

Oracle® Hyperion Profitability and Cost Management

管理员指南



11.2.x
F26415-07
2023 年 10 月

ORACLE®

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

目录

文档可访问性

文档反馈

第 I 部分 Profitability and Cost Management 入门

1 概览

| | |
|--|-----|
| 关于 Profitability and Cost Management | 1-1 |
| Profitability and Cost Management 中的辅助功能 | 1-2 |
| Profitability 类型 | 1-2 |
| 标准 Profitability | 1-2 |
| 明细 Profitability | 1-2 |
| 管理分类帐 Profitability | 1-3 |
| 其他产品组件 | 1-3 |
| 管理任务 | 1-4 |
| 启动 Profitability and Cost Management | 1-4 |

2 管理安全性和授权用户

| | |
|---|------|
| 关于用户设置 | 2-1 |
| 分配安全角色 | 2-3 |
| 标准 Profitability 的安全角色 | 2-4 |
| 明细 Profitability 的安全角色 | 2-6 |
| 管理分类帐 Profitability 的安全角色 | 2-9 |
| 审核 Profitability and Cost Management 中的更改 | 2-10 |
| 输出日志文件 | 2-13 |
| Oracle Diagnostic Logging (ODL) 文件 | 2-14 |

第 II 部分 创建 Profitability and Cost Management 应用程序

3 使用 Profitability 应用程序控制台创建应用程序

| | |
|---|------|
| 显示 Profitability 应用程序控制台 | 3-1 |
| 使用 Profitability 应用程序控制台的准则 | 3-2 |
| 使用 Essbase 主多维数据集中的维创建应用程序 | 3-3 |
| 创建管理分类帐应用程序 | 3-4 |
| 使用平面文件中的维创建管理分类帐应用程序 | 3-4 |
| 通过导入模板文件创建管理分类帐应用程序 | 3-5 |
| 在 Profitability 应用程序控制台中使用应用程序 | 3-5 |
| 关于在 Profitability 应用程序控制台中使用应用程序 | 3-6 |
| 编辑应用程序说明和 Shared Services 项目 | 3-6 |
| 执行其他应用程序操作 | 3-6 |
| 更新应用程序维 | 3-8 |
| 在 Profitability 应用程序控制台的作业库中查看任务 | 3-9 |
| 为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件 | 3-11 |
| 关于为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件 | 3-11 |
| 导出模板 | 3-11 |
| 为每个管理分类帐维准备平面文件 | 3-12 |
| 关于平面文件属性 | 3-13 |
| 关于平面文件中的注释 | 3-15 |
| 平面文件示例 | 3-16 |
| 为 Profitability and Cost Management 创建 Essbase 主多维数据集 | 3-16 |
| 维类型 | 3-17 |
| 用户定义的属性 | 3-18 |
| ASO 维和成员属性 | 3-19 |
| 指定 BSO 和成员维属性 | 3-20 |
| 查看分配给成员属性的块存储选项 (BSO) 数据存储值 | 3-23 |
| 自定义度量维 | 3-23 |

第 III 部分 使用标准 Profitability

4 了解标准 Profitability 产品体系结构

5 使用标准 Profitability 的数据库

6 使用标准 Profitability and Cost Management 维和成员

| | |
|--------------------------|------|
| 关于标准 Profitability 维和元数据 | 6-1 |
| 维类型 | 6-2 |
| 度量维 | 6-3 |
| 动因度量 | 6-3 |
| 成本层分配度量 | 6-5 |
| 收入层分配度量 | 6-6 |
| 报表度量 | 6-8 |
| AllocationType 维 | 6-9 |
| 别名维 | 6-10 |
| 业务维 | 6-11 |
| POV 维 | 6-12 |
| 属性维 | 6-12 |
| 维的排序顺序和密度设置 | 6-13 |
| 维的排序顺序建议 | 6-13 |
| 设置维排序顺序 | 6-13 |

7 标准 Profitability 导入临时表

| | |
|----------------------------|------|
| 使用导入临时表 | 7-1 |
| 为标准 Profitability 创建导入数据库表 | 7-2 |
| HPM_STG_STAGE | 7-2 |
| HPM_STG_POV | 7-3 |
| HPM_STG_DRIVER | 7-4 |
| HPM_STG_DRIVER_SEL | 7-7 |
| HPM_STG_DRIVER_EXCEP | 7-8 |
| HPM_STG_ASSIGNMENT | 7-9 |
| HPM_STG_ASGN_RULE_SEL | 7-10 |

8 导出标准 Profitability 的模型定义数据

| | |
|------------------------------|-----|
| 关于导出标准 Profitability 的模型定义数据 | 8-1 |
| HPM_EXP_STAGE | 8-2 |
| HPM_EXP_POV | 8-2 |
| HPM_EXP_DRIVER | 8-3 |
| HPM_EXP_DRIVER_SEL | 8-5 |
| HPM_EXP_DRIVER_EXCEP | 8-5 |
| HPM_EXP_ASSIGNMENT | 8-6 |
| HPM_EXP_ASGN_RULE_SEL | 8-7 |

第 IV 部分 使用明细 Profitability

9 了解明细 Profitability 产品体系结构

10 使用明细 Profitability 数据库

11 产品架构

12 明细 Profitability 的模型数据架构

| | |
|----------------------------|------|
| 关于明细 Profitability 的模型数据架构 | 12-1 |
| 对产品架构的模型数据架构表授权 | 12-1 |
| 模型数据架构表结构要求 | 12-2 |
| 注册数据库视图 | 12-2 |

13 使用明细 Profitability and Cost Management 维和元数据

| | |
|--------------------------|------|
| 关于明细 Profitability 维和元数据 | 13-1 |
| 明细 Profitability 维类型 | 13-2 |
| 别名维 | 13-3 |
| 业务维 | 13-3 |
| POV 维 | 13-4 |
| 属性维 | 13-4 |
| 维的排序顺序和密度设置 | 13-5 |
| 维的排序顺序建议 | 13-5 |
| 设置维排序顺序 | 13-6 |

14 计算明细 Profitability 模型

| | |
|--------------------|------|
| 预计算和后计算自定义脚本 | 14-1 |
| HPM_SQL_SCRIPT | 14-1 |
| 创建自定义脚本 | 14-2 |
| 创建 ODBC 数据源以支持数据传输 | 14-3 |
| 高级计算选项 | 14-4 |
| 动因操作类型 | 14-4 |

| | |
|--------|------|
| 其他流程类型 | 14-4 |
|--------|------|

15 导入明细 Profitability 临时表

| | |
|----------------------------|-------|
| 关于导入明细 Profitability 临时表 | 15-1 |
| 为明细 Profitability 创建导入数据库表 | 15-2 |
| HPM_STGD_POV | 15-2 |
| HPM_STGD_DRIVER | 15-3 |
| HPM_STGD_DRIVER_SEL | 15-4 |
| HPM_STGD_DRIVER_EXCEP | 15-5 |
| HPM_STGD_ASGN_RULE_SEL | 15-6 |
| HPM_STGD_CALCRLU_SNGLSRC | 15-7 |
| HPM_STGD_CALCRLU_CALCMSRS | 15-9 |
| HPM_STGD_CALCRLU_MULTISRC | 15-10 |

16 导出明细 Profitability 的模型定义数据

| | |
|------------------------------|------|
| 关于导出明细 Profitability 的模型定义数据 | 16-1 |
| HPM_EXPD_STAGE | 16-2 |
| HPM_EXPD_POV | 16-3 |
| HPM_EXPD_DRIVER | 16-3 |
| HPM_EXPD_DRIVER_SEL | 16-4 |
| HPM_EXPD_DRIVER_EXCEP | 16-5 |
| HPM_EXPD_ASGN_RUL_SEL | 16-6 |
| HPM_EXPD_CALCRLU_SNGLSRC | 16-6 |
| HPM_EXPD_CALCRLU_CALCMSRS | 16-7 |
| HPM_EXPD_CALCRLU_MULTISRC | 16-8 |
| 为明细 Profitability 应用程序生成统计信息 | 16-9 |

第 V 部分 使用管理分类帐 Profitability

17 了解管理分类帐 Profitability 应用程序体系结构

18 了解管理分类帐维

| | |
|----------|------|
| 关于管理分类帐维 | 18-1 |
| 管理分类帐系统维 | 18-2 |
| 管理分类帐规则维 | 18-3 |
| 管理分类帐余额维 | 18-3 |

| | |
|------------|------|
| 管理分类帐业务维 | 18-4 |
| POV 维 | 18-5 |
| 属性维 | 18-6 |
| 别名维 | 18-6 |
| 管理分类帐维排序顺序 | 18-7 |
| 维的排序顺序建议 | 18-7 |
| 设置维排序顺序 | 18-7 |

19 执行其他应用程序管理任务

| | |
|--|------|
| 管理管理分类帐 Profitability 应用程序 | 19-1 |
| 使用管理分类帐应用程序和维 | 19-1 |
| Profitability and Cost Management 维和成员属性 | 19-1 |
| 设置层次类型属性 | 19-1 |
| 使用向导验证和部署管理分类帐应用程序 | 19-2 |

A 应用程序设计最佳实践

| | |
|----------------------|-----|
| 管理应用程序可扩展性 | A-1 |
| 维计数 | A-1 |
| 维成员计数 | A-1 |
| 属性和 UDA | A-2 |
| 数据规模管理 | A-2 |
| 汇总起始数据 | A-2 |
| 汇总分配中点 | A-2 |
| 管理维设计 | A-2 |
| 维类型 | A-3 |
| 存储 | A-3 |
| 动态 | A-3 |
| 启用多个层次 | A-4 |
| 层次顶级 | A-4 |
| 汇总和报告 | A-4 |
| 无成员 | A-4 |
| 最佳实践示例 | A-4 |
| 重复的成员 | A-5 |
| 管理应用程序逻辑设计 | A-5 |
| 用于管理规则计数的简单策略 | A-5 |
| 确定可以在何处使用单个规则来应用公用逻辑 | A-6 |
| 使用“缺少动因数据”作为分配筛选器 | A-6 |
| 分配规则类型和选项 | A-6 |
| 简单维扩展 | A-6 |

| | |
|----------------------|-----|
| 复杂维扩展 | A-7 |
| 简单重新归类 | A-7 |
| 复杂重新归类 | A-7 |
| 与源相同，维不同 | A-7 |
| 每个规则类型允许的分配规则和成员选择选项 | A-7 |
| 规则范围 | A-8 |
| 范围大小和稀疏度 | A-8 |
| 使用源细分 | A-8 |
| 使用筛选器或备用层次 | A-8 |
| 避免使用负的动因数据值 | A-8 |
| 避免使用动态成员作为动因 | A-9 |
| 在规则集内使用并行执行 | A-9 |
| 自定义计算规则 | A-9 |

B 将数据导入 Profitability and Cost Management

| | |
|---------|-----|
| 关于导入数据 | B-1 |
| 关于临时表 | B-1 |
| 创建导入配置 | B-2 |
| 修改导入配置 | B-4 |
| 删除导入配置 | B-5 |
| 运行导入配置 | B-5 |
| 验证导入的数据 | B-6 |

C 使用 EPM System 生命周期管理迁移数据

| | |
|-----------------------------|-----|
| 关于生命周期管理 | C-1 |
| 管理分类帐的 ApplicationData 和维对象 | C-1 |
| 修改生命周期管理的默认超时设置 | C-2 |

D 备份 Profitability and Cost Management 组件

E Essbase 命名约定

| | |
|------------------------|-----|
| 用于生成的计算脚本的命名约定 | E-1 |
| 应用程序和数据库的 Essbase 命名限制 | E-1 |
| Essbase 对维、成员和别名的命名限制 | E-2 |
| Essbase 对属性计算维的命名约定 | E-6 |

文档可访问性

有关 Oracle 对可访问性的承诺，请访问 Oracle Accessibility Program 网站 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>。

获得 Oracle 支持

购买了支持服务的 Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

文档反馈

要提供有关此文档的反馈，请单击任意 Oracle 帮助中心主题中页面底部的“反馈”按钮。还可以向 epmdoc_ww@oracle.com 发送电子邮件。

第 I 部分

Profitability and Cost Management 入门

另请参阅：

- [概览](#)
- [管理安全性和授权用户](#)

1 概览

另请参阅：

- [关于 Profitability and Cost Management](#)
Oracle Hyperion Profitability and Cost Management 是从 Oracle Hyperion Enterprise Performance Management Workspace 访问的分析应用程序。
- [Profitability and Cost Management 中的辅助功能](#)
Oracle Hyperion Profitability and Cost Management 为主要功能提供了键盘快捷键。
- [Profitability 类型](#)
Oracle Hyperion Profitability and Cost Management 提供了三种不同类型的应用程序用于评估获利能力。
- [其他产品组件](#)
通过使用产品，扩展了 Oracle Hyperion Profitability and Cost Management 的强大功能和灵活性。
- [管理任务](#)
如果您是 Oracle Hyperion Profitability and Cost Management 管理员或具有 *admin* 角色，则可以执行多个任务。
- [启动 Profitability and Cost Management](#)
只能通过 Oracle Hyperion Enterprise Performance Management Workspace 访问 Oracle Hyperion Profitability and Cost Management。

关于 Profitability and Cost Management

Oracle Hyperion Profitability and Cost Management 是从 Oracle Hyperion Enterprise Performance Management Workspace 访问的分析应用程序。

它用于准确度量、分配和管理成本及收入；计算业务部门的获利能力；以及通过使用成本分解、基于使用量的成本计算和方案模拟来度量获利能力。

Profitability and Cost Management 是 EPM Workspace 的组成部分，它使用各种组件来构建和管理其应用程序：

- EPM Workspace 用于访问 Profitability and Cost Management 以及管理其他组件来构建应用程序、控件安全性和产品报表。有关 Profitability 应用程序类型，请参阅“[Profitability 类型](#)”。
- 使用 Oracle Hyperion Shared Services 创建和管理用户帐户，包括定义安全角色以确定用户可访问哪些模型。
- 使用 Profitability 应用程序构建和维护维及成员，以用于 Profitability and Cost Management。通过维库，您可以使用其他应用程序（如 Oracle Hyperion Planning）中已存在的公共维和成员来构建应用程序。

本指南假设您将使用 Profitability 应用程序控制台来管理您的模型的应用程序和维。请参阅“[使用 Profitability 应用程序控制台创建应用程序](#)”。

- 使用 Oracle Essbase 或关系数据库来创建应用程序大纲，以及存储和执行计算脚本。

- Oracle Hyperion Enterprise Performance Management System 生命周期管理用于在不同产品环境和操作系统间迁移应用程序、多维数据库、存储库或单个对象。
- 使用 Oracle Hyperion Reporting and Analysis、Oracle Hyperion Financial Reporting 或第三方产品（如 Microsoft Excel）创建计算结果的报表。

Profitability and Cost Management 中的辅助功能

Oracle Hyperion Profitability and Cost Management 为主要功能提供了键盘快捷键。

《Oracle Hyperion Profitability and Cost Management Accessibility Guide》中介绍了辅助功能。这些功能是 Profitability and Cost Management 所独有的。

Profitability 类型

Oracle Hyperion Profitability and Cost Management 提供了三种不同类型的应用程序用于评估获利能力。

- [标准 Profitability](#)
- [明细 Profitability](#)
- [管理分类帐 Profitability](#)

标准 Profitability

标准 Profitability 侧重于贡献分析，跟踪流程的所有阶段中的成本和收入资金流，从而确定资金来源和资金流向。

在标准 Oracle Hyperion Profitability and Cost Management 模型中，您可以监视和控制整个模型的直接贡献数据。可以同时跟踪成本和收入的输入金额、成本和收入流以及资金的最终目标，从而确保资源得到最佳利用并且可以轻易地证明获利能力。计算结果将发布到各个成本中心或帐户中。

标准 Profitability 模型的数据同时存在于 Oracle Essbase 多维数据库和关系数据库中。

明细 Profitability

明细 Profitability 可将来自一个源的池或利润率对象一步分配给一个目标，以便分析获利能力。明细 Profitability 利用关系数据库进行模型对象存储、计算和报表查看。

明细 Oracle Hyperion Profitability and Cost Management 模型利用用户定义的架构组织包含现有数据的关系表和关联的查找表以扩展这些数据。明细 Profitability 模型的数据仅存在于关系数据库中。

您在 Profitability 应用程序控制台中创建该模型，然后定义您的组织中的业务维、别名和度量。在 Profitability and Cost Management 中，数据将映射到应用程序，以使您可以构建明细 Profitability 模型。该应用程序可以处理极大的数据量。

该应用程序不使用层次结构，但处理单个源和目标组合之间的流中的所有分配。分配是通过一个受限的 MeasuresDetailed 维处理的，而不是像在标准 Profitability 中那样创建 AllocationType.MembersDetailed 维包含用于处理所有分配的一组有限成员。

有关使用明细 Profitability 的详细信息，请参阅“[了解明细 Profitability 产品体系结构](#)”。

管理分类帐 Profitability

管理分类帐应用程序是为以下分析人员设计的，他们在管理报表的计算和生成方法方面具有丰富的经验，但可能并不是很熟悉 Oracle Essbase 和脚本语法或编程语言。

与标准 Profitability 应用程序类似，管理分类帐应用程序的数据也同时存在于 Essbase 多维数据库和关系数据库中。您可以在 Profitability 应用程序控制台中创建应用程序，并使用维和维成员在组织内定义帐户、活动和操作的层次。

部署应用程序后，您可以构建模型，显示特定成本和收入分配的资金流。分配的源和目标范围都使用 Profitability and Cost Management 用户界面定义为计算规则。对标准和明细 Profitability 应用程序类型而言，视点 (POV) 表示特定的模型实例，可用于查看或计算不同版本的模型；例如，查看不同月份或季度的值、比较预算与实际数据或者创建方案以度量各种更改对效益的影响。

管理分类帐模型中并无阶段或层的概念。所有结构都通过 POV 下的规则集和规则加以控制。对于每个 POV，计算规则会分成多组，在同一时间或类似时间，针对同一或类似数据库区域运行。这些组称为规则集。它们会定义计算规则的运行顺序。计算规则可以从 POV 或规则集级别继承默认成员选定项，因此用户只需定义一个数据库区域一次，随后便能多次使用该区域，而无需每次都指定。这些默认值称为“上下文”。

创建模型之后将进行验证，以确保已考虑所有分配，并已平衡计算。验证后，需要部署数据库，然后计算模型，并分析结果。有关详细信息，请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》，以及本指南的第 IV 部分“使用管理分类帐应用程序”。

其他产品组件

通过使用产品，扩展了 Oracle Hyperion Profitability and Cost Management 的强大功能和灵活性。

表 1-1 Profitability and Cost Management 产品组件

| 产品 | 说明 |
|---|---|
| Oracle Essbase | 仅对于标准 Profitability，存储和计算 Profitability and Cost Management 应用程序数据。 |
| Oracle Essbase Administration Services | 仅对于标准 Profitability，使用 Essbase 管理服务控制台连接到 Essbase 服务器的接口。 用于设计、开发、维护和管理多个 Essbase 应用程序和数据库。 |
| Profitability and Cost Management Profitability 应用程序控制台 | 管理维和应用程序。 |
| Oracle Hyperion Enterprise Performance Management Workspace | 导航到 Profitability and Cost Management 和其他产品，并对应用程序进行管理 |
| Oracle Hyperion Shared Services | 为外部系统的用户设置访问 Profitability and Cost Management 的权限 迁移 Profitability and Cost Management 应用程序 |
| Oracle Smart View for Office | 在 Microsoft Excel 电子表格中输入 Profitability and Cost Management 数据并生成报表 |
| Oracle Hyperion Financial Reporting | 创建用于 Web 分发或打印分发的报表和图表 |

管理任务

如果您是 Oracle Hyperion Profitability and Cost Management 管理员或具有 *admin* 角色，则可以执行多个任务。

- 使用 Oracle Hyperion Shared Services 创建和管理用户帐户。
- 设置用户以进行 Shared Services 身份验证。
- 在 Profitability 应用程序控制台中管理共享库和 Profitability and Cost Management 维及成员。
- 生成多维 Oracle Essbase 数据库。
- 创建、更新和删除模型阶段、动因和视点 (POV)。
- 创建、更新和删除动因选定项、分配、分配规则和分配规则的选定项。
- 创建、更新和删除计算规则。
- 计算和重新计算模型。
- 查看和修改模型数据。
- 查看跟踪分配。
- 备份和还原 Profitability and Cost Management 模型组件。
- 使用生命周期管理实用程序将应用程序从一个环境传输到另一个环境。将数据从一个环境（例如开发或测试）提升到另一个环境（例如生产）。
- 监视对业务对象所做的更改。

Profitability and Cost Management 管理员还可以通过 Oracle Hyperion Enterprise Performance Management Workspace 管理模型元数据和数据。

请参阅以下几节：

- [使用标准 Profitability and Cost Management 维和成员](#)
- [使用明细 Profitability and Cost Management 维和元数据](#)

启动 Profitability and Cost Management

只能通过 Oracle Hyperion Enterprise Performance Management Workspace 访问 Oracle Hyperion Profitability and Cost Management。

要访问 Profitability and Cost Management：

1. 确保下列应用程序已配置且处于运行状态：
 - EPM Workspace
 - Oracle Hyperion Shared Services
 - Oracle Essbase（仅适用于标准 Profitability）
 - Profitability and Cost Management

有关说明，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》和《Oracle Enterprise Performance Management System 安装入门》。

2. 在 Web 浏览器中，访问 EPM Workspace 网页。
默认情况下，URL 为 `http://server name:19000/workspace/`，其中 `server name` 是 Oracle HTTP Server (OHS) 服务器名称。
3. 输入 EPM Workspace 用户名和密码。

 注：

用户名和密码都区分大小写。

4. 单击登录。
将显示 EPM Workspace 主页。
5. 依次选择导航、应用程序和 **Profitability**，然后选择要查看的应用程序。

2

管理安全性和授权用户

另请参阅：

- [关于用户设置](#)
在使用 Oracle Hyperion Profitability and Cost Management 之前，管理员必须设置用户和组，并将适当的安全角色分配给每个用户和组。
- [分配安全角色](#)
在 Oracle Hyperion Profitability and Cost Management 中，将为每个用户 ID 分配一个安全角色。
- [审核 Profitability and Cost Management 中的更改](#)
您可以使用 Oracle Hyperion Shared Services Console 中的“审核”功能监视应用程序中的活动和更改，然后生成详细描述结果的审核报表。
- [输出日志文件](#)
管理员可以在整个 Oracle Hyperion Enterprise Performance Management Workspace 中生成日志文件。
- [Oracle Diagnostic Logging \(ODL\) 文件](#)
对于每个 Oracle Hyperion Enterprise Performance Management Workspace 产品，Web 应用程序的所有配置日志文件的 Oracle Diagnostic Logging (ODL) 位置是统一的。

关于用户设置

在使用 Oracle Hyperion Profitability and Cost Management 之前，管理员必须设置用户和组，并将适当的安全角色分配给每个用户和组。

为每个安全角色提供的授权确定用户或组可以访问哪些功能和数据。在配置期间，选择 Oracle Hyperion Shared Services 作为身份验证模式，如《*Oracle Enterprise Performance Management System 安装与配置指南*》中所述。

对于管理安全和用户设置，提供了以下两个指南：

- 对于有关安全性的技术信息，请使用《*Oracle Enterprise Performance Management System 用户安全管理指南*》：
 - SSL（单向、双向、SSL 卸载、SSL 终端）
 - 单点登录
 - 默认 EPM System SSO
 - 安全代理
 - 自定义登录
 - 自定义身份验证模块
 - EPM System 安全准则
- 有关如何设置和管理用户设置的信息，请使用《*Oracle Enterprise Performance Management System 用户安全管理指南*》：
 - Oracle Hyperion Shared Services 控制台

- 用户目录
- 应用程序和应用程序组
- 授权用户管理
- 管理 Native Directory
- 管理设置
- 设置 EPM System

可通过 Oracle Hyperion Enterprise Performance Management Workspace 为 Profitability and Cost Management 创建、维护以及设置用户和组。设置过程要求您已相应配置 Shared Services 和 Profitability and Cost Management 且两者都在运行。外部身份验证可确保应用程序间的通信无缝进行，从而可轻松准确地设置用户。

下列步骤概述了设置用户和组的过程：

1. 从 EPM Workspace 中，依次选择导航、管理和 **Shared Services Console** 以访问 Shared Services 屏幕。

 **注：**

第一次登录时，会自动为您的产品创建管理员 (*admin*) 用户。

2. 将分配有 *admin* 角色的用户设置为“设置管理员”角色。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。
3. 创建用户。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。
4. 为用户设置适当安全角色和访问所需项目和应用程序的权限。

 **注意：**

如果用户需要访问 EPM Workspace 来执行 Profitability and Cost Management 以外的任务，您还必须为选定的用户设置 EPM Workspace 角色，例如 Oracle Hyperion Enterprise Performance Management System 生命周期管理。

5. 根据需要创建组。您可以在组内设置组。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。
6. 登录到 EPM Workspace，然后打开 Profitability and Cost Management。
7. 验证此用户可以登录到 Profitability and Cost Management，并能查看设置的项目和应用程序。

有关设置用户和组的详细说明，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。

分配安全角色

在 Oracle Hyperion Profitability and Cost Management 中，将为每个用户 ID 分配一个安全角色。

- 管理员 (*admin* 是您登录到 Oracle Hyperion Shared Services 时的默认安全角色)
- 超级用户
- 交互式用户
- 查看用户

分配的安全角色确定该用户的可用访问级别或权限。可以为一个用户授予多个角色。启动特定操作时将检查该操作的权限。

注：

必须在 Oracle Hyperion Shared Services Console 中为至少一个用户手动分配“设置管理员”角色。通过“设置管理员”角色，该用户可以将安全角色分配给应用程序的其他用户。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。

某个用户必须存在且具有分配的安全角色，您才能将该用户分配给某个组。为一组用户分配访问级别时，将为该组的所有成员授予类似的安全访问权限。根据对某个特定用户的访问权限要求，可以修改分配的安全权限以附加一个更大或更小的访问权限。例如，向拥有“高级用户”安全授权的组分配“查看用户”权限，即假定该组具有较高的安全级别。

如果用户必须启动和监视任务流，则还需要附加 Shared Services 角色，如表 1 中所示。

注意：

如果用户需要访问其他产品，如需要访问 Oracle Hyperion Enterprise Performance Management Workspace 中的应用程序创建器，或者通过 Oracle Hyperion Enterprise Performance Management System 生命周期管理导入或导出临时表，则必须单独分配这些附加安全角色。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。

以下各节中详细说明的安全角色特定于 Profitability and Cost Management。有关所有安全角色的完整说明，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》，查看详细说明。

标准 Profitability 的安全角色

表 2-1 标准 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|---------------------------------|--|
| 管理员 (<i>admin</i>); 角色类型 = 超级 | <ul style="list-style-type: none">• 使用 Oracle Hyperion Shared Services 创建和维护用户帐户及安全角色，并设置用户• 生成 Oracle Essbase 数据库• 设置和维护应用程序首选项• 创建和维护模型内的元素，例如阶段、动因、POV、动因选定项、分配以及应用程序首选项。• 执行 POV 复制、计算、验证、数据输入和跟踪分配• 部署到 Essbase 并生成计算脚本 <div data-bbox="641 730 1377 949" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"><p>▲ 注意：</p><p>还必须为 Oracle Hyperion Profitability and Cost Management 管理员分配 Essbase 访问权限，才能执行 Essbase ASO 和 BSO 部署。</p></div> <ul style="list-style-type: none">• 导入和导出数据• 使用生命周期管理实用程序将数据从一个环境（例如开发或测试环境）提升到另一个环境（例如生产环境）• 备份和还原 Profitability and Cost Management 模型组件• 监视对业务对象所做的更改。• 创建、编辑、复制、删除和启动查询。• 使用 Profitability 应用程序控制台创建和管理新的 Profitability and Cost Management 应用程序。 |

表 2-1 (续) 标准 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|---------------------------------|---|
| 超级用户；角色类型 = 超级 | <ul style="list-style-type: none"> 创建和维护模型内的元素，例如阶段、动因、POV、动因选定项、分配以及应用程序首选项。 执行 POV 复制、计算、验证、数据输入和跟踪分配。 导入和导出数据 部署到 Essbase 并生成计算脚本。 <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>注意：</p> <p>还必须为 Profitability and Cost Management 超级用户分配 Essbase 访问权限（如创建应用程序和应用程序管理器），才能将数据输入到 Essbase 并执行 Essbase ASO 和 BSO 部署。</p> </div> <ul style="list-style-type: none"> 创建、编辑、复制、删除和启动查询。 <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p>注：</p> <p>超级用户执行任务并不需要有特定的安全角色。例如，如果超级用户在“计算”屏幕上运行计算，此操作在幕后创建和执行任务流。超级用户不需要有“管理任务流”的角色来执行此任务，除非超级用户想从“管理任务流”任务来直接访问此任务。</p> </div> |
| 交互式用户；角色类型 = 交互式 | <ul style="list-style-type: none"> 查看所有建模屏幕 查看和修改“数据输入”屏幕中的数据 查看跟踪分配 启动查询。 |
| 查看用户；角色类型 = 交互式 | <p>以仅查看方式访问以下功能：</p> <ul style="list-style-type: none"> 跟踪分配 应用程序首选项 模型阶段、动因和 POV |
| 管理任务流；角色类型 = Shared Services 角色 | <p>创建和编辑任务流时需要该角色。</p> <p>有关详细信息，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。</p> |
| 运行任务流；角色类型 = Shared Services 角色 | <p>设置用户只能运行和查看任务流时需要该角色。具有此角色的用户不能创建和编辑任务流。</p> <p>有关详细信息，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。</p> |

明细 Profitability 的安全角色

表 2-2 明细 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|----------------------------------|--|
| 管理员 (<i>admin</i>); 角色类型 = 管理员 | <ul style="list-style-type: none">• 设置和维护应用程序首选项• 使用 Profitability 应用程序控制台构建模型数据库• 创建和部署关系数据库的报表视图• 创建、读取（查看）、更新和删除以下功能：<ul style="list-style-type: none">– 阶段– 动因– POV– 动因关联– 分配– 应用程序首选项– 计算规则– 作业库和状态– 表注册• 执行以下任务：<ul style="list-style-type: none">– POV 复制– 验证– 部署– 计算– 停止作业• 使用生命周期管理实用程序将数据从一个环境（例如开发或测试环境）提升到另一个环境（例如生产环境）。• 导入和导出数据• 备份和还原 Oracle Hyperion Profitability and Cost Management 模型组件。• 监视对业务对象所做的更改。• 部署、更新和替换 Essbase 报表数据库，以及传输数据。 |

注意：

还必须为 Profitability and Cost Management 管理员分配 Oracle Essbase 访问权限，才能执行 Essbase ASO 部署。

表 2-2 (续) 明细 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|----------------|--|
| 超级用户；角色类型 = 超级 | <ul style="list-style-type: none">• 使用 Oracle Hyperion Shared Services 创建和维护用户帐户及安全角色，并设置用户• 创建和部署关系数据库的报表视图• 创建、读取（查看）、更新和删除以下功能：<ul style="list-style-type: none">– 阶段– 动因– POV– 动因关联– 分配– 应用程序首选项– 计算规则– 作业库和状态– 表注册• 执行以下任务：<ul style="list-style-type: none">– POV 复制– 验证– 计算– 停止作业– 部署、更新和替换 Essbase 报表数据库，以及传输数据。– 部署 |

 **注意：**

还必须为 Profitability and Cost Management 超级用户分配 Essbase 访问权限（如创建应用程序和应用程序管理器），才能将数据输入到 Essbase 并执行 Essbase ASO 部署。

 **注：**

超级用户执行任务并不需要特定的安全角色。例如，如果超级用户在“计算”屏幕上运行计算，此操作在幕后创建和执行任务流。超级用户执行此任务不需要“管理任务流”角色，除非超级用户想从“管理任务流”任务来直接访问此任务。

表 2-2 (续) 明细 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|---------------------------------|---|
| 交互式用户；角色类型 = 交互式 | 查看（读取）以下功能： <ul style="list-style-type: none">• 阶段• 动因• POV• 动因关联• 分配• 应用程序首选项• 计算规则• 作业库和状态• 表注册 |
| 查看用户；角色类型 = 交互式 | 查看（读取）以下功能： <ul style="list-style-type: none">• 阶段• 动因• POV• 动因关联• 分配• 应用程序首选项• 作业库和状态• 表注册 |
| 管理任务流；角色类型 = Shared Services 角色 | 创建和编辑任务流时需要该角色。 有关详细信息，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。 |
| 运行任务流；角色类型 = Shared Services 角色 | 设置用户只能运行和查看任务流时需要该角色。具有此角色的用户不能创建和编辑任务流。 有关详细信息，请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。 |

管理分类帐 Profitability 的安全角色

表 2-3 管理分类帐 Profitability and Cost Management 安全角色

| 安全角色 | 说明 |
|---------------------------------|---|
| 管理员 (<i>admin</i>); 角色类型 = 超级 | <ul style="list-style-type: none"> 使用 Oracle Hyperion Shared Services 创建和维护用户帐户及安全角色，并设置用户 生成 Oracle Essbase 数据库 创建和管理 Oracle Hyperion Profitability and Cost Management 应用程序 使用 Profitability 应用程序控制台创建和管理新的 Profitability and Cost Management 应用程序。 设置和维护应用程序首选项 使用 Profitability 应用程序控制台构建模型数据库，以选择公共维和成员 创建和维护模型内的元素（如 POV、规则集和规则） 执行 POV 复制、计算、验证、数据输入和跟踪分配 部署到 Essbase 和计算模型 |
| | <p>注意：</p> <p>还必须为 Profitability and Cost Management 管理员分配 Essbase 访问权限，才能执行 Essbase ASO 部署。</p> |
| 超级用户; 角色类型 = 超级 | <ul style="list-style-type: none"> 导入和导出对象 使用生命周期管理实用程序将数据从一个环境（例如开发或测试环境）提升到另一个环境（例如生产环境） 备份和还原 Profitability and Cost Management 模型组件 监控对业务对象所做的更改 创建、编辑、复制、删除和启动查询 执行模型平衡和验证 创建和维护模型内的元素（如 POV、规则集和规则） 执行 POV 复制、计算、验证、数据输入和跟踪分配 导入和导出对象 部署到 Essbase 和计算模型 创建、编辑、复制、删除和启动查询 执行模型平衡和验证 |
| 交互式用户; 角色类型 = 交互式 | <ul style="list-style-type: none"> 查看所有建模屏幕 使用规则集和规则 查看规则平衡 查看跟踪分配 定义和运行查询。 |
| 查看用户; 角色类型 = 交互式 | <p>以仅查看方式访问以下功能：</p> <ul style="list-style-type: none"> 跟踪分配 规则平衡 |

审核 Profitability and Cost Management 中的更改

您可以使用 Oracle Hyperion Shared Services Console 中的“审核”功能监视应用程序中的活动和更改，然后生成详细描述结果的审核报表。

可以使用三种审核报表：

- 安全报表
- 对象报表
- 配置报表

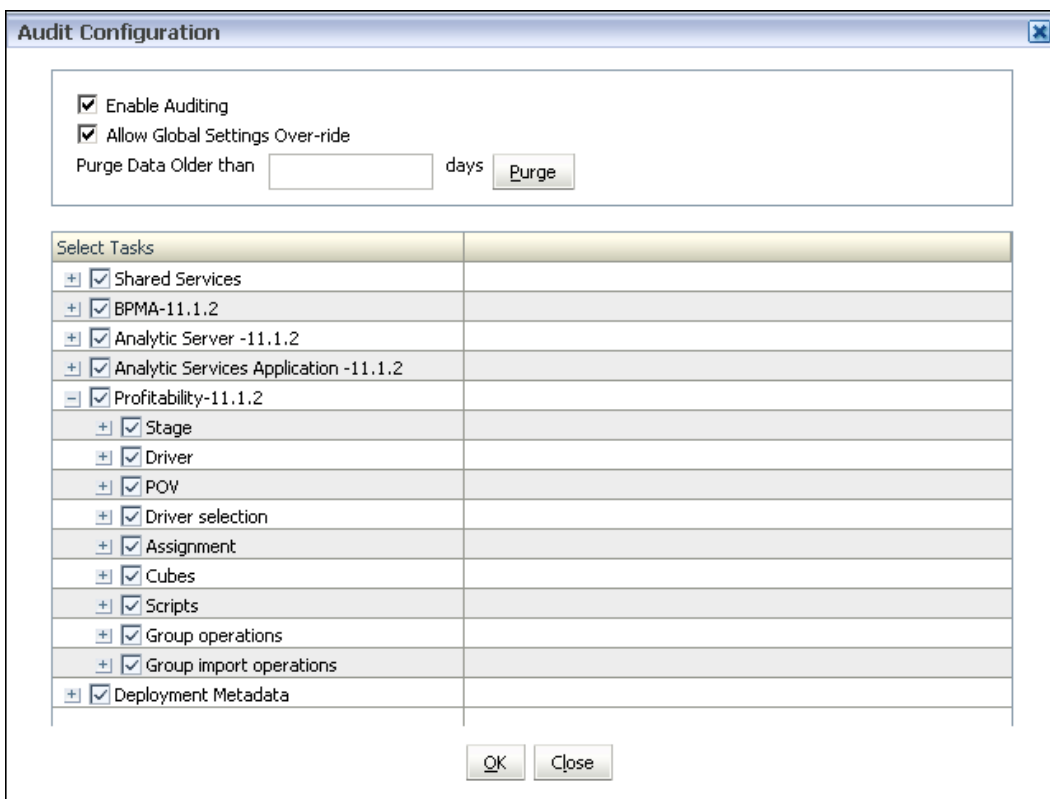
审核报表包含所选审核区域的活动详细信息，包括下列信息：

- 日期
- 应用程序
- 用户
- 对象类型和名称
- 已执行的任务

必须启用审核才能生成审核报表，如以下过程中所述。可以将这些报表导出为 CSV 文件。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。

要启用审核：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Shared Services Console**。
2. 从 Shared Services Console 中，依次选择管理和配置审核。
将显示“审核配置”屏幕。



3. 选择启用审核。

将激活允许覆盖全局设置和选择任务列表。

4. 在选择任务下，选择要启用审核的应用程序区域。您可以选择整个区域，也可以展开每个选项来选择要监视的单独步骤。

 注：

这些任务中的大多数仅适用于标准和明细 Profitability 应用程序。

表 2-4 Profitability and Cost Management 审核任务

| 监视区域 | 可用的监视任务 |
|------|--|
| 阶段 | <ul style="list-style-type: none"> • 创建阶段 • 修改阶段 • 删除阶段 • 导出阶段 • 导入阶段 |
| 动因 | <ul style="list-style-type: none"> • 创建动因 • 修改动因 • 删除动因 • 导出动因 • 导入动因 |

表 2-4 (续) Profitability and Cost Management 审核任务

| 监视区域 | 可用的监视任务 |
|---------------------------|--|
| POV | <ul style="list-style-type: none"> • 创建 POV • 修改 POV • 删除 POV • 导出 POV • 导入 POV |
| 动因选定项 | <ul style="list-style-type: none"> • 创建动因选定项 • 修改动因选定项 • 删除动因选定项 |
| 分配 | <ul style="list-style-type: none"> • 创建分配 • 修改分配 • 删除分配 |
| 分配规则 | <ul style="list-style-type: none"> • 创建分配规则 • 修改分配规则 • 删除分配规则 • 创建规则选定项 • 删除规则选定项 |
| 多维数据集 (仅标准 Profitability) | <ul style="list-style-type: none"> • 部署计算多维数据集 • 部署报告多维数据集 • 传输数据 |
| 系谱 (仅标准 Profitability) | 计算系谱 |
| 脚本 | <ul style="list-style-type: none"> • 部署分配脚本 • 部署系谱脚本 • 部署 POV 复制脚本 • 执行分配脚本 • 执行系谱脚本 • 执行 POV 复制脚本 |
| 成组操作 | <ul style="list-style-type: none"> • 复制分配 • 删除分配 • 删除分配规则的选定项 • 复制动因选定项 • 删除动因选择规则 • 删除动因选择例外 |
| 成组导入操作 | <ul style="list-style-type: none"> • 导入阶段 • 导入 POV • 导入动因 • 导入动因选定项 • 导入分配 |
| 成组导出操作 | <ul style="list-style-type: none"> • 导出阶段 • 导出 POV • 导出动因 • 导出动因选定项 • 导出分配 |

表 2-4 (续) Profitability and Cost Management 审核任务

| 监视区域 | 可用的监视任务 |
|--------------|--|
| 生命周期管理 (LCM) | <ul style="list-style-type: none"> • LCM 从文件加载 • LCM 保存到文件 • LCM 导出 • LCM 导入 |

5. 单击确定。
将显示一条消息，确认审核配置已保存。
6. 单击是。
7. 可选：生成审核报表以查看审核结果，如《Oracle Enterprise Performance Management System 用户安全管理指南》中所述。

输出日志文件

管理员可以在整个 Oracle Hyperion Enterprise Performance Management Workspace 中生成日志文件。

日志文件帮助技术人员识别系统或环境问题，或者帮助开发人员调试报表或 API 程序。

下列日志文件提供了有关 Oracle Hyperion Profitability and Cost Management 的信息：

表 2-5 Profitability and Cost Management 日志文件

| 日志文件 | 说明 |
|------------------------------------|---|
| hpcm.log | Profitability and Cost Management 会生成应用程序的服务器端日志文件，其中收集了从应用程序或服务器发送来的特定于应用程序的消息。 默认情况下，日志文件位于 C:\oracle\Middleware\user_projects\domains\EPMSsystem\servers\Profitability0\logs。 要访问此日志文件，请联系您的系统管理员。 |
| SharedServices_Security_Client.log | Oracle Hyperion Shared Services 客户端日志文件提供了有关 Profitability and Cost Management 与 Common Security Services 握手的详细信息。默认情况下，该日志文件位于 C:\oracle\Middleware\user_projects\domains\EPMSsystem\servers\Profitability0\logs。 |

要了解相关产品和应用程序的其他日志文件，请参阅《Oracle Enterprise Performance Management System 安装与配置指南》。

要修改日志文件收集信息的详细程度，请参阅《Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide》中的 "Using System Logs"。

Oracle Diagnostic Logging (ODL) 文件

对于每个 Oracle Hyperion Enterprise Performance Management Workspace 产品，Web 应用程序的所有配置日志文件的 Oracle Diagnostic Logging (ODL) 位置是统一的。

对于 Oracle Hyperion Profitability and Cost Management，配置日志文件位于：

C:\oracle\Middleware\user_projects\domains\EPMSystem\config\fmwconfig\servers\Profitability0。文件名是 logging.xml。

有关配置日志文件的其他信息，请参阅《*Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*》中的 "Using EPM Logs"。

第 II 部分

创建 Profitability and Cost Management 应用程序

另请参阅：

- [使用 Profitability 应用程序控制台创建应用程序](#)

3

使用 Profitability 应用程序控制台创建应用程序

另请参阅：



- [显示 Profitability 应用程序控制台](#)
可以从 Oracle Hyperion Enterprise Performance Management Workspace 打开 Profitability 应用程序控制台。
- [使用 Profitability 应用程序控制台的准则](#)
Profitability 应用程序控制台提供多种方式来管理 Oracle Hyperion Profitability and Cost Management 应用程序和维。
- [使用 Essbase 主多维数据集中的维创建应用程序](#)
可以使用 Essbase 主多维数据集中的维在 Profitability 应用程序控制台中创建应用程序。
- [创建管理分类帐应用程序](#)
可以使用平面文件中的维或通过导入模板文件来创建管理分类帐应用程序。
- [在 Profitability 应用程序控制台中使用应用程序](#)
使用 Profitability 应用程序控制台来使用从平面文件和模板文件创建的应用程序。
- [为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件](#)
您可以准备模板和平面文件来创建和更新管理分类帐 Profitability 应用程序。
- [为 Profitability and Cost Management 创建 Essbase 主多维数据集](#)
管理员或具有相应安全设置的其他用户可以在 Oracle Essbase 主多维数据集中创建维和维成员。

显示 Profitability 应用程序控制台

可以从 Oracle Hyperion Enterprise Performance Management Workspace 打开 Profitability 应用程序控制台。

要打开和查看 Profitability 应用程序控制台：

1. 从 EPM Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。

此时将显示 Profitability 应用程序控制台，其中显示了应用程序 () 和作业库 () 选项卡。Profitability 应用程序控制台会列出所有现有的应用程序；相应的应用程序类型；是通过主多维数据集还是文件部署的这些应用程序；以及是否已启用这些应用程序。

提示：



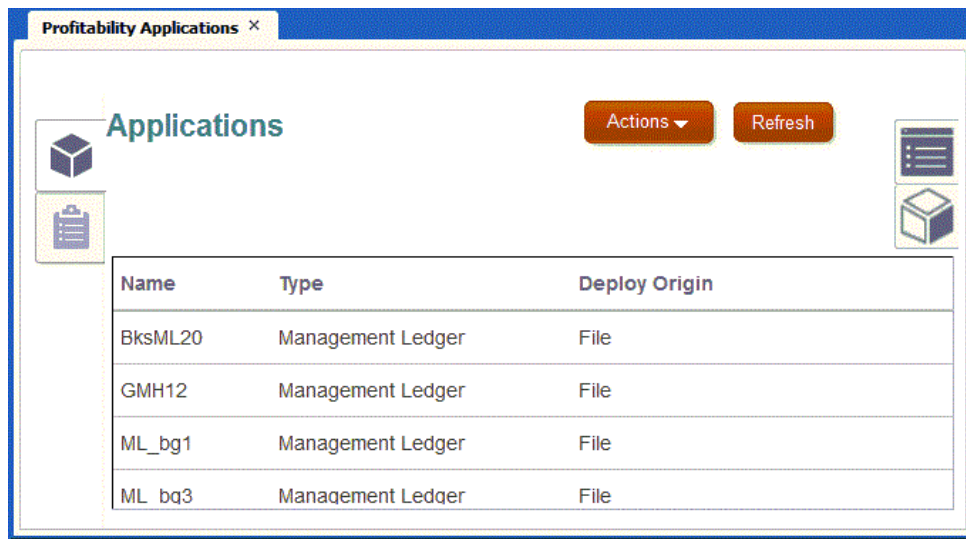
单击

可查看现有应用程序的摘要信息。单击



可查看选定应用程序中包含的维。

图 3-1 Profitability 应用程序控制台的“应用程序”选项卡



2. 选择操作以对选定的应用程序执行各种操作或创建一个新的应用程序。
请参阅“[在 Profitability 应用程序控制台中使用应用程序](#)”和“[在 Profitability 应用程序控制台的作业库中查看任务](#)”。
3. 单击刷新以更新当前的选项卡。

 注：

打开 Profitability 应用程序控制台可能需要几秒钟，在重新启动 Oracle Hyperion Profitability and Cost Management 服务之后尤其如此。使用 Chrome 和 Edge Chromium 浏览器时，这可能会导致弹出一条页面不响应的消息，以及一个等待或退出页面的选项。可以忽略此消息 – 页面最终将按预期的那样打开。

使用 Profitability 应用程序控制台的准则

Profitability 应用程序控制台提供多种方式来管理 Oracle Hyperion Profitability and Cost Management 应用程序和维。

- 在维源设置为主多维数据集的情况下，使用操作 > 新建选项创建 Profitability and Cost Management 应用程序。此类型的应用程序使用 Oracle Essbase 作为其维的源（使用 [Essbase 主多维数据集中的维创建应用程序](#)）。
- 在维源设置为文件的情况下，使用操作 > 新建选项创建应用程序。此类型的应用程序使用包含维定义的带有特殊格式的平面文件（仅适用于管理分类帐应用程序，请参阅“[使用平面文件中的维创建管理分类帐应用程序](#)”）。
- 使用操作 > 导入模板，通过模板文件（其中包含通过从现有管理分类帐应用程序导出模板而创建的维、元数据和其他对象）创建文件类型的应用程序（仅适用于管理分类帐应用程序，请参阅“[通过导入模板文件创建管理分类帐应用程序](#)”）。

 注:

只能使用原始维创建系统（如果“部署源”类型为主多维数据集，则维创建系统为 Essbase；如果“部署源”类型为文件，则维创建系统为平面文件）来修改应用程序维。请参阅：

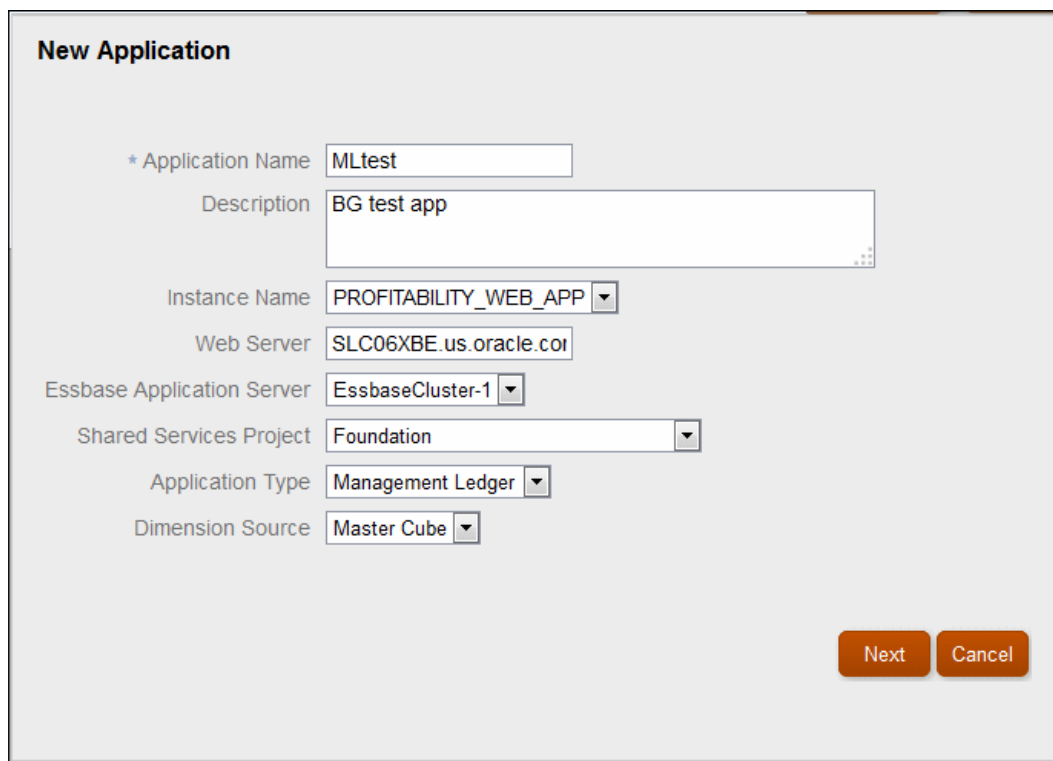
- 为 Profitability and Cost Management 创建 Essbase 主多维数据集
- 为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件

使用 Essbase 主多维数据集中的维创建应用程序

可以使用 Essbase 主多维数据集中的维在 Profitability 应用程序控制台中创建应用程序。

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。
2. 依次选择操作和新建。
此时将显示新建应用程序对话框。

图 3-2 Profitability 应用程序控制台中的“新建应用程序”对话框



The screenshot shows a 'New Application' dialog box with the following fields and values:


- * Application Name: MLtest
- Description: BG test app
- Instance Name: PROFITABILITY_WEB_APP
- Web Server: SLC06XBE.us.oracle.cor
- Essbase Application Server: EssbaseCluster-1
- Shared Services Project: Foundation
- Application Type: Management Ledger
- Dimension Source: Master Cube

At the bottom right, there are two buttons: 'Next' and 'Cancel'.

3. 在新建应用程序对话框中，输入以下信息，然后单击下一步：
 - 应用程序名称
 - 可选：应用程序的说明
 - 从下拉列表中选择实例名称

- 此时将显示 **Web 服务器**的地址
 - 应用程序的 **Essbase 应用程序服务器**
 - 应用程序的 **Shared Services 项目**
 - 应用程序类型；管理分类帐是默认值，但也可以选择标准 **Profitability** 和明细 **Profitability**
 - 维源；主多维数据集是默认值
4. 选择主多维数据集作为维源，选择要提供维的 Oracle Essbase 主多维数据集的名称，然后选择要包含在应用程序中的维。
 5. 单击完成。

在主多维数据集作为维源的情况下单击完成时，会启动以下操作：

- 将开始一个用于创建新应用程序的新任务流。可以在“作业库”选项卡  中查看结果。
- 对源 Essbase 数据库中的维进行验证。如果出现验证问题，任务流会终止并且“作业库”选项卡中会显示一条错误消息。单击错误链接可查看详细信息。请修复任何验证问题，并重复执行创建新应用程序的步骤。

 **提示：**

当您工作时，单击“作业库”选项卡中的刷新。然后，当作业成功完成时，单击“应用程序”选项卡。

验证完成后，新应用程序会添加到“应用程序”选项卡，且部署源为主多维数据集。

创建管理分类帐应用程序

可以使用平面文件中的维或通过导入模板文件来创建管理分类帐应用程序。

另请参阅：

- [使用平面文件中的维创建管理分类帐应用程序](#)
- [通过导入模板文件创建管理分类帐应用程序](#)

使用平面文件中的维创建管理分类帐应用程序

要使用平面文件中的维在 Profitability 应用程序控制台中创建管理分类帐应用程序：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability 应用程序**。
2. 依次选择操作和新建。
3. 在新建应用程序屏幕中，输入以下信息，然后选择下一步：
 - 应用程序名称
 - 可选：应用程序的说明
 - 从下拉列表中选择实例名称

- 此时将显示 **Web 服务器**的地址
 - 应用程序的 **Essbase 应用程序服务器**
 - 应用程序的 **Shared Services 项目**
 - 应用程序类型；管理分类帐是默认值，但也可以选择标准 **Profitability** 和 **明细 Profitability**
 - 维源；主多维数据集是默认值，对于管理分类帐应用程序，也可以选择文件
4. 在新建应用程序屏幕中选择文件。
 5. 输入要对新管理分类帐应用程序中的规则维和余额维使用的名称。单击**完成**。该应用程序将添加到应用程序选项卡，且部署源为文件。
 6. 执行“[更新应用程序维](#)”中的步骤，以将其他维添加到应用程序。
 7. 维完成后，使用验证并启用选项，以便可以添加规则（请参阅“[执行其他应用程序操作](#)”）。

通过导入模板文件创建管理分类帐应用程序

“[导出模板](#)”介绍了如何将管理分类帐应用程序导出到模板文件以便进行应用程序迁移和备份。本节介绍了如何导入这些文件，以使用之前从其他应用程序导出的应用程序元数据、维元数据和模型对象来创建新的应用程序。


要使用模板文件创建管理分类帐应用程序：

1. 导出模板文件，如“[导出模板](#)”中所述。
2. 在 Profitability 应用程序控制台的“应用程序”选项卡上，依次选择操作和导入模板。
3. 在服务器或本地计算机上为模板文件选择一个位置。

注：

必须先将从服务器加载的文件复制到 import_export 文件夹。请注意，此文件夹与用于 LCM（生命周期管理）的文件夹相同，如 <MIDDLEWARE_HOME/user_projects/epmsystem1/import_export。

4. 浏览以选择具有 .zip 扩展名的文件。
如果该文件包含一个输入数据文件夹，您可以选中导入输入数据，以将其包含在导入内容中。
5. 单击下一步。
如果文件采用了有效格式，则导入将开始。否则，将显示错误消息。

您可以依次单击“作业库”选项卡  和刷新来检查导入状态。

在 Profitability 应用程序控制台中使用应用程序

使用 Profitability 应用程序控制台来使用从平面文件和模板文件创建的应用程序。

另请参阅：

- [关于在 Profitability 应用程序控制台中使用应用程序](#)

- [编辑应用程序说明和 Shared Services 项目](#)
- [执行其他应用程序操作](#)
- [更新应用程序维](#)
- [在 Profitability 应用程序控制台的作业库中查看任务](#)

关于在 Profitability 应用程序控制台中使用应用程序

Profitability 应用程序控制台的应用程序选项卡列出了从平面文件和模板文件（类型为文件的部署源）以及 Oracle Essbase 主多维数据集创建的 Oracle Hyperion Profitability and Cost Management 应用程序。

您可以编辑选定应用程序的说明和 **Shared Services** 项目（请参阅[“编辑应用程序说明和 Shared Services 项目”](#)）。

对于在 Profitability 应用程序控制台中创建的应用程序，您可以使用操作菜单来执行以下任务：删除、复制、验证并启用、重新注册、元数据验证和更新维。对于管理分类帐应用程序，您还可以导入模板和导出模板。有关详细信息，请参阅本节开头列出的主题。


编辑应用程序说明和 Shared Services 项目

只有应用程序的说明和 Oracle Hyperion Shared Services 项目可以在“概览”窗口中修改。如果要更改与显示在“概览”窗口中的应用程序相关的其他任何信息，您必须创建一个新的应用程序。

要编辑应用程序：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。

此时将打开 Profitability 应用程序控制台，其中显示了应用程序和作业库选项卡。

2. 选择要修改的应用程序，然后单击概览按钮 。
3. 根据需要查看和修改信息：
 - 如果需要，修改说明。
 - 从下拉列表中选择一个不同的 **Shared Services** 项目。
4. 单击保存。

要重新注册应用程序，请参阅[“执行其他应用程序操作”](#)。

执行其他应用程序操作

对于在 Profitability 应用程序控制台中创建的应用程序，您可以在操作菜单中执行以下任务：删除、复制、验证并启用、重新注册、元数据验证和更新维。请参阅后面步骤 3 中复制下的“注意”。

 **注：**


对于管理分类帐应用程序，您还可以导出和导入模板文件（“[导出模板](#)”和“[通过导入模板文件创建管理分类帐应用程序](#)”）

要在 Profitability 应用程序控制台中执行应用程序操作：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。
2. 在应用程序选项卡上，选择目标应用程序。
3. 选择操作，然后选择以下选项之一：
 - 删除 - 删除选定应用程序
在删除此应用程序之前，请确保没有其他用户需要它。
 - 复制 - 复制选定应用程序
系统将要求您为新应用程序提供一个名称。

 **注：**

在 Profitability 应用程序控制台中复制某个应用程序时，会复制相应的维和成员并部署该应用程序。要将其他对象（如 POV、计算规则等）添加到标准和明细 Profitability 应用程序，必须从源系统导出这些对象，然后使用生命周期管理 (LCM) 将其导入新应用程序（[使用 EPM System 生命周期管理迁移数据](#)）。对于管理分类帐应用程序，您可以导出模板文件，然后将其导入新应用程序（包括输入数据，请参阅“[为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件](#)”）。

- 验证并启用 - 验证选定应用程序，如果有效，则启用它
结果显示在“作业库”选项卡 () 上。如果发生验证错误，请单击链接了解详细信息。可以创建记录了错误的文件以方便进行更正。
 - 重新注册 - 刷新选定应用程序的 Oracle Hyperion Shared Services 注册
 - 元数据验证 - 对所选应用程序的部署数据运行跨维验证，但不启用有效的应用程序
选择验证并启用以启用有效的应用程序。
 - 更新维 - 为所选应用程序添加或删除维成员
有关详细信息，请参阅“[更新应用程序维](#)”。
4. 对任何确认提示进行答复。

 **提示：**

可以查看和刷新作业库选项卡来检查 Profitability 应用程序控制台的当前状态（在 Profitability 应用程序控制台的作业库中查看任务）。

更新应用程序维

如果在“部署源”为主多维数据集的应用程序中需要对维或维成员进行任何更改（如添加、删除、更改或重命名），您必须先更新 Oracle Essbase 主多维数据集中的维，然后再更新 Oracle Hyperion Profitability and Cost Management 应用程序。

▲ 注意：

如果需要重命名或删除 Essbase 主多维数据集中的维，不妨考虑创建一个单独的 Essbase 主多维数据集，以便现有 Profitability and Cost Management 应用程序在必要时仍可以更新。此外，如果 Essbase 主多维数据集数据库的名称与基于该数据库的 Profitability and Cost Management 应用程序的名称不同，则您无法使用 Profitability 应用程序控制台来更新维。

每当使用 Profitability 应用程序控制台更新维时，都将自动在作业库中创建一个作业条目以验证和更新应用程序。

要更新在 Profitability 应用程序控制台中创建的 Profitability 应用程序中的 Essbase 维：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。
2. 在应用程序选项卡中，选择包含要更新的维和维成员的应用程序。
3. 依次选择操作和更新维。

通常，您将执行预更新分析，然后选择更新维。

4. 可选：在预更新分析下，选择验证维，以验证是否每项选定的更改都有效。此选项不会更新维。如果已选择了所有维，则此选项还将运行应用程序级别的维验证。

验证错误和影响分析的结果将显示在“作业库”中。

5. 可选：在预更新分析下，选择影响分析，以对正在部署的模型与 Profitability and Cost Management 中的现有模型进行比较。如果选中影响分析，则验证维选项将自动选中。

此时将显示以下信息：

- 新成员
- 删除的成员
- 重新设定父代的成员
- 发生了 Level0 更改的成员
- 对引用方模型对象的影响

验证错误和影响分析的结果将显示在“作业库”中。

6. 可选：选中更新维以使用更改更新选定维。
7. 在维下，执行以下操作之一：
 - 如果正在更新部署源为主多维数据集的应用程序，请选择要更新的维。

 注:

如果之前已包含未选择的维成员，将省略或删除这些成员。

- 如果正在使用平面文件添加或更新部署源为文件的管理分类帐应用程序中的维，请浏览以便为要更新的维选择平面文件。
8. 单击确定。任何异常都会报告在作业库中。
- 如果选择了预更新分析，则修改后的维会经过验证，但不会更新。
- 如果选择了更新维，则会根据维的选择来添加、更新或删除选定的维成员，或者导入选定的平面文件。

 注:

如果正在使用平面文件进行更新，请重复必要的步骤，直到已导入所有平面文件。务必使用验证并启用选项来确保正确导入了文件。

在 Profitability 应用程序控制台的作业库中查看任务

“作业库”选项卡提供在 Profitability 应用程序控制台中创建的所有作业的列表。

要查看作业库：


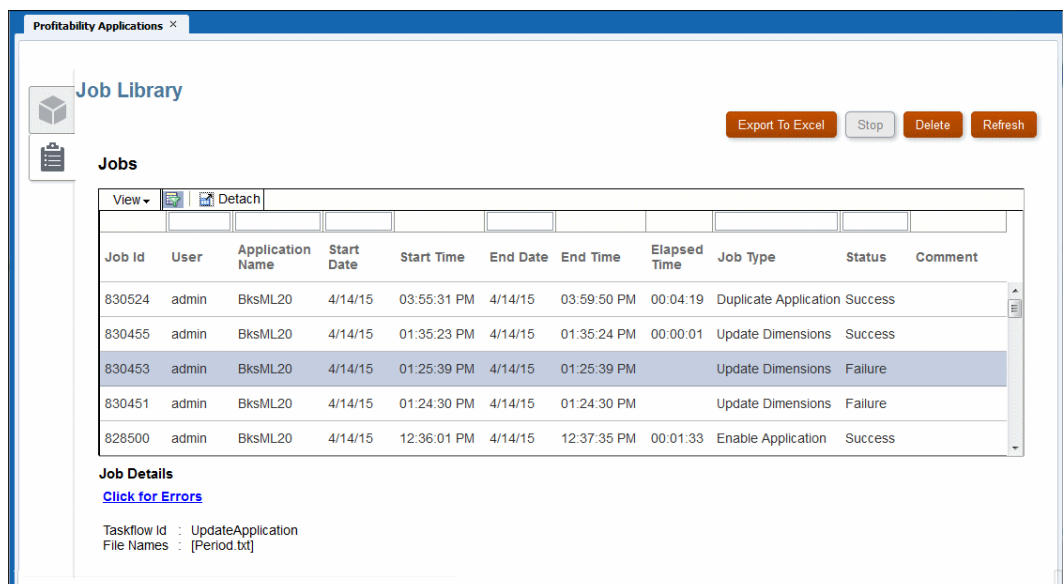


1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和 **Profitability** 应用程序。
2. 选择作业库选项卡 ()。

图 3-3 Profitability 应用程序控制台的“作业库”屏幕



作业库屏幕包含下列控件：

- 导出到 **Excel** 按钮 - 将作业库表保存到 Microsoft Excel 文件
- 停止按钮 - 取消选定作业
- 删除按钮 - 从作业库表中删除选定作业
- 刷新按钮 - 使用最新作业信息更新作业库表
- 查看菜单 - 隐藏、显示各列及重新排列各列的顺序；分离和重新附加作业表；在每一列顶部显示和隐藏“按示例查询”框
- 筛选器按钮 - ，在每一列顶部显示和隐藏“按示例查询”框
在“按示例查询”框中输入要匹配的文本，以在作业库表中选择特定条目。
- 分离按钮 - ，将作业表移到自己的窗口中；再次单击可将其附加进来。

3. 查看每个作业的信息：

- 作业 ID 是系统分配的序列标识号。
- 用户标识提交任务进行处理的个人的用户名。
- 应用程序名称显示正在运行任务的应用程序的名称。
- 开始日期/时间显示作业的提交日期和时间或者作业调度运行的日期和时间。
- 结束日期/时间显示作业停止的日期和时间，无论成功与否。
- 所用时间是开始时间和结束时间之间的差值。
- 作业类型显示要执行的任务类型，例如“创建应用程序”或“更新维”。
- 状态显示作业的当前状态，如“正在运行”、“成功”或“失败”。
- 注释显示用户输入的有关特定作业的备注或详细信息。注释是在提交任务时输入的。

4. 可选：在作业详细信息下，查看有关选定作业的其他摘要级详细信息。作业详细信息包括任务流 ID，并可能包括单击即可查看更多详细信息（例如错误或影响分析信息）的超链接。如果提供了超链接，您可以将该信息保存到文件中。

 注：

任务流 ID 是系统为特定任务生成的任务 ID，显示格式为 <应用程序名称>:<任务名称><生成的任务流编号>。

例如，生成的任务流编号可能会显示为

AppMgmt_DeployApplication_D20120824T08520_5ed，其中 **AppMgmt** 是应用程序名称，**DeployApplication** 是任务，**D20120824T08520_5ed** 是生成的任务流实例 ID。

- 5. 可选：如果您使用了查看选项或筛选器按钮来显示“按示例查询”框，可在每一列顶部的文本框中输入要匹配的文字来查找作业。
- 6. 可选：如果需要，单击停止结束处于“正在运行”状态的选定任务。

 注：

虽然按下按钮后任务流会快速停止，但影响结果的活动可能需要更多时间才能确保数据的状态一致。

 注：

已打开的应用程序内的“作业库”屏幕和 Profitability 应用程序控制台中的“作业库”屏幕都可用于将作业历史记录导出到 Excel 电子表格文件。导出的文件以 .html 格式创建。如果您在保存所导出的文件时为其提供 .html 后缀，则可以在任意浏览器、MS Word、MS Excel 或其他能够处理 .html 文件的程序中打开该文件。

为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件

您可以准备模板和平面文件来创建和更新管理分类帐 Profitability 应用程序。

另请参阅：

- [关于为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件](#)
- [导出模板](#)
- [为每个管理分类帐维准备平面文件](#)
- [关于平面文件属性](#)
- [关于平面文件中的注释](#)
- [平面文件示例](#)

关于为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件

您可以使用 Profitability 应用程序控制台，通过导入一个导出的模板来创建管理分类帐应用程序。此外，您也可以通过导入平面文件来添加或更新“部署源”为“文件”类型的管理分类帐应用程序中的维。

“[为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件](#)”中所列的主题介绍了如何为了创建和更新管理分类帐应用程序而准备模板和平面文件。

导出模板


模板文件是通过以下方式创建的：在单个操作中导出整个管理分类帐应用程序（包括应用程序元数据、维元数据和程序对象）以创建一个应用程序“模板”文件。导出和导入模板文件操作非常适用于对应用程序进行备份或将其迁移到其他环境（例如从测试环境到生产环境）。

要导出模板文件：

1. 显示 Profitability 应用程序控制台，并选择管理分类帐应用程序。

2. 依次选择操作和导出模板。
3. 查看文件名，并根据需要更改它。将为导出的模板文件添加扩展名 .zip。
4. 如果显示了 POV 数据，请选择是导出所有的 POV 数据还是清除不应导出的数据的复选框。
5. 可选：选中包括输入数据以导出输入数据。选中后，会导出所有选定 POV 的输入数据。
6. 单击确定以开始导出。

系统会在服务器上的 import_export 文件夹中创建相应的文件。请注意，此文件夹与用于 LCM（生命周期管理）的文件夹相同，如 <MIDDLEWARE_HOME>/user_projects/epmsystem1/import_export。

7. 您可以依次单击“作业库”选项卡  和刷新以检查导出状态。

创建后，便可导入模板文件来创建一个新的管理分类帐应用程序，如“[通过导入模板文件创建管理分类帐应用程序](#)”中所述。

为每个管理分类帐维准备平面文件

您可以使用维平面文件来更新管理分类帐应用程序以及创建此类应用程序。如果该文件中缺少现有成员，则在更新过程中会删除这些现有成员。有关详细信息，请参阅本节开头列出的主题。

要使用平面文件在管理分类帐应用程序中添加或替换维，请按如下方式准备平面文件：

1. 在文本编辑器中，按照本节介绍的格式为每个维创建一个平面文件。每个平面文件都必须完整地定义维。维平面文件是包含以下内容的文本文件：
 - 标题记录，用以指定维和成员属性的顺序
 - 包含属性值的维数据记录
 - 包含属性值的单个成员数据记录

注：

由于成员条目的顺序需要与最终大纲相同，因此必须先定义父代成员，再定义其子成员。文件中成员的顺序将与在 Oracle Hyperion Profitability and Cost Management 中显示维树的顺序相同（例如，在成员选择器中）。

2. 按照以下规则创建维和成员标题记录：
 - 可以按任何顺序指定属性（以逗号分隔）。多值属性的每个值必须用引号括起来。属性有多个值时需要使用引号。例如，用户定义属性 (UDA) 的给定成员可以有多个 UDA 字符串。用引号将这些成员括起来表示它们是一个集合，例如：
`, "myUDA1, myUDA2, myUDA3",`
 - 多值属性（例如属性名称或 UDA）中使用的逗号始终解释为值分隔符。只要值用引号括起来，单值属性就可以包含逗号（例如 "my,value"）。
 - 要在值中放置双引号，请用另一组引号将带双引号的值括起来。例如，要输入值 "myValue"，请按如下所示指定该值："`"myValue"`"

 注：

为了获得最佳效果，请避免使用需要双引号的成员名称。

- 属性名称不区分大小写，例如下列名称视为相同名称：name、NAME、Name。
- 并非平面文件中的每一行都需要提供所有属性。属性可以忽略或为空，这可通过在逗号之间通常列出属性的位置留空来指示，例如 ,,
例如，在成员记录中，标题中的任何维属性都会被忽略。
- 对于必需的属性，没有值或值无效的属性将采用默认值（请参阅[“关于平面文件属性”](#)）。
- 支持使用注释（请参阅[“关于平面文件中的注释”](#)）。
- 规则维和余额维的初始设置方式与从主多维数据集部署时的初始设置方式相同。您无需为这些维导入文件，系统会自动创建相应的成员。您不能为任何“规则”维成员指定别名。“规则”和“余额”维成员的维成员名称以编程方式定义，并会进行转换。
- 维排序和解析的先后顺序如下：根据维名称的字母顺序进行处理，但是属性维始终在最后处理（这些维也按字母顺序处理）。如果使用了“成员解析顺序”属性，它优先于其他考虑因素。

关于平面文件属性

[表 1](#) 介绍了维平面文件中每个属性的格式。有关平面文件的示例，请参阅[“平面文件示例”](#)。

可以包括成员的 Essbase“成员解析顺序”属性，但是必须将其添加到平面文件的第一行以显示其位置。

 注：

维名称在任何注释行之后的第一行中提供。

表 3-1 在维平面文件中定义的属性

| 属性名称 | 属性类型 | 维类型 | 默认值 | 说明 |
|-----------|------|--------------------|------|---|
| 维类型 | 维 | 全部 | 常规 | 标识维类型。有效示例如下：帐户、期间、年份、版本、方案、POV1、POV2、POV3、POV4、维、属性、规则。如果在此列表中没有属性名称，则默认值将为“常规”。此位置表示维名称（如果为 Gen1 行）或成员名称。 POV 顺序如下：年份/ POV1 = 1，期间/ POV2 = 2，方案/ POV3 = 3，版本/ POV4 = 4 |
| 存储类型 | 维 | 常规、帐户、实体、POV | 稀疏 | 稀疏和密集 |
| 维名称（作为属性） | 维、成员 | 常规 | 无 | 在标题中提供属性维名称时，该位置表示要加载的维的关联属性维。该位置中的任何字符串都将被视为“属性成员”。例如，假定在标题中指定“我的属性维”。在该文件的数据行中，同一位置中的字符串将假定为该属性维的成员。因此，可以在“所有产品”的数据行上指定“我的属性 Member1”，且加载程序会将该成员作为属性关联分配给“所有产品”。 |
| 注释 | 维、成员 | 常规、帐户、实体、国家/地区、POV | 无 | |
| 层次类型 | 维、成员 | 常规、帐户、实体、国家/地区、POV | 已存储 | 已启用、已存储、动态、已禁用 |
| BSO 数据存储 | 维、成员 | 常规、帐户、实体、国家/地区、POV | 从不共享 | 仅标签、存储、从不共享、共享、动态计算和存储、动态计算 |
| ASO 数据存储 | 维、成员 | 常规、帐户、实体、国家/地区、POV | 从不共享 | 仅标签、存储、从不共享、共享 |
| 两遍计算 | 维、成员 | 常规、帐户、实体、国家/地区、POV | N | True 或 False 是可接受的值（N 或 Y）。 |
| ASO 维公式 | 维、成员 | 常规、帐户、实体、国家/地区、POV | 无 | |

表 3-1 (续) 在维平面文件中定义的属性

| 属性名称 | 属性类型 | 维类型 | 默认值 | 说明 |
|--------|------|---------------------------|-----|---|
| 成员解析顺序 | 成员 | 常规、帐户、实体、 国家/地区 | 无 | 在 Oracle Essbase 中，解析序号决定了在维中计算成员的顺序。可以输入一个介于 1 和 127 之间的数字。首先计算解析序号最高的成员（例如，解析序号为 20 的公式在解析序号为 5 的公式之前进行计算）。 具有相同解析序号的成员按照其维出现在数据库大纲中的顺序进行计算。没有解析序号的成员在具有解析序号的所有成员之后进行计算。 |
| 合并类型 | 成员 | 常规、帐户、实体、 国家/地区 | 未使用 | 加、减、乘、除、百分比、忽略、未使用 |
| UDA | 维、成员 | 常规、帐户、实体、 国家/地区 | 无 | 单个文本值或文本值列表（用逗号括起来）。对于维，值是类型 = UDA 的维名称。对于成员，它是指定 UDA 维的成员。例如： myUda1 "myUda1,myUda2" |
| 父代 | 成员 | 常规、帐户、实体、 国家/地区、POV、属性 | | 标识父代成员名称。如果为空，则它表示成员是层代 2。顺序很重要；之前必须已定义所引用的父代。 |
| 别名：别名表 | 维、成员 | 全部 | | 示例："Alias: Default"、"Alias: T1" |
| 说明 | 维、成员 | 全部 | | 可选 - 无默认值。 |

关于平面文件中的注释

对于单行注释，放置井号字符作为该行的第一个字符。将忽略空白行。

块注释由开始注释块指示符 (#!) 描述并在一个单独的行上以结束块指示符 (#!) 终止。中间的行不需要进行注释。

例如：

```
#!—start of comment block

Comment within block

Another comment within block

#--!
```

平面文件示例

图 1 显示了 Customers 维的示例文件。

图 3-4 Customers.txt 管理分类帐平面文件示例

```
Generic,Storage Type,Hierarchy Type,Attributes Header,comment,bsc data storage,aso data storage,
two pass calculation,aso dimension formula,consolidation type,uda,parent,alias:Default,alias:English
Customers,SPARSE,STORED,,,LABELONLY,STOREDATA,N,,,UDA,,alias:Default,alias:English
NoCustomer,SPARSE,Disabled,,,StoreData,StoreData,N,+,,,No Customer,No Customer
AllCustomers,SPARSE,Disabled,,,StoreData,StoreData,N,+,,,,All Customers,All Customers
Big Box,SPARSE,,,StoreData,StoreData,N,+,,,AllCustomers,,
BB100,SPARSE,,,StoreData,StoreData,N,+,,,Big Box,Q Mart,Q Mart
BB200,SPARSE,,,StoreData,StoreData,N,+,,,Big Box,Bike Depot,Bike Depot
BB300,SPARSE,,,StoreData,StoreData,N,+,,,Big Box,Mountain Adventures,Mountain Adventures
Specialty Retailers,SPARSE,,,StoreData,StoreData,N,+,,,AllCustomers,,
SR100,SPARSE,,,StoreData,StoreData,N,+,,,Specialty Retailers,Bobs Bikes,Bobs Bikes
SR200,SPARSE,,,StoreData,StoreData,N,+,,,Specialty Retailers,Rose Town Bikes,Rose Town Bikes
SR300,SPARSE,,,StoreData,StoreData,N,+,,,Specialty Retailers,The Cyclery,The Cyclery
Webstore,SPARSE,,,StoreData,StoreData,N,+,,,AllCustomers,,
```

为 Profitability and Cost Management 创建 Essbase 主多维数据集

管理员或具有相应安全设置的其他用户可以在 Oracle Essbase 主多维数据集中创建维和维成员。

这些维和维成员随后在 Profitability 应用程序控制台中导入 Oracle Hyperion Profitability and Cost Management 应用程序。主多维数据集是一个 ASO 应用程序。同一个主多维数据集可用于创建多个 Profitability and Cost Management 应用程序。应用程序类型（“标准”、“明细”或“管理分类帐”）是创建应用程序时在控制台中设置的，而并非来自 Essbase 主多维数据集。

本节介绍了如何创建要用于 Profitability 应用程序控制台的 Essbase 主多维数据集。要使用平面文件，请参阅“[为了创建和更新管理分类帐 Profitability 应用程序而准备模板和平面文件](#)”。

要创建 Essbase 主多维数据集：

1. 为您的 Profitability and Cost Management 应用程序创建 Essbase 主数据库，请使用以下准则生成应用程序的维和成员：
 - [维类型](#)
 - [ASO 维和成员属性](#)
 - [指定 BSO 和成员维属性](#)

有关在 Essbase 中创建应用程序和数据库的详细说明，请参阅《Oracle Essbase Database Administrator's Guide》。

 **注意：**

Essbase 主数据库名称必须与 Essbase 主多维数据集名称相同。如果这两个名称不同，则用户在使用 Profitability 应用程序控制台进行部署时无法更新维。

2. 确保已将要创建 Profitability and Cost Management 应用程序的用户设置为管理员或具有应用程序创建设置的其他用户。
3. 创建 Profitability and Cost Management 应用程序。

 **注意：**

如果在 Essbase 主多维数据集中重命名或删除了维，将不能再在引用了这些维的已部署 Profitability and Cost Management 应用程序上正确更新这些维。换言之，绝不可修改定义维类型的 UDA。如果要重命名或删除维，请考虑创建一个单独的 Essbase 主数据库，以便现有 Profitability and Cost Management 应用程序在需要时仍可以更新。“更新前分析”选项中的现有验证不会报告这些情况。

 **注：**

部署 Essbase 主多维数据集时，会自动将 ASO 维成员及其属性（如 ASO 存储类型、ASO 公式、层次类型等）读取到 Essbase 中。无需手动操作。。

在部署期间，无法自动读取特定于 BSO 的维和成员属性。要对其进行管理，必须为 BSO 维成员和属性（如公式、数据存储和维存储类型）创建用户定义的属性维 (UDA)，以使 BSO 数据库与 Essbase 保持一致。

 **注：**

在复制通过主多维数据集方法创建的标准应用程序或明细应用程序后，可能需要重新启动 Profitability and Cost Management 服务，才能将复制的应用程序中的多维数据集成功部署到 Essbase。

维类型

创建 Oracle Essbase 主多维数据集时，必须为每个维分配一个 DIMTYPE UDA。

表 1 中显示了标准 Profitability 应用程序的有效维类型。

表 3-2 标准 Profitability 维类型

| 属性标签 | 属性名称 | 说明 |
|--------|--|---|
| 分配类型 | DIMTYPE_ALLOCTYPE | <p>仅对于标准 Profitability，使用以下条件创建 AllocationType 维：</p> <ul style="list-style-type: none"> 为 Generation1 成员分配 UDA 值 "DIMTYPE_ALLOCTYPE"，以便 Oracle Hyperion Profitability and Cost Management 可以识别该成员。成员名称可以是任何内容，但是 UDA 值必须按照所述内容设置。 在此维中创建一个虚拟成员使 Essbase 在 Gen1 成员没有子代时可以保存大纲。 |
| 度量 | DIMTYPE_MEASURES | <ul style="list-style-type: none"> 创建一个度量维，并为 Generation1 成员分配一个 UDA 值 "DIMTYPE_MEASURES"。成员名称可以是任何内容，但是 UDA 值必须按照所述内容设置。 在此维中创建一个虚拟成员使 Essbase 在 Gen1 成员没有子代时可以保存大纲。 |
| 常规（业务） | DIMTYPE_GENERIC | <p>为业务维分配一个 UDA 值 DIMTYPE_GENERIC。</p> |
| POV | <ul style="list-style-type: none"> DIMTYPE_POV1 DIMTYPE_POV2 DIMTYPE_POV3 DIMTYPE_POV4 | <p>为最多四个 POV 维按数字顺序分配 Generation1 成员值。例如，DIMTYPE_POV1、DIMTYPE_POV2，等等。</p> |

 注：

仅适用于标准 Profitability，要使用自定义度量维，请参阅“自定义度量维”。

明细和管理分类帐 Profitability 应用程序的维类型与之相似，不同之处在于明细 Profitability 具有系统维 "MeasuresDetailed" (DIMTYPE_MEASURES)，而管理分类帐 Profitability 具有系统维“规则”(DIMTYPE_RULES) 和“余额”(DIMTYPE_BALANCES)。

用户定义的属性

通过用户定义的属性 (UDA)，可以基于维成员的文本属性执行分析。例如，您可以添加一个名为“新产品”的 UDA。

为选定成员输入 UDA 名称。

您可以输入 80 个字母数字字符。

主多维数据集中的属性维名称只能由 ASCII 字符（英文字母和数字）构成，而且属性维名称的第一个字符必须是字母，例如 a-z 或 A-Z。

在任何 Oracle Essbase 大纲中，确保所有维都有唯一的名称。否则，创建大纲将失败。例如，属性维成员名称与常规维名称不匹配。

ASO 维和成员属性

在部署期间，系统会自动将表 1 中所示的 ASO 维属性和成员属性提取到 Oracle Essbase 中。无需手动操作来使这些属性与主 Essbase 应用程序保持一致。

表 3-3 Profitability and Cost Management ASO 维和成员属性

| 属性标签 | 属性名称 | 说明 |
|------------|--|--|
| 别名 | 别名 | 别名是在已部署的应用程序中显示的维成员名称。将显示除根成员之外的所有成员的别名。 |
| 注释 | 注释 | 可以为维或成员输入注释。注释最多可包含 255 个字符。默认情况下，此文本框显示当前注释（如果存在）。 |
| 合并 | 合并 | <p>成员合并属性确定子代上滚到其父代的方式。</p> <p>如果当前成员不是维或属性，则会向成员分配以下合并运算符之一：</p> <ul style="list-style-type: none"> • + (加) - 默认值 • - (减) • * (乘) • / (除) • % (百分比) • ~ (合并过程中忽略) • ^ (从不合并) • 未使用 <p>注意：在聚合存储大纲 (ASO) 中使用合并运算符时，存在一些限制。请参阅《Oracle Essbase Database Administrator's Guide》。</p> |
| 数据存储 (ASO) | <ul style="list-style-type: none"> • ASODimensionDataStorage (适用于维根成员) • ASOMemberDataStorage (适用于维成员) | <p>可用于维和成员的 ASO 存储选项：</p> <ul style="list-style-type: none"> • StoreData - 数据和维存储在一起。 • ShareData - 与此成员关联的数据可以共享。ShareData 属性仅适用于成员。维根成员不能共享。 • NeverShare - 与此维关联的数据不能予以共享，即便存在隐式共享关系，例如带有一个子代的父代。在这种情况下，数据复制到父代和子代中。 <p>注意：此选项不适用于聚合存储大纲中的存储层次。</p> <ul style="list-style-type: none"> • LabelOnly - 没有数据与此维关联。 |

表 3-3 (续) Profitability and Cost Management ASO 维和成员属性

| 属性标签 | 属性名称 | 说明 |
|--------------------|-------------------------------|--|
| 维公式 (ASO) | ASODimensionFormula (适用于维根成员) | 应用于维的计算公式。 |
| 维解析顺序 | DimensionSolveOrder | 选定维的解析序列中的数值。例如，如果此维是第二个解析，请输入 "2"。 |
| 层次类型 (仅适用于维) | DimensionHierarchyType | <p>层次类型属性仅适用于 Essbase 中的聚合存储 (ASO) 数据库。</p> <p>维的层次类型设置为以下值之一：</p> <ul style="list-style-type: none"> • STORED - 对于使用下列合并符号的任意维成员： <ul style="list-style-type: none"> - + ADDITION - ~ IGNORE <p>注意：如果成员的父代设为 LABEL_ONLY，则无合并或 IGNORE (~) 运算符只能用在 STORED 层次中</p> • DYNAMIC - 对于使用任何合并符号（包括 ADDITION 和 IGNORE）的维成员，或包含公式的维成员。 • ENABLED - 用以支持备用层次 |
| 层次类型 (仅适用于成员) | HierarchyType | <p>可用于成员的层次类型：</p> <ul style="list-style-type: none"> • Stored (对于第一个 Gen2 子代) • Dynamic (从第二个 Gen2 子代起适用。第二个 Gen2 成员可以容纳备用层次) • None (不使用) |
| 成员公式 (ASO) | ASOMemberFormula | 应用于成员的计算公式。 |
| 成员解析顺序 (仅适用于成员) | MemberSolveOrder | <p>为选定的成员输入解析序列的数值。例如，如果此成员是第二个解析，请输入 "2"。</p> <p>此属性仅适用于 ASO 数据库。</p> <p>解析顺序为 0 的成员继承其维的解析顺序。</p> <p>除非维排序顺序属性另行指定，否则解析顺序相同的成员将按照其维在数据库大纲中的显示顺序进行计算。</p> <p>无解析顺序的成员在有解析顺序的成员之后进行计算。</p> |

指定 BSO 和成员维属性

Oracle Essbase 主多维数据集是一个 ASO 应用程序，在部署期间会自动读取 ASO 维和成员。不过，特定于 BSO 的成员属性对 BSO 数据库而言是必需的（例如，针对标准 Profitability 应用程序），或者需要使用这些成员属性来处理无法自动读取的特殊维和成员。

要将这些 BSO 维和成员包含在 Essbase BSO 数据库中，要部署到 Oracle Hyperion Profitability and Cost Management 中的每个维都需要有 Generation1 成员。维在主大纲中的顺序是由在 Profitability and Cost Management 生成的大纲中的维顺序决定的。

您必须为表 1 中列出的每个维和成员创建 UDA。有关使用 Essbase 的其他信息，请参阅《Oracle Essbase Database Administrator's Guide》。

 **注意：**

不要将多个类型相同的 UDA 分配到 Essbase 中的同一个特定成员，这会导致 Profitability and Cost Management 部署不知道要使用哪个 UDA 类型，从而导致不一致的行为。例如，不要将 DIMTYPE_POV1 和 DIMTYPE_POV2 分配到同一个成员。

有关需要必须手动分配给应用程序的特殊 UDA 的属性，请参阅表 1。

 **注：**

因为 BSO 属性是使用 UDA 分配的，所以如果用户不慎在主 Essbase 主数据库中分配了不兼容的 BSO 属性，则 Profitability and Cost Management 中的后续“部署到报表多维数据集”操作可能会失败。如果出现这种情况，您可以通过手动加载报表数据库部署生成的数据和规则文件来识别不兼容的分配。

表 3-4 Profitability and Cost Management 的维和成员属性

| 属性标签 | 属性名称 | 说明 |
|------------|--|---|
| 数据存储 (BSO) | <ul style="list-style-type: none"> BSODimensionDataStorage (适用于维根成员) BSOMemberDataStorage (适用于维成员) | <p>对于 BSO 数据存储, 将以下 UDA 之一分配给任意子代成员的 Generation1 成员:</p> <ul style="list-style-type: none"> BSODS_LABELONLY BSODS_NEVERSHARE BSODS_SHAREDATA BSODS_DYNCALANDSTORE BSODS_DYNCALC BSODS_STOREDATA <p>设置 UDA 时, 如果没有为属性"BSO 数据存储"分配任何值, 则 Profitability and Cost Management 部署将假设已分配了以下默认值:</p> <ul style="list-style-type: none"> 为 Generation1 成员分配了 "BSODS_LABELONLY", 但有以下例外: <ul style="list-style-type: none"> POV Generation1 成员被设置为 "BSODS_STOREDATA" 度量 Generation1 成员被设置为 "BSODS_DYNCALC" 为所有其他成员分配了 "BSODS_STOREDATA" |
| 维公式 (BSO) | BSODimensionFormula (适用于维根成员) | <p>对于 BSO 维公式, 用户无法在 Essbase 主数据库中指定 BSO 维公式, 因为该数据库必须是一个 ASO 数据库 (因此该主数据库中可用的公式属性是 ASO 公式)。</p> <p>部署后, 要使 BSO 公式显示在 Essbase 报表数据库中, 请手动将 BSO 维公式直接输入到报表数据库中。如果重新部署, 必须重新输入 BSO 公式。</p> <p>请使用恰当的 BSO CALCULATOR 语法。</p> |
| 维排序顺序 | DimensionSortOrder | <p>按顺序输入数字值, 以设置维在 Essbase 大纲中的顺序, 该大纲是由 Profitability and Cost Management 生成的。</p> <p>例如, 如果此维将是 Essbase 大纲中的第二个维, 请输入 "2"。</p> <p>必须为模型中的每个维 (除别名维和 UDA 维外) 设置维排序顺序。</p> <p>维排序顺序必须连续、唯一且大于或等于 1。</p> |

表 3-4 (续) Profitability and Cost Management 的维和成员属性

| 属性标签 | 属性名称 | 说明 |
|------------------|--------------------------|---|
| 维存储类型 | DimensionStorageType | 维存储类型属性仅适用于 Essbase 中的块存储 (BSO) 数据库。 默认情况下, 该值设置为 "SPARSE"。 如果需要 DENSE 设置, 请将一个值为 "DIMSTORETYPE_DENSE" 的 UDA 分配给 Generation1 成员。 |
| 成员公式 (BSO) | BSOMemberFormula | 对于 BSO 成员公式, 用户无法在 Essbase 主多维数据集中指定 BSO 成员公式, 因为该数据库必须是一个 ASO 数据库 (因此, 主多维数据集中可用的公式属性是 ASO 公式)。 部署后, 要使 BSO 公式显示在 Essbase 报表数据库中, 请手动将 BSO 维公式直接输入到报表数据库中。如果重新部署, 必须重新输入 BSO 公式。 请使用恰当的 BSO CALCULATOR 语法。 |
| 两遍计算 (仅适用于维) | BSO_TWOPASS (适用于维根成员) | 仅对于 BSO 数据库, 指定 BSO_TWOPASS 以在第二次通过大纲时计算某个成员。 |
| 两遍计算 (仅适用于成员) | BSO_TWOPASS | 仅对于 BSO 数据库, 指定 BSO_TWOPASS 以在第二次通过大纲时计算某个成员。 |

查看分配给成员属性的块存储选项 (BSO) 数据存储值

要了解分配给成员的 BSO 数据存储值, 请在为 BSO 计算多维数据集部署 Oracle Essbase 失败时, 使用 HPM_DIM_MEMBER_PROP_V 视图进行调试。

该视图由以下列构成:

```
APPLICATION_NAME
DIMENSION_NAME
MEMBER_NAME
BSO_DATA_STORAGE
DIMENSION_STORAGE_TYPE
```

查询时, 该视图会为每个部署的 Oracle Hyperion Profitability and Cost Management 应用程序中的每个维成员返回一行。

自定义度量维

仅对于标准 Profitability, 如果要使用自定义度量成员, 请在“动因度量”下创建一个名为 "UserDefinedDriverMeasures" 的成员, 并添加自定义成员作为该成员的子代。Profitability 应

用程序控制台将这些自定义成员作为 "UserDefinedDriverMeasures" 的子代插入部署的度量层次。

UserDefinedDriverMeasures 成员用于存储应用程序特定的、用户定义的动因度量。按照如下方式设置 ASOMember 数据存储和 BSOMember 数据存储的属性：

- 如果成员没有子代，则设置为 **StoreData**。
- 如果成员作为子代添加到此成员，并且所有这些子代都具有 IGNORE 合并符号，则设置为 **LabelOnly**。

 **注：**

大纲中的所有动因度量必须是唯一的。不要将大纲中的维中的现有动因度量名称用作其他成员的名称（包括系统、POV 和业务维）；否则，“数据输入”屏幕将不会正确显示值。

第 III 部分

使用标准 Profitability

另请参阅：

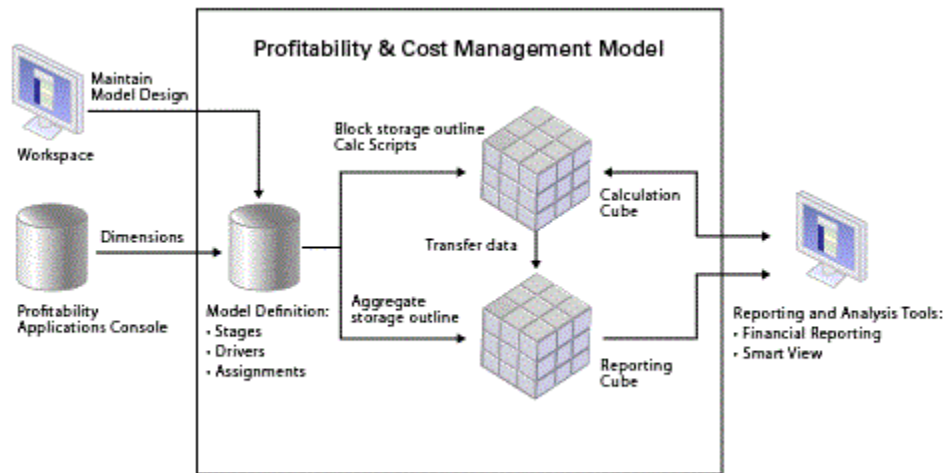
- [了解标准 Profitability 产品体系结构](#)
- [针对标准 Profitability 使用数据库](#)
- [使用标准 Profitability and Cost Management 维和成员](#)
- [标准 Profitability 导入临时表](#)
- [导出标准 Profitability 的模型定义数据](#)

4

了解标准 Profitability 产品体系结构

通过 Oracle Hyperion Enterprise Performance Management Workspace 访问的 Oracle Hyperion Profitability and Cost Management 是一个位于 Oracle Essbase 之上的分析工具。通过该应用程序，业务用户可以针对获利能力和成本管理创建其业务模型，并使用该模型信息创建可针对获利能力和成本分析进行微调的 Essbase 数据库。

图 4-1 标准 Profitability and Cost Management 产品体系结构



Profitability and Cost Management 利用 Profitability 应用程序控制台对应用程序元数据和安全性进行集中管理。

应用程序管理员使用 Profitability and Cost Management 维。使用 Oracle Hyperion Shared Services 对用户访问进行集中管理。准备好维元数据后，可将其部署到 Profitability and Cost Management 应用程序或模型。

模型设计包含生成 Essbase 大纲所需的信息和模型的 Essbase 组件所需的计算脚本。每个模型都需要访问下列数据库：

- 用于存储模型设计（包括维元数据）的关系数据库
- Essbase 数据库，包括计算数据库 (BSO) 和报表数据库 (ASO)。

注：

只需一个数据库即可存储多个模型。

可以在报表工具和分析工具（如 Oracle Hyperion Financial Reporting 和 Oracle Smart View for Office）中查看报表数据库中的结果。

5

使用标准 Profitability 的数据库

对于 Oracle Hyperion Profitability and Cost Management，既需要关系数据库，又需要 Oracle Essbase 数据库。应用程序中“模型摘要”的“系统信息”选项卡上指定了有关这些数据库及连接信息的详情。有关详细信息，请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》。

创建应用程序时，使用相应的值在 Profitability and Cost Management 关系数据库中填充表。模型部署期间会在 Profitability and Cost Management 中创建 Essbase 应用程序。每个数据库都存储特定的信息，如表 1 中所示。

表 5-1 Profitability and Cost Management 数据库

| 数据库类型 | 存储的信息类型 |
|---------|--|
| Essbase | <ul style="list-style-type: none">• 维层次• 成本数据• 收入数据• 动因数据• 汇率• 计算脚本 |
| 产品关系数据库 | <ul style="list-style-type: none">• 维• 阶段定义• POV 定义• 动因定义• 动因选定项• 分配• 分配规则• 分配规则选定项• 模型首选项 |

Profitability and Cost Management 使用在关系数据库中定义的动因定义、动因选定项和分配信息在 Essbase 应用程序中创建计算脚本。计算脚本执行所需的数据分配。

6

使用标准 Profitability and Cost Management 维和成员

另请参阅：

- [关于标准 Profitability 维和元数据](#)
Oracle Hyperion Profitability and Cost Management 管理员使用 Profitability 应用程序控制台创建和维护维和元数据。
- [维类型](#)
有多种维类型。
- [度量维](#)
度量维包含构建、验证和计算模型所需的成员。
- [AllocationType 维](#)
AllocationType 维，用于存储直接分配和分配系数数据。
- [别名维](#)
别名是指替代的名称、说明、语言或其他项，用于帮助定义维。
- [业务维](#)
业务维描述模型中每个阶段内的业务特定对象，例如产品、客户、区域和员工。
- [POV 维](#)
POV 维指示特定视点或模型版本，如年份、方案或期间。
- [属性维](#)
属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。
- [维的排序顺序和密度设置](#)
“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。

关于标准 Profitability 维和元数据

Oracle Hyperion Profitability and Cost Management 管理员使用 Profitability 应用程序控制台创建和维护维和元数据。

通过使用公共维和成员，Profitability and Cost Management 可以轻松地与其他产品（如 Oracle Hyperion Planning）一起使用公共数据和相互传输公共数据，从而节省时间和精力并提高准确性。有关创建和维护维和元数据的详细信息，请参阅[“使用 Profitability 应用程序控制台创建应用程序”](#)。

Profitability and Cost Management 使用在 Profitability 应用程序控制台中创建的维和成员来代表业务模型的许多结构元素：

- 存储直接分配和分配系数数据的 AllocationType 维
- 为模型和方案提供结构的维（如源度量维和目标度量维）。
- 反映模型的业务特定元素（如部门、总帐帐户、活动、客户或产品）的业务维

- 视点 (POV) 维，如期间、方案和版本
- 别名维（如果需要）
- 属性维（如果需要）

必须为每个维指定维类型和维名称：

- 维类型是一个维属性，通过它可以对选定的维使用预定义的功能。请参阅“[维类型](#)”。
- 分配维名称是为了标识维的内容，该名称与您的组织或企业相关联。例如，可以为帐户类型的维指定总帐或会计科目表之类的维名称。维名称无需反映维类型，尽管可以使其反映。有关命名限制，请参阅“[Essbase 命名约定](#)”。

▲ 注意：

虽然对可以创建的维的数量和成员数量没有实际限制，但大型维结构会发生性能问题。

Profitability and Cost Management 模型中维的排序顺序必须设置为特定顺序，以尽可能提高处理和计算性能。请参阅“[设置维排序顺序](#)”。

有关创建和维护维和成员的详细说明，请参阅“[使用 Profitability 应用程序控制台创建应用程序](#)”。有关维和成员的命名约定，请参阅“[Essbase 命名约定](#)”。

有关 Profitability and Cost Management 维的信息，请参阅以下几节：

- [度量维](#)
- [AllocationType 维](#)
- [别名维](#)
- [业务维](#)
- [POV 维](#)
- [属性维](#)

维类型

有多种维类型。

- 系统维（例如度量维和 AllocationType 维）必须根据以下各节中列出的要求来创建：
 - [度量维](#)
 - [AllocationType 维](#)
- 别名维用于分配替代的名称、说明、语言或其他项，用于帮助定义维。
- 创建业务维是为了描述模型内的业务元素，例如部门、总帐帐户、活动、客户或产品。请参阅“[业务维](#)”。
- 使用属性维可以基于维成员属性或品质执行分析。属性描述数据的特征，如产品的大小或颜色。

- 用户定义属性 (UDA) 允许基于文本属性进行分析，这些文本属性可用于对其他关联维的成员进行分类。例如，您可以添加一个名为“新产品”的 UDA。
- POV 维指示特定视点或模型版本，如年份、方案或期间。每个模型至少需要一个 POV 维。

还可以创建版本维类型以维护同一 POV 的不同版本。该版本用于跟踪同一模型的不同版本并评估更改或不同策略对模型的影响。

注：

定义维大纲时，某些限制字符不能在名称中使用。Oracle 强烈建议您查看《Oracle Essbase Database Administrator's Guide》中的 Oracle Essbase 命名约定。

度量维

度量维包含构建、验证和计算模型所需的成员。

除度量维所需的成员（这些成员不得修改）之外，用户可以将用户定义的动因度量（或子层次）添加到层次中的 'UserDefinedDriverMeasures' 成员下。用户可以修改这些维成员。

注：

如果需要，您可以为度量类型的维指定一个不同的名称。

在度量维中，度量类型可以分为下列类别：

- 动因度量，用于创建动因类型和公式。请参阅“[动因度量](#)”。
- 成本层分配度量，用于控制计算成本和输入成本的分配。请参阅“[成本层分配度量](#)”。
- 收入层分配度量，用于控制计算成本和输入成本的分配。请参阅“[收入层分配度量](#)”。
- 报表度量，用于生成报表，使用计算值和输入值为模型生成总成本和总收入。将计算所有非 0 级报表度量。请参阅“[报表度量](#)”。

动因度量

表 6-1 动因度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|------------------|---------|-----------------------|--------|
| FixedDriverValue | FixedDV | 用于需要固定动因值参数的动因类型的默认度量 | 输入 |
| 比率 | 比率 | 用于需要比率参数的动因类型的默认度量 | 输入 |
| 数量 | Qty | 用于需要数量参数的动因类型的默认度量 | 输入 |
| 权重 | 权重 | 用于需要权重参数的动因类型的默认度量 | 输入 |

表 6-1 (续) 动因度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|----------------------------------|-----------|---|--------|
| 百分比 | 百分比 | 用于百分比动因类型的默认度量 | 输入 |
| CalculatedDriverValue | CDV | 作为分配中使用的动因公式结果的度量 | 计算 |
| TotalDriverValue | TDV | 在分配公式 DV/TDV 中用作分母的度量 | 计算 |
| EffectiveTotalDriverValue | EffTDV | 如果定义动因时选中“允许空闲”框，此项为用于存储此类动因的有效动因总数的度量。 | 计算 |
| OverrideTotalDriverValue | OvrtdTDV | 用户输入的值，该值覆盖在分配中用作分母的 TotalDriverValue 度量。 | 输入 |
| TotalDriverValueAfterReciprocals | TDVAftRcp | 当倒数分配中涉及源时，在非倒数阶段内和阶段后分配的分配公式中用作分母的度量 | 计算 |
| IdleDriverValue | IdleDV | 计算 IdleCost 时用作动因值 (DV) 的度量 | 计算 |
| UserDefinedDriverMeasures | 不适用 | <p>UserDefinedDriverMeasures 成员用于存储应用程序特定的、用户定义的动因度量。</p> <p>按照如下方式设置 ASOMember 数据存储和 BSOMember 数据存储的属性：</p> <ul style="list-style-type: none"> - 如果此成员没有子代，则设置为 StoreData。 - 如果成员作为子代添加到此成员，并且所有这些子代都具有 IGNORE 合并符号，则设置为 LabelOnly。 <p>注意：大纲中的所有动因度量必须是唯一的。不要将大纲中的维中的现有动因度量名称用作其他成员的名称（包括系统、POV 和业务维）；否则，“数据输入”屏幕将不会正确显示值。</p> | 不适用 |

成本层分配度量

表 6-2 成本层分配度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|------------------------|------------------|---|--------|
| UnassignedCost | UnAsgCost | 完成所有分配和空闲计算之后源交叉点上的剩余成本 | 计算 |
| CostAssigned | CostAsg | 从源分配给阶段后目标和非倒数阶段内目标的总成本 | 计算 |
| CostAssignedIntraStage | CostAsgInt | 分配给阶段内目标（不包括倒数目标）的成本总和 | 计算 |
| CostAssignedPostStage | CostAsgPost | 分配给阶段后目标的成本总和 | 计算 |
| OverDrivenCost | OverDrivenCost | 对于标准基数动因，如果分配的总成本大于 NetCostForAssignment，则将超出金额过帐到 OverDrivenCost。 | 计算 |
| IdleCost | IdleCost | <p>闲置成本的生成方式随动因类型而异：</p> <ul style="list-style-type: none"> 对于实际基数动因，闲置成本使用以下分配公式生成： $\text{IdleDriverValue} / \text{OverrideTotalDriverValue}$ 对于标准基数动因，如果分配的总成本少于 NetCostForAssignment，则会生成闲置成本。 | 计算 |
| NetCostForAssignment | NetCostAsg | <p>在考虑所有阶段前、阶段内和倒数分配后，分配可用的总成本</p> <p>将 DataStorage (BSO) 属性设置为 StoreData。</p> | 计算 |
| GrossReceivedCost | GrRecCost | <p>从阶段前和阶段内分配中分配的所有成本总和，不包括倒数分配和用户输入成本</p> <p>将 DataStorage (BSO) 属性设置为 StoreData。</p> | 计算 |
| StandardCostRate | StandardCostRate | <p>对于标准基数动因，用户分配标准成本比率，并输入该值以在计算标准基数成本动因时使用，如下所示：</p> $\text{CostReceivedPriorStage} = \text{StandardCostRate} * \text{TotalDriverValue}$ | 输入 |
| CostInput | CostInput | 交叉点的用户输入成本值 | 输入 |

表 6-2 (续) 成本层分配度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|----------------------------|----------------------|---|--------|
| CostReceived | CostRec | 由阶段前分配和阶段内分配分配给交叉点的所有成本总和，不包括从倒数分配生成的成本 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| CostReceivedPriorStage | CostRecPri | 在阶段前分配上接收的成本总和 | 计算 |
| CostReceivedIntraStage | CostRecInt | 阶段内分配上接收的所有成本总和，不包括倒数分配 | 计算 |
| NetReciprocalCost | NetRcpCost | 倒数分配对阶段后目标和非倒数阶段内目标的分配可用金额的净效果 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| ReciprocalCostAssigned | RcpCostAsg | 分配给倒数目标的总成本，不包括从倒数接收的成本 | 计算 |
| ReciprocalCostReceived | RcpCostRec | 从倒数目标接收的总成本 | 计算 |
| ReciprocalIntermediateCost | RcpIntCost | 在应用联立方程后、调整倒数之前计算得出的交叉点的中间值 | 计算 |
| CostPerDrvUnit | Cost Per Driver Unit | 此度量是 AllocationMeasures 的子代。 此公式将分配的成本 (CostAssigned) 除以所有动因值的总和 (TotalDriverValue) 来计算每个动因值单位的成本。 | 计算 |
| UnitCost | Unit Cost | 此度量是 AllocationMeasures 的子代。 此公式将位于源交叉点的成本金额 (NetCostForAssignment) 除以用户输入数量来计算每单位的成本。 | 计算 |

收入层分配度量

表 6-3 收入层分配度量

| 成员名称 | 别名 | 说明 | 计算/输入 |
|-------------------|----------|---------------------------|-------|
| UnassignedRevenue | UnAsgRev | 完成所有分配计算和空闲计算之后源交叉点处的剩余收入 | 计算 |
| RevenueAssigned | RevAsg | 从源分配给阶段后和非倒数阶段内目标的总收入 | 计算 |

表 6-3 (续) 收入层分配度量

| 成员名称 | 别名 | 说明 | 计算/输入 |
|---------------------------|---------------------|---|-------|
| OverDrivenRevenue | OverDrivenRevenue | 对于标准基数动因, 如果总收入高于 NetRevenueForAssignment, 则将超出金额过帐到 OverDrivenRevenue。 | 计算 |
| RevenueAssignedIntraStage | RevAsgInt | 分配到阶段内目标 (不包括倒数目标) 的收入总和 | 计算 |
| RevenueAssignedPostStage | RevAsgPos | 分配到阶段后目标的收入总和 | 计算 |
| IdleRevenue | IdleRev | 闲置收入的生成方式随动因类型而异: <ul style="list-style-type: none"> 对于实际基数动因, 使用以下分配公式生成闲置收入: IdleDriverValue / OverrideTotalDriverValue。 对于标准基数动因, 如果分配的总成本少于 NetRevenueForAssignment, 则生成收入成本。 | 计算 |
| NetRevenueForAssignment | NetRevAsg | 在考虑所有阶段前、阶段内和倒数分配后, 分配可用的总收入 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| GrossReceivedRevenue | GrRecRev | 从阶段前和阶段内分配中分配的所有收入总和, 不包括倒数分配和用户输入收入 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| StandardRevenueRate | StandardRevenueRate | 对于标准基数动因, 用户分配标准收入比率, 并输入该值以在计算标准基数收入动因时使用, 如下所示: RevenueReceivedPriorStage =StandardRevenueRate * TotalDriverValue | 输入 |
| RevenueInput | RevInput | 交叉点的用户输入收入值 定义收入类别并将其作为层次存储在 RevenueInput 下。 将 DataStorage (BSO) 属性设置为 StoreData 。 | 输入 |

表 6-3 (续) 收入层分配度量

| 成员名称 | 别名 | 说明 | 计算/输入 |
|-------------------------------|-----------|---|-------|
| RevenueReceived | RevRec | 由阶段前分配和阶段内分配分配到交叉点的所有收入的总和, 不包括从倒数分配生成的收入 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| RevenueReceivedPriorStage | RevRecPri | 阶段前分配上接收的收入总和 | 计算 |
| RevenueReceivedIntraStage | RecRecInt | 阶段内分配上接收的所有收入的总和, 不包括倒数分配 | 计算 |
| NetReciprocalRevenue | NetRcpRev | 倒数分配对阶段后目标和非倒数阶段内目标的分配可用收入金额的净效果 将 DataStorage (BSO) 属性设置为 StoreData 。 | 计算 |
| ReciprocalRevenueAssigned | RcpRevAsg | 分配给倒数目标的总收入, 不包括从倒数分配接收的收入 | 计算 |
| ReciprocalRevenueReceived | RcpRevRec | 从倒数目标接收的总收入 | 计算 |
| ReciprocalIntermediateRevenue | RcpIntRev | 在应用联立方程后、调整倒数之前计算得出的交叉点的中间值 | 计算 |

报表度量

表 6-4 报表度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|--------------|-----------|---|--------|
| Profit | Profit | 选定交叉点的利润计算值。此值是以下计算的结果： $\text{NetRevenueForAssignment} - \text{NetCostForAssignment}$ | 计算 |
| GrossCost | GrossCost | 交叉点的总成本, 包括所有可能的输入： <ul style="list-style-type: none"> • 输入值 • 阶段前分配 • 阶段内分配 (包括倒数分配)。 此计算表示交叉点的真实总成本。 | 计算 |
| StandardCost | | 对于标准基数动因, 为 $\text{StandardCostRate} * \text{TotalDriverValue}$ 计算得出的成本 | 计算 |

表 6-4 (续) 报表度量

| 成员名称 | 别名 | 说明 | 计算还是输入 |
|---------------------------|---------------|---|--------|
| StandardRevenue | | 对于标准基数动因, 为 StandardRevenueRate * TotalDriverValue 计算 得出的收入 | 计算 |
| InitialCost | InitialCost | 在对阶段内成本或倒数成本 进行计算之前交叉点的成 本, 包括输入成本和从阶段 前分配接收的成本。 | 计算 |
| NetCostAfterIntraStage | NetCostAftInt | 交叉点的成本, 包括所有阶 段内成本分配 | 计算 |
| GrossRevenue | GrossRev | 交叉点的总收入, 包括所有 可能的输入: <ul style="list-style-type: none"> • 输入值 • 阶段前分配 • 阶段内分配 (包括倒数 分配)。 此计算表示交叉点的真实总 收入。 | 计算 |
| InitialRevenue | InitialRev | 在对阶段内收入或倒数收入 进行计算之前交叉点的收 入, 包括输入收入和从阶段 前分配接收的收入。 | 计算 |
| NetRevenueAfterIntraStage | NetRevAftInt | 考虑阶段内收入分配的所有 类型之后交叉点的收入 | 计算 |

AllocationType 维

AllocationType 维, 用于存储直接分配和分配系谱数据。

创建报表时, 借助 AllocationType 维, 您可以指定要检索的分配数据的类型。

注:

如果需要, 您可以为 AllocationType 类型的维指定一个不同的名称。

如果您使用向导创建应用程序并选中自动创建本地维, 则系统会自动生成 AllocationType 维。如果您选择创建空白应用程序, 则必须自行创建维并选择 **AllocationType** 维类型。

无法修改此维, 并且它在 Oracle Hyperion Profitability and Cost Management 中不可见。

在 Profitability and Cost Management 创建的 Oracle Essbase 大纲中, AllocationType 维包含下列成员:

- **AllAllocations**, 其包含下列子代成员:
 - **DirectAllocation**, 该成员存储在模型中的指定源交叉点和目标交叉点之间直接分配的计算数据。直接分配必须是由用户定义的分配。

- **GenealogyAllocation**，该成员存储对模型中的各种相关交叉点间的间接链接进行计算的分配系谱。
系谱分配不是直接由用户定义的，而是由两个或多个直接分配产生的。例如，存在 A-B-C 分配系谱，是因为存在从 A 到 B (A-B) 和 B 到 C (B-C) 的直接分配。
- **SysAllocVar1**，存储虚拟链接上的阶段内分配值，此值是分配到同一阶段另一节点的值的一部分，并且 **DirectAllocation** 成员中包含该值。
- **SysAllocVar2**，用于获取 **DirectAllocation**、**GenealogyAllocation** 和 **SysAllocVar1** 的源链接的总和。
- **SysAllocVar3**，存储系统中使用的计算系谱数据。不要在报表中使用此成员。
- **TotalAllocation**，动态计算 **DirectAllocation**、**GenealogyAllocation** 和 **SysAllocVar3** 的源链接的总和。
- **IndirectAllocation**，动态计算 **GenealogyAllocation** 和 **SysAllocVar3** 的源链接的总和。

▲ 注意：

不要编辑此维中的系统成员，因为这种修改可能会造成数据丢失或模型损坏。





别名维


别名是指替代的名称、说明、语言或其他项，用于帮助定义维。

例如，您可以引用系统中的客户编号，还可以分配一个在屏幕上显示公司名称的别名，从而更容易标识该客户。您可以为帐户、货币、实体、方案、期间、版本、年度和用户定义的维成员分配一个或多个别名。

不允许在同一个维中出现重复的成员名称或别名。如果在 Oracle Essbase 中克隆了特定的维，则可能会克隆别名。

要查看别名：

1. 在应用程序中，选择任何使用通用成员选择器的屏幕，例如“动因选定项”、“分配”、“数据输入”或“跟踪分配”。随即将显示包含通用成员选择器的相应屏幕，其中显示所有可用成员。
2. 可选：要选择成员：
 - 在动因选定项上，单击选择器  或添加 .
 - 在动因规则和例外选项卡上，单击添加新例外，然后选择一个维，再单击选择器 .
 - 在跟踪分配上，单击选择器 .

“选择成员”对话框随即打开，其中显示所有可用的成员。
3. 在通用成员选择器中，单击“上下文菜单”按钮 ，然后选择显示别名。别名将显示在成员列表中。

 注:

如果从上下文菜单中选择了“显示别名”，但未指定任何别名，则将在方括号中显示成员名称。例如，“产品”成员名称将显示为 [产品]。

业务维

业务维描述模型中每个阶段内的业务特定对象，例如产品、客户、区域和员工。

业务维可以使用以下部分或全部维类型，并且适用于一个或多个阶段或模型：

- 常规
- 帐户
- 实体
- 国家/地区

部署 Oracle Essbase 大纲后，在 Oracle Hyperion Profitability and Cost Management 应用程序中创建的业务维是没有类型的基本维或常规维。此功能使 Profitability and Cost Management 可重用为其他应用程序（例如 Oracle Hyperion Planning）定义的维成员和层次。

 注:

该维类型不适用于聚合存储大纲。

创建业务维时，应遵循下列要求：

- 维中 Gen1 成员的下列属性必须设置为 LABEL_ONLY：
 - DataStorage(BSO)
 - DataStorage(ASO)
- Gen1 维名称下的第一个 Gen2 子代通常设置为 All 成员。例如，部门维的 AllDepartments。

主层次在第一个 Gen2 子代下。分配模型只使用第一个 Gen2 层次，并且此层次不能包含共享成员。
- 其他 Gen2 成员可以容纳备用层次，但是这些层次不可用于构造分配模型。如果维要容纳备用层次，请将 DimensionHierarchyType 设置为 "Enabled"，将第一个 Gen2 成员的 HierarchyType 设置为 "Stored"，并将具有包含共享成员的备用层次的 Gen2 成员设置为 "Dynamic"。

这些备用层次在 Profitability and Cost Management 建模屏幕中不可见，只能在 Essbase 中看到。
- NoMember 成员是必需的。层次中最后一个 Gen2 子代必须始终为 NoMember，同时其合并设置为 IGNORE (~)。

POV 维

POV 维指示特定视点或模型版本，如年份、方案或期间。

可以自定义该维以反映组织的要求。例如，POV 可以包括季度、月份、季节性分组等等。

每个模型至少需要一个 POV 维，但最多可以创建 4 个 POV 维。

此外，还提供了版本维，该维用于创建模型的另一个实例。可以修改此版本以使用多种策略和业务选项来试验“假设分析”方案。通过修改此版本，可以实施多种特性，并比较结果以确定最佳操作。

属性维

属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。

属性维描述数据的特性，如产品的大小和颜色。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还可用于在创建分配规则时筛选目标交叉点。



注：

有关命名限制，请参阅“[Essbase 命名约定](#)”。

有两种属性维类型：

- 属性维：
 - 该属性可以使用不同的结构（如布尔、日期、数字和文本）创建。
 - 一个属性具有一个层次，可以上滚层次来获得聚合值。
 - 只能将给定属性维的一个属性与一个成员关联。
- 用户定义属性维 (UDA)：
 - 该属性只能使用文本创建。
 - UDA 没有层次，不能在报表中轻松地用来生成合计。
 - 一个成员可以与多个 UDA 相关联。

每种类型的属性维都提供了不同的优势，具体取决于您的模型和报表的需求。有关使用属性维的详细信息，请参阅《*Oracle Essbase Database Administrator's Guide*》。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还用于在创建分配规则时筛选目标交叉点。

有关命名限制，请参阅“[Essbase 命名约定](#)”。

维的排序顺序和密度设置

“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。

除了别名维和 UDA 维以外，模型中的所有维都必须设置维排序顺序。

▲ 注意：

如果维的排序顺序保留为空，验证将失败。

在维加载文件中设置“维排序顺序”属性。有关说明，请参阅[“设置维排序顺序”](#)。

维的排序顺序建议

Oracle 建议使用以下建议方法设置维的排序：

- 必须为模型中的每个维（别名维和 UDA 维除外）设置维排序顺序。

✎ 注：

维排序顺序会忽略别名维和 UDA 维，因为它们不以维的形式存在于 Oracle Hyperion Profitability and Cost Management 和 Oracle Essbase 中。

- 维排序顺序必须连续、唯一且大于或等于 1。
- 度量维默认设置为 1。
- AllocationType 维默认设置为 2。
- 业务维和 POV 维必须设置为 3 或更高。
- 属性维在排序时必须始终排在最后。例如，如果在由 12 个维组成的序列中有 4 个属性维，则这些属性维必须设置为 9、10、11 和 12。

设置维排序顺序

必须在维级别使用维加载文件中的维排序顺序属性设置模型中每个维的处理顺序。

必须遵循维排序顺序限制；否则，模型验证将失败。有关限制的完整列表，请参阅[“维的排序顺序建议”](#)。

要构造包括维排序顺序的加载文件：

1. 在外部程序或 ETL 流程中，构造加载文件，其中包含类似如下的列。确保包括对应于“维排序顺序”的列。

为维的根成员输入排序顺序。在此示例中，Accounts 的排序顺序为 9。

| Regular | Parent | Storage Type | Hierarchy Type | Comment | ASO Data Storage | BSO Data Storage | Two Pass Calculation | ASO | | Consolidation Type | UDA | Dimension Solve Order | Member | | Alias:English | Alias:Default | |
|-----------------------|--------|--------------|----------------|---------|------------------|------------------|----------------------|-------------------|----------------------|--------------------|-----|-----------------------|-------------|--------|---------------|----------------|----------------|
| | | | | | | | | Dimension Formula | Dimension Sort Order | | | | Solve Order | Member | | | |
| Accounts | | SPARSE | STORED | | LABELONLY | LABELONLY | N | | | | | 0 | 9 | | 0 | ALIAS:English | ALIAS:Default |
| AllAccounts | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | AllAccounts | AllAccounts |
| "NoAccour AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | NoAccounts | NoAccounts |
| "Cost of Gr AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Cost of Goods | Cost of Goods |
| "Material Cost of Gc | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Materials | Materials |
| "MAT51/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Frames | Frames |
| "MAT52/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Wheels | Wheels |
| "MAT53/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Gearsets | Gearsets |
| "MAT54/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Seats | Seats |
| "MAT55/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Controls | Controls |
| "MAT56/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Tubing | Tubing |
| "Equipme Cost of Gc | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Equipment Dep | Equipment Dep |
| "DEP610 Equipmen | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Bar Bender Dep | Bar Bender Dep |
| "Personne AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Personnel Exp | Personnel Exp |
| "PER2100 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Regular Salary | Regular Salary |
| "PER2200 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Regular Wage | Regular Wage |
| "PER2300 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Overtime | Overtime |
| "PER2400 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | | + | | | | | 0 | Vacation | Vacation |

- 在 Profitability 应用程序控制台中，依次选择“操作”和更新维以加载定义了维排序顺序的文件。

7

标准 Profitability 导入临时表

另请参阅：

- [使用导入临时表](#)
要将模型数据从关系数据库导入 Oracle Hyperion Profitability and Cost Management，必须在与 Profitability and Cost Management 产品架构分开的数据库架构中创建一组临时表。
- [为标准 Profitability 创建导入数据库表](#)
临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database。
- [HPM_STG_STAGE](#)
HPM_STG_STAGE 表提供模型内的阶段列表，并标识每个阶段的显示顺序、前缀、关联的维和动因维。
- [HPM_STG_POV](#)
HPM_STG_POV 表存储视点 (POV) 中包含的每个维成员组合的状态。
- [HPM_STG_DRIVER](#)
HPM_STG_DRIVER 表提供有关动因的详细信息，包括动因类型、显示顺序、层和度量维中的某些成员。
- [HPM_STG_DRIVER_SEL](#)
HPM_STG_DRIVER_SEL 表提供有关所选动因规则的 POV 和源阶段动因维成员的详细信息。
- [HPM_STG_DRIVER_EXCEP](#)
HPM_STG_DRIVER_EXCEP 表提供有关所选动因例外的 POV 和源阶段交叉点的详细信息。
- [HPM_STG_ASSIGNMENT](#)
HPM_STG_ASSIGNMENT 表提供有关每个分配的详细信息，包括源阶段、POV、层、源维成员、目标阶段和目标维成员。
- [HPM_STG_ASGN_RULE_SEL](#)
HPM_STG_ASGN_RULE_SEL 表存储有关控制所选阶段的分配的规则的源阶段和维成员的详细信息。

使用导入临时表

要将模型数据从关系数据库导入 Oracle Hyperion Profitability and Cost Management，必须在与 Profitability and Cost Management 产品架构分开的数据库架构中创建一组临时表。

可以重用与明细 Profitability 的模型数据架构相同的架构。然后用要导入的应用程序对象的详细信息填充这些临时表。

▲ 注意：

请勿修改产品架构。

临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database。使用适当的脚本在新的数据库架构中创建导入表：

 **注：**

如果两个阶段具有相同的维，但是处于不同的顺序，标准应用程序的临时导入无法正常工作。例如：

阶段 1：产品、客户、实体

阶段 2：客户、实体、产品

两个阶段具有相同的维数时，顺序也必须相同。

为标准 Profitability 创建导入数据库表

临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database。

使用 `create_staging.sql` 脚本在标准 Profitability 中创建表。

要创建临时表：

1. 在产品架构之外创建新的 Oracle 或 Microsoft SQL Server 数据库架构。
2. 在与数据库类型对应的默认位置下找到相应的 SQL 脚本：
 - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer`
 - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle`
3. 运行 `create_staging.sql` 脚本。

HPM_STG_STAGE

HPM_STG_STAGE 表提供模型内的阶段列表，并标识每个阶段的显示顺序、前缀、关联的维和动因维。

相关项：无

表 7-1 HPM_STG_STAGE

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-------------|-----------------|--------------------|------|--|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 是 | 模型阶段的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 阶段目的的简要说明。 |
| stage_order | identity(38,0) | integer(38,0) | 是 | 生成和计算脚本过程中建模（分配和规则创建）时在模型中使用所选阶段的顺序位置。 |

表 7-1 (续) HPM_STG_STAGE

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|--------------------|-----------------|--------------------|--------------|--|
| stage_prefix | nvarchar(80) | varchar2(80 CHAR) | 是 | 区分阶段名称的前缀。 |
| dim1_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 包含在阶段中的第一个维的名称。 |
| dim2_name | nvarchar(80) | varchar2(80 CHAR) | | 包含在阶段中的第二个维的名称 (如果可用)。 |
| dim3_name | nvarchar(80) | varchar2(80 CHAR) | | 包含在阶段中的第三个维的名称 (如果可用)。 |
| driver_dim_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 作为阶段的动因选择的维的名称。 |
| Is_intra | nvarchar(1) | varchar2(255 CHAR) | 是 | 输入 "Y" (是) 在模型阶段内启用阶段内分配。 默认值是 "N" (否)。此字段从应用程序中进行填充。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_POV

HPM_STG_POV 表存储视点 (POV) 中包含的每个维成员组合的状态。

相关项: 无

表 7-2 HPM_STG_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|------|-----------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 第一个 POV 维成员的名称。 |

表 7-2 (续) HPM_STG_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|--------------|---|
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 第二个 POV 维成员的名称 (如果可用)。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 第三个 POV 维成员的名称 (如果可用)。可能会根据需要添加其他维和成员名称。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 第四个 POV 维成员的名称 (如果可用)。可能会根据需要添加其他维和成员名称。 |
| pov_state | nvarchar(80) | varchar2(80 CHAR) | 是 | POV 的当前状态: <ul style="list-style-type: none"> 草稿 已发布 已归档 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_DRIVER

HPM_STG_DRIVER 表提供有关动因的详细信息, 包括动因类型、显示顺序、层和度量维中的某些成员。

相关项: 无

表 7-3 HPM_STG_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-------------|-----------------|--------------------|------|--------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 是 | 动因的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 对所选动因的用途的说明。 |

表 7-3 (续) HPM_STG_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-----------------|-----------------|-----------------------|------|--|
| driver_type | nvarchar(80) | varchar2(80 CHAR) | 是 | 动因类型： <ul style="list-style-type: none"> • 平均 • 简单 • 简单加权 • 简单变量 • 加权变量 • 固定值和变量 • 百分比 • 自定义 |
| fixed_member | nvarchar(80) | varchar2(80) | | 与标准动因度量 "FixedDriverValue" 对应的度量维中的成员。 |
| fixed_location | nvarchar(30) | varchar2(30) | | 标准动因度量 "FixedDriverValue" 的位置。 |
| rate_member | nvarchar(80) | varchar2(80) | | 与标准动因度量 "Rate" 对应的度量维 中的成员。 |
| rate_location | nvarchar(30) | varchar2(30) | | 标准动因度量 "Rate" 的位置。 |
| weight_member | nvarchar(80) | varchar2(80) | | 与标准动因度量 "Weight" 对应的度量 维中的成员。 |
| weight_location | nvarchar(30) | varchar2(30) | | 标准动因度量 "Weight" 的位置。 |
| volume_member | nvarchar(80) | varchar2(80) | | 与标准动因度量 "Volume" 对应的度量 维中的成员。 |
| volume_location | nvarchar(30) | varchar2(30) | | 标准动因度量 "Volume" 的位置。 |
| custom_formula | nvarchar(500) | varchar2(500 CHAR) | | 用户为动因创建的公 式。必须使用 Oracle Essbase 计算脚本语 法创建此公式。请参 阅《Oracle Hyperion Profitability and Cost Management 用 户指南》。 |
| cost_layer | nvarchar(30) | varchar2(30 CHAR) | | 输入 "Yes" 将动因层 设置为“成本”。它包 含模型的成本值。 注意：必须选择 cost_layer 或 revenue_layer 或者 两者。 |

表 7-3 (续) HPM_STG_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|--------------------|-----------------|--------------------|--------------|--|
| revenue_layer | nvarchar(30) | varchar2(30 CHAR) | | 输入 "Yes" 将动因层设置为“收入”。它包含模型的收入值。 注意：必须选择 cost_layer 或 revenue_layer 或者两者。 |
| allow_idle_cost | nvarchar(1) | varchar2(1 CHAR) | | 输入适当的值确定此动因是否允许闲置成本： <ul style="list-style-type: none"> • 输入 "Y" (是) 允许动因的闲置成本。 • 输入 "N" (否) 禁用动因的闲置成本。这是默认设置。 |
| driver_basis_type | nvarchar(80) | varchar2(80 CHAR) | 是 | 选择所需动因基数： <ul style="list-style-type: none"> • 实际基数 • 标准基数 注意：标准基数动因无法使用“平均”或“百分比”动因类型。 |
| priority | identity | integer(38,0) | 是 | 输入动因的计算优先级，以便按指定顺序运行阶段内的分配。首先处理最低优先级的动因。默认情况下，最低优先级设置为 100，最高优先级为 1。编号不必连续。同一优先级的动因不按特定顺序执行。只有正整数有效。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_DRIVER_SEL

HPM_STG_DRIVER_SEL 表提供有关所选动因规则的 POV 和源阶段动因维成员的详细信息。

相关项:

- HPM_STG_DRIVER
- HPM_STG_STAGE
- HPM_STG_POV

表 7-4 HPM_STG_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|------------------------|-----------------|--------------------|--------------|---|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第四个维成员的名称（如果适用）。 |
| driver_dim_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因维成员的名称 |
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 视点的所选层的名称： <ul style="list-style-type: none"> • 成本收入 • 收入 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点的模型阶段的名称。 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因维成员的动因名称。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 启动上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_DRIVER_EXCEP

HPM_STG_DRIVER_EXCEP 表提供有关所选动因例外的 POV 和源阶段交叉点的详细信息。

表 7-5 HPM_STG_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|--------------|---|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第四个维成员的名称（如果适用）。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 源阶段中第一个维成员的名称 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第二个维成员的名称（如果适用）。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第三个维成员的名称（如果适用）。 |
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 视点的所选层的名称： <ul style="list-style-type: none"> 成本收入 收入 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点的源阶段的名称。 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选交叉点的动因名称。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 启动上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_ASSIGNMENT

HPM_STG_ASSIGNMENT 表提供有关每个分配的详细信息，包括源阶段、POV、层、源维成员、目标阶段和目标维成员。

相关项：

- HPM_STG_POV
- HPM_STG_STAGE
- HPM_STG_DRIVER

表 7-6 HPM_STG_ASSIGNMENT

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|-------------------|---|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第四个维成员的名称（如果适用）。 |
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 视点的所选层的名称： <ul style="list-style-type: none"> • 成本收入 • 收入 注意：必须选择“成本”或“收入”。 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 分配的源阶段的名称。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 源阶段中第一个维成员的名称。 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 是（如果阶段源/目标阶段包含此维） | 源阶段中第二个维成员的名称（如果可用）。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 是（如果阶段源/目标阶段包含此维） | 源阶段中第三个维成员的名称（如果可用）。 |
| dst_stage_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 分配的目标阶段的名称。 |
| dst_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 目标阶段中第一个维成员的名称。 |
| dst_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 目标阶段中第二个维成员的名称（如果有）。 |

表 7-6 (续) HPM_STG_ASSIGNMENT

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|--------------|-----------------------------|
| dst_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 目标阶段中第三个维成员的名称（如果有）。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 启动上次导入的用户的 ID。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 启动上次导入的日期和时间。此值由导入程序在更新时填充。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充。 | 修改上次导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时填充。 | 修改的导入的日期和时间。 |

HPM_STG_ASGN_RULE_SEL

HPM_STG_ASGN_RULE_SEL 表存储有关控制所选阶段的分配的规则的源阶段和维成员的详细信息。

相关项：

- HPM_STG_POV
- HPM_STG_STAGE
- HPM_STG_DRIVER

表 7-7 HPM_STG_ASSGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|------|---------------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 视点中第四个维成员的名称（如果适用）。 |

表 7-7 (续) HPM_STG_ASSGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|---------------|--|
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 视点的所选层的名称： <ul style="list-style-type: none"> • 成本收入 • 收入 注意：“成本”或“收入”是必需的。 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 分配规则选定项的源阶段的名称。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 源阶段中第一个维成员的名称。 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 是，如果源阶段有适当的维。 | 源阶段中第二个维成员的名称（如果可用）。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 是，如果源阶段有适当的维。 | 源阶段中第三个维成员的名称（如果可用）。 |
| rule_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 控制所选分配的规则的规则的名称。规则必须存在于目标数据库中。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时间填充。 | 上次导入的日期和时间。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时间填充。 | 启动上次导入的用户的 ID。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时间填充。 | 启动重新导入的用户的 ID。 |
| modified_timestamp | timestamp(6) | timestamp(6) | 由导入程序在更新时间填充。 | 重新导入的日期和时间。 |

8

导出标准 Profitability 的模型定义数据

另请参阅：

- [关于导出标准 Profitability 的模型定义数据](#)
创建模型之后，可以查询数据库以将模型对象作为数据库视图中的输出显示。
- [HPM_EXP_STAGE](#)
HPM_EXP_STAGE 视图为所有应用程序检索在模型内定义的阶段列表。此视图还显示为阶段定义的所有其他属性。
- [HPM_EXP_POV](#)
HPM_EXP_POV 视图将检索视点 (POV) 中包含的每个维成员组合的状态。
- [HPM_EXP_DRIVER](#)
HPM_EXP_DRIVER 视图将检索有关动因的详细信息，包括动因类型、公式和层。
- [HPM_EXP_DRIVER_SEL](#)
HPM_EXP_DRIVER_SEL 视图为所有应用程序检索在模型内定义的“动因选定项 - 规则”的信息。
- [HPM_EXP_DRIVER_EXCEP](#)
HPM_EXP_DRIVER_EXCEP 视图将检索所有应用程序的模型内定义的“动因选定项 - 例外”。
- [HPM_EXP_ASSIGNMENT](#)
HPM_EXP_ASSIGNMENT 视图将检索有关每个分配的详细信息，包括源阶段、POV、层、源维成员、目标阶段和目标维成员。
- [HPM_EXP_ASGN_RULE_SEL](#)
HPM_EXP_ASGN_RULE_SEL 视图检索用于控制选定阶段的分配的规则的源阶段和维成员的详细信息。

关于导出标准 Profitability 的模型定义数据

创建模型之后，可以查询数据库以将模型对象作为数据库视图中的输出显示。

管理员可以在系统数据库中创建用于反映临时表中使用的列的数据库视图，从而显示系统中存储的模型数据：

- 阶段 ([HPM_EXP_STAGE](#))
- POV ([HPM_EXP_POV](#))
- 动因 ([HPM_EXP_DRIVER](#))
- 动因选定项 ([HPM_EXP_DRIVER_SEL](#))
- 动因例外 ([HPM_EXP_DRIVER_EXCEP](#))
- 分配 ([HPM_EXP_ASSIGNMENT](#))
- 分配规则选定项 ([HPM_EXP_ASGN_RULE_SEL](#))

HPM_EXP_STAGE

HPM_EXP_STAGE 视图为所有应用程序检索在模型内定义的阶段列表。此视图还显示为阶段定义的所有其他属性。

表 8-1 HPM_EXP_STAGE

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------|-----------------|--------------------|---|
| application_name | nvarchar(80) | varchar2(80) | 应用程序的名称 |
| id | identity | integer(38,0) | Oracle Hyperion Profitability and Cost Management 数据库中保存的阶段 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 模型阶段的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | 阶段目的的简要说明。 |
| stage_order | integer | integer | 生成和计算脚本过程中建模（分配和规则创建）时在模型中使用所选阶段的顺序位置。 |
| stage_prefix | nvarchar(80) | varchar2(80 CHAR) | 区分阶段名称的前缀。 |
| dim1_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第一个维的名称。 |
| dim2_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第二个维的名称（如果可用）。 |
| dim3_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第三个维的名称（如果可用）。 |
| driver_dim_name | nvarchar(80) | varchar2(80 CHAR) | 作为阶段的动因选择的维的名称。 |
| Is_intra | nvarchar(1) | varchar2(1 CHAR) | 确定是否允许阶段内分配的标志： <ul style="list-style-type: none"> • "Y"（是）指定在模型阶段内允许阶段内分配。 • "N"（否）指定在模型阶段内不允许阶段内分配。 |

HPM_EXP_POV

HPM_EXP_POV 视图将检索视点 (POV) 中包含的每个维成员组合的状态。

表 8-2 HPM_EXP_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------|-----------------|-------------------|---|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 应用程序的名称 |
| id | identity | integer(38,0) | Oracle Hyperion Profitability and Cost Management 产品数据库中保存的 POV 维成员组 ID |

表 8-2 (续) HPM_EXP_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|---|
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 第一个 POV 维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 第二个 POV 维成员的名称 (如果可用)。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 第三个 POV 维成员的名称 (如果可用)。 可能会根据需要添加其他维和成员名称。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 第四个 POV 维成员的名称 (如果可用)。 可能会根据需要添加其他维和成员名称。 |
| pov_state | nvarchar(80) | varchar2(80 CHAR) | POV 的当前状态: <ul style="list-style-type: none"> 草稿 已发布 已归档 |

HPM_EXP_DRIVER

HPM_EXP_DRIVER 视图将检索有关动因的详细信息, 包括动因类型、公式和层。

表 8-3 HPM_EXP_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------|-----------------|--------------------|--|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 应用程序的名称 |
| id | identity | integer | Oracle Hyperion Profitability and Cost Management 数据库中保存的动因 ID |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 动因的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | 对所选动因的用途的说明。 |
| display_order | integer | integer | 显示动因在模型中由所有动因组成的列表内的位置。 |
| driver_type | nvarchar(80) | varchar2(80 CHAR) | 动因类型: <ul style="list-style-type: none"> 平均 简单 简单加权 简单变量 加权变量 固定值和变量 百分比 自定义 |

表 8-3 (续) HPM_EXP_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|-------------------|-----------------|---------------------|---|
| complete_formula | nvarchar(500) | varchar2(4000 CHAR) | <p>用户为动因创建的公式，或预定义动因类型的转换公式。必须使用以下项创建此公式：</p> <p>Oracle Essbase 计算脚本语法。请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》。</p> |
| cost_layer | nvarchar(30) | varchar2(3 CHAR) | <p>将动因层设置为“成本”层，其中包括模型的成本值。</p> <p>注意：必须选择 cost_layer 或 revenue_layer 或者两者。</p> |
| revenue_layer | nvarchar(30) | varchar2(3 CHAR) | <p>将动因层设置为“收入”层，其中包括模型的收入值。</p> <p>注意：必须选择 cost_layer 或 revenue_layer 或者两者。</p> |
| allow_idle_cost | nvarchar(1) | varchar2(1 CHAR) | <p>确定此动因是否允许闲置成本的标志：</p> <ul style="list-style-type: none"> “Y”（是）指定此动因允许闲置成本。 “N”（否）指定此动因不允许闲置成本。 |
| driver_basis_type | nvarchar(80) | varchar2(80 CHAR) | <p>选择所需动因基数：</p> <ul style="list-style-type: none"> 实际基数 标准基数 <p>注意：标准基数动因无法使用“平均”或“百分比”动因类型。</p> |
| priority | integer | integer(38,0) | <p>输入动因的计算优先级，以便按指定顺序运行阶段内的分配。</p> <p>首先处理最低优先级的动因。默认情况下，优先级设置为 100，最低（或第一个）优先级为 0。数字不必是连续的。同一优先级的动因不按特定顺序执行。只有正整数有效。</p> |

HPM_EXP_DRIVER_SEL

HPM_EXP_DRIVER_SEL 视图为所有应用程序检索在模型内定义的“动因选定项 - 规则”的信息。

表 8-4 HPM_EXP_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------------|-----------------|-------------------|---|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第四个维成员的名称（如果适用）。 |
| driver_dim_member_name | nvarchar(80) | varchar2(80 CHAR) | 动因维成员的名称 |
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 视点的所选层的名称： <ul style="list-style-type: none"> • 成本收入 • 收入 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 所选视点的模型阶段的名称。 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 所选动因维成员的动因名称。 |

HPM_EXP_DRIVER_EXCEP

HPM_EXP_DRIVER_EXCEP 视图将检索所有应用程序的模型内定义的“动因选定项 - 例外”。

表 8-5 HPM_EXP_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|--|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 应用程序的名称 |
| id | identity | integer(38,0) | Oracle Hyperion Profitability and Cost Management 数据库中保存的动因选定项例外 ID。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 视点中第四个维成员的名称（如果适用）。 |

表 8-5 (续) HPM_EXP_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|---|
| layer_name | nvarchar(80) | varchar2(80 CHAR) | 视点的所选层的名称： <ul style="list-style-type: none"> 成本收入 收入 |
| src_stage_name | nvarchar(80) | varchar2(80 CHAR) | 所选视点的模型阶段的名称。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选阶段的第一个维的成员名称。 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选阶段的第二个维的成员名称（如果适用）。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选阶段的第三个维的成员名称（如果适用）。 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 所选交叉点的动因名称。 |

HPM_EXP_ASSIGNMENT

HPM_EXP_ASSIGNMENT 视图将检索有关每个分配的详细信息，包括源阶段、POV、层、源维成员、目标阶段和目标维成员。

表 8-6 HPM_EXP_ASSIGNMENT

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|---------------|---|
| application_name | nvarchar(80) | varchar2(80) | 应用程序的名称 |
| id | identity | integer(38,0) | Oracle Hyperion Profitability and Cost Management 数据库中保存的分配 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80) | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80) | 视点中第二个维成员的名称（如果适用）。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80) | 视点中第三个维成员的名称（如果适用）。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80) | 视点中第四个维成员的名称（如果适用）。 |
| layer_name | nvarchar(80) | varchar2(80) | 视点的所选层的名称： <ul style="list-style-type: none"> 成本 收入 |
| src_stage_name | nvarchar(80) | varchar2(80) | 分配的源阶段的名称。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80) | 源阶段中第一个维成员的名称。 |
| src_dim2_member_name | nvarchar(80) | varchar2(80) | 源阶段中第二个维成员的名称（如果可用）。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80) | 源阶段中第三个维成员的名称（如果可用）。 |

表 8-6 (续) HPM_EXP_ASSIGNMENT

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|--------------|-----------------------|
| dst_stage_name | nvarchar(80) | varchar2(80) | 分配的目标阶段的名称。 |
| dst_dim1_member_name | nvarchar(80) | varchar2(80) | 目标阶段中第一个维成员的名称。 |
| dst_dim2_member_name | nvarchar(80) | varchar2(80) | 目标阶段中第二个维成员的名称 (如果有)。 |
| dst_dim3_member_name | nvarchar(80) | varchar2(80) | 目标阶段中第三个维成员的名称 (如果有)。 |

HPM_EXP_ASGN_RULE_SEL

HPM_EXP_ASGN_RULE_SEL 视图检索用于控制选定阶段的分配的规则的源阶段和维成员的详细信息。

表 8-7 HPM_EXP_ASGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|---------------|--|
| application_name | nvarchar(80) | varchar2(80) | 应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80) | 所选视点中第一个维成员的名称。 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80) | 视点中第二个维成员的名称 (如果适用)。 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80) | 视点中第三个维成员的名称 (如果适用)。 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80) | 视点中第四个维成员的名称 (如果适用)。 |
| layer_name | nvarchar(80) | varchar2(80) | 视点的所选层的名称： <ul style="list-style-type: none"> 成本 收入 |
| src_stage_name | nvarchar(80) | varchar2(80) | 分配规则的源阶段的名称。 |
| src_dim1_member_name | nvarchar(80) | varchar2(80) | 源阶段中第一个维成员的名称。 |
| src_dim2_member_name | nvarchar(80) | varchar2(80) | 源阶段中第二个维成员的名称 (如果可用)。 |
| src_dim3_member_name | nvarchar(80) | varchar2(80) | 源阶段中第三个维成员的名称 (如果可用)。 |
| rule_name | nvarchar(80) | varchar2(80) | 控制所选分配源组合的分配规则的名称。应当已在 Oracle Hyperion Profitability and Cost Management 数据库中定义了该分配规则。 |

第 IV 部分

使用明细 Profitability

另请参阅：

- [了解明细 Profitability 产品体系结构](#)
- [使用明细 Profitability 数据库](#)
- [产品架构](#)
- [明细 Profitability 的模型数据架构](#)
- [使用明细 Profitability and Cost Management 维和元数据](#)
- [计算明细 Profitability 模型](#)
- [导入明细 Profitability 临时表](#)
- [导出明细 Profitability 的模型定义数据](#)

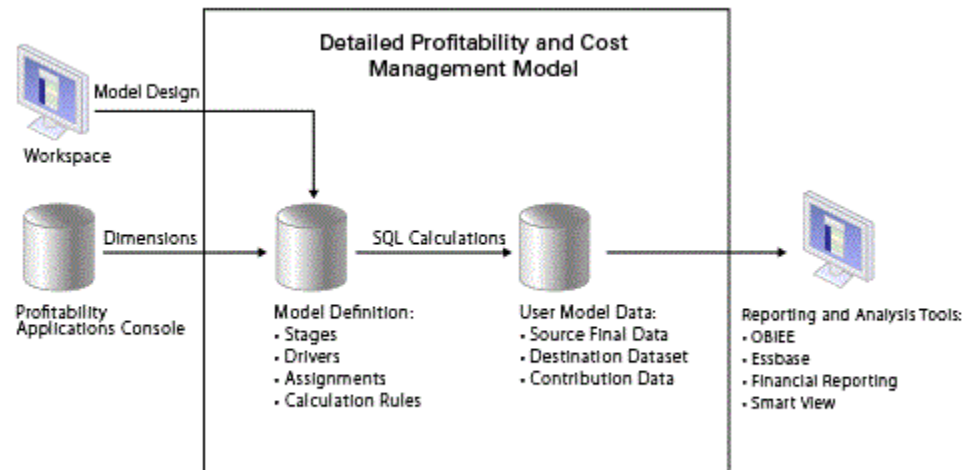
9

了解明细 Profitability 产品体系结构

通过 Oracle Hyperion Enterprise Performance Management Workspace 访问的明细 Profitability 是一个分析应用程序，它使用关系数据库中的数据。通过此应用程序，业务用户可以建立业务模型进行获利能力和成本管理，并使用该模型信息创建关系数据库，在其中存储模型信息、现有数据和此应用程序所计算出的结果。明细 Profitability 模型的数据（包括输入的和计算的数据）都存在于该关系数据库中。

Oracle Hyperion Profitability and Cost Management 利用 Oracle Hyperion Shared Services 对应用程序元数据和安全性进行集中管理。

图 9-1 明细 Profitability 体系结构



应用程序管理员使用 Profitability 应用程序控制台创建 Profitability and Cost Management 维。（请参阅[“使用 Profitability 应用程序控制台创建应用程序”](#)）。使用 Shared Services 对用户访问进行集中管理。准备好维元数据后，可将其部署到 Profitability and Cost Management 应用程序或模型。

模型设计包含生成 SQL 语句需要的信息，在数据库中执行计算需要这些语句。每个模型都需要访问数据库中的下列架构：

- 用于存储模型设计的架构，称为产品架构。请参阅[“产品架构”](#)。
- 称为模型数据架构的架构，用于存储现有数据和模型计算的结果。请参阅[“明细 Profitability 的模型数据架构”](#)。

10

使用明细 Profitability 数据库

存储明细 Profitability 应用程序的维数据和模型定义的关系数据库架构与用于存储标准 Profitability 应用程序的维数据和模型定义的关系数据库架构相同。该架构称为产品架构，它在安装 Profitability and Cost Management 时创建。从 Profitability 应用程序控制台部署应用程序时，维数据将填充到产品架构中。构建模型时，模型定义存储在此架构中。

对于明细 Profitability 应用程序，执行分配所基于的业务数据也存储在关系数据库中（而不存储在 Oracle Essbase 中，标准 Profitability 应用程序也是如此）。这些数据位于名为模型数据架构的单独数据库架构中。模型数据架构是用户定义的架构，必须与产品架构位于相同的数据库实例中。仅支持 Oracle 和 Microsoft SQL Server 数据库。

部署应用程序之后，维和成员将映射到相应的列并注册到 Oracle Hyperion Profitability and Cost Management 中的应用程序表。有关明细 Profitability 维的信息，请参阅“[使用明细 Profitability and Cost Management 维和元数据](#)”。

表 10-1 明细 Profitability and Cost Management 数据库架构

| 架构类型 | 存储的信息类型 |
|--------|--|
| 模型数据架构 | <ul style="list-style-type: none">• 维层次• 成本数据• 收入数据• 动因数据• 利润对象数据• 汇率 |
| 产品架构 | <ul style="list-style-type: none">• 维• 阶段定义• POV 定义• 动因定义• 动因选定项• 分配规则• 分配规则选定项• 模型首选项• 表注册• 计算规则 |

产品架构

标准和明细 Profitability 应用程序共享同一个产品架构，该产品架构用于存储 Oracle Hyperion Profitability and Cost Management 系统表，它是在安装期间创建的。Profitability and Cost Management UI 和明细 Profitability 计算引擎将连接到该架构。模型开发和模型计算期间在产品架构中创建报表视图。

对此架构的访问权限通常限于应用程序管理员和数据库管理员。系统将自动授予模型数据架构对报表视图的权限，并自动在模型数据架构中创建报表视图的同义词。同义词将视图公开给有权访问该架构的用户，而无需为这些用户授予对产品架构的权限。

安装之后，明细 Profitability 要求应授予对产品架构的以下系统权限（或包括这些权限的角色）：

- 对于 Oracle 数据库，明细 Profitability 要求对产品架构授予以下系统权限（或包括这些权限的角色）：
 - CREATE TABLE
 - CREATE VIEW
 - CREATE ANY SYNONYM
 - DROP ANY SYNONYM
- 对于 Microsoft SQL Server，明细 Profitability 要求对产品架构授予以下系统权限（或包括这些权限的角色）：
 - CREATE TABLE
 - CREATE VIEW
 - 授予产品架构对模型数据架构的 ALTER 或 CONTROL 权限（使产品架构能够更改模型数据架构中的对象）

例外：如果产品架构是管理员用户，如 "dbo" 或 "sa"，则已具有上述权限。在这种情况下，无需执行以下授权。

注：

请勿将 SYSADMIN 角色授予与 Profitability and Cost Management 产品架构关联的用户。因为这将更改该用户的默认架构，其默认架构必须一直设置为 Profitability and Cost Management 产品架构。

CREATE TABLE 和 CREATE VIEW 权限允许 Profitability and Cost Management 在模型开发和模型计算期间，在产品架构中创建系统生成的表和报表视图。

CREATE ANY SYNONYM 和 DROP ANY SYNONYM 权限允许 Profitability and Cost Management 管理模型数据架构中在产品架构中创建的系统生成报表视图的同义词。

 注:

如果授予这些系统权限是组织的一项安全隐患，可以禁用这些权限并在模型数据架构中手动为系统生成的报表视图创建同义词。通过此选项，可以仅对具有模型数据架构访问权限的业务数据用户公开这些视图。

12

明细 Profitability 的模型数据架构

另请参阅：

- [关于明细 Profitability 的模型数据架构](#)
模型数据架构可以是预先存在的架构或由数据库管理员创建以支持明细 Profitability 应用程序的架构。
- [对产品架构的模型数据架构表授权](#)
要在模型数据注册内查看模型数据架构的数据库对象，必须首先将表权限授予 Profitability and Cost Management 产品架构。
- [模型数据架构表结构要求](#)
要从模型数据架构注册表供明细 Profitability 应用程序使用，必须满足一些要求。
- [注册数据库视图](#)
如果为产品架构授予了相应权限，您也可以从模型数据架构注册数据库视图。

关于明细 Profitability 的模型数据架构

模型数据架构可以是预先存在的架构或由数据库管理员创建以支持明细 Profitability 应用程序的架构。

此架构存储的表和视图包含明细 Profitability 应用程序执行分配所基于的业务数据。Oracle Hyperion Profitability and Cost Management 仅直接连接到产品架构；要查看应用程序中的表或视图，数据库管理员必须对产品架构授予每个表或视图的表权限。

对产品架构的模型数据架构表授权

要在模型数据注册内查看模型数据架构的数据库对象，必须首先将表权限授予 Profitability and Cost Management 产品架构。

如下表所示授予每个表类型的权限：

表 12-1 针对每个表类型授予的权限

| 表类型 | 权限 |
|--------------|---|
| 水平或垂直走向的源阶段表 | GRANT SELECT ON <table_name or view_name><hpm_product_schema> |
| 目标阶段表 | GRANT SELECT, UPDATE ON <table_name or view_name><hpm_product_schema> |
| 所有类型的查找表 | GRANT SELECT ON <table_name or view_name><hpm_product_schema> |

 注:

对于 Oracle 数据库，必须授予对 dbms_stats 的 EXECUTE 权限。默认情况下，此权限将授予 Public。如果组织限制统计信息的收集，并从 Public 撤消了此权限，则必须将其授予产品架构。

部署应用程序之后，必须在模型数据架构中注册这些表以便在明细 Profitability 应用程序中使用。

您将它们的列映射到您在 Profitability 应用程序控制台中定义的维和度量。有关明细 Profitability 维的信息，请参阅[“使用明细 Profitability and Cost Management 维和元数据”](#)。

模型数据架构表结构要求

要从模型数据架构注册表供明细 Profitability 应用程序使用，必须满足一些要求。

- 数据库对象名称：必须是大写字母，可以包括数字 0-9 以及字符 "_" 和 "\$"。不支持小写字母或其他特殊字符。如果某些现有数据库对象名称使用了小写字母或特殊字符，可以创建视图，使用允许的标识符将这类表、视图和列名重命名，然后注册这些视图，而不注册其基础表。
- 仅限目标阶段表：必须具有名为 "WORKING"（使用大写字母）的数字列。
- 仅限目标阶段表：必须在其上定义主键约束，以实现表的唯一性。

注册数据库视图

如果为产品架构授予了相应权限，您也可以从模型数据架构注册数据库视图。

任何有效数据库视图都可以注册为源阶段表或查找表。仅简单的单一表可更新视图可注册为目标阶段表。

多种原因下可以使用数据库视图：

- 访问不位于模型数据架构中的表。对于查找表这可能特别有用。
- 筛选基础表中的行或隐藏基础表中的列，以阻止这些行或列对明细 Profitability 应用程序公开。
- 隐藏只读表的视图定义内的复杂联接以简化其在应用程序中的用法。

在模型开发和模型计算期间，Oracle Hyperion Profitability and Cost Management 将在产品架构中创建系统生成的报表视图。Profitability and Cost Management 还对视图授予模型数据架构的 SELECT 权限，并尝试为视图在模型数据架构中创建同义词。这将为用户提供对该架构的访问权限，而不提供对通常仅保留供管理员访问的产品架构的权限。通常应向模型构建者和报表开发人员授予对此架构的访问权限，除非管理员需要完成 Profitability and Cost Management 要求的所有表注册任务和所使用任何报表工具中的物理模型设置。

 注：

在 Oracle 中，数据库用户和数据库架构表示相同的内容；但是，在 Microsoft SQL Server 中，它们表示两种不同内容。如果使用的是 Microsoft SQL Server，则必须创建与模型数据架构同名的用户，并对该用户授予模型数据架构访问权限。由 Profitability and Cost Management 对系统生成的视图进行的权限授予表示模型数据用户。所创建的同义词将存储于模型数据架构中。

对于 SQL Server，在 Profitability and Cost Management 产品和文档中提到模型数据架构时，指的是该架构、相应用户或这两者。

 注：

SQL Server：必须在目标阶段表和查找表中创建包含 4 位或更多小数位的度量列以避免舍入错误。

 注意：

导入数据或对象之前，必须创建模型数据架构的备份。请参阅《*Oracle Enterprise Performance Management System Backup and Recovery Guide*》。

使用明细 Profitability and Cost Management 维和元数据

另请参阅：

- [关于明细 Profitability 维和元数据](#)
通过明细 Profitability 可将现有关系数据库用作明细 Profitability 应用程序的起点。
- [明细 Profitability 维类型](#)
对于明细 Oracle Hyperion Profitability and Cost Management，有多种可用维类型。
- [别名维](#)
别名是指替代的名称、说明、语言或其他项，用于帮助定义维。
- [业务维](#)
业务维描述模型中每个阶段内的业务特定对象，例如产品、客户、区域和员工。
- [POV 维](#)
POV 维指示特定视点或模型版本，如年份、方案或期间。可以自定义该维以反映组织的要求。
- [属性维](#)
属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。
- [维的排序顺序和密度设置](#)
“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。
- [设置维排序顺序](#)
必须在维级别使用维加载文件中的维排序顺序属性设置模型中每个维的处理顺序。

关于明细 Profitability 维和元数据

通过明细 Profitability 可将现有关系数据库用作明细 Profitability 应用程序的起点。

Oracle Hyperion Profitability and Cost Management 管理员使用 Profitability 应用程序控制台创建和维护维和元数据。通过使用公共维和成员，Profitability and Cost Management 可以轻松与其他产品（如 Oracle Hyperion Planning）一起使用公共数据和相互传输公共数据，从而节省时间和精力并提高准确性。有关创建和维护维和元数据的详细信息，请参阅[“使用 Profitability 应用程序控制台创建应用程序”](#)。

Profitability and Cost Management 使用在 Profitability 应用程序控制台中创建的维和成员来代表业务模型的许多结构元素。创建明细 Profitability 应用程序期间，将选择以下维。

- MeasuresDetailed（必需）：保留维，包含支持阶段平衡和其他验证活动所需的分配成员。为模型和方案提供结构。此维不包含动因度量。
- 至少一个 POV 维（必需）：期间、方案和版本
- 至少一个业务维（必需）：反映模型的业务特定元素（如部门、总帐帐户、活动、客户或产品）

- 别名维（可选）
- 属性维（可选）

必须为每个维指定维类型和维名称：

- 维类型是一个维属性，通过它可以对选定的维使用预定义的功能。请参阅“[维类型](#)”。
- 分配维名称是为了标识维的内容，该名称与您的组织或企业相关联。例如，可以为帐户类型的维指定总帐或会计科目表之类的维名称。维名称无需反映维类型，尽管可以使其反映。有关命名限制，请参阅“[Essbase 命名约定](#)”。

 **注意：**

虽然对可以创建的维的数量和成员数量没有实际限制，但大型维结构会发生性能问题。在已部署应用程序之后创建其他维或成员可能需要在“模型数据注册”中重新注册，以便使用新的对象。

有关创建和维护维和成员的详细说明，请参阅“[使用 Profitability 应用程序控制台创建应用程序](#)”。

明细 Profitability 维类型

对于明细 Oracle Hyperion Profitability and Cost Management，有多种可用维类型。

- MeasuresDetailed 为限制维，包含支持阶段平衡和其他验证活动所需的分配成员。
- 别名维用于分配替代的名称、说明、语言或其他项，用于帮助定义维。这是可选维，仅当要在模型中使用别名时才需要。

 **注：**

不允许出现重复的成员名称或别名。

- 创建业务维是为了描述模型内的业务元素，例如部门、总帐帐户、活动、客户或产品。应用程序至少需要一个业务维。
- 使用属性维可以基于维成员属性或品质执行分析。属性描述数据的特征，如产品的大小或颜色。

 **注：**

定义维大纲时，某些限制字符不能在名称中使用。Oracle 强烈建议您查看《*Oracle Essbase Database Administrator's Guide*》中的 Oracle Essbase 命名约定。

别名维

别名是指替代的名称、说明、语言或其他项，用于帮助定义维。


例如，您可以引用系统中的客户编号，还可以分配一个在屏幕上显示公司名称的别名，从而更容易标识该客户。您可以为帐户、货币、实体、方案、期间、版本、年度和用户定义的维成员分配一个或多个别名。

注：

不允许在同一个维中出现重复的成员名称或别名。

如果在 Oracle Essbase 中克隆了特定的维，则可能会克隆别名。

要查看别名：

1. 在应用程序中，选择任何使用通用成员选择器的屏幕，例如“动因选定项”、“分配”、“数据输入”或“跟踪分配”。随即将显示包含通用成员选择器的相应屏幕，其中显示所有可用成员。
2. 在通用成员选择器中，单击“上下文菜单”按钮 ，然后选择显示别名。
别名将显示在成员列表中。

注：

如果从上下文菜单中选择了“显示别名”，但未指定任何别名，则将在方括号中显示成员名称。例如，“产品”成员名称将显示为 [产品]。

业务维

业务维描述模型中每个阶段内的业务特定对象，例如产品、客户、区域和员工。

业务维可以使用以下部分或全部维类型，并且适用于一个或多个阶段或模型：

- 常规
- 帐户
- 实体
- 国家/地区

部署 Oracle Essbase 大纲后，在 Oracle Hyperion Profitability and Cost Management 应用程序中创建的业务维是没有类型的基本维或常规维。此功能使 Profitability and Cost Management 可重用为其他应用程序（例如 Oracle Hyperion Planning）定义的维成员和层次。

 注:

该维类型不适用于聚合存储大纲。

创建业务维时，应遵循下列要求：

- 维中 Gen1 成员的下列属性必须设置为 LABEL_ONLY:
 - DataStorage(BSO)
 - DataStorage(ASO)
- Gen1 维名称下的第一个 Gen2 子代通常设置为 All 成员。例如，部门维的 AllDepartments。
主层次在第一个 Gen2 子代下。分配模型只使用第一个 Gen2 层次，并且此层次不能包含共享成员。
- 其他 Gen2 成员可以容纳备用层次，但是这些层次不可用于构造分配模型。如果维要容纳备用层次，请将 DimensionHierarchyType 设置为 "Enabled"，将第一个 Gen2 成员的 HierarchyType 设置为 "Stored"，并将具有包含共享成员的备用层次的 Gen2 成员设置为 "Dynamic"
这些备用层次在 Profitability and Cost Management 建模屏幕中不可见，只能在 Essbase 中看到。
- NoMember 成员是必需的。层次中最后一个 Gen2 子代必须始终为 NoMember，同时其合并设置为 IGNORE (~)。

POV 维

POV 维指示特定视点或模型版本，如年份、方案或期间。可以自定义该维以反映组织的要求。

例如，POV 可以包括季度、月份、季节性分组等等。

每个模型至少需要一个 POV 维，但最多可以创建 4 个 POV 维。

此外，还提供了版本维，该维用于创建模型的另一个实例。可以修改此版本以使用多种策略和业务选项来试验“假设分析”方案。通过修改此版本，可以实施多种特性，并比较结果以确定最佳操作。

属性维

属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。

属性维描述数据的特性，如产品的大小和颜色。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还可用于在创建分配规则时筛选目标交叉点。

 **注：**

有关命名限制，请参阅“[Essbase 命名约定](#)”。

有两种属性维类型：

- 属性维：
 - 该属性可以使用不同的结构（如布尔、日期、数字和文本）创建。
 - 一个属性具有一个层次，可以上滚层次来获得聚合值。
 - 只能将给定属性维的一个属性与一个成员关联。
- 用户定义属性维 (UDA)：
 - 该属性只能使用文本创建。
 - UDA 没有层次，不能在报表中轻松地用来生成合计。
 - 一个成员可以与多个 UDA 相关联。

每种类型的属性维都提供了不同的优势，具体取决于您的模型和报表的需求。有关使用属性维的详细信息，请参阅《*Oracle Essbase Database Administrator's Guide*》。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还用于在创建分配规则时筛选目标交叉点。

有关命名限制，请参阅“[Essbase 命名约定](#)”。

维的排序顺序和密度设置

“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。

除了别名维和 UDA 维以外，模型中的所有维都必须设置维排序顺序。

 **注意：**

如果维的排序顺序保留为空，验证将失败。

在维加载文件中设置“维排序顺序”属性。有关说明，请参阅“[设置维排序顺序](#)”。

维的排序顺序建议

Oracle 建议使用以下建议方法设置维的排序：

- 必须为模型中的每个维（别名维和 UDA 维除外）设置维排序顺序。

 注:

维排序顺序会忽略别名维和 UDA 维，因为它们不以维的形式存在于 Oracle Hyperion Profitability and Cost Management 和 Oracle Essbase 中。

- 维排序顺序必须连续、唯一且大于或等于 1。
- 度量维默认设置为 1。
- AllocationType 维默认设置为 2。
- 业务维和 POV 维必须设置为 3 或更高。
- 属性维在排序时必须始终排在最后。例如，如果在由 12 个维组成的序列中有 4 个属性维，则这些属性维必须设置为 9、10、11 和 12。

设置维排序顺序

必须在维级别使用维加载文件中的维排序顺序属性设置模型中每个维的处理顺序。

必须遵循维排序顺序限制；否则，模型验证将失败。有关限制的完整列表，请参阅“[维的排序顺序建议](#)”。

要构造包括维排序顺序的加载文件：

1. 在外部程序或 ETL 流程中，构造加载文件，其中包含类似如下的列。确保包括对应于“维排序顺序”的列。

为维的根成员输入排序顺序。在此示例中，Accounts 的排序顺序为 9。

| Regular | Parent | Storage Type | Hierarchy Type | Comment | ASO Data Storage | BSO Data Storage | Two Pass Calculation | ASO Dimension Formula | Consolidation Type | UDA | Dimension Solve Order | Dimension Sort Order | Member Solve Order | Alias:English | Alias:Default |
|-------------|------------|--------------|----------------|---------|------------------|------------------|----------------------|-----------------------|--------------------|-----|-----------------------|----------------------|--------------------|----------------|----------------|
| Accounts | | SPARSE | STORED | | LABELONLY | LABELONLY | N | | | | 0 | 9 | | ALIAS:English | ALIAS:Default |
| AllAccounts | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | AllAccounts | AllAccounts |
| "NoAccour | AllAccoun | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | NoAccounts | NoAccounts |
| "Cost of G | AllAccoun | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Cost of Goods | Cost of Goods |
| "Material C | of Gc | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Materials | Materials |
| "MAT51 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Frames | Frames |
| "MAT52 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Wheels | Wheels |
| "MAT53 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Gearsets | Gearsets |
| "MAT54 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Seats | Seats |
| "MAT55 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Controls | Controls |
| "MAT56 | Materials | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Tubing | Tubing |
| "Equipme | Cost of Gc | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Equipment | De Equipment |
| "DEP610 | Equipmen | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Bar Bender | De Bar Bender |
| "Personne | AllAccoun | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Personnel | Exp |
| "PER2100 | Personnel | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Regular Salary | Regular Salary |
| "PER2200 | Personnel | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Regular Wage | Regular Wages |
| "PER2300 | Personnel | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Overtime | Overtime |
| "PER2400 | Personnel | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | 0 | Vacation | Vacation |

2. 在 Profitability 应用程序控制台中，依次选择“操作”和更新维以加载定义了维排序顺序的文件。

14

计算明细 Profitability 模型

在验证模型后，可以计算模型。

从“管理计算”屏幕中，明细 Profitability 用户可选择处理选项，包括自定义脚本或特定数据 POV。

除了基本计算功能，管理用户还可以查看其他操作和流程类型。

▲ 注意：

明细 Profitability 计算引擎对之前可选的维层次报表的表有了一种新的依赖性。这些表在 Oracle Hyperion Profitability and Cost Management 中的“计算”>“管理数据库”菜单>“报表的表和视图”选项卡中进行部署。每次重新部署明细 Profitability 应用程序时，都必须重新部署这些表。有关说明，请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》中的“部署明细 Profitability 报表视图”。

预计算和后计算自定义脚本

您可以创建自定义预计算脚本和后计算脚本，以在应用程序计算之前或之后执行相应任务来处理模型数据或阶段数据。

按如下所述运行自定义脚本：

- 在模型计算之前运行预计算 SQL 脚本。
- 在模型计算之后运行后计算 SQL 脚本。

在模型计算之前运行自定义脚本以准备分配，或者在模型计算之后运行自定义脚本以增强报表结果。

使用 SQL Developer 或者任何第三方 SQL 工具，创建用分隔符分隔的一系列简单 SQL 语句。自定义脚本存储于产品架构内数据库中的 HPM_SQL_SCRIPT 表的 SCRIPT 列中。

创建计算参数时选择相应自定义脚本。在任务区域下，依次选择计算和管理计算，然后选择所需的预计算脚本或后计算脚本。有关详细说明，请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》。

请参阅以下几节：

- [HPM_SQL_SCRIPT](#)
- [创建自定义脚本](#)

HPM_SQL_SCRIPT

此表存储自定义预计算脚本和后计算脚本。表中每一行都包含一条或多条可以在阶段模型期间执行的 SQL 语句。

**注：**

不要将脚本直接输入 HPM_SQL_SCRIPT 中。请参阅“[创建自定义脚本](#)”。

表 14-1 HPM_SQL_SCRIPT

| 列 | 数据类型 | 可为空 | 说明 |
|------------------|---------------|-----|--|
| NAME | Varchar2(80) | 否 | 自定义脚本的名称 |
| APPLICATION_NAME | Varchar2(8) | 否 | 已创建此自定义脚本的应用程序的名称 |
| MODEL_POV_NAME | Varchar2(80) | 是 | 选定的 POV 将以 "Year": "2012", "Period": "January", "Scenario": "Actual" 格式显示。 注意：如果此列设置为 NULL，则该脚本可用于任何模型 POV。 如果选择了特定 POV，则脚本只能用于该 POV。 |
| SCRIPT_TYPE | Varchar2(80) | 是 | 选择相应的脚本类型： <ul style="list-style-type: none"> • PRE（显示预脚本） • POST（显示后脚本） • EITHER of（显示预脚本或后脚本） • 如果未指定任何脚本类型，则将假定使用 EITHER。 |
| DESCRIPTION | Varchar2(255) | 是 | 输入脚本的目的或内容的说明 |
| SCRIPT | LONG | 否 | 在此处输入自定义脚本。 有关创建脚本的说明，请参阅“ 创建自定义脚本 ”。 |

创建自定义脚本

自定义脚本可以在文本编辑器中创建，如记事本、Textpad 或 Oracle SQL Developer。该脚本可以短如一条 SQL 语句，但如果脚本更长，则需要使用分隔符。需要的分隔符为自身位于单独一行中的正斜线字符（"/"）。请注意，可以在以下脚本中执行存储过程。

脚本将粘贴或加载到 HPM_SQL_SCRIPT 表的 SCRIPT 列，如 [HPM_SQL_SCRIPT](#) 中所示。不要将脚本直接键入 HPM_SQL_SCRIPT 表中。

**注意：**

只有具有 HPM 产品架构的密码的系统管理员才能创建预脚本和后脚本。

要创建和加载自定义脚本：

1. 在文本编辑器（如记事本、Textpad 或 Oracle SQL Developer）中创建自定义脚本。
2. 使用 Oracle SQL Developer 打开 HPM_SQL_SCRIPT 表。该表应位于 HPM 产品架构中，与产品表所在位置相同。
3. 插入一个新行。
4. 在表中输入所需值，包括脚本类型。请参阅“[HPM_SQL_SCRIPT](#)”。
5. 将脚本复制到 SCRIPT 列，如下所示：
 - 对于使用 SQL Developer 的 Oracle，复制并粘贴文本编辑器中的脚本，或者使用 Update 语句填充 SCRIPT 列。以下是 Oracle 的 SQL 脚本的一个示例：

```
update my_table set my_column =5  
/  
update your_table set your_column =6  
/  
begin myproc; end;  
/
```

- 对于使用 SQL Studio 的 Microsoft SQL Server，使用 Update 语句填充 SCRIPT 列。以下是 SQL Server 的 SQL 脚本的一个示例：

```
update my_table set my_column =5  
/  
update your_table set your_column =6  
/  
begin execute my_proc end  
/
```

▲ 注意：

您必须在同一脚本中的多个 SQL 语句之间包括分隔符“/”。“/”字符必须在每条语句末尾独占一行。确保最后一个“/”后跟一个换行符。

创建 ODBC 数据源以支持数据传输

“管理计算”选项卡中的“数据传输”选项需要名为 **PROFITABILITY_DS** 的 ODBC 数据源，该数据源连接到支持 Oracle Hyperion Profitability and Cost Management 实例的 Oracle 或 Microsoft SQL Server 数据库中的产品架构。

要创建该 ODBC 数据源：

1. 在应用程序服务器上，打开 ODBC 管理实用程序：在命令行上，执行 `odbcad32` 命令。
2. 选择系统 **DSN** 选项卡，然后单击添加…。
3. 选择相应的 ODBC 驱动程序，然后单击下一步。
4. 在数据源名称中，输入 **PROFITABILITY_DS**，然后填写连接详细信息以连接到产品架构。

高级计算选项

在“管理计算”屏幕上，将为管理员显示“动因操作类型”和“其他流程类型”选项卡，这两个选项卡对基本用户不可用：

- [动因操作类型](#)
- [其他流程类型](#)

▲ 注意：

Oracle 建议仅有经验的数据库管理员才应使用这些选项，修改任意动因操作类型或其他流程类型都可能会对应用程序产生重大影响，并且可能会损坏模型或数据。

动因操作类型

“管理计算”屏幕上的“动因操作类型”选项卡显示明细 Profitability 的当前动因操作类型。

▲ 注意：

不要在此选项卡中创建新动因操作类型。仅有经验的数据库管理员才应访问此选项卡，因为任何更改都可能会对应用程序产生重大影响，并且可能会损坏模型或数据。

“动因操作类型”是用于扩展支持的动因操作的一项管理功能，用于解决性能问题或特殊动因问题。使用该功能需要具备高级 SQL 知识、对数据库管理有深入了解并对没有文档说明的 Oracle Hyperion Profitability and Cost Management 明细 Profitability 应用程序功能有深入认识。Oracle 不建议使用此功能，除非是 Oracle 为响应客户服务请求而要求使用。

其他流程类型

此版本的明细 Profitability 提供了两种备用计算流程类型：

- Oracle Database 11g（默认）
- Oracle Database 10g

如果您的系统配置有 Oracle Database 10g，请使用“其他流程类型”选项卡选择适合您的数据库的计算流程。

注意：

Oracle 建议仅有经验的数据库管理员才应使用“其他流程类型”选项卡。不要对“其他流程类型”进行任何其他修改，因为所做的修改可能会对应用程序产生重大影响，并且可能会损坏模型或数据。


根据所用于明细 Profitability 的 Oracle 数据库版本，使用此屏幕可将您的数据库更改为正确版本。

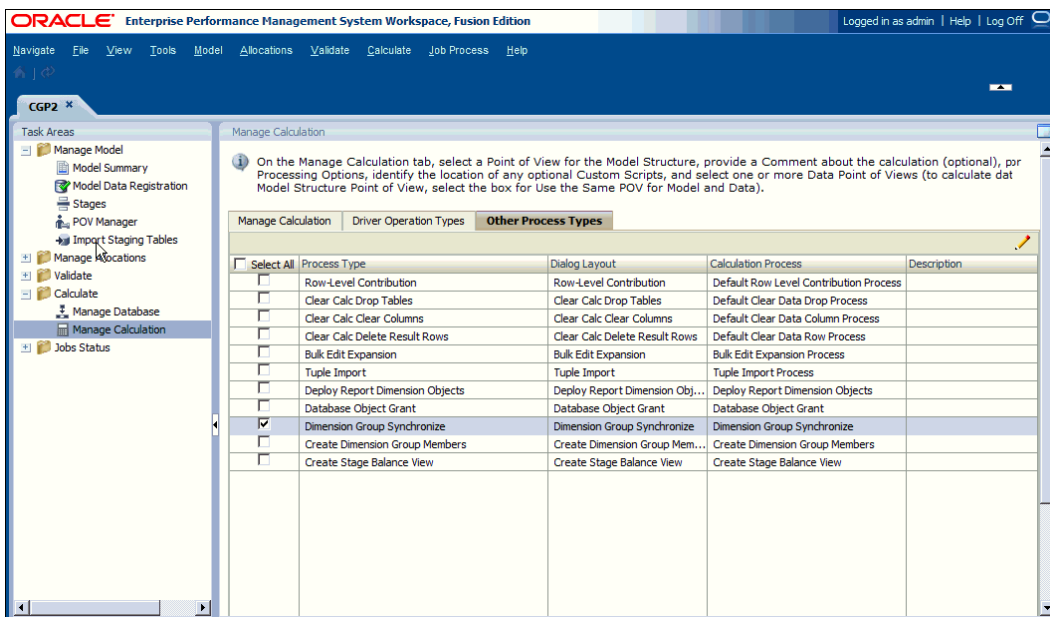
要更改您的 Oracle 数据库版本：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、应用程序和 **Profitability**，然后选择要为其修改 Oracle Database 版本的应用程序。
2. 在“任务区域”下，依次选择计算、管理计算和其他流程类型选项卡。

注：

仅当已为您分配管理员用户角色时，此选项卡才可见。

3. 选择维组同步，然后单击“编辑”按钮  将显示“定义其他流程类型”对话框。



4. 在“计算流程”下，选择适合 Oracle 数据库的选项：
 - 维组同步 (Oracle 11g)
 - 维组同步 10G (Oracle 10g)
5. 单击确定以保存更改。

导入明细 Profitability 临时表

另请参阅：

- [关于导入明细 Profitability 临时表](#)
要将模型数据从关系数据库导入 Oracle Hyperion Profitability and Cost Management，必须在与 Profitability and Cost Management 产品架构分开的数据库架构中创建一组临时表。
- [为明细 Profitability 创建导入数据库表](#)
临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database
- [HPM_STGD_POV](#)
HPM_STGD_POV 表存储着视点 (POV) 中包含的维成员的每个组合的状态。
- [HPM_STGD_DRIVER](#)
HPM_STGD_DRIVER 表提供有关动因的详细信息，包括动因类型、显示顺序和公式。
- [HPM_STGD_DRIVER_SEL](#)
HPM_STGD_DRIVER_SEL 表提供有关所选动因规则的 POV 和源阶段动因维成员的详细信息。
- [HPM_STGD_DRIVER_EXCEP](#)
HPM_STGD_DRIVER_EXCEP 表提供有关所选动因例外的 POV 和源阶段交叉点的详细信息。
- [HPM_STGD_ASGN_RULE_SEL](#)
HPM_STGD_ASGN_RULE_SEL 表提供有关分配规则选定项的详细信息。
- [HPM_STGD_CALCRULE_SNGLSRC](#)
HPM_STGD_CALCRULE_SNGLSRC 表提供有关创建或更新单源计算规则的详细信息。
- [HPM_STGD_CALCRULE_CALCMSRS](#)
HPM_STGD_CALCRULE_CALCMSRS 表提供有关创建或更新计算的度量规则的详细信息。
- [HPM_STGD_CALCRULE_MULTISRC](#)
HPM_STGD_CALCRULE_MULTISRC 表提供有关创建或更新多源计算规则的详细信息。

关于导入明细 Profitability 临时表

要将模型数据从关系数据库导入 Oracle Hyperion Profitability and Cost Management，必须在与 Profitability and Cost Management 产品架构分开的数据库架构中创建一组临时表。

可以重用与明细 Profitability 的模型数据架构相同的架构。然后用要导入的应用程序对象的详细信息填充这些临时表。

▲ 注意：

请勿修改产品架构。

临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database。使用适当的脚本在新的数据库架构中创建导入表：

为明细 Profitability 创建导入数据库表

临时数据库脚本可用于 Microsoft SQL Server 和 Oracle Database

- 。使用 `create_dp_staging.sql` 脚本在明细 Profitability 中创建表。

要创建临时表：

1. 在产品架构之外创建新的 Oracle 或 Microsoft SQL Server 数据库架构。
2. 在与您的数据库类型对应的默认位置找到 `create_dp_staging.sql`：
 - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer`
 - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle`
3. 运行 `create_dp_staging.sql` 脚本。

HPM_STGD_POV

HPM_STGD_POV 表存储着视点 (POV) 中包含的维成员的每个组合的状态。

表 15-1 HPM_STGD_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|-------------|---|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 |
| pov_state | nvarchar(80) | varchar2(80 CHAR) | 是 | POV 的当前状态： <ul style="list-style-type: none"> • 草稿 • 已发布 • 已归档 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 启动上次导入的用户的 ID |
| created_timestamp | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |

表 15-1 (续) HPM_STGD_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|--------------------|-----------------|-------------------|-------------|---------------|
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 修改上次导入的用户的 ID |
| modified_timestamp | timestamp(6) | timestamp(6) | | 修改的导入的日期和时间 |

HPM_STGD_DRIVER

HPM_STGD_DRIVER 表提供有关动因的详细信息，包括动因类型、显示顺序和公式。

表 15-2 HPM_STGD_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-----------------|-----------------|---------------------|------|--|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因的名称 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 对所选动因的用途的说明 |
| formula | nvarchar(500) | varchar2(500 CHAR) | 是 | 用户为动因创建的公式。此公式必须使用 SQL 语法创建，并且从数学方面来看是正确的运算。 |
| allow_idle_cost | nvarchar(1) | varchar2(1 CHAR) | | 输入适当的值确定此动因是否允许闲置成本： <ul style="list-style-type: none"> 输入 "Y" (是) 允许动因的闲置成本。 输入 "N" (否) 禁用动因的闲置成本。此选项为默认选项。 |
| priority | integer(38,0) | integer(38,0) | 是 | 输入动因的计算优先级，以便按指定顺序运行阶段内的分配。首先处理最低优先级的动因。默认情况下，最低优先级设置为 100，最高优先级为 1。编号不必连续。 <p>同一优先级的动因不按特定顺序执行。只有正整数有效。</p> |
| volume_formula | nvarchar(4000) | varchar2(4000 CHAR) | | 仅适用于基于比率的动因，其同时定义了比率和数量公式。 |

表 15-2 (续) HPM_STGD_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-------------------------|-----------------|--------------------|-------------|---|
| dst_measure_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 维表中度量的名称 |
| tdv_measure_member_name | nvarchar(80) | varchar2(80 CHAR) | | 在分配公式中用作分母的度量: Driver Value/ TotalDriverValue (DV/TDV) |
| operation_type | nvarchar(80) | varchar2(80 CHAR) | 是 | 为动因选择的运算类型: <ul style="list-style-type: none"> • 基于比率的分配 • 基于比例的分配 • 计算的度量 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 启动上次导入的用户的 ID |
| created_timestamp | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 修改上次导入的用户的 ID |
| modified_timestamp | timestamp(6) | timestamp(6) | | 修改的导入的日期和时间 |

HPM_STGD_DRIVER_SEL

HPM_STGD_DRIVER_SEL 表提供有关所选动因规则的 POV 和源阶段动因维成员的详细信息。

表 15-3 HPM_STGD_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|------|--------------------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 (如果适用) |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 (如果适用) |

表 15-3 (续) HPM_STGD_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|------------------------|-----------------|--------------------|-------------|--------------------------|
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 (如果适用) |
| driver_dim_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因维成员的名称 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因的名称 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 启动上次导入的用户的 ID |
| created_timestamp | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 修改上次导入的用户的 ID |
| modified_timestamp | timestamp(6) | timestamp(6) | | 修改的导入的日期和时间 |

HPM_STGD_DRIVER_EXCEP

HPM_STGD_DRIVER_EXCEP 表提供有关所选动因例外的 POV 和源阶段交叉点的详细信息。

表 15-4 HPM_STGD_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|------|--------------------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 (如果适用) |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 (如果适用) |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 (如果适用) |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 源阶段中第一个维成员的名称 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第二个维成员的名称 (如果适用) |

表 15-4 (续) HPM_STGD_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|-------------|----------------------|
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第三个维成员的名称 (如果适用) |
| src_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第四个维成员的名称 (如果适用) |
| src_dim5_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第五个维成员的名称 (如果适用) |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选动因的名称 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 启动上次导入的用户的 ID |
| created_timestamp | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 修改上次导入的用户的 ID |
| modified_timestamp | timestamp(6) | timestamp(6) | | 修改的导入的日期和时间 |

HPM_STGD_ASGN_RULE_SEL

HPM_STGD_ASGN_RULE_SEL 表提供有关分配规则选定项的详细信息。

表 15-5 HPM_STGD_ASGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|-------------------|------|----------------------|
| id | identity | integer(38,0) | 是 | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 源阶段中第一个维成员的名称 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第二个维成员的名称 (如果适用) |

表 15-5 (续) HPM_STGD_ASGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|-------------|-----------------------------|
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第三个维成员的名称 (如果适用) |
| src_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第四个维成员的名称 (如果适用) |
| src_dim5_member_name | nvarchar(80) | varchar2(80 CHAR) | | 源阶段中第五个维成员的名称 (如果适用) |
| rule_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 控制所选分配的规则的名称。规则必须存在于目标数据库中。 |
| last_upload_date | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 详细说明导入此表期间发生的错误的消息。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 启动上次导入的用户的 ID |
| created_timestamp | timestamp(6) | timestamp(6) | | 上次导入的日期和时间 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | 由导入程序在更新时填充 | 修改上次导入的用户的 ID |
| modified_timestamp | timestamp(6) | timestamp(6) | | 修改的导入的日期和时间 |

HPM_STGD_CALCRLU_SNGLSRC

HPM_STGD_CALCRLU_SNGLSRC 表提供有关创建或更新单源计算规则的详细信息。

表 15-6 HPM_STGD_CALCRLU_SNGLSRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|------|-----------------------------|
| id | identity | NUMBER(38,0) | 是 | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | VARCHAR2(80 CHAR) | 是 | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 |

表 15-6 (续) HPM_STGD_CALCRLU_SNGLSRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|-------------------------|-----------------|--------------------|------|--|
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | identity | NUMBER(4,0) | 是 | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | VARCHAR2(255) | | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | VARCHAR2(1 CHAR) | 是 | |
| first_sequence_priority | identity | NUMBER(38,0) | 是 | 在此规则将执行的动因范围内指定第一个动因序列优先级。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| last_sequence_priority | identity | NUMBER(38,0) | 是 | 在此规则将执行的动因范围内指定最后一个动因序列优先级。其数值必须大于等于 first_sequence_priority，否则在插入期间将收到检查约束错误。 |
| all_drivers_flag | nvarchar(1) | VARCHAR2(1 CHAR) | 是 | 如果为 Y，在计算期间将忽略第一个序列优先级和最后一个序列优先级字段中的值。 |
| last_upload_date | timestamp | timestamp | | 此列将在导入时填充，显示执行导入的最近日期。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 包含错误代码，指示导入行时是否存在问题。Null 表示导入成功。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | | 指出在此表中创建行的初始用户。Profitability and Cost Management 不使用此字段。 |
| created_timestamp | timestamp(6) | timestamp | | 指出创建行的初始日期/时间。Profitability and Cost Management 不使用此字段。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | | 标识上次修改此表中的行的用户。Profitability and Cost Management 不使用此字段。 |
| modified_timestamp | timestamp(6) | timestamp | | 标识上次更改行的日期/时间。Profitability and Cost Management 不使用此字段。 |

HPM_STGD_CALCRLUE_CALCMSRS

HPM_STGD_CALCRLUE_CALCMSRS 表提供有关创建或更新计算的度量规则的详细信息。

表 15-7 HPM_STGD_CALCRLUE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|------------------------|-----------------|--------------------|------|---|
| id | identity | number(38,0) | 是 | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | varchar2(80 CHAR) | 是 | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | number | number(4,0) | 是 | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | varchar2(255) | | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | varchar2(1 CHAR) | 是 | |
| driver_name1 | nvarchar(1) | varchar2(1 CHAR) | 是 | 要与此计算规则关联的动因名称。 |
| driver_name2 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的动因名称。 |
| driver_name3 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的动因名称。 |
| driver_name4 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的动因名称。 |
| driver_name5 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的动因名称。 |
| dest_assign_rule_name1 | nvarchar(80) | varchar2(80 CHAR) | 是 | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name2 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name3 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name4 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |

表 15-7 (续) HPM_STGD_CALCRLUE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|------------------------|-----------------|--------------------|------|--|
| dest_assign_rule_name5 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| last_upload_date | timestamp(6) | timestamp | | 此列将在导入时填充，显示执行导入的最近日期。 |
| import_exception | nvarchar(255) | varchar2(255 CHAR) | | 包含错误代码，指示导入行时是否存在问题。Null 表示导入成功。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | | 指出在此表中创建行的初始用户。 Profitability and Cost Management 不使用此字段。 |
| created_timestamp | timestamp(6) | timestamp | | 指出创建行的初始日期/时间。 Profitability and Cost Management 不使用此字段。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | | 指出在此表中创建行的初始用户。 Profitability and Cost Management 不使用此字段。 |
| modified_timestamp | timestamp(6) | timestamp | | 指出创建行的初始日期/时间。 Profitability and Cost Management 不使用此字段。 |

HPM_STGD_CALCRLUE_MULTISRC

HPM_STGD_CALCRLUE_MULTISRC 表提供有关创建或更新多源计算规则的详细信息。

表 15-8 HPM_STGD_CALCRLUE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|----------------------|-----------------|--------------------|------|-----------------------------|
| id | identity | number(38,0) | 是 | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | varchar2(80 CHAR) | 是 | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 所选 POV 中第一个维成员的名称 |

表 15-8 (续) HPM_STGD_CALCRLULE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|------------------------|-----------------|--------------------|------|--|
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | number | number(4,0) | 是 | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | varchar2(255) | | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | varchar2(1 CHAR) | 是 | |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 是 | 要与此计算规则关联的动因名称。 |
| src_assign_rule_name1 | nvarchar(80) | varchar2(80 CHAR) | 是 | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name2 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name3 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name4 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name5 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的源分配规则。 |
| dest_assign_rule_name1 | nvarchar(80) | varchar2(80 CHAR) | 是 | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name2 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name3 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name4 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name5 | nvarchar(80) | varchar2(80 CHAR) | | 要与此计算规则关联的目标分配规则。 |
| last_upload_date | timestamp(6) | timestamp | | 此列将在导入时填充，显示执行导入的最近日期。 |
| import_exception | nvarchar(255) | VARCHAR2(255 CHAR) | | 包含错误代码，指示导入行时是否存在问题。Null 表示导入成功。 |
| created_userid | nvarchar(32) | varchar2(32 CHAR) | | 指出在此表中创建行的初始用户。 Profitability and Cost Management 不使用此字段。 |

表 15-8 (续) HPM_STGD_CALCRLUE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 是否必需 | 说明 |
|--------------------|-----------------|-------------------|------|--|
| created_timestamp | timestamp(6) | timestamp | | 指出创建行的初始日期/时间。 Profitability and Cost Management 不使用此字段。 |
| modified_userid | nvarchar(32) | varchar2(32 CHAR) | | 指出在此表中创建行的初始用户。 Profitability and Cost Management 不使用此字段。 |
| modified_timestamp | timestamp(6) | timestamp | | 指出创建行的初始日期/时间。 Profitability and Cost Management 不使用此字段。 |

16

导出明细 Profitability 的模型定义数据

另请参阅：

- [关于导出明细 Profitability 的模型定义数据](#)
创建模型之后，可以查询数据库以将模型对象作为数据库视图中的输出显示。
- [HPM_EXPDP_STAGE](#)
HPM_EXPDP_STAGE 视图检索阶段名称和顺序、应用程序名称、源阶段和目标阶段维。
- [HPM_EXPDP_POV](#)
HPM_EXPDP_POV 视图检索视点 (POV) 中包含的维成员的每个组合的状态。
- [HPM_EXPDP_DRIVER](#)
HPM_EXPDP_DRIVER 视图检索有关动因的详细信息，包括动因类型、显示顺序和公式。
- [HPM_EXPDP_DRIVER_SEL](#)
HPM_EXPDP_DRIVER_SEL 视图将检索有关 POV 和动因选定项的详细信息。
- [HPM_EXPDP_DRIVER_EXCEP](#)
HPM_EXPDP_DRIVER_EXCEP 视图检索有关所选动因例外的 POV 和源阶段的详细信息。
- [HPM_EXPDP_ASGN_RULE_SEL](#)
HPM_EXPDP_ASGN_RULE_SEL 视图检索有关分配规则选定项的详细信息。
- [HPM_EXPDP_CALC_RULE_SINGLSRC](#)
HPM_EXPDP_CALC_RULE_SINGLSRC 视图检索有关单源计算规则的详细信息。
- [HPM_EXPDP_CALC_RULE_CALCMSRS](#)
HPM_EXPDP_CALC_RULE_CALCMSRS 视图检索有关计算的度量规则的详细信息。
- [HPM_EXPDP_CALC_RULE_MULTISRC](#)
HPM_EXPDP_CALC_RULE_MULTISRC 视图检索有关多源计算规则的详细信息。
- [为明细 Profitability 应用程序生成统计信息](#)
“明细应用程序模型计算统计信息”脚本可用于明细 Profitability 应用程序，生成对于理解和调试模型非常有用的模型和模型计算统计信息。

关于导出明细 Profitability 的模型定义数据

创建模型之后，可以查询数据库以将模型对象作为数据库视图中的输出显示。

管理员可以在系统数据库中创建用于反映临时表中使用的列的数据库视图，从而显示系统中存储的模型数据。

HPM_EXPD_STAGE

HPM_EXPD_STAGE 视图检索阶段名称和顺序、应用程序名称、源阶段和目标阶段维。

表 16-1 HPM_EXPD_STAGE

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------|-----------------|--------------------|---|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | Oracle Hyperion Profitability and Cost Management 数据库中使用的阶段 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 所选源阶段的名称 |
| description | nvarchar(255) | varchar2(255 CHAR) | 所选阶段的目的是说明 |
| stage_order | integer | number(38,0) | 生成和计算脚本过程中建模（分配和规则创建）时在模型中使用所选阶段的顺序位置。 |
| stage_prefix | nvarchar(80) | varchar2(80 CHAR) | 区分阶段名称的前缀。 |
| driver_dim_name | nvarchar(80) | varchar2(80 CHAR) | 所选动因维的名称 |
| Is_intra | nvarchar(1) | varchar2(1 CHAR) | 确定是否允许阶段内分配的标志： <ul style="list-style-type: none"> • "Y"（是）指定在模型阶段内允许阶段内分配。 • "N"（否）指定在模型阶段内不允许阶段内分配。 |
| dim1_name | nvarchar(80) | varchar2(80 CHAR) | 阶段中第一个维的名称 |
| dim2_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第二个维的名称（如果可用）。 |
| dim3_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第三个维的名称（如果可用）。 |
| dim4_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第四个维的名称（如果可用）。 |
| dim5_name | nvarchar(80) | varchar2(80 CHAR) | 包含在阶段中的第五个维的名称（如果可用）。 |
| dim6_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim7_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim8_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim9_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim10_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim11_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim12_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim13_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim14_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim15_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim16_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim17_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |

表 16-1 (续) HPM_EXPD_STAGE

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------|-----------------|--------------------|-----------|
| dim18_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim19_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim20_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim21_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim22_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim23_name | nvarchar(80) | vvarchar2(80 CHAR) | 目标阶段中维的名称 |
| dim24_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |
| dim25_name | nvarchar(80) | varchar2(80 CHAR) | 目标阶段中维的名称 |

HPM_EXPD_POV

HPM_EXPD_POV 视图检索视点 (POV) 中包含的维成员的每个组合的状态。

表 16-2 HPM_EXPD_POV

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|---|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 |
| pov_state | nvarchar(80) | varchar2(80 CHAR) | POV 的当前状态： <ul style="list-style-type: none"> 草稿 已发布 已归档 |

HPM_EXPD_DRIVER

HPM_EXPD_DRIVER 视图检索有关动因的详细信息，包括动因类型、显示顺序和公式。

表 16-3 HPM_EXPD_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------|-----------------|--------------------|-------------|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| name | nvarchar(80) | varchar2(80 CHAR) | 所选动因的名称 |
| description | nvarchar(255) | varchar2(255 CHAR) | 对所选动因的用途的说明 |

表 16-3 (续) HPM_EXPD_DRIVER

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|-------------------------|-----------------|---------------------|---|
| display_order | integer(38,0) | integer(38,0) | 显示动因在模型中由所有动因组成的列表内的位置 |
| formula | nvarchar(4000) | varchar2(4000 CHAR) | 用户为动因创建的公式。此公式必须使用 SQL 语法创建，并且从数学方面来看是正确的运算。 |
| allow_idle_cost | nvarchar(1) | varchar2(1 CHAR) | 确定此动因是否允许闲置成本的标志： <ul style="list-style-type: none"> “Y” (是) 指定此动因允许闲置成本。 “N” (否) 指定此动因不允许闲置成本。 |
| priority | integer | number(38,0) | 输入动因的计算优先级，以便按指定顺序运行阶段内的分配。 首先处理最低优先级的动因。默认情况下，最低优先级设置为 100，最高优先级为 1。编号不必连续。 同一优先级的动因不按特定顺序执行。只有正整数有效。 |
| volume_formula | nvarchar(4000) | varchar2(4000 CHAR) | 仅适用于基于比率的动因，同时定义了比率和数量公式。 |
| dst_measure_member_name | nvarchar(2000) | varchar2(2000 CHAR) | 维表中度量的名称 |
| tdv_measure_member_name | nvarchar(80) | varchar2(2000 CHAR) | 在分配公式中用作分母的度量： $Driver\ Value / TotalDriverValue (DV / TDV)$ |
| operation_type | nvarchar(80) | varchar2(80 CHAR) | 为动因选择的运算类型： <ul style="list-style-type: none"> 基于比率的分配 基于比例的分配 计算的度量 |

HPM_EXPD_DRIVER_SEL

HPM_EXPD_DRIVER_SEL 视图将检索有关 POV 和动因选定项的详细信息。

表 16-4 HPM_EXPD_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|-------------------|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |

表 16-4 (续) HPM_EXPD_DRIVER_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------------|-----------------|-------------------|--------------------------|
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 (如果适用) |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 (如果适用) |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 (如果适用) |
| driver_dim_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选动因维的名称 |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 所选动因的名称 |

HPM_EXPD_DRIVER_EXCEP

HPM_EXPD_DRIVER_EXCEP 视图检索有关所选动因例外的 POV 和源阶段的详细信息。

表 16-5 HPM_EXPD_DRIVER_EXCEP

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|--------------------------|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 (如果适用) |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 (如果适用) |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 (如果适用) |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第一个维成员的名称 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第二个维成员的名称 (如果适用) |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第三个维成员的名称 (如果适用) |
| src_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第四个维成员的名称 (如果适用) |
| src_dim5_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第五个维成员的名称 (如果适用) |
| driver_name | nvarchar(80) | varchar2(80 CHAR) | 所选动因的名称 |

HPM_EXPD_ASGN_RUL_SEL

HPM_EXPD_ASGN_RULE_SEL 视图检索有关分配规则选定项的详细信息。

表 16-6 HPM_EXPD_ASGN_RULE_SEL

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|-------------------|-----------------------------|
| application_name | nvarchar(80) | varchar2(80 CHAR) | 所选应用程序的名称 |
| id | identity | integer(38,0) | 唯一记录 ID |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 |
| src_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第一个维成员的名称 |
| src_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第二个维成员的名称 |
| src_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第三个维成员的名称 |
| src_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第四个维成员的名称 |
| src_dim5_member_name | nvarchar(80) | varchar2(80 CHAR) | 源阶段中第五个维成员的名称 |
| rule_name | nvarchar(80) | varchar2(80 CHAR) | 控制所选分配的规则的名称。规则必须存在于目标数据库中。 |

HPM_EXPD_CALCRULE_SNGLSRC

HPM_EXPD_CALCRULE_SNGLSRC 视图检索有关单源计算规则的详细信息。

表 16-7 HPM_EXPD_CALCRULE_SNGLSRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|--------------------|-----------------------------|
| application_name | | VARCHAR2(80 CHAR) | 所选应用程序的名称 |
| id | identity | NUMBER(38,0) | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | VARCHAR2(80 CHAR) | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 |

表 16-7 (续) HPM_EXPD_CALCRLU_SNGLSRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|-------------------------|-----------------|-------------------|--|
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | integer | NUMBER(4,0) | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | VARCHAR2(255) | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | VARCHAR2(1 CHAR) | |
| first_sequence_priority | integer | NUMBER(38,0) | 在此规则将执行的动因范围内指定第一个动因序列优先级。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| last_sequence_priority | integer | NUMBER(38,0) | 在此规则将执行的动因范围内指定最后一个动因序列优先级。其数值必须大于等于 <i>first_sequence_priority</i> ，否则在插入期间将收到检查约束错误。 |
| all_drivers_flag | nvarchar(1) | VARCHAR2(1 CHAR) | 如果为 Y，在计算期间将忽略 <i>first_sequence_priority</i> 和 <i>last_sequence_priority</i> 字段中的值。 |

HPM_EXPD_CALCRLU_CALCMSRS

HPM_EXPD_CALCRLU_CALCMSRS 视图检索有关计算的度量规则的详细信息。

表 16-8 HPM_EXPD_CALCRLU_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|--------------------|-----------------------------|
| application_name | nvarchar(80) | VARCHAR2(80 CHAR) | 所选应用程序的名称 |
| id | identity | NUMBER(38,0) | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | VARCHAR2(80 CHAR) | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 |

表 16-8 (续) HPM_EXPD_CALCRLUE_CALCMSRS

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------------|-----------------|-------------------|---|
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | integer | NUMBER(4,0) | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | VARCHAR2(255) | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | VARCHAR2(1 CHAR) | |
| driver_name1 | nvarchar(1) | VARCHAR2(1 CHAR) | 要与此计算规则关联的动因名称。 |
| driver_name2 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的动因名称。 |
| driver_name3 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的动因名称。 |
| driver_name4 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的动因名称。 |
| driver_name5 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的动因名称。 |
| dest_assign_rule_name1 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name2 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name3 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name4 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name5 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |

HPM_EXPD_CALCRLUE_MULTISRC

HPM_EXPD_MULTISRC 视图检索有关多源计算规则的详细信息。

表 16-9 HPM_EXPD_CALCRLUE_MULTISRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|----------------------|-----------------|--------------------|-----------------------------|
| application_name | nvarchar(80) | VARCHAR2(80 CHAR) | 所选应用程序的名称 |
| id | identity | NUMBER(38,0) | 每个行的唯一标识符。唯一标识符可以来自任意序列生成器。 |
| name | nvarchar(80) | VARCHAR2(80 CHAR) | 计算规则的名称。 |
| description | nvarchar(255) | varchar2(255 CHAR) | 计算规则的长说明。 |
| pov_dim1_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第一个维成员的名称 |

表 16-9 (续) HPM_EXPD_CALCRULE_MULTISRC

| 字段 | SQL Server 数据类型 | Oracle 数据类型 | 说明 |
|------------------------|-----------------|-------------------|---|
| pov_dim2_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第二个维成员的名称 |
| pov_dim3_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第三个维成员的名称 |
| pov_dim4_member_name | nvarchar(80) | varchar2(80 CHAR) | 所选 POV 中第四个维成员的名称 |
| calculation_sequence | integer | NUMBER(4,0) | 指定相对顺序的数值，在计算期间将按该顺序来处理计算规则。其数值必须大于 0，否则在插入期间将收到检查约束错误。 |
| tag | nvarchar(255) | VARCHAR2(255) | 为此列的计算规则指定搜索标记。 |
| enabled_flag | nvarchar(1) | VARCHAR2(1 CHAR) | |
| driver_name | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的动因名称。 |
| src_assign_rule_name1 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name2 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name3 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name4 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的源分配规则。 |
| src_assign_rule_name5 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的源分配规则。 |
| dest_assign_rule_name1 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name2 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name3 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name4 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |
| dest_assign_rule_name5 | nvarchar(80) | VARCHAR2(80 CHAR) | 要与此计算规则关联的目标分配规则。 |

为明细 Profitability 应用程序生成统计信息

“明细应用程序模型计算统计信息”脚本可用于明细 Profitability 应用程序，生成对于理解和调试模型非常有用的模型和模型计算统计信息。

该脚本会在 Oracle Hyperion Profitability and Cost Management 产品架构中创建一个表，存储有用的统计信息，可用于解决模型计算结果问题、描述模型的形状和特征，以及评估模型计算性能。

该 dpmodelcalcstats.sql 脚本位于

```
%EPM_ORACLE_HOME%  
\OPatch\HPCM_11_1_2_2_351_14845052\files\products\Profitability\database\C  
ommon\Oracle
```

(在安装 11.1.2.2.351 补丁之后)。该脚本当前仅适用于 Oracle 数据库安装。

该脚本设计为开放脚本并可在 SQL Developer 中运行。

要为请求的应用程序生成统计信息：

1. 在 Oracle SQL Developer 中打开 `dpmodelcalcstats.sql`。
2. 选择 Oracle Fusion Performance Management 产品架构的连接。
3. 将光标移至脚本的第一行。
4. 单击运行脚本或按 F5 键。
5. 输入要请求生成模型计算统计信息的应用程序的名称。

该脚本使用以下命名惯例在产品架构中创建表：

```
HPMD_<应用程序名称>_DIAGSTAT_<唯一编号>
```

以下简要说明 HPMD_BksDP20_DIAGSTAT_4129 表中各列：

APPLICATION：应用程序的名称。

ITEM：该行中内容的简短说明。

RATIO_DRV_TOT：与所有基于比例的动因关联的项的总数。

RATE_DRV_TOT：与所有基于比率的动因关联的项的总数。

OBJC_DRV_TOT：与所有对象计算动因关联的项的总数。

TOTAL：与所有分配关联的项的总数。

NAME：与此项关联的对象的名称。

此表中包含以下类型的诊断统计信息项：

分配表：分配表的名称，其中有指定的应用程序的所有分配详细信息。如果计数大于 1，则结果可能不正确，因为尚未运行清除计算。

分配的数量：在模型中标识的指定应用程序的分配数量。

源节点的数量：在模型中标识的源节点（交叉点）的数量。

包含数据的分配的数量：其源阶段业务对象中有一行或多行与潜在源节点匹配的分配的数量。

行更新的数量：更新语句对所有分配执行的修改的总行数。

分配规则的数量：模型中定义的至少有一个分配引用的分配规则对象的数量。

动因的数量：模型中定义的至少有一个分配引用的动因对象的数量。

目标行计数的最大值：目标中由分配更新的行数的最大值。

目标行计数的中间值：目标中由分配更新的行数的中间值。标准

目标行计数的偏差：目标中由分配更新的行数的标准偏差。

动因计数：至少被一个分配引用的每个动因都有一个动因计数项。此项包含动因对象的名称和分配引用该动因的次数。

分配规则计数：至少被一个分配引用的每个分配规则都有一个分配规则计数项。此项包含分配规则对象的名称和分配引用的该规则的次数。

第 V 部分

使用管理分类帐 Profitability

另请参阅：

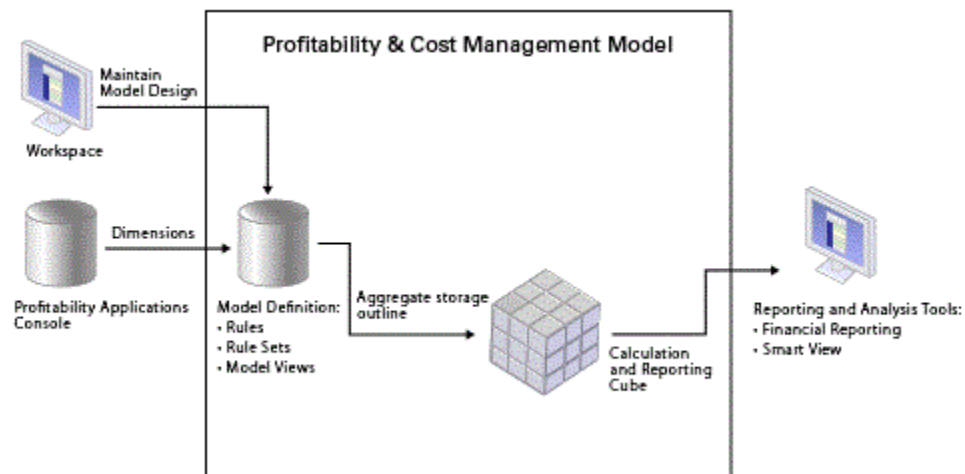
- [了解管理分类帐 Profitability 应用程序体系结构](#)
- [了解管理分类帐维](#)
- [执行其他应用程序管理任务](#)

了解管理分类帐 Profitability 应用程序体系结构

通过 Oracle Hyperion Enterprise Performance Management Workspace 访问的 Oracle Hyperion Profitability and Cost Management 是一个位于 Oracle Essbase 之上的分析工具。

通过 Profitability and Cost Management，业务用户可以针对获利能力和成本管理创建其业务模型，并使用该模型信息创建可针对获利能力和成本分析进行微调的 Essbase 数据库，而无需了解脚本语言。有关管理分类帐 Profitability 的说明，请参阅“[管理分类帐 Profitability](#)”。

图 17-1 管理分类帐 Profitability 产品体系结构



EPM Workspace 的模型元数据在 Profitability and Cost Management 模型的创建中使用，然后计算得到的结果可以在各种报表和分析工具中输出。

Profitability and Cost Management 利用 Oracle Hyperion Shared Services 对应用程序元数据和安全性进行集中管理。

应用程序管理员使用 Profitability 应用程序控制台创建 Profitability and Cost Management 维。使用 Shared Services 对用户访问进行集中管理。准备好维元数据后，可将其部署到 Profitability and Cost Management 应用程序或模型。

模型设计包含生成 Essbase 大纲所需的信息和模型的 Essbase 组件所需的计算脚本。每个模型都需要访问下列数据库：

- 用于存储模型设计的关系数据库
- 包含用于计算和报表的聚合存储 (ASO) 数据库的 Essbase 数据库。



只需一个数据库即可存储多个模型。

可以在报表工具和分析工具（如 Oracle Hyperion Financial Reporting 和 Oracle Smart View for Office）中查看结果。

了解管理分类帐维

另请参阅：

- [关于管理分类帐维](#)
Oracle Hyperion Profitability and Cost Management 使用在 Oracle Essbase 和 Profitability 应用程序控制台中创建的维和成员来代表业务模型的许多结构元素。
- [管理分类帐系统维](#)
管理分类帐应用程序必须包含两个系统维：规则维和余额维。
- [管理分类帐业务维](#)
业务维描述模型中的业务特定对象，如产品、客户、区域和员工。
- [POV 维](#)
POV 维指示特定视点或模型版本，如年份、方案或期间。
- [属性维](#)
属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。
- [别名维](#)
别名是指替代的名称、说明、语言或其他项，用于帮助定义维。
- [管理分类帐维排序顺序](#)
“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。

关于管理分类帐维

Oracle Hyperion Profitability and Cost Management 使用在 Oracle Essbase 和 Profitability 应用程序控制台中创建的维和成员来代表业务模型的许多结构元素。

维类型是一种允许使用预定义功能的维属性。维类型的特定特性可管理维的行为和功能。由于 Profitability and Cost Management 和其他 Oracle Hyperion Enterprise Performance Management Workspace 产品可能共享某些维类型，因此您可以对不同的产品使用维的功能。

管理分类帐 Profitability 应用程序具有以下维：

- [管理分类帐系统维](#)
- [管理分类帐业务维](#)
- [POV 维](#)
- [属性维](#)
- [别名维](#)

管理分类帐维要求

数据库大纲为模型提供数据结构，并包括计算说明和公式。Essbase 大纲中的维为层次结构。数据存储在维交叉点。以下是管理分类帐 Profitability 维要求：

- 应用程序或模型必须至少包含一个 POV 维，最多可以包含四个 POV 维。

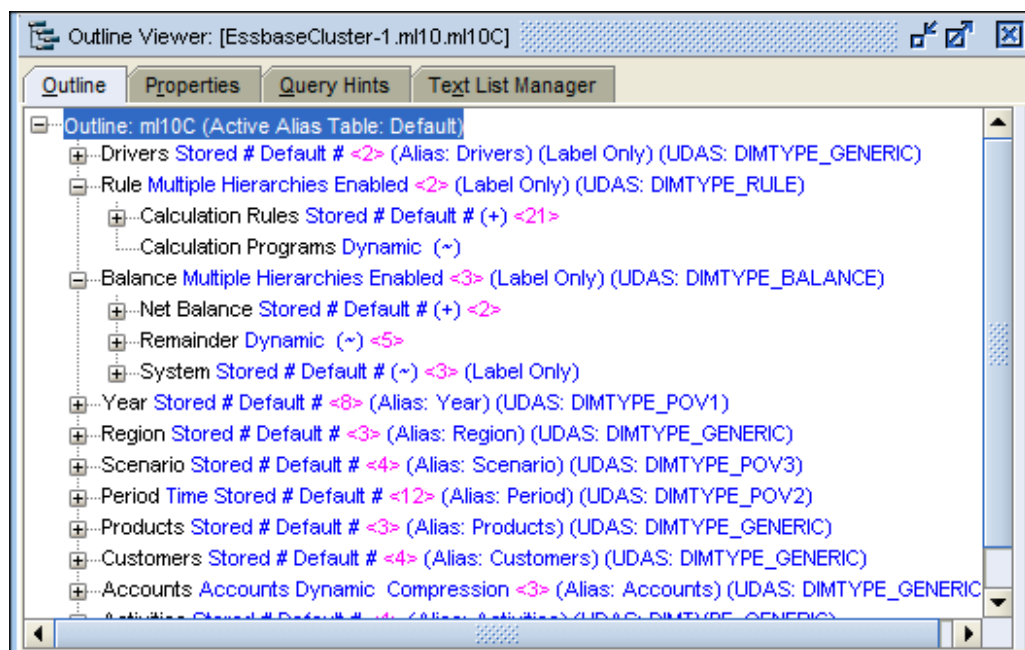
- 应用程序必须包含且仅包含一个名为规则的系统维。
用户可以编辑更多成员并将其添加到规则维中，例如 R1001 至 R1500。用户也可以选择删除该维。规则维中的“计算程序”成员是不可编辑的。
- 应用程序必须包含且仅包含一个名为余额的系统维。
无法编辑余额维中的系统维成员。不过，用户可以添加备用层次。
- 应至少有一个业务维并且在业务维的主层次中不存在重复成员。

▲ 注意：

在同一个维中成员不得重复。不过，成员可以在多个维之间重复。

图 1 显示了管理分类帐 Profitability 数据库的示例 Essbase 大纲（显示在 Essbase 控制台上）。

图 18-1 管理分类帐 Profitability 数据库的 Essbase 大纲



管理分类帐系统维

管理分类帐应用程序必须包含两个系统维：规则维和余额维。

部署或创建新的管理分类帐应用程序时，从 Profitability 应用程序控制台填充这些系统维。有关规则维和余额维的其他信息，请参阅列出的几节。

有关创建和维护维和成员的详细说明，请参阅“使用 Profitability 应用程序控制台创建应用程序”和《Oracle Essbase Database Administrator's Guide》。

注：

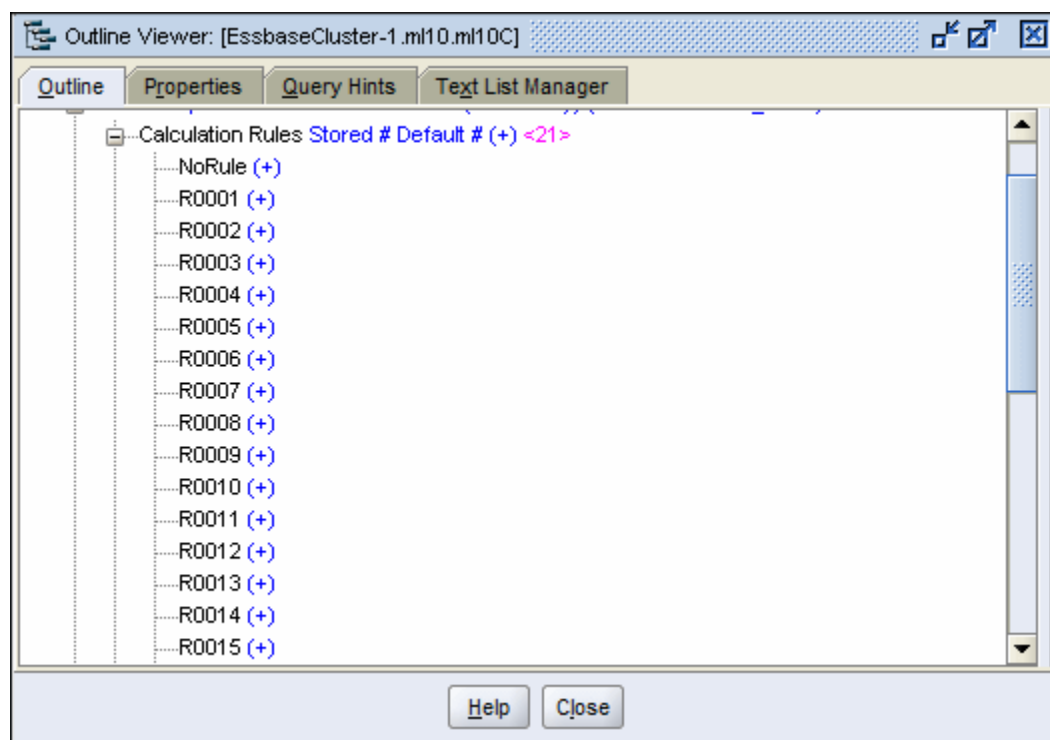
管理分类帐规则维和余额维是初始设置的系统维，不得以任何方式对其进行编辑，即使系统的某一部分允许执行此操作（例如，Profitability 应用程序控制台中的“更新维”）。这些维保留供系统使用。

管理分类帐规则维

规则维包含管理分类帐应用程序计算规则的定义。

图 1 显示了 Oracle Essbase 控制台中规则维的大纲。此图显示了“计算规则”成员。

图 18-2 管理分类帐规则维大纲



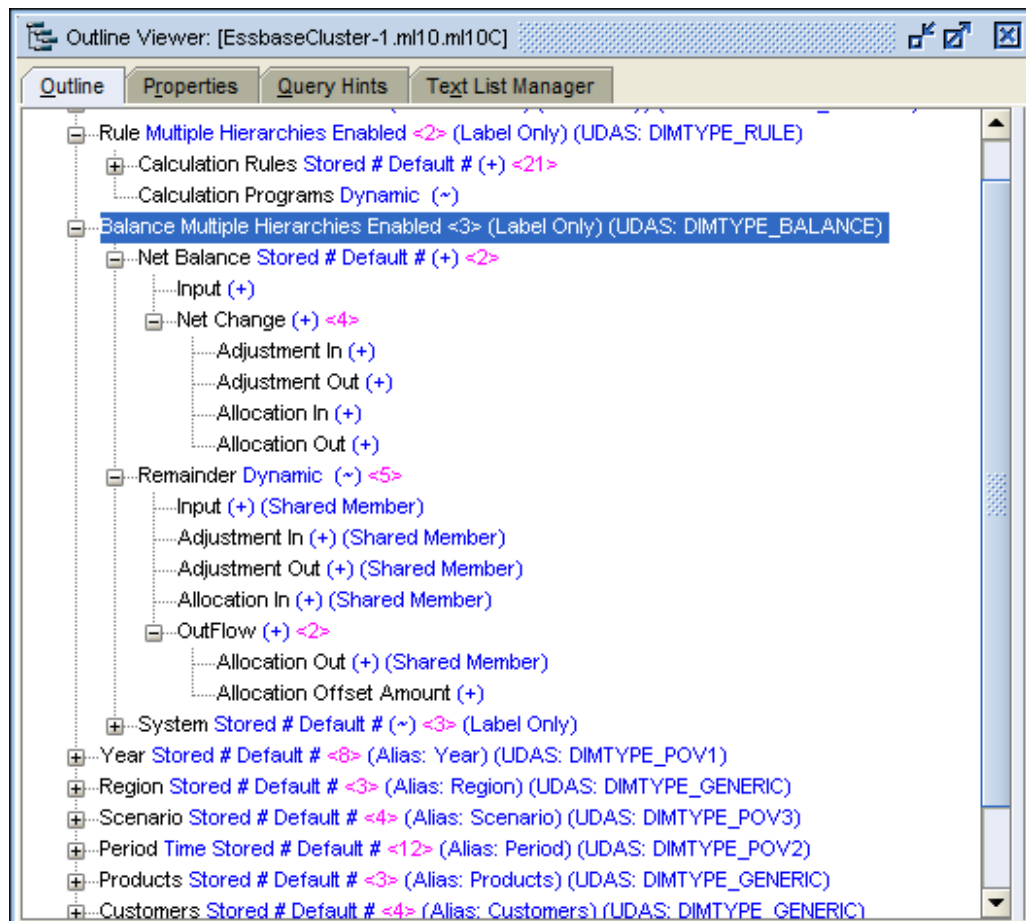
用户可以删除规则并将规则添加到“计算规则”，但用户只能应用 NoRule 规则。所有其他规则保留供系统使用。

图 1 显示了一个附加成员“计算程序”。该成员由系统进行控制，用户无法对其进行编辑。

管理分类帐余额维

图 1 显示了 Oracle Essbase 控制台中余额维的大纲。

图 18-3 管理分类帐余额维大纲



用户可以将数据添加到“净余额”的“输入”成员。其余的成员反映由规则集和规则确定的输入和输出。调整是动因计算的结果，分配是规则分配的结果，偏移是规则偏移定义的结果。有关规则集、规则及其定义的信息，请参阅《Oracle Hyperion Profitability and Cost Management 用户指南》。

可在“规则平衡”屏幕中查看存放在具有这些成员的交叉点中的数据。

管理分类帐业务维

业务维描述模型中的业务特定对象，如产品、客户、区域和员工。

业务维可能会使用以下部分或全部维类型，并且可能会应用于一个或多个模型：

- 常规
- 帐户
- 实体
- 国家/地区

部署 Oracle Essbase 大纲后，在 Oracle Hyperion Profitability and Cost Management 应用程序中创建的业务维是没有类型的基本维或常规维。此功能使 Profitability and

Cost Management 可重用为其他应用程序（例如 Oracle Hyperion Planning）定义的维成员和层次。

 注：

该维类型不适用于聚合存储大纲。

管理分类帐规则维和余额维是初始设置的系统维，不得以任何方式对其进行编辑，即使系统的某一部分允许执行此操作（例如，Profitability 应用程序控制台中的“更新维”）。这些维保留供系统使用。

业务维要求

创建管理分类帐应用程序的业务维时，应遵循下列要求：

- Gen1 维名称下的第一个 Gen2 子代通常设置为 All 成员。例如，部门维的 AllDepartments。
主层次在第一个 Gen2 子代下。分配模型只使用第一个 Gen2 层次，并且此层次不能包含共享成员。
- 其他 Gen2 成员可以容纳备用层次，但是这些层次不可用于构造分配模型。如果维要容纳备用层次，请将 DimensionHierarchyType 设置为 "Enabled"，将第一个 Gen2 成员的 HierarchyType 设置为 "Stored"，并将具有包含共享成员的备用层次的 Gen2 成员设置为 "Dynamic"
这些备用层次在 Profitability and Cost Management 建模屏幕中不可见，只能在 Essbase 中看到。

 注：

在管理分类帐业务维中，不要创建引用非 0 级基本成员的共享成员。如果这样做，将会导致复制 POV 和计算操作失败，因为在此类引用条件下共享会使得 Profitability and Cost Management 需要将数据写入非 0 级成员，而在 Essbase 中 ASO 多维数据集不支持此操作。

POV 维

POV 维指示特定视点或模型版本，如年份、方案或期间。

可以自定义该维以反映组织的要求。例如，POV 可以包括季度、月份、季节性分组等等。

每个模型至少需要一个 POV 维，但最多可以创建 4 个 POV 维。

此外，还提供了版本维，该维用于创建模型的另一个实例。可以修改此版本以使用多种策略和业务选项来试验“假设分析”方案。通过修改此版本，可以实施多种特性，并比较结果以确定最佳操作。

属性维

属性维是一种与业务维相关联的特殊类型的维，可以使用属性维包含的成员对其他关联维的成员进行分类。

属性维描述数据的特性，如产品的大小和颜色。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还可用于在创建分配规则时筛选目标交叉点。



注：

有关命名限制，请参阅“[Essbase 命名约定](#)”。

有两种属性维类型：

- 属性维：
 - 该属性可以使用不同的结构（如布尔、日期、数字和文本）创建。
 - 一个属性具有一个层次，可以上滚层次来获得聚合值。
 - 只能将给定属性维的一个属性与一个成员关联。
- 用户定义属性维 (UDA)：
 - 该属性只能使用文本创建。
 - UDA 没有层次，不能在报表中轻松地用来生成合计。
 - 一个成员可以与多个 UDA 相关联。

每种类型的属性维都提供了不同的优势，具体取决于您的模型和报表的需求。有关使用属性维的详细信息，请参阅《*Oracle Essbase Database Administrator's Guide*》。

您可以使用这些属性基于维成员的属性或品质分析数据。属性维还用于在创建分配规则时筛选目标交叉点。

有关命名限制，请参阅“[Essbase 命名约定](#)”。

别名维

别名是指替代的名称、说明、语言或其他项，用于帮助定义维。

例如，您可以引用系统中的客户编号，还可以分配一个在屏幕上显示公司名称的别名，从而更容易标识该客户。您可以为帐户、货币、实体、方案、期间、版本、年度和用户定义的维成员分配一个或多个别名。

安装完成后，即可使用“默认”别名表。重新部署后，您可以在使用管理分类帐成员选择器的所有屏幕（如“规则定义”屏幕）上查看别名。

使用管理分类帐别名维时，请记住以下几点：

- 不允许在同一个维中出现重复的成员名称或别名。

- 将维文件导入管理分类帐应用程序时，如果您在标题中定义别名表，则必须为维中的所有成员定义别名值。别名值不必与原始成员名称不同，但是您需要在文件中每个成员行上为每个别名表指定值。

管理分类帐维排序顺序

“维排序顺序”属性控制由 Oracle Hyperion Profitability and Cost Management 生成的 Oracle Essbase 大纲中的维顺序。

除了别名维和 UDA 维以外，模型中的所有维都必须设置维排序顺序。

▲ 注意：

如果维的排序顺序保留为空，验证将失败。

在维加载文件中设置“维排序顺序”属性。有关说明，请参阅[“设置维排序顺序”](#)。

维的排序顺序建议

Oracle 建议使用以下建议方法设置维的排序：

- 必须为模型中的每个维（别名维和 UDA 维除外）设置维排序顺序。

✎ 注：

维排序顺序会忽略别名维和 UDA 维，因为它们不以维的形式存在于 Oracle Hyperion Profitability and Cost Management 和 Oracle Essbase 中。

- 维排序顺序必须连续、唯一且大于或等于 1。
- 度量维默认设置为 1。
- AllocationType 维默认设置为 2。
- 业务维和 POV 维必须设置为 3 或更高。
- 属性维在排序时必须始终排在最后。例如，如果在由 12 个维组成的序列中有 4 个属性维，则这些属性维必须设置为 9、10、11 和 12。

设置维排序顺序

必须在维级别使用维加载文件中的维排序顺序属性设置模型中每个维的处理顺序。

必须遵循维排序顺序限制；否则，模型验证将失败。有关限制的完整列表，请参阅[“维的排序顺序建议”](#)。

要构造包括维排序顺序的加载文件：

1. 在外部程序或 ETL 流程中，构造加载文件，其中包含类似如下的列。确保包括对应于“维排序顺序”的列。

为维的根成员输入排序顺序。在此示例中，Accounts 的排序顺序为 9。

| Regular | Parent | Storage Type | Hierarchy Type | Comment | ASO Data Storage | BSO Data Storage | Two Pass Calculation | ASO | | Consolidation Type | UDA | Dimension Solve Order | Dimension Sort Order | Member Solve Order | Alias:English | Alias:Default |
|-----------------------|--------|--------------|----------------|---------|------------------|------------------|----------------------|-------------------|----------------|--------------------|-----|-----------------------|----------------------|--------------------|----------------|----------------|
| | | | | | | | | Dimension Formula | Dimension Type | | | | | | | |
| Accounts | | SPARSE | STORED | | LABELONLY | LABELONLY | N | | | | | 0 | 9 | | ALIAS:English | ALIAS:Default |
| AllAccounts | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | AllAccounts | AllAccounts |
| "NoAccour AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | NoAccounts | NoAccounts |
| "Cost of Gr AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Cost of Goods | Cost of Goods |
| "Material Cost of Gc | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Materials | Materials |
| "MAT51/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Frames | Frames |
| "MAT52/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Wheels | Wheels |
| "MAT53/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Gearsets | Gearsets |
| "MAT54/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Seats | Seats |
| "MAT55/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Controls | Controls |
| "MAT56/ Materials | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Tubing | Tubing |
| "Equipme Cost of Gc | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Equipment De | Equipment Dep |
| "DEP610 Equipmen | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Bar Bender De | Bar Bender Dep |
| "Personne AllAccoun | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Personnel Exp | Personnel Expe |
| "PER2100 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Regular Salary | Regular Salary |
| "PER2200 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Regular Wage | Regular Wages |
| "PER2300 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Overtime | Overtime |
| "PER2400 Personnel | | SPARSE | STORED | | StoreData | StoreData | N | | + | | | | | 0 | Vacation | Vacation |

- 在 Profitability 应用程序控制台中，依次选择“操作”和更新维以加载定义了维排序顺序的文件。

19

执行其他应用程序管理任务

另请参阅：

- [管理管理分类帐 Profitability 应用程序](#)
- [使用向导验证和部署管理分类帐应用程序](#)
创建 Oracle Hyperion Profitability and Cost Management 应用程序后，必须验证该应用程序并将其部署到 Profitability and Cost Management。

管理管理分类帐 Profitability 应用程序

另请参阅：

- [使用管理分类帐应用程序和维](#)
- [Profitability and Cost Management 维和成员属性](#)
- [设置层次类型属性](#)

使用管理分类帐应用程序和维

Oracle Hyperion Profitability and Cost Management 使用 Profitability 应用程序控制台选择维以构建用于收益模型的 Oracle Essbase 大纲。在 Profitability 应用程序控制台中创建所有维和成员，并导入到 Profitability and Cost Management 应用程序来构建模型。

Profitability and Cost Management 维和成员属性

Oracle Hyperion Profitability and Cost Management 维和成员的属性按字母顺序显示在标准 Profitability and Cost Management 维和成员属性表中，其中显示以下信息：

- 属性标签，为属性提供可读性更好的显示名称。如果适用，关联的数据库类型会附加到名称的末尾（ASO 或 BSO）。如果未指定数据库类型，则属性同时适用于两种类型。
- 各属性的说明
- 属性名称，为属性提供唯一标识符，在更新导入和批处理客户端中的数据时使用。

在维库中选择属性时，可以修改由下拉列表或数据输入文本框显示的任何属性。

▲ 注意：

Profitability and Cost Management 的所有属性均为本地值。如果修改一个层次中的属性，不能认为其他层次也会继承该值。

设置层次类型属性

“层次类型”属性仅适用于 Essbase 中的聚合存储数据库。

要使用 ADDITION 和 IGNORE 以外的其他合并类型符号，则层次类型应设置为 "DYNAMIC"。



注：

如果维成员具有公式，则层次类型必须设置为 "DYNAMIC"。

要设置层次类型：

1. 从 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、管理和维库。
2. 在应用程序下，选择一些维（别名、属性、UDA 除外）。
3. 在属性网格中，选择相应的层次类型属性：
 - 为维成员使用下列合并符号的所有维选择已存储：
 - + ADDITION
 - ~ IGNORE（仅位于 LABEL ONLY 成员下方。）
 - 为维成员需要使用所有受支持的合并符号或具有公式的所有维选择动态。
4. 将应用程序重新部署到 Oracle Hyperion Profitability and Cost Management，然后重新部署到 Oracle Essbase。

使用向导验证和部署管理分类帐应用程序

创建 Oracle Hyperion Profitability and Cost Management 应用程序后，必须验证该应用程序并将其部署到 Profitability and Cost Management。

如果使用应用程序向导，则在最终完成应用程序的创建之前，先验证应用程序并纠正任何错误。如果需要，您可以略过向导中的验证和部署选项，之后再使用应用程序库中的部署选项。

要使用向导验证并部署应用程序：

1. 在应用程序向导的应用程序设置屏幕上，单击验证。
所有错误（包括错误类型和错误消息）均显示在网格中。
2. 可选：要在验证后部署应用程序，请选择完成后部署。
3. 如果出现验证错误，则在部署应用程序前更正。
4. 单击完成。

如果选择了“完成后部署”，则显示“部署”对话框。



注：

如果更改共享维，则使用该共享维的所有应用程序都会受到影响。要使更改生效，必须重新部署应用程序。

5. 部署应用程序。

A

应用程序设计最佳实践

管理应用程序可扩展性

分配流程会使初始数据的大小扩展许多次。

分配规则提取叶级别的初始数据值并将值拆分到所有目标单元格中。不难发现，在执行分配规则之后，数据会增长到其初始大小的数百倍。许多分配规则可以将这些较小的金额拆分成许多更小的值。

为了避免在模型计算结束时处理数百万个微小数据点，需要在设计流程的早期考虑数据规模。如果要将 Oracle Hyperion Profitability and Cost Management 结果与下游报告系统集成，或生成发送回 GL 的数据摘要，扩展出来的数据会使这些处理大大复杂化。

维计数

为了使数据增长受到控制，最佳方法是在设计阶段的早期考虑会影响数据规模的因素（尤其是处于分配后状态时）。对于可扩展性，应用程序中所用维的数量是要考虑的第一个因素。

在理由不充分的情况下，请在添加更多维时一定要慎重。Oracle Hyperion Profitability and Cost Management 中的数据增长主要受到数据拆分次数的影响，在拆分过程中，数据会变成越来越小的值。在添加新维之前，请验证是否需要新的物理维。Profitability and Cost Management 设计人员起初通常采用源数据中存在的所有维，只是因为这些维是现成的并且有一天可能需要用到。

请考虑维对于报告最终结果或区分数据以支持分配流程是否必需。如果对于这两个功能都不是必需的，强烈建议您去除该维。

如果附加维实际上是现有维的另一种表达方式，请考虑改用备用层次或属性维。通过这种方式，可以报告所需类别而不增加数据大小。

尽管强烈建议限制维，但最好为将来的增长添加维。只要添加的维在所有的模型对象中仅使用单个“非成员”选择，“稀疏”维对性能的影响就很小。

维成员计数

Oracle Hyperion Profitability and Cost Management 中的数据规模是维数和维大小（成员计数）的函数。规则范围的大小受维计数和成员计数影响。

在设计应用程序时，明智的做法是慎重地使用非常大的维。当维较少而且较小时，Profitability and Cost Management 应用程序工作性能最佳。这并不是说 Profitability and Cost Management 无法处理较大的稀疏维，而是使用许多大型维可能会导致较大的稀疏范围，如果您设计了要在此范围中使用的规则，就会导致计算性能问题。

首先，对于维中的成员详细信息，使用选择维时的相同原理：

- 进行报告时是否需要详细信息？
- 为了支持分配逻辑，是否需要详细信息来区分数据？

如果这两个问题的答案都为否，则应当考虑在维中使用聚合度更高的成员。例如，使用“成本池”而非最低级别的帐户，或者使用“产品系列”而非 SKU。

属性和 UDA

对于报告逻辑和分配逻辑而言，属性是功能强大的工具；但有时，管理基本成员和属性成员之间的关联可能会变得非常麻烦。当您使用非常多的基本维（大于 10,000 个）或者非常多的唯一属性值时，请使用 UDA 而非属性。

数据规模管理

一定要在设计周期的早期意识到不受控制的数据增长可能产生的负面影响。千万不要为了支持可能出现的报告需求而力求向应用程序中添加每一个细节。请在开始设计应用程序之前，了解应用程序的报告需求。只有通过这种方式，设计人员才能回答“我们确实需要这么详细吗...？”这样的问题。

在了解了这些情况之后，可使用两种主要设计方法来限制增长：

- [汇总起始数据](#)
- [汇总分配中点](#)

汇总起始数据

并非所有的数据都真正需要使用。如前所述，在数据加载之前，在数据中使用汇总点是最好的工具之一。帐户详细信息是常见的示例。不在最低的详细程度加载费用，而是改用聚合成本池。尽可能针对数据中的每个维使用此策略。请参考前面那些与报告或分配流程所需的详细信息有关的问题。

通过询问以下问题，确定报告或分配流程是否需要详细信息：

- 进行报告时是否需要详细信息？
- 为了支持分配逻辑，是否需要详细信息来区分数据？

仅这一步就可以将起始数据大小缩小一个或多个数量级。

汇总分配中点

与起始数据点类似，在应用程序中也有自然的漏斗点，在这些点处，不再需要先前分配步骤中使用的详细程度。在这些情况下，在应用程序的中间阶段使用池化策略有助于将数据重新收缩到更易于管理的大小。

这在将财务数据分配到面向市场的维（例如“客户”、“区域/地区”、“产品/SKU”）之前的阶段特别有用。面向市场的维通常是应用程序中最大的维，并且是数据爆炸通常发生的阶段。

如果您可以在运营中心分配后缩小数据规模，则可以在各个市场交叉点分配较少的数据点，最终得到较小的数据集。

管理维设计

维设计会对报告性能和计算性能产生很大的影响。

Oracle Hyperion Profitability and Cost Management 使用 Oracle Essbase 聚合存储 (Aggregate Storage, ASO) 数据库。ASO 数据库的性能特征和优化方法与其他聚合存储业务流程使用的块存储 (Block Storage, BSO) 数据库不同。

维类型

Oracle Essbase ASO 提供三个维类型：

- 存储
- 动态
- 启用多个层次

存储

通常，存储维提供的计算和报告性能最佳。

对存储维的支持

- 聚合速度最快
- 共享成员
- 属性和 UDA

限制

- 成员只能使用加法 (+) 合并运算符
- 无成员公式
- 共享成员必须出现在其基本成员之后

动态

动态维提供极高的设计灵活性，但代价是牺牲计算性能和报告性能。

对动态维的支持

- 任何合并运算符 (+、-、*、/、~)
- 成员公式
- 共享成员
- 属性和 UDA

动态维存在的限制

- 聚合速度最慢
这会对计算性能和报告性能产生影响。在一个应用程序中使用多个动态维会使此问题加重。
- 共享成员必须出现在其基本成员之后

启用多个层次

启用多个层次的维中混合了存储层次和动态层次，这允许某些维具有与存储维相似的操作和行为，而其他维具有与动态维相似的操作和行为

对启用多个层次的维的支持

- 多个存储层次和动态层次的组合
- 存储层次具有与存储维类型相同的功能支持和限制。
- 动态层次具有与动态维类型相同的功能支持和限制。

启用多个层次的维存在的限制

- 第 1 个层代 2 成员层次必须为存储层次类型。
- 根成员必须为仅标签类型的数据存储

层次顶级

维的顶级会对报告工具和 Oracle Hyperion Profitability and Cost Management 屏幕（如使用模型视图的“规则平衡”）的易用性产生影响。维通常设置为将数据按自然方式汇总到根成员，这样更便于在报表中使用维。

汇总和报告

只要有可能，请确保将主层次中的所有数据汇总到维的顶级成员。这样，如果用户希望查看该维的所有数据，就可以跳过在报表中进行明确的成员选择。这也有助于简化规则平衡和其他即席报告工具的使用。

无成员

对于具有不规则维数的数据，常用的数据混合方法是使用那些指定数据点在维中没有引用的成员。使用 "No" 成员（例如 "NoProduct"）时，可存储尚未指定给特定产品的数据。

在启用多个层次的维中使用 "No" 成员时，建议将 "No" 成员放在第 1 个层代 2 层次中。这可以确保维的所有数据点（包括尚未归入该维的数据）聚合到维的顶层，从而帮助简化报告流程。

最佳实践示例

以产品层次为例：

单一层次维

维根成员（产品）数据存储：存储数据

第 1 个层代 2 层次（产品总计）聚合运算符：加法

- 第 1 个子代（所有产品）聚合运算符：加法。产品层次的其余部分置于此成员之下。
- 第 2 个子代 (NoProduct) 聚合运算符：加法。如果数据点没有关联的产品成员，则使用此成员。

多层次维

维根成员（产品）数据存储：仅标签

- 第 1 个层次 2 层次（产品总计）聚合运算符：加法。
 - 第 1 个子代（所有产品）聚合运算符：加法。产品层次的其余部分置于此成员之下。
 - 第 2 个子代 (NoProduct) 聚合运算符：加法。如果数据点没有关联的产品成员，则使用此成员。
- 第 2 个层次 2 层次（产品总计）聚合运算符：忽略。
- 第 3 个层次 2 层次（产品总计）聚合运算符：忽略。
- ...第 N 个层次 2 层次（备用层次）聚合运算符：忽略。

重复的成员

重复成员是用于存储数据而且具有相同名称的基本成员（不是共享成员）。Oracle Hyperion Profitability and Cost Management 支持不同的维中有重复的成员，但不支持在同一个维中有重复的成员。

如果在成本从源组织单位移到目标组织单位期间跟踪成本，而且有必要报告在源和目标之间移动的成本，则重复的维和成员非常有用。

尽管 Profitability and Cost Management 支持重复的成员，但是使用此类成员会带来一定的挑战。在 Oracle Essbase 中，必须对包含数据的数据单元格进行唯一的标识。在不使用重复的成员时，系统使用成员名称组合来标识单元格。

如果有重复的成员，则需要进一步限定。例如，引用中还必须包括维（例如，“Entity.TotalEntity.Administration.Marketing”），而不只是引用数据单元格组织成员（例如，“Marketing”）。在创建报表和修改维时，尤其是移动成员时，重复成员所需的更精细限定可能会产生问题。在移动重复的成员时，它的完全限定名称已更改。这会导致数据库重新部署流程在维发生修改之后找不到正确的单元格，从而导致数据值被删除。

为了避免产生重复项，应当在成员前面加上一个或两个维，以避免创建重复项。

管理应用程序逻辑设计

在创建规则之前，先创建应用程序设计图。

另请参阅：

- [用于管理规则计数的简单策略](#)
- [分配规则类型和选项](#)
- [规则范围](#)
- [在规则集内使用并行执行](#)
- [自定义计算规则](#)

用于管理规则计数的简单策略

在开始创建规则之前，最好创建应用程序设计图。许多用户往往从旧系统中复制分配，而没有先分析分配规则的总范围并尝试合理化整个流程。在许多情况下，如果您重新审视整个流程并

考虑 Oracle Hyperion Profitability and Cost Management 提供的许多工具，则可以找到减少流程步骤，使流程更易于理解，并易于向他人解释的方法。

- 确定可以在何处使用单个规则来应用公用逻辑
- 使用“缺少动因数据”作为分配筛选器

确定可以在何处使用单个规则来应用公用逻辑

这通常是将旧分配规则迁移到 Oracle Hyperion Profitability and Cost Management 时首先遇到的难题，不过我们为此提供了简便的解决方案。如果您确定许多部门或实体公用分配模式，则可以将旧流程中的许多步骤合并到 Profitability and Cost Management 中的单个规则。

检查是否存在以下情况：逻辑的许多区域都需要分配到使用相同动因值的同一组维。在这些情况下，“目标”选项卡上的“与源相同”选项将允许您在单个规则中运行多个并行分配步骤。

使用“缺少动因数据”作为分配筛选器

如果您的目标范围内应当接收分配的成员分散在不应当接收分配的成员之间，请记住，仅仅因为一个成员是目标的一部分，该成员并不会自动接收所分配的金额。这些分配仅发送到具有动因值的目标。

利用分配引擎的此行为，您通过创建单个规则就可以实现以下目标：规则覆盖较大的源和目标范围，但仅将数据分配到它找到动因数据的交叉点。这样，您只需创建单个规则，而在旧系统中则可能必须创建许多规则。

分配规则类型和选项

每个规则中的分配关系都会影响 Oracle Hyperion Profitability and Cost Management 用于构造计算逻辑的方法，以及 Oracle Essbase 引擎执行计算逻辑的方法。从性能角度看，并非所有的规则类型都相同。

下面是主要规则类型（按性能排序）：

- 简单维扩展
- 复杂维扩展
- 简单重新归类
- 复杂重新归类
- 与源相同，维不同

简单维扩展

- 维的每个源与目标关系要么设置为与源相同，要么为一（单个 0 层源）对多（多个目标）关系。
- 每个动因成员选择都是级别 0。
- 每个目标维都是单个成员选择（级别 0 或父代）。
- “简单维扩展”规则类型是执行速度最快的类型。

复杂维扩展

- 维的每个源与目标关系要么设置为与源相同，要么为一（单个 0 层源）对多（多个目标）关系。
- 任何动因成员选择都是父成员。
- 任何目标维都有多个不同的成员选择。

简单重新归类

维的一个源与目标关系是多对多关系，无论多个不同成员选择或一个父成员选择是否定义了多个成员。维的其余源与目标关系都是与源相同。

复杂重新归类

维的多个源与目标关系是多对多关系，无论多个不同成员选择或一个父成员选择是否定义了多个成员。维的其余源与目标关系都是与源相同。

复杂的重新归类规则通常是执行速度最慢的规则类型。

与源相同，维不同

维的一个或多个源与目标关系使用“与源相同”选项，而源维和目标维不同。

此选项可与任何分配规则类型一起使用，这将增加执行复杂度并降低性能。

使用此选项时一定要小心，在应用程序构建期间一定要进行频繁的性能基准测试以确定所产生的影响是否可接受。

每个规则类型允许的分配规则和成员选择选项

表 A-1 每个规则类型允许的分配规则和成员选择选项

| | 源维 | 目标维 | 动因维 |
|--|---|---|---|
| 简单维扩展 (Simple Dimension Extension, SDE) | <ul style="list-style-type: none"> • 单个 0 级成员 • 任何其他成员选择（如果同一个维的“目标”与“源”相同） | <ul style="list-style-type: none"> • 至少一个与源相同 • 单个 0 级成员 • 任何其他成员选择（如果没有为同一个维选择“动因”） | <ul style="list-style-type: none"> • 平均分配 • 指定的动因位置是 0 级元组 |
| 复杂维扩展 (Complex Dimension Extension, CDE) | <ul style="list-style-type: none"> • 单个 0 级成员 • 任何其他成员选择（如果同一个维的“目标”与“源”相同） | <ul style="list-style-type: none"> • 至少一个与源相同 • 单个 0 级成员 • 任何其他成员选择 | <ul style="list-style-type: none"> • 平均分配 • 指定的动因位置是 0 级元组 • 指定的动因位置不是 0 级元组 |
| 简单重新归类 (Simple Reclassification, SR) | <ul style="list-style-type: none"> • 单个 0 级成员 • 任何其他成员选择 | <ul style="list-style-type: none"> • 与源相同 • 单个 0 级成员 • 任何其他成员选择（如果没有为同一个维选择“动因”） | <ul style="list-style-type: none"> • 平均分配 • 指定的动因位置是 0 级元组 |

表 A-1 (续) 每个规则类型允许的分配规则和成员选择选项

| | 源维 | 目标维 | 动因维 |
|---------------------------------------|--|---|---|
| 复杂重新归类 (Complex Reclassification, CR) | <ul style="list-style-type: none"> • 单个 0 级成员 • 任何其他成员选择 | <ul style="list-style-type: none"> • 与源相同 • 单个 0 级成员 • 任何其他成员选择 | <ul style="list-style-type: none"> • 平均分配 • 指定的动因位置是 0 级元组 • 指定的动因位置不是 0 级元组 |
| 与源相同, 维不同 (SAD) | <ul style="list-style-type: none"> • 单个 0 级成员 • 任何其他成员选择 | <ul style="list-style-type: none"> • 与源相同 • 与源相同, 维不同 • 任何其他成员选择 | <ul style="list-style-type: none"> • 平均分配 • 指定的动因位置是 0 级元组 • 指定的动因位置不是 0 级元组 |

规则范围

在规则设计中, 影响性能的另一方面是源范围和目标范围。认真管理规则范围可以提高性能。

- [范围大小和稀疏度](#)
- [使用源细分](#)
- [使用筛选器或备用层次](#)
- [避免使用负的动因数据值](#)
- [避免使用动态成员作为动因](#)

范围大小和稀疏度

通常, 源、目标或目标范围越大, 执行规则所需的时间就越长。有些规则能够在极大的范围内执行, 但有些则不能。规则类型对计算时间有较大的影响; 但是, 对于给定的规则/规则类型, 源和目标范围越小, 执行速度越快。在创建范围极大的规则时一定要慎重。您可以在“规则定义”屏幕或执行统计信息报表中查看范围。

使用源细分

减小源范围的一种方法是将一个规则拆分为多个较小的规则, 即使用相同的目标和动因基础定义并将源范围拆分成较小的部分。您可以通过手动使用多个规则来完成拆分, 也可以使用规则的源选项卡上的“源细分”选项让系统自动将规则执行拆分为较小的段。

使用筛选器或备用层次

在某些情况下, 需要包括在源或目标中的成员在原有层次中没有天然地组合在一起。要限制规则范围并使规则逻辑更易于理解, 一种方法是使用替代层次, 按照更便于规则使用的方式对成员进行分组。另一种方法是使用属性筛选器。

避免使用负的动因数据值

使用具有负值的动因数据, 会导致分配引擎生成不可用的意外结果。分配引擎使用在每个目标位置找到的动因值作为“动因值/所有动因值总和”比率中的分子。然后此分配比率乘以源值, 确定要应用于目标的 *AllocationIn* 金额。如果混合使用正的和负的动因值,

会产生分母不再准确的情况，可能导致规则结果中的 *AllocationIn* 金额增大和缩小。这些结果非常难以解释和说明，所以最好是完全避免这种情况。

避免使用动态成员作为动因

如果动因值需要公式，最好在执行分配规则之前计算值。在规则执行期间，规则会检查每个可能的目标位置，以确定该位置的动因值。如果动因引用计算的成员，则系统必须为每个可能的位置执行成员公式。这可能会导致数十亿次不必要的计算，从而严重影响规则的性能。

更好的解决方案是为规则准备动因数据，方法是在规则前加一个自定义的计算规则来执行所需的公式，并将动因值存储在存储成员中。这允许分配规则按照通常方式执行，并快速跳过没有动因数据的位置。

在规则集内使用并行执行

在某些情况下，以并行方式执行规则有助于提高总体性能。计算选项是在规则集级别设置的，会影响规则集内的所有规则。所有规则集的默认选项是按顺序处理。这意味着每个规则都单独执行，而规则集内的所有规则都是逐一执行的。

确保规则是并行安全的

一些规则通常会影响到其他规则的执行。如果一个规则分配到另一个规则的源范围中，则这些规则的运行顺序会对最终结果产生显著影响。这就是默认计算选项为“按顺序”的原因。

如果存在可以安全地并行执行的规则，则使用并行计算选项有助于提高性能。针对要启用此功能的一个或多个规则集设置并行处理选项。您还需要在“应用程序属性”屏幕上设置“最大并发线程数”选项。

试验线程数量

可以通过试验来查找合适的并发线程级别。性能在某个点后不再继续改进，使用更多的线程启动其他规则会导致性能下降。并行计算的最佳实践是测试规则集，从四个并发线程开始并逐渐增加线程数，直到性能不再改进为止。尽管并行处理对于未使用混合源范围的规则来说是绝佳选择，但这些规则的目标范围可能仍存在共用的位置。在这些情况下，Oracle Essbase 必须先暂停一个规则，直到另一个规则完成目标数据写入为止。因此，您将看到在某个点处，增加线程数不再有助于提高性能。

自定义计算规则

较大的目标范围对自定义计算规则的影响和对分配规则的影响相同。通常，范围越小，执行速度越快。在几乎所有情况下，在规则公式中使用 `NONEMPTYTUPLE` 语句可帮助提高性能，当规则的目标范围非常大时尤其如此。

B

将数据导入 Profitability and Cost Management

关于导入数据

您可以将数据直接输入到 Oracle Hyperion Profitability and Cost Management；不过，数据输入可能占用大量时间。

为便于应用程序的填充，您可以使用一组导入临时表和导入配置直接将数据导入 Profitability and Cost Management 标准和明细 Profitability 应用程序。

模型数据可以从多个源导入：

- 将模型定义数据导入到 Profitability and Cost Management 中。此数据可以在另一个应用程序（如 Excel）中创建，然后使用临时表导入。
- 从 Oracle Essbase 中只能为标准 Profitability 应用程序导入和导出模型和最终用户数据。请参阅《Oracle Essbase Database Administrator's Guide》。
- 可以使用 Oracle Hyperion Enterprise Performance Management System 生命周期管理导入模型数据和应用程序。请参阅《Oracle Enterprise Performance Management System 生命周期管理指南》。

▲ 注意：

Oracle 建议在导入数据之前，先在 Oracle Hyperion Enterprise Performance Management Workspace 和 Essbase 中创建数据库的备份目录。

您需要创建导入配置来指定要导入哪些表和数据。可以保存配置，然后多次使用以导入相同的数据集。

关于临时表

将数据导入 Oracle Hyperion Profitability and Cost Management 时，用户会创建临时表，以提供预定义的结构来管理导入。

要将模型数据从关系数据库导入 Profitability and Cost Management 中，必须在不同于 Profitability and Cost Management 数据库表创建位置的数据库架构中创建一组临时表来设置信息格式供应用程序使用。

▲ 注意：

请勿在产品架构中创建导入临时表。不支持对产品架构进行修改，而且这样做会产生不可预料的结果。

这些表使用关系数据库（如 Oracle 或 SQL Server）创建，以便将数据组织为可以轻松与应用程序格式匹配的格式。临时数据库脚本在安装到安装文件夹中之后，Microsoft SQL Server 和 Oracle Database 可使用这些脚本。默认情况下，位置为 `%EPM_ORACLE_HOME%\products\Profitability\database\Common\`。使用适合应用程序类型的脚本在新数据库中创建临时表：

- 对于标准 Profitability 应用程序，使用 `create_staging.sql` 脚本。
- 对于明细 Profitability 应用程序，使用 `create_dp_staging.sql` 脚本。

您可以同时创建所有表，也可以仅创建要导入的表。您必须至少填充下列数据组之一：

- 阶段（仅限标准 Profitability）
- POV
- 动因
- 动因选定项
- 动因例外
- 分配（仅限标准 Profitability）
- 源分配规则关联
- 计算规则（仅限明细 Profitability）

临时表由 Profitability and Cost Management 管理员 (*admin*) 使用以下各节中指定的格式创建。

- [标准 Profitability 导入临时表](#)
- [导入明细 Profitability 临时表](#)

要导入数据，您必须有适当的用户角色和安全授权。请参阅《Oracle Enterprise Performance Management System 用户安全管理指南》。

创建导入配置

仅当多个元素可用时才能导入数据。

- 源数据库及其关联的临时表和数据
- 接收数据的目标应用程序
- 用于定义将导入哪些表和数据的导入配置。导入配置可以多次使用。

要简化导入，请考虑为不同的信息集创建单独的配置。通过使用更小的导入组，您可以减少导入次数，并避免静态信息的重复更新。例如，您可能会为导入配置对模型元素进行分组，如下所示：

- 偶尔更改：POV 和阶段
- 经常更改：动因、动因选定项、动因例外、分配、分配规则选定项和计算规则。

创建配置后，运行配置将数据导入到应用程序。

要创建导入配置：

1. 选择将用作导入源的数据库。

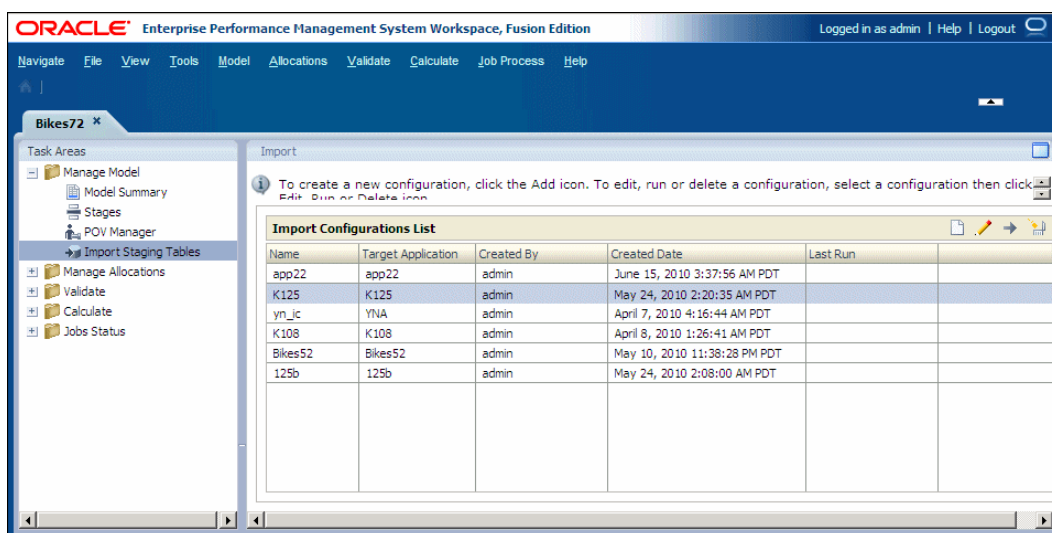
您可以使用已为导入进行格式化的现有数据库，或创建新的空白数据库。


数据库没有命名限制。


注意：

Oracle 建议在导入数据或对象之前，先在 Oracle Hyperion Enterprise Performance Management Workspace 和 Oracle Essbase 中创建数据库的备份目录。可与管理员联系以获得帮助。

2. 针对源数据库运行相应的脚本以自动创建临时表：
 - 对于标准 Profitability 应用程序，运行 `create_staging.sql` 脚本。系统将使用“[标准 Profitability 导入临时表](#)”中提供的架构生成这些表。
 - 对于明细 Profitability 应用程序，运行 `create_dp_staging.sql` 脚本。系统将使用“[导入明细 Profitability 临时表](#)”中提供的架构生成这些表。
3. 将模型数据加载到源临时表。
4. 查看临时表中的条目了解明显的问题，并从源数据库中删除空行。
5. 验证源数据库是可访问的。
6. 在 EPM Workspace 中，依次选择导航、应用程序和 **Profitability**，然后选择模型名称来访问 Oracle Hyperion Profitability and Cost Management 模型。
7. 从“任务区域”中，依次选择管理模型和导入临时表。




8. 在导入中，单击添加新的导入配置按钮  以创建新的导入配置。
将显示“导入数据”对话框的步骤 1。
9. 在数据源详细信息下，输入源临时表所在服务器的访问详细信息：

- 从数据库类型中，选择用于源数据库的数据库类型。
 - 在主机下，输入包含源临时表的源数据库的位置路径。
 - 在端口下，输入与源数据库相关的端口的名称。
 - 在数据库名称下，输入源数据库的名称。
10. 在凭据下，输入源数据库的授权用户的用户名和密码（如果需要）。
 11. 单击下一步。
将显示“导入数据”对话框的步骤 2。
 12. 在配置下，选择目标应用程序。
目标应用程序是数据要导入到的目标 Profitability and Cost Management 应用程序。
 13. 在临时表详细信息下，选择要导入的一个或多个临时表。
现有临时表的名称将列在“表名”下，并且每个临时表中的数据类型显示在“表类型”下。
 14. 单击下一步。
将显示“导入数据”对话框的步骤 3。
 15. 在配置摘要下，输入一个配置名称以存储此导入配置。
与此配置相关联的数据库名称显示在该配置名称下。该配置名称包含的字符不能超过 80 个。
 16. 单击完成。
导入配置将添加到“导入配置列表”中，并且可以多次重用以导入同一组数据。
 17. 可选：从“导入配置列表”中，选择导入配置，然后单击运行导入配置按钮  以运行导入配置。请参阅“运行导入配置”。

修改导入配置

更新导入配置时，可以更改目标应用程序、临时表组和导入配置名称。

要修改导入配置：

1. 从“任务区域”中，依次选择管理模型和导入临时表。
将显示“导入”页。
2. 在导入配置列表中，选择要修改的导入配置。
3. 单击编辑导入配置按钮 。
将显示“导入数据”对话框。
4. 查看选定导入配置的连接信息和凭据，然后单击下一步。
5. 可选：在目标应用程序下，为此导入配置选择一个目标应用程序。
6. 可选：在临时表详细信息中，更改要导入的临时表组。
7. 单击下一步。
8. 可选：在配置摘要下，输入其他配置名称以存储修改后的配置，并查看连接 URL。


9. 单击完成。

修改的配置会被存储，并可以随时运行。请参阅“[运行导入配置](#)”。

删除导入配置

可以删除导入配置。


要删除导入配置：

1. 从“任务区域”中，依次选择管理模型和导入临时表。
将显示“导入”页。
2. 选择要删除的配置，然后单击删除导入配置按钮 。
将显示一则消息，要求您确认删除。
3. 单击是。
将删除配置。

运行导入配置

创建导入配置后，您可以运行该配置以将数据导入到应用程序中。

要运行导入配置：

1. 确保您拥有临时表所在的关系数据库的连接信息。
2. 从“任务区域”中，依次选择管理模型和导入临时表。
将显示“导入”页。
3. 从导入配置列表中，选择一个导入配置。
4. 单击 **运行导入配置按钮** 。
将显示“导入数据”对话框。将列出所有数据连接的详细信息。
5. 单击下一步。
6. 在配置下，验证目标数据库和所选临时表是否正确。
7. 单击下一步。
此时将显示“配置摘要”。标识“配置名称”和连接 URI。
8. 选择下列操作之一：
 - 单击**立即运行**立即运行导入。
 - 单击**稍后运行**安排在某个更方便的日期和时间运行导入。
9. 单击完成。
 - 如果已选择**立即运行**，将运行导入操作，并使用所选数据填充目标应用程序。
 - 如果选择了**稍后运行**，将保存此作业。要运行保存的导入作业，选择“作业进程”，然后选择“管理任务流”。在此屏幕中，您可以运行该作业或者安排一个更方便的日期和时间来运行。有关详细说明，请参阅《*Oracle Hyperion Profitability and Cost Management 用户指南*》。
10. 完成导入后，请验证导入的数据。请参阅“[验证导入的数据](#)”。

验证导入的数据

将数据导入到目标数据库后，您必须验证数据已正确、完全地导入。

要验证导入的数据：

1. 在 Oracle Hyperion Enterprise Performance Management Workspace 中，依次选择导航、应用程序和 **Profitability**，然后选择应用程序名称来访问 Oracle Hyperion Profitability and Cost Management 模型。
2. 在目标应用程序中，打开该模型，查看预期导入的数据。
例如，如果已导入阶段，依次选择管理模型和阶段，并确保所有阶段的信息都存在且正确。
3. 查看每个导入的临时表中的“异常”列以确定是否存在错误或警告。
4. 在源临时表中更正错误，然后重新运行导入。
根据需要重复步骤 2 到步骤 3，直到导入期间不再生成错误。

C

使用 EPM System 生命周期管理迁移数据

关于生命周期管理

Oracle Hyperion Enterprise Performance Management System 生命周期管理为 Oracle Enterprise Performance Management System 产品在不同产品环境和操作系统间迁移应用程序、存储库或单个对象提供了统一的方式。

使用生命周期管理功能，您可以执行下列任务：

- 查看应用程序和文件夹
- 搜索对象
- 直接从一个应用程序迁移到另一个应用程序
- 迁入到文件系统和从文件系统迁出
- 保存和加载迁移定义文件
- 查看选定的对象
- 审核迁移
- 查看迁移状态
- 导入和导出单个对象以便快速更改文件系统

一般来说，Oracle Hyperion Shared Services Console 中的生命周期管理界面对支持生命周期管理的所有 EPM System 产品都是一致的。不过，EPM System 产品会在生命周期管理界面中显示不同的对象列表以及导出和导入选项。

除了在 Shared Services Console 中提供生命周期管理界面外，还有一个称为生命周期管理实用程序的命令行实用程序，它提供了另外一种将对象从源迁移到目标的方式。生命周期管理实用程序可以与第三方调度服务（如 Windows Task Scheduler 或 Oracle Enterprise Manager）结合使用。

最后，还有一个生命周期管理应用程序编程接口 (API)，让用户可以自定义和扩展生命周期管理功能。

有关标准和明细 Profitability 的生命周期管理和 Oracle Hyperion Profitability and Cost Management 对象的详细信息，请参阅《Oracle Enterprise Performance Management System 生命周期管理指南》。

管理分类帐的 ApplicationData 和维对象

对于管理分类帐应用程序，LCM 导出内容中包括一个元数据文件夹，该文件夹中包含以下对象：

- ApplicationData
- 维

这些对象分别位于以下文件夹结构中：

- resource\Metadata\ApplicationData\ - 此处只有一个文件
- resource\Metadata\Dimensions\ - 每个维有一个对应的文件

例如：

```
resource\Metadata\ApplicationData\application.xml
resource\Metadata\Dimensions\Account.txt
resource\Metadata\Dimensions\Product.txt
resource\Metadata\Dimensions\Year.txt
```

ApplicationData 对象类型供将来使用，当前不可用于导出或重新导入。要确保 LCM 导出正常运行，必须在导出或导入维时取消选择 ApplicationData。

要导入维，必须先在 Profitability 应用程序控制台中创建一个新的空应用程序。只能将维导入尚未启用的空应用程序。

 注：

对于这些对象，对象类型与对象名称相同。

修改生命周期管理的默认超时设置

如果您使用 Oracle Hyperion Enterprise Performance Management System 生命周期管理导入大型模型，处理导入的时间可能会超过在 Oracle WebLogic Server 的默认超时设置中指定的时间。

要解决此问题，您必须重置默认超时设置。

要修改 WebLogic Server 的默认超时：

1. 导航到

```
%Middleware_HOME%
\user_projects\epmsystem1\httpConfig\ohs\config\OHS\ohs_component\mod_
wl_ohs.conf
```

2. 在配置文件中，找到 **LocationMatch/profitability** 节：

```
<LocationMatch /profitability>
SetHandler weblogic-handler
PathTrim /
KeepAliveEnabled ON
KeepAliveSecs 20
WLIOTimeoutSecs 3000
Idempotent OFF
WeblogicCluster servername:6756
</LocationMatch>
```

```
SetHandler weblogic-handler
PathTrim
KeepAliveEnabled ON
```

```
KeepAliveSecs 20
WLIOTimeoutSecs 3000
WeblogicCluster servername:6756
```

3. 在 LocationMatch/profitability 节中添加以下行:

```
WLIOTimeoutSecs 3000
Idempotent OFF
```

4. 导航到

```
%Middleware_HOME%
\user_projects\epmsystem1\httpConfig\ohs\config\OHS\ohs_component\httpd.conf
```

5. 将超时设为 **3000**，如以下文本所示:

```
# Timeout: The number of seconds before receives and sends time out. Timeout
3000
```

 **注:**

上面所示的服务器超时是建议的限制值，可修改此设置以满足您应用程序服务器的文档中提供的特定超时设置。

D

备份 Profitability and Cost Management 组件

有几个 Oracle Hyperion Profitability and Cost Management 数据库需要定期备份，以确保数据的安全和完整性：

- 可操作的数据存储
- 导入临时表
- 块存储选项 (BSO) 数据库



注：

根据应用程序类型，您可能不使用其中某些数据库。

备份频率由数据库中的更改量和您的组织的常规标准确定。

表 1 列出了必须备份的数据库、相应的数据库类型以及建议的备份方案。



注意：

生命周期管理 (LCM) 当前不能用于备份导入临时区域或可操作的数据存储。

表 D-1 Profitability and Cost Management 数据库备份要求

| 数据库 | 数据库类型 | 备份方案 |
|--|----------------|--|
| 导入临时区域 | 关系数据库 (RDB) | 可以使用适用于任何 RDB 的标准备份技术，其中包括下列方案： <ul style="list-style-type: none">• 数据库脚本• 调度器脚本• 备份工具，例如 TOAD• 来自 SQL Server 或 Oracle 的备份过程。 |
| 可操作的数据存储 | 关系数据库 (RDB) | 可以使用适用于任何 RDB 的标准备份技术，其中包括下列方案： <ul style="list-style-type: none">• 数据库脚本• 调度器脚本• 备份工具，例如 TOAD• 来自 SQL Server 或 Oracle 的备份过程。 |
| 仅适用于标准 Profitability 的块存储选项 (BSO) 和聚合存储选项 (ASO) 数据库和分配规则 | Oracle Essbase | 应使用标准的 Essbase 备份技术。有关详细的备份和恢复过程，请参阅《Oracle Essbase Database Administrator's Guide》。 |

表 D-1 (续) Profitability and Cost Management 数据库备份要求

| 数据库 | 数据库类型 | 备份方案 |
|--------------------------|-------------|--|
| 明细 Profitability 的模型数据架构 | 关系数据库 (RDB) | 可以使用适用于任何 RDB 的标准备份技术, 其中包括下列方案: <ul style="list-style-type: none">• 数据库脚本• 调度器脚本• 备份工具, 例如 TOAD• 来自 SQL Server 或 Oracle 的备份过程。 |

有关详细的备份和恢复过程, 请参阅《Oracle Enterprise Performance Management System Backup and Recovery Guide》。

E

Essbase 命名约定

用于生成的计算脚本的命名约定

Oracle Hyperion Profitability and Cost Management 生成 Oracle Essbase 脚本时，将使用特定约定自动命名这些脚本。

每个阶段的动因计算脚本和分配计算脚本将组合为一个脚本。创建的脚本名称采用如下格式：

```
String scriptName = scriptSuffix + POV-identifier + Stage Order Number + "_" + index;
```

- 脚本后缀基于脚本类型。下表是一个后缀列表。

表 E-1 计算脚本后缀

| 脚本类型 | 成本层 | 收入层 |
|-------|-----|-----|
| 阶段间分配 | "a" | "r" |
| 阶段内分配 | "i" | "t" |

- POV 标识符基于 POV ID，最多由 3 位数组成。会针对每个 POV 生成一个脚本并标识出来。
- 阶段序号是源阶段的序号（例如，1、2、3 等）。
- 如果由于脚本拆分生成了多个脚本，_index 将显示具有相同的类型、POV、阶段和层的脚本的数字序号，从 001 开始，然后是 002，依此类推。

示例脚本名称

- a3682001.csc 表示 POV 标识符 368、源阶段 2 和成本层的计算脚本。
- t4533002.csc 表示 POV 标识符 453、源阶段 3 和收入层的阶段内计算脚本。

应用程序和数据库的 Essbase 命名限制

当为应用程序和数据库时创建名称时，输入名称并注意大小写。

应用程序或数据库名称将完全按照输入内容创建。如果以全部大写字母的方式输入名称（例如，NEWAPP），则 Oracle Essbase 不会自动将其转换为大写字母后跟小写字母（例如，Newapp）。

注：

此列表给出了一部分限制。有关所有限制的完整列表，Oracle 建议您参考《Oracle Essbase Database Administrator's Guide》：

在命名应用程序和数据库时，应遵循以下命名限制：

- 使用 8 个以内的字符。
- 不要在名称中使用空格。
- 不要在名称中使用以下特殊字符：
 - * 星号
 - + 加号
 - \ 反斜杠
 - ? 问号
 - [] 方括号
 - " 双引号
 - : 冒号
 - ; 分号
 - , 逗号
 - ' 单引号
 - = 等号
 - / 斜杠
 - > 大于号
 - 制表符
 - < 小于号
 - | 竖线
 - . 句点
- （仅限聚合存储数据库）下列字词不可用作应用程序或数据库名称：
 - DEFAULT
 - LOG
 - METADATA
 - TEMP

Essbase 对维、成员和别名的命名限制

定义维大纲时，不能在维名、成员名和别名中使用限制字符。

本部分中提供了最常见的限制字符的列表；不过，Oracle 强烈建议您认真查看《*Oracle Essbase Database Administrator's Guide*》中的 Essbase 命名约定，以了解完整的列表。

注：

此列表给出了一部分限制。有关所有限制的完整列表，Oracle 建议您参考《*Oracle Essbase Database Administrator's Guide*》：

在命名维、成员和别名时，应遵循下列命名约定：

- 使用 80 个以内的字符。
- 只有在启用了区分大写时，才区分大写和小写字母。要启用区分大小写，请参阅《Oracle Essbase Database Administrator's Guide》中的 "Setting Outline Properties"。
- 请不要在维或成员名称、别名和说明中使用 HTML 标记。
- 请勿在名称中使用引号、句点、方括号、反斜线或制表符。

▲ 注意：

块存储大纲中的括号在转换为聚合存储大纲时会导致错误，因此允许但不建议使用括号。

- 不允许在同一个维中出现重复的成员名称或别名。
- 维或成员名称不要以下列字符开头：
 - at 符号 (@)
 - 反斜杠 (\)
 - 中括号 ([])
 - 逗号 (,)
 - 短划线
 - 连字符
 - 减号 (-)
 - 等号 (=)
 - 小于号 (<)
 - 圆括号 ()
 - 句点 (。)
 - 加号 (+)
 - 单引号 (')
 - 引号 (")
 - 下划线 (_)
 - 竖线 (|)
- 不要将空格放在名称的开头或结尾，因为 Oracle Essbase 会忽略它们。
- 请勿在成员名称中使用正斜线。
- 对于自定义日历的期间，请勿在前缀中使用空格。
- 不要将下列各项作为维或成员名使用：
 - 计算脚本命令、运算符或关键字。有关命令的完整列表，请参阅《Oracle Essbase Database Administrator's Guide》。
 - 报表编写器命令
 - 函数名称和函数参数

- 如果启用动态时间系列成员，请勿使用：历史、年、季节、期间、季度、月、周或日。
- 数据库中其他维和成员的名称（除非成员已共享）、层代名称、级别名和别名。
- 不要使用下列字词：
 - ALL
 - AND
 - ASSIGN
 - AVERAGE
 - CALC
 - CALCMBR
 - COPYFORWARD
 - CROSSDIM
 - CURMBRNAME
 - DIM
 - DIMNAME
 - DIV
 - DYNAMIC
 - EMPTYPARM
 - EQ
 - EQOP
 - EXCEPT
 - EXP
 - EXPERROR
 - FLOAT
 - FUNCTION
 - GE
 - GEN
 - GENRANGE
 - GROUP
 - GT
 - ID
 - IDERROR
 - INTEGER
 - LE
 - LEVELRANGE
 - LOOPBLOCK

- LOOPPARMS
- LT
- MBR
- MBRNAME
- MBRONLY
- MINUS
- MISSING
- MUL
- MULOP
- NE
- NON
- NONINPUT
- NOT
- OR
- PAREN
- PARENPARM
- PERCENT
- PLUS
- RELOP
- SET
- SKIPBOTH
- SKIPMISSING
- SKIPNONE
- SKIPZERO
- TO
- TOLOCALRATE
- TRAILMISSING
- TRAILSUM
- UMINUS
- UPPER
- VARORXMBR
- XMBRONLY
- \$\$UNIVERSE\$\$
- #MISSING
- #MI

Essbase 对属性计算维的命名约定

Oracle Essbase 在属性计算维中创建的成员（Sum、Count、Min、Max 和 Avg）不考虑为保留字。

这是因为您可以在属性计算维中更改这些名称，然后在属性或标准维中使用标准名称。

如果大纲已标记为唯一的成员大纲，则避免使用 Sum、Count、Min、Max 和 Avg 作为成员名称。例如，如果您在标准维中使用 Max，然后创建一个属性维，并且 Essbase 在属性计算维中创建 Max 成员，则 Essbase 会检测到重复名称并返回以下错误消息：

```
"Analytic Server Error(1060115): Attribute Calculations dimension/member name already used."
```

如果大纲已标记为重复成员大纲，并且在您使用 Sum、Count、Min、Max 和 Avg 作为基成员之前已存在属性维（以及属性计算维），则 Essbase 允许存在重复的名称。但是，如果在创建属性维之前已使用 Sum、Count、Min、Max 和 Avg 作为基成员，则不允许出现重复的名称。