

# Database Modeling for the Live Streaming Platform: The Case of Twitch

By: Oscar Montoya and Adiel Hernández

## Abstract

This work addresses the development and optimization of the database modeling that supports the Twitch platform. This article presents a design that can handle millions of concurrent users, facilitating live streaming and real-time interaction. The proposed solution implements a scalable and efficient data model, which has improved the performance and stability of the platform. The results confirm a significant optimization in the responsiveness and handling of massive data, as well as in the user experience.

## Introduction

Twitch is one of the largest live streaming platforms in the world, allowing millions of users to interact with content creators in real time. A key component to its success is the ability to manage large volumes of data in real time, such as chat messages, user profiles, financial transactions for subscriptions and donations, as well as the metadata associated with each broadcast.

The exponential growth of the platform means the database must scale to support millions of concurrent users, without sacrificing speed or data integrity.

To keep interaction smooth and uninterrupted, the database must be able to respond to millions of queries in fractions of a second, especially during peaks of activity such as live events or popular tournaments.

This highlighted the need to implement a robust and scalable database, capable of handling not only the current demands of the platform, but also adapting to future growth. The challenges associated with ensuring data consistency and high availability in a distributed environment are also discussed, as well as solutions implemented to improve response speed for critical queries, such as chat messages and subscription transactions. Database optimization has been essential to ensuring a smooth and seamless user experience, allowing Twitch to remain the world's leading live streaming platform.

[https://idus.us.es/bitstream/handle/11441/87550/Estudio\\_de\\_las\\_plataformas\\_de\\_streaming.pdf](https://idus.us.es/bitstream/handle/11441/87550/Estudio_de_las_plataformas_de_streaming.pdf)

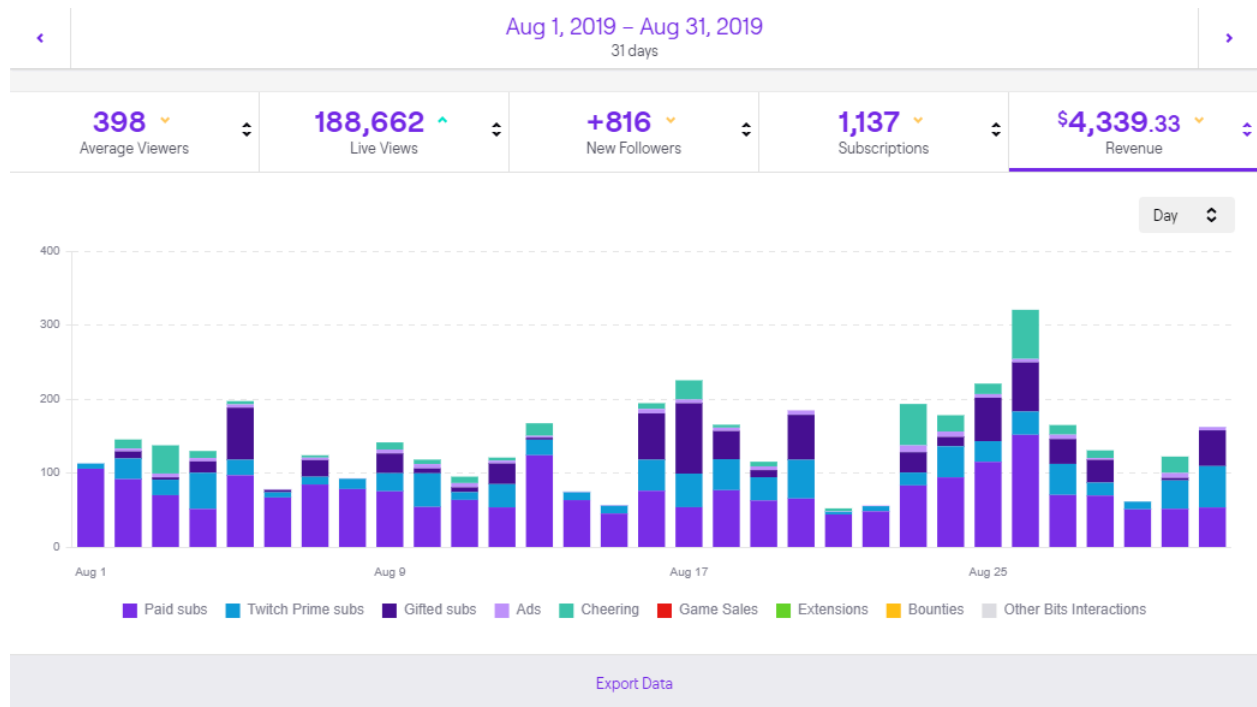
<https://help.twitch.tv/s/article/channel-analytics?language=es>

<https://thepower.education/blog/descubre-ahora-que-es-y-como-funciona-twitch>

## Method and Materials

The Twitch database was designed to handle large volumes of data generated in real time. A distributed database approach was chosen, using horizontal partitioning and data replication to ensure high availability and consistency. The main technical decisions included the use of databases. Additionally, caching techniques were implemented to improve response times for frequently asked queries, such as live chat messages and user profiles.

An exhaustive analysis of the system requirements was carried out, which led to the identification of the main entities to model: Users, Streams, Chats and Transactions. Each entity was designed with specific attributes that capture the information necessary for the platform's functionality. For example, the User entity model includes fields such as user\_id, username, email, while the Stream model contains stream\_id, title, category, and status.



some example statistics taken from the twitch help page where some data about a channel in the application is evident

## Results

Regarding the results, there really seems to be little that can be extracted since at the moment it is only about the modeling that has not yet been tested, not even in a controlled environment, but a high capacity is expected when processing the large amount of data for a level platform. as is twitch

## Conclusions

Database optimization was key to improving the experience for Twitch users. The implemented solutions made it possible not only to handle large volumes of data in real time, but also to guarantee the stability of the platform during peaks of use. The focus on scalability and availability ensures that the platform can continue to grow without impacting the end-user experience. This work demonstrates the importance of having a robust database infrastructure for large-scale applications like Twitch.

## Bibliography

- Analytics Summary [https://help.twitch.tv/s/article/channel-analytics?language=es\\_MX](https://help.twitch.tv/s/article/channel-analytics?language=es_MX)
- Regression Analysis & Visualization of Twitch Dataset  
<https://ieeexplore.ieee.org/document/10064568>
- Data Analysis and Friendship Prediction for Twitch Streamers  
<https://ieeexplore.ieee.org/document/10017732>