Personalized Workout Playlists

Sravya Kaniti
Geethika Balasubramanian
Dagim Bantikassegn
Vidhat Bandi

Use Case

- Create a workout playlist for a given user based on their desired workout
  - Uses Spotify database
  - Length of playlist dependant on user input
- Song choices based on workout intensity
  - Uses danceability, temp, and energy metrics from the database
- A song used used in a playlist within the past month will not be used in the creation of a new playlist
  - This is to keep playlist tracks fresh
- The user can:
  - save the playlist for future use
  - "dislike" a song to remove it from playlist consideration
  - "like" an artist so the app will select tracks by them more frequently
- App provides a couch-to-5k playlist plan to progressively improve fitness of the user

Business Rules

- From the original dataset:
  - Artists table - list of artists
  - Tracks table - list of tracks
  - Track_artists linking table
    - A track can be associated with many artists
    - An artist can be associated with many tracks
- Artists table and tracks table have a many-to-many relationship, thus the linking table
- New tables:
  - Workouts table - list of possible workouts
  - Saved_playlists - list of playlists the user saved
  - Playlist_workouts linking table
    - A playlist can be associated with many workouts
    - A workout can be associated with many playlists
  - Playlist_tracks linking table
    - A playlist can be associated with many tracks
    - A track can be associated with many playlists

Entity Relationship Diagram
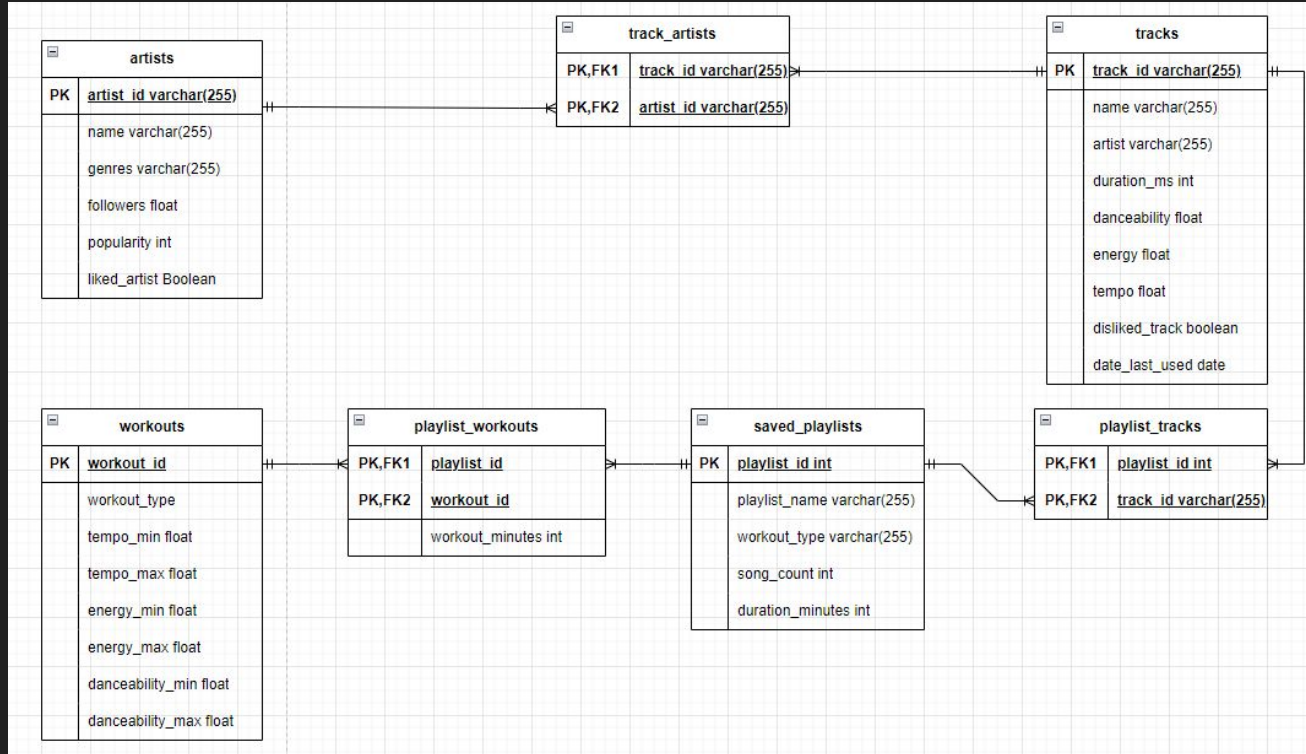
View 1

- This is a view that provides a list of all high energy workouts (minimum energy level=0.4) and the corresponding playlists the user has saved.

- The list also displays the number of songs in each playlist, playlist duration, names of tracks (comma separated) last heard from the playlist.
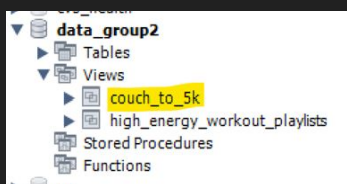
View 2

- Created a view for a couch-to-5k schedule with 6 weeks total
- playlists are provided to match the goal time for 5k
- Users are able to select the number of weeks they want to spread the goal across as well as desired 5k time

```sql
CREATE VIEW
    couch_to_5k
AS
    -- cte to filter song list down to those matching
energy/tempo/danceability
    -- settings for running
    WITH
        run_params
    AS
        (
        SELECT
            tempo_min
            , tempo_max
            , energy_min
            , energy_max
            , danceability_min
            , danceability_max
        FROM
            workouts
        WHERE
            workout_type = 'running'
        )

    -- cte to pull in all tracks matching the above parameters
    , run_tracks
    AS
        (
        SELECT
            ROW_NUMBER() OVER() row_num
            , track_id
            , track_guid
            , name
            , artist
            , duration_ms/1000/60 duration_min
        FROM
            tracks
        INNER JOIN
            run_params
        ON
            tracks.tempo > run_params.tempo_min
            AND tracks.tempo <= run_params.tempo_max
            AND tracks.energy > run_params.energy_min
            AND tracks.energy <= run_params.energy_max
            AND tracks.danceability > run_params.danceability_min
            AND tracks.danceability <= run_params.danceability_max
        WHERE
            tracks.disliked_track = 0
```

```sql
        )
    -- cte to calculate a running total of track length, which will be used
    -- to form playlists
    , playlist_duration
    AS
        (
        SELECT
            row_num
            , name
            , SUM(duration_min) over (order by row_num) AS duration_min
        FROM
            run_tracks
        GROUP BY row_num
        )

    -- cte to show couch to 5k schedule, in practice number of rows and
    -- time_for_5k increments will be costomized to user preferences
    , run_schedule
    AS
        (
        SELECT
            CONCAT('Week ', ROW_NUMBER() OVER()) week_num
            , 60 - (ROW_NUMBER() OVER() - 1) * 5 time_for_5k
        FROM
            run_tracks
        LIMIT 6
        )

    -- creating playlists based on user schedule
    SELECT
        week_num
        , time_for_5k
        , GROUP_CONCAT(name) track_list
        , MAX(duration_min) playlist_length
    FROM
        run_schedule
    LEFT JOIN
        playlist_duration
    ON
        run_schedule.time_for_5k > playlist_duration.duration_min
    WHERE
        duration_min <= time_for_5k
    GROUP BY
        week_num
        , time_for_5k
    ORDER BY
        week_num
        , duration_min desc
```

Data in view:

week_num	time_for_5k	track_list	playlist_length
Week 1	60	Solglasًgon,Krakkalakkarapp,THANXX,Classical Gas - Live At The Royal Festival Hall, London/1...	59.36226667
Week 2	55	Classical Gas - Live At The Royal Festival Hall, London/1977,Fue Por Mi,Chali Aayee,Magkombo,...	53.04026667
Week 3	50	Fue Por Mi,Krakkalakkarapp,Classical Gas - Live At The Royal Festival Hall, London/1977,Wir tick...	49.51160001
Week 4	45	Classical Gas - Live At The Royal Festival Hall, London/1977,Krakkalakkarapp,Wir ticken,¢§ ã,~...	43.37200001
Week 5	40	Fue Por Mi,Krakkalakkarapp,Vã ̃o das Águias,Errinnerungen - Live in Berlin 2000,Noonday Under...	39.41133334
Week 6	35	Magkombo,Ay Dios,Classical Gas - Live At The Royal Festival Hall, London/1977,Noonday Under...	32.11243334

- data_group2
  - Tables
  - Views
    - couch_to_5k
    - high_energy_workout_playlists
  - Stored Procedures
  - Functions

View 3

This is a view that provides a list of all the tracks of user's favorite artists that are not listened to or saved in the playlist before.

```sql
130  ●  DROP VIEW IF EXISTS   new_track_suggestions;
131  ●  CREATE VIEW new_track_suggestions
132     AS
133     SELECT track, artist
134     from(SELECT c.name track, a.name artist, b.track_id
135         FROM artists a
136             INNER JOIN track_artists b ON a.artist_id=b.track_id
137             INNER JOIN tracks c ON b.track_id=c.track_id
138         WHERE c.track_id NOT IN(SELECT track_id FROM playlist_tracks)
139         AND date_last_used IS NULL AND disliked_track=0 AND liked_artist=1)t;
```

Result Grid	Filter Rows:	Search	Export:
track		artist	
All Mata Penkawman		F. Suohoola	
Cardigan		TKKG Retro-Archiv	
EihÃ¤n TÃ¤mÃ¤n Nyt NÃ¤in PitÃ¤nyt Olla		Moira Dela Torre	
MÃ¥nen tur och retur, del 23		DEVO	
We're Alive - Mono; 2003 Remaster		Marty Robbins	
Don't Wanna Live My Life		Wohnout	
El Party		Marjatta LeppÃƒÂ¤nen	
Hail to the Princess Aurora		Fast Eddie	
When I'm Gone		Carpenters	
Yo No Me Voy Acostar		Joan Baez	
Can't Take My Eyes off You		Jacky Cheung	
The Edge of Heaven		Duangjan Suwannee	
The Gallis Pole		A.R. Rahman	
Kanashiki Kyodai Tachi (From "Saint Sei...		Lobo	
ç¾Žã—ã„ã"ä"		Randy Crawford	
Flying Home - Instrumental		Ã¨Â¨Â±Â±Ã©Ë†Â°Ã§Â§Â£Â®	
ÐšÐ°Ð°Ñ‚Ð¾ Ð¿Ñ€ÐµÐ´Ð		The Doors	
Love Me Instead		Armonia 10	
Invention No. 15 in B Minor, BWV 786 -...		Loqiemean	
×'×"/×œœ×·×"×™×"× - ×'×"×·×¤×¢×"¢×"		Chrisye	
ä¸€æ°°´éš"ä¤©©æ¶¶		JosÃƒÂ© Alfredo Jim...	
Haber Gelmiyor Yardan		Cornelius	
Chopin: Nocturne No. 17 in B Major, Op....		Nationalteatern	
Tum Aur Foroih Khoo (Live)		Teresa Teng	

	27	20:51:58	select * from new_track_suggestions LIMIT 0, 1000	51 row(s) returned	0.028 sec / 0.00018...
✓	28	20:52:31	DROP VIEW IF EXISTS new_track_suggestions	0 row(s) affected	0.031 sec
✓	29	20:52:31	CREATE VIEW new_track_suggestions AS SELECT track, artist from(SELECT c....	0 row(s) affected	0.034 sec