

Oracle Data Integrator 系统架构介绍

Oracle Data Integrator 是基于元数据管理的。元数据资料库存储的是 ODI 模型设计数据转换过程中的元数据信息等内容

- *Oracle Data Integrator 基础*
- *ODI 功能模型*
- *运行组件说明*
- *Metadata Navigator*



Oracle Data Integrator

系统架构介绍

同其他主流工具相比，Oracle 公司的数据加载工具 ODI (Oracle Data Integrator) 提出了一种 “ELT” 的架构，也是数据整合的一个开发平台，在不同信息系统或者数据库之间做数据传输、转换。——赵宇

在

上一期《Oracle 系列电子书》中，我们介绍了 [Oracle GoldenGate 数据同步技术](#)，其中我们对 Oracle 数据加载工具 Data Integrator 进行了篇幅较小的介绍，而在本期电子书中，我们将重点来了解一下这一款实用的工具，Oracle DBA 可以亲自尝试使用一下，或许可以得到意想不到的效果。

我们在做很多 IT 项目的过程中，经常会碰到需要把旧系统的数据迁移到新系统，或者把很多系统的数据整合到新系统做数据分析。碰到这样问题的时候，我们都通常会想到用 ETL (Extraction、Transformation、Loading –数据抽取、转换和加载) 工具来解决，主流的产品就是 IBM 公司的 Datastage、Informatica 公司的 Powercenter 以及 Cognos 公司的 DecisionStream 等，也有很多公司自己选择开发程序来解决。ETL 工具虽然能解决问题，但是成本很高，需要额外的中间转换服务器做数据转换。

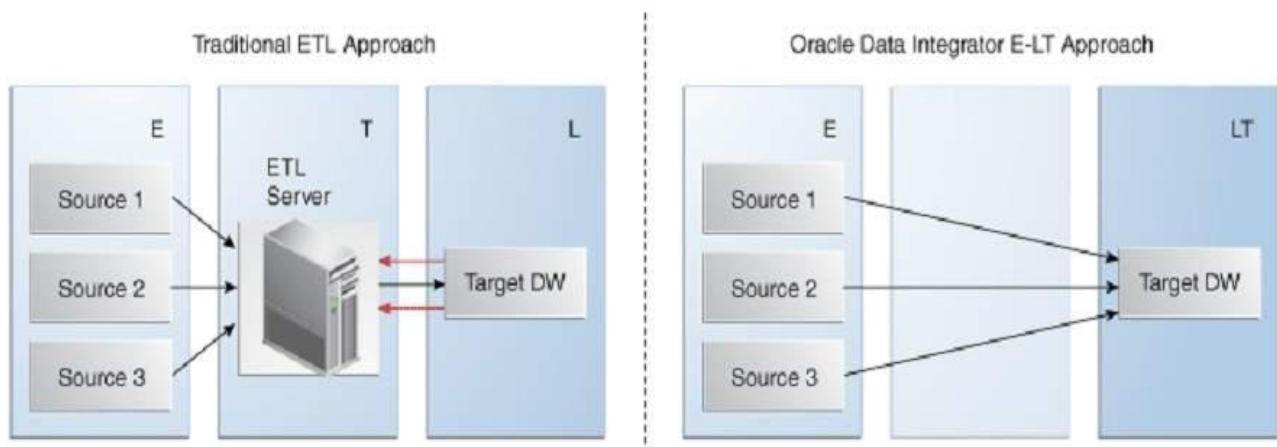
而 Oracle 公司的数据加载工具 **ODI(Oracle Data Integrator)** 提出了一种“ELT”的架构，也是数据整合的一个开发平台，在不同信息系统或者数据库之间做数据传输、转换。ODI 特点主要体现在两个方面：

- 1、ODI 是服务驱动的模式，也可以理解为时商业驱动的模式，它更多关注怎样符合业务规则，而不是把开发精力集中在技术层面；
- 2、ODI 在数据整合过程中，它自己本身不执行数据整合的处理，所有的执

行都是在现有的信息系统或者数据库中完成，直接把源数据在内存中加载在目标数据库以后再进行转换。

而且 ODI 集成了很多数据库的组件，可以充分利用数据库的功能进行异构数据库或者同构数据库之间的数据迁移，从而性能得到极大的提高（参考图 1-1）：

Figure 1-1 Traditional ETL versus ODI E-LT



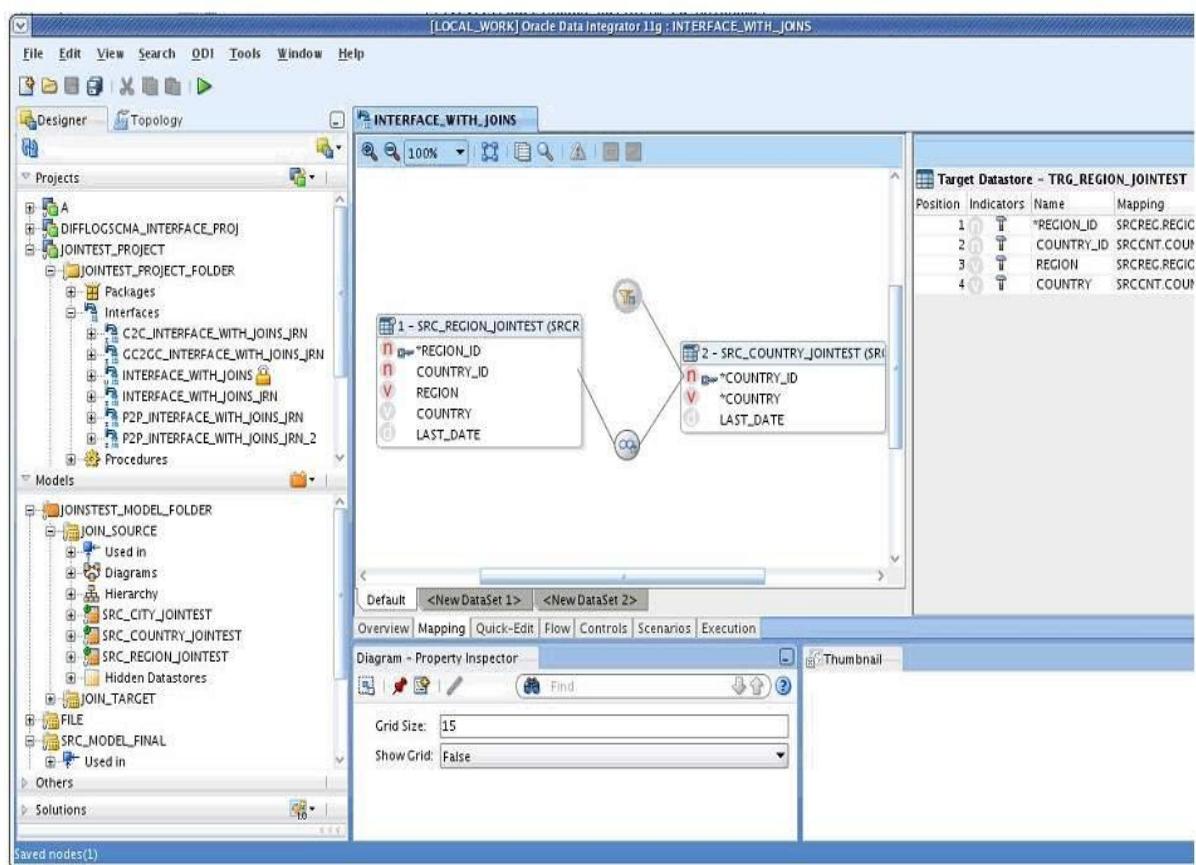
ODI 整体架构并没有 Oracle Database 复杂，因为它属于程序功能的使用，更多是程序开发和配置的工作，当然 ODI 的优化涉及到很多数据库优化的工作，从整体架构入手。基础有了，整体把握就会简单很多。

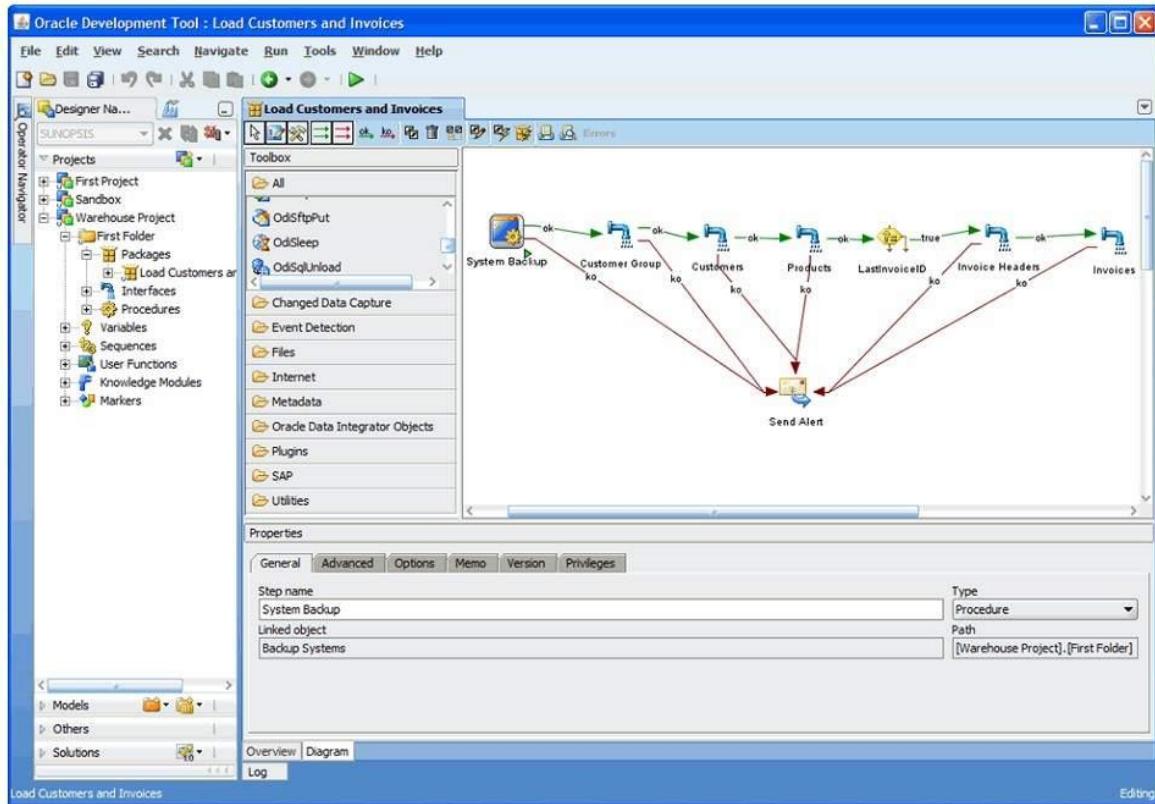
Oracle Data Integrator 简介

ODI 是基于元数据管理的。元数据资料库存储的是 ODI 模型设计数据转换过程中的元数据信息、数据库信息等内容。ODI 架构主要就是围绕这个模型资料库

来设计的，每个客户端都会通过自己的图形模型和执行代理来访问这个模型资料库，这个资料库安装在数据库一个用户里，存储着元数据的信息。客户也可以使用 Web application 和 Metadata Navigator 通过 Web 接口方式进行访问。

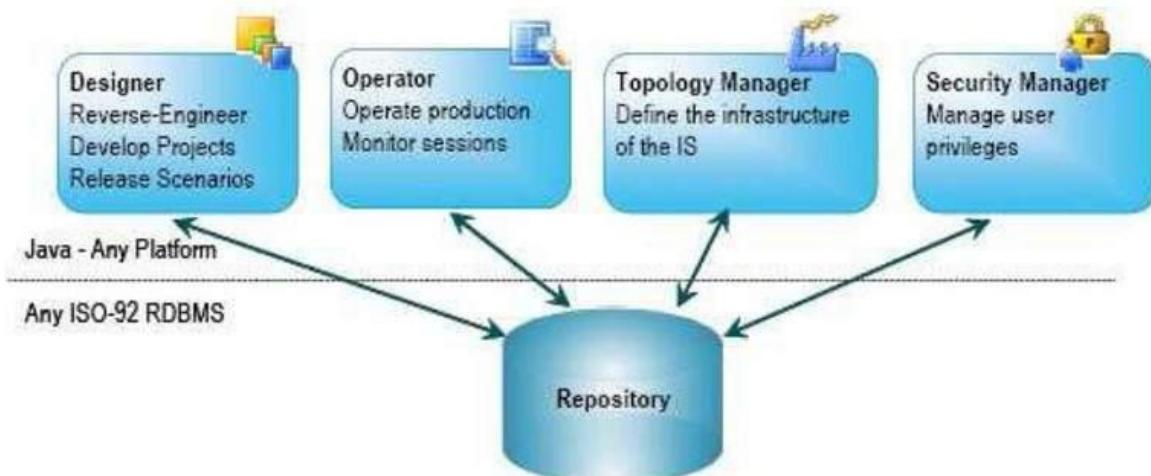
ODI 操作界面类似下图：





客户操作的图形模型包括四个组件：设计器(Designer)、操作器(Operator)、拓扑管理器 (Topology Manager) 和安全管理器 (Security Manager)，这些模型是在 ODI 安装时候就安装完整的，安装需要在 JVM1.5 的环境下，因为 ODI 是 Java 开发的产品，可以安装在 Windows、Linux、HP-UX、Solaris、AIX 和 Mac OS 平台。(参考图 1-2)

Figure 1-2 Oracle Data Integrator Component



功能模型说明

- **Designer**

定义数据转换和数据一致性的规则以及数据的过滤条件。Designer 是整个 ODI 的核心组件，所有的数据设计工作都在这个 Designer 模型中完成，数据库里的数据对象和 ODI 存储对象关系的元数据都在这里进行导入和定义。Designer 模型用数据库导入的元数据和自定义的规则产生场景(Scenarios)，最后发布到生产系统，很像我们以前编译好程序后进行打包，然后发布。这个模型主要是提供给开发人员和元数据管理员。

- **Operator**

主要用于对生产数据处理过程进行监控。这个模型主要提供给数据处理人员，并且提供执行的日志、过程中处理的行数、错误代码、执行统计信息、实际执行的代码等。Operator 在 ODI 中是非常重要组件，尤其在开发过程中为我们提供 DEBUG 的功能，可以知道我们数据处理的流程、代码出错的地方。

- **Topology Manager**

这个模型主要是用于定义 ODI 与数据库的链接串，在 ODI 中定义物理和逻辑结构。物理结构定义的是实际服务器名称、IP 地址、需要提取数据的模式、代理。

- **Security Manager**

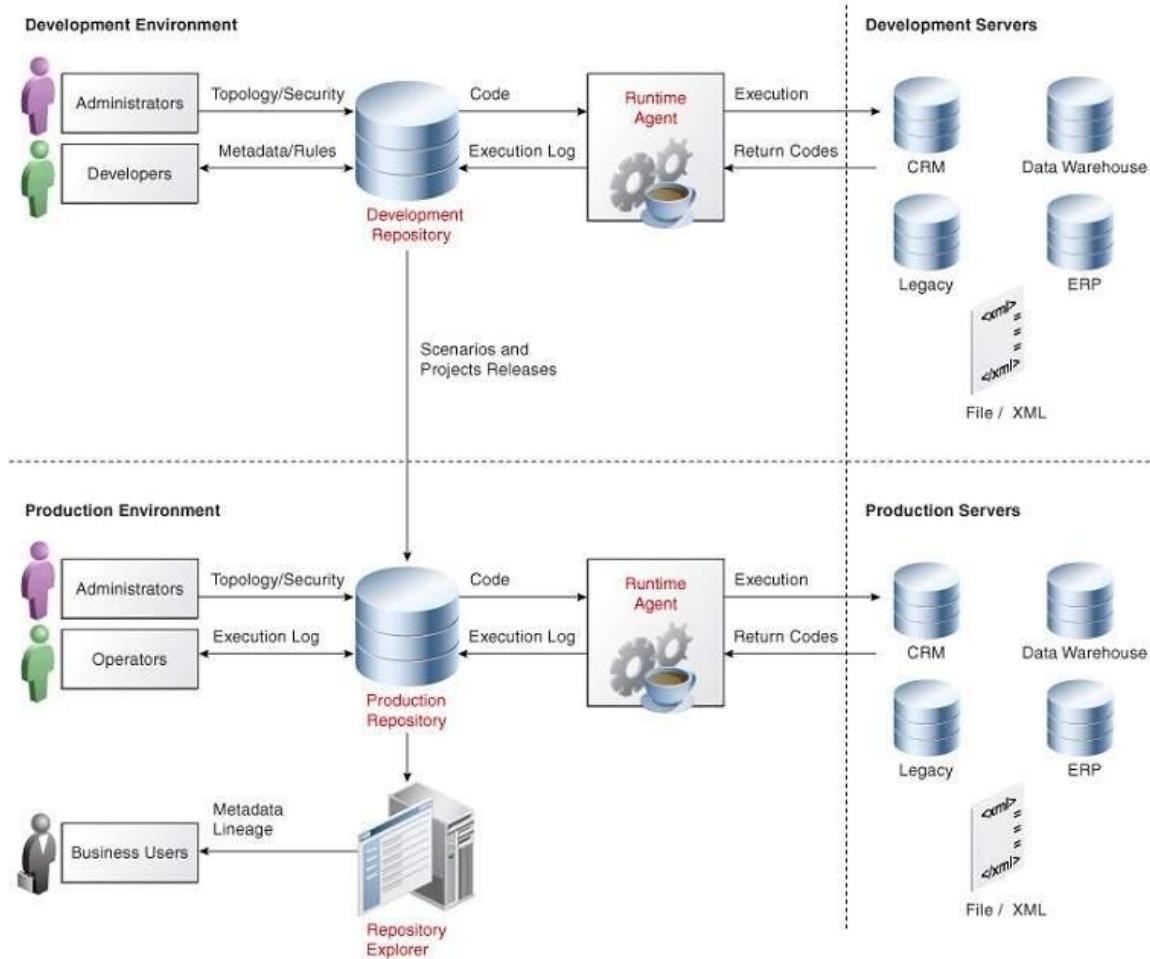
这个模型是 ODI 中用于控制 ODI 的用户权限管理，定义用户的概要文件和他们的访问权限。安全管理也给对象分配相应的权限。安全管理员会通常用这个模型。

注：所有模型定义的元数据都存储在 ODI 安装时候定义的主资料库中，通常是由一个管理人员完成 ODI 所有的配置工作。

运行组件说明

如下图 Figure1-6

Figure 1-6 Functional Architecture Overview



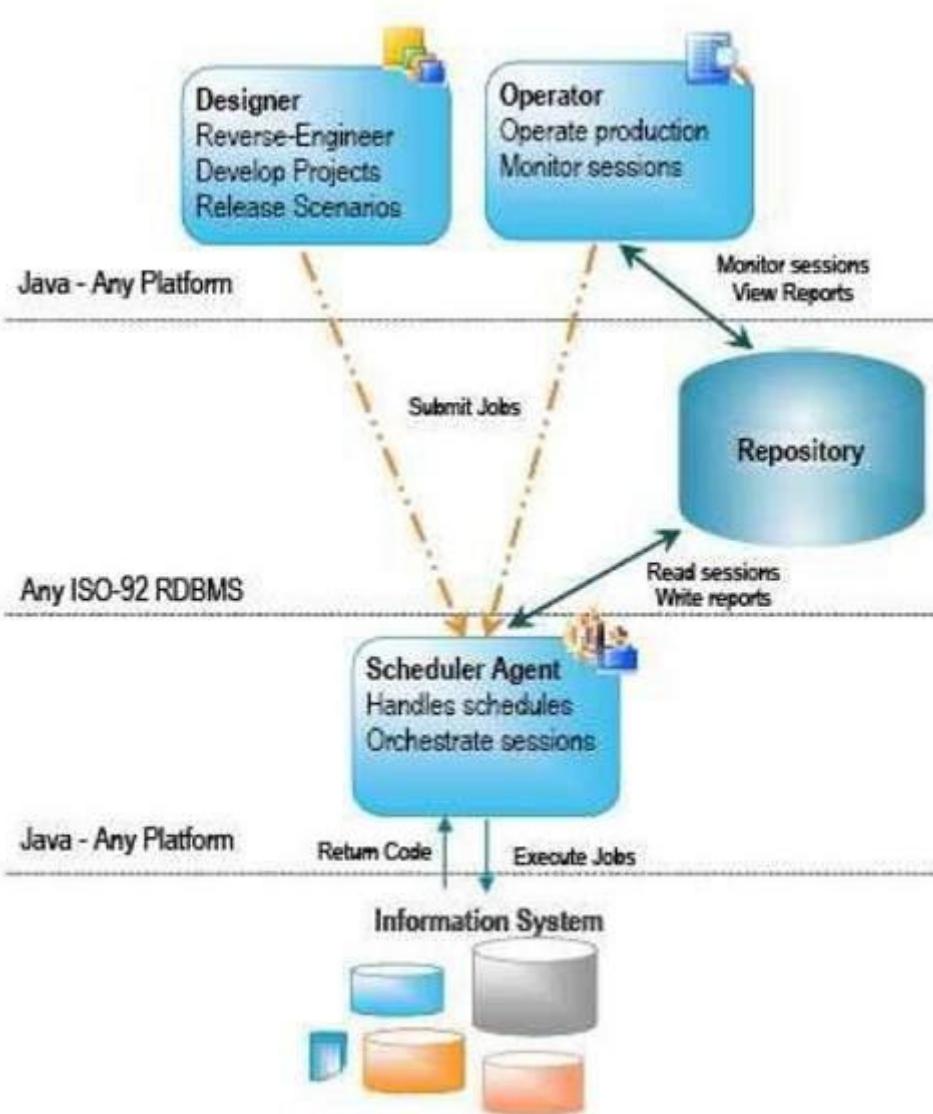
ODI 执行时，通过调度 Agent 去执行 Scenarios(打包好的项目)。这个调度 Agent 可以安装在任何支持 J2SE 的平台，Windows、Linux、HP-UX、Solaris、IBM AIX、iSeries/AS400、zSeries/OS/390 等。这个调度通常是我们从多源端的数据做提取时，安装在源端的服务器上。这些执行任务可以从任何一个界面模型发出，比如 Topology Manager 或者 Designer 、也可以创建一个调度任务来定期执行或者第三方调度程序来触发。

由于 ODI 是 Extract-Load Transformation(E-L-T)的架构，这样中间的调

度 Agent 实际上是不需要做任何中间数据转换的，而不像一些其他的 ETL 工具，需要在中间的服务器做耗时的数据转换。开发者在 ODI 的 Topology 定义实际的 JDBC 链接后，在 Designer 完成数据设计时，调度 Agent 只是从运行的资料库接受相应代码，然后要求相应的数据库服务器、操作系统、脚本引擎去执行代码。在执行完成以后，调度 Agent 会更新资料库的执行日志(Execution Logs)，并且报告错误信息和执行统计信息。开发人员或用户可以通过 Operator 或者 Metadata Navigator Web 接口来查看执行日志，判断执行的过程是否符合设计的要求或者进行诊断。

在 ODI 组件里，调度 Agent 很少作为转换引擎来使用，它更适合协调整合数据的中间流程，尤其是多个源对一个目标端，或者多个目标端与多个源做双向数据交互时，调度 Agent 就在整个数据整合过程中起到协调的作用，它会在相应的源端或目标端做调度的任务，定期执行程序。它是个多线程、负载均衡、分布式的组件。

下图就是 ODI 一个流程的架构图：

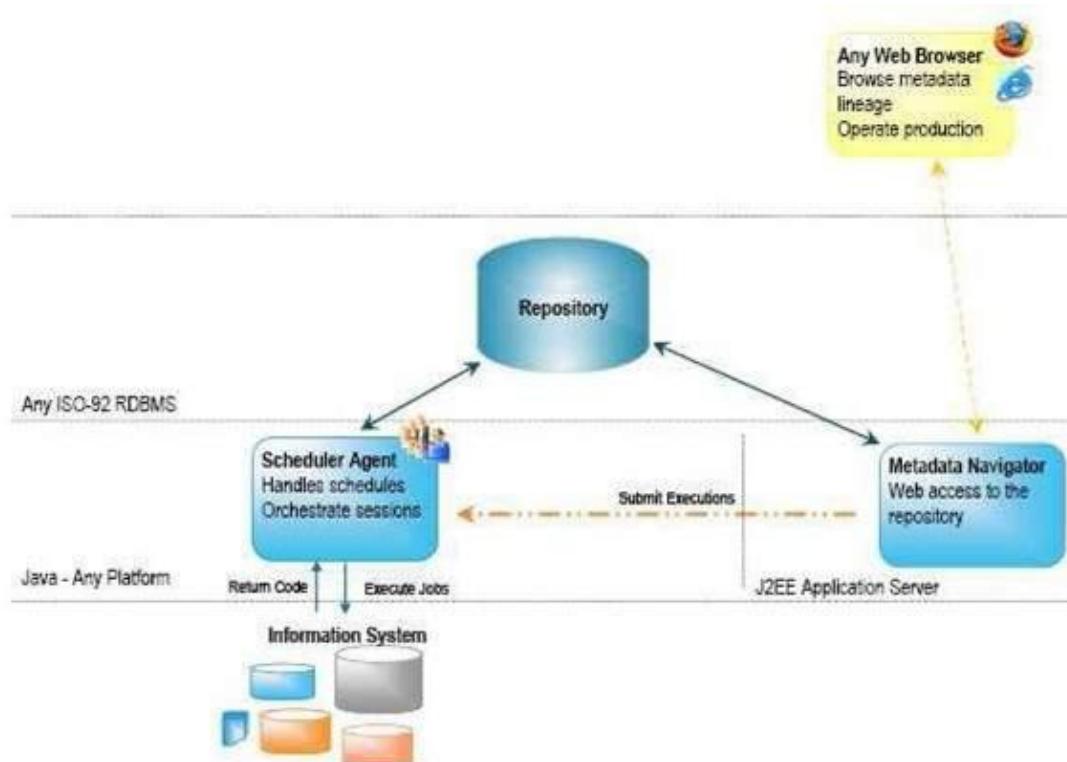


资料库说明

这个整个架构的核心组件就是资料库(Repository).资料库组成有两个部分，一个是 ODI 安装时候的主资料库，还有就是若干个工作资料库。它们存储了 IT 架构的配置信息、所有应用、项目、情景和执行日志的元数据。资料库可以安装在一个 RDBMS，也就是关系型数据库上。资料库也包含了 ODI 架构的信息。管理员、开

发人员、执行人员可以用 ODI 不同的操作界面来访问资料库。安全界面和拓扑管理器界面用于管理架构，设计器用来做元数据反向或者反向工程以及项目开发、接口等，操作器用来调度和执行实时操作。模型配置、开发或者使用的信息都是存储在这些资料库中的，这个架构里的各种组件都是通过 C/S 方式进行资料库的访问。

ODI 仅仅会有一个主资料库，包含了安全信息（用户概要文件、权限）、拓扑信息（利用哪种技术的定义和服务器信息，比如是 XML 源、数据库源还是开源技术等），还有对象版本的信息（在 ODI 里可以定义版本，类似我们项目发布的版本）。这些信息，都是通过 Topology Manager 和 Security Manager 来维护的。所有模型都有权访问主资料库，因为他们存储了 topology 和安全的信息在各自模型里。



我们在 Designer 中开发的项目对象是存储在工作资料库中的，也就是主资料库只是保存元数据信息。不同的资料库是可以同时存在一个主资料库中，也就是一个安装配置中。一个主资料库，可以对应多个工作资料库。这样的设置尤其适合项目过程中设置开发环节、质量环节和生产环节，也就是设计阶段的资料库和执行阶段的资料库是可以同时存在的。

工作资料库包含信息

● 模型

包括 Datastores, columns, data integrity constraints, cross reference, and data lineage

● 项目

包括 declarative rules, packages, procedures, folders, knowledge modules, and variables

● 运行信息

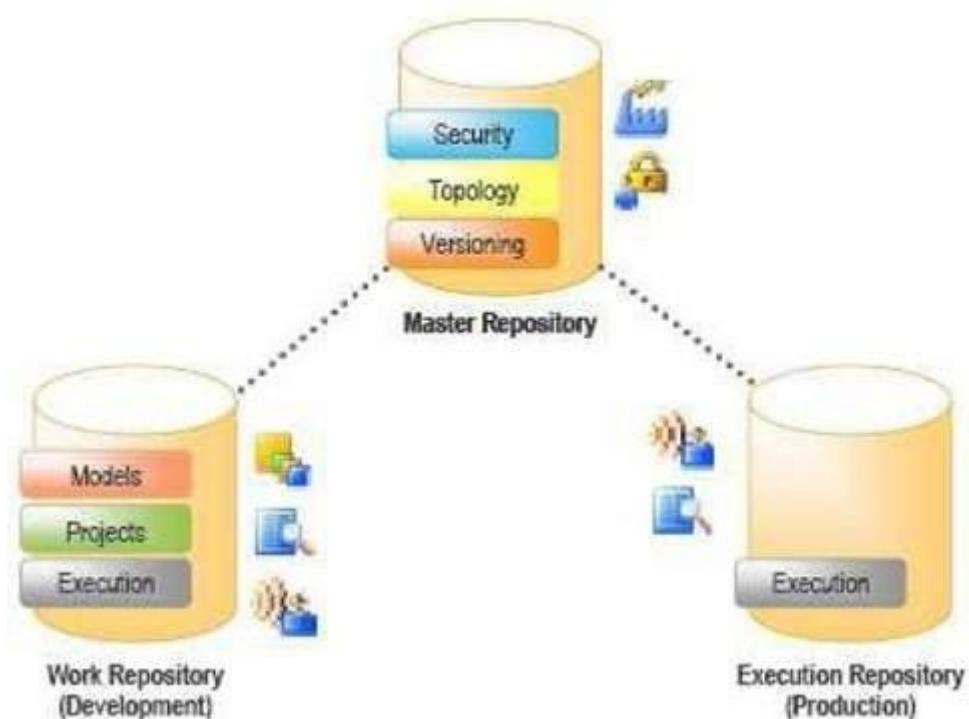
包括 including scenarios, scheduling information, and logs

开发者通过 Designer 和 Operator 模型管理工作资料库的内容。调度 Agent 在运行时也会访问工作资料库的内容。如果工作资料库只是存储执行的信息，仅仅为了做生产执行的目的，比如只执行打包好的开发程序（设计发布成 Scenarios），那么这个工作资料库就称为执行资料库。执行资料库在运行时被 Operator 和 Agent 所访问。

Metadata Navigator 说明

Metadata Navigator 是一个 J2EE 的应用程序，主要是用于通过 Web 方式访问 ODI 的资料库。用户可以浏览对象、项目、模型和执行日志。Metadata

Navigator 可以部署在类似 OC4J 或者 Tomcat 的应用服务器上。这样所有的 ODI 相关操作人员都可以通过 Web 浏览器进行元数据访问。通过 Metadata Navigator 可以看到整个数据设计的流程图、可以跟踪所有的数据源、可以钻取更细的级别查找数据转换的来源。通过 Metadata Navigator 也可以启动和监控情景 (Scenarios)。如下图：



我们的编辑团队

您若有何意见与建议，欢迎[与我们的编辑联系](#)。

诚挚感谢以下人员热情参与 TechTarget 中国《Oracle 系列电子书》的内容编辑工作！

诚邀更多的数据库专业人士加入我们的内容建设团队！



赵宇

TechTarget中国特邀技术专家。05年开始研究Oracle, 从事过电信级的数据库开发和管理工作。目前专注于ODI、GoldenGate等新产品技术，长于为客户做性能优化和问题诊断。