

## Operation and metric analysis

This project operation and metric analysis focuses on analysing the data which is provided by and answer the question asked by different departments. This project insights is helpful for ops team, support team, marketing team and also means better automation and better understanding between cross-functional teams, and more effective workflow. Investigating metric spike is also an important part of operation analytics to be able to understand or make other teams understand questions.

Here ,I have been given 2 different case studies and related data sets of total 4,where I have driven certain insights out of it and answered the questions.

**Approach :** I have already installed mysql workbench for the previous project ,I have created new query page in mysql and created database and loaded existing datasets in this project ,and then if there are any modifications like insufficient data I have modified it and then I wrote queries to extract the tables and drawn insights and extracted some useful information to answer the given question.

**Tech-stack :** used is mysql workbench for to extract data and for writing queries and to store the large amounts of data.

### **Insights:**

In case study-1(job data)

1.calculate the number of jobs reviewed per hour per day for November 2020?

Here in the first case the given data set which was provided was insufficient so I created a database and then a table and then filled of 30 value where 8 values were given and rest of them were filled manually.

MySQL Workbench

MySQL@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator

operation\_analytics

Limit to 1000 rows

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
1 create database operation_analytics;
2 use operation_analytics;
3 create table job_data(
4   ds date,
5   job_id int,
6   actors_id int,
7   event varchar(255),
8   language varchar(255),
9   time_spent int,
10  org varchar(255)
11 );
12 select*from job_data;
13 insert into job_data(ds,job_id,actors_id,event,language,time_spent,org)
14 values('2020-11-30',21,1001,'skip','English',15,'A');
```

Result Grid

dates	jobs reviewed per hour
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	13:42:10	select count(distinct job_id)/30*24 as num_jobs_reviewed from job_data where ds between 2020-11-01 and 20...	1 row(s) returned	0.000 sec / 0.000 sec
10	13:47:29	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	0 row(s) returned	0.000 sec / 0.000 sec
11	13:47:52	insert into job_data(ds,job_id,actors_id,event,language,time_spent,org) values(2020-11-30,21,1001,'skip','Engl...	32 row(s) affected Records: 32 Duplicates: 0 Warnings: 0	0.031 sec
12	13:47:59	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	30 row(s) returned	0.000 sec / 0.000 sec
13	13:49:31	select count(distinct job_id)/30*24 as num_jobs_reviewed from job_data where ds between 2020-11-01 and 2...	1 row(s) returned	0.000 sec / 0.000 sec
14	13:50:25	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	30 row(s) returned	0.016 sec / 0.000 sec

Connection Details

Name: H27.0.0.1 E3306

Use root

Use root@localhost

Use TLS\_AES\_256\_GCM\_SHA384

Server MySQL Community

Object Info Session

MySQL Workbench

MySQL@127.0.0.1:3306

File Edit View Query Database Server Tools Scripting Help

Navigator

operation\_analytics

Limit to 1000 rows

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
14 insert into job_data(ds,job_id,actors_id,event,language,time_spent,org)
15 values('2020-11-30',21,1001,'skip','English',15,'A'),
16 ('2020-11-30',22,1006,'transfer','Arabic',25,'B'),
17 ('2020-11-29',23,1003,'decision','Persian',20,'C'),
18 ('2020-11-28',23,1005,'transfer','Persian',22,'D'),
19 ('2020-11-28',25,1002,'decision','Hindi',11,'B'),
20 ('2020-11-27',11,1007,'decision','French',104,'D'),
21 ('2020-11-26',23,1004,'skip','Persian',56,'A'),
22 ('2020-11-25',20,1004,'transfer','Italian',45,'C'),
23 ('2020-11-24',21,1001,'skip','English',15,'A'),
24 ('2020-11-23',22,1006,'transfer','Arabic',25,'B'),
25 ('2020-11-22',24,1003,'decision','Persian',20,'C'),
26 ('2020-11-21',12,1005,'transfer','Persian',22,'D'),
27 ('2020-11-20',25,1002,'decision','Hindi',11,'B');
```

Result Grid

dates	jobs reviewed per hour
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	13:42:10	select count(distinct job_id)/30*24 as num_jobs_reviewed from job_data where ds between 2020-11-01 and 20...	1 row(s) returned	0.000 sec / 0.000 sec
10	13:47:29	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	0 row(s) returned	0.000 sec / 0.000 sec
11	13:47:52	insert into job_data(ds,job_id,actors_id,event,language,time_spent,org) values(2020-11-30,21,1001,'skip','Engl...	32 row(s) affected Records: 32 Duplicates: 0 Warnings: 0	0.031 sec
12	13:47:59	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	30 row(s) returned	0.000 sec / 0.000 sec
13	13:49:31	select count(distinct job_id)/30*24 as num_jobs_reviewed from job_data where ds between 2020-11-01 and 2...	1 row(s) returned	0.000 sec / 0.000 sec
14	13:50:25	select ds as dates, round(count(job_id)/sum(time_spent)/3600 as jobs reviewed per hour' from job_data where...	30 row(s) returned	0.016 sec / 0.000 sec

Connection Details

Name: H27.0.0.1 E3306

Use root

Use root@localhost

Use TLS\_AES\_256\_GCM\_SHA384

Server MySQL Community

Object Info Session

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'PERFORMANCE' tabs. The main editor displays a SQL query:

```

40 ('2020-11-05', 29, 1003, 'decision', 'Persian', 20, 'C'),
41 ('2020-11-05', 23, 1005, 'transfer', 'Persian', 22, 'D'),
42 ('2020-11-04', 24, 1002, 'decision', 'Hindi', 11, 'B'),
43 ('2020-11-03', 11, 1007, 'decision', 'French', 104, 'D'),
44 ('2020-11-02', 16, 1004, 'skip', 'Persian', 96, 'A'),
45 ('2020-11-01', 20, 1004, 'transfer', 'Italian', 45, 'C');
46
47
48 • select ds as dates,
49 round((count(job_id)/sum(time_spent))*3600)as 'jobs reviewed per hour'
50 from job_data
51 where ds between '2020-11-01' and '2020-11-30'
52 group by ds;
53

```

The 'Result Grid' shows the following data:

dates	jobs reviewed per hour
2020-11-30	180
2020-11-29	180
2020-11-28	218
2020-11-27	35

The 'Output' tab shows the execution log with the following messages:

- 10 13:42:10 select count(distinct job\_id)/3600 as num\_jobs\_reviewed from job\_data where ds between 2020-11-01 and 2020-11-30; 1 row(s) returned
- 11 13:47:25 select ds as dates, round(count(job\_id)/sum(time\_spent))\*3600 as jobs reviewed per hour from job\_data where ds between 2020-11-01 and 2020-11-30; 32 row(s) returned
- 12 13:47:52 insert into job\_data(job\_id,action\_id,event,language,time\_spent,org) values(2020-11-30,21,1001,'skip','Engl... 32 row(s) affected Records: 32 Duplicates: 0 Warnings: 0
- 13 13:49:31 select count(distinct job\_id)/3600 as num\_jobs\_reviewed from job\_data where ds between 2020-11-01 and 2020-11-30; 1 row(s) returned
- 14 13:50:25 select ds as dates, round(count(job\_id)/sum(time\_spent))\*3600 as jobs reviewed per hour from job\_data where ds between 2020-11-01 and 2020-11-30; 30 row(s) returned

2.calculate 7-day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL query:

```

53
54 /*calculate 7 day rolling average of throughput*/
55 • select round(count(event)/sum(time_spent),2)as 'weekly_throughput'
56 from job_data;
57
58
59
60
61
62

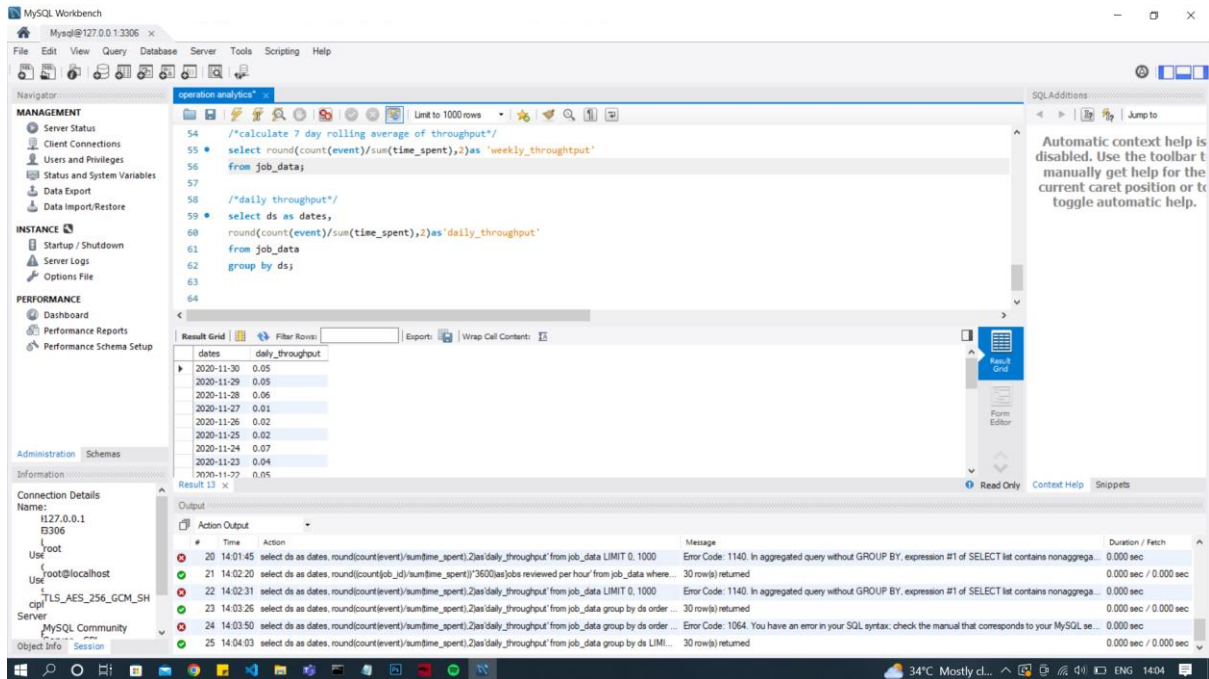
```

The 'Result Grid' shows the following data:

weekly_throughput
0.03

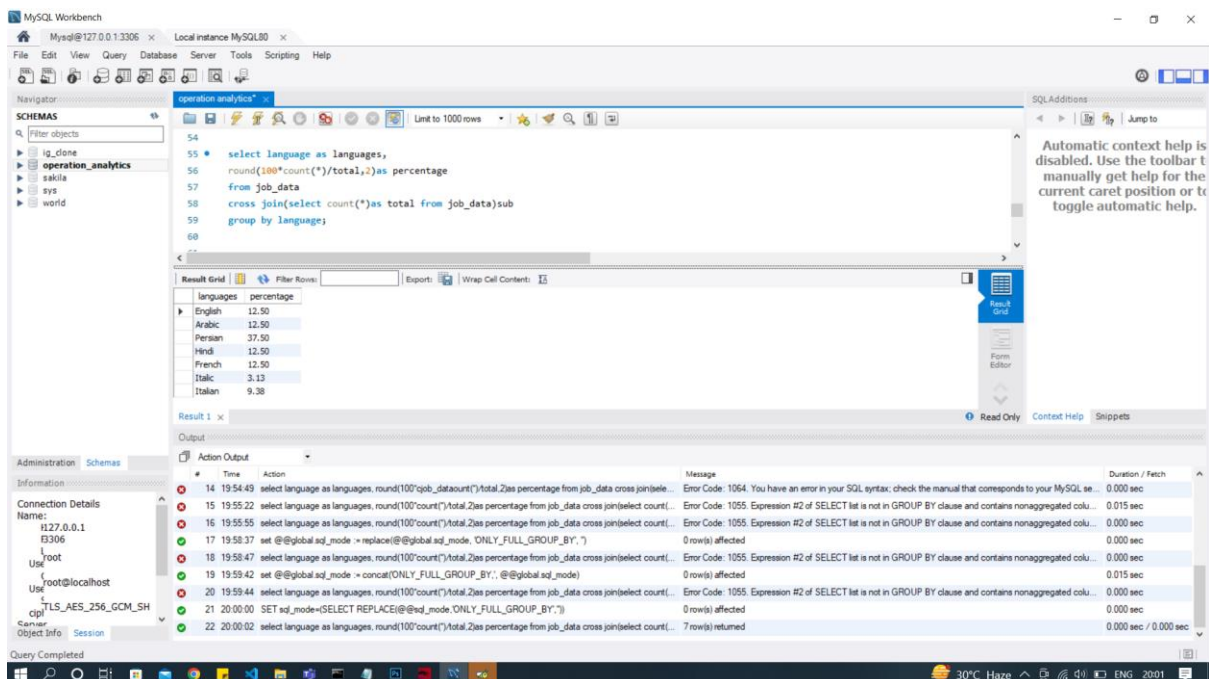
The 'Output' tab shows the execution log with the following messages:

- 11 13:47:52 insert into job\_data(job\_id,action\_id,event,language,time\_spent,org) values(2020-11-30,21,1001,'skip','Engl... 32 row(s) affected Records: 32 Duplicates: 0 Warnings: 0
- 12 13:49:31 select count(distinct job\_id)/3600 as num\_jobs\_reviewed from job\_data where ds between 2020-11-01 and 2020-11-30; 1 row(s) returned
- 13 13:50:25 select ds as dates, round(count(job\_id)/sum(time\_spent))\*3600 as jobs reviewed per hour from job\_data where ds between 2020-11-01 and 2020-11-30; 30 row(s) returned
- 14 13:55:13 select round(count(event)/sum(time\_spent),2) as 'weekly\_throughput' from job\_data LIMIT 0, 1000; 1 row(s) returned
- 15 13:55:29 select round(count(event)/sum(time\_spent),2) as 'weekly\_throughput' from job\_data LIMIT 0, 1000; 1 row(s) returned



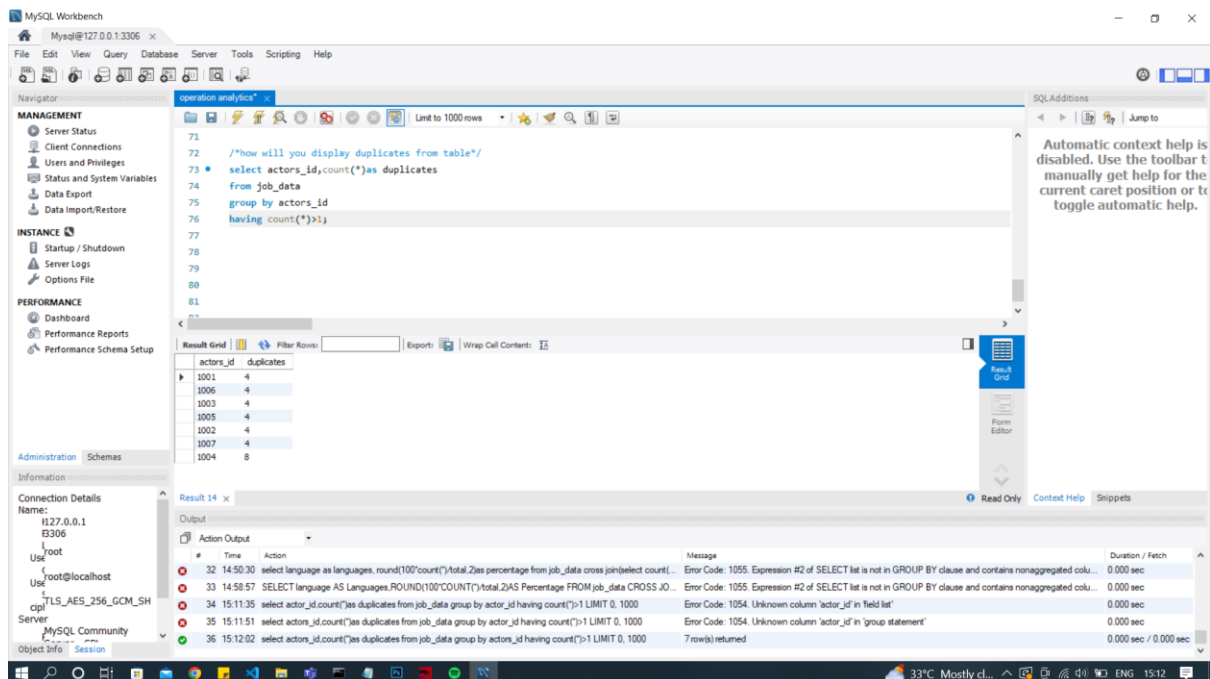
For throughput I prefer 7-day rolling because in daily metric the scale goes up and down and we cannot get pattern and in 7-day it gives you average value where we can get a conclusion.

3.calculate the percentage share of each language in the last 30 days?



In this Persian got the highest percentage of 37.50 and least is Italic of 3.13

#### 4.how will you display duplicates from the table?



This query helped me to display how many duplicates were present.

#### Case study 2(investigating metric spike)

This case study consists of 3 tables

1.table-1 users

2.table-2 events

3.table-3 email\_events

1.calculate the weekly user engagement?

I have imported a existing database table called events in mysql

This event table is the largest table among those two table which took a lot of time to get imported.

And after importing I loaded the data after loading the data I wrote query to get answer for the question



MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_clone

operation\_analytics

sakila

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Server: MySQL Community Server - GPL V8.0.33

Connector: VC++ 8.0.33

Object Info Session

events

Limit to 1000 rows

1 SELECT \* FROM events.events;

2 select extract(week from occurred\_at) as 'weekly numbers',

3 count(distinct user\_id) as 'weekly active users'

4 from events.events

5 where event\_type='engagement'

6 group by 1;

Result Grid

user_id	occurred_at	event_type	event_name	location	device	user_type
10522	2014-05-02 11:02:39	engagement	login	Japan	dell inspiron notebook	3
10522	2014-05-02 11:02:53	engagement	home_page	Japan	dell inspiron notebook	3
10522	2014-05-02 11:03:28	engagement	like_message	Japan	dell inspiron notebook	3
10522	2014-05-02 11:04:09	engagement	viewinbox	Japan	dell inspiron notebook	3
10522	2014-05-02 11:03:16	engagement	search_run	Japan	dell inspiron notebook	3
10612	2014-05-01 09:59:46	engagement	login	Netherlands	iphone 5	1
10612	2014-05-01 10:00:18	engagement	like_message	Netherlands	iphone 5	1
10617	2014-05-01 10:00:53	engagement	send_message	Netherlands	iphone 5	1

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	11:15:54	SHOW DATABASES	OK	0.000 sec
7	11:15:58	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
8	11:15:58	SHOW TABLES FROM 'events' like 'events'	OK	0.000 sec
9	11:16:01	CREATE TABLE 'events'.events ('user_id' double, 'occurred_at' text, 'event_type' text, 'event_name' text, 'location' text, 'device' text, 'user_type' int)	OK	0.000 sec
10	11:16:01	PREPARE stmt FROM 'INSERT INTO 'events'.events ('user_id','occurred_at','event_type','event_name','location','device','user_type') VALUES (10522,'2014-05-02 11:02:39','engagement','login','Japan','dell inspiron notebook',3);'	OK	0.000 sec
11	13:30:51	DEALLOCATE PREPARE stmt	OK	0.000 sec
12	13:38:37	SELECT * FROM events.events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
13	13:42:14	select extract(week from occurred_at) as 'weekly numbers', count(distinct user_id) as 'weekly active users' from events.events where event_type='engagement' group by 1;	19 row(s) returned	0.028 sec / 0.000 sec
14	13:42:24	SELECT * FROM events.events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_clone

operation\_analytics

sakila

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Server: MySQL Community Server - GPL V8.0.33

Connector: VC++ 8.0.33

Object Info Session

events

Limit to 1000 rows

1 SELECT \* FROM events.events;

2 select extract(week from occurred\_at) as 'weekly numbers',

3 count(distinct user\_id) as 'weekly active users'

4 from events.events

5 where event\_type='engagement'

6 group by 1;

Result Grid

weekly numbers	weekly active users
17	443
18	787
19	815
20	805
21	796
22	818
23	846
24	863
25	826
26	817

Output

Action Output

#	Time	Action	Message	Duration / Fetch
7	11:15:58	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
8	11:15:58	SHOW TABLES FROM 'events' like 'events'	OK	0.000 sec
9	11:16:01	CREATE TABLE 'events'.events ('user_id' double, 'occurred_at' text, 'event_type' text, 'event_name' text, 'location' text, 'device' text, 'user_type' int)	OK	0.000 sec
10	11:16:01	PREPARE stmt FROM 'INSERT INTO 'events'.events ('user_id','occurred_at','event_type','event_name','location','device','user_type') VALUES (10522,'2014-05-02 11:02:39','engagement','login','Japan','dell inspiron notebook',3);'	OK	0.000 sec
11	13:30:51	DEALLOCATE PREPARE stmt	OK	0.000 sec
12	13:38:37	SELECT * FROM events.events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
13	13:42:14	select extract(week from occurred_at) as 'weekly numbers', count(distinct user_id) as 'weekly active users' from events.events where event_type='engagement' group by 1;	19 row(s) returned	0.028 sec / 0.000 sec
14	13:42:24	SELECT * FROM events.events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.015 sec
15	13:42:33	select extract(week from occurred_at) as 'weekly numbers', count(distinct user_id) as 'weekly active users' from events.events where event_type='engagement' group by 1;	19 row(s) returned	0.028 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## 2.calculate the user growth for product?

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 SELECT * FROM users.`table-1` users;
```

The result grid displays the following data:

user_id	created_at	company_id	language	activated_at	state
0	2013-01-01 20:59:39	5737	english	2013-01-01 21:01:07	active
1	2013-01-01 13:07:46	28	english		pending
2	2013-01-01 10:59:05	51	english		pending
3	2013-01-01 18:40:36	2800	german	2013-01-01 18:42:02	active
4	2013-01-01 14:37:51	5110	indian	2013-01-01 14:39:05	active
5	2013-01-01 13:39:51	2463	spanish		pending
6	2013-01-01 18:37:27	11699	english	2013-01-01 18:38:45	active
7	2013-01-01 16:19:01	4765	french	2013-01-01 16:20:28	active
8	2013-01-01 04:38:30	2698	french	2013-01-01 04:40:10	active

The output pane shows the execution of the query:

```
1 09:51:57 SELECT * FROM users.`table-1` users LIMIT 0, 1000
1000 row(s) returned
```

The screenshot shows the MySQL Workbench interface with a more complex query in the SQL editor:

```
1 SELECT * FROM users.`table-1` users;
```

```
2 select months,users,round(((users/lag(users,1)over(order by months)-1)*100),2)as 'growth in %'
```

```
3 from
```

```
4 (select extract(month from created_at)as months,count(activated_at)as users
```

```
5 from users.`table-1` users
```

```
6 where activated_at not in(''))
```

```
7 group by 1
```

```
8 order by 1
```

```
9 )sub;
```

The result grid displays the following data:

months	users	growth in %
1	1424	100.00
2	1370	-3.79
3	1530	11.68
4	1814	18.56
5	1986	9.48
6	2172	9.37

The output pane shows the execution of the query:

```
1 09:51:57 SELECT * FROM users.`table-1` users LIMIT 0, 1000
1000 row(s) returned
```

```
2 10:31:17 select months,users,round(((users/lag(users,1)over(order by months)-1)*100),2)as 'growth in %' from (select extract... Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH... 0.000 sec
```

```
3 10:31:44 select months,users,round(((users/lag(users,1)over(order by months)-1)*100),2)as 'growth in %' from (select extract... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL serv... 0.000 sec
```

```
4 10:32:17 select months,users,round(((users/lag(users,1)over(order by months)-1)*100),2)as 'growth in %' from (select extract... 12 row(s) returned 0.063 sec / 0.000 sec
```

The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tab with a tree view of the database structure. The main editor displays a SQL query:

```

1  select first as 'week numbers',
2  sum(case when week_number=0 then 1 else 0 end) as 'week 0',
3  sum(case when week_number=1 then 1 else 0 end) as 'week 1',
4  sum(case when week_number=2 then 1 else 0 end) as 'week 2',
5  sum(case when week_number=3 then 1 else 0 end) as 'week 3',
6  sum(case when week_number=4 then 1 else 0 end) as 'week 4',
7  sum(case when week_number=5 then 1 else 0 end) as 'week 5',
8  sum(case when week_number=6 then 1 else 0 end) as 'week 6',
9  sum(case when week_number=7 then 1 else 0 end) as 'week 7',
10 sum(case when week_number=8 then 1 else 0 end) as 'week 8',
11 sum(case when week_number=9 then 1 else 0 end) as 'week 9',
12 sum(case when week_number=10 then 1 else 0 end) as 'week 10',
13 sum(case when week_number=11 then 1 else 0 end) as 'week 11',
14 sum(case when week_number=12 then 1 else 0 end) as 'week 12'
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938

```

The screenshot displays the MySQL Workbench environment. The top toolbar includes icons for File, Edit, View, Query, Database, Server, Tools, and Scripting. The left sidebar shows the 'Navigator' pane with 'SCHEMAS' and 'events' selected. The main editor window contains a SQL query:

```

26 sum(case when week_number=17 then 1 else 0 end)as'week 17',
27 sum(case when week_number=18 then 1 else 0 end)as'week 18',
28 from(
29   select m.user_id,n.login_week,n.first,m.login_week-first as week_number
30   from
31   (select user_id,extract(week from occurred_at)as login_week from events group by 1,2)m,
32   (select user_id,min(extract(week from occurred_at))as first from events group by 1)n
33   where m.user_id=n.user_id
34 )sub
35 group by first
36 order by first;
37
38

```

The 'Result' pane shows the output of the query, which is a table with 17 columns representing weeks (week 0 to week 16) and 17 rows representing users (17 to 33). The table data is as follows:

week numbers	week 0	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week 14	week 15	week 16
17	443	270	196	167	139	129	122	105	105	101	97	89	90	100	76	62	57
18	517	213	157	127	105	104	109	89	87	88	82	87	86	77	67	55	
19	406	140	98	88	71	67	64	58	67	59	53	48	49	31	41	38	2

The bottom pane shows the 'Action Output' tab with a list of messages and their durations. The messages are as follows:

#	Time	Action	Message	Duration / Fetch
10	15:29:07	SELECT * FROM events events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
11	15:29:10	select first as 'week numbers',sum(case when week_number=0 then 1 else 0 end)as'week 0',sum(case when week_number=1 then 1 else 0 end)as'week 1',sum(case when week_number=2 then 1 else 0 end)as'week 2',sum(case when week_number=3 then 1 else 0 end)as'week 3',sum(case when week_number=4 then 1 else 0 end)as'week 4',sum(case when week_number=5 then 1 else 0 end)as'week 5',sum(case when week_number=6 then 1 else 0 end)as'week 6',sum(case when week_number=7 then 1 else 0 end)as'week 7',sum(case when week_number=8 then 1 else 0 end)as'week 8',sum(case when week_number=9 then 1 else 0 end)as'week 9',sum(case when week_number=10 then 1 else 0 end)as'week 10',sum(case when week_number=11 then 1 else 0 end)as'week 11',sum(case when week_number=12 then 1 else 0 end)as'week 12',sum(case when week_number=13 then 1 else 0 end)as'week 13',sum(case when week_number=14 then 1 else 0 end)as'week 14',sum(case when week_number=15 then 1 else 0 end)as'week 15',sum(case when week_number=16 then 1 else 0 end)as'week 16',sum(case when week_number=17 then 1 else 0 end)as'week 17',sum(case when week_number=18 then 1 else 0 end)as'week 18',sum(case when week_number=19 then 1 else 0 end)as'week 19',sum(case when week_number=20 then 1 else 0 end)as'week 20',sum(case when week_number=21 then 1 else 0 end)as'week 21',sum(case when week_number=22 then 1 else 0 end)as'week 22',sum(case when week_number=23 then 1 else 0 end)as'week 23',sum(case when week_number=24 then 1 else 0 end)as'week 24',sum(case when week_number=25 then 1 else 0 end)as'week 25',sum(case when week_number=26 then 1 else 0 end)as'week 26',sum(case when week_number=27 then 1 else 0 end)as'week 27',sum(case when week_number=28 then 1 else 0 end)as'week 28',sum(case when week_number=29 then 1 else 0 end)as'week 29',sum(case when week_number=30 then 1 else 0 end)as'week 30',sum(case when week_number=31 then 1 else 0 end)as'week 31',sum(case when week_number=32 then 1 else 0 end)as'week 32',sum(case when week_number=33 then 1 else 0 end)as'week 33',sum(case when week_number=34 then 1 else 0 end)as'week 34',sum(case when week_number=35 then 1 else 0 end)as'week 35',sum(case when week_number=36 then 1 else 0 end)as'week 36',sum(case when week_number=37 then 1 else 0 end)as'week 37',sum(case when week_number=38 then 1 else 0 end)as'week 38',sum(case when week_number=39 then 1 else 0 end)as'week 39',sum(case when week_number=40 then 1 else 0 end)as'week 40',sum(case when week_number=41 then 1 else 0 end)as'week 41',sum(case when week_number=42 then 1 else 0 end)as'week 42',sum(case when week_number=43 then 1 else 0 end)as'week 43',sum(case when week_number=44 then 1 else 0 end)as'week 44',sum(case when week_number=45 then 1 else 0 end)as'week 45',sum(case when week_number=46 then 1 else 0 end)as'week 46',sum(case when week_number=47 then 1 else 0 end)as'week 47',sum(case when week_number=48 then 1 else 0 end)as'week 48',sum(case when week_number=49 then 1 else 0 end)as'week 49',sum(case when week_number=50 then 1 else 0 end)as'week 50',sum(case when week_number=51 then 1 else 0 end)as'week 51',sum(case when week_number=52 then 1 else 0 end)as'week 52',sum(case when week_number=53 then 1 else 0 end)as'week 53',sum(case when week_number=54 then 1 else 0 end)as'week 54',sum(case when week_number=55 then 1 else 0 end)as'week 55',sum(case when week_number=56 then 1 else 0 end)as'week 56',sum(case when week_number=57 then 1 else 0 end)as'week 57',sum(case when week_number=58 then 1 else 0 end)as'week 58',sum(case when week_number=59 then 1 else 0 end)as'week 59',sum(case when week_number=60 then 1 else 0 end)as'week 60',sum(case when week_number=61 then 1 else 0 end)as'week 61',sum(case when week_number=62 then 1 else 0 end)as'week 62',sum(case when week_number=63 then 1 else 0 end)as'week 63',sum(case when week_number=64 then 1 else 0 end)as'week 64',sum(case when week_number=65 then 1 else 0 end)as'week 65',sum(case when week_number=66 then 1 else 0 end)as'week 66',sum(case when week_number=67 then 1 else 0 end)as'week 67',sum(case when week_number=68 then 1 else 0 end)as'week 68',sum(case when week_number=69 then 1 else 0 end)as'week 69',sum(case when week_number=70 then 1 else 0 end)as'week 70',sum(case when week_number=71 then 1 else 0 end)as'week 71',sum(case when week_number=72 then 1 else 0 end)as'week 72',sum(case when week_number=73 then 1 else 0 end)as'week 73',sum(case when week_number=74 then 1 else 0 end)as'week 74',sum(case when week_number=75 then 1 else 0 end)as'week 75',sum(case when week_number=76 then 1 else 0 end)as'week 76',sum(case when week_number=77 then 1 else 0 end)as'week 77',sum(case when week_number=78 then 1 else 0 end)as'week 78',sum(case when week_number=79 then 1 else 0 end)as'week 79',sum(case when week_number=80 then 1 else 0 end)as'week 80',sum(case when week_number=81 then 1 else 0 end)as'week 81',sum(case when week_number=82 then 1 else 0 end)as'week 82',sum(case when week_number=83 then 1 else 0 end)as'week 83',sum(case when week_number=84 then 1 else 0 end)as'week 84',sum(case when week_number=85 then 1 else 0 end)as'week 85',sum(case when week_number=86 then 1 else 0 end)as'week 86',sum(case when week_number=87 then 1 else 0 end)as'week 87',sum(case when week_number=88 then 1 else 0 end)as'week 88',sum(case when week_number=89 then 1 else 0 end)as'week 89',sum(case when week_number=90 then 1 else 0 end)as'week 90',sum(case when week_number=91 then 1 else 0 end)as'week 91',sum(case when week_number=92 then 1 else 0 end)as'week 92',sum(case when week_number=93 then 1 else 0 end)as'week 93',sum(case when week_number=94 then 1 else 0 end)as'week 94',sum(case when week_number=95 then 1 else 0 end)as'week 95',sum(case when week_number=96 then 1 else 0 end)as'week 96',sum(case when week_number=97 then 1 else 0 end)as'week 97',sum(case when week_number=98 then 1 else 0 end)as'week 98',sum(case when week_number=99 then 1 else 0 end)as'week 99',sum(case when week_number=100 then 1 else 0 end)as'week 100',sum(case when week_number=101 then 1 else 0 end)as'week 101',sum(case when week_number=102 then 1 else 0 end)as'week 102',sum(case when week_number=103 then 1 else 0 end)as'week 103',sum(case when week_number=104 then 1 else 0 end)as'week 104',sum(case when week_number=105 then 1 else 0 end)as'week 105',sum(case when week_number=106 then 1 else 0 end)as'week 106',sum(case when week_number=107 then 1 else 0 end)as'week 107',sum(case when week_number=108 then 1 else 0 end)as'week 108',sum(case when week_number=109 then 1 else 0 end)as'week 109',sum(case when week_number=110 then 1 else 0 end)as'week 110',sum(case when week_number=111 then 1 else 0 end)as'week 111',sum(case when week_number=112 then 1 else 0 end)as'week 112',sum(case when week_number=113 then 1 else 0 end)as'week 113',sum(case when week_number=114 then 1 else 0 end)as'week 114',sum(case when week_number=115 then 1 else 0 end)as'week 115',sum(case when week_number=116 then 1 else 0 end)as'week 116',sum(case when week_number=117 then 1 else 0 end)as'week 117',sum(case when week_number=118 then 1 else 0 end)as'week 118',sum(case when week_number=119 then 1 else 0 end)as'week 119',sum(case when week_number=120 then 1 else 0 end)as'week 120',sum(case when week_number=121 then 1 else 0 end)as'week 121',sum(case when week_number=122 then 1 else 0 end)as'week 122',sum(case when week_number=123 then 1 else 0 end)as'week 123',sum(case when week_number=124 then 1 else 0 end)as'week 124',sum(case when week_number=125 then 1 else 0 end)as'week 125',sum(case		



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_done

operation\_analytic

sakila

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

Port: 3306

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Connector: MySQL Community Server - GPL V8.0.33

Object Info Session

events

Limit to 1000 rows

26 sum(case when week\_number=17 then 1 else 0 end)as'week 17',

27 sum(case when week\_number=18 then 1 else 0 end)as'week 18',

28 from(

29 select m.user\_id,m.login\_week,n.first,m.login\_week-first as week\_number

30 from

31 (select user\_id,extract(week from occurred\_at)as login\_week from events group by 1,2)m,

32 (select user\_id,min(extract(week from occurred\_at))as first from events group by 1)n

33 where m.user\_id=n.user\_id

Result Grid

week numbers	week 0	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week 14	week 15	week 16
17	443	270	196	167	139	129	122	105	105	101	97	89	90	100	76	62	57
18	517	213	157	127	105	104	109	89	87	88	82	87	97	86	77	67	55
19	456	140	98	88	71	67	64	58	67	59	53	48	49	31	41	38	2
20	341	105	85	67	52	41	37	32	39	41	40	24	20	21	25	0	0
21	317	86	64	42	43	40	44	43	39	28	31	24	19	19	2	0	0
22	325	97	72	53	43	34	34	35	29	29	29	27	24	1	0	0	0
23	321	83	56	44	35	34	31	31	32	24	19	19	0	0	0	0	0
24	333	77	62	47	41	32	32	34	22	20	18	0	0	0	0	0	0
25	298	65	48	43	36	32	29	28	21	17	2	0	0	0	0	0	0
26	280	56	36	21	26	20	15	16	11	0	0	0	0	0	0	0	0

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	15:29:07	SELECT * FROM events events LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.016 sec
11	15:29:10	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
12	15:29:34	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
13	15:29:34	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	Error Code: 1054. Unknown column 'occurred_at' in 'field list'	0.000 sec
14	15:30:17	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	Error Code: 1054. Unknown column 'firsteventevents' in 'group statement'	0.000 sec
15	15:30:36	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	13 row(s) returned	1.359 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## 4.calculate the weekly engagement per device?

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_done

operation\_analytic

sakila

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

Port: 3306

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Connector: MySQL Community Server - GPL V8.0.33

Object Info Session

events

Limit to 1000 rows

38 select extract(week from occurred\_at)as'week numbers',

39 count(distinct case when device in('dell inspiron notebook')then user\_id else null end)as 'dell inspiron notebook',

40 count(distinct case when device in('iphone 5')then user\_id else null end)as 'iphone 5',

41 count(distinct case when device in('iphone 4s')then user\_id else null end)as 'iphone 4s',

42 count(distinct case when device in('windows surface')then user\_id else null end)as 'windows surface',

43 count(distinct case when device in('macbook air')then user\_id else null end)as 'macbook air',

44 count(distinct case when device in('iphone 5s')then user\_id else null end)as 'iphone 5s',

45 count(distinct case when device in('macbook pro')then user\_id else null end)as 'macbook pro',

46 count(distinct case when device in('kindle fire')then user\_id else null end)as 'kindle fire',

47 count(distinct case when device in('ipad mini')then user\_id else null end)as 'ipad mini',

48 count(distinct case when device in('nexus 7')then user\_id else null end)as 'nexus 7',

49 count(distinct case when device in('nexus 5')then user\_id else null end)as 'nexus 5',

50 count(distinct case when device in('samsung galaxy s4')then user\_id else null end)as 'samsung galaxy s4',

51 count(distinct case when device in('lenovo thinkpad')then user\_id else null end)as 'lenovo thinkpad',

Result Grid

week numbers	iphone 5	iphone 4s	windows surface	macbook air	iphone 5s	macbook pro	kindle fire	ipad mini	nexus 7	nexus 5	samsung galaxy s4	lenovo thinkpad	samsung galaxy s4
48	14	3	32	30	100	4	10	12	27	38	58	38	51
49	38	4	83	49	187	22	19	20	59	56	124	56	11
50	29	10	80	56	159	12	24	30	63	61	128	61	11

Result 4 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
14	15:30:17	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	Error Code: 1054. Unknown column 'firsteventevents' in 'group statement'	0.000 sec
15	15:30:36	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as'week 0', sum(case when...	13 row(s) returned	1.359 sec / 0.000 sec
16	15:34:03	select extract(week from occurred_at)as'week numbers', count(distinct case when device in('dell inspiron notebook...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
17	15:34:31	select extract(week from occurred_at)as'week numbers', count(distinct case when device in('dell inspiron notebook...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
18	15:47:20	select extract(week from occurred_at)as'week numbers', count(distinct case when device in('dell inspiron notebook...	Error Code: 1054. Unknown column 'occurred_at' in 'field list'	0.000 sec
19	15:47:35	select extract(week from occurred_at)as'week numbers', count(distinct case when device in('dell inspiron notebook...	13 row(s) returned	0.938 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_done

operation\_analytcs

skala

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

Port: 3306

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Server

MySQL Community Server - GPL

Connector: VC++ 8.0.33

Object Info Session

events

Limit to 1000 rows

SQL

```

50 count(distinct case when device in('samsung galaxy s4') then user_id else null end)as 'samsung galaxy s4',
51 count(distinct case when device in('lenovo thinkpad') then user_id else null end)as 'lenovo thinkpad',
52 count(distinct case when device in('samsung galaxy s4') then user_id else null end)as 'samsung galaxy s4',
53 count(distinct case when device in('lenovo thinkpad') then user_id else null end)as 'lenovo thinkpad',
54 count(distinct case when device in('samsung galaxy tablet') then user_id else null end)as 'samsung galaxy tablet',
55 count(distinct case when device in('asus chromebook') then user_id else null end)as 'asus chromebook',
56 count(distinct case when device in('htc one') then user_id else null end)as 'htc one',
57 count(distinct case when device in('nokia lumia 365') then user_id else null end)as 'nokia lumia 365',
58 count(distinct case when device in('samsung galaxy note') then user_id else null end)as 'samsung galaxy note',
59 count(distinct case when device in('mac mini') then user_id else null end)as 'mac mini',
60 count(distinct case when device in('hp pavilion desktop') then user_id else null end)as 'hp pavilion desktop',
61 count(distinct case when device in('dell inspiron desktop') then user_id else null end)as 'dell inspiron desktop',
62 count(distinct case when device in('ipad air') then user_id else null end)as 'ipad air',
63 count(distinct case when device in('amazon fire phone') then user_id else null end)as 'amazon fire phone'.

```

Result Grid

	phone 14	phone 4s	windows surface	macbook air	iphone 5s	macbook pro	kindle fire	ipad mini	nexus 7	nexus 5	samsung galaxy s4	lenovo thinkpad	samsung galaxy s4
48	14	3	32	30	100	4	10	12	27	38	58	38	58
69	38	4	83	49	187	22	19	20	59	56	124	56	124
89	29	10	80	56	199	12	24	30	63	61	128	61	128

Output

Action Output

#	Time	Action	Message	Duration / Fetch
14	15:30:17	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as week 0, sum(case when...	Error Code: 1054. Unknown column 'firsteventsevents' in 'group statement'	0.000 sec
15	15:30:36	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as week 0, sum(case when...	13 row(s) returned	1.359 sec / 0.000 sec
16	15:34:03	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
17	15:34:31	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
18	15:47:20	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1054. Unknown column 'occured_at' in 'field list'	0.000 sec
19	15:47:35	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	13 row(s) returned	0.938 sec / 0.000 sec

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

events

Tables

Views

Stored Procedures

Functions

ig\_done

operation\_analytcs

skala

sys

users

world

Administration Schemas

Information

Connection Details

Local instance MySQL80

Host: localhost

Port: 3306

User: root

SSL: TLS\_AES\_256\_GCM\_SHA384

Server

MySQL Community Server - GPL

Connector: VC++ 8.0.33

Object Info Session

events

Limit to 1000 rows

SQL

```

61 count(distinct case when device in('dell inspiron desktop') then user_id else null end)as 'dell inspiron desktop',
62 count(distinct case when device in('ipad air') then user_id else null end)as 'ipad air',
63 count(distinct case when device in('amazon fire phone') then user_id else null end)as 'amazon fire phone',
64 count(distinct case when device in('nexus 10') then user_id else null end)as 'nexus 10'
65 from events
66 where event_type='engagement'
67 group by 1
68 order by 1;

```

Result Grid

	week numbers	dell inspiron notebook	iphone 5s	iphone 4s	windows surface	macbook air	iphone 5s	macbook pro	kindle fire	ipad mini	nexus 7	nexus 5	samsung galaxy s4
17	28	48	14	3	32	30	100	4	10	12	27	38	58
18	50	69	38	4	83	49	187	22	19	20	59	56	124
19	54	89	29	10	80	56	199	12	24	30	63	61	128
20	62	79	36	11	75	59	181	13	23	19	67	62	124
21	59	92	32	15	71	47	167	19	19	19	63	61	128
22	68	77	31	9	102	51	164	13	21	30	67	74	128
23	73	102	37	6	82	54	174	19	23	21	58	66	128
24	61	97	40	15	103	54	167	18	21	31	58	66	128
25	73	85	26	14	70	42	171	13	16	31	59	64	128
26	53	90	34	11	85	55	167	12	21	29	52	72	128

Output

Action Output

#	Time	Action	Message	Duration / Fetch
14	15:30:17	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as week 0, sum(case when...	Error Code: 1054. Unknown column 'firsteventsevents' in 'group statement'	0.000 sec
15	15:30:36	select first as 'week numbers', sum(case when week_number=0 then 1 else 0 end)as week 0, sum(case when...	13 row(s) returned	1.359 sec / 0.000 sec
16	15:34:03	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
17	15:34:31	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
18	15:47:20	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	Error Code: 1054. Unknown column 'occured_at' in 'field list'	0.000 sec
19	15:47:35	select extractweek from occurred_atas week numbers, count(distinct case when device in('dell inspiron noteb...	13 row(s) returned	0.938 sec / 0.000 sec

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## 5. Calculate the email engagement metrics?

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to calculate email engagement metrics from the `email_events` table. The query uses conditional aggregation to calculate weekly digest rate, email open rate, email clickthrough rate, and reengagement email rate, grouped by week.

```
1 SELECT * FROM email_events;
2
3 select week,
4 round((weekly_digest/total*100),2)as'weekly digest rate',
5 round((email_opens/total*100),2)as'email open rate',
6 round((email_clickthroughs/total*100),2)as'email clickthrough rate',
7 round((reengagement_emails/total*100),2)as'reengagement email rate'
8 from
9 (
10  select extract(week from occurred_at)as week,
11  count(case when action='sent_weekly_digest' then user_id else null end)as weekly_digest,
12  count(case when action='email_open' then user_id else null end)as email_opens,
13  count(case when action='email_clickthrough' then user_id else null end)as email_clickthroughs,
14  count(case when action='sent_reengagement_email' then user_id else null end)as reengagement_emails,
15  count(user_id)as total
16 from email_events
17 group by 1
18 )sub
19 group by 1
20 order by 1;
```

The Result Grid shows the following data:

week	weekly digest rate	email open rate	email clickthrough rate	reengagement email rate
17	62.32	21.28	11.39	5.01
18	63.45	22.24	10.49	3.83
19	62.16	22.67	11.13	4.04
20	61.62	22.64	11.43	4.31

The Output pane shows the execution of the query, with messages indicating the number of rows returned and the duration of each step.

This screenshot is identical to the one above, showing the same SQL query and results in the MySQL Workbench interface.

This project operation analytics and investigating metric spike is bit tricky and bit challenging, this project improved my critical thinking, analytical skill and logical too. Whenever I got stuck I approached the ask doubt form and some trainity videos for concept doubts.