

### Stored Procedure SQL Code:

The below code loops through the entries in the Top\_Saved\_Track to find the top-rated and trending songs based on their rating types and popularity for the tracks in the table. The purpose of the stored procedure is to display the different rating types that users listen to across the platform. Using the rating types which depicts the length of time the affinities have been calculated and analyzed, and after we filter based on popularity to provide the top and most popular suggestions to our users. This allows any given user to see trending and top-rated songs to utilize as recommendations for their own songs and track playlists. The main utilization we wanted to highlight was recommending our users new and trending songs that they might be interested in adding, which the stored procedure coupled with the CRUD operations accomplishes.

DELIMITER //

```
CREATE PROCEDURE GetHighestRatedTracks()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE trackIDVar VARCHAR(255);
    DECLARE trackNameVar VARCHAR(255);
    DECLARE userIDVar VARCHAR(255);
    DECLARE MaxShortRating DECIMAL(5,2);
    DECLARE MaxMediumRating DECIMAL(5,2);
    DECLARE MaxLargeRating DECIMAL(5,2);
    DECLARE maxRating DECIMAL(5,2);
    DECLARE ratingType VARCHAR(50);
    DECLARE trackPopularity INT;

    DECLARE curTracks CURSOR FOR
        SELECT DISTINCT TrackID, UserID
        FROM Top_Saved_Track;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN curTracks;

    trackLoop: LOOP
        FETCH curTracks INTO trackIDVar, userIDVar;
        IF done THEN
            LEAVE trackLoop;
        END IF;
```

```

SELECT
    MIN(ShortRating) AS MaxShortRating,
    MIN(MediumRating) AS MaxMediumRating,
    MIN(LargeRating) AS MaxLargeRating
INTO
    MaxShortRating, MaxMediumRating, MaxLargeRating
FROM Top_Saved_Track
WHERE TrackID = trackIDVar AND UserID = userIDVar;

SET maxRating = 101;
SET ratingType = "";

IF MaxShortRating <= maxRating THEN
    SET maxRating = MaxShortRating;
    SET ratingType = 'ShortRating';
END IF;

IF MaxMediumRating <= maxRating THEN
    SET maxRating = MaxMediumRating;
    SET ratingType = 'MediumRating';
END IF;

IF MaxLargeRating <= maxRating THEN
    SET maxRating = MaxLargeRating;
    SET ratingType = 'LargeRating';
END IF;

SELECT Track_name, Popularity INTO trackNameVar, trackPopularity
FROM Track
WHERE TrackID = trackIDVar;

IF trackPopularity > 90 THEN
    SELECT CONCAT('Highest rated track (' , ratingType, ') - Track ID: ' , trackIDVar, ' ,
        UserID: ' , userIDVar, ' , Track Name: ' , trackNameVar) AS Result;
    END IF;
END LOOP trackLoop;

CLOSE curTracks;

END //

DELIMITER ;

```

Trigger:

This trigger, named UpdateTrackPopularity, updates the popularity of the particular song in the Track Table or deletes the song from the Top\_Saved\_Track table. Everytime a new record is inserted into the table, it first pulls the popularity attribute from the song associated with the track\_id in the Track table and stores that as trackPopularity. Once that is finished, it then calculates an updated popularity value for the song based on the value of the LargeRating. If it is less than 3 then the song's popularity increases by 1, otherwise it decreases by 1. After that the trigger checks whether the adjusted popularity is 0, and if it is then it deletes the song from the Top\_Saved\_Track table, otherwise it updates the song's popularity in the Track table. This trigger reflects user interactions through dynamically adjusting the popularity of the song. If a track's popularity drops too low then it gets dropped from the Top\_Saved\_Track table, otherwise it gets an adjusted popularity in the Track Table.

DELIMITER //

```
CREATE TRIGGER UpdateTrackPopularity
AFTER INSERT ON Top_Saved_Track
FOR EACH ROW
BEGIN
    DECLARE trackPopularity INT;
    SET trackPopularity = (SELECT Popularity FROM Track WHERE TrackID = NEW.TrackID);

    IF NEW.LargeRating < 3 THEN
        SET trackPopularity = trackPopularity + 1;
    ELSE
        SET trackPopularity = trackPopularity - 1;
    END IF;

    IF trackPopularity = 0 THEN
        DELETE FROM Top_Saved_Track WHERE TrackID = NEW.TrackID;
    ELSE
        UPDATE Track SET Popularity = trackPopularity WHERE TrackID = NEW.TrackID;
    END IF;

END //
```

DELIMITER;

