

prefix sy: <<http://www.dei.unipd.it/Database2/FRANGI/spotifyYoutubeStatistics/>>

## 1 - Top 10 songs by number of streams

```
select distinct (group_concat(distinct ?aName ; separator=", ") as ?aNames) ?sName ?streams
where {
    ?artist sy:published ?song ;
        sy:personName ?aName .
    ?song sy:trackName ?sName ;
        sy:stream ?streams .
}
group by ?sName ?streams
order by desc (?streams)
limit 10
```

## 2 - Song with max duration for each playlist, ordered by duration

```
select distinct ?pName ?sName ?duration where {
    ?song sy:isPartOf ?playlist ;
        sy:trackName ?sName ;
        sy:duration ?duration .
    ?playlist sy:playlistName ?pName .
    {
        select distinct ?pName1 (MAX(?duration1) as ?mDuration) where{
            ?song1 sy:isPartOf ?playlist1 ;
                sy:duration ?duration1 .
            ?playlist1 sy:playlistName ?pName1 .
        }
        group by ?pName1
    }
    filter (?pName = ?pName1 && ?duration = ?mDuration)
}
order by desc (?duration)
```

## 3 - How many artists published n songs

```
select (count(?artist) as ?nArtists) ?songs where{
    {
        select ?artist (count(?song) as ?songs) where{
            ?artist sy:published ?song .
        }
        group by ?artist
        order by desc(?songs)
    }
}
group by ?songs
```

## 4 - Artist that produced the most streams

```

select ?aName (sum(?stream) as ?streams) where{
    ?artist sy:personName ?aName ;
    sy:published ?song .
    ?song sy:stream ?stream.
}
group by ?aName
order by desc(?streams)
limit 1

```

## 5 - Top 10 channels by number of views, with also likes and comments

```

select distinct ?name (sum(?views) as ?totalViews) (sum(?likes) as ?totalLikes) (sum(?comments)
as ?totalComments) (sum(?stream) as ?totalStreams) where{
    ?video rdf:type sy:YoutubeVideo;
        sy:views ?views;
        sy:likes ?likes;
        sy:comments ?comments;
        sy:videoTitle ?title;
        sy:isUploadedBy ?channel;
        sy:isVideoOf ?song.
    ?channel sy:channelName ?name.
    ?song sy:stream ?stream.
}
group by ?name
order by desc(?totalViews)
limit 10

```

## 6 - Is the playlist with most songs the one with most streams?

```

ask where{
    {
        select ?playlist1 (count(?song) as ?songs) where{
            ?playlist1 rdf:type sy:SpotifyPlaylist;
            ^sy:isPartOf ?song.
        }
        group by ?playlist1
        order by desc(?songs)
        limit 1
    }.

    {
        select ?playlist2 (sum(?streams) as ?totalStreams) where{
            ?playlist2 rdf:type sy:SpotifyPlaylist;
            ^sy:isPartOf ?song.
            ?song sy:stream ?streams;
        }
        group by ?playlist2
    }
}

```

```

    order by desc(?totalStreams)
    limit 1
  }.

```

```

filter(?playlist1 = ?playlist2)
}

```

## 7 - Most common 5 genres among playlists

```

select ?genreName (count(?playlist) as ?numPlaylists) where {
  ?playlist sy:hasGenre ?genre.
  ?genre sy:genreName ?genreName.
}
group by ?genreName
order by desc(?numPlaylists)
limit 5

```

## 8 - Genre of top 5 playlists with most played songs

```

select ?pName (group_concat(distinct ?gName ; separator = ", ") as ?genres) ?streams where{
  {
    select ?pName ?gName (sum(?stream) as ?streams) where{
      ?playlist sy:playlistName ?pName ;
      sy:hasGenre ?genre .
      ?song sy:isPartOf ?playlist ;
      sy:stream ?stream .
      ?genre sy:genreName ?gName .
    }
    group by ?pName ?gName
    order by desc(?streams)
  }
}
group by ?pName ?streams
limit 5

```

## 9 - Random pair of songs that have similar tempo and key (useful for djs)

```

select ?inputTrackName ?inputKey ?inputTempo ?similarTrackName ?similarKey ?similarTempo
where {
  {
    # Get a random song
    select ?inputSong ?inputTrackName ?inputKey ?inputTempo
    where {
      ?inputSong a sy:SpotifySong ;
      sy:trackName ?inputTrackName ;
      sy:key ?inputKey ;
      sy:tempo ?inputTempo .
    }
    bind(rand() as ?randOrder1)
  }
}

```

```

    }
    order by ?randOrder1
    limit 1
}

# Get all the other songs but ordered randomly
{
    select ?similarSong ?similarTrackName ?similarKey ?similarTempo
    where {
        ?similarSong a sy:SpotifySong ;
        sy:trackName ?similarTrackName ;
        sy:key ?similarKey ;
        sy:tempo ?similarTempo .

        bind(rand() as ?randOrder2)
    }
    order by ?randOrder2

}

#Filters to get matches
filter (?inputSong != ?similarSong) # Ensure it's a different song (can happen that they are the
same but very rare (different id but name is the same just written differently))
# Threshold for key and tempo can be changed
filter (?inputKey = ?similarKey)
filter (abs(?inputTempo - ?similarTempo) < 5.0)
}
#Just keep one result (the first)
limit 1

```

## 10 - Average number of views of songs for the 3 types of album types (single, compilation and album)

```

select ?albumType (round(avg(?streams)*100)/100 as ?averageStreams) where {
    ?artist sy:published ?song ;
    sy:personName ?aName .
    ?song sy:trackName ?sName ;
    sy:stream ?streams .
    ?song sy:belongsTo ?album .
    ?album sy:albumType ?albumType .
}
group by ?albumType
order by desc(?averageStreams)

```

## 11 - Are number of streams of a song higher than the number of views of their related youtube video (in general)?

```

ask
where {
  {
    select (count(?youtubeVideo) as ?numVideosMoreViews)
    where {
      ?youtubeVideo sy:isVideoOf ?spotifySong .
      ?youtubeVideo sy:views ?views .
      ?spotifySong sy:stream ?streams .
      filter (?views > ?streams)
    }
  }
  {
    select (count(?youtubeVideo) as ?numVideosLessViews)
    where {
      ?youtubeVideo sy:isVideoOf ?spotifySong .
      ?youtubeVideo sy:views ?views .
      ?spotifySong sy:stream ?streams .
      filter (?views < ?streams)
    }
  }
  filter (?numVideosLessViews > ?numVideosMoreViews)
}

```

## 12 - Average ratios of views/likes and views/comments of videos that are the official video of the songs and the ones that are not

```

select ?avgRatioVLOfficial ?avgRatioVLUnofficial ?avgRatioVCOfficial ?avgRatioVCUnofficial
where {
  {
    select (round(avg(?viewsOfficial/?likesOfficial)*100)/100 as ?avgRatioVLOfficial)
    where {
      ?youtubeVideo sy:officialVideo true.
      ?youtubeVideo sy:views ?viewsOfficial.
      ?youtubeVideo sy:likes ?likesOfficial.
    }
  }
  {
    select (round(avg(?viewsUnofficial/?likesUnofficial)*100)/100 as ?avgRatioVLUnofficial)
    where {
      ?youtubeVideo sy:officialVideo false.
      ?youtubeVideo sy:views ?viewsUnofficial.
      ?youtubeVideo sy:likes ?likesUnofficial.
    }
  }

  {
    select (round(avg(?viewsOfficial/?commentsOfficial)*100)/100 as ?avgRatioVCOfficial)
    where {

```

```

    ?youtubeVideo sy:officialVideo true.
    ?youtubeVideo sy:views ?viewsOfficial.
    ?youtubeVideo sy:comments ?commentsOfficial.
  }
}
{
  select (round(avg(?viewsUnofficial/?commentsUnofficial)*100)/100 as ?avgRatioVCUnofficial)
  where {
    ?youtubeVideo sy:officialVideo false.
    ?youtubeVideo sy:views ?viewsUnofficial.
    ?youtubeVideo sy:comments ?commentsUnofficial.
  }
}
}

```

### 13 - Top 10 artists by number of albums published

```

select ?artistName (count(?album) as ?albums) where{
  {
    select distinct ?artistName ?album where {
      ?artist sy:published ?song ;
      sy:personName ?artistName .
      ?song sy:belongsTo ?album .
      ?album sy:albumType ?aType .
      filter (?aType = "album"^^xsd:string)
    }
  }
} group by ?artistName
order by desc(?albums)
limit 10

```

### 14 - Relation between YouTube video views and corresponding song

```

select ?songName ?stream ?videoName ?views (round((?stream/?views)*10000)/10000 as
?SVratio) where {
  ?video sy:isVideoOf ?song;
  sy:views ?views;
  sy:videoTitle ?videoName.
  ?song sy:stream ?stream;
  sy:trackName ?songName.
}
order by (abs(?SVratio-1))

```