In this Lecture we will

- Learn to create a Relational Database and connect it to a ASP.NET Web Form (***ASPandRelationalDB***)
- Look at this Scenario
    - The "band" wants us to add a Favorite Songs page to the web site. They want it to show a fan's name, their favorite songs, a comment and if they are a member of the StreetTeam – the group of active fans that help promote the band. Easy enough since this information is all in the Fans table. Here's the hard part – they also want the Favorite Songs page to show the Album Name, Album Number and Track Number. But that information is in a different table; it's in the AlbumTracks table. So we have some work to do there..... That's where the concept of Relational Databases will come into play
    - **Fields in Fans Table**

      ID
      Password
      Name
      BirthDate
      Gender
      City
      State
      Email
      FavoriteSong***
      StreetTeam
      Comment

      Fields in AlbumTracks Table

      AlbumNumber
      AlbumName
      TrackNumber
      TrackName ***

    - If you look carefully at the data in these tables, you see that the information in the FavoriteSong field of the Fans table is the same as the information in the TrackName field of the AlbumTracks table. That is, both of these fields contain song titles. In other words, the data in the Fans table is connected to or "related" to the data in the AlbumTracks table through the fields FavoriteSong and TrackName. This is the essential element of a "relational database" – information in one table can be related to information in another table through fields that hold common information (in this case song titles). Let's say we wanted the name of the album (AlbumName) that contains a fan's favorite song. We would look for the name of the favorite song in the FavoriteSong field of the Fans table. Then we search for the same song in the TrackName field of the AlbumTracks table. Once

    we find the matching favorite song in the AlbumTracks table, we get the AlbumName for that particular record.
  - Here's another example. The Albums table contains a field called AlbumNumber. The AlbumTracks table also contains a field called AlbumNumber. Both fields contain an integer: 1 denoting the first album and 2 denoting the second album (and we hope there will be many more in the future). The information in both tables is related through their AlbumNumber fields. Using the AlbumNumber field in the AlbumTracks table, we can find the same AlbumNumber in the Albums table. Once we know which record has the same AlbumNumber, we can get other information from the Albums table such as Release Year and Label.
- First add a GridView and connect it to the BandDatabase
- When we configure the Select statement we need to select fields from the Fans and Album Tracks tables
  - Configure Select using custom SQL (Multi-Table Query-Query Builder) ... selecting fields from two tables
    - Add the Fans and AlbumTracks Tables to the Query Builder
    - Check the boxes next to the column names we want to display:

      Fans: Name, Favorite Song, Street Team, Comments

      Album Tracks: Album Number, Album Name, Track Number, TrackName

    - FavoriteSong field and TrackName are connecting fields
    - then click on FavoriteSong and drag toward TrackName (this creates relationship).This will match (or join) the two tables on those column names when their values are equal.
    - press the "Execute Query Button" ... and then "Test Query"
- Start your application. The grid is populated with data from the Fans table and the AlbumTracks table. Note that the same song is listed in the FavoriteSong field and the TrackName field.
  - clean up the grid ... eliminate redundant information (make TrackName invisible) and move AlbumNumber down once so that it is just before TrackNumber
  - Sort the records by AlbumNumber ***then*** by TrackNumber (need to get back into QueryBuilder)
    - Click SqlDataSource ... then go over to properties and choose SelectQuery
    - Click in the Sort Type cell in the AlbumNumber row and select Ascending. The Sort Order will be 1. Click in the Sort Type cell

      for the TrackNumber row and select Ascending. The Sort Order will be 2.