A/B TEST ON SITE

JAN 2024

Case, Join & Aggregated Pivot
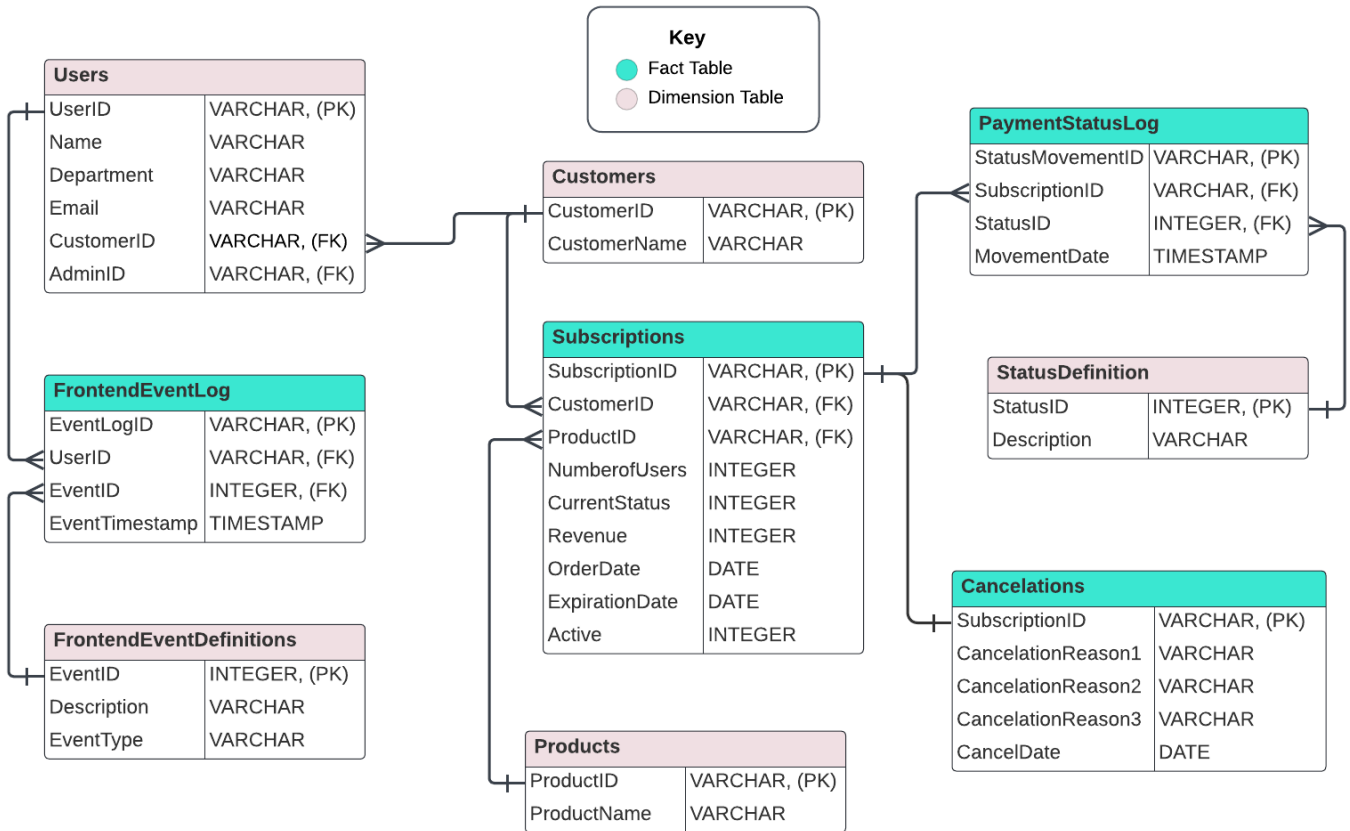
LinkedIn Coding Challenge

Intermediate

A design team is rolling out a new customer support page on their company's website and would like to run an A/B test to see how the newly designed page performs compared to the original. The control group users will see the original customer support page and the users in treatment will see the new page. In order to make a final recommendation to the product team based on the results, user activity must be tracked via frontend events (button clicking, page viewing, etc.).

Count the number of times a tag on the page fires for a frontend event related with Customer Support.

INPUT FORMAT

The source tables are **FRONTENDEVENTLOGS** and **FRONTENDEVENTDEFINITIONS**

Main Data Model

Key
- Fact Table
- Dimension Table

Users

UserID	VARCHAR, (PK)
Name	VARCHAR
Department	VARCHAR
Email	VARCHAR
CustomerID	VARCHAR, (FK)
AdminID	VARCHAR, (FK)

Customers

CustomerID	VARCHAR, (PK)
CustomerName	VARCHAR

PaymentStatusLog

StatusMovementID	VARCHAR, (PK)
SubscriptionID	VARCHAR, (FK)
StatusID	INTEGER, (FK)
MovementDate	TIMESTAMP

FrontendEventLog

EventLogID	VARCHAR, (PK)
UserID	VARCHAR, (FK)
EventID	INTEGER, (FK)
EventTimestamp	TIMESTAMP

Subscriptions

SubscriptionID	VARCHAR, (PK)
CustomerID	VARCHAR, (FK)
ProductID	VARCHAR, (FK)
NumberofUsers	INTEGER
CurrentStatus	INTEGER
Revenue	INTEGER
OrderDate	DATE
ExpirationDate	DATE
Active	INTEGER

StatusDefinition

StatusID	INTEGER, (PK)
Description	VARCHAR

FrontendEventDefinitions

EventID	INTEGER, (PK)
Description	VARCHAR
EventType	VARCHAR

Cancelations

SubscriptionID	VARCHAR, (PK)
CancelationReason1	VARCHAR
CancelationReason2	VARCHAR
CancelationReason3	VARCHAR
CancelDate	DATE

Products

ProductID	VARCHAR, (PK)
ProductName	VARCHAR

CONSTRAINTS

- The event tags and their corresponding event IDs from **FRONTENDEVENTLOG** are:
  - *VIEWEDHELPCENTERPAGE = 1*
  - *CLICKEDFAQS = 2*
  - *CLICKEDCONTACTSUPPORT = 3*
  - *SUBMITTEDTICKET = 4*
- Event type can be found in from the **FRONTENDEVENTDEFINITIONS** table

CODE SOLUTION

```sql
SELECT

logs.USERID,

SUM(CASE WHEN defs.EVENTID = 1 THEN 1 ELSE 0 END) AS VIEWEDHELPCENTERPAGE,

SUM(CASE WHEN defs.EVENTID = 2 THEN 1 ELSE 0 END) AS CLICKEDFAQS,

SUM(CASE WHEN defs.EVENTID = 3 THEN 1 ELSE 0 END) AS CLICKEDCONTACTSUPPORT,

SUM(CASE WHEN defs.EVENTID = 4 THEN 1 ELSE 0 END) AS SUBMITTEDTICKET

FROM frontendeventlog logs

JOIN frontendeventdefinitions defs

ON defs.EVENTID = logs.EVENTID

WHERE defs.EVENTTYPE ='Customer Support'

GROUP BY logs.USERID
```

SOLUTION PROCESS

- Case function: An individual CASE function for each desired column in order to create a pivot table
- Sum functions: Each CASE function must be wrapped in a SUM function, nesting all characters from CASE to END, in order to aggregate the results by its unique identifiers (UserID in this case)

OUTPUT

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