

Oracle® Hyperion Profitability and Cost Management

관리자 가이드



11.2.16
F92197-01
2024년 1월

ORACLE®

F92197-01

Copyright © 2008, 2024년, Oracle and/or its affiliates.

주요 작성자: EPM Information Development Team

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.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

목차

설명서 접근성

설명서 피드백

I부 Profitability and Cost Management 시작

1 개요

Profitability and Cost Management 정보	1-1
Profitability and Cost Management의 액세스 가능	1-2
Profitability 유형	1-2
관리 원장 Profitability	1-2
상세 Profitability	1-3
추가 제품 구성요소	1-3
관리 태스크	1-3
Profitability and Cost Management 실행	1-4

2 보안 관리 및 사용자 인증

사용자 설정 및 프로비저닝 정보	2-1
보안 역할 지정	2-3
관리 원장 Profitability에 대한 보안 역할	2-4
상세 Profitability에 대한 보안 역할	2-5
Profitability and Cost Management의 변경사항 감사	2-7
출력 로그 파일	2-10
ODL(Oracle Diagnostic Logging) 파일	2-11

II부 Profitability and Cost Management 애플리케이션 생성

3 Profitability 애플리케이션 콘솔로 애플리케이션 생성

Profitability 애플리케이션 콘솔 표시	3-1
Profitability 애플리케이션 콘솔 사용 가이드라인	3-2
Essbase 마스터 큐브의 차원으로 애플리케이션 생성	3-3
관리 원장 애플리케이션 생성	3-5
플랫 파일의 차원을 사용하여 관리 원장 애플리케이션 생성	3-5
템플릿 파일을 импорт하여 관리 원장 애플리케이션 생성	3-6
Profitability 애플리케이션 콘솔의 애플리케이션 작업	3-6
Profitability 애플리케이션 콘솔에서의 애플리케이션 작업 정보	3-7
애플리케이션 설명 및 Shared Services 프로젝트 편집	3-7
기타 애플리케이션 작업 수행	3-7
애플리케이션 차원 업데이트	3-9
Profitability 애플리케이션 콘솔 작업 라이브러리의 태스크 보기	3-10
관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫 파일 준비	3-12
관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫 파일 준비 정보	3-13
템플릿 익스포트	3-13
각 관리 원장 차원에 대한 플랫 파일 준비	3-13
플랫 파일 속성 정보	3-15
플랫 파일의 주석 정보	3-18
플랫 파일 샘플	3-18
Profitability and Cost Management용 Essbase 마스터 큐브 생성	3-18
차원 유형	3-20
사용자 정의 속성	3-20
ASO 차원 및 멤버 등록정보	3-20
BSO 및 멤버 차원 등록정보 지정	3-22
멤버 등록정보에 지정된 BSO(블록 저장영역 옵션) 데이터 저장영역 값 보기	3-26
사용자정의 측정항목 차원	3-26

III부 관리 원장 Profitability 작업

4 관리 원장 Profitability 애플리케이션 아키텍처 이해

5 관리 원장 차원 이해

관리 원장 차원 정보	5-1
관리 원장 시스템 차원	5-3
관리 원장 규칙 차원	5-3
관리 원장 잔액 차원	5-4

관리 원장 비즈니스 차원	5-5
POV 차원	5-6
속성 차원	5-7
별칭 차원	5-7
관리 원장 차원 정렬 순서	5-8
차원 정렬 순서 권장 사항	5-8
차원 정렬 순서 설정	5-8

6 다른 애플리케이션 관리 태스크 수행

관리 원장 Profitability 애플리케이션 관리	6-1
관리 원장 애플리케이션 및 차원 작업	6-1
Profitability and Cost Management 차원 및 멤버 등록정보	6-1
계층 유형 등록정보 설정	6-2
마법사를 사용하여 관리 원장 애플리케이션 검증 및 배포	6-2

IV부 상세 Profitability 작업

7 상세 Profitability 제품 아키텍처 이해

8 상세 Profitability 데이터베이스 작업

9 제품 스키마

10 상세 Profitability에 대한 모델 데이터 스키마

상세 Profitability에 대한 모델 데이터 스키마 정보	10-1
모델 데이터 스키마 테이블에서 제품 스키마에 관한 부여	10-1
모델 데이터 스키마 테이블 구조 요구사항	10-2
데이터베이스 뷰 등록	10-2

11 상세 Profitability and Cost Management 차원 및 메타데이터 작업

상세 Profitability 차원 및 메타데이터 정보	11-1
상세 Profitability 차원 유형	11-2
별칭 차원	11-3
비즈니스 차원	11-3
POV 차원	11-4

속성 차원	11-5
차원 정렬 순서 및 밀도 설정	11-5
차원 정렬 순서 권장 사항	11-6
차원 정렬 순서 설정	11-6

12 상세 Profitability 모델 계산

사전 및 사후 계산 사용자정의 스크립트	12-1
HPM_SQL_SCRIPT	12-2
사용자정의 스크립트 생성	12-3
ODBC 데이터 소스를 생성하여 데이터 전송 활성화	12-4
고급 계산 옵션	12-4
드라이버 작업 유형	12-4
기타 프로세스 유형	12-5

13 상세 Profitability 스테이지 테이블 임포트

상세 Profitability 스테이징 테이블 임포트 정보	13-1
상세 Profitability에 대한 임포트 데이터베이스 테이블 생성	13-2
HPM_STGD_POV	13-2
HPM_STGD_DRIVER	13-3
HPM_STGD_DRIVER_SEL	13-5
HPM_STGD_DRIVER_EXCEP	13-6
HPM_STGD_ASGN_RULE_SEL	13-7
HPM_STGD_CALCRULE_SNGLSRC	13-9
HPM_STGD_CALCRULE_CALCMSRS	13-10
HPM_STGD_CALCRULE_MULTISRC	13-12

14 상세 Profitability에 대한 모델 정의 데이터 익스포트

상세 Profitability에 대한 모델 정의 데이터 익스포트 정보	14-1
HPM_EXPD_STAGE	14-2
HPM_EXPD_POV	14-3
HPM_EXPD_DRIVER	14-4
HPM_EXPD_DRIVER_SEL	14-5
HPM_EXPD_DRIVER_EXCEP	14-6
HPM_EXPD_ASGN_RUL_SEL	14-6
HPM_EXPD_CALCRULE_SNGLSRC	14-7
HPM_EXPD_CALCRULE_CALCMSRS	14-8
HPM_EXPD_CALCRULE_MULTISRC	14-10

A 애플리케이션 디자인 모범 사례

애플리케이션 확장성 관리	A-1
차원 수	A-1
차원 멤버 수	A-1
속성 및 UDA	A-2
데이터 스케일 관리	A-2
시작 데이터 요약	A-2
할당 중간점 요약	A-2
차원 디자인 관리	A-3
차원 유형	A-3
저장	A-3
동적	A-3
계층 사용 설정	A-4
계층 최상위 레벨	A-4
롤업 및 보고	A-4
멤버 없음	A-4
모범 사례 예	A-4
중복 멤버	A-5
애플리케이션 로직 디자인 관리	A-5
규칙 수를 관리하는 단순한 전략	A-6
단일 규칙을 사용하여 일반 논리를 적용할 수 있는 위치 파악	A-6
드라이버 데이터 부재를 할당 필터로 사용	A-6
할당 규칙 유형 및 옵션	A-6
단순 차원 확장	A-6
복합 차원 확장	A-7
단순 재분류	A-7
복합 재분류	A-7
소스와 동일, 다른 차원	A-7
각 규칙 유형에 허용되는 할당 규칙 및 멤버 선택 옵션	A-7
규칙 범위	A-8
범위 크기 및 빈도	A-8
소스 세분화 사용	A-9
필터 또는 대체 계층 사용	A-9
음수 드라이버 데이터 값 사용 방지하기	A-9
동적 멤버를 드라이버로 사용 방지	A-9
규칙 세트에서 병렬 실행 사용	A-9
사용자정의 계산 규칙	A-10

B Profitability and Cost Management로 데이터 импорт

데이터 импорт 정보	B-1
스테이지 테이블 정보	B-1
임포트 구성 생성	B-2
임포트 구성 수정	B-4
임포트 구성 삭제	B-5
임포트 구성 실행	B-5
임포트된 데이터 확인	B-6

C EPM System Lifecycle Management를 사용한 데이터 마이그레이션

Lifecycle Management 정보	C-1
관리 원장의 ApplicationData 및 Dimensions 아티팩트	C-1
Lifecycle Management에 대한 기본 시간 초과 설정 수정	C-2

D Profitability and Cost Management 구성요소 백업

E Essbase 이름 지정 규칙

생성된 계산 스크립트 이름 지정 규칙	E-1
애플리케이션 및 데이터베이스에 대한 Essbase 이름 지정 제한 사항	E-1
차원, 멤버 및 별칭에 대한 Essbase 이름 지정 제한	E-3
속성 계산 차원에 대한 Essbase 이름 지정 규칙	E-6

설명서 접근성

오라클의 접근성 개선 노력에 대한 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>에서 Oracle Accessibility Program 웹 사이트를 방문하십시오.

오라클 고객지원센터 액세스

지원 서비스를 구매한 오라클 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

설명서 피드백

이 문서에 대한 피드백을 제공하려면 모든 Oracle 도움말 센터 항목의 페이지 맨아래에 있는 [피드백] 버튼을 누릅니다. epmdoc_ww@oracle.com으로 전자메일을 보낼 수도 있습니다.

I부

Profitability and Cost Management 시작

참조:

- [개요](#)
- [보안 관리 및 사용자 인증](#)

1 개요

참조:

- [Profitability and Cost Management 정보](#)
Oracle Hyperion Profitability and Cost Management는 Oracle Hyperion Enterprise Performance Management Workspace에서 액세스하는 분석 애플리케이션입니다.
- [Profitability and Cost Management의 액세스 가능](#)
Oracle Hyperion Profitability and Cost Management는 기본 기능에 대한 바로가기 키를 제공합니다.
- [Profitability 유형](#)
Oracle Hyperion Profitability and Cost Management에서는 수익성 평가에 사용되는 세 가지 애플리케이션을 제공합니다.
- [추가 제품 구성요소](#)
다음 제품을 사용하면 Oracle Hyperion Profitability and Cost Management의 성능과 유연성이 향상됩니다.
- [관리 태스크](#)
Oracle Hyperion Profitability and Cost Management 관리자 또는 *admin* 역할을 통해 여러 태스크를 수행할 수 있습니다.
- [Profitability and Cost Management 실행](#)
Oracle Hyperion Profitability and Cost Management는 Oracle Hyperion Enterprise Performance Management Workspace를 통해서만 액세스할 수 있습니다.

Profitability and Cost Management 정보

Oracle Hyperion Profitability and Cost Management는 Oracle Hyperion Enterprise Performance Management Workspace에서 액세스하는 분석 애플리케이션입니다.

비용 및 수익을 정확하게 측정, 할당 및 관리하고, 비즈니스 세그먼트의 수익성을 계산하며, 비용 분해, 소비 기반 원가 계산 및 시나리오 재현을 통해 수익성을 측정하는 데 사용됩니다.

Profitability and Cost Management는 EPM Workspace의 필수적인 부분으로, 다양한 구성요소를 사용하여 다음과 같은 애플리케이션을 빌드하고 관리합니다.

- EPM Workspace를 사용하여 Profitability and Cost Management에 액세스합니다. 기타 구성요소를 관리하여 애플리케이션을 빌드하고 보안을 제어하며 보고서를 생성합니다. Profitability 애플리케이션 유형의 경우 [Profitability 유형](#)을 참조하십시오.
- Oracle Hyperion Shared Services를 사용하여 사용자가 액세스할 수 있는 모델을 확인하는 보안 역할 정의를 비롯한 사용자 계정을 생성하고 관리합니다.
- Profitability 애플리케이션을 사용하여 Profitability and Cost Management에 사용할 차원과 멤버를 빌드하고 유지관리합니다. 차원 라이브러리를 통해 Oracle Hyperion Planning 등의 다른 애플리케이션에 이미 존재하는 공통 차원 및 멤버를 사용하는 애플리케이션을 빌드할 수 있습니다.

이 가이드에서는 모델의 애플리케이션 및 차원을 관리하는 데 Profitability 애플리케이션 콘솔을 사용한다고 가정합니다. [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#)을 참조하십시오.

- Oracle Essbase 또는 관계형 데이터베이스를 사용하여 온라인으로 애플리케이션을 생성하고 계산 스크립트를 저장 및 실행합니다.
- Oracle Hyperion Enterprise Performance Management System Lifecycle Management를 사용하여 제품 전체 및 운영 체제 전체에서 애플리케이션, 다차원 데이터베이스, 저장소 또는 개별 아티팩트를 마이그레이션합니다.
- Oracle Hyperion Reporting and Analysis, Oracle Hyperion Financial Reporting 또는 Microsoft Excel 등의 타사 제품을 사용하여 계산된 결과의 보고서를 생성합니다.

Profitability and Cost Management의 액세스 가능

Oracle Hyperion Profitability and Cost Management는 기본 기능에 대한 바로가기 키를 제공합니다.

접근성 기능은 *Oracle Hyperion Profitability and Cost Management Accessibility Guide*에 설명되어 있습니다. 이것은 Profitability and Cost Management의 고유한 기능입니다.

Profitability 유형

Oracle Hyperion Profitability and Cost Management에서는 수익성 평가에 사용되는 세 가지 애플리케이션을 제공합니다.

- [관리 원장 Profitability](#)
- [상세 Profitability](#)

관리 원장 Profitability

관리 원장 애플리케이션은 관리 보고 계산 및 보고 방법 분야에 대한 경험은 많으나 Oracle Essbase와 스크립팅 구문 또는 프로그래밍 언어에 대한 경험은 많지 않은 분석가가 사용하도록 디자인되었습니다.

관리 원장 애플리케이션에 대한 데이터는 Essbase 다차원 데이터베이스와 관계형 데이터베이스에 모두 포함됩니다. Profitability 애플리케이션 콘솔에서 애플리케이션을 생성하고 차원과 차원 멤버를 사용하여 조직 내의 계정, 활동 및 작업 계층을 정의합니다.

애플리케이션이 배포되면 모델을 빌드하여 특정 비용 및 수익 할당에 대한 자금 플로우를 표시합니다. 할당의 소스 및 대상 범위 둘 다 Profitability and Cost Management 사용자 인터페이스를 사용하여 계산 규칙으로 정의됩니다. 상세 Profitability 애플리케이션 유형의 경우 POV(Point Of View)는 모델의 특정 인스턴스를 나타내며, 모델의 여러 버전을 보거나 계산하는 데 사용될 수 있습니다. 예를 들어 여러 월 또는 분기의 값을 보거나, 예산 대 실제 수치를 비교하거나, 다양한 변화가 순익에 미치는 영향을 측정하기 위한 시나리오를 생성하는 데 사용될 수 있습니다.

관리 원장 모델에는 단계 또는 계층의 개념이 없습니다. 모든 구조는 POV의 규칙 세트 및 규칙 구성을 통해 제어됩니다. POV마다 동시 또는 유사한 시간에 데이터베이스의 동일 영역이나 유사한 영역에 대해 실행되는 계산 규칙이 그룹으로 구성됩니다. 이러한 그룹을 규칙 세트라고 합니다. 그룹은 계산 규칙이 실행되는 순서를 결정합니다. 계산 규칙은 POV 또는 규칙 세트 레벨에서 기본 멤버 선택을 상속할 수 있으므로 사용자가 데이터베이스 영역을 한 번 정의하여 여러 번 사용할 수 있으며 매번 영역을 지정할 필요가 없습니다. 이러한 기본값을 "컨텍스트"라고 합니다.

모델을 생성한 후 검증하여 모든 할당이 고려되었으며 계산이 밸런싱되었는지 확인합니다. 검증 후 데이터베이스를 배포하고, 모델을 계산하고, 결과를 분석합니다. 자세한 내용은

Oracle Hyperion Profitability and Cost Management 사용자 가이드 및 이 가이드의 4부 "관리 원장 애플리케이션 작업"을 참조하십시오.

상세 Profitability

상세 Profitability에서는 수익성 분석을 위해 단일 소스에서 대상까지의 풀 또는 환율을 한 단계로 할당합니다. 상세 Profitability는 모델 아티팩트 저장, 계산 및 보고 뷰에 대해 관계형 데이터베이스를 사용합니다.

상세 Oracle Hyperion Profitability and Cost Management 모델은 사용자 정의 스키마를 통해 기존 데이터 및 연계된 록업 테이블로 관계형 테이블을 구성하여 해당 데이터를 확장합니다. 상세 Profitability 모델에 대한 데이터는 관계형 데이터베이스에만 포함됩니다.

Profitability 애플리케이션 콘솔에서 모델을 생성하고 조직 내의 비즈니스 차원, 별칭, 측정항목을 정의합니다. Profitability and Cost Management 내에서 데이터가 애플리케이션에 매핑되어 상세 Profitability 모델을 빌드할 수 있게 합니다. 애플리케이션에서 매우 큰 볼륨을 처리할 수 있습니다.

애플리케이션은 계층 구조를 사용하지 않고 단일 소스 및 대상 조합 간의 플로우에서 모든 할당을 처리합니다. 할당은 제한된 MeasuresDetailed 차원을 통해 처리됩니다. 멤버 상세 차원에는 모든 할당을 처리할 제한된 멤버 세트가 포함됩니다.

상세 Profitability 작업에 대한 자세한 내용은 [상세 Profitability 제품 아키텍처 이해](#)를 참조하십시오.

추가 제품 구성요소

다음 제품을 사용하면 Oracle Hyperion Profitability and Cost Management의 성능과 유연성이 향상됩니다.

표 1-1 Profitability and Cost Management 제품 구성요소

제품	설명
Profitability and Cost Management Profitability 애플리케이션 콘솔	차원 및 애플리케이션 관리.
Oracle Hyperion Enterprise Performance Management Workspace	Profitability and Cost Management 및 기타 제품으로 이동하여 애플리케이션 관리
Oracle Hyperion Shared Services	Profitability and Cost Management의 외부 시스템에서 사용자 프로비전닝 Profitability and Cost Management 애플리케이션 마이그레이션
Oracle Smart View for Office	Microsoft Excel 스프레드시트에 Profitability and Cost Management 데이터를 입력 및 보고
Oracle Hyperion Financial Reporting	웹 또는 인쇄 배포용 보고서 및 차트 생성

관리 태스크

Oracle Hyperion Profitability and Cost Management 관리자 또는 *admin* 역할을 통해 여러 태스크를 수행할 수 있습니다.

- Oracle Hyperion Shared Services를 사용하여 사용자 계정을 생성하고 관리합니다.
- Shared Services 인증을 위한 사용자를 프로비전닝합니다.

- Profitability 애플리케이션 콘솔에서 공유 라이브러리와 Profitability and Cost Management 차원 및 멤버를 관리합니다.
- 다차원 Oracle Essbase 데이터베이스를 생성합니다.
- 모델 단계, 드라이버 및 POV를 생성, 업데이트 및 삭제합니다.
- 드라이버 선택, 지정, 지정 규칙 및 지정 규칙의 선택 사항을 생성, 업데이트 및 삭제합니다.
- 계산 규칙을 생성, 업데이트 및 삭제합니다.
- 모델을 계산 및 다시 계산합니다.
- 모델 데이터를 보고 수정합니다.
- 할당 추적을 봅니다.
- Profitability and Cost Management 모델 구성요소를 백업 및 복원합니다.
- Lifecycle Management 유틸리티를 사용하여 하나의 환경에서 다른 환경으로 애플리케이션을 전송합니다. 개발 또는 테스트 등의 환경에서 작업과 같은 다른 환경으로 승격합니다.
- 비즈니스 객체의 변경사항을 모니터링합니다.

또한 Profitability and Cost Management 관리자는 Oracle Hyperion Enterprise Performance Management Workspace를 통해 모델 메타데이터와 데이터를 관리합니다.

Profitability and Cost Management 실행

Oracle Hyperion Profitability and Cost Management는 Oracle Hyperion Enterprise Performance Management Workspace를 통해서만 액세스할 수 있습니다.

Profitability and Cost Management에 액세스하려면 다음을 수행합니다.

1. 다음 애플리케이션이 구성되어 실행 중인지 확인합니다.
 - EPM Workspace
 - Oracle Hyperion Shared Services
 - Profitability and Cost Management

지침은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드* 및 *Oracle Enterprise Performance Management System 설치 시작 페이지*를 참조하십시오.

2. 웹 브라우저에서 EPM Workspace 웹 페이지에 액세스합니다.

기본적으로 URL은 `http://server name:19000/workspace/`입니다. 여기서 서버 이름은 Oracle HTTP Server(OHS) 서버 이름입니다.

3. EPM Workspace 사용자 이름과 비밀번호를 입력합니다.



주:

사용자 이름과 비밀번호 모두 대소문자를 구분합니다.

4. 로그인을 누릅니다.

기본 EPM Workspace 페이지가 표시됩니다.

5. 탐색, 애플리케이션, **Profitability** 순으로 선택하고 보려는 애플리케이션을 선택합니다.

2

보안 관리 및 사용자 인증

참조:

- [사용자 설정 및 프로비저닝 정보](#)
관리자는 Oracle Hyperion Profitability and Cost Management 작업을 수행하기 전에 사용자 및 그룹을 설정하고 각각에 적절한 보안 역할을 지정해야 합니다.
- [보안 역할 지정](#)
Oracle Hyperion Profitability and Cost Management에서는 각 사용자 ID에 보안 역할이 지정됩니다.
- [Profitability and Cost Management의 변경사항 감사](#)
Oracle Hyperion Shared Services Console의 감사 기능을 사용하여 애플리케이션의 작업 및 변경사항을 모니터링한 다음 결과를 자세히 설명하는 감사 보고서를 생성할 수 있습니다.
- [출력 로그 파일](#)
관리자는 Oracle Hyperion Enterprise Performance Management Workspace에서 로그 파일을 생성할 수 있습니다.
- [ODL\(Oracle Diagnostic Logging\) 파일](#)
각 Oracle Hyperion Enterprise Performance Management Workspace 제품의 모든 구성 로그 파일에 대한 웹 애플리케이션의 ODL(Oracle Diagnostic Logging) 위치는 중앙 집중식입니다.

사용자 설정 및 프로비저닝 정보

관리자는 Oracle Hyperion Profitability and Cost Management 작업을 수행하기 전에 사용자 및 그룹을 설정하고 각각에 적절한 보안 역할을 지정해야 합니다.

각 보안 역할에 제공된 권한으로 사용자나 그룹이 액세스할 수 있는 기능 및 데이터가 결정됩니다. 구성 중 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*에 설명된 대로 Oracle Hyperion Shared Services를 인증 모드로 선택합니다.

보안 및 사용자 프로비저닝 관리에 다음 두 개의 가이드를 참조하십시오.

- 다음과 같은 보안에 대한 기술 정보는 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.
 - SSL(단방향, 양방향, SSL 오프로딩, SSL 종료)
 - 단일 사인온
 - 기본 EPM System SSO
 - 보안 에이전트
 - 사용자정의 로그인
 - 사용자정의 인증 모듈
 - EPM System 보안 가이드라인
- 다음과 같은 사용자 프로비저닝 설정 및 관리 방법에 대한 정보는 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

- Oracle Hyperion Shared Services Console
- 사용자 디렉토리
- 애플리케이션 및 애플리케이션 그룹
- 위임된 사용자 관리
- Native Directory 관리
- 프로비저닝 관리
- EPM System 프로비저닝

Oracle Hyperion Enterprise Performance Management Workspace를 통해 Profitability and Cost Management에 사용자 및 그룹을 생성, 유지관리 및 프로비저닝합니다. 프로비저닝 프로세스에는 Shared Services 및 Profitability and Cost Management를 모두 구성 및 실행해야 합니다. 외부 인증을 통해 애플리케이션 간의 통신이 원활하게 수행되어 쉽고 정확하게 사용자를 프로비저닝하도록 합니다.

다음 단계는 사용자 및 그룹을 설정하고 프로비저닝하는 프로세스 개요를 제공합니다.

1. EPM Workspace에서 **탐색, 관리, Shared Services Console** 순으로 선택하여 Shared Services 화면에 액세스합니다.

 **주:**

최초 로그인 시 해당 제품에 대해 자동으로 관리자(admin)가 생성됩니다.

2. *관리* 역할에 지정된 사용자를 프로비저닝 관리자 역할에 설정합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.
3. 사용자를 생성합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.
4. 사용자에게 적합한 보안 역할과 필요한 프로젝트 및 애플리케이션에 대한 액세스 권한을 프로비저닝합니다.

 **주의:**

사용자가 Profitability and Cost Management 외부에서 태스크를 수행하기 위해 EPM Workspace에 액세스해야 하는 경우 선택한 사용자에게 EPM Workspace 역할(예: Oracle Hyperion Enterprise Performance Management System Lifecycle Management)도 프로비저닝해야 합니다.

5. 필요에 따라 그룹을 생성합니다. 그룹 내에 그룹을 프로비저닝할 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.
6. EPM Workspace에 로그인하고 Profitability and Cost Management를 엽니다.
7. 사용자가 Profitability and Cost Management에 로그인하고 프로비저닝된 프로젝트 및 애플리케이션을 볼 수 있는지 확인합니다.

사용자 및 그룹을 설정하고 프로비저닝하는 방법에 대한 자세한 지침은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

보안 역할 지정

Oracle Hyperion Profitability and Cost Management에서는 각 사용자 ID에 보안 역할이 지정됩니다.

- 관리자(Oracle Hyperion Shared Services에 로그인하는 경우 *admin*이 기본 보안 역할임)
- 고급 사용자
- 대화식 사용자
- 보기 사용자

지정된 보안 역할에 따라 해당 사용자가 사용할 수 있는 액세스 또는 권한 레벨이 결정됩니다. 사용자에게 여러 역할을 부여할 수 있습니다. 특정 작업에 대한 권한은 작업 시작 시 확인됩니다.

주:

한 명 이상의 사용자에게 수동으로 Oracle Hyperion Shared Services Console의 프로비저닝 관리자 역할을 지정해야 합니다. 사용자는 이 프로비저닝 관리자 역할을 통해 애플리케이션의 다른 사용자에게 보안 역할을 지정할 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

그룹에 사용자를 지정하려면 해당 사용자가 있어야 하고 보안 역할이 지정되어 있어야 합니다. 사용자 그룹에 액세스 레벨이 지정되면 해당 그룹의 모든 멤버에게 유사한 보안 액세스 권한이 부여됩니다. 특정 사용자에 대한 액세스 요구사항에 따라 지정된 보안에 보다 넓거나 좁은 액세스 권한을 첨부하도록 수정될 수 있습니다. 예를 들어 고급 사용자 보안 권한을 가진 그룹에 지정된 사용자 보기의 경우 보안 레벨이 보다 높을 수 있습니다.

주의:

Oracle Hyperion Enterprise Performance Management Workspace의 애플리케이션 생성자와 같은 다른 제품에 대한 액세스 권한이 필요하거나 Oracle Hyperion Enterprise Performance Management System Lifecycle Management를 통해 스테이징 테이블을 импорт하거나 익스포트해야 하는 경우 해당하는 추가 보안 역할이 별도로 지정되어야 합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

다음 섹션에서 자세히 설명하는 보안 역할은 Profitability and Cost Management와 관련이 있습니다. 모든 보안 역할에 대한 전체 설명은 *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*에서 자세한 지침을 참조하십시오.

관리 원장 Profitability에 대한 보안 역할

표 2-1 관리 원장 Profitability and Cost Management 보안 역할

보안 역할	설명
관리자(admin), 역할 유형 = 고급	<ul style="list-style-type: none"> Oracle Hyperion Shared Services를 사용하여 계정 및 보안 역할을 생성 및 유지관리하고 사용자를 프로비저닝합니다. Oracle Essbase 데이터베이스 생성 Oracle Hyperion Profitability and Cost Management 애플리케이션 생성 및 관리 Profitability 애플리케이션 콘솔을 사용하여 새 Profitability and Cost Management 애플리케이션을 생성하고 관리합니다. 애플리케이션 환경설정을 지정하고 유지 관리 공동 차원 및 멤버를 선택할 수 있도록 Profitability 애플리케이션 콘솔을 사용하여 모델 데이터베이스 빌드 모델 내에서 요소(예: POV, 규칙 세트, 규칙) 생성 및 유지 관리 POV 복사, 계산, 검증, 데이터 입력, 추적 할당 수행 Essbase에 배포 및 모델 계산
	<p>▲ 주의:</p> <p>Profitability and Cost Management 관리자가 Essbase ASO 배포를 수행하려면 Essbase 액세스 권한도 지정되어야 합니다.</p>
고급 사용자, 역할 유형 = 고급	<ul style="list-style-type: none"> 아티팩트 임포트 및 익스포트 Lifecycle Management 유틸리티를 사용하여 개발 또는 테스트 등의 환경에서 작업과 같은 다른 환경으로 데이터를 승격합니다 Profitability and Cost Management 모델 구성요소를 백업 및 복원합니다 비즈니스 객체 변경사항 모니터 쿼리를 생성, 편집, 복사, 삭제 및 실행합니다 모델 밸런싱 및 검증 수행 모델 내에서 요소(예: POV, 규칙 세트, 규칙) 생성 및 유지 관리 POV 복사, 계산, 검증, 데이터 입력 및 할당 추적을 수행합니다. 아티팩트 임포트 및 익스포트 Essbase에 배포 및 모델 계산 쿼리를 생성, 편집, 복사, 삭제 및 실행합니다 모델 밸런싱 및 검증 수행
대화식 사용자, 역할 유형 = 대화식	<ul style="list-style-type: none"> 모든 모델링 화면 보기 규칙 세트 및 규칙 사용 규칙 밸런싱 보기 할당 추적 보기 쿼리를 정의 및 실행합니다.
보기 사용자, 역할 유형 = 대화형	<p>다음 기능에 대한 보기 권한만 갖습니다.</p> <ul style="list-style-type: none"> 할당 추적 규칙 밸런싱

상세 Profitability에 대한 보안 역할

표 2-2 상세 Profitability and Cost Management 보안 역할

보안 역할	설명
관리자(admin), 역할 유형 = 관리자	<ul style="list-style-type: none"> • 애플리케이션 환경설정을 지정하고 유지 관리 • Profitability 애플리케이션 콘솔을 사용하여 모델 데이터베이스 빌드 • 보고 뷰를 생성하고 관계형 데이터베이스에 배포 • 다음 기능 생성, 읽기(보기), 업데이트 및 삭제 <ul style="list-style-type: none"> - 단계 - 드라이버 - POV - 드라이버 연계 - 지정 - 애플리케이션 환경설정 - 계산 규칙 - 작업 라이브러리 및 상태 - 테이블 등록 • 다음 태스크 수행 <ul style="list-style-type: none"> - POV 복사 - 검증 - 배포 - 계산 - 작업 중지 • Lifecycle Management 유틸리티를 사용하여 개발 또는 테스트 등의 환경에서 작업과 같은 다른 환경으로 데이터를 승격합니다. • 데이터 импорт 및 익스포트 • Oracle Hyperion Profitability and Cost Management 모델 구성요소 백업 및 복원 • 비즈니스 객체의 변경사항을 모니터링합니다. • Essbase 보고 데이터베이스를 배포, 업데이트 및 바꾸고 데이터를 전송합니다.

▲ 주의:

Profitability and Cost Management 관리자가 Essbase ASO 배포를 수행하려면 Oracle Essbase 액세스 권한도 지정되어야 합니다.

표 2-2 (계속) 상세 Profitability and Cost Management 보안 역할

보안 역할	설명
고급 사용자, 역할 유형 = 고급	<ul style="list-style-type: none"> • Oracle Hyperion Shared Services를 사용하여 계정 및 보안 역할을 생성 및 유지관리하고 사용자를 프로비저닝합니다. • 보고 뷰를 생성하고 관계형 데이터베이스에 배포 • 다음 기능 생성, 읽기(보기), 업데이트 및 삭제 <ul style="list-style-type: none"> - 단계 - 드라이버 - POV - 드라이버 연계 - 지정 - 애플리케이션 환경설정 - 계산 규칙 - 작업 라이브러리 및 상태 - 테이블 등록 • 다음 태스크 수행 <ul style="list-style-type: none"> - POV 복사 - 검증 - 계산 - 작업 중지 - Essbase 보고 데이터베이스를 배포, 업데이트 및 바꾸고 데이터를 전송합니다. - 배포

▲ 주의:

Profitability and Cost Management 고급 사용자가 Essbase에 데이터를 입력하고 Essbase ASO 배포를 수행하려면 애플리케이션 생성 및 애플리케이션 관리자 같은 Essbase 액세스 권한도 지정되어야 합니다.

✎ 주:

고급 사용자가 태스크를 수행하기 위해 반드시 특정 보안 역할이 필요한 것은 아닙니다. 예를 들어 고급 사용자가 [계산] 화면에서 계산을 실행하는 경우 이 태스크는 태스크 플로우를 생성하고 백그라운드에서 실행합니다. 고급 사용자가 [태스크 플로우 관리] 태스크에서 직접 이 태스크에 액세스하려는 경우가 아니면 [태스크 플로우 관리] 역할이 없어도 이 태스크를 수행할 수 있습니다.

표 2-2 (계속) 상세 Profitability and Cost Management 보안 역할

보안 역할	설명
대화식 사용자, 역할 유형 = 대화식	다음 기능 보기(읽기) <ul style="list-style-type: none"> • 단계 • 드라이버 • POV • 드라이버 연계 • 지정 • 애플리케이션 환경설정 • 계산 규칙 • 작업 라이브러리 및 상태 • 테이블 등록
보기 사용자, 역할 유형 = 대화형	다음 기능 보기(읽기) <ul style="list-style-type: none"> • 단계 • 드라이버 • POV • 드라이버 연계 • 지정 • 애플리케이션 환경설정 • 작업 라이브러리 및 상태 • 테이블 등록
태스크플로우 관리, 역할 유형 = Shared Services 역할	태스크 플로우를 생성 및 편집하는 데 필요합니다. 자세한 내용은 <i>Oracle Enterprise Performance Management System 사용자 보안 관리 가이드</i> 를 참조하십시오.
태스크플로우 실행, 역할 유형 = Shared Services 역할	사용자가 태스크 플로우를 실행하고 볼 수만 있도록 하는 데 필요합니다. 이 역할을 가진 사용자는 태스크 플로우를 생성하거나 편집할 수 없습니다. 자세한 내용은 <i>Oracle Enterprise Performance Management System 사용자 보안 관리 가이드</i> 를 참조하십시오.

Profitability and Cost Management의 변경사항 감사

Oracle Hyperion Shared Services Console의 감사 기능을 사용하여 애플리케이션의 작업 및 변경사항을 모니터한 다음 결과를 자세히 설명하는 감사 보고서를 생성할 수 있습니다.

다음과 같은 세 가지 유형의 감사 보고서를 사용할 수 있습니다.

- 보안 보고서
- 아티팩트 보고서
- 구성 보고서

감사 보고서에는 다음 정보를 비롯하여 선택한 감사 영역에 대한 작업 세부정보가 포함됩니다.

- 날짜
- 애플리케이션
- 사용자
- 아티팩트 유형 및 이름

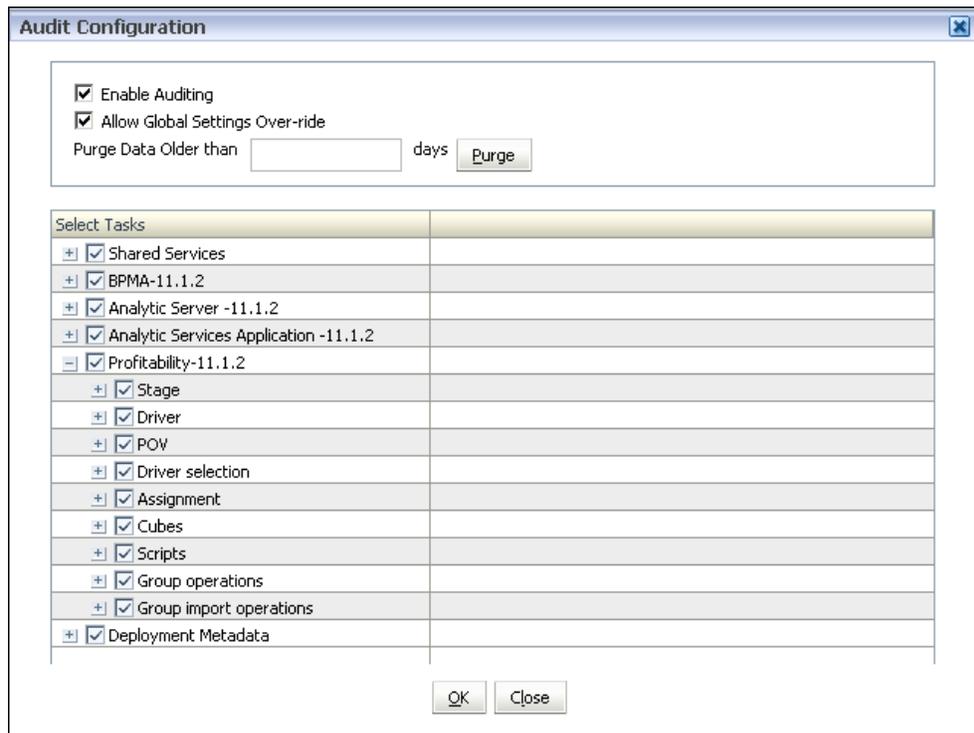
- 수행된 태스크

다음 절차에 설명된 대로 보고서를 생성하려면 먼저 감사를 활성화해야 합니다. 이러한 보고서를 .CSV 파일로 내보낼 수 있습니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

감사를 사용으로 설정하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Shared Services Console** 순으로 선택합니다.
2. Shared Services Console에서 **관리, 감사 구성** 순으로 선택합니다.

[감사 구성] 화면이 표시됩니다.



3. **감사 사용**을 선택합니다.

글로벌 설정 대체 허용 및 태스크 선택 목록이 활성화됩니다.

4. **태스크 선택** 아래에서 감사에 대해 활성화할 애플리케이션 영역을 선택합니다. 전체 영역을 선택하거나 각 옵션을 확장하여 모니터링할 별도의 단계를 선택할 수 있습니다.

주:

이러한 태스크는 상세 Profitability 애플리케이션에만 적용됩니다.

표 2-3 Profitability and Cost Management 감사 태스크

모니터된 영역	사용가능한 모니터된 태스크
단계	<ul style="list-style-type: none"> • 단계 생성 • 단계 수정 • 단계 삭제 • 단계 익스포트 • 단계 임포트
드라이버	<ul style="list-style-type: none"> • 드라이버 생성 • 드라이버 수정 • 드라이버 삭제 • 드라이버 익스포트 • 드라이버 임포트
POV	<ul style="list-style-type: none"> • POV 생성 • POV 수정 • POV 삭제 • POV 익스포트 • POV 임포트
드라이버 선택	<ul style="list-style-type: none"> • 드라이버 선택 생성 • 드라이버 선택 수정 • 드라이버 선택 삭제
지정	<ul style="list-style-type: none"> • 지정 생성 • 지정 수정 • 지정 삭제
지정 규칙	<ul style="list-style-type: none"> • 지정 규칙 생성 • 지정 규칙 수정 • 지정 규칙 삭제 • 규칙 선택 생성 • 규칙 선택 제거
스크립트	<ul style="list-style-type: none"> • 할당 스크립트 배포 • 계보 스크립트 배포 • POV 복사 스크립트 배포 • 할당 스크립트 실행 • 계보 스크립트 실행 • POV 복사 스크립트 실행
그룹 작업	<ul style="list-style-type: none"> • 지정 복사 • 지정 삭제 • 지정 규칙의 선택 항목 삭제 • 드라이버 선택 복사 • 드라이버 선택 규칙 삭제 • 드라이버 선택 예외 삭제
그룹 임포트 작업	<ul style="list-style-type: none"> • 단계 임포트 • POV 임포트 • 드라이버 임포트 • 드라이버 선택 임포트 • 지정 임포트

표 2-3 (계속) Profitability and Cost Management 감사 태스크

모니터된 영역	사용가능한 모니터된 태스크
그룹 익스포트 작업	<ul style="list-style-type: none"> • 단계 익스포트 • POV 익스포트 • 드라이버 익스포트 • 드라이버 선택 익스포트 • 지정 익스포트
LCM(Lifecycle Management)	<ul style="list-style-type: none"> • 파일에서 LCM 로드 • 파일에 LCM 저장 • LCM 익스포트 • LCM 임포트

5. **확인을 누릅니다.**
감사 구성이 저장되었음을 확인하는 메시지가 표시됩니다.
6. **예를 누릅니다.**
7. **선택사항:** Oracle Enterprise Performance Management System 사용자 보안 관리 가이드에 요약된 대로 감사 보고서를 생성하여 감사 결과를 검토합니다.

출력 로그 파일

관리자는 Oracle Hyperion Enterprise Performance Management Workspace에서 로그 파일을 생성할 수 있습니다.

로그 파일은 기술자가 시스템 또는 환경 문제를 확인하거나 개발자가 보고서나 API 프로그램을 디버그하도록 지원할 수 있습니다.

다음 로그 파일은 Oracle Hyperion Profitability and Cost Management 관련 정보에 사용할 수 있습니다.

표 2-4 Profitability and Cost Management 로그 파일

로그 파일	설명
hpcm.log	Profitability and Cost Management는 애플리케이션이나 서버에서 전송된 애플리케이션 특정 메시지를 수집하는 서버측 로그 파일인 애플리케이션을 생성합니다. 기본적으로 로그 파일은 C:\oracle\Middleware\user_projects\domains\EPMSys\servers\P 에 있습니다. 이 로그 파일에 액세스하려면 시스템 관리자에게 문의하십시오.
SharedServices_Security_Client	Oracle Hyperion Shared Services 클라이언트측 로그 파일은 Profitability and Cost Management와 Common Security Services 간의 핸드셰이크에 대한 세부정보를 제공합니다. 기본적으로 이 로그 파일은 C:\oracle\Middleware\user_projects\domains\EPMSys\servers\P 에 있습니다.

관련 제품 및 애플리케이션에 대한 추가 로그 파일은 *Oracle Enterprise Performance Management System 설치 및 구성 가이드*를 참조하십시오.

로그 파일에 캡처할 세부정보 레벨을 수정하려면 *Oracle Hyperion Enterprise Performance Management System 설치 및 구성 문제 해결 가이드*에서 "시스템 로그 사용"을 참조하십시오.

ODL(Oracle Diagnostic Logging) 파일

각 Oracle Hyperion Enterprise Performance Management Workspace 제품의 모든 구성 로깅 파일에 대한 웹 애플리케이션의 ODL(Oracle Diagnostic Logging) 위치는 중앙 집중식입니다.

Oracle Hyperion Profitability and Cost Management의 경우 구성 로깅 파일은 다음 위치에 있습니다.

C:\oracle\Middleware\user_projects\domains\EPMSys\config\fmwconfig\servers\Profitability0
파일 이름은 logging.xml입니다.

구성 로깅 파일에 대한 자세한 내용은 *Oracle Hyperion Enterprise Performance Management System 설치 및 구성 문제 해결 가이드*에서 "EPM 로그 사용"을 참조하십시오.

II부

Profitability and Cost Management 애플리케이션 생성

참조:

- [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#)

3

Profitability 애플리케이션 콘솔로 애플리케이션 생성

참조:

- [Profitability 애플리케이션 콘솔 표시](#)
Profitability 애플리케이션 콘솔은 Oracle Hyperion Enterprise Performance Management Workspace에서 열 수 있습니다.
- [Profitability 애플리케이션 콘솔 사용 가이드라인](#)
Profitability 애플리케이션 콘솔을 사용하면 여러 가지 방법으로 Oracle Hyperion Profitability and Cost Management 애플리케이션 및 차원을 관리할 수 있습니다.
- [Essbase 마스터 큐브의 차원으로 애플리케이션 생성](#)
Profitability 애플리케이션 콘솔에서 Essbase 마스터 큐브의 차원을 사용하여 애플리케이션을 생성할 수 있습니다.
- [관리 원장 애플리케이션 생성](#)
플랫 파일의 차원을 사용하거나 템플릿 파일을 임포트하여 관리 원장 애플리케이션을 생성할 수 있습니다.
- [Profitability 애플리케이션 콘솔의 애플리케이션 작업](#)
플랫 파일 및 템플릿 파일에서 생성된 애플리케이션으로 작업하려면 Profitability 애플리케이션 콘솔을 사용합니다.
- [관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫 파일 준비](#)
관리 원장 Profitability 애플리케이션을 생성하고 업데이트하기 위한 템플릿과 플랫 파일을 준비할 수 있습니다.
- [Profitability and Cost Management용 Essbase 마스터 큐브 생성](#)
관리자 또는 다른 적절한 보안 프로비저닝을 가진 사용자는 Oracle Essbase 마스터 큐브에 차원 및 차원 멤버를 생성할 수 있습니다.

Profitability 애플리케이션 콘솔 표시

Profitability 애플리케이션 콘솔은 Oracle Hyperion Enterprise Performance Management Workspace에서 열 수 있습니다.

수익성 애플리케이션 콘솔을 열어 보려면 다음을 수행합니다.

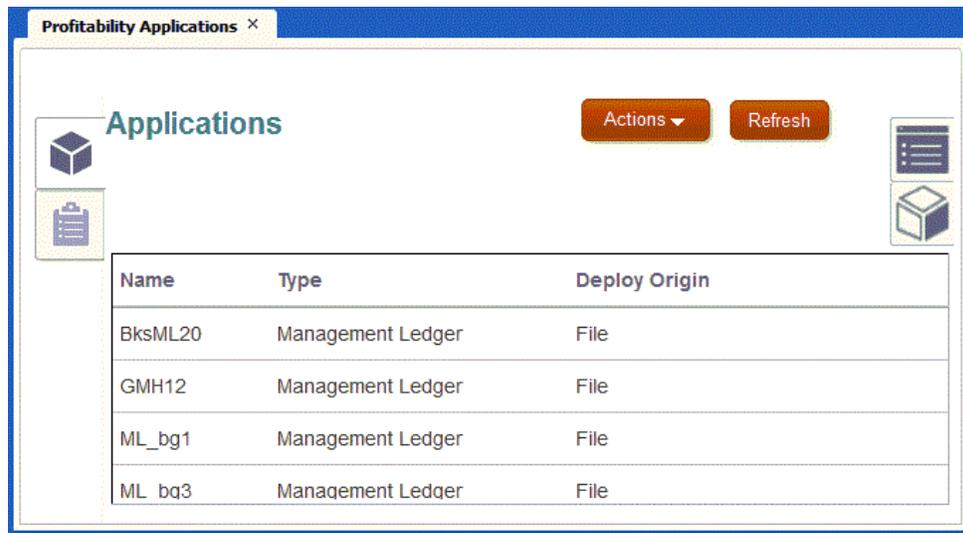
1. EPM Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.

Profitability 애플리케이션 콘솔이 표시되어 **애플리케이션** (📦) 및 **작업 라이브러리** (📋) 탭을 보여줍니다. Profitability 애플리케이션 콘솔에는 기존 애플리케이션과 애플리케이션 유형을 비롯하여 이러한 애플리케이션이 마스터 큐브 또는 파일에서 배포되었는지, 사용으로 설정되어 있는지가 나열됩니다.

팁:

기존 애플리케이션에 대한 요약 정보를 보려면  을 누릅니다. 선택된 애플리케이션에 포함된 차원을 보려면  을 누릅니다.

그림 3-1 Profitability 애플리케이션 콘솔, 애플리케이션 탭



2. 작업을 선택하여 선택된 애플리케이션에서 다양한 작업을 수행하거나 새 애플리케이션을 생성합니다.

[Profitability 애플리케이션 콘솔의 애플리케이션 작업](#) 및 [Profitability 애플리케이션 콘솔 작업 라이브러리의 태스크 보기](#)를 참조하십시오.

3. 현재 탭을 업데이트하려면 **새로고침**을 누릅니다.

주:

특히 Oracle Hyperion Profitability and Cost Management 서비스를 재시작한 후에는 Profitability 애플리케이션 콘솔을 여는 데 몇 초가 걸릴 수 있습니다. Chrome 및 Edge Chromium 브라우저를 사용하면 이로 인해 페이지가 페이지 대기 또는 종료 옵션에 응답하지 않는다는 팝업 메시지가 표시될 수 있습니다. 이 메시지는 무시할 수 있습니다. 페이지는 결국 예상대로 열립니다.

Profitability 애플리케이션 콘솔 사용 가이드라인

Profitability 애플리케이션 콘솔을 사용하면 여러 가지 방법으로 Oracle Hyperion Profitability and Cost Management 애플리케이션 및 차원을 관리할 수 있습니다.

- 차원 소스가 마스터 큐브로 설정된 **작업, 새로 작성** 옵션을 사용하여 Profitability and Cost Management 애플리케이션을 생성합니다. 이 유형의 애플리케이션은 Oracle

Essbase를 해당 차원의 소스로 사용합니다([Essbase 마스터 큐브의 차원으로 애플리케이션 생성](#)).

- **차원 소스가 파일로 설정된 작업, 새로 작성** 옵션을 사용하여 애플리케이션을 생성합니다. 이 유형의 애플리케이션은 차원 정의가 포함된, 특별히 형식 지정된 플랫폼 파일을 사용합니다([관리 원장 애플리케이션에만 해당, 플랫폼 파일의 차원을 사용하여 관리 원장 애플리케이션 생성](#) 참조).
- **작업, 템플릿 임포트**를 사용하여 기존 관리 원장 애플리케이션에서 템플릿 익스포트를 통해 생성된 차원, 메타데이터 및 기타 아티팩트가 포함된 템플릿 파일로 **파일** 유형의 애플리케이션을 생성합니다([관리 원장 애플리케이션에만 해당, 템플릿 파일을 임포트하여 관리 원장 애플리케이션 생성](#) 참조).

주:

원래 차원 생성 시스템([원본 배포] 유형 **마스터 큐브**의 경우 Essbase, [원본 배포] 유형 **기본**의 경우 플랫폼 파일)을 사용하면 애플리케이션 차원만 수정할 수 있습니다. 다음을 참조하십시오.

- [Profitability and Cost Management](#)용 Essbase 마스터 큐브 생성
- [관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비](#)

Essbase 마스터 큐브의 차원으로 애플리케이션 생성

Profitability 애플리케이션 콘솔에서 Essbase 마스터 큐브의 차원을 사용하여 애플리케이션을 생성할 수 있습니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.
2. **작업, 새로 작성** 순으로 선택합니다.
새 애플리케이션 대화상자가 표시됩니다.

그림 3-2 Profitability 애플리케이션 콘솔의 새 애플리케이션 대화상자

New Application

* Application Name: MLtest

Description: BG test app

Instance Name: PROFITABILITY_WEB_APP

Web Server: SLC06XBE.us.oracle.coi

Essbase Application Server: EssbaseCluster-1

Shared Services Project: Foundation

Application Type: Management Ledger

Dimension Source: Master Cube

Next Cancel

3. 새 애플리케이션 대화상자에서 다음 정보를 입력하고 다음:을 누릅니다.
 - 애플리케이션 이름
 - 선택 사항: 애플리케이션에 대한 설명
 - 드롭다운 목록에서 인스턴스 이름 선택
 - 웹 서버의 주소가 표시됩니다.
 - 애플리케이션의 **Essbase 애플리케이션 서버**
 - 애플리케이션의 **Shared Services 프로젝트**
 - 차원 소스는 마스터 큐브가 기본값입니다.
4. 마스터 큐브를 차원 소스로 선택하고 차원을 제공할 Oracle Essbase 마스터 큐브의 이름을 선택한 다음 애플리케이션에 포함할 차원을 선택합니다.
5. 완료를 누릅니다.

차원 소스가 마스터 큐브인 상태로 완료를 누르면 다음 작업이 시작됩니다.

 - 새 애플리케이션을 생성하기 위한 새 태스크 플로우가 시작됩니다. 결과는 [작업 라이브러리] 탭(📁)에서 볼 수 있습니다.
 - 소스 Essbase 데이터베이스의 차원이 검증됩니다. 검증 이슈가 있는 경우 태스크 플로우가 종료되고 작업 라이브러리 탭에 오류 메시지가 표시됩니다. 세부정보를 보려면 오류 링크를 누릅니다. 검증 오류를 수정하고 새 애플리케이션을 생성하는 단계를 반복합니다.

**팁:**

작업할 때 [작업 라이브러리] 탭에서 **새로고침**을 누르고 작업이 성공하면 [애플리케이션] 탭을 누릅니다.

검증이 완료되면 **마스터 큐브의 원본 배포**를 통해 애플리케이션 탭에 새 애플리케이션이 추가됩니다.

관리 원장 애플리케이션 생성

플랫 파일의 차원을 사용하거나 템플릿 파일을 임포트하여 관리 원장 애플리케이션을 생성할 수 있습니다.

참조:

- [플랫 파일의 차원을 사용하여 관리 원장 애플리케이션 생성](#)
- [템플릿 파일을 임포트하여 관리 원장 애플리케이션 생성](#)

플랫 파일의 차원을 사용하여 관리 원장 애플리케이션 생성

Profitability 애플리케이션 콘솔에서 플랫 파일의 차원을 사용하여 관리 원장 애플리케이션을 생성하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.
2. **작업, 새로 작성** 순으로 선택합니다.
3. **새 애플리케이션 화면**에서 다음 정보를 입력하고 다음을 선택합니다.
 - **애플리케이션 이름**
 - **선택 사항:** 애플리케이션에 대한 **설명**
 - 드롭다운 목록에서 **인스턴스 이름** 선택
 - **웹 서버**의 주소가 표시됩니다.
 - 애플리케이션의 **Essbase 애플리케이션 서버**
 - 애플리케이션의 **Shared Services 프로젝트**
 - **차원 소스**에서 **마스터 큐브**는 기본값이며 **기본도** 관리 원장 애플리케이션에 사용할 수 있습니다.
4. **새 애플리케이션 화면**에서 **기본**을 선택합니다.
5. 새 관리 원장 애플리케이션의 규칙 및 잔액 차원에 사용할 이름을 입력합니다. **완료**를 누릅니다. **고유 원본 배포**를 통해 **애플리케이션** 탭에 애플리케이션이 추가됩니다.
6. 추가 차원을 애플리케이션에 추가하려면 **애플리케이션 차원 업데이트**의 단계를 수행합니다.
7. 차원이 완료되면 **검증 및 사용** 옵션을 사용하여 규칙을 추가할 수 있습니다(**기타 애플리케이션 작업 수행**).

템플릿 파일을 임포트하여 관리 원장 애플리케이션 생성

템플릿 익스포트에서는 애플리케이션 마이그레이션 및 백업을 위해 관리 원장 애플리케이션을 템플릿 파일로 익스포트하는 방법에 대해 설명합니다. 이 섹션에서는 해당 파일을 임포트하여 다른 애플리케이션에서 이전에 익스포트된 애플리케이션 메타데이터, 차원 메타데이터, 모델 아티팩트로 새 애플리케이션을 생성하는 방법에 대해 설명합니다.

템플릿 파일을 사용하여 관리 원장 애플리케이션을 생성하려면 다음을 수행합니다.

1. **템플릿 익스포트**에 설명된 대로 템플릿 파일을 익스포트합니다.
2. Profitability 애플리케이션 콘솔의 애플리케이션 탭에서 **작업**, **템플릿 임포트** 순으로 선택합니다.
3. 서버 또는 로컬 컴퓨터에서 템플릿 파일의 위치를 선택합니다.

주:

서버에서 로드된 파일은 먼저 `import_export` 폴더에 복사해야 합니다. 이 폴더는 `<MIDDLEWARE_HOME/user_projects/epmsystem1/import_export`와 같이 LCM(Lifecycle Management)에 사용되는 폴더와 동일합니다.

4. 확장자가 `.zip`인 파일을 찾아 선택합니다.
해당 파일에 입력 데이터 폴더가 있는 경우 **입력 데이터 임포트**를 선택하여 임포트에 포함할 수 있습니다.
5. 다음을 누릅니다.
파일이 적합한 형식인 경우 임포트가 시작됩니다. 그렇지 않으면 오류 메시지가 표시됩니다.

작업 라이브러리 탭() , **새로고침** 순으로 눌러 임포트 상태를 확인할 수 있습니다.

Profitability 애플리케이션 콘솔의 애플리케이션 작업

플랫 파일 및 템플릿 파일에서 생성된 애플리케이션으로 작업하려면 Profitability 애플리케이션 콘솔을 사용합니다.

참조:

- Profitability 애플리케이션 콘솔에서의 애플리케이션 작업 정보
- 애플리케이션 설명 및 Shared Services 프로젝트 편집
- 기타 애플리케이션 작업 수행
- 애플리케이션 차원 업데이트
- Profitability 애플리케이션 콘솔 작업 라이브러리의 태스크 보기

Profitability 애플리케이션 콘솔에서의 애플리케이션 작업 정보

Profitability 애플리케이션 콘솔의 **애플리케이션** 탭에는 플랫폼 파일 및 템플릿 파일(기본 유형의 **원본 배포**)와 Oracle Essbase 마스터 큐브에서 생성된 Oracle Hyperion Profitability and Cost Management 애플리케이션이 나열됩니다.

선택한 애플리케이션의 **설명** 및 **Shared Services 프로젝트**를 편집할 수 있습니다([애플리케이션 설명 및 Shared Services 프로젝트 편집](#)).

Profitability 애플리케이션 콘솔에서 생성된 애플리케이션의 경우 **작업** 메뉴인 **삭제, 복제, 검증 및 사용, 재등록, 메타데이터 검증, 차원 업데이트**를 사용하여 다음 태스크를 수행할 수 있습니다. 관리 원장 애플리케이션의 경우 **템플릿 импорт 및 템플릿 익스포트**도 수행할 수 있습니다. 자세한 내용은 이 섹션의 시작 부분에 나열된 항목을 참조하십시오.

애플리케이션 설명 및 Shared Services 프로젝트 편집

개요 창에서는 애플리케이션 설명과 Oracle Hyperion Shared Services 프로젝트만 수정할 수 있습니다. 개요 창에 표시된 다른 애플리케이션 관련 정보를 변경하려면 새 애플리케이션을 생성해야 합니다.

애플리케이션을 편집하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.

Profitability 애플리케이션 콘솔이 열리고 **애플리케이션** 및 **작업 라이브러리** 탭을 보여줍니다.

2. 수정할 애플리케이션을 선택하고 **개요** 버튼()을 누릅니다.
3. 필요에 따라 정보를 보고 수정합니다.
 - 필요한 경우 **설명**을 수정합니다.
 - 드롭다운 목록에서 다른 **Shared Services 프로젝트** 선택
4. **저장**을 누릅니다.

애플리케이션을 재등록하려면 [기타 애플리케이션 작업 수행](#)을 참조하십시오.

기타 애플리케이션 작업 수행

Profitability 애플리케이션 콘솔에서 생성된 애플리케이션의 경우 **작업** 메뉴인 **삭제, 복제, 검증 및 사용, 재등록, 메타데이터 검증, 차원 업데이트**에서 다음 태스크를 수행할 수 있습니다. 다음의 3단계에서 **복제** 아래 참고를 참조하십시오.

주:

관리 원장 애플리케이션의 경우 템플릿 파일을 익스포트 및 임포트할 수도 있습니다 ([템플릿 익스포트 및 템플릿 파일을 임포트하여 관리 원장 애플리케이션 생성](#)).

Profitability 애플리케이션 콘솔에서 애플리케이션 작업을 수행하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.
2. **애플리케이션** 탭에서 타겟 애플리케이션을 선택합니다.
3. **작업**을 선택하고 다음 옵션 중 하나를 선택합니다.

- **삭제** - 선택한 애플리케이션을 삭제합니다.

애플리케이션을 삭제하기 전에 해당 애플리케이션을 필요로 하는 다른 사용자가 없는지 확인합니다.

- **복제** - 선택한 애플리케이션을 복사합니다.

새 애플리케이션 이름을 제공하라는 메시지가 표시됩니다.

 **주:**

Profitability 애플리케이션 콘솔에서 애플리케이션을 복제하면 차원 및 멤버가 복사되고 애플리케이션이 배포됩니다. POV, 계산 규칙 등과 같은 다른 아티팩트를 상세 Profitability 애플리케이션에 추가하려면 소스 시스템에서 해당 아티팩트를 익스포트하고 LCM(Lifecycle Management)을 사용하여 새 애플리케이션으로 임포트해야 합니다 ([EPM System Lifecycle Management를 사용한 데이터 마이그레이션](#)). 관리 원장 애플리케이션의 경우 템플릿 파일을 익스포트한 다음 새 애플리케이션으로 임포트할 수 있습니다(입력 데이터 포함, [관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비 참조](#)).

- **검증 및 사용** - 선택된 애플리케이션을 검증하고 적합한 경우 사용으로 설정합니다.

결과는 [작업 라이브러리] 탭()에 표시됩니다. 검증 오류가 발생하는 경우 자세한 내용을 보려면 링크를 누르십시오. 쉽게 수정할 수 있도록 오류 파일을 생성할 수 있습니다.

- **재등록** - 선택한 애플리케이션의 Oracle Hyperion Shared Services 등록을 새로고침합니다.

- **메타데이터 검증** - 선택된 애플리케이션의 배포된 데이터에 대해 횡차원 검증을 실행하지만 적합한 애플리케이션을 사용하도록 설정하지 않습니다.

적합한 애플리케이션을 사용하도록 설정하려면 **검증 및 사용**을 선택합니다.

- **차원 업데이트** - 선택된 애플리케이션의 차원 멤버를 추가하거나 제거합니다.

자세한 내용은 [애플리케이션 차원 업데이트](#)를 참조하십시오.

4. 모든 확인 프롬프트에 응답합니다.

 **팁:**

작업 라이브러리 탭을 보고 새로고쳐 Profitability 애플리케이션 콘솔 작업의 현재 상태를 확인합니다([Profitability 애플리케이션 콘솔 작업 라이브러리의 태스크 보기](#)).

애플리케이션 차원 업데이트

추가, 삭제, 변경 또는 이름 바꾸기 등 **기본**의 원본 배포를 통해 애플리케이션에서 차원 또는 차원 멤버를 변경해야 하는 경우 Oracle Essbase 마스터 큐브에서 차원을 업데이트한 다음, Oracle Hyperion Profitability and Cost Management 애플리케이션을 업데이트해야 합니다.

▲ 주의:

Essbase 마스터 큐브에서 차원 이름을 바꾸거나 차원을 삭제해야 하는 경우 필요하면 기존 Profitability and Cost Management 애플리케이션을 업데이트할 수 있도록 별도의 Essbase 마스터 큐브를 생성하는 것을 고려해 보십시오. 또한 Essbase 마스터 큐브 데이터베이스의 이름이 기반이 되는 Profitability and Cost Management 애플리케이션의 이름과 동일하지 않은 경우 Profitability 애플리케이션 콘솔을 사용하여 차원을 업데이트할 수 없습니다.

수익성 애플리케이션 콘솔을 사용하여 차원을 업데이트할 때마다 작업 라이브러리에 작업 항목이 자동으로 생성되어 애플리케이션을 검증하고 업데이트합니다.

Profitability 애플리케이션 콘솔에서 생성된 Profitability 애플리케이션의 Essbase 차원을 업데이트하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.
2. **애플리케이션** 탭에서 업데이트할 차원 및 차원 멤버가 포함된 애플리케이션을 선택합니다.
3. **작업, 차원 업데이트** 순으로 선택합니다.

일반적으로 **사전 업데이트 분석**을 수행하고 **차원 업데이트**를 선택합니다.

4. **선택사항: 사전 업데이트 분석**에서 **차원 검증**을 선택하여 선택한 각 변경사항이 적합한지 확인합니다. 이 옵션은 차원을 업데이트하지 않습니다. 모든 차원이 선택된 경우 이 옵션은 애플리케이션 레벨 차원 검증도 실행합니다.

검증 오류와 영향 분석 결과가 작업 라이브러리에 표시됩니다.

5. **선택사항: 사전 업데이트 분석**에서 **영향 분석**을 선택하여 배포되는 모델과 Profitability and Cost Management의 기존 모델을 비교합니다. 영향 분석이 선택된 경우에는 차원 검증 옵션이 자동으로 선택됩니다.

다음 정보가 표시됩니다.

- 새 멤버
- 삭제된 멤버
- 상위 멤버가 변경된 멤버
- 레벨0 변경사항이 있는 멤버
- 참조 모델 아티팩트에 대한 영향

검증 오류와 영향 분석 결과가 작업 라이브러리에 표시됩니다.

6. **선택사항: 차원 업데이트**를 선택하여 선택된 차원을 변경사항으로 업데이트합니다.
7. **차원에서** 다음 중 하나를 수행합니다.

- 기본의 원본 배포를 사용하여 애플리케이션을 업데이트하는 경우 업데이트할 차원을 선택합니다.

 주:

선택되지 않은 차원 멤버는 이전에 포함되어 있었어도 생략되거나 삭제됩니다.

- 플랫폼 파일을 사용하여 기본의 원본 배포를 통해 관리 원장 애플리케이션에 대해 차원을 추가하고 업데이트하는 경우 업데이트할 차원의 플랫폼 파일을 찾아 선택합니다.
8. 확인을 누릅니다. 예외사항은 모두 작업 라이브러리에 보고됩니다.

사전 업데이트 분석을 선택한 경우 수정된 차원이 검증되지만 업데이트되지는 않습니다.

차원 업데이트를 선택한 경우 선택된 차원 멤버가 차원 선택에 따라 추가, 업데이트, 삭제되거나 선택된 플랫폼 파일이 임포트됩니다.

 주:

플랫폼 파일로 업데이트하는 경우 모든 플랫폼 파일이 임포트될 때까지 필요한 단계를 반복합니다. 검증 및 사용 옵션을 사용하여 파일이 올바르게 임포트되었는지 확인해야 합니다.

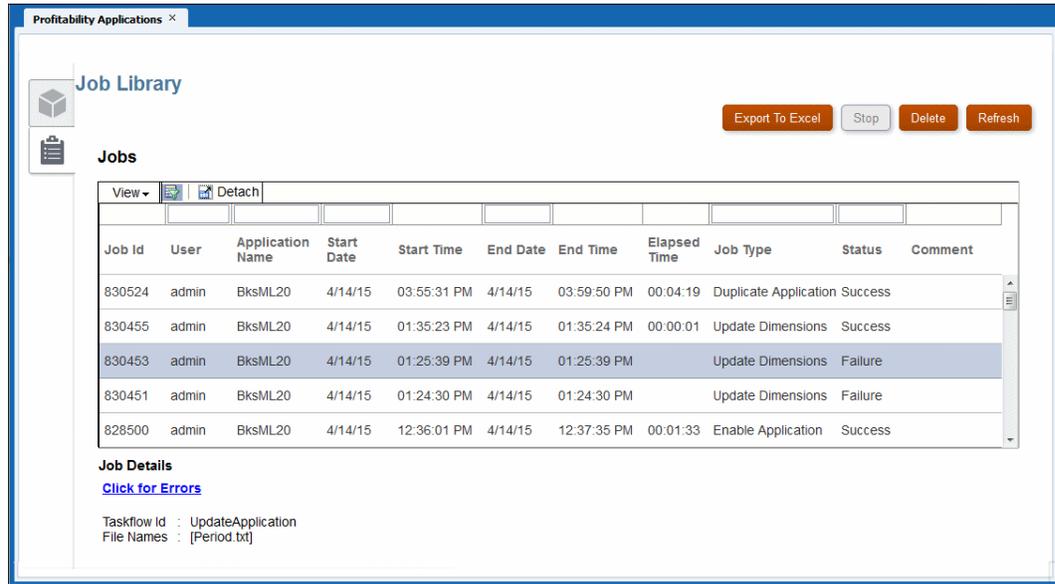
Profitability 애플리케이션 콘솔 작업 라이브러리의 태스크 보기

작업 라이브러리 탭은 Profitability 애플리케이션 콘솔에서 생성된 모든 작업 목록을 제공합니다.

작업 라이브러리를 보려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, Profitability 애플리케이션** 순으로 선택합니다.
2. 작업 라이브러리 탭을 선택합니다 ()

그림 3-3 수익성 애플리케이션 콘솔 작업 라이브러리 화면



작업 라이브러리 화면에 포함되어 있는 제어는 다음과 같습니다.

- **Excel로 익스포트 버튼** - 작업 라이브러리 테이블을 Microsoft Excel 파일에 저장합니다.
- **중지 버튼** - 선택된 작업을 취소합니다.
- **삭제 버튼** - 선택된 작업을 작업 라이브러리 테이블에서 제거합니다.
- **새로고침 버튼** - 가장 최신 작업 정보로 작업 라이브러리 테이블을 업데이트합니다.
- **보기 메뉴** - 열 순서를 숨기고, 표시하고, 재정렬합니다. 작업 라이브러리 테이블을 분리하고 다시 연결합니다. 각 열의 맨위에 예제별 쿼리 상자를 표시하고 숨깁니다.
- **필터 버튼** - , 각 열의 맨위에 예제별 쿼리 상자를 표시하고 숨깁니다. 예제별 쿼리 상자와 일치하는 텍스트를 입력하여 작업 라이브러리 테이블의 특정 항목을 선택합니다.
- **분리 버튼** - , 자체 창의 작업 라이브러리 테이블을 분리하거나 다시 눌러 연결합니다.

3. 각 작업에 대한 정보를 봅니다.

- **작업 ID**는 시스템에서 지정된 순차적 식별 번호입니다.
- **사용자**는 처리할 태스크를 제출한 개인의 사용자 이름을 식별합니다.
- **애플리케이션 이름**에는 태스크가 실행되는 애플리케이션의 이름이 표시됩니다.
- **시작 날짜/시간**에는 작업이 제출되었거나 실행 스케줄링된 날짜와 시간이 표시됩니다.
- **종료 일자/시간**에는 성공 여부에 상관없이 작업이 중지된 일자 및 시간이 표시됩니다.
- **경과 시간**은 시작 시간과 종료 시간 사이의 차이입니다.
- **작업 유형**에는 수행 중인 태스크의 유형(예: 애플리케이션 생성 또는 차원 업데이트)이 표시됩니다.
- **상태**에는 작업의 현재 상태(예: 실행 중, 성공 또는 실패)가 표시됩니다.
- **설명**에는 사용자가 입력한 설명이나 특정 작업에 대한 세부정보가 표시됩니다. 태스크를 제출할 때 설명이 입력됩니다.

4. **선택사항: 작업 세부정보**에서 선택한 작업에 대한 추가 요약 레벨 세부정보를 검토합니다. 작업 세부정보에는 **태스크 플로우 ID**가 포함되며 오류 또는 영향 분석 정보 같은 자세한 내용을 눌러 볼 수 있는 하이퍼링크가 포함될 수 있습니다. 하이퍼링크가 제공되는 경우 정보를 파일에 저장할 수 있습니다.

 **주:**

태스크 플로우 ID는 특정 태스크에 대한 시스템 생성 태스크 ID이며 <애플리케이션 이름>:<태스크 이름><생성된 태스크 플로우 번호> 형식으로 표시됩니다.

예를 들어 생성된 태스크 플로우 번호가

AppMgmt_DeployApplication_D20120824T08520_5ed로 표시될 수 있습니다. 여기서 **AppMgmt**는 애플리케이션 이름이고, **DeployApplication**은 태스크이고, **D20120824T08520_5ed**는 생성된 태스크 플로우 인스턴스 ID입니다.

5. **선택사항: 보기 옵션 또는 필터** 버튼을 사용하여 예제별 쿼리 상자를 표시한 경우 각 열의 맨위에 있는 텍스트 상자에서 일치하는 텍스트를 상자에 입력하여 작업을 찾습니다.
6. **선택사항: 필요한 경우 중지**를 눌러 실행 중 상태의 선택된 태스크를 종료합니다.

 **주:**

버튼을 누른 후 태스크 플로우가 빠르게 중지되는 동안 데이터 상태에 일관성이 있도록 결과에 영향을 주는 태스크에 추가 시간이 필요할 수도 있습니다.

 **주:**

열린 애플리케이션 내 작업 라이브러리 화면과 Profitability 애플리케이션 콘솔의 작업 라이브러리 화면에서는 둘 다 작업 내역을 Excel 스프레드시트 파일로 익스포트할 수 있습니다. 익스포트된 파일은 .html 형식으로 생성됩니다. 파일을 저장할 때 파일에 .html 접미사를 지정하면 MS Word, MS Excel 또는 .html 파일을 처리하는 기타 프로그램의 모든 브라우저에서 파일을 열 수 있습니다.

관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플리트 및 플랫폼 파일 준비

관리 원장 Profitability 애플리케이션을 생성하고 업데이트하기 위한 템플리트와 플랫폼 파일을 준비할 수 있습니다.

참조:

- [관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플리트 및 플랫폼 파일 준비 정보](#)

- 템플릿 익스포트
- 각 관리 원장 차원에 대한 플랫폼 파일 준비
- 플랫폼 파일 속성 정보
- 플랫폼 파일의 주석 정보
- 플랫폼 파일 샘플

관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비 정보

Profitability 애플리케이션 콘솔로 익스포트된 템플릿을 임포트하여 관리 원장 애플리케이션을 생성할 수 있습니다. 플랫폼 파일을 임포트하여 기본 유형 원본 배포를 통해 관리 원장 애플리케이션에서 차원을 추가하거나 업데이트할 수도 있습니다.

[관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비](#)에 나열된 항목은 관리 원장 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비 방법에 대해 설명합니다.

템플릿 익스포트

하나의 애플리케이션 "템플릿" 파일을 생성하는 단일 작업에서 애플리케이션 메타데이터, 차원 메타데이터 및 프로그램 아티팩트를 비롯한 전체 관리 원장 애플리케이션을 익스포트하여 템플릿 파일이 생성됩니다. 템플릿 파일 익스포트 및 임포트는 애플리케이션을 백업하거나 애플리케이션을 다른 환경(예: 테스트에서 프로덕션)으로 마이그레이션하는 데 유용합니다.

템플릿 파일을 익스포트하려면 다음을 수행합니다.

1. Profitability 애플리케이션 콘솔을 표시하고 관리 원장 애플리케이션을 선택합니다.
2. **작업, 템플릿 익스포트** 순으로 선택합니다.
3. 파일 이름을 검토하고 필요한 경우 변경하십시오. .zip 확장자는 익스포트된 템플릿 파일에 추가됩니다.
4. POV 데이터가 표시되면 모든 POV 데이터를 익스포트할지 여부를 선택하거나 익스포트되지 않아야 하는 데이터의 선택란을 지웁니다.
5. **선택사항: 입력 데이터 포함**을 선택하여 입력 데이터를 익스포트합니다. 선택하는 경우 모든 선택된 POV의 입력 데이터가 익스포트됩니다.
6. **확인**을 눌러 익스포트를 시작합니다.

파일은 서버의 import_export 폴더에 생성됩니다. 이 폴더는 <MIDDLEWARE_HOME>/user_projects/epmsystem1/import_export와 같이 LCM(Lifecycle Management)에 사용되는 폴더와 동일합니다.

7. 작업 라이브러리 탭() , **새로고침** 순으로 눌러 익스포트 상태를 확인할 수 있습니다.

[템플릿 파일을 임포트하여 관리 원장 애플리케이션 생성](#)에 설명된 대로 템플릿 파일이 생성되면 이 파일을 임포트하여 새 관리 원장 애플리케이션을 생성할 수 있습니다.

각 관리 원장 차원에 대한 플랫폼 파일 준비

차원 플랫폼 파일을 사용하여 관리 원장 애플리케이션을 업데이트하고 생성할 수 있습니다. 기존 멤버가 파일에서 생략되면 업데이트 중에 제거됩니다. 자세한 내용은 이 섹션의 시작 부분에 나열된 항목을 참조하십시오.

플랫폼 파일을 사용하여 관리 원장 애플리케이션에서 차원을 추가하거나 바꾸려면 다음과 같이 플랫폼 파일을 준비합니다.

1. 텍스트 편집기에서 이 섹션에 설명된 형식에 따라 차원마다 하나의 플랫폼 파일을 생성합니다. 각 플랫폼 파일이 완전하게 차원을 정의해야 합니다. 차원 플랫폼 파일은 다음이 포함된 텍스트 파일입니다.

- 차원 및 멤버 등록정보 순서가 지정된 머리글 레코드
- 해당 등록정보에 대한 값이 포함된 차원 데이터 레코드
- 해당 등록정보에 대한 값이 포함된 개별 멤버 데이터 레코드

주:

멤버 항목이 최종 아웃라인과 동일한 순서여야 하므로 하위 멤버 보다 먼저 상위 멤버가 정의되어야 합니다. 파일에서 멤버의 순서는 Oracle Hyperion Profitability and Cost Management 내에서(예: 멤버 선택기에서) 차원 트리가 표시되는 순서와 동일합니다.

2. 다음 규칙에 따라 차원 및 멤버 머리글 레코드를 생성합니다.

- 심표로 구분하여 원하는 순서대로 속성을 지정할 수 있습니다. 다중 값 속성의 개별 값은 따옴표로 묶어야 합니다. 속성에 값이 여러 개인 경우 따옴표가 필요합니다. 예를 들어 사용자정의 속성(UDA) 속성의 제공된 멤버에 복수의 UDA 문자열이 있을 수 있습니다. 다음과 같이 따옴표로 묶여 집합임을 나타냅니다.

```
, "myUDA1, myUDA2, myUDA3",
```

- 특성 이름 또는 UDA와 같은 복수 값 속성 내에서 사용되는 심표는 항상 값 구분자로 해석됩니다. 값이 따옴표로 묶여 있는 경우 단일 값 속성에 심표가 포함될 수 있습니다(예: "my, value").
- 값에 큰따옴표를 지정하려면 또 다른 따옴표 세트 내에서 값을 큰따옴표로 묶으십시오. 예를 들어 "myValue" 값을 입력하려면 "'myValue'"와 같이 지정하십시오.

주:

최상의 결과를 얻으려면 큰따옴표가 필요한 멤버 이름을 사용하지 마십시오.

- 속성 이름은 대소문자를 구분하지 않습니다. 예를 들어 name, NAME, Name 이름은 모두 동일하게 처리됩니다.
- 일부 속성은 플랫폼 파일의 모든 행에 대해 필수입니다. 속성이 원래 있던 위치의 심표 사이에 값이 표시되어 있지 않으면 속성이 무시되거나 null일 수 있습니다(예: , ,). 예를 들어, 멤버 레코드에서 머리글의 차원 속성이 무시됩니다.
- 값이 없거나 부적합한 값이 있는 등록정보는 필수 등록정보에 대해 기본 설정됩니다 (플랫폼 파일 속성 정보 참조).
- 주석이 지원됩니다(see 플랫폼 파일의 주석 정보 참조).
- 규칙 및 잔액 차원은 마스터 큐브에서 배포 시 초기 설정되는 것과 동일한 방식으로 초기 설정됩니다. 해당 차원에 대해 파일을 импорт할 필요가 없습니다. 멤버가

자동으로 생성됩니다. 규칙 차원 멤버에 대해 별칭을 지정할 수 없습니다. 규칙 및 간액 차원 멤버의 차원 멤버 이름이 프로그래밍 방식으로 정의되고 변환됩니다.

- 차원 정렬 및 해결 순서 선행 작업은 다음과 같이 처리됩니다. 속성 차원이 항상 마지막 (해당 위치에서 알파벳순으로 정렬됨)인 것을 제외하고는 차원 이름을 사용하여 알파벳순으로 처리가 진행됩니다. 멤버 해결 순서 속성이 사용된 경우 기타 고려 사항을 대체합니다.

플랫폼 파일 속성 정보

표 1에서는 차원 플랫폼 파일에 있는 각 속성의 형식을 설명합니다. 플랫폼 파일 예는 [플랫폼 파일 샘플](#)을 참조하십시오.

멤버의 Essbase 멤버 해결 순서 등록정보를 포함할 수 있으나 플랫폼 파일의 첫번째 라인에 등록정보를 추가하여 해당 위치를 표시해야 합니다.



주:

차원 이름은 주석 행 다음 첫 번째 행에 제공됩니다.

표 3-1 차원 플랫폼 파일에 정의된 속성

속성 이름	속성 유형	차원 유형	기본값	설명
차원 유형	차원	모두	일반	차원 유형을 식별합니다. 적합한 예로는 계정, 기간, 연도, 버전, 시나리오, POV1, POV2, POV3, POV4, 차원, 속성, 규칙이 있습니다. 이 목록에 등록정보 이름이 없는 경우 기본값은 "일반"입니다. 이 위치는 차원 이름(Gen1 행인 경우) 또는 멤버 이름을 나타냅니다. POV 순서는 연도/POV1 = 1, 기간/POV2 = 2, 시나리오/POV3 = 3, 버전/POV4 = 4순입니다.
저장영역 유형	차원	일반, 계정, 엔티티, POV	희소	희소 및 밀집

표 3-1 (계속) 차원 플랫폼 파일에 정의된 속성

속성 이름	속성 유형	차원 유형	기본값	설명
차원 이름(속성)	차원, 멤버	일반	없음	머리글에 속성 차원 이름을 입력하는 경우 해당 위치는 로드되고 있는 차원의 연결된 속성 차원을 나타냅니다. 해당 위치의 모든 문자열은 "속성 멤버"로 처리됩니다. 예를 들어 머리글에 "내 속성 차원"을 지정한다고 가정해 보십시오. 해당 파일의 데이터 행에서 동일한 위치의 문자열은 해당 속성 차원의 멤버라고 가정합니다. 따라서 "모든 제품"에 대해 데이터 행에 "내 속성 멤버1"을 지정할 수 있으며 로더는 "모든 제품"에 속성 연결로 해당 멤버를 지정합니다.
주식	차원, 멤버	일반, 계정, 엔티티, 국가, POV	없음	
계층 유형	차원, 멤버	일반, 계정, 엔티티, 국가, POV	저장	사용, 저장, 동적, 사용안함
BSO 데이터 저장영역	차원, 멤버	일반, 계정, 엔티티, 국가, POV	공유 안 함	레이블 전용, 저장, 공유 안 함, 공유, 동적 계산 및 저장, 동적 계산
ASO 데이터 저장영역	차원, 멤버	일반, 계정, 엔티티, 국가, POV	공유 안 함	레이블 전용, 저장, 공유 안 함, 공유
2단계 계산	차원, 멤버	일반, 계정, 엔티티, 국가, POV	N	허용되는 값은 True 또는 False입니다(N 또는 Y).
ASO 차원 공식	차원, 멤버	일반, 계정, 엔티티, 국가, POV	없음	

표 3-1 (계속) 차원 플랫폼 파일에 정의된 속성

속성 이름	속성 유형	차원 유형	기본값	설명
멤버 해결 순서	멤버	일반, 계정, 엔티티, 국가	없음	Oracle Essbase에서는 해결 순서 번호에 따라 차원의 멤버를 평가하는 순서를 결정합니다. 1에서 127 사이의 숫자를 입력할 수 있습니다. 해결 순서 번호가 가장 높은 멤버가 먼저 평가됩니다(예: 해결 순서 번호가 5인 공식보다 해결 순서가 20인 공식이 먼저 평가됨). 해결 순서 번호가 같은 멤버는 데이터베이스 아웃라인에 차원이 표시되는 순서대로 평가됩니다. 해결 순서 번호가 없는 멤버는 해결 순서 번호가 있는 멤버 뒤에 평가됩니다.
통합 유형	멤버	일반, 계정, 엔티티, 국가	사용되지 않음	더하기, 빼기, 곱하기, 나누기, 퍼센트, 무시, 사용되지 않음
UDA	차원, 멤버	일반, 계정, 엔티티, 국가	없음	단일 텍스트 값 또는 텍스트 값 목록(심표로 묶임)입니다. 차원의 경우 값은 유형의 차원 이름 = UDA입니다. 멤버의 경우 지정된 UDA 차원의 멤버입니다. 예: myUda1 "myUda1, myUda2"
상위	멤버	일반, 계정, 엔티티, 국가, POV, 속성		상위 멤버 이름을 식별합니다. null인 경우 멤버가 2세대임을 의미합니다. 순서가 중요합니다. 참조되는 상위 이름이 이전에 먼저 정의되어 있어야 합니다.
별칭: <i>별칭 표</i>	차원, 멤버	모두		예: "별칭: 기본값", "별칭: T1"

표 3-1 (계속) 차원 플랫폼 파일에 정의된 속성

속성 이름	속성 유형	차원 유형	기본값	설명
설명	차원, 멤버	모두		선택사항 - 기본값이 없습니다.

플랫폼 파일의 주석 정보

단일 라인 주석의 경우 라인의 첫 문자로 해시 문자를 추가합니다. 빈 라인은 무시됩니다.

블록 주석은 주석 블록 시작 표시기(!)로 나타내고 블록 종료 표시기(#--)로 종료됩니다. 중간 라인은 주석 처리할 필요가 없습니다.

예:

```
#!-start of comment block

Comment within block

Another comment within block

#--!
```

플랫폼 파일 샘플

그림 1에서는 고객 차원에 대한 샘플 파일을 보여 줍니다.

그림 3-4 Customers.txt 관리 원장 플랫폼 파일 샘플

```
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,,
BB200,SPARSE,,,StoreData,StoreData,N,+,,,Big Box,Q Mart,Q Mart
BB300,SPARSE,,,StoreData,StoreData,N,+,,,Big Box,Bike Depot,Bike Depot
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,,
```

Profitability and Cost Management용 Essbase 마스터 큐브 생성

관리자 또는 다른 적절한 보안 프로비저닝을 가진 사용자는 Oracle Essbase 마스터 큐브에 차원 및 차원 멤버를 생성할 수 있습니다.

그런 다음 Profitability 애플리케이션 콘솔에서 이러한 항목을 Oracle Hyperion Profitability and Cost Management 애플리케이션으로 임포트합니다. 마스터 큐브는 ASO 애플리케이션입니다. 동일한 마스터 큐브를 사용하여 Profitability and Cost Management 애플리케이션을 여러 개 생성할 수 있습니다. 애플리케이션 유형(상세 또는 관리 원장)은

애플리케이션을 생성할 때 콘솔에서 설정합니다. Essbase 마스터 큐브에서 제공되지 않습니다.

이 섹션에서는 Profitability 애플리케이션 콘솔과 함께 사용할 Essbase 마스터 큐브를 생성하는 방법에 대해 설명합니다. 대신 플랫폼 파일을 사용하려면 [관리 원장 Profitability 애플리케이션 생성 및 업데이트를 위한 템플릿 및 플랫폼 파일 준비](#)를 참조하십시오.

Essbase 마스터 큐브를 생성하려면 다음을 수행합니다.

1. 다음 지침에 따라 애플리케이션의 차원과 멤버를 생성하여 Profitability and Cost Management 애플리케이션에 대한 Essbase 마스터 데이터베이스를 생성합니다.
 - 차원 유형
 - ASO 차원 및 멤버 등록정보
 - BSO 및 멤버 차원 등록정보 지정

Essbase에서 애플리케이션과 데이터베이스를 생성하는 방법에 대한 자세한 지침은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

▲ 주의:

Essbase 마스터 데이터베이스 이름은 Essbase 마스터 큐브 이름과 동일해야 합니다. 이름이 다르면 사용자가 Profitability 애플리케이션 콘솔을 사용하여 배포하는 경우 차원을 업데이트할 수 없습니다.

2. Profitability and Cost Management 애플리케이션을 생성할 사용자가 관리자 또는 애플리케이션 생성 프로비저닝을 가진 기타 사용자로 프로비저닝되었는지 확인합니다.
3. Profitability and Cost Management 애플리케이션을 생성합니다.

▲ 주의:

Essbase 마스터 큐브에서 차원의 이름을 바꾸거나 차원을 삭제할 경우 이러한 차원을 참조하는 이미 배포된 Profitability and Cost Management 애플리케이션의 차원을 더 이상 제대로 업데이트할 수 없습니다. 즉, 차원 유형을 정의하는 UDA는 어떤 방식으로든 수정하면 안 됩니다. 차원의 이름을 바꾸거나 차원을 삭제하려면 필요한 경우 기존 Profitability and Cost Management 애플리케이션이 업데이트될 수 있도록 별도의 Essbase 마스터 데이터베이스를 생성하는 것이 좋습니다. [사전 업데이트 분석] 옵션의 기존 검증은 이러한 경우를 보고하지 않습니다.

✎ 주:

Essbase 마스터 큐브를 배포할 때 ASO 저장영역 유형, ASO 공식, 계층 유형 등 ASO 차원 멤버와 해당 등록정보를 자동으로 Essbase로 읽어옵니다. 수동 작업은 필요하지 않습니다..

BSO 특정 차원 및 멤버 등록정보는 배포 중에 자동으로 읽을 수 없습니다. 이를 관리하려면 공식, 데이터 저장영역, 차원 저장영역 유형 등의 BSO 차원 멤버 및 등록정보에 대한 UDA(사용자 정의 속성 차원)를 생성하여 BSO 데이터베이스를 Essbase와 맞춰야 합니다.



주:

마스터 큐브 메소드를 사용하여 생성된 상세 애플리케이션이 중복되는 경우, 중복된 애플리케이션의 큐브를 Essbase에 배포하려면 먼저 Profitability and Cost Management 서비스를 제한해야 할 수 있습니다.

차원 유형

Oracle Essbase 마스터 큐브를 생성하는 경우 각 차원에 DIMTYPE UDA가 지정되어야 합니다.

상세 Profitability에는 시스템 차원 측정항목 상세(DIMTYPE_MEASURES)가 있고 관리 원장 Profitability에는 시스템 차원 규칙(DIMTYPE_RULES) 및 잔액(DIMTYPE_BALANCES)이 있다는 점을 제외하고는 상세 및 관리 원장 Profitability 애플리케이션의 차원 유형은 비슷합니다.

사용자 정의 속성

사용자 정의 속성(UDA)는 차원 멤버의 텍스트 속성을 기준으로 분석을 활성화합니다. 예를 들어 "New Products"라는 UDA를 추가할 수 있습니다.

선택한 멤버의 UDA 이름을 입력합니다.

최대 80자의 영숫자 문자를 입력할 수 있습니다.

마스터 큐브의 속성 차원에는 ASCII 문자(영문자 및 숫자)로만 구성된 차원 이름이 있어야 하고 속성 차원 이름의 첫번째 문자는 a-z 또는 A-Z와 같은 문자여야 합니다.

Oracle Essbase 아웃라인에서는 모든 차원에 고유한 이름이 있어야 합니다. 그렇지 않으면 아웃라인 생성에 실패합니다. 예를 들어 속성 차원 멤버 이름은 일반 차원 이름과 일치할 수 없습니다.

ASO 차원 및 멤버 등록정보

배포 중 표 1에 표시된 ASO 차원 등록정보 및 멤버 등록정보를 Oracle Essbase로 자동으로 가져옵니다. 이러한 등록정보를 Essbase 애플리케이션과 맞추는 데 필요한 수동 작업은 없습니다.

표 3-2 Profitability and Cost Management ASO 차원 및 멤버 등록정보

등록정보 레이블	속성 이름	설명
별칭	별칭	별칭은 배포된 애플리케이션에 표시되는 차원 멤버 이름입니다. 루트 멤버를 제외한 모든 멤버 별칭이 표시됩니다.
주석	주석	차원 또는 멤버에 대한 설명을 입력할 수 있습니다. 설명은 최대 255자를 포함할 수 있습니다. 기본적으로 이 텍스트 상자에는 현재 설명(있는 경우)이 표시됩니다.

표 3-2 (계속) Profitability and Cost Management ASO 차원 및 멤버 등록정보

등록정보 레이블	속성 이름	설명
통합	통합	<p>멤버 통합 등록정보는 1차 하위 구성요소가 상위로 롤업되는 방식을 결정합니다.</p> <p>현재 멤버가 차원 또는 속성이 아닌 경우 다음 통합 연산자 중 하나가 멤버에 지정됩니다.</p> <ul style="list-style-type: none"> • +(더하기) - 기본값 • -(빼기) • *(곱하기) • /(나누기) • %(퍼센트) • ~(통합 중 무시) • ^(통합하지 않음) • NotUsed <p>참고: 집계 저장영역 아웃라인(ASO)에서 통합 연산자 사용과 관련하여 일부 제한이 있습니다. <i>Oracle Essbase Database Administrator's Guide</i>를 참조하십시오.</p>
데이터 저장영역(ASO)	<ul style="list-style-type: none"> • ASODimensionDataStorage(차원 루트 멤버의 경우) • ASOMemberDataStorage(차원 루트 멤버의 경우) 	<ul style="list-style-type: none"> • 차원 및 멤버에 사용할 수 있는 ASO 저장영역 옵션은 다음과 같습니다. • StoreData - 데이터가 차원과 함께 저장됩니다. • ShareData - 이 멤버와 연결된 데이터는 공유할 수 있습니다. ShareData 등록정보는 멤버에만 적용됩니다. 차원 루트 멤버는 공유할 수 없습니다. • NeverShare - 1차 하위 구성요소를 하나 포함하는 상위처럼 내재 공유 관계가 있는 경우에도 이 차원과 연결된 데이터는 공유할 수 없습니다. 이 경우 데이터는 상위 및 1차 하위 구성요소에서 복제됩니다. <p>참고: 이 옵션은 집계 저장영역 아웃라인의 저장된 계층에는 적용되지 않습니다.</p> <ul style="list-style-type: none"> • LabelOnly - 이 차원과 연결된 데이터가 없습니다.
차원 공식(ASO)	ASODimensionFormula(차원 루트 멤버에 해당)	차원에 적용된 계산 공식입니다.
차원 해결 순서	DimensionSolveOrder	선택한 차원의 해결 순서를 나타내는 숫자 값입니다. 예를 들어 이 차원을 두번째로 해결하려면 "2"를 입력합니다.

표 3-2 (계속) Profitability and Cost Management ASO 차원 및 멤버 등록정보

등록정보 레이블	속성 이름	설명
계층 유형 (차원만 해당)	DimensionHierarchyType	<p>[계층 유형] 등록정보는 Essbase의 합산 저장영역(ASO) 데이터베이스에만 적용됩니다.</p> <p>차원의 계층 유형은 다음 값 중 하나로 설정됩니다.</p> <ul style="list-style-type: none"> • STORED - 다음 통합 기호를 사용하는 차원 멤버의 경우 <ul style="list-style-type: none"> - +ADDITION - ~ IGNORE <p>참고: 비통합 또는 IGNORE(~) 연산자는 멤버의 상위가 LABEL_ONLY로 설정된 경우 STORED 계층에서만 사용할 수 있습니다.</p> • DYNAMIC - ADDITION 및 IGNORE를 비롯한 통합 기호를 사용하는 차원 멤버의 경우 또는 차원 멤버에 공식이 포함된 경우 • ENABLED - 대체 계층 구조를 지원하려는 경우
계층 유형 (멤버만 해당)	HierarchyType	<p>멤버에 사용 가능한 계층 유형은 다음과 같습니다.</p> <ul style="list-style-type: none"> • 저장(첫 번째 Gen2 하위 멤버의 경우) • 동적(두 번째 Gen2 하위 멤버 이상의 경우. 두 번째 Gen2 멤버는 대체 계층을 호스트할 수 있음) • 없음(사용하면 안 됨)
멤버 공식(ASO)	ASOMemberFormula	<p>멤버에 적용된 계산 공식입니다.</p>
멤버 해결 순서 (멤버만 해당)	MemberSolveOrder	<p>선택한 멤버의 해결 순서에 대한 숫자 값을 입력합니다. 예를 들어 이 멤버를 두번째로 해결하려면 "2"를 입력합니다. 이 등록정보는 ASO 데이터베이스에만 적용됩니다.</p> <p>해결 순서가 0인 멤버는 해당 차원의 해결 순서를 상속합니다.</p> <p>해결 순서가 같은 멤버는 차원 경렬 순서 등록정보에서 달리 지정하지 않은 경우 데이터베이스 아웃라인에 차원이 표시되는 순서대로 평가됩니다.</p> <p>해결 순서가 없는 멤버는 해결 순서가 있는 멤버 뒤에 평가됩니다.</p>

BSO 및 멤버 차원 등록정보 지정

Oracle Essbase 마스터 큐브는 ASO 애플리케이션이고 ASO 차원과 멤버는 배포 중에 자동으로 읽어옵니다.

이러한 BSO 차원과 멤버를 Essbase BSO 데이터베이스에 포함하려면 Oracle Hyperion Profitability and Cost Management에 배포할 각 차원에 대한 Generation1 멤버가 있어야 합니다. 마스터 아웃라인의 차원 순서는 Profitability and Cost Management 생성 아웃라인의 차원 순서에 의해 결정됩니다.

표 1에 나열된 각 차원 및 멤버에 대해 UDA를 생성해야 합니다. Essbase 작업에 대한 자세한 내용은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

▲ 주의:

Essbase의 특정 멤버에 동일한 유형의 UDA를 여러 개 지정하지 마십시오. 이 경우 Profitability and Cost Management 배포에서 사용할 UDA 유형을 알지 못하므로 일관성 없는 동작이 발생합니다. 예를 들어 동일한 멤버에 DIMTYPE_POV1 및 DIMTYPE_POV2를 둘 다 지정하지 마십시오.

애플리케이션에 수동으로 지정해야 하는 특수 UDA가 필요한 등록정보는 표 1의 내용을 참조하십시오.

✎ 주:

BSO 등록정보는 UDA를 사용하여 지정되므로 사용자가 Essbase 마스터 데이터베이스에서 호환되지 않는 BSO 등록정보를 실수로 지정할 경우 Profitability and Cost Management 내에서 이후에 수행하는 "보고 큐브에 배포" 작업이 실패할 수 있습니다. 이 경우 보고 데이터베이스 배포에 의해 생성된 데이터 및 규칙 파일을 수동으로 로드하여 호환되지 않는 지정을 식별할 수 있습니다.

표 3-3 Profitability and Cost Management에 대한 차원 및 멤버 등록정보

등록정보 레이블	속성 이름	설명
데이터 저장영역(BSO)	<ul style="list-style-type: none"> • BSODimensionDataStorage(차원 루트 멤버에 해당) • BSOMemberDataStorage(차원 멤버에 해당) 	<p>BSO 데이터 저장영역의 경우 다음 UDA 중 하나를 하위 멤버의 Generation1 멤버에 지정합니다.</p> <ul style="list-style-type: none"> • BSODS_LABELONLY • BSODS_NEVERSHARE • BSODS_SHAREDATA • BSODS_DYNCALANDSTORE • BSODS_DYNCALC • BSODS_STOREDATA <p>UDA를 설정할 때 등록정보 BSO 데이터 저장영역에 값이 지정되지 않은 경우 Profitability and Cost Management 배포에서 다음 기본값이 지정된다고 가정합니다.</p> <ul style="list-style-type: none"> • Generation1 멤버에는 "BSODS_LABELONLY"가 지정되지만 다음과 같은 예외가 있습니다. <ul style="list-style-type: none"> – POV Generation1 멤버는 "BSODS_STOREDATA"로 설정됩니다. – 측정항목 Generation1 멤버는 "BSODS_DYNCALC"로 설정됩니다. • 다른 모든 멤버에는 "BSODS_STOREDATA"가 지정됩니다.
차원 공식(BSO)	BSODimensionFormula(차원 루트 멤버에 해당)	<p>BSO 차원 공식의 경우 사용자가 Essbase 마스터 데이터베이스에서 BSO 차원 공식을 지정할 수 없습니다. 데이터베이스가 ASO 데이터베이스여야 하므로 마스터 데이터베이스에서 사용 가능한 공식 속성은 ASO 공식이기 때문입니다. 배포 후에 BSO 공식이 Essbase 보고 데이터베이스에 표시되게 하려면 보고 데이터베이스에 직접 BSO 차원 공식을 입력합니다. 재배포하는 경우 BSO 공식을 다시 입력해야 합니다.</p> <p>해당 BSO CALCULATOR 구문을 사용합니다.</p>

표 3-3 (계속) Profitability and Cost Management에 대한 차원 및 멤버 등록정보

등록정보 레이블	속성 이름	설명
차원 정렬 순서	DimensionSortOrder	Profitability and Cost Management에서 생성된 Essbase 아웃라인에 차원 순서를 설정할 순서로 숫자 값을 입력합니다. 예를 들어 이 차원이 Essbase 아웃라인에서 두 번째인 경우 "2"를 입력합니다. 차원 정렬 순서는 별칭 및 UDA 차원을 제외하고 모델에 있는 모든 차원에 대해 설정해야 합니다. 차원 정렬 순서는 순차적이고 고유하며 1보다 크거나 같아야 합니다.
차원 저장영역 유형	DimensionStorageType	차원 저장영역 유형 등록정보는 Essbase의 블록 저장영역(BSO) 데이터베이스에만 적용됩니다. 기본적으로 값은 "SPARSE"로 설정됩니다. 밀집 설정이 필요한 경우 Generation1 멤버에 "DIMSTORETYPE_밀집" 값을 지정하여 UDA를 지정합니다.
멤버 공식(BSO)	BSOMemberFormula	BSO 멤버 공식의 경우 사용자가 Essbase 마스터 큐브에서 BSO 멤버 공식을 지정할 수 없습니다. 데이터베이스가 ASO 데이터베이스여야 하므로 마스터 큐브에서 사용가능한 공식 속성은 ASO 공식이기 때문입니다. 배포 후에 BSO 공식이 Essbase 보고 데이터베이스에 표시되게 하려면 보고 데이터베이스에 직접 BSO 차원 공식을 입력합니다. 재배포하는 경우 BSO 공식을 다시 입력해야 합니다. 해당 BSO CALCULATOR 구문을 사용합니다.
2단계 계산 (차원만 해당)	BSO_TWOPASS (차원 루트 멤버의 경우)	아웃라인을 통과하는 두 번째 단계에서 멤버를 계산하려면 BSO_TWOPASS를 지정합니다(BSO 데이터베이스에만 해당).
2단계 계산 (멤버만 해당)	BSO_TWOPASS	아웃라인을 통과하는 두 번째 단계에서 멤버를 계산하려면 BSO_TWOPASS를 지정합니다(BSO 데이터베이스에만 해당).

멤버 등록정보에 지정된 BSO(블록 저장영역 옵션) 데이터 저장영역 값 보기

멤버에 지정되는 BSO 데이터 저장영역 값에 대해 이해하려면 BSO 계산 큐브의 Oracle Essbase에 대한 배포가 실패하는 경우 HPM_DIM_MEMBER_PROP_V 뷰를 사용하여 디버그합니다.

뷰는 다음 열로 구성되어 있습니다.

```
APPLICATION_NAME
DIMENSION_NAME
MEMBER_NAME
BSO_DATA_STORAGE
DIMENSION_STORAGE_TYPE
```

쿼리되는 경우 뷰는 배포된 각 Oracle Hyperion Profitability and Cost Management 애플리케이션의 차원 멤버마다 한 행을 반환합니다.

사용자정의 측정항목 차원

이 UserDefinedDriverMeasures 멤버는 애플리케이션 특정 사용자 정의 드라이버 측정항목이 저장되는 위치입니다. ASOMember DataStorage 및 BSOMember DataStorage 등록정보를 다음과 같이 설정합니다.

- 멤버에 하위 멤버가 없는 경우 **StoreData**로 설정됩니다.
- 이 멤버에 멤버가 하위 멤버로 추가되고 이러한 모든 하위 멤버에 IGNORE의 통합 기호가 있으면 **LabelOnly**로 설정됩니다.

주:

모든 드라이버 측정항목은 아웃라인에서 고유해야 합니다. 아웃라인에서 차원의 기존 드라이버 측정항목 이름을 다른 멤버(시스템, POV, 비즈니스 차원 포함)의 이름으로 사용하지 마십시오. 그렇지 않으면 [데이터 입력] 화면에 값이 제대로 표시되지 않습니다.

III부

관리 원장 Profitability 작업

참조:

- 관리 원장 Profitability 애플리케이션 아키텍처 이해
- 관리 원장 차원 이해
- 다른 애플리케이션 관리 태스크 수행

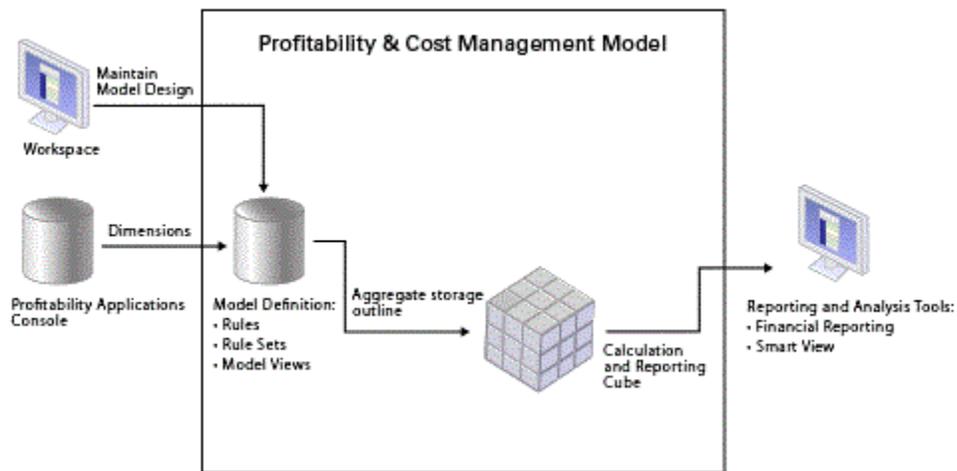
4

관리 원장 Profitability 애플리케이션 아키텍처 이해

Oracle Essbase 맨위에 있는 분석 툴은 Oracle Hyperion Enterprise Performance Management Workspace, Oracle Hyperion Profitability and Cost Management를 통해 액세스할 수 있습니다.

Profitability and Cost Management를 통해 비즈니스 사용자는 수익성 및 비용 관리를 위한 비즈니스를 모델링하고 해당 모델 정보를 사용하여 스크립팅 언어에 대한 이해 없이도 수익성 및 비용 분석을 위해 세분화할 수 있는 Essbase 데이터베이스를 생성할 수 있습니다. 관리 원장 Profitability에 대한 자세한 내용은 [관리 원장 Profitability](#)을 참조하십시오.

그림 4-1 관리 원장 Profitability 제품 아키텍처



EPM Workspace의 모델 메타데이터는 Profitability and Cost Management 모델을 생성할 때 사용되며, 다양한 보고 및 분석 툴에서 계산된 결과가 출력될 수 있습니다.

Profitability and Cost Management는 Oracle Hyperion Shared Services를 활용하여 애플리케이션 메타데이터 및 보안을 중앙 집중식으로 관리합니다.

애플리케이션 관리자는 Profitability 애플리케이션 콘솔을 사용하여 Profitability and Cost Management 차원을 생성합니다. 사용자 액세스는 Shared Services를 사용하여 중앙에서 관리됩니다. 차원 메타데이터가 준비되면 Profitability and Cost Management 애플리케이션이나 모델로 배포됩니다.

모델 디자인에는 Essbase 아웃라인 생성에 필요한 정보와 해당 모델의 Essbase 구성요소에 필요한 계산 스크립트가 포함됩니다. 각 모델에는 다음과 같은 데이터베이스에 대한 액세스 권한이 필요합니다.

- 모델 디자인을 저장하는 관계형 데이터베이스입니다.

- 계산 및 보고를 위한 ASO(합산 저장영역) 데이터베이스가 포함된 Essbase 데이터베이스입니다.

 주:

여러 모델을 저장하는 데 하나의 데이터베이스만 필요합니다.

결과는 Oracle Hyperion Financial Reporting 및 Oracle Smart View for Office와 같은 보고 및 분석 툴에서 볼 수 있습니다.

5

관리 원장 차원 이해

참조:

- [관리 원장 차원 정보](#)
Oracle Hyperion Profitability and Cost Management는 Oracle Essbase 및 Profitability 애플리케이션 콘솔에서 생성된 차원 및 멤버를 사용하여 비즈니스 모델의 여러 구조 요소를 나타냅니다.
- [관리 원장 시스템 차원](#)
관리 원장 애플리케이션에는 규칙 및 잔액 시스템 차원이 포함되어야 합니다.
- [관리 원장 비즈니스 차원](#)
비즈니스 차원은 모델 내에서 제품, 고객, 지역, 직원 등의 비즈니스 특정 객체를 설명합니다.
- [POV 차원](#)
POV 차원은 연도, 시나리오 또는 기간과 같은 모델의 특정 POV나 버전을 나타냅니다.
- [속성 차원](#)
속성 차원은 비즈니스 차원과 연결된 특수한 유형의 차원으로, 다른 연결된 차원의 멤버를 분류하는 데 사용할 수 있는 멤버를 포함합니다.
- [별칭 차원](#)
별칭은 대체 이름, 설명, 언어 또는 차원 정의를 돕는 기타 항목입니다.
- [관리 원장 차원 정렬 순서](#)
차원 정렬 순서 등록정보는 Oracle Hyperion Profitability and Cost Management로 생성된 Oracle Essbase 아웃라인의 차원 순서를 제어합니다.

관리 원장 차원 정보

Oracle Hyperion Profitability and Cost Management는 Oracle Essbase 및 Profitability 애플리케이션 콘솔에서 생성된 차원 및 멤버를 사용하여 비즈니스 모델의 여러 구조 요소를 나타냅니다.

차원 유형은 사전 정의된 기능을 사용할 수 있는 차원 등록정보입니다. 차원 유형의 지정된 특성이 차원의 동작과 기능을 관리합니다. Profitability and Cost Management 및 기타 Oracle Hyperion Enterprise Performance Management Workspace 제품에서 특정 차원 유형이 공유될 수 있으므로 관리자가 다른 제품에 차원 기능을 활용할 수 있습니다.

관리 원장 Profitability 애플리케이션에는 다음 차원이 있습니다.

- [관리 원장 시스템 차원](#)
- [관리 원장 비즈니스 차원](#)
- [POV 차원](#)
- [속성 차원](#)
- [별칭 차원](#)

관리 원장 차원 요구사항

데이터베이스 아웃라인은 모델의 데이터 구조를 제공하며 계산 지침 및 공식을 포함하고 있습니다. Essbase 아웃라인의 차원은 계층 구조식입니다. 데이터는 차원 교차점에 저장됩니다. 관리 원장 Profitability 차원 요구사항은 다음과 같습니다.

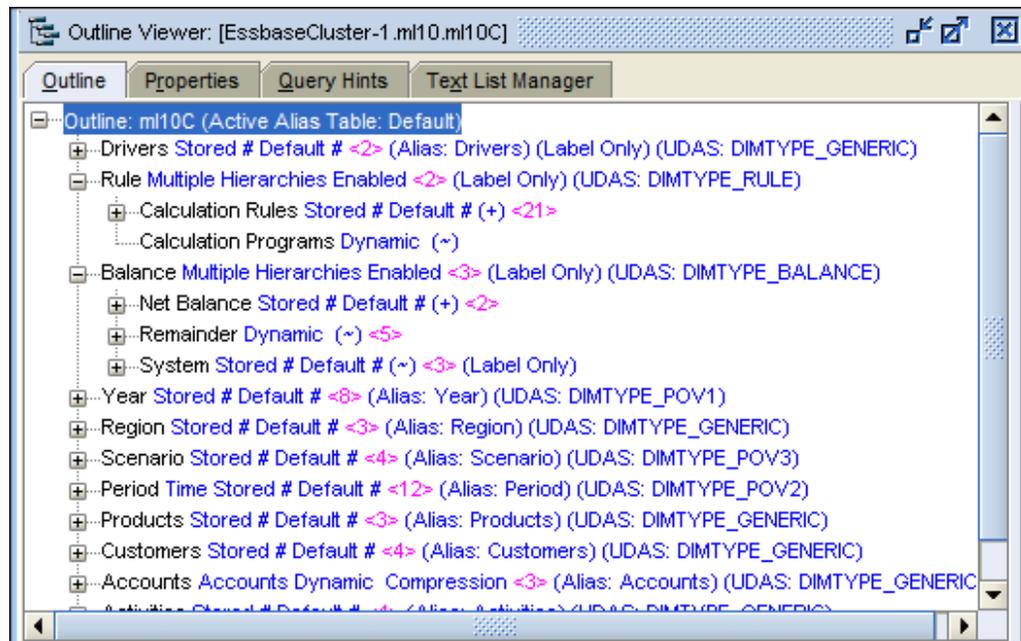
- 애플리케이션 또는 모델에는 하나 이상의 POV 차원이 포함되어야 하고 최대 4개의 POV 차원이 포함될 수 있습니다.
- 애플리케이션에는 **규칙**이라는 시스템 차원이 하나만 포함되어야 합니다.
사용자는 규칙 차원을 편집하고 해당 차원에 더 많은 멤버를 추가할 수 있습니다(예: R1001~R1500). 선택적으로 해당 차원을 지울 수도 있습니다. 규칙 차원의 계산 프로그램 멤버는 편집할 수 없습니다.
- 애플리케이션에는 **잔액**이라는 시스템 차원이 하나만 포함되어야 합니다.
잔액 차원의 시스템 차원 멤버는 편집할 수 없습니다. 그러나 사용자는 대체 계층 구조를 추가할 수 있습니다.
- 비즈니스 차원의 기본 계층에서 중복 멤버가 없는 비즈니스 차원이 하나 이상 있어야 합니다.

▲ 주의:

멤버는 같은 차원 내에서 반복되지 않아야 합니다. 그러나 멤버가 여러 차원에 걸쳐 반복될 수 있습니다.

그림 1에서는 Essbase 콘솔에 표시된 관리 원장 Profitability 데이터베이스의 샘플 Essbase 아웃라인을 보여줍니다.

그림 5-1 관리 원장 Profitability 데이터베이스의 Essbase 아웃라인



 주:

릴리스 11.2.15 이상의 경우 Essbase 21c 웹 콘솔 아웃라인 화면을 사용해야 합니다.

관리 원장 시스템 차원

관리 원장 애플리케이션에는 규칙 및 잔액 시스템 차원이 포함되어야 합니다.

이러한 시스템 차원은 새 관리 원장 애플리케이션이 배포되거나 생성될 때 Profitability 애플리케이션 콘솔에서 채워집니다. 규칙 및 잔액 차원에 대한 자세한 내용은 나열된 섹션을 참조하십시오.

차원 및 멤버를 생성하고 유지관리하는 방법에 대한 자세한 지침은 [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#) 및 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

 주:

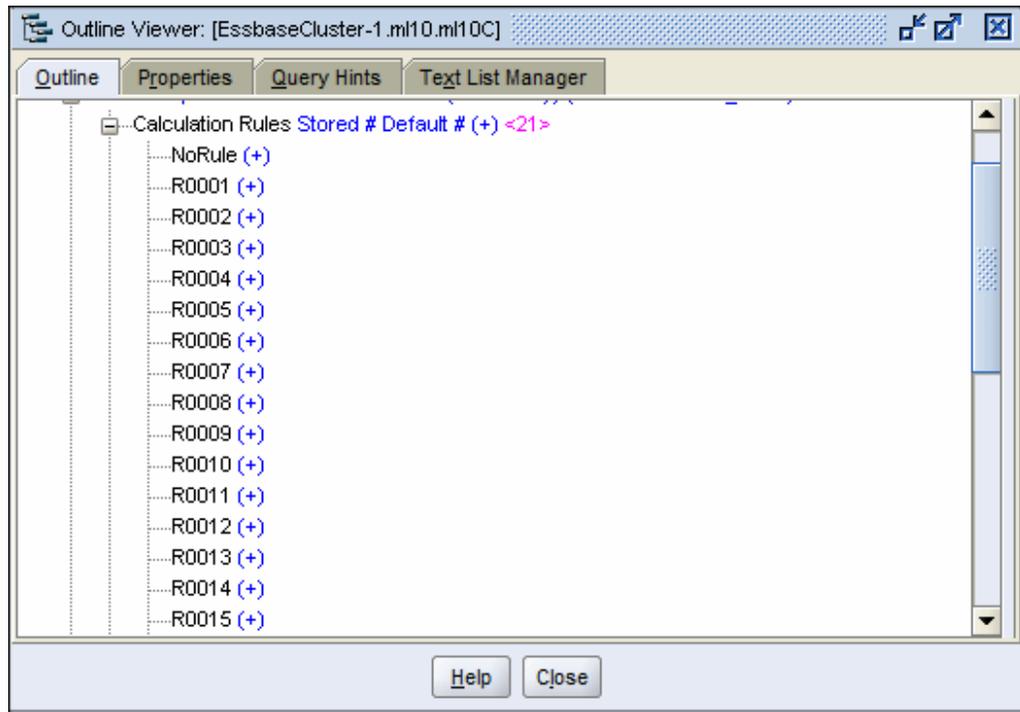
관리 원장 규칙 및 잔액 차원은 초기 설정되며 시스템의 일부에서 허용하는 경우에도(예: Profitability 애플리케이션 콘솔의 차원 업데이트) 편집되지 않아야 하는 시스템 차원입니다. 이러한 차원은 시스템용으로 예약되어 있습니다.

관리 원장 규칙 차원

규칙 차원에는 관리 원장 애플리케이션에 대한 계산 규칙 정의가 포함됩니다.

[그림 1](#)에서는 Oracle Essbase 콘솔의 규칙 차원 아웃라인을 보여줍니다. 다음은 계산 규칙 멤버에 대한 그림입니다.

그림 5-2 관리 원장 규칙 차원 아웃라인



 주:

릴리스 11.2.15 이상의 경우 Essbase 21c 웹 콘솔 아웃라인 화면을 사용해야 합니다.

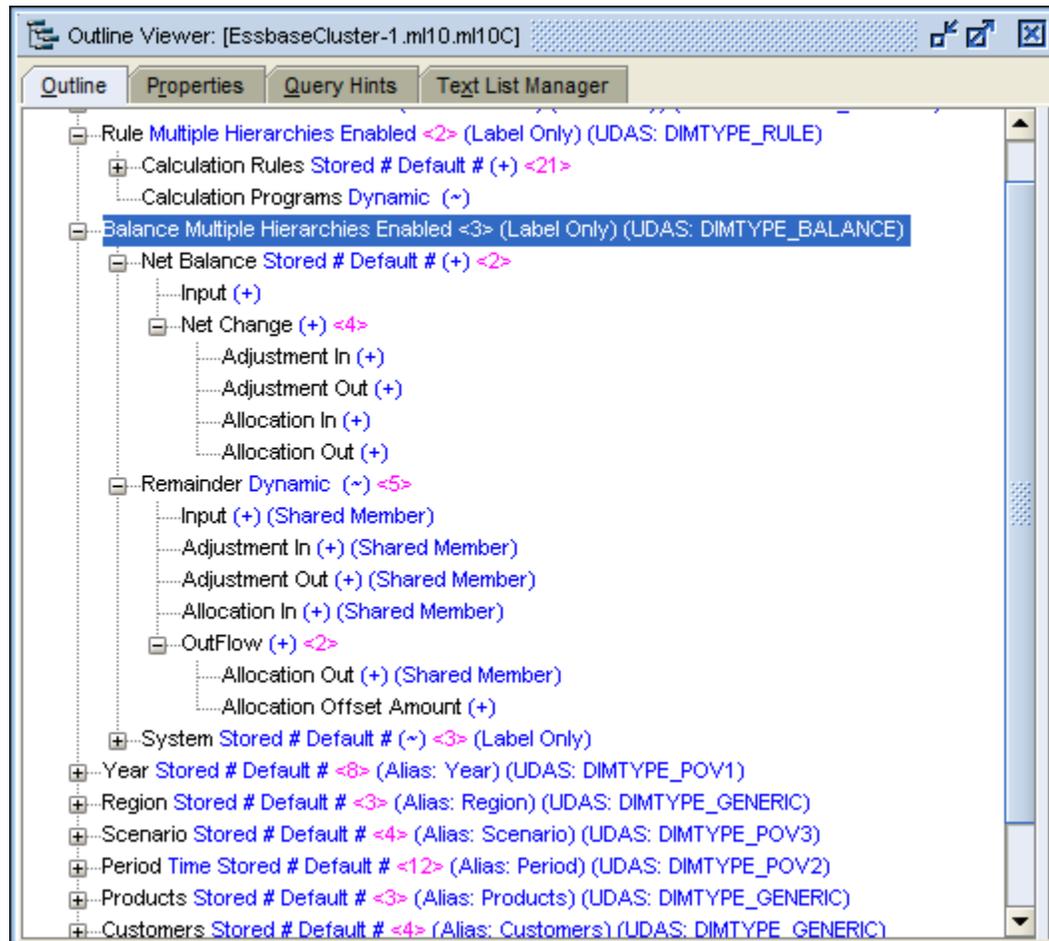
사용자는 계산 규칙을 삭제하고 계산 규칙에 규칙을 추가할 수 있지만 적용할 수 있는 규칙은 NoRule뿐입니다. 기타 규칙은 모두 시스템용으로 예약되었습니다.

그림 1에서는 추가 멤버, 계산 프로그램을 보여줍니다. 이 멤버는 시스템에서 제어되고 사용자가 편집할 수 없습니다.

관리 원장 잔액 차원

그림 1에서는 Oracle Essbase 콘솔의 잔액 차원 아웃라인을 보여줍니다.

그림 5-3 관리 원장 잔액 차원 아웃라인



사용자는 순 잔액의 입력 멤버에 데이터를 추가할 수 있습니다. 나머지 멤버는 규칙 세트 및 규칙에 따라 결정되는 입력 및 출력을 반영합니다. 조정은 드라이버 계산의 결과이고 할당은 규칙 할당의 결과이며 오프셋은 규칙 오프셋 정의에서 생성됩니다. 규칙 집합, 규칙 및 해당 정의에 대한 정보는 *Oracle Hyperion Profitability and Cost Management 사용자 가이드*를 참조하십시오.

이러한 멤버와의 교차에 저장된 데이터는 [규칙 밸런싱] 화면에 표시됩니다.

관리 원장 비즈니스 차원

비즈니스 차원은 모델 내에서 제품, 고객, 지역, 직원 등의 비즈니스 특정 객체를 설명합니다.

비즈니스 차원은 다음 차원 유형의 일부 또는 전체를 사용할 수 있으며 하나 이상의 모델에 적용될 수 있습니다.

- 일반
- 계정
- 엔티티
- 국가

Oracle Essbase 아웃라인이 배포되면 비즈니스 차원이 Oracle Hyperion Profitability and Cost Management 애플리케이션에서 유형 없이 기본 또는 일반 차원으로 생성됩니다. 이 기능으로 Profitability and Cost Management에서 Oracle Hyperion Planning 등의 다른 애플리케이션에 대해 정의된 차원 멤버 및 계층을 재사용할 수 있습니다.

주:

이 차원 유형은 합산 저장영역 아웃라인에는 적용되지 않습니다.

관리 원장 규칙 및 잔액 차원은 초기 설정되며 시스템의 일부에서 허용하는 경우에도(예: Profitability 애플리케이션 콘솔의 차원 업데이트) 편집되지 않아야 하는 시스템 차원입니다. 이러한 차원은 시스템용으로 예약되어 있습니다.

비즈니스 차원 요구사항

관리 원장 애플리케이션의 비즈니스 차원을 생성하는 경우 다음 요구사항이 적용됩니다.

- Gen1 차원 이름 아래의 첫번째 Gen2 1차 하위는 일반적으로 All 멤버로 설정됩니다. 예를 들어 부서 차원에 대해 AllDepartments입니다.
기본 계층은 첫 번째 Gen2 1차 하위 구성요소 아래에 호스트됩니다. 첫 번째 Gen2 계층만 할당 모델링에 사용되며 이 계층에는 공유 멤버를 포함할 수 없습니다.
- 추가 Gen2 멤버는 대체 계층을 호스트할 수 있지만 이러한 계층이 할당 모델링에 사용되지 않습니다. 차원에서 대체 계층을 호스트하려는 경우 차원의 HierarchyType을 "사용"으로 설정하고 첫번째 Gen2 멤버의 HierarchyType을 "저장"으로 설정하고 공유 멤버가 포함된 대체 계층이 있는 Gen2 멤버를 "동적"으로 설정합니다.
이러한 대체 계층은 Profitability and Cost Management 모델링 화면에 표시되지 않으며 Essbase에서만 볼 수 있습니다.

주:

관리 원장 비즈니스 차원에서 레벨 0이 아닌 기준 멤버를 참조하는 공유 멤버를 만들지 마십시오. 만들 경우 POV 복사 및 계산 작업이 실패합니다. 이러한 참조를 공유하면 Essbase에서 ASO 큐브에 대해 지원되지 않는, Profitability and Cost Management에서 레벨 0이 아닌 멤버를 써야 하는 상황이 발생하기 때문입니다.

POV 차원

POV 차원은 연도, 시나리오 또는 기간과 같은 모델의 특정 POV나 버전을 나타냅니다.

조직의 요구사항에 맞게 차원을 사용자정의할 수 있습니다. 예를 들어 POV는 분기, 월, 계절별 그룹 등으로 구성될 수 있습니다.

각 모델에 대해 POV 차원이 하나 이상 필요하며, 최대 4개의 POV 차원을 생성할 수 있습니다.

버전 차원도 사용할 수 있으며, 이 차원은 모델의 다른 인스턴스를 생성하는 데 사용됩니다. 이 버전을 수정하여 여러 가지 전략이나 비즈니스 옵션으로 "가정" 시나리오를 재생할 수 있습니다. 버전 수정을 통해 기능을 구현하고 결과를 비교하여 최상의 작업 방향을 결정할 수 있습니다.

속성 차원

속성 차원은 비즈니스 차원과 연결된 특수한 유형의 차원으로, 다른 연결된 차원의 멤버를 분류하는 데 사용할 수 있는 멤버를 포함합니다.

속성 차원은 제품의 크기 및 색상과 같은 데이터의 특성을 설명합니다.

이러한 속성을 사용하여 차원 멤버의 속성 또는 품질을 기준으로 데이터를 분석할 수 있습니다. 또한 속성 차원은 지정 규칙 생성 시 대상 교차를 필터링하는 데 사용됩니다.



주:

이름 지정 제한에 대한 자세한 내용은 [Essbase 이름 지정 규칙](#)을 참조하십시오.

속성 차원 유형은 다음 두 가지가 있습니다.

- 속성 차원:
 - 부울, 날짜, 숫자 및 텍스트 같은 여러 구조를 사용하여 속성을 생성할 수 있습니다.
 - 속성에는 계층이 있고, 합산 값을 위해 계층을 롤업할 수 있습니다.
 - 지정된 속성 차원의 속성 한 개만 한 멤버에 연결할 수 있습니다.
- UDA(사용자 정의 속성) 차원
 - 속성은 텍스트로만 생성할 수 있습니다.
 - UDA에는 계층이 없으며 보고서에서 쉽게 사용하여 합계를 생성할 수 없습니다.
 - 한 멤버에 여러 UDA를 연결할 수 있습니다.

각 유형의 속성 차원은 모델 및 보고 요구에 따라 다른 이점을 제공합니다. 속성 차원 작업에 대한 자세한 내용은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

이러한 속성을 사용하여 차원 멤버의 속성 또는 품질을 기준으로 데이터를 분석할 수 있습니다. 또한 속성 차원은 지정 규칙 생성 시 대상 교차를 필터링하는 데 사용됩니다.

이름 지정 제한에 대한 자세한 내용은 [Essbase 이름 지정 규칙](#)을 참조하십시오.

별칭 차원

별칭은 대체 이름, 설명, 언어 또는 차원 정의를 돕는 기타 항목입니다.

예를 들어 시스템의 고객 번호를 참조할 수 있지만 화면에 해당 회사 이름을 표시하는 별칭을 지정하여 보다 쉽게 해당 클라이언트를 식별할 수 있습니다. 계정, 통화, 엔티티, 시나리오, 기간, 버전, 연도 및 사용자 정의 차원 번호에 하나 이상의 별칭을 지정할 수 있습니다.

설치가 완료되면 "기본" 별칭 테이블을 사용할 수 있습니다. 재배포 후에는 [규칙 정의] 화면과 같은 관리 원장 멤버 선택기를 사용하는 모든 화면에서 별칭을 볼 수 있습니다.

관리 원장 별칭 차원으로 작업하는 경우 다음 사항에 유의하십시오.

- 동일한 차원 내에서는 중복 멤버 이름 또는 별칭을 사용할 수 없습니다.

- 차원 파일을 관리 원장 애플리케이션으로 임포트하는 경우 머리글에 별칭 테이블을 정의하면 차원의 모든 멤버에 대해 별칭 값을 정의해야 합니다. 별칭 값이 원래 멤버 이름과 달라야 하는 것은 아니지만 파일의 각 멤버 행에 있는 각 별칭 테이블에 대해 값을 지정해야 합니다.

관리 원장 차원 정렬 순서

차원 정렬 순서 등록정보는 Oracle Hyperion Profitability and Cost Management로 생성된 Oracle Essbase 아웃라인의 차원 순서를 제어합니다.

차원 정렬 순서는 모델 내부에서 별칭 및 UDA를 제외한 모든 차원에 대해 설정되어야 합니다.

▲ 주의:

차원 정렬 순서를 비워두면 검증이 실패합니다.

차원 정렬 순서 등록정보는 차원 로드 파일에서 설정합니다. 지침은 [차원 정렬 순서 설정](#)을 참조하십시오.

차원 정렬 순서 권장 사항

다음 권장 사항을 사용하여 차원 정렬을 설정하는 것이 좋습니다.

- 별칭 및 UDA를 제외한 모델의 모든 차원에 대해 차원 정렬 순서를 설정해야 합니다.

✎ 주:

별칭 및 UDA 차원은 Oracle Hyperion Profitability and Cost Management 및 Oracle Essbase의 차원으로 존재하지 않으므로 차원 정렬 순서에 대해 무시됩니다.

- 차원 정렬 순서는 순차적이고 고유하며 1보다 크거나 같아야 합니다.
- 측정항목 차원의 기본값은 1로 설정됩니다.
- AllocationType 차원의 기본값은 2로 설정됩니다.
- 비즈니스 및 POV 차원은 3 이상으로 설정되어야 합니다.
- 속성 차원은 항상 마지막 차원으로 정렬되어야 합니다. 예를 들어 연속적인 12개 차원 중 속성 차원이 4개인 경우 속성 차원은 9, 10, 11 및 12로 설정되어야 합니다.

차원 정렬 순서 설정

모델에 있는 모든 차원의 처리 순서는 차원 로드 파일에 있는 차원 정렬 순서 등록정보를 사용하여 차원 레벨에서 설정해야 합니다.

차원 정렬 순서 제한 사항을 충족해야 하며, 그렇지 않으면 모델 검증에 실패합니다. 제한 사항 전체 목록은 [차원 정렬 순서 권장 사항](#)을 참조하십시오.

차원 정렬 순서가 포함된 로드 파일을 생성하려면 다음을 수행합니다.

1. 외부 프로그램 또는 ETL 프로세스에서 이와 유사한 열을 사용하여 로드 파일을 생성합니다. 차원 정렬 순서에 대한 열이 포함되도록 해야 합니다.

차원의 루트 멤버에 대한 정렬 순서를 입력합니다. 이 예에서는 계/경에 정렬 순서 9가 있습니다.

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 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
"NoAccoun	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	Cost of Gc	SPARSE	STORED		StoreData	StoreData	N		+				0	Materials	Materials
"MAT511	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Frames	Frames
"MAT521	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Wheels	Wheels
"MAT531	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Gearsets	Gearsets
"MAT541	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Seats	Seats
"MAT551	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Controls	Controls
"MAT561	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

2. Profitability 애플리케이션 콘솔에서 작업을 선택한 후 차원 업데이트를 선택하여 차원 정렬 순서가 정의된 파일을 로드합니다.

6

다른 애플리케이션 관리 태스크 수행

참조:

- 관리 원장 Profitability 애플리케이션 관리
- 마법사를 사용하여 관리 원장 애플리케이션 검증 및 배포
Oracle Hyperion Profitability and Cost Management 애플리케이션을 생성한 후에는 애플리케이션을 검증하여 Profitability and Cost Management에 배포해야 합니다.

관리 원장 Profitability 애플리케이션 관리

참조:

- 관리 원장 애플리케이션 및 차원 작업
- Profitability and Cost Management 차원 및 멤버 등록정보
- 계층 유형 등록정보 설정

관리 원장 애플리케이션 및 차원 작업

Oracle Hyperion Profitability and Cost Management에서는 Profitability 애플리케이션 콘솔을 사용하여 Profitability 모델에 사용되는 Oracle Essbase 아웃라인을 빌드할 차원을 선택합니다. 모든 차원 및 멤버는 Profitability 애플리케이션 콘솔에서 생성되며 모델 빌드를 위해 Profitability and Cost Management 애플리케이션으로 임포트됩니다.

Profitability and Cost Management 차원 및 멤버 등록정보

Oracle Hyperion Profitability and Cost Management 차원 및 멤버에 대한 등록정보는 다음 정보를 표시하는 관리 원장 및 원가 관리 차원 및 멤버 등록정보 테이블에 알파벳순으로 표시됩니다.

- 등록정보에 대해 더 읽기 쉬운 표시 이름을 제공하는 등록정보 레이블. 해당되는 경우 연관된 데이터베이스 유형이 이름에 추가됩니다(ASO 또는 BSO). 데이터베이스 유형이 지정되지 않은 경우 등록정보가 두 유형 모두에 적용됩니다.
- 각 등록정보에 대한 설명
- 임포트 및 배치 클라이언트에서 데이터를 업데이트할 때 사용되는 등록정보의 고유한 식별자를 제공하는 등록정보 이름

차원 라이브러리에서 등록정보를 선택할 때 드롭다운 목록 또는 데이터 입력 텍스트 상자를 표시하는 등록정보를 수정할 수 있습니다.

▲ 주의:

모든 Profitability and Cost Management 등록정보는 로컬 값입니다. 한 계층에서 등록정보를 수정한 경우 해당 값이 다른 계층으로 상속될 것으로 간주할 수 없습니다.

계층 유형 등록정보 설정

'계층 유형' 등록정보는 Essbase의 합산 저장영역 데이터베이스에만 적용됩니다.

ADDITION 및 IGNORE 이외의 다른 통합 유형 기호를 사용할 수 있으려면 계층 유형을 'DYNAMIC'으로 설정해야 합니다.

주:

차원 멤버에 공식이 있는 경우 계층 유형을 'DYNAMIC'으로 설정해야 합니다.

계층 유형을 설정하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 관리, 차원 라이브러리** 순으로 선택합니다.
2. 애플리케이션에서 별칭, 속성, UDA를 제외한 일부 차원을 선택합니다.
3. 등록정보 그리드에서 적절한 계층 유형 등록정보를 선택합니다.
 - 차원 멤버가 다음 통합 기호를 사용하는 모든 차원에 대해 **저장**을 선택합니다.
 - + ADDITION
 - ~ IGNORE(LABEL ONLY 멤버 아래에만 있음)
 - 차원 멤버가 지원되는 모든 통합 기호를 사용하거나 공식을 포함해야 하는 모든 차원에 대해 **동적**을 선택합니다.
4. 애플리케이션을 Oracle Hyperion Profitability and Cost Management 및 Oracle Essbase에 차례로 재배포합니다.

마법사를 사용하여 관리 원장 애플리케이션 검증 및 배포

Oracle Hyperion Profitability and Cost Management 애플리케이션을 생성한 후에는 애플리케이션을 검증하여 Profitability and Cost Management에 배포해야 합니다.

애플리케이션 마법사를 사용하는 경우에는 애플리케이션 생성을 완료하기 전에 애플리케이션을 검증하고 오류를 해결합니다. 필요한 경우 마법사의 검증 및 배포 옵션을 건너뛰고 나중에 [애플리케이션 라이브러리]의 배포 옵션을 사용할 수 있습니다.

마법사를 사용하여 애플리케이션을 검증 및 배포하려면 다음을 수행합니다.

1. 애플리케이션 마법사의 **애플리케이션 설정** 화면에서 **검증**을 누릅니다.
모든 오류가 그리드에 표시되어 오류 유형과 오류 메시지를 나타냅니다.
2. **선택 사항**: 검증 후 애플리케이션을 배포하려면 **완료 시 배포**를 선택합니다.
3. 검증 오류가 발생하는 경우 애플리케이션을 배포하기 전에 모든 문제를 수정합니다.
4. **완료**를 누릅니다.
"완료 시 배포"를 선택한 경우 [배포] 대화상자가 표시됩니다.

 주:

공유 차원을 변경하는 경우 해당 공유 차원을 사용하는 모든 애플리케이션이 영향을 받습니다. 변경사항을 적용하려면 애플리케이션을 다시 배포해야 합니다.

5. 애플리케이션을 배포합니다.

IV부

상세 Profitability 작업

참조:

- 상세 Profitability 제품 아키텍처 이해
- 상세 Profitability 데이터베이스 작업
- 제품 스키마
- 상세 Profitability에 대한 모델 데이터 스키마
- 상세 Profitability and Cost Management 차원 및 메타데이터 작업
- 상세 Profitability 모델 계산
- 상세 Profitability 스테이지 테이블 임포트
- 상세 Profitability에 대한 모델 정의 데이터 익스포트

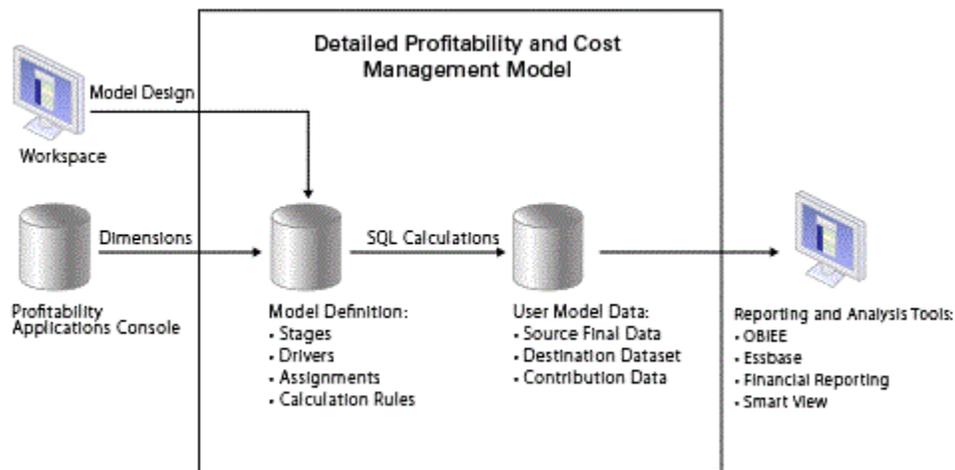
7

상세 Profitability 제품 아키텍처 이해

Oracle Hyperion Enterprise Performance Management Workspace를 통해 액세스되는 상세 Profitability는 관계형 데이터베이스의 데이터를 사용하는 분석 애플리케이션입니다. 이 애플리케이션을 통해 비즈니스 사용자는 수익성 및 비용 관리용으로 비즈니스를 모델링하고, 해당 모델 정보를 사용하여 모델 정보, 기존 데이터 및 애플리케이션에서 계산된 결과가 저장되는 관계형 데이터베이스를 생성할 수 있습니다. 입력 및 계산된 상세 Profitability 모델에 대한 데이터는 관계형 데이터베이스에 포함됩니다.

Oracle Hyperion Profitability and Cost Management는 Oracle Hyperion Shared Services를 활용하여 애플리케이션 메타데이터 및 보안을 중앙 집중식으로 관리합니다.

그림 7-1 상세 Profitability 아키텍처



애플리케이션 관리자는 Profitability 애플리케이션 콘솔을 사용하여 Profitability and Cost Management 차원을 생성합니다. [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#)을 참조하십시오. 사용자 액세스는 Shared Services를 사용하여 중앙에서 관리됩니다. 차원 메타데이터가 준비되면 Profitability and Cost Management 애플리케이션이나 모델로 배포됩니다.

이 모델 디자인에는 데이터베이스 내의 계산 수행에 필요한 SQL 문을 생성하는 데 필요한 정보가 포함됩니다. 각 모델에서 데이터베이스 내의 다음 스키마에 액세스해야 합니다.

- 모델 디자인을 저장하기 위한 스키마로, 제품 스키마라고 합니다. [제품 스키마](#)를 참조하십시오.
- 기존 데이터 및 모델 계산 결과를 저장하기 위한 스키마(모델 데이터 스키마라고도 함) [상세 Profitability에 대한 모델 데이터 스키마](#)를 참조하십시오.

8

상세 Profitability 데이터베이스 작업

상세 Profitability 애플리케이션에 대한 차원 데이터와 모델 정의는 관리 원장 애플리케이션의 차원 데이터와 모델 정의를 저장하는 데 사용되는 것과 동일한 관계형 데이터베이스 스키마에 저장됩니다. 이 스키마는 제품 스키마라고도 하며 Profitability and Cost Management를 설치할 때 생성됩니다. Profitability 애플리케이션 콘솔에서 애플리케이션을 배포하면 제품 스키마의 차원 데이터가 채워집니다. 모델 정의는 모델을 빌드할 때 이 스키마에 저장됩니다.

상세 Profitability 애플리케이션의 경우 할당 수행 시 사용되는 비즈니스 데이터도 관리 원장 애플리케이션처럼 Oracle Essbase에 저장되지 않고 관계형 데이터베이스에 저장됩니다. 이 데이터는 모델 데이터 스키마라는 별도의 데이터베이스 스키마에 있습니다. 모델 데이터 스키마는 사용자 정의되며 제품 스키마와 동일한 데이터베이스 인스턴스에 있어야 합니다. Oracle 및 Microsoft SQL Server 데이터베이스만 지원됩니다.

애플리케이션을 배포하면 차원과 멤버가 열에 매핑되며 Oracle Hyperion Profitability and Cost Management의 애플리케이션 테이블에 등록됩니다. 상세 Profitability 차원에 대한 자세한 내용은 [상세 Profitability and Cost Management 차원 및 메타데이터 작업을 참조하십시오.](#)

표 8-1 상세 Profitability and Cost Management 데이터베이스 스키마

스키마 유형	저장된 정보 유형
모델 데이터 스키마	<ul style="list-style-type: none">• 차원 계층• 비용 데이터• 수익 데이터• 드라이버 데이터• 수익 객체 데이터• 환율
제품 스키마	<ul style="list-style-type: none">• 차원• 단계 정의• POV 정의• 드라이버 정의• 드라이버 선택• 지정 규칙• 지정 규칙 선택• 모델 환경설정• 테이블 등록• 계산 규칙

9

제품 스키마

상세 Profitability 애플리케이션은 Oracle Hyperion Profitability and Cost Management 시스템 테이블을 저장하고 설치 시 생성되는 동일한 제품 스키마를 공유합니다.

Profitability and Cost Management UI 및 상세 Profitability 계산 엔진은 이 스키마에 연결합니다. 모델을 개발하고 계산하는 동안 보고 뷰가 제품 스키마에 생성됩니다.

이 스키마는 대체로 애플리케이션 관리자와 데이터베이스 관리자만 액세스할 수 있습니다. 모델 데이터 스키마에 대한 보고 뷰의 권한은 자동으로 부여되며 해당 동의어가 모델 데이터 스키마에 자동으로 생성됩니다. 동의어를 사용하면 제품 스키마에 대한 권한을 부여하지 않고도 스키마에 대한 액세스 권한을 가진 사용자에게 권한이 노출됩니다.

설치 후에 상세 Profitability에서는 다음 시스템 권한(또는 권한이 포함된 역할)이 제품 스키마에 부여되어야 합니다.

- Oracle 데이터베이스의 경우 상세 Profitability에서 다음 시스템 권한(또는 권한이 포함된 역할)이 제품 스키마에 부여되어야 합니다.
 - CREATE TABLE
 - CREATE VIEW
 - CREATE ANY SYNONYM
 - DROP ANY SYNONYM
- Microsoft SQL Server의 경우 상세 Profitability에서 다음 시스템 권한(또는 권한이 포함된 역할)이 제품 스키마에 부여되어야 합니다.
 - CREATE TABLE
 - CREATE VIEW
 - 제품 스키마에 대한 모델 데이터 스키마의 ALTER 또는 CONTROL(제품 스키마가 모델 데이터 스키마의 객체를 변경할 수 있음)

예외: 제품 스키마가 "dbo" 또는 "sa" 같은 관리자 사용자인 경우 위의 권한이 이미 있습니다. 이 경우 다음 권한을 부여할 필요가 없습니다.

주:

Profitability and Cost Management 제품 스키마와 연계된 사용자에게 SYSADMIN 역할을 부여하지 마십시오. Profitability and Cost Management 제품 스키마로 설정되어 있어야 하는 해당 사용자의 기본 스키마가 변경됩니다.

CREATE TABLE 및 CREATE VIEW 권한이 부여되면 모델을 개발하고 계산하는 동안 Profitability and Cost Management에서 제품 스키마에 시스템 생성 테이블과 보고 뷰를 생성할 수 있습니다.

CREATE ANY SYNONYM 및 DROP ANY SYNONYM 권한이 부여되면 Profitability and Cost Management에서 제품 스키마에 생성된 시스템 생성 보고 뷰에 대한 모델 데이터 스키마의 동의어를 관리할 수 있습니다.

 주:

이러한 시스템 권한 부여 시 조직의 보안이 염려되는 경우 권한을 취소하고 모델 데이터 스키마에 시스템 생성 보고 뷰에 대한 동의어를 수동으로 생성할 수 있습니다. 이 옵션을 사용하면 모델 데이터 스키마에 대한 액세스 권한만 있는 비즈니스 데이터 사용자에게 권한을 노출시킬 수 있습니다.

10

상세 Profitability에 대한 모델 데이터 스키마

참조:

- [상세 Profitability에 대한 모델 데이터 스키마 정보](#)
모델 데이터 스키마는 데이터베이스 관리자가 상세 Profitability 애플리케이션을 지원하기 위해 생성한 스키마 또는 기존 스키마일 수 있습니다.
- [모델 데이터 스키마 테이블에서 제품 스키마에 권한 부여](#)
모델 데이터 등록 내에서 모델 데이터 스키마의 데이터베이스 객체를 보려면 먼저 Profitability and Cost Management 제품 스키마에 테이블 권한을 부여해야 합니다.
- [모델 데이터 스키마 테이블 구조 요구사항](#)
상세 Profitability 애플리케이션에서 사용할 모델 데이터 스키마의 테이블을 등록하려면 몇 가지 요구사항을 충족해야 합니다.
- [데이터베이스 뷰 등록](#)
제품 스키마에 적절한 권한을 부여한 경우 모델 데이터 스키마에서 데이터베이스 뷰를 등록할 수도 있습니다.

상세 Profitability에 대한 모델 데이터 스키마 정보

모델 데이터 스키마는 데이터베이스 관리자가 상세 Profitability 애플리케이션을 지원하기 위해 생성한 스키마 또는 기존 스키마일 수 있습니다.

이 스키마는 상세 Profitability 애플리케이션에서 할당을 수행하는 비즈니스 데이터가 포함된 테이블과 뷰를 저장합니다. Oracle Hyperion Profitability and Cost Management는 제품 스키마에만 직접 연결하므로 애플리케이션의 테이블 또는 뷰를 보려면 데이터베이스 관리자가 각 애플리케이션의 테이블 권한을 제품 스키마에 부여해야 합니다.

모델 데이터 스키마 테이블에서 제품 스키마에 권한 부여

모델 데이터 등록 내에서 모델 데이터 스키마의 데이터베이스 객체를 보려면 먼저 Profitability and Cost Management 제품 스키마에 테이블 권한을 부여해야 합니다.

아래 표에 나와 있는 대로 각 테이블 유형에 대한 권한을 부여합니다.

표 10-1 각 테이블 유형에 부여된 권한

테이블 유형	권한
가로 또는 세로 방향 소스 스테이지 테이블	GRANT SELECT ON <table_name 또는 view_name> TO <hpm_product_schema> WITH GRANT OPTION
대상 스테이지 테이블	GRANT SELECT, UPDATE ON <table_name 또는 view_name> TO <hpm_product_schema>
모든 유형의 록업 테이블	GRANT SELECT ON <table_name 또는 view_name> TO <hpm_product_schema>

 주:

Oracle 데이터베이스의 경우 `dbms_stats`에 대한 EXECUTE 권한을 부여해야 합니다. 기본적으로 이 권한은 공용에 부여됩니다. 조직에서 통계 입력을 제한했으며 이 권한을 공용에서 취소한 경우 제품 스키마에 권한을 부여해야 합니다.

애플리케이션을 배포한 후 상세 Profitability 애플리케이션에서 사용할 테이블을 모델 데이터 스키마에 등록해야 합니다.

해당 열을 Profitability 애플리케이션 콘솔에서 정의한 차원 및 측정항목에 매핑합니다. 상세 Profitability 차원에 대한 자세한 내용은 [상세 Profitability and Cost Management 차원 및 메타데이터 작업](#)을 참조하십시오.

모델 데이터 스키마 테이블 구조 요구사항

상세 Profitability 애플리케이션에서 사용할 모델 데이터 스키마의 테이블을 등록하려면 몇 가지 요구사항을 충족해야 합니다.

- **데이터베이스 객체 이름:** 대문자여야 하며 숫자 0-9와 "_" 및 "\$" 문자를 포함할 수 있습니다. 소문자와 기타 특수 문자는 지원되지 않습니다. 소문자나 특수 문자를 사용하는 기존 데이터베이스 객체 이름이 있는 경우 뷰를 생성하여 허용되는 식별자로 테이블, 뷰 및 열 이름을 바꾼 다음 기본 테이블 대신 뷰를 등록할 수 있습니다.
- **대상 스테이지 테이블에만 해당:** "WORKING"(대문자)이라는 숫자 열이 있어야 합니다.
- **대상 스테이지 테이블에만 해당:** 테이블의 고유성을 적용하려면 PRIMARY KEY 제약 조건이 정의되어 있어야 합니다.

데이터베이스 뷰 등록

제품 스키마에 적절한 권한을 부여한 경우 모델 데이터 스키마에서 데이터베이스 뷰를 등록할 수도 있습니다.

적합한 모든 데이터베이스를 소스 스테이지 테이블 또는 록업 테이블로 등록할 수 있습니다. 간단한 단일 테이블 업데이트 가능 뷰만 대상 스테이지 테이블로 등록할 수 있습니다.

데이터베이스 뷰는 다음과 같은 여러 이유로 유용할 수 있습니다.

- 모델 데이터 스키마에 없는 테이블 액세스. 특히 록업 테이블에 유용할 수 있습니다.
- 상세 Profitability 애플리케이션에 노출되지 않도록 기본 테이블에서 행 필터링 또는 열 숨김
- 애플리케이션 내에서 사용을 간소화하기 위해 읽기 전용 테이블의 뷰 정의 내에서 복잡한 조인 숨김

모델을 개발하고 계산하는 동안 Oracle Hyperion Profitability and Cost Management는 제품 스키마에 시스템 생성 보고 뷰를 생성합니다. Profitability and Cost Management는 모델 데이터 스키마에 대한 SELECT 권한도 부여하며 모델 데이터 스키마에 해당 동의어를 생성합니다. 이렇게 하면 사용자가 대체로 관리자만 액세스할 수 있도록 예약되는 제품 스키마에 대한 권한이 없어도 해당 스키마에 액세스할 수 있습니다. 관리자가 Profitability and Cost Management에 필요한 모든 테이블 등록 태스크와 사용된 임의의 보고 도구의 물리적 모델 설정을 완료하는 경우가 아니면 모델 빌더와 보고서 개발자에게 이 스키마에 대한 액세스 권한이 부여되어야 합니다.

 주:

Oracle에서 데이터베이스 사용자와 데이터베이스 스키마는 같은 항목이지만 Microsoft SQL Server에서는 두 개의 개별 항목입니다. Microsoft SQL Server를 사용하는 경우 모델 데이터 스키마와 동일한 이름의 사용자를 생성하고 모델 데이터 스키마 액세스 권한을 이 사용자에게 부여해야 합니다. Profitability and Cost Management에서 만든 시스템 생성 뷰에 대한 권한을 부여하는 경우 모델 데이터 사용자가 참조됩니다. 생성된 동의어는 모델 데이터 스키마에 저장됩니다.

SQL Server의 경우 Profitability and Cost Management 제품 및 설명서의 모델 데이터 스키마 참조는 스키마, 사용자 또는 둘 다를 참조합니다.

 주:

SQL Server: 반올림 오류를 방지하려면 대상 스테이지 테이블 및 록업 테이블에 소수 자릿수가 4 이상인 측정항목 열을 생성해야 합니다.

 주의:

데이터나 아티팩트를 임포트하기 전에 모델 데이터 스키마의 백업을 생성해야 합니다. *Oracle Enterprise Performance Management System 백업 및 복구 가이드*를 참조하십시오.

상세 Profitability and Cost Management 차원 및 메타데이터 작업

참조:

- 상세 Profitability 차원 및 메타데이터 정보**
 상세 Profitability를 통해 기존 관계형 데이터베이스를 상세 Profitability 애플리케이션의 시작점으로 사용할 수 있습니다.
- 상세 Profitability 차원 유형**
 상세 Oracle Hyperion Profitability and Cost Management에는 여러 유형의 사용가능한 차원이 있습니다.
- 별칭 차원**
 별칭은 대체 이름, 설명, 언어 또는 차원 정의를 돕는 기타 항목입니다.
- 비즈니스 차원**
 비즈니스 차원은 모델의 각 스테이지 내에서 제품, 고객, 지역, 직원 등의 비즈니스 특정 객체를 설명합니다.
- POV 차원**
 POV 차원은 연도, 시나리오 또는 기간과 같은 모델의 특정 POV나 버전을 나타냅니다. 조직의 요구사항에 맞게 차원을 사용자정의할 수 있습니다.
- 속성 차원**
 속성 차원은 비즈니스 차원과 연결된 특수한 유형의 차원으로, 다른 연결된 차원의 멤버를 분류하는 데 사용할 수 있는 멤버를 포함합니다.
- 차원 정렬 순서 및 밀도 설정**
 차원 정렬 순서 등록정보는 Oracle Hyperion Profitability and Cost Management로 생성된 Oracle Essbase 아웃라인의 차원 순서를 제어합니다.
- 차원 정렬 순서 설정**
 모델에 있는 모든 차원의 처리 순서는 차원 로드 파일에 있는 차원 정렬 순서 등록정보를 사용하여 차원 레벨에서 설정해야 합니다.

상세 Profitability 차원 및 메타데이터 정보

상세 Profitability를 통해 기존 관계형 데이터베이스를 상세 Profitability 애플리케이션의 시작점으로 사용할 수 있습니다.

차원 및 메타데이터는 Oracle Hyperion Profitability and Cost Management 관리자가 Profitability 애플리케이션 콘솔을 사용하여 생성하고 유지관리합니다. 공통 차원 및 멤버를 사용하면 Profitability and Cost Management에서 손쉽게 Oracle Hyperion Planning 같은 기타 제품의 공통 데이터를 사용 및 전송하여 시간과 노력을 절약하고 정확성을 높일 수 있습니다. 차원 및 메타데이터를 생성하고 유지관리하는 방법에 대한 자세한 내용은 [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#)을 참조하십시오.

Profitability and Cost Management는 Profitability 애플리케이션 콘솔에서 생성된 차원 및 멤버를 사용하여 비즈니스 모델의 여러 구조 요소를 나타냅니다. 상세 Profitability 애플리케이션을 생성하는 동안 다음 차원이 선택됩니다.

- 측정항목 상세(필수): 스테이지 밸런싱 및 기타 확인 활동을 지원하는 데 필요한 할당 멤버가 포함되어 있는 예약된 차원입니다. 모델 및 시나리오의 구조를 제공합니다. 이 차원에는 드라이버 측정항목이 포함되지 않습니다.
- 하나 이상의 POV 차원(필수): 기간, 시나리오 및 버전입니다.
- 하나 이상의 비즈니스 차원(필수): 모델의 부서, 일반 원장 계정, 활동, 고객 또는 제품과 같은 비즈니스 특정 요소를 반영합니다.
- 별칭 차원(선택사항)
- 속성 차원(선택사항)

각 차원에 대해 차원 유형 및 차원 이름을 모두 지정해야 합니다.

- 차원 유형은 선택한 차원에 대해 사전 정의된 기능을 사용할 수 있도록 하는 차원 등록정보입니다.
- 차원 이름은 조직이나 비즈니스와 연관된 차원의 내용을 식별하기 위해 지정됩니다. 예를 들어, 계정 유형의 차원에는 총계정원장, 계정 차트 등의 차원 이름을 지정할 수 있습니다. 차원 이름에는 차원 유형이 반영된 경우라도 해당 차원 유형을 반영할 필요가 없습니다. 이름 지정 제한에 대한 자세한 내용은 [Essbase 이름 지정 규칙](#).

▲ 주의:

생성할 수 있는 차원 및 멤버 수에 물리적인 제한은 없지만 큰 차원 구조에서 성능 이슈가 발생할 수 있습니다. 애플리케이션이 배포된 후 추가로 차원이나 멤버를 생성하려면 새 객체를 사용하기 위해 모델 데이터 등록에 재등록해야 할 수 있습니다.

차원 및 멤버를 생성하고 유지관리하는 방법에 대한 자세한 지침은 [Profitability 애플리케이션 콘솔로 애플리케이션 생성](#)을 참조하십시오.

상세 Profitability 차원 유형

상세 Oracle Hyperion Profitability and Cost Management에는 여러 유형의 사용가능한 차원이 있습니다.

- 측정항목 상세는 단계 밸런싱 및 기타 검증 작업을 지원하는 데 필요한 할당 멤버가 포함된 제한된 차원입니다.
- 별칭 차원은 대체 이름, 설명, 언어 또는 차원 정의에 도움이 되는 기타 항목을 지정하는 데 사용됩니다. 이 차원은 선택사항으로, 모델에서 별칭을 사용하려는 경우에만 필요합니다.

✎ 주:

중복 멤버 이름 또는 별칭은 허용되지 않습니다.

- 비즈니스 차원은 부서, 일반 원장 계정, 활동, 고객 또는 제품 같은 모델 내의 비즈니스 요소를 설명하기 위해 생성됩니다. 애플리케이션에 대한 비즈니스 차원이 하나 이상 필요합니다.

- 속성 차원은 차원 멤버의 속성 및 품질을 기준으로 분석을 실행합니다. 속성은 제품의 크기 및 색상과 같이 데이터의 특성을 설명합니다.

 주:

차원 아웃라인을 정의하는 경우 이름 지정에 사용할 수 없는 제한 문자가 있습니다. *Oracle Essbase Database Administrator's Guide*에서 Oracle Essbase 이름 지정 규칙을 검토하는 것이 좋습니다.

별칭 차원

별칭은 대체 이름, 설명, 언어 또는 차원 정의를 돕는 기타 항목입니다.

예를 들어 시스템의 고객 번호를 참조할 수 있지만 화면에 해당 회사 이름을 표시하는 별칭을 지정하여 보다 쉽게 해당 클라이언트를 식별할 수 있습니다. 계정, 통화, 엔티티, 시나리오, 기간, 버전, 연도 및 사용자 정의 차원 번호에 하나 이상의 별칭을 지정할 수 있습니다.

 주:

동일한 차원 내에서는 중복 멤버 이름 또는 별칭을 사용할 수 없습니다.

Oracle Essbase에서 특정 차원이 복제된 경우 별칭이 복제될 수 있습니다.

별칭을 보려면 다음을 수행합니다.

- 애플리케이션에서 공통 멤버 선택기를 사용하는 화면(예: 드라이버 선택, 지정, 데이터 입력 또는 추적 지정)을 선택합니다. [공통 멤버 선택기]가 있는 해당 화면이 표시되고 사용 가능한 모든 멤버를 보여 줍니다.
- [공통 멤버 선택기]에서 [컨텍스트 메뉴] 버튼()을 누르고 **별칭 표시**를 선택합니다. 별칭이 멤버 목록에 표시됩니다.

 주:

[컨텍스트 메뉴]에서 [별칭 표시]를 선택했는데 별칭이 지정되지 않은 경우에는 멤버 이름이 대괄호 안에 표시됩니다. 예를 들어 멤버 이름 Product는 [Product]로 표시됩니다.

비즈니스 차원

비즈니스 차원은 모델의 각 스테이지 내에서 제품, 고객, 지역, 직원 등의 비즈니스 특정 객체를 설명합니다.

비즈니스 차원은 다음 차원 유형의 일부 또는 전체를 사용할 수 있으며 하나 이상의 단계나 모델에 적용될 수 있습니다.

- 일반

- 계정
- 엔티티
- 국가

Oracle Essbase 아웃라인이 배포되면 비즈니스 차원이 Oracle Hyperion Profitability and Cost Management 애플리케이션에서 유형 없이 기본 또는 일반 차원으로 생성됩니다. 이 기능으로 Profitability and Cost Management에서 Oracle Hyperion Planning 등의 다른 애플리케이션에 대해 정의된 차원 멤버 및 계층을 재사용할 수 있습니다.

주:

이 차원 유형은 합산 저장영역 아웃라인에는 적용되지 않습니다.

비즈니스 차원 생성 시 다음 요구사항이 적용됩니다.

- 차원의 Gen1 멤버에 대해 다음 등록정보를 LABEL_ONLY로 설정해야 합니다.
 - DataStorage(BSO)
 - DataStorage(ASO)
- Gen1 차원 이름 아래의 첫번째 Gen2 1차 하위는 일반적으로 All 멤버로 설정됩니다. 예를 들어 부서 차원에 대해 AllDepartments입니다.

기본 계층은 첫 번째 Gen2 1차 하위 구성요소 아래에 호스트됩니다. 첫 번째 Gen2 계층만 할당 모델링에 사용되며 이 계층에는 공유 멤버를 포함할 수 없습니다.
- 추가 Gen2 멤버는 대체 계층을 호스트할 수 있지만 이러한 계층이 할당 모델링에 사용되지 않습니다. 차원에서 대체 계층을 호스트하려는 경우 차원의 HierarchyType을 "사용"으로 설정하고 첫번째 Gen2 멤버의 HierarchyType을 "저장"으로 설정하고 공유 멤버가 포함된 대체 계층이 있는 Gen2 멤버를 "동적"으로 설정합니다.

이러한 대체 계층은 Profitability and Cost Management 모델링 화면에 표시되지 않으며 Essbase에서만 볼 수 있습니다.
- NoMember 멤버가 필요합니다. 이 계층에서 마지막 Gen2 1차 하위 구성요소는 항상 통합이 IGNORE (~)로 설정된 NoMember여야 합니다.

POV 차원

POV 차원은 연도, 시나리오 또는 기간과 같은 모델의 특정 POV나 버전을 나타냅니다. 조직의 요구사항에 맞게 차원을 사용자정의할 수 있습니다.

예를 들어 POV는 분기, 월, 계절별 그룹 등으로 구성될 수 있습니다.

각 모델에 대해 POV 차원이 하나 이상 필요하며, 최대 4개의 POV 차원을 생성할 수 있습니다.

버전 차원도 사용할 수 있으며, 이 차원은 모델의 다른 인스턴스를 생성하는 데 사용됩니다. 이 버전을 수정하여 여러 가지 전략이나 비즈니스 옵션으로 "가정" 시나리오를 재생할 수 있습니다. 버전 수정을 통해 기능을 구현하고 결과를 비교하여 최상의 작업 방향을 결정할 수 있습니다.

속성 차원

속성 차원은 비즈니스 차원과 연결된 특수한 유형의 차원으로, 다른 연결된 차원의 멤버를 분류하는 데 사용할 수 있는 멤버를 포함합니다.

속성 차원은 제품의 크기 및 색상과 같은 데이터의 특성을 설명합니다.

이러한 속성을 사용하여 차원 멤버의 속성 또는 품질을 기준으로 데이터를 분석할 수 있습니다. 또한 속성 차원은 지정 규칙 생성 시 대상 교차를 필터링하는 데 사용됩니다.



주:

이름 지정 제한에 대한 자세한 내용은 [Essbase 이름 지정 규칙](#)을 참조하십시오.

속성 차원 유형은 다음 두 가지가 있습니다.

- 속성 차원:
 - 부울, 날짜, 숫자 및 텍스트 같은 여러 구조를 사용하여 속성을 생성할 수 있습니다.
 - 속성에는 계층이 있고, 합산 값을 위해 계층을 롤업할 수 있습니다.
 - 지정된 속성 차원의 속성 한 개만 한 멤버에 연결할 수 있습니다.
- UDA(사용자 정의 속성) 차원
 - 속성은 텍스트로만 생성할 수 있습니다.
 - UDA에는 계층이 없으며 보고서에서 쉽게 사용하여 합계를 생성할 수 없습니다.
 - 한 멤버에 여러 UDA를 연결할 수 있습니다.

각 유형의 속성 차원은 모델 및 보고 요구에 따라 다른 이점을 제공합니다. 속성 차원 작업에 대한 자세한 내용은 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

이러한 속성을 사용하여 차원 멤버의 속성 또는 품질을 기준으로 데이터를 분석할 수 있습니다. 또한 속성 차원은 지정 규칙 생성 시 대상 교차를 필터링하는 데 사용됩니다.

이름 지정 제한에 대한 자세한 내용은 [Essbase 이름 지정 규칙](#)을 참조하십시오.

차원 정렬 순서 및 밀도 설정

차원 정렬 순서 등록정보는 Oracle Hyperion Profitability and Cost Management로 생성된 Oracle Essbase 아웃라인의 차원 순서를 제어합니다.

차원 정렬 순서는 모델 내부에서 별칭 및 UDA를 제외한 모든 차원에 대해 설정되어야 합니다.



주의:

차원 정렬 순서를 비워두면 검증이 실패합니다.

차원 정렬 순서 등록정보는 차원 로드 파일에서 설정합니다. 지침은 [차원 정렬 순서 설정](#)을 참조하십시오.

차원 정렬 순서 권장 사항

다음 권장 사항을 사용하여 차원 정렬을 설정하는 것이 좋습니다.

- 별칭 및 UDA를 제외한 모델의 모든 차원에 대해 차원 정렬 순서를 설정해야 합니다.

주:

별칭 및 UDA 차원은 Oracle Hyperion Profitability and Cost Management 및 Oracle Essbase의 차원으로 존재하지 않으므로 차원 정렬 순서에 대해 무시됩니다.

- 차원 정렬 순서는 순차적이고 고유하며 1보다 크거나 같아야 합니다.
- 측정항목 차원의 기본값은 1로 설정됩니다.
- AllocationType 차원의 기본값은 2로 설정됩니다.
- 비즈니스 및 POV 차원은 3 이상으로 설정되어야 합니다.
- 속성 차원은 항상 마지막 차원으로 정렬되어야 합니다. 예를 들어 연속적인 12개 차원 중 속성 차원이 4개인 경우 속성 차원은 9, 10, 11 및 12로 설정되어야 합니다.

차원 정렬 순서 설정

모델에 있는 모든 차원의 처리 순서는 차원 로드 파일에 있는 차원 정렬 순서 등록정보를 사용하여 차원 레벨에서 설정해야 합니다.

차원 정렬 순서 제한 사항을 충족해야 하며, 그렇지 않으면 모델 검증에 실패합니다. 제한 사항 전체 목록은 [차원 정렬 순서 권장 사항](#)을 참조하십시오.

차원 정렬 순서가 포함된 로드 파일을 생성하려면 다음을 수행합니다.

1. 외부 프로그램 또는 ETL 프로세스에서 이와 유사한 열을 사용하여 로드 파일을 생성합니다. 차원 정렬 순서에 대한 열이 포함되도록 해야 합니다.

차원의 루트 멤버에 대한 정렬 순서를 입력합니다. 이 예에서는 *계*경에 정렬 순서 9가 있습니다.

Regular	Parent	Storage Type	Hierarchy Type	Comment	ASO Data Storage	BSO Data Storage	Two Pass Calculation	Dimension Formula	Consolidation Type	UDA	Dimension Solve Order	Dimension Sort Order	Solve Order	Member Alias:English	Member 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 Gc 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
"MAT51Materials		SPARSE	STORED		StoreData	StoreData	N		+				0	Frames	Frames
"MAT52Materials		SPARSE	STORED		StoreData	StoreData	N		+				0	Wheels	Wheels
"MAT53Materials		SPARSE	STORED		StoreData	StoreData	N		+				0	Gearsets	Gearsets
"MAT54Materials		SPARSE	STORED		StoreData	StoreData	N		+				0	Seats	Seats
"MAT55Materials		SPARSE	STORED		StoreData	StoreData	N		+				0	Controls	Controls
"MAT56Materials		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

2. Profitability 애플리케이션 콘솔에서 작업을 선택한 후 차원 업데이트를 선택하여 차원 정렬 순서가 정의된 파일을 로드합니다.

12

상세 Profitability 모델 계산

모델을 검증한 후 모델을 계산할 수 있습니다.

[계산 관리] 화면에서 상세 Profitability 사용자는 [사용자정의 스크립트] 또는 특정 데이터 POV를 비롯한 처리 옵션을 선택합니다.

기본 계산 함수 외에도 관리 사용자는 추가 작업과 프로세스 유형을 볼 수 있습니다.

▲ 주의:

이전 선택사항이었던 차원 계층 보고 테이블의 상세 Profitability 계산 엔진에 대한 새로운 종속성이 있습니다. 이러한 테이블은 Oracle Hyperion Profitability and Cost Management 내 [계산], [데이터베이스 관리] 메뉴, [보고 테이블 및 뷰] 탭에서 배포됩니다. 상세 Profitability 애플리케이션을 재배포할 때마다 이러한 테이블을 재배포해야 합니다. 지침은 *Oracle Hyperion Profitability and Cost Management 사용자 가이드*의 "상세 Profitability 보고 뷰 배포"를 참조하십시오.

사전 및 사후 계산 사용자정의 스크립트

사용자정의 사전 계산 및 사후 계산 스크립트를 생성하여 애플리케이션 계산 전이나 후에 모델 또는 단계 데이터를 조작하는 작업을 수행할 수 있습니다.

사용자정의 스크립트는 다음과 같이 실행됩니다.

- 사전 계산 SQL 스크립트는 모델 계산 전에 실행됩니다.
- 사후 계산 SQL 스크립트는 모델 계산 후에 실행됩니다.

사용자정의 스크립트는 모델 계산 전에 할당을 준비하거나 모델 계산 후 보고할 결과를 개선하기 위해 실행됩니다.

SQL Developer 또는 타사 SQL 도구를 사용하여 간단한 일련의 SQL 문을 구분자로 구분하여 생성합니다. 사용자정의 스크립트는 제품 스키마 내의 데이터베이스에 있는 HPM_SQL_SCRIPT 테이블의 SCRIPT 열에 저장됩니다.

계산 매개변수를 생성할 때 사용자정의 스크립트를 선택합니다. **태스크 영역**에서 **계산**, **계산 관리** 순으로 선택한 다음 필요한 사전 계산 또는 사후 계산 스크립트를 선택합니다. 자세한 지침은 *Oracle Hyperion Profitability and Cost Management 사용자 가이드*를 참조하십시오.

다음 섹션을 참조하십시오.

- [HPM_SQL_SCRIPT](#)
- [사용자정의 스크립트 생성](#)

HPM_SQL_SCRIPT

이 테이블에는 사용자정의 사전 및 사후 계산 스크립트가 저장됩니다. 테이블의 각 행에는 단계 모델 중 실행할 수 있는 SQL 문이 하나 이상 포함됩니다.



주:

HPM_SQL_SCRIPT에 스크립트를 직접 입력하지 마십시오. [사용자정의 스크립트 생성](#)을 참조하십시오.

표 12-1 HPM_SQL_SCRIPT

열	데이터 유형	Null 허용 가능	설명
NAME	Varchar2 (80)	아니요	사용자정의 스크립트의 이름
APPLICATION_NAME	Varchar2 (8)	아니요	이 사용자정의 스크립트가 생성된 애플리케이션의 이름
MODEL_POV_NAME	Varchar2 (80)	예	선택한 POV는 "연도": "2012", "기간": "1월", "시나리오": "실제" 형식으로 표시됩니다. 참고: 이 열이 NULL로 설정된 경우 임의 모델 POV에 스크립트를 사용할 수 있습니다. 특정 POV가 선택된 경우 해당 POV에만 스크립트를 사용할 수 있습니다.
SCRIPT_TYPE	Varchar2 (80)	예	적합한 스크립트 유형을 선택합니다. <ul style="list-style-type: none"> • PRE(사전 스크립트 표시) • POST(사후 스크립트 표시) • EITHER(사전 스크립트 또는 사후 스크립트 표시) • 스크립트 유형을 지정하지 않으면 EITHER가 가정됩니다.
DESCRIPTION	Varchar2 (255)	예	스크립트의 용도나 내용에 대한 설명 입력
SCRIPT	LONG	아니요	여기에 사용자정의 스크립트를 입력합니다. 스크립트 생성에 대한 지침은 사용자정의 스크립트 생성 을 참조하십시오.

사용자정의 스크립트 생성

메모장, Textpad, Oracle SQL Developer 등의 텍스트 편집기에서 사용자정의 스크립트를 생성할 수 있습니다. 스크립트는 간단한 SQL 문처럼 짧을 수도 있지만 더 긴 경우 문 사이에 구분자가 필요합니다. 필요한 구분자는 별도 라인에 있는 슬래시 문자("/")입니다. 이 스크립트에서 저장 프로시저를 실행할 수 있습니다.

[HPM_SQL_SCRIPT](#)에 표시된 대로 스크립트가 HPM_SQL_SCRIPT 테이블의 SCRIPT 열에 붙여넣어지거나 로드됩니다. HPM_SQL_SCRIPT 테이블에 스크립트를 직접 입력하지 마십시오.

▲ 주의:

HPM 제품 스키마에 대한 비밀번호가 있는 시스템 관리자만 사전 및 사후 스크립트를 생성할 수 있습니다.

사용자정의 스크립트를 생성하고 로드하려면 다음을 수행합니다.

1. 메모장, Textpad, Oracle SQL Developer 등의 텍스트 편집기에서 사용자정의 스크립트를 생성합니다.
2. Oracle SQL Developer를 사용하여 HPM_SQL_SCRIPT 테이블을 엽니다. 이 테이블은 HPM 제품 스키마에서 제품 테이블과 동일한 위치에 있어야 합니다.
3. 새 행을 삽입합니다.
4. 스크립트 유형을 비롯하여 필요한 값을 테이블에 입력합니다. [HPM_SQL_SCRIPT](#)을 참조하십시오.
5. 스크립트를 다음과 같이 SCRIPT 열로 복사합니다.
 - SQL Developer를 사용하는 Oracle의 경우 텍스트 편집기에서 스크립트를 복사하여 붙여넣거나 Update 문을 사용하여 SCRIPT 열을 채웁니다. 다음은 Oracle용 SQL 스크립트의 예입니다.

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

- SQL Studio를 사용하는 Microsoft SQL Server의 경우 Update 문을 사용하여 SCRIPT 열을 채웁니다. 다음은 SQL Server용 SQL 스크립트의 예입니다.

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

▲ 주의:

동일한 스크립트에 SQL 문이 여러 개 있는 경우 각 문 사이에 구분자 ";"를 포함해야 합니다. ";" 문자는 각 문의 끝에서 별도의 라인에 있어야 합니다. 최종 ";" 뒤에는 줄바꿈 문자가 옵니다.

ODBC 데이터 소스를 생성하여 데이터 전송 활성화

[계산 관리] 탭의 [데이터 전송] 옵션에는 Oracle Hyperion Profitability and Cost Management 인스턴스를 지원하는 Oracle 또는 Microsoft SQL Server 데이터베이스의 제품 스키마에 연결하는 **PROFITABILITY_DS**라는 ODBC 데이터 소스가 필요합니다.

이 ODBC 데이터 소스를 생성하려면 다음을 수행합니다.

1. 애플리케이션 서버에서 ODBC 관리 유틸리티를 엽니다. 명령행에서 `odbcad32` 명령을 실행합니다.
2. 시스템 DSN 탭을 선택하고 **추가...**를 누릅니다.
3. 적절한 ODBC 드라이버를 선택하고 **다음**을 누릅니다.
4. 데이터 소스 이름에 **PROFITABILITY_DS**를 입력하고 제품 스키마에 연결하기 위한 연결 세부정보를 채웁니다.

고급 계산 옵션

[계산 관리] 화면에는 기본 사용자가 사용할 수 없는 [드라이버 작업 유형] 및 [기타 프로세스 유형] 탭이 관리자를 위해 표시됩니다.

- [드라이버 작업 유형](#)
- [기타 프로세스 유형](#)

▲ 주의:

속려된 데이터베이스 관리자만 이러한 옵션을 사용하는 것이 좋습니다. 임의 유형을 [드라이버 작업 유형]이나 [기타 프로세스 유형]으로 수정하면 애플리케이션에 심각한 영향을 줄 수 있으며 모델 또는 데이터가 손상될 수도 있습니다.

드라이버 작업 유형

[계산 관리] 화면의 [드라이버 작업 유형] 탭에는 상세 Profitability의 현재 드라이버 작업 유형이 표시됩니다.

▲ 주의:

이 탭에서 새 드라이버 작업 유형을 생성하지 마십시오. 숙련된 데이터베이스 관리자만 이 탭에 액세스해야 합니다. 변경 시 애플리케이션에 심각한 영향을 줄 수 있으며 모델 또는 데이터가 손상될 수도 있습니다.

[드라이버 작업 유형]은 성능 이슈나 고유한 드라이버 이슈를 해결하기 위해 지원되는 드라이버 작업을 확장하는 관리 기능입니다. 이 기능을 사용하려면 고급 SQL 지식, 데이터베이스 관리에 대한 전문적 이해 및 Oracle Hyperion Profitability and Cost Management 상세 Profitability 애플리케이션의 문서화되지 않은 기능에 대한 전문적 이해가 필요합니다. 고객 서비스 요청에 대한 응답으로 Oracle에서 요청되지 않은 경우 이 기능을 사용하지 않는 것이 좋습니다.

기타 프로세스 유형

이 상세 Profitability 릴리스에는 다음 두 가지 대체 계산 프로세스 유형을 사용할 수 있습니다.

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

시스템이 Oracle Database 10g로 구성되어 있는 경우 [기타 프로세스 유형] 탭을 사용하여 데이터베이스에 적합한 계산 프로세스를 선택합니다.

▲ 주의:

숙련된 데이터베이스 관리자만 [기타 프로세스 유형] 탭을 사용하는 것이 좋습니다. [기타 프로세스 유형]에서 다른 수정 작업은 하지 마십시오. 내용 변경 시 애플리케이션에 심각한 영향을 줄 수 있으며 모델 또는 데이터가 손상될 수도 있습니다.

상세 Profitability에 사용 중인 Oracle Database 버전에 따라 이 화면에서 데이터베이스를 정확한 버전으로 변경합니다.

Oracle Database 버전을 변경하려면 다음을 수행합니다.

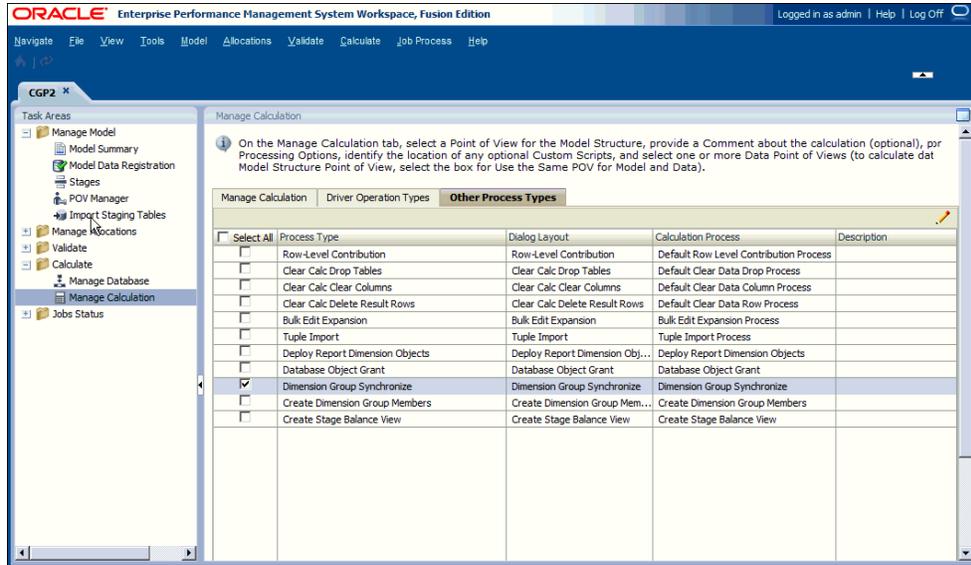
1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 애플리케이션, Profitability** 순으로 선택하고 Oracle Database 버전을 수정할 애플리케이션을 선택합니다.
2. [태스크 영역]에서 **계산, 계산 관리, 기타 프로세스 유형** 탭 순으로 선택합니다.

✎ 주:

이 탭은 관리자 사용자 역할이 지정된 경우에만 표시됩니다.

3. **차원 그룹 동기화**를 선택하고 [편집] 버튼을 누릅니다. 

[기타 프로세스 유형 정의] 대화상자가 표시됩니다.



4. [계산 프로세스]에서 사용 중인 Oracle Database에 적합한 옵션을 선택합니다.
 - 차원 그룹 동기화(Oracle 11g)
 - 차원 그룹 동기화 10G(Oracle 10g)
5. 확인을 눌러 변경사항을 저장합니다.

13

상세 Profitability 스테이지 테이블 импорт

참조:

- [상세 Profitability 스테이징 테이블 импорт 정보](#)
관계형 데이터베이스에서 Oracle Hyperion Profitability and Cost Management로 모델 데이터를 импорт하려면 Profitability and Cost Management 제품 스키마와 별도의 데이터베이스 스키마에 스테이지 테이블 세트를 생성해야 합니다.
- [상세 Profitability에 대한 импорт 데이터베이스 테이블 생성](#)
스테이지 데이터베이스 스크립트는 Microsoft SQL Server 및 Oracle 데이터베이스에 사용할 수 있습니다
- [HPM_STGD_POV](#)
HPM_STGD_POV 테이블은 POV에 포함된 각 차원 멤버 조합의 상태를 저장합니다.
- [HPM_STGD_DRIVER](#)
HPM_STGD_DRIVER 테이블은 드라이버 유형, 표시 순서 및 공식을 비롯하여 드라이버에 대한 세부정보를 제공합니다.
- [HPM_STGD_DRIVER_SEL](#)
HPM_STGD_DRIVER_SEL 테이블은 선택한 드라이버 규칙의 POV 및 소스 단계 드라이버 차원 멤버에 대한 세부정보를 제공합니다.
- [HPM_STGD_DRIVER_EXCEP](#)
HPM_STGD_DRIVER_EXCEP 테이블은 선택한 드라이버 예외의 POV 및 소스 단계 교차에 대한 세부정보를 제공합니다.
- [HPM_STGD_ASGN_RULE_SEL](#)
HPM_STGD_ASGN_RULE_SEL 테이블은 지정 규칙 선택에 대한 세부정보를 제공합니다.
- [HPM_STGD_CALCRULE_SNGLSRC](#)
HPM_STGD_CALCRULE_SNGLSRC 테이블은 단일 소스 계산 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.
- [HPM_STGD_CALCRULE_CALCMSRS](#)
HPM_STGD_CALCRULE_CALCMSRS 테이블은 계산된 측정항목 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.
- [HPM_STGD_CALCRULE_MULTISRC](#)
HPM_STGD_CALCRULE_MULTISRC 테이블은 다중 소스 계산 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.

상세 Profitability 스테이징 테이블 импорт 정보

관계형 데이터베이스에서 Oracle Hyperion Profitability and Cost Management로 모델 데이터를 импорт하려면 Profitability and Cost Management 제품 스키마와 별도의 데이터베이스 스키마에 스테이지 테이블 세트를 생성해야 합니다.

상세 Profitability의 모델 데이터 스키마로 사용되는 것과 동일한 스키마를 다시 사용할 수 있습니다. 그런 다음 가져올 애플리케이션 아티팩트의 세부정보로 이러한 스테이지 테이블을 채웁니다.

▲ 주의:

제품 스키마를 수정하지 마십시오.

스테이지 데이터베이스 스크립트는 Microsoft SQL Server 및 Oracle 데이터베이스에 사용할 수 있습니다. 적절한 스크립트를 사용하여 새 데이터베이스 스키마에 임포트 테이블을 생성합니다.

상세 Profitability에 대한 임포트 데이터베이스 테이블 생성

스테이지 데이터베이스 스크립트는 Microsoft SQL Server 및 Oracle 데이터베이스에 사용할 수 있습니다

. create_dp_staging.sql 스크립트를 사용하여 상세 Profitability에 테이블을 생성합니다.

준비 테이블을 생성하려면 다음을 수행합니다.

1. 새 Oracle 또는 Microsoft SQL Server 데이터베이스 스키마는 제품 스키마 외부에서 생성합니다.
2. 데이터베이스 유형의 기본 위치에서 create_dp_staging.sql을 찾습니다.
 - %EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer
 - %EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle
3. create_dp_staging.sql 스크립트를 실행합니다.

HPM_STGD_POV

HPM_STGD_POV 테이블은 POV에 포함된 각 차원 멤버 조합의 상태를 저장합니다.

표 13-1 HPM_STGD_POV

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	integer(38,0)	예	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름

표 13-1 (계속) HPM_STGD_POV

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
pov_state	nvarchar (80)	varchar2(80 CHAR)	예	POV의 현재 상태는 다음과 같습니다. <ul style="list-style-type: none"> • 초안 • 게시됨 • 아카이브됨
last_upload_date	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
exception	nvarchar (255)	varchar2(255 CHAR)		이 테이블을 가져오는 중에 발생한 오류를 설명하는 메시지.
created_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 시작한 사용자의 ID
created_timestamp	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
modified_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 수정한 사용자의 ID
modified_timestamp	timestamp (6)	timestamp (6)		수정된 임포트의 날짜 및 시간

HPM_STGD_DRIVER

HPM_STGD_DRIVER 테이블은 드라이버 유형, 표시 순서 및 공식을 비롯하여 드라이버에 대한 세부정보를 제공합니다.

표 13-2 HPM_STGD_DRIVER

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	integer(38,0)	예	고유 레코드 ID
name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 드라이버의 이름
description	nvarchar (255)	varchar2(255 CHAR)		선택한 드라이버의 용도에 대한 설명
formula	nvarchar (500)	varchar2(500 CHAR)	예	사용자가 드라이버에 대해 생성한 공식. 이 공식은 SQL 구문과 수학적으로 정확한 연산을 사용하여 생성해야 합니다.

표 13-2 (계속) HPM_STGD_DRIVER

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
allow_idle_cost	nvarchar (1)	varchar2 (1 CHAR)		적절한 값을 입력하여 이 드라이버에 유틸 비용을 허용할지 여부를 결정합니다. <ul style="list-style-type: none"> • 드라이버에 유틸 비용을 허용하려면 "Y"(Yes)를 입력합니다. • 드라이버의 유틸 비용을 비활성화하려면 "N"(No)을 입력합니다. 이 옵션이 기본값입니다.
priority	integer(38,0)	integer(38,0)	예	드라이버의 계산 우선순위를 입력하여 단계 내의 할당이 지정된 순서대로 실행될 수 있도록 합니다. 우선순위가 가장 낮은 드라이버가 가장 먼저 처리됩니다. 기본적으로 우선순위는 100으로 설정되며 가장 높은 우선순위는 1입니다. 번호는 순차적이지 않아도 됩니다. 동일한 우선순위를 가진 드라이버는 특정한 순서 없이 실행됩니다. 양의 정수만 적합합니다.
volume_formula	nvarchar (4000)	varchar2(4000 CHAR)		공식에 비율 및 볼륨 공식이 둘 다 정의되어 있는 비율 기반(Rate-Based) 드라이버에만 해당됩니다.
dst_measure_member_name	nvarchar (80)	varchar2(80 CHAR)	예	대상 테이블의 측정항목 이름
tdv_measure_member_name	nvarchar (80)	varchar2(80 CHAR)		할당 공식: P 드라이버 값 / TotalDriverValue(DV/TDV)에서 분모로 사용되는 측정항목

표 13-2 (계속) HPM_STGD_DRIVER

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
operation_type	nvarchar (80)	varchar2(80 CHAR)	예	드라이버에 대해 선택된 작업 유형 <ul style="list-style-type: none"> • 비율 기반(Rate-Based) 할당 • 비율 기반(Ratio-Based) 할당 • 계산된 측정항목
last_upload_date	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
exception	nvarchar (255)	varchar2(255 CHAR)		이 테이블을 가져오는 중에 발생한 오류를 설명하는 메시지.
created_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 시작한 사용자의 ID
created_timestamp	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
modified_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 수정한 사용자의 ID
modified_timestamp	timestamp (6)	timestamp (6)		수정된 임포트의 날짜 및 시간

HPM_STGD_DRIVER_SEL

HPM_STGD_DRIVER_SEL 테이블은 선택한 드라이버 규칙의 POV 및 소스 단계 드라이버 차원 멤버에 대한 세부정보를 제공합니다.

표 13-3 HPM_STGD_DRIVER_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	integer(38,0)	예	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름 (해당되는 경우)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름 (해당되는 경우)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름 (해당되는 경우)

표 13-3 (계속) HPM_STGD_DRIVER_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 드라이버 차원 멤버의 이름
driver_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 드라이버의 이름
last_upload_date	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
import_exception	nvarchar (255)	varchar2(255 CHAR)		이 테이블을 가져오는 중에 발생한 오류를 설명하는 메시지.
created_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 시작한 사용자의 ID
created_timestamp	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
modified_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 수정한 사용자의 ID
modified_timestamp	timestamp (6)	timestamp (6)		수정된 임포트의 날짜 및 시간

HPM_STGD_DRIVER_EXCEP

HPM_STGD_DRIVER_EXCEP 테이블은 선택한 드라이버 예외의 POV 및 소스 단계 교차에 대한 세부정보를 제공합니다.

표 13-4 HPM_STGD_DRIVER_EXCEP

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	integer(38,0)	예	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름 (해당되는 경우)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름 (해당되는 경우)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름 (해당되는 경우)
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	소스 단계의 첫 번째 차원 멤버 이름

표 13-4 (계속) HPM_STGD_DRIVER_EXCEP

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 두 번째 차원 멤버 이름 (해당되는 경우)
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 세 번째 차원 멤버 이름 (해당되는 경우)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 네 번째 차원 멤버 이름 (해당되는 경우)
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 다섯 번째 차원 멤버 이름 (해당되는 경우)
driver_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 드라이버의 이름
last_upload_date	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
import_exception	nvarchar (255)	varchar2(255 CHAR)		이 테이블을 가져오는 중에 발생한 오류를 설명하는 메시지.
created_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 시작한 사용자의 ID
created_timestamp	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
modified_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 수정한 사용자의 ID
modified_timestamp	timestamp (6)	timestamp (6)		수정된 임포트의 날짜 및 시간

HPM_STGD_ASGN_RULE_SEL

HPM_STGD_ASGN_RULE_SEL 테이블은 지정 규칙 선택에 대한 세부정보를 제공합니다.

표 13-5 HPM_STGD_ASGN_RULE_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	integer(38,0)	예	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름

표 13-5 (계속) HPM_STGD_ASGN_RULE_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	소스 단계의 첫 번째 차원 멤버 이름
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 두 번째 차원 멤버 이름 (해당되는 경우)
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 세 번째 차원 멤버 이름 (해당되는 경우)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 네 번째 차원 멤버 이름 (해당되는 경우)
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)		소스 단계의 다섯 번째 차원 멤버 이름 (해당되는 경우)
rule_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 지정을 제어하는 규칙 이름. 이 규칙은 대상 데이터베이스에 있어야 합니다.
last_upload_date	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
import_exception	nvarchar (255)	varchar2(255 CHAR)		이 테이블을 가져오는 중에 발생한 오류를 설명하는 메시지.
created_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 시작한 사용자의 ID
created_timestamp	timestamp (6)	timestamp (6)		최종 임포트의 날짜 및 시간
modified_userid	nvarchar (32)	varchar2(32 CHAR)	업데이트 시 임포트 프로그램에 의해 채워짐	최종 임포트를 수정한 사용자의 ID
modified_timestamp	timestamp (6)	timestamp (6)		수정된 임포트의 날짜 및 시간

HPM_STGD_CALCRLUE_SNGLSRC

HPM_STGD_CALCRLUE_SNGLSRC 테이블은 단일 소스 계산 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.

표 13-6 HPM_STGD_CALCRLUE_SNGLSRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	NUMBER(38,0)	예	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	VARCHAR2(80 CHAR)	예	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)		계산 규칙에 대한 자세한 설명.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름
calculation_sequence	identity	NUMBER(4,0)	예	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	VARCHAR2(255)		이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	예	
first_sequence_priority	identity	NUMBER(38,0)	예	이 규칙에 대해 실행할 드라이버 범위의 첫 번째 드라이버 순서 우선순위를 지정합니다. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
last_sequence_priority	identity	NUMBER(38,0)	예	이 규칙에 대해 실행할 드라이버 범위의 최종 드라이버 순서 우선순위를 지정합니다. 이 값은 first_sequence_priority 보다 크거나 같아야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	예	계산 중에 Y가 첫 번째 순서 및 최종 순서 우선순위 필드의 값을 무시하는 경우.
last_upload_date	timestamp	timestamp		이 열은 임포트를 통해 채워지며, 임포트가 수행된 가장 최근 날짜를 표시합니다.

표 13-6 (계속) HPM_STGD_CALCRLULE_SNGLSRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
import_exception	nvarchar (255)	varchar2(255 CHAR)	.	행 임포트에 문제가 있는지 여부를 나타내는 오류 코드를 포함합니다. Null은 임포트에 성공했음을 의미합니다.
created_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 원래 생성한 사용자를 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
created_timestamp	timestamp (6)	timestamp		행이 원래 생성된 날짜/시간을 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 최종 수정한 사용자를 확인합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_timestamp	timestamp (6)	timestamp		행이 최종 변경된 날짜/시간을 확인합니다. Profitability and Cost Management에서 사용되지 않습니다.

HPM_STGD_CALCRLULE_CALCMSRS

HPM_STGD_CALCRLULE_CALCMSRS 테이블은 계산된 측정항목 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.

표 13-7 HPM_STGD_CALCRLULE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	number(38,0)	예	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	varchar2(80 CHAR)	예	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)		계산 규칙에 대한 자세한 설명.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름

표 13-7 (계속) HPM_STGD_CALCRLULE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
calculation_sequence number		number(4,0)	예	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	varchar2(255)		이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	예	
driver_name1	nvarchar (1)	varchar2(1 CHAR)	예	이 계산 규칙과 연결할 드라이버 이름.
driver_name2	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 드라이버 이름.
driver_name3	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 드라이버 이름.
driver_name4	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 드라이버 이름.
driver_name5	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 드라이버 이름.
dest_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	예	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
last_upload_date	timestamp (6)	timestamp		이 열은 임포트를 통해 채워지며, 임포트가 수행된 가장 최근 날짜를 표시합니다.
import_exception	nvarchar (255)	varchar2(255 CHAR)		행 임포트에 문제가 있는지 여부를 나타내는 오류 코드를 포함합니다. Null은 임포트에 성공했음을 의미합니다.

표 13-7 (계속) HPM_STGD_CALCRLUE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
created_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 원래 생성한 사용자를 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
created_timestamp	timestamp (6)	timestamp		행이 원래 생성된 날짜/시간을 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 원래 생성한 사용자를 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_timestamp	timestamp (6)	timestamp		행이 원래 생성된 날짜/시간을 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.

HPM_STGD_CALCRLUE_MULTISRC

HPM_STGD_CALCRLUE_MULTISRC 테이블은 다중 소스 계산 규칙 생성 또는 업데이트에 대한 세부정보를 제공합니다.

표 13-8 HPM_STGD_CALCRLUE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
id	identity	number(38,0)	예	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	varchar2(80 CHAR)	예	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)		계산 규칙에 대한 자세한 설명.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	예	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 두 번째 차원 멤버 이름

표 13-8 (계속) HPM_STGD_CALCRULE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		선택한 POV의 네 번째 차원 멤버 이름
calculation_sequence_number		number(4,0)	예	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	varchar2(255)		이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	예	
driver_name	nvarchar (80)	varchar2(80 CHAR)	예	이 계산 규칙과 연결할 드라이버 이름.
src_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	예	이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 소스 지정 규칙.
dest_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	예	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		이 계산 규칙과 연결할 대상 지정 규칙.
last_upload_date	timestamp (6)	timestamp		이 열은 임포트를 통해 채워지며, 임포트가 수행된 가장 최근 날짜를 표시합니다.

표 13-8 (계속) HPM_STGD_CALCRLUE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	필수	설명
import_exception	nvarchar (255)	VARCHAR2(255 CHAR)		행 임포트에 문제가 있는지 여부를 나타내는 오류 코드를 포함합니다. Null은 임포트에 성공했음을 의미합니다.
created_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 원래 생성한 사용자를 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
created_timestamp	timestamp (6)	timestamp		행이 원래 생성된 날짜/시간을 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		이 테이블의 행을 원래 생성한 사용자를 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.
modified_timestamp	timestamp (6)	timestamp		행이 원래 생성된 날짜/시간을 식별합니다. Profitability and Cost Management에서 사용되지 않습니다.

상세 Profitability에 대한 모델 정의 데이터 익스포트

참조:

- [상세 Profitability에 대한 모델 정의 데이터 익스포트 정보](#)
모델이 생성되면 데이터베이스를 쿼리하여 모델 아티팩트를 데이터베이스 뷰에 출력으로 표시할 수 있습니다.
- [HPM_EXPDP_STAGE](#)
HPM_EXPDP_STAGE 뷰는 단계 이름 및 순서, 애플리케이션 이름, 소스 단계 및 대상 단계 차원을 검색합니다.
- [HPM_EXPDP_POV](#)
HPM_EXPDP_POV 뷰는 POV에 포함된 각 차원 멤버 조합의 상태를 검색합니다.
- [HPM_EXPDP_DRIVER](#)
HPM_EXPDP_DRIVER 뷰는 드라이버 유형, 표시 순서 및 공식을 비롯하여 드라이버에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_DRIVER_SEL](#)
HPM_EXPDP_DRIVER_SEL 뷰는 POV 및 드라이버 선택에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_DRIVER_EXCEP](#)
HPM_EXPDP_DRIVER_EXCEP 뷰는 선택한 드라이버 예외의 POV 및 소스 단계에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_ASGN_RUL_SEL](#)
HPM_EXPDP_ASGN_RULE_SEL 뷰는 지정 규칙 선택에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_CALCRLULE_SNGLSRC](#)
HPM_EXPDP_CALCRLULE_SNGLSRC 뷰는 단일 소스 계산 규칙에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_CALCRLULE_CALCMSRS](#)
HPM_EXPDP_CALCRLULE_CALCMSRS 뷰는 계산된 측정항목 규칙에 대한 세부정보를 검색합니다.
- [HPM_EXPDP_CALCRLULE_MULTISRC](#)
HPM_EXPDP_MULTISRC 뷰는 다중 소스 계산 규칙에 대한 세부정보를 검색합니다.
- [상세 Profitability 애플리케이션에 대한 통계 생성](#)
상세 애플리케이션 모델 계산 통계 스크립트를 상세 Profitability 애플리케이션과 함께 사용하여 모델 이해와 디버깅에 유용할 수 있는 모델 및 모델 계산 통계를 생성할 수 있습니다.

상세 Profitability에 대한 모델 정의 데이터 익스포트 정보

모델이 생성되면 데이터베이스를 쿼리하여 모델 아티팩트를 데이터베이스 뷰에 출력으로 표시할 수 있습니다.

관리자는 스테이지 테이블에 사용된 열을 미리링하는 데이터베이스 뷰를 시스템 데이터베이스에 생성하여 시스템에 저장된 모델 데이터를 표시할 수 있습니다.

HPM_EXPDP_STAGE

HPM_EXPDP_STAGE 뷰는 단계 이름 및 순서, 애플리케이션 이름, 소스 단계 및 대상 단계 차원을 검색합니다.

표 14-1 HPM_EXPDP_STAGE

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	Oracle Hyperion Profitability and Cost Management 데이터베이스에서 사용되는 단계 ID
name	nvarchar (80)	varchar2(80 CHAR)	선택한 소스 단계의 이름
description	nvarchar (255)	varchar2(255 CHAR)	선택한 단계의 용도에 대한 설명
stage_order	integer	number (38,0)	계산 스크립트 생성 및 계산에서 모델링(지정 및 규칙 생성)하는 동안 선택한 단계가 사용되는 시퀀스 위치
stage_prefix	nvarchar (80)	varchar2(80 CHAR)	단계 이름의 비교 접두어
driver_dim_name	nvarchar (80)	varchar2(80 CHAR)	선택한 드라이버 차원의 이름
Is_intra	nvarchar (1)	varchar2(1 CHAR)	단계 내 지정을 허용할지 여부를 결정하는 플래그 <ul style="list-style-type: none"> "Y"(예)를 지정하면 모델 단계에서 단계 내 지정이 허용됩니다. "N"(아니요)을 지정하면 모델 단계에서 단계 내 지정이 허용되지 않습니다.
dim1_name	nvarchar (80)	varchar2(80 CHAR)	단계에 포함된 첫 번째 차원의 이름
dim2_name	nvarchar (80)	varchar2(80 CHAR)	단계에 포함된 두 번째 차원의 이름(사용 가능한 경우)
dim3_name	nvarchar (80)	varchar2(80 CHAR)	단계에 포함된 세 번째 차원의 이름(사용 가능한 경우)
dim4_name	nvarchar (80)	varchar2(80 CHAR)	단계에 포함된 네 번째 차원의 이름(사용 가능한 경우)
dim5_name	nvarchar (80)	varchar2(80 CHAR)	단계에 포함된 다섯 번째 차원의 이름(사용 가능한 경우)
dim6_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim7_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim8_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim9_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름

표 14-1 (계속) HPM_EXPD_STAGE

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
dim10_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim11_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim12_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim13_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim14_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim15_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim16_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim17_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim18_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim19_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim20_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim21_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim22_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim23_name	nvarchar (80)	vvarchar2(80 CHAR)	대상 단계의 차원 이름
dim24_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름
dim25_name	nvarchar (80)	varchar2(80 CHAR)	대상 단계의 차원 이름

HPM_EXPD_POV

HPM_EXPD_POV 뷰는 POV에 포함된 각 차원 멤버 조합의 상태를 검색합니다.

표 14-2 HPM_EXPD_POV

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름
pov_state	nvarchar (80)	varchar2(80 CHAR)	POV의 현재 상태는 다음과 같습니다. <ul style="list-style-type: none"> • 초안 • 게시됨 • 아카이브됨

HPM_EXPD_DRIVER

HPM_EXPD_DRIVER 뷰는 드라이버 유형, 표시 순서 및 공식을 비롯하여 드라이버에 대한 세부정보를 검색합니다.

표 14-3 HPM_EXPD_DRIVER

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	고유 레코드 ID
name	nvarchar (80)	varchar2(80 CHAR)	선택한 드라이버의 이름
description	nvarchar (255)	varchar2 (255 CHAR)	선택한 드라이버의 용도에 대한 설명
display_order	integer(38,0)	integer(38,0)	모델에 있는 모든 드라이버 목록 내의 드라이버 표시 위치
formula	nvarchar (4000)	varchar2 (4000 CHAR)	사용자가 드라이버에 대해 생성한 공식. 이 공식은 SQL 구문과 수학적으로 정확한 연산을 사용하여 생성해야 합니다.
allow_idle_cost	nvarchar (1)	varchar2 (1 CHAR)	이 드라이버에 유휴 비용을 허용할지 여부를 결정하는 플래그 <ul style="list-style-type: none"> "Y"(Yes)를 지정하면 이 드라이버에 유휴 비용이 허용됩니다. "N"(No)을 지정하면 이 드라이버에 유휴 비용이 허용되지 않습니다.
priority	integer	number (38,0)	드라이버의 계산 우선순위를 입력하여 단계 내의 할당이 지정된 순서대로 실행될 수 있도록 합니다. 우선순위가 가장 낮은 드라이버가 가장 먼저 처리됩니다. 기본적으로 우선순위는 100으로 설정되며 가장 높은 우선순위는 1입니다. 번호는 순차적이지 않아도 됩니다. 동일한 우선순위를 가진 드라이버는 특정한 순서 없이 실행됩니다. 양의 정수만 적합합니다.
volume_formula	nvarchar (4000)	varchar2(4000 CHAR)	비율 기반(Rate-Based) 드라이버에만 해당되며, 비율 및 볼륨 공식이 둘 다 정의되어 있는 공식
dst_measure_member_name	nvarchar (2000)	varchar2(2000 CHAR)	대상 테이블의 측정항목 이름

표 14-3 (계속) HPM_EXPDP_DRIVER

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
tdv_measure_member_name	nvarchar (80)	varchar2(2000 CHAR)	할당 공식 드라이버 값/ TotalDriverValue(DV/ TDV)에서 분모로 사용되는 측정항목
operation_type	nvarchar (80)	varchar2(80 CHAR)	드라이버에 대해 선택된 작업 유형 <ul style="list-style-type: none"> • 비율 기반(Rate-Based) 할당 • 비율 기반(Ratio- Based) 할당 • 계산된 측정항목

HPM_EXPDP_DRIVER_SEL

HPM_EXPDP_DRIVER_SEL 뷰는 POV 및 드라이버 선택에 대한 세부정보를 검색합니다.

표 14-4 HPM_EXPDP_DRIVER_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름(해당되는 경우)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름(해당되는 경우)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름(해당되는 경우)
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 드라이버 차원의 이름
driver_name	nvarchar (80)	varchar2(80 CHAR)	선택한 드라이버의 이름

HPM_EXPD_DRIVER_EXCEP

HPM_EXPD_DRIVER_EXCEP 뷰는 선택한 드라이버 예외의 POV 및 소스 단계에 대한 세부정보를 검색합니다.

표 14-5 HPM_EXPD_DRIVER_EXCEP

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	고유 레코드 ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름 (해당되는 경우)
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름 (해당되는 경우)
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름 (해당되는 경우)
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 첫 번째 차원 멤버 이름
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 두 번째 차원 멤버 이름 (해당되는 경우)
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 세 번째 차원 멤버 이름 (해당되는 경우)
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 네 번째 차원 멤버 이름 (해당되는 경우)
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 다섯 번째 차원 멤버 이름 (해당되는 경우)
driver_name	nvarchar (80)	varchar2(80 CHAR)	선택한 드라이버의 이름

HPM_EXPD_ASGN_RULE_SEL

HPM_EXPD_ASGN_RULE_SEL 뷰는 지정 규칙 선택에 대한 세부정보를 검색합니다.

표 14-6 HPM_EXPD_ASGN_RULE_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	varchar2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	integer(38,0)	고유 레코드 ID

표 14-6 (계속) HPM_EXPD_ASGN_RULE_SEL

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 첫 번째 차원 멤버 이름
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 두 번째 차원 멤버 이름
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 세 번째 차원 멤버 이름
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 네 번째 차원 멤버 이름
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	소스 단계의 다섯 번째 차원 멤버 이름
rule_name	nvarchar (80)	varchar2(80 CHAR)	선택한 지정을 제어하는 규칙 이름. 이 규칙은 대상 데이터베이스에 있어야 합니다.

HPM_EXPD_CALCRLULE_SNGLSRC

HPM_EXPD_CALCRLULE_SNGLSRC 뷰는 단일 소스 계산 규칙에 대한 세부정보를 검색합니다.

표 14-7 HPM_EXPD_CALCRLULE_SNGLSRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name		VARCHAR2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	NUMBER(38,0)	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	VARCHAR2(80 CHAR)	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)	계산 규칙에 대한 자세한 설명.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름

표 14-7 (계속) HPM_EXPD_CALCRLULE_SNGLSRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
calcuation_sequence	integer	NUMBER(4,0)	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	VARCHAR2(255)	이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
first_sequence_priority	integer	NUMBER(38,0)	이 규칙에 대해 실행할 드라이버 범위의 첫 번째 드라이버 순서 우선순위를 지정합니다. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
last_sequence_priority	integer	NUMBER(38,0)	이 규칙에 대해 실행할 드라이버 범위의 최종 드라이버 순서 우선순위를 지정합니다. 이 값은 <i>first_sequence_priority</i> 보다 크거나 같아야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	Y인 경우 계산 중 <i>first_sequence_priority</i> 및 <i>last_sequence_priority</i> 필드의 값을 무시합니다.

HPM_EXPD_CALCRLULE_CALCMSRS

HPM_EXPD_CALCRLULE_CALCMSRS 뷰는 계산된 측정항목 규칙에 대한 세부정보를 검색합니다.

표 14-8 HPM_EXPD_CALCRLULE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar (80)	VARCHAR2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	NUMBER(38,0)	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	VARCHAR2(80 CHAR)	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)	계산 규칙에 대한 자세한 설명.

표 14-8 (계속) HPM_EXPD_CALCRULE_CALCMSRS

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름
calculation_sequence	integer	NUMBER(4,0)	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	VARCHAR2(255)	이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
driver_name1	nvarchar (1)	VARCHAR2(1 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
driver_name2	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
driver_name3	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
driver_name4	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
driver_name5	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.

HPM_EXPD_CALCRLUE_MULTISRC

HPM_EXPD_MULTISRC 뷰는 다중 소스 계산 규칙에 대한 세부정보를 검색합니다.

표 14-9 HPM_EXPD_CALCRLUE_MULTISRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
application_name	nvarchar(80)	VARCHAR2(80 CHAR)	선택한 애플리케이션의 이름
id	identity	NUMBER(38,0)	각 행의 고유 식별자. 임의 순서 생성기에서 제공될 수 있습니다.
name	nvarchar (80)	VARCHAR2(80 CHAR)	계산 규칙의 이름.
description	nvarchar (255)	varchar2(255 CHAR)	계산 규칙에 대한 자세한 설명.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 첫 번째 차원 멤버 이름
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 두 번째 차원 멤버 이름
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 세 번째 차원 멤버 이름
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	선택한 POV의 네 번째 차원 멤버 이름
calculation_sequence	integer	NUMBER(4,0)	계산 중에 계산 규칙이 처리되는 상대 순서를 지정하는 숫자 값. 이 값은 0보다 커야 하며, 그렇지 않으면 삽입 중에 CHECK 제약 조건 오류가 발생합니다.
tag	nvarchar (255)	VARCHAR2(255)	이 열의 계산 규칙에 대한 검색 태그를 지정합니다.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
driver_name	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 드라이버 이름.
src_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 소스 지정 규칙.
src_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 소스 지정 규칙.
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.

표 14-9 (계속) HPM_EXPD_CALCRULE_MULTISRC

필드	SQL Server 데이터 유형	Oracle 데이터 유형	설명
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	이 계산 규칙과 연결할 대상 지정 규칙.

상세 Profitability 애플리케이션에 대한 통계 생성

상세 애플리케이션 모델 계산 통계 스크립트를 상세 Profitability 애플리케이션과 함께 사용하여 모델 이해와 디버깅에 유용할 수 있는 모델 및 모델 계산 통계를 생성할 수 있습니다.

이 스크립트는 Oracle Hyperion Profitability and Cost Management 제품 스키마 내에 테이블을 생성하고 모델 계산 결과 문제 해결, 모델의 모양과 특성 설명, 모델 계산 성능 평가에 유용한 통계를 저장합니다.

dpmodelcalcstats.sql 스크립트는 11.1.2.2.351 패치가 설치된 후

```
%EPM_ORACLE_HOME%
\OPatch\HPCM_11_1_2_2_351_14845052\files\products\Profitability\database\Common\Oracle
```

내에 있습니다. 현재 Oracle 데이터베이스 설치에만 사용할 수 있습니다.

이 스크립트는 SQL Developer 내에서 열리고 실행됩니다.

요청된 애플리케이션에 대한 통계를 생성하려면 다음을 수행합니다.

1. Oracle SQL Developer 내에서 dpmodelcalcstats.sql을 엽니다.
2. Oracle Fusion Performance Management 제품 스키마에 대한 연결을 선택합니다.
3. 커서를 스크립트의 첫 번째 라인으로 이동합니다.
4. 스크립트 실행을 누르거나 F5 키를 누릅니다.
5. 모델 계산 통계가 요청된 애플리케이션의 이름을 입력합니다.

이 스크립트는 다음 이름 지정 규칙을 사용하여 제품 스키마 내에 테이블을 생성합니다.

HPMD_<애플리케이션 이름>_DIAGSTAT_<고유 번호>

HPMD_BksDP20_DIAGSTAT_4129 테이블 내의 열에 대한 간단한 설명은 다음과 같습니다.

APPLICATION: 애플리케이션의 이름.

ITEM: 행 내의 콘텐츠에 대한 간단한 설명.

RATIO_DRV_TOT: 모든 비율 기반(Ratio-Based) 드라이버와 관련된 총 항목 수.

RATE_DRV_TOT: 모든 비율 기반(Rate-Based) 드라이버와 관련된 총 항목 수.

OBJC_DRV_TOT: 모든 객체 계산 드라이버와 관련된 총 항목 수.

TOTAL: 모든 지정과 관련된 총 항목 수.

NAME: 항목과 관련된 아티팩트(artifact) 또는 아티팩트(object)의 이름.

다음 유형의 진단 통계 항목이 테이블 내에 포함됩니다.

지정 테이블: 지정한 애플리케이션에 대한 모든 지정 세부정보가 포함된 지정 테이블의 이름. 개수가 1보다 큰 경우 명확한 계산이 실행되지 않았기 때문에 결과가 부정확할 수 있습니다.

지정 수: 지정한 애플리케이션의 모델 내에서 식별된 지정 수.

소스 노드 수: 모델 내에서 식별된 소스 노드(교차) 수.

데이터가 있는 지정 수: 소스 단계 비즈니스 객체 내의 하나 이상 행이 잠재적 소스 노드와 일치하는 지정 수.

행 업데이트 수: 모든 지정에 대해 수행된 Update 문에서 변경되는 총 행 수.

지정 규칙 수: 지정 규칙을 참조하는 지정이 하나 이상 있는 모델 내에서 정의된 지정 규칙 아티팩트 수.

드라이버 수: 드라이버를 참조하는 지정이 하나 이상 있는 모델 내에서 정의된 드라이버 아티팩트 수.

최대 대상 행 개수: 지정에 의해 대상 내에서 업데이트되는 최대 행 수.

중앙값 대상 행 개수: 지정에 의해 대상 내에서 업데이트되는 중앙값 행 수. 표준

대상 행 개수의 편차: 지정에 의해 대상 내에서 업데이트되는 행의 표준 편차.

드라이버 개수: 하나 이상의 지정에서 참조되는 드라이버당 하나의 드라이버 개수 항목이 있습니다. 항목에는 드라이버 아티팩트의 이름과 지정에서 참조되는 횟수가 포함됩니다.

지정 규칙 개수: 하나 이상의 지정에서 참조되는 지정 규칙당 하나의 지정 규칙 개수 항목이 있습니다. 항목에는 지정 규칙 아티팩트의 이름과 지정에서 참조되는 횟수가 포함됩니다.

A

애플리케이션 디자인 모범 사례

애플리케이션 확장성 관리

할당 프로세스에서는 초기 데이터의 크기를 여러 번 확장할 수 있습니다.

할당 규칙에서는 리프 레벨에서 원래 데이터 값을 사용하고 모든 대상 셀에서 분할합니다. 할당 규칙을 실행한 후, 데이터가 원래 크기보다 수백 배 증가하는 것은 드문 일이 아닙니다. 할당 규칙의 워터폴은 이 작은 크기를 여러 개의 더 작은 값으로 분할할 수 있습니다.

모델 계산의 마지막 부분에서 수백만 개의 미세 데이터 포인트를 처리하지 않으려면 디자인 프로세스 초기에 데이터 스케일을 고려해야 합니다. 데이터가 확장되면 다운스트림 보고 시스템 또는 GL로 다시 보낼 데이터 추출을 생성하여 Oracle Hyperion Profitability and Cost Management의 통합 결과를 상당히 복잡하게 만들 수 있습니다.

차원 수

데이터 증가를 계속 제어하는 가장 좋은 방법은 데이터 스케일에 영향을 줄 요소를 디자인 단계의 초기에, 특히 사후 할당 상태에서 고려하는 것입니다. 애플리케이션에서 사용되는 차원 수가 확장성의 첫번째 고려 사항입니다.

명확한 이유 없이 데이터에 차원을 추가하는 데 유의하십시오. Oracle Hyperion Profitability and Cost Management의 데이터 증가는 데이터를 더 작은 값으로 분할한 수의 영향을 가장 많이 받습니다. 새 차원을 추가하기 전에 새 물리적 차원이 필요한지 확인하십시오. 처음에 Profitability and Cost Management 디자이너는 일반적으로 소스 데이터에 있는 모든 차원을 채택합니다. 단순히 해당 차원이 소스 데이터에 있으며 나중에 필요할 수도 있기 때문입니다.

차원이 최종 결과 보고 또는 할당 프로세스를 지원하기 위해 데이터를 구분하는 데 필요한지 고려하십시오. 이 모두가 해당되지 않는 경우 차원 제거를 심각하게 고려해야 합니다.

추가 차원이 기존 차원의 대체 표현식이면 대체 계층 또는 속성 차원을 대신 사용하십시오. 그러면 데이터 크기를 늘리지 않고 원하는 범주에서 보고할 수 있습니다.

차원을 제한하도록 적극 권장하지만, 나중에 증가를 위해 차원을 추가하는 것도 좋습니다. 차원이 모든 모델 아티팩트에서 단일 "nomember" 선택사항만 사용하는 한, "spare" 차원은 성능에 거의 영향을 미치지 않습니다.

차원 멤버 수

Oracle Hyperion Profitability and Cost Management의 데이터 스케일은 차원 수 및 해당 차원 크기(멤버 수)의 함수입니다. 규칙의 범위 크기는 차원 수와 멤버 수 둘 다의 영향을 받습니다.

애플리케이션을 디자인할 때 매우 큰 차원을 사용하는 것이 좋습니다. 차원 수가 적고 크기가 작을 때 Profitability and Cost Management 애플리케이션이 가장 적합합니다. 그렇다고 해서 Profitability and Cost Management에서 수가 적고 큰 차원을 처리할 수 없다는 의미는 아닙니다. 단, 큰 차원을 많이 사용하는 경우 광대하지만 밀도가 희박한 범위에서 작동하도록 규칙을 디자인하기 시작할 때 계산 성능에 문제가 발생할 수 있습니다.

차원에서 멤버 세부정보를 선택할 때는 처음 차원을 선택할 때와 동일한 이유를 사용하십시오.

- 보고하는 데 세부정보가 필요합니까?
- 할당 논리를 지원하기 위해 데이터를 구분해야 합니까?

두 질문의 답변이 모두 아니요이면 차원에서 더 집계된 수준의 멤버를 사용해야 합니다. 예를 들어 가장 낮은 레벨 계정 대신 비용 풀을 사용하거나 SKU 대신 제품군을 사용합니다.

속성 및 UDA

속성은 보고 및 할당 논리 모두에 강력한 틀이지만, 기본 멤버와 속성 멤버 간의 연계 관리는 복잡해질 수 있습니다. 매우 큰 기본 차원(10,000+) 또는 다수의 고유 속성 값을 사용하여 작업할 때 속성이 아니라 UDA를 사용하는 것이 좋습니다.

데이터 스케일 관리

디자인 주기 초기에 제어되지 않는 데이터 증가가 미칠 수 있는 부정적인 영향을 파악하는 것이 중요합니다. 발생할 수 있는 보고를 지원하기 위해 탐색 중인 애플리케이션에 세부정보를 일일이 추가하지 않는 것이 중요합니다. 디자인을 시작하기 전에 애플리케이션의 보고 요구사항을 파악하십시오. 이 방법을 통해서만 디자이너가 "정말로 그 정도로 상세해야 합니까...?"라는 질문에 답할 수 있습니다.

이 요구사항을 알고 있으면 다음과 같은 두 가지 주요 디자인 방법을 통해 증가를 제한할 수 있습니다.

- [시작 데이터 요약](#)
- [할당 중간점 요약](#)

시작 데이터 요약

실제로 일부 데이터는 필요하지 않습니다. 앞서 설명한 대로 데이터를 로드하기 전에 데이터의 요약 포인트를 사용하는 것이 가장 좋은 툴 중 하나입니다. 계정 세부정보가 일반적인 예입니다. 가장 낮은 레벨의 세부정보에 비용을 로드하는 대신, 집계 비용 풀을 사용하십시오. 데이터의 가능한 모든 차원에서 이 전략을 사용합니다. 보고 또는 할당 프로세스에 필요한 세부정보는 이전 질문을 참조하십시오.

다음 질문을 통해 보고 또는 할당 프로세스에 세부정보가 필요한지 판별하십시오.

- 보고하는 데 세부정보가 필요합니까?
- 세부정보에서 할당 논리를 지원하기 위해 데이터를 구분해야 합니까?

이 단계만 수행해도 하나 이상의 스케일 순서로 시작 데이터 크기를 축소할 수 있습니다.

할당 중간점 요약

시작 데이터 포인트와 비슷하게, 이전 할당 단계에서 사용된 세부정보 레벨이 더 이상 필요하지 않은 애플리케이션에서 퍼널 포인트가 있습니다. 이 경우 애플리케이션의 중간 단계에 있는 풀링 전략을 사용하면 데이터를 보다 관리 가능한 크기로 다시 축소할 수 있습니다.

이 기능은 고객, 지역/영역, 제품/SKU와 같은 시장 관련 차원에 재무 데이터를 할당하기 바로 전 단계에 특히 유용합니다. 일반적으로 시장 관련 차원은 애플리케이션에서 가장 큰 차원이며 일반적으로 데이터 급증이 발생하는 단계입니다.

운영 센터 할당 후에 데이터 스케일을 줄이면 시장 교차점 간에 할당할 데이터 포인트 수가 줄어들고 결과적으로 데이터 세트가 작아집니다.

차원 디자인 관리

차원 디자인은 보고 및 계산 성능에 큰 영향을 줄 수 있습니다.

Oracle Hyperion Profitability and Cost Management는 Oracle Essbase 집계 저장영역(ASO) 데이터베이스를 사용합니다. ASO 데이터베이스에는 다른 Oracle EPM 비즈니스 프로세스에서 사용한 블록 저장영역(BSO) 데이터베이스와 다른 성능 특징 및 최적화 방법이 있습니다.

차원 유형

Oracle Essbase ASO에서는 다음 세 가지 차원을 제공합니다.

- 저장
- 동적
- 계층 사용 설정

저장

일반적으로 저장된 차원에서는 계산 및 보고에 가장 적합한 성능을 제공합니다.

저장된 차원 지원

- 가장 빠른 집계
- 공유 멤버
- 속성 및 UDA

제한사항

- 멤버는 추가(+) 통합 연산자만 사용할 수 있습니다.
- 멤버 공식 없음
- 공유 멤버는 공유의 기본 멤버 뒤에 표시되어야 합니다.

동적

동적 차원은 가장 뛰어난 디자인 유연성을 제공하지만, 계산 및 보고 성능은 제한됩니다.

동적 차원 지원

- 모든 통합 연산자(+, -, *, /, ~)
- 멤버 공식
- 공유 멤버
- 속성 및 UDA

동적 차원 제한사항

- 가장 느린 집계
이는 계산 및 보고 성능에 영향을 미칩니다. 애플리케이션에서 여러 동적 차원을 사용하면 이 문제가 발생합니다.
- 공유 멤버는 공유의 기본 멤버 뒤에 표시되어야 합니다.

계층 사용 설정

계층 사용 차원에서는 차원의 일부가 저장된 차원과 비슷하게 작동하고 수행하며 다른 계층은 동적 차원과 비슷하게 작동하고 수행할 수 있는 저장된 계층과 동적 계층의 하이브리드를 제공합니다.

계층 사용 설정 차원 지원

- 여러 저장된 계층과 동적 계층의 조합
- 저장된 계층에는 저장된 차원 유형과 동일한 기능 지원 및 제한사항이 있습니다.
- 동적 계층에는 동적 차원 유형과 동일한 기능 지원 및 제한사항이 있습니다.

계층 사용 설정 차원 제한사항

- 1세대 2 멤버 계층은 저장된 계층 유형이어야 합니다.
- 루트 멤버에는 레이블 전용 데이터 저장영역이 있어야 합니다.

계층 최상위 레벨

차원의 최상위 레벨은 모델 뷰를 사용하는 규칙 간액조정과 같은 Oracle Hyperion Profitability and Cost Management 화면과 보고 툴의 유용성에 영향을 미칠 수 있습니다. 일반적으로 데이터가 루트 멤버로 결합될 수 있도록 차원을 설정하면 보고서에서 차원을 더 쉽게 사용할 수 있습니다.

롤업 및 보고

가능할 때마다 기본 계층의 모든 데이터가 차원의 최상위 멤버로 롤업됩니다. 따라서 사용자가 해당 차원의 모든 데이터를 보려는 경우 보고서에 명시적으로 멤버를 선택하는 작업을 건너뛸 수 있습니다. 또한 임시로 규칙 밸런싱 및 기타 보고 툴을 사용하기가 간단해 집니다.

멤버 없음

불규칙 차원과 데이터를 혼합하는 일반적인 방법은 차원에 참조가 없는 데이터 포인트를 지정하는 멤버를 사용하는 것입니다. "NoProduct"와 같은 "No" 멤버를 사용하면 아직 특정 제품으로 지정되지 않은 데이터를 저장하도록 지원합니다.

계층 사용 차원에 "No" 멤버를 사용하는 경우 1세대 2계층에 "No" 멤버를 배치하는 것이 좋습니다. 그러면 아직 해당 차원에 속하지 않은 데이터를 포함하여 차원의 모든 데이터 포인트를 차원의 맨 위에 집계하여 보고를 간소화합니다.

모범 사례 예

제품 계층을 예로 사용:

단일 계층 차원

차원 루트 멤버(제품) 데이터 저장영역: 데이터 저장

1세대 2 계층(총 제품) 집계 연산자: 추가

- 첫번째 1차 하위(모든 제품) 집계 연산자: 추가. 제품 계층의 나머지는 이 멤버 아래에 배치됩니다.

- 두번째 1차 하위(NoProduct) 집계 연산자: 추가. 데이터 포인트에 연계된 제품 멤버가 없는 경우에 사용되는 멤버입니다.

다중 계층 차원

차원 루트 멤버(제품) 데이터 저장영역: 레이블 전용

- 1세대 2 계층(총 제품) 집계 연산자: 추가.
 - 첫번째 1차 하위(모든 제품) 집계 연산자: 추가. 제품 계층의 나머지는 이 멤버 아래에 배치됩니다.
 - 두번째 1차 하위(NoProduct) 집계 연산자: 추가. 데이터 포인트에 연계된 제품 멤버가 없는 경우에 사용되는 멤버입니다.
- 2세대 2계층(대체 계층) 집계 연산자: 무시.
- 3세대 2계층(대체 계층) 집계 연산자: 무시.
- ...N세대 2계층(대체 계층) 집계 운영자: 무시

중복 멤버

중복 멤버는 데이터를 저장하고 이름이 동일한 기본 멤버(공유 멤버가 아님)입니다. Oracle Hyperion Profitability and Cost Management에서는 여러 차원에서 중복 멤버를 지원하지만 동일한 차원 내의 중복 멤버는 지원하지 않습니다.

중복 차원 멤버는 소스 조직 구성 단위에서 타겟 조직 구성 단위로 이동할 때 비용을 추적하는 경우 유용하며, 소스와 타겟 사이에서 이동된 비용을 보고해야 합니다.

Profitability and Cost Management에서 중복 멤버를 지원하는 반면, 이를 사용하는 데 따른 문제가 있습니다. Oracle Essbase에서 데이터를 포함하는 데이터 셀은 고유하게 확인되어야 합니다. 중복 멤버를 사용하지 않으면 시스템에서 멤버 이름 조합을 사용하여 셀을 확인합니다.

중복 멤버를 사용하면 추가 정규화가 필요합니다. 예를 들어 데이터 셀 조직 멤버(예: "Marketing")를 참조하는 대신, 참조에 차원(예: "Entity.TotalEntitiy.Administration.Marketing")도 포함해야 합니다. 중복 멤버에 필요한 세분화된 정규화 때문에 보고서를 생성하고 차원을 수정할 때, 특히 멤버를 이동할 때 문제가 발생할 수 있습니다. 중복 멤버를 이동하면 완전한 이름이 변경된 것입니다. 그러면 데이터베이스 재배포 프로세스에서 차원을 수정한 후 데이터의 적절한 셀을 찾을 수 없으므로, 데이터 값이 삭제됩니다.

중복 차원을 생성하는 대신, 중복 항목이 작성되지 않도록 하나 또는 두 차원 모두의 멤버를 접두부로 지정해야 합니다.

애플리케이션 로직 디자인 관리

규칙을 생성하기 전에 애플리케이션 디자인의 맵을 생성합니다.

참조:

- 규칙 수를 관리하는 단순한 전략
- 할당 규칙 유형 및 옵션
- 규칙 범위
- 규칙 세트에서 병렬 실행 사용
- 사용자정의 계산 규칙

규칙 수를 관리하는 단순한 전략

규칙을 작성하기 전에 애플리케이션 디자인의 맵을 작성하는 것이 가장 좋습니다. 많은 사용자가 할당 규칙의 총 범위를 보고 전체 프로세스를 개선하려고 시도하기 전에 레거시 시스템에서 할당을 복제하려고 합니다. 대부분의 경우 프로세스를 새로 살펴보고 Oracle Hyperion Profitability and Cost Management에서 제공하는 여러 툴을 살펴보면 프로세스의 단계를 줄이고 프로세스를 더 쉽게 이해하여 다른 사람에게 더 쉽게 설명할 수 있는 방법을 찾을 수 있습니다.

- 단일 규칙을 사용하여 일반 논리를 적용할 수 있는 위치 파악
- 드라이버 데이터 부재를 할당 필터로 사용

단일 규칙을 사용하여 일반 논리를 적용할 수 있는 위치 파악

Oracle Hyperion Profitability and Cost Management로 레거시 할당 규칙을 마이그레이션할 때 일반적으로 여기에서 첫번째 문제가 발생하지만 쉽게 해결할 수 있습니다. 여러 부서와 엔티티 간에 공통인 할당 패턴을 확인하면 레거시 프로세스의 여러 단계를 Profitability and Cost Management의 단일 규칙으로 결합할 가능성이 큼니다.

동일한 드라이버 값을 사용하여 동일한 대상 세트에 많은 논리 영역을 할당해야 하는 경우가 있는지 확인하십시오. 이 경우 대상 탭에서 소스와 동일 옵션을 사용하면 여러 병렬 할당 단계를 단일 규칙으로 실행할 수 있습니다.

드라이버 데이터 부재를 할당 필터로 사용

할당을 받아야 하는 대상 범위의 멤버가 할당을 받지 않아야 하는 멤버 간에 분산되어 있으면 멤버가 대상의 일부여도 할당된 금액이 자동으로 수신되지 않습니다. 할당은 드라이버 값이 있는 대상에만 전송됩니다.

이 할당 엔진의 동작을 사용하면 광범위한 소스 및 대상을 교차하는 단일 규칙을 작성할 수 있지만, 드라이버 데이터가 있는 교차에만 할당됩니다. 그러면 여러 규칙을 생성해야 할 수도 있는 레거시 시스템에서 단일 규칙을 작성할 수 있습니다.

할당 규칙 유형 및 옵션

각 규칙의 할당 관계는 Oracle Hyperion Profitability and Cost Management가 계산 논리를 수행하는 데 사용하는 메소드 및 Oracle Essbase 엔진이 이를 실행하는 방법에 영향을 미칩니다. 모든 규칙 유형이 성능 면에서 동일하지는 않습니다.

성능 순서별 주요 규칙 유형은 다음과 같습니다.

- 단순 차원 확장
- 복합 차원 확장
- 단순 재분류
- 복합 재분류
- 소스와 동일, 다른 차원

단순 차원 확장

- 모든 차원의 소스 및 대상 관계가 소스와 동일하게 설정되었거나 하나의 소스(단일 레벨 0 소스)와 많은 대상이 연결된 관계입니다.

- 선택한 모든 드라이버 멤버가 레벨 0입니다.
- 모든 대상 차원이 레벨 0 또는 상위의 단일 멤버 선택사항입니다.
- 단순 차원 확장 규칙 유형이 가장 적합한 유형입니다.

복합 차원 확장

- 모든 차원의 소스 및 대상 관계가 소스와 동일하게 설정되었거나 하나의 소스(단일 레벨 0 소스)와 많은 대상이 연결된 관계입니다.
- 모든 드라이버 멤버 선택사항은 상위 멤버입니다.
- 모든 대상 차원에는 고유 멤버 선택사항이 여러 개 있습니다.

단순 재분류

한 차원의 소스 및 대상 관계는 *여러* 멤버의 정의가 여러 개별 멤버 선택사항인지 아니면 상위 멤버의 선택사항인지에 상관없이 다대다 관계입니다. 나머지 차원 소스 및 대상 관계는 소스와 동일합니다.

복합 재분류

다중 차원의 소스 및 대상 관계는 *여러* 멤버의 정의가 여러 개별 멤버 선택사항인지 아니면 상위 멤버의 선택사항인지에 상관없이 다대다 관계입니다. 나머지 차원 소스 및 대상 관계는 소스와 동일합니다.

일반적으로 복합 분류 규칙은 성능이 가장 느린 규칙 유형입니다.

소스와 동일, 다른 차원

하나 이상의 차원 소스 및 대상 관계에서 소스 및 대상 차원이 서로 다른 소스 옵션을 사용합니다.

이 옵션은 할당 규칙 유형과 함께 사용할 수 있으며, 실행에 복잡도를 추가하고 성능을 저하시킵니다.

이 옵션을 신중하게 사용하고, 애플리케이션 빌드 중에 성능 벤치마크를 자주 수행하여 미치는 영향이 허용 가능한지 판별하십시오.

각 규칙 유형에 허용되는 할당 규칙 및 멤버 선택 옵션

표 A-1 각 규칙 유형에 허용되는 할당 규칙 및 멤버 선택 옵션

	소스 차원	대상 차원	드라이버 차원
SDE(단순 차원 확장)	<ul style="list-style-type: none"> • 단일 레벨 0 멤버 • 동일한 차원에 대해 대상이 소스와 동일한 경우 다른 멤버 선택 	<ul style="list-style-type: none"> • 하나 이상 소스와 동일함 • 단일 레벨 0 멤버 • 동일한 차원에 대해 드라이버가 선택되지 않은 경우 다른 멤버 선택 	<ul style="list-style-type: none"> • 균등 할당 • 지정된 드라이버 위치는 레벨 0 튜플임

표 A-1 (계속) 각 규칙 유형에 허용되는 할당 규칙 및 멤버 선택 옵션

	소스 차원	대상 차원	드라이버 차원
CDE(복합 차원 확장)	<ul style="list-style-type: none"> 단일 레벨 0 멤버 동일한 차원에 대해 대상이 소스와 동일한 경우 다른 멤버 선택 	<ul style="list-style-type: none"> 하나 이상 소스와 동일함 단일 레벨 0 멤버 다른 멤버 선택 	<ul style="list-style-type: none"> 균등 할당 지정된 드라이버 위치는 레벨 0 튜플임 지정된 드라이버 위치는 레벨 0 튜플이 아님
SR(단순 재분류)	<ul style="list-style-type: none"> 단일 레벨 0 멤버 다른 멤버 선택 	<ul style="list-style-type: none"> 소스와 동일함 단일 레벨 0 멤버 동일한 차원에 대해 드라이버가 선택되지 않은 경우 다른 멤버 선택 	<ul style="list-style-type: none"> 균등 할당 지정된 드라이버 위치는 레벨 0 튜플임
CR(복합 재분류)	<ul style="list-style-type: none"> 단일 레벨 0 멤버 다른 멤버 선택 	<ul style="list-style-type: none"> 소스와 동일함 단일 레벨 0 멤버 다른 멤버 선택 	<ul style="list-style-type: none"> 균등 할당 지정된 드라이버 위치는 레벨 0 튜플임 지정된 드라이버 위치는 레벨 0 튜플이 아님
SAD(소스와 동일, 다른 차원)	<ul style="list-style-type: none"> 단일 레벨 0 멤버 다른 멤버 선택 	<ul style="list-style-type: none"> 소스와 동일함 소스와 동일, 다른 차원 다른 멤버 선택 	<ul style="list-style-type: none"> 균등 할당 지정된 드라이버 위치는 레벨 0 튜플임 지정된 드라이버 위치는 레벨 0 튜플이 아님

규칙 범위

성능에 영향을 주는 규칙 디자인의 또 다른 요소는 소스 및 타겟 범위입니다. 규칙 범위를 신중하게 관리하면 성능을 향상시킬 수 있습니다.

- 범위 크기 및 빈도
- 소스 세분화 사용
- 필터 또는 대체 계층 사용
- 음수 드라이버 데이터 값 사용 방지하기
- 동적 멤버를 드라이버로 사용 방지

범위 크기 및 빈도

일반적으로, 소스, 대상 또는 타겟 범위가 클수록 규칙을 실행하는 시간이 길어집니다. 일부 규칙은 매우 큰 범위에서 실행할 수 있지만 다른 규칙은 그렇지 않습니다. 규칙 유형은 계산 시간에 더 큰 영향을 미칩니다. 그러나 지정된 규칙/규칙 유형의 소스와 대상 범위가 작을수록 더 빠르게 실행됩니다. 매우 큰 범위의 규칙을 작성할 때 주의하십시오. 규칙 정의 화면 또는 실행 통계 보고서에서 범위를 검토할 수 있습니다.

소스 세분화 사용

소스 범위를 줄이는 한 가지 옵션은 동일한 대상 및 드라이버 기준 정의를 사용하여 규칙을 여러 개의 작은 규칙으로 분할하고 소스 범위를 더 작은 파트로 분할하는 것입니다. 여러 규칙을 사용하여 수동으로 수행하거나, 규칙 소스 탭에서 소스 세분화 옵션을 사용하여 시스템이 더 작은 세그먼트로 실행을 자동으로 분할하게 할 수 있습니다.

필터 또는 대체 계층 사용

경우에 따라 소스 또는 대상에 포함해야 하는 멤버는 기본적으로 기본 계층에 함께 포함되지 않습니다. 대체 계층을 사용하여 더욱 "규칙을 준수"하는 방식으로 멤버를 그룹화하면 규칙 범위를 제한하고 규칙 논리를 이해하기 쉽게 만들 수 있습니다. 또 다른 옵션은 속성 필터를 사용하는 것입니다.

음수 드라이버 데이터 값 사용 방지하기

음수 값이 있는 드라이버 데이터는 할당 엔진에 문제를 일으켜 예기치 않게 사용 불가능한 결과를 낼 수 있습니다. 할당 엔진은 각 대상 위치에서 찾은 드라이버 값을 드라이버 값/모든 드라이버 값 함께 비율의 분자로 사용합니다. 이 할당 비율에 소스 값을 곱하여 대상에 적용할 *AllocationIn* 금액을 결정합니다. 양수 및 음수가 혼합된 드라이버 값을 사용하면 분모가 더 이상 정확하지 않아 규칙 결과에서 *AllocationIn* 금액의 인플레이션과 디플레이션을 일으킬 수 있습니다. 이러한 결과는 설명하고 정당화하기가 매우 어려우므로 이와 같은 상황이 발생하지 않도록 하는 것이 가장 좋습니다.

동적 멤버를 드라이버로 사용 방지

드라이버 값에 공식이 필요하다면 할당 규칙을 실행하기 전에 값을 계산하는 것이 좋습니다. 규칙 실행 중에 가능한 모든 대상 위치는 규칙에서 검토하여 해당 위치에 있는 드라이버 값을 판별합니다. 계산한 멤버를 드라이버로 참조하는 경우, 시스템에서 가능한 모든 위치의 멤버 공식을 실행해야 합니다. 그러면 규칙 영향에 심각한 영향을 미치는 불필요한 계산이 수없이 이루어질 수 있습니다.

필요한 공식을 실행하고 저장된 멤버에 드라이버 값을 저장하는 사용자정의 계산 규칙을 앞에 두어 규칙에 맞는 드라이버 데이터를 준비하는 것이 더 나은 해결책입니다. 그러면 할당 규칙을 정상적으로 실행하고 드라이버 데이터가 없는 위치를 신속하게 건너뛸 수 있습니다.

규칙 세트에서 병렬 실행 사용

일부 경우에 병렬 방식으로 규칙을 실행하면 전체 성능에 도움이 될 수 있습니다. 계산 옵션은 규칙 세트 레벨에서 설정되며 규칙 세트의 모든 규칙에 영향을 줍니다. 모든 규칙 세트의 기본 옵션은 *순차 처리*입니다. 즉, 모든 규칙이 개별적으로 실행되고 규칙 세트의 모든 규칙이 하나씩 실행됩니다.

규칙이 병렬로 안전한지 확인

규칙은 종종 다른 규칙의 실행에 영향을 줍니다. 한 규칙이 다른 규칙의 소스 범위에 할당되는 경우 규칙 실행 순서는 최종 결과에 막대한 영향을 미칠 수 있습니다. 따라서 기본 계산 옵션이 순차적입니다.

병렬로 안전하게 실행할 수 있는 규칙이 있는 경우 병렬 계산 옵션을 사용하면 성능을 개선할 수 있습니다. 사용 설정하려는 규칙 세트에서 병렬 처리 옵션을 설정하십시오. 애플리케이션 등록정보 화면에서 최대 동시 스레드 옵션을 설정해야 합니다.

스레드 수 실험

동시 스레드의 올바른 레벨을 찾는 것은 실험에 달려 있습니다. 성능 개선이 중지되고 추가 스레드를 사용하여 추가 규칙을 실행하는 것이 역효과를 내는 지점이 있습니다. 병렬 계산을 위한 최적의 방법은 네 개의 동시 스레드로 시작하고 성능 향상이 중지될 때까지 스레드 수를 점진적으로 늘려 규칙 세트를 테스트하는 것입니다. 소스 범위를 혼합하지 않는 규칙을 사용하는 것이 병렬 처리에 좋지만, 대상 범위는 여전히 공통 위치를 공유할 수 있습니다. 이 경우 Oracle Essbase에서 다른 규칙의 대상 데이터 쓰기가 완료될 때까지 한 규칙을 일시 정지해야 합니다. 따라서 스레드 수를 늘려도 성능에 더 이상 도움이 되지 않는 지점이 표시됩니다.

사용자정의 계산 규칙

사용자정의 계산 규칙은 할당 규칙과 동일한 방식으로 대규모 타겟 범위의 영향을 받습니다. 일반적으로 범위가 작을수록 더 빨리 실행됩니다. 대부분의 경우 규칙 공식에 `NONEMPTYTUPLE` 문을 사용하면 특히 규칙의 타겟 범위가 매우 큰 경우 성능에 도움이 됩니다.

B

Profitability and Cost Management로 데이터 임포트

데이터 임포트 정보

Oracle Hyperion Profitability and Cost Management에 데이터를 직접 입력할 수도 있지만, 이 경우 상당한 시간이 소요될 수 있습니다.

애플리케이션 채우기를 원활하게 수행하기 위해 임포트 스테이지 테이블 및 임포트 구성 세트를 사용하여 Profitability and Cost Management 상세 Profitability 애플리케이션으로 데이터를 직접 임포트할 수 있습니다.

모델 데이터는 다음과 같은 여러 소스에서 가져옵니다.

- 모델 정의 데이터는 Profitability and Cost Management에서 가져옵니다. 이 데이터는 Excel 등의 다른 애플리케이션에서 생성되어 준비 테이블을 사용하여 가져올 수 있습니다.
- Oracle Hyperion Enterprise Performance Management System Lifecycle Management를 사용하여 모델 데이터와 애플리케이션을 가져올 수 있습니다. *Oracle Enterprise Performance Management System Lifecycle Management 가이드*를 참조하십시오.

▲ 주의:

데이터를 임포트하기 전에 Oracle Hyperion Enterprise Performance Management Workspace 및 Oracle Essbase 에서 데이터베이스의 백업 디렉토리를 생성하는 것이 좋습니다.

가져올 테이블과 데이터를 지정하려면 임포트 구성을 생성해야 합니다. 이 구성을 저장하여 동일한 데이터 세트를 가져올 때 여러 번 사용할 수 있습니다.

스테이지 테이블 정보

데이터를 Oracle Hyperion Profitability and Cost Management로 임포트하는 경우 사용자는 임포트를 관리하기 위한 사전 정의된 구조를 제공하는 스테이지 테이블을 생성합니다.

관계형 데이터베이스에서 Profitability and Cost Management로 모델 데이터를 가져오려면 Profitability and Cost Management 데이터베이스 테이블이 생성된 위치에서 별도의 데이터베이스 스키마에 준비 테이블 세트를 생성하여 애플리케이션에서 사용할 정보에 서식을 지정해야 합니다.

▲ 주의:

제품 스키마 내에서 임포트 준비 테이블을 생성하지 마십시오. 제품 스키마를 수정하는 일은 지원되지 않을 뿐 아니라 예기치 않은 결과가 생길 수 있습니다.

이 테이블은 Oracle이나 SQL Server 등의 관계형 데이터베이스를 사용하여 생성되며 데이터를 애플리케이션에 쉽게 일치시킬 수 있는 형식으로 구성합니다. 설치 폴더에 설치한 후 Microsoft SQL Server 및 Oracle 데이터베이스에 스테이지 데이터베이스 스크립트를 사용할 수 있습니다. 기본적으로 위치는 `%EPM_ORACLE_HOME%\products\Profitability\database\Common\`입니다. 애플리케이션 유형에 적절한 스크립트를 사용하여 새 데이터베이스에 스테이지 테이블을 생성합니다.

상세 Profitability 애플리케이션의 경우 `create_dp_staging.sql` 스크립트를 사용합니다.

모든 테이블을 동시에 생성하거나 가져오려는 테이블만 생성할 수 있습니다. 다음 데이터 그룹 중 하나 이상을 채워야 합니다.

- POV
- 드라이버
- 드라이버 선택
- 드라이버 예외
- 소스 지정 규칙 연결
- 계산 규칙(상세 Profitability에만 해당)

데이터를 가져오려면 적절한 사용자 역할과 보안 인증이 있어야 합니다. *Oracle Enterprise Performance Management System 사용자 보안 관리 가이드*를 참조하십시오.

임포트 구성 생성

여러 요소를 사용할 수 있는 경우에만 데이터를 임포트할 수 있습니다.

- 소스 데이터베이스 및 연결된 해당 준비 테이블 및 데이터
- 데이터를 수신하는 대상 애플리케이션
- 가져올 테이블과 데이터를 정의하는 임포트 구성. 임포트 구성은 여러 번 사용할 수 있습니다.

임포트를 효율적으로 수행하려면 서로 다른 정보 세트에 대해 별도의 구성을 생성하십시오. 작은 임포트 그룹을 사용하면 임포트 시간을 절감하고 정적 정보의 반복적 업데이트를 방지할 수 있습니다. 예를 들어 다음과 같이 임포트 구성의 모델 요소를 그룹화할 수 있습니다.

- 가끔 변경됨: POV 및 단계
- 자주 변경됨: 드라이버, 드라이버 선택, 드라이버 예외, 지정, 지정 규칙 선택 및 계산 규칙

구성을 생성한 후에는 해당 구성을 실행하여 데이터를 애플리케이션으로 가져옵니다.

임포트 구성을 생성하려면 다음을 수행합니다.

1. 임포트의 소스로 사용할 데이터베이스를 선택합니다.

임포트용으로 형식이 지정된 기존 데이터베이스를 사용하거나 새로운 빈 데이터베이스를 생성할 수 있습니다.

데이터베이스에 대한 이름 지정 제한은 없습니다.

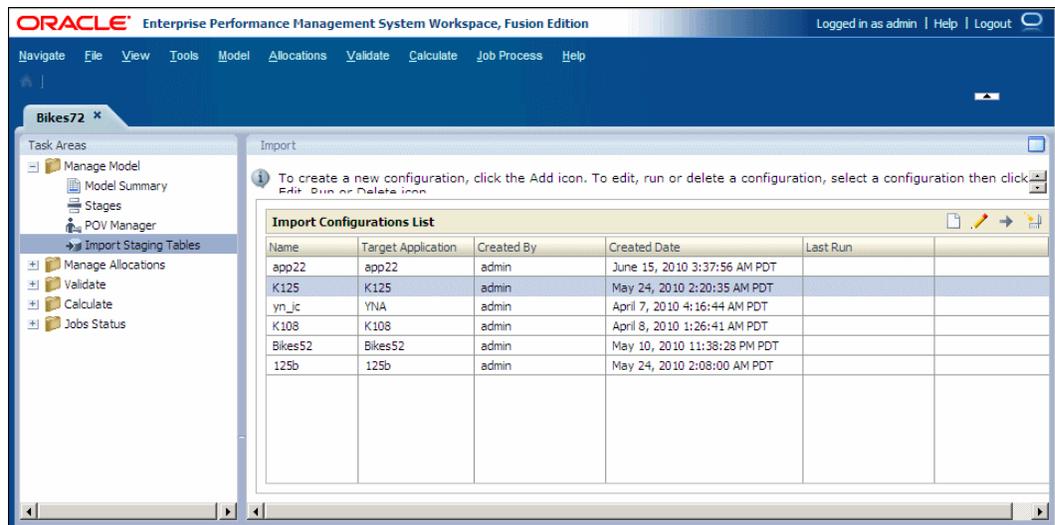
▲ 주의:

데이터나 아티팩트를 임포트하기 전에 Oracle Hyperion Enterprise Performance Management Workspace 및 Oracle Essbase에서 해당 데이터베이스의 백업 디렉토리를 생성하는 것이 좋습니다. 필요한 경우 관리자에게 지원을 요청하십시오.

2. 소스 데이터베이스에 대해 적절한 스크립트를 실행하여 자동으로 스테이지 테이블을 생성합니다.

상세 Profitability 애플리케이션의 경우 create_dp_staging.sql 스크립트를 실행합니다. 상세 Profitability 스테이지 테이블 임포트에 제공된 스키마를 사용하여 테이블이 생성됩니다.

3. 모델을 소스 준비 테이블로 로드합니다.
4. 명확한 이슈는 준비 테이블의 항목을 검토하고 소스 데이터베이스에서 null 행을 제거합니다.
5. 소스 데이터베이스에 액세스할 수 있는지 확인합니다.
6. EPM Workspace에서 **탐색, 애플리케이션, Profitability**, 모델 이름 순으로 선택하여 Oracle Hyperion Profitability and Cost Management 모델에 액세스합니다.
7. 태스크 영역에서 **모델 관리, 스테이지 테이블 임포트** 순으로 선택합니다.



8. 임포트에서 새 임포트 구성 추가 버튼(📄)을 눌러 새 임포트 구성을 생성합니다.
 [데이터 임포트]의 [1단계] 대화상자가 표시됩니다.
9. 데이터 소스 세부정보 아래에서 소스 준비 테이블이 있는 서버에 대한 액세스 세부정보를 입력합니다.
 - 데이터베이스 유형에서 소스 데이터베이스에 사용되는 데이터베이스 유형을 선택합니다.
 - 호스트 아래에서 소스 준비 테이블이 포함된 소스 데이터베이스의 위치 경로를 입력합니다.

- 포트 아래에서 소스 데이터베이스에 연결된 포트 이름을 입력합니다.
 - 데이터베이스 이름 아래에서 소스 데이터베이스의 이름을 입력합니다.
10. 인증서 아래에서 필요한 경우 소스 데이터베이스의 인증된 사용자에 대한 사용자 이름 및 비밀번호를 입력합니다.
 11. 다음을 누릅니다.
 [데이터 임포트]의 [2단계] 대화상자가 표시됩니다.
 12. 구성 아래에서 타겟 애플리케이션을 선택합니다.
 대상 애플리케이션은 데이터를 가져오는 Profitability and Cost Management 애플리케이션입니다.
 13. 준비 테이블 세부정보 아래에서 가져올 준비 테이블을 하나 이상 선택합니다.
 기존 준비 테이블의 이름이 [테이블 이름] 아래에 나열되고 각 스테이지 테이블의 유형이 [스테이지 유형] 아래에 표시됩니다.
 14. 다음을 누릅니다.
 [데이터 임포트]의 [3단계] 대화상자가 표시됩니다.
 15. 구성 요약 아래에 이 임포트 구성을 저장할 구성 이름을 입력합니다.
 이 구성에 연결된 데이터베이스 이름이 구성 이름 아래에 표시됩니다. 이 구성 이름은 80자를 넘을 수 없습니다.
 16. 완료를 누릅니다.
 이 임포트 구성이 [임포트 구성 목록]에 추가되어 동일한 데이터 세트를 가져올 때 여러 번 재사용할 수 있습니다.
 17. 선택사항: [임포트 구성 목록]에서 임포트 구성을 선택하고 **임포트 구성 실행** 버튼()을 눌러 임포트 구성을 실행합니다. [임포트 구성 실행](#)을 참조하십시오.

임포트 구성 수정

임포트 구성을 업데이트하면 대상 애플리케이션, 준비 테이블 세트 및 임포트 구성 이름이 변경될 수 있습니다.

임포트 구성을 수정하려면 다음을 수행합니다.

1. 태스크 영역에서 **모델 관리, 스테이지 테이블 임포트** 순으로 선택합니다.
 [임포트] 페이지가 표시됩니다.
2. **임포트 구성 목록**에서 수정할 임포트 구성을 선택합니다.
3. **임포트 구성 편집** 버튼()을 누릅니다.
 [데이터 임포트] 대화상자가 표시됩니다.
4. 선택한 임포트 구성에 대한 연결 정보 및 인증서를 검토한 후 다음을 누릅니다.
5. 선택 사항: 대상 애플리케이션 아래에서 이 임포트 구성에 대한 대상 애플리케이션을 선택합니다.
6. 선택 사항: 준비 테이블 세부정보에서 가져올 준비 테이블 세트를 변경합니다.
7. 다음을 누릅니다.

8. **선택사항:** 구성 요약에서 다른 구성 이름을 입력하여 수정된 구성을 저장하고 해당 연결 URL을 검토합니다.
9. **완료**를 누릅니다.
수정된 구성이 저장되고 언제든지 실행할 수 있습니다. [임포트 구성 실행](#)을 참조하십시오.

임포트 구성 삭제

임포트 구성을 삭제할 수 있습니다.

임포트 구성을 삭제하려면 다음을 수행합니다.

1. **태스크 영역**에서 **모델 관리**, **스테이지 테이블 임포트** 순으로 선택합니다.
[임포트] 페이지가 표시됩니다.
2. 삭제할 구성을 선택하고 **임포트 구성 삭제** 버튼()을 누릅니다.
삭제 여부를 묻는 메시지가 나타납니다.
3. **예**를 누릅니다.
구성이 삭제됩니다.

임포트 구성 실행

임포트 구성을 생성한 후에는 해당 구성을 사용하여 애플리케이션으로 데이터를 가져올 수 있습니다.

임포트 구성을 실행하려면 다음을 수행합니다.

1. 준비 테이블이 있는 관계형 데이터베이스에 대한 연결 정보가 있는지 확인합니다.
2. **태스크 영역**에서 **모델 관리**, **스테이지 테이블 임포트** 순으로 선택합니다.
[임포트] 페이지가 표시됩니다.
3. **임포트 구성 목록**에서 실행하려는 임포트 구성을 선택합니다.
4. **임포트 구성 실행** 버튼()을 누릅니다.
[데이터 임포트] 대화상자가 표시됩니다. 모든 데이터 연결 세부정보가 나열됩니다.
5. **다음**을 누릅니다.
6. 구성 아래에서 대상 데이터베이스와 선택된 준비 테이블이 올바른지 확인합니다.
7. **다음**을 누릅니다.
[구성 요약]이 표시됩니다. [구성 이름] 및 연결 URI가 식별됩니다.
8. 다음 작업 중 하나를 선택합니다.
 - **지금 실행**을 눌러 즉시 임포트를 실행합니다.
 - **나중에 실행**을 눌러 편리한 날짜 및 시간으로 임포트 실행을 예약합니다.
9. **완료**를 누릅니다.
 - **지금 실행**을 선택한 경우 임포트가 실행되고 대상 애플리케이션이 선택된 데이터로 채워집니다.

- **나중에 실행**을 선택한 경우 작업이 저장됩니다. 저장된 임포트 작업을 실행하려면 [작업 프로세스], [태스크 플로우 관리] 순으로 선택합니다. 이 화면에서 작업을 실행하거나 편리한 날짜 및 시간으로 작업 실행을 예약할 수 있습니다. 자세한 지침은 *Oracle Hyperion Profitability and Cost Management 사용자 가이드*를 참조하십시오.
10. 임포트가 완료되면 가져온 데이터를 검증합니다. **임포트된 데이터 확인**을 참조하십시오.

임포트된 데이터 확인

데이터를 대상 데이터베이스로 가져온 후에는 해당 데이터를 제대로 완전하게 가져왔는지 확인해야 합니다.

가져온 데이터를 확인하려면 다음을 수행합니다.

1. Oracle Hyperion Enterprise Performance Management Workspace에서 **탐색, 애플리케이션, Profitability**, 애플리케이션 이름 순으로 선택하여 Oracle Hyperion Profitability and Cost Management 모델에 액세스합니다.
2. 대상 애플리케이션에서 모델을 열고 가져올 예상 데이터를 검토합니다.
예를 들어 단계를 임포트한 경우 **모델 관리, 단계** 순으로 선택하여 모든 단계에 대한 정보가 존재하며 올바른지 확인합니다.
3. 가져온 준비 테이블의 [예외] 열을 검토하여 오류나 경고가 있는지 확인합니다.
4. 소스 준비 테이블의 오류를 수정한 다음 임포트를 다시 실행합니다.
임포트 중 오류가 생성되지 않을 때까지 필요에 따라 2단계에서 3단계를 반복합니다.

C

EPM System Lifecycle Management를 사용한 데이터 마이그레이션

Lifecycle Management 정보

Oracle Hyperion Enterprise Performance Management System Lifecycle Management는 Oracle Enterprise Performance Management System 제품에서 애플리케이션, 저장소 또는 개별 아티팩트를 제품 환경과 운영 체제 간에 마이그레이션하기 위한 일관된 방법을 제공합니다.

Lifecycle Management 기능을 사용하여 다음 태스크를 수행할 수 있습니다.

- 애플리케이션 및 폴더 보기
- 아티팩트 검색
- 한 애플리케이션에서 다른 애플리케이션으로 직접 마이그레이션
- 파일 시스템 간 마이그레이션
- 마이그레이션 정의 파일 저장 및 로드
- 선택한 아티팩트 보기
- 마이그레이션 감사
- 마이그레이션 상태 보기
- 파일 시스템의 신속한 변경을 위해 개별 아티팩트 импорт 및 익스포트

일반적으로 Oracle Hyperion Shared Services Console의 Lifecycle Management 인터페이스는 Lifecycle Management를 지원하는 모든 EPM System 제품에 대해 일관됩니다. 하지만 EPM System 제품은 Lifecycle Management 인터페이스에 다른 아티팩트 목록과 익스포트 및 импорт 옵션을 표시합니다.

Shared Services Console에 Lifecycle Management 인터페이스를 제공하는 것 외에도 아티팩트를 소스에서 대상으로 마이그레이션하는 또 다른 방법을 제공하는 Lifecycle Management 유틸리티라는 명령행 유틸리티가 있습니다. Lifecycle Management 유틸리티는 Windows 태스크 스케줄러 또는 Oracle Enterprise Manager와 같은 타사 스케줄링 서비스와 함께 사용할 수 있습니다.

마지막으로, 사용자가 Lifecycle Management 기능을 사용자정의하고 확장할 수 있도록 해주는 Lifecycle Management API(Application Programming Interface)가 있습니다.

상세 Profitability의 Lifecycle Management 및 Oracle Hyperion Profitability and Cost Management 아티팩트에 대한 자세한 내용은 *Oracle Enterprise Performance Management System Lifecycle Management 가이드*를 참조하십시오.

관리 원장의 ApplicationData 및 Dimensions 아티팩트

관리 원장 애플리케이션의 경우 LCM 익스포트에는 다음 아티팩트가 포함된 메타데이터 폴더가 있습니다.

- ApplicationData
- 차원

이러한 아티팩트는 각각 다음 폴더 구조에 있습니다.

- resource\Metadata\ApplicationData\ - 이 위치에는 하나의 파일만 있음
- resource\Metadata\Dimensions\ - 차원마다 하나의 파일이 있음

예:

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

ApplicationData 아티팩트 유형은 나중에 사용하기 위한 것이며 현재 익스포트하거나 다시 임포트하는 데 사용할 수 없습니다. 차원을 익스포트하거나 임포트하는 경우 LCM 익스포트 작업을 올바르게 수행하려면 ApplicationData를 선택 취소해야 합니다.

차원을 임포트하려면 먼저 Profitability 애플리케이션 콘솔에 비어 있는 새 애플리케이션을 생성해야 합니다. 사용으로 설정되지 않은 빈 애플리케이션으로만 차원을 임포트할 수 있습니다.



주:

이 아티팩트의 경우 아티팩트 유형이 아티팩트 이름과 동일합니다.

Lifecycle Management에 대한 기본 시간 초과 설정 수정

Oracle Hyperion Enterprise Performance Management System Lifecycle Management를 사용하여 크기가 큰 모델을 가져오는 경우 임포트 처리에 걸리는 시간이 Oracle WebLogic Server의 기본 시간 초과 설정에 지정된 시간보다 길어질 수 있습니다.

이 이슈를 해결하려면 기본 시간 초과 설정을 재설정해야 합니다.

WebLogic Server에 대한 기본 시간 초과를 수정하려면 다음을 수행합니다.

1. 다음 위치로 이동합니다.

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

2. 구성 파일에서 **LocationMatch/profitability** 섹션을 찾습니다.

```
<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. LocationMatch/profitability 섹션에 다음 라인을 추가합니다.

```
WLIOTimeoutSecs 3000  
Idempotent OFF
```

4. 다음 위치로 이동합니다.

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

5. 다음 텍스트에 표시된 대로 시간 초과를 **3000**으로 설정합니다.

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

 주:

위에 표시된 서버 시간 초과는 제안된 제한이며, 해당 애플리케이션 서버의 설명서에 제공된 특정 시간 초과 설정에 맞게 수정할 수 있습니다.

D

Profitability and Cost Management 구성요소 백업

다음 몇 가지 Oracle Hyperion Profitability and Cost Management 데이터베이스는 데이터의 안전과 무결성을 보장하기 위해 정기적으로 백업해야 합니다.

- 운용 데이터 저장
- 스테이지 테이블 임포트
- 블록 저장영역 옵션(BSO) 데이터베이스

주:

애플리케이션 유형에 따라 일부 데이터베이스를 사용하지 않을 수도 있습니다.

백업 빈도는 데이터베이스의 변경량과 해당 조직의 일반적인 표준에 따라 결정됩니다.

표 1에는 백업해야 하는 데이터베이스, 해당 유형 및 제안되는 백업 옵션이 나열되어 있습니다.

주의:

현재 LCM(Lifecycle Management)을 사용하여 임포트 준비 영역이나 운용 데이터 저장을 백업할 수 없습니다.

표 D-1 Profitability and Cost Management 데이터베이스 백업 요구사항

데이터베이스	데이터베이스 유형	백업 옵션
스테이지 영역 임포트	관계형 데이터베이스(RDB)	다음 옵션을 비롯하여 RDB에 대한 표준 백업 기술을 사용할 수 있습니다. <ul style="list-style-type: none">• 데이터베이스 스크립팅• 스케줄러 스크립트• 백업 도구(예: TOAD)• SQL Server 또는 Oracle의 백업 절차
운용 데이터 저장	관계형 데이터베이스(RDB)	다음 옵션을 비롯하여 RDB에 대한 표준 백업 기술을 사용할 수 있습니다. <ul style="list-style-type: none">• 데이터베이스 스크립팅• 스케줄러 스크립트• 백업 도구(예: TOAD)• SQL Server 또는 Oracle의 백업 절차

표 D-1 (계속) Profitability and Cost Management 데이터베이스 백업 요구사항

데이터베이스	데이터베이스 유형	백업 옵션
상세 Profitability에 대한 모델 데이터 스키마	관계형 데이터베이스(RDB)	<p>다음 옵션을 비롯하여 RDB에 대한 표준 백업 기술을 사용할 수 있습니다.</p> <ul style="list-style-type: none"> • 데이터베이스 스크리핑 • 스케줄러 스크립트 • 백업 도구(예: TOAD) • SQL Server 또는 Oracle의 백업 절차

자세한 백업 및 복구 절차는 *Oracle Enterprise Performance Management System 백업 및 복구 가이드*를 참조하십시오.

E

Essbase 이름 지정 규칙

생성된 계산 스크립트 이름 지정 규칙

Oracle Hyperion Profitability and Cost Management에서 Oracle Essbase 스크립트를 생성하면 특정 규칙을 사용하여 해당 스크립트에 자동으로 이름이 지정됩니다.

드라이버 계산 스크립트 및 할당 계산 스크립트는 단계별로 하나의 스크립트로 결합됩니다. 스크립트 이름은 다음과 같은 형식으로 생성됩니다.

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

- 스크립트 접미어는 스크립트 유형을 기반으로 합니다. 다음 표는 접미어 목록을 표시합니다.

표 E-1 계산 스크립트 접미어

스크립트 유형	비용 계층	수익 계층
단계 간 할당	"a"	"r"
단계 내 할당	"i"	"t"

- POV 식별자는 POV ID를 기반으로 하며 최대 3자리까지 포함할 수 있습니다. 각 POV에 대해 스크립트가 생성 및 식별됩니다.
- 단계 정렬 번호는 소스 단계의 정렬 번호(예: 1, 2, 3 등)입니다.
- 스크립트 분할로 인해 여러 스크립트가 생성되는 경우 `_index`는 동일한 유형, POV, 단계 및 계층에 대해 001, 002 등으로 시작하는 스크립트의 숫자 시퀀스를 표시합니다.

샘플 스크립트 이름

- `a3682001.csc`는 POV 식별자 368, 소스 단계 2 및 비용 계층에 대한 계산 스크립트를 나타냅니다.
- `t4533002.csc`는 POV 식별자 453, 소스 단계 3 및 수익 계층에 대한 단계 내 계산 스크립트를 나타냅니다.

애플리케이션 및 데이터베이스에 대한 Essbase 이름 지정 제한 사항

애플리케이션 및 데이터베이스에 대한 이름 생성 시 원하는 단어가 표시되는 경우에는 해당 이름을 입력합니다.

애플리케이션 및 데이터베이스 이름은 입력하는 그대로 생성됩니다. 이름을 모두 대문자(예: NEWAPP)로 입력하면 Oracle Essbase에서 이 이름을 Newapp와 같이 대-소문자 조합으로 자동 변환하지 않습니다.

 주:

이 목록은 부분적인 제한 사항을 제공합니다. 전체 제한 사항 목록을 보려면 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

애플리케이션 및 데이터베이스 이름 지정 시 다음과 같은 이름 지정 제한 사항이 적용됩니다.

- 8자 이하를 사용합니다.
- 이름에 공백을 사용하지 마십시오.
- 이름에 다음 특수 문자를 사용하지 마십시오.
 - * 별표
 - + 더하기 기호
 - \ 백슬래시
 - ? 물음표
 - [] 대괄호
 - " 큰따옴표
 - : 콜론
 - ; 세미콜론
 - , 쉼표
 - ' 작은따옴표
 - = 등호
 - / 슬래시
 - > 보다 큼 기호
 - 탭
 - < 보다 작음 기호
 - | 세로 막대
 - . 마침표
- 합산 저장영역 데이터베이스의 경우에만 애플리케이션이나 데이터베이스 이름으로 다음 단어를 사용하지 마십시오.
 - DEFAULT
 - LOG
 - METADATA
 - TEMP

차원, 멤버 및 별칭에 대한 Essbase 이름 지정 제한

차원 아웃라인을 정의하는 경우 차원, 멤버 및 별칭 이름 지정 시 사용할 수 없는 문자 제한이 존재합니다.

가장 일반적인 제한 문자 목록은 다음 섹션에 나와 있습니다. 하지만 *Oracle Essbase Database Administrator's Guide*에 설명된 Essbase 이름 지정 규칙에서 전체 목록을 검토하는 것이 좋습니다.

주:

이 목록은 부분적인 제한 사항을 제공합니다. 전체 제한 사항 목록을 보려면 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.

차원, 멤버 및 별칭에 대한 이름을 지정할 때는 다음과 같은 이름 지정 제한을 준수합니다.

- 80자 이하를 사용합니다.
- 대소문자 구분이 활성화된 경우 대문자와 소문자를 구분합니다. 대소문자 구분을 사용으로 설정하려면 *Oracle Essbase Database Administrator's Guide*에서 "아웃라인 등록정보 설정"을 참조하십시오.
- 차원 또는 멤버 이름, 별칭 및 설명에는 HTML 태그를 사용하지 마십시오.
- 이름에는 따옴표, 마침표, 대괄호, 백슬래시 또는 탭을 사용하지 마십시오.

주의:

대괄호는 허용되지만 블록 저장영역 아웃라인에 사용하지 않는 것이 좋습니다. 합산 저장영역 아웃라인으로 변환할 때 오류가 발생합니다.

- 동일한 차원 내에서는 중복 멤버 이름 또는 별칭을 사용할 수 없습니다.
- 다음 단어로 차원 또는 멤버 이름을 시작하지 마십시오.
 - at 기호(@)
 - 백슬래시(\)
 - 대괄호([])
 - 쉼표(,)
 - 대시
 - 하이픈
 - 빼기 기호(-)
 - 등호(=)
 - 보다 작음 기호(<)
 - 괄호()
 - 마침표(.)

- 더하기 기호(+)
- 작은 따옴표(')
- 따옴표("")
- 밑줄(_)
- 세로 막대(|)
- 이름의 맨 앞 또는 맨 뒤에 공백을 포함하지 마십시오. Oracle Essbase에서는 이러한 공백을 무시합니다.
- 멤버 이름에 슬래시를 사용하지 마십시오.
- 사용자정의 달력의 기간인 경우 접두어에 공백을 사용하지 마십시오.
- 다음 항목을 차원 또는 멤버 이름으로 사용하지 마십시오.
 - 계산 스크립트 명령, 연산자 및 키워드. 전체 명령 목록을 보려면 *Oracle Essbase Database Administrator's Guide*를 참조하십시오.
 - 보고서 작성기 명령
 - 함수 이름 및 함수 인수
 - [동적 시계열]이 사용된 경우 [내역], [연도], [계절], [기간], [분기], [월], [주] 또는 [일]을 사용하지 않습니다.
 - 기타 차원 및 멤버 이름(멤버를 공유하지 않을 경우), 계보 이름, 레벨 이름 및 데이터베이스의 별칭
- 다음 단어를 사용하지 마십시오.
 - 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

속성 계산 차원에 대한 Essbase 이름 지정 규칙

Oracle Essbase에서 속성 계산 차원에 생성하는 멤버(Sum, Count, Min, Max 및 Avg)는 예약어로 간주되지 않습니다.

속성 계산 차원에서 이러한 이름을 변경한 후 속성 또는 표준 차원에서 표준 이름을 사용할 수 있기 때문입니다.

아웃라인이 고유한 멤버 아웃라인으로 태그가 붙은 경우 멤버 이름에 Sum, Count, Min, Max 및 Avg를 사용하지 마십시오. 예를 들어 표준 차원에서 Max를 사용한 다음 속성 차원을 생성하면(Essbase에서 속성 계산 차원에 Max 멤버 생성) Essbase에서 중복 이름을 감지한 후 다음과 같은 오류 메시지를 반환합니다.

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

아웃라인이 중복 멤버 아웃라인으로 태그가 지정되었으며 Sum, Count, Min, Max 및 Avg를 기본 멤버로 사용하기 전에 속성 차원(및 속성 계산 차원)이 있는 경우 Essbase에서 중복 이름을 허용합니다. 그러나 속성 차원을 생성하기 전에 Sum, Count, Min, Max 및 Avg를 기본 멤버로 사용하는 경우에는 중복 이름이 허용되지 않습니다.