

## Pivoting rows into aggregated columns with CASE

### Business problem: Tracking user activity with frontend events

The design team recently redesigned the customer support page on the website, and they want to run an AB test to see how the newly designed page performs compared to the original. An AB test is a statistical test used frequently in data science to measure the impact changes have on specified metrics. The users will be randomly assigned into two groups: control and treatment. The users in control group will see the current customer support page, and the users in treatment will see the new page design. The analytics team needs to track user activity via frontend events (button clicking, page viewing, etc.) to inform the product team for future iterations. At the end of the experiment, the results of the control and treatment group will be compared to make a final product recommendation.

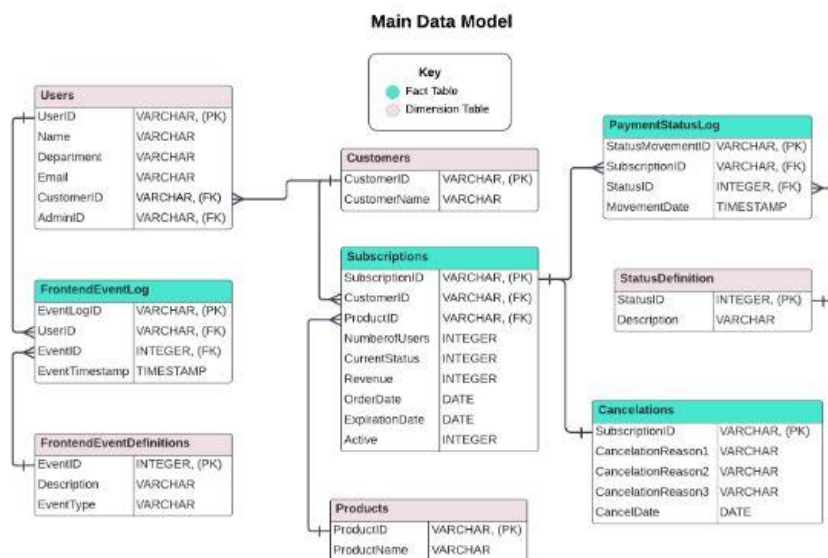
You decide that it will be important to track user activity and ticket submissions on the customer support page since they could be impacted, either positively or negatively, by the design changes. You choose the following events to track:

- When a user views the help center page: `ViewedHelpCenterPage`
- When a user clicks on the Frequently Asked Questions link: `ClickedFAQs`
- When a user clicks the contact customer support button: `ClickedContactSupport`
- When a user clicks the submit ticket button: `SubmittedTicket`

#### Task:

Using the `FrontendEventLog` table and CASE, count the number of times a user completes the following events:

- `ViewedHelpCenterPage` (eventid = 1)
  - `ClickedFAQs` (eventid = 2)
  - `ClickedContactSupport` (eventid = 3)
  - `SubmittedTicket` (eventid = 4)
- 
- Filter the events with `eventtype = 'Customer Support'` from the `frontendevedeventdefinitions` table to pull only the events related to customer support.



```

1  -- SQL code by John Uzoma
2  select
3      A.userid,
4      sum(case when A.eventid = 1 then 1 else 0 end) as VIEWEDHELPCENTERPAGE,
5      sum(case when A.eventid = 2 then 1 else 0 end) as CLICKEDFAQS,
6      sum(case when A.eventid = 3 then 1 else 0 end) as CLICKEDCONTACTSUPPORT,
7      sum(case when A.eventid = 4 then 1 else 0 end) as SUBMITTEDTICKET
8  from fronteventlog A
9  left join fronteventdefinitions B
10 on A.eventid = B.eventid
11 where B.eventtype = 'Customer Support'
12 group by A.userid

```

## Result

USERID	VIEWEDHELPCENTERPAGE	CLICKEDFAQS	CLICKEDCONTACTSUPPORT	SUBMITTEDTICKET
U172	1	0	1	1
U837	4	1	2	2