尽管数据库是一种通用软件，不过它也为各种各样的技术要求提供了解决方案，例如：

记录和存储数据：

可靠地存储数据，保护每个用户的数据不会因其他用户的改变而受到影响。

读取数据以便在线查看和生成报表：

提供数据的一致性视图。

分析数据：

汇总数据，探测数据的趋势和数据间的关联，并进行预测。

最后两方案以数据仓库来实现。企业或组织的战略战术管理需要使用商业智能，而数据仓库正是提供这些商业智能的基础设施的一部分。借助这些解决方案，可以将组织内数据存储中有价值的商业信息展示出来。

目前，数据仓库和商业智能解决方案已经被广泛部署，新的项目也非常流行。在这种趋势后面有一个相当简单的理由：这些项目被看作商业的核心，它们可以让商业团体获得投资的回报。

这种趋势一致就存在。20世纪90年代初期，Oracle就开始在Oracle 7中加入数据仓库的特性。后续版本为数据仓库和商业智能提供特性，性能更好、功能更强、更易于扩展和管理。Oracle还提供了用于构造和使用商业智能基础设施的工具，包括数据迁移和业务分析工具及应用程序。

商业智能基础设施可以帮助业务分析员确定：

* 新的商业方案与过去的经营成果有什么关系。
* 通过不同的角度考虑数据，得出新的方案。
* 将来的发展如何。
* 如何改变商业策略以便对未来的发展产生正面影响。

本章将介绍这些基本概念、技术以及在数据仓库和商业智能中使用的工具。

## 1、商业智能的基础知识

为什么需要数据仓库？或者说为什么需要商业智能解决方案呢？为什么在线事务处理（OLTP）数据库中的数据仅仅只是商业智能解决方案中的一部分？记住，数据仓库通常设计成具有下面的这些特征：

决策分析可以认清数据的发展趋势：

数据仓库中的数据几乎都是用来读而不是用于写的：

跨越多个系统时，OLTP系统中的数据并不是干净的或一致的：

设计一个有效的数据仓库不同于关系型数据库的标准规范设计过程：

### 1.1商业智能的发展

收集商业智能并不是一个新概念。将公司数据用于战略决策，而非用于跟踪和操作，这种形式随着计算的发展已存在很长时间了。

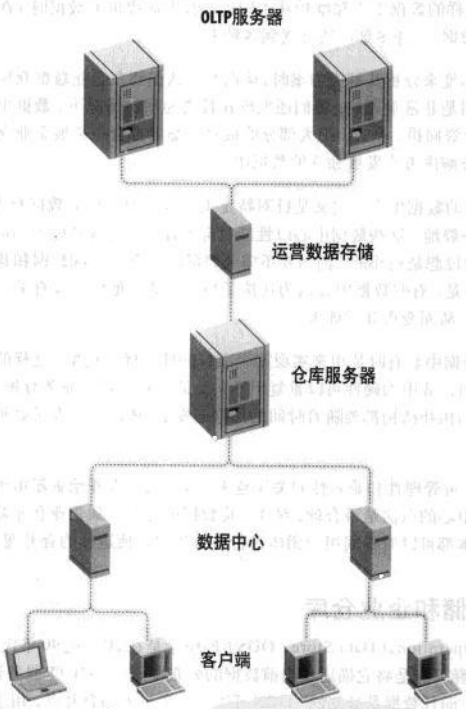
目前，数据仓库主要用于商业领域，如客户关系管理、销售和市场运营分析、产品管理和包装、进入分析以及风险估计和欺诈分析等。

20世纪80年代，许多公司开始对这些应用使用专用的系统，一般称为决策支持系统（DDS），作为对管理信息系统的补充。决策支持查询往往与CPU和内存相关，它们主要在只读环境中执行。而在传统的OLTP中，大部分I/O操作则都是写操作。查询的这种特征使得它比OLTP系统更具有不可预测性。这导致了用于决策支持的数据存储的开发与OLTP系统的开发相分离。

20世纪90年代早期，当Bill Inmon以及其他一些普及了术语“数据仓库”后，才有了一个用于构造数据仓库的规范化的公共基础设施。从此以后，商业智能解决方案的拓扑结构得到进一步发展，下面将逐一介绍。今天的商业智能解决方案通常包括用于数据仓库和OLTP系统中的数据以报表的形式展现出来的基础设施。目前基础硬件已经相当成熟；对于数据仓库硬件平台而言，I/O是一个在设计上需要重点考虑的事项。

### 1.2商业智能的拓扑结构

为服务于企业范围内庞大的信息源，典型的数据仓库的拓扑结构是多层的，如图：



由于各种原因，这个拓扑结构发展了很多年。最初创建单一企业数据仓库的努力一再陷入“分析瘫痪”。如同定义一个企业范围的OLTP模型需要花几年的时间一样（由于跨部门的管理不一样，工作的方向也不一样），数据仓库所要花费的时间比业务能够接受的时间要长得多。而且这些努力还常常会受到不断改变的需求分析的影响，需求分析则与不断变化的市场相关。虽然运营系统的数据元素和需求可以保持较长时间的稳定，但要理解业务的发展趋势则像视图将闪电装进瓶子里那样不可能。

因而，试图构建这样一个企业范围的模型以满足每个用户的需求，最终结果将是谁也不满意。

### 1.3数据中心（Data Mart）

当某些大规模、面向企业的数据仓库以失败而告终时，挫折感和急躁随之而来。为此，有些人开始从合适的运营源系统中提取数据，构建特定部门的独立数据中心。许多数据中心在最初非常成功，因为它们能够迅速满足特定业务的需求。

不过，问题也随之而来。部门之间经常会发生对于某些基本概念没有统一定义的情况，例如，“客户”这个概念。如果一个高级管理员向不同部门提出同样的问题，由这些“独立”的数据中心提供的答案往往是不同的，这样就导致了数据中心有效性的问题。在管理多个这样的数据中心和维护从不同运营数据源提取的数据时（在多个部门之间经常需要赋值数据），许多部门还会遇到各种困难。

当设计师从不同角度来分析其解决方案时，他们开始认识到在企业数据仓库级拥有明细数据的一致性视图是非常重要的。他们还发现在投资递增的情况下，数据中心可以解决商业问题，提供投资回报。今天，绝大部分的成功实施者都会在扩展企业仓库服务器的同时为自己的业务解决方案发展独立的数据中心。

简言之，当前公认的数据中心的定义是针对特定主体或针对应用的数据仓库，通常在一个部门内实现。一般地，这些数据中心以性能目标而建立，包括庞大的汇总表。关于数据中心，最初的设想是较小的，因为并不需要将部门所有的明细数据和其他部门的数据都加载进来。但是，有些数据中心因为连接了与业务的其他部分没有关系的外部来源，从而变得非常庞大。

在一些组织中，数据中心有时是用来实现某个项目的性能优化模型。这样的数据中心在项目完成时即退出，其中的硬件可以重复用于其他的项目。随着业务分析需求的变化，所有的数据中心的拓扑结构都要随着时间的推移而发展，所以开发人员必须要注意这种可能性。

随着对成本的控制、可管理性和兼容性的关注越来越多，很多人开始重新审视拥有大量的物理分离的数据中心的做法是否合理。结果，将数据中心合并到企业仓库是一个普遍的趋势。最近的版本都可以对不同用户团体进行有效管理，使这样的合并变得可能。

### 1.4运营数据存储和企业仓库

运营数据存储（Operational Data Store，ODS）的概念是在20世纪90年代开始流行的。对DOS最好的解释可能是将它描述成当前数据的分布中心。与OLTP服务器类似，其模式是非常规范的，而且数据是最新的。DOS可作为生成报表的合并点，用于查看不同部门的当前数据。DOS发展部分地是因为许多大公司正处于收购与合并的过程中。这些组织经常面对混合应用的环境。DOS就像一个中继站，可作为进一步变换为数据仓库或数据中心的数据源。

仓库服务器或企业数据仓库是指多主题历史信息存储，通常支持多部门，而且可作为记录的联合数据库。当ODS建立后，仓库服务器经常从ODS中提取数据。若没有ODS，仓库中的数据则直接由运营数据源提取并加以转换。仓库服务器也可能从外部数据源获取数据。

正如前面提到的，目前在多层结构中流行平台合并。企业数据仓库可以是ODS和多个数据中心的合并点。尽管存在不同的逻辑模型，它们可被合并到单一的平台和数据库中。

### 1.5OLTP系统和商业智能

真正的实时数据驻留在OTLP系统中。Oracle可以采用数据仓库系统的信息，将来自这些事务处理系统的报告同时显示在门户或面板上。一个有意义的面板关键之处在于提供一致的高质量数据。OTLP系统中数据的质量直接关系到数据输入的控制，以删除重复或有错误倾向的条目。

使用主数据管理（Master Data Management，MDM）可以解决一致性意义的问题。MDM解决方案由数据中心组成，这些数据中心是作为数据的一个共同的控制点，支持关键业务的质量，如客户、产品或财务。为了构造这样一个基础设施，Oracle为这些以及其他的商业领域提供了大量的数据中心。

这些利用数据仓库、OLTP系统以及MDM解决方案中数据的项目叫做数据集成项目。

## 2、数据仓库的设计

数据库服务器是商业智能的基础，它是数据存储的地方。但是商业智能不是简单的数据，只有当商业用户要使用数据仓库中的信息时，它才变得非常有用。这看起来似乎微不足道，但是在许多实际情况中，公司建立了数据仓库，而没有真正了解业务的实际需要。通常，这样部署的项目只支持很少的用户，起的作用也很小，只能获得很少的商业智能。

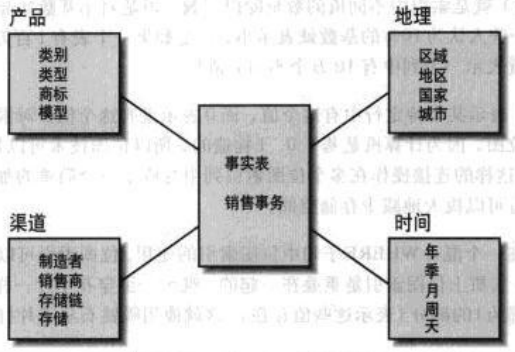
假设数据仓库已经得到很好的规划，而且有数据需求，下一个问题就是确定如何处理这个需求。你可能需要设计数据仓库和其他的基础组件从而为用户提供合理的性能，而此性能最初看起来可能远远高于你的系统能力，因为数据仓库请求的信息可能涉及大量明细数据的合计和比较。

开始设计的时候还需要记住：一个数据仓库不可能完善。当业务出现变化，数据仓库就要做相应的调整。这样，跟踪数据仓库中存储的元数据变化的能力在设计阶段就显得非常关键。许多设计工具可以提供这个能力。Oracle数据库企业版、标准版以及第一标准版包含了Oracle仓库构建器（Oracle Warehouse Builder，OWB）。该工具提供了一个元数据库，还提供从操作表中导入元数据，然后对新的模式和表进行前向工程的能力。数据仓库设计者可以在新表中添加新列，在新模式中添加约束。进行适当转化后，就可以建立了源列和目标列的映射。创建新表的DML脚本和用于ETL的PL/SQL或SQL Loader脚本将自动生成。

前面已经提到过，数据仓库和OLTP数据库有许多不同的使用特性。使数据仓库能够满足性能要求的一个方面是读操作要占较高的比列。第8章中详细描述的Oracle的锁模式最适合于数据仓库的操作。Oracle不对任何要读的数据加锁，在有许多数据库读操作的情况下可以减少竞争和资源请求。因为锁没有扩大，Oracle非常适合于在一个与OLTP工作负荷类似的场景中为数据仓库提供实时数据。

数据仓库的不同使用特点导致了与OLTP不同的设计模式。在OLTP数据库中，事务数据通常存放在多个表中，而数据项只存储一次。如果从不同事务表中查询数据，必须将表加以关联。一般地，数据库查询优化器将决定使用哪个表作为关联的开始点，不过这里要基于这样一个前提假定，即表中数据都同等重要。

虽然基于Oracle的数据仓库有时可以被规范成第三范式（3NF），然而当业务用户需要用一个易于理解的模式来标识他们自己的即兴查询或分析过程时，关键事务数据可能更适合存储在一个被维表或查询表围绕的中心事实表中，如图所示。事实表可以包含在数据仓库其他地方复制的数据项的汇总数据，而维表可以包含多个层次。正如前面提到的，当组织将他们的数据中心合并到企业数据仓库时，很多人都修改了以前的模式，采取了混合模式，它是第三范式和星型模式的混合。



数据仓库的使用者往往都采用经典的查询方式，如图所示的星型模式就是这样一个模型。一个典型的查询可能是这样的：

在过去6个月（时间）中在威斯康星洲（地理位置）的一家连锁店（渠道）卖出去多少台计算机（产品类型）。

图中显示出一个相对比较大的事务表（称为事实表），可被许多小表（称为维表或者查询表）围绕。刚才描述的查询通过称为多维查询。因为包括了几个维表（时间几乎总在其中）。因为这些查询在数据仓库中是很典型的，因此，如果Oracle的基于成本的优化器能够识别星型模式，那么将会带来巨大的性能收益。

## 3、查询优化

最早在Oracle 7中提供了在查询优化器中识别星型模式的能力，并且在后续的数据库版本中使得基于成本的查询优化器在响应商业智能查询上更敏捷。后来又更近异步提升了优化器预报的精度，自从10g以来，优化器预报被当作了实时性能，所有的错误被随之自动修正。通过物化视图，优化器也可以透明地提供对概要层的查询重写，这些概要层一般采用星型模式部署。Oracle 11g为OLTP选项添加了查询重写，以及改进的包含内联视图的查询。

优化器是如何处理一个星型模型的查询呢？首先，它注意到销售事务事实表比周围的维表有更多的项。这就是星型模式存在的线索。随着Oracle 7的发展，优化器开始产生了更加巧妙的计划。一个标准关系型数据库的优化器通常都是将每个维表并入事实表中，每次关联一个。因为事实表通常非常大，在多次关联过程中涉及事实表将会花费许多时间。

笛卡尔积关联首次在Oracle 7中引入，用于关联维表，然后在最后一步中，将维表关联的结果并入事实表中。在维表不太多（从经验来看，为了使笛卡尔积较小，一般为6个或更少），并且数据分布相对集中的情况下，这种技术非常合适。

在某些情况下，会有相当多的维表，或者在事实表的数据很稀疏。为了关联这些表，优化器可能会选择一个并行的位图星型连接。

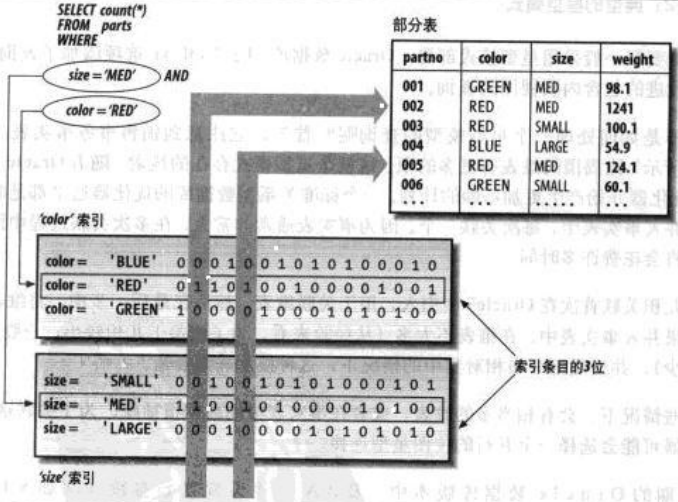
在早期的Oracle数据库版本中，DBA必须设置初始参数，收集统计表，以使优化器找出解决这些查询的最好办法。现今，需要的参数是在安装时调整的，统计表也是由Oracle数据库自动收集的。

### 3.1位图索引和并行

位图索引是Oracle 7中首次引入，可用来加速数据获取的速度，以及加速数据仓库查询连接的速度。Oracle中的位图索引一般采用基数小的数据的列。基数（Cardinality）就是索引中不同值的数量除以行数。但是对基数实际的含义有各种不同的意见。一些人认为10%的基数就表示小，但是如果有一个表有1百万行，那么10%的“小”基数就表示一个列中有10万行个不同的值！

位图索引中的1表示某个特定行中有这个值，而0表示没有这个值。对索引列中的每个值都建立一个位图。因为计算机是基于0，1构造的，所以位图基数可以加速数据获取。另外，如AND这样的连接操作在多个位图索引列中变成了一个简单的加法运算。位图索引的额外收益可以极大地减少存储空间。

如图显示了一个混合WHERE子句中位图索引的使用。位图索引可以联合使用以提供更好的性能。实质上位图索引是重叠在一起的，就像一组穿孔卡片一样。只要简单查找所有的位都为1的部分（表示这些值存在），这就像用吸铁石从卡片打孔的位置吸出一根针。



在Oracle中，改进星型查询的方法是建立事实表中外键列的位图索引，这些列与周围的维表相关联。位图从事实表中获取必要的行，并且将它们关联到维表中，这时出现的是一个并行的位图星型连接。在连接中，稀疏连接操作（例如大量的空值）被认为是位图的固有属性，维表的数量不再是问题。这个算法也可以用于雪花模式（snowflake schema），它是标准星型模式的扩展，每个维表中有多个表。

为了进一步加速查询，Oracle 9i添加了一个从事实表到维护表中的位图连接索引。位图连接索引只不过是两个或多个表的连接的位图索引。性能的提升来自于避免了事实表的连接或通过在连接数据前考虑约束减少了数据量。具有多维表的星型查询的性能可以得到很大的改善，因为星型转换中的未操作现在可以被消除了。

并行查询也可以显著得提高性能。决策支持查询中经常都有连接和分类操作。第7章描述过这种并行操作。当时列出了Oracle可以进行并行运算的所有功能。

从Oracle 9i开始，实时应用程序集群取代了Oracle并行服务武器（OPS），它通过让查询在集群或计算机系统网格的节点间透明地进行，进一步加大了并行化。

### 3.2汇总表

维表中的数据通常都是自然分层的（举例来说，在时间维表中，天积累成星期，星期积累成月，月积累成季度，季度积累成年）。如果查询只是要查看月份级的数据，那还有什么必要对天和星期的信息进行分类呢？相反地，它可以仅仅只在月份级或更高的级别中查看数据。原来，是数据仓库的性能专家们设计了这些类型的汇总表，其中包括多个层次的预先计算的统计结果。例如，在图中，列出所有时间周期都可以根据天数的不同分组来进行实时计算。但是，为了加快基于不同时间序列的查询，数据仓库可以实现计算好星期和月份的统计数据，并存放在汇总表中，以便直接查询获取这些数据。

### 3.3物化视图

Oracle 8i为创建汇总表引入了物化视图的概念，这些汇总表主要是针对可以表现出层次积累的事实表和维表的。物化视图可提供预计算摘要数据，更重要的是，在合适的时候，一个物化视图可以自动取代一个更大的明细表。基于成本的查询优化器可以执行查询，重写这些汇总表和层次中的积累层，这对于用户而言是不可见的，而且经常会使性能得到显著提高。举例来说，如果一个查询请求一个基于月份销售额明细表的汇总数据，在处理查询时查询优化器将自动用物化视图取代明细表。季度级数据的查询可能使用物化视图中的月数据进行合计，选择季度中的那些月。Oracle 10g增加了查询重写的能力，这样优化器可以利用多个合适的物化视图。

物化视图可以通过Oracle企业管理器进行管理。企业管理器中的SQL顾问包含了一个SQL访问顾问，它可以推荐何时创建物化视图。

## 4、数据库的分析、OLTP和数据挖掘

## 5、管理数据仓库

## 6、其他数据仓库软件

## 7、元数据的挑战

## 8、最佳实践