DML Queries of StreamWave Database

1.Number of streams broadcasted by a streamer on the platform:

•Purpose: To get the total number of streams broadcasted by a specific streamer on the platform.

```
SELECT COUNT(StreamID) AS Total_Streams

FROM Stream

WHERE StreamerID = 'pewd@S001';
```

2.All the chats a viewer has ever made (Chat History):

•Purpose: To retrieve all the chat logs, timestamps, and associated stream IDs for a specific viewer.

```
SELECT Chat_log, chat_timestamp, StreamID

FROM Chat_Reactions

WHERE ViewerID = 'V0001';
```

3.The total amount of donations a viewer has made on the platform:

•Purpose: To calculate the total amount of donations made by a specific viewer on the platform.

```
SELECT SUM(Don_Amt) AS Total_Donations

FROM Donations

WHERE ViewerID = 'V0001';
```

4.All the channels a viewer is currently subscribed to:

•Purpose: To retrieve the list of channels that a specific viewer is currently subscribed to.

```
SELECT s.StreamerID, s.ChannelName

FROM Streamer s

JOIN Subscriptions d ON s.StreamerID = d.StreamerID

WHERE d.ViewerID = 'V0001'

AND d.Subs_End > CURRENT_DATE;
```

5.Specific category of the streamers a viewer wants to watch:

•Purpose: To get a list of recent streams from a specific category (e.g., Gaming) that a viewer might be interested in watching.

```
SELECT s.StreamID, s.Title, s.Language, sc.Category

FROM Stream s

JOIN Stream_Cat sc ON s.StreamID = sc.StreamID

WHERE sc.Category = 'Gaming' -- or any other category

ORDER BY s.Stream_Date DESC;
```

6.What streams a viewer has watched (Watch History):

•Purpose: To retrieve the watch history of a specific viewer, including details about the streams they have watched, such as title, language, date, and their watch duration.

```
SELECT s.StreamID, s.Title, s.Language, s.Stream_Date,w.Watch_Dur

FROM Stream s NATURAL JOIN Watches w

WHERE  w.ViewerID = 'V0001'

GROUP BY s.StreamID,w.Watch_Dur
```

```
ORDER BY s.Stream_Date DESC;
```

7.See what kind of sponsors are available for a streamer:

•Purpose: To retrieve the list of sponsorship companies and their numbers that are available for a specific streamer.

```
SELECT G.Sponsorship_No, S.Spons_Company
FROM Sponsorships_Platform_Deals S NATURAL JOIN Given_To G
WHERE G.Valid_For_Sponsor = 'Y' and G.StreamerID = 'poki@S004';
```

8.To get to know the age range of a streamer’s viewers:

•Purpose: To calculate the age range of viewers subscribed to a specific streamer's channel.

```
SELECT MIN(V_Age) AS MinimumAge, MAX(V_Age) AS MaximumAge
FROM Viewer v
JOIN Watches w ON v.ViewerID = w.ViewerID
NATURAL JOIN Stream s
WHERE s.StreamerID = 'pewd@S001';
```

9.To find the percentage of subscribers and non-subscribers for a streamer:

•Purpose: To count the percentage of viewers who have subscribed or not subscribed to a specific streamer.

```
WITH CTE AS (
 SELECT
  CASE WHEN Subs_Start IS NOT NULL THEN 'Subscriber' ELSE 'Non-Subscriber' END AS SubscriberStatus,
  COUNT(*) AS Count
 FROM Subscriptions
 WHERE StreamerID = 'pewd@S001'
```

```
  GROUP BY SubscriberStatus
)
SELECT
  SubscriberStatus,
  ROUND(100.0 * Count / (SELECT SUM(Count) FROM CTE), 2) AS Percentage
FROM CTE;
```

10.The Comments a streamer got in a particular stream/video:

•Purpose: To list the comments received by a streamer during a specific stream, and their chat timestamps accordingly.

```
SELECT Chat_Log AS Comment, Chat_timestamp
FROM Chat_Reactions
WHERE StreamID = 'ST001';
```

11.The Number of Follows a streamer gained/lost in a limited duration:

•Purpose: To count the number of follows the streamer got between start date and end date.

```
WITH CTE AS (
  SELECT
    StreamerID,
    Follows.Fol_Date,
    CASE WHEN Fol_Status = TRUE THEN 1 ELSE -1 END AS SubscriberChange
  FROM Follows
  WHERE StreamerID = 'pewd@S001'
    AND Follows.Fol_Date BETWEEN '2011-11-01' AND '2024-01-01'
)
SELECT SUM(SubscriberChange) AS SubscriberChange
FROM CTE;
```

12.The Number of Returning viewers to the stream:

- **Purpose: To count the number of distinct viewers who have participated in more than one stream of a specific streamer, indicating their returning audience.**

```
SELECT COUNT(DISTINCT ViewerID) AS ReturningViewers
FROM (
  SELECT ViewerID
  FROM Watches
  WHERE StreamID = 'ST001'
  GROUP BY ViewerID
  HAVING COUNT(DISTINCT StreamID) > 1
) AS SubQuery;
```

13.The Total donations a streamer has received in a stream:

- **Purpose: Calculate total donations for a specific stream.**

```
SELECT SUM(Don_Amt) AS TotalDonations
FROM Donations
WHERE StreamerID = 'pewd@S001';
```

14.The Revenue the streamer has earned from ads:

- **Purpose: Compute ad revenue earned from streamer's streams.**

```
SELECT SUM(a.Revenue_Split_Ratio* a.Ad_Earnings) AS TotalAdRevenue
FROM Advertisements a
JOIN Shown_On so ON a.AdvID = so.AdvID
WHERE a.StreamerID = 'pewd@S001';
```

15.Analyze the type of viewers a streamer gets in a stream:

•Purpose: Understand viewer demographics and interests.

```
SELECT v.V_Country, v.V_Age, vi.Interests, COUNT(*) AS ViewerCount
FROM Viewer v
JOIN Viewer_Int vi ON v.ViewerID = vi.UserID
JOIN Watches w ON v.ViewerID = w.ViewerID
WHERE w.StreamID = 'ST002'
GROUP BY v.V_Country, v.V_Age, vi.Interests;
```

16.List percentage distribution of streamers from different countries:

•Purpose: Analyse the streaming population of countries.

```
SELECT Country, ROUND((COUNT(*) * 100.0) / (SELECT COUNT(*) FROM Streamer),2) AS
Percentage
FROM Streamer
GROUP BY Country
ORDER BY Percentage desc;
```

17.List streamers who have registered in the year 2022:

•Purpose: Get the list of streamers who have registered in a specific date range.

```
SELECT StreamerID, ChannelName, S_Reg_Date
FROM Streamer
WHERE CAST(S_Reg_Date as VARCHAR) LIKE  '2011%';
```

18.List top 10 streamers with the highest number of followers:

•Purpose: Find the top 10 streamers with the highest number of followers.

```
SELECT s.ChannelName
FROM Streamer s
```

```
JOIN Follows AS f ON s.StreamerID = f.StreamerID

GROUP BY s.StreamerID

ORDER BY COUNT(s.StreamerID) DESC

LIMIT 10;
```

19. Percentage of viewer’s unbanned by a streamer:

•Purpose: To find the percentage of viewer’s unbanned by a streamer on reviewing their request

```
Select (100*count(*))/

( Select count(*) from Chat_Unban_Requests natural join streamer s where
s.StreamerID='poki@S004')

As PercentageUnbanned

From Chat_Unban_Requests c natural join streamer s

Where c.Unban_Approved = 'Y' and s.StreamerID='poki@S004';
```

20.List earnings of channel administrators on the platform:

•Purpose: To find the list of total earnings as channel administrators for streamers on the platform.

```
Select h.StreamerID, sum(c.Earnings) as Total_Earnings

from Channel_Administrators c natural join Hires h

group by h.StreamerID

order by sum(c.Earnings) desc;
```

21.List the streamers who received the highest donation in May 2022:

•Purpose: Find the streamers who have received the highest donations in a specific month.

```
SELECT S.STREAMERID, S.CHANNELNAME, SUM(D.DON_AMT) AS TOTAL_DONATIONS

FROM Streamer s
```

```
JOIN Donations d ON s.StreamerID = d.StreamerID
WHERE CAST(d.Don_timestamp AS VARCHAR) LIKE '_____04%' AND
CAST(d.Don_timestamp AS VARCHAR) LIKE '2024%'
GROUP BY s.StreamerID
ORDER BY Total_Donations DESC;
```

22.List date of ban of streamers who are banned currently:

•Purpose: Get the list of streamers who have been banned along with the reason and date of ban.

```
SELECT s.StreamerID, s.ChannelName, b.Reason_for_Ban, b.Date_of_Ban
FROM Streamer s
JOIN Streamer_Bans b ON s.StreamerID = b.StreamerID
WHERE b.Status_of_Ban = 'Y';
```

23.List details of advertisements shown on a specific stream:

•Purpose: Retrieve the details of advertisements shown on a specific stream.

```
SELECT distinct a.AdvID, a.Ad_Name, a.Company_Name, so.Ad_Duration,
so.Ad_timestamp,a.Ad_Earnings
FROM Advertisements a
JOIN Shown_On so ON a.AdvID = so.AdvID
WHERE so.StreamID = 'ST001';
```

24.List streamers that have active sponsorships or platform deals:

•Purpose: Find the streamers who have active sponsorships or platform deals.

```
SELECT s.StreamerID, s.ChannelName, pd.Spons_Company, pd.Spons_Date,
pd.Spons_Duration
FROM Streamer s
JOIN Given_To gt ON s.StreamerID = gt.StreamerID
JOIN Sponsorships_Platform_Deals pd ON gt.Membership_No = pd.Membership_No OR
gt.Sponsorship_No = pd.Sponsorship_No;
```

25.List viewers that have been banned from a specific streamer's channel, and state the reasons and timestamps for their bans:

•Purpose: Get the list of viewers who have been banned from a specific streamer's channel.

```
SELECT v.ViewerID, v.V_Email, b.Reason_For_Ban, b.Ban_timestamp
FROM Viewer v
JOIN Chat_Reactions cr ON v.ViewerID = cr.ViewerID
JOIN Chat_Unban_Requests b ON cr.ViewerID = b.ViewerID
WHERE b.StreamerID = 'pewd@S001' AND b.Unban_Approved = 'N';
```

26.List the requirements stated by a specific company during sponsorships

•Purpose: To get an idea of the common requirements of a particular company

```
SELECT distinct r.comp_req
from requirements r natural join sponsorships_platform_deals s
where s.Spons_company='ABC Inc';
```

27.Find streamers that are currently open for hiring channel administrators:

•Purpose: Find the streamers who are open for hiring channel administrators.

```
SELECT s.StreamerID, s.ChannelName
FROM Streamer s
JOIN Hires h ON s.StreamerID = h.StreamerID
WHERE h.Open_For_Hiring = 'Yes';
```

28. Stream categories available on the platform:

- Purpose: Get the list of streams with their respective categories.

```
SELECT s.StreamID, s.Title, sc.Category
FROM Stream s
JOIN Stream_Cat sc ON s.StreamID = sc.StreamID;
```

29.Viewers status of subscription

- Purpose: To help viewer see his status of subscription for a streamer

```
SELECT d.StreamerID, v.V_Email, d.Subs_Start, d.Subs_End, d.Subs_Type
FROM Viewer v
JOIN Subscriptions d ON v.ViewerID = d.ViewerID
WHERE v.ViewerID = 'V0002' AND d.Subs_Start IS NOT NULL
```

30.List streamers that have been granted a specific type of membership or sponsorship (e.g., 'Gold' membership):

- Purpose: Find the streamers who have been granted a specific type of membership or sponsorship.

```
SELECT s.StreamerID, s.ChannelName, pd.Mem_Type, pd.Mem_Duration
FROM Streamer s
JOIN Given_To gt ON s.StreamerID = gt.StreamerID
JOIN Sponsorships_Platform_Deals pd ON gt.Membership_No = pd.Membership_No
WHERE pd.Mem_Type = 'Gold';
```