Add client IP’s to firewall

1. Basics of TSQL. First half of PowerPoint.
2. Explain Database, Tables, Columns, Rows
3. Let’s get connected.
4. Azure Data Studio is a querying tool. There are many tools
5. Queries on dbo.flat
   1. Find the tracks with the most streams. Introduce order by
      1. SELECT track, artist, streams FROM flatdata ORDER BY streams desc
   2. Find the Top 10 track with the most streams. Introduce order by and TOP.
      1. SELECT TOP 10 track, artist, streams FROM flatdata ORDER BY streams desc
   3. Find songs by your favorite artist. Introduce WHERE.
      1. SELECT track, artist, streams FROM flatdata WHERE artist = 'lizzo'
   4. Find your favorite song or artists. Introduce some string manipulation.
      1. Wildcard searches
      2. SELECT track, artist, streams FROM flatdata WHERE track like '%September%'
      3. SELECT track, artist, streams FROM flatdata WHERE artist like 'Lil%'
   5. Find million stream songs. Introduce some numeric comparison
      1. SELECT track, artist, streams FROM flatdata WHERE streams > 1000000
   6. Find number of songs per artist. Introduce group by.
      1. SELECT artist, count(track) FROM flatdata GROUP BY artist
      2. Order it up
      3. SELECT artist, count(track) FROM flatdata GROUP BY artist ORDER BY count(track) desc
      4. Add an alias
      5. SELECT artist, Tracks = count(track) FROM flatdata GROUP BY artist ORDER BY count(track) desc
      6. Filter on multiples
      7. SELECT artist, Tracks = count(track) FROM flatdata GROUP BY artist HAVING count(track) > 1 ORDER BY count(track) desc
   7. Count isn’t the only aggregate we can use, let’s do sum.
      1. SELECT artist, Tracks = SUM(streams) FROM flatdata GROUP BY artist HAVING sum(streams) > 10000000 ORDER BY sum(streams) desc
   8. Put it all together. Find songs that share the first 8 characters in common.
      1. SELECT left(track,8), COUNT(track) FROM flatdata GROUP BY left(track,8) HAVING COUNT(track) > 1
   9. Who can find an top artist collaboration? Find an instance where an artist appears in the name?
      1. select r.artist, Track, x.artist

from flatdata r

inner join (select artist = '%'+artist+'%' from rawimport) x

on r.Track like x.artist

group by r.artist, Track, x.Artist

order by Track, x.Artist

* + 1. select \* from flatdata where track like '%post malone%'
    2. select \* from flatdata where track like '%meek mill%'
    3. select \* from flatdata where track like '%cardi b%'

1. Relational formatting creates multiple tables. Back to PowerPoint.
2. Queries on the relational tables. Notice how each table now has its own key. This key doesn’t change, even if the artist changes their name, we can still link them to all their songs.
   1. select \* from track
   2. select \* from artist
   3. select \* from chart
   4. Let’s start joining them together.
   5. select \* from track inner join artist on track.artistID = artist.ID
   6. select \* from chart   
      inner join track on chart.trackid = track.id   
      inner join artist on track.artistID = artist.ID
   7. But now we have all these extra columns so lets just pick the columns we want. Same data as we had before from the relational. But this serves us much better for eventually inserting data.
   8. select track.name, artist.name, chart.streams from chart

inner join track on chart.trackid = track.id

inner join artist on track.artistID = artist.ID

* 1. We can further clean it up with aliases.
  2. select t.name, a.name, c.streams

from chart AS c

inner join track AS t on c.trackid = t.id

inner join artist AS a on t.artistID = a.ID

* 1. The chart table has a rank to it.
  2. select t.name, a.name, c.streams

from chart AS c

inner join track AS t on c.trackid = t.id

inner join artist AS a on t.artistID = a.ID

Where c.rank = 1

* 1. Let’s add a new artist to our database.
  2. alter role db\_datawriter add member student
  3. First, let’s take a look at the table. It’s got an IDENTITY column, remember that column just counts up with each row. So we don’t need to provide that value.
  4. INSERT INTO artist (name)

SELECT 'Metallica'

* 1. Now, we need the id of the new artist in order to insert a track by metallica.
  2. SELECT \* FROM ARTIST where name = 'Metallica'
  3. Now, we’ll insert the new track name. We don’t have a URL for this one, so we can just insert a NULL value. A NULL is the absence of a value.
  4. INSERT INTO track (name, url, artistid)

SELECT 'Enter Sandman', NULL, 121

* 1. Now, we’ll add the song back to the charts! Let’s get the id value of the track we just added.

SELECT id from track where name = 'Enter Sandman'

* 1. Now, add a chart listing. Make up numbers of rank and the number of streams, whatever you want.

insert into dbo.chart (rank, streams, artistid, trackid)

Select 1000, 999999999, 121, 249

* 1. Now we can take a look at the data again, scrolling around to see everyone’s new tracks.

select t.name, a.name, c.streams

from chart AS c

inner join track AS t on c.trackid = t.id

inner join artist AS a on t.artistID = a.ID

1. Get started with [codecademy.com/learn/learn-sql](https://www.codecademy.com/learn/learn-sql)
2. Talk about jobs, consulting vs in-house