

intra-mart WebPlatform/AppFramework Ver.7.2

アプリケーション共通マスタ仕様書

2010/05/31 第2版

<< 変更履歴 >>

変更年月日	変更内容
2010/04/01	初版
2010/05/31	第 2 版 「3.3.3 リスナの設定」を追記しました。

<< 目次 >>

1	アプリケーション共通マスタについて	1
1.1	はじめに	1
1.2	アプリケーション共通マスタ	2
1.2.1	目的	2
1.2.2	統合化の範囲と要件	2
1.2.3	データ構造	2
2	構造	9
2.1	ユーザ	9
2.1.1	データ構造	9
2.1.2	プロフィールとアカウント	10
2.2	分類	10
2.2.1	データ構造	10
2.2.2	初期登録時の分類の内容	12
2.3	会社と組織	19
2.3.1	データ構造	19
2.3.2	会社および組織の情報	29
2.3.3	会社組織の構成	30
2.3.4	会社組織構成のバージョン管理	31
2.3.5	役職	34
2.3.6	会社組織への所属	34
2.3.7	分類	37
2.4	パブリックグループ	39
2.4.1	データ構造	39
2.4.2	パブリックグループセットおよびパブリックグループの情報	47
2.4.3	パブリックグループの構成	48
2.4.4	パブリックグループ構成のバージョン管理	49
2.4.5	パブリックグループへの所属	52
2.4.6	分類	53
2.5	プライベートグループ	54
2.5.1	データ構造	54
2.5.2	プライベートグループの情報	56
2.5.3	プライベートグループへの所属	57
2.6	商品	58
2.6.1	データ構造	58
2.6.2	商品の情報	73
2.6.3	商品属性テンプレート	73
2.6.4	商品の分類	74
2.6.5	商品の取扱	76
3	API	80
3.1	アプリケーション共通マスタとアプリケーションデータの整合性について	80
3.1.1	アプリケーション共通マスタ間の整合性の考慮は不要	80
3.1.2	テーブルの拡張に対して同期を取ることが可能	80
3.2	マネージャ	81
3.2.1	マネージャの取得	83
3.2.2	マネージャの使用	84
3.2.3	マネージャによる検索	86

3.2.4	マネージャによる拡張情報の操作.....	88
3.3	リスナ.....	90
3.3.1	リスナの動作.....	90
3.3.2	リスナの種類.....	91
3.3.3	リスナの設定.....	95
4	検索例.....	96
4.1	ある組織の詳細情報を取得する場合.....	96
4.2	ある会社以下の全ての組織を取得する.....	97
4.3	ある組織に所属するユーザを取得する.....	98

1 アプリケーション共通マスタについて

1.1 はじめに

ベースモジュール ver2.1 より、アプリケーション共通マスタ unit が追加されました（ベースモジュール ver2.0X にあった汎用マスタが機能アップされています）。

また、intra-mart 5.0 以降ではデータベースの構造を見直すことにより、国際化や期間化といった新しい概念が追加され機能が増えました。

このアプリケーション共通マスタの中で、会社データ、組織データ、グループデータ、取引先データ、顧客データ、商品データなどシステム開発でよく利用するマスタが標準で用意されています。このマスタを利用することで、より短期間でのシステム開発が可能になると同時に、各 intra-mart アプリケーションシリーズ（スタートパックや営業支援システム、Web 受発注システムなど）と連携したシステムが開発できます。

これらアプリケーション共通マスタの利用方法については、アドミニストレータガイド第2編 ログイングループ管理者編の「8 アプリケーション共通マスタ」をご覧ください。

本資料では、intra-mart 上で動作するアプリケーション共通マスタの仕組みおよびデータベース構造を解説するとともに、im-BizAPI を利用したプログラミング方法等について記述しています。

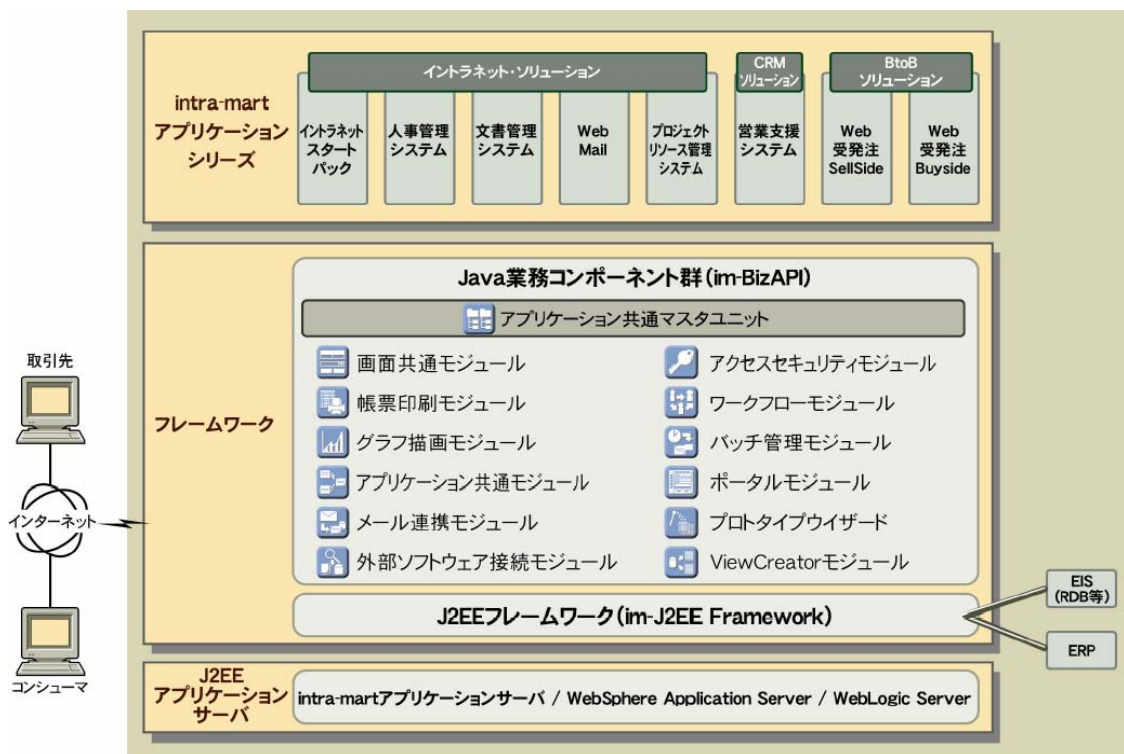


図 1-1 intra-mart

1.2 アプリケーション共通マスタ

1.2.1 目的

会社・組織・取引先・ユーザマスタを統合化することで、イントラマートユーザのマスタメンテナンスにかかる負担を軽減するとともに、イントラマートシステム全体を統合的に構築できる環境を整備します。

1.2.2 統合化の範囲と要件

アプリケーション共通マスタにはさまざまな要件を実現するために設計・実装されています。その内容を「表 1-1 統合化の範囲と要件」に示します。

表 1-1 統合化の範囲と要件

No.	要件	方法
1	組織やパブリックグループを指定階層以下の集合として扱うことができること。 (例:ある組織以下の社員すべてを検索したい)	内包型のデータ構造で実現する。
2	画面の性格によっては自社のみを表示したり、顧客一覧を抽出したりしたい。	「分類区分」、「分類詳細」、「分類」の各テーブルを使用して実現する。
3	多数の取引先が混在する環境で、互いの可視範囲を設定したい。	「分類区分」、「分類詳細」、「分類」の各テーブルを使用して実現する。
4	多数の取引先が混在する環境で、販売チャンネル毎に顧客一覧抽出したい。	「分類区分」、「分類詳細」、「分類」の各テーブルを使用して実現する。
5	会社やパブリックグループの階層構成のバージョン管理を行いたい。	「会社バージョン」、「パブリックグループセットバージョン」の各テーブルを使用して実現する。
6	ユーザ名や組織名などが変更された場合でも過去の情報を残したい、または変更される予定の情報をあらかじめ登録しておきたい。	対象となるテーブルに対する期間化されたテーブルを使用して実現する。
7	組織名などの表示名を言語によって変更したい。	対象となるテーブルに対する国際化されたテーブルを使用して実現する。
8	商品の情報(名称、単価など)を期間および言語により変更したい。	「商品」に対して期間化、国際化されたテーブルを使用して実現する。
9	商品によって異なる属性情報を保持したい。	「商品属性テンプレート」テーブルを使用して実現する。
10	商品を複数の取引先に対して扱いたい。またいくつかの商品をセットで取扱いたい。	「取扱」テーブルを使用することにより実現する。
11	複数の異なる視点から商品を検索したい。	ツリー構造のカテゴリを作成することにより実現する。

1.2.3 データ構造

アプリケーション共通マスタはさまざまなテーブルから構成されています。これらのテーブルの関連概要を「図 1-2 データ構成(全体)」に示します。

1.2.3.1 期間化と国際化

「図 1-2 データ構成(全体)」で示した図のエンティティは期間化や国際化されているものもあります。

期間化されているテーブルでは、現在のデータだけではなく過去や未来のデータも保存することができるようになります。これにより「昔の帳票を出力したい」、「新しい組織情報をあらかじめ登録しておきたい」といったような要件に対応することができます。

国際化されているテーブルでは、各言語に対応したデータを持つことができるようになります。これにより「ログインしているユーザが使用している言語で組織名を表示する」といったような要件に対応することができます。

1.2.3.2 テーブル構造

「図 1-2 データ構成(全体)」で示したエンティティはあくまで概念的なものであり、実際のテーブル構造は期間化や国際化によってその構造に影響を受けます。

ER図中に出てくるエンティティは、「図 1-3 テーブル構造」で示されるようにデータベース上では最大 4 つのテーブルで管理されています。

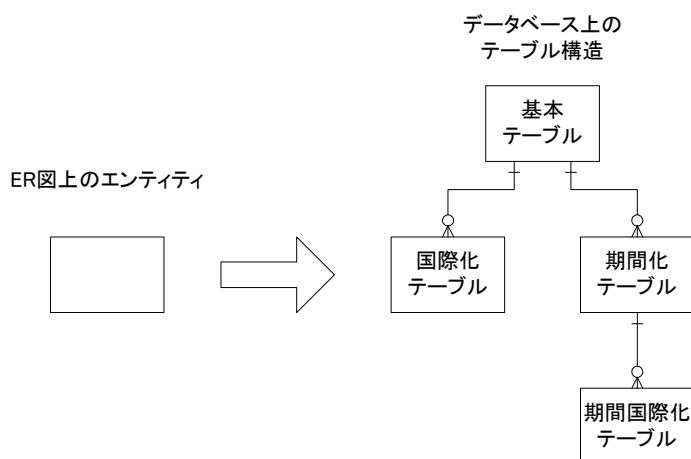


図 1-3 テーブル構造

「図 1-3 テーブル構造」の 4 つのテーブルはそれぞれ以下のような役割を持ちます。

- 基本テーブル
期間化も国際化もする必要がない情報(時間や言語に依存しない情報)が管理されます。
- 国際化テーブル
国際化だけ行う必要がある情報(言語に依存するが、時間には依存しない情報)が管理されます。
- 期間化テーブル
期間化だけ行う必要がある情報(時間に依存するが、言語には依存しない情報)が管理されます。
- 期間国際化テーブル
期間化も国際化も行う必要がある情報(時間や言語によって変更される情報)が管理されます。

1.2.3.2.1 基本テーブル

基本テーブルでは期間化も国際化もする必要がない情報(時間や言語に依存しない情報)が管理されます。基本テーブルは以下のようなカラムを持ちます。

- エンティティのプライマリキー
データを一意に識別するキーです。

- 国際化も期間化もされていない項目
時間や言語に依存しない情報を表現する項目です。

1つのエンティティに対して必ず1つの基本テーブルが存在します。

1.2.3.2.2 国際化テーブル

国際化テーブルでは国際化だけ行う必要がある情報(言語に依存するが、時間には依存しない情報)が管理されます。国際化テーブルは以下のようなカラムを持ちます。

- エンティティのプライマリーキー
基本テーブルのプライマリーキーと一致する必要があります。
- ロケール ID
どの言語に対する情報かを指定します。intra-martではlocale_idというカラム名を使用しています。指定できる値はjava.util.Localeクラスで表現できるものだけです。
- 国際化のみされている項目
言語に依存するが、時間には依存しない情報を表現する項目です。

国際化テーブルのプライマリーキーは以下のカラムの組合せとなります。

- エンティティのプライマリーキー
- ロケール ID

エンティティに国際化のみされた項目がない場合、国際化テーブルは必要ありません。

1.2.3.2.3 期間化テーブル

期間化テーブルでは期間化だけ行う必要がある情報(時間に依存するが、言語には依存しない情報)が管理されます。期間化テーブルは以下のようなカラムを持ちます。

- エンティティのプライマリーキー
基本テーブルのプライマリーキーと一致する必要があります。
- 期間コード
どの言語に対する情報かを指定します。intra-martではterm_cdというカラム名を使用しています。エンティティのプライマリーキーに対して一意となるものを割り当てる必要があります。
- 有効期間開始日時
この情報が有効である期間の開始日時を指定します。intra-martではstart_dateというカラム名を使用しています。
- 有効期間終了日時
この情報が有効である期間の終了日時を指定します。intra-martではend_dateというカラム名を使用しています。
- 期間化のみされている項目
時間に依存するが、言語には依存しない情報を表現する項目です。

期間化テーブルのプライマリーキーは以下のカラムの組合せとなります。

- エンティティのプライマリーキー
- 期間コード

また、この情報が有効である期間は以下の式で表現されます。

$$\text{有効期間開始日時} \leq \text{有効期間} < \text{有効期間終了日時}$$

たとえば、有効期間開始日時が2005/10/01、有効期間終了日が2006/04/01となっている場合、「表 1-2 有効期間」のように扱われます。

表 1-2 有効期間

対象日時	有効/無効
2005/09/30 23:59:59	無効
2005/10/01 00:00:00	有効
2006/01/01 12:00:00	有効
2006/03/31 23:59:59	有効
2006/04/01 00:00:00	無効

エンティティに期間化のみされた項目がない場合、期間化テーブルは必要ありません。ただし、「1.2.3.2.4 期間国際化テーブル」で述べているテーブルを作成する必要がある場合、期間化テーブルは必須です。

1.2.3.2.4 期間国際化テーブル

期間国際化テーブルでは期間化も国際化も行う必要がある情報(時間や言語によって変更される情報)が管理されます。期間国際化テーブルは以下のようなカラムを持ちます。

- エンティティのプライマリキー
基本テーブルのプライマリキーと一致する必要があります。
- 期間コード
期間化テーブルの期間コードと一致する必要があります。
- ロケール ID
どの言語に対する情報かを指定します。intra-mart では locale_id というカラム名を使用しています。指定できる値は java.util.Locale クラスで表現できるものだけです。
- 期間化も国際化もされている項目
時間や言語によって変更される情報を表現する項目です。

期間国際化テーブルのプライマリキーは以下のカラムの組合せとなります。

- エンティティのプライマリキー
- 期間コード
- ロケール ID

エンティティに期間化も国際化もする必要がある項目がない場合、期間国際化テーブルは必要ありません。

1.2.3.2.5 エンティティとテーブルの例

ここではエンティティとそれに対応するテーブルの例として顧客情報を取り上げます。ここで説明する顧客は以下のような情報を持っています。

- 顧客コード
顧客を一意に識別するコードです。
- 名前
顧客名です。顧客名は言語によって表記が変わり、ある時点で変更される可能性もあります。
- 誕生日
顧客の誕生日です。
- 出身地
顧客の出身地の表記です。さまざまな言語で表記される場合があります。
- e-mail
顧客のメールアドレスです。ある時点で変更される可能性があります。

これらのことから、顧客情報は期間化も国際化もされることがわかります。この内容をエンティティとして表現すると「図 1-4 エンティティの例(顧客)」のようになります。

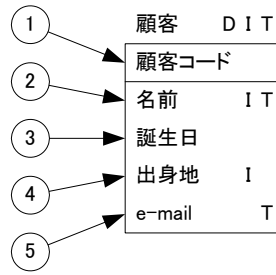


図 1-4 エンティティの例(顧客)

「図 1-4 エンティティの例(顧客)」の表記法の詳細については、アプリケーション共通マスタAPI方針の「2.1.4 エンティティに表記法」および「付録A intra-mart拡張ERD表記法一覧」をご覧ください。

「図 1-4 エンティティの例(顧客)」のエンティティをテーブルに変換すると「図 1-5 テーブルの例(顧客)」のようになります。

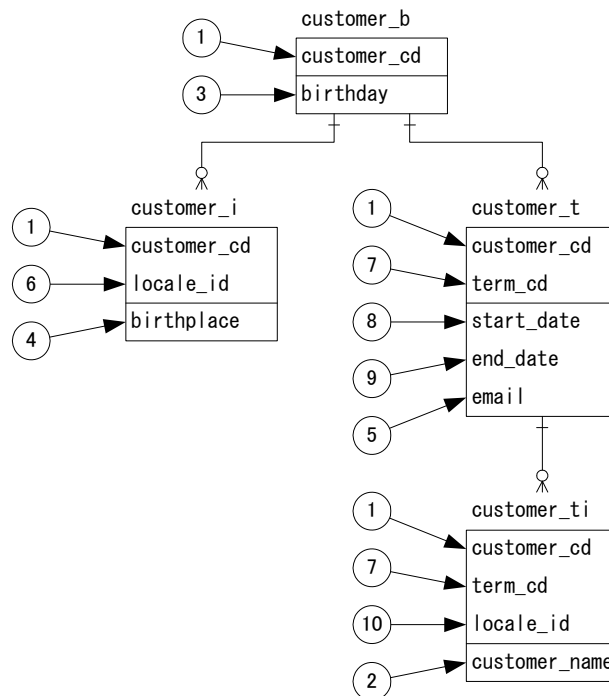


図 1-5 テーブルの例(顧客)

「図 1-4 エンティティの例(顧客)」と「図 1-5 テーブルの例(顧客)」の各数字はそれぞれ「表 1-3 エンティティとテーブルの対応(例)」のような意味を持ちます。

表 1-3 エンティティとテーブルの対応(例)

番号	意味
1	エンティティのプライマリキー
2	期間化も国際化もされている項目
3	期間化も国際化もされていない項目
4	国際化のみされている項目
5	期間化のみされている項目
6	ロケール ID(国際化テーブル用)
7	期間コード
8	有効期間開始日時
9	有効期間終了日時
10	ロケール ID(期間国際化テーブル用)

2 構造

2.1 ユーザ

ユーザは intra-mart で扱う個人情報です。アプリケーション共通マスタではユーザをプロフィールと呼んでいます。プロフィールはアクセスセキュリティで管理されるアカウントとは異なります。同一人物である場合プロフィールとアカウントはいずれも同じ ID を持ちますが、プロフィールとアカウントのどちらか一方が存在しなくてもかまいません。その場合、そのユーザはログインのみ可能なユーザ、または登録されているがログインできないユーザといったような状態になります。これは、EC 関連アプリケーションでログインの必要ない個人情報を大量に扱う可能性を考慮したものです。ログインアカウントの必要なユーザのみ intra-mart のアクセスセキュリティで管理しているログインアカウントにも情報が保存されます。

2.1.1 データ構造

ユーザのエンティティと実際のテーブルの構造を「表 2-1 ユーザのデータ構造」と「図 2-1 ユーザテーブルの ER 図」に示します。

表 2-1 ユーザのデータ構造

項目	カラム名	PK	期間化	国際化	備考
ユーザコード	user_cd	○	—	—	
アカウントフラグ	account_flag	—	×	×	ログインユーザアカウントが存在する場合は "1"、存在しなければ "0"
ユーザ名	user_name	—	○	○	
ユーザ名カナ	user_name_syllabary	—	○	○	
ユーザ名英字	user_name_eng	—	○	×	
電話番号	telephone_number	—	○	×	
携帯電話番号	mobile_number	—	○	×	
FAX 番号	fax_number	—	○	×	
内線番号	extension_number	—	○	×	
内線 FAX 番号	extension_fax_number	—	○	×	
国コード	country_cd	—	○	×	
郵便番号	zip_code	—	○	×	
住所	address	—	○	○	
メールアドレス1	email_address1	—	○	×	
メールアドレス2	email_address2	—	○	×	
携帯用メール	mobile_email_address	—	○	×	
URL	url	—	○	×	
備考	notes	—	○	×	
ソートキー	sort_key	—	○	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

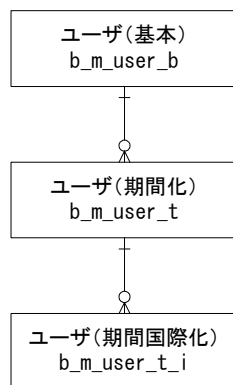


図 2-1 ユーザテーブルの ER 図

2.1.2 プロファイルとアカウント

intra-mart ではユーザを表現する情報としてプロファイルとアカウントがあります。プロファイルは組織への所属や人事情報など業務で扱われるユーザです。アカウントは主にログインやセキュリティなど、システムを対象としたユーザです。本書で説明しているユーザはプロファイルに該当します。

プロファイルとアカウントはそれらのユーザコードが一致する場合、同一のユーザであるとみなされます。一人のユーザがプロファイルとアカウントのいずれを持つのか、あるいは両方を持つのかは自由に設定することができます。

プロファイルが存在しているときに同一ユーザのアカウントも存在する場合、プロファイルのアカウントフラグ(「表 2-1 ユーザのデータ構造」の項目を参照してください)は"1"となります。プロファイルは存在していても同一ユーザのアカウントが存在しない場合、プロファイルのアカウントフラグは"0"となります。この項目はシステムで自動的に設定され、ユーザが独自に設定することはできません。

2.2 分類

「2.3 会社と組織」や「2.4 パブリックグループ」で説明するグループにはさまざまな角度で分類することができます。アプリケーション共通マスタでは次のような単位で分類することができます。

- 公開ゾーン
- 検索属性
- 販売チャネル

2.2.1 データ構造

分類に関連するER図を「図 2-2 分類関連のER図」に示します。

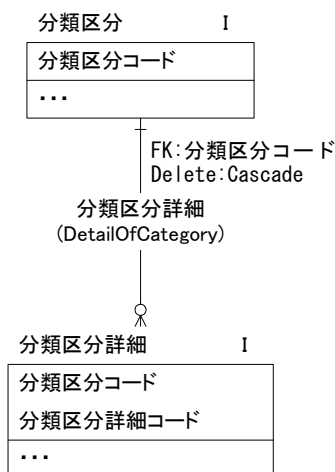


図 2-2 分類関連の ER 図

「図 2-2 分類関連のER図」において各エンティティは次のような役割があります。

- 分類区分エンティティ
 - ◆ 分類の種類(公開ゾーン、検索属性、販売チャネル等)を管理します。
- 分類区分詳細エンティティ
 - ◆ 分類の種類ごとに定義されている詳細(公開ゾーンにはイントラネットゾーン、エクストラネットゾーン、マーケットプレースが存在等)を管理します。

また、「図 2-2 分類関連のER図」から以下のような整合性チェックが必要であることが読み取れます。

- 分類区分エンティティのデータが削除された場合、
 - ◆ その分類区分の詳細情報である分類区分詳細も削除する必要があります。
- 分類区分詳細エンティティのデータを追加する場合、
 - ◆ 該当する分類区分が存在している必要があります。

2.2.1.1 分類区分

分類区分のエンティティと実際のテーブルの構造を「表 2-2 分類区分のデータ構造」と「図 2-3 分類区分テーブルのER図」に示します。

表 2-2 分類区分のデータ構造

項目	カラム名	PK	期間化	国際化	備考
分類区分コード	category_cls_cd	○	—	—	
分類区分名	category_cls_name	—	×	○	
備考	notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

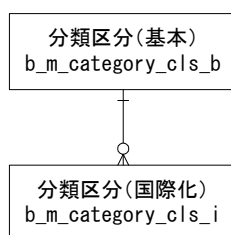


図 2-3 分類区分テーブルの ER 図

2.2.1.2 分類区分詳細

分類区分詳細のエンティティと実際のテーブルの構造を「表 2-3 分類区分詳細のデータ構造」と「図 2-4 分類区分詳細テーブルのER図」に示します。

表 2-3 分類区分詳細のデータ構造

項目	カラム名	PK	期間化	国際化	備考
分類区分コード	category_cls_cd	○	—	—	
分類コード	category_cd	○	—	—	
分類名	category_name	—	×	○	
備考	notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

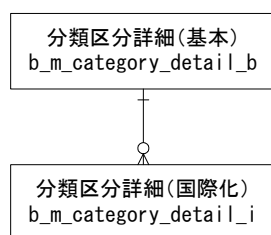


図 2-4 分類区分詳細テーブルの ER 図

2.2.2 初期登録時の分類の内容

アプリケーション共通マスタでは標準で以下の分類情報を用意しています。

- 公開ゾーン
組織等の公開範囲
- 検索属性
システムの中での組織の役割
- 販売チャネル
システムを運用する企業にとっての販売ルートを表現する概念

検索時にはこれらの条件を組み合わせで会社、組織およびパブリックグループ等を絞り込むことができます。

2.2.2.1 公開ゾーン

公開ゾーンは、アプリケーションの性格によって組織等の公開範囲を変更したい場合に使用します。例えば、「社内のスケジュール管理では全組織および全社員を表示するが、取引先も含めたコミュニティ APP では自社の一部の組織のみ公開したい」といった要件を実現するためのものです。

「図 2-5 公開ゾーンの例」に、最も簡単な公開ゾーンの例を示します。

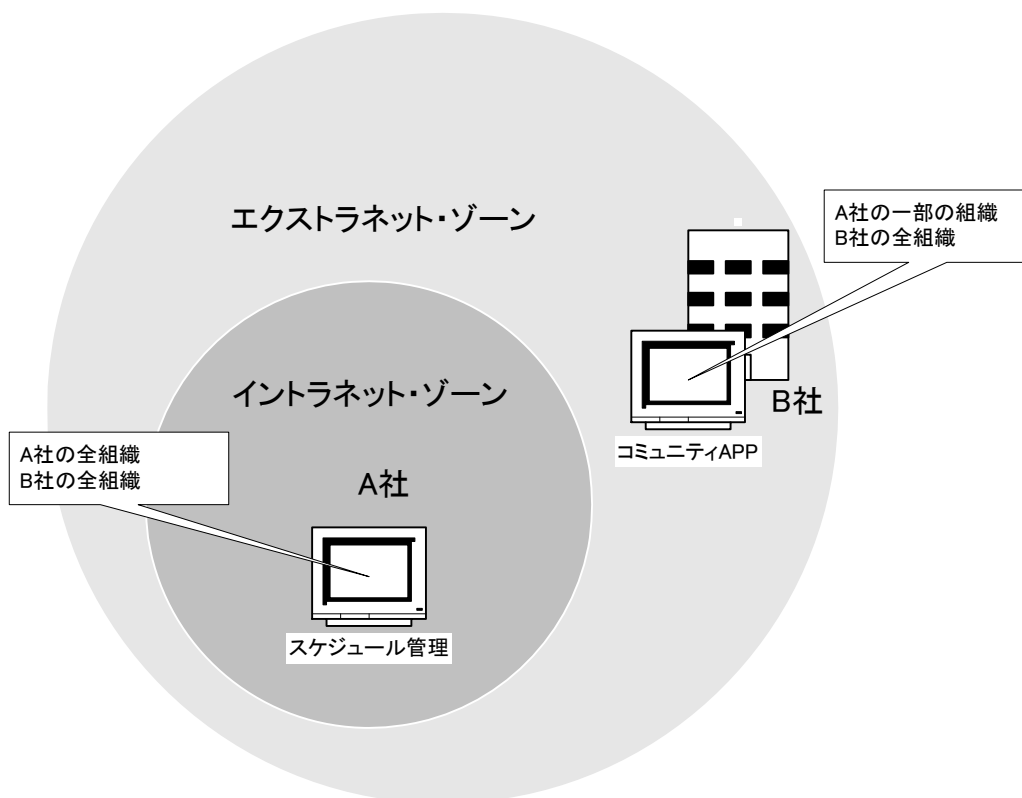
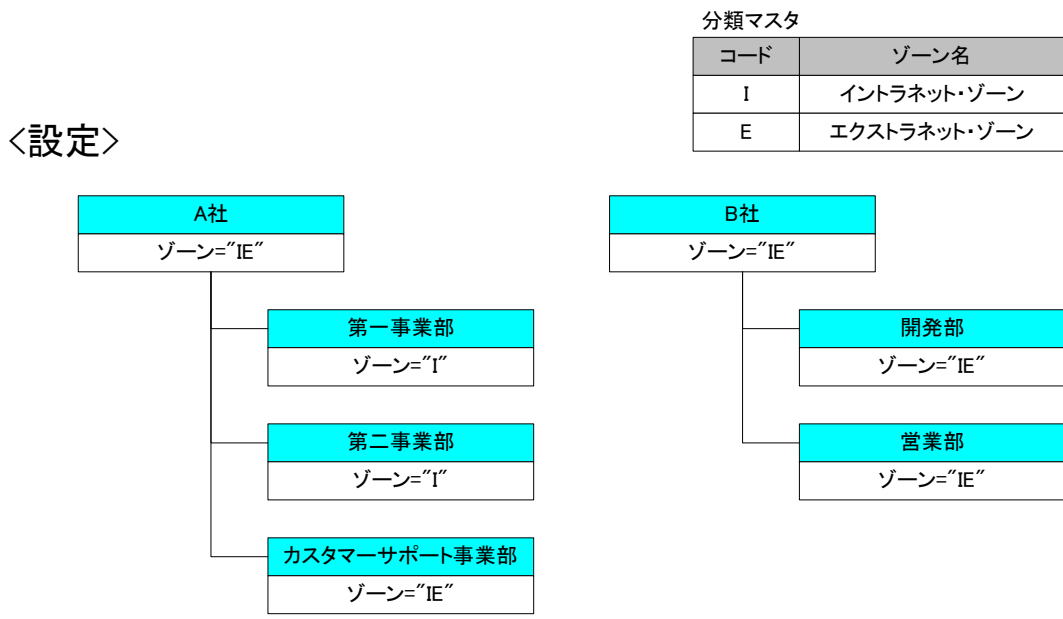


図 2-5 公開ゾーンの例

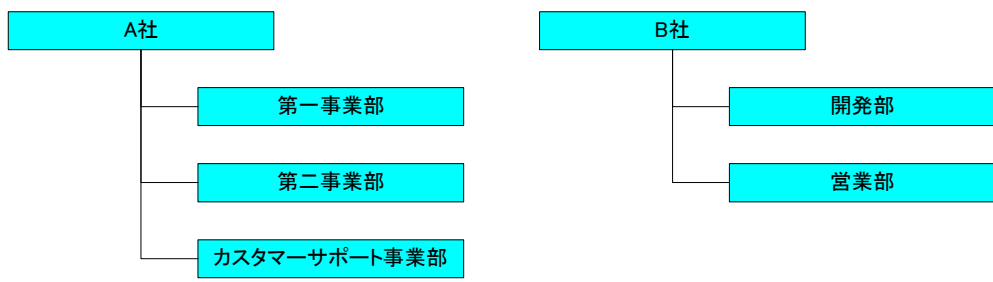
「図 2-5 公開ゾーンの例」では以下のような設定を行います。

- A社の全組織をイントラネット・ゾーンに公開する。
- A社の一部組織をエクストラネット・ゾーンに公開する。
- B社の全組織をエクストラネット・ゾーンに公開する。

各アプリケーションでは「図 2-6 公開ゾーンによるデータの抽出」のようなデータの抽出を行います。

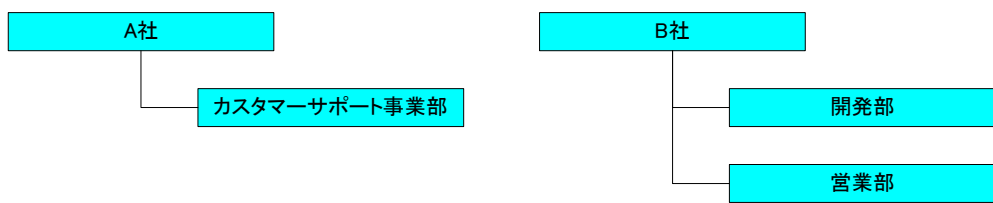


<スケジュール管理>
(イントラネット・ゾーンに公開されたデータのみ抽出)



*ゾーンに“I”が含まれるデータを抽出

<コミュニティAPP>
(エクストラネット・ゾーンに公開されたデータのみ抽出)



*ゾーンに“E”が含まれるデータを抽出

図 2-6 公開ゾーンによるデータの抽出

公開ゾーンの種類の種類は、分類区分・分類詳細マスタで定義します。登録されるデータとしては「表 2-4 公開ゾーン

の設定値]のようになります。

表 2-4 公開ゾーンの設定値

分類区分名	分類区分コード	分類名	分類コード
公開ゾーン	zone	イントラネットゾーン	I
		エクストラネットゾーン	E
		マーケットプレイスゾーン	M

公開ゾーンを使用する場合、以下のような点に注意する必要があります。

- 会社、組織、パブリックグループセットおよびパブリックグループについて設定することができます。プライベートグループについては設定できません。
- 公開ゾーンの設定はコーディネイトすることができますが、動的に公開ゾーンを設定することを想定していません。ある程度普遍的な集合について、公開ゾーンを設定してください。例えば2~3社がほぼ固定で、それぞれのイントラネットゾーンと共同のエクストラネットゾーンを設定することは可能ですが、ASP 事業でユーザ企業が無制限に増えることが前提であるにもかかわらずユーザ企業毎にゾーン設定するといった運用はできません
- 公開された組織内のデータ(所属ユーザ情報等)のセキュリティポリシーについては、各アプリケーションの仕様に依存します。
- 公開ゾーンはマスタメンテナンスによって変更または削除される可能性があるため、アプリケーションで使用するゾーンの分類コードはハードコーディングせず、設定ファイルにするなどの対策をする必要があります。

2.2.2.2 検索属性

検索属性は、システムの中での組織の役割を定義します。例えば、以下のような要件を実現するために使用します。

- この組織は顧客検索に表示したい
- この組織は協力会社一覧に表示したい

また、勤怠管理アプリケーション等のイントラネット型システムのユーザ検索画面等から、必要のない取引先の情報排除するためにも使用します。

検索属性は、公開ゾーンやその他の分類区分とは無関係(並列)に設定します。検索時は公開ゾーンや販売チャネルなどと組み合わせて抽出条件に加えることができます。

「図 2-7 検索属性の例」に簡単な検索属性の例を示します。

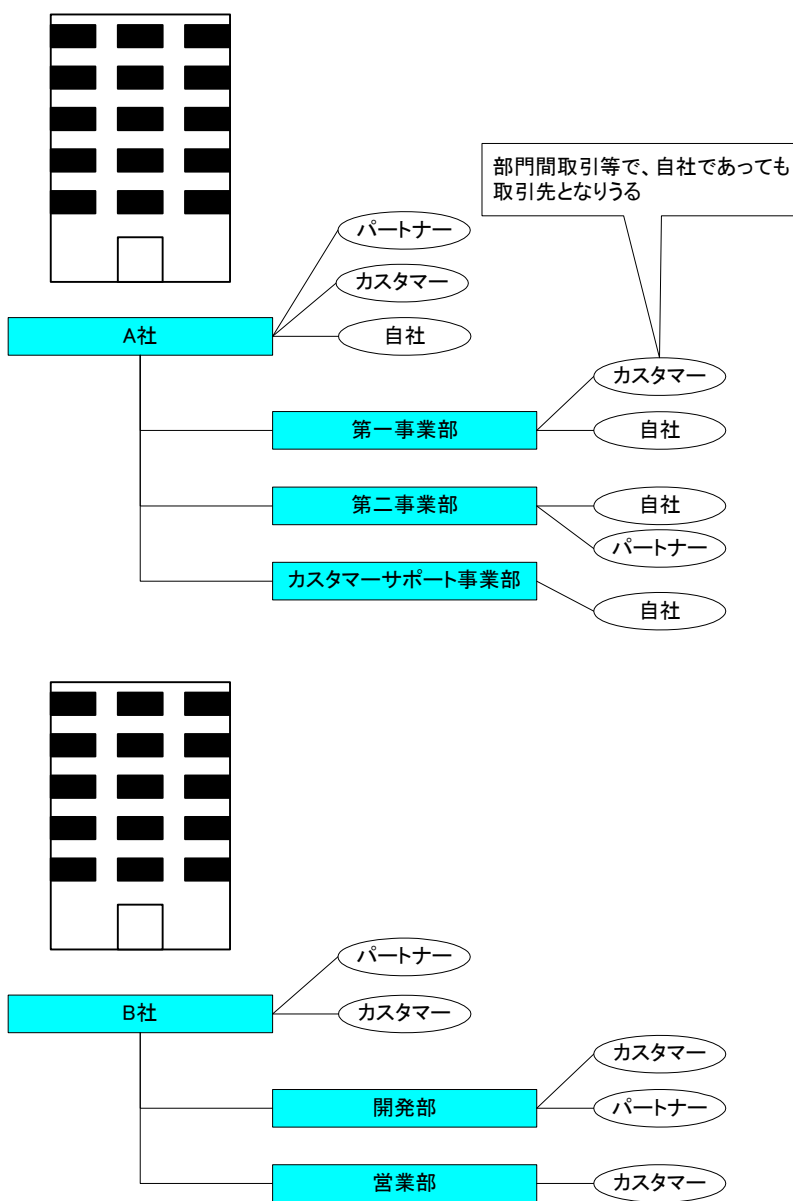


図 2-7 検索属性の例

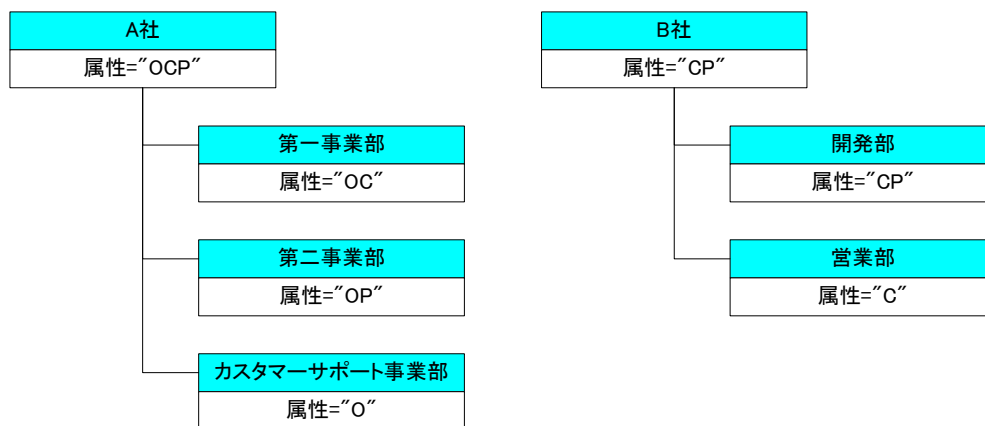
「図 2-7 検索属性の例」では、A社のイントラを中心としているため、A社から見た組織毎の検索属性を定義します。ユーザ会社毎の立場が平等である場合は、システム内における組織の役割をそれぞれ定義します。

各アプリケーションでは「図 2-8 検索属性によるデータ抽出」のようなデータの抽出を行います。

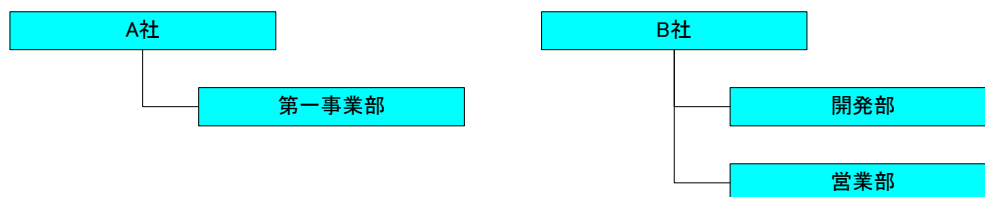
<設定>

分類マスタ

コード	分類名
O	自社
C	カスタマー
P	パートナー

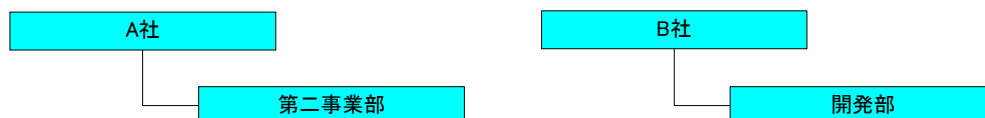


<顧客(カスタマー)検索画面>



*属性に“C”が含まれるデータを抽出

<協力会社(パートナー)検索画面>



*属性に“P”が含まれるデータを抽出

図 2-8 検索属性によるデータ抽出

検索属性の種類は、分類区分マスタおよび分類詳細マスタで定義します。登録されるデータとしては「表 2-5 検索属性の設定値」のようになります。

表 2-5 検索属性の設定値

分類区分名	分類区分コード	分類名	分類コード
検索属性	attribute	自社	O
		カスタマー	C
		パートナー	P

検索属性を使用する場合、以下のような点に注意する必要があります。

- 会社および組織について設定することができます。パブリックグループセット、パブリックグループおよびプライベートグループについては設定できません。
- 顧客はマスタメンテナンスによって変更・削除される可能性があるため、アプリケーションで使用する検索属性の分類コードはハードコーディングせず、設定ファイルにするなどの対策をする必要があります。

2.2.2.3 販売チャネル

販売チャネルとは、システムを運用する企業にとっての販売ルートを表現する概念です。

この概念を使って、Web 受発注や営業支援システムなどで多数の顧客を扱う際に、顧客をさらに販売チャネルという概念で細分化して管理または分析することを容易にします。

表 2-6 販売チャネル

分類区分名	分類区分コード	分類名	分類コード
販売チャネル	channel	未定義	未定義

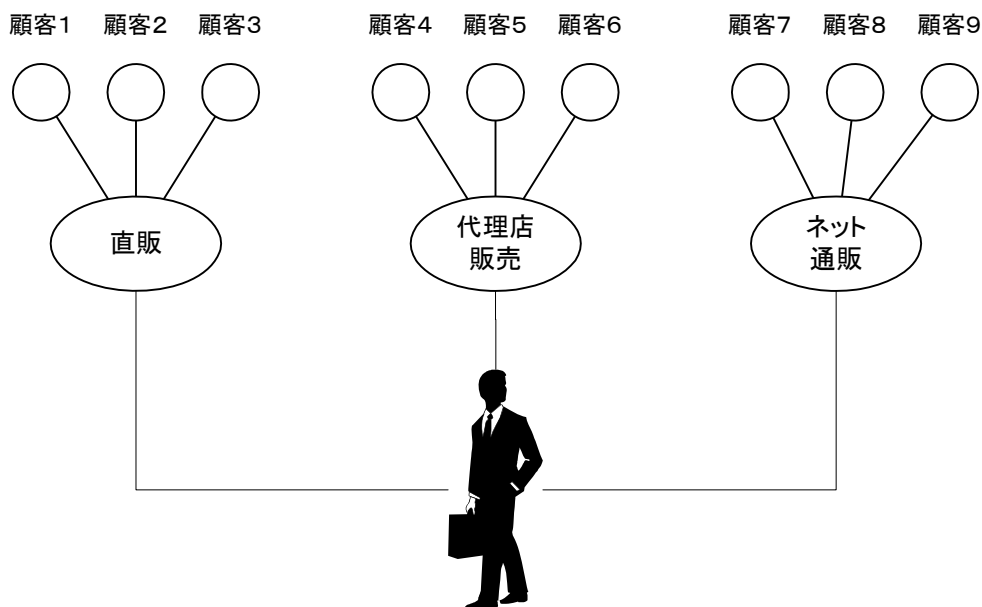


図 2-9 販売チャネル

検索属性を使用する場合、以下のような点に注意する必要があります。

- 会社および組織について設定することができます。パブリックグループセット、パブリックグループおよびプライベートグループについては設定できません。

- 最初から分類詳細マスタにいくつかのデータが用意されている公開ゾーンや検索属性とは異なり、販売チャネルに関してはあらかじめ分類詳細データが登録されていません。運用する環境やアプリケーションの性格等に応じて、データを登録します。

2.3 会社と組織

アプリケーション共通マスタでは会社やその組織に対する情報を扱うことができます。取り扱える情報には以下のようなものがあります。

- 会社や組織そのものの情報
- 組織構成の情報
- 組織構成のバージョン管理
- 会社ごとで扱う役職
- 会社または組織に所属するユーザ
- 任意のユーザのある時点における主所属
- 会社や組織の分類情報

2.3.1 データ構造

2.3.1.1 会社と組織の構成

会社および組織の構成に関連するER図を「図 2-10 会社・組織の構成に関連するER図」に示します。

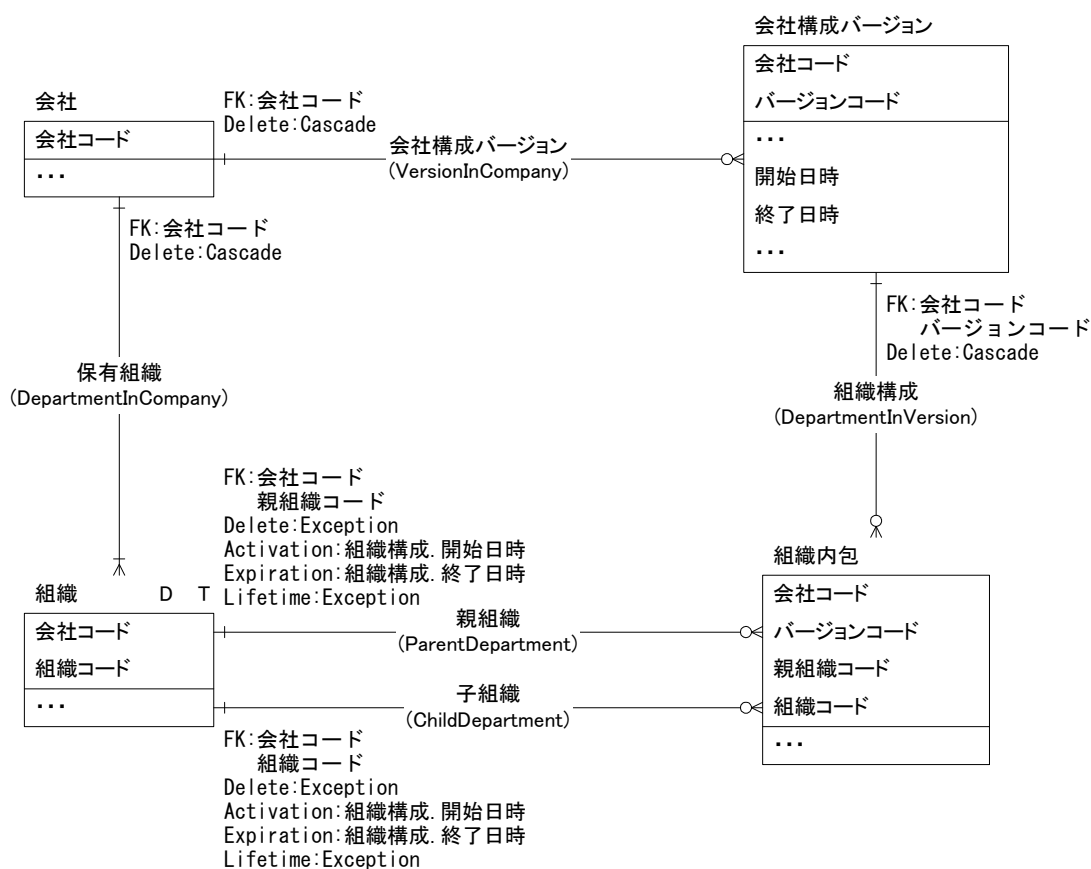


図 2-10 会社・組織の構成に関連する ER 図

「図 2-10 会社・組織の構成に関連するER図」において各エンティティは次のような役割があります。

- 会社
会社の概念を管理します。会社自身の詳細情報は組織エンティティで管理します。
- 組織
会社や会社内に存在する組織の詳細情報を管理します。
- 会社構成バージョン
会社内の組織構成のバージョンを管理します。構成については組織内包エンティティで管理します。
- 組織内包
会社内の組織構成の詳細情報を管理します。

また、「図 2-10 会社・組織の構成に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 会社エンティティのデータが削除された場合、
 - ◆ その会社で管理している会社構成バージョンも削除する必要があります。
 - ◆ その会社が保有している組織も削除する必要があります。
- 会社構成バージョンエンティティのデータを追加する場合、
 - ◆ 該当する会社が存在している必要があります。
- 会社構成バージョンエンティティのデータが削除された場合、
 - ◆ そのバージョンを構成している組織内包も削除する必要があります。
- 会社構成バージョンエンティティのデータの開始日時や終了日時が変更されようとした場合、
 - ◆ そのバージョンを構成している組織内包において親組織または子組織として使用されている組織が、変更後の期間内に一時でも存在しない場合は例外を発生させます。
- 組織エンティティのデータを追加する場合、
 - ◆ 該当する会社が存在している必要があります。
- 組織エンティティのデータの存在期間を変更しようとした場合、
 - ◆ 変更後の期間がその組織が使用されている会社構成バージョンの有効期間内に一時でも存在しない場合は例外を発生させます。
- 組織エンティティのデータが削除されようとした場合、
 - ◆ その組織を親組織または子組織として使用している組織内包が存在している場合は例外を発生させます。
- 組織内包エンティティのデータを追加する場合、
 - ◆ 親組織に該当する組織が存在している必要があります。
 - ◆ 子組織に該当する組織が存在している必要があります。
 - ◆ 親組織または子組織として使用される組織の存在期間が、該当する会社構成の有効期間内に一時でも存在しないような場合は例外を発生させます。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。マネージャを利用するとさらに「図 2-10 会社・組織の構成に関連するER図」からでは読み取れない以下のような整合性もチェックします。

- 組織内包が循環構造にならないかチェックします。
- 組織内包のデータを追加または削除したとき、関連する他の組織内包のデータの追加や削除も行います。詳細については「2.3.3 会社組織の構成」を参照してください。
- 会社をまたがった組織の移動はできません。

2.3.1.1.1 会社

会社のエンティティと実際のテーブルの構造を「表 2-7 会社のデータ構造」と「図 2-11 会社テーブルのER図」に示します。

表 2-7 会社のデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

会社(基本)
b_m_company_b

図 2-11 会社テーブルの ER 図

2.3.1.1.2 会社構成バージョン

会社構成バージョンのエンティティと実際のテーブルの構造を「表 2-8 会社構成バージョンのデータ構造」と「図 2-12 会社構成バージョンテーブルのER図」に示します。

表 2-8 会社構成バージョンのデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	—	—	
バージョンコード	version_cd	○	—	—	同一の会社コードに対して一意
開始日	start_date	—	×	×	
終了日	end_date	—	×	×	
備考	notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

会社構成バージョン(基本)
b_m_company_version_b

図 2-12 会社構成バージョンテーブルの ER 図

2.3.1.1.3 組織

組織のエンティティと実際のテーブルの構造を「表 2-9 組織のデータ構造」と「図 2-13 組織テーブルのER図」に示します。

表 2-9 組織のデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	—	—	
組織コード	department_cd	○	—	—	同一の会社コードに対して一意 会社コード=組織コードの場合、会社自身の情報とみなされる
組織名	department_name	—	○	○	
組織名カナ	department_name_syllabary	—	○	○	
組織名英字	department_name_eng	—	○	○	
電話番号	telephone_number	—	○	×	
FAX 番号	fax_number	—	○	×	
内線番号	extension_number	—	○	×	
内線 FAX 番号	extension_fax_number	—	○	×	
国コード	country_cd	—	○	×	
郵便番号	zip_code	—	○	×	
住所 1	address1	—	○	○	通常は住所 1 のみ使用
住所 2	address2	—	○	○	帳票印刷の折返し表示等で意図的に表示を分割したい場合はこちらも使用
メールアドレス 1	email_address1	—	○	×	
メールアドレス 2	email_address2	—	○	×	
URL	url	—	○	×	
備考	notes	—	×	×	
ソートキー ¹	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

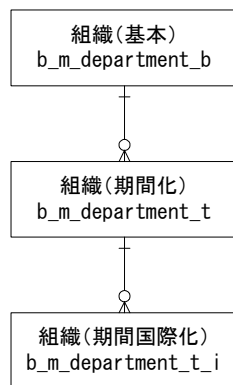


図 2-13 組織テーブルの ER 図

¹ 現在、ソートキーは組織設定画面および API にて行うことができますが、組織設定のソートキーとして使用されていません。

2.3.1.1.4 組織内包

組織内包のエンティティと実際のテーブルの構造を「表 2-10 組織内包のデータ構造」と「図 2-14 組織内包テーブルのER図」に示します。

表 2-10 組織内包のデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	—	—	
バージョンコード	version_cd	○	—	—	同一の会社コードに対して一意
親組織コード	parent_department_cd	○	—	—	
組織コード	department_cd	○	—	—	
階層レベル	depth	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

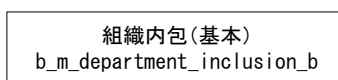


図 2-14 組織内包テーブルの ER 図

2.3.1.2 会社と組織への所属

ユーザが所属する会社および組織に関連するER図を「図 2-15 会社・組織とユーザの所属に関連するER図」に示します。

- 会社エンティティのデータが削除された場合、
 - ◆ その会社で定義している役職も削除する必要があります。
 - ◆ その会社が保有している組織も削除する必要があります。
- 役職エンティティのデータを追加する場合、
 - ◆ 該当する会社が存在している必要があります。
- 役職エンティティのデータが削除された場合、
 - ◆ 組織所属からその役職を参照している場合、参照を null にする必要があります。
- 役職エンティティのデータの有効期間が変更された場合、
 - ◆ 組織所属からその役職を参照している期間を調整する必要があります。役職そのものは参照していても一時的にその役職が休止しているような期間がある場合、その期間は参照を null にする必要があります。この場合、必要に応じて組織所属のエンティティのデータを分割する場合があります。
- 組織エンティティのデータを追加する場合、
 - ◆ 該当する会社が存在している必要があります。
- 組織エンティティのデータが削除された場合、
 - ◆ その組織に所属する構成部員も削除する必要があります。
- 組織エンティティのデータの有効期間が変更された場合、
 - ◆ その組織に所属する構成部員の所属期間を調整する必要があります。ある組織に所属しているにもかかわらず一時的にその組織が休止しているような期間がある場合、その期間は所属データを削除する必要があります。この場合、必要に応じて組織所属エンティティのデータを分割する場合があります。
- 組織所属エンティティのデータが削除されようとした場合、
 - ◆ その組織を主所属としていた情報も削除する必要があります。
- 組織所属エンティティのデータの有効期間が変更されようとした場合、
 - ◆ その期間において該当する組織が確実に存在している必要があります。一時的にでも所属しているにも関わらず組織が存在しないという状態がありうる場合、例外を発生させます。
 - ◆ その期間において該当するユーザが確実に存在している必要があります。一時的にでも所属しているにも関わらずユーザが存在しないという状態がありうる場合、例外を発生させます。
- 組織所属エンティティのデータの有効期間が変更された場合、
 - ◆ その組織を主所属としていた情報の所属期間を調整する必要があります。一時的にその組織に所属していないような期間がある場合、その期間は主所属データを削除する必要があります。この場合、必要に応じて主所属エンティティのデータを分割する場合があります。
- 組織所属エンティティのデータを追加する場合、
 - ◆ 該当する組織が存在している必要があります。
 - ◆ 該当するユーザが存在している必要があります。
 - ◆ 役職が付く場合、その役職が存在している必要があります。
- ユーザエンティティのデータが削除された場合、
 - ◆ そのユーザに関連する所属情報も削除する必要があります。
 - ◆ そのユーザに関連する主所属情報も削除する必要があります。
- ユーザエンティティのデータの有効期間が変更された場合、
 - ◆ そのユーザに関連する所属情報の所属期間を調整する必要があります。一時的にそのユーザが無効であるような期間がある場合、その期間は所属データを削除する必要があります。この場合、必要に応じて所属エンティティのデータを分割する場合があります。
 - ◆ そのユーザに関連する主所属情報の所属期間を調整する必要があります。一時的にそのユーザが無効であるような期間がある場合、その期間は主所属データを削除する必要があります。この場合、必要に応じて主所属エンティティのデータを分割する場合があります。
- 主所属エンティティのデータの有効期間が変更されようとした場合、
 - ◆ その期間において該当するユーザが確実に存在している必要があります。一時的にでもユーザが無

効であるにも関わらず所属しているという状態がありうる場合、例外を発生させます。

- ◆ その期間において該当するユーザが組織に確実に所属している必要があります。一時的にでもそのユーザがその組織に所属していない状態がありうる場合、例外を発生させます。
- 主所属エンティティのデータを追加する場合、
 - ◆ 該当する組織所属が存在している必要があります。
 - ◆ 該当するユーザが存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。

2.3.1.2.1 役職

役職のエンティティと実際のテーブルの構造を「表 2-11 役職のデータ構造」と「図 2-16 役職テーブルのER図」に示します。

表 2-11 役職のデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	—	—	
役職コード	post_cd	○	—	—	同一の会社コードに対して一意
役職名	post_name	—	○	○	
備考	notes	—	×	×	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

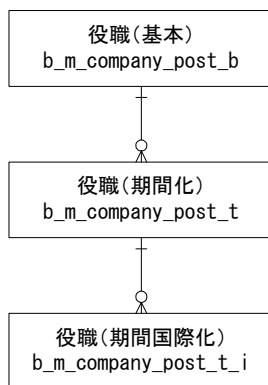


図 2-16 役職テーブルのER図

2.3.1.2.2 組織所属

組織所属のエンティティと実際のテーブルの構造を「表 2-12 組織所属のデータ構造」と「図 2-17 組織所属テーブルのER図」に示します。

表 2-12 組織所属のデータ構造

項目	カラム名	PK	期間化	国際化	備考
ユーザコード	user_cd	○	—	—	
会社コード	company_cd	○	—	—	
組織コード	department_cd	○	—	—	
役職コード	post_cd	—	○	×	役職が付かない場合は null
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

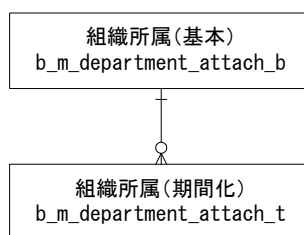


図 2-17 組織所属テーブルの ER 図

2.3.1.2.3 主所属

主所属のエンティティと実際のテーブルの構造を「表 2-13 主所属のデータ構造」と「図 2-18 主所属テーブルの ER 図」に示します。

表 2-13 主所属のデータ構造

項目	カラム名	PK	期間化	国際化	備考
ユーザコード	user_cd	○	—	—	ユーザコード、会社コード、組織コードの組み合わせは組織所属に存在する必要がある
会社コード	company_cd	—	○	×	同上
組織コード	department_cd	—	○	×	同上
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

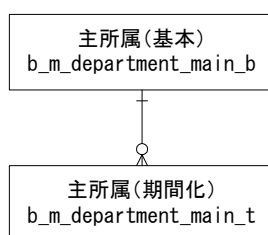


図 2-18 主所属テーブルの ER 図

2.3.1.3 会社と組織の分類

会社および組織の分類に関連するER図を「図 2-19 会社・組織の分類に関連するER図」に示します。

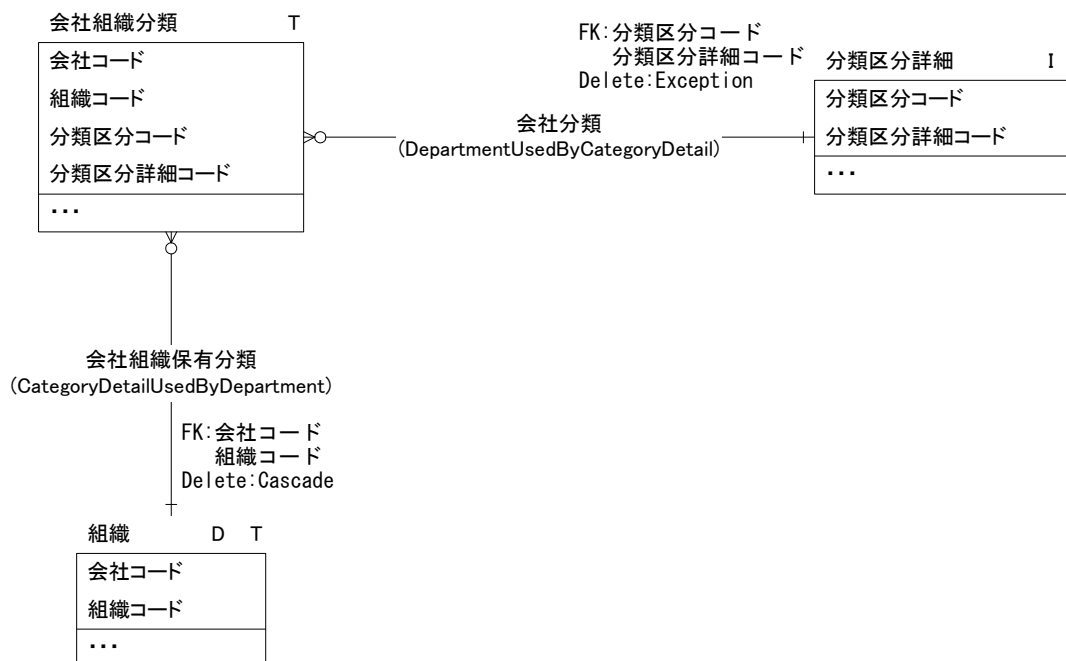


図 2-19 会社・組織の分類に関連する ER 図

「図 2-19 会社・組織の分類に関連するER図」において各エンティティは次のような役割があります。

- 分類区分詳細
分類の種類ごとに定義されている詳細を管理します。詳細は「2.2.1.2 分類区分詳細」を参照してください。
- 組織
会社や会社内に存在する組織の詳細情報を管理します。詳細は「2.3.1.1.3 組織」を参照してください。
- 会社組織分類
会社または組織がどのような分類情報を持つか管理します。

また、「図 2-19 会社・組織の分類に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 分類区分詳細エンティティのデータが削除された場合、
 - ◆ その分類区分詳細情報を使用している会社組織分類が存在している場合は例外を発生させます。
- 組織エンティティのデータが削除された場合、
 - ◆ その組織が持つ会社組織分類も削除する必要があります。
- 会社組織分類エンティティのデータを追加する場合、
 - ◆ 該当する分類区分詳細が存在している必要があります。
 - ◆ 該当する組織が存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。

2.3.1.3.1 会社組織分類

会社組織分類のエンティティと実際のテーブルの構造を「表 2-14 会社組織分類のデータ構造」と「図 2-20 会社組織分類テーブルのER図」に示します。

表 2-14 会社組織分類のデータ構造

項目	カラム名	PK	期間化	国際化	備考
会社コード	company_cd	○	○	—	
組織コード	department_cd	○	○	—	
分類区分コード	category_cls_cd	○	○	—	
分類コード	category_cd	○	○	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

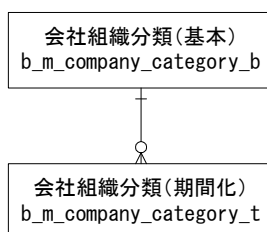


図 2-20 会社組織分類テーブルの ER 図

2.3.2 会社および組織の情報

アプリケーション共通マスタでは、会社の詳細情報は組織の詳細情報として保存されます。会社コードと組織コードが一致する組織が会社を表現する組織としてみなします。

会社と組織の違いは以下のとおりです。

- 会社は複数の組織を持ちますが、組織は会社を持ちません。
- 組織構成のバージョン(「2.3.4 会社組織構成のバージョン管理」を参照)は会社単位で管理されます。

組織は期間化管理され、さらに期間国際化されています。「図 2-21 組織の期間化」の例では会社とその組織の名称が期間管理されています。

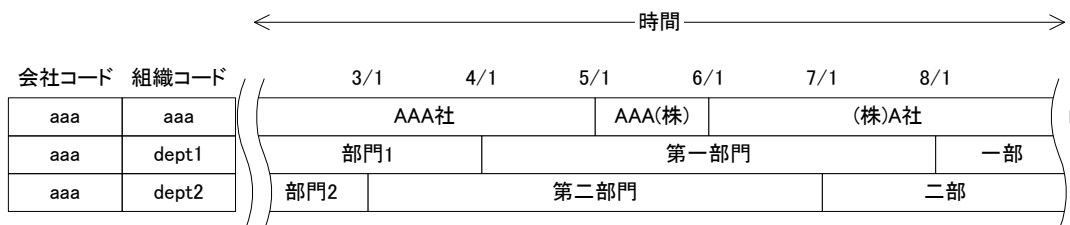


図 2-21 組織の期間化

「図 2-21 組織の期間化」の例をテーブルに登録したデータの状態を「図 2-22 組織の期間国際化(テーブル)」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

会社テーブル

会社コード
aaa
bbb
⋮

組織テーブル

会社コード	組織コード
aaa	aaa
aaa	dept1
aaa	dept2
bbb	bbb
⋮	⋮

組織テーブル(期間化)

会社コード	組織コード	期間コード	期間
aaa	aaa	term_0	~5/1
aaa	aaa	term_1	5/1~6/1
aaa	aaa	term_2	6/1~
aaa	dept1	term_0	~4/1
aaa	dept1	term_1	4/1~8/1
aaa	dept1	term_2	8/1~
aaa	dept2	term_0	~3/1
aaa	dept2	term_1	3/1~7/1
aaa	dept2	term_2	7/1~
bbb	bbb	term_0	~1/1
⋮	⋮	⋮	⋮

組織テーブル(期間国際化)

会社コード	組織コード	期間コード	ロケール	組織名
aaa	aaa	term_0	日本語	AAA社
aaa	aaa	term_0	英語	AAA Co.
aaa	aaa	term_1	日本語	AAA(株)
aaa	aaa	term_1	英語	AAA corp.
aaa	aaa	term_2	日本語	(株)A社
aaa	aaa	term_2	英語	A Co, Ltd.
aaa	dept1	term_0	日本語	部門1
aaa	dept1	term_0	英語	Dept. 1
aaa	dept1	term_1	日本語	第一部門
aaa	dept1	term_1	英語	Dept. One
aaa	dept1	term_2	日本語	一部
aaa	dept1	term_2	英語	D-1
aaa	dept2	term_0	日本語	部門2
aaa	dept2	term_0	英語	Dept. 2
aaa	dept2	term_1	日本語	第二部門
aaa	dept2	term_1	英語	Dept. Two
aaa	dept2	term_2	日本語	二部
aaa	dept2	term_2	英語	D-2
bbb	bbb	term_0	日本語	BBB社
⋮	⋮	⋮	⋮	⋮

図 2-22 組織の期間国際化(テーブル)

「図 2-22 組織の期間国際化(テーブル)」では期間国際化されたデータも示しています。ここでは、会社コード "aaa" という会社が存在し、その詳細情報は組織コード "aaa" の組織に登録されていることがわかります。

2.3.3 会社組織の構成

アプリケーション共通マスタでは、会社組織の構成は会社をトップノードとしたツリー構造で表現できます。このツリー構造を表現するテーブルとして組織内包があります。組織内包は次のような構造を持ちます。

- ツリー上のすべての組織(会社を含みます)から下方にたどることができる組織の情報を持ちます。この情報には親組織からみて子組織が何階層目に該当するかという情報が含まれます。
- 親組織と子組織が同じ場合、階層数は 0 となります。
- 親組織から見て直下の階層の組織は階層数 1 とみなされます。

会社組織の構成の例を「図 2-23 会社組織の構成例」に、この場合におけるテーブルの状態を「図 2-24 会社組織の構成内容」に示します。

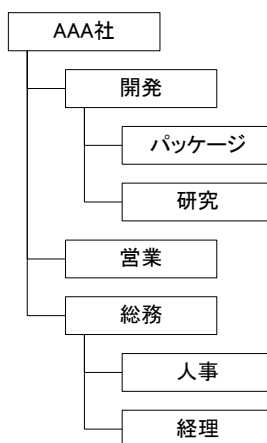


図 2-23 会社組織の構成例

組織内包テーブル

会社	バージョン	親組織	子組織	階層
AAA社	version_1	AAA社	AAA社	0
AAA社	version_1	AAA社	開発	1
AAA社	version_1	AAA社	パッケージ	2
AAA社	version_1	AAA社	研究	2
AAA社	version_1	AAA社	営業	1
AAA社	version_1	AAA社	総務	1
AAA社	version_1	AAA社	人事	2
AAA社	version_1	AAA社	経理	2
AAA社	version_1	開発	開発	0
AAA社	version_1	開発	パッケージ	1
AAA社	version_1	開発	研究	1
AAA社	version_1	パッケージ	パッケージ	0
AAA社	version_1	研究	研究	0
AAA社	version_1	営業	営業	0
AAA社	version_1	総務	総務	0
AAA社	version_1	総務	人事	1
AAA社	version_1	総務	経理	1
AAA社	version_1	人事	人事	0
AAA社	version_1	経理	経理	0

図 2-24 会社組織の構成内容

「図 2-24 会社組織の構成内容」では簡略化のために会社名や組織名がプライマリキーの一部になっていますが、実際のテーブル構造では会社コードおよび組織コードがプライマリキーの一部として使用されています。また、バージョンについては「2.3.4 会社組織構成のバージョン管理」を参照してください。

2.3.4 会社組織構成のバージョン管理

アプリケーション共通マスタではこの構成を次のような単位で管理することができます。

- ある時点における会社組織の構成は会社ごとに定義することができます。
- 同一の会社であっても期間によって構成を変更することができます。つまり、各会社ごとに構成のバージョン管理ができます。

会社組織の構成がバージョン管理されている様子を「図 2-25 会社組織の構成とバージョン」に示します。また、この場合における会社と組織の登録内容を「図 2-26 会社と組織の情報」に、バージョンごとの構成を「図 2-27 会社組織の構成(バージョン1)」および「図 2-28 会社組織の構成(バージョン2)」に示します。

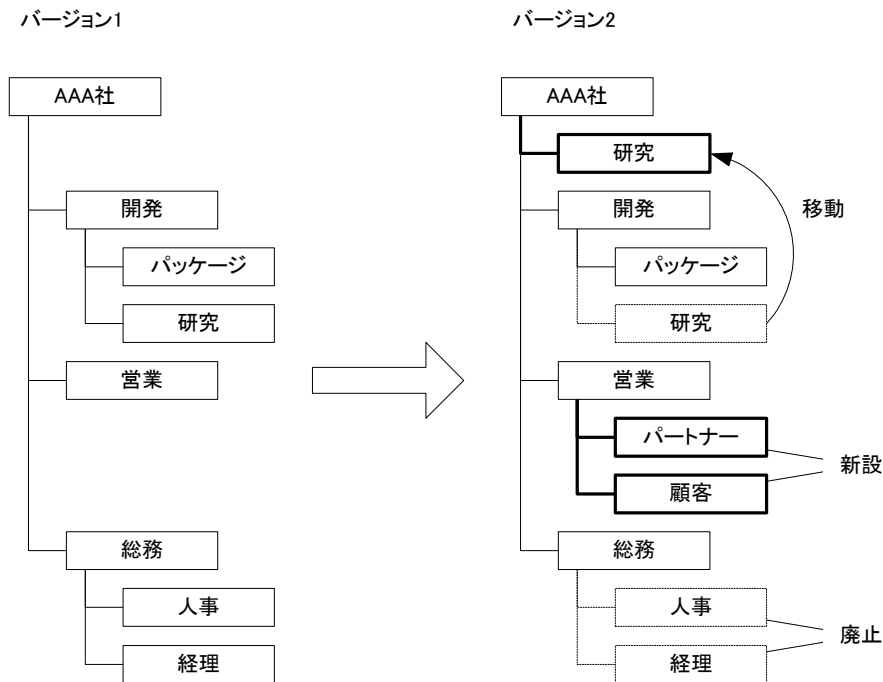


図 2-25 会社組織の構成とバージョン

会社テーブル		会社バージョンテーブル		組織テーブル	
会社		会社	バージョン	会社	組織
AAA社		AAA社	version_1	AAA社	AAA社
		AAA社	version_2	AAA社	開発
				AAA社	パッケージ
				AAA社	研究
				AAA社	営業
				AAA社	総務
				AAA社	人事
				AAA社	経理
				AAA社	パートナー
				AAA社	顧客

図 2-26 会社と組織の情報

組織内包テーブル

会社	バージョン	親組織	子組織	階層
AAA社	version_1	AAA社	AAA社	0
AAA社	version_1	AAA社	開発	1
AAA社	version_1	AAA社	パッケージ	2
AAA社	version_1	AAA社	研究	2
AAA社	version_1	AAA社	営業	1
AAA社	version_1	AAA社	総務	1
AAA社	version_1	AAA社	人事	2
AAA社	version_1	AAA社	経理	2
AAA社	version_1	開発	開発	0
AAA社	version_1	開発	パッケージ	1
AAA社	version_1	開発	研究	1
AAA社	version_1	パッケージ	パッケージ	0
AAA社	version_1	研究	研究	0
AAA社	version_1	営業	営業	0
AAA社	version_1	総務	総務	0
AAA社	version_1	総務	人事	1
AAA社	version_1	総務	経理	1
AAA社	version_1	人事	人事	0
AAA社	version_1	経理	経理	0

図 2-27 会社組織の構成(バージョン 1)

組織内包テーブル

会社	バージョン	親組織	子組織	階層
AAA社	version_2	AAA社	AAA社	0
AAA社	version_2	AAA社	研究	1
AAA社	version_2	AAA社	開発	1
AAA社	version_2	AAA社	パッケージ	2
AAA社	version_2	AAA社	営業	1
AAA社	version_2	AAA社	パートナー	2
AAA社	version_2	AAA社	顧客	2
AAA社	version_2	AAA社	総務	1
AAA社	version_2	研究	研究	0
AAA社	version_2	開発	開発	0
AAA社	version_2	開発	パッケージ	1
AAA社	version_2	パッケージ	パッケージ	0
AAA社	version_2	営業	営業	0
AAA社	version_2	営業	パートナー	1
AAA社	version_2	営業	顧客	1
AAA社	version_2	総務	総務	0
AAA社	version_2	パートナー	パートナー	0
AAA社	version_2	顧客	顧客	0

図 2-28 会社組織の構成(バージョン 2)

「図 2-26 会社と組織の情報」、「図 2-27 会社組織の構成(バージョン 1)」および「図 2-28 会社組織の構成(バージョン 2)」では簡略化のために会社名や組織名がプライマリキーの一部になっていますが、実際のテーブル構造では会社コードおよび組織コードがプライマリキーの一部として使用されています。

2.3.5 役職

アプリケーション共通マスタでは、ユーザが会社や組織に所属(所属については「2.3.6 会社組織への所属」を参照してください)するときの役職を会社ごとに定義できます。

役職は期間化管理され、さらに期間国際化されています。「図 2-29 役職情報」の例では役職の名称が期間管理されています。

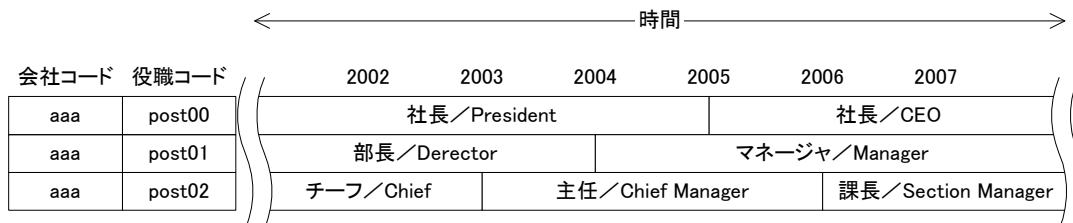


図 2-29 役職情報

「図 2-29 役職情報」の例をテーブルに登録したデータの状態を「図 2-30 役職の期間国際化(テーブル)」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

会社テーブル		役職テーブル		組織テーブル(期間国際化)				
会社コード*	会社コード*	役職コード*	会社コード*	組織コード*	期間コード*	ローケル	役職名	
aaa	aaa	post00	aaa	post00	term_0	日本語	社長	
bbb	aaa	post01	aaa	post00	term_0	英語	President	
⋮	aaa	post02	aaa	post00	term_1	日本語	社長	
	bbb	post0	aaa	post00	term_1	英語	CEO	
	⋮	⋮	aaa	post01	term_0	日本語	部長	
	⋮	⋮	aaa	post01	term_0	英語	Director	
	⋮	⋮	aaa	post01	term_1	日本語	マネージャ	
	⋮	⋮	aaa	post01	term_1	英語	Manager	
	⋮	⋮	aaa	post02	term_0	日本語	チーフ	
	⋮	⋮	aaa	post02	term_0	英語	Chief	
	⋮	⋮	aaa	post02	term_1	日本語	主任	
	⋮	⋮	aaa	post02	term_1	英語	Chief Manager	
	⋮	⋮	aaa	post02	term_2	日本語	課長	
	⋮	⋮	aaa	post02	term_2	英語	Section Manager	
	⋮	⋮	bbb	post0	term_0	日本語	会長	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

図 2-30 役職の期間国際化(テーブル)

2.3.6 会社組織への所属

アプリケーション共通マスタでは、ユーザはある時点において複数の会社や組織に所属することができます。ユーザが複数の会社や組織に所属している場合、主所属を決めることもできます。主所属は任意の時点において最高でも1つだけです。所属していない会社や組織を主所属にすることはできません。

2.3.6.1 所属

アプリケーション共通マスタでは、ユーザは複数の組織に所属することができます。この場合、ユーザは複数の会社をまたがって所属することも可能です。所属情報は会社や組織の期間化情報やバージョンには依存しません。

「このユーザはこの時点ではこの組織に所属している」という情報のみが保存されます。

ユーザは会社または組織に所属している場合、会社ごとに定義されている役職(役職については「2.3.5 役職」を参照してください)に就任することができます。同一の所属期間中では同一の役職に就任しているか、役職に就任していないかのいずれかである必要があります。ある時点から役職を変更したい場合、所属期間を新しくする必要があります。

「図 2-31 会社組織の所属情報」に所属の例を示します。

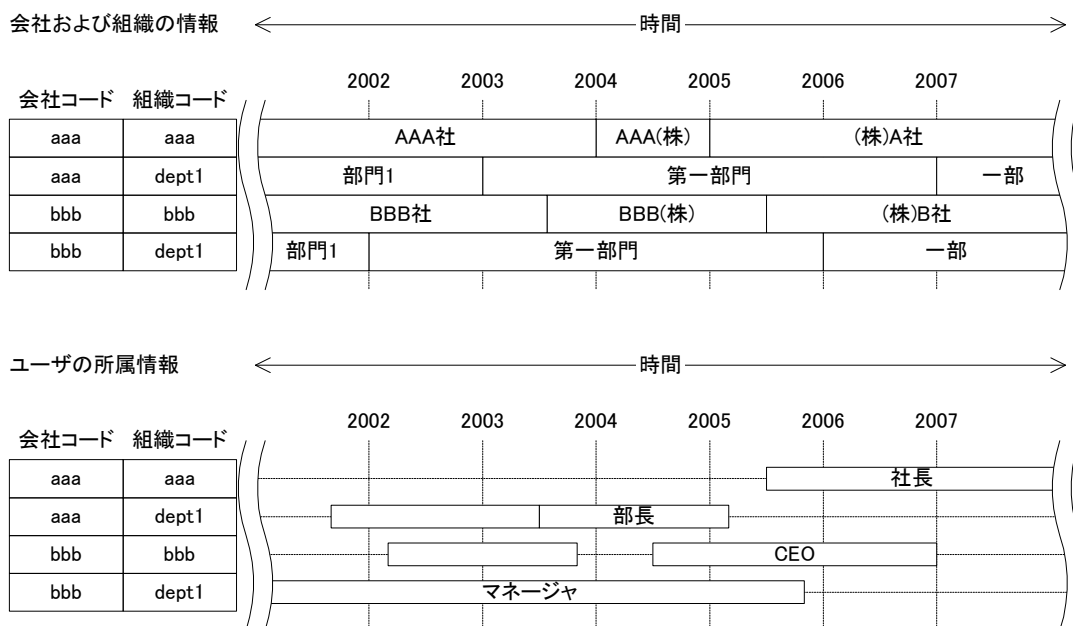


図 2-31 会社組織の所属情報

「図 2-31 会社組織の所属情報」で示されるユーザの経歴および予定は「表 2-15 ユーザの所属経歴」のようになります。

表 2-15 ユーザの所属経歴

時期	内容
	BBB 社の部門 1 にマネージャとして就任
2001 下旬	AAA 社の部門 1 に就任
2002 初旬	BBB 社に就任 BBB 社の部門 1 にはマネージャとして就任したまま
2003 中旬	AAA 社の第一部門(旧:部門 1)に部長として再就任
2003 下旬	BBB(株)(旧:BBB 社)から退任 BBB(株)の第一部門(旧:BBB 社の部門 1)にはマネージャとして就任したまま
2004 中旬	BBB(株)に CEO として再就任
2005 中旬	(株)A 社に社長として就任
2005 下旬	(株)B 社の第一部門(旧:BBB 社の部門 1)から退任
2007 末	(株)B 社(旧:BBB(株))から退任

「図 2-31 会社組織の所属情報」および「表 2-15 ユーザの所属経歴」の内容はアプリケーション共通マスタには「図 2-32 ユーザの所属履歴(テーブル)」のように登録されます。

組織所属テーブル

ユーザ	会社コード	組織コード
user0001	aaa	aaa
user0001	aaa	dept1
user0001	bbb	bbb
user0001	bbb	dept1

会社役職テーブル

会社コード	役職
aaa	社長
aaa	部長
bbb	CEO
bbb	マネージャ

組織所属テーブル(期間化)

ユーザ	会社コード	組織コード	期間コード	開始	終了	役職
user0001	aaa	aaa	term0	2005中旬	(未定)	社長
user0001	aaa	dept1	term0	2001下旬	2003中旬	(なし)
user0001	aaa	dept1	term1	2003中旬	2005初旬	部長
user0001	bbb	bbb	term0	2002初旬	2003下旬	(なし)
user0001	bbb	bbb	term1	2004中旬	2007末	CEO
user0001	bbb	dept1	term0	(未定)	2005下旬	マネージャ

図 2-32 ユーザの所属履歴(テーブル)

「図 2-32 ユーザの所属履歴(テーブル)」では簡略化のために役職名を使用していますが、実際のテーブル構造では役職コードになります。

2.3.6.2 主所属

ユーザは同時に複数の会社や組織に所属することができますが、このうちの一つを主所属として扱うこともできます。

「図 2-33 主所属情報」に主所属の例を示します。

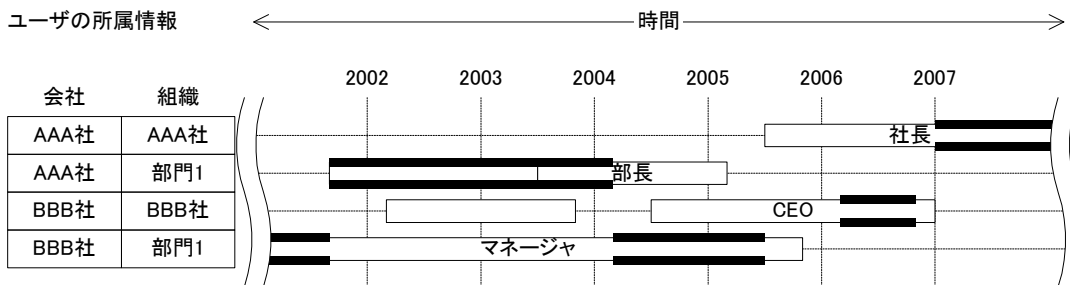


図 2-33 主所属情報

「図 2-33 主所属情報」の所属情報は「図 2-31 会社組織の所属情報」と同じですが、主所属のところが太線で表現されています。

- 主所属は必ず所属情報が存在する場所で設定される必要があります。
- 主所属は同一時点で二箇所以上設定することはできません。
- 主所属は所属そのものの期間が連続していれば役職の変更とは無関係に設定できます(「図 2-33 主所属情報」ではAAA社の部門1に所属している途中で役職が部長に変更されていますが、主所属期間としては切断されていません)。

「図 2-33 主所属情報」の内容はアプリケーション共通マスタには「図 2-34 ユーザの主所属履歴(テーブル)」のように登録されます。

主所属テーブル

ユーザ	会社	組織
user0001	AAA社	AAA社
user0001	AAA社	部門1
user0001	BBB社	BBB社
user0001	BBB社	部門1

主所属テーブル(期間化)

ユーザ	期間コード	開始	終了	会社	組織
user0001	term0	(未定)	2001下旬	BBB社	部門1
user0001	term1	2001下旬	2004初旬	AAA社	部門1
user0001	term2	2004初旬	2005中旬	BBB社	部門1
user0001	term3	2006初旬	2006下旬	BBB社	BBB社
user0001	term4	2007初頭	(未定)	AAA社	AAA社

図 2-34 ユーザの主所属履歴(テーブル)

「図 2-34 ユーザの主所属履歴(テーブル)」では簡略化のために会社名や組織名を使用していますが、実際のテーブル構造では会社コードや組織コードになります。

2.3.7 分類

会社や組織はさまざまな角度から分類することができます。アプリケーション共通マスタでは、標準では以下のカテゴリで会社や組織を分類できます。

- 公開ゾーン
- 検索属性
- 販売チャネル

これらの詳細については「2.2.2 初期登録時の分類の内容」を参照してください。

会社や組織に対する分類の設定は期間化することが可能です。「図 2-35 会社組織の分類の変遷例」にその例を示します。

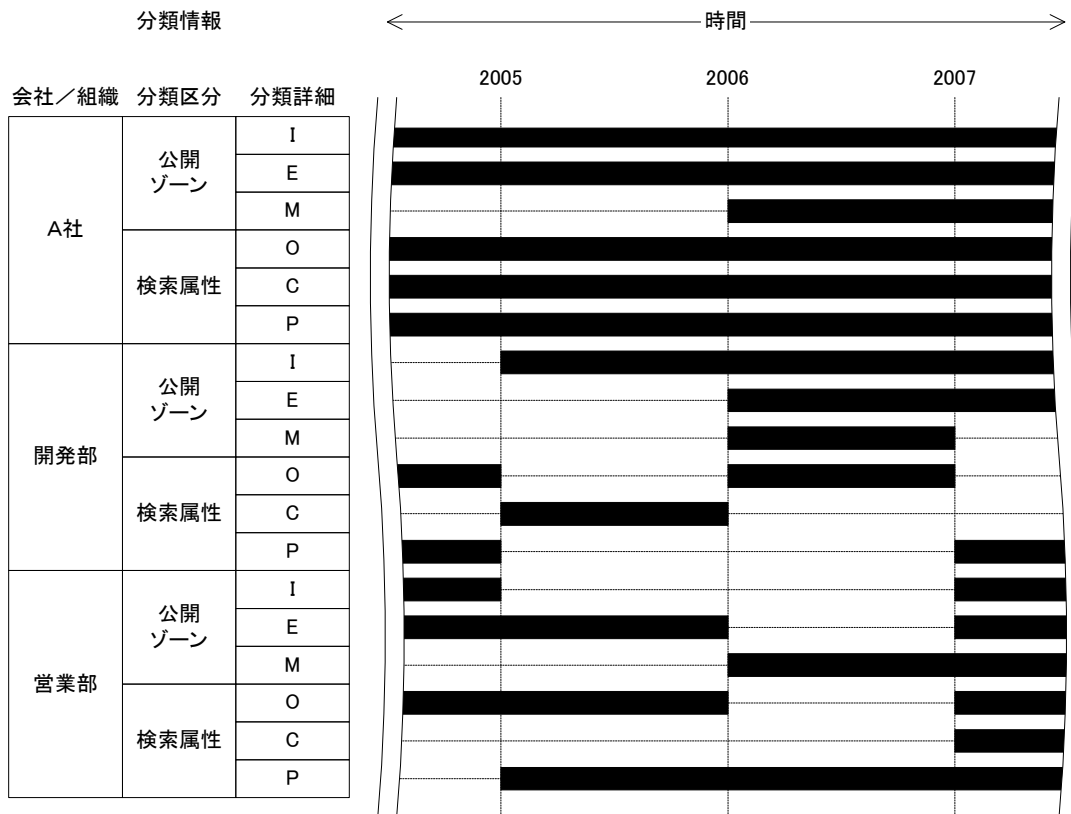


図 2-35 会社組織の分類の変遷例

「図 2-35 会社組織の分類の変遷例」がアプリケーション共通マスタのテーブルに登録されている状態を「図 2-36 会社組織の分類の変遷例(基本テーブル)」および「図 2-37 会社組織の分類の変遷例(期間化テーブル)」に示します。

会社組織分類テーブル

会社	組織	分類区分	分類
A社	A社	ゾーン	イントラネット・ゾーン
A社	A社	ゾーン	エクストラネット・ゾーン
A社	A社	ゾーン	マーケットプレイス・ゾーン
A社	A社	検索属性	自社
A社	A社	検索属性	カスタマー
A社	A社	検索属性	パートナー
A社	開発部	ゾーン	イントラネット・ゾーン
A社	開発部	ゾーン	エクストラネット・ゾーン
A社	開発部	ゾーン	マーケットプレイス・ゾーン
A社	開発部	検索属性	カスタマー
A社	開発部	検索属性	パートナー
A社	開発部	検索属性	自社
A社	営業部	ゾーン	イントラネット・ゾーン
A社	営業部	ゾーン	エクストラネット・ゾーン
A社	営業部	ゾーン	マーケットプレイス・ゾーン
A社	営業部	検索属性	自社
A社	営業部	検索属性	カスタマー
A社	営業部	検索属性	パートナー

図 2-36 会社組織の分類の変遷例(基本テーブル)

会社組織分類テーブル(期間化)

会社	組織	分類区分	分類	期間コード	期間
A社	A社	ゾーン	イントラネット・ゾーン	term0	～
A社	A社	ゾーン	エクストラネット・ゾーン	term0	～
A社	A社	ゾーン	マーケットプレイス・ゾーン	term0	2006～
A社	A社	検索属性	自社	term0	～
A社	A社	検索属性	カスタマー	term0	～
A社	A社	検索属性	パートナー	term0	～
A社	開発部	ゾーン	イントラネット・ゾーン	term0	2005～
A社	開発部	ゾーン	エクストラネット・ゾーン	term0	2006～
A社	開発部	ゾーン	マーケットプレイス・ゾーン	term0	2006～2007
A社	開発部	検索属性	自社	term0	～2005
A社	開発部	検索属性	自社	term1	2006～2007
A社	開発部	検索属性	カスタマー	term0	2005～2006
A社	開発部	検索属性	パートナー	term0	～2005
A社	開発部	検索属性	パートナー	term1	2007～
A社	営業部	ゾーン	イントラネット・ゾーン	term0	～2005
A社	営業部	ゾーン	イントラネット・ゾーン	term1	2007～
A社	営業部	ゾーン	エクストラネット・ゾーン	term0	～2006
A社	営業部	ゾーン	エクストラネット・ゾーン	term1	2007～
A社	営業部	ゾーン	マーケットプレイス・ゾーン	term0	2006～
A社	営業部	検索属性	自社	term0	～2006
A社	営業部	検索属性	自社	term1	2007～
A社	営業部	検索属性	カスタマー	term0	2007～
A社	営業部	検索属性	パートナー	term0	2005～

図 2-37 会社組織の分類の変遷例(期間化テーブル)

「図 2-36 会社組織の分類の変遷例(基本テーブル)」および「図 2-37 会社組織の分類の変遷例(期間化テーブル)」では簡略化のために会社名、組織名、分類区分名および分類名を使用していますが、実際のテーブル構造では会社コード、組織コード、分類区分コードおよび分類コードになります。

2.4 パブリックグループ

「2.3 会社と組織」では会社や組織を扱うときの情報について説明しました。しかし、そのような概念には当てはまらない団体も存在する場合があります(サークル、非営利団体等)。アプリケーション共通マスタではこのような団体をパブリックグループとして情報を扱うことができます。取り扱える情報には以下のようなものがあります。

- パブリックグループそのものの情報
- グループ構成の情報
- グループ構成のバージョン管理
- グループに所属するユーザ
- グループの分類情報

2.4.1 データ構造

2.4.1.1 パブリックグループの構成

パブリックグループセットおよびパブリックグループの構成に関連するER図を「図 2-38 パブリックグループの構成に関連するER図」に示します。

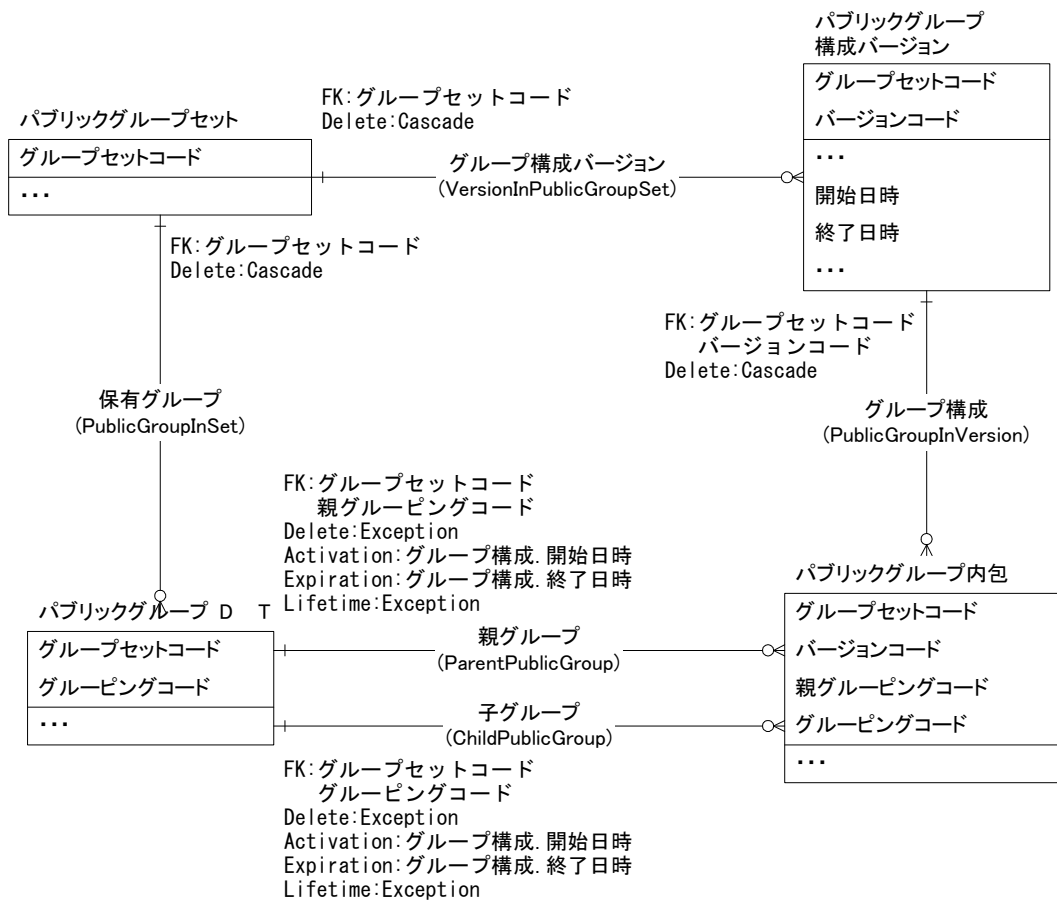


図 2-38 パブリックグループの構成に関連する ER 図

「図 2-38 パブリックグループの構成に関連するER図」において各エンティティは次のような役割があります。

- **パブリックグループセット**
パブリックグループの集合の概念を管理します。パブリックグループセット自身の詳細情報はパブリックグループエンティティで管理します。
- **パブリックグループ**
パブリックグループセットやパブリックグループセット内に存在するパブリックグループの詳細情報を管理します。
- **パブリックグループ構成バージョン**
パブリックグループセット内のパブリックグループ構成のバージョンを管理します。構成についてはパブリックグループ内包エンティティで管理します。
- **パブリックグループ内包**
パブリックグループセット内のパブリックグループ構成の詳細情報を管理します。

また、「図 2-38 パブリックグループの構成に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- **パブリックグループセットエンティティのデータが削除された場合、**
 - ◆ そのパブリックグループセットで管理しているパブリックグループ構成バージョンも削除する必要があります。
 - ◆ そのパブリックグループセットが保有しているパブリックグループも削除する必要があります。
- **パブリックグループセット構成バージョンエンティティのデータを追加する場合、**
 - ◆ 該当するパブリックグループセットが存在している必要があります。

- パブリックグループセット構成バージョンエンティティのデータが削除された場合、
 - ◆ そのバージョンを構成しているパブリックグループ内包も削除する必要があります。
- パブリックグループ構成バージョンエンティティのデータの開始日時や終了日時が変更されようとした場合、
 - ◆ そのバージョンを構成しているパブリックグループ内包において親グループまたは子グループとして使用されているパブリックグループが、変更後の期間内に一時でも存在しない場合は例外を発生させます。
- パブリックグループエンティティのデータを追加する場合、
 - ◆ 該当するパブリックグループセットが存在している必要があります。
- パブリックグループエンティティのデータの存在期間を変更しようとした場合、
 - ◆ 変更後の期間がその組織が使用されているパブリックグループ構成バージョンの有効期間内に一時でも存在しない場合は例外を発生させます。
- パブリックグループエンティティのデータが削除されようとした場合、
 - ◆ そのパブリックグループを親グループまたは子グループとして使用しているパブリックグループ内包が存在している場合は例外を発生させます。
- パブリックグループ内包エンティティのデータを追加する場合、
 - ◆ 親グループに該当するパブリックグループが存在している必要があります。
 - ◆ 子グループに該当するパブリックグループが存在している必要があります。
 - ◆ 親グループまたは子グループとして使用されるパブリックグループの存在期間が、該当するパブリックグループ構成の有効期間内に一時でも存在しないような場合は例外を発生させます。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。マネージャを利用するとさらに「図 2-38 パブリックグループの構成に関連するER図」からでは読み取れない以下のような整合性もチェックします。

- パブリックグループ内包が循環構造にならないかチェックします。
- パブリックグループ内包のデータを追加または削除したとき、関連する他のパブリックグループ内包のデータの追加や削除も行います。詳細については「2.4.3 パブリックグループの構成」を参照してください。

2.4.1.1.1 パブリックグループセット

パブリックグループセットのエンティティと実際のテーブルの構造を「表 2-16 会社のデータ構造」と「図 2-39 パブリックグループセットテーブルのER図」に示します。

表 2-16 会社のデータ構造

項目	カラム名	PK	期間化	国際化	備考
グループセットコード	Public_group_set_cd	○	—	—	
最終更新者	Record_user_cd	—	×	×	
最終更新日	Record_date	—	×	×	

パブリックグループセット(基本)
b_m_public_group_set_b

図 2-39 パブリックグループセットテーブルの ER 図

2.4.1.1.2 パブリックグループ構成バージョン

パブリックグループ構成バージョンのエンティティと実際のテーブルの構造を「表 2-17 パブリックグループ構成バージョンのデータ構造」と「図 2-40 パブリックグループ構成バージョンテーブルのER図」に示します。

表 2-17 パブリックグループ構成バージョンのデータ構造

項目	カラム名	PK	期間化	国際化	備考
グループセットコード	public_group_set_cd	○	—	—	
バージョンコード	version_cd	○	—	—	
開始日	Start_date	—	×	×	
終了日	end_date	—	×	×	
備考	Notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

パブリックグループ構成バージョン(基本)
b_m_public_group_version_b

図 2-40 パブリックグループ構成バージョンテーブルの ER 図

2.4.1.1.3 パブリックグループ

パブリックグループのエンティティと実際のテーブルの構造を「表 2-18 パブリックグループのデータ構造」と「図 2-41 パブリックグループテーブルのER図」に示します。

表 2-18 パブリックグループのデータ構造

項目	カラム名	PK	期間化	国際化	備考
グループセットコード	public_group_set_cd	○	—	—	
グループコード	public_group_cd	○	—	—	
グループ名	public_group_name	—	○	○	
グループ名カナ	public_group_name_syllabary	—	○	○	
グループ名英字	public_group_name_eng	—	○	○	
備考	Notes	—	×	×	
ソートキー ²	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

² 現在、ソートキーはパブリックグループ設定画面および API にて行うことができますが、パブリックグループ設定のソートキーとして使用はされていません。

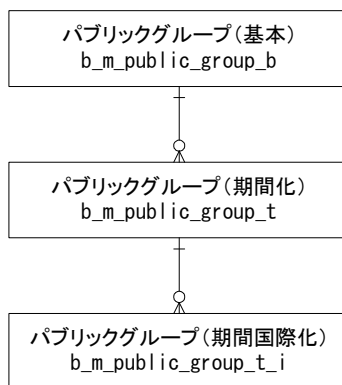


図 2-41 パブリックグループテーブルの ER 図

2.4.1.1.4 パブリックグループ内包

パブリックグループ内包のエンティティと実際のテーブルの構造を「表 2-19 パブリックグループ内包のデータ構造」と「図 2-42 パブリックグループ内包テーブルのER図」に示します。

表 2-19 パブリックグループ内包のデータ構造

項目	カラム名	PK	期間化	国際化	備考
グループセットコード	public_group_set_cd	○	—	—	
バージョンコード	version_cd	○	—	—	
親グループコード	parent_public_group_cd	○	—	—	
グループコード	public_group_cd	○	—	—	
階層レベル	depth	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

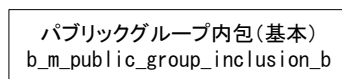


図 2-42 パブリックグループ内包テーブルの ER 図

2.4.1.2 パブリックグループへの所属

ユーザが所属するパブリックグループセットおよびパブリックグループに関連するER図を「図 2-43 パブリックグループとユーザの所属に関連するER図」に示します。

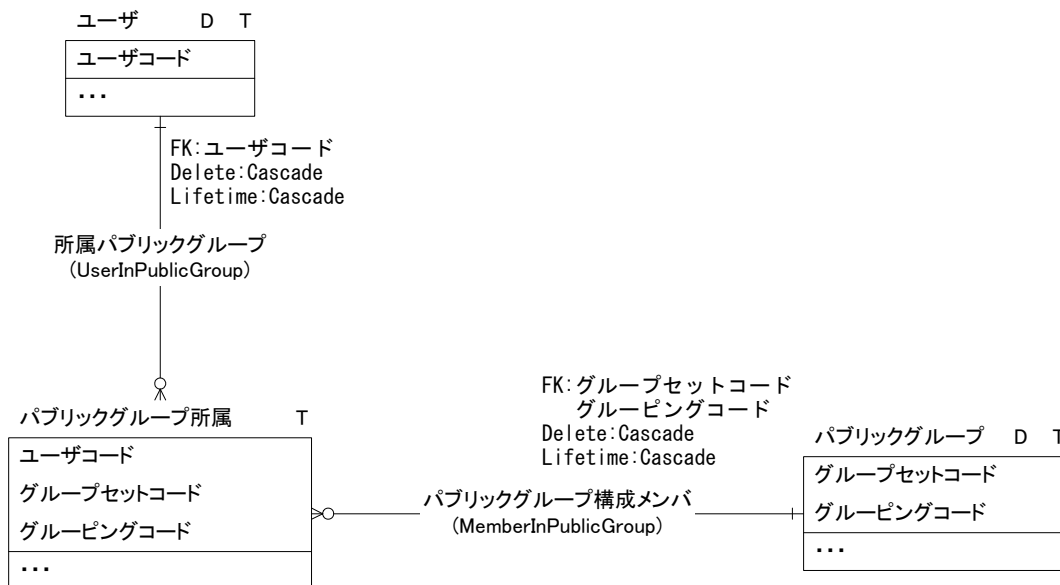


図 2-43 パブリックグループとユーザの所属に関する ER 図

「図 2-43 パブリックグループとユーザの所属に関するER図」において各エンティティは次のような役割があります。

- ユーザ
ユーザを管理します。詳細は「2.1 ユーザ」を参照してください。
- パブリックグループ
パブリックグループの詳細情報を管理します。詳細は「2.4.1.1.3 パブリックグループ」を参照してください。
- パブリックグループ所属
ユーザがどのパブリックグループに所属するかを管理します。

また、「図 2-43 パブリックグループとユーザの所属に関するER図」から以下のような整合性チェックが必要であることが読み取れます。

- パブリックグループエンティティのデータが削除された場合、
 - ◆ そのパブリックグループに所属する構成メンバも削除する必要があります。
- パブリックグループエンティティのデータの有効期間が変更された場合、
 - ◆ そのパブリックグループに所属する構成メンバの所属期間を調整する必要があります。あるパブリックグループに所属しているにもかかわらず一時的にそのパブリックグループが休止しているような期間がある場合、その期間は所属データを削除する必要があります。この場合、必要に応じてパブリックグループ所属エンティティのデータを分割する場合があります。
- パブリックグループ所属エンティティのデータの有効期間が変更されようとした場合、
 - ◆ その期間において該当するパブリックグループが確実に存在している必要があります。一時的にでも所属しているにもかかわらずパブリックグループが存在しないという状態がありうる場合、例外を発生させます。
 - ◆ その期間において該当するユーザが確実に存在している必要があります。一時的にでも所属しているにもかかわらずユーザが存在しないという状態がありうる場合、例外を発生させます。
- パブリックグループ所属エンティティのデータを追加する場合、
 - ◆ 該当するパブリックグループが存在している必要があります。
 - ◆ 該当するユーザが存在している必要があります。
- ユーザエンティティのデータが削除された場合、
 - ◆ そのユーザに関連する所属情報も削除する必要があります。

- ユーザエンティティのデータの有効期間が変更された場合、
 - ◆ そのユーザに関連する所属情報の所属期間を調整する必要があります。一時的にそのユーザが無効であるような期間がある場合、その期間は所属データを削除する必要があります。この場合、必要に応じて所属エンティティのデータを分割する場合があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。

2.4.1.2.1 パブリックグループ所属

組織所属のエンティティと実際のテーブルの構造を「表 2-20 パブリックグループ所属のデータ構造」と「図 2-44 パブリックグループ所属テーブルのER図」に示します。

表 2-20 パブリックグループ所属のデータ構造

項目	カラム名	PK	期間化	国際化	備考
ユーザコード	user_cd	○	—	—	
グループセットコード	public_group_set_cd	○	—	—	
グループコード	public_group_cd	○	—	—	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

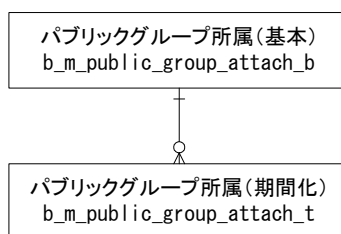


図 2-44 パブリックグループ所属テーブルの ER 図

2.4.1.3 パブリックグループの分類

パブリックグループセットおよびパブリックグループの分類に関連するER図を「図 2-45 パブリックグループの分類に関連するER図」に示します。

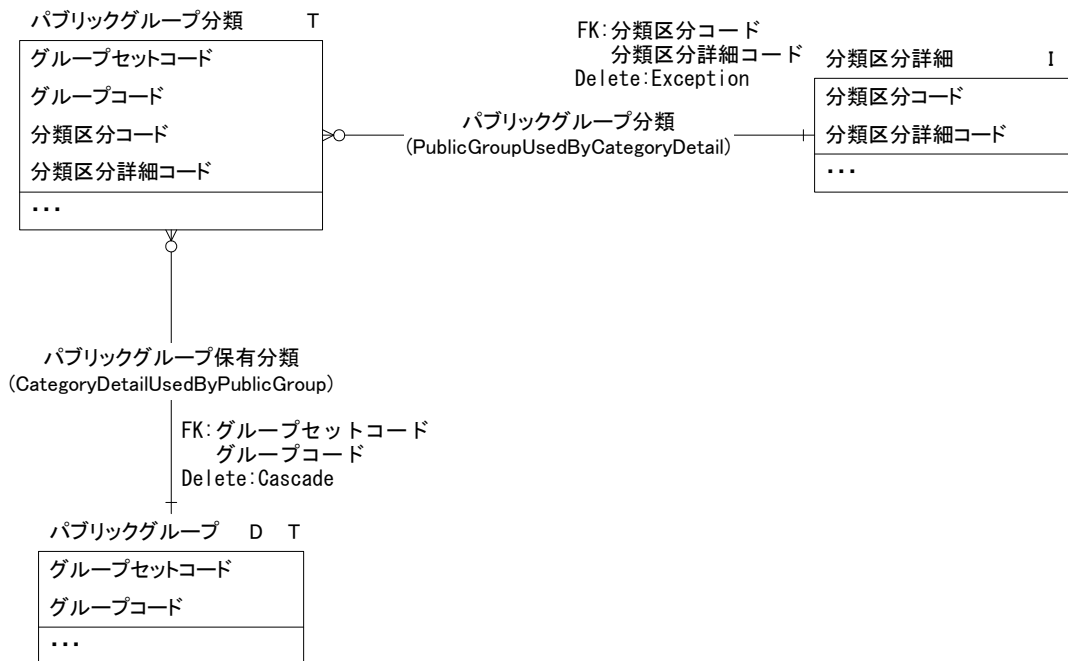


図 2-45 パブリックグループの分類に関連する ER 図

「図 2-45 パブリックグループの分類に関連するER図」において各エンティティは次のような役割があります。

- 分類区分詳細
分類の種類ごとに定義されている詳細を管理します。詳細は「2.2.1.2 分類区分詳細」を参照してください。
- パブリックグループ
パブリックグループセットやパブリックグループセット内に存在するパブリックグループの詳細情報を管理します。詳細は「2.4.1.1.3 パブリックグループ」を参照してください。
- パブリックグループ分類
パブリックグループセットまたはパブリックグループがどのような分類情報を持つか管理します。

また、「図 2-45 パブリックグループの分類に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 分類区分詳細エンティティのデータが削除された場合、
 - ◆ その分類区分詳細情報を使用しているパブリックグループ分類が存在している場合は例外を発生させます。
- パブリックグループエンティティのデータが削除された場合、
 - ◆ そのパブリックグループが持つパブリックグループ分類も削除する必要があります。
- パブリックグループ分類エンティティのデータを追加する場合、
 - ◆ 該当する分類区分詳細が存在している必要があります。
 - ◆ 該当するパブリックグループが存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。

2.4.1.3.1 パブリックグループ分類

パブリックグループ分類のエンティティと実際のテーブルの構造を「表 2-21 パブリックグループ分類のデータ構造」と「図 2-43 パブリックグループとユーザの所属に関連するER図」に示します。

表 2-21 パブリックグループ分類のデータ構造

項目	カラム名	PK	期間化	国際化	備考
グループセットコード	public_gorup_set_cd	○	○	—	
グループコード	public_group_cd	○	○	—	
分類区分コード	category_cls_cd	○	○	—	
分類コード	category_cd	○	○	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

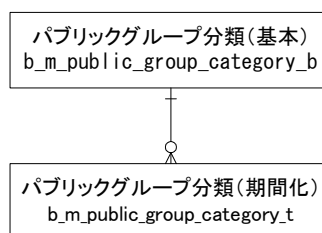


図 2-46 パブリックグループ分類テーブルの ER 図

2.4.2 パブリックグループセットおよびパブリックグループの情報

アプリケーション共通マスタでは、パブリックグループセットの詳細情報はパブリックグループの詳細情報として保存されます。グループセットコードとグループコードが一致するパブリックグループがパブリックグループセットを表現するパブリックグループとしてみなします。

パブリックグループセットとパブリックグループの違いは以下のとおりです。

- パブリックグループセットは複数のパブリックグループを持ちますが、パブリックグループはパブリックグループセットを持ちません。
- パブリックグループ構成のバージョン(「2.4.4 パブリックグループ構成のバージョン管理」を参照)はパブリックグループセット単位で管理されます。

パブリックグループは期間化管理され、さらに期間国際化されています。「図 2-47 パブリックグループの期間化」の例ではパブリックグループセットとそのパブリックグループの名称が期間管理されています。

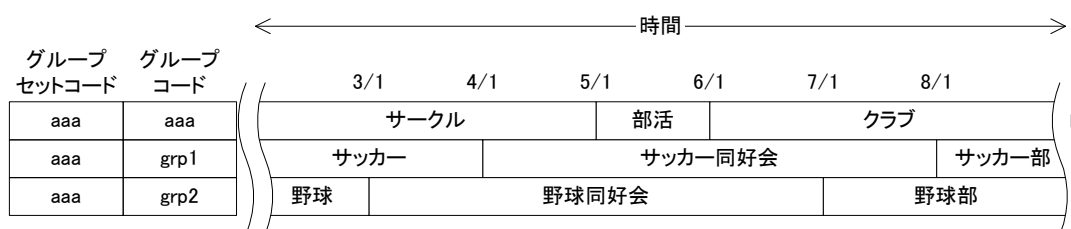


図 2-47 パブリックグループの期間化

「図 2-47 パブリックグループの期間化」の例をテーブルに登録したデータの状態を「図 2-48 パブリックグループの期間国際化(テーブル)」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

パブリックグループ
セットテーブル

グループ セットコード
aaa
bbb
⋮

パブリックグループテーブル

セットコード	グループ コード
aaa	aaa
aaa	grp1
aaa	grp2
bbb	bbb
⋮	⋮

組織テーブル(期間国際化)

グループ セットコード	グループ コード	期間コード	ローケル	パブリック グループ名
aaa	aaa	term_0	日本語	サークル
aaa	aaa	term_0	英語	Circle
aaa	aaa	term_1	日本語	部活
aaa	aaa	term_1	英語	Club
aaa	aaa	term_2	日本語	クラブ
aaa	aaa	term_2	英語	Club
aaa	grp1	term_0	日本語	サッカー
aaa	grp1	term_0	英語	Soccer
aaa	grp1	term_1	日本語	サッカー同好会
aaa	grp1	term_1	英語	Soccer
aaa	grp1	term_2	日本語	サッカー部
aaa	grp1	term_2	英語	Football
aaa	grp2	term_0	日本語	野球
aaa	grp2	term_0	英語	Baseball
aaa	grp2	term_1	日本語	野球同好会
aaa	grp2	term_1	英語	Baseball
aaa	grp2	term_2	日本語	野球部
aaa	grp2	term_2	英語	Baseball
bbb	bbb	term_0	日本語	ボランティア
⋮	⋮	⋮	⋮	⋮

組織テーブル(期間化)

グループ セットコード	グループ コード	期間コード	期間
aaa	aaa	term_0	~5/1
aaa	aaa	term_1	5/1~6/1
aaa	aaa	term_2	6/1~
aaa	grp1	term_0	~4/1
aaa	grp1	term_1	4/1~8/1
aaa	grp1	term_2	8/1~
aaa	grp2	term_0	~3/1
aaa	grp2	term_1	3/1~7/1
aaa	grp2	term_2	7/1~
bbb	bbb	term_0	~1/1
⋮	⋮	⋮	⋮

図 2-48 パブリックグループの期間国際化(テーブル)

「図 2-48 パブリックグループの期間国際化(テーブル)」では期間国際化されたデータも示しています。ここでは、グループセットコード"aaa"というパブリックグループセットが存在し、その詳細情報はグループコード"aaa"のパブリックグループに登録されていることがわかります。

2.4.3 パブリックグループの構成

アプリケーション共通マスタでは、パブリックグループの構成はパブリックグループセットをトップノードとしたツリー構造で表現できます。このツリー構造を表現するテーブルとしてパブリックグループ内包があります。パブリックグループ内包は次のような構造を持ちます。

- ツリー上のすべてのパブリックグループ(パブリックグループセットを含みます)から下方にたどることができるパブリックグループの情報を持ちます。この情報には親グループからみて子グループが何階層目に該当するかという情報が含まれます。
- 親グループと子グループが同じ場合、階層数は0となります。
- 親グループから見て直下の階層のパブリックグループは階層数1とみなされます。

パブリックグループの構成の例を「図 2-49 パブリックグループの構成例」に、この場合におけるテーブルの状態を「図 2-50 パブリックグループの構成内容」に示します。

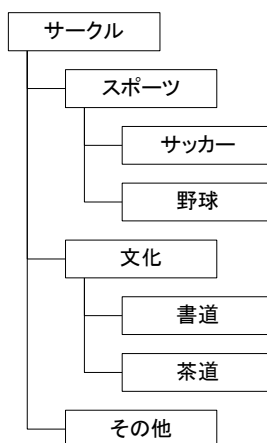


図 2-49 パブリックグループの構成例

パブリックグループ内包テーブル

パブリックグループセット	バージョン	親グループ	子グループ	階層
サークル	version_1	サークル	サークル	0
サークル	version_1	サークル	スポーツ	1
サークル	version_1	サークル	サッカー	2
サークル	version_1	サークル	野球	2
サークル	version_1	サークル	文化	1
サークル	version_1	サークル	書道	2
サークル	version_1	サークル	茶道	2
サークル	version_1	サークル	その他	1
サークル	version_1	スポーツ	スポーツ	0
サークル	version_1	スポーツ	サッカー	1
サークル	version_1	スポーツ	野球	1
サークル	version_1	サッカー	サッカー	0
サークル	version_1	野球	野球	0
サークル	version_1	文化	文化	0
サークル	version_1	文化	書道	1
サークル	version_1	文化	茶道	1
サークル	version_1	書道	書道	0
サークル	version_1	茶道	茶道	0
サークル	version_1	その他	その他	0

図 2-50 パブリックグループの構成内容

「図 2-50 パブリックグループの構成内容」では簡略化のためにパブリックグループセット名やパブリックグループ名がプライマリーの一部になっていますが、実際のテーブル構造ではグループセットコードおよびグループコードがプライマリーの一部として使用されています。また、バージョンについては「2.4.4 パブリックグループ構成のバージョン管理」を参照してください。

2.4.4 パブリックグループ構成のバージョン管理

アプリケーション共通マスタではこの構成を次のような単位で管理することができます。

- ある時点におけるパブリックグループの構成はパブリックグループセットごとに定義することができます。
- 同一のパブリックグループセットであっても期間によって構成を変更することができます。つまり、各パブリックグループセットごとに構成のバージョン管理ができます。

パブリックグループの構成がバージョン管理されている様子を「図 2-51 パブリックグループの構成とバージョン」に示します。また、この場合におけるパブリックグループセットとパブリックグループの登録内容を「図 2-52 パブリックグループセットとパブリックグループの情報」に、バージョンごとの構成を「図 2-53 パブリックグループの構成 (バージョン 1)」および「図 2-54 パブリックグループの構成 (バージョン 2)」に示します。

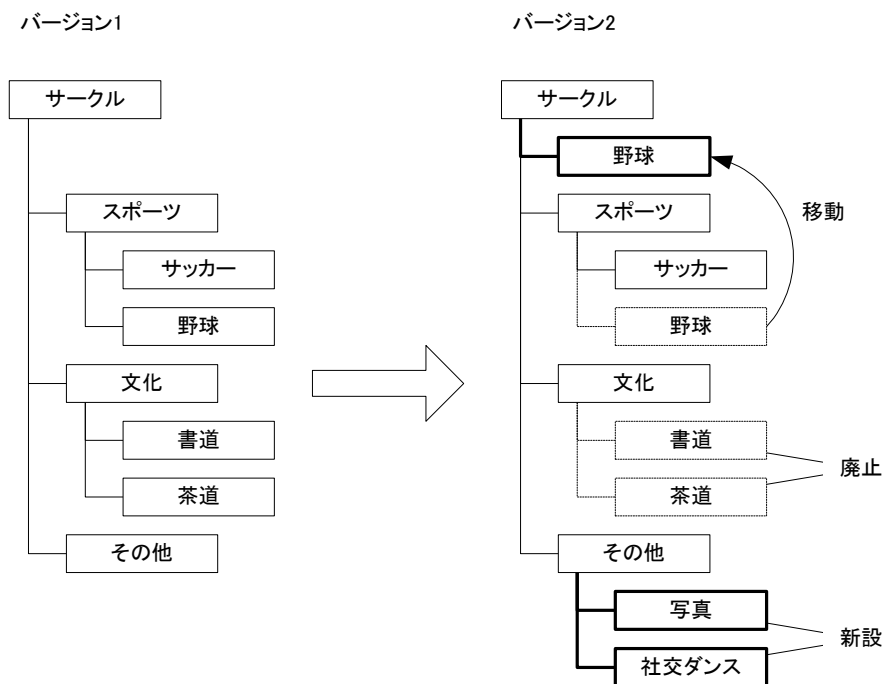


図 2-51 パブリックグループの構成とバージョン

パブリックグループセットテーブル		パブリックグループテーブル	
パブリックグループセット		パブリックグループセット	パブリックグループ
サークル		サークル	サークル
		サークル	スポーツ
		サークル	サッカー
		サークル	野球
		サークル	文化
		サークル	書道
		サークル	茶道
		サークル	その他
		サークル	写真
		サークル	社交ダンス

パブリックグループバージョンテーブル	
パブリックグループセット	バージョン
サークル	version_1
サークル	version_2

図 2-52 パブリックグループセットとパブリックグループの情報

パブリックグループ内包テーブル

パブリックグループセット	バージョン	親グループ	子グループ	階層
サークル	version_1	サークル	サークル	0
サークル	version_1	サークル	スポーツ	1
サークル	version_1	サークル	サッカー	2
サークル	version_1	サークル	野球	2
サークル	version_1	サークル	文化	1
サークル	version_1	サークル	書道	2
サークル	version_1	サークル	茶道	2
サークル	version_1	サークル	その他	1
サークル	version_1	スポーツ	スポーツ	0
サークル	version_1	スポーツ	サッカー	1
サークル	version_1	スポーツ	野球	1
サークル	version_1	サッカー	サッカー	0
サークル	version_1	野球	野球	0
サークル	version_1	文化	文化	0
サークル	version_1	文化	書道	1
サークル	version_1	文化	茶道	1
サークル	version_1	書道	書道	0
サークル	version_1	茶道	茶道	0
サークル	version_1	その他	その他	0

図 2-53 パブリックグループの構成(バージョン 1)

パブリックグループ内包テーブル

パブリックグループセット	バージョン	親組織	子組織	階層
サークル	version_2	サークル	サークル	0
サークル	version_2	サークル	野球	1
サークル	version_2	サークル	スポーツ	1
サークル	version_2	サークル	サッカー	2
サークル	version_2	サークル	文化	1
サークル	version_2	サークル	その他	1
サークル	version_2	サークル	写真	2
サークル	version_2	サークル	社交ダンス	2
サークル	version_2	野球	野球	0
サークル	version_2	スポーツ	スポーツ	0
サークル	version_2	スポーツ	サッカー	1
サークル	version_2	サッカー	サッカー	0
サークル	version_2	文化	文化	0
サークル	version_2	その他	その他	0
サークル	version_2	その他	写真	1
サークル	version_2	その他	社交ダンス	1
サークル	version_2	写真	写真	0
サークル	version_2	社交ダンス	社交ダンス	0

図 2-54 パブリックグループの構成(バージョン 2)

「図 2-52 パブリックグループセットとパブリックグループの情報」、「図 2-53 パブリックグループの構成(バージョン 1)」および「図 2-54 パブリックグループの構成(バージョン 2)」では簡略化のためにパブリックグループセット名やパブリックグループ名がプライマリキーの一部になっていますが、実際のテーブル構造ではグループセットコ

ードおよびグループコードがプライマリキーの一部として使用されています。

2.4.5 パブリックグループへの所属

アプリケーション共通マスタでは、ユーザはある時点において複数のパブリックグループセットやパブリックグループに所属することができます。この場合、ユーザは複数のパブリックグループセットをまたがって所属することも可能です。所属情報はパブリックグループセットやパブリックグループの期間化情報やバージョンには依存しません。「このユーザはこの時点ではこのパブリックグループに所属している」という情報のみが保存されます。

「図 2-55 パブリックグループの所属情報」に所属の例を示します。

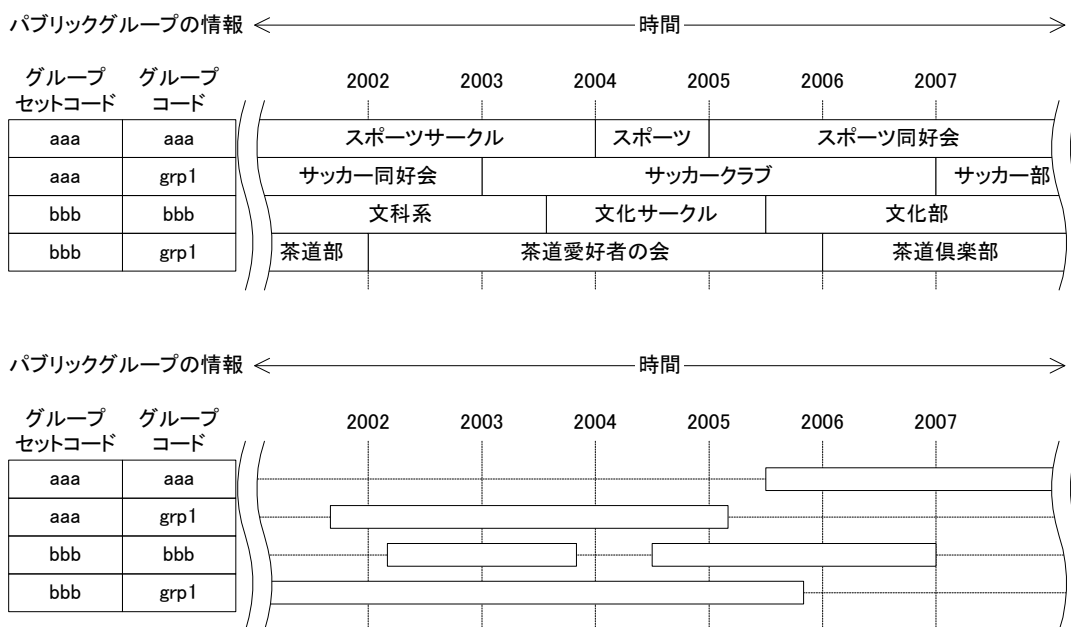


図 2-55 パブリックグループの所属情報

「図 2-55 パブリックグループの所属情報」で示されるユーザの経歴および予定は「表 2-22 ユーザのパブリックグループ所属経歴」のようになります。

表 2-22 ユーザのパブリックグループ所属経歴

時期	内容
	文科系の茶道部に入部
2001 下旬	スポーツサークルのサッカー同好会に入部
2002 初旬	文科系に所属 文科系の茶道部には籍を置いたまま
2003 下旬	文科系から退任 文化サークルの茶道愛好者の会(旧:文科系の茶道部)には籍を置いたまま
2004 中旬	文化サークルに再び所属
2005 中旬	スポーツ同好会に所属
2005 下旬	文化サークルの茶道愛好者の会(旧:文科系の茶道部)を退部
2007 末	文化部(旧:文化サークル)から退任

「図 2-55 パブリックグループの所属情報」および「表 2-22 ユーザのパブリックグループ所属経歴」の内容はアプリケーション共通マスタには「図 2-56 ユーザのパブリックグループ所属履歴(テーブル)」のように登録されます。

パブリックグループ所属テーブル

ユーザ	グループセットコード	グループコード
user0001	aaa	aaa
user0001	aaa	grp1
user0001	bbb	bbb
user0001	bbb	grp1

パブリックグループ所属テーブル(期間化)

ユーザ	グループセットコード	グループコード	期間コード	開始	終了
user0001	aaa	aaa	term0	2005中旬	(未定)
user0001	aaa	grp1	term0	2001下旬	2005初旬
user0001	bbb	bbb	term0	2002初旬	2003下旬
user0001	bbb	bbb	term1	2004中旬	2007末
user0001	bbb	grp1	term0	(未定)	2005下旬

図 2-56 ユーザのパブリックグループ所属履歴(テーブル)

2.4.6 分類

パブリックグループセットやパブリックグループはさまざまな角度から分類することができます。アプリケーション共通マスターでは、標準では以下のカテゴリでパブリックグループセットやパブリックグループを分類できます。

- 公開ゾーン

これらの詳細については「2.2.2 初期登録時の分類の内容」を参照してください。

パブリックグループセットやパブリックグループに対する分類の設定は期間化することが可能です。「図 2-57 パブリックグループの分類の変遷例」にその例を示します。

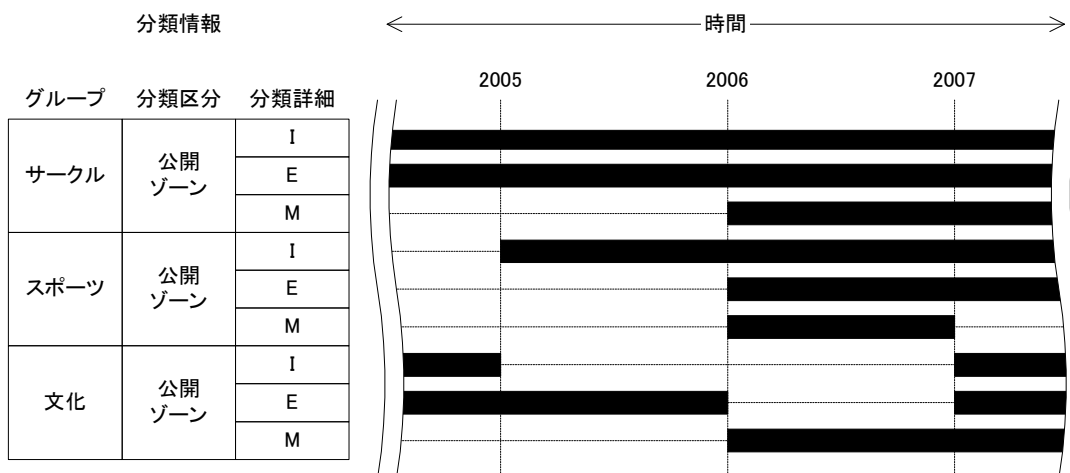


図 2-57 パブリックグループの分類の変遷例

「図 2-57 パブリックグループの分類の変遷例」がアプリケーション共通マスターのテーブルに登録されている状態を「図 2-58 パブリックグループの分類の変遷例(基本テーブル)」および「図 2-59 パブリックグループの分類の変遷例(期間化テーブル)」に示します。

パブリックグループテーブル

パブリックグループセット	パブリックグループ	分類区分	分類
サークル	サークル	ゾーン	イントラネット・ゾーン
サークル	サークル	ゾーン	エクストラネット・ゾーン
サークル	サークル	ゾーン	マーケットプレイス・ゾーン
サークル	スポーツ	ゾーン	イントラネット・ゾーン
サークル	スポーツ	ゾーン	エクストラネット・ゾーン
サークル	スポーツ	ゾーン	マーケットプレイス・ゾーン
サークル	文化	ゾーン	イントラネット・ゾーン
サークル	文化	ゾーン	エクストラネット・ゾーン
サークル	文化	ゾーン	マーケットプレイス・ゾーン

図 2-58 パブリックグループの分類の変遷例(基本テーブル)

パブリックグループ分類テーブル(期間化)

パブリックグループセット	パブリックグループ	分類区分	分類	期間コード	期間
サークル	サークル	ゾーン	イントラネット・ゾーン	term0	～
サークル	サークル	ゾーン	エクストラネット・ゾーン	term0	～
サークル	サークル	ゾーン	マーケットプレイス・ゾーン	term0	2006～
サークル	スポーツ	ゾーン	イントラネット・ゾーン	term0	2005～
サークル	スポーツ	ゾーン	エクストラネット・ゾーン	term0	2006～
サークル	スポーツ	ゾーン	マーケットプレイス・ゾーン	term0	2006～2007
サークル	文化	ゾーン	イントラネット・ゾーン	term0	～2005
サークル	文化	ゾーン	イントラネット・ゾーン	term1	2007～
サークル	文化	ゾーン	エクストラネット・ゾーン	term0	～2006
サークル	文化	ゾーン	エクストラネット・ゾーン	term1	2007～
サークル	文化	ゾーン	マーケットプレイス・ゾーン	term0	2006～

図 2-59 パブリックグループの分類の変遷例(期間化テーブル)

「図 2-58 パブリックグループの分類の変遷例(基本テーブル)」および「図 2-59 パブリックグループの分類の変遷例(期間化テーブル)」では簡略化のためにパブリックグループセット名、パブリックグループ名、分類区分名および分類名を使用していますが、実際のテーブル構造ではグループセットコード、グループコード、分類区分コードおよび分類コードになります。

2.5 プライベートグループ

会社やパブリックグループはシステムで決定する団体ですが、個人でグループを管理したい(アドレス帳等)場合はプライベートグループを利用することができます。

2.5.1 データ構造

プライベートグループに関連するER図を「図 2-60 プライベートグループに関連するER図」に示します。

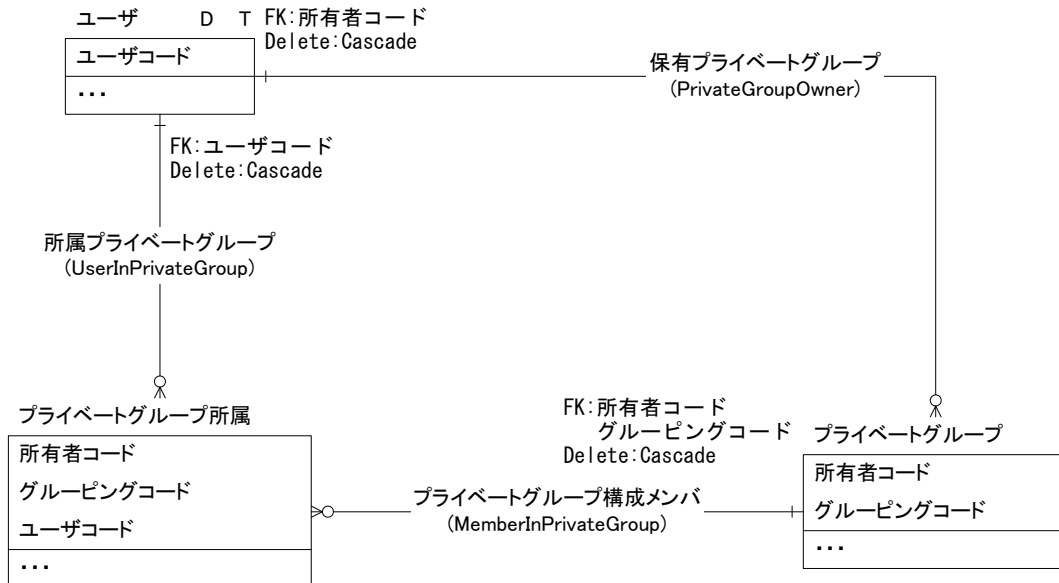


図 2-60 プライベートグループに関連する ER 図

「図 2-60 プライベートグループに関連するER図」において各エンティティは次のような役割があります。

- ユーザ
プライベートグループを持つユーザ、またはプライベートグループに所属するユーザを管理します。詳細は「2.1 ユーザ」を参照してください。
- プライベートグループ
プライベートグループの詳細情報を管理します。
- プライベートグループ所属
ユーザがどのプライベートグループに所属するかを管理します。

また、「図 2-60 プライベートグループに関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- ユーザエンティティのデータが削除された場合、
 - ◆ そのユーザに関連する所属情報も削除する必要があります。
 - ◆ そのユーザが保有するプライベートグループも削除する必要があります。
- プライベートグループエンティティのデータが削除された場合、
 - ◆ そのプライベートグループに所属する構成メンバも削除する必要があります。
- プライベートグループ所属エンティティのデータを追加する場合、
 - ◆ 該当するプライベートグループが存在している必要があります。
 - ◆ 該当するユーザが存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックはすべて自動的に行われます。

2.5.1.1 プライベートグループ

プライベートグループのエンティティと実際のテーブルの構造を「表 2-23 プライベートグループのデータ構造」と「図 2-61 プライベートグループテーブルのER図」に示します。

表 2-23 プライベートグループのデータ構造

項目	カラム名	PK	期間化	国際化	備考
所有者コード	owner_cd	○	—	—	
グループコード	private_group_cd	○	—	—	所有者コードに対して一意
グループ名	private_group_name	—	×	×	
グループ名カナ	private_group_name_syllabary	—	×	×	
グループ名英字	private_group_name_eng	—	×	×	
備考	notes	—	×	×	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

プライベートグループ(基本)
b_m_private_group_b

図 2-61 プライベートグループテーブルの ER 図

2.5.1.2 プライベートグループ所属

プライベートグループ所属のエンティティと実際のテーブルの構造を「表 2-24 プライベートグループ所属のデータ構造」と「図 2-62 プライベートグループ所属テーブルのER図」に示します。

表 2-24 プライベートグループ所属のデータ構造

項目	カラム名	PK	期間化	国際化	備考
所有者コード	owner_cd	○	—	—	
グループコード	private_group_cd	○	—	—	
ユーザコード	user_cd	○	—	—	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

プライベートグループ所属(基本)
b_m_private_group_attach_b

図 2-62 プライベートグループ所属テーブルの ER 図

2.5.2 プライベートグループの情報

プライベートグループは会社やパブリックグループと同様に団体を扱う情報です。しかし、プライベートグループは個人で扱えることを最優先の目的としているため、簡素な管理体系となっています。そのためプライベートグループは会社やパブリックグループとは以下の点で異なります。

- 階層構造を持ちません。
- バージョンによる管理がされません。
- 期間化や国際化がされていません。

ユーザはプライベートグループを複数定義することが可能です。複数のプライベートグループのグループコードが同じでも、プライベートグループを定義したユーザが異なれば別のプライベートグループとして扱われます。

2.5.3 プライベートグループへの所属

プライベートグループではユーザは「所有者」と「所属ユーザ」の二つの意味に捉えることができます。所有者と所属ユーザは次のように定義されます。

- 所有者
 - プライベートグループを定義したユーザ
- 所属ユーザ
 - プライベートグループに所属するユーザ

アプリケーション共通マスタでは、ユーザは複数のプライベートグループに所属することができます。また、所有者自身が所属ユーザとなることも可能です。この様子の例を「図 2-63 プライベートグループの所有者と所属ユーザ」に示します。

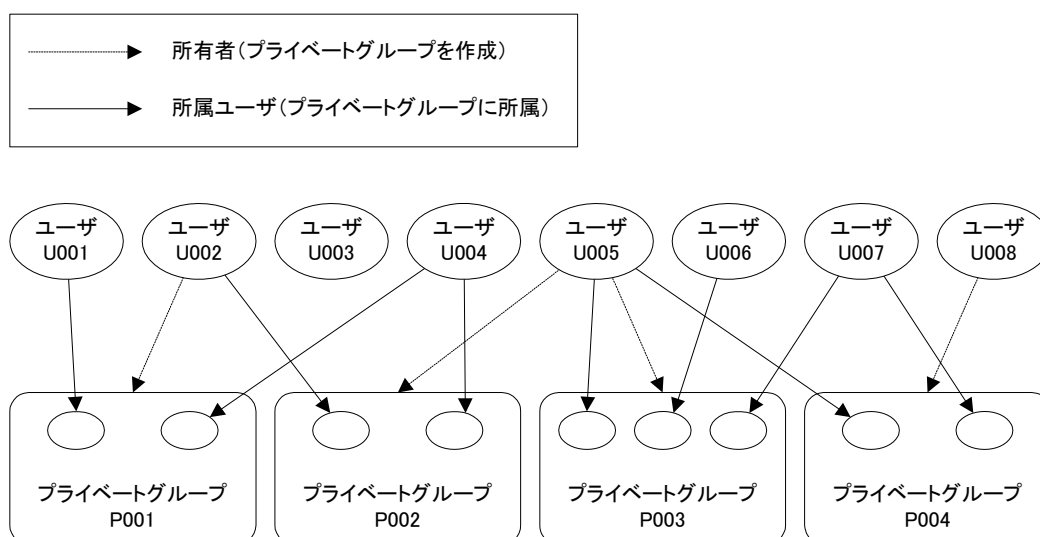


図 2-63 プライベートグループの所有者と所属ユーザ

「図 2-63 プライベートグループの所有者と所属ユーザ」の内容はアプリケーション共通マスタには「図 2-64 ユーザのプライベートグループ所属(テーブル)」のように登録されます。

ユーザテーブル	プライベートグループテーブル		プライベートグループ所属テーブル		
ユーザコード	所有者コード	グループコード	所有者コード	グループコード	ユーザコード
U001	U002	P001	U002	P001	U001
U002	U005	P001	U002	P001	U004
U003	U005	P002	U005	P001	U002
U004	U008	P001	U005	P001	U004
U005			U005	P002	U005
U006			U005	P002	U006
U007			U005	P002	U007
U008			U008	P001	U005
			U008	P001	U007

図 2-64 ユーザのプライベートグループ所属(テーブル)

2.6 商品

アプリケーション共通マスタでは商品に関連する情報を扱うことができます。取り扱える情報には以下のようなものがあります。

- 商品についての情報(名称、単価、製品型番、属性、カタログ情報など)
- 商品の分類情報
- 商品の取扱についての情報

2.6.1 データ構造

2.6.1.1 商品の構成

商品の構成に関するER図を「図 2-65 商品に関連するER図」に示します。

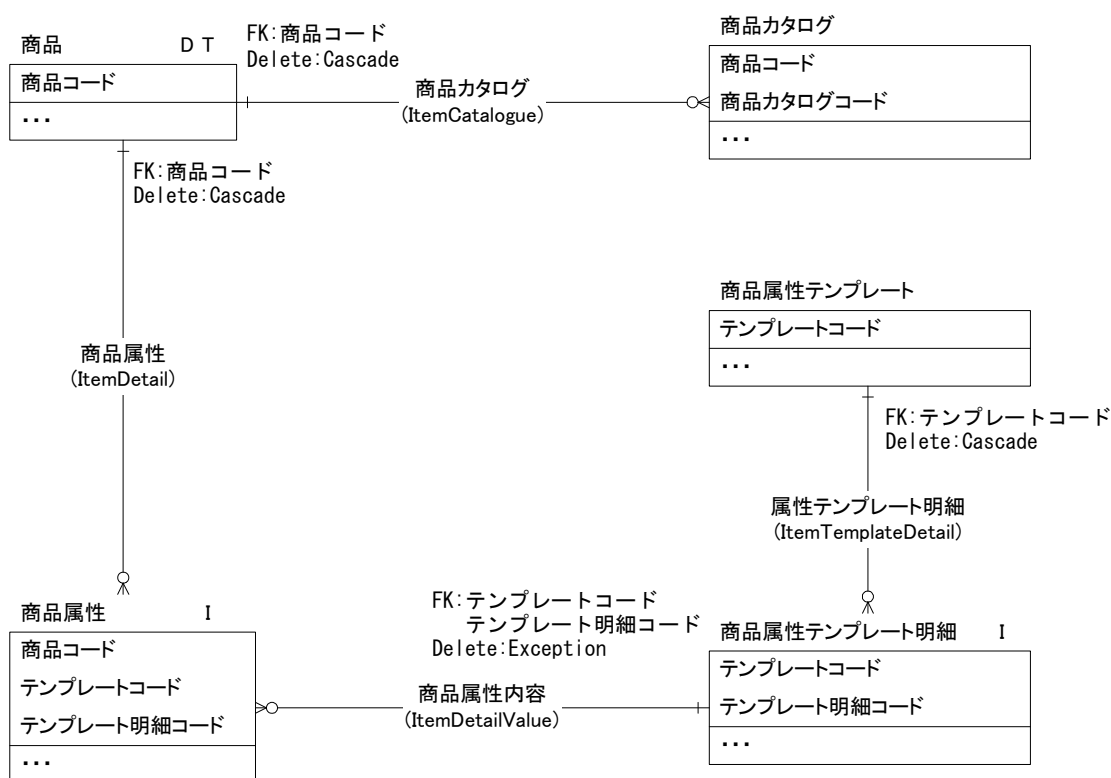


図 2-65 商品に関連する ER 図

「図 2-65 商品に関連するER図」において各エンティティは次のような役割があります。

- 商品
商品そのものの詳細情報を管理します。
- 商品属性テンプレート
商品に対して適用されるテンプレート情報を管理します。
- 商品属性テンプレート明細
商品属性テンプレートの明細情報を管理します。
- 商品属性
商品属性テンプレート明細に対する属性情報を管理します。
- 商品カタログ
商品に対して掲載されているカタログについての情報を管理します。

また、「図 2-65 商品に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 商品が削除された場合
 - ◆ その商品に対する商品属性情報も削除する必要があります。
 - ◆ その商品に対するカタログ情報も削除する必要があります。
- 商品属性テンプレートが削除された場合
 - ◆ その商品属性テンプレートの明細情報も削除する必要があります。
- 商品属性テンプレート明細が削除された場合
 - ◆ その商品属性テンプレート明細情報を参照する商品属性情報が存在してはいけません。もし参照する商品属性が存在する場合は、例外を発生させます。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。マネージャを利用するとさらに「図 2-65 商品に関連するER図」からは読み取れない下記チェックも行います。

- 商品に対して設定される期間が連続しているかどうかチェックします。もし期間が途切れている場合は、例外を発生させます。
- 商品情報の設定時に、単価や商品名称など期間管理されるデータが設定されているかどうかチェックします。もし期間管理データが設定されていない場合は、例外を発生させます。
- 商品属性テンプレート明細で設定されているデータ型(文字列型または数値型)と、その明細を参照する商品属性に対して設定されたデータの整合性をチェックします。つまり、商品属性テンプレート明細のデータ型が『文字列型』の場合は国際化された文字列データ、『数値型』の場合は数値データ以外の時は、例外を発生させます。
- 商品属性テンプレート明細を参照する商品属性情報が存在する場合、テンプレート明細の更新を制限します。つまり、ある商品から参照されている商品属性テンプレート明細のデータ型が更新された場合は、例外を発生させます。
- ある商品に対して、複数の商品属性テンプレートを適用しようとした場合は、例外を発生させます。
- 商品カタログ情報が削除されたとき、カタログ媒体そのものを削除します。

2.6.1.1.1 商品

商品のエンティティと実際のテーブルの構造を「表 2-25 商品のデータ構造」と「図 2-66 商品テーブルのER図」に示します。

表 2-25 商品のデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品コード	item_cd	○	—	—	
商品名	item_name	○	○	○	
商品通称	item_caption	—	○	○	
製品型番	product_model_code	—	×	×	
販売単価	sales_unit_price	—	○	×	
仕入単価	purchase_unit_price	—	○	×	
備考	notes	—	×	×	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

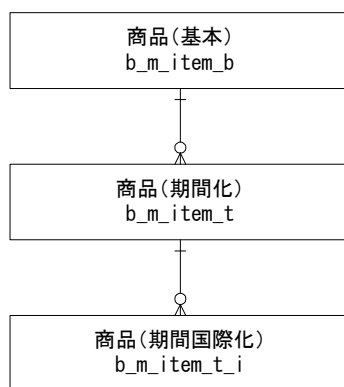


図 2-66 商品テーブルの ER 図

2.6.1.1.2 商品属性テンプレート

商品属性テンプレートのエンティティと実際のテーブル構造を「表 2-26 商品属性テンプレートのデータ構造」と「図 2-67 商品属性テンプレートテーブルのER図」に示します。

表 2-26 商品属性テンプレートのデータ構造

項目	カラム名	PK	期間化	国際化	備考
テンプレートコード	template_cd	○	—	—	
テンプレート名	template_name	—	×	×	
備考	notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	



図 2-67 商品属性テンプレートテーブルの ER 図

2.6.1.1.3 商品属性テンプレート明細

商品属性テンプレート明細のエンティティと実際のテーブル構造を「表 2-27 商品属性テンプレート明細のデータ構造」と「図 2-68 商品属性テンプレート明細テーブルのER図」に示します。

表 2-27 商品属性テンプレート明細のデータ構造

項目	カラム名	PK	期間化	国際化	備考
テンプレートコード	template_cd	○	—	—	
テンプレート明細コード	template_detail_cd	○	—	—	
テンプレート明細名	template_detail_name	—	×	○	
データ型	Data_type	—	×	×	
ソートキー	Sort_key	—	×	×	
備考	notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

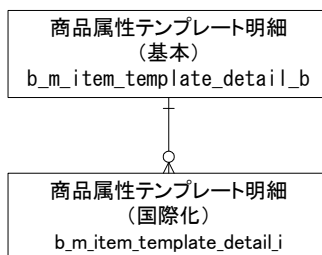


図 2-68 商品属性テンプレート明細テーブルの ER 図

2.6.1.1.4 商品属性

商品属性のエンティティと実際のテーブル構造を「表 2-28 商品属性のデータ構造」と「図 2-69 商品属性テーブルのER図」に示します。

表 2-28 商品属性のデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品コード	Item_cd	○	—	—	
テンプレートコード	template_cd	○	—	—	
テンプレート明細コード	template_detail_cd	○	—	—	
属性数値	number_value	—	×	×	
属性文字列	string_value	—	×	○	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

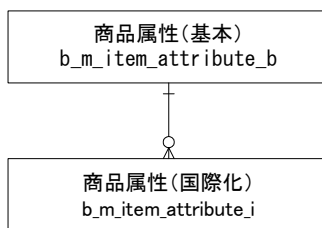


図 2-69 商品属性テーブルの ER 図

2.6.1.1.5 商品カタログ

商品カタログのエンティティと実際のテーブルの構造を「表 2-29 商品カタログのデータ構造」と「図 2-70 商品カタログテーブルのER図」に示します。

表 2-29 商品カタログのデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品コード	item_cd	○	—	—	
商品カタログコード	item_catalogue_cd	○	—	—	
媒体種類	catalogue_cls	—	×	×	
媒体	media	—	×	×	
備考	notes	—	×	×	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

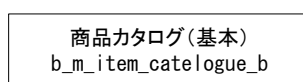


図 2-70 商品カタログテーブルの ER 図

2.6.1.2 商品カテゴリの構成

商品の構成に関するER図を「図 2-71 商品カテゴリに関連するER図」に示します。

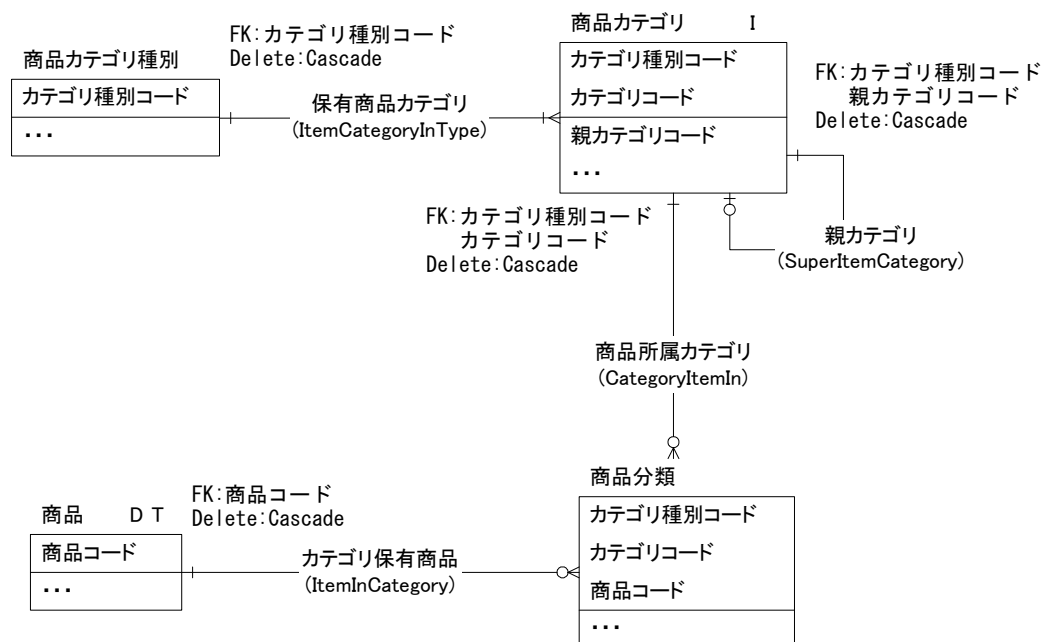


図 2-71 商品カテゴリに関連する ER 図

「図 2-71 商品カテゴリに関連するER図」において各エンティティは次のような役割があります。

- 商品カテゴリ種別
商品カテゴリの種類を管理します。
- 商品カテゴリ
商品カテゴリの名前およびカテゴリ構成についての情報を管理します。1つの商品カテゴリ(ルートとなる商品カテゴリは除く)はその属性情報に親カテゴリをもち、その構成は階層構造になります。
- 商品分類
商品がどのカテゴリに分類されるかを管理します。商品は複数のカテゴリに分類することができます。
- 商品
商品を管理します。詳細は「2.6.1.1.1 商品」を参照してください。

また、「図 2-71 商品カテゴリに関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 商品カテゴリ種別が削除された場合、
 - ◆ その商品カテゴリ種別を構成する商品カテゴリをすべて削除する必要があります。
 - ◆ その商品カテゴリ種別を構成する商品カテゴリに対して設定されている商品分類情報もすべて削除する必要があります。
- 商品カテゴリが削除された場合、
 - ◆ その商品カテゴリより下位の商品カテゴリをすべて削除する必要があります。
 - ◆ その商品カテゴリより下位の商品カテゴリに対して設定されている商品分類情報もすべて削除する必要があります。
- 商品が削除された場合、
 - ◆ その商品に関する分類情報をすべて削除する必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。マネージャを利用するとさらに「図 2-71 商品カテゴリに関連するER図」からは読み取れない下記のチェックも行います。

- 商品カテゴリの登録および更新時に階層構成が循環しないようにチェックします。
- 商品カテゴリ階層の末端となる商品カテゴリ以外に、商品进行分类させることはできません。
- 商品が分類されている商品カテゴリを親とする商品カテゴリを作成することはできません。

2.6.1.2.1 商品カテゴリ種別

商品カテゴリ種別のエンティティと実際のテーブルの構造を「表 2-30 商品カテゴリ種別のデータ構造」と「図 2-72 商品カテゴリ種別テーブルのER図」に示します。

表 2-30 商品カテゴリ種別のデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品カテゴリ種別コード	item_category_type_cd	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

商品カテゴリ種別(基本)
b_m_item_category_type_b

図 2-72 商品カテゴリ種別テーブルの ER 図

2.6.1.2.2 商品カテゴリ

商品カテゴリのエンティティと実際のテーブルの構造を「表 2-31 商品カテゴリのデータ構造」と「図 2-73 商品カテゴリテーブルのER図」に示します。

表 2-31 商品カテゴリのデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品カテゴリ種別コード	item_category_type_cd	○	—	—	
商品カテゴリコード	item_category_cd	○	—	—	
商品カテゴリ名	item_category_name	—	×	○	
親商品カテゴリコード	parent_item_category_cd	—	×	×	
備考	notes	—	×	×	
ソートキー	sort_key	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

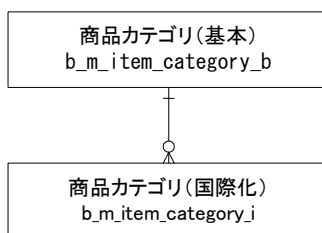


図 2-73 商品カテゴリテーブルの ER 図

2.6.1.2.3 商品分類

商品分類のエンティティと実際のテーブルの構造を「表 2-32 商品分類のデータ構造」と「図 2-74 商品分類テーブルのER図」に示します。

表 2-32 商品分類のデータ構造

項目	カラム名	PK	期間化	国際化	備考
商品カテゴリ種別コード	item_category_type_cd	○	—	—	
商品カテゴリコード	item_category_cd	○	—	—	
商品コード	item_cd	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

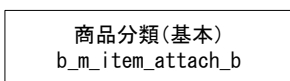


図 2-74 商品分類テーブルの ER 図

2.6.1.3 取扱の構成

取扱の構成に関するER図を「図 2-75 取扱の構成に関連するER図」に示します。

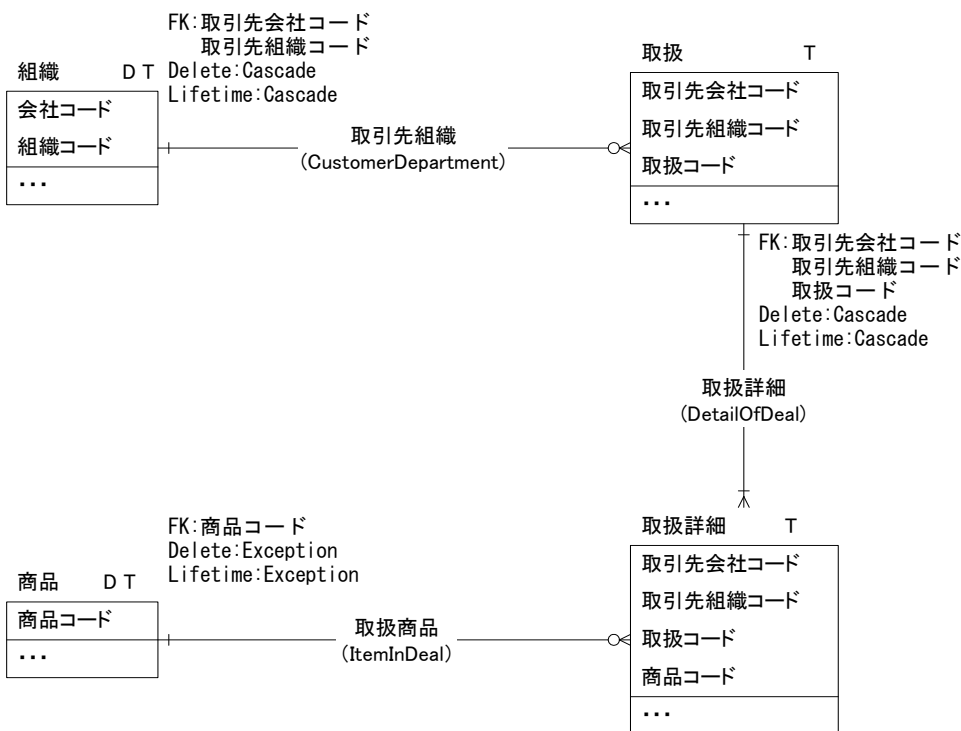


図 2-75 取扱の構成に関連する ER 図

「図 2-75 取扱の構成に関連するER図」において各エンティティは次のような役割があります。

- 取扱
 - 商品がどの取引先に対してどのように取り扱われるかを管理します。
- 取扱詳細
 - 取扱情報の中で実際に扱われる商品の情報を管理します。

- 組織
組織を管理します。詳細は「2.3.1.1.3 組織」を参照してください。
- 商品
商品を管理します。詳細は「2.6.1.1.1 商品」を参照してください。

また、「図 2-75 取扱の構成に関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 取扱エンティティのデータを追加する場合、
 - ◆ 追加される取扱情報で取引先となる組織が、存在している必要があります。
 - ◆ 追加される取扱情報の期間は、取引先となる組織の有効期間内に存在する必要があります。
- 取扱エンティティのデータを削除する場合、
 - ◆ 取扱に対する詳細情報も削除する必要があります。
- 取扱エンティティのデータの有効期間が変更(追加)された場合、
 - ◆ 変更(追加)された取扱の期間は、取引先となる組織の有効期間内に存在する必要があります。
- 取扱詳細エンティティのデータを追加する場合、
 - ◆ 追加される取扱詳細に対する取扱情報が、存在している必要があります。
 - ◆ 追加される取扱詳細の期間は、取扱情報の有効期間内に存在する必要があります。
 - ◆ 追加される取扱詳細で扱われる商品が、存在している必要があります。
 - ◆ 追加される取扱詳細の期間は、つねに商品の有効期間内に存在する必要があります。
- 取扱詳細エンティティのデータの有効期間が変更(追加)される場合、
 - ◆ 変更(追加)される取扱詳細の期間が、つねに取扱情報の有効期間内に存在する必要があります。
 - ◆ 追加される期間が、商品の有効期間内に存在する必要があります。
- 組織エンティティのデータを削除する場合、
 - ◆ その組織が取引先となる取扱情報が、すべて削除される必要があります。
- 組織エンティティのデータの有効期間が変更された場合、
 - ◆ その組織を取引先とする取扱情報の期間が、組織の有効期間内に存在するように調整される必要があります。
- 商品エンティティのデータを削除する場合、
 - ◆ その商品を扱っている取扱詳細が存在すれば場合は、例外を発生させる必要があります。
- 商品エンティティのデータの有効期間が削除された場合、
 - ◆ 削除された期間内で商品が取扱われていた場合は、例外を発生させる必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。マネージャを利用するとさらに「図 2-75 取扱の構成に関連するER図」からは読み取れない下記のチェックも行います。

- 期間情報をもたない取扱情報の作成はできません。
- 取扱詳細をもたない取扱情報は作成できません。
- 取扱と取扱詳細の期間は、つねに同期が取られます。つまり、取扱情報に対して設定される期間とその詳細情報に対して設定される期間は、つねに同じものになります。

2.6.1.3.1 取扱

取扱のエンティティと実際のテーブルの構造を「表 2-33 取扱のデータ構造」と「図 2-76 取扱テーブルのER図」に示します。

表 2-33 取扱のデータ構造

項目	カラム名	PK	期間化	国際化	備考
取引先会社コード	customer_company_cd	○	—	—	
取引先組織コード	customer_department_cd	○	—	—	
取扱コード	deal_cd	○	—	—	
取扱名	deal_name	—	×	×	
取扱種別	deal_type	—	×	×	
通貨コード	currency_cd	—	×	×	
単価	unit_price	—	○	×	
備考	Notes	—	×	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

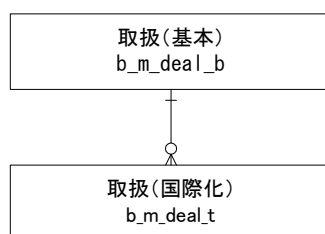


図 2-76 取扱テーブルのER図

2.6.1.3.2 取扱詳細

取扱詳細のエンティティと実際のテーブルの構造を「表 2-34 取扱詳細のデータ構造」と「図 2-77 取扱詳細テーブルのER図」に示します。

表 2-34 取扱詳細のデータ構造

項目	カラム名	PK	期間化	国際化	備考
取引先会社コード	customer_company_cd	○	—	—	
取引先組織コード	customer_department_cd	○	—	—	
取扱コード	deal_cd	○	—	—	
商品コード	item_cd	○	—	—	
単位コード	unit_cd	—	○	×	
数量	Amount	—	○	×	
単価	unit_price	—	○	×	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

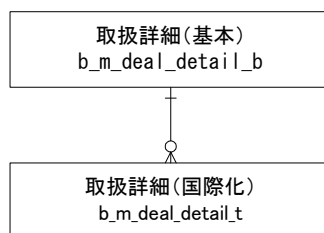


図 2-77 取扱詳細テーブルの ER 図

2.6.1.4 組織毎取扱パーミッションの構成

取扱情報と組織に対して付与されるパーミッションの構成に関するER図を「図 2-78 組織毎取扱パーミッションに関連するER図」に示します。

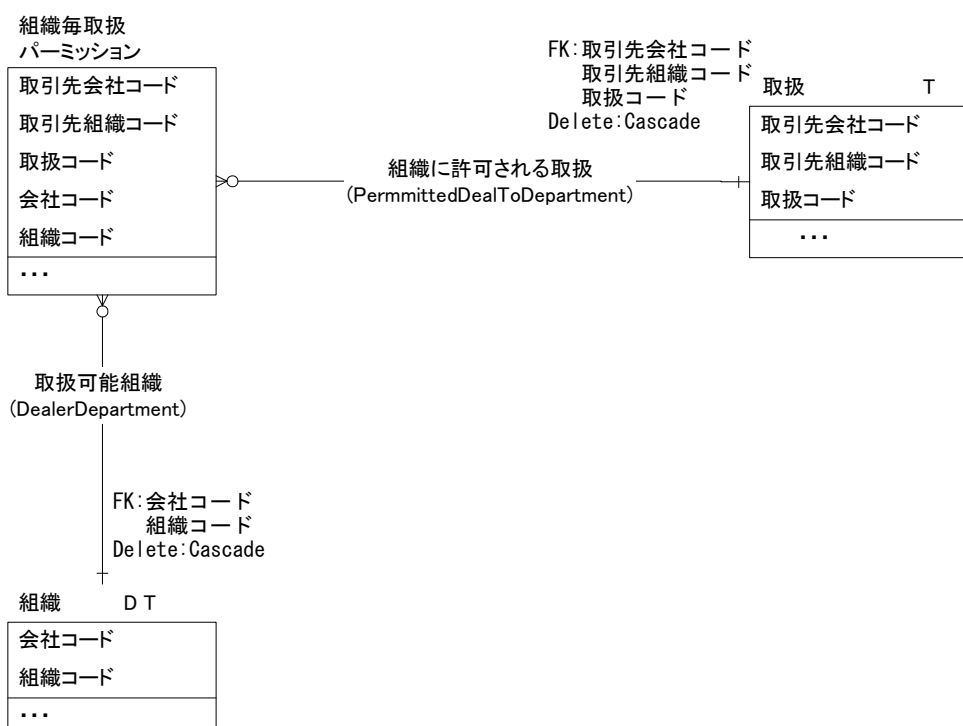


図 2-78 組織毎取扱パーミッションに関連する ER 図

「図 2-78 組織毎取扱パーミッションに関連するER図」において各エンティティは次のような役割があります。

- 組織毎取扱パーミッション
どの組織に対して、どの取扱情報の参照が許可されているのかを管理します。
- 取扱
商品に対する取扱情報を管理します。詳細は「2.6.1.3.1 取扱」を参照してください。
- 組織
組織を管理します。詳細は「2.3.1.1.3 組織」を参照してください。

また、「図 2-78 組織毎取扱パーミッションに関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 組織エンティティのデータが削除された場合、
 - ◆ その組織に対して付与された取扱パーミッション情報がすべて削除される必要があります。
- 取扱エンティティのデータが削除された場合、

- ◆ その取扱についてのパーミッション情報がすべて削除される必要があります。
- 組織毎取扱パーミッションエンティティのデータを追加する場合
 - ◆ 追加されるパーミッション情報に対する取扱情報が、存在している必要があります。
 - ◆ 追加されるパーミッション情報に対する組織情報が、存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。

2.6.1.4.1 組織毎取扱パーミッション

組織毎取扱パーミッションのエンティティと実際のテーブルの構造を「表 2-35 組織毎取扱パーミッションのデータ構造」と「図 2-79 組織毎取扱パーミッションのER図」に示します。

表 2-35組織毎取扱パーミッションのデータ構造

項目	カラム名	PK	期間化	国際化	備考
取引先会社コード	Customer_company_cd	○	—	—	
取引先組織コード	Customer_department_cd	○	—	—	
取扱コード	Deal_cd	○	—	—	
会社コード	company_cd	○	—	—	
組織コード	department_cd	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

組織毎取扱パーミッション
(基本)
b_m_dealer_department_b

図 2-79組織毎取扱パーミッションの ER 図

2.6.1.5 パブリックグループ毎取扱パーミッションの構成

取扱情報とパブリックグループに対して付与されるパーミッションの構成に関するER図を「図 2-80 パブリックグループ毎取扱パーミッションに関連するER図」に示します。

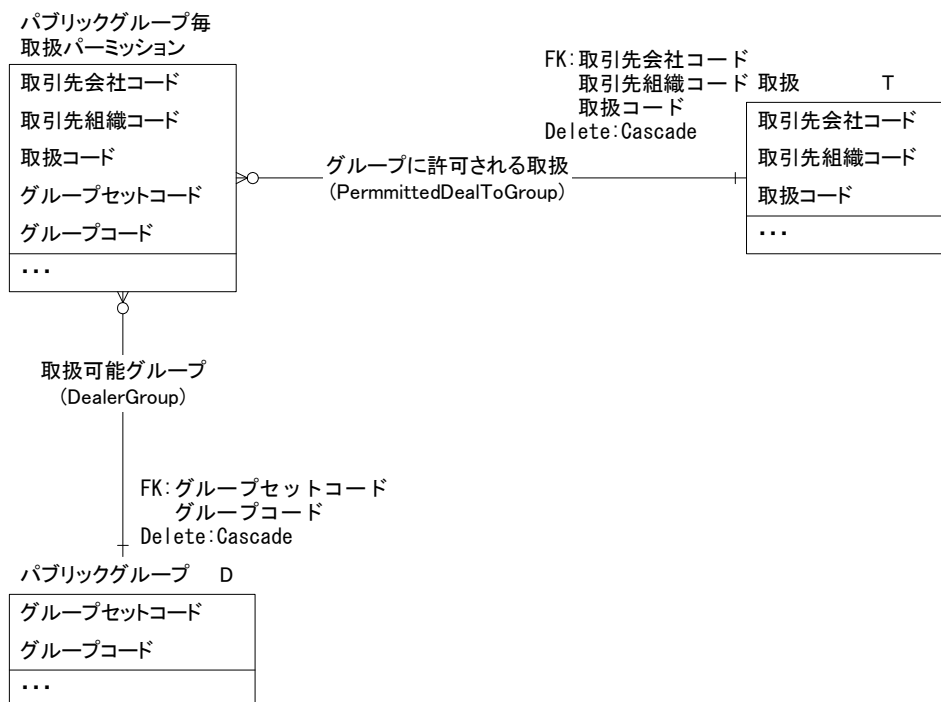


図 2-80 パブリックグループ毎取扱パーミッションに関連する ER 図

「図 2-80 パブリックグループ毎取扱パーミッションに関連するER図」において各エンティティは次のような役割があります。

- パブリックグループ毎取扱パーミッション
どのパブリックグループに対して、どの取扱情報の参照が許可されているのかを管理します。
- 取扱
商品に対する取扱情報を管理します。詳細は「2.6.1.3.1 取扱」を参照してください。
- パブリックグループ
パブリックグループを管理します。詳細は「2.4.1.1.3 パブリックグループ」を参照してください。

また、「図 2-80 パブリックグループ毎取扱パーミッションに関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- パブリックグループのエンティティのデータが削除された場合、
 - ◆ そのパブリックグループに対して付与された取扱パーミッション情報がすべて削除される必要があります。
- 取扱エンティティのデータが削除された場合、
 - ◆ その取扱についてのパーミッション情報がすべて削除される必要があります。
- パブリックグループ毎取扱パーミッションエンティティのデータを追加する場合
 - ◆ 追加されるパーミッション情報に対する取扱情報が、存在している必要があります。
 - ◆ 追加されるパーミッション情報に対するパブリックグループ情報が、存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。

2.6.1.5.1 パブリックグループ毎取扱パーミッション

パブリックグループ毎取扱パーミッションのエンティティと実際のテーブルの構造を「表 2-36 パブリックグループ毎取扱パーミッションのデータ構造」と「図 2-81 パブリックグループ毎取扱パーミッションのER図」に示します。

表 2-36パブリックグループ毎取扱パーミッションのデータ構造

項目	カラム名	PK	期間化	国際化	備考
取引先会社コード	customer_company_cd	○	—	—	
取引先組織コード	customer_department_cd	○	—	—	
取扱コード	deal_cd	○	—	—	
グループセットコード	public_group_set_cd	○	—	—	
グループコード	public_group_cd	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

パブリックグループ毎
取扱パーミッション(基本)
b_m_dealer_public_group_b

図 2-81パブリックグループ毎取扱パーミッションの ER 図

2.6.1.6 ロール毎取扱パーミッションの構成

取扱情報とロールに対して付与されるパーミッションの構成に関するER図を「図 2-82 ロール毎取扱パーミッションに関連するER図」に示します。

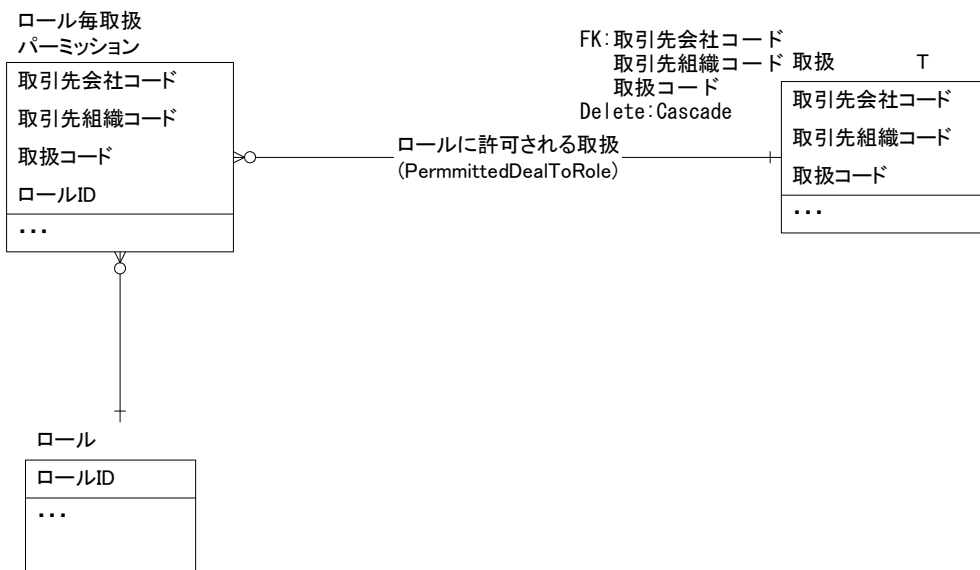


図 2-82 ロール毎取扱パーミッションに関連する ER 図

「図 2-82 ロール毎取扱パーミッションに関連するER図」において各エンティティは次のような役割があります。

- ロール毎取扱パーミッション
どのロールに対して、どの取扱情報の参照が許可されているのかを管理します。
- 取扱
組織を管理します。詳細は「2.6.1.3.1 取扱」を参照してください。
- ロール
システムで使用されるロールを管理します(アクセスセキュリティで管理されます)。

また、「図 2-82 ロール毎取扱パーミッションに関連するER図」から以下のような整合性チェックが必要であることが読み取れます。

- 取扱エンティティのデータが削除された場合、

- ◆ その取扱についてのパーミッション情報がすべて削除される必要があります。
- ロール毎取扱パーミッションエンティティのデータが追加される場合、
 - ◆ 追加されるパーミッション情報に対する取扱情報が、存在している必要があります。
 - ◆ 追加されるパーミッション情報に対するロール情報が、存在している必要があります。

マネージャ(マネージャの詳細は「3.2 マネージャ」を参照してください)のAPIを利用した場合、上記の整合性チェックは自動的に行われます。

2.6.1.6.1 ロール毎取扱パーミッション

ロール毎取扱パーミッションのエンティティと実際のテーブルの構造を「表 2-37 ロール毎取扱パーミッションのデータ構造」と「図 2-83 ロール毎取扱パーミッションのER図」に示します。

表 2-37ロール毎取扱パーミッションのデータ構造

項目	カラム名	PK	期間化	国際化	備考
取引先会社コード	customer_company_cd	○	—	—	
取引先組織コード	customer_department_cd	○	—	—	
取扱コード	deal_cd	○	—	—	
ロール ID	role_id	○	—	—	
最終更新者	record_user_cd	—	×	×	
最終更新日	record_date	—	×	×	

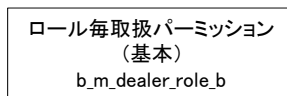


図 2-83ロール毎取扱パーミッションの ER 図

2.6.1.7 その他

2.6.1.7.1 通貨

通貨に関する情報を保持するエンティティおよびテーブルの構造を「表 2-38 通貨のデータ構造」と「図 2-84 通貨のER図」に示します。

表 2-38 通貨のデータ構造

項目	カラム名	PK	期間化	国際化	備考
通貨コード	currency_cd	○	—	—	
通貨名	currency_name	—	—	○	
備考	notes	—	—	—	

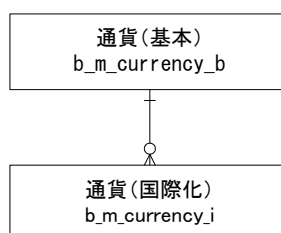


図 2-84通貨の ER 図

通貨コードおよび通貨名に関する情報は以下のマネージャを使用することにより取得することができます。

- スクリプト開発モデル
CurrencyManager
- J2EE 開発モデル
jp.co.intra_mart.foundation.datastore.application.domain.currency.CurrencyManager

通貨に関する情報の設定は直接データベースに対して操作を行う必要があります。

2.6.1.7.2 システム通貨

システム通貨は、システムで基準となる通貨を表します。「2.6.1.7.1 通貨」で設定された通貨の中から1つだけを選んで設定する必要があります。

システム通貨に関する情報を保持するエンティティおよびテーブルの構造を「表 2-39 システム通貨のデータ構造」と「図 2-85 システム通貨のER図」に示します。

表 2-39 システム通貨のデータ構造

項目	カラム名	PK	期間化	国際化	備考
通貨コード	currency_cd	○	—	—	



図 2-85 システム通貨の ER 図

システム通貨に関する情報は以下のマネージャを使用することにより取得することができます。

- スクリプト開発モデル
CurrencyManager
- J2EE 開発モデル
jp.co.intra_mart.foundation.datastore.application.domain.currency.CurrencyManager

また、システム通貨に関する情報の設定は直接データベースに対して操作を行う必要があります。

2.6.1.7.3 単位

商品の単位に関する情報を保持するエンティティおよびテーブルの構造を「表 2-40 単位のデータ構造」と「図 2-86 単位のER図」に示します。

表 2-40 単位のデータ構造

項目	カラム名	PK	期間化	国際化	備考
単位コード	unit_cd	○	—	—	
単位名	unit_name	—	—	—	

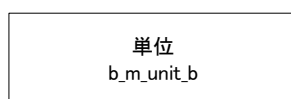


図 2-86 単位の ER 図

単位に関する情報の更新および参照は、以下のマネージャを使用することにより取得することができます。

- スクリプト開発モデル
ItemUnitManager
- J2EE 開発モデル
jp.co.intra_mart.foundation.datastore.application.domain.unit.ItemUnitManager

2.6.2 商品の情報

アプリケーション共通マスタでは、商品の情報は期間化管理され、さらに期間国際化されます。「図 2-87 商品の期間化」の例で商品名および単価情報が期間管理される様子を示します。

← 時間 →

商品	属性	1990/10	2001/10	2002/10	2004/10
item001	名称	製品A		製品B	
	仕入単価	300円		350円	400円
	販売単価	350円		450円	

図 2-87 商品の期間化

「図 2-87 商品の期間化」の例をテーブルに登録したデータの状態を「図 2-88 商品の構成内容」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

商品テーブル

商品コード*	製品型番
item001	model001
⋮	⋮

商品テーブル(期間化)

商品コード*	期間コード*	期間	仕入単価	販売単価
item001	term_0	1990/10~2001/10	300円	350円
item001	term_1	2001/10~2002/10	350円	450円
item001	term_2	2002/10~2004/10	350円	450円
item001	term_3	2004/10~9999/12	400円	450円
⋮	⋮	⋮	⋮	⋮

商品テーブル(期間国際化)

会社コード*	期間コード*	ローケル	組織名
item001	term_0	日本語	製品A
item001	term_0	英語	Product A
item001	term_1	日本語	製品A
item001	term_1	英語	Product A
item001	term_2	日本語	製品B
item001	term_2	英語	Product B
item001	term_3	日本語	製品B
item001	term_3	英語	Product B
⋮	⋮	⋮	⋮

図 2-88 商品の構成内容

2.6.3 商品属性テンプレート

アプリケーション共通マスタでは、商品固有の属性を定義するためのテンプレートを作成することができます。商品属性テンプレートおよび商品の属性値の情報は以下のように管理されます。

- テンプレートには、属性名とその属性値に対するデータ型を定義する
- テンプレートとして定義される属性名は国際化項目である
- テンプレートとして定義される属性のデータ型には、『文字列型』または『数値型』のいずれかを指定する
- 商品は、作成されたテンプレートを任意に1つだけ適用することができる（適用しなくても構わない）
- 属性値のデータ型が文字列型の場合、その商品属性は国際化項目となる

商品属性テンプレートの構成の例を「図 2-89 商品属性テンプレートの構成」に、この場合におけるテーブルの

状態を「図 2-90 商品属性テンプレートの構成内容」に示します。

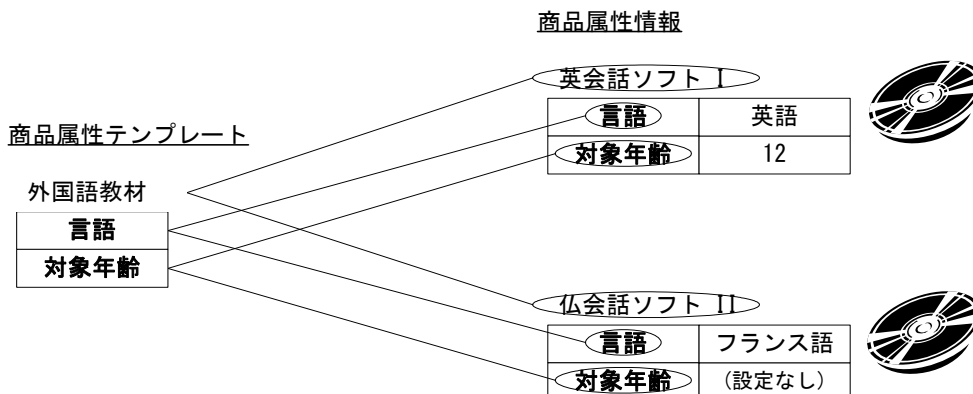


図 2-89 商品属性テンプレートの構成

商品属性テンプレートテーブル

テンプレートコード	テンプレート名
teach_material	外国語教材

商品属性テンプレート明細テーブル

テンプレートコード	テンプレート明細コード	データ型
teach_material	lang	文字列型
teach_material	age	数値型

商品属性テンプレート明細テーブル(国際化)

テンプレートコード	テンプレート明細コード	ロケール	テンプレート明細名
teach_material	lang	日本語	言語
teach_material	lang	英語	Language
teach_material	age	日本語	対象年齢
teach_material	age	英語	Age

商品属性テーブル

商品コード	テンプレートコード	テンプレート明細コード	属性数値
english_01	teach_material	lang	
english_01	teach_material	age	12
french_02	teach_material	lang	

商品属性テーブル(国際化)

商品コード	テンプレートコード	テンプレート明細コード	ロケール	属性文字列
english_01	teach_material	lang	日本語	英語
english_01	teach_material	lang	英語	English
french_02	teach_material	lang	日本語	フランス語
french_02	teach_material	lang	英語	French

図 2-90 商品属性テンプレートの構成内容

2.6.4 商品の分類

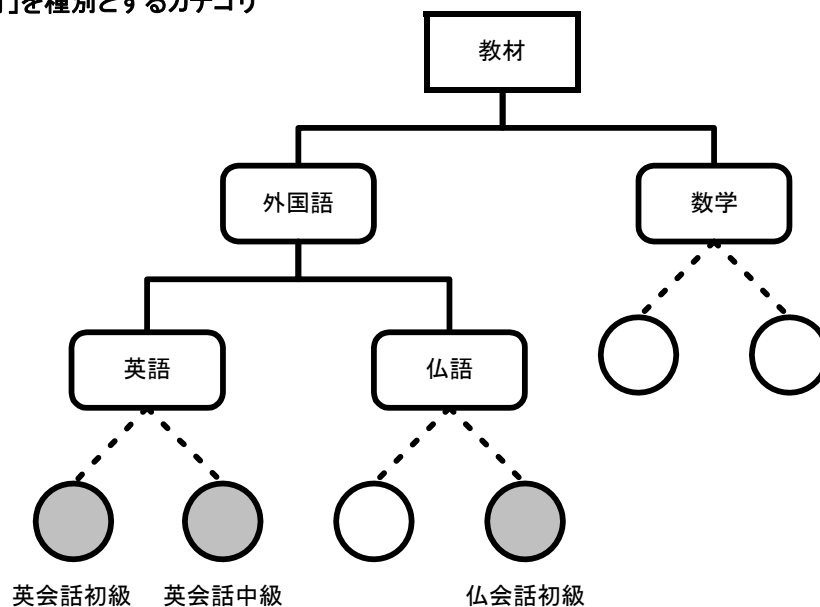
商品はカテゴリに分類することにより、複数の異なる視点から参照することができるようになります。アプリケーション共通マスタでは、商品カテゴリおよび商品の分類は以下のように管理されます

- 商品カテゴリは、商品カテゴリ種別をルートに持つ階層構造をもつ
- 商品が実際に分類される商品カテゴリは末端のカテゴリのみである
- 1商品は複数のカテゴリに分類できる

「図 2-91 商品カテゴリと分類」の例で商品がカテゴリに分類される様子を示します。ここでは、『教材』と『媒体』と

いうふたつのカテゴリ種別が存在します。

(1) 「教材」を種別とするカテゴリ



(2) 「媒体」を種別とするカテゴリ

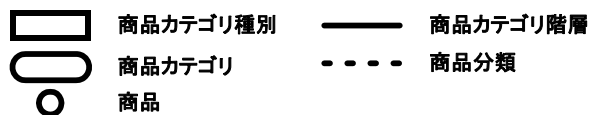
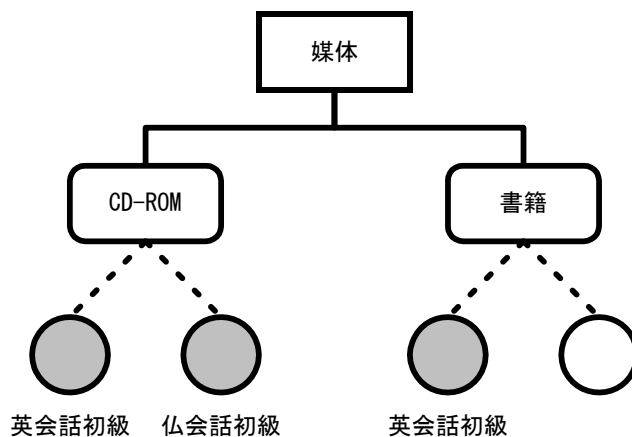


図 2-91 商品カテゴリと分類

「図 2-91 商品カテゴリと分類」の例をテーブルに登録したデータの状態を「図 2-92 商品カテゴリの構成内容」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

商品カテゴリ種別テーブル

カテゴリ種別コード
material
media

商品テーブル

商品コード	
english_b	(…英会話初級)
english_m	(…英会話中級)
french_b	(…仏会話初級)

商品分類テーブル

カテゴリ種別コード	カテゴリコード	商品コード
material	english	english_b
material	english	english_m
material	french	french_b
media	cdrom	english_b
media	cdrom	french_b
media	book	english_m

商品カテゴリテーブル

カテゴリ種別コード	カテゴリコード	親カテゴリコード
material	material	
material	foreign_lang	material
material	mathematics	material
material	english	foreign_lang
material	french	foreign_lang
media	media	
media	cdrom	media
media	book	media

商品カテゴリテーブル(国際化情報)

カテゴリ種別コード	カテゴリコード	ロケール	カテゴリ名
material	material	日本語	教材
material	material	英語	Teaching Material
material	foreign_lang	日本語	外国語
material	foreign_lang	英語	Foreign Language
material	mathematics	日本語	数学
material	mathematics	英語	Mathematics
material	english	日本語	英語
material	english	英語	English
material	french	日本語	仏語
material	french	英語	French
media	media	日本語	媒体
media	media	英語	Media
media	cdrom	日本語	CD-ROM
media	cdrom	英語	CD-ROM
media	book	日本語	書籍
media	book	英語	Book

図 2-92 商品カテゴリの構成内容

2.6.5 商品の取扱

2.6.5.1 取扱情報

アプリケーション共通マスタでは、「どの取引先に対してどの商品をいくらで売るか」または「どの取引先からどの商品をいくらで仕入れるか」といった商品の取扱情報をもつことができます。取扱情報は以下のように管理されます。

- 取扱情報は、取扱コードと取引先となる組織の2つで一意に特定される
- 1つの取扱情報で複数の商品を扱うことができる（セット販売など）
- 取扱情報は期間管理され、その期間は取引先となる組織および扱われる商品の期間に依存する

「図 2-93 商品の取扱情報」の例で組織、商品、取扱情報の関係を示します。

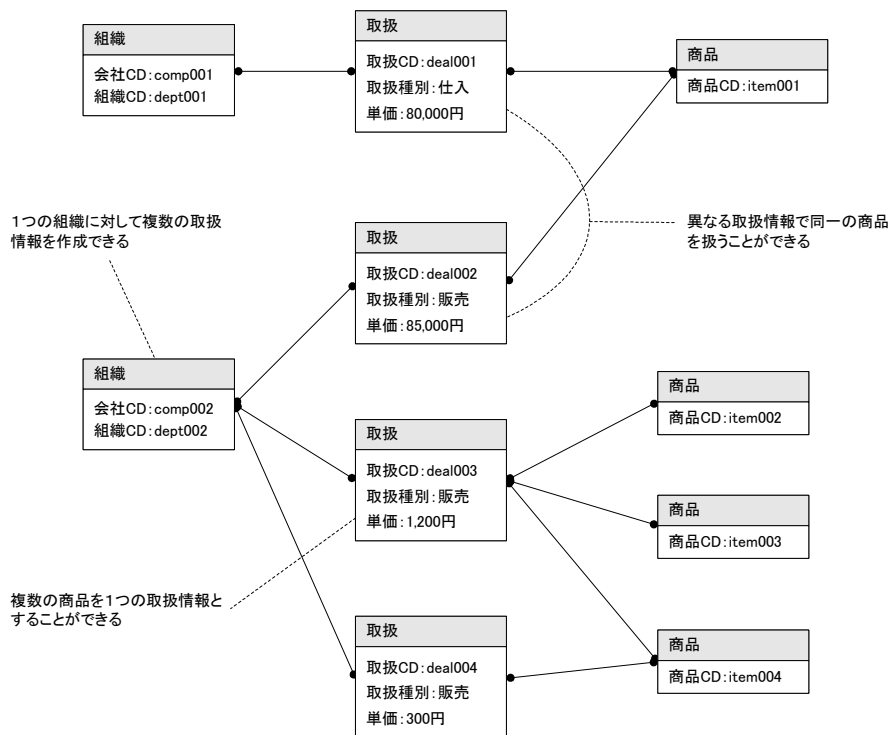


図 2-93 商品の取扱情報

「図 2-93 商品の取扱情報」の例をテーブルに登録したデータの状態を「図 2-94 取扱情報の構成内容」に示します。ここでは表示を簡略化するため、主要な項目のみを抜き出しています。

組織テーブル

会社コード	組織コード
comp001	dept001
comp002	dept002

取扱テーブル

取引先会社コード	取引先組織コード	取扱コード	取扱種別
comp001	dept001	deal001	仕入
comp002	dept002	deal002	販売
comp002	dept002	deal003	販売
comp002	dept002	deal004	販売

商品テーブル

商品コード	取引先会社コード	取引先組織コード	取扱コード	期間コード	期間	単価
item001	comp001	dept001	deal001	term0	1990/01/01～9999/12/31	80,000
item002	comp002	dept002	deal002	term0	2000/01/01～9999/12/31	85,000
item003	comp002	dept002	deal003	term0	2005/01/01～9999/12/31	1,200
item004	comp002	dept002	deal004	term0	2002/01/01～9999/12/31	300

取扱テーブル(期間化)

取引先会社コード	取引先組織コード	取扱コード	期間コード	期間	単価
comp001	dept001	deal001	term0	1990/01/01～9999/12/31	80,000
comp002	dept002	deal002	term0	2000/01/01～9999/12/31	85,000
comp002	dept002	deal003	term0	2005/01/01～9999/12/31	1,200
comp002	dept002	deal004	term0	2002/01/01～9999/12/31	300

取扱明細テーブル

取引先会社コード	取引先組織コード	取扱コード	商品コード
comp001	dept001	deal001	item001
comp002	dept002	deal002	item001
comp002	dept002	deal003	item002
comp002	dept002	deal003	item003
comp002	dept002	deal003	item004
comp002	dept002	deal004	item004

取扱明細テーブル(期間化)

取引先会社コード	取引先組織コード	取扱コード	商品コード	期間コード	期間
comp001	dept001	deal001	item001	term0	1990/01/01～9999/12/31
comp002	dept002	deal002	item001	term0	2000/01/01～9999/12/31
comp002	dept002	deal003	item002	term0	2005/01/01～9999/12/31
comp002	dept002	deal003	item003	term0	2005/01/01～9999/12/31
comp002	dept002	deal003	item004	term0	2005/01/01～9999/12/31
comp002	dept002	deal004	item004	term0	2002/01/01～9999/12/31

図 2-94 取扱情報の構成内容

2.6.5.2 取扱権限

アプリケーション共通マスタでは、取扱情報に対して参照のための権限を付与することができます。取扱権限を付与することにより、ある取扱情報(商品の取引情報)を特定の部署だけが利用できるようなるといったアクセス制御ができるようになります。取扱権限は以下のように管理されます。

- 取扱権限は任意の組織に対して付与することができる
- 取扱権限は任意のパブリックグループに対して付与することができる
- 取扱権限は任意のロールに対して付与することができる
-

取扱権限の構成の例を「図 2-95 取扱情報と権限」に、この場合におけるテーブルの状態を「図 2-96 取扱権限情報の構成内容」に示します。

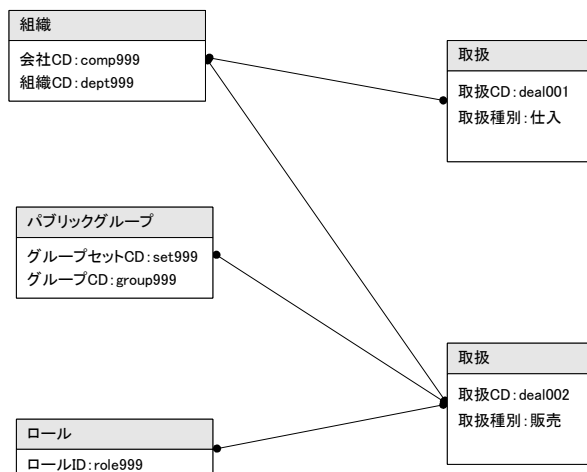


図 2-95 取扱情報と権限

組織テーブル

会社コード*	組織コード*
comp001	dept001
comp002	dept002
comp999	dept999

組織毎取扱パーミッションテーブル

取引先会社コード*	取引先組織コード*	取扱コード*	会社コード*	組織コード*
comp001	dept001	deal001	comp999	dept999
comp002	dept002	deal002	comp999	dept999

パブリックグループテーブル

セットコード*	グループコード*
set999	group999

パブリックグループ毎取扱パーミッションテーブル

取引先会社コード*	取引先組織コード*	取扱コード*	セットコード*	グループコード*
comp002	dept002	deal002	set999	group999

ロール

ロールID
role999

ロール毎取扱パーミッションテーブル

取引先会社コード*	取引先組織コード*	取扱コード*	ロールID
comp002	dept002	deal002	role999

取扱テーブル

取引先会社コード*	取引先組織コード*	取扱コード*	取扱種別
comp001	dept001	deal001	仕入
comp002	dept002	deal002	販売

図 2-96 取扱権限情報の構成内容

3 API

3.1 アプリケーション共通マスタとアプリケーションデータの整合性について

アプリケーション共通マスタに登録、更新および削除を行う場合、データの整合性を保つ必要があります。また、アプリケーション共通マスタは比較的汎用的に設計されていますが、独自に拡張したい場合もあります。

この場合、直接データベースにデータを登録、更新および削除したり、テーブル構造を変更したりすると様々な不具合が発生したり、試験工数が増加したりすることが考えられます。

これらの問題を解決するため、intra-mart ではアプリケーション共通マスタに対する API を用意しています。

API を利用することで、以下の利点があります。

- アプリケーション共通マスタ間の整合性の考慮は不要
- テーブルの拡張に対して同期を取ることが可能

各アプリケーションおよびテンプレートは、アプリケーション共通マスタで使用しているテーブルに対してデータを直接登録、更新および削除してはいけません。アプリケーション共通マスタで使用しているテーブルのデータに変更する処理を行う場合、必ずこの API で行う必要があります。

3.1.1 アプリケーション共通マスタ間の整合性の考慮は不要

アプリケーション共通マスタはデータベース上の複数のテーブルから構成されますが、API を利用することで整合性の考慮は不要となります。組織の登録、更新および削除時には組織テーブル、組織テーブル(期間化)、組織テーブル(期間国際化)、組織内包テーブルという 4 つのテーブルにアクセスする必要がありますが、API を利用すると 1 回の呼び出しで両方のテーブルが更新されます(「図 3-1 アプリケーション共通マスタ間の整合性の考慮は不要」を参照してください)。

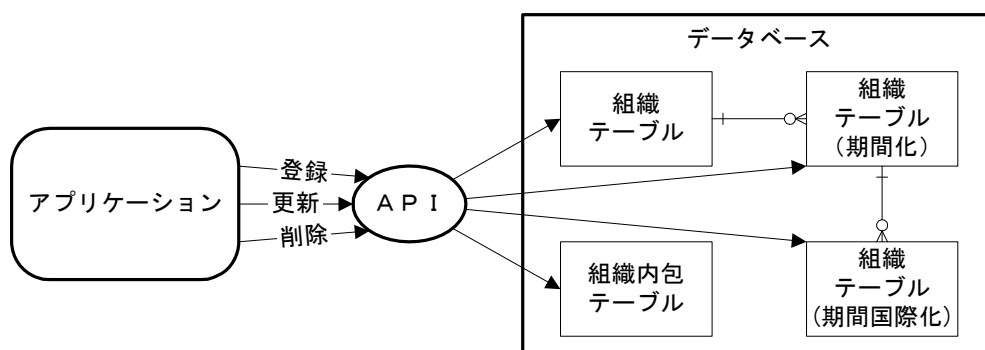


図 3-1 アプリケーション共通マスタ間の整合性の考慮は不要

3.1.2 テーブルの拡張に対して同期を取ることが可能

アプリケーション共通マスタの変更に対して、各アプリケーションのマスタおよびトランザクションデータの整合性を維持するために、追加、更新および削除毎に実行すべきリスナを設定することが出来ます。

アプリケーション共通マスタに対する変更は intra-mart に付属のメンテナンス画面を使用すると、インストールされ

ているすべてのプロダクトに対するデータ整合性が維持されます。アプリケーションで組織種類を追加し、データを登録および編集する場合は提供される共通関数を使用します(これにより、定義されたリスナが起動します)。

3.2 マネージャ

アプリケーション共通マスタのデータを変更または取得するとき、マネージャと呼ばれるAPI群を使用します。アプリケーション共通マスタのどのデータを扱うかによって使用するマネージャが変わります。「表 3-1 マネージャ一覧」にその一覧を示します。APIの詳細はAPIリストを参照してください。

表 3-1 マネージャ一覧

取り扱い情報	マネージャ	
	スクリプト開発モデル API	J2EE 開発モデル API
ユーザ	UserManager	jp.co.intra_mart.foundation.datastore.application.domain.user.UserManager
会社/組織	CompanyManager	jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyManager
パブリックグループ	PublicGroupManager	jp.co.intra_mart.foundation.datastore.application.domain.public_group.PublicGroupManager
プライベートグループ	PrivateGroupManager	jp.co.intra_mart.foundation.datastore.application.domain.private_group.PrivateGroupManager
分類	CategoryManager	jp.co.intra_mart.foundation.datastore.application.domain.category.CategoryManager
商品	ItemManager	jp.co.intra_mart.foundation.datastore.application.domain.item.ItemManager
商品テンプレート	ItemTemplateManager	jp.co.intra_mart.foundation.datastore.application.domain.item_template.ItemTemplateManager
商品カテゴリ	ItemCategoryManager	jp.co.intra_mart.foundation.datastore.application.domain.item_category.ItemCategoryManager
取扱	DealManager	jp.co.intra_mart.foundation.datastore.application.domain.deal.DealManager
取扱パーミッション	DealPermissionManager	jp.co.intra_mart.foundation.datastore.application.domain.deal.DealPermissionManager

また、各マネージャが使用するエンティティおよびエンティティを構成するテーブルを「表 3-2 マネージャが使用するエンティティとテーブルの一覧」に示します。

表 3-2 マネージャが使用するエンティティとテーブルの一覧

マネージャ	エンティティ	テーブル
UserManager	ユーザ	b_m_user_b
		b_m_user_t
		b_m_user_t_i
CompanyManager	会社	b_m_company_b
	組織	b_m_department_b
		b_m_department_t
		b_m_department_t_i
	会社構成バージョン	b_m_company_version_b
	組織内包	b_m_department_inclusion_b
	役職	b_m_company_post_b
		b_m_company_post_t
		b_m_company_post_t_i
	会社組織所属	b_m_department_attach_b
		b_m_department_attach_t
会社組織主所属	b_m_department_main_b	
	b_m_department_main_t	
会社組織分類	b_m_company_category_b	
	b_m_company_category_t	
PublicGroupManager	パブリックグループセット	b_m_public_group_set_b
	パブリックグループ	b_m_public_group_b
		b_m_public_group_t
		b_m_public_group_t_i
	パブリックグループ構成バージョン	b_m_public_group_version_b
	パブリックグループ内包	b_m_public_group_inclusion_b
	パブリックグループ所属	b_m_public_group_attach_b
b_m_public_group_attach_t		
パブリックグループ分類	b_m_public_group_category_b	
	b_m_public_group_category_t	
PrivateGroupManager	プライベートグループ	b_m_private_group_b
	プライベートグループ所属	b_m_private_group_attach_b
CategoryManager	分類区分	b_m_category_cls_b
		b_m_category_cls_i
	分類区分詳細	b_m_category_detail_b
	b_m_category_detail_i	
ItemManager	商品	b_m_item_b
		b_m_item_t
		b_m_item_t_i
	商品カタログ	b_m_item_catalogue_b
	商品属性	b_m_item_attribute_b
b_m_item_attribute_i		
ItemTemplateManager	商品テンプレート	b_m_item_template_b
	商品テンプレート明細	b_m_item_template_detail_b
ItemTemplateManager	商品テンプレート明細	b_m_item_template_detail_i
ItemCategoryManager	商品カテゴリ種別	b_m_item_category_type_b
	商品カテゴリ	b_m_item_category_b
		b_m_item_category_i
商品分類	b_m_item_attach_b	
DealManager	取扱	b_m_deal_b
		b_m_deal_t
	取扱詳細	b_m_deal_detail_b
b_m_deal_detail_t		
DealPermissionManager	組織毎取扱パーミッション	b_m_deal_department_b
	パブリックグループ毎取扱パーミッション	b_m_deal_public_group_b
	ロール毎取扱パーミッション	b_m_deal_role_b

マネージャは内部ではさまざまなマップ(マップの詳細についてはアプリケーション共通マスタ API 方針「3.2.2 マップ」を参照してください。)を使用しています。マップを直接扱ってもアプリケーション共通マスタにデータを登録できますが、整合性の保持などで大きな危険を伴いますので、データの変更はマネージャを通じて行うようにしてください。

3.2.1 マネージャの取得

アプリケーション共通マスタではさまざまなマネージャを用意していますが、マネージャの取得方法は同様です。以下に J2EE 開発モデルおよびスクリプト開発モデルにおけるマネージャの取得方法について説明します。

3.2.1.1 J2EE開発モデルの場合

J2EE開発モデルではマネージャは抽象クラスとして定義されています。マネージャを取得する場合、それぞれのクラスにクラスメソッドとして定義されているget~Managerというメソッドを呼び出すことで取得できます。「表 3-3 マネージャの取得 (J2EE開発モデル)」にマネージャとその取得メソッドの一覧を示します。

表 3-3 マネージャの取得(J2EE 開発モデル)

取り扱う情報	マネージャ	マネージャを取得するメソッド
ユーザ	UserManager	getUserManager
会社／組織	CompanyManager	getCompanyManager
パブリックグループ	PublicGroupManager	getPublicGroupManager
プライベートグループ	PrivateGroupManager	getPrivateGroupManager
分類	CategoryManager	getCategoryManager
商品	ItemManager	getItemManager
商品テンプレート	ItemTemplateManager	getItemTemplateManager
商品カテゴリ	ItemCategoryManager	getItemCategoryManager
取扱	DealManager	getDealManager
取扱パーミッション	DealPermissionManager	getDealPermissionManager

これらのメソッドは必ず引数にログインユーザコードとログイングループコードを必要とします。

「リスト 3-1 CompanyManagerの取得例 (J2EE開発モデル)」に会社に関連するマネージャ(CompanyManager)を取得するときの例を示します。

リスト 3-1 CompanyManager の取得例 (J2EE 開発モデル)

```
import jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyManager;
...
String user = ...; // ログインユーザコードの取得
String login = ...; // ログイングループコードの取得
CompanyManager manager = CompanyManager.getCompanyManager(user, login);
...
```

3.2.1.2 スクリプト開発モデルの場合

スクリプト開発モデルではマネージャは定義済みのオブジェクトとして定義されています。マネージャを取得する場合、それぞれのオブジェクトのコンストラクタを呼び出すことで取得できます。「表 3-4 マネージャの取得(スクリプト開発モデル)」にマネージャとその取得メソッドの一覧を示します。

表 3-4 マネージャの取得(スクリプト開発モデル)

取り扱う情報	マネージャ
ユーザ	UserManager
会社/組織	CompanyManager
パブリックグループ	PublicGroupManager
プライベートグループ	PrivateGroupManager
分類	CategoryManager
商品	ItemManager
商品テンプレート	ItemTemplateManager
商品カテゴリ	ItemCategoryManager
取扱	DealManager
取扱パーミッション	DealPermissionManager

これらのコンストラクタは必ず引数にログインユーザコードとログイングループコードを必要とします。

「リスト 3-2 CompanyManager の取得例(スクリプト開発モデル)」に会社に関連するマネージャ(CompanyManager)を取得するときの例を示します。

リスト 3-2 CompanyManager の取得例(スクリプト開発モデル)

```

...
var user = ...; // ログインユーザコードの取得
var login = ...; // ログイングループコードの取得
var manager = new CompanyManager(user, login);
...

```

3.2.2 マネージャの使用

マネージャの各メソッドを呼び出すことでアプリケーション共通マスタに対してデータの登録や削除などが行えるようになります。これらのメソッドはトランザクションが有効となっている状態で呼び出す必要があります。以下に J2EE 開発モデルおよびスクリプト開発モデルにおけるマネージャの使用方法について説明します。

3.2.2.1 J2EE開発モデルの場合

J2EE開発モデルのマネージャのメソッド(get~ を除きます)はユーザトランザクション内で実行されることが前提となっています。「リスト 3-3 CompanyManagerの使用例(J2EE開発モデル)」に会社に関連するマネージャ(CompanyManager)を使用して会社情報を新規に登録するときの例を示します。

リスト 3-3 CompanyManager の使用例(J2EE 開発モデル)

```
import javax.naming.InitialContext;
import javax.transaction.UserTransaction;
import jp.co.intra_mart.foundation.datastore.application.domain.company.company.Company;
import jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyManager;
...
// トランザクションの開始
InitialContext ctx = new InitialContext();
UserTransaction ut = (UserTransaction)ctx.lookup("java:comp/UserTransaction");
ut.begin();
try {
    ...
    CompanyManager manager = ...; // マネージャの取得
    Company company = ...; // 会社情報の生成
    manager.addCompany(company);
    ...
    // トランザクションの終了
    ut.commit();
} catch (Exception e) {
    // 例外処理
}
...
```

3.2.2.2 スクリプト開発モデルの場合

スクリプト開発モデルのマネージャのメソッド(get~ を除きます)はユーザトランザクション内で実行されることが前提となっています。「リスト 3-4 CompanyManagerの使用例(スクリプト開発モデル)」に会社に関連するマネージャ(CompanyManager)を使用して会社情報を新規に登録するときの例を示します。

リスト 3-4 CompanyManager の使用例(スクリプト開発モデル)

```
...
// トランザクションの開始
if (DatabaseManager.beginTransaction().isSuccess()) {
    ...
    var manager = ...; // マネージャの取得
    var company = ...; // 会社情報の生成
    manager.addCompany(company);
    ...
    // トランザクションの終了
    if (!DatabaseManager.commit().isSuccess()) {
        // コミットに失敗したときの処理
    }
}
...
```

3.2.3 マネージャによる検索

いくつかのマネージャにはデータを取得するためのメソッド(詳細については API リストを参照して下さい)が提供されています。このメソッドは大きく分けて以下の2つに分類されます。

- 詳細情報を取得するメソッド
- 検索条件に一致するデータを1件だけ取得します。取得されたデータには、すべての期間の情報およびすべての国際化情報が設定されます。
- 一覧を取得するメソッド
- 検索条件に一致するデータを配列として取得します。取得されたデータには、システム日付(または指定された日付)を含む期間の情報およびログインユーザのデフォルト言語(または指定された言語)に一致する国際化情報が設定されます。

以下に J2EE 開発モデルおよびスクリプト開発モデルにおけるマネージャによる検索について説明します。

3.2.3.1 J2EE開発モデルの場合

3.2.3.1.1 詳細情報を取得する場合

J2EE開発モデルのマネージャの詳細取得メソッドを使用すると、結果はDTO(Data Transfer Object)として返却されます。「リスト 3-5 CompanyManagerによる詳細検索の例 (J2EE開発モデル)」に会社に関連するマネージャ (CompanyManager) を使用して組織の詳細情報を取得するときの例を示します。

リスト 3-5 CompanyManager による詳細検索の例(J2EE 開発モデル)

```
import jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyManager;
import jp.co.intra_mart.foundation.datastore.application.domain.company.company.Department;
import jp.co.intra_mart.foundation.datastore.common.structure.Term;
...
CompanyManager manager = ...; // マネージャの取得
Department dept = manager.getDepartment("comp01", "dept10"); // 組織詳細情報の取得
if (dept == null) { // 検索結果が存在するか?
    return;
}
Term[] terms = dept.getTerms(); // 期間一覧を取得
for (int i = 0; i < terms.length; i++) {
    DepartmentTerminable tDept = dept.getDepartmentTerminable(terms[i]); // 期間化情報
    String telNo = tDept.getTelephoneNumber(); // 電話番号の取得
    ...
    Locale[] locales = tDept.getLocales(); // 言語一覧を取得
    for (int j = 0; j < locales.length; j++) {
        DepartmentTerminableInternational tiDept = tDept
            .getDepartmentTerminableInternational(locales[j]); // 期間国際化情報
        String name = tiDept.getName(); // 組織名を取得
        ...
    }
}
...
```

3.2.3.1.2 一覧情報を取得する場合

J2EE開発モデルのマネージャの一覧取得メソッドを使用すると、結果はDTOの配列として返却されます。「リスト 3-6 CompanyManagerによる一覧検索の例 (J2EE 開発モデル)」に会社に関連するマネージャ (CompanyManager) を使用して組織に所属するユーザの一覧を取得するときの例を示します。

リスト 3-6 CompanyManager による一覧検索の例 (J2EE 開発モデル)

```
import jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyManager;
import jp.co.intra_mart.foundation.datastore.application.domain.company.company.Department;
import jp.co.intra_mart.foundation.datastore.application.domain.user.User;
...
CompanyManager manager = ...; // マネージャの取得
User[] users = manager.getMember("comp01", "dept10"); // 所属ユーザの一覧取得
for (int i = 0; i < users.length; i++) {
    String code = users[i].getUserCode(); // ユーザコードの取得
    String telNo = users[i].getTelephoneNumber(); // 電話番号の取得
    String name = users[i].getName(); // ユーザ名の取得
    ...
}
...
```

3.2.3.2 スクリプト開発モデルの場合

3.2.3.2.1 詳細情報を取得する場合

スクリプト開発モデルのマネージャの詳細取得メソッドを使用すると、結果は処理結果オブジェクトに格納されて返却されます。「リスト 3-7 CompanyManagerによる詳細検索の例 (スクリプト開発モデル)」に会社に関連するマネージャ (CompanyManager) を使用して組織の詳細情報を取得するときの例を示します。

リスト 3-7 CompanyManager による詳細検索の例 (スクリプト開発モデル)

```
...
var manager = ...; // マネージャの取得
var result = manager.getDepartment("comp01", "dept10"); // 組織詳細情報を検索
if (result.error || result.data == null) { // 検索結果が存在するか?
    return;
}
var dept = result.data;
var deptCd = dept.departmentCode; // 組織コードの取得
for (var i = 0; i < dept.terms.length; i++) { // 期間化情報を取得
    var telNo = dept.terms[i].telephoneNumber; // 電話番号を取得
    ...
    for (var locale in dept.terms[i].locales) { // 期間国際化情報を取得
        var name = dept.terms[i].locales[locale].name; // 組織名を取得
        ...
    }
}
...
```

3.2.3.2.2 一覧情報を取得する場合

スクリプト開発モデルのマネージャの一覧取得メソッドを使用すると、結果は処理結果オブジェクトに配列として格納されて返却されます。「リスト 3-8 CompanyManagerによる一覧検索の例 (スクリプト開発モデル)」に会社に関連するマネージャ (CompanyManager) を使用して組織に所属するユーザの一覧を取得するときの例を示します。

リスト 3-8 CompanyManager による一覧検索の例(スクリプト開発モデル)

```

...
var manager = ...; // マネージャの取得
var result = manager.getMember("comp01", "dept10"); // 組織詳細情報を検索
if (result.error || result.data.length < 1) { // 検索結果が存在するか?
    return;
}
var users = result.data;
var locale = AccessSecurityManager.getSessionInfo().locale; // ログインユーザの言語を取得
for (var i = 0; i < users.length; i++) {
    var code = users[i].userCode; // ユーザコードの取得
    var telNo = users[i].terms[0].telephoneNumber; // 電話番号の取得
    var name = users[i].terms[0].locales[locale].name; // ユーザ名の取得
    ...
}
...

```

3.2.4 マネージャによる拡張情報の操作

ユーザ情報および組織情報については、対応するマネージャをとおして拡張情報の登録、更新、参照の操作を行うことができます。以下にJ2EE開発モデルおよびスクリプト開発モデルにおけるマネージャによる拡張情報の操作について説明します。³

3.2.4.1 J2EE開発モデルの場合

J2EE開発モデルでは、「表 3-5 拡張情報を操作できるメソッド(J2EE開発モデル)」に示すマネージャのメソッドにより拡張情報を操作することができます。

表 3-5 拡張情報を操作できるメソッド(J2EE開発モデル)

マネージャ	操作メソッド	拡張情報の操作内容
ユーザマネージャ	getUser()	ユーザ拡張情報を参照
	addUser(User)	ユーザ拡張情報を登録
	updateUser(User)	ユーザ拡張情報を更新
会社マネージャ	getDepartment()	組織拡張情報を参照
	addDepartment(Department)	組織拡張情報を登録
	updateDepartment(Department)	組織拡張情報を更新

「リスト 3-9 UserManagerによる拡張基本情報の参照の例(J2EE開発モデル)」にユーザマネージャによるユーザ拡張基本情報の参照、「リスト 3-10 UserManagerによる拡張基本情報の登録の例(J2EE開発モデル)」に登録の例を示します。(拡張期間化情報、拡張期間国際化情報についても同様に扱うことができます)

³拡張情報を扱うためには、ユーザエンティティまたは組織エンティティに対して、拡張テーブルおよび拡張モデルの定義、作成をする必要があります。詳細については、アプリケーション共通マスタ API 方針の「2.5 リレーショナルデータベースのマッピング」、「3.2 モデル」、「4 開発者の役割」および「付録 B DTD」をご覧ください。

リスト 3-9 UserManager による拡張基本情報の参照の例 (J2EE 開発モデル)

```
import java.util.Map;
import jp.co.intra_mart.foundation.datastore.application.domain.user.UserManager;
import jp.co.intra_mart.foundation.datastore.application.domain.user.User;
...
UserManager manager = ...; // マネージャの取得
User user = manager.getUser("aoyagi"); // ユーザ情報の取得

// "extendedUser"という拡張名の拡張基本情報を取得
Map extendedUser = user.getExtendedUser("extendedUser");
// 拡張情報の詳細情報を取得
System.out.println(extendedUser.get("birthday")); // 誕生日
System.out.println(extendedUser.get("bloodType")); // 血液型
...
```

リスト 3-10 UserManager による拡張基本情報の登録の例 (J2EE 開発モデル)

```
import java.util.*;
import jp.co.intra_mart.foundation.datastore.application.domain.user.UserManager;
import jp.co.intra_mart.foundation.datastore.application.domain.user.User;
...
UserManager manager = ...; // マネージャの取得

// ユーザ情報の新規作成
User user = new User();
user.setUserCode("aoyagi");
// ユーザ拡張基本情報の新規作成
Map extendedUser = new HashMap();
// 誕生日
extendedUser.put("birthday", new GregorianCalendar(2000,0,1).getTime());
// 血液型
extendedUser.put("bloodType", "B");
// ユーザ情報に"extendedUser"という拡張名で拡張情報を設定
user.putExtendedUser("extendedUser" extendedUser);
...
manager.addUser(user); // ユーザ情報の登録
...
```

3.2.4.2 スクリプト開発モデルの場合

J2EE開発モデルでは、「表 3-6 拡張情報を操作できるメソッド(スクリプト開発モデル)」に示すマネージャのメソッドにより拡張情報を操作することができます。

表 3-6 拡張情報を操作できるメソッド(スクリプト開発モデル)

マネージャ	操作メソッド	拡張情報の操作内容
ユーザマネージャ	getUser()	ユーザ拡張情報を参照
	addUser(User)	ユーザ拡張情報を登録
	updateUser(User)	ユーザ拡張情報を更新
会社マネージャ	getDepartment()	組織拡張情報を参照
	addDepartment(Department)	組織拡張情報を登録
	updateDepartment(Department)	組織拡張情報を更新

「リスト 3-11 UserManagerによる拡張基本情報の参照の例(スクリプト開発モデル)」にユーザマネージャによるユーザ拡張基本情報の参照、「リスト 3-12 UserManagerによる拡張基本情報の登録の例(スクリプト開発モデル)」に登録の例を示します。(拡張期間化情報、拡張期間国際化情報についても同様に扱うことができます)

リスト 3-11 UserManager による拡張基本情報の参照の例(スクリプト開発モデル)

```
...
var manager = ...; // マネージャの取得
var user = manager.getUser("aoyagi"); // ユーザ情報の取得
...
// "extendedUser"という拡張名の拡張基本情報を取得
var extendedUser = user.extended.get("extendedUser");
// 拡張情報の詳細情報を取得
Debug.print(extendedUser.birthday); // 誕生日
Debug.print(extendedUser.bloodType); // 血液型
...
```

リスト 3-12 UserManager による拡張基本情報の登録の例(スクリプト開発モデル)

```
...
var manager = ...; // マネージャの取得

// ユーザ情報の新規作成
var user = new Object();
user.userCode = "aoyagi";
// 拡張マップを生成
user.extended = new ExtendedMap();
// ユーザ拡張基本情報の新規作成
var extendedUser = new Object();
// 誕生日
extendedUser.birthday = Module.date.get(2000,1,1);
// 血液型
extendedUser.bloodType = 'B';
// ユーザ情報に"extendedUser"という拡張名で拡張情報を設定
user.extended.put("extendedUser" extendedUser);
...
manager.addUser(user); // ユーザ情報の登録
...
```

3.3 リスナ

マネージャはさまざまな更新処理(登録や削除を含みます)を行います。このとき、本来の処理の後に独自の処理を追加することができます。

3.3.1 リスナの動作

「図 3-2 リスナの動作概要」にリスナの動作概要を示します。

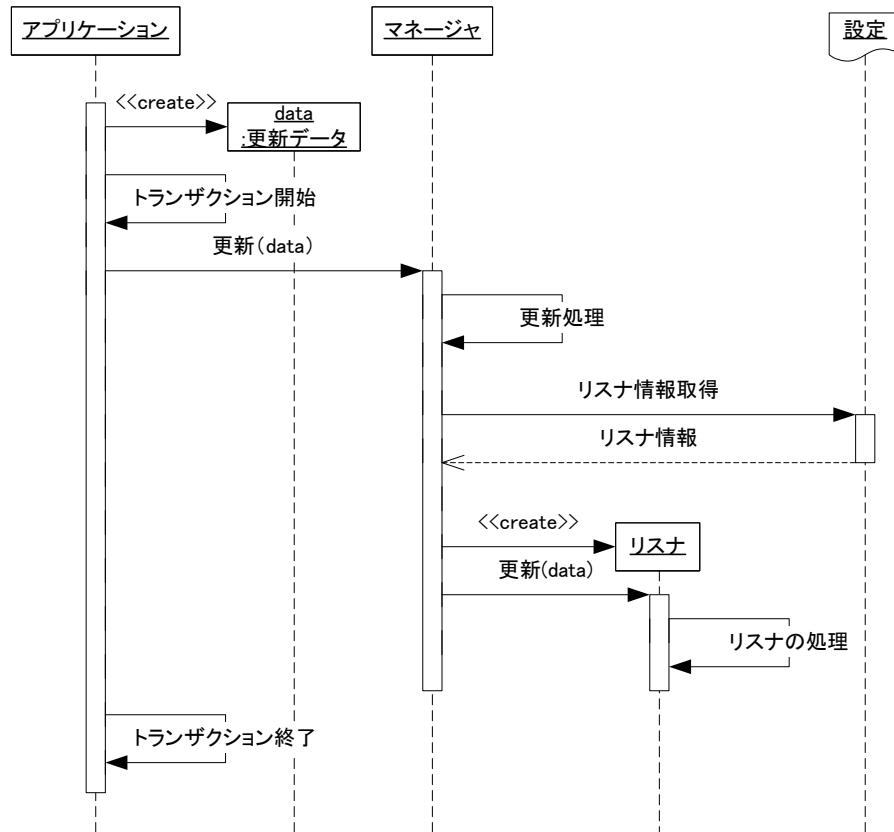


図 3-2 リスナの動作概要

- ① アプリケーションは更新(登録や削除を含みます)用のオブジェクトを生成します。
- ② アプリケーションはトランザクションを開始します。
- ③ アプリケーションはマネージャに対して更新処理を依頼します。
- ④ マネージャは本来の更新処理を行います。
- ⑤ マネージャは設定をもとにリスナ情報を取得します。
- ⑥ マネージャは更新用のオブジェクトをもとにリスナを起動します。
- ⑦ リスナでは更新用のオブジェクトを参照し、独自の処理を行います。
- ⑧ マネージャおよびすべてのリスナの処理が正常終了した場合、アプリケーションはトランザクションをコミットします。一部でも例外を返した場合、アプリケーションはトランザクションをロールバックする必要があります。

3.3.2 リスナの種類

リスナは以下の情報操作に対して設定できます。

- ユーザ
- 会社／組織
- パブリックグループ
- プライベートグループ

3.3.2.1 ユーザに関連するリスナ

ユーザの登録、更新および削除を行うとき、`UserManager` は設定ファイルに定義されたリスナを呼び出します。この場合のリスナは以下の条件をすべて満たす必要があります。

- `jp.co.intra_mart.foundation.datastore.application.domain.user.UserChangedListener` インタフェースを実装している。
- 引数なしの `public` なコンストラクタ(デフォルトコンストラクタ)が定義されている。
- `java.util.Map` を引数にもつ `init` メソッドが実装されている。
- `java.lang.String` を引数にもつ `setUserId` メソッドが実装されている。

「表 3-7 ユーザのリスナ」に実行されるメソッドを実行します。

表 3-7 ユーザのリスナ

タイミング	UserManager のメソッド	UserListener で実行されるメソッド
ユーザ登録	addUser	userAdded
ユーザ更新	updateUser	userUpdated
ユーザ削除	deleteUser	userDeleted

「表 3-7 ユーザのリスナ」で示したUserListenerのメソッドはすべて戻り値をとりません。

「表 3-7 ユーザのリスナ」に示したメソッドの中では引数で受け取った内容を変更しないでください。

3.3.2.2 会社／組織に関連するリスナ

会社や組織に関連する情報に対してなんらかの変更を行うとき、CompanyManager は設定で定義されたリスナを呼び出します。この場合のリスナは以下の条件をすべて満たす必要があります。

- jp.co.intra_mart.foundation.datastore.application.domain.company.CompanyChangeListener インタフェースを実装している。
- 引数なしの public なコンストラクタ(デフォルトコンストラクタ)が定義されている。
- java.util.Map を引数にもつ init メソッドが実装されている。
- java.lang.String を引数にもつ setUserId メソッドが実装されている。

「表 3-8 会社／組織のリスナ」に実行されるメソッドを実行します。

表 3-8 会社／組織のリスナ

タイミング	CompanyManager のメソッド	CompanyListener で実行されるメソッド
会社登録	addCompany	companyAdded
会社削除	deleteCompany	companyDeleted
組織登録	addDepartment	departmentAdded
組織更新	updateDepartment	departmentUpdated
組織削除	deleteDepartment	departmentDeleted
組織構成登録	joinDepartment	departmentJoined
組織構成移動	moveDepartment	departmentMoved
組織構成削除	leaveDepartment	departmentLeft
分類区分設定	setCategoryDetail	categoryDetailSet
分類区分削除	removeCategoryDetail	rategoryDetailRemoved
人員割当	setMember	memberSet
人員削除	deleteMember	memberDeleted
主所属設定	setMain	mainSet
主所属削除	removeMain	mainRemoved
バージョン登録	addVersion	versionAdded
バージョン更新	updateVersion	versionUpdated
バージョン削除	deleteVersion	versionDeleted
バージョンコピー	copyVersion	versionCopied
役職登録	AddPost	postAdded
役職更新	updatePost	postUpdated
役職削除	deletePost	postDeleted

「表 3-8 会社／組織のリスナ」に示したメソッドの中では引数で受け取った内容を変更しないでください。

3.3.2.3 パブリックグループに関連するリスナ

パブリックグループに関連する情報に対してなんらかの変更を行うとき、`PublicGroupManager` は設定で定義されたリスナを呼び出します。この場合のリスナは以下の条件をすべて満たす必要があります。

- `jp.co.intra_mart.foundation.datastore.application.domain.public_group.PublicGroupChangeListener` インタフェースを実装している。
- 引数なしの `public` なコンストラクタ (デフォルトコンストラクタ) が定義されている。
- `java.util.Map` を引数にもつ `init` メソッドが実装されている。
- `java.lang.String` を引数にもつ `setUserId` メソッドが実装されている。

「表 3-9 パブリックグループのリスナ」に実行されるメソッドを実行します。

表 3-9 パブリックグループのリスナ

タイミング	<code>PublicGroupManager</code> のメソッド	<code>PublicGroupListener</code> で実行されるメソッド
グループセット登録	<code>addGroupSet</code>	<code>groupSetAdded</code>
グループセット削除	<code>deleteGroupSet</code>	<code>groupSetDeleted</code>
グループ登録	<code>addPublicGroup</code>	<code>publicGroupAdded</code>
グループ更新	<code>updatePublicGroup</code>	<code>publicGroupUpdated</code>
グループ削除	<code>deletePublicGroup</code>	<code>publicGroupDeleted</code>
グループ構成登録	<code>joinPublicGroup</code>	<code>publicGroupJoined</code>
グループ構成移動	<code>movePublicGroup</code>	<code>publicGroupMoved</code>
グループ構成削除	<code>leavePublicGroup</code>	<code>publicGroupLeft</code>
分類区分設定	<code>setCategoryDetail</code>	<code>categoryDetailSet</code>
分類区分削除	<code>removeCategoryDetail</code>	<code>rategoryDetailRemoved</code>
人員割当	<code>setMember</code>	<code>memberSet</code>
人員削除	<code>deleteMember</code>	<code>memberDeleted</code>
バージョン登録	<code>addVersion</code>	<code>versionAdded</code>
バージョン更新	<code>updateVersion</code>	<code>versionUpdated</code>
バージョン削除	<code>deleteVersion</code>	<code>versionDeleted</code>
バージョンコピー	<code>copyVersion</code>	<code>versionCopied</code>

「表 3-9 パブリックグループのリスナ」に示したメソッドの中では引数で受け取った内容を変更しないでください。

3.3.2.4 プライベートグループに関連するリスナ

プライベートグループに関連する情報に対してなんらかの変更を行うとき、`PrivateGroupManager` は設定で定義されたリスナを呼び出します。この場合のリスナは以下の条件をすべて満たす必要があります。

- `jp.co.intra_mart.foundation.datastore.application.domain.private_group.PrivateGroupChangeListener` インタフェースを実装している。
- 引数なしの `public` なコンストラクタ(デフォルトコンストラクタ)が定義されている。
- `java.util.Map` を引数にもつ `init` メソッドが実装されている。
- `java.lang.String` を引数にもつ `setUserId` メソッドが実装されている。

「表 3-10 プライベートグループのリスナ」に実行されるメソッドを実行します。

表 3-10 プライベートグループのリスナ

タイミング	<code>PrivateGroupManager</code> のメソッド	<code>PrivateGroupListener</code> で実行されるメソッド
グループ登録	<code>addPrivateGroup</code>	<code>privateGroupAdded</code>
グループ更新	<code>updatePrivateGroup</code>	<code>privateGroupUpdated</code>
グループ削除	<code>deletePrivateGroup</code>	<code>privateGroupDeleted</code>
人員割当	<code>setMember</code>	<code>memberSet</code>
人員削除	<code>deleteMember</code>	<code>memberDeleted</code>

「表 3-10 プライベートグループのリスナ」に示したメソッドの中では引数で受け取った内容を変更しないでください。

3.3.3 リスナの設定

作成したリスナを設定します。

リスナを設定するには、conf/system-install.xml内の要素system-install/listener内に「表 3-11 リスナを設定するための要素名」で示した要素を記述します。

表 3-11 リスナを設定するための要素名

リスナ名	設定ファイル内の要素名
UserChangedListener	profile-listener/listener-class
CompanyChangedListener	department-listener/listener-class
PublicGroupChangedListener	public-group-listener/listener-class
PrivateGroupChangedListener	private-group-listener/listener-class

「リスト 3-13 各リスナの設定例」に作成したリスナのクラス名を記述する例を示します。

リスト 3-13 各リスナの設定例

```
<system-install>
<listener>
...
<department-listener>
  <listener-class>sample.listener.company.SampleCompanyChangedListener</listener-class>
</department-listener>

<public-group-listener>
  <listener-class>sample.listener.public_group.SamplePublicGroupChangedListener</listener-class>
</public-group-listener>

<private-group-listener>
  <listener-class>sample.listener.private_group.SamplePrivateGroupChangedListener</listener-class>
</private-group-listener>

<profile-listener>
  <listener-class>sample.listener.user.SampleUserChangedListener</listener-class>
</profile-listener>
...
</listener>
</system-install>
```

4 検索例

アプリケーション共通マスタに関連する情報を検索する場合、直接 SQL を実行する必要がある場合があります。そのとき、検索目的に応じて対象となるマスタを使い分ける必要があります。

4.1 ある組織の詳細情報を取得する場合

2005/10/01時点の会社A-部門B（会社コード= 'comp_A'、組織コード= 'dept_B'）の詳細情報を日本語（ロケール= 'ja'）で取得する場合

```
select ~ from b_m_department_b dpt_b, b_m_department_t dpt_t
left outer join b_m_department_t_i dpt_ti
on (dpt_t.company_cd = dpt_ti.company_cd and dpt_t.department_cd = dpt_ti.department_cd
and dpt_t.term_cd = dpt_ti.term_cd and dpt_ti.locale_id = 'ja')
where dpt_b.company_cd = 'comp_A' and dpt_b.department_cd = 'dept_B'
and dpt_b.company_cd = dpt_t.company_cd and dpt_b.department_cd = dpt_t.department_cd
and dpt_t.start_date <= '2004/10/01|00:00:00' and dpt_t.end_date > '2004/10/01|00:00:00'
```

組織テーブル (b_m_department_b)

会社コード (company_cd)	組織コード (department_cd)	備考 (notes)	...
comp_A	comp_A	会社Aの備考	...
comp_A	dept_A	部門Aの備考	...
comp_A	dept_B	部門Bの備考	...
...

組織テーブル (期間化) (b_m_department_t)

会社コード (company_cd)	組織コード (department_cd)	期間コード (term_cd)	開始日時 (start_date)	終了日時 (end_date)	電話番号 (telephone_number)	...
...
comp_A	dept_A	term_02	2008/04/01	-	03-XXXX-1012	...
comp_A	dept_B	term_00	-	2003/04/01	03-XXXX-1020	...
comp_A	dept_B	term_01	2003/04/01	2006/04/01	03-XXXX-1021	...
comp_A	dept_B	term_02	2006/04/01	-	03-XXXX-1022	...
...

組織テーブル (期間国際化) (b_m_department_t_i)

会社コード (company_cd)	組織コード (department_cd)	期間コード (term_cd)	ロケール (locale_id)	組織名 (department_name)	...
...
comp_A	dept_A	term_02	en	Branch A	...
comp_A	dept_B	term_00	ja	部門B	...
comp_A	dept_B	term_00	en	Department B	...
comp_A	dept_B	term_01	ja	B部門	...
comp_A	dept_B	term_01	en	Section B	...
comp_A	dept_B	term_02	ja	B部	...
comp_A	dept_B	term_02	en	Branch B	...
...

結果

会社コード (company_cd)	組織コード (department_cd)	備考 (notes)	電話番号 (telephone_number)	組織名 (department_name)	...
comp_A	dept_A	部門Aの備考	03-XXXX-1021	B部門	...

図 4-1 ある組織の詳細情報を取得する

4.2 ある会社以下の全ての組織を取得する

A社(会社コード='comp_a')のバージョンver001(バージョンコード='ver001')
時点の詳細情報を取得する場合

```
select ~ from b_m_department_inclusion_b_inc_b
  where company_cd = 'comp_a'
        and version_cd = 'ver001'
        and parent_department_cd = 'comp_a'
```

組織内包マスタ (b_m_department_inclusion_b)

会社コード (company_cd)	バージョンコード (version_cd)	親組織コード (parent_department_cd)	組織コード (department_cd)	階層レベル (depth)	...
comp_a	ver001	comp_a	comp_a	0	...
comp_a	ver001	dept_b	dept_b	0	...
comp_a	ver001	comp_a	dept_b	1	...
comp_a	ver001	dept_b1	dept_b1	0	...
comp_a	ver001	dept_b	dept_b1	1	...
comp_a	ver001	comp_a	dept_b1	2	...
comp_a	ver001	dept_c	dept_c	0	...
comp_a	ver001	comp_a	dept_c	1	...

結果

会社コード (company_cd)	バージョンコード (version_cd)	親組織コード (parent_department_cd)	組織コード (department_cd)	階層レベル (depth)	...
comp_a	ver001	comp_a	comp_a	0	...
comp_a	ver001	comp_a	dept_b	1	...
comp_a	ver001	comp_a	dept_b1	2	...
comp_a	ver001	comp_a	dept_c	1	...

図 4-2 ある会社以下の全ての組織を取得する

4.3 ある組織に所属するユーザを取得する

2005/10/01時点における会社A-部門B1（会社コード=`comp_a`、組織コード=`dept_b1`）に所属するユーザを取得する場合

```
select ~ from b_m_department_attach_t atc_t
  where atc_t.company_cd = 'comp_a'
        and atc_t.department_cd = 'dept_b1'
        and atc_t.start_date <= '2005/10/01|00:00:00'
        and atc_t.end_date > '2005/10/01|00:00:00'
```

所属マスタ（期間化）（b_m_department_attach_t）

ユーザコード* (user_cd)	会社コード (company_cd)	組織コード (department_cd)	期間コード (term_cd)	開始日時 (start_date)	終了日時 (end_date)	...
user_a	comp_a	comp_a	term_00	—	2005/01/01	...
user_a	comp_a	dept_b	term_00	2003/01/01	2006/01/01	...
user_a	comp_a	dept_b1	term_00	2004/01/01	—	...
user_b	comp_a	comp_a	term_00	2005/01/01	—	...
user_b	comp_a	dept_b	term_00	—	2005/01/01	...
user_b	comp_a	dept_b	term_01	2006/01/01	—	...
user_b	comp_a	dept_b1	term_00	2003/01/01	2006/01/01	...
user_b	comp_a	dept_b1	term_01	2007/01/01	—	...
user_b	comp_a	dept_c	term_00	2005/01/01	—	...
user_c	comp_a	comp_a	term_00	—	2005/01/01	...
user_c	comp_a	dept_b	term_00	2005/01/01	2006/01/01	...
user_c	comp_a	dept_b1	term_00	2006/01/01	—	...

結果

ユーザコード* (user_cd)	会社コード (company_cd)	組織コード (department_cd)	...
user_a	comp_a	dept_b1	...
user_b	comp_a	dept_b1	...

図 4-3 ある組織に所属するユーザを取得する

**intra-mart WebPlatform/AppFramework Ver. 7.2
アプリケーション共通マスタ仕様書**

2010/05/31 第2版

**Copyright 2000-2010 株式会社NTTデータ イントラマート
All rights Reserved.**

TEL: 03-5549-2821

FAX: 03-5549-2816

E-MAIL: info@intra-mart.jp

URL: <http://www.intra-mart.jp/>