Interview Question

SQL

Customer Churn

DIFFICULTY LEVEL :HARD

Question From

Ankit Bansal Yotube Channel

CUSTOMER CHURN RATE

Customer churn refers to the rate at which customers stop doing business with a company or stop using its products or services.

It is a measure of customer loyalty and retention, and is often used as an indicator of a company's overall health and performance.

Customer churn rate is the percentage of customers who stop doing business with a company or stop using its products or services over a given period of time.
It is calculated by dividing the number of customers lost during the period by the total number of customers at the beginning of the period.

For example, if a company starts with 100 customers and loses 10 customers over a given month, the customer churn rate for that month would be 10%.

We are going to find customerchurn rate for following data on monhtly basis.

transactions Table

order_id	cust_id	order_date	amount
1	1	2020-01-15	150
2	1	2020-02-10	150
3	2	2020-01-16	150
4	2	2020-02-25	150
5	3	2020-01-10	150
6	3	2020-02-20	150
7	4	2020-01-20	150
8	5	2020-02-20	150

QUERY

```sql
WITH curr_next_cte AS (
SELECT cust_id
      ,order_date curr_order_date
      ,LEAD(order_date) OVER (PARTITION BY cust_id ORDER BY order_date)
       AS next_order_date
FROM transactions
),
count_cte AS (
SELECT MONTH(curr_order_date) month_no
      ,SUM(CASE WHEN DATEDIFF(MONTH,curr_order_date,next_order_date)=1
            THEN 0
             ELSE 1
       END) AS cust_counts
      ,COUNT(DISTINCT cust_id) AS total_custs
FROM curr_next_cte
GROUP BY MONTH(curr_order_date)
)
SELECT month_no+1 month_no
      ,CAST(cust_counts*100.0/total_custs AS DECIMAL(15,2))
       AS churn_rate
FROM count_cte
```

QUERY EXPLANATION

1.With curr_next_cte, we are simply obtaining the current and next order dates for corresponding customers.
We have used the LEAD function to retrieve the previous order date.

2.With count_cte, we are extracting the month from the current order date and calculating the count of customers.

To count the customers, we first flagged them with 0 or 1 and then summed them up. We also check if the difference between the next and current order date is 1 to calculate the count. Additionally, we count the number of distinct customers in that month using the

4.Finally, we are querying the month and calculating the churn rate.

output from curr_next_cte

cust_id	curr_order_date	next_order_date
1	2020-01-15	2020-02-10
1	2020-02-10	NULL
2	2020-01-16	2020-02-25
2	2020-02-25	NULL
3	2020-01-10	2020-02-20
3	2020-02-20	NULL
4	2020-01-20	NULL
5	2020-02-20	NULL

Output from CASE WHEN

cust_id	curr_order_date	next_order_date	month_no	flag
1	2020-01-15	2020-02-10	1	0
1	2020-02-10	NULL	2	1
2	2020-01-16	2020-02-25	1	0
2	2020-02-25	NULL	2	1
3	2020-01-10	2020-02-20	1	0
3	2020-02-20	NULL	2	1
4	2020-01-20	NULL	1	1
5	2020-02-20	NULL	2	1

Ouput from count_cte

month_no	cust_counts	total_custs
1	1	4
2	4	4

Final Output

month_no	churn_rate
2	25.00
3	100.00

Here we willl have churn rate in next month that 1 month prior to month fro which we are having count of customers.

Here for 3rd month we are having 100% churn rate beacuse we don't have data for 3rd month in our records.

BY MANISH KUMAR CHAUDHARY

THANK YOU

“Every moment is a fresh beginning.” –

T.S. Eliot

BY MANISH KUMAR CHAUDHARY

Interview Question

SQL

Customer Retention

DIFFICULTY LEVEL :HARD

Question From

Ankit Bansal

Yotube Channel

CUSTOMER RETENTION RATE

Customer retention is a measure of how well a business is able to keep its existing customers over a period of time.
It refers to the percentage of customers who continue to do business with a company after their initial purchase, and is an important metric for evaluating the overall health and success of a business.

Customer retention rate (CRR) is a metric that is used to measure customer retention.
It is calculated by taking the number of customers a business has at the end of a given period, subtracting the number of new customers acquired during that period, and dividing the result by the number of customers the business had at the beginning of the period.
This calculation provides a percentage that represents the percentage of customers who have been retained by the business over the period.

We are going to find customer retention rate for following data on monhtly basis.

transactions Table

order_id	cust_id	order_date	amount
1	1	2020-01-15	150
2	1	2020-02-10	150
3	2	2020-01-16	150
4	2	2020-02-25	150
5	3	2020-01-10	150
6	3	2020-02-20	150
7	4	2020-01-20	150
8	5	2020-02-20	150

QUERY

```sql
WITH curr_prev_cte AS (
SELECT cust_id
      ,order_date curr_order_date
      ,LAG(order_date) OVER (PARTITION BY cust_id ORDER BY order_date)
       AS prev_order_date
FROM transactions
),
count_cte AS (
SELECT MONTH(curr_order_date) month_no
      ,SUM(CASE WHEN DATEDIFF(MONTH,prev_order_date,curr_order_date)=1
            THEN 1
            ELSE 0
       END) AS cust_counts
      ,LAG(COUNT(DISTINCT cust_id)) OVER (ORDER BY MONTH(curr_order_date))
       AS total_prev_month_custs
FROM curr_prev_cte
GROUP BY MONTH(curr_order_date)
)
SELECT month_no
      ,CAST(cust_counts*100.0/total_prev_month_custs AS DECIMAL(15,2))
       AS retention_rate
FROM count_cte
```

QUERY EXPLANATION

1.With curr_prev_cte, we are simply obtaining the current and previous order dates for corresponding customers.
We have used the LAG function to retrieve the previous order date.

2.With count_cte, we are extracting the month from the current order date and calculating the count of customers.

To count the customers, we first flagged them with 1 or 0 and then summed them up. We also check if the difference between the previous and current order date is 1 to calculate the count. Additionally, we count the number of distinct customers in the previous month using the LAG function to calculate the retention rate.

4.Finally, we are querying the month and calculating the retention rate.

output from curr_prev_cte

cust_id	curr_order_date	prev_order_date
1	2020-01-15	NULL
1	2020-02-10	2020-01-15
2	2020-01-16	NULL
2	2020-02-25	2020-01-16
3	2020-01-10	NULL
3	2020-02-20	2020-01-10
4	2020-01-20	NULL
5	2020-02-20	NULL

Ouput from count_cte

month_no	cust_counts	total_prev_month_custs
1	0	NULL
2	3	4

Final Output

month_no	retention_rate
1	NULL
2	75.00

BY MANISH KUMAR CHAUDHARY

THANK YOU

The good life is one inspired by love and guided by knowledge.

Bertrand Russell

BY MANISH KUMAR CHAUDHARY

Interview Question

SQL

Data Analyst Spotify Case Study

Spotify

Question From
Ankit Bansal
Yotube Channel

PROBLEM STATEMENT

The activity table shows the app installed and app purchase activities for spotify along with country details

We solve following questions

1.Find total active users each day

2.Find total active users each week

3.Date wise total number of users who made the purchase same day they installed the app

4.Percentage of Paid Users in India, USA and ay other country should be tagged as others

5.Among all the users who installed the app on a given day, how many did in app puchased on the very next day -- day wise result

activity Table

user_id	event_name	event_date	country
1	app-installed	2022-01-01	India
1	app-purchase	2022-01-02	India
2	app-installed	2022-01-01	USA
3	app-installed	2022-01-01	USA
3	app-purchase	2022-01-03	USA
4	app-installed	2022-01-03	India
4	app-purchase	2022-01-03	India
5	app-installed	2022-01-03	SL
5	app-purchase	2022-01-03	SL
6	app-installed	2022-01-04	Pakistan
6	app-purchase	2022-01-04	Pakistan

1.Find total active users each day

QUERY

```sql
SELECT event_date
      ,COUNT(DISTINCT user_id) total_active_users
FROM activity
GROUP BY event_date
```

QUERY EXPANATION

1.For each event date we are simply counting the distinct users there are to get total active users

Here active users means either they have installed app on that date or else they purchased the app on that date. So if a user have installed and purchase on same date then we want count it only 1 time, thus we have used DISTINCT.

OUTPUT

event_date	total_active_users
2022-01-01	3
2022-01-02	1
2022-01-03	3
2022-01-04	1

2.Find total active users each week

QUERY

```sql
SELECT DATEPART(WEEK,event_date) week_no
      ,COUNT(DISTINCT user_id) total_active_users
FROM activity
GROUP BY DATEPART(WEEK,event_date)
```

QUERY EXPLANATION

1.To get week number we have used DATEPART function.

2.WITH count we are simply counting distinct users for that particular week.

OUTPUT

week_no	total_active_users
1	3
2	5

3.Date wise total number of users who made the purchase same day they installed the app

QUERY

```sql
WITH cte AS (
SELECT user_id
      ,MIN(CASE WHEN event_name ='app-installed'
                THEN event_date END) installed_date
      ,MIN(CASE WHEN event_name='app-purchase'
                THEN event_date END) purchase_date
      ,event_date
FROM activity
GROUP BY user_id,event_date
)
SELECT event_date
      ,SUM(CASE WHEN installed_date=purchase_date
                THEN 1 ELSE 0 END) total_users
FROM cte
GROUP BY event_date
```

QUERY EXPLANATION

1.WITH cte for each user we are creating two separet columns for app isntalled date and app purchase date using CASE WHEN statment.

2.Now for each event date we are simply counting the users who purchased the app on same day they installed the app.

To count such users we first flagged with 1 using the CASE WHEN statement if the purchase date and installed date are same.

After this we summed them up to get the total count of such users.

OUTPUT FROM CTE

user_id	installed_date	purchase_date	event_date
1	2022-01-01	NULL	2022-01-01
2	2022-01-01	NULL	2022-01-01
3	2022-01-01	NULL	2022-01-01
1	NULL	2022-01-02	2022-01-02
3	NULL	2022-01-03	2022-01-03
4	2022-01-03	2022-01-03	2022-01-03
5	2022-01-03	2022-01-03	2022-01-03
6	2022-01-04	2022-01-04	2022-01-04

FINAL OUTPUT

event_date	total_users
2022-01-01	0
2022-01-02	0
2022-01-03	2
2022-01-04	1

4.Percentage of Paid Users in India, USA and ay other country should be tagged as others

QUERY

```sql
WITH cte AS (
SELECT user_id
      ,event_name
      ,CASE WHEN country='India' THEN 'India'
            WHEN country='USA' THEN 'USA'
            ELSE 'Others'
       END AS country
FROM activity
WHERE event_name='app-purchase'
)
SELECT country
      ,CAST(COUNT(*)*100.0/(SELECT COUNT(*) FROM cte)
            AS DECIMAL(15,2)) perc_paid_users
FROM cte
GROUP BY country
```

QUERY EXPLANATION

1.WITH cte we are are simply flagging the country using CASE WHEN statement.

Here we have filtered for those record where event type is app pruchased one.

2.Now we are simply selecting the country and calculating the percentage of paid users.

Here, we have counted the users so that for each country we get total users and divided it by total number of users who have purchased app from all countries.

To get total users we have used subquery.

OUTPUT FROM CTE

user_id	event_name	country
1	app-purchase	India
3	app-purchase	USA
4	app-purchase	India
5	app-purchase	Others
6	app-purchase	Others

FINAL OUTPUT

country	perc_paid_users
India	40.00
Others	40.00
USA	20.00

5.Among all the users who installed the app on a given day, how many did in app puchased on the very next day -- day wise result

QUERY

```sql
WITH cte AS (
SELECT user_id
      ,event_name
      ,event_date installed_date
      ,LEAD(event_date) OVER (PARTITION BY user_id
                              ORDER BY event_date) purchase_date
FROM activity
)
SELECT ISNULL(purchase_date,installed_date) event_date
      ,SUM(CASE WHEN DATEDIFF(DAY,installed_date,purchase_date)=1
                THEN 1
                  ELSE 0
            END) AS total_users
FROM cte
GROUP BY ISNULL(purchase_date,installed_date)
```

QUERY EXPLANATION

Assumption for one user there can be maximum two event date one for installed and one for purchased in this data.

1.With cte for each event we are getting the installed date and purchase date.

Here we are also assuming that someone will first install the app then only they will purchase it.
So to get the purchase date we have used LEAD function.

2.Now we have selected all event dates and for that we are counting users who have purchased the app on next day when they installed the app.

To count this we have used CASE WHEN statement with SUM funtion. We used ISNULL function so that we get all event dates.

OUTPUT FROM CTE

user_id	event_name	installed_date	purchase_date
1	app-installed	2022-01-01	2022-01-02
1	app-purchase	2022-01-02	NULL
2	app-installed	2022-01-01	NULL
3	app-installed	2022-01-01	2022-01-03
3	app-purchase	2022-01-03	NULL
4	app-installed	2022-01-03	2022-01-03
4	app-purchase	2022-01-03	NULL
5	app-installed	2022-01-03	2022-01-03
5	app-purchase	2022-01-03	NULL
6	app-installed	2022-01-04	2022-01-04
6	app-purchase	2022-01-04	NULL

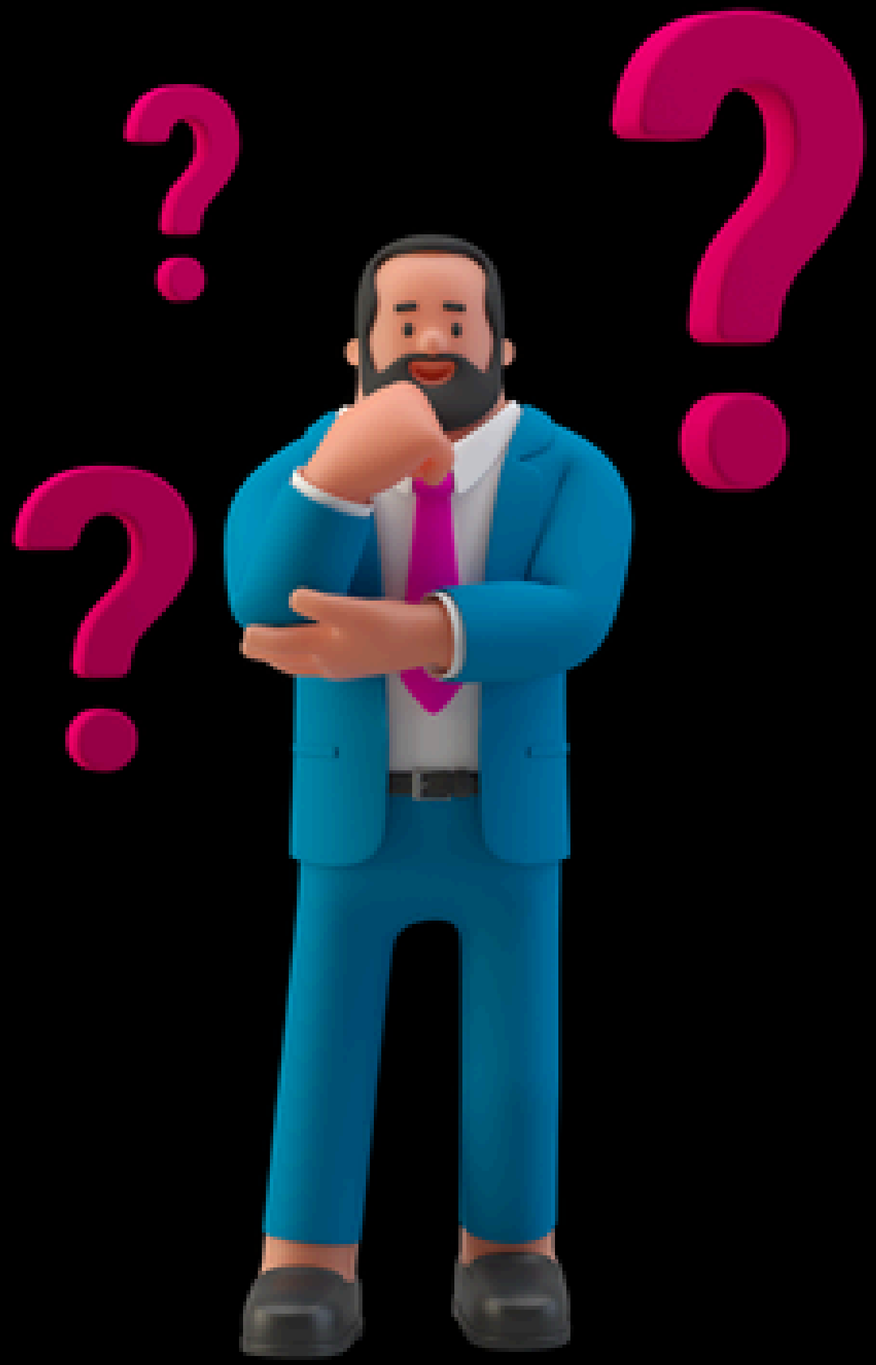
FINAL OUTPUT

event_date	total_users
2022-01-01	0
2022-01-02	1
2022-01-03	0
2022-01-04	0

BY MANISH KUMAR CHAUDHARY

THANK YOU

It is often the small steps, not the giant leaps, that bring about the most lasting change.



DO YOU KNOW?

#5SQLquestions

Interview Question

SQL

BY MANISH KUMAR CHAUDHARY

WHAT IS SQL?

SQL (Structured Query Language) is a programming language that is designed to manage and manipulate relational databases.

It is widely used to store, retrieve, and modify data in relational database management systems (RDBMS) such as MySQL, Oracle, Microsoft SQL Server, PostgreSQL, and SQLite. SQL allows users to interact with databases through a set of commands or queries that can perform various operations like creating tables, inserting data, updating records, deleting data, and retrieving data based on specific criteria.

SQL is an essential tool for data analysts, developers, and database administrators who work with large amounts of structured data.

What is a database?

A database is a crucial tool used in managing and storing structured data in computer systems. It is designed to efficiently handle large amounts of information, providing quick and easy access to users.

Databases are used in a wide range of applications such as websites, mobile apps, and business systems.

Different types of databases exist, classified based on their structure and usage. Relational databases are the most commonly used type and utilize a structured format to store data in tables with relationships between them. Other types of databases include NoSQL databases, object-oriented databases, and graph databases.

What is DBMS?

DBMS stands for Database Management System. It is software that is designed to manage and organize large amounts of data efficiently. DBMS provides an interface for users and applications to interact with databases, allowing for the creation, modification, retrieval, and deletion of data.

DBMS is responsible for handling the storage, backup, security, and recovery of data, ensuring data consistency and integrity. It allows multiple users to access and manipulate data simultaneously while enforcing data security and access control.

Examples of popular DBMS software include Oracle, Microsoft SQL Server, MySQL, PostgreSQL, and MongoDB. These systems use different data models, such as relational, NoSQL, object-oriented, and graph databases, to manage data efficiently.

What is RDBMS?

RDBMS stands for Relational Database Management System. It is a type of database management system that is based on the relational data model. In an RDBMS, data is organized in tables that are related to each other based on common attributes.

An RDBMS provides an interface for users and applications to interact with the database, allowing for the creation, modification, retrieval, and deletion of data. It is responsible for managing the storage, backup, security, and recovery of data, ensuring data consistency and integrity.

Some of the most popular RDBMS software include Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and SQLite. They use Structured Query Language (SQL) to interact with the database, allowing users to query data, insert new records, modify existing records, and delete records.

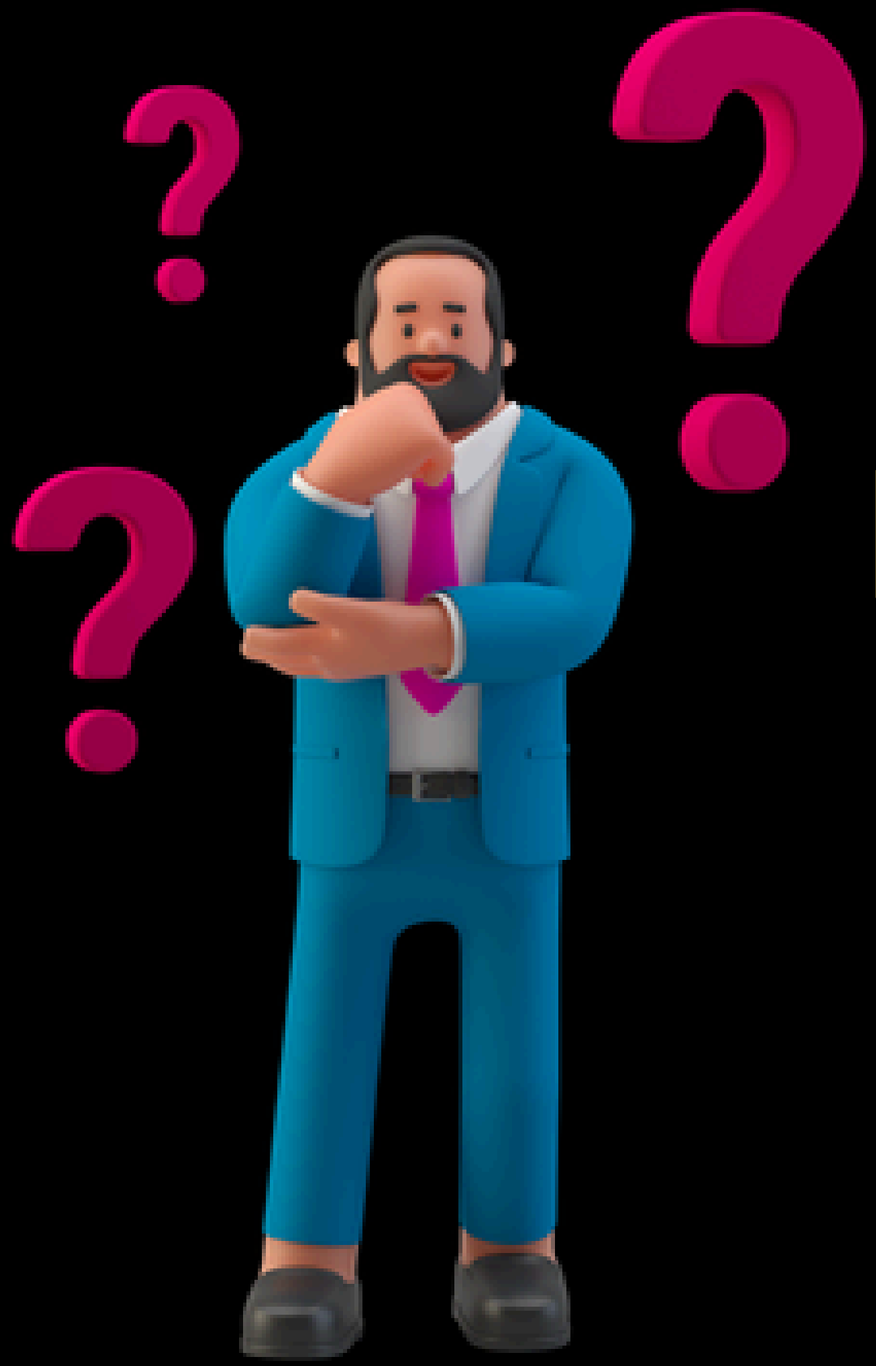
What is difference between DBMS and RDBMS?

Feature	DBMS	RDBMS
Data model	May use various data models (e.g. hierarchical, network, object-oriented)	Uses relational data model
Data organization	Flat, without explicit relationships between tables	Organized into tables with explicit relationships between them
Query language	Proprietary language or languages specific to the system	Standardized query language, such as SQL
Data redundancy	May allow for redundancy, which can lead to inconsistency and wasted storage space	Designed to minimize redundancy and maintain data consistency
Data consistency	Not as strict as RDBMS, may not have built-in mechanisms to ensure consistency	Built-in mechanisms such as foreign keys and constraints to ensure data consistency
Scalability	May be less scalable due to lack of relationship support	Highly scalable with support for complex relationships between tables
Examples	Examples include FileMaker and Microsoft Access	Examples include Oracle, MySQL, and Microsoft SQL Server

THANK YOU

“SOMETIMES IT TAKES A GOOD FALL TO REALLY KNOW WHERE YOU STAND”

HAYLEY WILLIAMS



DO YOU KNOW?

Data Modeling

Interview Question

SQL

BY MANISH KUMAR CHAUDHARY

What is Data Modeling?

Data Modeling is the visual concept used for creating a conceptual represntation of data and its relationship. This representation helps define the structure of different data and relationship between them.

A example before making an actual house you create a design or map of that house on paper or device which acts as refrence while building the actual house. Similarly Data Modeling also acts as refrence for keeping the required data is well structured manner and it acts like a blueprint for database design.

The goal of data modeling is to create a structured and organized representation of data that can be easily understood and used to support business decision-making and analysis.

Features of Good Data Model

Simplicity: A good data model should be simple and easy to understand, with clear and well-defined entities, attributes, and relationships.

Flexibility: The data model should be flexible enough to accommodate changes or updates to the data structure without requiring significant modifications to the model.

Scalability: The data model should be scalable, able to handle large volumes of data without sacrificing performance or accuracy.

Accuracy: The data model should accurately represent the data and the relationships between different entities and attributes.

Consistency: The data model should be consistent throughout, with a clear and defined set of rules and standards for naming conventions and data structures.

Features of Good Data Model

Reusability: A good data model should be reusable, with components that can be easily adapted and reused for other projects or applications.

Completeness: A good data model should include all the necessary entities, attributes, and relationships required to represent the data accurately and completely.

Maintainability: The data model should be easy to maintain and update over time, with clear documentation and processes for making changes and updates.

A good data model should be able to answer the following questions:

1.What are the entities that make up the data, and how are they related to each other?

2.What are the attributes of each entity, and what are their data types and constraints?

3.What are the relationships between the entities, and what type of relationship exists between them (one-to-one, one-to-many, many-to-many)?

4.How is data stored and organized within the data model, and what is the data access and retrieval mechanism?

5.How does the data model support business requirements and analysis, and what insights can be gained from analyzing the data?

Benefits of Data Modeling

Improved Data Quality: By defining the structure and relationships of data, data modeling can improve data quality, making it more consistent, accurate, and reliable.

Better Decision Making: Data modeling provides a clear and organized view of the data, enabling better decision-making based on insights gained from data analysis.

Reduced Data Redundancy: By eliminating data redundancy and optimizing data storage, data modeling can improve data efficiency and reduce data storage costs.

Increased Data Consistency: With clear rules and standards for data naming conventions and structures, data modeling can increase data consistency throughout the organization, promoting better communication and collaboration.

Benefits of Data Modeling

Improved System Design: Data modeling provides a blueprint for designing data storage systems and databases, ensuring that they are well-designed and meet the needs of the organization.

Easier System Maintenance: With clear documentation and processes for making changes and updates, data modeling can make system maintenance easier and more efficient.

NOTE: Apart from these all benefits a **Good Data Model in Power BI** always helps to overcome the use of **complex DAX** functions. If you have Good Data Model then without using complex DAX we will be able to perform our analysis.

My Thoughts on Data Modeling

I think Data Model as one of the most important concept that one should know if they are working as Data Analyst or are aspiring to be one.

If you are able to create a Good Data Model that means you understand the business you are working for and also understand how data should be related to each other.

If you create a Good Data Model then 70-80% of our work becomes easier.

Note: These are only my thoughts.

THANK YOU

"DON'T BOTHER JUST TO BE BETTER THAN YOUR CONTEMPORARIES OR PREDECESSORS. TRY TO BE BETTER THAN YOURSELF."

WILLIAM FAULKNER

BY MANISH KUMAR CHAUDHARY

Interview Question

SQL

Prime subscription rate by product action

DIFFICULTY LEVEL :HARD

Question From

Ankit Bansal

Yotube Channel

PROBLEM STATEMENT

- Given the following two tables, return the fraction of users, rounded to two decimal places, who accessed Amazon music and upgraded to prime membership within the first 30 days of signing up.

users Table

user_id	name	join_date
1	Jon	2020-02-14
2	Jane	2020-02-14
3	Jill	2020-02-15
4	Josh	2020-02-15
5	Jean	2020-02-16
6	Justin	2020-02-17
7	Jeremy	2020-02-18

events Table

user_id	type	access_date
1	Pay	2020-03-01
2	Music	2020-03-02
2	P	2020-03-12
3	Music	2020-03-15
4	Music	2020-03-15
1	P	2020-03-16
3	P	2020-03-22

MENTAL APPROACH

- Firstly, we need to identify users who have an Amazon Music subscription.

- Next, we need to determine if the same user also has an Amazon Prime subscription.

- If the user has Amazon Prime, we need to check whether they accessed their Prime subscription within 30 days of their join date.

- If the user accessed their Prime subscription within 30 days of their join date, we will count them as a relevant user for our fraction calculation.

- Finally, we will divide the number of relevant users by the total number of users who have access to Amazon Music to get the user fraction.

QUERY

```sql
WITH users_music AS (
SELECT u.user_id
      ,u.join_date
FROM users u
INNER JOIN events e
ON u.user_id=e.user_id
WHERE e.type='Music'
)
SELECT CAST(COUNT(e.user_id)*100.0/(SELECT COUNT(1) FROM users_music)
            AS DECIMAL(15,2)) fraction_of_users
FROM events e
INNER JOIN users_music m
ON e.user_id=m.user_id
WHERE type='P'
AND access_date<=DATEADD(DAY,30,m.join_date)
```

OUTPUT

fraction_of_users
33.33

QUERY EXPLANATION

- First, we are using the users_music CTE to identify the users who have an Amazon Music subscription and their join date.

- Next, we are directly calculating the user fraction by counting the relevant users. To do this, we are counting the users who have an Amazon Music subscription and have also subscribed to Amazon Prime within 30 days of their join date. We are using the DATEADD function to check if they accessed Amazon Prime within 30 days of their join date.

- To count the total number of users who have an Amazon Music subscription, we are using a subquery and referring to the users_music CTE.

- Overall, this approach accurately identifies and counts the relevant users for our fraction calculation.

BY MANISH KUMAR CHAUDHARY

THANK YOU

"Success is not about being the best. It's about always getting better."

Behdad Sami

BY MANISH KUMAR CHAUDHARY

Interview Question

SQL

Product Market Share

DIFFICULTY LEVEL :HARD

Question From

stratascratch

PROBLEM STATEMENT

- Write a query to find the Market Share at the Product Brand level for each Territory, for Time Period Q4-2021.

- Market Share is the number of Products of a certain Product Brand brand sold in a territory, divided by the total number of Products sold in this Territory.

- Output the ID of the Territory, name of the Product Brand and the corresponding Market Share in percentages.

- Only include these Product Brands that had at least one sale in a given territory.

fct_customer_sales Sample input Table

cust_id	prod_sku_id	order_date	order_value	order_id
C274	P474	2021-06-28	1500	O110
C285	P472	2021-06-28	899	O118
C282	P487	2021-06-30	500	O125
C282	P476	2021-07-02	999	O146
C284	P487	2021-07-07	500	O149
C285	P478	2021-07-12	700	O150
C287	P489	2021-07-13	189	O151
C284	P482	2021-07-15	725	O156

map_customer_territory Sample input Table

cust_id	territory_id
C273	T3
C274	T3
C275	T1
C276	T1
C277	T1
C278	T2
C279	T2
C280	T4
C281	T4

dim_product

prod_sku_id	prod_sku_name	prod_brand	market_name
P472	iphone-13	Apple	Apple IPhone 13
P473	iphone-13-promax	Apple	Apply IPhone 13 Pro Max
P474	macbook-pro-13	Apple	Apple Macbook Pro 13" Apple Makbook
P475	macbook-air-13	Apple	Air 13" Apple IPad Apple
P476	ipad	Apple	IPad Pro
P477	ipad-pro	Apple	Samsung Galaxy S21
P478	galaxy-s21	Samsung	Samsung Galaxy Tab A
P481	galaxy-tab-a	Samsung	

MENTAL APPROACH

- To calculate the market share of a particular product brand in a given territory, we need to find the total number of products sold by that brand in the territory, as well as the total number of products sold in the territory.

- Once we have the total number of products sold by a particular brand in a given territory and the total number of products sold in that territory, we can calculate the market share of that brand in the territory using the following formula:

- (number of products sold by the particular brand in the territory)*100 / (total number of products sold in the territory)

QUERY

```sql
SELECT t.territory_id
      ,p.prod_brand
      ,CAST(COUNT(s.prod_sku_id)*100.0/
       SUM(COUNT(s.prod_sku_id)) OVER (PARTITION BY t.territory_id)
        AS DECIMAL(15,2)) AS market_share
FROM fct_customer_sales s
INNER JOIN map_customer_territory t
ON s.cust_id=t.cust_id
INNER JOIN dim_product p
ON s.prod_sku_id=p.prod_sku_id
WHERE DATEPART(QQ,s.order_date)=4
AND YEAR(s.order_date)=2021
GROUP BY t.territory_id, p.prod_brand
ORDER BY t.territory_id
```

QUERY EXPLANATION

1.Here with COUNT(prod_sku_id) we are simply counting the total number of products sold by particular brand in a particular territory as we have grouped on these two.

2. With SUM(COUNT(prod_sku_id)) OVER (PARTITION BY Territory). First counting of products will be done similar to first step and then SUM of these counts will be produced for each territory. So we will get total number of products sold in a particular territory.

3.For calculating the markte share we took help of the formula.

SAMPLE OUTPUT

territory_id	prod_brand	market_share
T1 T1 T1 T2	Apple	33.33 16.67
T2 T3 T3 T3	JBL	50.00 25.00
T3 T3	Samsung	75.00 37.50
	Apple	12.50 12.50
	Samsung	25.00 12.50
	Apple	
	Canon	
	Dell	
	GoPro	
	JBL	

BY MANISH KUMAR CHAUDHARY

THANK YOU

"The only limit to our realization of tomorrow will be our doubts of today."

Franklin D. Roosevelt

BY MANISH KUMAR CHAUDHARY

Interview Question

SQL

Scenario Based SQL Question | Solving Using SCD Type 2 Concept | SQL Interview Question

DIFFICULTY LEVEL :HARD

Question From

Ankit Bansal

Yotube Channel

PROBLEM STATEMENT

Write an SQL query to calculate the total bill for each employee based on the number of hours worked and the applicable billing rate from the "billings" table.

The output should include the employee name and the total bill and exclude employees who have no hours worked in the "HoursWorked" table.

The billing rate in the "billings" table should be valid until the next billing rate starts, or until a new billing rate is specified for that employee, whichever comes first.

billings Table

emp_name	bill_date	bill_rate
Sachin	1990-01-01	25
Sehwag	1989-01-01	15
Dhoni	1989-01-01	20
Sachin	1991-02-05	30

HoursWorked Table

emp_name	work_date	bill_hrs
Sachin	1990-07-01	3
Sachin	1990-08-01	5
Sehwag	1990-07-01	2
Sachin	1991-07-01	4

Types of SCD (Slow Changing Dimension) Concept in SQL

SCD Type 1

In this type, the old data is simply overwritten with the new data.

It means that the dimension is updated without any history tracking.

This approach is suitable when the history of the dimension data is not important or there is no need to store it.

SCD Type 2

In this type, the new data is inserted into the dimension table as a new row, along with a new surrogate key and a start and end date.

The old data is kept in the table with an end date that corresponds to the start date of the new row.

This approach allows us to keep track of the history of the dimension data.

SCD Type 3

In this type, the dimension table is updated with a new attribute column to hold the new data.

However, the old data is not kept in the table, so there is no history tracking.

This approach is useful when we only need to track a small amount of history and the size of the dimension table is a concern.

QUERY By Ankit Bansal Sir

Solved using SCD Type 2 Concept

```sql
WITH bill_dates AS (
SELECT emp_name
       ,bill_date current_bill_date
       ,LEAD(bill_date,1,'9999-12-31') OVER (PARTITION BY emp_name
                                             ORDER BY bill_date) next_bill_date
       ,bill_rate
FROM billings
)
SELECT w.emp_name
       ,SUM(w.bill_hrs*b.bill_rate) total_bill
FROM HoursWorked w
LEFT JOIN bill_dates b
ON w.emp_name=b.emp_name
AND w.work_date BETWEEN b.current_bill_date AND DATEADD(DAY,-1,b.next_bill_date)
GROUP BY w.emp_name
```

QUERY EXPLANATION

1.With bill_dates CTE we are simply getting the current and next bill date when the bill rate has changed.
Here we have taken use of LEAD function to get next bill date and for default we have provided with a date (i.e for null it will replace with default date we have provided)

2.We are now SELECTING the employee name and SUM of product of bill rate and bill hours to get the total amount to be paid to the employee.

Here we have joined on the basis of employee name and where work date is BETWEEN current bill date and one less than next bill date.

We are doing next_bill_date -1 because after that bill rate has changed so for that date we want new bill rate.

OUTPUT FROM bill_dates CTE

emp_name	current_bill_date	next_bill_date	bill_rate
Dhoni	1989-01-01	9999-12-31	20
Sachin	1990-01-01	1991-02-05	25
Sachin	1991-02-05	9999-12-31	30
Sehwag	1989-01-01	9999-12-31	15

FINAL OUTPUT

emp_name	total_bill
Sachin	320
Sehwag	30

QUERY BY ME

```sql
WITH bills_cte AS (
SELECT w.emp_name
       ,w.work_date
       ,w.bill_hrs
       ,MAX(b.bill_rate) bill_rate
FROM HoursWorked w
INNER JOIN billings b
ON w.emp_name=b.emp_name
AND w.work_date>=bill_date
GROUP BY w.emp_name, w.work_date,w.bill_hrs
)
SELECT emp_name
     ,SUM(bill_rate*bill_hrs) total_bill
FROM bills_cte
GROUP BY emp_name
```

QUERY EXPLANATION

1.With bill_cte CTE we are fetching employee name, bill hours, work date, and MAX of bill rate.

Here we have joined on the basis of employee name and where the work_date is greater than equal to bill date.

We have used MAX of bill rate so that where there will be null it will replace with MAX date for particular condition.

We have grouped on the basis of employee name, bill hours and work date so that we get only distinct such values.

2.Now we are simply selecting the employee name with the SUM of product of bill rate and bill hours to get total amount to be paid to an employee.

OUTPUT FROM bills_cte

emp_name	work_date	bill_hrs	bill_rate
Sachin	1990-07-01	3	25
Sachin	1990-08-01	5	25
Sachin	1991-07-01	4	30
Sehwag	1990-07-01	2	15

FINAL OUTPUT

emp_name	total_bill
Sachin	320
Sehwag	30

THANK YOU

“Every moment is a fresh beginning.”