OPERATION ANALYTICS

&

INVESTGATING METRIC SPIKE

ADVANCED SQL PROJECT

BY - AKASH GAIKWAD

- **Description:**

Operation Analytics is the analysis done for the complete end to end operations of a company. With the help of this, the company then finds the areas on which it must improve upon. You work closely with the ops team, support team, marketing team, etc and help them derive insights out of the data they collect.

Being one of the most important parts of a company, this kind of analysis is further used to predict the overall growth or decline of a company's fortune. It means better automation, better understanding between cross-functional teams, and more effective workflows.

Investigating metric spike is also an important part of operation analytics as being a Data Analyst you must be able to understand or make other teams understand questions like- Why is there a dip in daily engagement? Why have sales taken a dip? Etc. Questions like these must be answered daily and for that it's very important to investigate metric spike.

You are working for a company like Microsoft designated as Data Analyst Lead and is provided with different data sets, tables from which you must derive certain insights out of it and answer the questions asked by different departments.

- **Case Study 1 (Job Data):**

Below is the structure of the table with the definition of each column**:**

- **Table-1:** job_data
  - job_id: unique identifier of jobs
  - actor_id: unique identifier of actor
  - event: decision/skip/transfer
  - language: language of the content
  - time_spent: time spent to review the job in seconds
  - org: organization of the actor
  - ds: date in the yyyy/mm/dd format. It is stored in the form of text and we use presto to run. no need for date function

Use the dataset and solve the following Questions:

1) **Number of jobs reviewed:** Number of jobs reviewed over time.
   Your task: Calculate the number of jobs reviewed per hour per day for November 2020?
2) **Throughput:** It is the no. of events happening per second.
   Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
3) **Percentage share of each language:** Share of each language for different contents.
   Your task: Calculate the percentage share of each language in the last 30 days?
4) **Duplicate rows:** Rows that have the same value present in them.
   Your task: Let's say you see some duplicate rows in the data. How will you display duplicates from the table?

- **Tech Stack Used:**

1) MYSQL VERSION 8.0: Software used to fetch the details from the database.
2) MS EXCEL: Microsoft Excel used for data analysis.

- **Insights and Solution:**

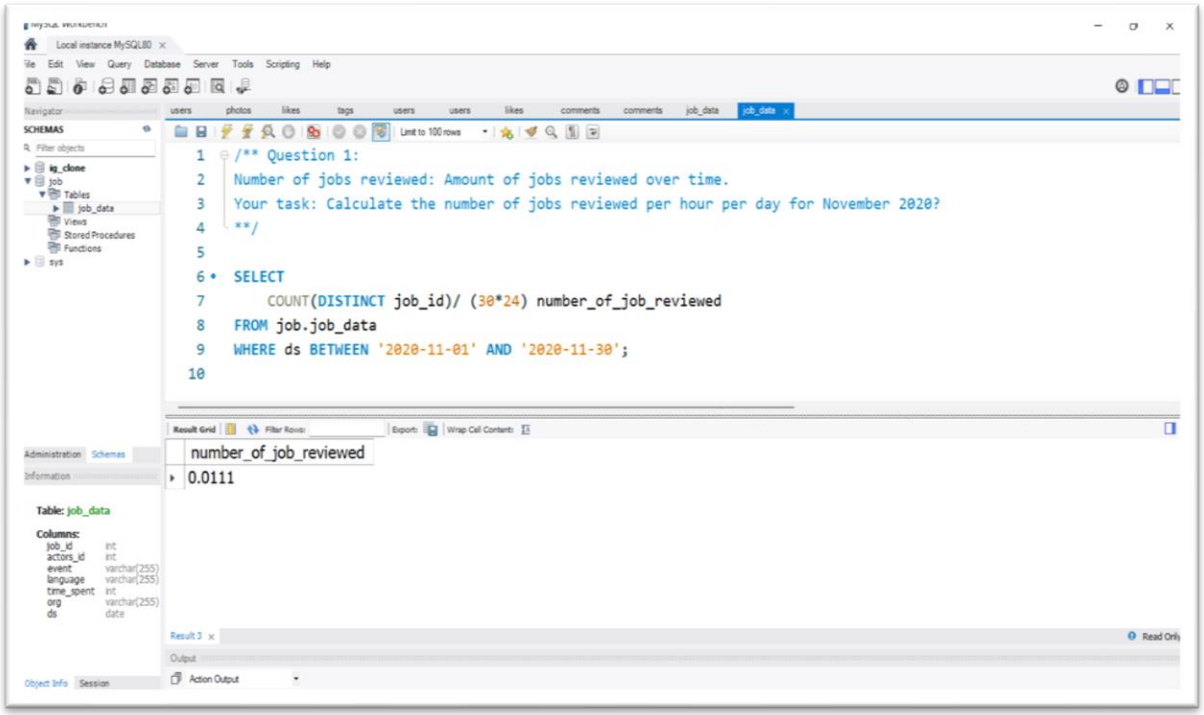
❖ Table Creation:

```
create table job_data
(
job_id int,
actors_id int,
event varchar(255),
language varchar(255),
time_spent int,
org varchar(255),
ds date
);
```

❖ Data Insert into Table Query:

```
insert into job_data (job_id,
actors_id, event, language, time_spent, org, ds)
values
(21,1001,'skip','English',15,'A','2020-11-30'),
(22,1006,'transfer', 'Arabic',25,'B','2020-11-30'),
(23,1003,'decision','Persian',20,'C','2020-11-29'),
(24,1005,'transfer','Persian',22,'D','2020-11-28'),
(25,1002,'decision','Hindi', 11,'B','2020-11-28'),
(26,1007, 'decision', 'French', 104,'D', '2020-11-27'),
(27, 1004, 'skip', 'Persian', 56, 'A', '2020-11-26'),
(28,1008, 'transfer', 'Italian', 45,'C','2020-11-25'),
(28, 1008, 'transfer', 'Italian', 45,'C','2020-11-25');
```

1) **Number of jobs reviewed:** Number of jobs reviewed over time.
Your task: Calculate the number of jobs reviewed per hour per day for November 2020?

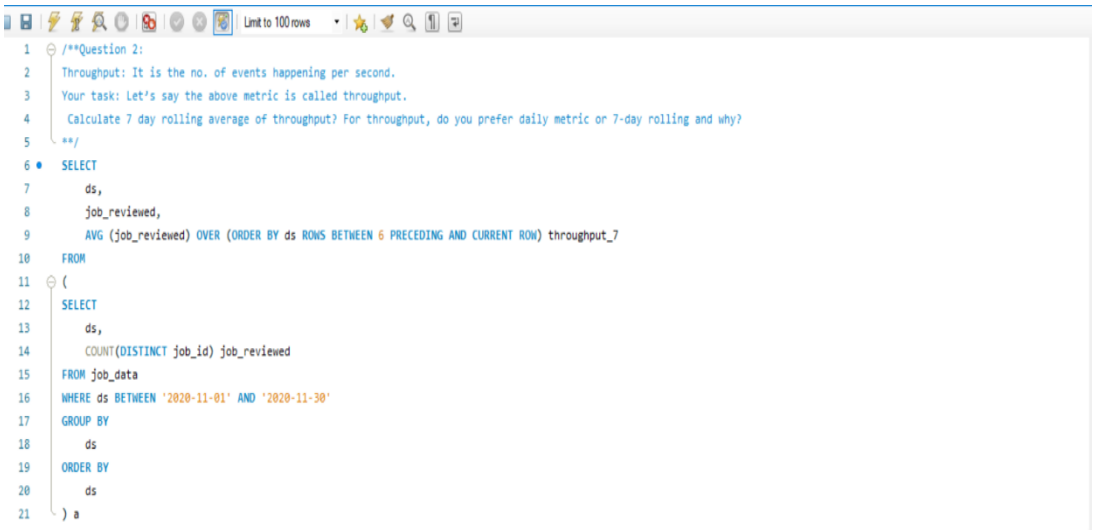


IMG.1 Query and Answer

Procedure:

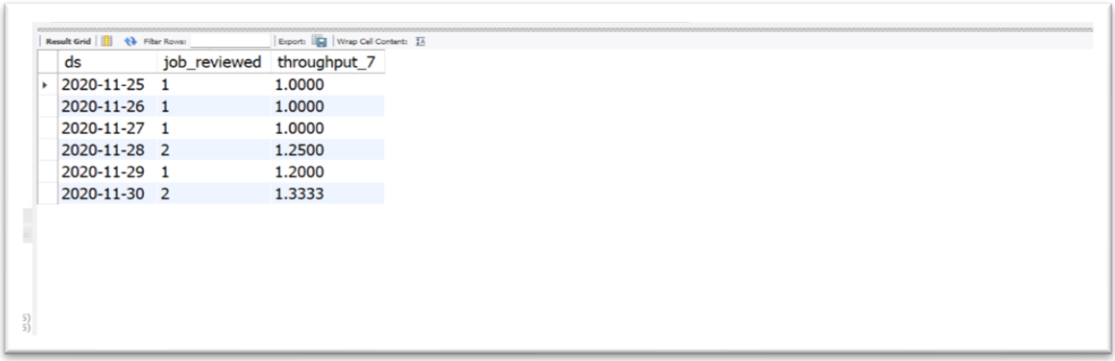
For calculating the number of jobs reviewed per hour per day for November 2020, use the SQL function COUNT and DISTINCT. The DISTINCT use in the job_id column for getting the only unique records. Then used the WHERE clause to get records only November.

2) **Throughput:** It is the no. of events happening per second.
Your task: Let's say the above metric is called throughput. Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?



```
/**Question 2:
Throughput: It is the no. of events happening per second.
Your task: Let's say the above metric is called throughput.
 Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?
**/
SELECT
    ds,
    job_reviewed,
    AVG (job_reviewed) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) throughput_7
FROM
(
SELECT
    ds,
    COUNT(DISTINCT job_id) job_reviewed
FROM job_data
WHERE ds BETWEEN '2020-11-01' AND '2020-11-30'
GROUP BY
    ds
ORDER BY
    ds
) a
```

IMG.2 Query



ds	job_reviewed	throughput_7
2020-11-25	1	1.0000
2020-11-26	1	1.0000
2020-11-27	1	1.0000
2020-11-28	2	1.2500
2020-11-29	1	1.2000
2020-11-30	2	1.3333

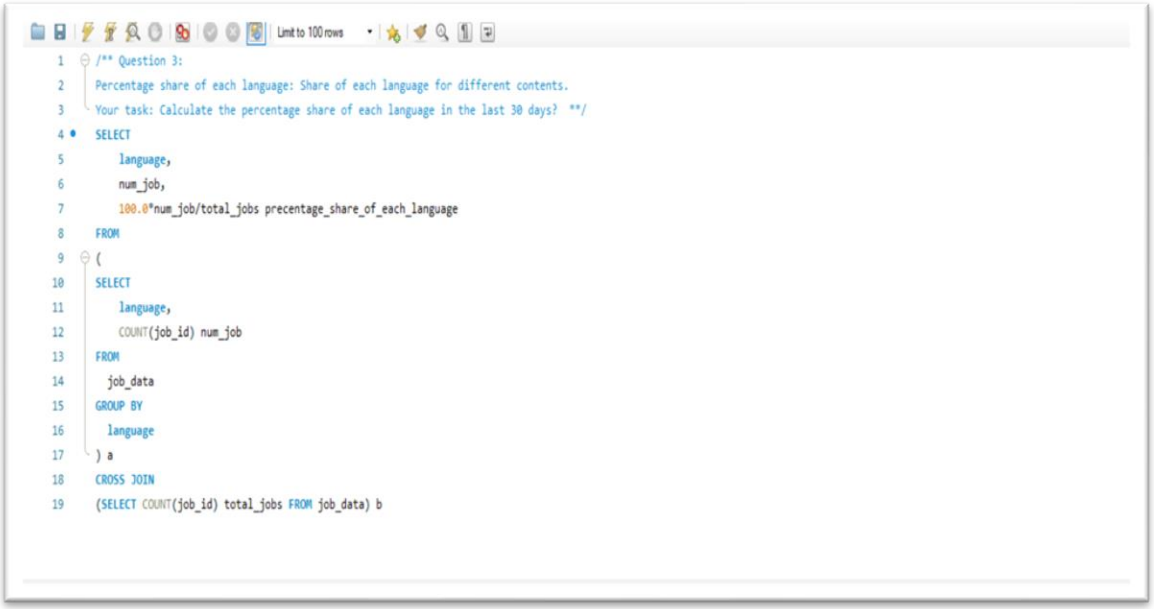
IMG 3. Answer

Procedure:

For calculating 7 days average , Firstly I used the ds and job_id columns from job_data table then use the COUNT function for calculating the count of job_id and the DISTINCT also used for job_id column to get the unique Job_id and avoid the duplicate job_id then uses WHERE clause for get the job_id between '2020-11-01 to 2020-11-30', after that used the GROUP BY clause to get the record in group of date and ORDER BY clause used for to get the date in ascending order. then used the window function's for getting the throughput.

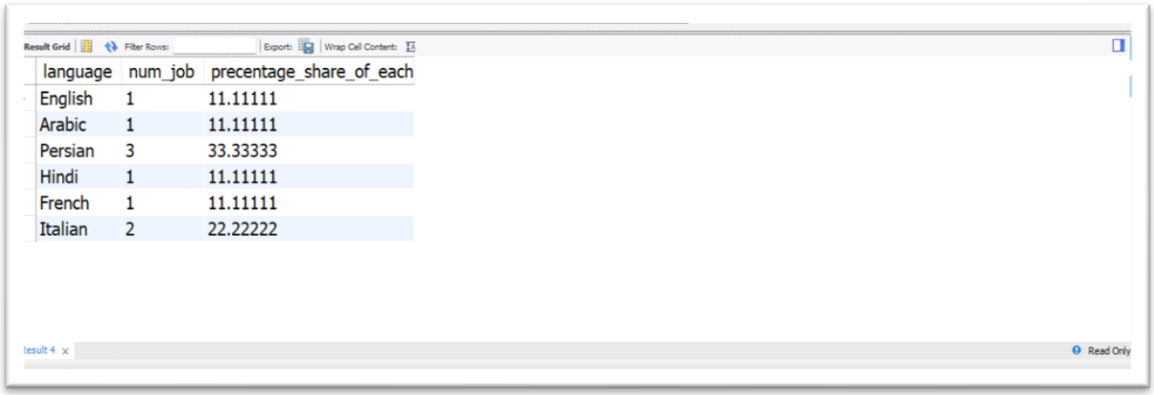
3) **Percentage share of each language:** Share of each language for different contents.

Your task: Calculate the percentage share of each language in the last 30 days?



```
/** Question 3:
Percentage share of each language: Share of each language for different contents.
Your task: Calculate the percentage share of each language in the last 30 days?  **/
SELECT
    language,
    num_job,
    100.0*num_job/total_jobs precentage_share_of_each_language
FROM
(
SELECT
    language,
    COUNT(job_id) num_job
FROM
  job_data
GROUP BY
  language
) a
CROSS JOIN
(SELECT COUNT(job_id) total_jobs FROM job_data) b
```

IMG 4. Query



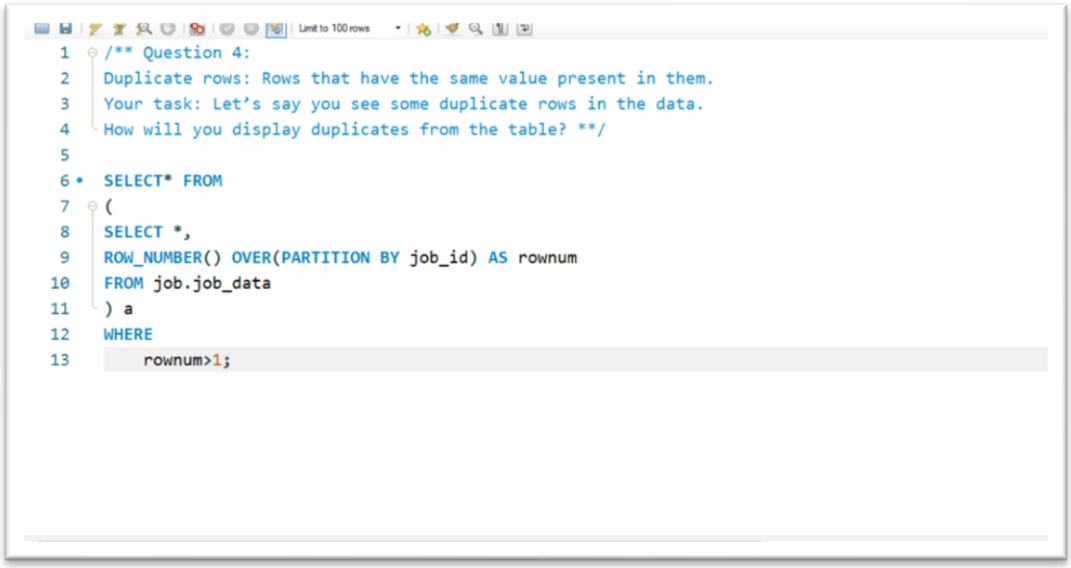
language	num_job	precentage_share_of_each
English	1	11.11111
Arabic	1	11.11111
Persian	3	33.33333
Hindi	1	11.11111
French	1	11.11111
Italian	2	22.22222

IMG 5. Answer

Procedure:

In this question calculated the share percentage of each language. In these queries used the subquery. In first subquery taken the language and job_id columns from the job_data table. Then used the COUNT function for counting the total job ids and they grouped using GROUP by function. then used the cross join on job_id column after that in main query, the subquery called.

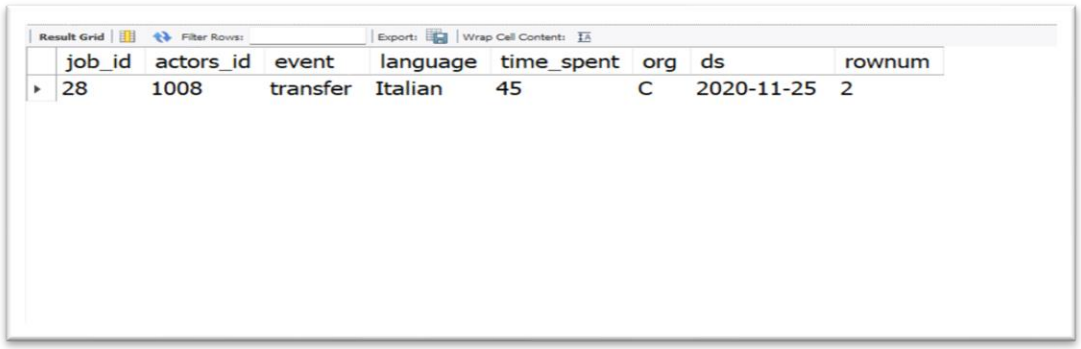
4) **Duplicate rows:** Rows that have the same value present in them. **Your task:** Let's say you see some duplicate rows in the data. How will you display duplicates from the table?



```
/** Question 4:
Duplicate rows: Rows that have the same value present in them.
Your task: Let's say you see some duplicate rows in the data.
How will you display duplicates from the table? **/

SELECT* FROM
(
SELECT *,
ROW_NUMBER() OVER(PARTITION BY job_id) AS rownum
FROM job.job_data
) a
WHERE
    rownum>1;
```

IMG.6 Query



job_id	actors_id	event	language	time_spent	org	ds	rownum
28	1008	transfer	Italian	45	C	2020-11-25	2

IMG.7 Answer

Procedure:

In this question calculated the duplicate row. In this used subquery in that the ROW_NUMBER function used for assign the sequentially values and PARTITION by job_id and gave the name for column is rownum and used the WHERE clause on rownum for get the value greater than 1.

- **Case study 2 (Investigating Metric Spike):**
  Following is the structure of the table with the definition of each column which is present in table

  - **Table-1:** users

    This table includes one row per user, with descriptive information about that user's account.
  - **Table-2:** events

    This table includes one row per event, where an event is an action that a user has taken. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.
  - **Table-3:** email_events

    This table contains events specific to the sending of emails. It is similar in structure to the events table above.

Use the dataset and solve the following Questions:

1) **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.
   Your task: Calculate the weekly user engagement?
2) **User Growth:** Number of users growing over time for a product.
   Your task: Calculate the user growth for product?
3) **Weekly Retention:** Users getting retained weekly after signing-up for a product.
   Your task: Calculate the weekly retention of users-sign up cohort?
4) **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
   Your task: Calculate the weekly engagement per device?
5) **Email Engagement:** Users engaging with the email service.
   Your task: Calculate the email engagement metrics?

- **Insights and Solution:**

1) **User Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service.

   Your task: Calculate the weekly user engagement?

```sql
/** Question 1:
 User Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service.
 Your task: Calculate the weekly user engagement?
 **/
SELECT
  EXTRACT(
    week
    FROM
      occurred_at
  ) weeknum,
  COUNT(DISTINCT user_id)
FROM
  tutorial.yammer_events
GROUP BY
  weeknum
```

IMG.8 Query

	weeknum	count
1	18	791
2	19	1244
3	20	1270
4	21	1341
5	22	1293
6	23	1366
7	24	1434
8	25	1462
9	26	1443
10	27	1477
11	28	1556
12	29	1556
13	30	1593
14	31	1685
15	32	1483
16	33	1438
17	34	1412
18	35	1442

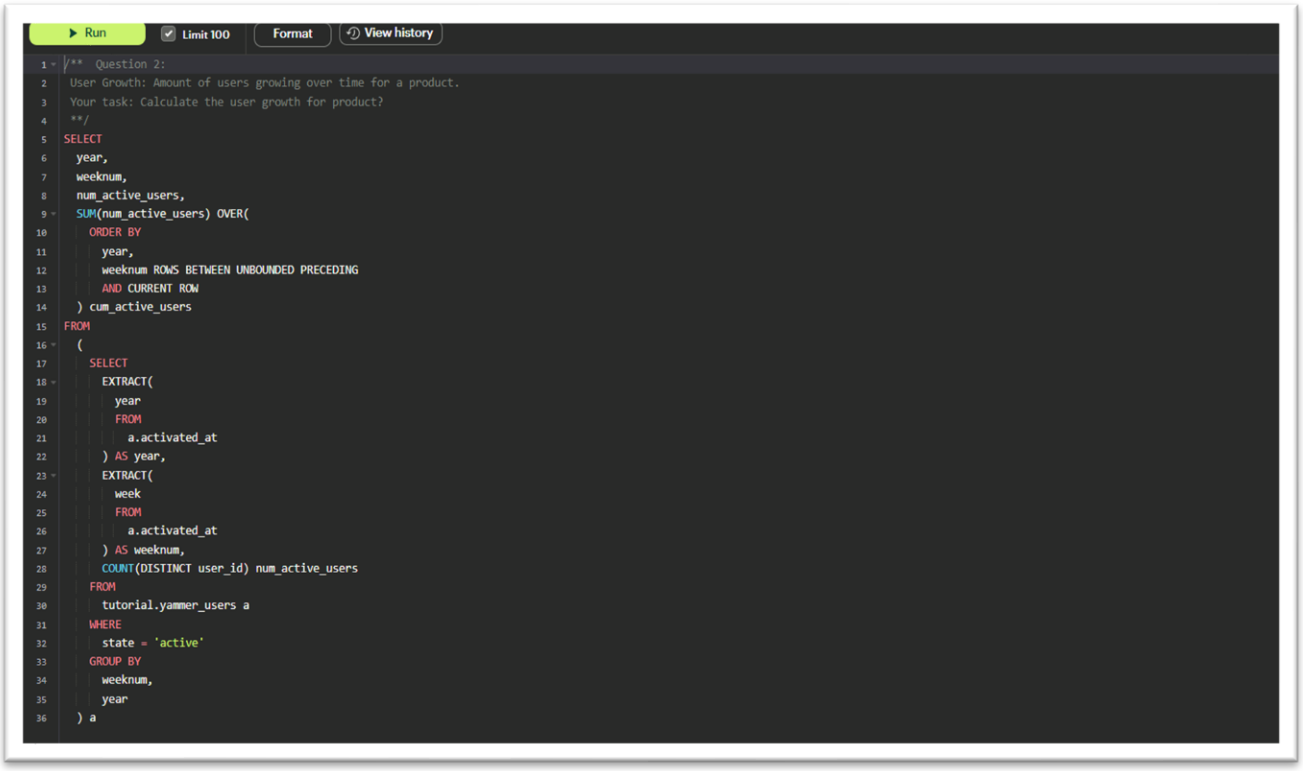
IMG.9 Answer

Procedure:

In this question calculated the weekly user engagement. selected the occurred at and user_id columns from the tutorial.yammer sevent table and EXTRACT function used on occurred_at column for find out the total weeks then COUNT function used for counting the user_id and GROUP BY function used for grouping the week.

2) **User Growth:** Number of users growing over time for a product.

   Your task: Calculate the user growth for product?



IMG.10 Query

year	weeknum	num_active_users	cum_active_users	year	weeknum	num_active_users	cum_active_users
2013	1	67	67	2013	44	92	2467
2013	2	29	96	2013	45	97	2564
2013	3	47	143	2013	46	94	2658
2013	4	36	179	2013	47	82	2740
2013	5	30	209	2013	48	103	2843
2013	6	48	257	2013	49	96	2939
2013	7	41	298	2013	50	117	3056
2013	8	39	337	2013	51	123	3179
2013	9	33	370	2013	52	104	3283
2013	10	43	413	2014	1	91	3374
2013	11	33	446	2014	2	122	3496
2013	12	32	478	2014	3	112	3608
2013	13	33	511	2014	4	113	3721
2013	14	40	551	2014	5	130	3851
2013	15	35	586	2014	6	132	3983
2013	16	42	628	2014	7	135	4118
2013	17	48	676	2014	8	127	4245
2013	18	48	724	2014	9	127	4372
2013	19	45	769	2014	10	135	4507
2013	20	55	824	2014	11	152	4659
2013	21	41	865	2014	12	132	4791
2013	22	49	914	2014	13	151	4942
2013	23	51	965	2014	14	161	5103
2013	24	51	1016	2014	15	166	5269
2013	25	46	1062	2014	16	165	5434
2013	26	57	1119	2014	17	176	5610
2013	27	57	1176	2014	18	172	5782
2013	28	52	1228	2014	19	160	5942
2013	29	71	1299	2014	20	186	6128
2013	30	66	1365	2014	21	177	6305
2013	31	69	1434	2014	22	186	6491
2013	32	66	1500	2014	23	197	6688
2013	33	73	1573	2014	24	198	6886
2013	34	70	1643	2014	25	222	7108
2013	35	80	1723	2014	26	210	7318
2013	36	65	1788	2014	27	199	7517
2013	37	71	1859	2014	28	223	7740
2013	38	84	1943	2014	29	215	7955
2013	39	92	2035	2014	30	228	8183
2013	40	81	2116	2014	31	234	8417
2013	41	88	2204	2014	32	189	8606
2013	42	74	2278	2014	33	250	8856
2013	43	97	2375	2014	34	259	9115

IMG.11 Answer

Procedure:

In this question calculated the user growth for product. In these Query used the subquery for extracting the year and weeks from activated_at column and the column is taken from tutorial.yammer_user table. The COUNT function used because to find out the total user_id. Then WHERE clause used for taken the active user from the state column. GROUP BY clause used for grouping the week and year and this SUBQUERY called in main QUERY.

3) **Weekly Retention:** Users getting retained weekly after signing-up for a product.

Your task: Calculate the weekly retention of users-sign up cohort?

```sql
/* Question 3:
Weekly Retention: Users getting retained weekly after signing-up for a product.
Your task: Calculate the weekly retention of users-sign up cohort?
**/
SELECT
COUNT(user_id),
SUM(CASE WHEN retention_week = 1 THEN 1 ELSE 0 END) AS week_1,
SUM(CASE WHEN retention_week = 2 THEN 1 ELSE 0 END) AS week_2
FROM
(
SELECT
a.user_id,
a.signup_week,
b.engagement_week,
b.engagement_week - a.signup_week AS retention_week
FROM
(SELECT DISTINCT user_id, EXTRACT(week FROM occurred_at) AS signup_week
FROM tutorial.yammer_events
WHERE event_type = 'signup_flow'
AND event_name = 'complete_signup'
AND EXTRACT(week FROM occurred_at) = 18
) a
LEFT JOIN
(
SELECT DISTINCT user_id,
  EXTRACT(week FROM occurred_at) AS engagement_week
FROM tutorial.yammer_events
WHERE event_type = 'engagement'
) b
ON a.user_id = b.user_id
ORDER BY
 a.user_id
) c
```

IMG.12 Query

	count	week_1	week_2
1	317	64	27

IMG.13 Answer

Procedure:

In this question calculated the weekly retention of users sign up cohort. In this query used SUBQUERY and LEFT JOIN also used DISINCT, EXTRACT, WHERE Clause, ORDER BY Clause for various operation.

4) **Weekly Engagement:** To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.

Your task: Calculate the weekly engagement per device?

```sql
/** Question 4:
 Weekly Engagement: To measure the activeness of a user. Measuring if the user finds quality in a product/service weekly.
 Your task: Calculate the weekly engagement per device?
 **/
SELECT
  EXTRACT(
    year
    FROM
      occurred_at
  ) AS year,
  EXTRACT(
    week
    FROM
      occurred_at
  ) AS week,
  device,
  COUNT(DISTINCT user_id)
FROM
  tutorial.yammer_events
WHERE
  event_type = 'engagement'
GROUP BY
  year,
  week,
  device
ORDER BY
  year,
  week,
  device
```

IMG.14 Query

year	week	device	count
2014	18	acer aspire desktop	10
2014	18	acer aspire notebook	21
2014	18	amazon fire phone	4
2014	18	asus chromebook	23
2014	18	dell inspiron desktop	21
2014	18	dell inspiron notebook	49
2014	18	hp pavilion desktop	15
2014	18	htc one	16
2014	18	ipad air	30
2014	18	ipad mini	21
2014	18	iphone 4s	21
2014	18	iphone 5	70
2014	18	iphone 5s	45
2014	18	kindle fire	6
2014	18	lenovo thinkpad	90
2014	18	macbook air	57
2014	18	macbook pro	154
2014	18	mac mini	8
2014	18	nexus 10	16
2014	18	nexus 5	43
2014	18	nexus 7	20
2014	18	nokia lumia 635	19
2014	18	samsumg galaxy tablet	8
2014	18	samsung galaxy note	7
2014	18	samsung galaxy s4	56
2014	18	windows surface	10
2014	19	acer aspire desktop	26
2014	19	acer aspire notebook	34
2014	19	amazon fire phone	9
2014	19	asus chromebook	42
2014	19	dell inspiron desktop	58
2014	19	dell inspiron notebook	78
2014	19	hp pavilion desktop	37
2014	19	htc one	19
2014	19	ipad air	52
2014	19	ipad mini	29
2014	19	iphone 4s	47
2014	19	iphone 5	114
2014	19	iphone 5s	70
2014	19	kindle fire	26
2014	19	lenovo thinkpad	155
2014	19	macbook air	119
2014	19	macbook pro	248
2014	19	mac mini	12
2014	19	nexus 10	30
2014	19	nexus 5	73
2014	19	nexus 7	29
2014	19	nokia lumia 635	34
2014	19	samsumg galaxy tablet	11
2014	19	samsung galaxy note	15
2014	19	samsung galaxy s4	80
2014	19	windows surface	10
2014	20	acer aspire desktop	22
2014	20	acer aspire notebook	40
2014	20	amazon fire phone	12
2014	20	asus chromebook	26
2014	20	dell inspiron desktop	36
2014	20	dell inspiron notebook	82
2014	20	hp pavilion desktop	40
2014	20	htc one	32
2014	20	ipad air	53
2014	20	ipad mini	37
2014	20	iphone 4s	40
2014	20	iphone 5	113
2014	20	iphone 5s	77
2014	20	kindle fire	20
2014	20	lenovo thinkpad	176
2014	20	macbook air	110
2014	20	macbook pro	261
2014	20	mac mini	19
2014	20	nexus 10	25
2014	20	nexus 5	84
2014	20	nexus 7	41
2014	20	nokia lumia 635	22
2014	20	samsumg galaxy tablet	6
2014	20	samsung galaxy note	11
2014	20	samsung galaxy s4	90
2014	20	windows surface	15
2014	21	acer aspire desktop	23
2014	21	acer aspire notebook	40
2014	21	amazon fire phone	10
2014	21	asus chromebook	39
2014	21	dell inspiron desktop	52
2014	21	dell inspiron notebook	84
2014	21	hp pavilion desktop	31
2014	21	htc one	27
2014	21	ipad air	54
2014	21	ipad mini	32
2014	21	iphone 4s	56
2014	21	iphone 5	128
2014	21	iphone 5s	75
2014	21	kindle fire	22
2014	21	lenovo thinkpad	177

IMG.15 Answer

Procedure:

In this question calculated the weekly engagement per device. Retrieved the occurred_at and user_id columns from tutorial.yammer_events. Used the EXTRACT function on occurred_at for getting years and weeks from this column. The COUNT function used for finding the total user ids. for taking only engagement user from event_type column used the WHERE clause also used the GROUP BY clause for grouping purpose. For getting the records in ascending order used the ORDER BY clause.

5) **Email Engagement:** Users engaging with the email service.

Your task: Calculate the email engagement metrics?

```sql
/** Question 5
 Email Engagement: Users engaging with the email service.
 Your task: Calculate the email engagement metrics?
 **/
SELECT
  100.0 * SUM(
    CASE
      WHEN email_cat = 'email_open' THEN 1
      ELSE 0
    END
  ) / SUM(
    CASE
      WHEN email_cat = 'email_sent' THEN 1
      ELSE 0
    END
  ) email_open_rate,
  100.0 * SUM(
    CASE
      WHEN email_cat = 'email_clicked' THEN 1
      ELSE 0
    END
  ) / SUM(
    CASE
      WHEN email_cat = 'email_sent' THEN 1
      ELSE 0
    END
  ) email_clicked_rate
FROM
  (
    SELECT
      *,
      CASE
        WHEN ACTION IN ('sent_weekly_digest', 'sent_reengagement_email') THEN 'email_sent'
        WHEN ACTION IN ('email_open') THEN 'email_open'
        WHEN ACTION IN ('email_clickthrough') THEN 'email_clicked'
      END AS email_cat
    FROM
      tutorial.yammer_emails
  ) a
```

IMG.16 Query

	email_open_rate	email_clicked_rate
1	33.5834	14.7899

IMG.17 Answer

Procedure:

In this question calculated the email engagement metrics. Used the CASE WHEN statement. This data retrieved from tutorial.yammer_emails.

- **Result:**

Operational analytics is the process of using data analysis and business intelligence to improve efficiency and streamline everyday operations in real time. A subset of business analytics, operational analytics is supported by data mining, artificial intelligence, and machine learning.

- **Operation Analytics Result:**

1) Number of jobs reviewed:

Number of jobs reviewed over time which helps to find the number of jobs viewed per hour. From the query we can conclude that a minimum of 1 job is searched by the users.

2) Number of Events happening per second:

Throughout is a number of events happening per second. Here we are using 7days rolling average which helps to find more accurate events. Here we can conclude that the rolling average is constantly increasing day by day. But on 29/11/2011 it slightly decreased.

3) Percentage share of each language:

Percentage Share of each language for different contents in 30 days. From the query above we can conclude that most of the users speak Persian language which has 33.33 percent.

4) Duplicate rows:

Rows that have the same value present in the list. In the user list job_id 28 appears two times who speak Italian language.

- **Investigating Metric Spike Result:**

1) User Engagement:

Measuring if the user finds quality in a product/service. From the query most users are engaged in week 31. And the remaining weeks have recorded the common engagement.

2) User Growth:

Number of users growing over time for a product. In the starting year of 2013 week 1, the number of users started from 67 and at the end of the year the total number of users was 3283. Moreover, at the end of 2014 the user number reached 9115. The user growth has been increasing day by day from last week.

3) Weekly Retention:

Users getting retained weekly after signing-up for a product. From the query we can conclude that every device has at least one user retention happening.

4) Weekly Engagement:

Measuring if the user finds quality in a product/service. Every device has at least 1 user engagement every week.

5) Email Engagement:

In email engagement metrics calculated the email open rate is 33.58 and email click rate is 14.78 so from this conclude that the email open rate is greater.

- **Knowledge gained:**
  - The dataset provide in the dashboard is in query format and which is suitable for MySQL
  - I did some searching and successfully modified the query and created a database in MySQL Workbench.
  - This has helped me to know more about creating and handling databases because I am a fresher in this field.
  - I have gained more hands-on experience in this project because, I have handled a real life like situation.
  - It helped me to understand the GROUP BY function well, as I was a little confused about that when I started the project.
  - I have gained knowledge about the aggregate functions in SQL like COUNT, SUM, etc.
  - In this project I got the hands-on experience in window function, joins, conditional statement, EXTRACT function, date time function.