数据建模DAX语言（新建度量值、新建列）

- 函数
  - ALL系列函数可以引用多列，并且返回的结果模表。
    - ALL系列函数全部在度量值中是不能独立使用的，需要配合CALCULATE和COUNTROWS这些可以引用表的函数使用
    - CALCULATE(度量值, <筛选条件1>, <筛选条件1>……)
      - 度量值=calculate([度量], 产品表[咖啡种类]="拿铁")
        - 正确的写法=calculate([度量] 产品表[咖啡种类]="拿铁")
        - 在调制多条件筛选时，可以使用逗号表示同时满足情况。calculate([度量], 产品表[咖啡种类]="拿铁", "美式", "卡布奇诺")
        - 也可以表达不包含关系。度量值=calculate([销售额],NOT产品表[咖啡种类]="拿铁", "美式", "卡布奇诺", 以 意思为：客户选择不包含拿铁、美式、卡布奇诺咖啡的销售额
        - 产品意义：为引用表不能只引用列多列，并且返回的结果模表。
      - ALL筛选器函数
        - 功能为解除（排除）某个筛选条件对"大范围
          - ALL销售量=calculate([销售量],ALL('销售额九表'[ ])
          - all是在下字数据表中清除了所有筛选条件的影响
            - 注意all函数就清除的筛选列和初始筛选条件中的筛选列完全一致（即同一张表的同一列）
          - 还可以按照列来清除筛选影响：ALL销量=calculate（[销量], ALL(产品表[咖啡种类]) 意思为：客产品表中咖啡咖啡种类的筛选条件的影响之外的销量
      - ALLEXCEPT
        - 意思为除了指定列外清除筛选影响，清除其他不在引用的列
          - ALLEXCEPT=CALCULATE([销量], ALLEXCEPT('产品表'[所型]) 意思为：客户产品表中所有的咖啡（大、中、小）只受所型筛选影响，清除种型之外的条件影响。
          - ALL系列函数是可以引用多列，并且返回的结果模表。
            - all系列函数全部在度量值中是不能独立使用的，需要配合CALCULATE和COUNTROWS这些可以引用表的函数使用
      - ALLSELECTED
        - 对数据上放次的筛选条件进行清除筛选（即切片器），而其他筛选条件保留，进行数据统计
          - ALLSELECTED=CALCULATE([销量], ALLSELECTED('产品表'[ ]) 合计=sum([销量])/ALLSELECTED() 这段语句以保证合计是与切片器筛选条件下的数据合计占比，保证每次占比都是100%。
      - FILTER函数：高级筛选函数
        - filter是最强大的筛选器，不是计算函数，返回是一张表，所以它是表格函数，常与CALCULATE函数搭配，也可与聚合函数搭配如：COUNTROWS(FILTER(表, 筛选条件))计算相关的行数
          - FILTER(表, 筛选条件)
            - FILTER的筛选条件为判断逻辑 由于FILTER函数会对筛选的表进行逐行遍历以检索出结果，如果与函数一样（SUM）与其它函数主要区别是，在工作中可以表达所写第一行，后面这个工作时间是行上下文。
            - 其中常可以是任意一张表，包括ALL、VALUES等函数返回的表格型数据作为FILTER函数返回结果
              - 例如：FILTER(FILTER('订单明细', [单价]>50&&[数量]>10),[数量]>50)
              - 第二部分为筛选条件：FILTER(销量=CALCULATE([销量]), FILTER('产品表', '产品表'[咖啡种类]="拿铁"&&"产品表"[所型]="中"))
              - 解析：
                - 在例如：A的为表格数据=CALCULATE([销售额], FILTER(VALUES('销售记录'[产品名称]), [累计数量]>=0.7))
                - 解析：在销售记录表中求销量累计数占比小于等于0.7的A表格数据
                - FILTER2使用中注意事项：
                  1、尽量在LOOKUP查询使用它的数据量边小于事实表（数据表），避免进行扫描带来的计算机进行压力
                  2、当能使用CALCULATE筛选解决时，不要用FILTER,慎不要用FILTER函数在CALCULATE+FILTER函数
            - 表格算法行与列构成，所以上下文也有两种：1、横向以下文 2、行上下文
              - 筛选上下文：指的就是反复遍历的"筛选条件"这个词比如CALCULATE([销量],'产品表'[咖啡种类]="拿铁", '产品表'[所型]="中")
                - 它的筛选上下文就是针对产品表中的"咖啡种类列"，"所型列"以及等号，"中"为条件筛选出来的表
                - 行上下文=当前行，当我们在遍历的第一个单元格时输入数值时，它的所在的单元格数值等于当前行
                - 例如在新建列输入2+3表格，所有的这列单元格数值是5，并不会在这一行的范围上下文
                - 行上下文不会自动转为筛选上下文：当前所在行不会转成筛选数据的信号作为数据查询结果中，也就是说对于数据数据条件没有进行任何筛选，所有数据的原始数据都一样。
                - CALCULATE能当于整个表格数据的计算范围呢，当加入CALCULATE时，系统就会启动，所有的信号都消失了，这个表格信号等行上下文转换为筛选上下文，能增加当前行中的筛选条件时候数据数据进行筛选（如果点击不明白）
                - 1、在CALCULATE(销量)、FILTER(门店信息表, SUM(销售明细表'[数量]>200) 中SUM会符合计数量总计传递到中每一行，行上下文不会变成筛选上下文（也就是说对销售数据表没有任何筛选，所以是计算销售量超过200）
                - 2、导致的结果是FILTER在进行逐行扫描时，每一行都是总计大于200
                - 3.通过复合的度量值导致的CALCULATE函数把行上下文转换成筛选上下文，而在FILTER函数里SUM受CALCULATE函数引导不会再上下文转化，只会在类似矩阵表中添加的初始筛选上下文条件下运算
      - FILTER2函数：安全除法
        - DIVIDE(<numerator>, <denominator>[,<alternateresult>])
          - 术语
          - 定义
          - numerator
          - 被除数，即被除的数字。
          - denominator
          - 除数，即除以的数字。
          - 备用结果
          - （可选）若零除而导致错误时返回的值。 如果没有提供，则默认值为 BLANK()。
      - IF/SWITCH函数：逻辑判断
        - SWITCH语句语句
          - 如果是某种数据类型语句例如：switch ('日历表'[月], 1, "1月", 2"2月", 3"3月", 4, "4月", 未识别月)
          - SWITCH+TRUE函数用于逻辑判断：SWITCH(TRUE([数]>208&[年数]>=30,"21-30",>30,"30以上")
      - 关系函数：RELATED RELATEDTABLE LOOKUPVALUE
        - RELATED是一个特别的函数，它是专为关系型建立的，所以它不同于上下文只关心表格的结果问题
          - RELATED也是语法之一，在入个数量为"RALATED('产品表'[价格])
        - RELATEDTABLE
          - 是一项函数能引用多条数据，当用我们需要对多项的多条数据进行聚合（订单数量=COUNTROWS(RELATEDTABLE ('销售数据表') ）
        - LOOKUPVALUE函数中的条件可以不断增加，精准定位搜索信息，不需要像RELATED使用表格作为基础，相同的字段条件在一起时，主要也是三个部分。
          - 1、输入所在单元格的字段名称，即是中"产品表中的"价格"的列。
          - 2、输入所在单元格要搜索的数值
          - 3、输入表格中需要搜索的关联字段
      - TIME INTELLIGENCE函数：时间智能函数（以2012年1月1日到2014年12月31日）
        - DATESYTD
          - 语法与语法：
          - DATESYTD(日历表'[日期]
          - 说明如果2014年1月1日到2014年12月31日
          - 注意all函数就清除的筛选和初始筛选条件的影响
          - 注意给是时间段函数和时间点函数返回的都是一个
            - DATEADD函数：按照固定的间隔加减一个时间区间
              - DATEADD(日历表'[日期], -1, YEAR)意为向前，还可以用 日 为最近区，注意本本。
            - Datesbetween函数用于固定开始和结束日期之间的时间段
              - 例如：Datesbetween('日历表'[日期],"2015-01-01",MAX('日历表'[日期]) 说明如是2015年1月1日到2014年9月1日的时间段
            - Dateslnperiod函数可以限制某一个时间点来调整时间区间。如果以Dateslnperiod('日历表'[日期], "2015-01-01",1, month) 将以2015年1月1日为起点，向后数一个月，得到2015年1月1日到2015年2月1日的时间段。
              - DATESINPERIOD('日历表'[日期], 开始日期, 间隔,间隔类型)
              - DATESINPERIOD
                - 30天移动平均：calculate([销量], dateslnperiod ('日历表'[日期], 日期序列, MAX ('日历表'[日期]) , -30,day)/30
            - 注意all是时间段函数和时间点函数返回的都是表，要配合CALCULATE函数把他们能为CALCULATE的筛选条件
          - DATESQTD函数返回的是本季度累计
            - DATESMTD函数返回的是本月累计
          - 时间点函数
            - 用于指定某一个特定日期，他们返回的是一个包含一值表。
            - 这个值的是某一个日期，比如FIRSTDATA用于求最年轻的日期
            - LASTDATA用于求最晚的日期
            - 不同于min 和 max只是时间列求最小和最大
              - FIRSTDATA和LASTDATA函数可以用表，比如我们开公司成立之日日期到现在的累计值，则需要找到最开始的日期FIRSTDATA(ALL('日历表'[日期]))
              - END函数返回的是最后一天，比如ENDOFMONTH('日历表'[日期]) 则是，日期序，MAX ('日历表'[日期]) , -30,day)/30
            - SAMEPERIODLASTYEAR
              - SAMEPERIODLASTYEAR函数的语文意思是上年同期，
              - SAMEPERIODLASTYEAR('日历表'[日期]) =
              - =DATEADD('日历表'[日期], -1, YEAR)
      - 日历表的使用
        - 对日历表进行排序，比如需要按产品中"添加日期时"功能添加一个"DAYOFWEEK"（每周的第一日）列，然后关闭按照列DAYOFWEEK排序即可
          - 子主题 1
        - 对于特殊节日的表现方式
          - 对于非常规日历，方金品记录，上月销量=CALCULATE([销售量], FILTER(ALL('时间日历表'),'时间日历表'[日]=MAX('时间日历表'[日])
        - 非常规日期对比
          -
            - 创建两个日历表进行相对值对比
      - DIVIDE函数：安全除法
        - DIVIDE(<numerator>, <denominator> [,<alternateresult>])
          - 术语
          - 定义
          - numerator
          - 被除数，即被除的数字。
          - denominator
          - 除数，即除以的数字。
          - 备用结果
          - （可选）若零除而导致错误时返回的值。 如果没有提供，则默认值为 BLANK()。
      - IF/SWITCH函数：逻辑判断
        - SWITCH语句语句
          - 如果是某种数据类型语句例如：switch ('日历表'[月], 1, "1月", 2"2月", 3"3月", 4, "4月", 未识别月)
        - SWITCH+TRUE函数用于逻辑判断：SWITCH(TRUE([数]>208&[年数]>=30,"21-30",>30,"30以上")
      - 关系函数：RELATED RELATEDTABLE LOOKUPVALUE
        - RELATED是一个特别的函数，它是专为关系型建立的，所以它不同于上下文只关心表格的结果问题
          - RELATED也是语法之一，在入个数量为"RALATED('产品表'[价格])
        - RELATEDTABLE
          - 是一项函数能引用多条数据，当用我们需要对多项的多条数据进行聚合（订单数量=COUNTROWS(RELATEDTABLE ('销售数据表') ）
        - LOOKUPVALUE函数中的条件可以不断增加，精准定位搜索信息，不需要像RELATED使用表格作为基础，相同的字段条件在一起时，主要也是三个部分。
          - 1、输入所在单元格的字段名称，即是中"产品表中的"价格"的列。
          - 2、输入所在单元格要搜索的数值
          - 3、输入表格中需要搜索的关联字段
      - TIME INTELLIGENCE函数：时间智能函数（以2012年1月1日到2014年12月31日）
        - DATESYTD
          - 语法与语法：
          - DATESYTD(日历表'[日期]
          - 说明如果2014年1月1日到2014年12月31日
          - 注意all函数就清除的筛选和初始筛选条件的影响
          - 注意给是时间段函数和时间点函数返回的都是表，要配合CALCULATE函数把他们能为CALCULATE的筛选条件
            - DATEADD函数：按照固定的间隔加减一个时间区间
              - DATEADD(日历表'[日期], -1, YEAR)意为向前，还可以用 日 为最近区，注意本本。
            - Datesbetween函数用于固定开始和结束日期之间的时间段
              - 例如：Datesbetween('日历表'[日期],"2015-01-01",MAX('日历表'[日期]) 说明如是2015年1月1日到2014年9月1日的时间段
            - Dateslnperiod函数可以限制某一个时间点来调整时间区间。如果以Dateslnperiod('日历表'[日期], "2015-01-01",1, month) 将以2015年1月1日为起点，向后数一个月，得到2015年1月1日到2015年2月1日的时间段。
              - DATESINPERIOD('日历表'[日期], 开始日期, 间隔,间隔类型)
              - DATESINPERIOD
                - 30天移动平均：calculate([销量], dateslnperiod ('日历表'[日期], 日期序列, MAX ('日历表'[日期]) , -30,day)/30
            - 注意all是时间段函数和时间点函数返回的都是表，要配合CALCULATE函数把他们能为CALCULATE的筛选条件
          - DATESQTD函数返回的是本季度累计
            - DATESMTD函数返回的是本月累计
          - 时间点函数
            - 用于指定某一个特定日期，他们返回的是一个包含一值表。
            - 这个值的是某一个日期，比如FIRSTDATA用于求最年轻的日期
            - LASTDATA用于求最晚的日期
            - 不同于min 和 max只是时间列求最小和最大
              - FIRSTDATA和LASTDATA函数可以用表，比如我们开公司成立之日日期到现在的累计值，则需要找到最开始的日期FIRSTDATA(ALL('日历表'[日期]))
              - END函数返回的是最后一天，比如ENDOFMONTH('日历表'[日期]) 则是，日期序，MAX ('日历表'[日期]) , -30,day)/30
            - SAMEPERIODLASTYEAR
              - SAMEPERIODLASTYEAR函数的语文意思是上年同期，
              - SAMEPERIODLASTYEAR('日历表'[日期]) =
              - =DATEADD('日历表'[日期], -1, YEAR)
      - 日历表的使用
        - 对日历表进行排序，比如需要按产品中"添加日期时"功能添加一个"DAYOFWEEK"（每周的第一日）列，然后关闭按照列DAYOFWEEK排序即可
          - 子主题 1
        - 对于特殊节日的表现方式
          - 对于非常规日历，方金品记录，上月销量=CALCULATE([销售量], FILTER(ALL('时间日历表'),'时间日历表'[日]=MAX('时间日历表'[日])
        - 非常规日期对比
          -
            - 创建两个日历表进行相对值对比
  - 函数
    - VALUES函数：不重复值
      - 说白由一列构成的一个表，该表包含来自指定表或列的非重复值。
      - 用VALUES达到同样效果 CALCULATE([销量],ALLEXCEPT('产品表'[所型]))
      - 原理是先用ALL函数删除产品表中所有筛选条件，在选择性地保留"产品表中的"所型"列数据
      - ALLEXCEPT=CALCULATE([销量],ALLEXCEPT('产品表'[所型]))
      - 此处的意思是：
      - 用VALUES达到同样效果 CALCULATE([销量],ALLEXCEPT('产品表'[所型]))
      - 原理是先用ALL函数删除产品表中所有筛选条件，在选择性地保留"产品表中的"所型"列数据
    - 关系函数及关系DAX公式
      - USERELATIONSHIP函数（"销售记录"[实际送货日期]=多维表[日历表'[日期]]——强关系->）
        - 可以使用CALCULATE和CALCULATE(SUM('销售记录'[销量]),USERELATIONSHIP('销售记录'[实际送货日期], '日历表'[日期])) 来调用第2张表中不同关系日期的产品销量。
      - HASONEVALUE函数是判断表"有"或"没"，即判断的是是否有一个值，可以把它的表达式用来替代COUNTROWS()
        - VALUES(列名称)=1，也就是说某列的不重复的数计算行数是否等于1
      - RASONEVALUE函数：只取一个值
        -
          - 第一种现在方法是找到度量值在字段进行筛选筛选
      - RANKX函数的数据排序度量
        - RANKX是上下文函数，会计整张表进行计算
        - 例如：RANKX(ALL('门店信息'),[销售量])
          - 意思：按照整张门店信息表，进行销量排名
        - RANKX(<table>, <expression>[, <value>[, <order>[, <rtie>]])
        - RANKX函数：1.table函数，2.expression是要求表达式，3.VALUE是以空值 4.order是以为小排序顺序，5为升序，6.tie为是排名方式
      - TOPN函数及其他N开头的聚集CALCULATE函数或者其他计算函数的合并用
        - TOPN(<n_value>, <table>, <orderby_expression>, [<order>[, <orderby_expression>, [<order>]]...])
        - TOPN函数1.N_value是收取N位，2.table是要筛选的表 3.orderby_expression是以它那个比重量表来筛选，4.order是升可选，0降序，1升序。
        - TOPN=CALCULATE([销量],TOPN(5,ALL('门店信息表'),[销量]),销量)，按照门店信息表筛选一张前5名销量的表
        - 前百名的销量占比：
        - DIVIDE(CALCULATE([销量],TOPN(5,ALL('门店信息表'),[销量]), CALCULATE([销量],ALL('门店信息表')))

试用版 XMind:ZEN 试用版