



**TRANSPORT AND  
TELECOMMUNICATION  
INSTITUTE**

# **Database processing Course Project**

**Technical & User Documentation**

Student's name: Artjoms  
Student's surname: Voroncovs  
St. code: st83663

**RIGA**

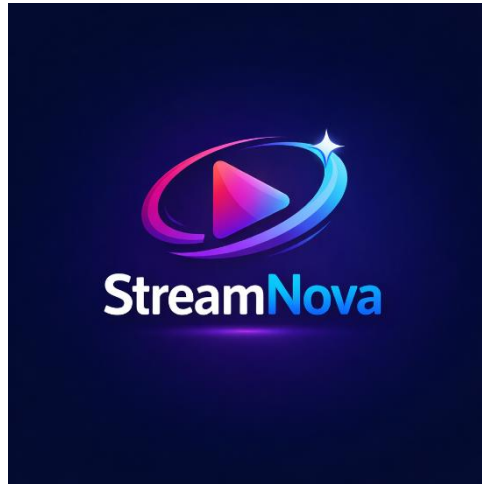
# TABLE OF CONTENTS

1. Introduction .....	3
1.1 Scenario Description .....	3
Business Requirements Summary.....	3
2.1 Key Business Questions.....	3
2.2 Key KPIs (Summary) .....	4
Data Model Overview .....	4
3.1 Tables and Relationships.....	4
3.2 Design Rationale .....	5
Technical Design .....	5
4.1 Report Structure .....	5
4.2 Key Business Rules (Implementation).....	6
4.3 Regional Breakdown .....	6
4.4 Assumptions and Simplifications .....	6
How to Run the Project.....	6
5.1 Execution Order .....	6
5.2 Parameters.....	6

# 1. Introduction

## 1.1 Scenario Description

This project is about video streaming platform which is fully subscription based, platform name is StreamNova. StreamNova provides high quality digital video streaming with exclusive content worldwide. Content provided to individual users under monthly subscription plans. Users able to acquire different subscription plans with varying prices and features online. There is possibility to upgrade or downgrade plans, cancel or reactivate subscription. System is designed to support detailed business analysis of platform revenue, engagement and user lifecycle behavior, underlying system is Oracle database.



*1.0 StreamNova platform logo*

## 1.2 Report Purpose

The Monthly Performance and Engagement Report update management with a aggregated monthly description of:

- Revenue and discounts
- Active subscriber base
- Plan performance
- User engagement
- Unsubscribe activity, retention, and subscriber movement
- Potential data quality issues

The report is generated using Oracle engine directly with help of a PL/SQL programming language and printed via DBMS\_OUTPUT facility.

# Business Requirements Summary

## 2.1 Key Business Questions

The report answers the following business core questions.

### Revenue and Subscription Base

- How much net revenue was generated during selected month?
- How much discount was applied to subscriptions?
- How many paid subscriptions were billed?
- How many active subscribers exist per plan?

### Engagement / Usage

- How much total streaming time occurred in the month?
- What is the average usage per active user?
- Who are the top users by engagement?
- Which device types are most used?

### Churn and Customer Movement

- How many users churned, joined, or reactivated in the month?
- What is the churn and retention rate?

### Data Quality

- Are there invalid or suspicious payments?
- Are there subscriptions or usage records with inconsistent dates or values?

## 2.2 Key KPIs (Summary)

KPI: Monthly Net Revenue

Description: Payments minus discounts for the report month

KPI: Total Discounts

Description: Sum of all discounts applied

KPI: ARPU

Description: Net revenue divided by active subscribers

KPI: Active Subscribers

Description: Users with an active subscription during the month

KPI: Churn Rate

Description: Churned subscribers divided by active subscribers at month start

KPI: Retention Rate

Description: One minus churn rate

KPI: Total Usage Time

Description: Sum of streaming minutes

KPI: Average Usage per User

Description: Total usage divided by active users

KPI: Trial Conversions

Description: Trials converted to paid plans

KPI: Data Quality Issues

Description: Count of detected bad data records

## Data Model Overview

### 3.1 Tables and Relationships

Table Name	Purpose / Description
USERS	Subscriber master data (region, signup date)
PLANS	Subscription plans and prices
SUBSCRIPTIONS	Subscription lifecycle periods
SUBSCRIPTION_EVENTS	Plan changes and status changes
PAYMENTS	Billing and discounts
USAGE_SESSIONS	User engagement and usage events
DATA_QUALITY_ISSUES	Data quality issues, incorrect payments, and anomalous records

Relationships table:

Parent Table	Child Table	Relationship Type	Foreign Key Column(s)
USERS	SUBSCRIPTIONS	One-to-Many	subscriptions.user_id
PLANS	SUBSCRIPTIONS	One-to-Many	subscriptions.plan_id
SUBSCRIPTIONS	SUBSCRIPTION_EVENTS	One-to-Many	subscription_events.subscription_id
PLANS	SUBSCRIPTION_EVENTS	One-to-Many (optional)	subscription_events.old_plan_id
PLANS	SUBSCRIPTION_EVENTS	One-to-Many (optional)	subscription_events.new_plan_id
SUBSCRIPTIONS	PAYMENTS	One-to-Many	payments.subscription_id
USERS	USAGE_SESSIONS	One-to-Many	usage_sessions.user_id

### 3.2 Design Rationale

The table is intentionally small with seven tables and is denormalized only where helpful for sake of reporting. Each table directly supports one or more KPI metrics. Historical accuracy is preserved by modeling subscription periods and events separately.

# Technical Design

## 4.1 Report Structure

The PL/SQL report is divided into the following sections:

1. Header Information
2. Revenue and Subscription Summary
3. Plan Performance
4. Engagement and Usage Metrics
5. Churn and Retention
6. Data Quality Checks

Each section uses SQL aggregation queries executed inside PL/SQL blocks and prints results using internal DBMS\_OUTPUT.PUT\_LINE function.

## 4.2 Key Business Rules (Implementation)

### Active Subscriber Definition

A subscriber considered active if the subscription start date is on or before the last day of the report month, the end date is null or after the start of the report month, and the status is ACTIVE.

### New Subscribers

Subscriptions which start date fall within the selected report month.

### Churned Subscribers

Subscriptions with an end date in the report month and status set to CANCELLED.

### Reactivated Subscribers

Users with a cancelled subscription before the month and a new active subscription starting in the report month.

### Revenue Inclusion Rule

Payments are included if the payment date falls inside the report month.

### Churn Formula

Churn Rate equals churned subscribers divided by active subscribers at the start of the month.

Retention Rate equals one minus the churn rate.

## 4.3 Assumptions and Simplifications

- A single currency is assumed.
- Taxes are ignored.
- Subscription overlap edge cases are not handled.

# How to Run the Project

## 5.1 Execution Order

1. Run the DDL script: `project_ddl_vorontsov.sql`
2. Run the DML script: `project_dml_vorontsov.sql`
3. Run the report script after enabling server output: `project_report_vorontsov.sql`

## 5.2 Parameters

- Modify `p_report_month` to change the analyzed month.
- Modify `p_region_filter` to filter by region or use ALL.

# Limitations and Ideas for Improvement

## Current Limitations

- No indexing or performance optimization.
- Small data volumes only.
- No time zone handling.
- Simplified churn logic.

## Possible Improvements

- Add indexes and partitions for large datasets.
- Introduce daily snapshots for subscriptions.
- Extend usage analytics with content-level metrics.
- Convert the report into a stored procedure or dashboard feed.