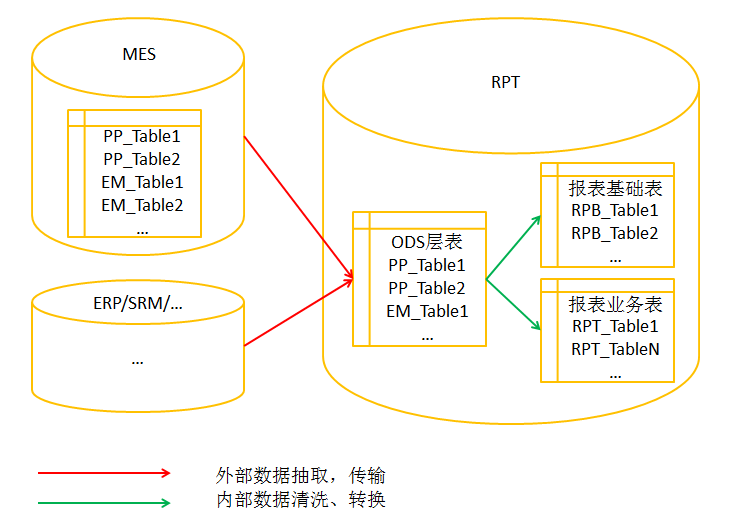
## 总体架构



如上图所示，报表ETL设计总体架构从大的方面分为以下两层。

* **第一层：**从MES、ERP等外部数据源提取数据，传输到RPT的ODS层。为了最大程度地加快这个传输过程，ODS层的表是跟源系统中的表结构一一对应的，从原系统提取数据后直接传输过来就行了，而不会涉及到数据转换等额外操作。
* **第二层：**在第一层的数据传输操作完成后，所有需要的数据都已经在ODS层表中了，因此第二层的数据转换操作就只在RPT内部进行了。这一层的主要工作是，将ODS层的数据进行标准化从而形成标准的基础数据（比如：颜色、车型等），或者统计汇总成最终的报表业务数据。视具体业务场景的复杂度，这个过程可能会经历一系列的中间表，层层汇总而来；也可能根本就没有这个过程，比如从第一层传输过来的数据可能就可以直接使用了，而不需要汇总操作了。

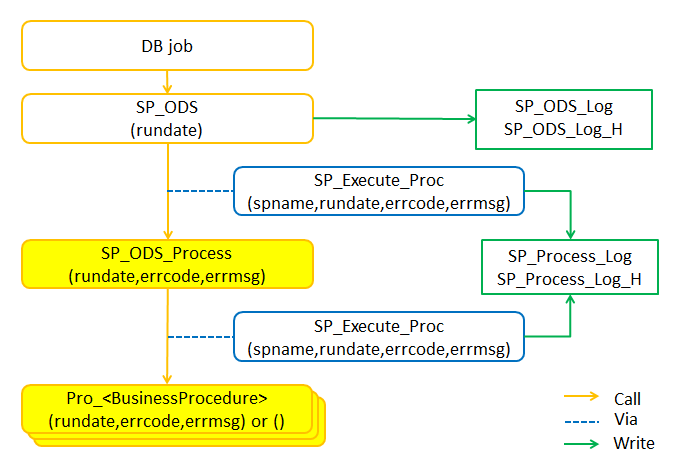
## 刷新策略：

根据业务不同，可采用如下的几种刷新策略：

* **全表刷新：**将原系统中的表完整刷新到报表系统中，适合于一些比较小的主数据表。具体实现也可有两种方式：a）不管三七二十一，先删除，再插入；b）比较源表与目标表中的记录，根据比较的结果在目标表中应用相应的操作（插入或更新），在Oracle数据库中可以使用merge语句在一个事务内完成。
* **增量刷新（周期不固定）：**自上次刷新以来，将源系统中变更的记录（包括新增的和修改的）应用到报表系统中这个策略通常要求原系统的相关表中有创建时间和最后修改时间的字段，通常适合用于比较大的主数据表。
* **增量刷新（固定周期）：**每次刷新只考虑本周期内的数据，比如按天为固定周期，则每次刷新只考虑某一天的数据，通常适合于业务数据表。

## 存储过程规划

参考之前动力BI项目的设计思路，整理如下图：



1. DB job：ETL过程所使用的一个定时作业调度任务。在其内部调用统一的存储过程SP\_ODS。
2. SP\_ODS：ETL过程所使用的一个入口存储过程，在其内部通过SP\_Execute\_Proc存储过程调用SP\_ODS\_Process存储过程。
3. SP\_ODS\_Process：定义ETL过程所需要的各个步骤，以及它们执行顺序（一般来说，先执行外部数据抽取、再执行内部数据转换。先执行主数据的传输、再执行业务数据的传输），在其内部通过SP\_Execute\_Proc存储过程调用各个业务存储过程Pro\_<BusinessProcedure>s。
4. Pro\_<BusinessProcedure>：对应具体的一个步骤，通常，每一个要处理的表，会对应这里的一个存储过程。

作为示例，目前我已建立了一个PRO\_MD\_COLOUR存储过程，用于传输颜色主数据。相应地，如果要传输其他数据，则只要新增一个Pro\_<BusinessProcedure>，并修改SP\_ODS\_Process存储过程，在合适的位置加入对这个新增存储过程的调用即可。

## ETL表清单

目前为止，已知的需要做ETL的表清单如下：

（注：部分主数据，比如发动机排量、变速箱类型、总装/涂装/焊装车型等，要通过某些规则来抽取，由于尚不清楚从源系统中的哪些表中进行抽取，故没有在下表中列出）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **源系统** | **源系统表名** | **报表系统表名** | **刷新方式** | **刷新频率** |
| MES | MD\_COLOUR | 同源系统表名 | 增量刷新 | 天 |
| MES | BA\_Material | 同源系统表名 | 增量刷新 | 天 |
| MES | PP\_MANUDETAIL | 同源系统表名 | 增量刷新 | 天 |
| MES | PP\_VEHCONF | 同源系统表名 | 增量刷新 | 天 |
| MES | PP\_VEHINFO | 同源系统表名 | 增量刷新 | 天 |
| MES | PP\_VINGENERATE | 同源系统表名 | 增量刷新 | 天 |
| MES | QM\_FailureSymptom | 同源系统表名 | 增量刷新 | 天 |
| MES | QM\_RepairRecord | 同源系统表名 | 增量刷新 | 天 |
| MES | FTT\_QUALITYINFO | RPT\_QuanlityIssue | 增量刷新 | 天 |
| MES | EM\_EQUIPMENT | 同源系统表名 | 增量刷新 | 月 |
| MES | EM\_EQUIPMENTSTATUSCHANGERECORD | 同源系统表名 | 增量刷新 | 月 |