# Appendix

## Queries

### Data Cleaning

Mobile\_Game\_Inapp\_Purchases Dataset

# Check nulls per column

SELECT

COUNT(\*) AS total\_rows,

COUNTIF(Customer\_ID IS NULL) AS null\_customer\_id,

COUNTIF(Device IS NULL) AS null\_device,

COUNTIF(GameGenre IS NULL) AS null\_game\_genre,

COUNTIF(SessionCount IS NULL) AS null\_session\_count,

COUNTIF(AverageSessionLength IS NULL) AS null\_average\_session\_length,

COUNTIF(SpendingSegment IS NULL) AS null\_spending\_segment,

COUNTIF(InAppPurchaseAmount IS NULL) AS null\_in\_app\_purchase\_amount,

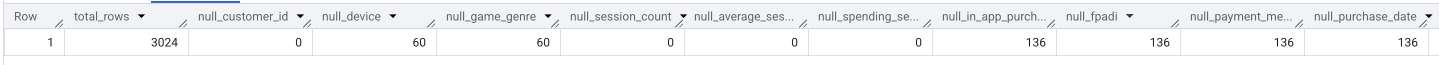
COUNTIF(FirstPurchaseDaysAfterInstall IS NULL) AS null\_fpadi,

COUNTIF(PaymentMethod IS NULL) AS null\_payment\_method,

COUNTIF(LastPurchaseDate IS NULL) AS null\_purchase\_date

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_A3`;

# Initial inspection shows that Device and GameGenre have 60 null values each. InAppPurcahseAmount, FirstPurchaseDaysAfterInstall, PaymentMethod, and LastPurchaseDate have 136 null values each => this makes sense becasue if the customers don't purchase anything then there's no records of these 4 variables.



# Create cleaned table

CREATE OR REPLACE TABLE `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_Cleaned` AS

SELECT

Customer\_ID,

IFNULL(Device, "Unknown") AS Device,

IFNULL(GameGenre, "Unknown") AS GameGenre,

IFNULL(SessionCount, 0) AS SessionCount,

IFNULL(AverageSessionLength, 0.0) AS AverageSessionLength,

IFNULL(SpendingSegment, "Unknown") AS SpendingSegment,

IFNULL(InAppPurchaseAmount, 0.0) AS InAppPurchaseAmount,

IFNULL(FirstPurchaseDaysAfterInstall, 0) AS FirstPurchaseDaysAfterInstall,

IFNULL(PaymentMethod, "Not Applicable") AS PaymentMethod,

LastPurchaseDate,

CASE

WHEN LastPurchaseDate IS NULL THEN "No Purchase"

ELSE "Has Purchased"

END AS PurchaseStatus

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_A3`;

### Donor Behaviour Patterns

Traditional Donor Monetary Contribution

# Monetary Contribution

SELECT

MIN(DonationAmount) AS min\_value,

ROUND(APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(1)],2) AS q1,

ROUND(APPROX\_QUANTILES(DonationAmount, 2)[OFFSET(1)],2) AS median,

ROUND(APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(3)],2) AS q3,

MAX(DonationAmount) AS max\_value,

ROUND(AVG(DonationAmount),2) AS avg\_value,

-- IQR = Q3 - Q1

ROUND(

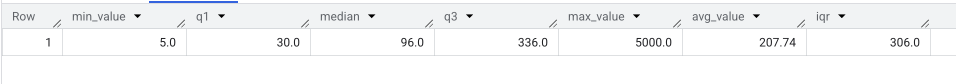
APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(3)]

- APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(1)],2

) AS iqr

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`;

**Results**



# Monetary Contribution by product type

SELECT

Product,

MIN(DonationAmount) AS min\_value,

ROUND(APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(1)],2) AS q1,

ROUND(APPROX\_QUANTILES(DonationAmount, 2)[OFFSET(1)],2) AS median,

ROUND(APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(3)],2) AS q3,

MAX(DonationAmount) AS max\_value,

ROUND(AVG(DonationAmount),2) AS avg\_value,

ROUND(

APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(3)]

- APPROX\_QUANTILES(DonationAmount, 4)[OFFSET(1)],2

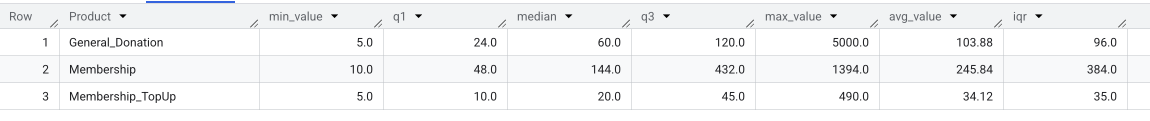
) AS iqr

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Product

ORDER BY Product;

**Results**



# Exploring top 10% Contribution Share

WITH donor\_totals AS (

SELECT

Customer\_ID,

SUM(DonationAmount) AS total\_donated

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Customer\_ID

),

ranked AS (

SELECT

total\_donated,

NTILE(10) OVER (ORDER BY total\_donated DESC) AS decile

FROM donor\_totals

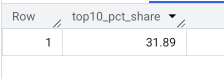
)

SELECT

ROUND(SUM(CASE WHEN decile = 1 THEN total\_donated ELSE 0 END) / SUM(total\_donated) \* 100,2) AS top10\_pct\_share

FROM ranked;

**Results**



In-app Monetary Contribution

# Monetary Contribution

SELECT

MIN(InAppPurchaseAmount) AS min\_value,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(1)],2) AS q1,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 2)[OFFSET(1)],2) AS median,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(3)],2) AS q3,

MAX(InAppPurchaseAmount) AS max\_value,

ROUND(AVG(InAppPurchaseAmount),2) AS avg\_value,

-- IQR = Q3 - Q1

ROUND(

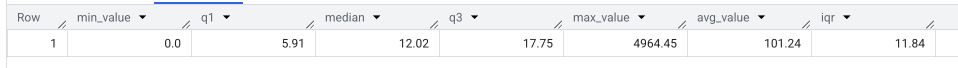
APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(3)]

- APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(1)],2

) AS iqr

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`;

**Results**



# Monetary Contribution by Spending Segment

SELECT

MIN(InAppPurchaseAmount) AS min\_value,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(1)],2) AS q1,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 2)[OFFSET(1)],2) AS median,

ROUND(APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(3)],2) AS q3,

MAX(InAppPurchaseAmount) AS max\_value,

ROUND(AVG(InAppPurchaseAmount),2) AS avg\_value,

-- IQR = Q3 - Q1

ROUND(

APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(3)]

- APPROX\_QUANTILES(InAppPurchaseAmount, 4)[OFFSET(1)],2

) AS iqr

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`;

**Results**



# Exploring top 10% Contribution Share

WITH ranked AS (

SELECT

InAppPurchaseAmount AS total\_donated,

NTILE(10) OVER (ORDER BY InAppPurchaseAmount DESC) AS decile

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`

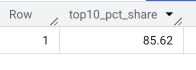
)

SELECT

ROUND(SUM(CASE WHEN decile = 1 THEN total\_donated ELSE 0 END) / SUM(total\_donated) \* 100,2) AS top10\_pct\_share

FROM ranked;

**Results**



RFM Segmemtation

Traditional Donor Payment Methods

SELECT

Channel\_Pay,

COUNT(\*) AS frequency,

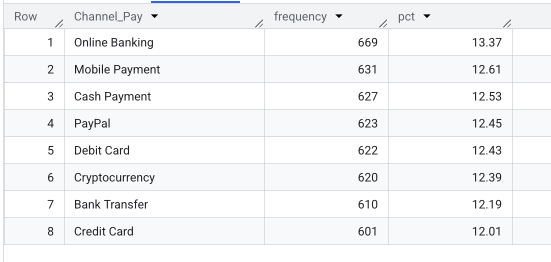
ROUND(COUNT(\*) \* 100.0 / SUM(COUNT(\*)) OVER(),2) AS pct

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Channel\_Pay

ORDER BY frequency DESC;

**Results**



In-app Payment Methods

SELECT

PaymentMethod,

COUNT(\*) AS frequency,

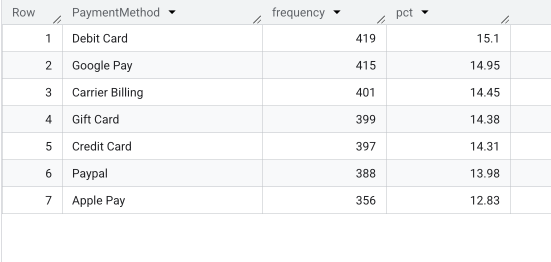
ROUND(COUNT(\*) \* 100.0 / SUM(COUNT(\*)) OVER(),2) AS pct

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`

GROUP BY PaymentMethod

ORDER BY frequency DESC;

**Results**



Traditional Donor Frequency

# Overall Frequency

WITH donor\_freq AS (

SELECT

Customer\_ID,

COUNT(Donation\_ID) AS donation\_count

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Customer\_ID

)

SELECT

COUNT(\*) AS total\_donors,

ROUND(AVG(donation\_count),2) AS avg\_donations,

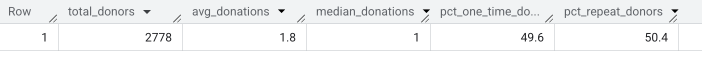
APPROX\_QUANTILES(donation\_count, 2)[OFFSET(1)] AS median\_donations,

ROUND(SUM(CASE WHEN donation\_count=1 THEN 1 ELSE 0 END)\*100.0/COUNT(\*),2) AS pct\_one\_time\_donors,

ROUND(SUM(CASE WHEN donation\_count>1 THEN 1 ELSE 0 END)\*100.0/COUNT(\*),2) AS pct\_repeat\_donors

FROM donor\_freq;

**Results**



# Frequency segment

WITH donor\_freq AS (

SELECT

Customer\_ID,

COUNT(Donation\_ID) AS donation\_count

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Customer\_ID

),

freq\_with\_customer AS (

SELECT

c.Customer\_ID,

c.Age,

c.Gender,

c.Family\_Size,

f.donation\_count

FROM donor\_freq f

JOIN `mis784t22025-466123.MIS784\_A3.customer\_A3` c

ON f.Customer\_ID = c.Customer\_ID

),

freq\_with\_agegroup AS (

SELECT \*,

CASE

WHEN Age < 30 THEN 'Under 30'

WHEN Age BETWEEN 30 AND 44 THEN '30–44'

WHEN Age BETWEEN 45 AND 59 THEN '45–59'

ELSE '60+'

END AS Age\_Group

FROM freq\_with\_customer

)

SELECT

Age\_Group,

CASE

WHEN donation\_count = 1 THEN 'One-time'

WHEN donation\_count BETWEEN 2 AND 4 THEN 'Occasional (2–4)'

ELSE 'Frequent (5+)'

END AS frequency\_segment,

COUNT(\*) AS donor\_count,

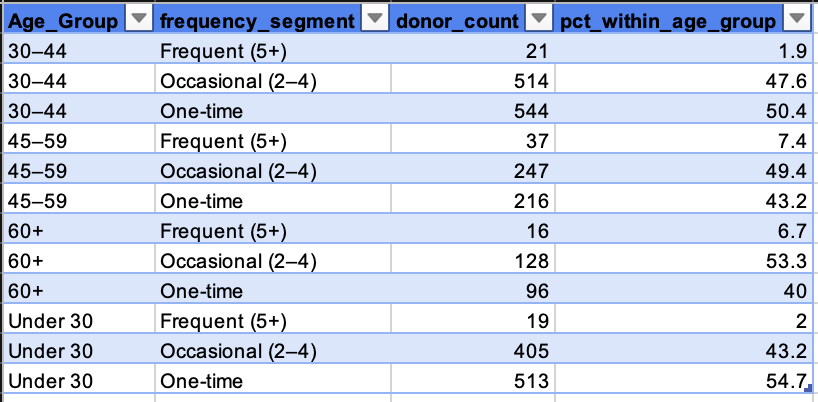
ROUND(100 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY Age\_Group), 1) AS pct\_within\_age\_group

FROM freq\_with\_agegroup

GROUP BY Age\_Group, frequency\_segment

ORDER BY Age\_Group, frequency\_segment;

**Results**



Traditional Donor Recency

# Overall recency

WITH donor\_recency AS (

SELECT

Customer\_ID,

MAX(DonationDate) AS last\_donation

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Customer\_ID

),

max\_date AS (

SELECT MAX(DonationDate) AS dataset\_max\_date

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

)

SELECT

ROUND(AVG(DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY)),1) AS avg\_days\_since,

APPROX\_QUANTILES(DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY), 2)[OFFSET(1)] AS median\_days\_since,

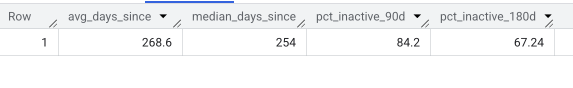
ROUND(100\*AVG(CAST(DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY) > 90 AS INT64)),2) AS pct\_inactive\_90d,

ROUND(100\*AVG(CAST(DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY) > 180 AS INT64)),2) AS pct\_inactive\_180d

FROM donor\_recency r

CROSS JOIN max\_date m;

**Results**



# Recency segments

WITH donor\_recency AS (

SELECT

Customer\_ID,

MAX(DonationDate) AS last\_donation

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

GROUP BY Customer\_ID

),

max\_date AS (

SELECT MAX(DonationDate) AS dataset\_max\_date

FROM `mis784t22025-466123.MIS784\_A3.Tradition\_Donation\_Patterns\_A3`

),

recency\_segmented AS (

SELECT

c.Customer\_ID,

c.Age,

c.Family\_Size,

c.Gender,

r.last\_donation,

DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY) AS days\_since\_last,

CASE

WHEN DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY) <= 90 THEN 'Active (≤90d)'

WHEN DATE\_DIFF(m.dataset\_max\_date, r.last\_donation, DAY) <= 180 THEN 'At Risk (91–180d)'

ELSE 'Lapsed (>180d)'

END AS recency\_segment

FROM donor\_recency r

JOIN `mis784t22025-466123.MIS784\_A3.customer\_A3` c

ON r.Customer\_ID = c.Customer\_ID

CROSS JOIN max\_date m

),

recency\_with\_agegroup AS (

SELECT \*,

CASE

WHEN Age < 30 THEN 'Under 30'

WHEN Age BETWEEN 30 AND 44 THEN '30–44'

WHEN Age BETWEEN 45 AND 59 THEN '45–59'

ELSE '60+'

END AS Age\_Group

FROM recency\_segmented

)

SELECT

Age\_Group,

recency\_segment,

COUNT(\*) AS donor\_count,

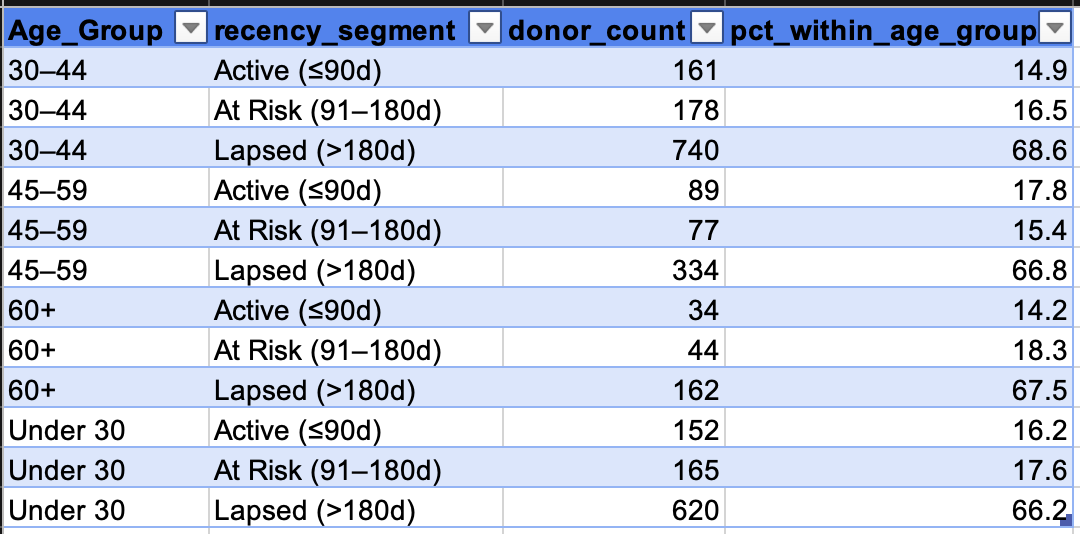
ROUND(100 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY Age\_Group), 1) AS pct\_within\_age\_group

FROM recency\_with\_agegroup

GROUP BY Age\_Group, recency\_segment

ORDER BY Age\_Group, recency\_segment;

**Results**



In-app Engagement

# Engagement summary

SELECT

ROUND(AVG(SessionCount),1) AS avg\_sessions,

APPROX\_QUANTILES(SessionCount, 2)[OFFSET(1)] AS median\_sessions,

ROUND(AVG(AverageSessionLength),1) AS avg\_session\_length,

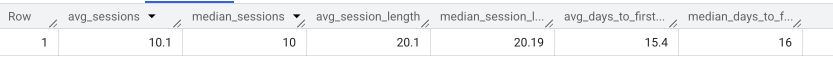
APPROX\_QUANTILES(AverageSessionLength, 2)[OFFSET(1)] AS median\_session\_length,

ROUND(AVG(FirstPurchaseDaysAfterInstall),1) AS avg\_days\_to\_first\_purchase,

APPROX\_QUANTILES(FirstPurchaseDaysAfterInstall, 2)[OFFSET(1)] AS median\_days\_to\_first\_purchase

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`;

**Results**



# Spending Segment Engagement Summary

SELECT

SpendingSegment,

ROUND(AVG(SessionCount),1) AS avg\_sessions,

ROUND(AVG(AverageSessionLength),1) AS avg\_session\_length,

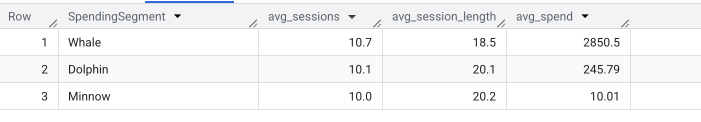
ROUND(AVG(InAppPurchaseAmount),2) AS avg\_spend

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`

GROUP BY SpendingSegment

ORDER BY avg\_spend DESC;

**Results**



# Game Genre choice per Spending Segment

SELECT

SpendingSegment,

GameGenre,

COUNT(\*) AS donor\_count

FROM `mis784t22025-466123.MIS784\_A3.Mobile\_Game\_Inapp\_Purchases\_No\_Nulls`

GROUP BY SpendingSegment, GameGenre

ORDER BY SpendingSegment, donor\_count DESC;

**Results**



### Churn Risk & Retention

**1.** **Data Check for Nulls:**

**Query:**

SELECT

COUNTIF(Customer\_ID IS NULL) AS null\_customer,

COUNTIF(Donation\_ID IS NULL) AS null\_donation,

COUNTIF(DonationDate IS NULL) AS null\_donationdate,

COUNTIF(DonationEndDate IS NULL) AS null\_donationenddate,

COUNTIF(DonationAmount IS NULL) AS null\_amount,

COUNTIF(Product IS NULL) AS null\_product,

COUNTIF(Channel\_Pay IS NULL) AS null\_channel

FROM `mis784-sem1.Assignment\_3.Tradition\_Donation\_Patterns`;

**Result:**



**Query:**

SELECT

COUNTIF(Customer\_ID IS NULL) AS null\_customer,

COUNTIF(LastPurchaseDate IS NULL) AS null\_lastpurchase,

COUNTIF(SessionCount IS NULL) AS null\_sessioncount,

COUNTIF(AverageSessionLength IS NULL) AS null\_avgsessionlength,

COUNTIF(SpendingSegment IS NULL) AS null\_segment,

COUNTIF(PaymentMethod IS NULL) AS null\_payment,

COUNTIF(InAppPurchaseAmount IS NULL) AS null\_inapppurchase

FROM `mis784-sem1.Assignment\_3.Mobile\_Game\_Inapp\_Purchases`;

**Result:**



**Query:**

SELECT

COUNTIF(Customer\_ID IS NULL) AS null\_customer,

COUNTIF(Campaign\_ID IS NULL) AS null\_campaign,

COUNTIF(Response IS NULL) AS null\_response,

COUNTIF(ClickThroughRate IS NULL) AS null\_ctr,

COUNTIF(EngagementFrequency IS NULL) AS null\_engagement

FROM `mis784-sem1.Assignment\_3.Campaign\_Response`;

**Result:**



**Query**:

SELECT

COUNTIF(Customer\_ID IS NULL) AS null\_customer,

COUNTIF(Age IS NULL) AS null\_age,

COUNTIF(Gender IS NULL) AS null\_gender,

COUNTIF(Occupation IS NULL) AS null\_occupation,

COUNTIF(Income\_Level IS NULL) AS null\_incomelevel,

COUNTIF(Location IS NULL) AS null\_location,

COUNTIF(City IS NULL) AS null\_city,

COUNTIF(Family\_Size IS NULL) AS null\_familysize

FROM `mis784-sem1.Assignment\_3.Customers`;

**Result**:



**2.** **Duplicates check:**

**Query:**

SELECT COUNT(\*) AS total\_rows,

COUNT(DISTINCT Donation\_ID) AS unique\_donation\_ids

FROM `mis784-sem1.Assignment\_3.Tradition\_Donation\_Patterns`;

**Result:**



**NOTE:** Says there are 5003 unique donations made.

**Query:**

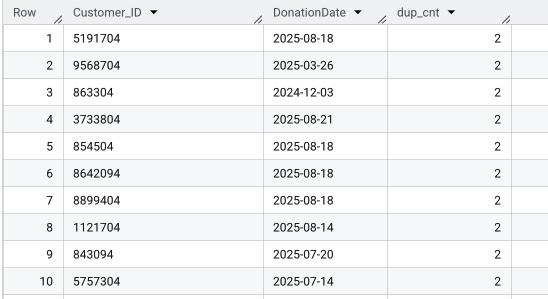
SELECT Customer\_ID, DonationDate, COUNT(\*) AS dup\_cnt

FROM `mis784-sem1.Assignment\_3.Tradition\_Donation\_Patterns`

GROUP BY Customer\_ID, DonationDate

HAVING COUNT(\*) > 1;

**Result:**



**NOTE:** However, checking as per Customer\_ID and DonationDate, some donors are appearing two times (dup\_cnt=2), which means they donated multiple times on the same date. These duplicates were not removed in order to maintain accurate donation patterns, because they represent actual separate donations, not data errors.

**Query:**

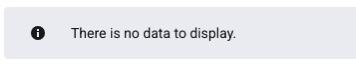
SELECT Customer\_ID, LastPurchaseDate, COUNT(\*) AS dup\_cnt

FROM `mis784-sem1.Assignment\_3.Mobile\_Game\_Inapp\_Purchases`

GROUP BY Customer\_ID, LastPurchaseDate

HAVING COUNT(\*) > 1;

**Result:**



**NOTE:** Hence no duplicate data.

**Query:**

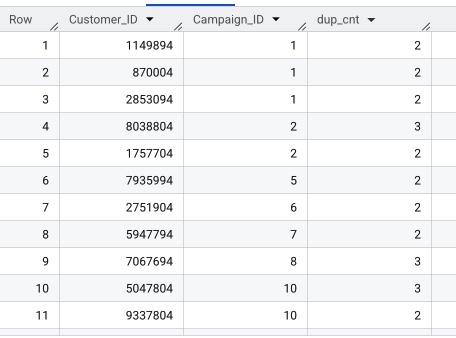
SELECT Customer\_ID, Campaign\_ID, COUNT(\*) AS dup\_cnt

FROM `mis784-sem1.Assignment\_3.Campaign\_Response`

GROUP BY Customer\_ID, Campaign\_ID

HAVING COUNT(\*) > 1;

**Result:**



**3.** **Outliers Check:**

**Query:**

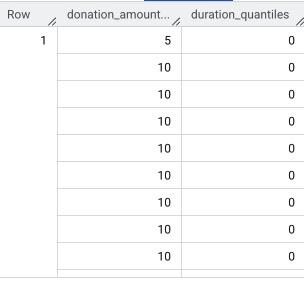
SELECT

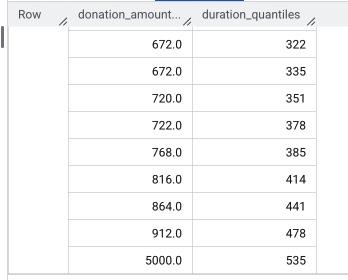
APPROX\_QUANTILES(DonationAmount, 100) AS donation\_amount\_quantiles,

APPROX\_QUANTILES(DATE\_DIFF(DonationEndDate, DonationDate, DAY), 100) AS duration\_quantiles

FROM `mis784-sem1.Assignment\_3.Tradition\_Donation\_Patterns`;

**Result:**





**Query:**

SELECT

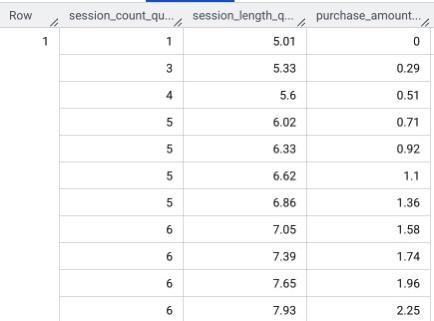
APPROX\_QUANTILES(SessionCount, 100) AS session\_count\_quantiles,

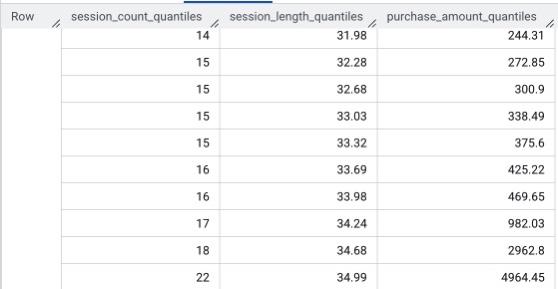
APPROX\_QUANTILES(AverageSessionLength, 100) AS session\_length\_quantiles,

APPROX\_QUANTILES(InAppPurchaseAmount, 100) AS purchase\_amount\_quantiles

FROM `mis784-sem1.Assignment\_3.Mobile\_Game\_Inapp\_Purchases`;

**Result:**





**Query:**

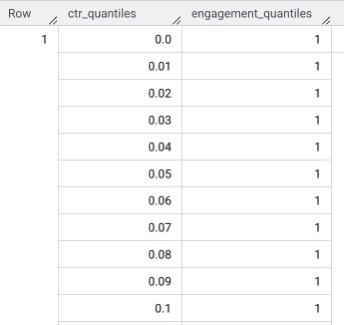
SELECT

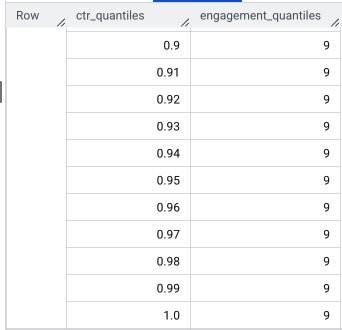
APPROX\_QUANTILES(ClickThroughRate, 100) AS ctr\_quantiles,

APPROX\_QUANTILES(EngagementFrequency, 100) AS engagement\_quantiles

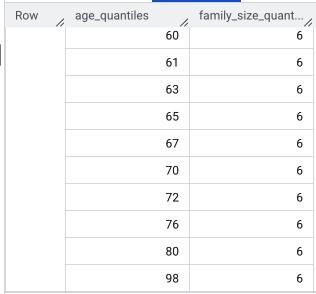
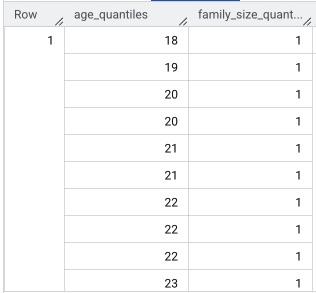
FROM `mis784-sem1.Assignment\_3.Campaign\_Response`;

**Result:**

****

****

**Result**:



**4.** **Unified RFM Analysis (Create Views)**

**5. RFM Segment Counts**

**Query:**

SELECT donation\_rfm\_segment AS segment, COUNT(\*) AS donor\_count

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment`

GROUP BY segment

ORDER BY donor\_count DESC;

**Result:**

****

**Query:**

SELECT inapp\_rfm\_segment AS segment, COUNT(\*) AS donor\_count

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment`

GROUP BY segment

ORDER BY donor\_count DESC;

**Result:**

****

**6.** **Campaign Response by Churn Segment (Traditional & In-App):**

**a)** **Campaign Response Rate by Churn Segment – Traditional Donors**

**Query:**

WITH seg AS (

SELECT CAST(Customer\_ID AS STRING) AS customer\_id\_s,

donation\_rfm\_segment AS churn\_segment

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment`

),

camp AS (

SELECT

CAST(Customer\_ID AS STRING) AS customer\_id\_s,

CASE

WHEN SAFE\_CAST(Response AS BOOL) IS NOT NULL THEN SAFE\_CAST(Response AS BOOL)

WHEN LOWER(CAST(Response AS STRING)) IN ('yes','y','true','1') THEN TRUE

WHEN LOWER(CAST(Response AS STRING)) IN ('no','n','false','0') THEN FALSE

ELSE FALSE

END AS responded\_flag

FROM `mis784-sem1.Assignment\_3.Campaign\_Response`

),

customer\_level AS (

SELECT s.churn\_segment,

c.customer\_id\_s,

MAX(c.responded\_flag) AS any\_response

FROM seg s

JOIN camp c USING (customer\_id\_s)

GROUP BY s.churn\_segment, c.customer\_id\_s

)

SELECT

churn\_segment,

COUNT(\*) AS total\_customers,

SUM(CASE WHEN any\_response THEN 1 ELSE 0 END) AS responders,

ROUND(100.0 \* SUM(CASE WHEN any\_response THEN 1 ELSE 0 END) / COUNT(\*), 2) AS response\_rate\_pct

FROM customer\_level

GROUP BY churn\_segment

ORDER BY response\_rate\_pct DESC;

**Result:**

****

**b)** **Campaign Response Rate by Churn Segment – Mobile In-App**

**Query:**

WITH seg AS (

SELECT CAST(Customer\_ID AS STRING) AS customer\_id\_s,

inapp\_rfm\_segment AS churn\_segment

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment`

),

camp AS (

SELECT

CAST(Customer\_ID AS STRING) AS customer\_id\_s,

CASE

WHEN SAFE\_CAST(Response AS BOOL) IS NOT NULL THEN SAFE\_CAST(Response AS BOOL)

WHEN LOWER(CAST(Response AS STRING)) IN ('yes','y','true','1') THEN TRUE

WHEN LOWER(CAST(Response AS STRING)) IN ('no','n','false','0') THEN FALSE

ELSE FALSE

END AS responded\_flag

FROM `mis784-sem1.Assignment\_3.Campaign\_Response`

),

customer\_level AS (

SELECT s.churn\_segment, c.customer\_id\_s, MAX(c.responded\_flag) AS any\_response

FROM seg s

JOIN camp c USING (customer\_id\_s)

GROUP BY s.churn\_segment, c.customer\_id\_s

)

SELECT

churn\_segment,

COUNT(\*) AS total\_customers,

SUM(CASE WHEN any\_response THEN 1 ELSE 0 END) AS responders,

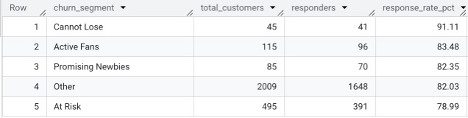
ROUND(100.0 \* SUM(CASE WHEN any\_response THEN 1 ELSE 0 END) / COUNT(\*), 2) AS response\_rate\_pct

FROM customer\_level

GROUP BY churn\_segment

ORDER BY response\_rate\_pct DESC;

**Result:**

****

**c)** **Click-Through & Engagement by Churn Segment – Traditional Donors**

**Query:**

WITH customer\_avg AS (

SELECT

ds.donation\_rfm\_segment AS churn\_segment,

cr.Customer\_ID,

AVG(SAFE\_CAST(cr.ClickThroughRate AS FLOAT64)) AS customer\_avg\_ctr,

AVG(SAFE\_CAST(cr.EngagementFrequency AS FLOAT64)) AS customer\_avg\_engagement

FROM `mis784-sem1.Assignment\_3.Campaign\_Response` cr

JOIN `mis784-sem1.Assignment\_3.v\_donation\_segment` ds

ON CAST(cr.Customer\_ID AS STRING) = CAST(ds.Customer\_ID AS STRING)

WHERE cr.ClickThroughRate IS NOT NULL

AND cr.EngagementFrequency IS NOT NULL

GROUP BY churn\_segment, cr.Customer\_ID

)

SELECT

churn\_segment,

ROUND(AVG(customer\_avg\_ctr) \* 100.0, 2) AS ctr\_pct\_customer\_avg,

ROUND(AVG(customer\_avg\_engagement), 2) AS eng\_freq\_customer\_avg,

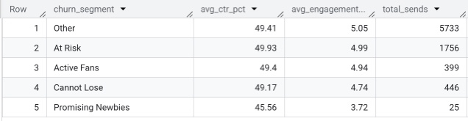
COUNT(DISTINCT Customer\_ID) AS customers\_count

FROM customer\_avg

GROUP BY churn\_segment

ORDER BY eng\_freq\_customer\_avg DESC;

**Result:**

****

**d)** **Click-Through & Engagement by Churn Segment – In-App**

**Query:**

WITH customer\_avg AS (

SELECT

ds.inapp\_rfm\_segment AS churn\_segment,

cr.Customer\_ID,

AVG(SAFE\_CAST(cr.ClickThroughRate AS FLOAT64)) AS customer\_avg\_ctr,

AVG(SAFE\_CAST(cr.EngagementFrequency AS FLOAT64)) AS customer\_avg\_engagement

FROM `mis784-sem1.Assignment\_3.Campaign\_Response` cr

JOIN `mis784-sem1.Assignment\_3.v\_inapp\_segment` ds

ON CAST(cr.Customer\_ID AS STRING) = CAST(ds.Customer\_ID AS STRING)

WHERE cr.ClickThroughRate IS NOT NULL

AND cr.EngagementFrequency IS NOT NULL

GROUP BY churn\_segment, cr.Customer\_ID

)

SELECT

churn\_segment,

ROUND(AVG(customer\_avg\_ctr) \* 100.0, 2) AS ctr\_pct\_customer\_avg,

ROUND(AVG(customer\_avg\_engagement), 2) AS eng\_freq\_customer\_avg,

COUNT(DISTINCT Customer\_ID) AS customers\_count

FROM customer\_avg

GROUP BY churn\_segment

ORDER BY eng\_freq\_customer\_avg DESC;

**Result:**

****

**7.** **Behavioral Predictors Attrition:**

**a)** **Outliers (Traditional Donors)**

**Query (Bottom Donors- Lowest Totals):**

SELECT

donation\_total\_amount,

COUNT(\*) AS donor\_count

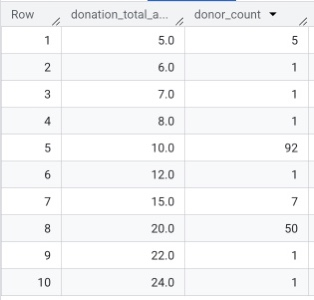
FROM `mis784-sem1.Assignment\_3.v\_donation\_agg`

GROUP BY donation\_total\_amount

ORDER BY donation\_total\_amount ASC

LIMIT 10;

**Result:**

****

**Query (Top Donors- Highest Totals):**

SELECT

Customer\_ID,

donation\_total\_amount,

donation\_frequency,

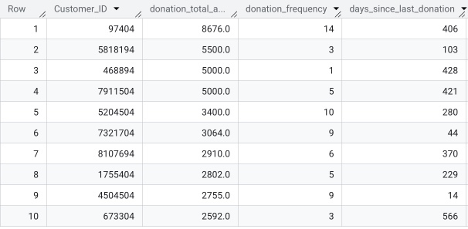
days\_since\_last\_donation

FROM `mis784-sem1.Assignment\_3.v\_donation\_agg`

ORDER BY donation\_total\_amount DESC

LIMIT 10;

**Result:**

****

**b)** **Outliers (In-app)**

**Query (Bottom Donors- Lowest Totals):**

SELECT

purchase\_total\_amount,

COUNT(\*) AS customer\_count

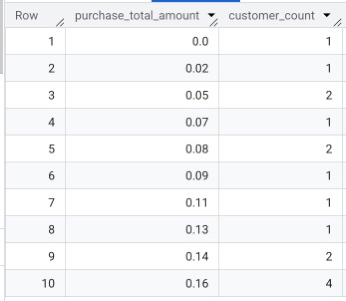
FROM `mis784-sem1.Assignment\_3.v\_inapp\_agg`

GROUP BY purchase\_total\_amount

ORDER BY purchase\_total\_amount ASC

LIMIT 10;

**Result:**

****

**Query (Top Donors- Highest Totals):**

SELECT

Customer\_ID,

purchase\_total\_amount,

purchase\_frequency,

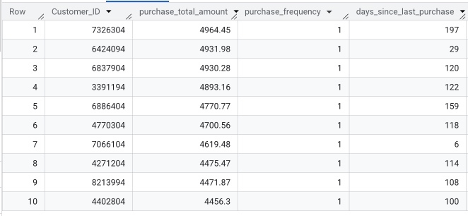
days\_since\_last\_purchase

FROM `mis784-sem1.Assignment\_3.v\_inapp\_agg`

ORDER BY purchase\_total\_amount DESC

LIMIT 10;

**Result:**

****

**c)** **RFM Drivers (Traditional Donors)**

**Query:**

SELECT

s.donation\_rfm\_segment AS churn\_segment,

ROUND(AVG(a.days\_since\_last\_donation), 1) AS avg\_days\_since\_last,

ROUND(AVG(a.donation\_frequency), 2) AS avg\_frequency,

ROUND(AVG(a.donation\_total\_amount), 2) AS avg\_total\_donations,

COUNT(\*) AS donors\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_donation\_agg` a

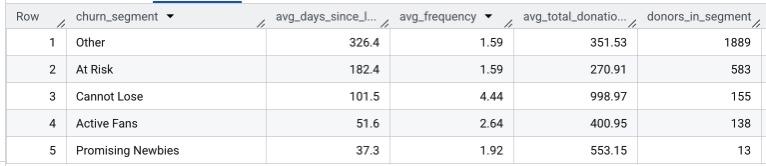
JOIN `mis784-sem1.Assignment\_3.v\_donation\_segment` s

ON a.Customer\_ID = s.Customer\_ID

GROUP BY churn\_segment

ORDER BY avg\_days\_since\_last DESC;

**Result:**

****

**c)** **RFM Drivers (In-App)**

**Query:**

SELECT

s.inapp\_rfm\_segment AS churn\_segment,

ROUND(AVG(a.days\_since\_last\_purchase), 1) AS avg\_days\_since\_last,

ROUND(AVG(a.purchase\_frequency), 2) AS avg\_frequency,

ROUND(AVG(a.purchase\_total\_amount), 2) AS avg\_total\_purchases,

COUNT(\*) AS customers\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_inapp\_agg` a

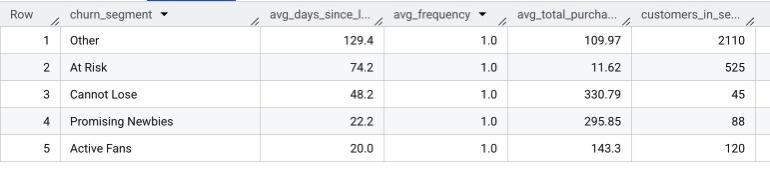
JOIN `mis784-sem1.Assignment\_3.v\_inapp\_segment` s

ON a.Customer\_ID = s.Customer\_ID

GROUP BY churn\_segment

ORDER BY avg\_days\_since\_last DESC;

**Result:**



**d)** **Revenue Concentration by Top 10% vs Rest**

**Query:**

WITH trad\_totals AS (

SELECT Customer\_ID, SUM(DonationAmount) AS total\_amt

FROM `mis784-sem1.Assignment\_3.Tradition\_Donation\_Patterns`

WHERE DonationAmount IS NOT NULL

GROUP BY Customer\_ID

),

trad\_ranked AS (

SELECT

\*,

NTILE(10) OVER (ORDER BY total\_amt DESC) AS decile

FROM trad\_totals

),

inapp\_totals AS (

SELECT Customer\_ID, SUM(IFNULL(InAppPurchaseAmount,0)) AS total\_amt

FROM `mis784-sem1.Assignment\_3.Mobile\_Game\_Inapp\_Purchases`

GROUP BY Customer\_ID

),

inapp\_ranked AS (

SELECT

\*,

NTILE(10) OVER (ORDER BY total\_amt DESC) AS decile

FROM inapp\_totals

)

-- Traditional

SELECT

'Traditional' AS channel,

ROUND(SUM(CASE WHEN decile = 1 THEN total\_amt ELSE 0 END) / SUM(total\_amt) \* 100, 2) AS top10\_share\_pct,

ROUND(SUM(CASE WHEN decile <= 2 THEN total\_amt ELSE 0 END) / SUM(total\_amt) \* 100, 2) AS top20\_share\_pct,

COUNT(\*) AS donors\_count

FROM trad\_ranked

UNION ALL

-- In-App

SELECT

'In-App' AS channel,

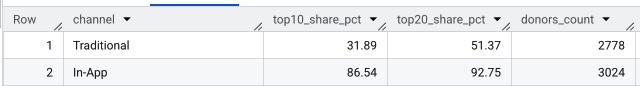
ROUND(SUM(CASE WHEN decile = 1 THEN total\_amt ELSE 0 END) / SUM(total\_amt) \* 100, 2) AS top10\_share\_pct,

ROUND(SUM(CASE WHEN decile <= 2 THEN total\_amt ELSE 0 END) / SUM(total\_amt) \* 100, 2) AS top20\_share\_pct,

COUNT(\*) AS customers\_count

FROM inapp\_ranked;

**Result:**



**8. Demographics Impact on Retention:**

1. **Age Distribution Across Churn Groups - Traditional Donors**

**Query:**

SELECT

ds.donation\_rfm\_segment AS churn\_segment,

ROUND(AVG(c.Age), 1) AS avg\_age,

MIN(c.Age) AS min\_age,

MAX(c.Age) AS max\_age,

COUNT(\*) AS donors\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment` ds

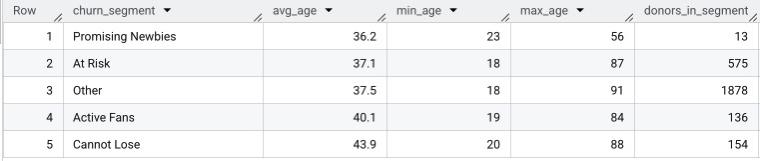
JOIN `mis784-sem1.Assignment\_3.Customers` c

ON ds.Customer\_ID = c.Customer\_ID

GROUP BY churn\_segment

ORDER BY avg\_age;

**Result:**

****

1. **Age Distribution Across Churn Groups - In-App**

**Query:**

SELECT

iseg.inapp\_rfm\_segment AS churn\_segment,

ROUND(AVG(c.Age), 1) AS avg\_age,

MIN(c.Age) AS min\_age,

MAX(c.Age) AS max\_age,

COUNT(\*) AS customers\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment` iseg

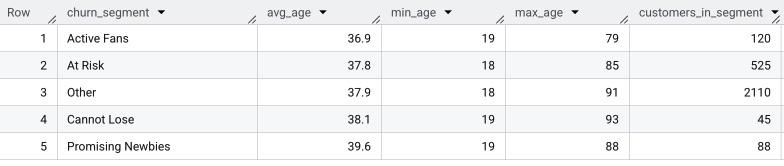
JOIN `mis784-sem1.Assignment\_3.Customers` c

ON iseg.Customer\_ID = c.Customer\_ID

GROUP BY churn\_segment

ORDER BY avg\_age;

**Result:**

****

* + 1. **Gender vs Churn Segment - Traditional Donors**

**Query:**

SELECT

ds.donation\_rfm\_segment AS churn\_segment,

c.Gender,

COUNT(\*) AS donor\_count

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment` ds

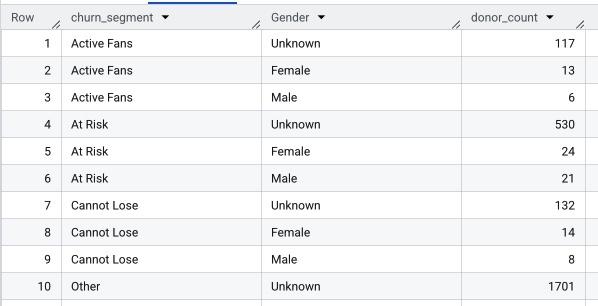
JOIN `mis784-sem1.Assignment\_3.Customers` c

ON ds.Customer\_ID = c.Customer\_ID

GROUP BY churn\_segment, c.Gender

ORDER BY churn\_segment, donor\_count DESC;

**Result:**



* + 1. **Gender vs Churn Segment - In-App**

**Query:**

SELECT

iseg.inapp\_rfm\_segment AS churn\_segment,

c.Gender,

COUNT(\*) AS customer\_count

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment` iseg

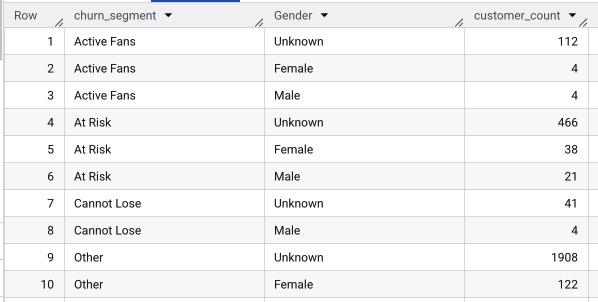
JOIN `mis784-sem1.Assignment\_3.Customers` c

ON iseg.Customer\_ID = c.Customer\_ID

GROUP BY churn\_segment, c.Gender

ORDER BY churn\_segment, customer\_count DESC;

**Result:**



* + 1. **Income vs Churn Segment - Traditional Donors**

**Query:**

SELECT

ds.donation\_rfm\_segment AS churn\_segment,

c.Income\_Level,

COUNT(\*) AS donor\_count,

ROUND(100.0 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY ds.donation\_rfm\_segment), 2) AS pct\_within\_segment

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment` ds

JOIN `mis784-sem1.Assignment\_3.Customers` c

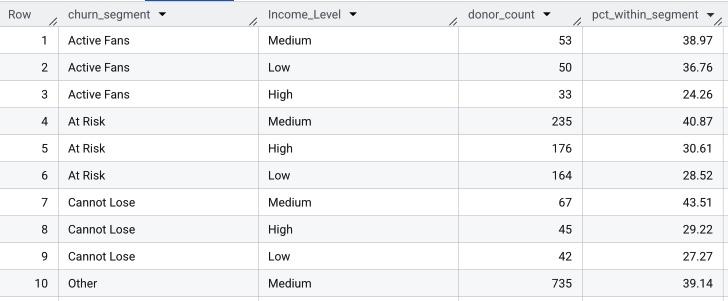
ON ds.Customer\_ID = c.Customer\_ID

WHERE c.Income\_Level IS NOT NULL

GROUP BY churn\_segment, c.Income\_Level

ORDER BY churn\_segment, pct\_within\_segment DESC;

**Result:**



* + 1. **Income vs Churn Segment - In-App**

**Query:**

SELECT

iseg.inapp\_rfm\_segment AS churn\_segment,

c.Income\_Level,

COUNT(\*) AS customer\_count,

ROUND(100.0 \* COUNT(\*) / SUM(COUNT(\*)) OVER (PARTITION BY iseg.inapp\_rfm\_segment), 2) AS pct\_within\_segment

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment` iseg

JOIN `mis784-sem1.Assignment\_3.Customers` c

ON iseg.Customer\_ID = c.Customer\_ID

WHERE c.Income\_Level IS NOT NULL

GROUP BY churn\_segment, c.Income\_Level

ORDER BY churn\_segment, pct\_within\_segment DESC;

**Result:**



* + 1. **Family Size vs Churn Segment – Traditional Donor**

**Query:**

SELECT

dseg.donation\_rfm\_segment AS churn\_segment,

ROUND(AVG(SAFE\_CAST(c.Family\_Size AS FLOAT64)), 2) AS avg\_family\_size,

COUNT(\*) AS donors\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_donation\_segment` dseg

JOIN `mis784-sem1.Assignment\_3.Customers` c

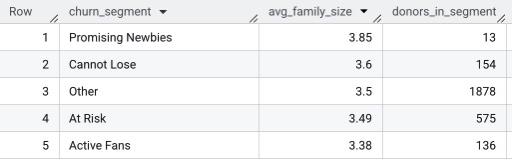
ON dseg.Customer\_ID = c.Customer\_ID

WHERE SAFE\_CAST(c.Family\_Size AS FLOAT64) IS NOT NULL

GROUP BY churn\_segment

ORDER BY avg\_family\_size DESC;

**Result:**



* + 1. **Family Size vs Churn Segment - In-App**

**Query:**

SELECT

iseg.inapp\_rfm\_segment AS churn\_segment,

ROUND(AVG(SAFE\_CAST(c.Family\_Size AS FLOAT64)), 2) AS avg\_family\_size,

COUNT(\*) AS customers\_in\_segment

FROM `mis784-sem1.Assignment\_3.v\_inapp\_segment` iseg

JOIN `mis784-sem1.Assignment\_3.Customers` c

ON iseg.Customer\_ID = c.Customer\_ID

WHERE SAFE\_CAST(c.Family\_Size AS FLOAT64) IS NOT NULL

GROUP BY churn\_segment

ORDER BY avg\_family\_size DESC;

**Result:**



### Revenue Optimisation

-- reusable views

-- donations – aggregates and rfm

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_donation\_agg` AS

WITH asof AS (

SELECT MAX(DonationDate) AS as\_of

FROM `mis784-466303.A3.tradition\_donation\_patterns`

WHERE DonationDate IS NOT NULL)

SELECT

t.Customer\_ID,

MAX(t.DonationDate) AS last\_donation\_date,

COUNT(\*) AS donation\_frequency,

SUM(t.DonationAmount) AS donation\_total\_amount,

DATE\_DIFF((SELECT as\_of FROM asof), MAX(t.DonationDate), DAY) AS days\_since\_last\_donation

FROM `mis784-466303.A3.tradition\_donation\_patterns` t

WHERE t.DonationDate IS NOT NULL AND t.DonationAmount IS NOT NULL

GROUP BY t.Customer\_ID;

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_donation\_rfm` AS

SELECT

a.\*,

NTILE(5) OVER (ORDER BY a.days\_since\_last\_donation DESC, a.Customer\_ID) AS R\_score,

NTILE(5) OVER (ORDER BY a.donation\_frequency ASC, a.Customer\_ID) AS F\_score,

NTILE(5) OVER (ORDER BY a.donation\_total\_amount ASC, a.Customer\_ID) AS M\_score

FROM `mis784-466303.A3.v\_donation\_agg` a;

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_donation\_segment` AS

SELECT

Customer\_ID,

CASE

WHEN R\_score IN (4,5) AND F\_score = 5 AND M\_score = 5 THEN 'Cannot Lose'

WHEN R\_score = 5 AND F\_score IN (4,5) AND M\_score IN (3,4,5) THEN 'Active Fans'

WHEN R\_score = 5 AND F\_score IN (2,3) AND M\_score IN (4,5) THEN 'Promising Newbies'

WHEN R\_score IN (3,4,5) AND F\_score IN (2,3,4) AND M\_score IN (2,3,4) THEN 'At Risk'

ELSE 'Other'

END AS donation\_rfm\_segment

FROM `mis784-466303.A3.v\_donation\_rfm`;

-- in-app – aggregates and rfm

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_inapp\_agg` AS

WITH asof AS (

SELECT MAX(LastPurchaseDate) AS as\_of

FROM `mis784-466303.A3.mobile\_game\_inapp\_purchases`

WHERE LastPurchaseDate IS NOT NULL)

SELECT

m.Customer\_ID,

MAX(m.LastPurchaseDate) AS last\_purchase\_date,

COUNTIF(m.InAppPurchaseAmount IS NOT NULL) AS purchase\_frequency,

SUM(IFNULL(m.InAppPurchaseAmount,0)) AS purchase\_total\_amount,

DATE\_DIFF((SELECT as\_of FROM asof), MAX(m.LastPurchaseDate), DAY) AS days\_since\_last\_purchase

FROM `mis784-466303.A3.mobile\_game\_inapp\_purchases` m

WHERE m.LastPurchaseDate IS NOT NULL

GROUP BY m.Customer\_ID;

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_inapp\_rfm` AS

SELECT

a.\*,

NTILE(5) OVER (ORDER BY a.days\_since\_last\_purchase DESC, a.Customer\_ID) AS R\_score,

NTILE(5) OVER (ORDER BY a.purchase\_frequency ASC, a.Customer\_ID) AS F\_score,

NTILE(5) OVER (ORDER BY a.purchase\_total\_amount ASC, a.Customer\_ID) AS M\_score

FROM `mis784-466303.A3.v\_inapp\_agg` a;

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_inapp\_segment` AS

SELECT

Customer\_ID,

CASE

WHEN R\_score IN (4,5) AND F\_score = 5 AND M\_score = 5 THEN 'Cannot Lose'

WHEN R\_score = 5 AND F\_score IN (4,5) AND M\_score IN (3,4,5) THEN 'Active Fans'

WHEN R\_score = 5 AND F\_score IN (2,3) AND M\_score IN (4,5) THEN 'Promising Newbies'

WHEN R\_score IN (3,4,5) AND F\_score IN (2,3,4) AND M\_score IN (2,3,4) THEN 'At Risk'

ELSE 'Other'

END AS inapp\_rfm\_segment

FROM `mis784-466303.A3.v\_inapp\_rfm`;

-- campaigns – dedup customer+campaign and attach campaign type

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_campaign\_resp\_dedup` AS

SELECT

Customer\_ID, Campaign\_ID,

MAX(Response) AS any\_response,

AVG(IFNULL(ClickThroughRate,0)) AS avg\_ctr\_per\_person,

AVG(IFNULL(EngagementFrequency,0)) AS avg\_eng\_per\_person

FROM `mis784-466303.A3.campaign\_response`

GROUP BY Customer\_ID, Campaign\_ID;

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_campaign\_kpi\_person` AS

SELECT

r.Customer\_ID,

r.Campaign\_ID,

m.CampaignType,

r.any\_response,

r.avg\_ctr\_per\_person,

r.avg\_eng\_per\_person

FROM `mis784-466303.A3.v\_campaign\_resp\_dedup` r

JOIN `mis784-466303.A3.marketing\_campaigns` m USING (Campaign\_ID);

-- unified rfm view

CREATE OR REPLACE VIEW `mis784-466303.A3.v\_rfm\_union` AS

SELECT

'donations' AS domain,

d.Customer\_ID,

d.last\_donation\_date AS last\_txn\_date,

d.days\_since\_last\_donation AS days\_since\_last,

d.donation\_frequency AS frequency,

d.donation\_total\_amount AS total\_amount,

d.R\_score, d.F\_score, d.M\_score,

s.donation\_rfm\_segment AS segment

FROM `mis784-466303.A3.v\_donation\_rfm` d

JOIN `mis784-466303.A3.v\_donation\_segment` s USING (Customer\_ID)

UNION ALL

SELECT

'inapp' AS domain,

i.Customer\_ID,

i.last\_purchase\_date AS last\_txn\_date,

i.days\_since\_last\_purchase AS days\_since\_last,

i.purchase\_frequency AS frequency,

i.purchase\_total\_amount AS total\_amount,

i.R\_score, i.F\_score, i.M\_score,

s.inapp\_rfm\_segment AS segment

FROM `mis784-466303.A3.v\_inapp\_rfm` i

JOIN `mis784-466303.A3.v\_inapp\_segment` s USING (Customer\_ID);

-- donor behaviour profiling

-- unified rfm breakdown

SELECT

'inapp' AS domain,

d.Customer\_ID,

d.last\_purchase\_date AS last\_txn\_date,

d.days\_since\_last\_purchase AS days\_since\_last,

d.purchase\_frequency AS frequency,

ROUND(d.purchase\_total\_amount,2) AS total\_amount,

d.R\_score, d.F\_score, d.M\_score,

s.inapp\_rfm\_segment AS segment,

FROM `mis784-466303.A3.v\_inapp\_rfm` d

JOIN `mis784-466303.A3.v\_inapp\_segment` s USING (Customer\_ID)

UNION ALL

SELECT

'donations' AS domain,

d.Customer\_ID,

d.last\_donation\_date AS last\_txn\_date,

d.days\_since\_last\_donation AS days\_since\_last,

d.donation\_frequency AS frequency,

ROUND(d.donation\_total\_amount,2) AS total\_amount,

d.R\_score, d.F\_score, d.M\_score,

s.donation\_rfm\_segment AS segment,

FROM `mis784-466303.A3.v\_donation\_rfm` d

JOIN `mis784-466303.A3.v\_donation\_segment` s USING (Customer\_ID)

ORDER BY domain, R\_score DESC, F\_score DESC, M\_score DESC, total\_amount DESC;

A screenshot of a computer

AI-generated content may be incorrect.

-- unified rfm segment sizes and value

SELECT

domain,

segment,

COUNT(\*) AS num\_customers,

ROUND(SUM(total\_amount),2) AS segment\_total\_amount,

ROUND(AVG(total\_amount),2) AS avg\_amount\_per\_customer,

ROUND(AVG(frequency),2) AS avg\_frequency

FROM `mis784-466303.A3.v\_rfm\_union`

GROUP BY domain, segment

ORDER BY domain, segment\_total\_amount DESC;

A screenshot of a computer

AI-generated content may be incorrect.  
 -- revenue optimisation

-- monthly/seasonal trends by product (donations)

SELECT

FORMAT\_DATE('%Y-%m', DATE\_TRUNC(DonationDate, MONTH)) AS month,

Product AS product\_type,

COUNT(\*) AS num\_transactions,

ROUND(SUM(DonationAmount),2) AS total\_amount,

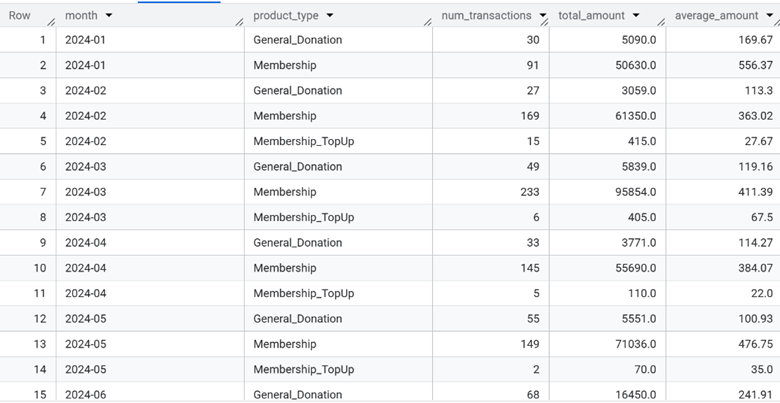
ROUND(AVG(DonationAmount),2) AS average\_amount

FROM `mis784-466303.A3.tradition\_donation\_patterns`

WHERE DonationDate IS NOT NULL AND DonationAmount IS NOT NULL

GROUP BY month, product\_type

ORDER BY month, product\_type;



-- high-value contributors (top 5%) across both domains

WITH don\_b AS (

SELECT APPROX\_QUANTILES(DonationAmount,100)[OFFSET(95)] AS p95

FROM `mis784-466303.A3.tradition\_donation\_patterns`

WHERE DonationAmount IS NOT NULL),

inapp\_b AS (

SELECT APPROX\_QUANTILES(InAppPurchaseAmount,100)[OFFSET(95)] AS p95

FROM `mis784-466303.A3.mobile\_game\_inapp\_purchases`

WHERE InAppPurchaseAmount IS NOT NULL),

u AS (

SELECT 'donations' AS domain, Customer\_ID, DonationAmount AS amount

FROM `mis784-466303.A3.tradition\_donation\_patterns`

WHERE DonationAmount IS NOT NULL

UNION ALL

SELECT 'inapp' AS domain, Customer\_ID, InAppPurchaseAmount AS amount

FROM `mis784-466303.A3.mobile\_game\_inapp\_purchases`

WHERE InAppPurchaseAmount IS NOT NULL AND LastPurchaseDate IS NOT NULL)

SELECT

domain,

COUNT(\*) AS hv\_txn,

ROUND(SUM(amount),2) AS hv\_total\_amount,

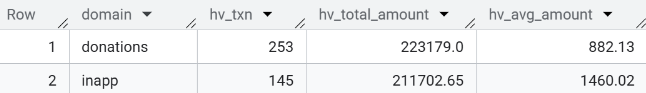
ROUND(AVG(amount),2) AS hv\_avg\_amount

FROM u, don\_b, inapp\_b

WHERE (domain = 'donations' AND amount >= don\_b.p95) OR (domain = 'inapp' AND amount >= inapp\_b.p95)

GROUP BY domain

ORDER BY hv\_total\_amount DESC;



-- game genre x device performance (in-app) – revenue + engagement

SELECT

IFNULL(GameGenre,'Missing') AS game\_type,

IFNULL(Device,'Missing') AS device\_type,

COUNT(\*) AS txn\_count,

ROUND(SUM(IFNULL(InAppPurchaseAmount,0)),2) AS total\_revenue,

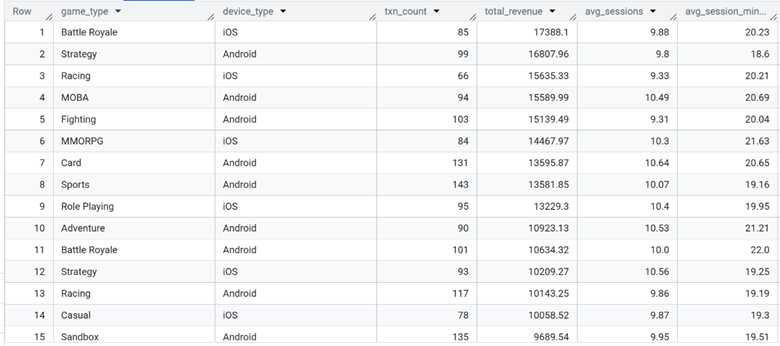
ROUND(AVG(SessionCount),2) AS avg\_sessions,

ROUND(AVG(AverageSessionLength),2) AS avg\_session\_minutes

FROM `mis784-466303.A3.mobile\_game\_inapp\_purchases`

GROUP BY game\_type, device\_type

ORDER BY total\_revenue DESC;



-- campaign effectiveness by type and audience

WITH camp\_kpi AS (

SELECT

Campaign\_ID,

AVG(IF(any\_response, 1, 0)) AS response\_rate,

AVG(avg\_ctr\_per\_person) AS ctr,

AVG(avg\_eng\_per\_person) AS engagement

FROM `mis784-466303.A3.v\_campaign\_kpi\_person`

GROUP BY Campaign\_ID)

SELECT

m.CampaignType AS campaign\_type,

m.TargetAudience AS audience,

COUNT(\*) AS num\_campaigns,

ROUND(AVG(k.response\_rate), 2) AS avg\_response\_rate,

ROUND(AVG(k.ctr), 2) AS avg\_ctr,

ROUND(AVG(k.engagement), 2) AS avg\_engagement,

ROUND(SUM(m.CampaignBudget), 2) AS total\_budget

FROM `mis784-466303.A3.marketing\_campaigns` m

LEFT JOIN camp\_kpi k USING (Campaign\_ID)

GROUP BY campaign\_type, audience

ORDER BY avg\_response\_rate DESC, avg\_engagement DESC;

A screenshot of a computer

AI-generated content may be incorrect.

-- campaign response by donation rfm segment and campaign type

SELECT

s.donation\_rfm\_segment,

p.CampaignType AS campaign\_type,

ROUND(AVG(IF(p.any\_response,1,0)),2) AS response\_rate,

ROUND(AVG(p.avg\_ctr\_per\_person),2) AS avg\_ctr,

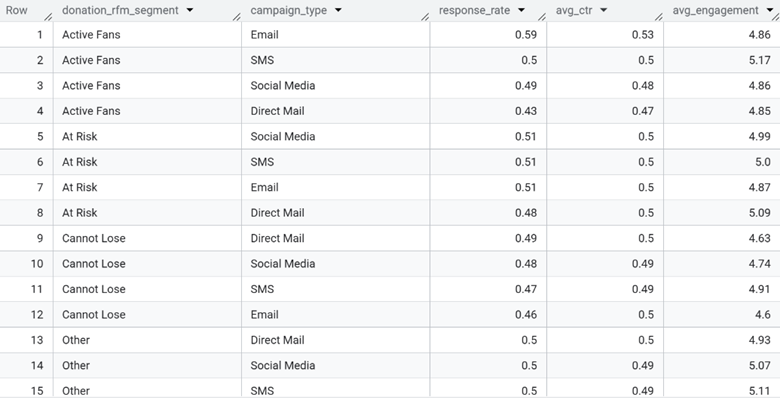
ROUND(AVG(p.avg\_eng\_per\_person),2) AS avg\_engagement

FROM `mis784-466303.A3.v\_campaign\_kpi\_person` p

JOIN `mis784-466303.A3.v\_donation\_segment` s USING (Customer\_ID)

GROUP BY s.donation\_rfm\_segment, campaign\_type

ORDER BY s.donation\_rfm\_segment, response\_rate DESC;



-- campaign cost per engaged person by campaign type

-- calculates both cost per unique engaged person and cost per engagement

WITH joined AS (

SELECT

m.CampaignType,

m.Campaign\_ID,

m.CampaignBudget,

p.Customer\_ID,

p.any\_response

FROM `mis784-466303.A3.marketing\_campaigns` m

LEFT JOIN `mis784-466303.A3.v\_campaign\_kpi\_person` p USING (Campaign\_ID)),

agg AS (

SELECT

CampaignType,

ROUND(SUM(CampaignBudget), 2) AS total\_budget,

SUM(IF(any\_response, 1, 0)) AS total\_engagements,

COUNT(DISTINCT IF(any\_response, Customer\_ID, NULL)) AS engaged\_people\_unique

FROM joined

GROUP BY CampaignType)

SELECT

CampaignType,

total\_budget,

total\_engagements,

engaged\_people\_unique,

ROUND(SAFE\_DIVIDE(total\_budget, NULLIF(engaged\_people\_unique, 0)), 2) AS cost\_per\_unique\_engaged,

ROUND(SAFE\_DIVIDE(total\_budget, NULLIF(total\_engagements, 0)), 2) AS cost\_per\_engagement

FROM agg

ORDER BY cost\_per\_unique\_engaged ASC;

A screenshot of a computer

AI-generated content may be incorrect.

## Meeting Minute

### Meeting Minute 1

| **Unit Code:** | | MIS784 Marketing Analytics T2 2025 | | |
| --- | --- | --- | --- | --- |
| **Assignment Group Number:** | | Group 60 | | |
| **Meeting attended by:** | | · Jayseon Choi  · Simran Thakur  · Lee Hoang | | |
| **Absent from meeting:** | | None | | |
| **Date:** | | 2025/9/14 | | |
| **Time:** | | 5:00pm-5:45pm | | |
|  | | | | |
| **Meeting Agenda #1: Introduction** | | | | |
| **Note:** team members introduce themselves | | | | |
|  | | | | |
| **Meeting Agenda #2: Assignment Discussion** | | | | |
| **Note:** go through the group assignment description together to ensure that all group members are familiar with the requirements of the assignment. | | | | |
|  | | | | |
| **Meeting Agenda #3: Division of Work** | | | | |
| **Note**: assign work contents to each group member for further assignment progress. | | | | |
|  | | | | |
| **Action items** | | | | |
| **Task** | **Status** | | **Who** | **Due by** |
| Identify patterns of donor behaviour  Analyse donor churn  Guide future donation activities |  | | Lee    Simran  Jayseon | Sunday (21/09/2025) |
|  | | | | |
| **Agenda for the next meeting:** | | · Pitch each member’s analysis  · Discuss the analysis  · Discuss the 4th (main requirement)  · By Thursday (18/09): Flag data quality to the team via the chat channel (no meeting) | | |
|  | | | | |
| **Adjournment** | | | | |
| · Adjourned at: 2025/9/18 8:51pm  · Next meeting will be held at: 2025/09/21 5:00-6:00pm | | | | |
|  | | | | |
| **Minute submitted by:** | | Lee Hoang | | |
|  | | | | |
| **Approved by:** | | Simran Thakur  Jayseon | | |

### Meeting Minute 2

| **Unit Code:** | | MIS784 Marketing Analytics T2 2025 | | |
| --- | --- | --- | --- | --- |
| **Assignment Group Number:** | | Group 60 | | |
| **Meeting attended by:** | | · Jayseon Choi  · Simran Thakur  · Lee Hoang | | |
| **Absent from meeting:** | | None | | |
| **Date:** | | 2025/9/28 | | |
| **Time:** | | 6:00pm-6:45pm | | |
|  | | | | |
| **Meeting Agenda #1: Summary of individual analysis** | | | | |
| **Note:** | | | | |
|  | | | | |
| **Meeting Agenda #2: Consolidate individual’s analysis to make the analysis section consistent** | | | | |
| **Note:** | | | | |
|  | | | | |
| **Action items** | | | | |
| **Task** | **Status** | | **Who** | **Due by** |
| Fix rfm and rerun churn    All team members to finalise analysis |  | | Jayseon and Simran | 29/09/2025    30/09/2025 |
|  | | | | |
| **Agenda for the next meeting:** | | · Recommendations | | |
|  | | | | |
| **Adjournment** | | | | |
| · Adjourned at: 2025/9/28 8:25 pm  · Next meeting will be held at: to be finalized once analysis is done | | | | |
|  | | | | |
| **Minute submitted by:** | | Lee Hoang | | |
|  | | | | |
| **Approved by:** | | Simran Thakur  Jayseon | | |

### Meeting Minute 3

| **Unit Code:** | | MIS784 Marketing Analytics T2 2025 | | |
| --- | --- | --- | --- | --- |
| **Assignment Group Number:** | | Group 60 | | |
| **Meeting attended by:** | | · Jayseon Choi  · Simran Thakur  · Lee Hoang | | |
| **Absent from meeting:** | | None | | |
| **Date:** | | 2025/10/01 | | |
| **Time:** | | 7:10pm-7:25pm | | |
|  | | | | |
| **Meeting Agenda #1: Finalise analysis** | | | | |
|  | | | | |
| **Meeting Agenda #2: Discuss recommendations** | | | | |
|  | | | | |
| **Action items** | | | | |
| **Task** | **Status** | | **Who** | **Due by** |
| Personalized outreach  Predictive Retention Model  App-based incentives |  | | Lee    Simran    Jayseon | Friday (03/10/2025) |
|  | | | | |
| **Agenda for the next meeting:** | | N/A | | |
|  | | | | |
| **Adjournment** | | | | |
| Adjourned at: 2025/10/01 7:25pm | | | | |
|  | | | | |
| **Minute submitted by:** | | Lee Hoang | | |
|  | | | | |
| **Approved by:** | | Simran Thakur  Jayseon | | |