ETL的主要步骤

ETL的主要步骤

ETL的主要步骤

ETL(Extract Transform Loading, 数据抽取转化装载规则)是负责完成是数据源数据向数据仓库数据的转化的过程。是实施数据仓库中最重要的步骤。可以形象的说，ETL的角色相当于砖石修葺成房子的过程。在数据仓库系统设计中最难的部分是用户需求分析和模型设计，那么工作量最大的就是ETL规则的设计和实施了，它要占到整个数据仓库设计工作量的60%-70%，甚至更多。

下面是本人对ETL的几个重要步骤理解，和大家分享！

一、ODS区的数据采集：　　最主要作用为了尽量减少对业务系统的影响。表结构可以不必和DW一致。根据具体业务需求和数据量情况，将数据源的数据放入ODS有各种不同的方法，比如Oracle的数据库链路，表复制，SQL*LOADER，Teradata的Fastload，Sysbase的BCP等等。

需要解决的问题包括：

a、数据的时间差异性问题

在抽取旧有数据时，要将不同时期的数据定义统一，较早的数据不够完整或不符合新系统的数据规范，一般可以根据规则，在存入中转区的过程中予以更新或补充。

b、数据的平台多样性问题

在抽取旧有数据时，大部分数据都可采用表复制方式直接导入数据中转区集中，再做处理，但有部分数据可能需要转换成文本文件或使用第三方工具如Informatica等装载入数据中转区。这部分数据主要是与数据中转区数据库平台不一致的数据库数据，或非存储于数据库内的文本、excel等数据。

c 、数据的不稳定性问题

对于重要信息的完整历史变更记录，在抽取时可以根据各时期的历史信息，在抽取需要信息等基本属性的旧有数据时，要与相应时段的信息关联得到真实的历史属性。

d 、数据的依赖性问题

旧有业务系统的数据关联一般已有约束保证，代码表和参照表等数据也比较准确，但仍有少量数据不完整，对这部分数据，需根据地税的需求采取清洗策略，保证数据仓库各事实表和维表之间的关联完整有效。

数据仓库各事实表和维表的初始装载顺序有先后关系，要有一个集中的数据装载任务顺序方案，确保初始数据装载的准确。这可以通过操作系统或第三方工具的任务调度机制来保证。

二、数据转换、清洗：

将ODS中的数据，按照数据仓库中数据存储结构进行合理的转换，转换步骤一般还要包含数据清洗的过程。数据清洗主要是针对源数据库中出现二义性、重复、不完整、违反业务或逻辑规则等问题的数据数据进行统一的处理，一般包括如：NULL值处理，日期格式转换，数据类型转换等等。在清洗之前需要进行数据质量分析，以找出存在问题的数据，否则
数据清洗将无从谈起。数据装载是通过装载工具或自行编写的SQL程序将抽取、转换后的结果数据加载到目标数据库中。

数据质量问题具体表现在以下几个方面：

a、正确性（Accuracy）：数据是否正确的表示了现实或可证实的来源?

b、完整性（Integrity）：数据之间的参照完整性是否存在或一致?

c、一致性（Consistency）：数据是否被一致的定义或理解?

d、完备性（Completeness）：所有需要的数据都存在吗?

e、有效性（Validity）：数据是否在企业定义的可接受的范围之内?

f、时效性（Timeliness）：数据在需要的时侯是有效的吗?

g、可获取性（Accessibility）：数据是否易于获取、易于理解和易于使用?

以下综合说明数据仓库中数据质量要求，包括格式、完整性要求。

a、业务描述统一，对数据模型的不同版本融合、映射为唯一版本。包括：

1、在业务逻辑没有变化的前提下，旧的业务数据映射在新模型上。

2、 遗留系统的人事信息、考核相关信息与业务系统、行政其他模块要一致。

b、信息描述规范、完整。

1、不存在格式违规

数据类型不存在潜在错误。

2 、参照完整性未被破坏

数据不会找不到参照。

3 、不存在交叉系统匹配违规，数据被很好集成

相同的数据存在于多个系统中，数据之间要匹配。

4 、数据在内部一致

同样的纪录字段在同一个表中重复出现，不能有差别。

以下是对主要数据质量问题的清洗策略：

主要问题

表现形式

产生原因

清洗策略

数据完整性问题

大量的空值字段的出现

原OLTP系统中对很多字段没有做非空限制

1. 交由OLTP系统重新录入, 补齐

2. 在数据仓库对应的维表中建立一个新的字段, 将这些空值字段的值统一的赋值

超出字典表范围

填写这些值的时候是直接让用户填写而非下拉框选择

1. 交由OLTP系统重新录入, 补齐

2. 在数据仓库对应的维表中建立一个新的字段, 将这些空值字段的值统一的赋值

数据一致性问题

一个特定的字段在不同的表中内容不同

录入, 同步的问题

1. 选取最可靠的表中的字段为确定值

应该成为主键的值不唯一

原OLTP系统中未建立有效的主键关系

1. 消除错误, 重复的主键

三、数据加载：

将转换和清洗完的数据按照数据仓库的结构进行数据加载。需要考虑初始数据装载、数据刷新、加载顺序等等问题。

a、针对数据现状，初始导入有这样一些问题需要考虑：

1、如何解决时间差异性？

2、如何解决平台差异性？

3、如何适应数据的不稳定性？

4、如何解决数据依赖性？

b、数据刷

新的策略要根据业务需求和应用系统的承受能力和数据情况决定。主要有这样一些问题需要考虑：1、如何解决时间差异性？

2、如何适应数据的不稳定性？

3、如何解决平台差异性？

4、如何解决数据依赖性？

5、如何减少对业务系统的影响？

c、不同的刷新任务类型，对业务系统的影响不同，刷新任务有以下种归类特性：

1、刷新频率：

实时刷新、每数小时、每日、每周、每月、不定期手动刷新。

2、刷新方式：

数据库表复制、文本文件ftp再装载、物化视图、数据库trigger。

3、数据加工方式：

简单插入更新、增加计算项字段、多表关联更新、汇总、多表关联汇总计算。

并可针对各种异常情况做处理：回滚，重新装载，断点重新装载等等，还可在任务完成后（或失败后）将日志以Email方式发给数据仓库管理人员。

四、汇总层、CUBE加载：

ODS加载进入数据仓库的数据只是底层详细层数据，还需按定义的汇总规则进行汇总，生成数据集市用的汇总表或CUBE。ETL流程是指完成每个维表数据及事实表数据导入的顺序, 其包括两个部分, 初始导入数据时的ETL流程, 及增量导入时的ETL流程。

初始导入数据时的ETL流程

第一步: 自动生成维的数据装载

自动生成维一般来说就是日期，年度月份，年度等时间类维度（年度月份，年度其实都是日期维的一个层次，但某些事实表中没有日期信息，只有月份信息，所以需额外建立此二维度），几乎数据仓库中每个数据模型都需使用时间类维度，在加载其它维度和事实之前，需要先将时间维度生成出来。

第二步: 手工维护维度装载

实际数据仓库开发中，很可能会有些维度的数据在业务系统中无发得到，典型的是一些外部信息指表的类型代码，是由数据仓库开发人员设计的。所以需要手工方式建立这些信息，然后导入数据仓库。

第三步: 缓慢变化维表数据装载

这些维度可以从业务系统中找到来源，但变化比较缓慢。对于初始装载时，需要考虑对缓慢变化维的处理方式要和增量刷新方式一致。

在装载事实表数据之前，需要先装载这些维表。需要注意的是，有些维本身就是事实表，其所依赖的维必须先装载完成。

第四步: 事实表数据装载

然后是初始装载所有的事实表数据。事实表之间也有依赖关系，某些事实表需在其他事实表装载之前装载。（如果ETL程序已保障了数据的完整性，也可以在将关联约束禁用的情况下不考虑先后顺序，但一般不建议）。

第五步：聚合表初始生成

许多数据仓库的前端应用，并非直接使用主题星型模型中的事实表数据
，而是聚合表中汇总，运算好的数据。（Oracle OLAP Service所建立的ROLAP 和数据集市实际上也是使用一系列的经过大量预先计算得到的聚合表）

增量导入

第一步: 缓慢变化维表数据装载

每天将所有变化过的维度信息刷新到数据仓库中，维表数据的刷新必须现于事实表。

第二步: 事实表数据装载阶段

每天新增事实数据的导入，如同初始化导入一样，需要考虑任务之间的先后顺序。

第三步: 数据汇总和聚合

根据设定的聚合规则和时间段对数据进行聚合。

第四步: 作业调度和异常情况处理

五、任务调度策略

驱动策略

前导Job驱动：只有满足另外一个JOB成功后，自己才运行。文件驱动：当下传的文件到达，并经过检验准确后JOB才运行。时间驱动：当到达某个时点时，Job便开始运行。事件驱动：如人工参与，导致JOB执行。

通知设计：重要信息（成功/失败）的通知

1、成功退出

分段提交方式，当分段提交的当次任务都正确完成，即Job运行状态临时表中登记的作业状态全部为完成时，退出ETL调度。

自动提交方式，当当期所有的任务都正确完成，即Job运行状态表中登记的作业状态全部为完成时，退出ETL调度。

2、失败退出

关键作业异常，关键作业运行异常时，影响剩下的作业不能运行时，则退出ETL调度。

超过ETL时限，当超过预先设定的ETL?时限时，退出ETL调度。

数据库异常，当不能正常操作数据库时，退出ETL调度。

操作系统异常，当发生操作系统异常，导致程序不能正常运行，如文件系统异常导致读写文件错时，需要退出ETL调度。

3、手工退出，需要人为干预ETL调度的时候，能以手工操作的方式退出ETL调度。
ETL过程中，怎么保证数据的准确性，这个准确性包含两个方面：数据量的准确性，数值的正确性。
A、字符集的转换——怎么将基于不同字符集的数据转换到目的数据库。
1)保证目标数据库的字符集是其他源数据库字符集的超集
2）数据库超集
3）在ETL前详细调查源的字符集，落地解析的字符集以及目的地字符集，如果这个过程
是字符集的子集或者超集那就最好，不是的话就要评估数据的损失量或者制定ASCII的
转换机制（字符集是将‘字’存储为ASCII码，而这个ASCII码在另个字符集展现中是什么样子）
4）
B、数据量的准确性——使用ETL工具的质量监控工具就那么准么？我可遇到过不准的情况哦。
1）不符合标准的数据上作个记号，全部写入数据仓库中
2）至今还没什么好的方法，ETL不是数据平移－－用个count统计对比就知道数据量是否正确，当一个指标经过
多次join，过滤，判断，计算以后就已经很难保证抽取过来的数据在量上是正确的。
一般采取的方法是定义抽样检测，得出一定的概率，这个概率就算是指标体系的一个误差了。
3）制定严格的etl规则，符合规则的数据抽取到数据仓库中来，不符合规则的数据不在业务范畴之内。
C、discard的数据怎么重新能够加载回目的数据库中（ETL已经完毕了，并不是所有的数据能够重新加载进取的）。
1）不符合条件数据的维度值给以一些缺省值，一般情况下流水作业很难重新补登再入库。
2）a.制定手工更改抽取的指标；b.启用前一天业务数据备份，在另一个环境下在人为的干预下重新跑一次
该模块的ETL流程。
3）根据具体的业务逻辑，常见的方法是使用一个error table记录下被discard的数据，然后再一次
load的时候做一次union。
4）要把discard的数据重新加载进去，不是简单的union。
5）discard的数据应放到一张临时表中，有待业务和开发人员确认。
D、ETL一般都是在晚上生产库结束后进行的，如果第二天发现ETL数据不对，怎么保证还能够抽取到昨天的业务数据？
1）做个好的脚本，只负责删除和处理某个时间段的数据。
2）对于生产库的一些关键性交易数据都要每天在抽取之前做一次逻辑的备份，类似于create table
tb_2008_1_5 as select * from ...,一旦第二天发现抽取问题，还有昨天的静态数据源；而对于
流水性质的数据来说我们可以不用做tmp表在第二天还能重新抽取，但对于那种可以删除流水的业务系统就有点变态了：）
3）可把每天的业务数据用csv文件保存下来with timestamp，在做完当天的etl之后备份当天的，然后设计一个clear backup的方案，比如三个月删除一次。
4）在数据量很大的情况下使用csv文件并不现实，毕竟是平面文件，查询备份中的数据就是一个比较麻烦的问题，不能自由抽取当中一部分数据，如果生产库几千张表，将近100g的数据量时问题就很大，如果是
交易流水数据，没必要每天全量保存副本。
E、一个指标的计算是经过多层次的抽取后完成的，怎么保证在抽取的过程中数值精度不丢失？怎么让业务人员理解其指标
进行运算的过程（ETL抽取的时提供的可是技术元数据，客户所关心的可是业务员数据）？
1）自己理解后做好元数据维护和释义
2）举个简单的例子int－float－double－money－varchar。
3）数据精度看要求多少位了，尽量db中留出足够的精度，前台展现可以进行适量的舍入和平衡。
在展现层以文字或在线帮助的形式对可能会引起歧义的指标给出后台的计算公式，帮助用户更好的理解指标含义
ETL过程中，怎么保证数
据的准确性，这个准确性包含两个方面：数据量的准确性，数值的正确性。
大致包含以下内容：
A、字符集的转换——怎么将基于不同字符集的数据转换到目的数据库。
保证目标数据库的字符集是其他源数据库字符集的超集

B、数据量的准确性——使用ETL工具的质量监控工具如何保证
至今还没什么好的方法，ETL不是数据平移－－用个count统计对比就知道数据量是否正确，当一个指标经过多次join，过滤，判断，计算以后就已经很难保证抽取过来的数据在量上是正确的。一般采取的方法是定义抽样检测，得出一定的概率，这个概率就算是指标体系的一个误差了。
C、discard的数据怎么重新能够加载回目的数据库中（ETL已经完毕了，并不是所有的数据能够重新加载进取的）。
1）制定手工更改抽取的指标；2）启用前一天业务数据备份，在另一个环境下在人为的干预下重新跑一次该模块的ETL流程。
D、ETL一般都是在晚上生产库结束后进行的，如果第二天发现ETL数据不对，怎么保证还能够抽取到昨天的业务数据？
对于生产库的一些关键性交易数据都要每天在抽取之前做一次逻辑的备份，类似于create table tb_2008_1_5 as select * from ...,一旦第二天发现抽取问题，还有昨天的静态数据源；而对于流水性质的数据来说我们可以不用做tmp表在第二天还能重新抽取，但对于那种可以删除流水的业务系统就有点变态了：）
E、(a)一个指标的计算是经过多层次的抽取后完成的，怎么保证在抽取的过程中数值精度不丢失？(b)怎么让业务人员理解其指标?
(a).数据精度看要求多少位了，尽量db中留出足够的精度，前台展现可以进行适量的舍入和平衡.
(b).在展现层以文字或在线帮助的形式对可能会引起歧义的指标给出后台的计算公式，帮助用户更好的理解指标含义