MILESTONE 3

DATABASE SYSTEM FOR SEATTLE CAST

HAERY LEE, JILLIAN PFLUGRATH, SAKSHI MADAN

1. Problem Statement

Our mission is to design a database system for Seattle Cast, a podcast network in the Seattle area, so they can efficiently aggregate download data and calculate key metrics for measuring podcasts' success. Building an efficient database system will improve existing listener analytic strategies, equipping them to make better business decisions for their audience and attract more lucrative advertising campaigns.

2. Description

For new podcast networks, creating consistent revenue streams from their podcasts can be a challenge. Many podcasts make money by signing deals with sponsors that pay cost per mile (CPM): sponsor pays a set amount for every 1,000 downloads. For this approach to be lucrative, a show must have many thousands of downloads. Shows can also arrange cost per acquisition (CPA) deals with sponsors: sponsor pays a set amount for every customer that is acquired through the podcast advertisement. This is often what is happening when advertisements provide a specific discount code or URL. Cost per acquisition can be effective for newer shows that have not yet amassed hundreds of thousands of downloads but have smaller but very engaged audiences. For CPA deals to be effective, podcasts must understand who is downloading their podcast and what products those people are likely to buy. People can also rate and/or review a podcast. These ratings can improve a podcast's rankings, making it easier for new people to discover the podcast and hopefully download episodes or subscribe. With robust analytics, podcast networks can understand their podcasts' downloaders and analyze the effectiveness of sponsor campaigns. Ideally, their data is structured so that key metrics and insights can be extracted easily.

Seattle Cast is a new podcast network with ten Seattle-based podcasts: sports, tech, food, music, comedy, news and outdoor activities. Each of these podcasts already has multiple episodes available for download via Apple's podcast application. People can subscribe to a podcast, ensuring that each new episode will be automatically downloaded to their listening device. Or, without subscribing, people can manually download individual episodes of the podcast. For Seattle Cast, podcast episodes have three advertisement slots: pre-roll (before episode content), mid-roll (in the middle of episode content), and post-roll (after episode content).

Seattle Cast's goal is to attract more sponsors so they can fill all open advertisement slots on upcoming episodes and establish a more consistent revenue stream from their podcasts. To attract and retain the right sponsors, they need robust analytics about the people who download and subscribe to their podcasts. Our custom database system will equip Seattle Cast to make sure all advertising slots are filled, track the success of advertising campaigns, and easily extract the metrics and insights needed to attract future sponsors as well as choose the right campaigns to match their audience.

3. SQL Queries

1. Number of subscribers per podcast category
Each podcast is labeled a certain category based on the topics covered in the Podcast. This table returns the number of subscribers per category type. Number of subscribers is one way to measure performance. This helps us know which categories are most popular and can provide guidance on which categories to focus on in the future or which categories to drop.

Query

```
SELECT P.Category, COUNT(C.CustomerID) as 'Subscribers per Category'
FROM mm_cpsc5910team06.Podcast as P
JOIN mm_cpsc5910team06.Podcast_Customer as C
ON P.PodcastID = C.PodcastID
GROUP BY P.Category
ORDER BY COUNT(C.CustomerID) DESC
```

Output

Category	Subscribers per Category
Food	367
News	361
Music	167
Comedy	167
Sports	135
Recreation	98
Tech	51

2. Average Cost per Acquisition per Podcast
This table displays two measures of podcast revenue generation performance: Which podcasts have been able to negotiate higher CPA rates on average for sponsor campaigns, and which podcasts have secured the most ad campaigns? By having these performance measures, we can monitor a podcast's sponsor campaign performance and create goals for future sponsor campaigns.

Query

```
SELECT P.PodcastName, avg(S.CPAinUSD) as AverageCPA, count(AdSpot) as
NoOfCampaigns
FROM mm_cpsc5910team06.Podcast_Sponsor S
JOIN mm_cpsc5910team06.Podcast P
ON S.PodcastID = P.PodcastID
GROUP BY S.PodcastID
ORDER BY AverageCPA ASC
```

Output

Podcast Name	AverageCPA	NoOfCampaigns
Seahawk Session	14.125	8
Seattle Sounds	15.125	16
Seattle Laughs	15.38888888888889	18
Grapes and Hops	15.4375	16
Sport Report	15.541666666666666	24
Emerald City Good	15.6	25
Puget Sound	16.130434782608695	23
PNW Hiking Guide	18.571428571428573	7
Tech Talk	18.842105263157894	19
Seattle Weekly Update	20.105263157894736	19

3. Total Number of Subscribers per Podcast
This provides another way of measuring podcast performance: the total number of subscribers. We can see which podcasts have the most subscribers (i.e., committed listeners) and which have the lowest.

Query
```
SELECT P.PodcastName, count(distinct(C.CustomerID)) as 'TotalNoOfSubscribers'
FROM mm_cpsc5910team06.Customer C
JOIN mm_cpsc5910team06.Episode_Customer E
ON C.CustomerID = E.CustomerID
JOIN mm_cpsc5910team06.Podcast P
ON E.PodcastID = P.PodcastID
GROUP BY E.PodcastID
```

Output

PodcastName	TotalNoOfSubscribers
Tech Talk	'327'
Sport Report	'300'
Grapes and Hop	'311'
Seattle Sounds	'377'
Emerald City Good	'331'
PNW Hiking Guide	'274'
Seattle Laughs	'380'
Seahawk Session	'357'
Puget Sound	'482'
Seattle Weekly Update	'525'

4. Average rating for each podcast
This creates a table with average ratings for each podcast. The information is crucial to know the trend/style of podcast programs that attract listeners, also to inform sponsors which podcast is rated highly from listeners.

Query

```
SELECT P.PodcastID, P.PodcastName, ROUND(avg(PC.Rating),2) as AverageRating
FROM mm_cpsc5910team06.Podcast as P, mm_cpsc5910team06.Podcast_Customer as PC
WHERE P.PodcastID = PC.PodcastID
GROUP BY PodcastID;
```

Output

PodcastID	PodcastName	AverageRating
1	Tech Talk	3.73
2	Sport Report	3.46
3	Grapes and Hops	3.53
4	Seattle Sounds	3.51
5	Emerald City Good	3.39
6	PNW Hiking Guide	3.60
7	Seattle Laughs	3.26
8	Seahawk Session	3.59
9	Puget Sound	3.57
10	Seattle Weekly Update	3.36

5. The average day difference between date of launch and date of download of episodes for a particular podcast

This query gives a table that shows the average number of days between the launch date of an episode and the date that each customer downloaded that episode for a particular podcast. In this case, we have used Podcast = 1. From this information, we can track which episodes are most popular (with lesser AvgDaysLaunchToDownload) for each Podcast and even the advertisers can benefit the most from them.

Query

```
SELECT E.PodcastID, E.EpisodeID, ROUND(AVG(datediff(EC.DateOfDownload,
            E.DateOfLaunch)),0) AS AvgDaysLaunchToDownload
FROM mm_cpsc5910team06.Episode as E, mm_cpsc5910team06.Episode_Customer as EC
WHERE E.EpisodeID = EC.EpisodeID AND
         E.PodcastID = EC.PodcastID AND
         E.PodcastID = 1
GROUP BY E.PodcastID, E.EpisodeID
ORDER BY E.PodcastID, AVG(datediff(EC.DateOfDownload, E.DateOfLaunch))
```

Output

PodcastID	EpisodeID	AvgDaysLaunchToDownload
1	10	62
1	9	79
1	8	80
1	7	84
1	6	96
1	5	107
1	4	110
1	3	126

1	1	147
1	2	149

6. Number of subscribers who haven't reviewed or rated the shows
The query shows a table that gives an idea of the number of subscribers who have not rated and reviewed the shows. We can target those people by sending emails asking them to review/rate a show as those attract more people

Query

```
SELECT PC.PodcastID, P.PodcastName, Count(PC.CustomerID) as CountOfNoFeedback
FROM mm_cpsc5910team06.Podcast_Customer AS PC JOIN mm_cpsc5910team06.Podcast
        AS P
ON PC.PodcastID = P.PodcastID
WHERE Subscriber = 'T' AND
        (Rating  is NULL or Review is NULL)
GROUP BY PodcastID
```

Output

PodcastID	PodcastName	CountOfNoFeedback
1	Tech Talk	30
4	Seattle Sounds	65
6	PNW Hiking Guide	70

7. Country with maximum number of subscriptions/reviews per podcast
This creates a table that shows which country has the maximum number of subscriptions/reviews for each podcast. This information can be utilized to target sponsors having their presence in other countries as well.

Query

```
SELECT Temp1.PodcastID, PodcastName, Temp1.Country
FROM
        (SELECT PC.PodcastID ,C.Country, COUNT(PC.CustomerID) as 'CountofCustomers'
FROM mm_cpsc5910team06.Podcast_Customer as PC join
        mm_cpsc5910team06.Customer as C
        ON PC.CustomerID = C.CustomerID
        GROUP BY PC.PodcastID, C.Country
        ORDER BY PC.PodcastID, COUNT(PC.CustomerID) DESC) as Temp1,
        mm_cpsc5910team06.Podcast
WHERE Temp1.CountofCustomers = (SELECT MAX(Temp2.CountofCustomers)
                                FROM
                                        (SELECT PC.PodcastID ,C.Country,
                                         COUNT(PC.CustomerID) as 'CountofCustomers'
                                         FROM mm_cpsc5910team06.Podcast_Customer
                                                as PC
                                         JOIN mm_cpsc5910team06.Customer as C
                                         ON PC.CustomerID = C.CustomerID
                                         GROUP BY PC.PodcastID, C.Country
                                         ORDER BY PC.PodcastID,
COUNT(PC.CustomerID)DESC) as Temp2
WHERE Temp2.PodcastID = Temp1.PodcastID)
AND Temp1.PodcastID = Podcast.PodcastID;
```

Output

PodcastID	PodcastName	Country
1	Tech Talk	Russia
2	Sport Report	United States
3	Grapes and Hops	Indonesia
4	Seattle Sounds	Nigeria
5	Emerald City Good	United Kingdom
6	PNW Hiking Guide	United States
7	Seattle Laughs	United States
8	Seahawk Session	China
9	Puget Sound	Russia
10	Seattle Weekly Update	Canada
10	Seattle Weekly Update	Indonesia

There are two rows for Seattle Weekly Update because Canada and Indonesia have same number of subscribers/reviews for this podcast.

8. Total number of downloads per podcast
This creates a table that shows the total number of downloads for each podcast. The cost per acquisition (CPA) can be decided based on the number of downloads. For higher number of downloads, the CPA can be higher as well.

Query
```
SELECT PC.PodcastID, P.PodcastName, COUNT(PC.CustomerID) as 'NoofDownloads'
FROM mm_cpsc5910team06.Podcast_Customer AS PC JOIN mm_cpsc5910team06.Podcast
AS P
ON PC.PodcastID = P.PodcastID
GROUP BY PC.PodcastID
ORDER BY PC.PodcastID;
```

Output

PodcastID	PodcastName	NoofDownloads
1	Tech Talk	51
2	Sport Report	96
3	Grapes and Hops	171
4	Seattle Sounds	91
5	Emerald City Good	196
6	PNW Hiking Guide	98
7	Seattle Laughs	167
8	Seahawk Session	39
9	Puget Sound	76

10	Seattle Weekly Update	361

9. Average age of customers subscribing to the podcast

This creates a table that shows the average age of people subscribing to the show. Based on this age, the podcast company can select which sponsors should be targeted.

Query

```
SELECT Temp.PodcastID, P.PodcastName, Temp.AverageAge
FROM
        (SELECT PC.PodcastID, ROUND(AVG((DATEDIFF(now(), C.Dob)/365)),0) as
                'AverageAge'
        FROM mm_cpsc5910team06.Podcast_Customer AS PC JOIN
                mm_cpsc5910team06.Customer AS C
        ON C.CustomerID = PC.CustomerID
        GROUP BY PC.PodcastID) AS Temp
        JOIN mm_cpsc5910team06.Podcast as P
        ON Temp.PodcastID = P.PodcastID
ORDER BY Temp.AverageAge
```

Output

PodcastID	PodcastName	AverageAge
2	Sport Report	33
4	Seattle Sounds	34
9	Puget Sound	34
10	Seattle Weekly Update	34
3	Grapes and Hops	35
6	PNW Hiking Guide	35
5	Emerald City Good	36
1	Tech Talk	37
7	Seattle Laughs	39
8	Seahawk Session	43

10. Number of customers greater than 50 with at least one-click per podcast's episode
This creates a table with the podcasts and episodes having at least 50 customers who are interested in advertisements; hence they click them at least once.

Query

```
SELECT EC.PodcastID, P.PodcastName, EC.EpisodeID, COUNT(DISTINCT EC.CustomerID)
         AS 'NoOfOneClickCustomers'
FROM mm_cpsc5910team06.Episode_Customer AS EC JOIN mm_cpsc5910team06.Podcast
        AS P
ON EC.PodcastID = P.PodcastID
WHERE ClicksOnAds != 0
GROUP BY PodcastID, EpisodeID
```

```
HAVING COUNT(DISTINCT CustomerID) >= 50
ORDER BY COUNT(DISTINCT EC.CustomerID) DESC;
```

Output

PodcastID	PodcastName	EpisodeID	NoOfOnceClick Customers
10	Seattle Weekly Update	10	113
10	Seattle Weekly Update	2	113
10	Seattle Weekly Update	6	107
9	Puget Sound	2	101
9	Puget Sound	6	99
1	Tech Talk	5	93
8	Seahawk Session	1	93
1	Tech Talk	9	92
9	Puget Sound	10	89
4	Seattle Sounds	5	66
4	Seattle Sounds	9	66
10	Seattle Weekly Update	3	64
7	Seattle Laughs	2	59
10	Seattle Weekly Update	7	58
5	Emerald City Good	10	56
5	Emerald City Good	6	54
7	Seattle Laughs	4	53
6	PNW Hiking Guide	3	52
6	PNW Hiking Guide	7	52
7	Seattle Laughs	6	52
8	Seahawk Session	4	52
6	PNW Hiking Guide	4	51
5	Emerald City Good	2	51
7	Seattle Laughs	10	50
2	Sport Report	5	50

Stored Procedures

1) Retrieve average rating of given podcast

Creation of Store Procedure

```
CREATE procedure Podcast_AvgRating_info(IN input int)
```

```
SELECT P.PodcastID, P.PodcastName, ROUND(avg(PC.Rating),2) as AverageRating
FROM Podcast as P, Podcast_Customer as PC
WHERE P.PodcastID = input
GROUP BY P.PodcastID;
```

<u>Calling Stored Procedure</u>

```
call mm_cpsc5910team06.Podcast_AvgRating_info(2);
```

<u>Output</u>

PodcastID	PodcastName	Average Rating
2	Sport Report	4.06

2) Retrieve number of customers greater than 50 with at least one-click per podcast's episode

<u>Creation of Stored Procedure</u>

```
CREATE procedure sp_active_customers (IN count int)

SELECT EC.PodcastID, P.PodcastName, EC.EpisodeID, COUNT(DISTINCT EC.CustomerID)
        AS 'NoOfOneClickCustomers'
FROM mm_cpsc5910team06.Episode_Customer AS EC JOIN mm_cpsc5910team06.Podcast
       AS P
ON EC.PodcastID = P.PodcastID
WHERE ClicksOnAds != 0
GROUP BY PodcastID, EpisodeID
HAVING COUNT(DISTINCT CustomerID) >= count
ORDER BY COUNT(DISTINCT EC.CustomerID) DESC
```

<u>Calling Stored Procedure</u>

```
call mm_cpsc5910team06.sp_active_customers(90);
```

<u>Output</u>

PodcastID	PodcastName	EpisodeID	NoOfResponsiveCustomers
10	Seattle Weekly Update	10	113
10	Seattle Weekly Update	2	113
10	Seattle Weekly Update	6	107
9	Puget Sound	2	101
9	Puget Sound	6	99
1	Tech Talk	5	93
8	Seahawk Session	1	93
1	Tech Talk	9	92

3) Retrieve average day difference between date of launch and date of download of episodes for a particular podcast

```
CREATE procedure mm_cpsc5910team06.sp_day_difference (IN podID int)

SELECT E.PodcastID, E.EpisodeID, ROUND(AVG(datediff(EC.DateOfDownload,
            E.DateOfLaunch)),0) AS AvgDaysLaunchToDownload
FROM mm_cpsc5910team06.Episode as E, mm_cpsc5910team06.Episode_Customer as EC
WHERE E.EpisodeID = EC.EpisodeID AND
         E.PodcastID = EC.PodcastID AND
         E.PodcastID = podID
GROUP BY E.PodcastID, E.EpisodeID
ORDER BY E.PodcastID, AVG(datediff(EC.DateOfDownload, E.DateOfLaunch))
```

Calling Stored Procedure

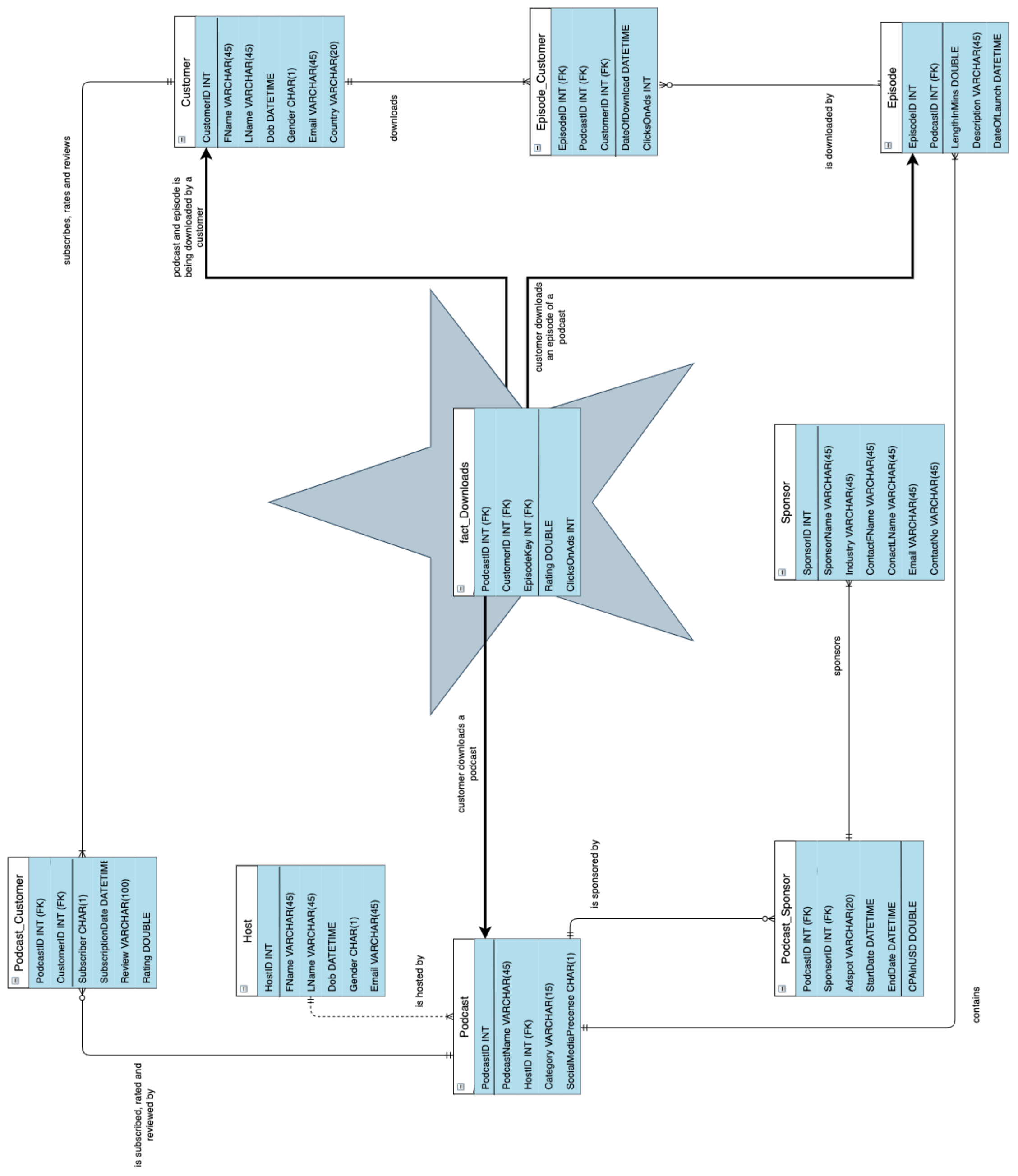
```
CALL mm_cpsc5910team06.sp_day_difference(2);
```

Output

PodcastID	EpisodeID	AvgDaysLaunchToDownload
2	10	71
2	9	73
2	8	87
2	6	88
2	7	90
2	5	97
2	4	113
2	3	116
2	1	144
2	2	147

All the queries have been converted to views in the database on AWS.

4. Physical Model Diagram



5. Accessing MYSQL Database

Following are the credentials to access our MYSQL database stored in AWS:

URL/Hostname	mmcpsc5910team06.cn2c7f4fqc61.us-east-1.rds.amazonaws.com
Username	admin
Password	adminadmin

6. Accessing Tableau Public Site

Following is the link to access our Tableau Public site:

https://public.tableau.com/views/PodcastVisualizations/PocastMetrics?:display_count=y&publish=yes&:origin=viz_share_link

Notes:

- We have attached the database dump as file name **milestone3.sql** in the submission.
- We have attached the power point as file name **milestone3.pptx** in the submission.
- We have attached the updated physical model as file name **milestone3.mwb** in the submission.
- We have attached the tableau workbook as file name **milestone3.twbx** in the submission.