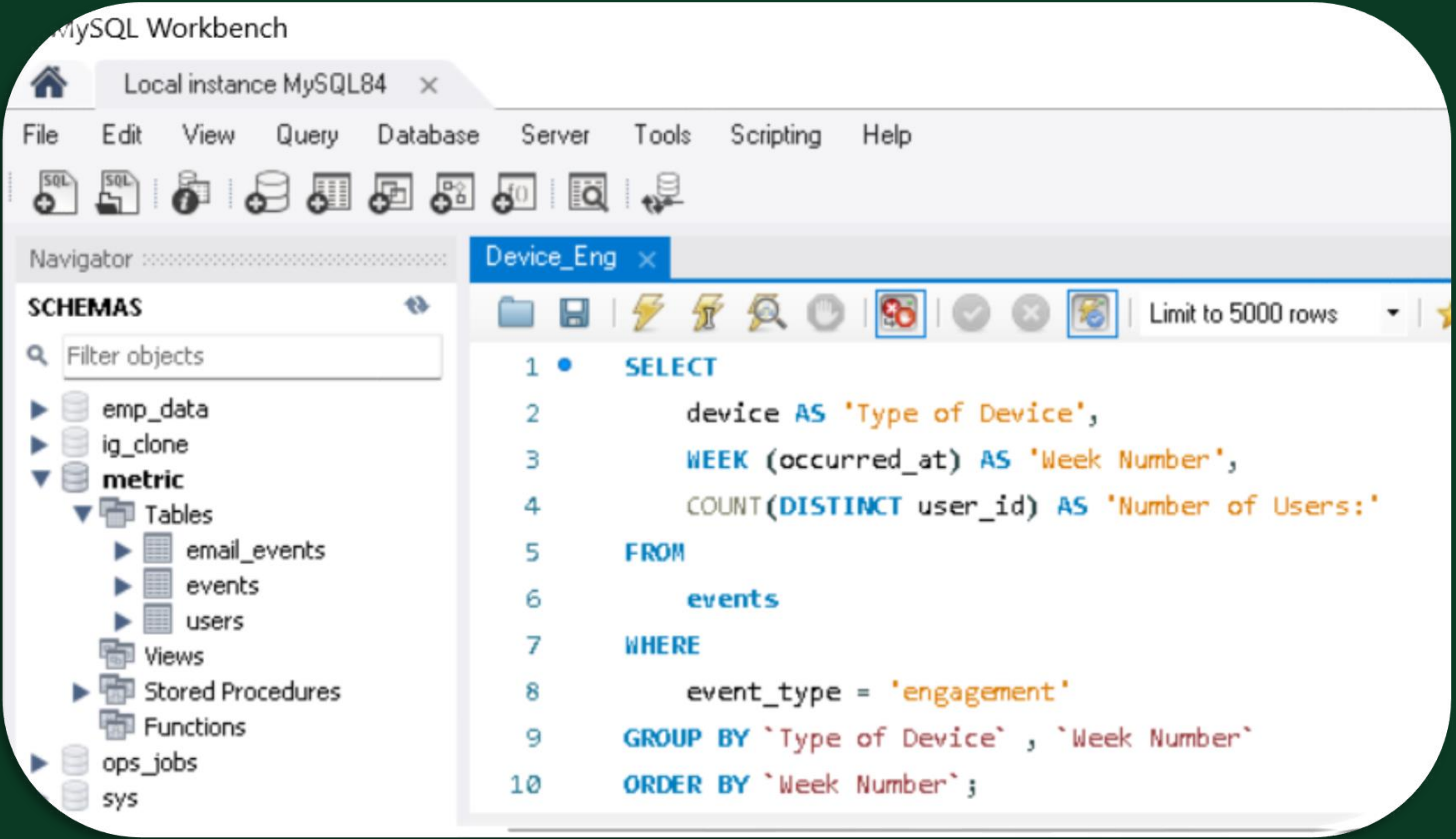
Cohort Size	Weeks User retained for	Count of Users retained	Percent Retention
3680	1	1540	41.800%
3680	2	605	16.400%
3680	3	432	11.700%
3680	4	252	6.800%
3680	5	199	5.400%
3680	6	132	3.600%
3680	7	119	3.200%
3680	8	94	2.600%
3680	9	55	1.500%
3680	10	64	1.700%
3680	11	42	1.100%
3680	12	43	1.200%
3680	13	30	0.800%
3680	14	28	0.800%
3680	15	23	0.600%
3680	16	15	0.400%
3680	17	4	0.100%
3680	18	3	0.100%

By: Nilesh Kulkarni

Case Study 2: Insights D. Weekly Engagement Per Device (SQL Query)

SQL Query for extracting weekly engagement per device

- Objective: Measure the activeness of users on a weekly basis per device.
- Your Task: Write an SQL query to calculate the weekly engagement per device



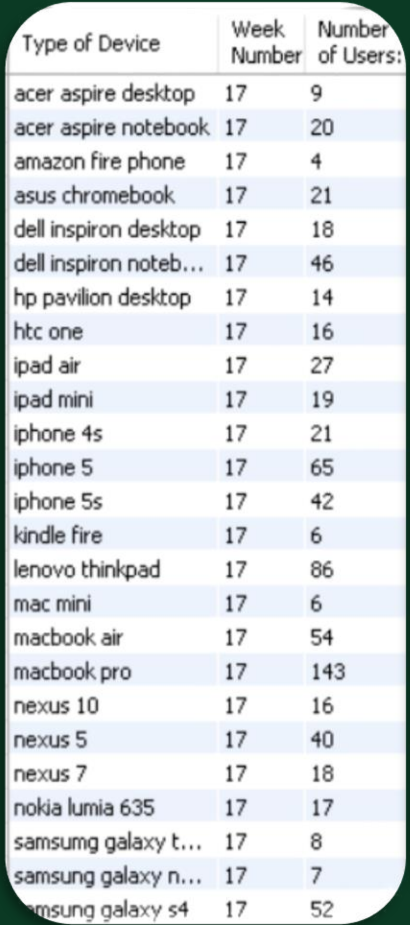
```
SELECT
    device AS 'Type of Device',
    WEEK (occurred_at) AS 'Week Number',
    COUNT(DISTINCT user_id) AS 'Number of Users:'
FROM
    events
WHERE
    event_type = 'engagement'
GROUP BY `Type of Device` , `Week Number`
ORDER BY `Week Number`;
```

By: Nilesh Kulkarni

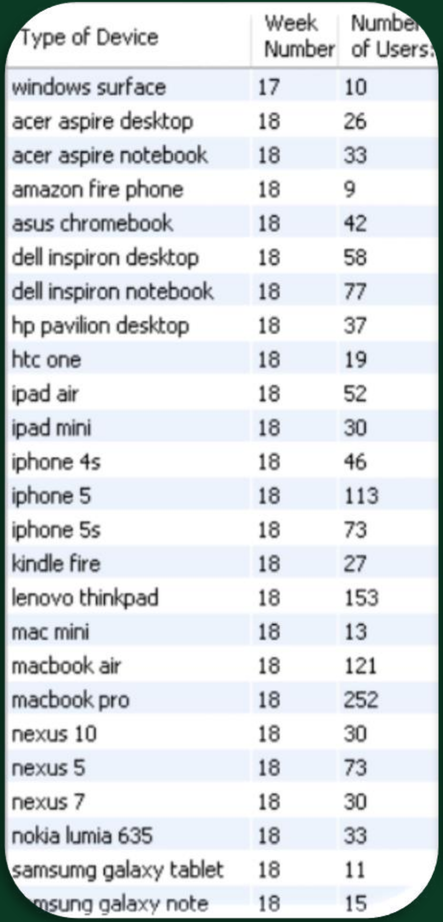
(Insight D Continued..)

Case Study 2: Insights D. Weekly Engagement Per Device (Output)

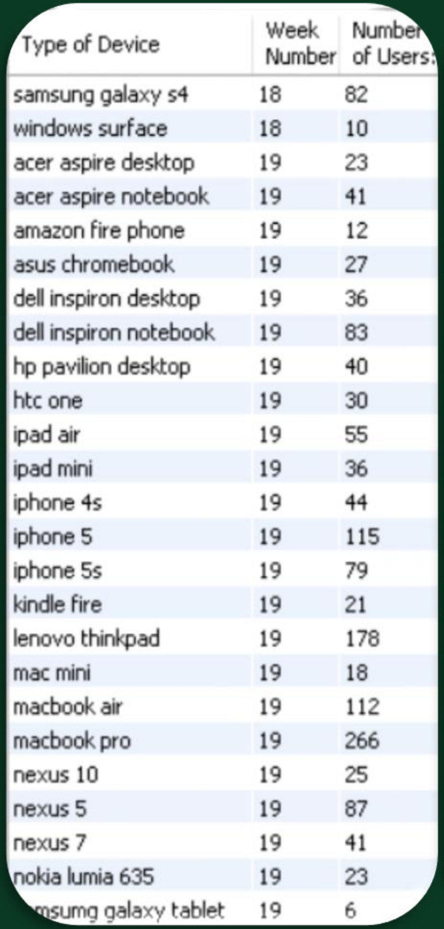
SQL Output data for weekly engagement per device**



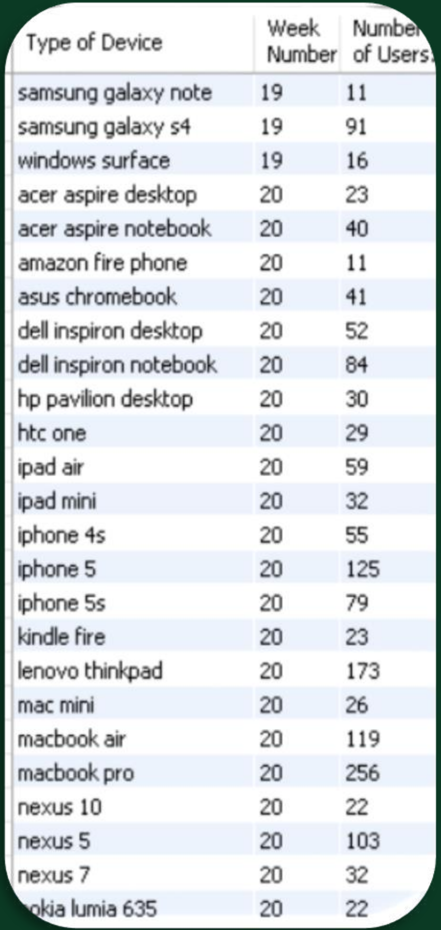
Type of Device	Week Number	Number of Users:
acer aspire desktop	17	9
acer aspire notebook	17	20
amazon fire phone	17	4
asus chromebook	17	21
dell inspiron desktop	17	18
dell inspiron noteb...	17	46
hp pavilion desktop	17	14
htc one	17	16
ipad air	17	27
ipad mini	17	19
iphone 4s	17	21
iphone 5	17	65
iphone 5s	17	42
kindle fire	17	6
lenovo thinkpad	17	86
mac mini	17	6
macbook air	17	54
macbook pro	17	143
nexus 10	17	16
nexus 5	17	40
nexus 7	17	18
nokia lumia 635	17	17
samsung galaxy t...	17	8
samsung galaxy n...	17	7
msung galaxy s4	17	52



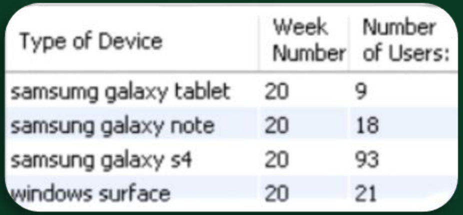
Type of Device	Week Number	Number of Users:
windows surface	17	10
acer aspire desktop	18	26
acer aspire notebook	18	33
amazon fire phone	18	9
asus chromebook	18	42
dell inspiron desktop	18	58
dell inspiron notebook	18	77
hp pavilion desktop	18	37
htc one	18	19
ipad air	18	52
ipad mini	18	30
iphone 4s	18	46
iphone 5	18	113
iphone 5s	18	73
kindle fire	18	27
lenovo thinkpad	18	153
mac mini	18	13
macbook air	18	121
macbook pro	18	252
nexus 10	18	30
nexus 5	18	73
nexus 7	18	30
nokia lumia 635	18	33
samsung galaxy tablet	18	11
msung galaxy note	18	15



Type of Device	Week Number	Number of Users:
samsung galaxy s4	18	82
windows surface	18	10
acer aspire desktop	19	23
acer aspire notebook	19	41
amazon fire phone	19	12
asus chromebook	19	27
dell inspiron desktop	19	36
dell inspiron notebook	19	83
hp pavilion desktop	19	40
htc one	19	30
ipad air	19	55
ipad mini	19	36
iphone 4s	19	44
iphone 5	19	115
iphone 5s	19	79
kindle fire	19	21
lenovo thinkpad	19	178
mac mini	19	18
macbook air	19	112
macbook pro	19	266
nexus 10	19	25
nexus 5	19	87
nexus 7	19	41
nokia lumia 635	19	23
msumg galaxy tablet	19	6



Type of Device	Week Number	Number of Users:
samsung galaxy note	19	11
samsung galaxy s4	19	91
windows surface	19	16
acer aspire desktop	20	23
acer aspire notebook	20	40
amazon fire phone	20	11
asus chromebook	20	41
dell inspiron desktop	20	52
dell inspiron notebook	20	84
hp pavilion desktop	20	30
htc one	20	29
ipad air	20	59
ipad mini	20	32
iphone 4s	20	55
iphone 5	20	125
iphone 5s	20	79
kindle fire	20	23
lenovo thinkpad	20	173
mac mini	20	26
macbook air	20	119
macbook pro	20	256
nexus 10	20	22
nexus 5	20	103
nexus 7	20	32
okia lumia 635	20	22



Type of Device	Week Number	Number of Users:
samsumg galaxy tablet	20	9
samsung galaxy note	20	18
samsung galaxy s4	20	93
windows surface	20	21

Insights:

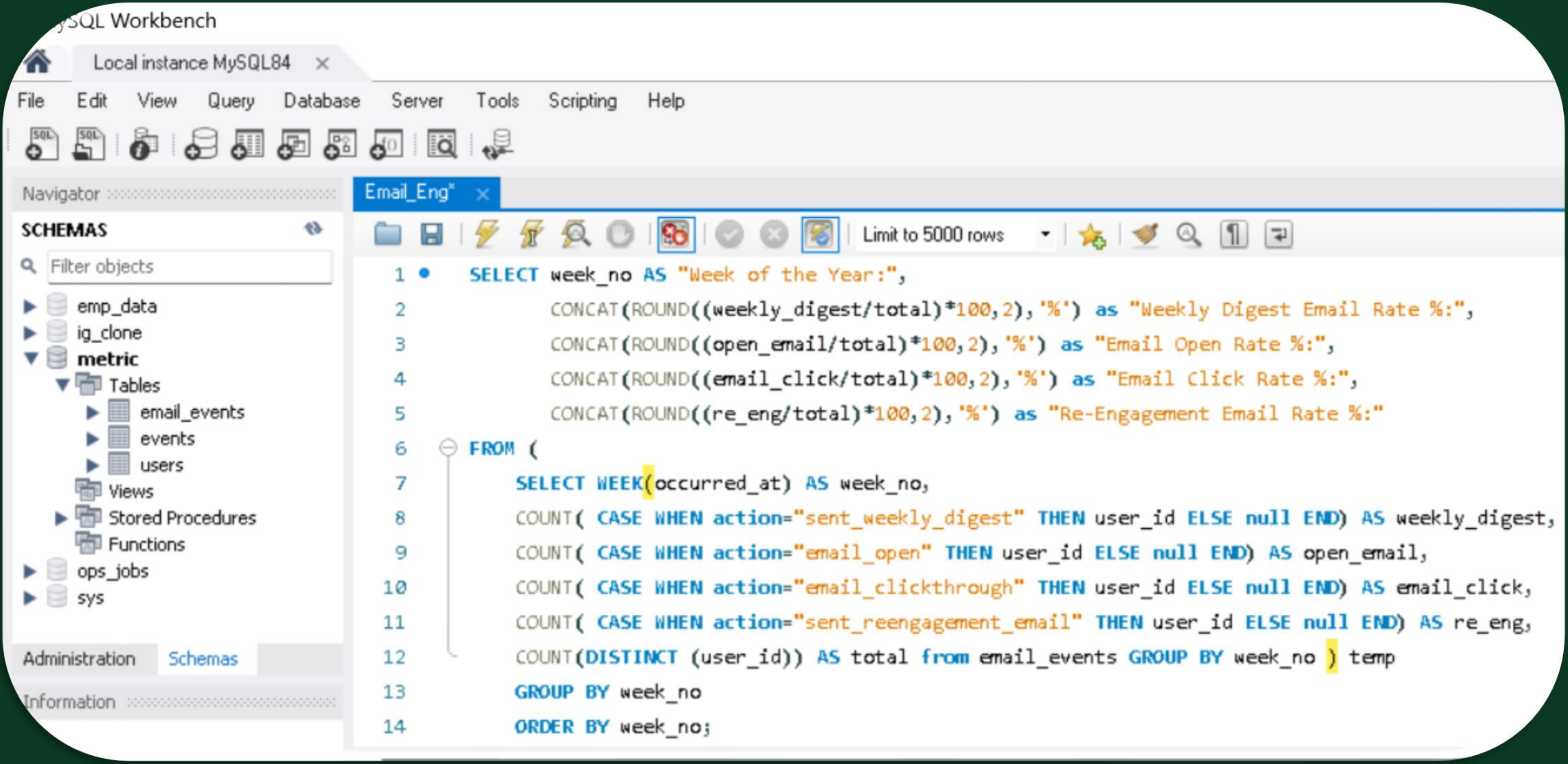
- ➢ Device Engagement varies greatly across devices and weeks.
- ➢ Target devices with higher potential of engagement and work on strategies to increase engagement for identified devices
- ➢ Identify devices with lower user Engagement and work on Root Cause Analysis for any device specific issues for app.

**Data for only week 17 to 20 is shown, provided SQL will return data for all the weeks

Case Study 2: Insights E. Email Engagement Analysis (SQL Query)

- Objective: Analyze how users are engaging with the email service.
- Task: Write an SQL query to calculate the email engagement metrics.

SQL Query for extracting Email Engagement Analysis Data



```sql
SELECT week_no AS "Week of the Year:",
        CONCAT(ROUND((weekly_digest/total)*100,2),'%') as "Weekly Digest Email Rate %:",
        CONCAT(ROUND((open_email/total)*100,2),'%') as "Email Open Rate %:",
        CONCAT(ROUND((email_click/total)*100,2),'%') as "Email Click Rate %:",
        CONCAT(ROUND((re_eng/total)*100,2),'%') as "Re-Engagement Email Rate %:"
FROM (
    SELECT WEEK(occurred_at) AS week_no,
    COUNT( CASE WHEN action="sent_weekly_digest" THEN user_id ELSE null END) AS weekly_digest,
    COUNT( CASE WHEN action="email_open" THEN user_id ELSE null END) AS open_email,
    COUNT( CASE WHEN action="email_clickthrough" THEN user_id ELSE null END) AS email_click,
    COUNT( CASE WHEN action="sent_reengagement_email" THEN user_id ELSE null END) AS re_eng,
    COUNT(DISTINCT (user_id)) AS total from email_events GROUP BY week_no ) temp
    GROUP BY week_no
    ORDER BY week_no;
```

By: Nilesh Kulkarni

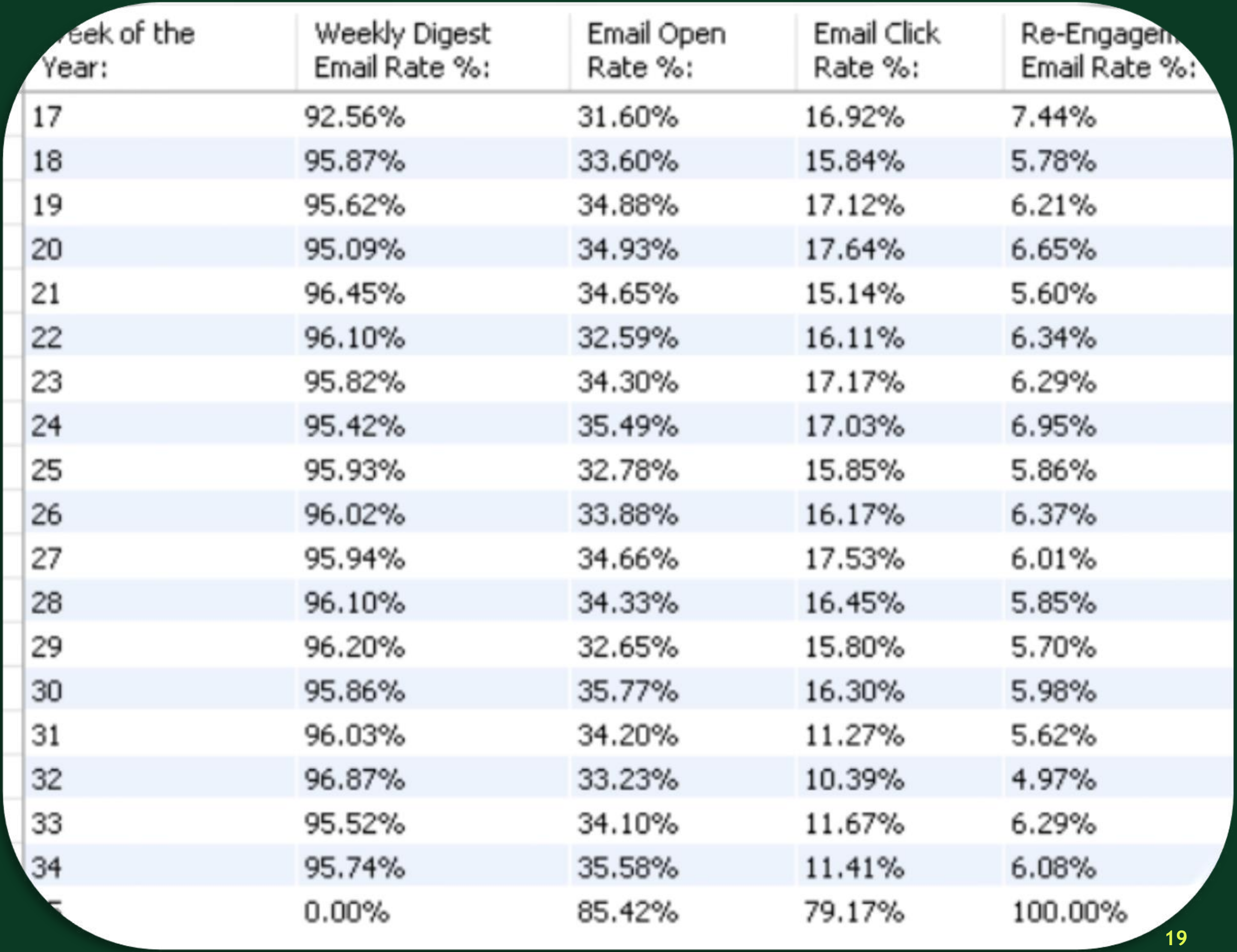
(Insight E Continued..)

Case Study 2: Insights E. Email Engagement Analysis (SQL Output)

SQL output for Weekly Email Engagement Analysis Data

Insights:

- Weekly Digest Rate is constant for all weeks (except last week). Little scope to improve it overall.
- Email Open rate is in the range of 30-35%, need to find ways to engage users more to make open emails with ways like catchy subject lines, time of email delivery and frequency.
- Email click rate is in the range 10-17 % so need to refine email campaign with better & engaging content.
- Re-engagement Email rate is less than 10%. Need more targeting re-engagement strategy.



eek of the Year:	Weekly Digest Email Rate %:	Email Open Rate %:	Email Click Rate %:	Re-Engagem Email Rate %:
17	92.56%	31.60%	16.92%	7.44%
18	95.87%	33.60%	15.84%	5.78%
19	95.62%	34.88%	17.12%	6.21%
20	95.09%	34.93%	17.64%	6.65%
21	96.45%	34.65%	15.14%	5.60%
22	96.10%	32.59%	16.11%	6.34%
23	95.82%	34.30%	17.17%	6.29%
24	95.42%	35.49%	17.03%	6.95%
25	95.93%	32.78%	15.85%	5.86%
26	96.02%	33.88%	16.17%	6.37%
27	95.94%	34.66%	17.53%	6.01%
28	96.10%	34.33%	16.45%	5.85%
29	96.20%	32.65%	15.80%	5.70%
30	95.86%	35.77%	16.30%	5.98%
31	96.03%	34.20%	11.27%	5.62%
32	96.87%	33.23%	10.39%	4.97%
33	95.52%	34.10%	11.67%	6.29%
34	95.74%	35.58%	11.41%	6.08%
[illegible]	0.00%	85.42%	79.17%	100.00%

Results

Conclusion/Business Value-Add from Analysis

- Case Study 1 - Job Data Analysis
  - Business team can use provided insights for improving job review rate as well as Throughput by finding root cause for variation/up-downs
  - Business team can use language specific share % to target at low % share language users
- Case Study 2 - Investigating Metric Spike
  - Business Team can use insights to find out weekly user engagement as well as device specific weekly engagement and prepare targeting engagement strategies for future.
  - Business Teams can use insights on User growth & User retention analysis to measure success of marketing campaigns and also impact of external events on User stats
  - Business Team can use Email engagement insights to find out current levels as well as improvements needed to foster the email engagements
- Overall, the data analysis & derived insights are significantly useful & value-add to business teams

Personal Achievement/Upskilling

- Good learning Opportunity to learn skills to extract business insights from given data
- Learned how to cleanse data with large datasets, create MySQL tables with DDL statements and query data using DML statements.
- Gained skills to analyse large data and come up with optimal SQL queries to extract business insights.
- Leaned advanced topics in MySQL like CTE (Common Table Expressions), Joins, Window functions & how to use those to extract useful business insights with practical hands-on.
- Overall, it is excellent learning and upskilling experience with Trainity. This has enhanced my know-how, understanding and general interest in data analytics.

Thank you

Nilesh Kulkarni
inileshkulkarni@gmail.com