REPORT
ON
DTH SERVICE PROVIDER

Submitted By

Abhishek Bhosle (2024H1120217P)
Naggender Singh (2024H1120216P)
Shiven Keshav (2024H1120268P)

Submitted To

Dr L. Rajya Lakshmi

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS
BITS PILANI

Introduction:

This project presents a comprehensive data warehouse solution for Direct-to-Home (DTH) service providers, designed to analyse content performance, understand unsubscribed customers and their reasons, and evaluate subscriber engagement. Together, these components provide a holistic view of service performance and content effectiveness, enabling granular analysis of viewer behaviour and consumption trends. This solution supports data-driven strategies aimed at optimizing customer retention and driving revenue growth in a highly competitive market environment.

Problem Statement:

In competitive DTH market, providers face the challenge of managing vast amounts of data from subscriber and un subscription details to content performance and engagement metrics spread across multiple systems. This fragmented data environment hinders the ability to obtain a complete picture of business performance and customer behaviour. Our project addresses these challenges by developing a centralized data warehouse that consolidates data from disparate sources, integrating new dimensions such as content and series details, alongside enhanced customer engagement metrics. The solution is designed to handle large data volumes, execute complex queries efficiently, and generate comprehensive reports for various stakeholders. By bringing together subscription, un subscription, content, and engagement data into a unified platform, the data warehouse enables more informed decision-making, operational efficiency, and strategic customer retention, ultimately elevating the overall viewer experience.

1. Customer Engagement Analysis:

Objective:
Learn how and when people watch TV so we can offer better shows and tailor their experience.

Insights & Actions:
Peak Viewing Times: Most viewers tune in on weekends and evenings, so schedule new releases then.
Genre Preferences: When regional-language content scores high, add more shows

in that language.
Personalization: Binge-watchers get full-season recommendations to keep them watching longer.

2. Feedback & Service Quality Analysis:

Objective:
Find and fix the main reasons customers complain to keep them happy.

Insights & Actions:
Technical Issues: In areas with frequent signal drops, upgrade local network equipment.
Content Gaps: If viewers ask for sports channels, add cricket and other popular sports coverage.
Pricing Feedback: When "too expensive" is common, launch lower-priced budget packs.

3. Subscriber Lifecycle Analysis:

Objective:
Determine subscriber tenure and uncover cancellation drivers. Leverage these insights to reduce churn rates and increase new sign-ups**.**

Insights & Actions:
Churn Hotspots: Basic plans see high drop-off after three months, so introduce loyalty bonuses then.
Premium Plans: Premium plans have low cancellation rates but fewer sign-ups, so we give free trials or short upgrades to let more people try them.
Geographic Trends: Tier-2 cities retain better with regional packs, so expand localized offerings there.

4. **Unsubscriber Root-Cause Analysis:**

Objective:
Spot what drives people to quit and stop them before they leave.

Insights & Actions:
Price Sensitivity: Roll out mid-tier plans for users who find existing prices too high.
Technical Issues: Partner with local ISPs to boost signal quality in ZIP codes with heavy churn.
Competitive Pressure: Bundle our DTH service with OTT apps like Netflix to stay ahead of rivals.

Business Motivation:

The primary business motivation for implementing this project to design a data warehouse for a Direct-to-Home (DTH) service provider revolves around achieving several strategic goals:

Enhanced Customer Personalization and Innovation
By integrating comprehensive subscriber behaviour, content consumption, and engagement data, the data warehouse enables advanced personalization strategies. This fosters innovation in tailored content recommendations, targeted promotional campaigns, and dynamic pricing models allowing the provider to offer a truly individualized viewing experience that can significantly boost customer loyalty and satisfaction.

Improved Marketing ROI and Strategic Partnerships
By leveraging detailed analytics on subscriber demographics, content preferences, and engagement metrics, the DTH provider can refine marketing strategies to target high-value customer segments. Furthermore, insights into content performance and partner channel metrics facilitate more effective negotiations and collaborations with content providers, ultimately enhancing return on marketing investments and creating new revenue opportunities.

Dynamic Pricing and Subscription Management
By consolidating customer usage data, demand patterns, and seasonal trends in the data warehouse, the provider can implement flexible pricing strategies that adapt in real time. Through personalized discounts, bundle offers, and tiered subscription plans, the company maximizes revenue while offering customers tailored, value-driven pricing options.

Cross-Selling and Upselling Opportunities
By storing detailed viewing histories and genre preferences in the data warehouse, the provider identifies which customers are most receptive to add-ons. Targeted offers premium content bundles, pay-per-view events, or combined internet and streaming packages drive incremental revenue and deepen engagement. Real-time campaign triggers can surface the right offer immediately after a big match or new series premiere. Closed-loop reporting then tracks uptake and refines future promotions, continuously improving personalization and revenue potential.

Agility in Business Transformation
A modern, scalable data warehouse supports agile business processes and faster decision-making. With a dynamic and interactive analytics environment, the organization can quickly adapt to market changes, launch innovative products or services, and respond to competitive pressures. This agility is crucial in a rapidly evolving media landscape where timely insights directly influence market positioning and growth.

Requirements for DTH Service Data Warehouse:

Multidimensional Conceptual View
The data warehouse must provide an analytical, multidimensional model. Design organizes data into distinct dimensions Customer, Plan, Time, Reason, Channel, Content, and Series to allow stakeholders to analyse subscriber behaviour, content performance, and engagement from various perspectives.

Accessibility and Consistent Reporting
Offer a single, consistent view of data that is easily accessible regardless of data volume or the number of dimensions. By integrating data from subscription, un subscription, feedback, and customer engagement tables, the warehouse delivers

uniform, reliable reports that help in assessing performance and trends across various operational aspects.

Generic Dimensionality
All dimensions should be equivalent in structure and operational capability. Dimensions such as Customer, Plan, Time, Channel, Content, and Reason are uniformly designed to support consistent drill-down and roll-up operations, ensuring analytical processes are seamless across the board.

Comprehensive Data Integration:
Consolidate data from multiple sources such as subscriber databases, content management platforms, and customer support systems into a single, unified repository

Rigorous Data Quality:
Ensure data accuracy, completeness, and consistency through robust ETL processes that incorporate extensive validation and cleansing, providing reliable insights for analysis.

Flexible Reporting and Unlimited Aggregations
Support at least 15 dimensions and accommodate unlimited aggregate measures. The system's design enables complex analytical queries that aggregate data across multiple dimensions (e.g., subscriber demographics, content performance, and engagement metrics) to support strategic decisions on customer retention, content strategy, and revenue optimization.

Specific Analytics Requirements:

Unsubscription Analytics
Examine un subscription patterns by linking data from the Un subscription Fact and Reason Dimension tables. This analysis reveals key factors driving churn such as pricing, content dissatisfaction, or technical issues and supports the development of retention strategies.

Subscriber Analytics
Evaluate subscriber demographics, viewing preferences, and plan selections by

leveraging data from the Customer and Subscription Fact tables. This analysis supports segmentation, churn prediction, and targeted marketing efforts.

Feedback Analysis

Analyse customer feedback in depth to uncover insights on service quality, channel selection, and overall satisfaction. By correlating feedback with subscriber and engagement data, we can identify specific areas for improvement and measure the impact of any changes implemented

Enhanced Engagement Analytics

With_Customer Engagement Fact and Series Monthly Aggregate Fact table, our system captures detailed viewership metrics_such as view counts, total viewing durations, and engagement scores_which provide a granular view of how customers interact with content across channels.

Financial Analytics

Track revenue streams, subscription renewals, and billing cycles through aggregated data from the Subscription Fact and Monthly Aggregate Fact tables, enabling precise financial forecasting and improved billing efficiency.

Information Package Diagrams:

Information Subject: Subscriber Analysis

Hierarchies/Categories	Dimensions				
	Time	**Plan**	**Customer**	**Promo**	**Event**
	Date/Day	Name	Name	Promo Name	Season Name
	Week	Package/Category	Email	Promo Type	Major Sport Event
	Month	Price	Address	Start Date	Promo Window Flag
	Quarter	Duration	City	End Date	
	Year		Zip Code	Discount	
				Channel Name	
	Facts: subscription_start_date, subscription_end_date, subscription_status				

Fig 1

Information Subject: Unsubscriber Analysis

Hierarchies/Categories	Dimensions		
	Time	**Reason**	**Customer**
	Date/Day	Category	Name
	Week	Description	Email
	Month		Address
	Quarter		City
	Year		Zip Code
	Facts: unsubscription_date		

Fig 2

Information Subject: Feedback Analysis

Hierarchies/Categories	Dimensions		
	Plan	**Channel**	**Customer**
	Name	Name	Name
	Price	Genre	Email
	Package	Language	Address
	Duration		City
			Zip Code
	Facts: feedback_comment		

Fig 3

Information Subject: Customer Engagement

Hierarchies/Categories	Dimensions				
	Time	**Channel**	**Customer**	**Content**	**Ad Exposure**
	Date/Day	Name	Name	Name	Ad Type
	Week	Genre	Email	Type	Impression Count
	Month	Language	Address	Series Name	Skip Flag
	Quarter		City	Release Date	
	Year		Zip Code	Language	
				Rating	
				Genre	
	Facts: view_count, viewing_duration, engagement_score				

Fig 4

Information Subject: Series Monthly Aggregate

Hierarchies/Categories	Dimensions		
	dim_month	**dim_series**	**Customer**
	Month	Series Name	Name
	Quarter		Email
	Year		Address
			City
			Zip Code
	Facts: total_view_count, total_viewing_duration, avg_engagement_score		

Fig 5

Information Subject: Monthly Aggregate

Hierarchies/Categories	Dimensions	
	dim_month	**Customer**
	Month	Name
	Quarter	Email
	Year	Address
		City
		Zip Code
	Facts: total_subscriptions, total_unsubscriptions, total_customers	

Fig 6

Data Cube Diagrams:

1. Feedback Fact Data Cube

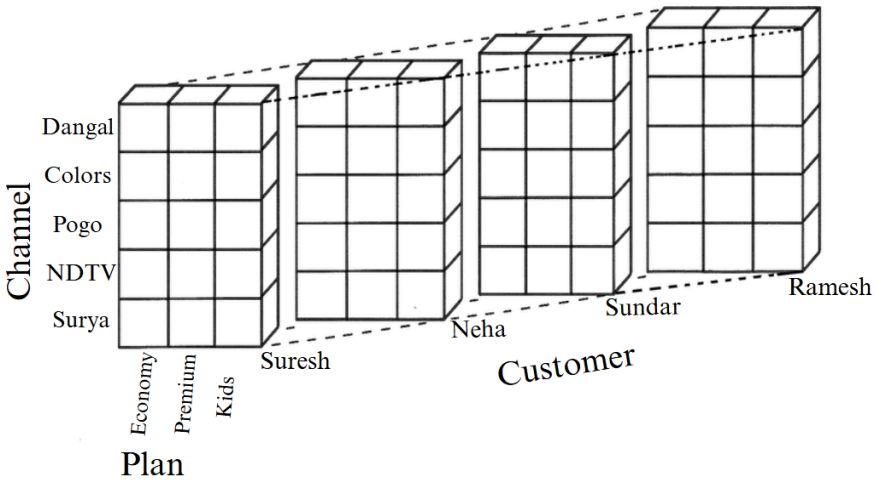

Fig 7

2. Unsubscription Fact Data Cube

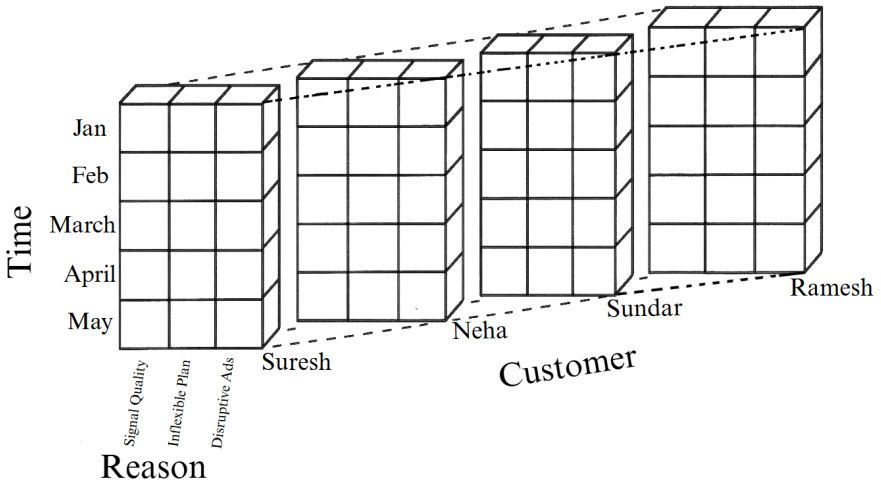

Fig 8

Dimension Tables:

1. **Customer_Dimension**

   Keys & Attributes: customer_id (PK), customer_name, customer_email, customer_address, customer_city, customer_state, customer_zipcode

   Purpose: Stores subscriber profile and contact details.

2. **Plan_Dimension**

   Keys & Attributes: plan_id (PK), plan_name, plan_price, channel_package, plan_duration

   Purpose: Defines each DTH package's price and channel lineup

3. **Time_Dimension**

   Keys & Attributes: time_id (PK), date_record, week_number, month_record, quarter_record, year_record

   Purpose: Provides a calendar hierarchy aligned to subscription and engagement events

4. **Dim_Month**

   Keys & Attributes: month_id (PK), month_record, quarter_record, year_record

   Purpose: Pre-aggregated month lookup for fast monthly roll-ups

5. **Reason_Dimension**

   Keys & Attributes: reason_id (PK), reason_category, reason_description

   Purpose: Captures standardized churn reasons (e.g., Price, Content, Technical)

6. **Channel_Dimension**

   Keys & Attributes: channel_id (PK), channel_name, channel_genre,

channel_language

Purpose**:** Lists every TV channel and its metadata

7. **Content_Dimension**

   Keys & Attributes: content_id (PK), episode_name, series_name, content_type, release_date, language, rating

   Purpose: Details individual episodes or series for content-level analysis

8. **Dim_Series**

   Keys & Attributes: series_id (PK), series_name

   Purpose: A derived dimension to group episodes into series

9. **Promotion_Dimension**

   Keys & Attributes: promotion_id (PK), promotion_name, promo_type, start_date, end_date, discount_pct, channel_name

   Purpose: Defines marketing or discount campaigns linked to subscriptions

10. **Event_Dimension**

    Keys & Attributes: event_id (PK), season_name, major_sport_event, promo_window_flag

    Purpose: Captures special broadcast events (e.g., World Cup matches, season premieres)

11. **Ad_Exposure_Dimension**

    Keys & Attributes: ad_exposure_id (PK), ad_type, impressions_count, skipped_flag

    Purpose: Tracks in-app or in-TV ad impressions and skips

12. Genre_Dimension

Keys & Attributes: genre_id (PK), genre_name

Purpose: Lists content genres (e.g., Drama, Comedy)

13. Content_Genre_Bridge

Keys & Attributes: content_id (FK), genre_id (FK)

Purpose: Many-to-many link between content items and genres

Fact Tables:

1. Subscription_Fact

Keys & Measures: subscription_id (PK), customer_id (FK), time_id (FK), plan_id (FK), promotion_id (FK), event_id (FK), subscription_start_date, subscription_end_date, subscription_status

Purpose: Records each subscription event along with any applied promotion or special event

2. Unsubscription_Fact

Keys & Measures: unsubscription_id (PK), customer_id (FK), time_id (FK), reason_id (FK), unsubscription_date

Purpose: Logs each churn event and its cause

3. Feedback_Fact

Keys & Measures: feedback_id (PK), customer_id (FK), plan_id (FK), channel_id (FK), feedback_comment

Purpose: Captures customer comments and ratings linked to specific plans or channels

4. **Monthly_Aggregate_Fact**

   Keys & Measures: monthly_aggregate_id (PK), customer_id (FK), time_id (FK), month_id (FK), total_subscriptions, total_unsubscriptions, total_customers

   Purpose: Pre-computes one-way monthly metrics for fast dashboarding

5. **Customer_Engagement_Fact**

   Keys & Measures: customer_engagement_id (PK), customer_id (FK), channel_id (FK), content_id (FK), time_id (FK), ad_exposure_id (FK), view_count, viewing_duration, engagement_score

   Purpose: Stores fine-grained viewer interactions and ad exposure details.

6. **Series_Monthly_Aggregate_Fact**

   Keys & Measures: series_monthly_aggregate_id (PK), series_id (FK), month_id (FK), customer_id (FK), total_view_count, total_viewing_duration, avg_engagement_score

   Purpose: Two-way aggregate summarizing how each customer watches entire series each month

Family Of Stars:

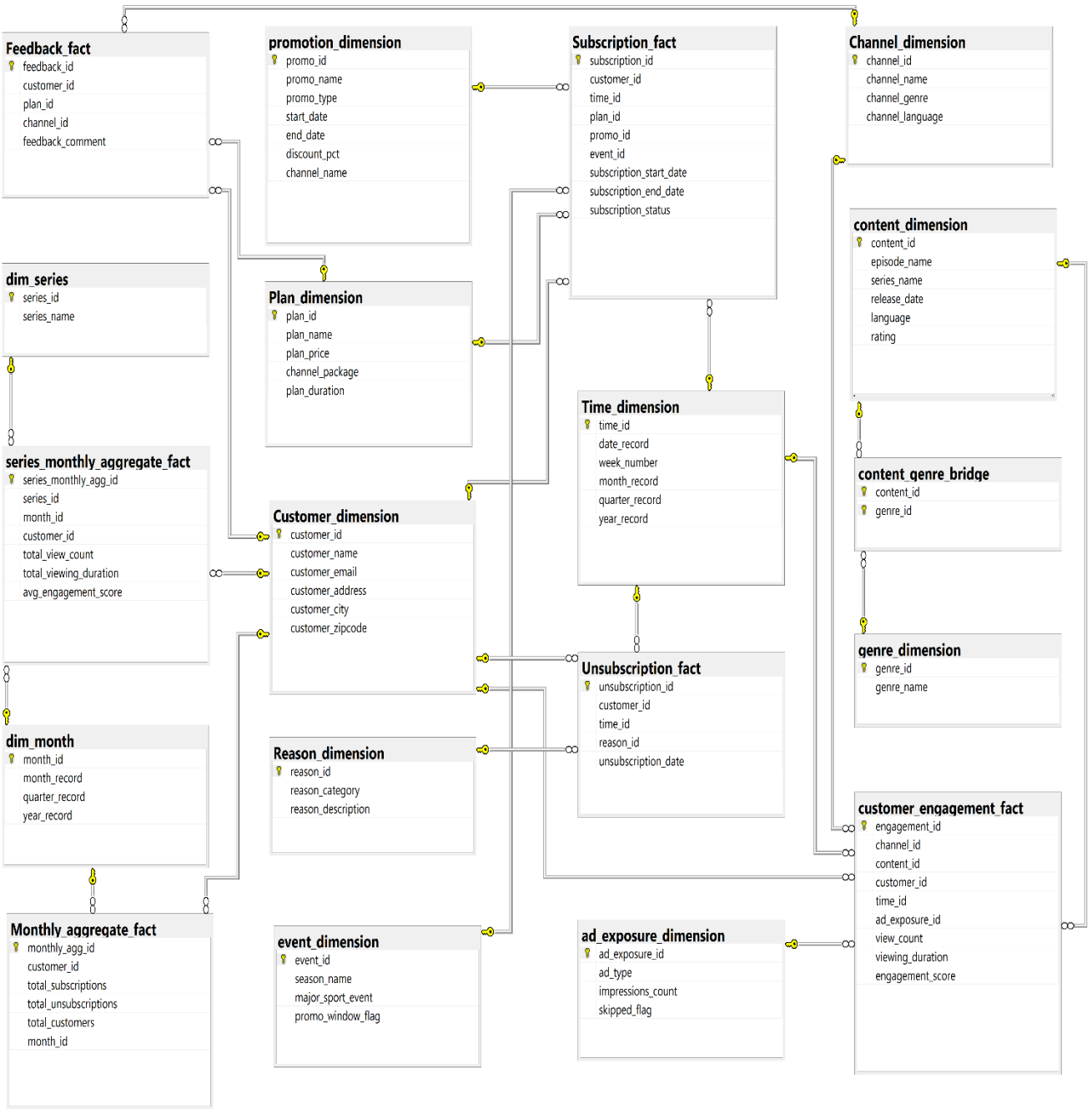

Fig 9

Star Schemas:

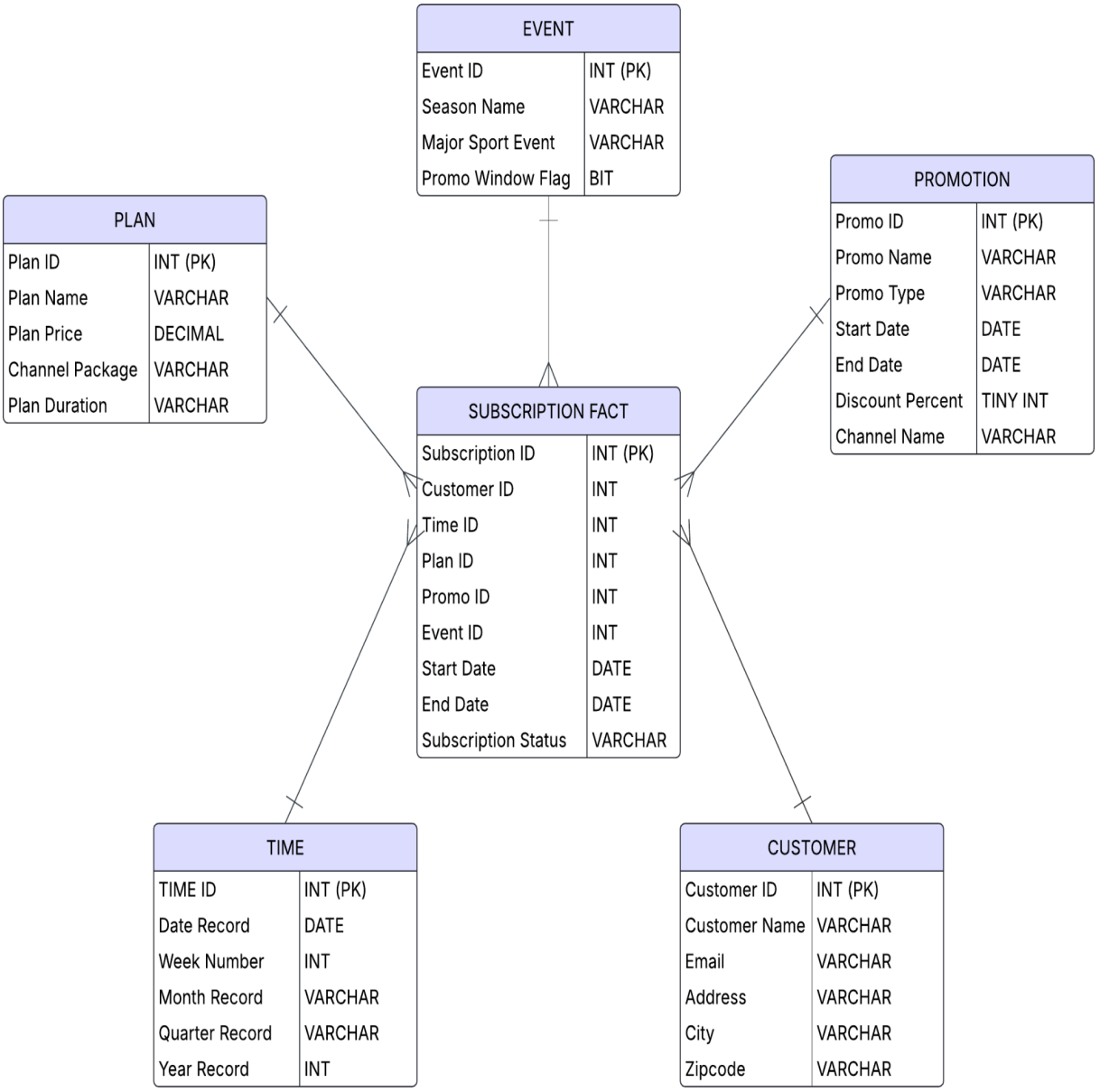

Fig 10

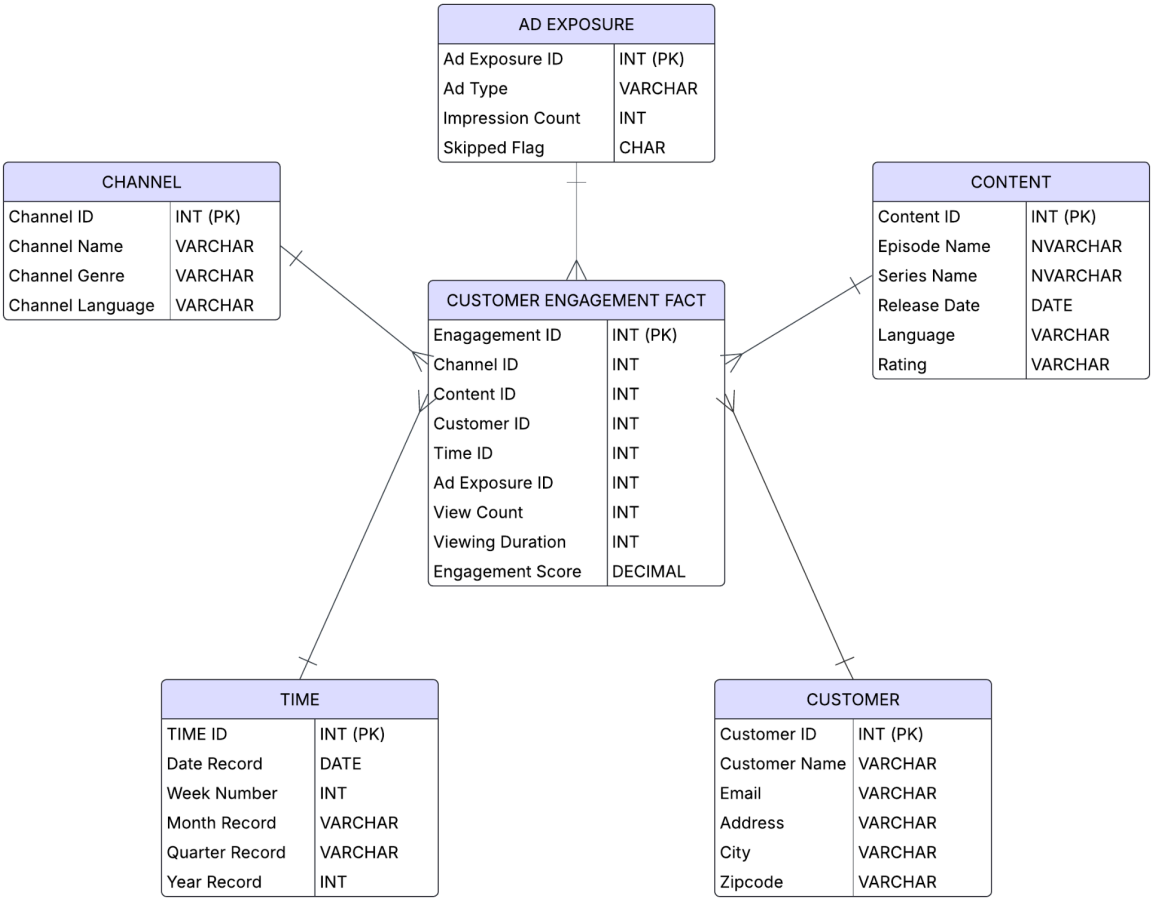
AD EXPOSURE
Ad Exposure ID
INT (PK)
Ad Type
VARCHAR
Impression Count
INT
Skipped Flag
CHAR
CHANNEL
Channel ID
INT (PK)
Channel Name
VARCHAR
Channel Genre
VARCHAR
Channel Language
VARCHAR
CONTENT
Content ID
INT (PK)
Episode Name
NVARCHAR
Series Name
NVARCHAR
Release Date
DATE
Language
VARCHAR
Rating
VARCHAR
CUSTOMER ENGAGEMENT FACT
Enagagement ID
INT (PK)
Channel ID
INT
Content ID
INT
Customer ID
INT
Time ID
INT
Ad Exposure ID
INT
View Count
INT
Viewing Duration
INT
Engagement Score
DECIMAL
TIME
TIME ID
INT (PK)
Date Record
DATE
Week Number
INT
Month Record
VARCHAR
Quarter Record
VARCHAR
Year Record
INT
CUSTOMER
Customer ID
INT (PK)
Customer Name
VARCHAR
Email
VARCHAR
Address
VARCHAR
City
VARCHAR
Zipcode
VARCHAR

Fig 11

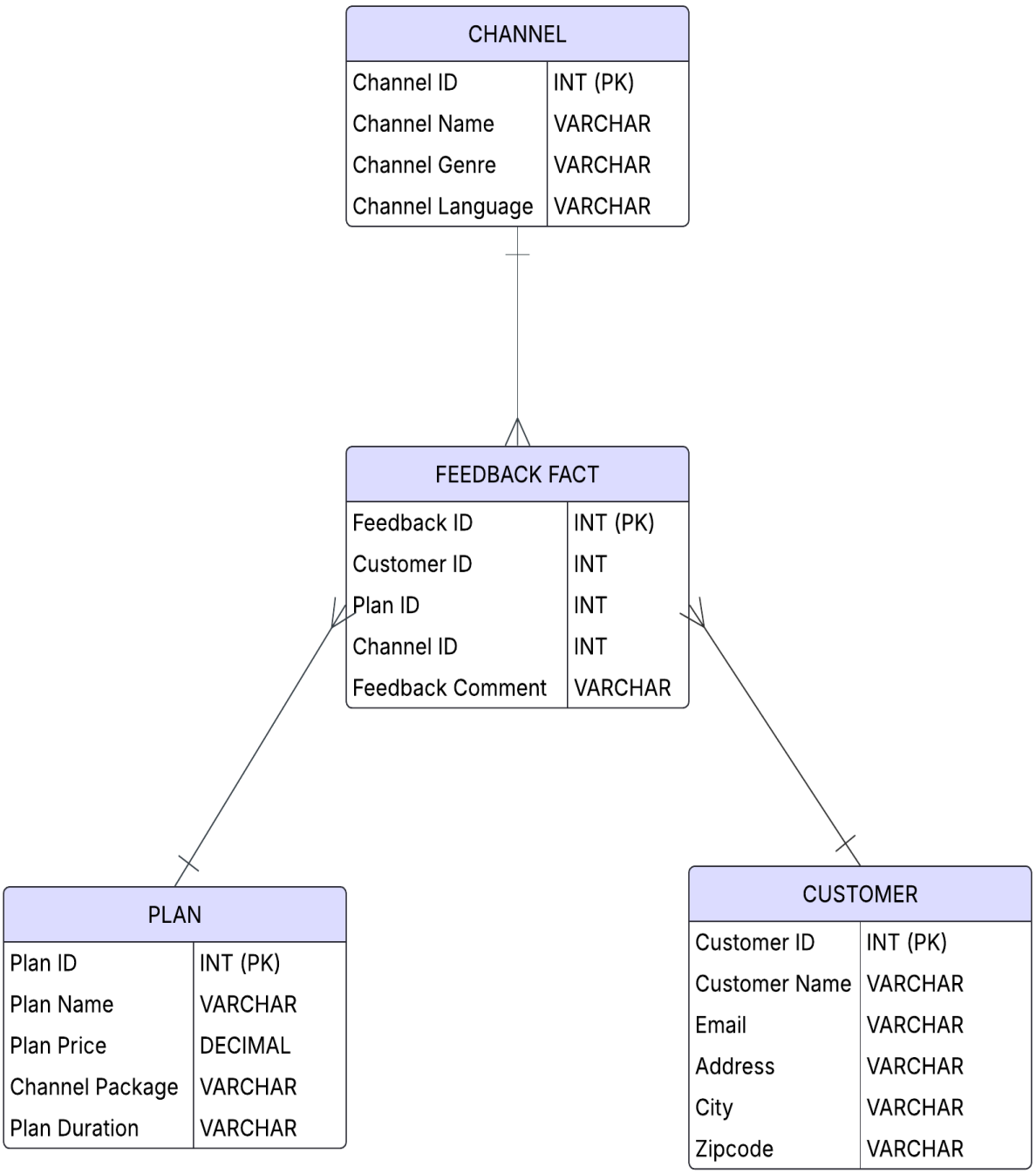
CHANNEL
Channel ID
INT (PK)
Channel Name
VARCHAR
Channel Genre
VARCHAR
Channel Language
VARCHAR
FEEDBACK FACT
Feedback ID
INT (PK)
Customer ID
INT
Plan ID
INT
Channel ID
INT
Feedback Comment
VARCHAR
PLAN
Plan ID
INT (PK)
Plan Name
VARCHAR
Plan Price
DECIMAL
Channel Package
VARCHAR
Plan Duration
VARCHAR
CUSTOMER
Customer ID
INT (PK)
Customer Name
VARCHAR
Email
VARCHAR
Address
VARCHAR
City
VARCHAR
Zipcode
VARCHAR

Fig 12

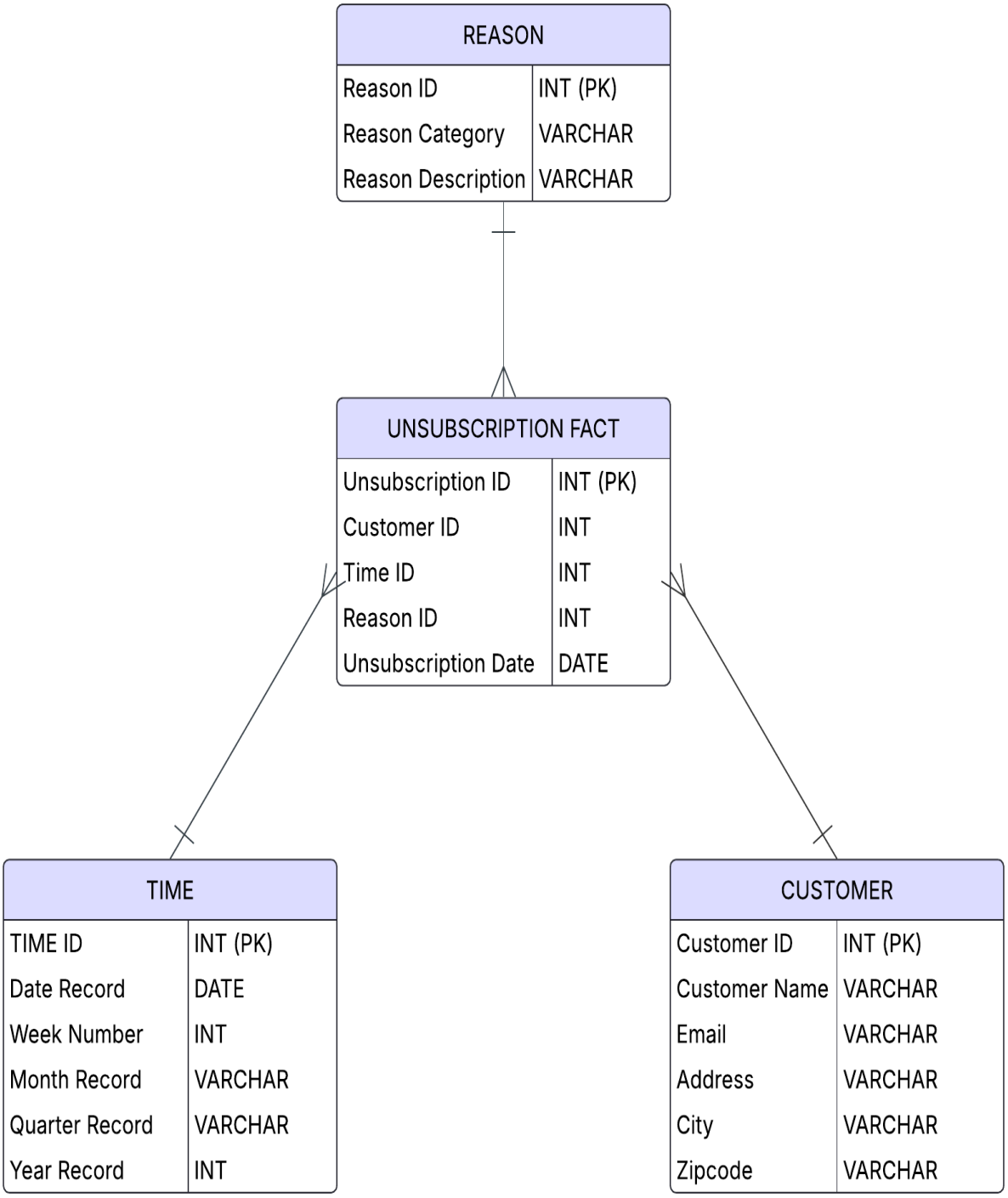
REASON
Reason ID
INT (PK)
Reason Category
VARCHAR
Reason Description
VARCHAR
UNSUBSCRIPTION FACT
Unsubscription ID
INT (PK)
Customer ID
INT
Time ID
INT
Reason ID
INT
Unsubscription Date
DATE
TIME
TIME ID
INT (PK)
Date Record
DATE
Week Number
INT
Month Record
VARCHAR
Quarter Record
VARCHAR
Year Record
INT
CUSTOMER
Customer ID
INT (PK)
Customer Name
VARCHAR
Email
VARCHAR
Address
VARCHAR
City
VARCHAR
Zipcode
VARCHAR

Fig 13

SQL Queries:

Query 1: List of Un-Subscribed Customers with Reasons.

```sql
SELECT
    u.unsubscription_date,
    cd.customer_id,
    cd.customer_name,
    cd.customer_email,
    p.plan_name,
    r.reason_category,
    r.reason_description
FROM Unsubscription_fact AS u
INNER JOIN Customer_dimension AS cd
    ON u.customer_id = cd.customer_id
INNER JOIN Reason_dimension AS r
    ON u.reason_id = r.reason_id
INNER JOIN Subscription_fact AS sf
    ON u.customer_id = sf.customer_id
   AND u.unsubscription_date BETWEEN sf.subscription_start_date AND sf.subscription_end_date
INNER JOIN Plan_dimension AS p
    ON sf.plan_id = p.plan_id
ORDER BY
    u.unsubscription_date DESC;
```

120 %

Results | Messages

	unsubscription_date	customer_id	customer_name	customer_email	plan_name	reason_category	reason_description
1	2024-05-07	16	Pooja Malik	pooja.malik@example.com	Sports Plan	Price	Plan is too expensive compared to competitors
2	2024-05-07	186	Sukhwinder Kaur	sukhwinder.kaur@example.com	Sports Plus	Ad Overload	Excessive advertisements during programming
3	2024-05-06	192	Divya Nair	divya.nair@example.com	Basic	Financial Constraints	Customer unable to afford the subscription
4	2024-05-06	95	Sanjay Kumar	sanjay.kumar@example.com	Gold Plan	Competitor Innovation	Competitor offers more innovative services
5	2024-05-05	116	Sita Menon	sita.menon@example.com	International Plan	Service Reliability	Frequent interruptions affecting reliability
6	2024-05-05	35	Vikas Verma	vikas.verma@example.com	Lifestyle Plan	Unfriendly Interface	Difficult and outdated user interface on set-top box
7	2024-05-04	200	Sneha Menon	sneha.menon@example.com	Value Plan	Signal Quality	Poor signal strength and constant disruptions
8	2024-05-04	156	Kiran Patel	kiran.patel@example.com	Kids Plan	Technical Issues	Frequent technical glitches and signal loss
9	2024-05-03	60	Rina Chatterjee	rina.chatterjee@example.com	Flex Plan	Service	Poor customer service experience overall
10	2024-05-03	140	Neeta Patel	neeta.patel@example.com	Standard Plan	Contract Disputes	Disagreements with contract terms and conditions
11	2024-05-02	106	Rajdeep Singh	rajdeep.singh@example.com	Gold Plan	Relocation	Moving to an area where the service is unavailable
12	2024-05-02	179	Rajesh Desai	rajesh.desai@example.com	HD Plan	Change in Preferences	Shift in viewing habits and content preferences

Query executed successfully.

Query 2: Count of Un-Subscribed Customers by reason category.

```sql
-- Count of Un-Subscribed Customers by Reason Category
SELECT
    r.reason_category,
    COUNT(DISTINCT u.customer_id) AS num_unsubscribed_customers
FROM Unsubscription_fact AS u
JOIN Reason_dimension  AS r
  ON u.reason_id = r.reason_id
GROUP BY
    r.reason_category
ORDER BY
    num_unsubscribed_customers DESC;
```

159 %

Results | Messages

	reason_category	num_unsubscribed_customers
1	Price	6
2	Technical Issues	4
3	Ad Overload	4
4	Billing Issues	3
5	Change in Preferences	3
6	Streaming Services	3
7	Signal Quality	3
8	Relocation	3
9	Service	3
10	Inadequate Features	3
11	Unfriendly Interface	3
12	Maintenance Downtime	2
13	Package Complexity	2
14	Service Reliability	2
15	Unclear Pricing	2
16	Competitor Innovation	2
17	Contract Disputes	2
18	Customer Loyalty Issues	2
19	Customer Service	2

Query executed successfully.

Query 3: List of Active Subscription with Plan Details.

```sql
-- List of Active Subscriptions with Plan Details
SELECT
    s.customer_id,
    cd.customer_name,
    p.plan_name,
    p.plan_price,
    p.channel_package,
    p.plan_duration,
    s.subscription_start_date,
    s.subscription_end_date
FROM Subscription_fact AS s
JOIN Plan_dimension     AS p
  ON s.plan_id = p.plan_id
JOIN Customer_dimension AS cd
  ON s.customer_id = cd.customer_id
WHERE s.subscription_status = 'Active'
ORDER BY s.subscription_start_date DESC;
```

159 %

Results | Messages

	customer_id	customer_name	plan_name	plan_price	channel_package	plan_duration	subscription_start_date	subscription_end_date
1	144	Sonal Shah	Family Plan	499.00	Family Pack	30 days	2024-04-30	2024-06-14
2	120	Neeta Menon	Drama Plan	249.00	Drama Channels	30 days	2024-04-30	2024-05-15
3	158	Sunita Desai	Family Plus	549.00	Family Plus Pack	30 days	2024-04-29	2024-06-13
4	33	Ajay Kumar	Movie Pack	399.00	Movie Channels	30 days	2024-04-29	2024-05-29
5	131	Rajesh Kumar	Focus Plan	299.00	Focused Channels	30 days	2024-04-28	2024-05-28
6	43	Sivakumar Iyer	Economy Plan	199.00	Economy Channels	30 days	2024-04-27	2024-05-12
7	118	Divya Nair	Adventure Plan	399.00	Adventure Channels	30 days	2024-04-27	2024-05-27
8	197	Arun Kumar	Kids Plan	249.00	Kids Pack	30 days	2024-04-26	2024-05-11
9	55	Pradeep Roy	Ultra HD Plan	549.00	Ultra HD Pack	30 days	2024-04-26	2024-06-10
10	19	Amitabh Joshi	Digital Plan	299.00	Digital Channels	30 days	2024-04-25	2024-05-25

Query executed successfully.

Query 4: Feedback Counts by Channel (Sentiment Proxy)

```sql
-- Feedback Counts by Channel (Sentiment Proxy)

SELECT
    ch.channel_name,
    COUNT(f.feedback_id) AS feedback_count
FROM feedback_fact f
JOIN channel_dimension ch
    ON f.channel_id = ch.channel_id
GROUP BY
    ch.channel_name
ORDER BY
    feedback_count DESC;
```

133 %

Results | Messages

	channel_name	feedback_count
1	Colors	5
2	Zee TV	3
3	Zee TV Tamil	2
4	Public TV	2
5	ET Now	2
6	Sony BBC Earth	2
7	ABP News	2
8	Star Utsav	2
9	Ten Golf	2
10	Times Now	1
11	UTV Movies	1
12	WION	1
13	Zee Cinema	1
14	Zee Kannada	1

Query executed successfully.

Query 5: Highest Rated Content with Engagement, Genre & Ad-Exposure

```sql
-- 1) Highest Rated Content with Engagement, Genre & Ad-Exposure
SELECT
    c.series_name,
    c.rating,
    gd.genre_name,
    ae.ad_type,
    COUNT(ce.engagement_id)            AS total_engagement_records,
    SUM(ce.view_count)                 AS total_views,
    ROUND(AVG(ce.engagement_score),2)  AS avg_engagement_score
FROM content_dimension AS c
JOIN content_genre_bridge AS cgb
  ON c.content_id = cgb.content_id
JOIN genre_dimension AS gd
  ON cgb.genre_id = gd.genre_id
JOIN customer_engagement_fact AS ce
  ON c.content_id = ce.content_id
JOIN ad_exposure_dimension AS ae
  ON ce.ad_exposure_id = ae.ad_exposure_id
GROUP BY
    c.series_name,
    c.rating,
    gd.genre_name,
    ae.ad_type
HAVING c.rating IS NOT NULL
ORDER BY
    c.rating DESC,
    avg_engagement_score DESC;
```

100 %

Results | Messages

	series_name	rating	genre_name	ad_type	total_engagement_records	total_views	avg_engagement_score
1	The Simpsons	TV-PG	Animated	Pre-Roll Fullscreen	1	170	4.120000
2	The Big Bang Theory	TV-PG	Comedy	Mid-Roll Overlay	2	363	3.540000
3	The Simpsons	TV-PG	Animated	Mid-Roll Overlay	1	102	3.530000
4	The Simpsons	TV-PG	Animated	Banner (Header)	1	183	3.180000
5	The Simpsons	TV-PG	Animated	Teaser Overlay	2	280	3.160000
6	The Big Bang Theory	TV-PG	Comedy	Post-Roll Fullscreen	2	274	3.130000
7	The Big Bang Theory	TV-PG	Comedy	Fullscreen Interstitial	2	275	3.110000
8	The Simpsons	TV-PG	Animated	Side-Banner (Left)	1	157	3.090000
9	The Big Bang Theory	TV-PG	Comedy	Side-Banner (Right)	1	125	3.010000
10	The Simpsons	TV-PG	Animated	Fullscreen Interstitial	1	121	2.750000

Query executed successfully.

Query 6: Average Subscription Duration by Plan Price, with & without Promo for Comparison

```sql
-- 3) Average Subscription Duration by Plan Price, WITH & WITHOUT Promo for Comparison
SELECT
    p.plan_name,
    p.plan_price,
    pd.promo_name,
    ROUND(AVG(DATEDIFF(DAY, s.subscription_start_date, s.subscription_end_date)), 1) AS avg_sub_duration_with_promo,
    ROUND(o.avg_sub_duration_overall, 1) AS avg_sub_duration_overall
FROM plan_dimension      AS p
JOIN subscription_fact   AS s
  ON p.plan_id = s.plan_id
JOIN promotion_dimension AS pd
  ON s.promo_id = pd.promo_id
JOIN (
    SELECT
      plan_id,
      AVG(DATEDIFF(DAY, subscription_start_date, subscription_end_date)) AS avg_sub_duration_overall
    FROM subscription_fact
    WHERE subscription_end_date IS NOT NULL
    GROUP BY plan_id
) AS o
  ON p.plan_id = o.plan_id
WHERE s.subscription_end_date IS NOT NULL
GROUP BY
    p.plan_name,
    p.plan_price,
    pd.promo_name,
    o.avg_sub_duration_overall
ORDER BY
    p.plan_price ASC;
```

91 %

Results | Messages

	plan_name	plan_price	promo_name	avg_sub_duration_with_promo	avg_sub_duration_overall
1	Economy Plan	199.00	CNN-News18 Rugby World Cup Offer	15	15
2	Economy Plan	199.00	Times Now Tour de France Offer	15	15
3	Express Plan	199.00	CNN-News18 Navratri Offer	29	29
4	Music Plan	199.00	Republic TV Summer Offer	15	15
5	News Plan	199.00	CNN-News18 Navratri Offer	15	25
6	News Plan	199.00	Times Now Super Bowl Offer	29	25
7	News Plan	199.00	Zee News ICC Champions Trophy Offer	29	25
8	Retro Plan	229.00	Zee News ICC Champions Trophy Offer	29	29
9	Drama Plan	249.00	Times Now Tour de France Offer	15	15
10	Drama Plan	249.00	Zee News ICC Champions Trophy Offer	15	15

Query executed successfully.

Query 7: Monthly Engagement for Series by Customer Segment (City)

```sql
-- Monthly Engagement for Series by Customer Segment (City)
SELECT
    ds.series_name,
    cd.customer_city,
    dm.month_record,
    SUM(sma.total_view_count) AS total_series_views,
    AVG(sma.avg_engagement_score) AS avg_series_engagement
FROM series_monthly_aggregate_fact sma
JOIN dim_series ds
    ON sma.series_id = ds.series_id
JOIN customer_dimension cd
    ON sma.customer_id = cd.customer_id
JOIN dim_month dm
    ON sma.month_id = dm.month_id
GROUP BY
    ds.series_name,
    cd.customer_city,
    dm.month_record
ORDER BY
    ds.series_name,
    cd.customer_city,
    dm.month_record;
```

110 %

Results | Messages

	series_name	customer_city	month_record	total_series_views	avg_series_engagement
1	Breaking Bad	Ludhiana	April	133	3.530000
2	Breaking Bad	Ludhiana	May	64	3.180000
3	Breaking Bad	Mumbai	April	1550	4.500000
4	Breaking Bad	Patna	April	193	2.580000
5	Breaking Bad	Surat	April	330	3.320000
6	Breaking Bad	Surat	May	180	3.330000
7	Breaking Bad	Vadodara	April	352	3.486666
8	Friends	Bangalore	April	875	3.800000
9	Friends	Coimbatore	April	309	4.160000
10	Friends	Ludhiana	April	69	2.260000
11	Friends	Ludhiana	May	159	2.420000
12	Friends	Patna	May	376	4.025000
13	Friends	Surat	April	293	3.240000

Query executed successfully.

Query 8: Top 5 plans subscribed by the customers.

```sql
-- Top 5 Most-Subscribed Plans
SELECT TOP 5
    p.plan_id,
    p.plan_name,
    COUNT(*) AS total_subscriptions
FROM Subscription_fact AS s
JOIN Plan_dimension    AS p
  ON s.plan_id = p.plan_id
GROUP BY
    p.plan_id,
    p.plan_name
ORDER BY
    total_subscriptions DESC;
```

175 %

Results | Messages

	plan_id	plan_name	total_subscriptions
1	2	Premium	6
2	9	Kids Plan	6
3	5	Sports Plan	5
4	6	Entertainment Plan	5
5	7	HD Plan	4

Query executed successfully.

Query 9: Busiest channel/month combinations by total viewing duration.

```sql
-- Identify the busiest channel/month combinations by total viewing duration.
SELECT TOP 20
  ch.channel_name,
  dm.month_record,
  SUM(cef.viewing_duration) AS total_viewing_minutes
FROM customer_engagement_fact cef
JOIN Channel_dimension ch ON cef.channel_id = ch.channel_id
JOIN Time_dimension t     ON cef.time_id    = t.time_id
JOIN dim_month dm         ON dm.month_record = t.month_record
GROUP BY ch.channel_name, dm.month_record
ORDER BY total_viewing_minutes DESC;
```

146 %

Results Messages

	channel_name	month_record	total_viewing_minutes
1	Times Now	April	1832
2	CNN-News18	April	1600
3	NDTV	April	1487
4	NDTV	May	785
5	CNN-News18	May	764
6	Times Now	May	503

Query 10: Compute the month-over-month change in view count for each series.

```sql
-- Compute the month-over-month change in view count for each series.
SELECT TOP 100
  sma.series_id,
  ds.series_name,
  sma.month_id,
  dm.month_record,
  sma.total_view_count,
  sma.total_view_count
    - LAG(sma.total_view_count) OVER (
        PARTITION BY sma.series_id
        ORDER BY sma.month_id
      ) AS mom_growth
FROM series_monthly_aggregate_fact sma
JOIN dim_series ds ON sma.series_id = ds.series_id
JOIN dim_month dm ON sma.month_id = dm.month_id
ORDER BY mom_growth DESC;
```

133 %

Results Messages

	series_id	series_name	month_id	month_record	total_view_count	mom_growth
1	10	The Crown	2	May	182	107
2	9	The Big Bang Theory	1	April	190	105
3	14	Vikings	1	April	186	98
4	10	The Crown	1	April	155	87
5	3	Game of Thrones	2	May	178	80
6	13	The Simpsons	2	May	180	78
7	10	The Crown	1	April	134	76
8	5	How I Met Your Mother	1	April	187	75
9	7	Peaky Blinders	1	April	180	71
10	2	Friends	1	April	137	68
11	9	The Big Bang Theory	1	April	191	63
12	15	Westworld	2	May	176	62
13	4	House of Cards	2	May	165	60
14	8	Stranger Things	2	May	140	59
15	11	The Mandalorian	2	May	185	56

Query 11: Customers who have watched the most.

```sql
-- Identify the customers who have watched the most minutes of content.s
SELECT TOP 10
  cef.customer_id,
  c.customer_name,
  SUM(cef.viewing_duration) AS total_viewing_minutes
FROM customer_engagement_fact cef
JOIN Customer_dimension c ON cef.customer_id = c.customer_id
GROUP BY cef.customer_id, c.customer_name
ORDER BY total_viewing_minutes DESC;
```

177 %

Results | Messages

	customer_id	customer_name	total_viewing_minutes
1	50	Meera Krishnan	69
2	49	Sundar Reddy	68
3	48	Priya Nair	67
4	47	Arvind Menon	66
5	46	Nandini Rao	65
6	45	Karthik Kumar	64
7	44	Divya Ramesh	63
8	43	Sivakumar Iyer	62
9	42	Lakshmi Subramanian	61
10	41	Murali Krishnan	60

Query 12: Monthly Subscription and Unsubscription Roll-Up.

```sql
-- Monthly Subscription and Unsubscription Roll-Up:
WITH subs AS (
    SELECT
      t.month_record,
      COUNT(*) AS cnt
    FROM Subscription_fact sf
    JOIN Time_dimension t
      ON sf.time_id = t.time_id
    GROUP BY t.month_record
),
unsubs AS (
    SELECT
      t.month_record,
      COUNT(*) AS cnt
    FROM Unsubscription_fact uf
    JOIN Time_dimension t
      ON uf.time_id = t.time_id
    GROUP BY t.month_record
)
SELECT
  dm.month_id,
  dm.month_record,
  COALESCE(subs.cnt, 0)   AS total_subscriptions,
  COALESCE(unsubs.cnt, 0) AS total_unsubscriptions,
  COALESCE(subs.cnt, 0) - COALESCE(unsubs.cnt, 0) AS net_change
FROM dim_month dm
LEFT JOIN subs   ON subs.month_record   = dm.month_record
LEFT JOIN unsubs ON unsubs.month_record = dm.month_record
ORDER BY dm.month_id;
```

110 %

Results | Messages

	month_id	month_record	total_subscriptions	total_unsubscriptions	net_change
1	1	April	60	58	2
2	2	May	40	20	20

Query 13: Feedback Rollup Analysis by Customer City & Plan.

```sql
-- Feedback Rollup Analysis by Customer City & Plan.
SELECT
  COALESCE(c.customer_city,      'All Cities') AS CustomerCity,
  COALESCE(p.plan_name,          'All Plans')  AS PlanName,
  COUNT(*)                                     AS TotalFeedback,
  SUM(CASE WHEN f.feedback_comment LIKE '%good%'
             OR f.feedback_comment LIKE '%excellent%'
             OR f.feedback_comment LIKE '%great%'
           THEN 1 ELSE 0 END)       AS PositiveCount,
  SUM(CASE WHEN f.feedback_comment LIKE '%bad%'
             OR f.feedback_comment LIKE '%poor%'
             OR f.feedback_comment LIKE '%complaint%'
           THEN 1 ELSE 0 END)       AS NegativeCount,
  SUM(CASE WHEN f.feedback_comment NOT LIKE '%good%'
             AND f.feedback_comment NOT LIKE '%excellent%'
             AND f.feedback_comment NOT LIKE '%great%'
             AND f.feedback_comment NOT LIKE '%bad%'
             AND f.feedback_comment NOT LIKE '%poor%'
             AND f.feedback_comment NOT LIKE '%complaint%'
           THEN 1 ELSE 0 END)       AS NeutralCount
FROM feedback_fact AS f
INNER JOIN customer_dimension AS c
  ON f.customer_id = c.customer_id
INNER JOIN plan_dimension     AS p
  ON f.plan_id     = p.plan_id
GROUP BY ROLLUP (c.customer_city, p.plan_name)
ORDER BY
  GROUPING(c.customer_city) DESC,
  GROUPING(p.plan_name)     DESC,
  CustomerCity,
  PlanName;
```

91 %

Results | Messages

	CustomerCity	PlanName	TotalFeedback	PositiveCount	NegativeCount	NeutralCount
1	All Cities	All Plans	58	18	8	32
2	Ahmedabad	All Plans	6	1	1	4
3	Bhopal	All Plans	5	1	0	4
4	Chandigarh	All Plans	5	1	0	4
5	Chennai	All Plans	4	0	2	2
6	Coimbatore	All Plans	2	0	1	1
7	Delhi	All Plans	1	1	0	0
8	Hyderabad	All Plans	2	1	1	0
9	Indore	All Plans	5	2	1	2

Query executed successfully.

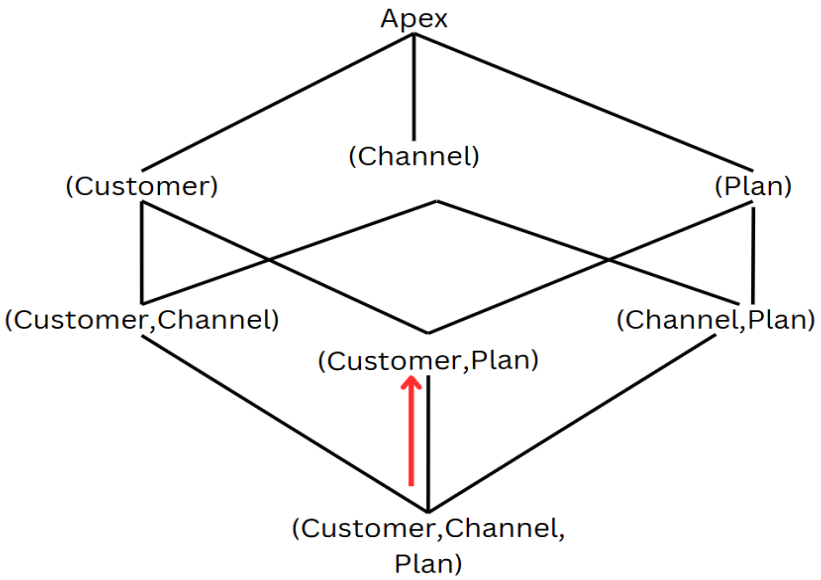

Fig 14

Query 14: Feedback Sentiment by Channel.

```sql
-- Feedback Sentiment by Channel:
SELECT
  ch.channel_name,
  CASE
    WHEN f.feedback_comment LIKE '%good%'
      OR f.feedback_comment LIKE '%excellent%' THEN 'Positive'
    WHEN f.feedback_comment LIKE '%poor%'
      OR f.feedback_comment LIKE '%bad%' THEN 'Negative'
    ELSE 'Neutral'
  END AS sentiment,
  COUNT(*) AS feedback_count
FROM Feedback_fact f
JOIN Channel_dimension ch ON f.channel_id = ch.channel_id
GROUP BY
  ch.channel_name,
  CASE
    WHEN f.feedback_comment LIKE '%good%'
      OR f.feedback_comment LIKE '%excellent%' THEN 'Positive'
    WHEN f.feedback_comment LIKE '%poor%'
      OR f.feedback_comment LIKE '%bad%' THEN 'Negative'
    ELSE 'Neutral'
  END
ORDER BY ch.channel_name, feedback_count DESC;
```

110 %

Results | Messages

	channel_name	sentiment	feedback_count
1	&TV	Neutral	1
2	ABP News	Neutral	1
3	ABP News	Positive	1
4	Animal Planet	Neutral	1
5	Big Magic	Neutral	1
6	Colors	Neutral	2
7	Colors	Negative	2
8	Colors	Positive	1
9	Colors Kannada	Neutral	1
10	Colors Tamil	Neutral	1
11	DD Sports	Neutral	1

Query executed successfully.

Query 15: Average Time to Churn by Reason.

```sql
-- Average Time to Churn by Reason:
SELECT
  r.reason_category,
  r.reason_description,
  ROUND(AVG(DATEDIFF(day, s.subscription_start_date, u.unsubscription_date)), 1) AS avg_days_to_unsub
FROM Unsubscription_fact u
JOIN Subscription_fact s
  ON u.customer_id = s.customer_id
  AND u.unsubscription_date BETWEEN s.subscription_start_date AND s.subscription_end_date
JOIN Reason_dimension r
  ON u.reason_id = r.reason_id
GROUP BY
  r.reason_category,
  r.reason_description
ORDER BY
  avg_days_to_unsub DESC;
```

133 %

Results | Messages

	reason_category	reason_description	avg_days_to_unsub
1	Inadequate Features	Missing interactive or advanced features	9
2	Streaming Services	Switching to on-demand streaming platforms	9
3	Signal Quality	Poor signal strength and constant disruptions	7
4	Price	Plan is too expensive compared to competitors	6
5	Lack of Use	Plan underutilized due to low viewing frequency	5
6	Service	Poor customer service experience overall	5
7	Customer Loyalty Issues	No rewards or recognition for loyal customers	5
8	Package Complexity	Overly complicated package structure	5
9	Market Saturation	Too many similar services in the market causing ...	5
10	Customer Service	Unresponsive support when issues arise	5
11	Outdated Technology	Service equipment is outdated and inefficient	5
12	Quality of Channels	Channels do not offer expected HD quality	5
13	Unclear Pricing	Confusing pricing structure and hidden costs	5
14	Installation Problems	Delayed or unsatisfactory installation service	5
15	Disruptive Ads	Excessive commercials disrupt viewing experience	5

Query executed successfully. SHIVEN\SQL

Query 16: Feedback–Engagement Anomaly Detection.

```sql
--Feedback-Engagement Anomaly Detection
SELECT
    cd.customer_id,
    cd.customer_name,
    COALESCE(e.total_viewing_minutes,0) AS total_viewing_minutes,
    COALESCE(e.avg_engagement,0)        AS avg_engagement,
    COALESCE(s.avg_sentiment,0)         AS avg_sentiment,
    CASE
      WHEN COALESCE(s.avg_sentiment,0) < 0
           AND COALESCE(e.avg_engagement,0) > (
             SELECT AVG(engagement_score)
             FROM customer_engagement_fact
           )
        THEN 'High engagement but negative sentiment'
      WHEN COALESCE(s.avg_sentiment,0) >= 0
           AND COALESCE(e.avg_engagement,0) < (
             SELECT AVG(engagement_score)
             FROM customer_engagement_fact
           )
        THEN 'Low engagement but positive sentiment'
      ELSE 'Aligned'
    END AS anomaly_flag
```

100 %

Results | Messages

	customer_id	customer_name	total_viewing_minutes	avg_engagement	avg_sentiment	anomaly_flag
1	1	Amit Desai	20	4.500000	0	Aligned
2	2	Priya Shah	21	4.500000	0	Aligned
3	3	Rahul Mehta	22	4.500000	0	Aligned
4	4	Sneha Joshi	23	4.500000	0	Aligned
5	5	Rohan Patel	24	4.500000	0	Aligned
6	6	Neha Kulkarni	25	4.500000	0	Aligned
7	7	Vikram Nair	26	4.500000	0	Aligned
8	8	Shreya Reddy	27	4.500000	0	Aligned
9	9	Suresh Patil	28	4.500000	0	Aligned
10	10	Divya Iyer	29	4.500000	0	Aligned
11	11	Manoj Singh	30	4.200000	0	Aligned
12	12	Anita Verma	31	4.200000	0	Aligned
13	13	Sandeep Kumar	32	4.200000	0	Aligned
14	14	Kavita Gupta	33	4.200000	0	Aligned
15	15	Vijay Sharma	34	4.200000	1	Aligned

Query executed successfully.

Query 17: Customer Loyalty & Revenue Potential Score.

```sql
--Customer Loyalty & Revenue Potential Score
WITH subscription_metrics AS (
    SELECT
        sf.customer_id,
        AVG(DATEDIFF(day, sf.subscription_start_date,
             COALESCE(sf.subscription_end_date, GETDATE()))) AS avg_subscription_duration,
        COUNT(*) AS total_subscriptions,
        AVG(p.plan_price) AS avg_plan_price
    FROM Subscription_fact sf
    JOIN Plan_dimension p
      ON sf.plan_id = p.plan_id
    GROUP BY sf.customer_id
),
engagement_metrics AS (
    SELECT
        customer_id,
        AVG(engagement_score) AS avg_engagement_score,
        SUM(viewing_duration) AS total_view_duration
    FROM customer_engagement_fact
    GROUP BY customer_id
),
max_plan_price AS (
    SELECT MAX(plan_price) AS max_price FROM Plan_dimension
)
```

100 %

Results | Messages

	customer_id	customer_name	avg_subscription_duration	total_subscriptions	avg_plan_price	avg_engagement_score	total_view_duration	loyalty_index	loyalty_segment
1	174	Neeta Desai	60	1	749.000000	3.910000	53	3.12	High Loyalty
2	165	Vikram Singh	45	1	449.000000	4.590000	55	3.09	High Loyalty
3	18	Sunita Chawla	45	1	449.000000	4.200000	37	2.97	High Loyalty
4	71	Rakesh Patel	45	1	599.000000	3.880000	17	2.94	High Loyalty
5	180	Sonal Patel	44	2	724.000000	3.700000	40	2.92	High Loyalty
6	101	Harpreet Kaur	45	1	499.000000	3.940000	28	2.91	High Loyalty
7	10	Divya Iyer	29	1	499.000000	4.500000	29	2.91	High Loyalty
8	47	Arvind Menon	45	1	499.000000	3.900000	66	2.9	High Loyalty
9	134	Sonal Mehta	45	1	499.000000	3.900000	32	2.9	High Loyalty
10	179	Rajesh Desai	29	1	449.000000	4.470000	31	2.88	High Loyalty
11	63	Aakash Patil	29	1	349.000000	4.470000	33	2.84	High Loyalty
12	15	Vijay Sharma	29	1	449.000000	4.200000	34	2.8	High Loyalty
13	11	Manoj Singh	30	1	399.000000	4.200000	30	2.79	High Loyalty

Query executed successfully.

Query 18: Churn-Risk Composite Score.

```sql
--Churn-Risk Composite Score
WITH tenure AS (
    SELECT
        sf.customer_id,
        AVG(DATEDIFF(
            day,
            sf.subscription_start_date,
            COALESCE(sf.subscription_end_date, GETDATE())
        )) AS avg_tenure_days
    FROM Subscription_fact sf
    GROUP BY sf.customer_id
),
unsubs AS (
    SELECT
        customer_id,
        COUNT(*) AS total_unsubs
    FROM unsubscription_fact
    GROUP BY customer_id
),
neg_reason_unsubs AS (
    SELECT
        u.customer_id,
        COUNT(*) AS neg_unsubs_count
    FROM unsubscription_fact u
```

100 %

Results | Messages

	customer_id	customer_name	avg_tenure_days	total_unsubs	neg_unsubs_count	neg_feedback_count	churn_risk_score	risk_segment
1	115	Vijay Nair	37	3	1	1	0.80	High Risk
2	90	Anjali Singh	30	3	1	1	0.80	High Risk
3	197	Arun Kumar	24	3	1	1	0.80	High Risk
4	150	Rita Patel	15	2	2	1	0.75	High Risk
5	45	Karthik Kumar	37	2	2	1	0.75	High Risk
6	36	Anjali Kapoor	15	2	2	1	0.75	High Risk
7	43	Sivakumar Iyer	22	2	1	1	0.55	Medium Risk
8	78	Kiran Desai	30	2	1	1	0.55	Medium Risk
9	81	Rohan Singh	45	1	0	0	NULL	Low Risk
10	85	Suresh Yadav	29	1	1	0	NULL	Low Risk
11	86	Komal Mehta	30	1	0	0	NULL	Low Risk
12	89	Manoj Kumar	22	1	0	0	NULL	Low Risk
13	198	Anita Reddy	22	1	0	0	NULL	Low Risk

Query executed successfully.

Query 19: Ad-Effectiveness vs. Engagement Lift.

```sql
-- Ad-Effectiveness vs. Engagement Lift
WITH ad_engagement AS (
  SELECT
    cef.customer_id,
    ae.ad_type,
    SUM(cef.view_count)       AS tot_views,
    AVG(cef.engagement_score) AS avg_score
  FROM customer_engagement_fact AS cef
  JOIN ad_exposure_dimension AS ae
    ON cef.ad_exposure_id = ae.ad_exposure_id
  GROUP BY cef.customer_id, ae.ad_type
),
no_ad_engagement AS (
  SELECT
    cef.customer_id,
    SUM(cef.view_count)       AS base_views,
    AVG(cef.engagement_score) AS base_score
  FROM customer_engagement_fact AS cef
  LEFT JOIN ad_exposure_dimension AS ae
    ON cef.ad_exposure_id = ae.ad_exposure_id
  WHERE ae.ad_exposure_id IS NULL
  GROUP BY cef.customer_id
)
SELECT
  ae.ad_type,
  ROUND(AVG(ae.avg_score - ISNULL(na.base_score,0)), 2)     AS avg_engagement_lift,
  ROUND(AVG(ae.tot_views  - ISNULL(na.base_views, 0)), 0)    AS avg_view_delta,
  COUNT(DISTINCT ae.customer_id)                             AS customers_exposed
FROM ad_engagement AS ae
LEFT JOIN no_ad_engagement AS na
  ON ae.customer_id = na.customer_id
GROUP BY ae.ad_type
ORDER BY avg_engagement_lift DESC;
```

91 %

Results | Messages

	ad_type	avg_engagement_lift	avg_view_delta	customers_exposed
1	Mid-Roll Overlay	3.500000	137	30
2	Pre-Roll Fullscreen	3.410000	142	10
3	Side-Banner (Left)	3.400000	139	20
4	Teaser Overlay	3.350000	142	20
5	Side-Banner (Right)	3.280000	139	10
6	Post-Roll Fullscreen	3.230000	138	40
7	Banner (Header)	3.200000	137	20

Query executed successfully.

 # Query 20: Promo × Event 3-Month Retention.

```sql
-- Promo × Event 3-Month Retention
;WITH promo_cohort AS (
  SELECT
    s.customer_id,
    pd.promo_type,
    ed.season_name,
    MIN(s.time_id)            AS cohort_time
  FROM subscription_fact AS s
  JOIN promotion_dimension AS pd
    ON s.promo_id = pd.promo_id
  JOIN event_dimension AS ed
    ON s.event_id = ed.event_id
  WHERE s.subscription_status = 'Active'
  GROUP BY s.customer_id, pd.promo_type, ed.season_name
),
retention AS (
  SELECT
    pc.promo_type,
    pc.season_name,
    pc.cohort_time,
    s2.time_id,
    COUNT(DISTINCT s2.customer_id) AS retained_count
  FROM promo_cohort AS pc
  JOIN subscription_fact AS s2
```

110 %

Results | Messages

	promo_type	season_name	cohort_month	activity_month	retained_count	pct_retained
1	Black Friday	Diwali	April	April	1	50.000000000000
2	Black Friday	Diwali	April	May	1	50.000000000000
3	ICC Champions Trophy	Ganesh Chaturthi	May	April	1	100.000000000000
4	Navratri	Christmas	April	April	1	33.300000000000
5	Navratri	Christmas	April	April	1	33.300000000000
6	Navratri	Christmas	April	May	1	33.300000000000
7	Rugby World Cup	Summer	May	April	1	50.000000000000
8	Rugby World Cup	Summer	May	April	1	50.000000000000

Conclusion:

The implementation of a centralized data warehouse has unified subscription lifecycles, unsubscription drivers, content performance, and viewer engagement into a single, robust analytics platform. By consolidating disparate source systems and modeling dimensions such as Customer, Plan, Content, Series, and Time, the organization now gains consistent, multidimensional insights that drive precise decision-making. Detailed retention analyses reveal pricing sensitivities and service-quality bottlenecks, while engagement metrics illuminate which programs keep viewers tuned in. Targeted strategies ranging from loyalty offers and dynamic pricing to personalized content recommendations and cross-sell campaigns are now executed with data-backed confidence. Ultimately, this scalable, warehouse-driven solution empowers the DTH provider to optimize operational efficiency, elevate the customer experience, and achieve stronger, sustainable performance in a competitive media landscape.

Future Scope:

Partition large fact tables (e.g., Subscription_fact, Customer_engagement_fact) by date or month_id and add appropriate indexes on foreign keys and high-cardinality columns. This will significantly speed up queries, especially those rolling up data over long periods. Alternatively, or in addition to real-time DW, establishing Operational Data Store (ODS) could be beneficial. An ODS would hold frequently updated, integrated copies of operational data, suitable for operational reporting and potentially feeding the data warehouse with more timely information

Establishing a comprehensive metadata repository is crucial for the long-term success and usability of the data warehouse. This would document data sources, transformations, data definitions, and business rules, making it easier for users and developers to understand and utilize the data warehouse effectively

Develop a set of standardized, scheduled reports (e.g., weekly churn summary, monthly top-channels) delivered via email or a portal. Simultaneously, roll out self-service BI dashboards with role-based access so business users can explore data without SQL.