Spotify Data Analysis Using SQL

```
create table spotify (
Artist varchar(100),
Track varchar(100),
Album   varchar(100),
Album_type varchar(100),
Danceability float,
Energy  float ,
Loudness float,
Speechiness float,
Acousticness     float,
Instrumentalness float,
Liveness float ,
Valence float ,
Tempo float ,
Duration_min float ,
Title varchar(100),
Channel          varchar(100),
Views    float,
Likes bigint ,
Comments         bigint,
Licensed boolean ,
official_video boolean ,
Stream  bigint,
EnergyLiveness  float ,
most_playedon varchar(50)

)
```

```
Copy spotify
(
Artist,Track,Album,Album_type,Danceability,Energy,Loudness,Speechiness,Acousticness,Instrumentalness,Liveness,Valence,Tempo,Duration_min,Title,Channel,Views,Likes,Comments,Licensed,official_video,Stream,EnergyLiveness,most_playedon
)
from 'C:\Excel\spotify_data.csv'
DELIMITER','
CSV HEADER;
```

Easy

```
--Q.1 Retrieve the names of all tracks that have more than 1 billion streams.
Select * from spotify
where stream > 1000000000

--Q.2 List all albums along with their respective artists.
select artist , album
from spotify
group by 1,2

--Q.3 Get the total number of comments for tracks where licenced = True
Select
sum(comments) as total_comments
From spotify
where licensed = 'true'

--Q.4 Find all the tracks that belong to the album type single
  Select * from spotify
  where album_type = 'single'
```

--Q.5 Count the total number of tracks by each artists

```
Select artist ,
count(*) as total_no_songs
from spotify
group by artist
```

MEDIUM

--Q.1 Calculate the average danceability of tracks in each album

```
Select
album ,
avg(danceability) as average_dancebility
from spotify
group by 1
order by 2 desc
```

--Q.2 Find the top 5 tracks with the highest energy values.

```
Select
 track ,
max(energy )
from spotify
group by 1
Order by 2 desc
limit 5
```

```
--Q.3 List all the tracks along with their views and likes where official video = True
Select
track ,
sum(likes) as total_likes ,
sum(views) as total_views
from spotify
where official_video = 'true'
group by 1
order by 2 desc

--Q.4 For each album , calculate the total views of all associated tracks
select
album ,
sum(views) as total_views,
track
from spotify
group by 1,3
order by 2 desc

--Q.5 Retrieve the track names that have been streamed on spotify more than youtube
Select * from
(Select
track ,
COALESCE(sum(CASE when most_playedon = 'Youtube' then stream END),0)  as
streamed_on_youtube ,
COALESCE(sum(Case when most_playedon = 'Spotify' then stream end),0)  as streamed_on_spotify
from spotify
group by 1
)
as t1
where streamed_on_spotify > streamed_on_youtube and streamed_on_youtube <> 0
```

HARD

```
--Q.1 Find top 3 most viewed tracks for each artist using window functions
with ranking_artist
as
(
Select
artist ,track,
sum(views) as total_view,
Dense_rank() Over (partition by artist order by sum(views) desc) as rank
from spotify
group by 1,2
order by 1,3 desc
)
select * from ranking_artist
where rank <= 3
limit 9

--Q.2 Write a query to find tracks where the liveness score is above the average.

select
track ,
artist ,
liveness from spotify
where liveness > (Select avg(liveness)from spotify )
```

--Q.3 Use a with clause to calculate the difference between the highest and the lowest energy values for tracks in each album

```
with cte
as
(
Select
album,
max(energy) as highest_energy,
min(energy) as lowest_energy
from spotify
group by 1
)
select album ,
highest_energy - lowest_energy as energy_differece
from cte
order by 2 desc
```

--Q.4 find tracks where the energy-to-liveness ratio is greater than 1.2

```
Select track , energyliveness from spotify
where energyliveness > 1.2
order by track desc
```

--Q.5 Calculate the cumulative sum of likes for tracks ordered by the number of views using window function

```
select
track , artist , views , likes ,
sum(likes)  over(order by views desc) as cumulative_likes
from spotify
```