1. [Mostly Done, just need to check] [DEPENDS as needed] What should the “Blank String” be for the various Analytic fields, grab from this list when obvious:
   1. ‘None’
   2. ‘Unknown’
2. AnalyticsDimPersonHistorical, AnalyticsDimPersonCurrent, AnalyticsDimFamilyHistorical, AnalyticsDimFamilyCurrent are available to DataViews and Reports, but only the fields are part of the Model are currently available, even though the additional fields created from Attributes are in the SQL Tables. It would take some work to figure out how to deal with these Dynamic fields (Attributes)
   1. [ToDo] When you pick AttributeDimPerson for example, the field picker shows all the fields, including the attribute fields
   2. See <https://weblogs.asp.net/scottgu/dynamic-linq-part-1-using-the-linq-dynamic-query-library> for some possible ideas
3. When a new AttributeField is marked as IsAnalytic and IsAnalyticHistory, the next ETL will mark a bunch of records as History.
   1. For example
      1. Lets say that there are 100,000 people in the database, and 4000 of them have a FavoriteColor attribute that is NOT NULL
      2. Next, add FavoriteColor as IsAnalytic and IsAnalyticHistory
      3. The next time the Person ETL runs, it will create a history record those 4000 people
   2. If we don’t like that behavior, we’ll have to figure out what we want to do instead
      1. Give the new FavoriteColor value to all of the HistoryRecords for that person..
         1. [THINK ABOUT, it would be cool if we could, but not if it’s a mess]…factoring in LastModifiedDateTime
            1. If the LastModifiedDateTime was 1/1/2013, create a HistoryRecord for that Person for that Date, based on the PreviousHistory record of the person
4. Group Attribute Fields are determined by GroupType, so if there are 50 group types, there are up to 50 different sets of Attribute Fields. If we turn these into Normal fields (like AnalyticsDimPerson and AnalyticsDimFamily), it could be a challenge
   1. Options for AnalyticsDimGroup AttributeFields
      1. Option #1) Create a different AnalyticsDimGroup table for each GroupType
      2. Option #2) Gave a single AnalyticsDimGroup table, but put all the possible AttributeFields (the ones that marked as IsAnalytic ) as fields on the table
         1. Option #2a) Name the Fields {{ Attribute.Key }}. But have a potential issue of collision if different group type’s have the same Attribute.Key name, but different a FieldType
         2. [It seems like we were leaning towards this option]Option #2b) Name the Fields {{ GroupType.Name + ‘\_’ + Attribute.Key }} for example AnalyticsDimGroup.NeighborhoodGroup\_MeetingTime
         3. Option #2c) Call the base table AnalyticsSourceGroup, Name the fields Attribute\_{{ AttributeId }}, and have a bunch of Views named AnalyticsDimGroup{{ GroupType.Name }} with just the attributes that are specific to them
   2. NOTE: Family is a separate Dim table with simple Attribute FieldNames, so no matter what we do, GroupType.Family is already taken care of
5. Repackage PowerBI Template (prompt for connection string, etc), like CCVRockit
   1. Looked into this and
      1. it is a Zip file under the covers
      2. The connection string is stored as an encrypted thing ☹
      3. There is a REST Api that talks about creating a Template, but it seems limited. Maybe they’ll make it better?
6. [Todo] Tooltip on the Calendar UI to explain some things
7. [Not Done, but maybe PowerBI can do it?/Nice to Have] It would be great to have a new field to AnalyticsSourcePersonHistorical for AgeRange with the following groupings
   1. Under 12 years old
   2. 12-17 years old
   3. 18-24 years old
   4. 25-34 years old
   5. 35-44 years old
   6. 45-54 years old
   7. 55-64 years old
   8. 65-74 years old
   9. 75 years or older