

Perfect — this is **exactly the kind of schema** OTT analytics teams love 😊
I'll do this in **two layers**:

1. **What retention analyses matter (business view – beIN style)**
2. **Actual dp1yr-ready code patterns** you can run on a real table

I'll assume your data frame is called **subs**.

1 Retention Analyses That Make Sense for *Your* Schema

Your schema is powerful because you have:

- Subscription lifecycle
- Renewal continuity
- Product & offer periods
- Winback vs churn
- Latest vs historical versions

This enables **true cohort-based retention**, not just activity retention.

A. Logo Retention (Customer-level)

Question: Do customers stay subscribed?

KPI

- Retained = Customer has **active subscription after expiry**
- Churned = No active subscription after expiry

📌 Key lens:

- Monthly vs Annual
 - Sports-only vs 4K bundle
 - Country (with caveats)
-

B. Subscription Retention (Contract-level)

Question: Do subscriptions renew or break?

Use:

- SubscriptionID
- subscription_version
- subscriptionLatest

This distinguishes:

- Continuous renewals (same ID)
 - Break & rejoin (new ID)
-

C. Cohort Retention (Gold Standard)

Question: How long do users stay after joining?

Cohort = subscription_start_Date (month)

Track:

- Active status at Month 0, 1, 3, 6, 12
-

D. Offer Period Retention

Question: Which offers retain better long-term?

Compare:

- Monthly vs 6-month vs Annual
-

E. Product Retention

Question: Which product mix reduces churn?

Compare:

- Sports-only
 - 4K
 - Non-4K
-

F. Winback Analysis

Question: Who churns but comes back?

Use:

- Subscription_type = "winback"
 - New SubscriptionID
-

G. Tenure / Lifetime Analysis

Question: How long does a customer stay subscribed?

Use:

- First start date → last expiry date
-

2 dplyr Code – Practical & Reusable

Step 0 Prep (Always Do This)

```
library(dplyr)
library(lubridate)

subs_clean <- subs %>%
  mutate(
    start_date = as.Date(subscription_start_Date),
    expiry_date = as.Date(ExpiryDate),
    start_month = floor_date(start_date, "month")
  )
```

1 Current Active Base (Baseline KPI)

```
active_base <- subs_clean %>%
  filter(subscriptionLatest == 1, subscription_Status == "active") %>%
  summarise(active_customers = n_distinct(CustomerID))
```

2 Logo Retention Rate (Month-on-Month)

```
logo_retention <- subs_clean %>%
  filter(subscriptionLatest == 1) %>%
  group_by(start_month) %>%
  summarise(
    total_customers = n_distinct(CustomerID),
    active_customers = n_distinct(
      CustomerID[subscription_Status == "active"]
    ),
    retention_rate = active_customers / total_customers
  )
```

3 Cohort Retention Table (Classic OTT View)

Step 1: Identify Cohort Month

```
cohort_df <- subs_clean %>%
  group_by(CustomerID) %>%
  summarise(cohort_month = min(start_month)) %>%
  left_join(subs_clean, by = "CustomerID")
```

Step 2: Months Since Start

```
cohort_df <- cohort_df %>%
  mutate(
    months_since_start = interval(cohort_month, start_month) %/% months(1)
  )
```

Step 3: Retention Matrix

```
cohort_retention <- cohort_df %>%
  filter(subscription_Status == "active") %>%
  group_by(cohort_month, months_since_start) %>%
  summarise(
    retained_customers = n_distinct(CustomerID),
    .groups = "drop"
  )
```

4 Offer Period Retention Comparison

```
offer_retention <- subs_clean %>%
  filter(subscriptionLatest == 1) %>%
  group_by(offer_period) %>%
  summarise(
    customers = n_distinct(CustomerID),
    active_customers = n_distinct(
      CustomerID[subscription_Status == "active"]
    ),
    retention_rate = active_customers / customers
  )
```

5 Product Retention

```
product_retention <- subs_clean %>%
  filter(subscriptionLatest == 1) %>%
  group_by(ProductName) %>%
  summarise(
    customers = n_distinct(CustomerID),
    retained = n_distinct(
      CustomerID[subscription_Status == "active"]
    ),
    retention_rate = retained / customers
  )
```

6 Continuous Renewal vs Break & Rejoin

Same Subscription ID = Healthy Retention

```
renewal_behavior <- subs_clean %>%
  group_by(CustomerID) %>%
  summarise(
    total_subscriptions = n_distinct(SubscriptionID),
    max_version = max(subscription_version, na.rm = TRUE)
  ) %>%
  mutate(
    renewal_type = case_when(
      total_subscriptions == 1 & max_version > 1 ~ "continuous",
      total_subscriptions > 1 ~ "break_and_rejoin",
      TRUE ~ "single_term"
    )
  )
```

7 Winback Rate

```
winback_rate <- subs_clean %>%
  group_by(CustomerID) %>%
  summarise(
    churned = any(subscription_type == "churned"),
    winback = any(subscription_type == "winback")
  ) %>%
  summarise(
    churned_customers = sum(churned),
    winback_customers = sum(winback),
    winback_rate = winback_customers / churned_customers
  )
```

8 Customer Lifetime (Tenure)

```
customer_lifetime <- subs_clean %>%
  group_by(CustomerID) %>%
  summarise(
    first_start = min(start_date),
    last_expiry = max(expiry_date),
    lifetime_days = as.numeric(last_expiry - first_start)
  )
```

9 Country-Aware Retention (VPN-Safe)

Use **dominant country per customer**:

```
customer_country <- subs_clean %>%
  count(CustomerID, Country) %>%
  group_by(CustomerID) %>%
  slice_max(n, n = 1)

country_retention <- subs_clean %>%
  filter(subscriptionLatest == 1) %>%
  left_join(customer_country, by = "CustomerID") %>%
  group_by(Country) %>%
  summarise(
    customers = n_distinct(CustomerID),
    retained = n_distinct(
      CustomerID[subscription_Status == "active"]
    )
  )
```

3 Retention KPIs for Management Slide

KPI	Meaning
Logo Retention %	Customers still active
Cohort Month-3 Retention	Stickiness
Offer Period Retention	Contract efficiency
Product Retention	Content value
Continuous Renewal %	Brand loyalty
Winback Rate	Recovery power
Avg Customer Lifetime	Long-term value

4 What I'd Suggest Next (High Impact)

- Convert cohort table → **heatmap**
- Add **survival curves**
- Merge **usage minutes** for *engagement-driven churn*
- Predict churn with **last 30-day inactivity**

If you want, next I can:

- Build **cohort heatmap code**
- Add **survival analysis (Kaplan–Meier)**
- Combine **usage + subscription**
- Design **executive retention dashboard**

Just tell me where you want to go next 🏹