1. [TODO] [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. [TODO]Primary Family.  The GetFamily of Person (<https://github.com/SparkDevNetwork/Rock/blob/3ee719f7b68d273bf8a5b7ed2baf28ad00055ce7/Rock/Model/Person.cs#L2668>), isn’t guaranteed to give a consistent result in the cases of multiple active families.  I propose we add an “ORDER BY ID” so that it will be consistent. That way it would always be the “NEWEST” family record first (that’s what CCV’s datamart tables do).  Another solution is to introduce a “PrimaryFamilyId” to Person so that Rock no longer has to guess which family is their primary family (it would also speed up a lot of stuff).  It could default to whichever family they were added to first, but would be editable (sort of like GivingGroup). It might take a bit of work to figure out how we want to do this since a person can be moved from families, families can become inactive, removed, etc.  So, maybe we could go with guessing like we do for now and figure this out later.
   1. [THIS WAY]
      1. Sort so that [IsAdult] (Order so that Adult then Child), then by [OLDEST] (order by Group.Id ASC), wins.  In other words, if they are in two families, and they are an adult in one and a child in another, choose the family where they are an Adult.  If they are the same in both, choose the oldest family.
3. [TODO] [YES] Do we want AnalyticsDimPersonHistorical and AnalyticsDimPersonCurrent to be available to DataViews and Reports? I assume yes, but it would take some work to figure out how to deal with the Dynamic fields (Attributes)
   1. [YES] 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. Random Notes
      1. Have a AttributeValuesXML column on AnalyticsDimPersonHistorical so that Attribute Values can be loaded efficiently from SQL without having to do a “One at a Time/Record By Record thingy”
      2. Have a “AnalyticsDynamicFields” Data Select that is smart enough to figure out how to create the Linq expression for SELECTs
      3. Have a “AnalyticsDynamicFields” Data Filter that is smart enough to figure out how to create the Linq expression for WHERE statements.
4. [DONE, but Jon will look at his book to verify a few things] DimDate UI
   1. [Look for Mockup]
   2. Giving Month
      1. [] Use Sunday Date
         1. Hardcode don’t show as option ([X] Don’t cross Fiscal Years)
   3. Fiscal Year has a Month Picker
5. Rock.Model.Attribute
   1. We had talked about adding an “IsAnalytic” as a field Rock.Model.Attribute,  but that would only be shown in the Attribute Config UI when the Model implement IAnalytic or IAnalyticHistorical.  I assume that would default to false for all existing Person/Family attributes? Or do we need to talk about that on a case by case (for the core ones) [YES, Jon will send me a list]
6. [TODO] Dim Family
   1. [YES] CampusId and CampusName
   2. ConnectionStatus of “Most Connected family member” (based on DefinedValue.Order where First is most connected)
   3. Name (Group.Name)
   4. FamilyTitle (getFamilyTitle)
   5. MailingAddress (first one, preference on Home address)
      1. Flattened out fields (AnalyticsDimFamily.Street1, Street1, etc)
      2. Lat
      3. Long
      4. LocationId
      5. GeoPoint (Spatial)
   6. MappedAddress (first one, preference on Home address)
      1. Same fields as MailingAddress
   7. IsFamilyActive (Is at least one FamilyMember Recordstatus “Active” (Core DefinedValue)\_
   8. Adult Count
   9. Child Count
   10. HeadOfHouseholdPersonId
       1. Which can be linked to new View off of DimCurrentPerson
          1. \*DimHeadOfHouseHold\*
   11. IsEra (Core ERA if any of the FamilyMembers is ERA)
   12. [Can Wait] Attributes (Just like Person)

* When a new AttributeField is marked as IsAnalytic and IsAnalyticHistory, the next ETL will mark a bunch of records as History. 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. Option #1) The next time the Person ETL runs, it will create a history record those 4000 people
  4. 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
     2. Skip the “Mark As History” if the Field was just added in that ETL session?
     3. Factor in LastModifiedDateTime on the AttributeValue??

1. [LATER] 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), 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. 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: It might make sense to keep Family a separate Dim table with simple Attribute FieldNames (also we wouldn’t have the situation of tracking of when a Person moved from a FamilyGroup to a NeighborhoodGroup, but for other grouptypes you might, such as “UsherTeamGroup” to “PrayerTeamGroup”)
2. [TODO] AnalyticsDimFamily as its own thing with Attributes
3. [OK] CCV Priorities vs core discussion
   1. CCV needs the following with a “Begin Testing” date of 12/26/2016, Production 1/3/2017, and an expectation that I would be coding on 12/16/2016
      1. Attendance
      2. Financial
      3. Access to Person Fields and Attributes as real fields in PowerBI
      4. Person History, nice but not a priority
      5. Groups. Enough for Attendance BI to make sense.
      6. Group Attributes. Not a priority
4. Attendance
   1. AnalyticsDimAttendanceLocation (View)
      1. (Most of the Location fields)
5. Attendance And FinancialTransaction
   1. Don’t do AnalyticsDimAttendanceCampus. Just put CampusName, CampusId, CampusShortCode on the AnalyticsFact/SourceAttendance Table
   2. Try to get rid of PersonAliasId and PersonId’s
      1. CurrentPersonKey (current)
      2. PersonKey (at the time of the Transaction or Attendance)