转载ETL架构师面试题

1. What is a logical data mapping and what does it mean to the ETL team?

什么是逻辑数据映射？它对ETL项目组的作用是什么？

答：

逻辑数据映射（Logical Data Map）用来描述源系统的数据定义、目标数据仓库的模型以及将源系统的数据转换到数据仓库中需要做操作和处理方式的说明文档，通常以表格或Excel的格式保存如下的信息：

目标表名：

目标列名：

目标表类型：注明是事实表、维度表或支架维度表。

SCD类型：对于维度表而言。

源数据库名：源数据库的实例名，或者连接字符串。

源表名：

源列名：

转换方法：需要对源数据做的操作，如Sum(amount)等。

逻辑数据映射应该贯穿数据迁移项目的始终，在其中说明了数据迁移中的ETL策略。在进行物理数据映射前进行逻辑数据映射对ETL项目组是重要的，它起着元数据的作用。项目中最好选择能生成逻辑数据映射的数据迁移工具。

2. What are the primary goals of the data discovery phase of the data warehouse project?

在数据仓库项目中，数据探索阶段的主要目的是什么？

答：

在逻辑数据映射进行之前，需要首先对所有的源系统进行分析。对源系统的分析通常包括两个阶段，一个是数据探索阶段（Data Discovery Phase），另一个是异常数据检测阶段。

数据探索阶段包括以下内容：

1．收集所有的源系统的文档、数据字典等内容。

2．收集源系统的使用情况，如谁在用、每天多少人用、占多少存储空间等内容。

3．判断出数据的起始来源（System-of-Record）。

4．通过数据概况（Data Profiling）来对源系统的数据关系进行分析。

数据探索阶段的主要目的是理解源系统的情况，为后续的数据建模和逻辑数据映射打下坚实的基础。

3. How is the system-of-record determined?

如何确定起始来源数据？

答：

这个问题的关键是理解什么是System-of-Record。System-of-Record和数据仓库领域内的其他很多概念一样，不同的人对它有不同的定义。在Kimball的体系中，System-of-Record是指最初产生数据的地方，即数据的起始来源。在较大的企业内，数据会被冗余的保存在不同的地方，在数据的迁移过程中，会出现修改、清洗等操作，导致与数据的起始来源产生不同。

起始来源数据对数据仓库的建立有着非常重要的作用，尤其是对产生一致性维度来说。我们从起始来源数据的越下游开始建立数据仓库，我们遇到垃圾数据的风险就会越大。

Architecture

4. What are the four basic Data Flow steps of an ETL process?

在ETL过程中四个基本的过程分别是什么？

答：

Kimball数据仓库构建方法中，ETL的过程和传统的实现方法有一些不同，主要分为四个阶段，分别是抽取（extract）、清洗（clean）、一致性处理（comform）和交付（delivery），简称为ECCD。

1．抽取阶段的主要任务是：

读取源系统的数据模型。

连接并访问源系统的数据。

变化数据捕获。

抽取数据到数据准备区。

2．清洗阶段的主要任务是：

清洗并增补列的属性。

清洗并增补数据结构。

清洗并增补数据规则。

增补复杂的业务规则。

建立元数据库描述数据质量。

将清洗后的数据保存到数据准备区。

3．一致性处理阶段的主要任务是：

一致性处理业务标签，即维度表中的描述属性。

一致性处理业务度量及性能指标，通常是事实表中的事实。

去除重复数据。

国际化处理。

将一致性处理后的数据保存到数据准备区。

4．交付阶段的主要任务是：

加载星型的和经过雪花处理的维度表数据。

产生日期维度。

加载退化维度。

加载子维度。

加载1、2、3型的缓慢变化维度。

处理迟到的维度和迟到的事实。

加载多值维度。

加载有复杂层级结构的维度。

加载文本事实到维度表。

处理事实表的代理键。

加载三个基本类型的事实表数据。

加载和更新聚集。

将处理好的数据加载到数据仓库。

从这个任务列表中可以看出，ETL的过程和数据仓库建模的过程结合的非常紧密。换句话说，ETL系统的设计应该和目标表的设计同时开始。通常来说，数据仓库架构师和ETL系统设计师是同一个人。

5. What are the permissible data structures for the data staging area? Briefly describe the pros and cons of each.

在数据准备区中允许使用的数据结构有哪些？各有什么优缺点？

答：

1．固定格式的文本文件。（Flat File）

Flat File指的是一种保存在系统上的一种文本文件格式，它以类似数据库的表的方式用行和列来保存数据。这种文件格式经常用来进行数据交换。用于保存数据不太合适。

2．XML数据集。

多用于数据交换，用户保存数据不太合适。

3．关系数据库的表。

保存数据的较理想选择。

4．独立的数据库表。

独立的数据库表一般指建立的表和其他表没有外键约束关系。这样的表多用于数据处理。

5．三范式或者关系型模型。

6．非关系型数据源。

非关系型数据源一般包括COBOL copy books、VSAM文件、Flat文件、Spreadsheets等。

7．维度模型。

8．原子事实表和聚集事实表。

9．代理键查找表。

6. When should data be set to disk for safekeeping during the ETL?

简述ETL过程中哪个步骤应该出于安全的考虑将数据写到磁盘上？

答：

Staging的意思就是将数据写到磁盘上。出于安全及ETL能方便重新开始，在数据准备区（Staging Area）中的每个步骤中都应该将数据写到磁盘上，即生成文本文件或者将建立关系表保存数据，而不应该以数据不落地方式直接进行ETL。

例如，在数据抽取阶段，我们需要连接到源系统，为了对源系统的影响尽量小，我们需要将抽取的数据保存成文本文件或者放入数据准备区的表中，这样，当ETL过程出现错误而失败时，我们就可以从这些文本文件开始ETL，而不需要再次影响源系统。

Extract

7. Describe techniques for extracting from heterogeneous data sources.

简述异构数据源中的数据抽取技术。

答：在数据仓库项目中，需要抽取的数据经常来自不同的数据源，它们的逻辑结构和物理结构都可能不同，即称之为异构数据源。

在对异构数据源进行整合抽取时，我们需要做的事情依次是标识出所有的源系统，对源系统进行概况分析，定义数据匹配逻辑，建立筛选规则，生成一致性维度。

对于源数据的操作系统平台和数据平台各不相同的情况，我们需要根据实际情况来确定如何进行数据抽取，通常的方法有建立ODBC连接、定义接口文件、建立DBLINK等方法。

8. What is the best approach for handling ERP source data?

从ERP源系统中抽取数据最好的方法是什么？

答：ERP系统的产生是为了解决企业内异构数据的整合。这个问题也是数据仓库系统面临的主要问题。ERP的解决方案是将企业内的各个应用（包括销售、会计、人力资源、库存和产品等）建立在相同的平台和相同的应用框架下，即在应用操作层将企业内的数据进行了一致性处理。而数据仓库是在应用操作层之上建立一致性的规则并进行一致性处理。目前比较流行的ERP系统有SAP、PeopleSoft、Oracle、Baan和J.D.EDwards（大部分没接触过）。

如果企业内只有一套ERP系统，那么数据就已经是一致的了，为数据抽取提供了方便。如果企业内除了ERP外还有其他系统，则数据抽取会变得复杂。因为目前的ERP系统的数据模型都非常复杂，可能有几百几千个表，并且较难理解。直接在ERP系统上建立数据捕获和抽取是非常复杂的。最好的办法是购买能针对ERP系统数据抽取提供功能的ETL工具，将ERP内部的复杂性留给ETL厂商处理。

9. Explain the pros and cons of communicating with databases natively versus ODBC.

简述直接连接数据库和使用ODBC连接数据库进行通讯的优缺点。

答：通常连接数据库的方式分为两类，一类是直接连接，另一类是通过ODBC连接。

直接连接的方式主要是通过COBOL、PL/SQL、Transact-SQL等方式连接数据库。这种方式的优点是运行性能高，可以使用DBMS提供的一些特殊功能。缺点是通用性差。

ODBC是为windows应用程序访问数据库提供的一组接口。ODBC的优点是灵活性，通过改变驱动和连接方式可以使用不同的数据库。ODBC方式的缺点是性能差。使用ODBC连接方式实现ETL的话，在ETL程序和至少要有两层，分别是ODBC Manager层和ODBC Driver层。另外，使用ODBC方式不能使用DBMS提供的一些特殊的功能。

10. Describe three change data capture (CDC) practices and the pros and cons of each.

简述出三种变化数据捕获技术及其优缺点。

答：

变化数据捕获（CDC）技术是ETL工作中的重点和难点，通常需要在增量抽取时完成。实现变化数据捕获时最理想的是找到源系统的DBA。如果不能找到，就需要ETL项目组自己进行检测数据的变化。下面是一些常用的技术。

1．采用审计列

审计列指表中如“添加日期”、“修改日期”、“修改人”等信息的字段。应用程序在对该表的数据进行操作时，同时更新这些字段，或者建立触发器来更新这些字段。采用这种方式进行变化数据捕获的优点是方便，容易实现。缺点是如果操作型系统没有相应的审计字段，需要改变已有的操作型系统的数据结构，以保证获取过程涉及的每张表都有审计字段。

2．数据库日志

DBMS日志获取是一种通过DBMS提供的日志系统来获得变化的数据。它的优点是对数据库或访问数据库的操作系统的影响最小。缺点是要求DBMS支持，并且对日志记录的格式非常了解。

3．全表扫描

全表扫描或者全表导出文件后进行扫描对比也可以进行变化数据捕获，尤其是捕获删除的数据时。这种方法的优点是，思路清晰，适应面广，缺点是效率比较差。

Data Quality

11. What are the four broad categories of data quality checks? Provide an implementation

technique for each.

数据质量检查的四大类是什么？为每类提供一种实现技术。

答：数据质量检查是ETL工作中非常重要的一步，主要关注一下四个方面。

1．正确性检查（Corret）

检查数据值及其描述是否真实的反映了客观事务。例如地址的描述是否完全。

2．明确性检查（Unambiguous）

检查数据值及其描述是否只有一个意思或者只有一个解释。例如地名相同的两个县需要加区分方法。

3．一致性检查（Consistent）

检查数据值及其描述是否统一的采用固定的约定符号来表示。例如币别中人民币用'CNY'。

4．完全性检查（Complete）

完全性有两个需要检查的地方，一个是检查字段的数据值及其描述是否完全。例如检查是否有空值。另一个是检查记录的合计值是否完全，有没有遗忘某些条件。

12. At which stage of the ETL should data be profiled?

简述应该在ETL的哪个步骤来实现概况分析？

答：数据概况分析是对源数据内容的概况进行分析，应该在项目的开始后尽早完成，它会对设计和实现有很大的影响。在完成需求收集后就应该立即开始数据概况分析。

数据概况分析不光是对源系统的数据概况的定量描述，而且为ETL系统中需要建立的错误事件事实表（Error Event Table）和审计维度表（Audit Dimension）打下基础，为其提供数据。

13. What are the essential deliverables of the data quality portion of ETL?

ETL项目中的数据质量部分核心的交付物有那些？

答：ETL项目中数据质量部分的核心的交付物主要有下面三个：

1．数据概况分析结果

数据概况分析结果是对源系统的数据状况的分析产物，包括如源系统中有多少个表，每个表有多少字段，其中多少为空，表间的外键关系是否存在等反映源系统数据质量的内容。这些内容用来决定数据迁移的设计和实现，并提供给错误事件事实表和审计维度表需要的相关数据。

2．错误事件事实表

错误事件事实表及相关的一系列维度表是数据质量检查部分的一个主要交付物。粒度是每一次数据质量检查中的错误信息。相关维度包括日期维度表、迁移信息维度表、错误事件信息维度表，其中错误事件信息维度表中检查的类型、源系统的信息、涉及的表信息、检查使用的SQL等内容。错误事件事实表不提供给前台用户。

3．审计维度表

审计维度表是给最终用户提供数据质量说明的一个维度表。它描述了用户使用的事实表的数据来源，数据质量情况等内容。

14. How can data quality be quantified in the data warehouse?

如何来量化数据仓库中的数据质量？

答：在数据仓库项目中，通常通过不规则数据的检测工作（Anomaly Detection）来量化源系统的数据质量。除非成立专门的数据质量调查项目组，否则这个工作应该由ETL项目组完成。通常可以采用分组SQL来检查数据是否符合域的定义规则。

对于数据量小的表，可以直接使用类似下面的SQL完成。

select state, count(\*) from order\_detail group by state

对于数据量大的表，一般通过采样技术来减少数据量，然后进行不规则数据检测。类似SQL如下。

select a.\* from employee a, (select rownum counter, a.\* from employee a) B where a.emp\_id = b.emp\_id and mod(b.counter, trunc((select count(\*) from employee)/1000,0)) = 0

如果可以采用专门的数据概况分析工具进行的话，可以减少很大的工作量。

Building mappings

15. What are surrogate keys? Explain how the surrogate key pipeline works.

什么是代理键？简述代理键替换管道如何工作。

答：在维度表的迁移过程中，有一种处理方式是使用无意义的整型值分配给维度记录并作为维度记录的主键，这些作为主键的整型值称为代理键（Surrogate Key）。使用代理键有很多好处，如隔离数据仓库与操作环境，历史记录的保存，查询速度快等。

同时，在事实表的迁移过程中，为了保证参照完整性也需要进行代理键的替换工作。为了代理键替换的效率高一些，我们通常在数据准备区中建立代理键查找表（Surrogate Mapping Table or Lookup Table）。代理键查找表中保存最新的代理键和自然键的对应关系。在对事实表进行代理键替换时，为了保证效率高，需要把代理键查找表中的数据加载到内存中，并可以开多线程依次替换同一记录的中的不同代理键，使一条事实记录在所有的代理键都替换完后再写如磁盘中，这样的替换过程称为代理键替换管道（Surrogate Key Pipeline）。

16. Why do dates require special treatment during the ETL process?

为什么在ETL的过程中需要对日期进行特殊处理？

答：在数据仓库的项目中，分析是主导需求，而基于日期和时间的分析更是占了很大的比重。而在操作型源系统中，日期通常都是SQL的DATETIME型的。如果在分析时，使用SQL对这种类型的字段临时处理会出现一些问题，如效率很差，不同的用户会采用不同的格式化方法导致报表不统一。所以，在数据仓库的建模时都会建立日期维度表和时间维度表，将用到的和日期相关的描述都冗余到该表中。

但是，并不是所有的日期都被转化为日期维度表的外键。日期维度表中的记录是有限的，有些日期如生日等可能会比日期维度表中记录的最小日期还要早，这类字段可以直接在数据仓库中保存SQL的DATETIME型。而像购买日期等与分析的业务紧密相关的通常都需要转化为日期维度表的外键，可以用日期维度表中统一的描述信息进行分析。

17. Explain the three basic delivery steps for conformed dimensions.

简述对一致性维度的三种基本的交付步骤。

答：数据整合的关键就是生成一致性维度，再通过一致性维度将来自不同数据源的事实数据合并到一起，供分析使用。通常来说，生成一致性维度有如下三个步骤：

1.标准化（Standardizing）

标准化的目的是使不同数据源的数据编码方式，数据格式等相同，为下一步数据匹配打下基础。

2.匹配（Matching and Deduplication）

数据匹配的工作有两种，一种是将不同数据源的标识同一事物的不同属性匹配到一起，是数据更完善；另一种是将不同数据源的相同数据标识成重复，为下一步的筛选打下基础。

3.筛选（Surviving）

数据筛选的主要目的是选定一致性维度作为主数据（Master Data），也就是最终交付的一致性维度数据。

18. Name the three fundamental fact grains and describe an ETL approach for each.

简述三种基本事实表，并说明ETL的过程中如何处理它们。

答：事实表从粒度的角色来划分可以分为三类，分别是交易粒度事实表（Transaction Grain）、周期快照粒度事实表（Periodic Snapshot）和累计快照粒度事实表（Accumulating Snapshot）。在事实表的设计时，一定要注意一个事实表只能有一个粒度，不能将不同粒度的事实建立在同一张事实表中。

交易粒度事实表的来源伴随交易事件成生的数据，例如销售单。在ETL过程中，以原子粒度直接进行迁移。

周期快照事实表是用来记录有规律的，固定时间间隔的业务累计数据，例如库存日快照。在ETL过程中，以固定的时间间隔生成累计数据。

累积快照事实表用来记录具有时间跨度的业务处理过程的整个过程的信息。在ETL过程中，随着业务处理过程的步骤逐步完善该表中的记录。

19. How are bridge tables delivered to classify groups of dimension records associated to a singlefact?

简述桥接表是如何将维度表和事实表进行关联的？

答：桥接表（Bridge Table）是维度建模中的一类比较特殊的表。

在数据仓库的建模时，会遇到具有层次结构的维度表，对于这样的表有一种建模方式是建立父子表，即每条记录上包括一个指向其父记录的字段。这种父子表的建立在层级深度可变时尤其有用，是一个紧凑而有效的建模方式。但是这种建模方式也有缺点，就是用标准SQL很难对递归结构进行操作。

与这种递归结构的父子表不同，桥接表采用不同的建模方式也可以表示这种层级结构。桥接表是建立在维度表和事实表中间的一个具有较多冗余信息的表，其中的记录包含层级结构中节点到其下面每个节点的路径。表结构如下所示：

父关键字

子关键字

父层数

层名

底端标识

顶端标识

在桥接表中，节点与其下面的任意一个节点都建立一个关联记录保存在表中，即父子关系不再局限在相邻层，如第一层与第三层同样有父子关系，通过父层数可以区分相隔了几层。这样，可以通过父层数和父子关系来进行层级结构的查询。

当然，桥接表也不是一个完备的解决方案，它只能是在某些情况下是查询变得容易。

20. How does late arriving data affect dimensions and facts? Share techniques for handling each.

迟到的数据对事实表和维度表有什么影响？怎样来处理这个问题？

答：迟到的数据分为两种，一种是迟到的事实表数据，另一种是迟到的维度表数据。

对于迟到的事实记录，我们可以插入到相应的事实表中。在插入的同时，还需要做一些处理。首先，对于具有SCD TYPE 2型维度的事实记录需要在插入前判断该事实记录的发生日期到目前为止，维度记录是否发生过变化，如果有变化，该事实记录需要对应到事实发生时的维度记录上。其次，在事实记录插入完成后，与该事实表相关的聚集事实表和合并事实表需要做相应的处理。

对于迟到的维度记录，我们需要做的处理要复杂一些。首先，如果迟到的维度记录是第一次进入数据仓库中，那么需要在维度表中生成一条维度记录，并将与该维度记录对应的事实记录的外键进行更新。其次，如果迟到的维度记录是对原维度进行的修改，那么我们在维度表中生成一条新记录的同时，还需要找到维度本次变化到下次变化间的事实行，并将其维度外键更新为新加维度的代理关键字。

Metadata

21. Describe the different types of ETL metadata and provide examples of each.

举例说明各种ETL过程中的元数据。

答：元数据是ETL项目组面对的一个非常重要的主题，对于整个数据仓库项目也是非常重要的一部分。对于元数据的分类和使用没有很确定的定义。

通常来说，我们可以把元数据分为三类，分别为业务元数据（Business Metadata），技术元数据（Technical Metadata）和过程处理元数据（Process Execution Metadata）。

业务元数据，是从业务的角度对数据的描述。通常是用来给报表工具和前端用户对数据进行分析和使用提供帮助。

技术元数据，是从技术的角度对数据的描述。通常包括数据的一些属性，如数据类型、长度、或者数据概况分析后一些结果。

过程处理元数据，是ETL处理过程中的一些统计数据，通常包括有多少条记录被加载，多少条记录被拒绝接受等数据

22. Share acceptable mechanisms for capturing operational metadata.

简述获取操作型元数据的方法。

答：操作型元数据（Operational Metadata），也就是过程处理元数据，记录的是ETL过程中数据迁移情况，如上次迁移日期，加载的记录数等信息。这部分元数据在ETL加载失败时会非常重要。

一般来说，对于使用ETL工具的数据加载，像迁移调度时间、迁移调度顺序，失败处理等内容都可以在由在迁移工具中定义生成。像上次迁移日期等数据可以建表保存。

如果是手工编写ETL程序的话，操作型元数据的处理会麻烦一些，需要自己来获取和存储。获取的方式，不同的编程方式会不尽相同。

23. Offer techniques for sharing business and technical metadata.

Optimization/Operations

简述共享业务元数据和技术元数据的方法。

答：为了能共享各种元数据，在数据仓库的构建过程中必须要有一些元数据标准，并在实际开发中遵守这些标准。这些标准包括元数据命名规则、存储规则及共享规则等内容。有关元数据标准的内容可以参看公共仓库元模型（Common Warehouse Metamodel，CWM）的相关资料 。

在最基本的层面上，企业应该在下面三个方面制定好标准。

1．命名规则

命名规则应该在ETL组开始编码前制定好，范围包括表、列、约束、索引等等数据库对象以及其他一些编码规则。如果企业有自己的命名规则，ETL组应该遵守企业的命名规则。当企业的命名规则不能完全满足需求时，ETL组可以制定补充规则或者新的规则。对企业命名规则的改变需要有详细的文档记录，并提交企业相关部门审核。

2．架构

在ETL组开始工作前，架构应该先被设计好。例如ETL引擎是和数据仓库放在同一台服务器上还是单独设立服务器；数据准备区是建立成临时的还是持久的；数据仓库是基于维度建模的还是3NF建模的。并且这些内容应该有详细的文档记录。

3．基础结构

系统的基础结构也应该先确定好。例如解决方案是基于Windows的还是基于UNIX的。这些企业基础结构元数据应该在ETL组开始工作前制定好。这些内容也应该有详细的文档记录。

在ETL的开发中，制定好元数据标准并能很好的遵守，那么建立好的数据仓库的元数据就可以很好的完成共享功能。

24. State the primary types of tables found in a data warehouse and the order which they must be loaded to enforce referential integrity.

简述数据仓库中的表的基本类型，以及为了保证引用完整性该以什么样的顺序对它们进行加载。

答：数据仓库中的表的基本类型有维度表、事实表、子维度表、桥接表等几类。其中子维度表即雪花模型由支架维度技术处理，桥接表用来处理多值维度或层级结构。

数据仓库中需要加载的各类表之间有相互依赖的关系，所以加载时需要以一定的顺序进行加载。下面是一些加载的基本原则：

子维度表加载成功后，再加载维度表。

维度表加载成功后，再加载桥接表。

子维度表、维度表和桥接表都加载成功后，再加载事实表。

这个加载顺序可以通过主外键的关系来确定。

（注意，此回答为总线架构的数据仓库的表的加载顺序。）

25. What are the characteristics of the four levels of the ETL support model?

简述ETL技术支持工作的四个级别的特点。

答：数据仓库上线后，ETL组需要为保证ETL工作的正常运行提供技术支持。通常这种技术支持工作分为四个级别。

1．第一级别的技术支持通常是电话支持人员，属于技术支持服务窗口（Help Desk）类型。如果数据迁移出现错误或者用户发现数据有问题，问题通过电话反映到第一级别的技术支持处。第一级别支持人员通过ETL项目组提供的一些问题的解决办法尽可能的解决发现的问题，阻止问题升级。

2．第二级别的技术支持通常是系统管理员和DBA。如果第一级别不能解决问题，问题反映到第二级别。第二级别的人员通常技术上比较强，硬件基础结构和软件架构上的问题都可以解决。

3．第三级别的技术支持通常是ETL项目负责人。如果第二级别不能解决问题，问题反映到第三级别。ETL项目负责人应该具备足够的知识，能够解决生产环境中的绝大部分问题。ETL项目负责人在必要时可以和开发人员或者外部产品提供商对某些问题进行交流，以便找出解决问题的办法。

4．第四级别的技术支持通常是ETL的实际开发人员。如果第三级别不能解决问题，问题反映到第四级别。ETL的实际开发人员可以对代码进行跟踪分析并找到问题的解决办法。如果问题出现在产品供应商的应用中，还需要供应商提供技术支持。

在小一些的数据仓库环境中，也是通常的情况下，第三级别和第四级别可以合并在一起。合并后对第二级别的要求会高一些。不建议每次出现问题都找ETL的开发人员。第一级别的技术支持人员不应该仅仅提供电话支持服务，在将问题反映给下一个级别前，要尽自己的能力去解决问题。

26. What steps do you take to determine the bottleneck of a slow running ETL process?

如果ETL进程运行较慢，需要分哪几步去找到ETL系统的瓶颈问题。

答：ETL系统遇到性能问题，运行很慢是一件较常见的事情，这时要做的是逐步找到系统的瓶颈在哪里。

首先要确定是由CPU、内存、I/O和网络等产生的瓶颈，还是由ETL处理过程产生的瓶颈。

如果环境没有瓶颈，那么需要分析ETL的代码。这时，我们可以采用排除的方法，需要隔离不同的操作，并分别对它们进行测试。如果是采用纯手工编码方式的ETL处理，隔离不同的操作要麻烦一些，这时需要根据编码的实际情况来处理。如果是采用ETL工具的话，目前的ETL工具应该都有隔离不同处理的功能，隔离起来相对容易一些。

分析最好从抽取操作开始，然后依次分析各种计算、查找表、聚集、过滤等转换环节的处理操作，最后分析加载操作。

实际的处理中，可以按照下面的七个步骤来查找瓶颈。

1．隔离并执行抽取查询语句。

先将抽取部分隔离出来，去掉转换和交付，可以将数据直接抽取到文件中。如果这一步效率很差，基本确定是抽取SQL的问题。从经验来看，未经调优的SQL是一个最常见的导致ETL效率差的原因。如果这步没有问题进入第二步。

2．去掉过滤条件。

这一条是针对全抽取，然后在ETL处理中进行过滤的处理方式而言。在ETL处理中做过滤处理有时会产生瓶颈。可以先将过滤去掉，如果确定为这个原因，可以考虑在抽取时进行数据过滤。

3．排除查找表的问题。

参照数据在ETL处理过程中通常会加载到内存中，目的是做代码和名称的查找替换，也称查找表。有时查找表的数据量过大也会产生瓶颈。可以逐个隔离查找表，来确定是否是这里出现问题。注意要将查找表的数据量降到最低，通常一个自然键一个代理键就可以，这样可以减少不必要的数据I/O。

4．分析排序和聚集操作。

排序和聚集操作都是非常费资源的操作。对这部分隔离，来判断是否因为它们引起性能问题。如果确定是因为这个，需要考虑是否可以将排序和聚集处理移出数据库和ETL工具，移到操作系统中来处理。

5．隔离并分析每一个计算和转换处理。

有时转换过程中的处理操作也会引起ETL工作的性能。逐步隔离移除它们来判断哪里出了问题。要注意观察像默认值、数据类型转换等操作。

6．隔离更新策略。

更新操作在数据量非常大时是性能非常差的。隔离这部分，看看是否这里出了问题。如果确定是因为大批量更新出了性能问题。应该考虑将insert、update和delete分开处理。

7．检测加载数据的数据库I/O。

如果前面各部分都没有问题，最后需要检测是目标数据库的性能问题。可以找个文件代替数据库，如果性能提高很多，需要仔细检测目标数据库的加载过程中的操作。例如是否关闭了所有的约束，关闭了所有的索引，是否使用了批量加载工具。如果性能还没有提高，可以考虑使用并行加载策略。

27. Describe how to estimate the load time of a large ETL job.

Real Time ETL

简述如何评估大型ETL数据加载时间。

答：评估一个大型的ETL的数据加载时间是一件很复杂的事情。数据加载分为两类，一类是初次加载，另一类是增量加载。

在数据仓库正式投入使用时，需要进行一次初次加载，而这次初次加载需要的时间一般较难预料。在数据仓库的日常使用和维护中，每天需要对数据仓库进行增量加载。增量加载的数据量要比初次加载小很多。

下面以初次加载为例来谈谈如何评估大型ETL的数据加载时间。

对初次加载的加载时间进行预估，需要将整个ETL过程分成抽取、转换和加载三部分，分别对这三部分进行评估。

1．对抽取时间的评估。

抽取通常占用的ETL的大部分时间，而且对这部分需要时间的评估也是非常困难的。为了对这部分时间进行评估，我们可以将查询时间分成两部分，一部分是查询响应时间，另一部分是数据返回时间。查询响应时间指从查询开始执行到结果开始返回这段时间。数据返回时间指第一条记录返回到最后一条记录返回的时间。

另外，初次加载的数据量太大，我们可以考虑选择其中的一部分来评估整体的时间，实际处理中，可以选择事实表的一个分区。一般来说各个分区的数据量差不多，评估出一个分区的时间，乘上分区数可以作为整体的评估时间。

2．对数据转换时间的评估

数据转换工作通常在内存中完成，一般来说都有着非常快的速度，占总体时间的比重比较小。如果要评估这部分需要的时间的话，最简单的评估方法是先评估出抽取时间和加载时间，然后运行整个过程，用整体时间减去抽取时间和加载时间。

3．对加载时间的评估

很多原因都可能影响加载时间，其中最重要的两个分别是索引和日志。

对加载时间的评估，也可以像评估抽取时间时一样，选择加载数据的一部分，如1/200进行加载，计算出时间后乘以200来作为整体加载时间。

总之，大型ETL数据的加载时间的评估是很困难的，我们采用的方法主要是类比评估，即选择一部分数据减少整体时间进行评估。在进行评估时要注意到测试环境和生产环境的配置等的差别会引起评估结果的偏差。虽然这种对时间的评估一定会有误差，但是可以做为整体加载时间的一个参考。

28. Describe the architecture options for implementing real-time ETL.

简述在架构实时ETL时的可以选择的架构部件。

答：在建立数据仓库时，ETL通常都采用批处理的方式，一般来说是每天的夜间进行跑批。

随着数据仓库技术的逐步成熟，企业对数据仓库的时间延迟有了更高的要求，也就出现了目前常说的实时ETL（Real-Time ETL）。实时ETL是数据仓库领域里比较新的一部分内容。

在构建实时ETL架构的数据仓库时，有几种技术可供选择。

1．微批处理（microbatch ETL，MB-ETL）

微批处理的方式和我们通常的ETL处理方式很相似，但是处理的时间间隔要短，例如间隔一个小时处理一次。

2．企业应用集成（Enterprise Application Integration，EAI）

EAI也称为功能整合，通常由中间件来完成数据的交互。而通常的ETL称为数据整合。

对实时性要求非常高的系统，可以考虑使用EAI作为ETL的一个工具，可以提供快捷的数据交互。不过在数据量大时采用EAI工具效率比较差，而且实现起来相对复杂。

3．CTF（Capture, Transform and Flow）

CTF是一类比较新的数据整合工具。它采用的是直接的数据库对数据库的连接方式，可以提供秒级的数据。CTF的缺点是只能进行轻量级的数据整合。通常的处理方式是建立数据准备区，采用CTF工具在源数据库和数据准备区的数据库之间相连接。数据进入数据准备区后再经过其他处理后迁移入数据仓库。

4．EII（Enterprise Information Integration）

EII是另一类比较新的数据整合软件，可以给企业提供实时报表。EII的处理方式和CTF很相似，但是它不将数据迁移入数据准备区或者数据仓库，而是在抽取转换后直接加载到报表中。

在实际建立实时ETL架构的数据仓库时，可以在MB-ETL, EAI, CTF, EII及通常的ETL中作出选择或者进行组合。

29. Explain the different real-time approaches and how they can be applied in different business scenarios.

简述几种不同的实时ETL实现方法以及它们的适用范围。

答：实时数据仓库在目前来说还不是很成熟，成功案例也比较少，下面列举了一些实时数据仓库架构的实现方法。

1．EII ONLY

使用EII技术来代替实时的数据仓库，数据延迟可以保证在1分钟左右，支持数据整合的复杂程度较低。无法保存历史数据。

2．EII + Static DW

使用EII技术联合非实时的数据仓库，数据延迟可以保证在1分钟左右，1天内的数据整合的复杂程度较低，1天前的数据整合的复杂程度可以较高。可以保存历史数据。

3．ETL + Static DW

普通的ETL处理，数据延迟在1天。支持复杂程度较高的数据整合。保存历史数据。

4．CTF + Real-Time Partition + Static DW

使用CTF技术建立实时数据仓库，数据延迟可保证在15分钟左右。数据整合的复杂程度较低。保存历史数据。

5．CTF + MB-ETL + Real-Time Partition + Static DW

使用CTF技术和MB-ETL联合处理数据迁移，数据延迟可保证在1小时左右，支持数据整合的复杂程度较高，保存历史数据。

6．MB-ETL + Real-Time Partition + Static DW

直接使用MB-ETL建立实时数据仓库，数据延迟可保证在1小时左右，支持数据整合的复杂程度较高，保存历史数据。

7．EAI + Real-Time Partition + Static DW

使用EAI技术建立实时数据仓库，数据延迟可保证在1分钟左右，支持数据整合的复杂程度较高。保存历史数据。

上面列出了一些实时数据仓库架构的选择，写的不是很详细，只是提出个思路，供大家自己去找资料学习。

30. Outline some challenges faced by real-time ETL and describe how to overcome them.

简述实时ETL的一些难点及其解决办法。

答：实时ETL的引入给数据仓库的建设带来了很多新的问题和挑战，下面列举了一些问题，其中有些问题有具体的解决办法，有些只能在实际情况下去斟酌。

1．连续的ETL处理对系统可靠性提出更高的要求。

2．离散快照数据的间隔时间变短。

3．缓慢变化维变成快速变化维。

4．如何确定数据仓库中数据的刷新频率。

5．目的是只出报表还是要实现数据整合。

6．做数据整合还是应用整合。

7．采用点对点的方式还是集中的方式。

8．前端展现工具的数据刷新方式如何确定。