小型电讯公司星型模型、雪花模型设计

141250XXX XXX

2017年2月24日

目录

[一、星型模型 2](#_Toc473759397)

[模型图 2](#_Toc473759398)

[映射关系 2](#_Toc473759399)

[二、雪花模型 5](#_Toc473759400)

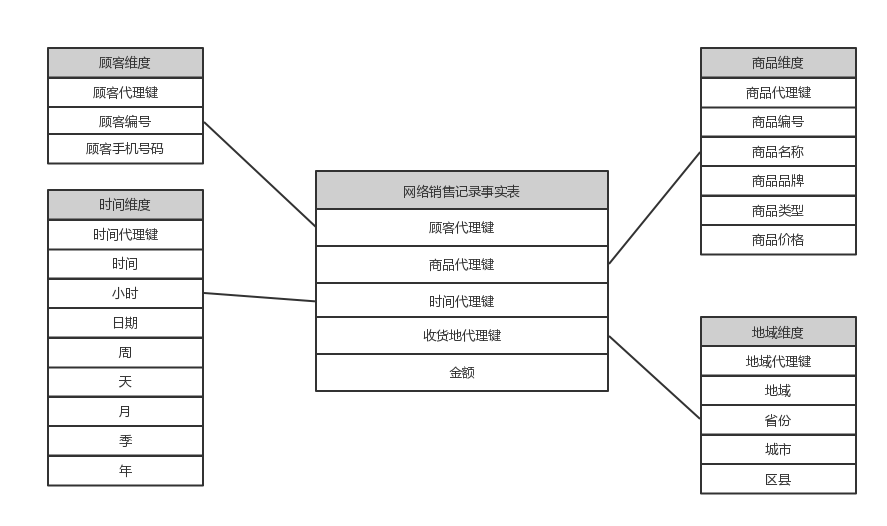
[模型图 5](#_Toc473759401)

[映射关系 5](#_Toc473759402)

[三、作业说明 7](#_Toc473759403)

## 一、星型模型

### 模型图



### 映射关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **数据映射来源** | **数据来源格式** | **文件名或者表名** |
| **顾客维度表**  **customer\_dim** | 顾客代理键  customer\_sk | / | / | / |
| 顾客编号  customer\_id | 源数据库customer表中的 customer\_id字段 | 源数据库的customer表中数据 | 源数据库的customer表 |
| 顾客手机号码  customer\_mobile | 源数据库customer表中的customer\_mobile字段 |
| **商品维度表**  **product\_dim** | 商品代理键  product\_sk | / | / | / |
| 商品编号  product\_id | 源数据库product表中的 product\_id字段 | 源数据库的product表中数据导出的csv格式 | product.csv |
| 商品名称  product\_name | 源数据库product表中的 product\_name字段 |
| 商品品牌编号  product\_brand\_id | 源数据库brand表中的 brand\_id字段 |
| 商品品牌名称  product\_brand\_name | 源数据库brand表中的 brand\_name字段 |
| 商品类型  product\_type | 源数据库product表中的 product\_type字段 |
| 商品价格  product\_cost | 源数据库product表中的 product\_cost字段 |
| **时间维度表**  **time\_dim** | 时间代理键  time\_sk | / | / | / |
| 时间  time | 源数据库sale表中的 sale\_time字段 | 源数据库的sale表中数据 | 源数据库的sale表 |
| 小时  hour | 对time字段的值应用HOUR()函数得到 | / | / |
| 日期  day | 对time字段的值应用Date()函数得到 |
| 周（一周中的第几天）  day\_of\_week | 对time字段的值应用DAYOFWEEK()函数得到 |
| 天（一月中的第几天）  day\_of\_month | 对time字段的值应用DAYOFMONTH()函数得到 |
| 月  month | 对time字段的值应用MONTH()函数得到 |
| 季  quarter | 对time字段的值应用QUARTER()函数得到 |
| 年  year | 对time字段的值应用YEAR()函数得到 |
| **地域维度表**  **area\_dim** | 地域代理键  area\_sk | / | / | / |
| 地域  area | 源数据库sale表中的 sale\_receiver\_address字段 | 源数据库的sale表数据导出的txt格式 | address.txt |
| 省份  province | 对area字段的值应用自定义函数得到 | / | / |
| 城市  city | 对area字段的值应用自定义函数得到 |
| 区县  district | 对area字段的值应用自定义函数得到 |
| **多种不同粒度的事实表** | | | | |
| **网络销售记录事实表**  **sale\_fact** | 顾客代理键  customer\_sk | 本数据库customer\_dim表中的 customer\_sk字段 | 本数据库的customer\_dim表中数据 | 本数据库的customer\_dim表 |
| 商品代理键  product\_sk | 本数据库product\_dim表中的 product\_sk字段 | 本数据库的product\_dim表中数据 | 本数据库的product\_dim表 |
| 时间代理键  time\_sk | 本数据库time\_dim表中的 time\_sk字段 | 本数据库的time\_dim表中数据 | 本数据库的time\_dim表 |
| 收货地代理键  area\_sk | 本数据库area\_dim表中的 area\_sk字段 | 本数据库的area\_dim表中数据 | 本数据库的area\_dim表 |
| 金额  sale\_money | 源数据库sale表中的 sale\_money字段 | 源数据库的sale表中数据 | 源数据库的sale表 |
| **以具体“商品品牌”为商品属性的统计单位的综合数据事实表1**  **sale\_fact\_1** | 商品品牌编号  product\_brand\_id | 本数据库product\_dim表中的 product\_brand\_id字段 | 本数据库的product\_dim表中数据 | 本数据库的product\_dim表 |
| 商品品牌名称  product\_brand\_name | 本数据库product\_dim表中的product\_brand\_name字段 |
| 该品牌总销售额  total\_sale\_money | 对本数据库sale\_fact表sale\_money字段的值应用函数得到 | / | / |
| **以具体“每周第几天”为时间属性的统计单位的综合数据事实表2**  **sale\_fact\_2** | 每周第几天  day\_of\_week | 本数据库time\_dim表中的day\_of\_week字段 | 本数据库的time\_dim表中数据 | 本数据库的time\_dim表 |
| 该时段总销售额  total\_sale\_money | 对本数据库sale\_fact表sale\_money字段的值应用函数得到 | / | / |
| **以具体“城市”为地域属性的统计单位的综合数据事实表3**  **sale\_fact\_3** | 省份  province | 本数据库area\_dim表中的province字段 | 本数据库的area\_dim表中数据 | 本数据库的area\_dim表 |
| 城市  city | 本数据库area\_dim表中的city字段 |
| 该地域总销售额  total\_sale\_money | 对本数据库sale\_fact表sale\_money字段的值应用函数得到 | / | / |

## 二、雪花模型

### 模型图



### 映射关系

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | **字段名** | **数据映射来源** | **数据来源格式** | **文件名或者表名** |
| **客户维度表**  **client\_dim** | 客户编号  client\_id | 源数据库client表中的 client\_id字段 | 源数据库的client表中数据 | 源数据库的client表 |
| 客户姓名  client\_name | 源数据库client表中的client\_name字段 |
| **地域维度表**  **area\_dim** | 地域编号  area\_id | 源数据库city表中的 city\_id字段 | 源数据库的city表中数据导出的txt格式 | city.txt |
| 地域名称  area\_name | 源数据库city表中的 city\_name字段 |
| 省份  province | 对area\_name字段的值应用自定义函数得到 |
| 城市  city | 对area\_name字段的值应用自定义函数得到 |
| **手机卡维度表**  **card\_dim** | 手机卡代理键  card\_sk | / | / | / |
| 手机号码  card\_num | 源数据库card表中的 card\_num字段 | 源数据库的card表中数据 | 源数据库的card表 |
| 所属客户编号  card\_client\_id | 源数据库card表中的 card\_client\_id字段 |
| 所属城市编号  cars\_area\_id | 源数据库card表中的 card\_city\_id字段 |
| **业务/套餐维度表**  **sp\_dim** | 业务/套餐代理键  sp\_sk | / | / | / |
| 业务/套餐编号  sp\_id | 源数据库sp表中的sp\_id字段 | 源数据库的sp表中数据导出的txt格式 | sp.txt |
| 业务/套餐类型  sp\_type | 源数据库sp表中的 sp\_type字段 |
| 业务/套餐计费标准  sp\_cost | 源数据库sp表中的 sp\_cost字段 |
| **时间维度表**  **time\_dim** | 时间代理键  time\_sk | / | / | / |
| 时间  time | 源数据库bill\_cardsp表中的 bill\_time字段 | 源数据库的bill\_cardsp表中数据 | 源数据库的bill\_cardsp表 |
| 日期  day | 对time字段的值应用Date()函数得到 |  |  |
| 月  month | 对time字段的值应用MONTH()函数得到 |
| 季  quarter | 对time字段的值应用QUARTER()函数得到 |
| 年  year | 对time字段的值应用YEAR()函数得到 |
| **多种不同粒度的事实表** | | | | |
| **手机卡业务/套餐账单事实表**  **bill\_sp\_fact** | 手机卡代理键  card\_sk | 本数据库card \_dim表中的 card \_sk字段 | 本数据库的card \_dim表中数据 | 本数据库的card \_dim表 |
| 业务/套餐代理键  sp\_sk | 本数据库sp\_dim表中的 sp\_sk字段 | 本数据库的sp\_dim表中数据 | 本数据库的sp\_dim表 |
| 时间代理键  time\_sk | 本数据库time\_dim表中的 time\_sk字段 | 本数据库的time\_dim表中数据 | 本数据库的time\_dim表 |
| 金额  bill\_money | 源数据库bill\_cardsp表中的 bill\_fee字段 | 源数据库的bill\_cardsp表中数据 | 源数据库的bill\_cardsp表 |
| **以具体“业务/套餐月消费额”为业务/套餐属性的统计单位的综合数据事实表1**  **bill\_sp\_fact\_1** | 业务/套餐月消费额  sp\_cost | 本数据库sp\_dim表中的sp\_cost字段 | 本数据库的sp\_dim表中数据 | 本数据库的sp\_dim表 |
| 该业务/套餐总账单额  total\_bill\_money | 对本数据库bill\_sp\_fact表bill\_money字段的值应用函数得到 | / | / |
| **以具体“月份”为时间属性的统计单位的综合数据事实表2**  **bill\_sp\_fact\_2** | 月份  month | 本数据库time\_dim表中的month字段 | 本数据库的time\_dim表中数据 | 本数据库的time\_dim表 |
| 该时段总账单额  total\_bill\_money | 对本数据库bill\_sp\_fact表bill\_money字段的值应用函数得到 | / | / |

## 三、作业说明

① 考虑到实际情况中数据仓库的源数据来源广泛，不仅仅限于Mysql数据库中的数据，所以增加了txt文件和csv这两种便于导入Mysql数据库的文件格式作为其他常见的数据来源。但同时，这种数据来源中的数据应该是较固定的、不经常变化的、且数据量不随着时间变化，所以选择了业务/套餐sp、商品product、城市city作为这些数据来源的数据。实际商业中还可能有xls、doc等数据来源格式，这些格式均可以通过txt文件进行转换后导入。本次作业由于时间的限制，没有考虑数据源是其他类型数据库这种较为复杂的情况。

② 数据仓库中维度表的属性和对应数据库中表的属性并不是完全一致的，只选取了那些有实际分析意义的属性列重新载入，同时在某些维度表中额外增加了一些属性，便于以后的数据定期载入等。

如，源数据库中product表的属性列：



对应星型模型中product\_dim表的属性列：



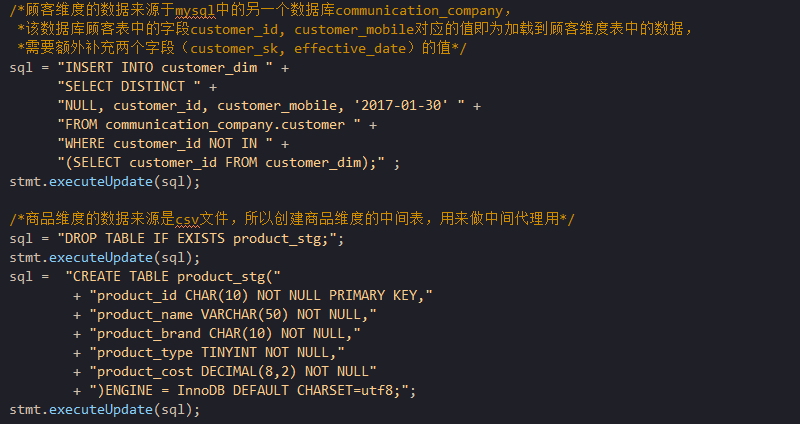
可以看出，数据仓库中的product\_dim删除了product\_stock\_num一列，因为它的值变化频率较快，而其他列的值相对来说较为稳定，如果数据仓库中仍保留该字段会导致每次刷新时要更新的内容非常多，并且该字段的值在数据仓库中没有分析意义，所以不再保留。

其次，product\_dim将product\_brand拆为了product\_brand\_id和product\_brand\_name两列，当分析人员进行数据查看时，如果仅仅查看编号，也许不能理解它的含义，加了名称后，分析人员能明确的知道编号对应现实中的意义，增加了数据的可理解性。

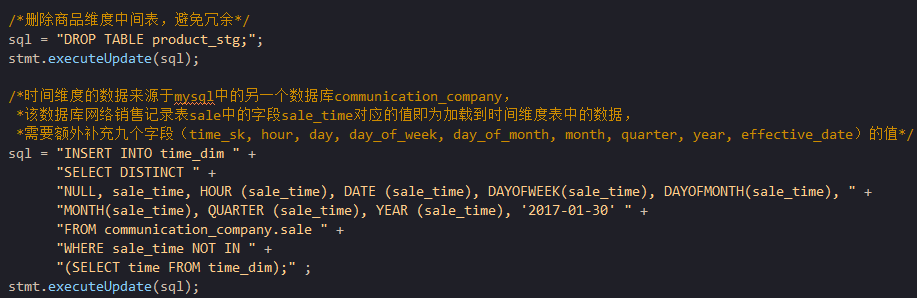
最后，product\_dim还增加了product\_sk和effective\_date两个字段，其中product\_sk作为代理键存在，effective\_date是数据被载入数据仓库生效的时间，在日后的数据定期载入中可以便于数据刷新和历史保留。

③ 作为一个分析应用，数据仓库应该提供各种粒度的事实表来供分析人员使用，因此，我在每一个模型中都设置了不同粒度的事实表（具体请见“映射关系”中的表格），这个作业中的数据量并不大，因此这些不同粒度的表格看上去没有什么意义。但是在现实中，对于大规模数据，根据不同维度的不同属性提取不同的综合数据并存储，可以缓解每一次分析都要重新查看数据造成的负担。

④ETL过程中需要用到一些表格作为中间表格存在，同时也用到了一些自定义的函数，具体请见工程文件中的DataETL中的代码。一些代码如下：









参考资料： Dimensional Data Warehousing with MySQL : A Tutorial by Djoni Darmawikarta