Name: SATYAM GUPTA

College: Delhi Technological University

Graduation year-2022

Doubt & Resolution Time Analysis

- **Creating the database:**

```
CREATE DATABASE `codingninjas` ;
```

- **Creating tables :**

1) Doubts

```
CREATE TABLE `doubts` (
  `id` int NOT NULL,
  `user_id` int DEFAULT NULL,
  `course_id` int DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `content_type` text,
  `user_rating` int DEFAULT NULL,
  `state` text,
  PRIMARY KEY (`id`)
);
```

2) Activities

```
CREATE TABLE `activities` (
  `id` int NOT NULL,
  `user_id` int DEFAULT NULL,
  `doubt_id` int DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `activity_type` text,
  PRIMARY KEY (`id`)
);
```

- Now the data is imported from doubts.csv and activities.csv files to the doubts and activities table respectively using 'Table data import wizard'.
- After importing the doubts.csv the structure of the table looks like –

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content

id	user_id	course_id	created_at	content_type	user_rating	state
248188	280541	10	2020-01-01 00:58:22	Code Problem	5	resolved
248189	32227	12	2020-01-01 02:02:21	Multiple Problem	5	resolved
248190	357687	1	2020-01-01 02:14:37	Single_choice Problem	5	resolved
248191	486150	11	2020-01-01 02:15:15	No_submission Problem	4	resolved
248192	367920	13	2020-01-01 03:16:19	Datascience Problem	5	resolved
248193	382823	11	2020-01-01 03:24:23	Code Problem	5	resolved
248194	382823	11	2020-01-01 03:26:55	Code Problem	3	resolved
248195	38918	13	2020-01-01 03:33:30	VideoOffering	5	resolved
248196	367920	13	2020-01-01 03:38:42	Datascience Problem	5	resolved
248197	409753	13	2020-01-01 03:40:53	Datascience Problem	5	resolved
248198	486150	11	2020-01-01 03:58:37	VideoOffering	4	resolved
248199	498910	1	2020-01-01 04:02:49	No_submission Problem	5	resolved
248200	332079	10	2020-01-01 04:06:06	VideoOffering	5	resolved

- After importing the activities.csv the structure of the table looks like –

Result Grid | Filter Rows: | Edit: | Export/Import:

id	user_id	doubt_id	created_at	activity_type
2121357	120623	248193	2020-01-01 04:26:08	assign
2121358	120623	248193	2020-01-01 04:26:15	accept
2121359	165368	248199	2020-01-01 04:29:18	assign
2121360	165368	248199	2020-01-01 04:29:34	reject
2121399	265305	248191	2020-01-01 04:35:07	assign
2121400	265305	248191	2020-01-01 04:35:16	accept
2121401	265305	248194	2020-01-01 04:35:39	assign
2121402	265305	248194	2020-01-01 04:35:43	accept
2121403	265305	248198	2020-01-01 04:35:49	assign
2121404	265305	248198	2020-01-01 04:35:57	accept
2121423	165368	248190	2020-01-01 04:45:18	assign
2121425	165368	248190	2020-01-01 04:45:40	accept
2121428	265305	248194	2020-01-01 04:48:07	review_reso

Overall analysis around the resolution time

- For this we club both the tables using MySql query (inner join):

Query:

use codingninjas;

select activities.doubt_id,activities.created_at as endd, doubts.created_at as startt, timediff(activities.created_at,doubts.created_at) as resolution_time from doubts inner join activities ON doubts.id=activities.doubt_id

where activities.activity_type='resolve'

ORDER BY doubt_id;

- After clubbing and finding the resolution time it looks:

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

doubt_id	endd	startt	resolution_time
248188	2020-01-01 13:57:16	2020-01-01 00:58:22	12:58:54
248189	2020-01-01 12:57:32	2020-01-01 02:02:21	10:55:11
248190	2020-01-01 05:07:36	2020-01-01 02:14:37	02:52:59
248191	2020-01-02 14:39:49	2020-01-01 02:15:15	36:24:34
248192	2020-01-02 13:05:58	2020-01-01 03:16:19	33:49:39
248193	2020-01-01 05:05:30	2020-01-01 03:24:23	01:41:07
248194	2020-01-01 04:56:07	2020-01-01 03:26:55	01:29:12
248195	2020-01-08 20:00:43	2020-01-01 03:33:30	184:27:13
248196	2020-01-01 17:57:28	2020-01-01 03:38:42	14:18:46
248197	2020-01-02 04:23:33	2020-01-01 03:40:53	24:42:40
248198	2020-01-01 16:44:22	2020-01-01 03:58:37	12:45:45
248199	2020-01-02 07:02:53	2020-01-01 04:02:49	27:00:04
248200	2020-01-02 07:00:18	2020-01-01 04:06:06	26:54:12
248202	2020-01-02 17:13:27	2020-01-01 04:18:17	36:55:10

- We found that for 17 doubt_id we get two resolve activity type ,they are :

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

doubt_id	count(activity_type)
248780	2
249064	2
249419	2
249426	2
249073	2
250019	2
250109	2
250772	2
250929	2
250808	2
250133	2
251210	2
251968	2
252091	2
252558	2
252754	2
252535	2

select doubt_id,count(activity_type) from activities where activity_type='resolve'

group by doubt_id

having count(activity_type) >1;

The table extracted now looks like :

https://github.com/satyam-gupta-github/doubts_and_doubts_analysis/blob/main/final_resolution_time.csv

- Then I created a VIEW resol_time

create or replace VIEW resol_time

as

select activities.doubt_id,activities.created_at as endd, doubts.created_at as startt, timediff(activities.created_at,doubts.created_at) as resolution_time from doubts inner join activities ON doubts.id=activities.doubt_id

where activities.activity_type='resolve'

ORDER BY doubt_id;

- The maximum resolution time is

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

doubt_id	endd	startt	resolution_time
249316	2020-02-06 19:00:51	2020-01-02 18:51:59	838:59:59

*select * from resol_time*

order by resolution_time desc limit 1;

- The minimum resolution time is

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

doubt_id	endd	startt	resolution_time
249728	2020-01-03 13:29:36	2020-01-03 13:29:32	00:00:04

*select * from resol_time*

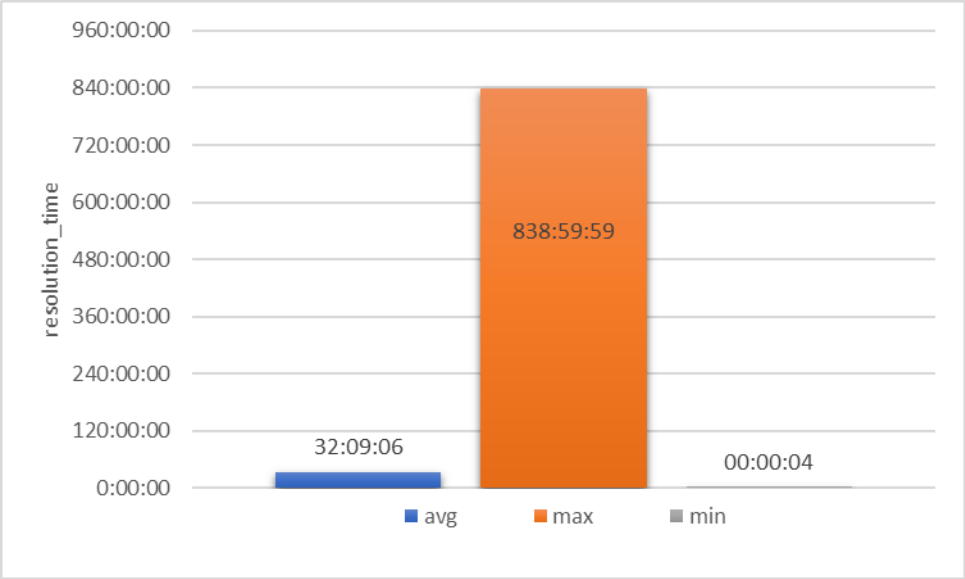
order by resolution_time limit 1;

- The average resolution time is

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

	average_resol_time
▶	32:09:06

- *SELECT cast(avg(resolution_time)as time) as average_resol_time from resol_time;*



An analysis on the effect of Resolution time on the user rating

- We created a view for this task to easily use it again and again.

  create or replace VIEW resol_timee

  as

  select timediff(activities.created_at,doubts.created_at) as resolution_time,doubts.user_rating as rating from doubts inner join activities ON doubts.id=activities.doubt_id

  where activities.activity_type='resolve'

  ORDER BY doubt_id;

- The overall average rating is

Result Grid | Filter Rows: | Export:

avg(rating)
4.5365

  select avg(rating) from resol_timee;

- The rating given to the highest resolution time is :

Result Grid | Filter Rows: | Export:

resolution_time	rating
838:59:59	1

  select resolution_time,rating from resol_timee

  order by resolution_time desc limit 1;

- The rating given to the lowest resolution time is :

Result Grid | Filter Rows:

resolution_time	rating
00:00:04	5

  select resolution_time,rating from resol_timee

  order by resolution_time limit 1;

- Now we will find the average resolution time for each rating :
  - Rate 1 (avg resolution time)

Result Grid | Filter Rows: | Export:

avg_resolution_time_for_rate_1
52:16:15.7056

SELECT SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as avg_resolution_time_for_rate_1 from resol_timee where rating =1;

- Rate 2 (avg resolution time)

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

avg_resolution_time_for_rate_2
74:22:47.3454

SELECT SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as avg_resolution_time_for_rate_2 from resol_timee where rating =2;

- Rate 3 (avg resolution time)

Result Grid | Filter Rows: | Export:

avg_resolution_time_for_rate_3
49:50:42.7080

SELECT SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as avg_resolution_time_for_rate_3 from resol_timee where rating =3;

- Rate 4 (avg resolution time)

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

avg_resolution_time_for_rate_4
44:26:50.8711

SELECT SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as avg_resolution_time_for_rate_4 from resol_timee where rating =4;

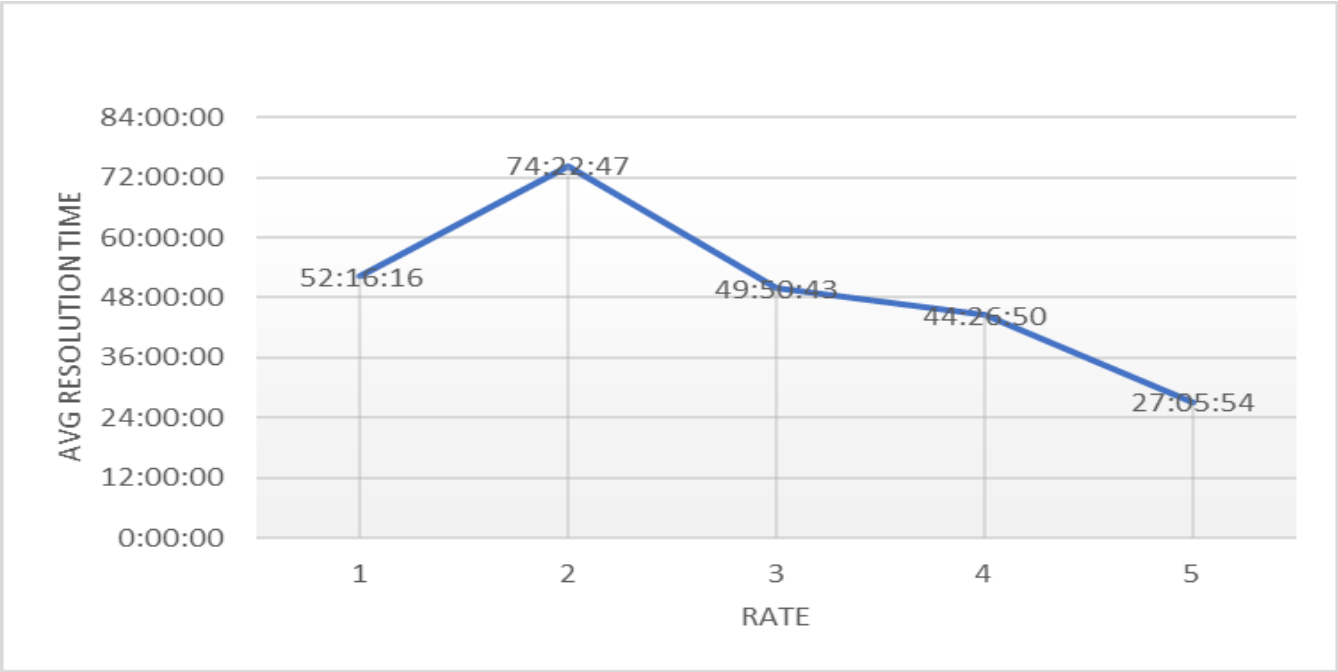
- Rate 5 (avg resolution time)

Result Grid | Filter Rows:

avg_resolution_time_for_rate_5
27:05:54.0944

SELECT SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as avg_resolution_time_for_rate_5 from resol_timee where rating =5;

Rating	1	2	3	4	5
Avg resolution time	52:16:16	74:22:47	49:50:43	44:26:51	27:05:54



Discussion : From the above graph we can conclude that the Rating given by the user is directly proportion to the resolution time .

The highest rating i.e. 5 is given to the one with less resolution time.

So we can say the resolution time is the major factor for the rating .

How many doubts are resolved by each TA?

```
create or replace view T
as
select activities.user_id,activities.doubt_id,activities.created_at,activity_type from
doubts inner join activities ON doubts.id=activities.doubt_id
where activities.activity_type='accept'
ORDER BY doubt_id asc,activities.created_at desc;

select T1.* from T T1 inner join (
select doubt_id,max(created_at) as max_date from T
group by doubt_id) T2
on T1.doubt_id=T2.doubt_id and T1.created_at=T2.max_date
order by T1.doubt_id;
select user_id as TA_id,count(doubt_id) as No_of_doubts from X
group by user_id
order by user_id ;
```

https://github.com/satyam-gupta-github/doubts_and_doubts_analysis/blob/main/TA%20ques3.csv

the above link contains the total doubts solved by each TA

Our performance in doubt resolution in different courses.

```
create or replace view CT

as

select   timediff(activities.created_at,doubts.created_at) as
resolution_time,content_type from doubts inner join activities ON
doubts.id=activities.doubt_id

where activities.activity_type='resolve'

ORDER BY doubt_id;


select content_type,SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) as
avg_time from CT

group by content_type

order by SEC_TO_TIME(AVG(TIME_TO_SEC(resolution_time))) ;
```

table : this table contain the content_type and its average resolution time.

Multiple Problem	16:31:28
Single_choice Problem	22:11:51
Frontend Problem	23:34:22
Fill_up Problem	25:29:58
SingleProblemOffering	26:25:06
No_submission Problem	29:51:16
Code Problem	32:13:40
Datascience Problem	34:45:27
Machine_learning Problem	36:46:55
NoteOffering	37:57:17
VideoOffering	43:58:29
ProjectsOffering	61:23:01

The maximum avg time is taken in ‘ProjectsOffering’ type , and the minimum in the ‘Multiple Problem’.

https://github.com/satyam-gupta-github/doubts_and_doubts_analysis/blob/main/QT%20final.csv


Multiple Problem
Single_choice Problem
Frontend Problem
Fill_up Problem
SingleProblemOffering
No_submission Problem
Code Problem
Datascience Problem
Machine_learning Problem
NoteOffering
VideoOffering
ProjectsOffering
Multiple Problem
Single_choice Problem
Frontend Problem
Fill_up Problem
SingleProblemO...
No_submission Problem
Code Problem
Datascience Problem
Machine_learning Problem
NoteOffering
VideoOffering
ProjectsOffering

Any other suggestions from your side

Normalisation of database will have systematic approach of decomposing tables to eliminate data redundancy(repition),removing duplicated data from the relation table.

I would make two separate tables for TA and Student which will have the TA_id and Student_id as the primary key respectively and doubt_id as foreign key in both.

It will make the accessibility and managing the data better without affecting the other table.