

# ERwin チュートリアル

日揮情報ソフトウェア株式会社

## ERwin チュートリアル

本書の著作権は、日揮情報ソフトウェア株式会社にあります。

本書に記載されている内容は予告なく変更することがあります。

本書の一部または全部を、日揮情報ソフトウェア株式会社との書面による同意なしに、複写、複製、転載、他言語への翻訳、書き換え、または転送をすることは法律で禁じられています。



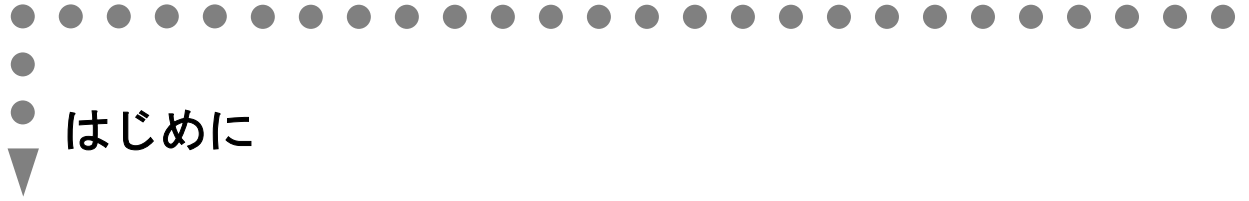
Copyright © 2004 J-SYS Software, Co., LTD.  
All rights reserved.

AllFusion ERwin Data Modeler は米国 Computer Associates International, Inc.の登録商標です。

Acrobat® Reader copyright ©1987-2001 Adobe Systems Incorporated. All right reserved.

Adobe および Acrobat はアドビシステムズ社の商標です。

その他の社名または製品名は各社の商標もしくは登録商標です。



## はじめに

このチュートリアルでは、ERwin のデータベース設計機能や直観的な基本的な操作方法を学習できます。ERwin の基本機能の習得にお役立てください。

---

### 対象とする読者

このチュートリアルは、以下の方を対象としています。


- ERwin を初めて使用する。
- ERwin を使用したデータベース設計に携わっている。
- ERwin 体験版を使用して ERwin を評価している。

### チュートリアルの目標

- ERwin の基本的なデータモデリング機能とオペレーションを理解する。
- ERwin によるスキーマ生成/完全比較機能を理解する。
- ERwin を使用したドキュメント作成方法を理解する。

### ERwin 4.1.4 体験版の動作環境

- ERwin をインストールする前に、使用するコンピュータが必要なシステム構成を満たしているかどうかを確認してください。
- IBM-PC およびその互換機（CPU は Pentium クラス以上を推奨）
- 85MB のハードディスク空き容量
- 256MB 以上のメモリ（大型のモデルを扱う場合は、512MB）
- Microsoft Windows（Windows Server 2003、XP、2000、ME）


## チュートリアル構成

このチュートリアルでは、ERwin のインストールと起動、ER 図の作成方法、DDL の生成、テーブル定義書の作成方法、データベースのメンテナンスの方法などを習得できます。

---

Lesson 1	インストールと起動
Lesson 2	ERwin の画面説明
Lesson 3	エンティティと属性の作成
Lesson 4	エンティティと属性の定義入力
Lesson 5	ドメインの作成
Lesson 6	リレーションシップの作成
Lesson 7	リレーションシップの定義入力
Lesson 8	正規化
Lesson 9	物理設計（物理名の設定）
Lesson 10	物理設計（インデックスの設定）
Lesson 11	物理設計（バリデーション ルールの設定）
Lesson 12	物理設計（デフォルト設定）
Lesson 13	物理設計（ドメイン階層構造化）
Lesson 14	参照整合性の設定
Lesson 15	ERwin マクロを利用したスキーマ生成
Lesson 16	データベースとの接続
Lesson 17	スキーマ生成
Lesson 18	サブジェクト エリア/ストアド ディスプレイの作成
Lesson 19	レポート作成
Lesson 20	完全比較
Lesson 21	リバース エンジニアリング

## インストールと起動

ここでは、ERwin のインストールと起動について学習します。

インストールは、[キャンセル] をクリックすることでいつでも中止できます。中止を確認するメッセージが表示されたとき、そのまま中止するには [はい] を、インストールを再開するには [いいえ] をクリックしてください。中止した後に ERwin を使用するには、再度インストールする必要があります。

### ERwin をインストールするには

1. ダウンロードした実行ファイル（ERwin4.1.4 の場合は、AllFusionEDM\_414\_JPN.exe）をダブルクリックします。  
(CD-ROM からの場合：Windows Server 2003、XP、2000、または ME が動作している状態で、CD-ROM ドライブに CD-ROM を挿入します。自動実行が無効になっている場合は、CD-ROM ドライブをクリックし、次に Setup.exe をダブルクリックします。)
2. InstallShield ウィザードが起動します。
3. [次へ] をクリックします。
4. ライセンス契約に関する確認画面が表示されます。ライセンス契約に記載されている条件に同意する場合は、[同意する] をクリックします。同意しない場合は [キャンセル] をクリックし、インストール プロセスを終了します。
5. インストール プログラムの指示に従ってインストールを継続してください。
6. インストールが完了したことを知らせる画面が表示されたら、[完了] をクリックします。



既に ERwin 4.x がインストールされている場合は、コントロール パネルの [アプリケーションの追加と削除] を使用して、既存の ERwin 4.x を必ずアンインストールしてください。また、アンインストール後に Windows システム ディレクトリにある INI ファイル (ERWIN40.INI) を削除してください。

### ERwin を起動するには

[スタート] ボタンをクリックし、[プログラム] をポイントします。次に、[Computer Associates] をポイントし、[AllFusion] をポイントします。[ERwin Data Modeler] をポイントし、[ERwin] をクリックすると、ERwin が起動します。

☆ **MEMO** ☆

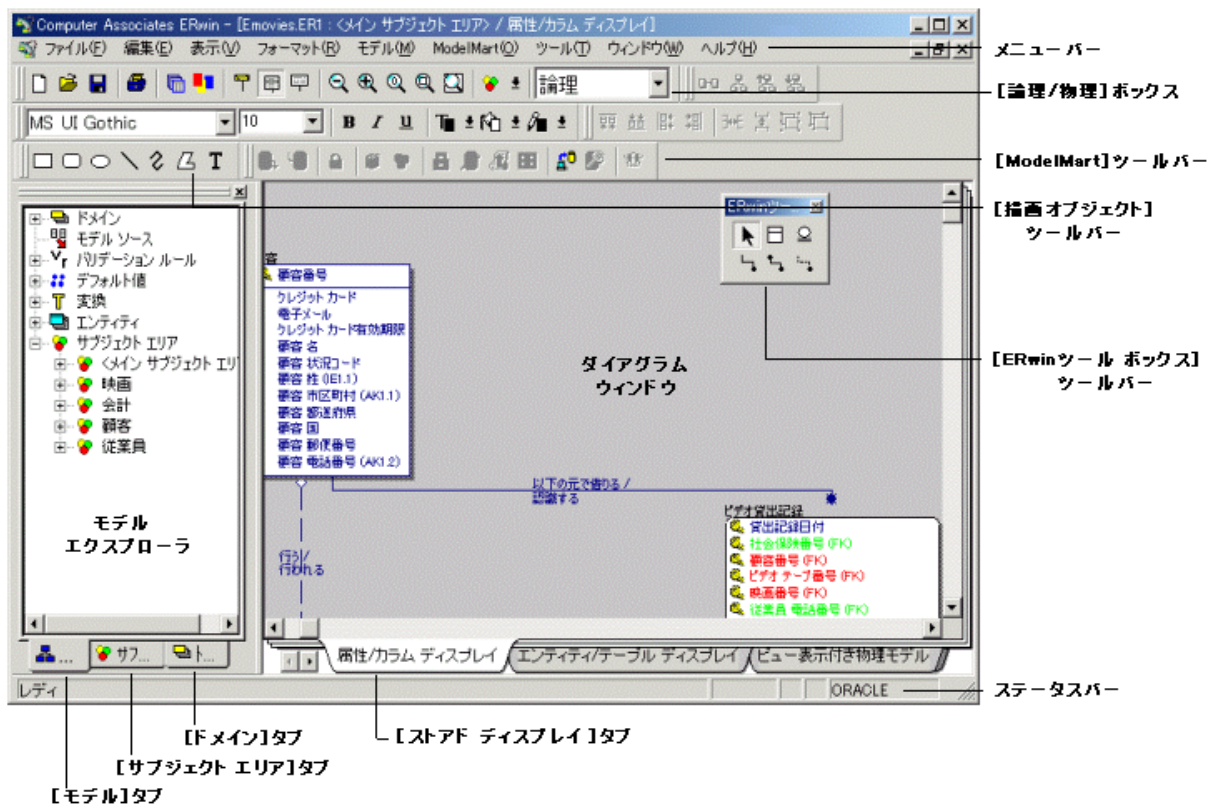
# Lesson

# 2

実習時間：約 10 分

## ERwin の画面説明

ここでは、ERwin 画面で表示する各種ツールやタブについて学習します。



### メニューバー


それぞれの機能を実行するためのメニューです。

### [論理/物理] ボックス

論理モデルから物理モデルに、または物理モデルから論理モデルに切り替えるためのリスト ボックスです。

### 【ModelMart】 ツールバー

ERwin の関連製品である Model Manager の各機能を実行するためのツールバーです。 [表示] メニューの [ツールバー] をポイントし、 [ModelMart] をクリックして表示/非表示を設定できます。

また、 をクリックして [ModelMart コネクション マネージャ] ダイアログ ボックスを開き、Model Manager に接続すると、 [ModelMart] ツールバーのボタンがアクティブになります。

### 【描画オブジェクト】 ツールバー

描画オブジェクトをダイアグラム ウィンドウに追加したり、テキストを描画オブジェクトに追加したりできます。 [表示] メニューの [ツールバー] をポイントし、 [描画] で表示/非表示を設定できます。

### 【ERwin ツールボックス】 ツールバー

データモデルを設計するために使用する選択ツール(ポインタ)とオブジェクト ツールです。 IDEF1X と IE 表記法では外観が少し異なります。

[表示] メニューの [ツールバー] をポイントし、 [ツールボックス] で表示/非表示を設定できます。

### モデル エクスプローラ

モデル内のオブジェクトをツリー構造で階層表示し、各オブジェクトを作成、編集、および削除できます。ダイアグラム上のオブジェクトにアクセスしないでオブジェクトを設計したり、モデル エクスプローラでオブジェクト選択して、ダイアグラム上の対象に移動したりできるため、大規模モデルの設計に有効です。モデル エクスプローラには、 [モデル] タブ、 [サブジェクト エリア] タブ、および [ドメイン] タブが用意されています。

### 【ストアド ディスプレイ】 タブ

現在のサブジェクト エリアに対するストアド ディスプレイです。タブをクリックして切り替えます。

### ステータスバー

現在選択されている機能やメニュー オプションに関する情報を表示します。通常は、「レディ」と表示されています。

[表示] メニューの [ステータスバー] で表示/非表示を設定できます。



## エンティティと属性の作成

ここでは、エンティティと属性の作成方法とデータ型と NULL オプションの設定方法について学習します。エンティティや属性を作成するには、ダイアログ ボックスから入力する方法と、ダイアグラム上で直接入力する方法があります。

### 説明



エンティティの作成は、ER 図を使用したデータベース設計での最初の作業です。エンティティとは、最終的には物理データベースでのテーブルとして定義されるオブジェクトの論理名で、ERwin では独立エンティティと依存エンティティを作成できます。

ダイアグラムを属性レベルで表示すると、エンティティは仕切り線で上下に分割されます。仕切り線の上部にはエンティティの主キー属性を入力し、下部には非キー属性を入力します。

属性とは、エンティティに関連するプロパティや特性を表すものです。最終的には物理データベースのカラムとして定義されるオブジェクトの論理名です。

#### 独立エンティティとは？

E/1



エンティティのインスタンス識別が、他のエンティティに依存することなく、自らの属性だけで可能となるエンティティ。角が四角い箱で表現されます。

#### 依存エンティティとは？

E/2



エンティティの存在とインスタンス識別を、他のエンティティに依存しなければならないエンティティ。角が丸い箱で表現されます。

#### 主キー属性とは？

E/3



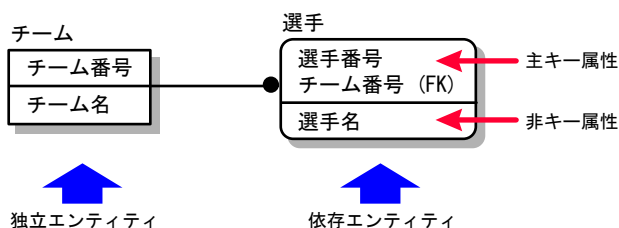
1つ以上の属性の組み合わせで、エンティティを一意に識別できる属性。

#### 非キー属性とは？




E/4



主キー属性以外の属性。



## エンティティの作成

1. [ファイル] メニューの [新規作成] をクリックするか、[ERwin 標準ツールバー] の [モデル作成]  をクリックし、[モデルの作成—テンプレート選択] ダイアログ ボックスを開きます。
2. [新規モデルのタイプ] の [論理/物理] をクリックします。
3. チュートリアルで使用するデータベースを [対象データベース] に指定します。ここでは、[データベース] ボックスの一覧の [ORACLE] をクリックし、[バージョン] ボックスの一覧の [9.x] をクリックします。次に、[OK] をクリックして [モデルの作成—テンプレート選択] ダイアログ ボックスを閉じます。
4. [ERwin ツールボックス] ツールバーの [エンティティ]  をクリックします。次に、マウス ポインタをダイアグラム上に移動し、カーソルが  になることを確認してください。
5. ダイアグラム上の任意の場所をクリックします。クリックした場所に [E/1] というエンティティが作成されます。
6. 同様の操作で、ダイアグラム上に [E/2] というエンティティを作成します。

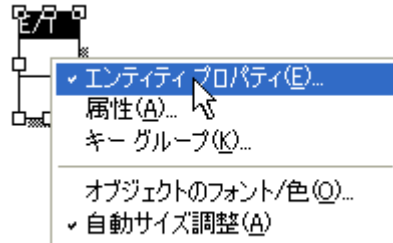


[ERwin ツールボックス] ツールバーのツールをクリックしてマウスをダイアグラム上に移動すると、カーソルの形状が選択したツールに合わせて変化します。これにより、現在どのツールを選択しているかを判断できます。

## エンティティと属性に名前を付ける

### －エンティティ エディタと属性 エディタを使用する－

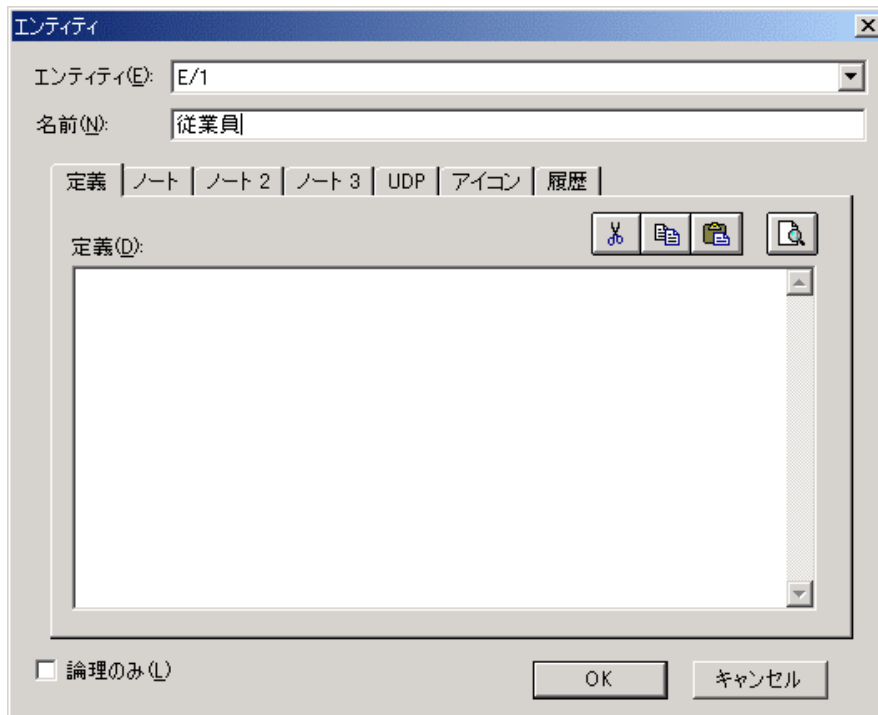
1. ダイアグラム上に作成した [E/1] エンティティを右クリックし、ショートカット メニューの [エンティティ プロパティ] をクリックして [エンティティ] ダイアログ ボックスを開きます。



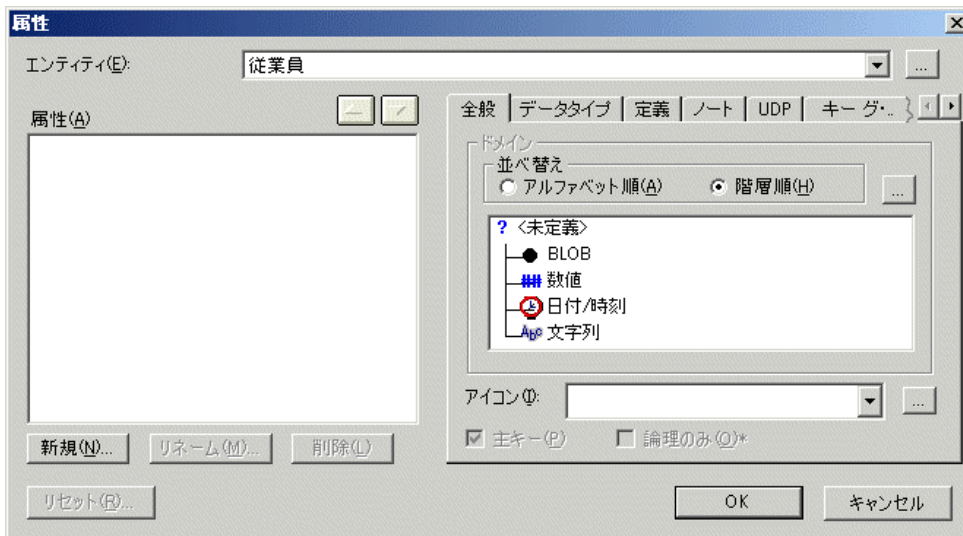
ダイアグラム オブジェクト（エンティティ、リレーションシップ、サブタイプ クラスタ、テキスト ブロックなど）を右クリックすると、各オブジェクトに対応したショートカットメニューが表示されます。メニュー項目をクリックすると、オブジェクトに対応するダイアログ ボックスを開くことができます。

また、最後に選択したダイアログ ボックスには、メニュー項目の左にチェックマーク (✓) が付きます。ダイアグラム ウィンドウでオブジェクトをダブルクリックすると、チェックマークが付いているオブジェクトのダイアログ ボックスが開きます。

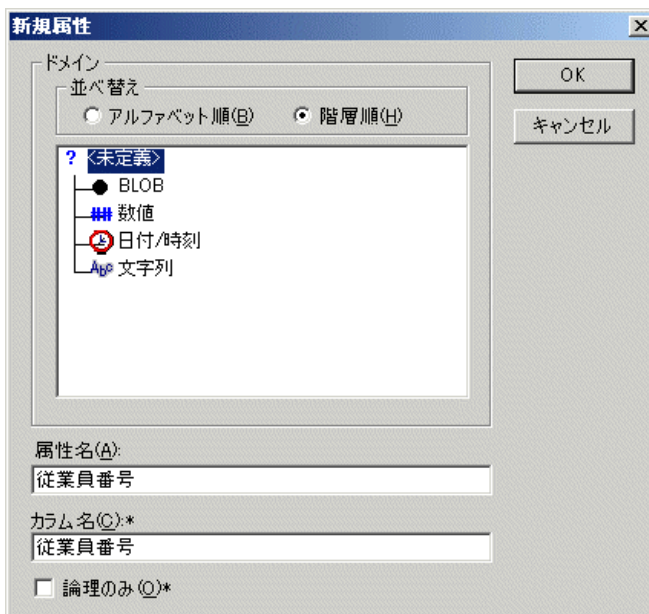
2. [名前] ボックスに「従業員」と入力し、[OK] をクリックして [エンティティ] ダイアログ ボックスを閉じます。



3. [従業員] エンティティを右クリックし、ショートカットメニューの[属性]をクリックして[属性]ダイアログボックスを開きます。



4. [新規]をクリックして[新規属性]ダイアログボックスを開き、[属性名]ボックスに「従業員番号」と入力します。次に、[OK]をクリックして[新規属性]ダイアログボックスを閉じます。



5. [属性]ダイアログボックスの[全般]タブの[主キー]チェックボックスをオンにして、この属性を主キーに設定します。
6. 同様の操作で「従業員名」、「従業員住所」、および「従業員電話番号」という名前の属性を作成します（これらの属性は非キー属性にします）。
7. [OK]をクリックして[属性]ダイアログボックスを閉じます。

## エンティティと属性に名前を付ける —オン・ダイアグラム編集を使用する—

1. ダイアグラム上に作成した [E/2] エンティティをクリックします。
2. エンティティ名の入力領域がアクティブになるまで<Tab>キーを押し、エンティティ名に「注文」と入力します。
3. 次に、主キー領域がアクティブになるまで<Tab>キーを押し、属性を入力するための [属性名] という入力領域が表示されたら、「注文番号」と入力します。
4. 続けて<Tab>キーを押すと、非キー領域がアクティブになります。[属性名] という入力領域が表示されたら、「注文日」と入力します。
5. <Enter>キーを押すと [注文日] の下に入力領域が移動するので、「注文ステータス」と入力します。  
この手順で、[注文] エンティティに以下の属性を入力します。

### 従業員

従業員番号
従業員名
従業員住所
従業員電話番号

### 注文

注文番号
注文日
注文ステータス
顧客番号
顧客名
属性名

### 非キー属性に入力する属性

注文日、注文ステータス、顧客番号、顧客名、顧客住所、顧客電話番号、製品番号、製品名、製品価格、注文数量



オン・ダイアグラム編集を行っている時に<Tab>キーを押すと、編集対象となる入力領域を以下のように移動できます。

[エンティティ名] → [主キー名] → [非キー名] → [エンティティ名]

属性の入力後、<Tab>キーではなく<Enter>キーを押した場合は、同一領域内で属性を追加できます。たとえば、主キー領域で属性を入力して<Enter>キーを押すと、さらに主キー属性を追加できます。これにより、複数のキー属性で構成される複合キーを作成できます。

6. 属性の入力が完了したら、<Shift + Enter>キーを押して、オン・ダイアグラム編集を終了します。

## データ型の設定



ERwin の論理モデルでは、属性に論理的なデータ型を設定できます。データ型を設定するには、リストボックスの一覧から適当なデータ型を選択します。物理モデルでは、ここで設定した論理データ型が、選択したデータベースがサポートする最適なデータ型に自動変換されます。

### 属性データ型の設定

1. [従業員] エンティティの [属性] ダイアログ ボックスを開きます。
2. [属性] ボックスの一覧の [従業員番号] 属性をクリックし、[データタイプ] タブをクリックします。
3. [従業員番号] に設定するデータ型を決定します。ここでは、char 型の長さ 5(char(5))を設定します。データ型のデフォルトは char(18) になっているので、データ型はそのままにして長さだけを「18」から「5」に変更します。[データタイプ] ボックスの一覧の [CHAR()] をクリックすると、[データタイプ] ボックスのデータ型が「CHAR(|)」と表示されます。「|」の部分に「5」と入力してください。
4. これで [従業員番号] 属性は「char(5)」と設定されました。同様の操作で [従業員] エンティティのすべての属性にデータ型を設定してください。

[従業員] エンティティ

属性名	データ型
従業員番号	CHAR(5)
従業員名	VARCHAR(20)
従業員住所	VARCHAR (40)
従業員電話番号	CHAR (10)

5. すべての属性にデータ型を設定したら、[OK] をクリックして [属性] ダイアログ ボックスを閉じます。



デフォルト データ型は CHAR(18)です。デフォルト データ型を変更するには、[モデル]メニューの [モデル プロパティ] をクリックして [モデル プロパティ] ダイアログ ボックスを開きます。[デフォルト] タブをクリックし、[デフォルトのデータタイプ] の [デフォルトの論理データタイプ] ボックスにデフォルトに設定するデータ型と桁数を設定してください。[モデル プロパティ] ダイアログ ボックスでは、表記法、NULL オプション、デフォルトの物理データ型などさまざまな設定ができます。それぞれの設定内容について確認することをお勧めします。[ERwin ツールボックス] ツールバーのツールをクリックしてマウスをダイアグラム上に移動すると、カーソルの形状が選択したツールに合わせて変化します。これにより、現在どのツールを選択しているかを判断できます。

上記の操作に従って、[注文] エンティティにデータ型を設定してください。

[注文] エンティティ

属性名	データ型
注文番号	CHAR(8)
注文日	DATE
注文ステータス	CHAR (4)
顧客番号	CHAR(6)
顧客名	VARCHAR(30)
顧客住所	VARCHAR(40)
顧客電話番号	CHAR(10)
製品番号	CHAR(7)
製品名	VARCHAR(20)
製品価格	NUMBER
注文数量	CHAR(3)

☆ MEMO ☆



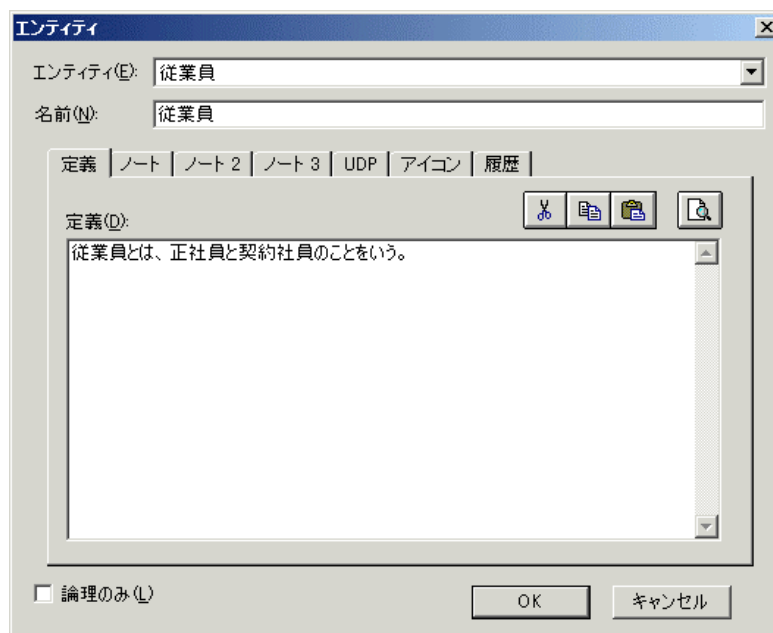
## エンティティと属性の定義入力

ここでは、エンティティと属性の定義の入力方法について学習します。

エンティティと属性の定義をきちんと入力しておけば、ER 図によるプレゼンテーションやテーブル定義書の作成をスピードアップできます。

### エンティティ定義の入力

1. [従業員] エンティティを右クリックし、ショートカットメニューの [エンティティ プロパティ] をクリックして [エンティティ] ダイアログボックスを開きます。
2. [定義] タブをクリックし、[定義] ボックスに「従業員とは、正社員と契約社員のことをいう。」と入力します。



3. 必要に応じて、さらに詳細なエンティティ情報を入力できます。[ノート] タブをクリックして、以下の情報を入力します。

入力タブ	入力値
ノート	従業員の登録は雇用契約の締結後に行う。



ERwinには、3種類のテーブル ノートが用意されています。これらのタブに入力した内容は、データベースには実装されません。エンティティに関する詳細な情報（サンプル データなど）を入力すると、エンティティ情報の共有に有効です。

4. [OK] をクリックして [エンティティ] ダイアログ ボックスを閉じます。
5. 同様の操作で、[注文] エンティティに以下のように入力します。

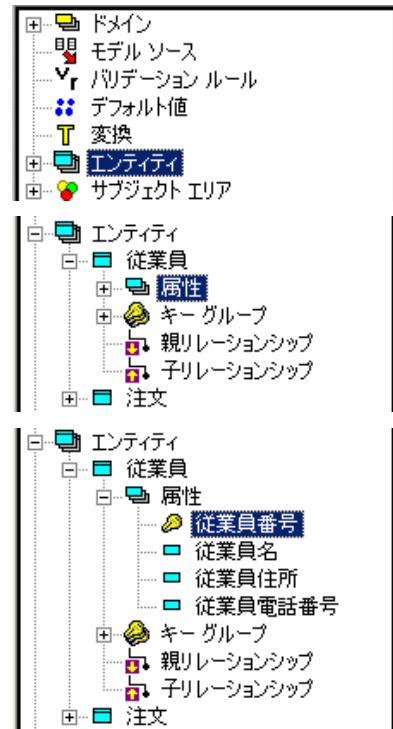
[注文] エンティティ

入力タブ	入力値
定義	注文とは、登録顧客より取り扱い製品を受注することをいう。
ノート	受注した場合、当日または翌日には情報を入力する。

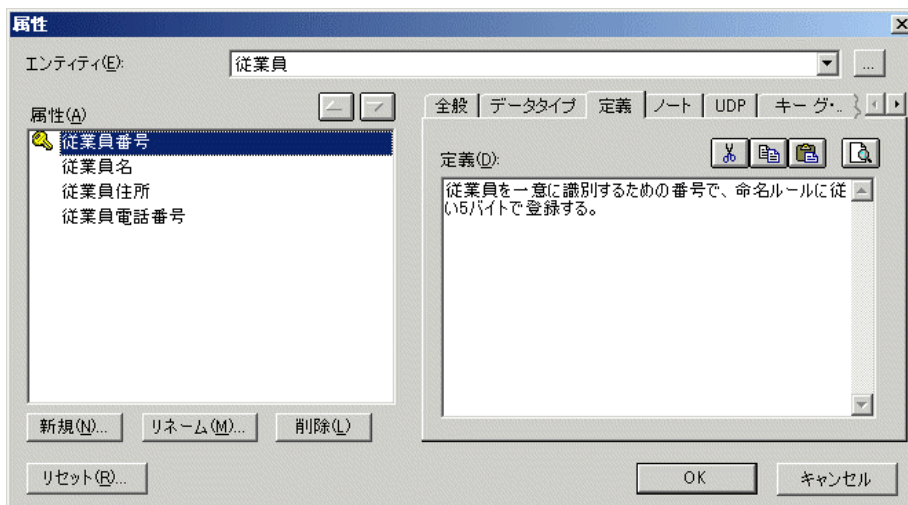
## 属性定義の入力

次は、モデル エクスプローラを使用してダイアログ ボックスを開きます。ダイアグラム ウィンドウの左側にあるモデル エクスプローラでは、ダイアグラムで作成したすべてのエンティティや属性を一覧表示できます。モデル エクスプローラから、任意のエンティティや属性のダイアログ ボックスを直接開くことができます。

1. モデル エクスプローラの[モデル]タブの[エンティティ] のプラス記号をクリックし、サブフォルダを展開すると、モデルに作成したすべてのエンティティが一覧表示されます。
2. エンティティ一覧から [従業員] エンティティのプラス記号をクリックし、さらに [属性] のプラス記号をクリックします。作成したすべての属性が一覧表示されます。
3. [従業員] エンティティの属性一覧から [従業員番号] をダブルクリックして、[属性] ダイアログ ボックスを開きます。



4. 属性に [従業員番号] が選択された状態で [属性] ダイアログ ボックスが開きます。[定義] タブをクリックし、[定義] ボックスに「従業員を一意に識別するための番号で、命名ルールに従い5バイトで登録する。」と入力します。



5. [属性] ボックスから [従業員名] をクリックして属性定義に「略字を使用せず正式な姓名で登録する。」と入力します。以下の表に従い、すべての属性に定義を入力し、[OK] をクリックして [属性] ダイアログ ボックスを閉じます。

[従業員] エンティティ

属性名	属性定義例
従業員番号	従業員を一意に識別するための番号で、命名ルールに従い5バイトで登録する。
従業員名	略字を使用せず正式な姓名で登録する。
従業員住所	都道府県から登録する。
従業員電話番号	市外局番から登録する。

[注文] エンティティ

属性名	属性定義例
注文番号	命名ルールに従い8バイトで登録する。
注文日	受注した年月日を西暦で登録する。
注文ステータス	受注状況を登録する。
顧客番号	製品を購入する企業を命名ルールに従い6バイトで登録する。
顧客名	略称を使用せず正式な社名で登録する。
顧客住所	都道府県から登録する。
顧客電話番号	市外局番から登録する。
製品番号	取り扱い製品を命名ルールに従い7バイトで登録する。
製品名	略称を使用せず正式な製品名で登録する。
製品価格	標準価格を登録する。
注文数量	受注本数を登録する。

## ドメインの作成

ここでは、ERwin の優れたデータベース設計機能の 1 つである、ドメインの作成方法について学習します。

### 説明

ワープロソフトにフォント名、フォントサイズ、左右インデントなどの書式をグループ化しておき、その書式グループを文書に設定することにより文書作成を効率化する機能があります。ERwin のドメインはこの機能と類似しています。

ERwin のドメインとは、論理モデルの属性プロパティ（データ型、属性定義、バリデーションルール、デフォルト値など）や物理モデルのカラムプロパティ（データ型、NULL オプションなど）をグループ化して名前を付けたオブジェクトを指します。ドメインを設定すると、予めドメインに定義しておいた全属性/カラムプロパティが自動的に設定されます。

ビジネスルールが変わり、属性やカラムのプロパティを変更する必要があるときも、ドメインのプロパティを変更すると、そのドメインを設定した全属性/カラムに変更内容が一括で反映されます。

通常、属性やカラムにプロパティを設定するには、それぞれに適したデータ型、定義、NULL オプションなどをプロパティごとに個別に指定しますが、ドメインを使用すれば、モデリングの工数を削減し、メンテナンスが容易になります。

### ドメインの洗い出し

ドメインとして登録する項目を洗い出します。まず、複数の属性で共通したプロパティを設定している場合、その共通した属性プロパティはドメインの候補となります。たとえば、金銭に関する属性が「標準単価」、「販売価格」、および「注文合計額」のように複数存在していて、それらの属性には共通してデータ型に `NUMBER` を設定しているとします。このとき、「金銭」という名のドメインを作成して、「`NUMBER`」で登録できます。また、世間一般で共通する規約もドメインの候補となります。たとえば、郵便番号情報を入力するカラムが複数存在するとします。郵便番号は郵政省の定めにより、7 桁以内と決まっています。このとき、「郵便番号」という名のドメインを「`VARCHAR2(7)`」で登録できます。

郵便番号が 5 桁から 7 桁へ変更された際には、郵便番号ドメインのデータ型を「`CHAR(5)`」から「`CHAR(7)`」に変更するだけで、郵便番号関連のすべての属性やカラムデータ型を一括して「`CHAR(7)`」に更新できます。このように、ドメインの使用はモデリングのメンテナンスに有効です。

## 複数の属性で使用している共通したプロパティを探す

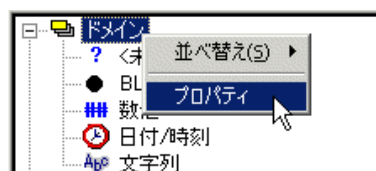
チュートリアルで設定した属性プロパティの中で、複数の属性で使用されている共通プロパティはないでしょうか？

エンティティ名	属性名	データ型	属性定義	共通プロパティ
従業員	従業員住所	VARCHAR(40)	都道府県から登録する。	データ型 属性定義
注文	顧客住所	VARCHAR(40)	都道府県から登録する。	
従業員	従業員電話番号	CHAR(10)	市外局番から登録する。	データ型 属性定義
注文	顧客電話番号	CHAR(10)	市外局番から登録する。	

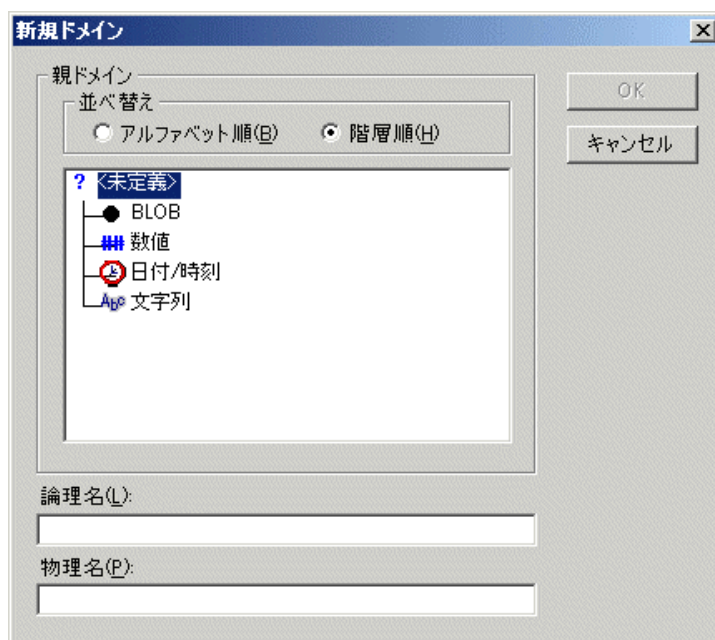
## ドメインの作成

上の表で網掛け表示された属性プロパティは、複数の属性で共通するプロパティです。この共通したプロパティをドメインとして登録します。

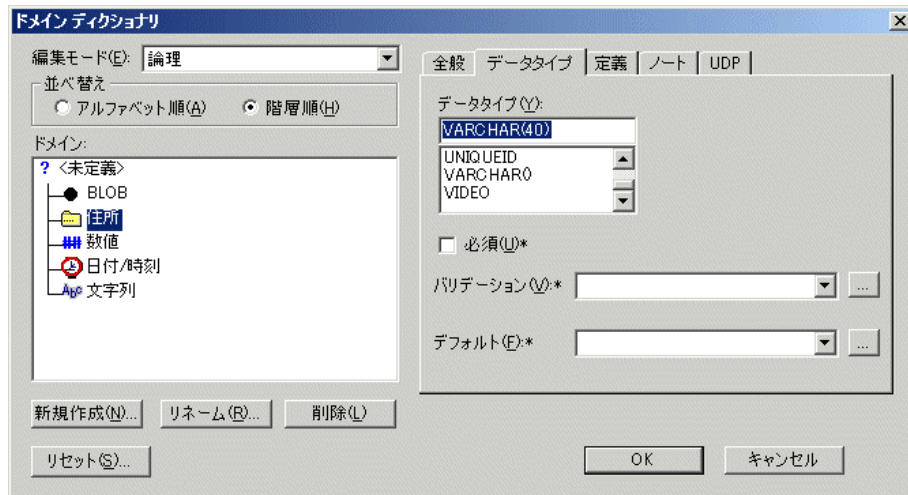
1. モデル エクスプローラの [モデル] タブをクリックします。[ドメイン] を右クリックし、ショートカット メニューの [プロパティ] をクリックして [ドメイン ディクショナリ] ダイアログ ボックスを開きます。



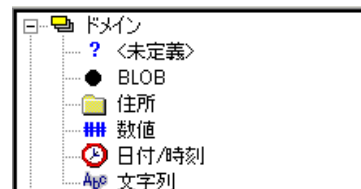
2. [ドメイン ディクショナリ] ダイアログ ボックスで、[新規作成] をクリックして [新規ドメイン] ダイアログ ボックスを開きます。



3. [論理名] ボックスに「住所」と入力し、[OK] をクリックして [ドメイン ディクショナリ] ダイアログ ボックスに戻ります。
4. [データタイプ] タブをクリックし、[データタイプ] ボックスの一覧の [VARCHAR0] をクリックします。[データタイプ] ボックスの「( )」内に「40」と入力し、桁数を 40 に設定します。



5. [OK] をクリックして [ドメイン ディクショナリ] ダイアログ ボックスを閉じます。
6. モデル エクスプローラの [ドメイン] の下に住所ドメインが作成されていることを確認してください。



## ドメインを属性に設定

1. [従業員] エンティティを右クリックし、ショートカットメニューの[属性]をクリックして[属性] ダイアログボックスを開きます。
2. [従業員住所] をクリックし、[全般] タブをクリックします。[ドメイン] グループから先ほど作成した[住所] ドメインをクリックします。この操作により、[従業員住所] 属性に住所ドメインを設定できました。
3. ダイアログボックス上部の[エンティティ] ボックスの一覧の[注文] をクリックし、対象エンティティを[注文] エンティティに切り替えます。
4. [顧客住所] をクリックし、[全般] タブの[ドメイン] グループから[住所] ドメインをクリックします。
5. [OK] をクリックして[属性] ダイアログボックスを閉じます。
6. 同様の操作で、「CHAR(10)」のデータ型を持つ電話番号ドメインを作成して、以下の属性に設定してください。

ドメイン名	データ型	テーブル名	カラム名
電話番号	CHAR(10)	従業員	従業員電話番号
		注文	顧客電話番号



## リレーションシップの作成

ここでは、リレーションシップの説明や作成方法を学習します。

ERwin は、リレーションシップを作成するだけで、親エンティティの主キーが子エンティティの外部キーとして自動的に継承され、エンティティ間の関連付けを自動的に行います。

### 説明

リレーションシップはエンティティ間の接続、リンク、関連などを表現するものです。最終的には物理データベースの外部キー制約や参照整合トリガとなるオブジェクトです。

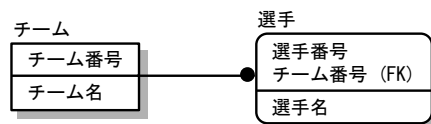
リレーションシップには、依存型、非依存型、および多対多型の 3 種類があります。依存型および非依存型は親テーブルの主キーを子テーブルに主キー属性として継承するか、非キー属性として継承するかによって使い分けます。多対多型は論理モデルだけに存在するモデル オブジェクトです。

#### 依存リレーションシップとは？



あるエンティティのインスタンスを一意に識別するために、他のエンティティから主キーを移行する必要がある場合は、親（参照元）エンティティから子（参照先）エンティティへ依存型リレーションシップを作成します。この場合、子エンティティの独自性は親に依存し、親から移行された外部キーは、子の主キーとして表示されます。

#### 例



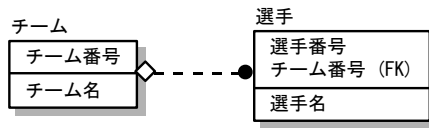
「選手」エンティティのインスタンスを一意に識別するためには、「選手番号」だけでなく、選手がどのチームに所属しているかを表す「チーム番号」との組み合わせが必要となります。

#### 非依存リレーションシップとは？



あるエンティティのインスタンスを一意に識別するために他のエンティティの主キーを参照する必要がない場合は、親から子へ非依存型リレーションシップを作成します。この場合、親から移行された外部キーは、子の非キーとして表示されます。非依存型リレーションシップには必須型とオプション型があり、必須型は外部キーの値に NULL を許可せず、オプション型は NULL 値を許可します。オプション型の場合、リレーションシップの始点（親エンティティ側）に白いひし形のマークが表示されます。

例



「選手」エンティティのインスタンスを一意に識別するためには、「選手番号」のデータがあればよく、選手がどのチームに所属しているかは必要ありません。

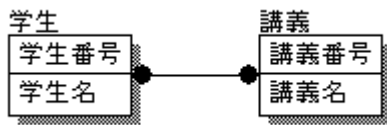
\*この例の非依存リレーションシップはオプション型です。

多対多リレーションシップとは？



多対多リレーションシップは、一方のエンティティの各インスタンスがもう一方のエンティティの複数インスタンスと関連し、反対側から見た場合にも同じ構造になっている2つのエンティティ間のリレーションシップです。リレーションシップの始点と終点（親子の両端）が黒丸の実線として表示されます。

例



学生と講義の関係を例に考えます。

1人の学生は複数の講義を履修します。また、1つの講義を複数の学生が受講します。[学生]エンティティ側からも[講義]エンティティ側からも相互に複数インスタンスと関連しています。

## 【従業員】エンティティと【注文】エンティティとの間のリレーションシップを考える



注文情報には、どの担当営業が受注した注文なのかを表すための従業員（担当営業）情報が必ず必要だと仮定します。このとき、【注文】エンティティで注文ごとに担当した営業担当者の名前を登録するのではリレーショナル データベースを効果的に活用しているとは言えません。なぜなら、もしある注文を担当した担当営業の姓が事情により変わったとすると、データ整合性を失わずに姓の変更をデータベース上に反映させるには、【従業員】エンティティと【注文】エンティティの両方を更新する必要があるからです。

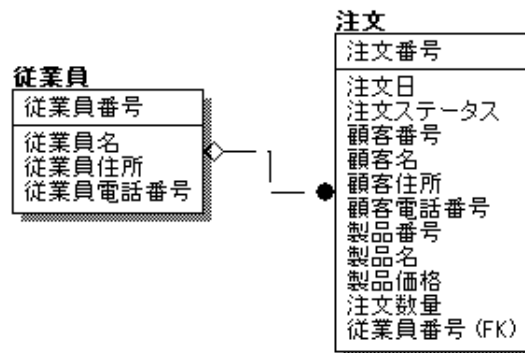
ここで【従業員】エンティティを【注文】エンティティに関連付けてみてはどうでしょうか？【注文】エンティティに登録する営業担当者の情報は、【従業員】エンティティから「従業員番号」を参照することにすれば、従業員の姓の変更は【従業員】エンティティに行うだけで済みます。

では、依存型と非依存型のどちらのリレーションシップを使用すべきでしょうか？【注文】エンティティのインスタンスを一意に識別するために、「従業員番号」というデータは必要ありません。すなわち、非依存型リレーションシップで関連付ければよいと分かります。

次に、非依存型リレーションシップのタイプを検討します。非依存型リレーションシップには、必須型とオプション型があります。【注文】エンティティには必ず担当営業情報が必要というビジネスルールがあるので、必須型の非依存型リレーションシップを選択する必要があります。

## 非依存型リレーションシップの作成

1. [ERwin ツールボックス] ツールバーの [非依存型リレーションシップ]  をクリックします。次に、マウス ポインタをダイアグラム上に移動し、カーソルが  に変わることを確認してください。
2. 親と子の順番でエンティティをクリックしてリレーションシップを作成します。まず、親の【従業員】エンティティをクリックし、次に【注文】エンティティをクリックします。





破線のリレーションシップ（非依存リレーションシップ）を作成すると、[従業員] エンティティの主キーである [従業員番号] が [注文] エンティティの非キーに継承されました。 [注文] エンティティの [属性] ダイアログ ボックスを開いて「従業員番号」のデータ型を確認してください。デフォルト データ型の CHAR(18)とは異なる、CHAR(5)が設定されています。ERwin では、リレーションシップを作成すると、親エンティティの主キーのプロパティが子エンティティの外部キーに継承されるのです。継承されるプロパティは、ドメイン、データ タイプ、定義、UDP、バリデーション ルール、およびデフォルトです。

3. [従業員] エンティティから [注文] エンティティへ破線のリレーションシップが作成され、リレーションシップと子エンティティとの接続部分に黒丸が付いていることを確認します。



リレーションシップを使ってエンティティ間に親子関係を作成すると、リレーションシップの到着先（子エンティティとの接点）に黒丸が表示されます。ただし、親子関係が決定していない多対多リレーションシップを使用する場合は、黒丸がリレーションシップの両端に表示されます。

## リレーションシップの定義入力

ここでは、親子エンティティのビジネス ルールをリレーションシップに定義する方法を学習します。  
ERwin は、親子エンティティのさまざまなビジネス ルールを ER 図上に表示できます。

### リレーションシップ定義の入力

1. [従業員] エンティティと [注文] エンティティ間のリレーションシップを右クリックし、ショートカット メニューの [リレーションシップ プロパティ] をクリックして [リレーションシップ] ダイアログ ボックスを開きます。
2. [全般] タブをクリックし、[動詞句] グループの [親から子へ] ボックスに「受ける」と入力します。



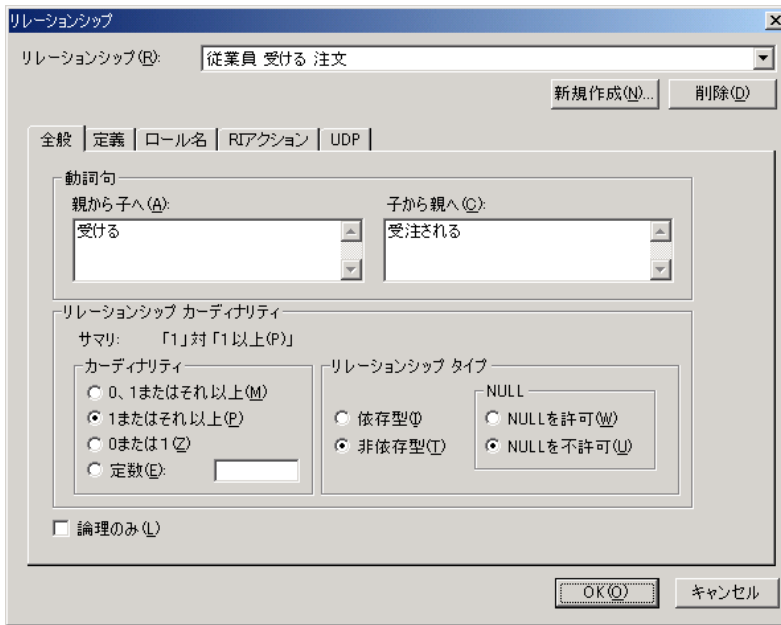
動詞句はエンティティ間の関連を表します。[親から子へ]は親から子への関係を表し、「親エンティティは子エンティティを<XXX (動詞句)>」と定義するように入力します。今回の例では、「従業員は注文を<承る>」となります。また、[子から親へ]の動詞句は、子から親への関係を表します。

3. [子から親へ] ボックスに「受注される」と入力します。
4. [ロール名] タブをクリックし、[ロール名] ボックスに「担当営業番号」と入力します。

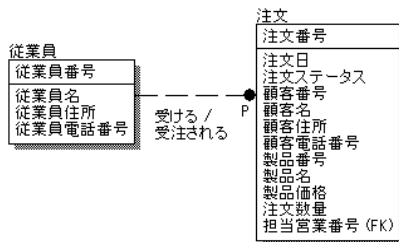


ロール名とは、親エンティティから子エンティティに移行される外部キーに定義する名前です。ロール名を使用することで、業務内容をより分かりやすく表現することができます。

5. [全般] タブをクリックし、[カーディナリティ] グループの [1 またはそれ以上 (P)] をクリックします。
6. [NULL] グループの [NULL を不許可] をクリックします。
7. [OK] をクリックして [リレーションシップ] ダイアログ ボックスを閉じます。



8. ダイアグラムの空白部分（何も無い部分）を右クリックし、ショートカットメニューの[リレーションシップの表示]をポイントし、[動詞句] および[カーディナリティ]をクリックします。リレーションシップに動詞句、および端の黒丸に「P」という文字が表示されます。



9. [従業員]エンティティと[注文]エンティティの間のリレーションシップにビジネスルールが表示されました。

☆以下に、[リレーションシップ] ダイアログボックスのオプションを簡単にまとめます。

オプション グループ	意味	オプション	表記
カーディナリティ	親と子がそれぞれ何対何の関係にあるかを定義します。カーディナリティは、親エンティティの1つのインスタンスが子エンティティのいくつかのインスタンスと結びつくかを指定します。	0、1 またはそれ以上	なし
		1 またはそれ以上 (P)	P
		0 または 1 (Z)	Z
		定数	定数
NULL	親エンティティから移行された外部キーにNULLを許すか否かを指定します。	NULL を許可	オプションナルシンボル (◇) あり
		NULL を不許可	オプションナルシンボル (◇) なし
リレーションシップタイプ	親エンティティから子エンティティへ外部キーをどのように移行するかを指定します。	依存型	主キーとして移行
		非依存型	非キーとして移行

## 正規化

ここでは、ERwin を使用した正規化について学習します。

ERwin を使用すれば、正規化を行っていない状態も ER 図で表現できるので、分析段階に合った ER 図を作成できます。

ここで学習する正規化ルールに従って、ER 図を注意深く分析して選択ツールによる操作を行えば、機械的に正規化作業を進めていくことができます。

### 説明

正規化とは、「属性が適切なエンティティに属しているか?」、「データに冗長性はないか?」などを確認していくデータモデリング アプローチを言います。一般的に、正規化は第 1 正規化から第 3 正規化まで行えば十分であると言われています。

正規化について学習する前に、ここで多用する選択ツールについて説明します。

属性を操作するこのツールは、キーボードのキーと合わせて使うことにより、属性のコピーや複数選択ができます。操作方法は、Windows のエクスプローラと同じなのでとても簡単です。しっかりと覚えて効率よくデータベース設計作業を行ってください。

選択ツールは、[ERwin ツールボックス] ツールバーから [選択]  をクリックして使用します。操作対象の属性をクリックすると、選択ツールが  (属性操作モード) になります。

### 属性の操作方法

- **属性を単体で選択するには**  
[選択] をクリックし、選択したい属性を含むエンティティをクリックします。次に、属性をクリックします。
- **連続した複数の属性を選択するには**  
[選択] をクリックし、選択したい属性を含むエンティティをクリックします。次に、〈Shift〉キーを押しながら属性をクリックします。
- **非連続の複数の属性を選択するには**  
[選択] をクリックし、選択したい属性を含むエンティティをクリックします。次に、〈Ctrl〉キーを押しながら属性をクリックします。
- **属性を移動するには**  
選択した (複数の) 属性を他のエンティティの希望の移動位置にドラッグ アンド ドロップします。
- **属性をコピーするには**  
選択した (複数の) 属性を〈Ctrl〉キーを押しながら他のエンティティの希望の移動位置にドラッグ アンド ドロップします。

- 属性を削除するには  
(複数の) 属性を選択し、<Del>キーまたは<Ctrl + x>キーを押します。選択された属性を削除するかを確認するダイアログ ボックスが表示されるので、[はい] をクリックします。



#### 選択ツールに関する備考

・依存型リレーションシップで子エンティティの主キー領域に移行された外部キーを非キー領域に移動することはできません。

・ [ERwin ツールボックス] ツールバーが表示されていない場合は、<Ctrl + t>キーを使用して表示/非表示を切り替えます。

## 第 1 正規化

### 第 1 正規形とは

すべての非キー属性は、唯一の値だけを持つことが要求される。一連の値あるいは繰り返し使われる値を持つことはない（繰り返しの排除）。

つまり、ひとつの属性に複数のデータが入ってしまうことがない状態を言います。日本語の名詞には単数形/複数形概念がないので、属性名だけでは実際に入ってくるデータを検証しきれないことがあります。サンプル インスタンス表などを利用して確認してみてください。

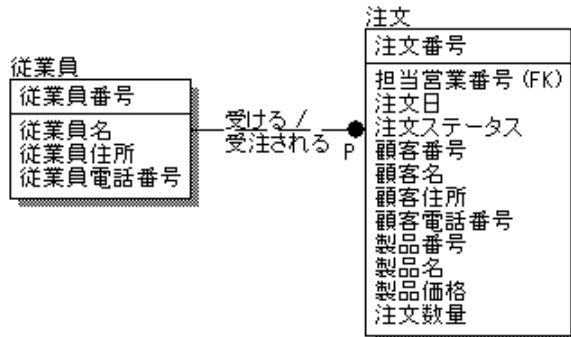
### ERwin を使用した第 1 正規化作業ルール

1. すべてのエンティティ内で、第 1 正規形に違反する繰り返しデータを持つ属性を探します。
2. エンティティから繰り返しデータを持つ属性を排除します。排除する属性を移動するための新規エンティティを作成します。
3. 選択ツールを使用して排除する属性を新規エンティティの非キー領域に移動します。
4. 元のエンティティから新しく作成したエンティティへ依存型リレーションシップを作成し、元のエンティティの主キー属性を新しいエンティティに移行します。
5. 移行された外部キーと組み合わせて、新しいエンティティのインスタンスを一意に識別できる主キーを再検討します。
6. 繰り返しデータを持つ属性に主キーとなり得る属性があれば、非キー領域から主キー領域へと移動し、ない場合は新しく主キーを追加します。
7. 新しいエンティティに適切な名前を付けます。



## 第1 正規化作業

[注文] エンティティのサンプル インスタンス表を見ながら第1 正規形について学習します。



注文番号	担当営業番号		製品番号	製品名	製品価格	注文数量
ORD56789	A1234		P-00123,P-00124	AllFusion ERwin Data Modeler 4.1.4, AllFusion Model Manager 4.1.4	900000,240000	5,5
ORD00001	C0012		P-00123	AllFusion ERwin Data Modeler4.1.4	900000	12
ORD00002	C7777		P-00124	AllFusion Model Manager 4.1.4	240000	2
ORD00003	B3321		P-00222	AllFusion Process Modeler 4.1.4	400000	3




▲ 注文エンティティのサンプル インスタンス表（一部の列を省略しています）

[注文] エンティティのサンプル インスタンス表を確認すると、「製品番号」、「製品名」、「製品価格」、および「注文数量」のデータに繰り返し属性があることが分かります。この繰り返し属性を排除して第1 正規形への変換を行います。



一般的に、サンプル インスタンス表の列にはカラム情報を書き、行にはサンプル インスタンス データを書く形式が多いようです。表を縦に分割している2重線の左側が主キーカラム、右側が非キーカラムとなります。つまり、2重線の左側に複数のカラムがあれば、それは複合キーということになります。


### ERwin を使用した第1 正規形への変換

1. [注文] エンティティの第1 正規形に違反する属性を排除します。排除する属性（「製品番号」、「製品名」、「製品価格」、および「注文数量」）を移動するための新しいエンティティをあらかじめ作成しておきます。[エンティティ]  をクリックし、新しいエンティティ [E/3] を作成します。
2. [選択]  をクリックし、[注文] エンティティをクリックします。属性の「製品番号」、「製品名」、「製品価格」、および「注文数量」を選択して、ドラッグ アンド ドロップで新規エンティティの非キー領域へ移動します。
3. [依存型リレーションシップ]  を使用して、[注文] エンティティから [E/3] エンティティへ依存型リレーションシップ「R/2」を作成します。これにより、「注文番号」が [E/3] エンテ

イティに移行されました。



独立エンティティは、親エンティティから依存型リレーションシップを引かれることによって子エンティティとなり、自動的に依存エンティティとなります。

4. [E/3]エンティティの主キーを検討します。[注文]エンティティから排除してきた属性の中で、「注文番号」属性との組み合わせで注文の明細を一意に識別できる属性を探してみると、「製品番号」属性が適当であると分かります。[選択]  をクリックし、「製品番号」属性を主キー領域に移動します。「注文番号」属性と「製品番号」属性を組み合わせると、どの注文で何という製品を何本受注したかが明確になります。このように複数の属性の組み合わせで作られる主キーを複合キーと呼びます。



**複合キーとは:**

複数の主キー属性のデータの組み合わせによりインスタンスを一意に識別するキー属性です。

5. 新規エンティティ [E/3] に適当な名前を付けます。この場合は、注文の明細情報を登録するエンティティですから、「注文明細」と入力します。

## 第 2 正規化

### 第 2 正規形とは

ダイアグラムが第 1 正規形であり、かつ主キー（一意識別子）の一部しか依存しない属性が存在しない状態である。

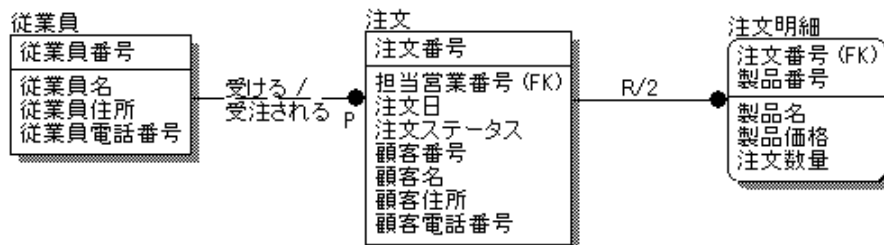
主キーの一部にしか依存しない属性がないかをチェックするということは、第 2 正規形の検討対象は、複合キーを持つエンティティだけとなります。複合キーの一部が決まれば、自動的に値が確定するような非キー属性が存在しなければ、それが第 2 正規形です。

### ERwin を使用した第 2 正規化作業ルール

1. 複合キーを持つエンティティの中で、第 2 正規形に違反する属性を探します。
2. 第 2 正規形に違反する属性を排除します。排除する属性を移動するための新規エンティティを作成します。
3. 選択ツールを使用して、非キー領域にあった属性は新規エンティティの非キー領域に、主キー領域にあった属性は新規エンティティの主キー領域に移動します。
4. 新しく作成したエンティティから元のエンティティへ依存型リレーションシップを引き、新しいエンティティの主キー属性を元のエンティティに移行して、最初の複合キーの状態に戻します。
5. 新しいエンティティに適当な名前を付けます。

### 第 2 正規化作業

[注文明細] エンティティのサンプル インスタンス表を見ながら第 2 正規形について学習します。






注文番号	製品番号	製品名	製品価格	注文数量
ORD56789	P-00123	AllFusion ERwin Data Modeler 4.1.4	900000	5
ORD56789	P-00124	AllFusion Model Manager 4.1.4	240000	5
ORD00001	P-00123	AllFusion ERwin Data Modeler 4.1.4	900000	12
ORD00002	P-00124	AllFusion Model Manager 4.1.4	240000	2

#### ▲ 注文明細エンティティのサンプル インスタンス表

[注文明細] エンティティのサンプル インスタンス表を確認すると、「製品名」および「製品価格」のデータは、複合キーの「製品番号」にだけ依存していることが分かります。この属性を排除して第 2 正規形への変換を行います。

### ERwin を使用した第 2 正規形への変換

1. [注文明細] エンティティの第 2 正規形に違反する属性を排除します。排除する属性（「製品番号」、「製品名」、および「製品価格」）を移動する新しいエンティティを予め作成しておきます。[エンティティ]  をクリックし、新しいエンティティ [E/4] を作成します。
2. [選択]  をクリックし、[注文明細] エンティティをクリックして、[注文明細] エンティティの [製品番号] 属性を選択して新規エンティティ [E/4] の主キー領域にドラッグ アンド ドロップします。同様の手順で、[製品名] 属性および [製品価格] 属性を非キー領域にドラッグ アンド ドロップします。
3. [依存型リレーションシップ]  をクリックして、[E/4] エンティティから [注文明細] エンティティへ依存型リレーションシップ「R/3」を作成します。これにより、「製品番号」属性が [注文明細] エンティティの主キーになり、[注文明細] エンティティは「注文番号」属性と「製品番号」属性による複合キーとなりました。
4. 新規エンティティ [E/4] に適切な名前を付けます。この場合は、製品情報を登録するエンティティですから、「製品」と入力します。

## 第3正規化

### 第3正規形とは

ダイアグラムが第2正規形であり、かつ非キー属性に依存する非キー属性が存在しない状態である。

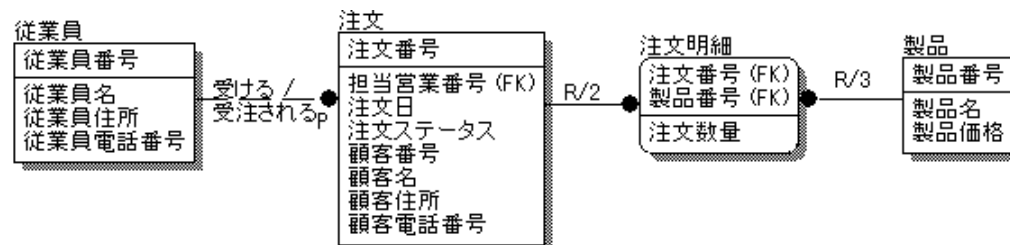
非キー領域の属性だけを使用して、主キーと非キーを持った別の新しいエンティティとして切り出せるかどうかを検討します。

### ERwin を使用した第3正規化作業ルール

1. すべてのエンティティで、第3正規形に違反する属性を探します。
2. 第3正規形に違反する属性を排除します。排除する属性を移動するための新規エンティティを作成します。
3. 選択ツールを使用して、排除する属性を新規エンティティの非キー領域に移動します。
4. 排除された属性の中から、新しいエンティティのインスタンスを一意に識別できる主キーを検討し、選択ツールを使用して非キー領域から主キー領域へと移動します。
5. 新しく作成したエンティティから元のエンティティに非依存型リレーションシップを引き、新しいエンティティの主キー属性を元のエンティティの非キー領域に移行します。
6. 新しいエンティティに適切な名前を付けます。

### 第3正規化作業

[注文] エンティティのサンプル インスタンス表を見ながら第3正規形について学習します。







注文番号	担当営業番号	注文日	注文ステータス	顧客番号	顧客名	顧客住所	顧客電話番号
ORD56789	A1234	1997/06/20	直販	CC0011	SPIRITS	XXXXXX	XXX-XXX-XXXX
ORD00001	C0012	1997/07/08	代理店経由	CC0022	信和工業	XXXXXX	XXX-XXX-XXXX
ORD00002	C7777	1997/07/21	当社販売	CC0033	横浜技術	XXXXXX	XXX-XXX-XXXX
ORD00003	B3321	1997/08/26	代理店	CC0044	ピアンキ	XXXXXX	XXX-XXX-XXXX

#### ▲ 注文エンティティのサンプル インスタンス表

[注文] エンティティのサンプル インスタンス表を確認すると、「顧客名」、「顧客住所」、および「顧客電話番号」のデータは、非キー領域に存在する「顧客番号」に依存していて、[注文] エンティティの主キーには依存していないことがわかります。この属性を排除して第3正規形への変換を行います。

### ERwin を使用した第 3 正規形への変換

1. [注文] エンティティの第 3 正規形に違反する属性を排除します。排除する属性（「顧客番号」、「顧客名」、「顧客住所」、および「顧客電話番号」）を移動する新しいエンティティをあらかじめ作成しておきます。[エンティティ]  をクリックし、新しいエンティティ [E/5] を作成します。
2. [選択]  をクリックし、[注文] エンティティをクリックして、第 3 正規形に違反していた属性（[顧客番号]、[顧客名]、[顧客住所]、および [顧客電話番号]）を選択して、[E/5] エンティティの非キー領域へドラッグ アンド ドロップします。
3. [E/5] エンティティの主キーを検討します。[注文] エンティティから排除してきた属性の中で、[E/5] エンティティのインスタンスを一意に識別できる属性を探してみると、「顧客番号」属性が適当であると分かります。[選択]  をクリックして [顧客番号] 属性を非キー領域から主キー領域に移動して、[E/5] エンティティの主キーとします。
4. [非依存型リレーションシップ]  を使用して、[E/5] エンティティから [注文] エンティティに非依存型リレーションシップ「R/4」を作成します。これにより、「顧客番号」が [注文] エンティティの非キーに移行されました。
5. 新規エンティティ [E/5] に適切な名前を付けます。この場合は、製品情報を登録するエンティティですから、「顧客」と入力します。

## 導出属性の登録

データベース設計では、データの依存関係を明らかにし、冗長性を排除する正規化作業は必ず必要です。しかし、実際のシステム構築においては、データベースやクライアント アプリケーションのプラットフォーム環境、マシンのスペック、データ件数などのさまざまな要因を考慮して、アプリケーションのパフォーマンスを向上させる必要があります。現実の開発においては、パフォーマンスの向上のために故意にデータベース上に冗長データや導出属性を持たせることがあります。

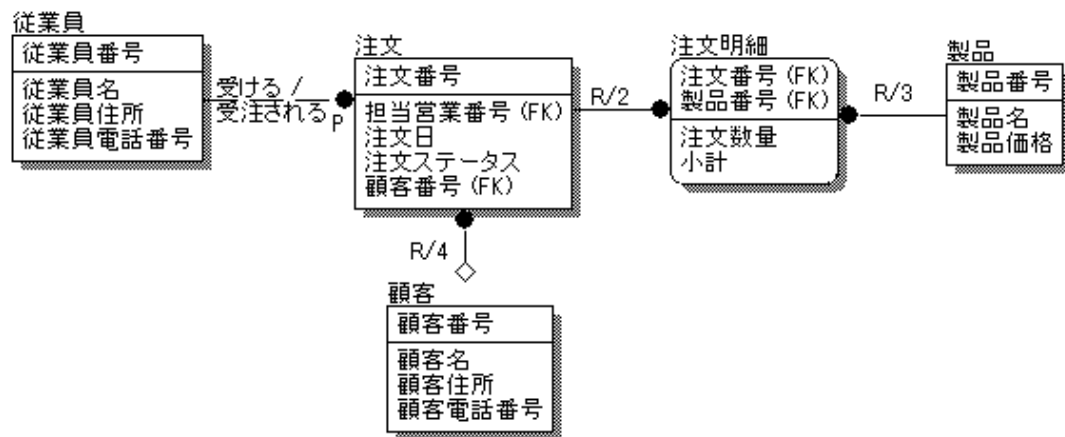


### 導出属性とは:

導出属性とは、属性のデータを加減乗除または統計演算することにより導き出すデータを格納する属性です。

[注文明細] エンティティの非キーに「小計」という属性を入力してください。データタイプは、NUMBER にします。この属性は、受注製品の単価と数量を掛け算して、製品ごとの受注金額を導き出した値を登録する属性です。

ここまでの操作で以下のような ER 図が作成されたことを確認してください。



ここで論理設計は終了します。ここまで作成してきた ER 図をファイルに保存します。

1. [ファイル] メニューの [名前を付けて保存] をクリックして [名前を付けて保存] ダイアログ ボックスを開きます。
2. [ファイルの種類] ボックスで [ERwin (\*.er1)] が選択されていることを確認し、ファイルを保存するディレクトリを選択します。次に、[ファイル名] ボックスに「MYMODEL」と入力します。
3. [OK] をクリックして [名前をつけて保存] ダイアログ ボックスを閉じます。

☆ MEMO ☆



## 物理設計（物理名の設定）

ここでは、ERwin のダイアグラム オブジェクトに物理名を設定する方法について学習します。物理名とは、データベースに実装するオブジェクト名を表します。

### 説明

ERwin は、ダイアグラム オブジェクトの名前を論理と物理で別々に登録できます。これまでの Lesson では、オブジェクトはすべて論理名で登録してきました。


概念設計および論理設計フェーズでは、オブジェクト名に業務に沿った名前を使用します。このように作成した ER 図は、データ構造を理解しやすいので ER 図の討論や実際の業務担当者へのプレゼンテーションに役立ちます。ここで使用しているオブジェクト名（論理名）に対して実際のデータベースに実装する名前を「物理名」と呼びます。

論理名と物理名を別々に登録できるので、ダブルバイトのオブジェクトを作成できないデータベースをお使いの方でも、ダブルバイトを使用したデータベース設計図を書くことができます。たとえば、論理名は実際の帳票で使用している日本語の項目名を使用し、物理名はシングル バイトのアルファベットを使用することもできます（例：論理名は「会社」、物理名は「KAISHA」）。

論理名と物理名を別々に登録できるオブジェクト タイプには以下の 3 種類があります。

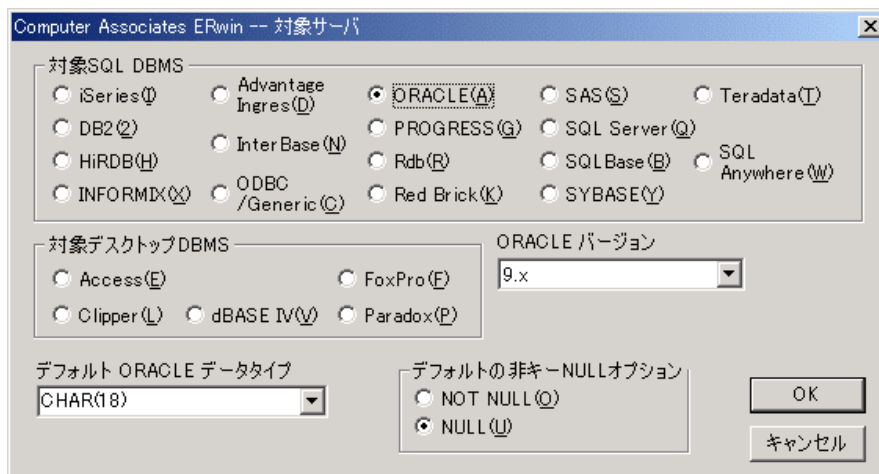
論理名	物理名
エンティティ名	テーブル名
属性名	カラム名
親から子への動詞句	外部キー制約名

### 論理モデルから物理モデルへの切り替え

1. [ERwin 標準ツールバー] の [論理/物理モデル]  ボックスの一覧の [物理] をクリックします。
2. ダイアグラム ウィンドウが物理モデルに切り替わります。

## 対象データベースの指定

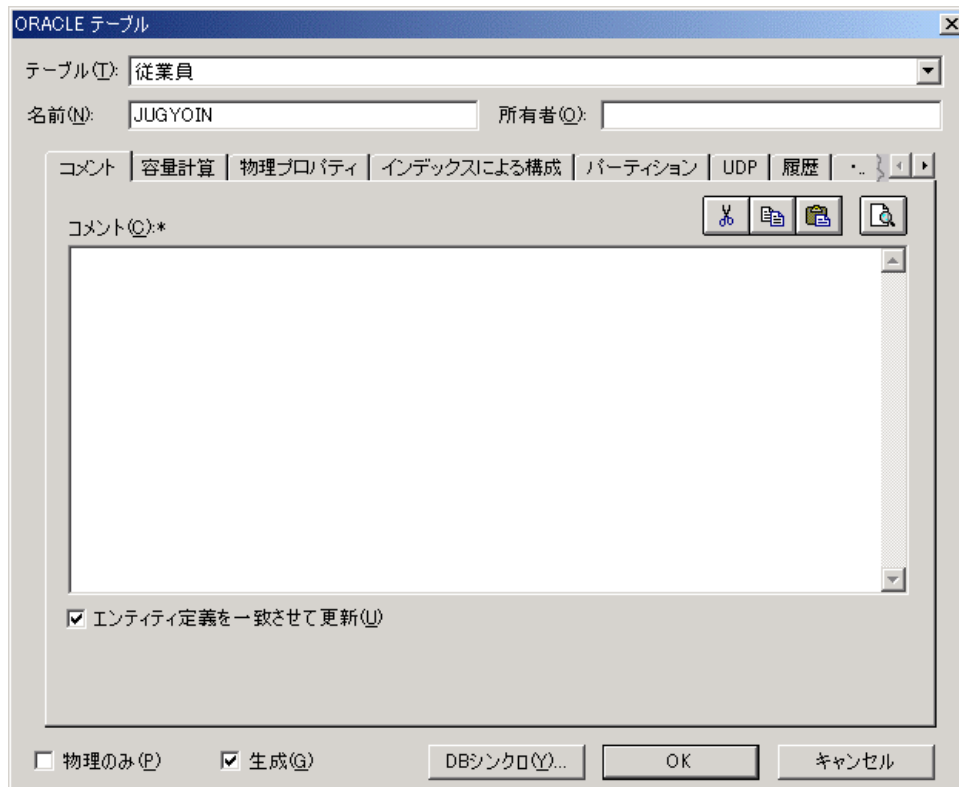
1. [データベース] メニューの [データベースの選択] をクリックして [対象サーバ] ダイアログ ボックスを開きます。
2. 対象データベースとして「ORACLE 9.x」が選択されていることを確認します。
3. [OK] をクリックして [対象サーバ] ダイアログ ボックスを閉じます。



このチュートリアルは、対象データベースに Oracle を選択して進めていきますが、お使いの環境が Oracle と接続できない場合は、ERwin がサポートする他のデータベースを選択して Lesson を進めてください。この場合は、チュートリアルの挿絵や選択オプションがお使いの ERwin の画面と異なります。

## 物理名の設定（テーブル名、カラム名）

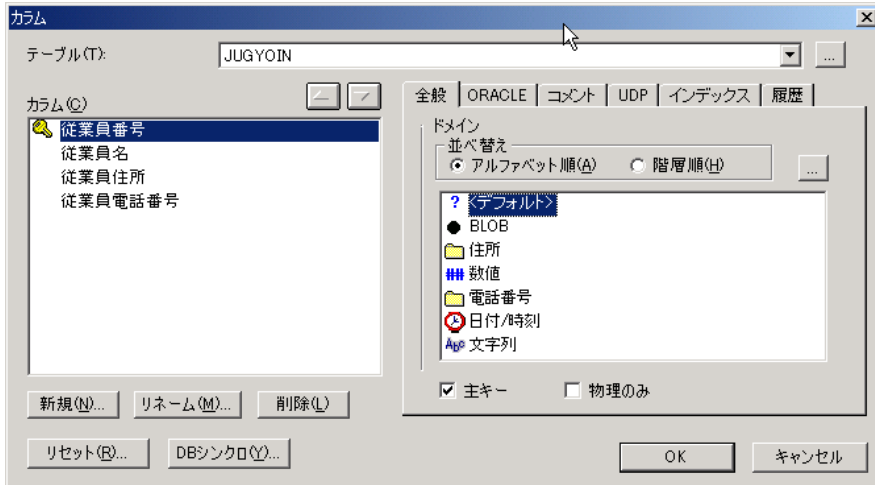
1. [従業員] テーブルを右クリックし、ショートカット メニューの [テーブル プロパティ] をポイントします。次に、任意のショートカット メニューをクリックして [ORACLE テーブル] ダイアログ ボックスを開きます。
2. [名前] ボックスに入力されているテーブル名を「従業員」から「JUGYOIN」に変更します。



### 対象データベース別の命名ルール:

オブジェクト名に使用できる最大文字数は、データベースにより異なります（たとえば Oracle のテーブル名に使用できる最大文字数は 30 バイトと決められています）。ERwin は対象データベースで決められている最大文字数を超過してダイアグラム オブジェクトの物理名を登録できません（もちろん、データベースへの実装と無関係な論理名に関しては問題ありません）。

3. [OK] をクリックして [ORACLE テーブル] ダイアログ ボックスを閉じます。
4. [JUGYOIN] テーブルを右クリックし、ショートカット メニューの [カラム] をクリックして [カラム] ダイアログ ボックスを開きます。
5. [カラム] ボックスの一覧の [従業員番号] カラムをクリックします。

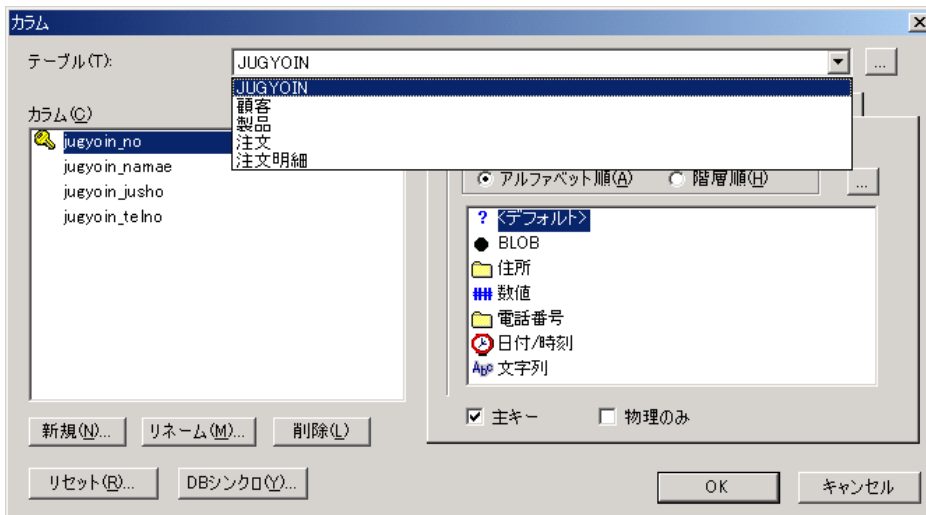


6. [リネーム] をクリックして、[カラムのリネーム] ダイアログ ボックスを開き、カラム名を「従業員番号」から「jugyoin\_no」に変更します。
7. 同様の手順で [JUGYOIN] テーブルのその他の属性のカラム名を設定し、最後に [OK] をクリックします。

[従業員] エンティティ ( [JUGYOIN] テーブル)

属性名	カラム名
従業員名	jugyoin_namae
従業員住所	jugyoin_jusho
従業員電話番号	jugyoin_telno

8. [JUGYOIN] テーブルの全カラム名の登録が終了したら、他のテーブルにあるカラム名の登録を行います。[テーブル] ボックスの一覧の任意のテーブルをクリックし、次ページの表にあるテーブルとカラムの名前を設定してください。すべての登録が終了したら、[OK] をクリックして [カラム] ダイアログ ボックスを閉じます。





### 編集対象のテーブルを切り替えるには:

あるテーブルのプロパティの編集を完了したら、ダイアログ ボックスを開き直さずに編集対象のテーブルを切り替えることができます。[カラム] ダイアログ ボックスの [テーブル] ボックスには、ダイアグラムに存在するすべてのテーブルが一覧表示されます。[テーブル] ボックスの一覧から次に編集したいテーブルをクリックすると、編集対象のテーブルが切り替わります。編集対象のテーブルを切り替えた場合、直前に編集したテーブルの内容はすべて保存されます。[OK] をクリックしてダイアログ ボックスを閉じ、次にテーブルをダブルクリックすると、最後に開いたときのダイアログ ボックスが開きます。

[顧客] エンティティ ( [KOKYAKU] テーブル)

論理名		物理名	
エンティティ名	属性名	テーブル名	カラム名
顧客	顧客番号	KOKYAKU	kokyaku_no
	顧客名		kokyaku_namae
	顧客住所		kokyaku_jusho
	顧客電話番号		kokyaku_telno

[製品] エンティティ ( [SEIHIN] テーブル)

論理名		物理名	
エンティティ名	属性名	テーブル名	カラム名
製品	製品番号	SEIHIN	seihin_no
	製品名		seihin_namae
	製品価格		seihin_kakaku

[注文] エンティティ ( [CHUMON] テーブル) 注: カラム順を以下のように並べ替えてください。

論理名		物理名	
エンティティ名	属性名	テーブル名	カラム名
注文	注文番号	CHUMON	chumon_no
	顧客番号		kokyaku_no
	担当営業番号		tantou_eigyō_no
	注文日		chumon_date
	注文ステータス		chumon_status

[注文明細] エンティティ ( [CHUMON\_MEISAI] テーブル)

論理名		物理名	
エンティティ名	属性名	テーブル名	カラム名
注文明細	注文番号	CHUMON_MEISAI	chumon_no
	製品番号		seihin_no
	注文数量		chumon_suuryo
	小計		shoukei



#### カラム名（物理属性名）の外部キーへの継承:

リレーションシップで接続された親テーブルの主キー カラム名を更新した場合、デフォルトではカラム名の更新は子テーブルの外部キーに継承されません。

## カラムのデータ型と NULL オプションの設定

物理モデルでは、論理モデルで設定したデータ型が物理モデルの対象データベースがサポートするデータ型に自動変換されます。

1. [JUGYOIN] テーブルの [カラム] ダイアログ ボックスを開きます。
2. [カラム] ボックスの一覧の [jugyoin\_namae] カラムをクリックし、[ORACLE] タブをクリックします。データ型が「varchar2 (20)」であることを確認します。
3. 次に、[NULL オプション] の [NOT NULL] をクリックします。
4. [jugyoin\_namae] カラムは、「varchar2 (20)」の「NOT NULL」と設定されました。同様の操作で [JUGYOIN] テーブルのカラムすべてのデータ型を確認し、NULL オプションを設定してください。

[JUGYOIN] テーブル

カラム名	データ型	NULL オプション
jugyoin_no	char(5)	NOT NULL
jugyoin_namae	varchar2(20)	NOT NULL
jugyoin_jusho	varchar2(40)	NULL
jugyoin_telno	char(10)	NULL

5. すべてのカラムにデータ型と NULL オプションを設定したら、[OK] をクリックして [カラム] ダイアログ ボックスを閉じます。

次ページの表に従って、他のテーブルでのカラムのデータ型を確認し、NULL オプションを設定してください。

[KOKYAKU] テーブル

カラム名	データ型	NULL オプション
kokyaku_no	char(6)	NOT NULL
kokyaku_nameae	varchar2(30)	NOT NULL
kokyaku_jusho	varchar2(40)	NULL
kokyaku_telno	char(10)	NOT NULL

[CHUMON] テーブル

カラム名	データ型	NULL オプション
chumon_no	char(8)	NOT NULL
kokyaku_no	char(6)	NOT NULL
tantou_eigyō_no	char(5)	NULL
chumon_date	date	NULL
chumon_status	char(4)	NULL

[CHUMON\_MEISAI] テーブル

カラム名	データ型	NULL オプション
chumon_no	char(8)	NOT NULL
seihin_no	char(7)	NOT NULL
chumon_suuryo	char(3)	NOT NULL
shoukei	number	NULL

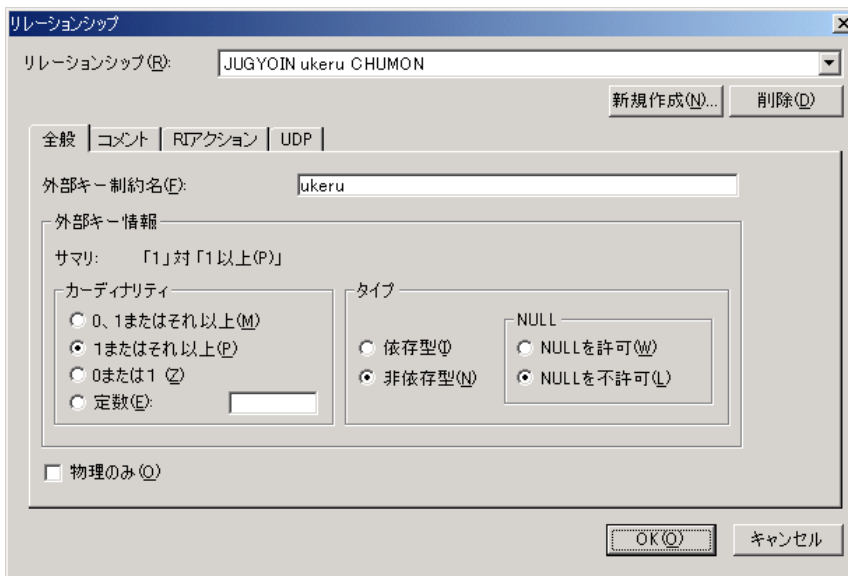
[SEIHIN] テーブル

カラム名	データ型	NULL オプション
seihin_no	char(7)	NOT NULL
seihin_nameae	varchar2(20)	NOT NULL
seihin_kakaku	number	NULL

## 物理名の設定（外部キー制約名）

物理モデルでは、論理モデルの [親から子へ] ボックスで設定した動詞句の値が [外部キー制約名] に置き換わります。

1. [JUGYOIN] テーブルと [CHUMON] テーブルの間のリレーションシップを右クリックし、[リレーションシップ プロパティ] をクリックして、[リレーションシップ] ダイアログ ボックスを開きます。
2. [リレーションシップ] ダイアログ ボックスの [外部キー制約名] ボックスに「ukeru」と入力します。



以下の表に従って、Lesson 8 で追加したリレーションシップの外部キー制約名を変更してください。表にない項目は自由に設定してください。リレーションシップの入力方法の詳細については、Lesson 7 を参照してください。

これからの Lesson で必要な定義だけを入力します。

変更前の 外部キー制約名	NULL オプション	リレーションシップ タイプ	変更後の 外部キー制約名
R/2	—	依存型	motsu
R/3	—	依存型	hitsuyou
R/4	NULL を不許可	非依存型	dasu



非依存型リレーションシップの [リレーションシップ] ダイアログ ボックスでは、NULL オプションを指定できます。ただし、外部キーの NULL オプションは、[カラム] ダイアログ ボックスでの設定が [リレーションシップ] ダイアログ ボックスでの設定に優先して保存されます。



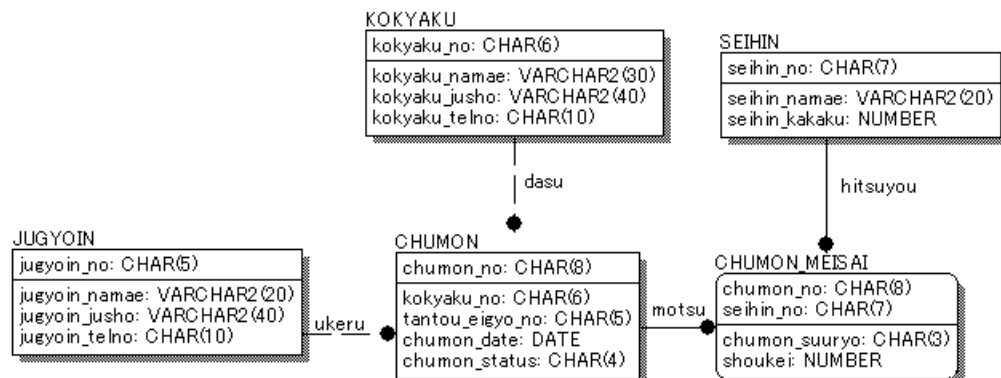
## 表示レベルの変更

ERwin では、モデル情報をさまざまなレベルで表示できます。たとえば、テーブルの名前だけを表示する「テーブル レベル」、テーブルの主キーだけを表示する「主キー レベル」などがあります。

表示レベル	内容
テーブル	テーブル名だけを表示します。
カラム	テーブル名、主キー カラム名および非キー カラム名を表示します。
主キー	テーブル名と主キー カラムを表示します。
コメント	テーブル名とテーブルのコメントを表示します。
物理順序	テーブル名とカラム名を表示します。ただし、カラムはデータベースに格納される順序で表示します。この表示レベルでカラム順序を変更しても、「カラム レベル」表示レベルには反映されません。
テーブルの表示	物理情報を表示します。
カラムのデータタイプ	カラムのデータ型を表示します。
カラムのドメイン	カラムのデータ型および設定ドメインを表示します。
NULL オプション	カラムの NULL オプションを表示します。
主キー識別子	主キー識別子を表示します。
外部キー識別子 (FK)	外部キー識別子 (FK) を表示します。
代替キー識別子 (AK)	代替キー (AK) および逆方向エントリ (IE) 識別子を表示します。
テーブル所有者	テーブルの所有者を表示します。

ここでは、表示レベルを「カラムのデータタイプ」に設定します。

1. ER ダイアグラムの空白部分を右クリックし、ショートカット メニューの [表示レベル] をポイントします。次に、[カラム] をクリックします。
2. 同様に、ER ダイアグラムの空白部分を右クリックし、ショートカット メニューの [テーブルの表示] をポイントします。次に、[カラムのデータタイプ] をクリックします。
3. ER ダイアグラムの物理情報 (テーブル名、カラム名、およびデータ型) が表示されます。



☆ MEMO ☆

## 物理設計（インデックスの設定）

ここでは、ERwin でインデックスを作成する方法について学習します。

適切なインデックスを作成することにより、データベースの検索スピードを向上できます。

### 説明

インデックスとは、データベースの検索をスピードアップするための機能です。たとえば、100 万件の注文情報を格納した [CHUMON] テーブルで、ある顧客の注文情報だけが必要であるとします。インデックスを使用しないと 100 万件のデータを持つ [CHUMON] テーブルを全表走査する必要がありますが、注文顧客情報を格納する「kokyaku\_no」カラムにインデックスを作成すれば、顧客情報だけを検索して必要なデータが存在するデータベース上の番地を突き止めることができます。

### ERwin で作成できるインデックス

ERwin では以下の 4 種類のインデックスを作成できます。

インデックスの種類	ERwin での呼び名	内容
主キー インデックス	PK インデックス	主キーに作成するインデックス
外部キー インデックス	FK インデックス	外部キーに作成するインデックス
ユニーク インデックス	AK インデックス	インデックス データに重複を許さないインデックス
非ユニーク インデックス	IE インデックス	インデックス データに重複を許すインデックス

設計者は、ダイアグラム オブジェクトにどのような名前を付けるかの命名規則を事前に定義しておくべきです。ERwin は、以下の命名規則に従ってインデックスを作成します。

「X」 + 「PK」（または「FK」、「AK」、「IE」 + n） + テーブル名


[CHUMON] テーブルにインデックスを作成したときの例

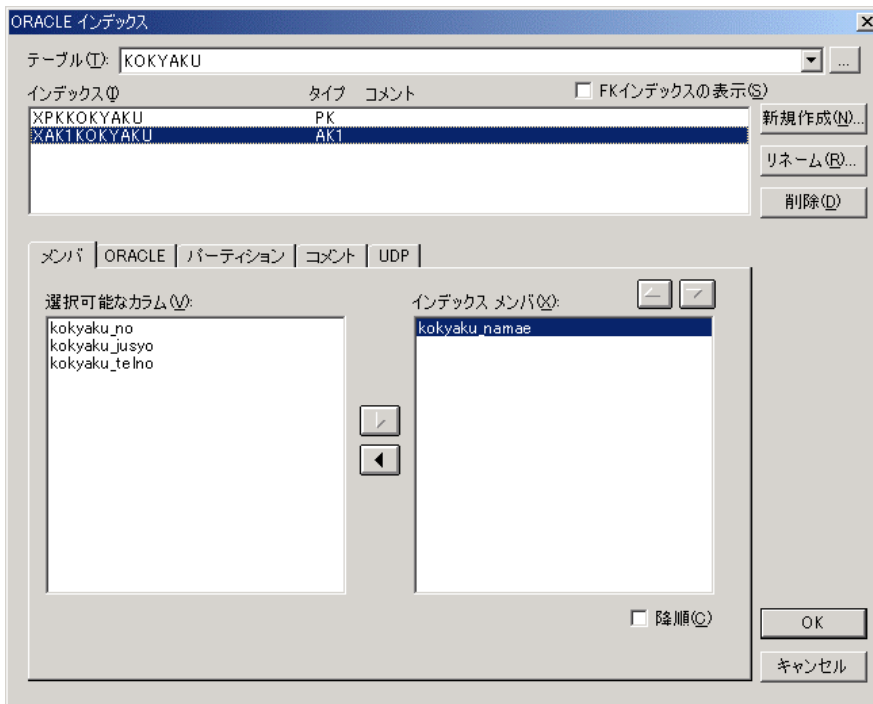
インデックスの種類	ERwin の命名ルール	例
主キー インデックス	「XPK」 + 「テーブル名」	XPKCHUMON
外部キー インデックス	「XIFn」 + 「テーブル名」	XIF1CHUMON
ユニーク インデックス	「XAKn」 + 「テーブル名」	XAK1CHUMON
非ユニーク インデックス	「XIE n」 + 「テーブル名」	XIE1CHUMON

## インデックスの作成

### ユニーク インデックスを作成するには

顧客情報の検索スピードを向上させるために [KOKYAKU] テーブルの「kokyaku\_namae」カラムにインデックスを作成します。顧客情報のマスター テーブルである [KOKYAKU] テーブルには、同じ会社を重複して登録することはありません。このような場合、ユニーク インデックスが適当であると言えます。ユニーク インデックスを設定したカラムには、データベース側で自動的に NOT NULL 制約が設定され、重複したデータと NULL 値が挿入されることを防ぐことができます。


1. [KOKYAKU] テーブルを右クリックし、ショートカット メニューの [インデックス] をクリックして [ORACLE インデックス] ダイアログ ボックスを開きます。
2. [新規作成] をクリックして [インデックスの新規作成] ダイアログ ボックスを開きます。
3. [キー グループ] と [インデックス] の値はそのままにします。ここではユニーク インデックス (AK) を作成するため、[一意] チェック ボックスをオンにし、[OK] をクリックします。
4. 「XAK1KOKYAKU」という名前のインデックスが [インデックス] ボックスに追加されていることを確認します。
5. [インデックス] ボックスで [XAK1KOKYAKU] をクリックし、[メンバ] タブの [選択可能なカラム] ボックスで [kokyaku\_namae] カラムをクリックします。次に、 をクリックして [kokyaku\_namae] カラムを [インデックス メンバ] ボックスへ移動します。

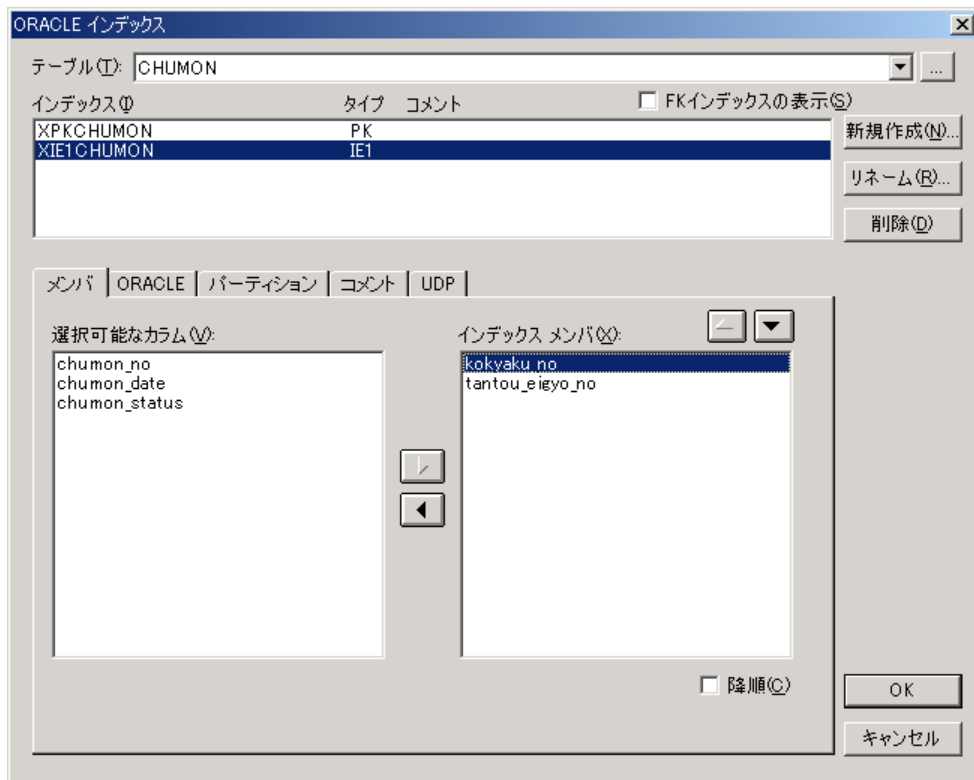


6. [ORACLE] タブをクリックして、[一意] チェック ボックスがオンになっていることを確認します。このチェック ボックスをオンまたはオフにすることによって、ユニーク インデックスと非ユニーク インデックスを変更できます。
7. [OK] をクリックして [ORACLE インデックス] ダイアログ ボックスを閉じます。

## 非ユニーク インデックスを作成するには

次に、「ある営業が担当した、ある顧客に関する情報」の検索スピードを向上させるためのインデックスを作成します。[CHUMON] テーブルの「kokyaku\_no」カラムと「tantou\_eigy\_no」カラムとの複合インデックスを作成します。ある担当営業は同じ顧客から複数の注文を受けることがあるため、このインデックスは非ユニーク インデックスが適当です。非ユニーク インデックスを設定したカラムには、重複したデータと NULL 値を挿入できます。


1. [CHUMON] テーブルを右クリックし、ショートカット メニューの [インデックス] をクリックして [ORACLE インデックス] ダイアログ ボックスを開きます。
2. [新規作成] をクリックし、[インデックスの新規作成] ダイアログ ボックスを開きます。
3. [キー グループ] と [インデックス] の値はそのままにします。ここでは非ユニーク インデックス (IE) を作成するため、[一意] チェック ボックスをオフにし、[OK] をクリックします。
4. 「XIE1CHUMON」という名前のインデックスが、[インデックス] ボックスに追加されていることを確認します。
5. [インデックス] ボックスで [XIE1CHUMON] をクリックし、[メンバ] タブの [選択可能なカラム] ボックスで [kokyaku\_no] カラムをクリックします。次に、 をクリックして [kokyaku\_no] カラムを [インデックス メンバ] ボックスへ移動します。[tantou\_eigy\_no] カラムも同様に [インデックス メンバ] ボックスへ移動します。



6. [ORACLE] タブをクリックして、[一意] チェック ボックスがオフになっていることを確認します。このチェック ボックスをオンまたはオフにすることによって、ユニーク インデックスと非ユニーク インデックスを変更できます。
7. [OK] をクリックして [ORACLE インデックス] ダイアログ ボックスを閉じます。

### インデックス カラムの並び順を変更するには

複合インデックスを作成する場合は、インデックス カラムの並び順を考慮する必要があります。インデックス カラムの並び順を変更すると、検索スピードを向上できます。

1. [CHUMON] テーブルを右クリックし、ショートカット メニューの [インデックス] をクリックして [ORACLE インデックス] ダイアログ ボックスを開きます。
2. [インデックス] ボックスで [XIE1CHUMON] をクリックし、[メンバ] タブをクリックします。
3. [XIE1CHUMON] インデックスは、①「kokyaku\_no」カラム、②「tantou\_eigy\_no」カラムの並び順であることがわかります。この並び順を逆にします。
4. [インデックス メンバ] ボックスの [tantou\_eigy\_no] カラムをクリックします。次に、[インデックス メンバ] の上部にある  をクリックすると、「tantou\_eigy\_no」カラムが「kokyaku\_no」カラムの上に移動します。
5. [OK] ボタンをクリックして [ORACLE インデックス] ダイアログ ボックスを閉じます。

### 作成したインデックスをダイアグラムに表示するには

ダイアグラムの空白部分を右クリックして、ショートカット メニューの [テーブルの表示] をポイントします。次に、[代替キー識別子 (AK)] をクリックします。これにより、テーブルのカラム名の後ろに、「AKn」（AK インデックスの場合）または「IE n」（IE インデックスの場合）が表示されます。

## 物理設計（バリデーション ルールの設定）

ここでは、データベースに設定するバリデーション ルールの作成方法とカラムへの設定方法について学習します。

### 説明

バリデーション ルールとは、ORACLE では「チェック制約」、SYBASE では「ルール」と呼ばれる制約のことであり、データベースの入力に明示的な条件を定義して、条件を満たすデータだけが入力を許可される機能です。これを利用すればデータベースに不正なデータが格納されることを防ぐことができます。

#### ERwin で作成できるバリデーション ルール

ERwin の [バリデーション ルール] ダイアログ ボックスでは、以下のような制約を定義できます。

種類	内容	サンプル
最小値	入力可能な最小値を定義する。	CHECK (カラム 1 >= 1)
最大値	入力可能な最大値を定義する。	CHECK (カラム 2 <= 10)
範囲	入力可能なデータの範囲を定義する。	CHECK (カラム 3 BETWEEN 1 AND 10)
有効値	入力可能なデータの種類を定義する。	CHECK (カラム 4 IN (松, 竹, 梅))
NOT	上記の条件を満たさないデータを許可する。	CHECK (カラム 5 NOT BETWEEN 1 AND 10)
引用符	条件となる値を引用符で囲む。	CHECK (カラム 6 NOT BETWEEN '1' AND '10')

### バリデーション ルールの作成

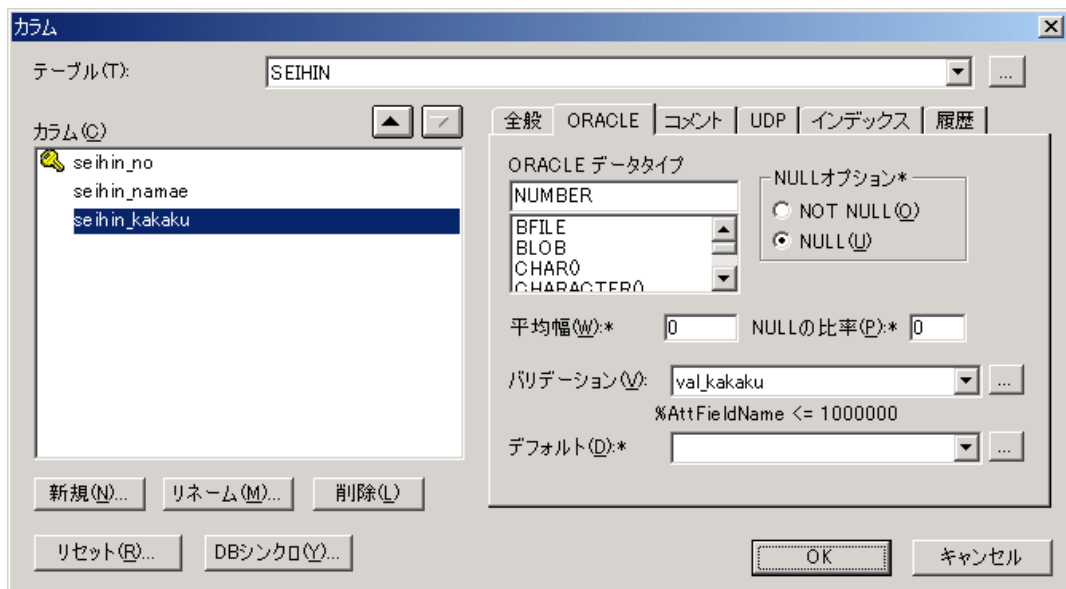
以下のカラムに入力されるデータを検討してみると、あるルールが存在することがわかりました。

カラム名（属性名）	ビジネスルール	入力可能な値
seihin_kakaku (製品価格)	製品の価格は 100 万円以下とする。	100 万円以下
chumon_status (注文ステータス)	直接販売か、代理店による販売かを登録する。	直販、代理

ユーザーの入力データが上記のルールと合致するかを確認するためのバリデーション ルールを作成します。バリデーション ルールをデータベースに実装すると、定義に反するデータが挿入されるときにエラーが発行されるため、不正なデータの入力を防ぐことができます。

まず、「seihin\_kakaku」カラムのルールを実現するバリデーションルールを作成します。

1. [モデル] メニューの [バリデーション ルール] をクリックし、[バリデーション ルール] ダイアログ ボックスを開きます。
2. [新規作成] をクリックし、[新規バリデーション ルール] ダイアログ ボックスを開きます。
3. [論理] ボックスおよび [物理] ボックスに「val\_kakaku」と入力し、[OK] をクリックします。
4. [全般] タブの [最小/最大] オプション ボタンをオンにし、[最大] ボックスに「1000000」と入力します。
5. [ORACLE] タブをクリックして、[ORACLE チェック条件] ボックスに「%AttFieldName <= 1000000」とバリデーション ルールが表示されていることを確認します。[OK] をクリックして [バリデーション ルール] ダイアログ ボックスを閉じます。  
(バリデーション ルールに記述された「%AttFieldName」は ERwin で使用するマクロです。このマクロに関しては 53 ページで説明します。)
6. [SEIHIN] テーブルの [カラム] ダイアログ ボックスを開きます。
7. [カラム] ボックスの一覧の [seihin\_kakaku] カラムをクリックし、[ORACLE] タブをクリックします。
8. [バリデーション] ボックスの一覧から、先ほど作成したバリデーションルール [val\_kakaku] をクリックします。
9. [バリデーション] ボックスの下に、「%AttFieldName <= 1000000」とバリデーション ルールが表示されていることを確認します。[OK] をクリックして [カラム] ダイアログ ボックスを閉じます。







### ERwin マクロ:

ここで作成したバリデーション ルールは、「%AttFieldName <= 1000000」と記述されています。この表現の先頭（「%AttFieldName」）には、「ERwin マクロ」と呼ばれるマクロが使用されています。このマクロは、展開するとマクロを設定したカラムの名前が表示されます。つまり展開後は、「seihin\_kakaku <= 1000000」となります。ERwin には、このようなマクロが多数用意されており、ERwin のさまざまなスクリプト テンプレートに活用されています。これらのマクロは公開されていますので、ユーザー独自のスクリプトを作成することもできます。たとえば、DDL の先頭にスキーマ作成日をコメント行として挿入し、CREATE TABLE 文の頭にテーブル名、定義、カラム一覧などをコメント行として挿入します。このようなテンプレートを作成しておけば、スキーマ定義文の品質は向上します。マクロの使用方法の詳細については、Lesson 15 で説明します。

次に、「chumon\_status」カラムのルールを実現するバリデーション ルールを作成します。

1. [モデル] メニューの [バリデーション ルール] をクリックして [バリデーション ルール] ダイアログ ボックスを開きます。
2. [新規作成] をクリックし、[新規バリデーション ルール] ダイアログ ボックスを開きます。
3. [論理] ボックスおよび [物理] ボックスに「val\_chumon\_status」と入力し、[OK] をクリックします。
4. [全般] タブの [有効値リスト] オプション ボタンをオンにします。
5. [有効値] ボックスに「直販」、[定義] ボックスに「弊社からの直接販売」と入力します。
6. 続けて、[有効値] ボックスに「代理」、[定義] ボックスに「代理店による販売」と入力します。
7. [ORACLE] タブをクリックして、[ORACLE チェック条件] ボックスに「%AttFieldName IN ('直販','代理) 」とバリデーション ルールが表示されていることを確認します。[OK] をクリックして [バリデーション ルール] ダイアログ ボックスを閉じます。
8. [CHUMON] テーブルの [カラム] ダイアログ ボックスを開きます。
9. [カラム] ボックスの一覧の [chumon\_status] カラムをクリックし、[ORACLE] タブをクリックします。
10. [バリデーション] ボックスの一覧から、先ほど作成したバリデーションルール [val\_chumon\_status] をクリックします。
11. [バリデーション] ボックスの下に、「%AttFieldName IN ('直販','代理) 」とバリデーション ルールが表示されていることを確認します。[OK] をクリックして [カラム] ダイアログ ボックスを閉じます。



### [カラム] ダイアログ ボックスから [バリデーション ルール] ダイアログ ボックスを開く:

[カラム] ダイアログ ボックスの [バリデーション] ボックスの右側に [...] ボタンがあります。このボタンをクリックすると、[カラム] ダイアログ ボックスから [バリデーション ルール] ダイアログ ボックスを開くことができます。同様に、[デフォルト] ボックスの右側にある [...] ボタンをクリックすると、[デフォルト値/初期値] ダイアログ ボックスを開くことができます。

☆ **MEMO** ☆

## 物理設計（デフォルト値の設定）

ここでは、データベースに設定するデフォルト値の作成方法とカラムへの設定方法について学習します。

### 説明

デフォルトとは、データ挿入時に明示的にデータが指定されなかった場合に、カラムに挿入するデータを指定する機能をいいます。デフォルトを作成しておくことで、入力するデータが決まっているようなカラムを新規作成するときに、自動的に指定したデータを挿入できます。

### デフォルト値/初期値の作成

以下のカラムに入力されるデータを検討してみると、挿入すべきデフォルト値が存在することがわかりました。

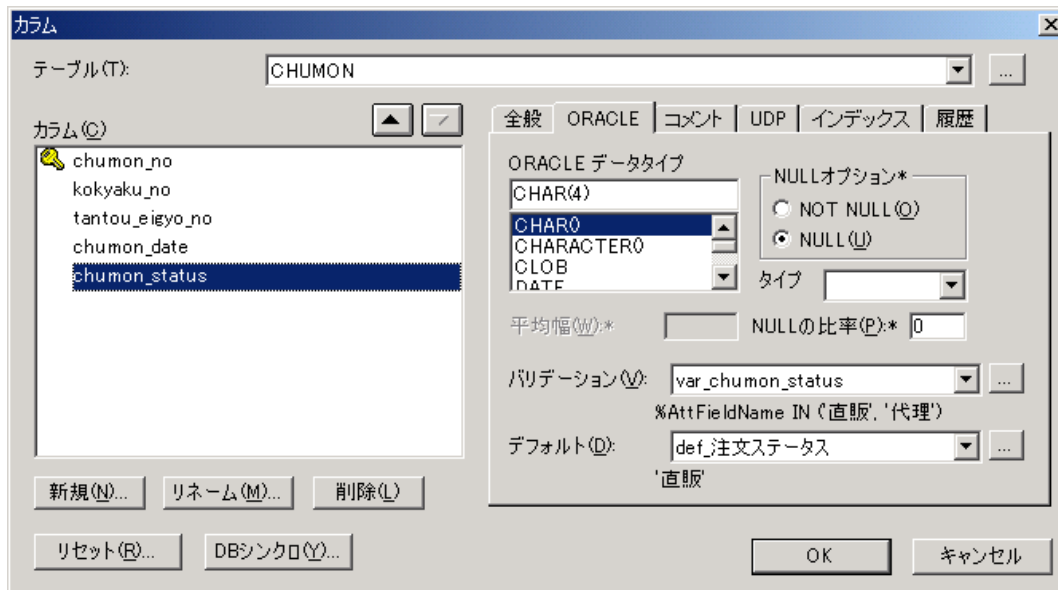
カラム名（属性名）	ビジネスルール	デフォルト値
chumon_status (注文ステータス)	直接販売を基本とする。	直販

注文データを入力する場合、「chumon\_status」カラムにデフォルトで「直販」を挿入します。

「chumon\_status」カラムのデフォルト値を実現するデフォルト/初期値を作成します。

1. [モデル] メニューの [デフォルト値] をクリックして [デフォルト/初期値] ダイアログ ボックスを開きます。
2. [新規] をクリックして [新規デフォルト値] ダイアログ ボックスを開きます。
3. [論理] ボックスおよび [物理] ボックスに「def\_注文ステータス」と入力し、[OK] をクリックします。
4. [ORACLE] タブをクリックし、[サーバー値 - ORACLE デフォルト] ボックスに「直販」と入力します。
5. [OK] をクリックして、[デフォルト値/初期値] ダイアログ ボックスを閉じます。
6. [CHUMON] テーブルの [カラム] ダイアログ ボックスを開きます。
7. [カラム] ボックスの一覧の [chumon\_status] カラムをクリックし、[ORACLE] タブをクリックします。
8. [デフォルト] ボックスの一覧から、先ほど作成したデフォルト [def\_注文ステータス] をクリックします。

9. [デフォルト] ボックスの下に、「直販」とデフォルト/初期値が表示されていることを確認します。[OK] をクリックして [カラム] ダイアログ ボックスを閉じます。



#### デフォルト/初期値:

ERwin は、[デフォルト/初期値] ダイアログ ボックスの [サーバー値] ボックスに入力した値をそのままデフォルト値に使用して DDL を生成します。デフォルト値に数値型データを使用するときは問題ありませんが、文字型データを使用する場合はデフォルト値を「」（シングル クォーテーション）で囲んでください。データベースによっては、引用符で囲まれていない文字型データを挿入できないことがあります。

## ドメインの階層構造

ここでは、ERwin の優れたデータベース設計機能の 1 つである、ドメインの階層構造について学習します

### 説明

カラムに設定するすべてのプロパティが共通している場合、簡単にドメインを検討できます。しかし実際には、あるプロパティは共通しているが、共通していないプロパティも存在するというケースが多く発生します。このような場合、ドメインを階層化する方法があります。



このチュートリアルでは、ドメインについて理解しやすいように「ドメインの作成」(Lesson5)と「ドメインの階層構造」に分けて説明していますが、通常はドメインを検討する時に階層構造を決定することが一般的です。

### ドメインに階層構造を持たせる

たとえば、部内で使用するコードはすべて数値型で登録するというルールがあるとします。コードを登録するカラムには、「製品コード」、「従業員コード」、「所属支店コード」、および「所属部門コード」があります。以下の表を参照し、それぞれのカラム プロパティを確認してください。

カラム名	データ型	長さ	NULL	バリデーション ルール	デフォルト値
製品コード	NUMBER	10	NOT NULL	—	—
従業員コード	NUMBER	5	NOT NULL	—	—
所属部署コード	NUMBER	3	NOT NULL	101, 102, 103	—
所属支店コード	NUMBER	3	NOT NULL	501, 502	501

これらのカラムはデータ型に NUMBER 型を使用することは共通していますが、他のプロパティは異なります。たとえば、「所属部署コード」カラムには、「101」（営業部）、「102」（技術部）、および「103」（総務部）だけ入力可能というバリデーション ルールが存在します。また、「所属支店コード」カラムには、「501」（東京本社）と「502」（大阪支店）だけ入力可能というバリデーション ルールと、新入社員は必ず「501」（東京本社）に配属されるというデフォルト値が存在します。

まず、すべてのカラムに共通するプロパティを登録します。データ型が「NUMBER」、NULL オプションが「NOT NULL」であるという情報だけ入力します。ここでは、「コード」ドメインという名前で登録します。

```
コード - - NUMBER, NOT NULL
```

次に、データ型「NUMBER」の長さを指定します。この場合、長さを指定してもデータが「NUMBER」であるということは何も変わりません。そこで、「コード」ドメインのプロパティを継承して長さだけ個別に指定した子ドメインを作成します。長さの種類としては3種類あります。「製品コード」の10バイト、「従業員コード」の5バイト、および「所属部署コード」と「所属支店コード」の3バイトです。長さ3バイトの「所属部署コード」と「所属支店コード」は、「所属コード」としてまとめます。

```
コード - - NUMBER, NOT NULL
├── 製品コード - - NUMBER(10), NOT NULL
├── 従業員コード - - NUMBER(5), NOT NULL
└── 所属コード - - NUMBER(3), NOT NULL
```

さらに、「所属部署コード」と「所属支店コード」について考えてみます。データ型が「NUMBER (3)」というプロパティは「所属コード」ドメインから継承し、バリデーションルールとデフォルトの設定を個別に指定したドメインを作成します。つまり、「所属コード」ドメインのプロパティを継承した子ドメインとして、「所属部署コード」ドメインと「所属支店コード」ドメインを作成します。

```
コード - - NUMBER, NOT NULL
├── 製品コード - - NUMBER(10), NOT NULL
├── 従業員コード - - NUMBER(5), NOT NULL
└── 所属コード - - NUMBER(3), NOT NULL
    ├── 所属部署コード - - NUMBER(3), NOT NULL, バリデーションルール(101, 102, 103)
    └── 所属支店コード - - NUMBER(3), NOT NULL, バリデーションルール(501, 502)
        デフォルト値(501)
```

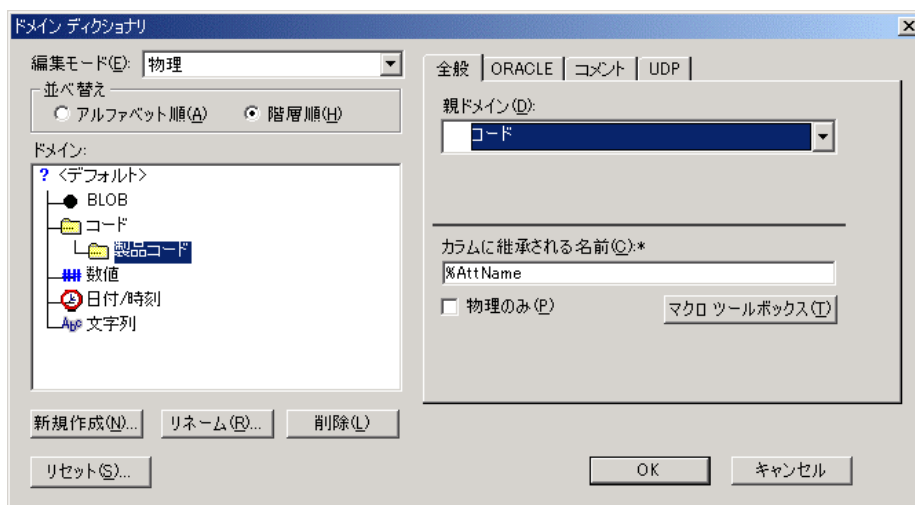
## 階層構造を持つドメインの作成

ここでは、ドメインの階層化を学習するために、新しいダイアグラムを作成して階層構造を持つドメインを作成してみます。

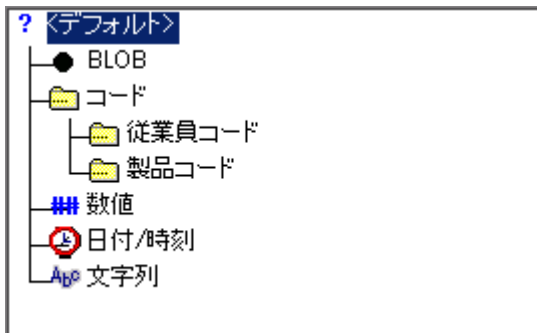
1. [ファイル] メニューの [新規作成] をクリックし、練習用の [論理/物理] ダイアグラムを作成します。対象データベースとして、「ORACLE 9.x」を選択してください。
2. 論理モデルでは NULL オプションの設定ができないため、物理モデルに切り替えます。
3. モデル エクスプローラの [モデル] タブの [ドメイン] を展開し、[<デフォルト>] を右クリックし、ショートカット メニューの [新規] をクリックして「新規\_ドメイン」を作成します。[新規\_ドメイン] がアクティブな状態になっていますので、名前を「コード」に変更します。



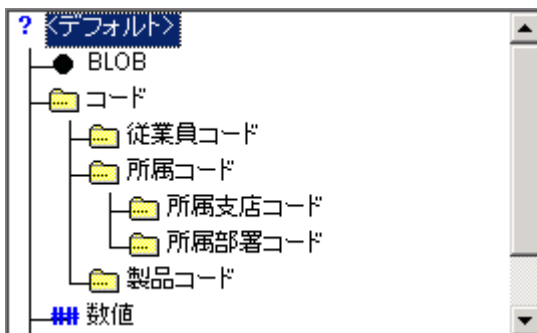
4. 同様の操作で、「製品コード」ドメインを作成します。
5. モデル エクスプローラで新しく作成したドメインの [コード] をダブルクリックして [ドメインディクショナリ] ダイアログボックスを開きます。[ORACLE] タブをクリックして、[ORACLE データタイプ] ボックスの一覧の [NUMBER0] をクリックします。[NULL オプション] の [NOT NULL] をクリックします。
6. 「コード」ドメインを「製品コード」ドメインの親ドメインにして階層構造を設定します。[ドメイン] ボックスで [製品コード] をクリックし、[全般] タブをクリックします。
7. [親ドメイン] ボックスの一覧の [コード] をクリックします。これにより、ドメインの階層構造が設定されました。
8. [ドメイン] ボックスに [コード] ドメインと [製品コード] ドメインが階層構造で表示されていることを確認します（階層構造になっていない場合は、[並べ替え] の [階層順] をクリックしてください）。



9. [ドメイン] ボックスで [製品コード] ドメインをクリックし、[ORACLE] タブをクリックします。親ドメインの情報が引き継がれていることを確認します。データ型の「NUMBER0」に桁数「10」を入力します。また、「NOT NULL」が選択されていることを確認します。
10. 次に、「従業員コード」ドメインを作成します。[ドメイン ディクショナリ] ダイアログ ボックスの [ドメイン] ボックスで [コード] ドメインを選択した状態で [新規作成] をクリックして、[新規ドメイン] ダイアログ ボックスを開きます。[論理名] ボックスに「従業員コード」と入力し、[OK] をクリックして [新規ドメイン] ダイアログ ボックスを閉じます。[ドメイン] ボックスに [コード] ドメインと [従業員コード] ドメインが階層構造で表示されていることを確認してください。



11. [ORACLE] タブをクリックし、データ型の「NUMBER0」に桁数「5」を入力します。
12. 同様の操作でドメインの継承階層を作成します。「コード」ドメインの子ドメインとしてデータ型に「NUMBER(3)」を指定した「所属コード」ドメインを作成します。
13. 「所属コード」ドメインの子ドメインとして、バリデーション ルールやデフォルトを指定した「所属部署コード」ドメインと「所属支店コード」ドメインを作成します。



14. [ドメイン ディクショナリ] ダイアログ ボックスの [ドメイン] ボックスで [所属部署コード] ドメインをクリックし、[ORACLE] タブをクリックします。[バリデーション] ボックスの右の [...] をクリックして [バリデーション ルール] ダイアログ ボックスを開きます。次に、「101」（営業部）、「102」（技術部）、および「103」（総務部）の有効値/定義を持つ「val\_所属部署コード」バリデーション ルールを作成します（バリデーション ルールの作成方法の詳細については、Lesson11を参照してください）。[OK] をクリックして [バリデーション ルール] ダイアログ ボックスを閉じます。
15. 同様の操作で、「所属支店コード」ドメインに「501」（東京本社）および「502」（大阪支店）の有効値/定義を持つ「val\_所属支店コード」バリデーション ルールを作成します。また、デフォルト「501」を持つ「def\_所属支店コード」デフォルト値を作成します（デフォルトの作成方法については、Lesson12を参照してください）。[OK] をクリックして [デフォルト値/初期値] ダイアログ ボックスを閉じます。



16. [ドメイン ディクショナリ] ダイアログ ボックスの [ドメイン] ボックスで、作成したドメインの階層構造を確認してください。

このダイアグラムは後の Lesson では使用しないので、保存せずに閉じてください。



---

**コラム プロパティの移行:**

主キーにドメインを設定している場合は、外部キー カラムのドメインに注意してください。ERwin では、親から子へとプロパティが移行されます。主キー カラムにドメインを設定すると、ドメイン設定はリレーションシップで結ばれたすべての外部キー カラムに移行されます。ERwin では、外部キー カラムに親キー カラムとは別のドメインを設定できますので、設定するドメインを検討してください。



---

**子ドメイン エディタに表示される「\*」（アスタリスク）について:**

[ドメインディクショナリ] ダイアログ ボックス上では、階層化された子ドメインの各プロパティ名の後に、「\*」（アスタリスク）記号が表示されます。このアスタリスクは、親ドメインのカラム プロパティをそのまま使用している場合だけ表示されます。ERwin の子ドメインで、親ドメインから継承したプロパティを上書きすると、「\*」は表示されなくなります。

☆ **MEMO** ☆

## 参照整合性制約の設定

ここでは、ERwin を使用して参照整合性制約を設定する方法について学習します。  
ERwin は、外部キー制約とトリガを使用してデータ整合性を実現します。

### 説明

データ整合性とは、設計者やデータベース管理者が事前に定義した規則に合ったデータのみがデータベースに格納されることであり、データの不正な削除、挿入、更新などを禁止します。たとえば Oracle を使用してデータ整合性を実現するには、宣言型（エンティティ整合性制約および参照整合性制約）と手続き型（データベース トリガ）の 2 種類の手段があります。

ここで学習する参照整合性制約（宣言型）とは、テーブル間の主キーや外部キーの整合性を保証するものです。参照整合性制約で実現できない問題は、データベース トリガ（手続き型）を使用して実現します。

### データベースがサポートする参照整合性を利用する

データベースの仕様によってサポートする参照整合性制約は異なりますが、たとえば Oracle の場合は外部キー制約を使用して、以下の 4 種類の参照整合性制約を実現しています。

データへの操作	参照整合性制約の動作	表現
親の主キーのデータの削除…	…を禁止します。	DELETE RESTRICT
	…を依存関係にある子の外部キーの削除へとカスケード（連鎖）します。	DELETE CASCADE
	…が行われると、依存関係にある子の外部キーの値が NULL 値に変換されます。	DELETE SET NULL
親の主キーのデータの更新…	…を禁止します。	UPDATE RESTRICT

## ERwin がサポートするデータ整合性を利用する

ERwin は、データ整合性の実現のためにデータベースの参照整合性制約やデータベース トリガを使用します。

以下に ERwin がサポートするデータ整合性を記します。これらを組み合わせることで、36 種類が利用できます。

データの操作対象 (表現)	データの操作 (表現)	参照整合性制約の動作 (表現)
<ul style="list-style-type: none"> <li>親の主キーへの操作 (PARENT)</li> <li>子の外部キーへの操作 (CHILD)</li> </ul>	<ul style="list-style-type: none"> <li>削除 (DELETE)</li> <li>挿入 (INSERT)</li> <li>更新 (UPDATE)</li> </ul>	<ul style="list-style-type: none"> <li>禁止する。(RESTRICT)</li> <li>関連データに操作をカスケード (連鎖) する。(CASCADE)</li> <li>関連データを NULL にする。(SET NULL)</li> <li>関連データをデフォルト値にする。(SET DEFAULT)</li> <li>何もしないことを明示的に宣言する。(NO ACTION)</li> <li>何もしない。(NONE)</li> </ul>
2 種類	×	3 種類
	×	6 種類
= 36 種類		



### ERwin の優れたデータ整合性実現機能:

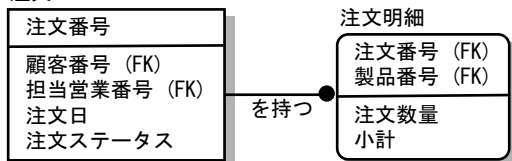
参照整合性制約で実現できる整合性は、データベース トリガで実現できる整合性に比べると適用範囲が狭く (Oracle は 4 種類)、柔軟性に欠けます。ERwin は、必要とされるすべての参照整合ルールを実現するためにデータベース トリガをテンプレート (ERwin マクロ) として用意しています。また、この ERwin マクロはカスタマイズしたり、新規作成したりすることもできます。詳細については、Lesson15 の「ERwin マクロを利用したスキーマ生成」を参照してください。

## 参照関係を明確にする

親子間のデータに関連を持たせるということは、親 (または子) データへの処理が子 (または親) データにも影響するという事です。親子テーブル間のデータに整合性を持たせるには、テーブル間のビジネス ルールを把握する必要があります。つまり、親の主キー (参照されるデータ) を操作した場合、子の外部キー (親に依存するデータ) をどのように操作するか、また子の外部キーが存在する場合どのように親の主キーを扱うかについて検討します。

まずは、[CHUMON] テーブルと [CHUMON\_MEISAI] テーブルの間のビジネス ルールを考えてみましょう。親である [CHUMON] テーブルのデータに削除、挿入、更新などの変更が発生したとき、子である [CHUMON\_MEISAI] テーブルのデータをどのように扱うべきなのでしょうか？

注文



親子間の関係を理解しやすくするために、左の図の ER ダイアグラムでは、日本語オブジェクト名を使用した論理モデル表示を使用しています。

[CHUMON] テーブルへの操作	[CHUMON_MEISAI] テーブルへの操作
<b>削除:</b> 注文がキャンセルになったので注文情報を削除する。	注文がキャンセルになったときは、その注文で受注した製品情報などの <b>注文明細情報も合わせて削除したい</b> 。
<b>挿入:</b> 新しい受注データを登録する。	新規受注が発生したときは、まず注文を受けた担当営業や注文日などの情報を入力し、その後で注文の明細情報を入力する。
<b>更新:</b> 注文番号に変更が発生し、注文番号を変更したい。	一度、注文明細に登録したデータの <b>注文番号の変更は許さない</b> 。仮に注文番号を変更する必要があるときは、注文情報を一度削除してから新しく登録し直す必要がある。

▲ [CHUMON] テーブルと [CHUMON\_MEISAI] テーブルの間のビジネス ルール

では、それぞれのケースでビジネス ルールを実現するためには、どのようなデータ整合性が必要になるか考えてみましょう。

[CHUMON] テーブルへの操作	ビジネス ルールをどのように実現するか	ERwin での表現方法
削除	[CHUMON] テーブルの「chumon_no」カラムのデータを削除するときは、それを参照する [CHUMON_MEISAI] テーブルの関連データもすべて削除する。つまり親データの削除を子の依存データの削除へとカスケード（連鎖）する。これにより、注文情報の元になる注文番号が削除された後に、不正な注文明細データが存在することを防ぐ。	PARENT DELETE CASCADE
挿入	[CHUMON] テーブルに受注データを新規登録する操作は、[CHUMON_MEISAI] テーブルへ何の影響も及ぼさない。	PARENT INSERT NONE、PARENT INSERT NO ACTION
更新	[CHUMON_MEISAI] テーブルのデータが [CHUMON] テーブルのデータを参照している場合は、[CHUMON_MEISAI] テーブルの依存データが削除されるまで、元となる親の参照データを更新できない。	PARENT UPDATE RESTRICT

上で解析したデータ整合性を実現するためには、Oracle のどの整合性ルールを使用できるでしょうか？

[CHUMON] テーブルへの操作	ERwin での表現方法	Oracle での実現方法
削除	PARENT DELETE CASCADE	外部キー制約の ON DELETE CASCADE により実現可能
挿入	PARENT INSERT NONE	何もしなくてよい
更新	PARENT UPDATE RESTRICT	外部キー制約の ON UPDATE RESTRICT により実現可能

## 参照整合性の設定

[CHUMON] テーブルと [CHUMON\_MEISAI] テーブルの間のビジネス ルールを実現するために、テーブル間を結ぶリレーションシップに参照整合性ルールを設定します。

1. [CHUMON] テーブルと [CHUMON\_MEISAI] テーブル間の [motsu] リレーションシップを右クリックし、ショートカット メニューの [リレーションシップ プロパティ] をクリックして [リレーションシップ] ダイアログ ボックスを開きます。
2. まず、親である [CHUMON] テーブルの参照整合性ルールを設定します。 [RI アクション] タブをクリックし、 [PARENT DELETE] ボックスの一覧の [CASCADE] をクリックします。これにより、 [CHUMON] テーブルの [chumon\_no] カラムのデータを削除すると、子の [CHUMON\_MEISAI] テーブルに存在する、 [CHUMON] テーブルの [chumon\_no] カラムに依存するすべてのデータが削除されます。
3. [PARENT INSERT] ボックスの一覧の [NONE] をクリックします。 [NONE] を指定した場合、特に整合性を保つためのスキーマは作成されません。
4. [PARENT UPDATE] ボックスの一覧の [RESTRICT] をクリックします。これにより、 [CHUMON] テーブルの [chumon\_no] カラムのデータは、それを参照する子の [CHUMON\_MEISAI] テーブルの [chumon\_no] カラムのデータが存在する限り更新できません。
5. その他の参照整合性については、すべて [NONE] を選択し、 [OK] をクリックして [リレーションシップ] ダイアログ ボックスを閉じます。



以下の表に従い、同様の操作で他のリレーションシップについても参照整合性を設定してください。

[JUGYOIN] テーブルと [CHUMON] テーブル間の、〈受ける〉(物理名<ukeru>) リレーションシップ

トリガ アクション タイプ	参照整合オプション	Oracle での実現方法
PARENT DELETE	RESTRICT	外部キー制約の ON DELETE RESTRICT により実現可能
PARENT INSERT	NONE	何もしなくてよい
PARENT UPDATE	RESTRICT	外部キー制約の ON UPDATE RESTRICT により実現可能
CHILD DELETE	NONE	何もしなくてよい
CHILD INSERT	NONE	何もしなくてよい
CHILD UPDATE	NONE	何もしなくてよい

[KOKYAKU] テーブルと [CHUMON] テーブル間の、〈を出す〉(物理名<dasu>) リレーションシップ

トリガ アクション タイプ	参照整合オプション	Oracle での実現方法
PARENT DELETE	RESTRICT	外部キー制約の ON DELETE RESTRICT により実現可能
PARENT INSERT	NONE	何もしなくてよい
PARENT UPDATE	RESTRICT	外部キー制約の ON UPDATE RESTRICT により実現可能
CHILD DELETE	NONE	何もしなくてよい
CHILD INSERT	NONE	何もしなくてよい
CHILD UPDATE	NONE	何もしなくてよい

[SEIHIN] テーブルと [CHUMON\_MEISAI] テーブル間の、〈に必要である〉(物理名<hitsuyou>) リレーションシップ

トリガ アクション タイプ	参照整合オプション	Oracle での実現方法
PARENT DELETE	RESTRICT	外部キー制約の ON DELETE RESTRICT により実現可能
PARENT INSERT	NONE	何もしなくてよい
PARENT UPDATE	CASCADE	データベース トリガにより実現可能
CHILD DELETE	NONE	何もしなくてよい
CHILD INSERT	NONE	何もしなくてよい
CHILD UPDATE	NONE	何もしなくてよい



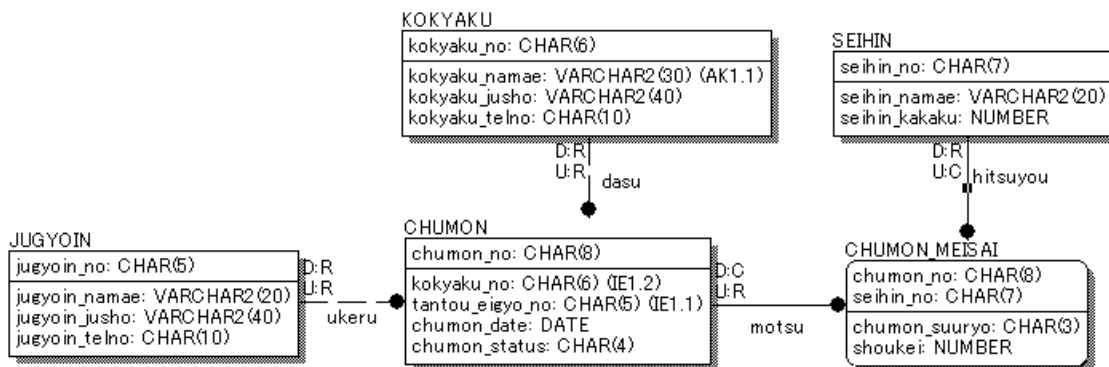
#### 参照整合性制約とデータベース トリガ:

参照整合性制約とデータベース トリガは、どちらもデータの整合性を保つために重要な機能です。ただし、不用意に生成したデータ整合性機能が互いに矛盾してしまい、データ整合性が保てないこともありますので、その使用には十分な検討が必要になります。

## 参照整合設定の表示方法

ダイアグラムの空白部分（何もない部分）を右クリックして、ショートカットメニューの「リレーションシップの表示」をポイントし、「参照整合性」をクリックします。これにより、リレーションシップのライン上に参照整合性設定が表示されます。

ここまでの Lesson で以下のような ER ダイアグラムが作成されたことを確認したら、ファイルを上書き保存してください。



リレーションシップの参照性合成設定を表示すると、リレーションシップの親側または子側に「X:Y」という文字が表示されます。2つの文字がコロン(:)で区切られて表示されますが、初めの文字(X)はデータに対する操作を表します。後ろの文字(Y)は選択した整合性オプションを表します。

データに対する操作	
<b>D</b>	削除 (DELETE)
<b>I</b>	挿入 (INSERT)
<b>U</b>	更新 (UPDATE)

整合性オプション	
<b>R</b>	RESTRICT
<b>C</b>	CASCADE
<b>SN</b>	SET NULL
<b>SD</b>	SET DEFAULT
<b>NOA</b>	NO ACTION
なし	NONE



## ERwin マクロを利用したスキーマ生成

ここでは、ERwin マクロを使用してさまざまな情報を追加したスキーマを作成する方法について学習します。

### 説明

ERwin マクロを使用すると、データベース トリガ、ストアド プロシージャ、プリ/ポスト スクリプトなどを生成するテンプレートを作成できます。また、ERwin マクロを活用すれば、ERwin が生成するスキーマにさまざまな付加情報を追加してより充実した DDL を作成できます。

### ERwin マクロの使用例

ERwin のどのようなオブジェクトで ERwin マクロが使用されているかをご紹介します。

#### トリガ テンプレート

ERwin には、参照整合性を実現するために非常に多くのトリガ テンプレートが用意されています。これらはすべて ERwin マクロで作成されています。テンプレートに ERwin マクロを使用することにより、1 つのトリガ テンプレートをさまざまな組み合わせの親子関係で使用できます。たとえば、参照関係の親テーブルを指定するマクロ (%Parent) は、どのような親子関係においても必ず親テーブル名を展開します。

### ERwin マクロの活用例

ERwin は生成するスキーマに、ユーザーが作成したスクリプトを追加できます。このスクリプトを活用すると、生成する DDL をより充実させることができます。

#### スキーマ プリ/ポスト スクリプトの活用例

スキーマ プリ/ポスト スクリプトは、ERwin が生成するスキーマの最初/最後に生成するスクリプトのことです。たとえば、スキーマの最初には、自分の名前と生成するスキーマの元となった ER1 ファイルの名前とスキーマの作成日を、そして最後には対象データベースの名前と接続ユーザー名をコメントとして記述します。

#### テーブル プリ/ポスト スクリプトの活用例

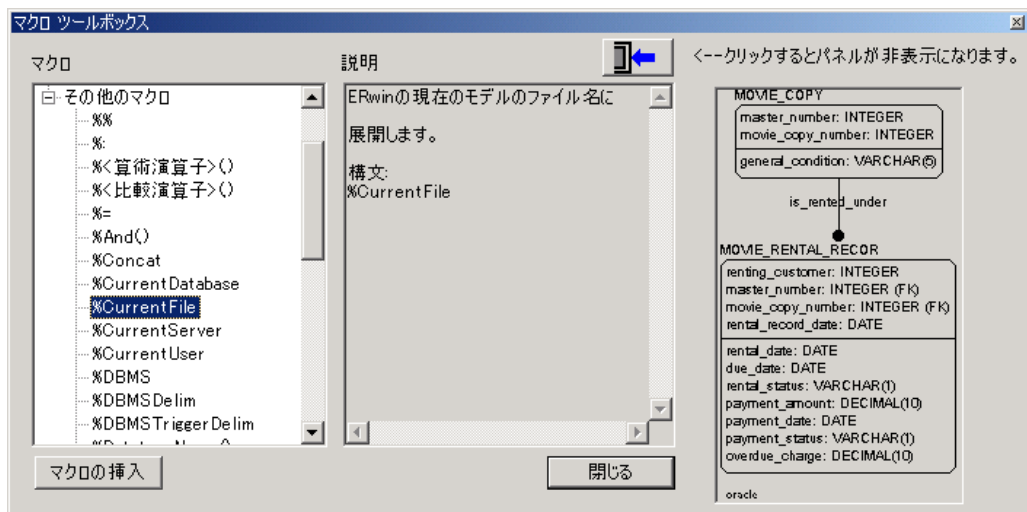
テーブル プリ/ポスト スクリプトは、ERwin が生成する CREATE TABLE 文の前/後に生成するスクリプトのことです。たとえば、CREATE TABLE 文の前には、生成するテーブルの名前、およびカラム名とデータ型の一覧を記述します。

## ERwin マクロを使ったスクリプトの作成

前述の ERwin マクロの活用例に従って、プリ/ポスト スクリプト テンプレートを作成します。まずは、スキーマ プリ/ポスト スクリプト テンプレートを作成します。

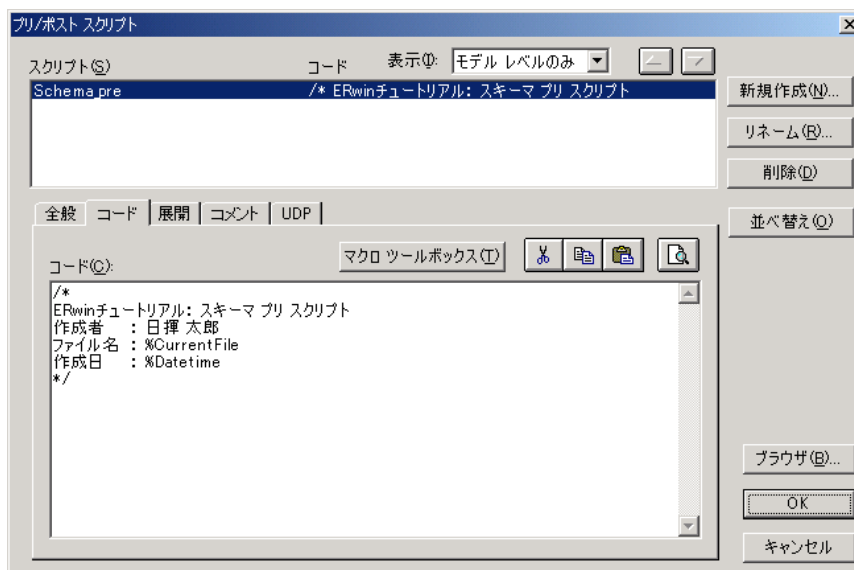
### スキーマ プリ/ポスト スクリプト テンプレートの作成

1. [データベース] メニューの [プリ/ポスト スクリプト] をポイントし、[モデル レベル] をクリックして [プリ/ポスト スクリプト] ダイアログ ボックスを開きます。
2. [新規作成] をクリックして、[新規スクリプト] ダイアログ ボックスの [名前] ボックスに「Schema\_pre」と入力します。[OK] をクリックして [新規スクリプト] ダイアログ ボックスを閉じ、[プリ/ポスト スクリプト] ダイアログ ボックスに戻ります。
3. [全般] タブをクリックし、[生成オプション] グループの [前に作成] をクリックします。
4. [コード] タブをクリックします。それでは、[コード] ボックスにスクリプト テンプレートを入力していきます。
5. まず、コメントの始まりを表す「/\*」を入力します（「\*/」でコメントを閉じるまでのスクリプトは、データベースへのスキーマ生成時にコメントとして扱われます）。
6. 改行後、「ERwin チュートリアル: スキーマ プリ スクリプト」と入力します。
7. 改行後、「作成者 : XXX (あなたの名前)」と入力します。
8. 改行後、「ファイル名:」と入力したら、「:」の後ろにカーソルを合わせた状態で [マクロ ツールボックス] をクリックして [マクロ ツールボックス] ダイアログ ボックスを開きます。[コード] ボックスの入力内容が見える場所に [マクロ ツールボックス] ダイアログ ボックスを移動してください。
  - i. [マクロ ツールボックス] ダイアログ ボックスの [マクロ] ボックスの [その他のマクロ] を展開し、[%CurrentFile] マクロをクリックします。これにより、ダイアログ ボックス中央の [説明] ボックスに、シンタックス例や展開後のサンプル コードなどの参考情報が表示されます。
  - ii. 使用すべき ERwin マクロが決まったら、[マクロの挿入] をクリックします。これにより、[プリ/ポスト スクリプト] ダイアログ ボックスの [コード] ボックスのカーソルがあった位置、すなわち「ファイル名:」の後に ERwin マクロがコピーされます。
  - iii. [コード] ボックスに [%CurrentFile] マクロがコピーされたことを確認したら、[閉じる] をクリックして [マクロ ツールボックス] ダイアログ ボックスを閉じます。



▲ [マクロ ツールボックス] ダイアログ ボックスでは、すべての ERwin マクロを選択でき、任意の ERwin マクロを簡単に [コード] ボックスにコピーできます。また、表示されるシンタックス例や展開コード サンプルなどを活用して、ユーザー テンプレートを作成してください。

9. 改行後、「作成日 : %Datetime」と入力します。この「%Datetime」も、[マクロ ツールボックス] ダイアログ ボックスの [マクロ] ボックスから、[その他のマクロ] を展開して挿入します。
10. 最後に、コメントの終わりを表す「\*/」を入力し、体裁を整えます。

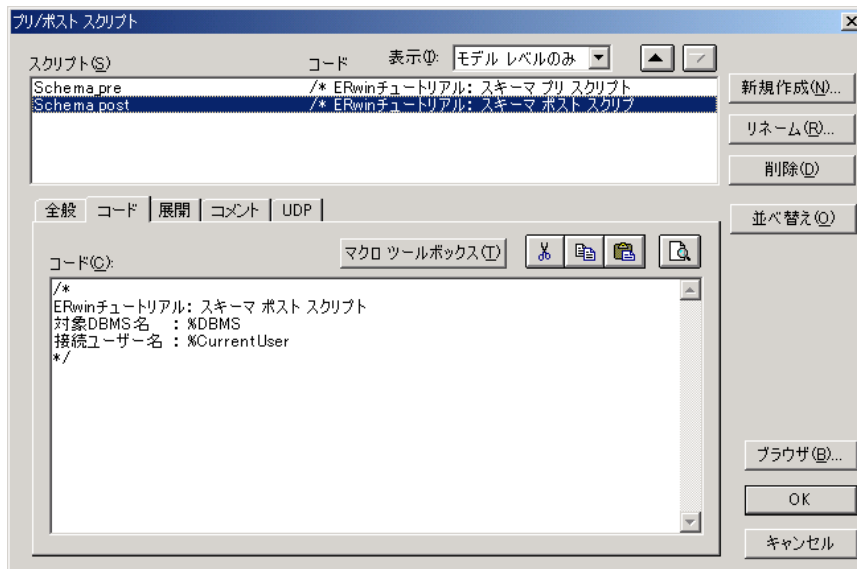


同様の操作で、以下のスキーマ ポスト スクリプト テンプレートを作成してください。

スキーマ スクリプト テンプレート名 : Schema\_post  
 生成オプション : 後で作成

入力するスクリプト	ERwin マクロの意味 / [マクロ ツールボックス] ダイアログ ボックスでの存在場所
<pre> /* ERwin チュートリアル : スキーマ ポスト スクリプト 対象 DBMS 名      : %DBMS 接続ユーザー名   : %CurrentUser */                     </pre>	コメント開始 / - コメント / - 選択した DBMS 名 / その他のマクロ 接続したユーザー名 / その他のマクロ コメント終了 / -

「%CurrentUser」マクロは、データベースとの接続時に [接続] ダイアログ ボックスの [ユーザー名] ボックスに入力したユーザー名を展開します。 [ユーザー名] ボックスに何も入力されていない場合は、空欄となります。また、データベースに接続していない場合も、空欄となります。



**スキーマ スクリプトを実際に生成するには**



スキーマ スクリプトは作成しただけでは、実際に生成されません。実際に生成するには、スキーマ生成時に [ORACLE スキーマ生成] ダイアログ ボックスで生成する項目として、それぞれ [プリスクリプト] チェック ボックスと [ポストスクリプト] チェック ボックスをオンにする必要があります。

スキーマ スクリプトの展開例

Schema_pre	Schema_post
<pre> /* ERwin チュートリアル : スキーマ プリ スクリプト 作成者      : 日揮 太郎 ファイル名  : C:\ERWIN\MYMODEL.ER1 作成日      : Thu May 06 18:21:09 2004 */                     </pre>	<pre> /* ERwin チュートリアル : スキーマ ポスト スクリプト 対象 DBMS 名      : ORACLE 接続ユーザー名   : scott */                     </pre>

## テーブル プリ/ポスト スクリプト テンプレートの作成

前述の ERwin マクロ活用例に従って、次にテーブル プリ/ポスト スクリプト テンプレートを作成します。

1. ダイアグラム上で何も選択していない状態で、[データベース] メニューの [プリ/ポスト スクリプト] をポイントし、[テーブル レベル] をクリックして [プリ/ポスト スクリプト] ダイアログ ボックスを開きます。
2. [新規作成] をクリックして、[新規スクリプト] ダイアログ ボックスの [名前] ボックスに「Table\_pre」と入力します。[OK] をクリックして [新規スクリプト] ダイアログ ボックスを閉じ、[プリ/ポスト スクリプト] ダイアログ ボックスに戻ります。
3. [全般] タブをクリックし、[スクリプト タイプ] グループの [テーブル レベル] がクリックされていることを確認します。
4. [生成オプション] グループの [前に作成] をクリックします。
5. 後は、先ほどスキーマ プリ スクリプト テンプレートを作成したのと同様の操作で作成します。以下のテーブル プリ スクリプト テンプレートを作成してください。

テーブル スクリプト テンプレート名 : Table\_pre  
生成オプション : 前に作成

入力するスクリプト	ERwin マクロの意味 / [マクロツールボックス] ダイアログ ボックスでの存在場所
<pre> /* //////////////////////////////////// テーブル名 : %TableName ----- カラム リスト %ForEachColumn() { %AttFieldName - %ColumnDatatype } //////////////////////////////////// */ </pre>	<p>コメント開始 / -</p> <p>コメント / -</p> <p>テーブル名 / エンティティ マクロ</p> <p>コメント / -</p> <p>コメント / -</p> <p>当該テーブルの各カラムのマクロ コードを展開する / エンティティ マクロ</p> <p>カラム名 - データ型 / 属性マクロ</p> <p>コメント / -</p> <p>コメント終了 / -</p>

6. [コード] タブの [コード] ボックスへの入力が終了したら、[ブラウザ] をクリックして [スクリプト ブラウザ] ダイアログ ボックスを開きます。[新規テーブルにアタッチ] チェック ボックスをオンにします。これをオンにすると、新規作成したテーブルには自動的にこのプリ スクリプト テンプレートが設定されます。

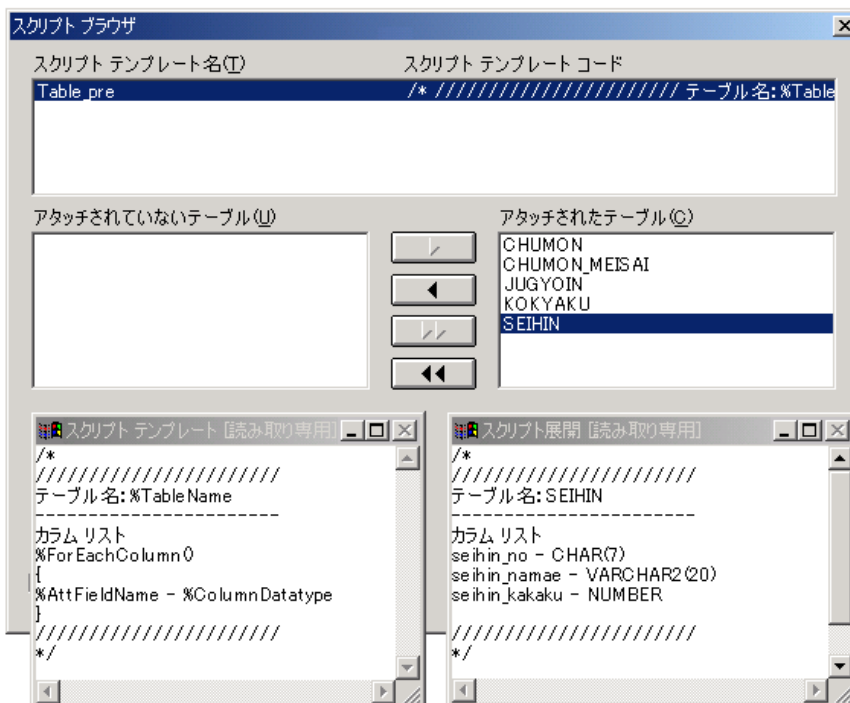


### テーブル スクリプトを実際に生成するには

テーブル スクリプト テンプレートもスキーマ スクリプト テンプレートと同様に作成しただけでは、実際に生成されません。実際に生成するには、以下の2つの手順が必要になります。

- ① [スクリプト ブラウザ] ダイアログ ボックスでスクリプトをテーブルに設定する。
- ② スキーマ生成時に [ORACLE スキーマ生成] ダイアログ ボックスで生成する項目として、[プリスクリプト] チェック ボックスと [ポストスクリプト] チェック ボックスをオンにする。

7. [スクリプト ブラウザ] ダイアログ ボックスを使用すると、1つのテーブル スクリプト テンプレートを一度に複数のテーブルに設定/設定の解除の指定ができます。先ほど作成したテーブル スクリプト テンプレートがダイアログ ボックス上部のボックスに表示されています。[アタッチされたテーブル] ボックスには何も表示されていないので、「Table\_pre」スクリプトはまだテーブルにアタッチされていません。[アタッチされていないテーブル] ボックスで、テーブルを1つクリックし、**▶▶**をクリックして [アタッチされたテーブル] ボックスにすべてのテーブルを移動します。すべてのテーブルに「Table\_pre」スクリプトを設定すれば、各テーブルの CREATE TABLE 文の前にそのテーブルを説明するコメント スクリプトが生成されます。
8. ダイアログ ボックス下段の [スクリプト展開] ウィンドウで作成したスクリプトを確認します。[スクリプト展開] ウィンドウには、ERwin マクロが展開され、テーブル名、カラム名、データ型などの具体的な値が入ったスクリプトが表示されます。また、左側の [スクリプト テンプレート] ウィンドウには、作成したスクリプトが表示されます。これらのウィンドウはサイズを変更できるので、展開されたスクリプトが見えるようにサイズを変えてみてください。確認したら [閉じる] をクリックして [スクリプト ブラウザ] ダイアログ ボックスを閉じます。



9. [OK] をクリックして [プリ/ポスト スクリプト] ダイアログ ボックスを閉じます。

## トリガ テンプレートの作成

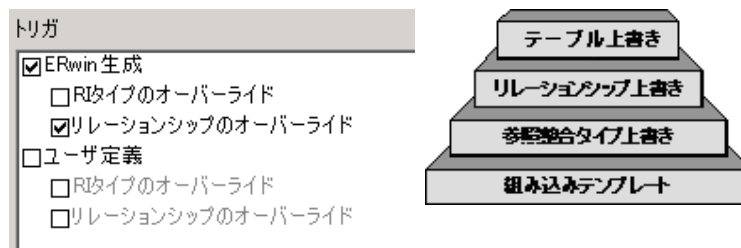
ERwin マクロの使用方法与 [マクロ ツールボックス] ダイアログ ボックスに慣れてきたところで、今度はトリガ テンプレートの作成方法について学習します。このチュートリアルでは、デフォルトのトリガ テンプレートにコメント行を追加するだけですが、トリガ テンプレートの作成/更新方法を学習すれば、ERwin マクロを使用してトリガ テンプレートをカスタマイズしたり、新規作成したりできます。

### ERwin が作成できるトリガ テンプレートの種類

ERwin は、すべての参照整合条件において以下の 4 種類のトリガ テンプレートを持っています。

トリガ タイプ	テンプレートを作成エディタ名
組み込みテンプレート (デフォルト)	編集不可
参照整合タイプ上書き	[トリガ テンプレート] ダイアログ ボックス
リレーションシップ上書き	[リレーションシップ テンプレート] ダイアログ ボックス
テーブル上書き	[トリガ] ダイアログ ボックス

ERwin は、特定の参照整合条件において生成するトリガ テンプレートをそれぞれのダイアログ ボックスで作成および設定できます。複数のダイアログ ボックスで複数個作成したトリガ テンプレートから、実際にどのトリガ テンプレートを生成するかを指定するには、[スキーマ生成] ダイアログ ボックスのチェック ボックスで切り替えます。このチェック ボックスは複数選択できるようになっていますので、各オプションには優先順位がついています。下の図は、どのトリガ テンプレートが優先されるかを示しています。それぞれの上書きレベルは、それより下のレベルで設定されているテンプレートより優先されます。つまり、下図左側のようにトリガ オプションを指定すると、組み込みテンプレート (デフォルトのトリガ テンプレート) に優先して、リレーションシップ上書き ([リレーションシップ テンプレート] ダイアログ ボックスで作成したトリガ) が生成されます。

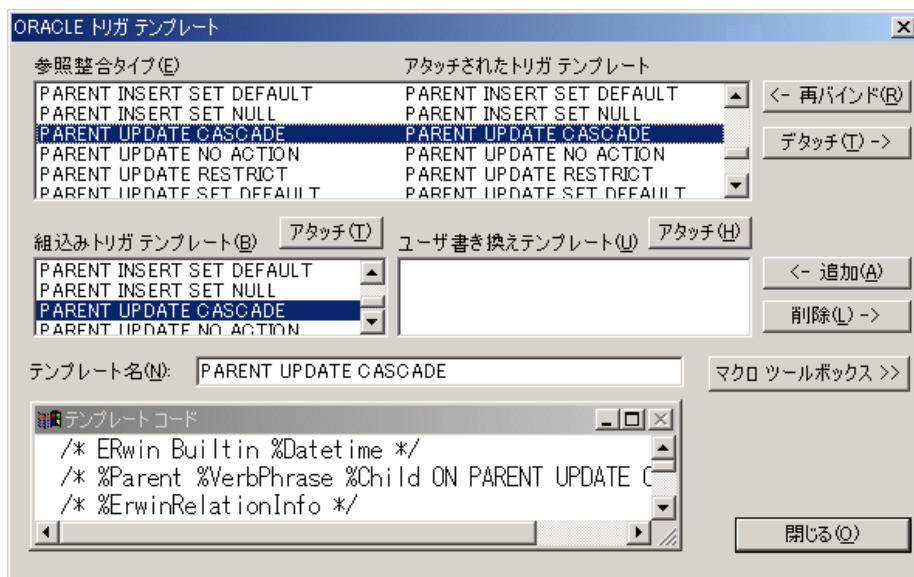


## トリガ テンプレートを作成し、特定の参照整合性に設定する

Lesson14 の中で、[SEIHIN] テーブルと [CHUMON\_MEISAI] テーブルの間のリレーションシップに、PARENT UPDATE CASCADE の参照整合を設定しました。この参照整合は、Oracle の外部キー制約では実現できませんので、データベース トリガを使用する必要があります。ERwin は PARENT UPDATE CASCADE を実現するデータベース トリガ テンプレートを準備していますので、このトリガをカスタマイズ（ここでは、コメント行を追加するだけです）することにより、トリガ テンプレートの作成方法を学習します。作成したトリガ テンプレートを PARENT UPDATE CASCADE の参照整合性に設定すると、ERwin がデフォルトで用意しているトリガ テンプレートに優先して生成できます。

前述の通り、ユーザー独自のトリガ テンプレートを作成する方法は 3 種類あり、それぞれに優先順位があります。ここでは、[トリガ テンプレート] ダイアログ ボックスを使用した参照整合タイプ上書きトリガを使用します。

[データベース] メニューの [RI トリガ] をポイントし、[グローバル トリガ テンプレート] をクリックして [ORACLE トリガ テンプレート] ダイアログ ボックスを開きます。



ダイアログ ボックスの上部のボックスには、どの参照整合タイプで、どのトリガ テンプレートを生成するかの一覧が表示されます。[ORACLE トリガ テンプレート] ダイアログ ボックスのデフォルト設定は、すべての参照整合タイプにおいて、デフォルトの組込みトリガ テンプレート（デフォルト トリガ）が指定されています。上図の例で言うと、PARENT UPDATE CASCADE という参照整合タイプにおいて、PARENT UPDATE CASCADE の組込みトリガ テンプレートが生成される設定となっています。それでは、ユーザー独自のトリガ テンプレートを作成して、PARENT UPDATE CASCADE の参照整合タイプに設定してみましょう。



1. [ORACLE トリガ テンプレート] ダイアログ ボックス上部のボックスに表示されている参照整合タイプ一覧から [PARENT UPDATE CASCADE] をクリックします。一覧の右側にアタッチされたトリガ テンプレート名が表示されます。このトリガ テンプレートの実際のテンプレートコードを見るには、ダイアログ ボックス中段の [組込みトリガ テンプレート] ボックスからテンプレート名を指定します。これにより、ダイアログ ボックス下段の [テンプレート名] ボックスにはテンプレート名が、[テンプレート コード] ウィンドウにはテンプレート コードが表示されます。
2. [テンプレート名] ボックスに「my\_trigger」と入力し、[追加] ボタンをクリックして [ユーザー書き換えテンプレート] ボックスに「my\_trigger」を追加します。
3. [テンプレート コード] ウィンドウのサイズを拡大して、テンプレート コードを見てください。1行目に、「/\* ERwin Builtin %Datetime \*/」というテンプレート コードが見えます。この行の前に、「/\* ERwin チュートリアル: my\_trigger \*/」というコメントを挿入します。
4. [ユーザー書き換えテンプレート] ボックスの上にある、[アタッチ] をクリックします。これにより、ダイアログ ボックス上部のボックス一覧の [PARENT UPDATE CASCADE] にはアタッチされたトリガ テンプレートとして「my\_trigger」が設定されます。
5. 下図のように設定できたら、[閉じる] をクリックして [ORACLE トリガ テンプレート] ダイアログ ボックスを閉じます。なお、参照整合タイプに設定したトリガ テンプレートをデフォルトの [組込みトリガ テンプレート] に戻すには [再バインド] をクリックします。設定を解除するには [デタッチ] をクリックします。

参照整合タイプ(E)	アタッチされたトリガ テンプレート
PARENT INSERT SET NULL	PARENT INSERT SET NULL
PARENT UPDATE CASCADE	my_trigger
PARENT UPDATE NO ACTION	PARENT UPDATE NO ACTION
PARENT UPDATE RESTRICT	PARENT UPDATE RESTRICT
PARENT UPDATE SET DEFAULT	PARENT UPDATE SET DEFAULT
PARENT UPDATE SET NULL	PARENT UPDATE SET NULL

6. ここで設定したトリガ テンプレートの「my\_trigger」を PARENT UPDATE CASCADE の参照整合タイプで実際に生成するには、[ORACLE スキーマ生成] ダイアログ ボックスで [オプション] タブをクリックし、[トリガ] の [ERwin 生成] の [RI タイプのオーバーライド] チェック ボックスをオンにする必要があります。

トリガ
<input checked="" type="checkbox"/> ERwin 生成
<input checked="" type="checkbox"/> RIタイプのオーバーライド
<input type="checkbox"/> リレーションシップのオーバーライド
<input type="checkbox"/> ユーザー定義
<input type="checkbox"/> RIタイプのオーバーライド
<input type="checkbox"/> リレーションシップのオーバーライド

## 特定の参照整合性からトリガ テンプレートを解除する

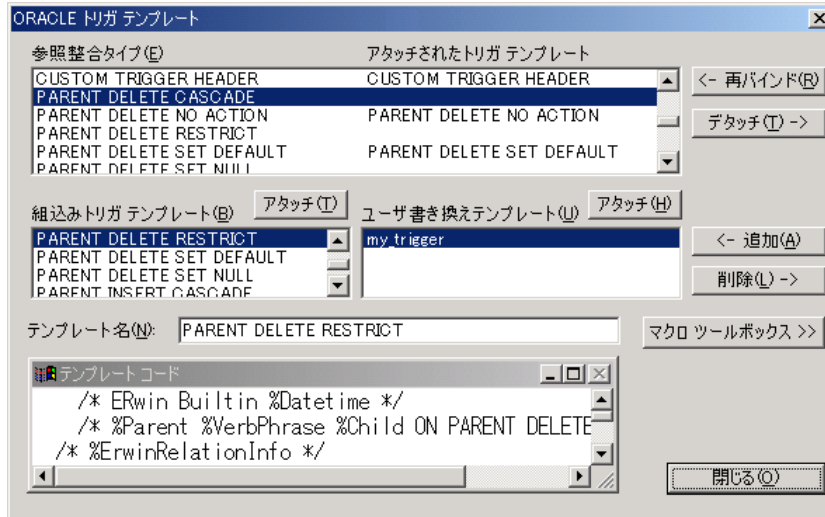
Lesson14 で設定した参照整合には、Oracle の外部キー制約を使用して整合性を実現できるものとデータベース トリガを使用すべきものがあります。前節で、データベース トリガで実現すべき参照整合 (PARENT UPDATE CASCADE) に、コメント行を追加したユーザー トリガ テンプレートを設定しました。この状態でスキーマ生成を行うと、PARENT UPDATE CASCADE を実現するためのユーザー トリガ テンプレートを生成すると同時に、外部キー制約で実現できる参照整合 (PARENT DELETE CASCADE、PARENT DELETE RESTRICT、PARENT DELETE SET NULL (このチュートリアルでは設定されていません)、PARENT UPDATE RESTRICT) に関しても、それを実現するためのトリガが生成されてしまいます。この状態は、外部キー制約とデータベース トリガで2重の参照整合設定を設けていることとなります。

参照整合性	外部キー制約	トリガ
PARENT DELETE CASCADE	○	○
PARENT DELETE RESTRICT	○	○
PARENT UPDATE CASCADE	×	○
PARENT UPDATE RESTRICT	○	○

この設定をデータベースに実装すると、外部キー制約とトリガの参照整合機能が互いにぶつかってしまい、エラーが発生する可能性があります。このエラーはデータベースへの実装時には発生せず、実際にデータを削除/更新したときに初めて発生します。理由としては、データベース トリガは作成するだけでは実行されないの、それが内包する矛盾に気が付かず、指定したタイミングでトリガが実行されて初めて矛盾が発覚するという訳です。

では、参照整合機能を2重に作成することを回避するために、外部キー制約で実現できる参照整合に関してはトリガを生成しないようにトリガ テンプレートを解除しましょう。参照整合に設定されているトリガ テンプレートを解除 (未設定に) すると、その参照整合に関してトリガは生成されません。

1. [データベース] メニューの [RI トリガ] をポイントし、[グローバルトリガ テンプレート] をクリックして [ORACLE トリガ テンプレート] ダイアログ ボックスを開きます。
2. ダイアログ ボックス上部のボックスの参照整合タイプ一覧からトリガを生成させない参照整合タイプを選択します。まず、[PARENT DELETE CASCADE] をクリックします。
3. [ORACLE トリガ テンプレート] ダイアログ ボックスの右上にある、[デタッチ] をクリックします。これにより、ダイアログ ボックス上部にあるボックスの参照整合タイプ一覧の PARENT DELETE CASCADE のアタッチされたトリガ テンプレートに何も表示されなくなり、トリガ テンプレートが解除されます。
4. 同様の操作で、PARENT DELETE RESTRICT と PARENT UPDATE RESTRICT のトリガ テンプレートも解除し、[閉じる] をクリックして [ORACLE トリガ テンプレート] ダイアログ ボックスを閉じます。



☆ **MEMO** ☆

## Lesson

# 16

実習時間：約 15 分

## データベースとの接続

ここでは、ERwin とデータベースを接続する方法について学習します。

ERwin がサポートするデータベースとの接続方法は、データベースごとに異なります。


### 説明

ERwin がデータベースと接続する方法には、大きく分けて 2 種類あります。データベースベンダーが提供するネイティブドライバによる接続、および ODBC ドライバによる接続です。

### ERwin がサポートするデータベース

ERwin は現在、以下のデータベースとのオンライン接続をサポートしています。

iSeries、Advantage Ingres、ORACLE、SAS、Teradata、DB2、InterBase、PROGRESS、SQL Server、HiRDB、Rdb、SQLBase、SQL Anywhere、INFORMIX、ODBC/Generic、Red Brick、SYBASE、Access、Clipper、dBASE IV、FoxPro、Paradox



Computer Associates ERwin -- 対象サーバ

対象SQL DBMS

iSeries(I)     Advantage Ingres(D)     ORACLE(A)     SAS(S)     Teradata(T)

DB2(Q)     InterBase(N)     PROGRESS(G)     SQL Server(Q)

HiRDB(H)     Rdb(R)     SQLBase(B)     SQL Anywhere(W)

INFORMIX(X)     ODBC /Generic(O)     Red Brick(K)     SYBASE(Y)

対象デスクトップDBMS

Access(E)     FoxPro(F)

Clipper(L)     dBASE IV(V)     Paradox(P)

ORACLE バージョン

9.x

デフォルト ORACLE データタイプ

CHAR(18)

デフォルトの非キーNULLオプション

NOT NULL(Q)

NULL(U)

OK

キャンセル

## ERwin とデータベースとの接続

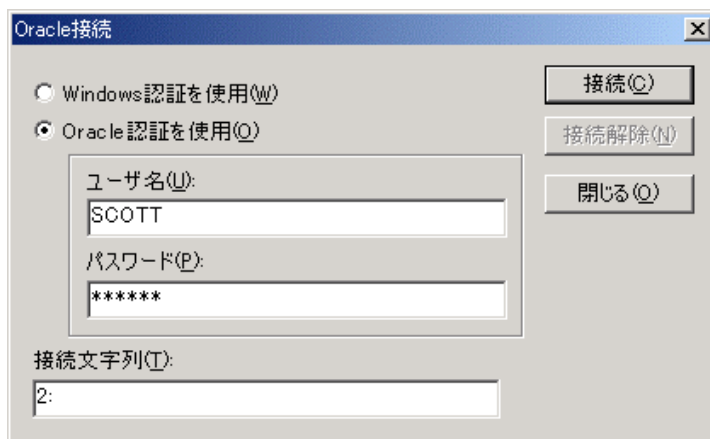
ERwin は多くのデータベースに接続できますが、このチュートリアルでは、比較的多くの ERwin ユーザーに使用されているデータベースとの接続方法をご紹介します。

データベース名	ミドルウェア	提供元
Oracle	SQL*Net または構築ファイル (hosts、tnsnames.ora) 付きの SQL Net8	日本オラクル株式会社
SQL Server	SQL Server クライアント ツール	マイクロソフト株式会社
SYBASE	SYBASE OPEN CLIENT	サイベース株式会社
ACCESS	ODBC ドライバ	マイクロソフト株式会社
SQL Anywhere	ODBC ドライバ	サイベース株式会社

▲ ERwin がサポートする代表的なデータベースとの接続環境

### ERwin と Oracle との接続方法

1. ERwin を使用するマシンに Oracle クライアントをインストールします。
2. Oracle データベースの接続設定を行い、サーバーに接続できることを確認します。
3. [データベース]メニューの[データベースの選択]をクリックして[Computer Associates ERwin - 対象サーバー] ダイアログ ボックスを開きます。[対象 SQL DBMS] の [ORACLE] をクリックし、適切な ORACLE バージョンを指定したら、[OK] をクリックして [Computer Associates ERwin - 対象サーバー] ダイアログ ボックスを閉じます。
4. [データベース] メニューの [データベース接続] をクリックして [Oracle 接続] ダイアログ ボックスを開きます。



5. [Oracle 認証を使用] をクリックし、[ユーザー名]、[パスワード]、および [接続文字列] ボックスに適切な文字列を入力して [接続] をクリックします。

	説明
ユーザー名	Oracle との接続ユーザー名
パスワード	接続ユーザーのパスワード
接続文字列	Oracle との接続文字列

## ERwin と Microsoft SQL Server との接続方法

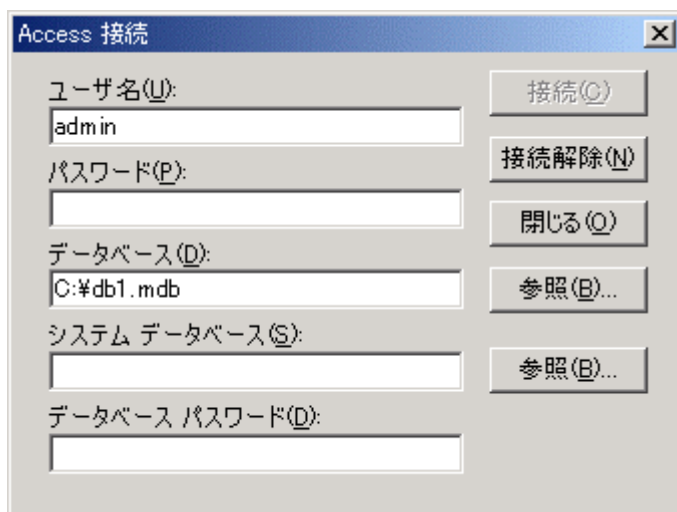
1. ERwin を使用するマシンに SQL Server クライアントをインストールします。
2. SQL Server データベースの接続設定を行い、サーバーに接続できることを確認します。
3. [データベース]メニューの[データベースの選択]をクリックして[Computer Associates ERwin – 対象サーバー] ダイアログ ボックスを開きます。[対象 SQL DBMS] の [SQL Server] をクリックし、適当な SQL Server バージョンを指定したら、[OK] をクリックして [Computer Associates ERwin – 対象サーバー] ダイアログ ボックスを閉じます。
4. [データベース] メニューの [データベース接続] をクリックして [SQL Server 接続] ダイアログ ボックスを開きます。

5. [ユーザー名]、[パスワード]、[データベース]、および [サーバー名] ボックスに適当な文字列を入力して [接続] をクリックします。

	説明
ユーザー名	SQL Server との接続ユーザー名
パスワード	接続ユーザーのパスワード
データベース	接続データベース名
サーバー名	データベースが存在するコンピュータ名

### ERwin と ACCESS との接続方法

1. Microsoft ACCESS をインストールし、ODBC データソースとして定義します。
2. [データベース]メニューの[データベースの選択]をクリックして[Computer Associates ERwin ー 対象サーバー] ダイアログ ボックスを開きます。[対象デスクトップ DBMS] の [Access] をクリックし、適当な Access バージョンを指定したら、[OK] をクリックして [Computer Associates ERwin ー 対象サーバー] ダイアログ ボックスを閉じます。
3. [データベース] メニューの [データベース接続] をクリックして [Access 接続] ダイアログ ボックスを開きます。



4. [ユーザー名] および [データベース] ボックスに適当な文字列を入力して [接続] をクリックします。

	説明
ユーザー名	ACCESS との接続ユーザー名 (デフォルトは「admin」)
パスワード	接続ユーザーのパスワード (デフォルトはパスワードなし)
データベース	接続に使用するデータソース名



## スキーマ生成

ここでは、ERwin とデータベースを接続し、ここまでのチュートリアルで作成してきたダイアグラム オブジェクトをデータベースに生成する方法について学習します。

このチュートリアルでは、対象データベースに Oracle を指定しています。Oracle との接続環境がない方は他のデータベースでも構いませんが、チュートリアル中のエディタ名や挿絵が異なることがあります。

### 説明



ERwin で作成したダイアグラム オブジェクトをデータベースに作成するには、ERwin からデータベースに接続して直接 DDL スクリプトを生成する方法と、DDL スクリプトをファイルに保存して他のツールから保存したファイルを実行する方法の 2 種類があります。このチュートリアルでは、ERwin からデータベースに接続して直接 DDL スクリプトを生成する方法でスキーマ生成を行います。

また、データベースに接続して直接 DDL スクリプトを生成する方法には、[スキーマ生成] ダイアログ ボックスを使用する方法と、[完全比較] ダイアログ ボックスを使用する方法の 2 種類があります。ここでは、[スキーマ生成] ダイアログ ボックスを使用する方法を使用します。[完全比較] ダイアログ ボックスを使用する方法については、Lesson 20 で行います。

### [スキーマ生成] ダイアログ ボックス

このダイアログ ボックスには、スキーマを生成するオブジェクトを選択するためのチェック ボックスが用意されています。選択したオプションに基づいて、スキーマが生成されます。

#### スキーマ生成オプション グループ


[スキーマ生成] ダイアログ ボックスのスキーマ生成オプションは、対象データベースにより異なります。[スキーマ生成] ダイアログ ボックスには、対象データベースがサポートしているスキーマ生成オプションだけが表示されます。

ERwin には、以下のようなスキーマ生成オプション グループが用意されています。

オプション グループ	説明
スキーマ	スキーマ レベルの設定
ビュー	ビューに関する設定
テーブル	テーブルに関する設定
カラム	カラムに関する、制約情報、物理順序、デフォルト値などの設定
インデックス	インデックスに関する設定
参照整合性	主キーおよび外部キーの作成、および主キーや外部キーの値が変更または削除されたときの設定
トリガ	トリガ（参照性合成トリガおよびユーザー定義トリガ）の設定
その他	対象サーバーが提供する特別な機能をサポートするための設定

## スキーマの生成

ここまでのチュートリアルで作成してきたダイアグラム オブジェクトをデータベースに生成します。  
 [スキーマ生成] ダイアログ ボックスで必要なオプションを選択してスキーマをデータベースに生成します。

1. [データベース] ツールバーの [フォワード エンジニアリング]  をクリックするか、または [ツール] メニューの [フォワード エンジニアリング/スキーマ生成] をクリックして [ORACLE スキーマ生成] ダイアログ ボックスを開きます。
2. [ORACLE スキーマ生成] ダイアログ ボックスで、[新規] をクリックして [新規スキーマ生成オプション セット] ダイアログ ボックスを開きます。
3. [新規スキーマ生成オプション セット] ダイアログ ボックスの [名前] ボックスに「ERwin チュートリアル」と入力し、[OK] をクリックします。
4. これにより、[ORACLE スキーマ生成] ダイアログ ボックスで選択したオプションの設定を「ERwin チュートリアル」という名前で保存できます。[オプション セット] ボックスの一覧の [ERwin チュートリアル] をクリックすると、いつでも同じオプションの設定でスキーマを生成できます。
5. 以下のオプションのチェック ボックスをオンにしてください。


オプショングループ	オプション名	意味
スキーマ	プリ スクリプト	スキーマを生成する前にプリ スクリプトを生成します。
	ポスト スクリプト	スキーマを生成した後にポスト スクリプトを生成します。
テーブル	CREATE TABLE	CREATE TABLE ステートメントを生成します。
	プリ スクリプト	CREATE TABLE ステートメントを生成する前にプリ スクリプトを生成します。
カラム	CHECK 制約	カラムに CHECK 制約を作成します。
	デフォルト値	カラムにデフォルト値を作成します。
インデックス	CREATE INDEX	選択したインデックスの CREATE INDEX ステートメントを生成します。
	代替キー(AK)	代替キー(AK) インデックスを作成します。
	逆方向エントリ(IE)	逆方向エントリ(IE) インデックスを作成します。
参照整合性	主キー(PK)	テーブルに主キーを作成します。
	変更/PK	ALTER ステートメントで主キーを生成します。
	外部キー(FK)	テーブルに外部キーを作成します。
	削除時	外部キーが削除されたときに、参照整合性オプションを実行します。
	更新時	外部キーが変更されたときに、参照整合性オプションを実行します。
	ステートメント形式 変更/FK	ALTER ステートメントで外部キーを生成します。
トリガ	ERwin 生成	トリガを生成します。
	RI タイプのオーバーライド	参照整合タイプ上書きのトリガを生成します。
その他のオプション	制約名	各制約に制約名を生成します。
	コメント	エンティティ定義/属性定義をテーブル コメント/カラム コメントとして生成します。

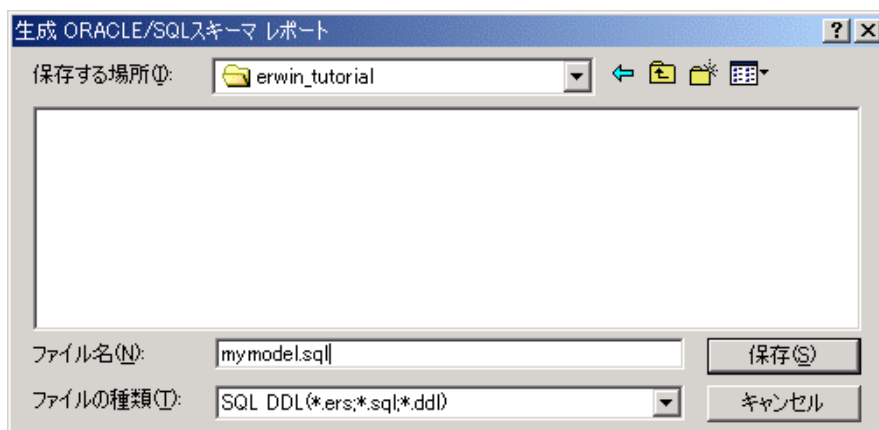
6. 指定したオプションのチェック ボックスをオンにしたら、[プレビュー]をクリックし、[ORACLE スキーマ生成のプレビュー] ダイアログ ボックスで生成されたスキーマ (DDL スクリプト) を見てみましょう。選択したオプションに基づいて、スキーマ (DDL スクリプト) が生成されたことが確認できます。

注 :生成される DDL スクリプトは、この Lesson の最後に記載してあります。

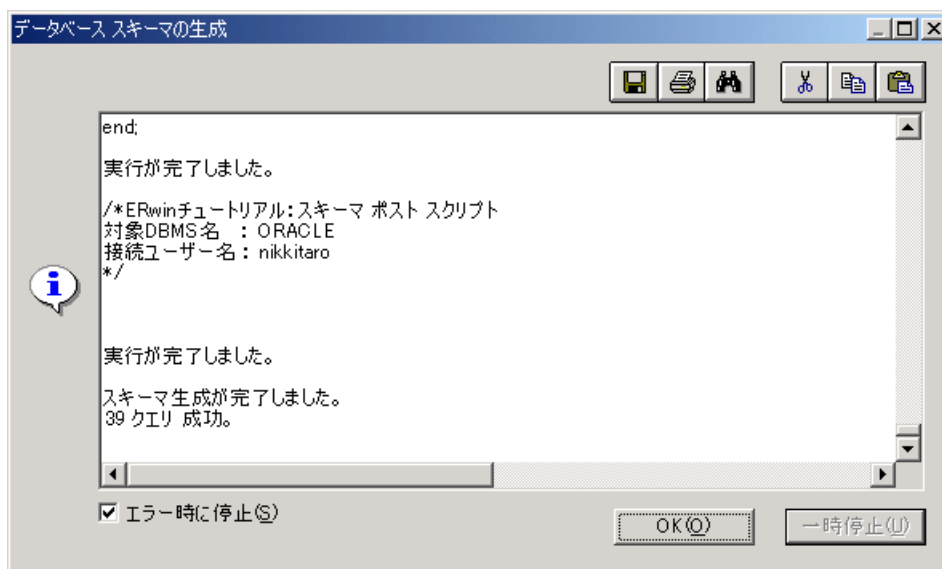


ERwin を使用して間もないユーザーの中には、[ORACLE スキーマ生成] ダイアログ ボックスの各オプションの意味を理解せずに、無意味に多くのオプションのチェック ボックスをオンにしてスキーマを生成する方がいるようです。スキーマ生成を行う前に、必要な定義がすべてあるか、不必要な定義が含まれていないかを確認してください。

7. [ORACLE スキーマ生成のプレビュー] ダイアログ ボックスで生成されたスキーマ (DDL スクリプト) を確認したら、ファイルに保存します。スキーマ (DDL スクリプト) をファイルに保存するには、をクリックして [生成 ORACLE/SQL スキーマ レポート] ダイアログ ボックスを開きます。次に、保存する場所を指定し、[ファイル名] ボックスに「mymodel.sql」と入力して [保存] をクリックします。



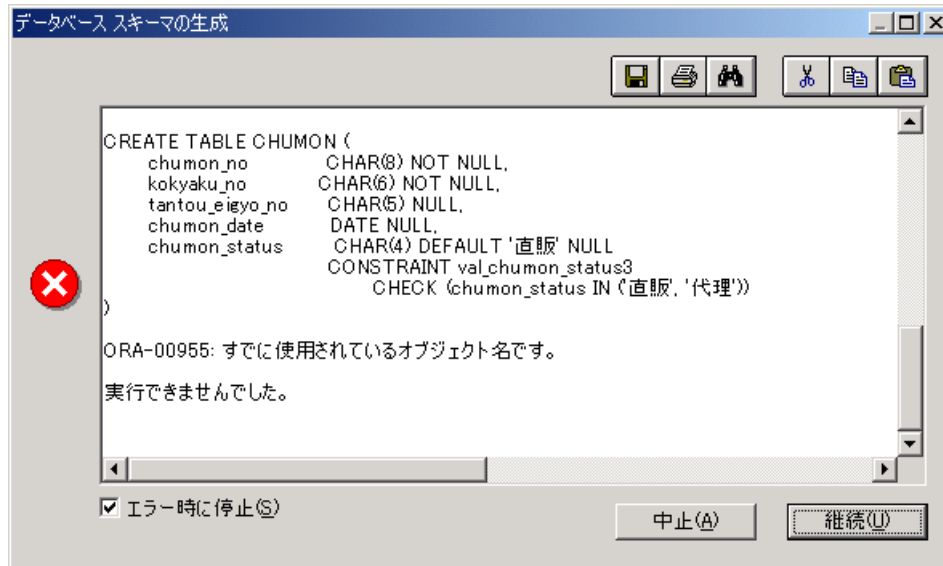
- スキーマ (DDL スクリプト) を保存したら、[ORACLE スキーマ生成のプレビュー] ダイアログ ボックスの [生成] をクリックしてデータベースにスキーマ (DDL スクリプト) を生成します。データベースに接続するための [Oracle 接続] ダイアログ ボックスが表示されたら、[Oracle 認証を使用] をクリックし、適切な [ユーザー名]、[パスワード]、および [接続文字列] を入力して [接続] をクリックします。
- 接続が完了すると、[データベース スキーマの生成] ダイアログ ボックスが開き、スキーマ (DDL スクリプト) がデータベースに生成されます。生成が完了した場合、DDL スクリプトに対して「実行が完了しました。」というメッセージが表示されます。生成が失敗した場合、DDL スクリプトに対してデータベースが返したエラー メッセージと「実行できませんでした。」という ERwin のメッセージが表示されます。すべてのスキーマ (DDL スクリプト) の生成が完了したら、「スキーマ生成が完了しました。」というメッセージが表示されます。また、生成が成功した DDL スクリプトの数と失敗した DDL スクリプトの数が表示されます。



- スキーマ生成が完了したら、[OK] をクリックして [データベース スキーマの生成] ダイアログ ボックスを閉じます。次に、[ORACLE スキーマ生成のプレビュー] ダイアログ ボックスで [閉じる] をクリックします。最後に、[ORACLE スキーマ生成] ダイアログ ボックスで [OK] をクリックします。

## スキーマ生成に失敗したら

デフォルトでは、DDL スクリプトの実行時にエラーが発生すると、DDL スクリプトの実行が一時停止されます。この場合、[データベース スキーマの生成] ダイアログ ボックスに表示されるエラー メッセージを確認してエラーの原因を調査してください。



エラーが発生した原因が判明した後、スキーマ生成を継続する場合は [継続] をクリックします。スキーマ生成を中止する場合は [中止] をクリックします。スキーマ生成時にエラーが発生しても生成を中止せずにスキーマ生成を続行するには、[エラー時に停止] チェック ボックスをオフにしてください。



既存のテーブルと同名のテーブルを生成したことによるエラーが発生したら、テーブル オプション グループの [DROP TABLE] チェック ボックスをオンにします。これにより、CREATE TABLE 文を生成する前に、そのテーブルを削除する DROP TABLE 文が生成されます。

## ERwin が生成したスキーマ (DDL スクリプト)

ファイルに保存したスキーマ (DDL スクリプト) をテキスト エディタで開き、以下のオブジェクトのスキーマが正しく生成されていることを確認してください。

- スキーマ プリ スクリプト
- スキーマ ポスト スクリプト
- テーブル プリ スクリプト
- トリガ テンプレート (PARENT UPDATE CASCADE)

## ERwin が生成したスキーマ スクリプト例

```

/*
ERwin チュートリアル : スキーマ プリ スクリプト
作成者   : 日揮 太郎
ファイル名 : C:\ERWIN\MYMODEL.ER1
作成日   : Thu May 06 18:21:09 2004
*/

```

} スキーマ プリ スクリプト

```

/*
////////////////////////////////////
テーブル名 : CHUMON
-----

カラム リスト
chumon_no - CHAR(8)
kokyaku_no - CHAR(6)
tantou_eigyo_no - CHAR(5)
chumon_date - DATE
chumon_status - CHAR(4)

////////////////////////////////////
*/

```

} テーブル プリ スクリプト

```

CREATE TABLE CHUMON (
    chumon_no          CHAR(8) NOT NULL,
    kokyaku_no         CHAR(6) NOT NULL,
    tantou_eigyo_no    CHAR(5) NULL,
    chumon_date        DATE NULL,
    chumon_status      CHAR(4) DEFAULT '直販' NULL
                    CONSTRAINT val_chumon_status
                    CHECK (chumon_status IN ('直販', '代理'))
);

COMMENT ON TABLE CHUMON IS '注文とは、登録顧客より取り扱い製品を受注することをいう。';
COMMENT ON COLUMN CHUMON.chumon_no IS '命名ルールに従い8バイトで登録する。';
COMMENT ON COLUMN CHUMON.kokyaku_no IS '製品を購入する企業を命名ルールに従い6バイトで登録する。';
COMMENT ON COLUMN CHUMON.tantou_eigyo_no IS '従業員を一意に識別するための番号で、命名ルールに従い5バイトで登録する。';
COMMENT ON COLUMN CHUMON.chumon_date IS '受注した年月日を西暦で登録する。';
COMMENT ON COLUMN CHUMON.chumon_status IS '受注状況を登録する。';

CREATE INDEX XIE1CHUMON ON CHUMON
(
    tantou_eigyo_no          ASC,
    kokyaku_no              ASC
);

ALTER TABLE CHUMON
    ADD ( CONSTRAINT XPKCHUMON PRIMARY KEY (chumon_no) );

/*
////////////////////////////////////
テーブル名 : CHUMON_MEISA1
-----

カラム リスト
chumon_no - CHAR(8)
seihin_no - CHAR(7)
chumon_suuryo - CHAR(3)
shoukei - NUMBER

////////////////////////////////////
*/

```

```

CREATE TABLE CHUMON_MEISAI (
    chumon_no          CHAR(8) NOT NULL,
    seihin_no          CHAR(7) NOT NULL,
    chumon_suuryo      CHAR(3) NOT NULL,
    shoukei            NUMBER NULL
);

COMMENT ON COLUMN CHUMON_MEISAI.chumon_no IS '命名ルールに従い8バイトで登録する。';
COMMENT ON COLUMN CHUMON_MEISAI.seihin_no IS '取り扱い製品を命名ルールに従い7バイトで登録する。';
COMMENT ON COLUMN CHUMON_MEISAI.chumon_suuryo IS '受注本数を登録する。';

ALTER TABLE CHUMON_MEISAI
    ADD ( CONSTRAINT XPKCHUMON_MEISAI PRIMARY KEY (chumon_no, seihin_no) );

/*
////////////////////////////////////
テーブル名 : JUGYOIN
-----
カラム リスト
jugyoin_no - CHAR(5)
jugyoin_namae - VARCHAR2(20)
jugyoin_jusho - VARCHAR2(40)
jugyoin_telno - CHAR(10)

////////////////////////////////////
*/

CREATE TABLE JUGYOIN (
    jugyoin_no          CHAR(5) NOT NULL,
    jugyoin_namae      VARCHAR2(20) NOT NULL,
    jugyoin_jusho      VARCHAR2(40) NULL,
    jugyoin_telno      CHAR(10) NULL
);

COMMENT ON TABLE JUGYOIN IS '従業員とは、正社員と契約社員のことをいう。';
COMMENT ON COLUMN JUGYOIN.jugyoin_no IS '従業員を一意に識別するための番号で、命名ルールに従い5バイトで登録する。';
COMMENT ON COLUMN JUGYOIN.jugyoin_namae IS '略字を用いず正式な姓で登録する。';
COMMENT ON COLUMN JUGYOIN.jugyoin_jusho IS '都道府県から登録する。';
COMMENT ON COLUMN JUGYOIN.jugyoin_telno IS '市外局番から登録する。';

ALTER TABLE JUGYOIN
    ADD ( CONSTRAINT XPKJUGYOIN PRIMARY KEY (jugyoin_no) );

/*
////////////////////////////////////
テーブル名 : KOKYAKU
-----
カラム リスト
kokyaku_no - CHAR(6)
kokyaku_namae - VARCHAR2(30)
kokyaku_jusho - VARCHAR2(40)
kokyaku_telno - CHAR(10)

////////////////////////////////////
*/

CREATE TABLE KOKYAKU (
    kokyaku_no          CHAR(6) NOT NULL,
    kokyaku_namae      VARCHAR2(30) NOT NULL,
    kokyaku_jusho      VARCHAR2(40) NULL,
    kokyaku_telno      CHAR(10) NOT NULL
);

```

```

COMMENT ON COLUMN KOKYAKU.kokyaku_no IS '製品を購入する企業を命名ルールに従い6バイトで登録する。';
COMMENT ON COLUMN KOKYAKU.kokyaku_namae IS '略称を使用せず正式な社名で登録する。';
COMMENT ON COLUMN KOKYAKU.kokyaku_jusho IS '都道府県から登録する。';
COMMENT ON COLUMN KOKYAKU.kokyaku_telno IS '市外局番から登録する。';

```

```

CREATE UNIQUE INDEX XAK1KOKYAKU ON KOKYAKU
(
    kokyaku_namae          ASC
);

```

```

ALTER TABLE KOKYAKU
ADD ( CONSTRAINT XPKKOKYAKU PRIMARY KEY (kokyaku_no) );

```

```

/*
//////////
テーブル名 : SEIHIN
-----

```

```

カラム リスト
seihin_no - CHAR(7)
seihin_namae - VARCHAR2(20)
seihin_kakaku - NUMBER

```

```

//////////
*/

```

```

CREATE TABLE SEIHIN (
    seihin_no          CHAR(7) NOT NULL,
    seihin_namae      VARCHAR2(20) NOT NULL,
    seihin_kakaku     NUMBER NULL
                    CONSTRAINT val_kakaku
                    CHECK (seihin_kakaku <= 1000000)
);

```

```

COMMENT ON COLUMN SEIHIN.seihin_no IS '取り扱い製品を命名ルールに従い7バイトで登録する。';
COMMENT ON COLUMN SEIHIN.seihin_namae IS '略称を使用せず正式な製品名で登録する。';
COMMENT ON COLUMN SEIHIN.seihin_kakaku IS '標準価格を登録する。';

```

```

ALTER TABLE SEIHIN
ADD ( CONSTRAINT XPKSEIHIN PRIMARY KEY (seihin_no) );

```

```

ALTER TABLE CHUMON
ADD ( CONSTRAINT ukeru
    FOREIGN KEY (tantou_eigyo_no)
    REFERENCES JUGYOIN );

```

```

ALTER TABLE CHUMON
ADD ( CONSTRAINT dasu
    FOREIGN KEY (kokyaku_no)
    REFERENCES KOKYAKU );

```

```

ALTER TABLE CHUMON_MEISAI
ADD ( CONSTRAINT motsu
    FOREIGN KEY (chumon_no)
    REFERENCES CHUMON
    ON DELETE CASCADE );

```

```

ALTER TABLE CHUMON_MEISAI
ADD ( CONSTRAINT hitsuyou
    FOREIGN KEY (seihin_no)
    REFERENCES SEIHIN );

```



```
create or replace trigger tU_SEIHIN after UPDATE on SEIHIN for each row
-- ERwin Builtin Thu May 06 18:21:09 2004
-- UPDATE trigger on SEIHIN
declare numrows INTEGER;
begin
  /* ERwinチュートリアル : my_trigger */
  /* ERwin Builtin Thu May 06 18:21:09 2004 */
  /* SEIHIN R/3 CHUMON_MEISAI ON PARENT UPDATE CASCADE */
  /* ERWIN_RELATION:PARENT_OWNER="", PARENT_TABLE="SEIHIN"
  CHILD_OWNER="", CHILD_TABLE="CHUMON_MEISAI"
  P2C_VERB_PHRASE="R/3", C2P_VERB_PHRASE="",
  FK_CONSTRAINT="hitsuyou", FK_COLUMNS="seihin_no" */
  if
    /* %JoinPKPK(:%Old,:%New," <> "," or ") */
    :old.seihin_no <> :new.seihin_no
  then
    update CHUMON_MEISAI
    set
      /* %JoinFKPK(CHUMON_MEISAI,:%New," = "," and") */
      CHUMON_MEISAI.seihin_no = :new.seihin_no
    where
      /* %JoinFKPK(CHUMON_MEISAI,:%Old," = "," and") */
      CHUMON_MEISAI.seihin_no = :old.seihin_no;
  end if;

-- ERwin Builtin Thu May 06 18:21:09 2004
end;
/
```

トリガー テンプレート

```
/*
ERwinチュートリアル : スキーマ ポスト スクリプト
対象 DBMS 名      : ORACLE
接続ユーザー名 :
*/
```

スキーマ ポスト スクリプト

☆ **MEMO** ☆

## サブジェクト エリア/ストアド ディスプレイの作成

ここでは、データベースの設計作業を効率化し、プレゼンテーション効果を高めるために大変有効な 2 つの機能を紹介します。

これらの機能を使用すると、エンタープライズ モデルからある業務に関連したエンティティだけを抽出させて表示したり、ダイアグラムを異なった観点から検討するために、別のレイアウトや別の表示レベルで保存したりすることができます。

### 説明

#### サブジェクト エリア

サブジェクト エリアとは、ダイアグラム全体から特定のダイアグラム オブジェクト（エンティティ、リレーションシップ、サブタイプ リレーションシップ、テキスト ブロックなど）だけを切り出して名前を付けたものです。たとえば、エンタープライズ モデルから製品の在庫管理業務に関するオブジェクトだけを切り出して表示すると、他の業務を意識せずに設計作業を進めることができるため、業務担当者はより簡単で分かりやすく設計図を見ることができます。サブジェクト エリアは元となるダイアグラム全体から一部分を切り出したものですが、すべてのダイアグラム オブジェクトを含むサブジェクト エリアを「<メイン サブジェクト エリア>」と呼びます。



#### ストアド ディスプレイ

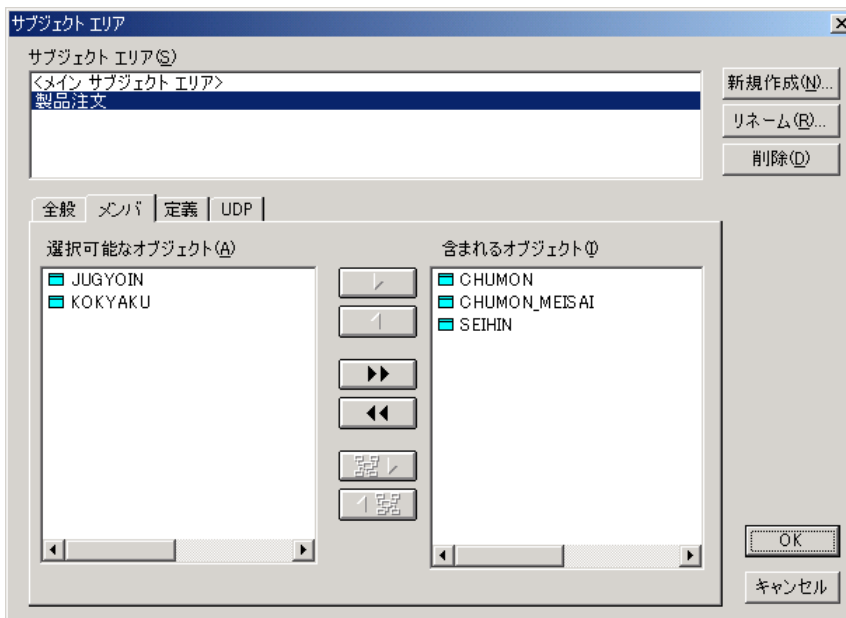
ストアド ディスプレイとは、ダイアグラム オブジェクトのレイアウトや表示レベルの設定に名前を付けたものです。たとえば、在庫管理業務の全体を把握するためにエンティティ レベルで表示したストアド ディスプレイと在庫数の変動の処理に注目したレイアウトを属性レベルで表示するストアド ディスプレイを個別に保存できます。それぞれ、注目している観点は異なりますが、双方とも在庫管理業務という 1 つのサブジェクト エリアから作成されています。つまり、ストアド ディスプレイを使用すると、同じダイアグラム オブジェクトに対して異なったレイアウトと表示レベルを保存することができます。

## サブジェクト エリア

どの製品が、いつ、どの程度の数量を受注されたかに注目したサブジェクト エリアを作成します。現在の<メイン サブジェクト エリア>から [CHUMON] テーブル、 [CHUMON\_MEISAI] テーブル、および [SEIHIN] テーブルだけを切り出します。

### サブジェクト エリアの作成

- 以下のいずれかの方法で、[サブジェクト エリア] ダイアログ ボックスを開きます。
  - ・ [ERwin 標準ツールバー] の [サブジェクト エリアの作成]  をクリックします。
  - ・ [モデル] メニューの [サブジェクト エリア] をクリックします。
  - ・ モデル エクスプローラの [サブジェクト エリア] タブをクリックし、[サブジェクト エリア] を右クリックして、ショートカット メニューの [プロパティ] をクリックします。
- [新規作成] をクリックして [新規サブジェクト エリア] ダイアログ ボックスを開きます。[名前] ボックスに「製品注文」と入力し、[OK] をクリックしてダイアログ ボックスを閉じます。
- [メンバ] タブをクリックし、[選択可能なオブジェクト] ボックスで、<Ctrl>キーを押しながら [CHUMON] テーブル、 [CHUMON\_MEISAI] テーブル、および [SEIHIN] テーブルをクリックします。ダイアログ ボックス中央の  をクリックすると、3つのテーブルが [含まれるオブジェクト] ボックスに移動します。また、他の矢印ボタンを使用して選択するテーブルを自由に変更できます。
- [全般] タブをクリックして、[作成者] ボックスに自分の名前を入力します。次に、[定義] タブをクリックして、[定義] ボックスに「製品の注文に関する SA」と入力します。[OK] をクリックして [サブジェクト エリア] ダイアログ ボックスを閉じます。



- ダイアグラムに戻ると、[CHUMON] テーブル、 [CHUMON\_MEISAI] テーブル、および [SEIHIN] テーブルだけが表示されています。ダイアグラムのタイトルバーに「MYMODEL.ER1 \*: 製品注文 / ディスプレイ 1」と表示され、現在「製品注文」サブジェクト エリアが選択されていることわかります。

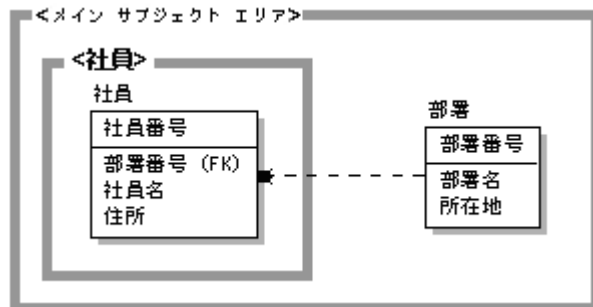
## サブジェクト エリアの切り替え

現在、「製品注文」サブジェクト エリアが選択されているため、ダイアグラム上に [CHUMON] テーブル、[CHUMON\_MEISAI] テーブル、および [SEIHIN] テーブルだけが表示されています。他のテーブルを表示するためにはサブジェクト エリアを切り替える必要があります。モデル エクスプローラの [サブジェクト エリア] タブをクリックして、サブジェクト エリアの一覧から、<メイン サブジェクト エリア>を右クリックして、ショートカット メニューの [移動] をクリックします。すべてのダイアグラム オブジェクトが表示されます。



## [スキーマ生成時に関連するリレーションシップのフィルタ] チェック ボックス

あるサブジェクト エリアのテーブルが他のサブジェクト エリアのテーブルから外部キーを継承しているとします。たとえば、[社員] サブジェクト エリアには [社員] テーブルしか表示されていませんが、実際は<メイン サブジェクト エリア>の [部署] テーブルとリレーションシップで結ばれているとします。



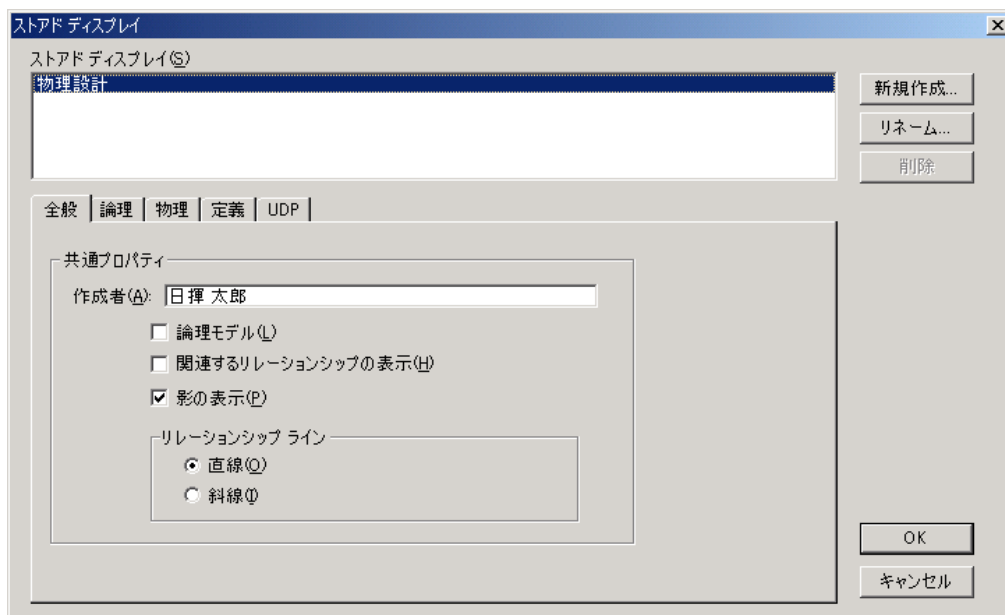
たとえば、[社員] サブジェクト エリアからスキーマ生成を行った場合、[CREATE TABLE] チェックボックスをオンにして生成されるテーブルは [社員] テーブルのみですが、[外部キー (FK)] チェックボックスをオンにした場合は、<メイン サブジェクト エリア>の [部署] テーブルに対して外部キー制約が生成されてしまいます。データベースに [部署] テーブルが作成されていない場合は、外部キー制約の宣言文はエラーとなってしまいます。これを回避するために、[サブジェクト エリア] ダイアログ ボックスの [全般] タブの [スキーマ生成に関連するリレーションシップのフィルタ] チェックボックスを使用します。このチェックボックスをオンにすると、他のサブジェクト エリアから継承している外部キーに関しては、外部キー制約を宣言しないようにフィルタをかけることができます。

## ストアド ディスプレイ

ダイアグラムの表示レベルとレイアウトを変更し、名前を付けて保存できます。ここでは、エンティティ名のみを表示する「概念設計」、論理情報を表示する「論理設計」、および物理情報を表示する「物理設計」の3つのストアド ディスプレイを作成します。

### ストアド ディスプレイの作成

- 以下のいずれかの方法で、[ストアド ディスプレイ] ダイアログ ボックスを開きます。
  - [フォーマット] メニューから [ストアド ディスプレイの設定] をクリックします。
  - モデル エクスプローラの [サブジェクト エリア] タブをクリックし、<メイン サブジェクト エリア>を展開します。[ストアド ディスプレイ] を右クリックして、ショートカット メニューの [プロパティ] をクリックします。
- 新規ダイアグラムを作成すると、デフォルトで作成される [ディスプレイ 1] スタド ディスプレイの名前を「物理設計」に変更します。[リネーム] をクリックして [リネーム スタド ディスプレイ] ダイアログ ボックスを開きます。[名前] ボックスに「物理設計」と入力し、[OK] をクリックしてダイアログ ボックスを閉じます。[全般] タブをクリックして、[作成者] ボックスに自分の名前を入力します。次に、[定義] タブをクリックして、[定義] ボックスに「物理情報を表示する SD」と入力します。



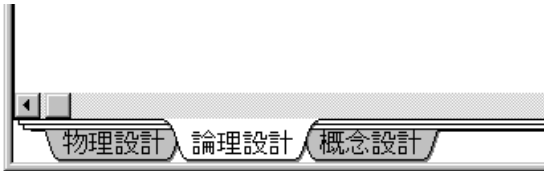
- [OK] をクリックして [ストアド ディスプレイ] ダイアログ ボックスを閉じると、ダイアグラム左下の [ストアド ディスプレイ] タブが「ディスプレイ 1」から「物理設計」に変更され、ダイアグラムのタイトルバーが「MYMODEL.ER1 \*:<メイン サブジェクト エリア>/物理設計」と表示され、<メイン サブジェクト エリア>の [物理設計] スタド ディスプレイが選択されていることがわかります。
- 現在の表示レベルが物理スキーマ レベルに合っていない方は、[ストアド ディスプレイ] ダイアログ ボックスを開き、[物理] タブをクリックして表示レベルを変更してください。

5. 以下の表を参照して、[論理設計] ストアド ディスプレイと [概念設計] ストアド ディスプレイを作成してください。作成したら [OK] をクリックして [ストアド ディスプレイ] ダイアログ ボックスを閉じます。

SD 名	作成者	定義
論理設計	お名前	論理情報を表示する SD
概念設計	お名前	エンティティ名のみを表示して、注文明細に注目したレイアウトを持つ SD

### ストアド ディスプレイの切り替え

ストアド ディスプレイを切り替えるには、ダイアグラム左下の [ストアド ディスプレイ] タブをクリックします。3つのストアド ディスプレイが作成されたら、[論理設計] スストアド ディスプレイ タブをクリックしてください。



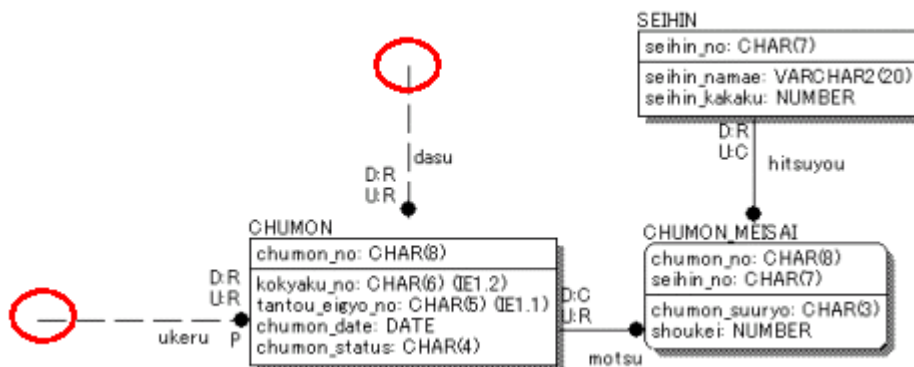
選択した [論理設計] スストアド ディスプレイが表示するダイアグラムは、物理スキーマ レベルで表示されているはずですが、これは、新規に作成するストアド ディスプレイは元となったストアド ディスプレイの表示レベルとレイアウトを継承するからです。作成したストアド ディスプレイにふさわしい表示レベルとレイアウトに変更しましょう。それぞれのストアド ディスプレイごとに、論理モデルと物理モデルで、異なる表示レベルとレイアウトを設定できます。論理モデルでの表示設定は、[ストアド ディスプレイ] ダイアログ ボックスの [論理] タブで、物理モデルでの表示設定は [物理] タブで行います。以下の表を参考に、お好きなように設定してみてください。

SD 名	表示レベル	レイアウト
論理設計	属性	物理設計 SD と同様で構いません。
概念設計	エンティティ	[注文明細] エンティティが目立つ位置にくるようにレイアウトします。

### [関連するリレーションシップの表示] チェック ボックス

あるサブジェクト エリアのテーブルが他のサブジェクト エリアのテーブルと関連しているとします。通常、このサブジェクト エリアには他のサブジェクト エリアにあるテーブルとの間に作成されているリレーションシップは表示されません。つまり、リレーションシップを表示せずに、外部キーだけは移行します。

[ストアド ディスプレイ] ダイアログ ボックスの [全般] タブの [関連するリレーションシップの表示] チェック ボックスをオンにすると、当該サブジェクト エリアには選択されていないテーブルとの間に結ばれたリレーションシップでもリレーションシップ ラインだけを表示できます。このとき、リレーションシップで結ばれている関連先テーブルは、当該サブジェクト エリアに表示することはできないため、リレーションシップはダイアグラムの空白部分から (へと) 現れて (消える) ように表示されます。





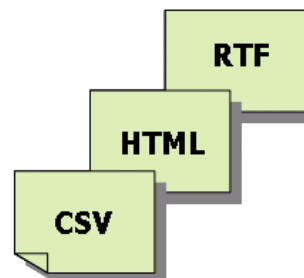
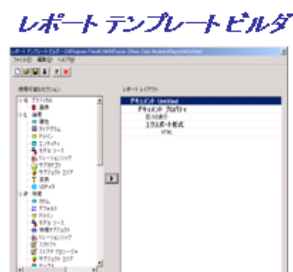
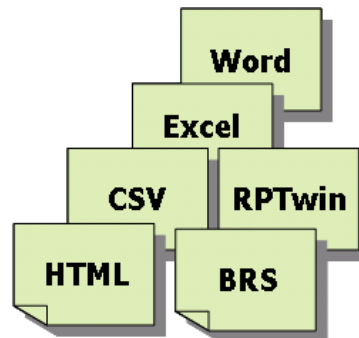
## レポート作成

ここでは、ERwin でデータベース設計のドキュメントを作成する方法について学習します。  
ERwin のレポート機能を使用すると ER 図から簡単にテーブル定義書などのドキュメントを作成できます。

説明 

旧来のやり方では、データベースのテーブル構造をドキュメントするために Excel などのソフトを使用してテーブル定義書を作成していました。作成したテーブル定義書は、実際にデータベース上に構築したテーブル構造やその DDL とは何のリンクも持っていないため、テーブル構造の仕様に変更が発生した場合、データベースのテーブルの更新、Excel のテーブル定義書の更新、模造紙などに手書きしたテーブルの関連図（ER 図）の更新などを個別に行う必要がありました。

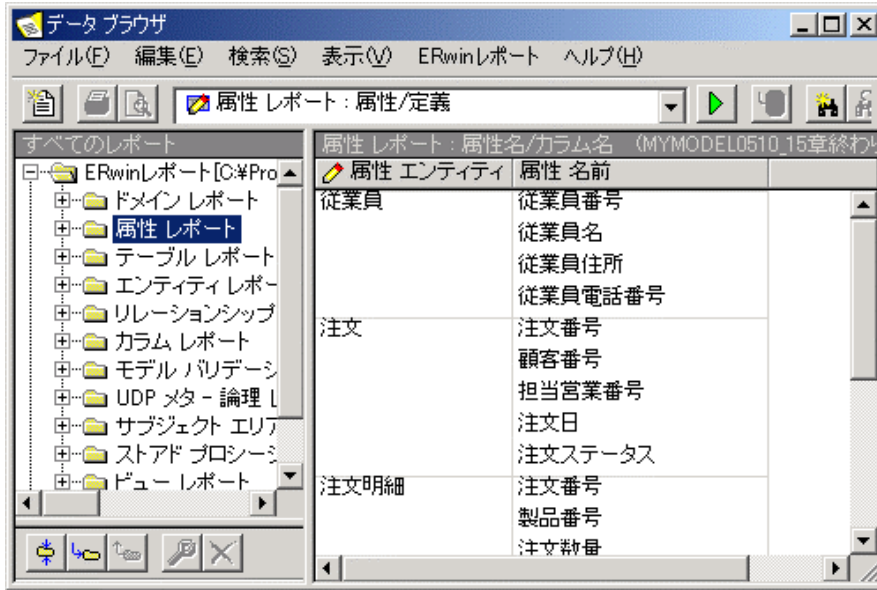
ERwin を使用すると、ERwin のダイアグラムを更新することにより、データベースの更新や新しいテーブル定義書の作成などが簡単に行えます。



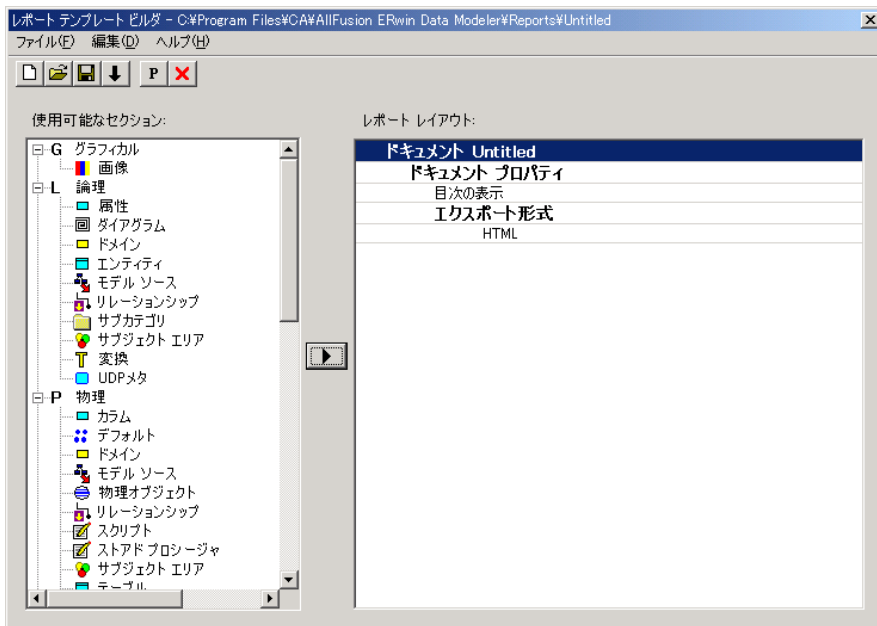
### ERwin のレポート機能

ERwin には、ER ダイアグラムに関するレポートを参照および作成するための、高い汎用性と高いカスタマイズ機能を持ち合わせた以下の 2 種類のツールが用意されています。

- データ ブラウザ**  
 データ ブラウザの事前定義レポート (カテゴリ) のセットを選択すると、表形式のレポートが UI 上に表示されます。オブジェクト情報を検索したい場合などに便利です。データ ブラウザに出力した内容は、印刷したり、Word、Excel、HTML などの形式で出力したりできます。



- レポート テンプレート ビルダ (RTB)**  
 レポート テンプレート ビルダを使用すると、再利用可能なレポート テンプレートを作成して ER ダイアグラムの情報を出力できます。テンプレートも多数用意されています。作成したレポートは、HTML、RTF、およびテキストの形式で出力できます。




## レポートの作成

レポート テンプレート ビルダを使用して、ここまでのチュートリアルで作成した ER ダイアグラム情報をレポートします。

### 既存のテンプレートを使用したレポートの作成

既存のテンプレートを使用して、エンティティ名を CSV 形式で出力します。

1. [ERwin 標準ツールバー] の [レポート テンプレート ビルダ]  をクリックするか、[ツール] メニューの [レポート ビルダ] をポイントし、[レポート ビルダ] をクリックして [レポート テンプレート] ダイアログ ボックスを開きます。
2. [使用可能なテンプレート] ボックスの [CSV エンティティ名.rtb] をクリックします。
3. [実行] をクリックして、以下エンティティ名が CSV 形式で出力されていることを確認します。

	A
1	エンティティ 名前
2	顧客
3	従業員
4	製品
5	注文
6	注文明細
7	


CSV エンティティ名

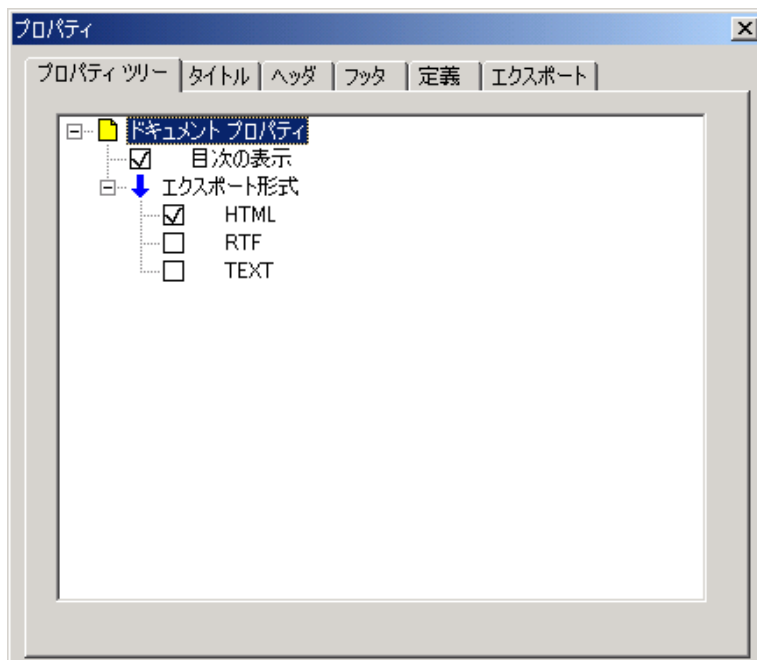




ERwin の既存のテンプレートを編集して新しいテンプレートを作成すると、短時間で好みのレポートを作成できます。

## 新規作成したテンプレートを使用したレポートの作成

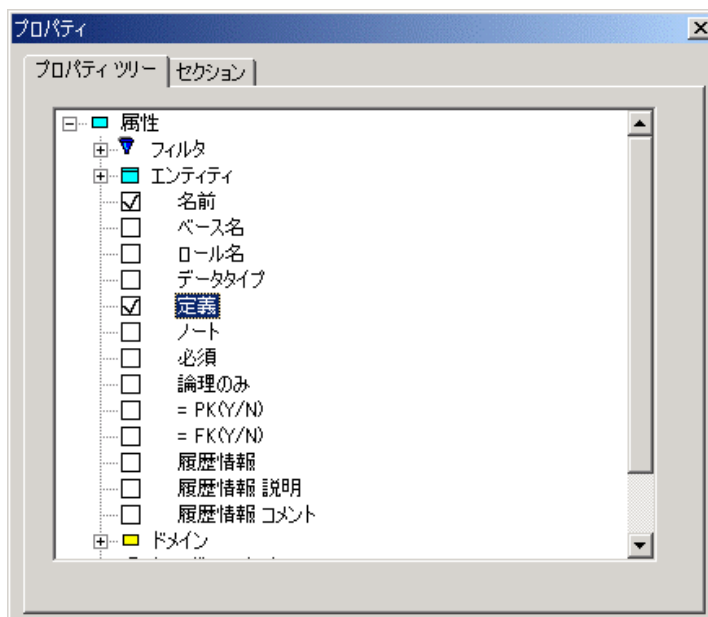
レポート テンプレートを新規作成して、エンティティと属性の情報を HTML 形式で出力します。

1. [ERwin 標準ツールバー] の [レポート テンプレート ビルダ]  をクリックするか、[ツール] メニューの [レポート ビルダ] をポイントし、[レポート ビルダ] をクリックして [レポート テンプレート] ダイアログ ボックスを開きます。
2. [新規作成] をクリックして [レポート テンプレート ビルダ] ダイアログ ボックスを開きます。
3. この [レポート テンプレート ビルダ] ダイアログ ボックスで、レポートの内容、形式、およびプロパティを指定します。
4. まず、レポート テンプレート プロパティを定義します。[レポート レイアウト] ボックスの [ドキュメント Untitled] をダブルクリックして [プロパティ] ダイアログ ボックスを開きます。
5. [プロパティ ツリー] タブをクリックし、[目次の表示] チェックボックスおよび [HTML] チェックボックスをオンにします。

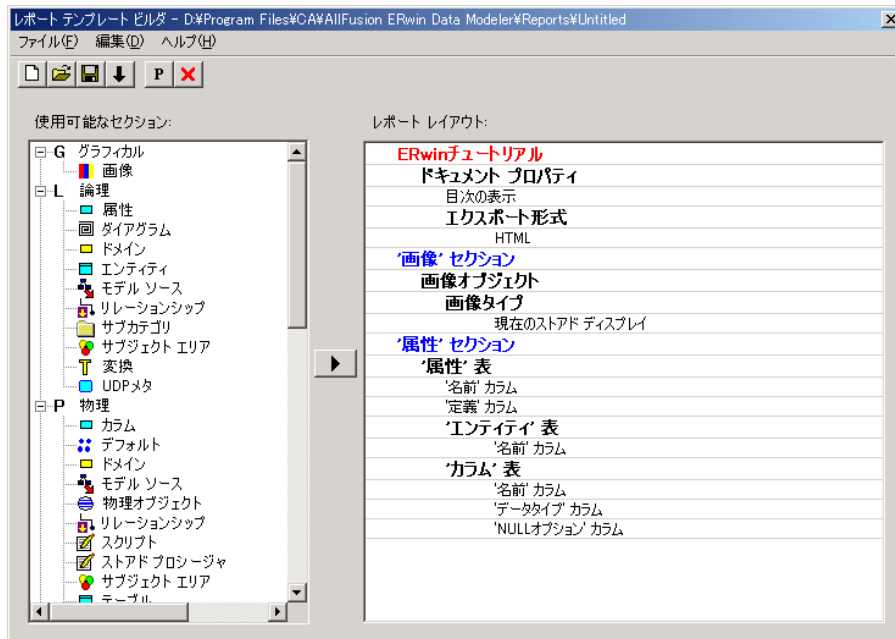



6. [タイトル] タブをクリックしてレポート タイトルを設定します。[ドキュメントのタイトル] ボックスに「ERwin チュートリアル」と入力します。閉じるボタンをクリックして [プロパティ] ダイアログ ボックスを閉じます。
7. 次に、レポート セクション、レポート カラム、およびプロパティを設定します。[使用可能なセクション] ボックスで [グラフィカル] の [画像] をクリックし、 をクリックして [レポート レイアウト] ボックスに移動します。同様に、[論理] の [属性] をクリックし、 をクリックして、[レポート レイアウト] ボックスに移動します。

8. [レポート レイアウト] ボックスの [属性'セクション] をクリックし、[アイテム プロパティ] **P** をクリックするか、[レポート レイアウト] ボックスの [属性'セクション] を右クリックし、ショートカット メニューの [プロパティ] をクリックして [プロパティ] ダイアログ ボックスを開きます。
9. [プロパティ] ダイアログ ボックスの [プロパティ ツリー] タブでレポートに含めるプロパティを設定します。[属性] のプラス記号 (+) をクリックしてツリーを展開し、[名前] チェックボックスおよび [定義] チェックボックスをオンにします。次に、[エンティティ] のプラス記号 (+) をクリックしてツリーを展開し、[名前] チェックボックスをオンにします。[カラム] のプラス記号 (+) をクリックしてツリーを展開し、[名前] チェックボックス、[データタイプ] チェックボックス、および [NULL オプション] チェックボックスをオンにします。閉じるボタンをクリックして [プロパティ] ダイアログ ボックスを閉じます。



10. [レポート テンプレート ビルダ]ダイアログ ボックスで、設定したプロパティ情報を確認します。



11. [実行]  をクリックするか、[ファイル] メニューの [実行] をクリックすると、HTML 形式のレポートが生成されます。
12. [ファイル] メニューの [名前を付けて保存] をクリックして、[レポート テンプレート ビルダ - 名前を付けて保存] ダイアログ ボックスを開きます。任意の名前をつけて作成したレポート テンプレートを保存してください。

## 完全比較

ここでは、ERwin の完全比較機能を使用して、作成した ER ダイアグラムとデータベース オブジェクトとの相違点を検出し、解決する方法について学習します。

### 説明

データベースの設計および実装の完了後、プログラム開発に移行した後で当初の設計仕様と実際のデータベース定義にギャップが生じ、データモデルが形骸化することがよくあります。ERwin が提供する完全比較機能は、このギャップを埋める先進的なメカニズムで、モデルとデータベースの間の相違点を検出して、モデルからデータベース、またはデータベースからモデルの双方向に相違点を更新できます。

完全比較機能では、データを持っているテーブルに対してデータの保持を考慮した変更スクリプトを自動生成するので、データが格納されているデータベースの定義変更も確実に行うことができます。データベースを変更するときに、いったんテーブルを削除しなければならない場合には、テーブルに格納されていたデータを新しいテーブルにコピーします。開発、テスト、および運用中のアクティブなデータベースに対して設計変更を行うときの定義変更作業を自動化できるため、作業の省力化やミスの軽減に役立ちます。

完全比較機能の比較対象には、データベース、SQL スクリプト、ER1 ファイル、および Model Manager に保存されているモデルの 4 種類があります。

#### ERwin の完全比較機能がサポートする比較対象

比較対象	特徴
データベース	ERwin がサポートするデータベースとの相違点を検出し、相違点を双方向にエクスポート、インポート、および削除します。
SQL スクリプト (DDL)	SQL スクリプト (DDL) との相違点を検出し、現在のモデルに対して相違点を削除およびインポートします。また、SQL スクリプトに対して相違点を解決するための DDL を生成します。
ER1 ファイル	ER1 ファイルとの相違点を検出し、現在のモデルに対して相違点を削除およびインポートします。
Model Manager	Model Manager に保存されているモデルとの相違点を検出し、相違点を双方向にエクスポート、インポート、および削除します。

この機能を使用すれば、データベースからリバース エンジニアリングした ER ダイアグラムと過去に作成した既存の SQL スクリプトや ER1 ファイル、および Model Manager に保存されているモデルとの差分を確認できます。差分が検出された場合は、ER ダイアグラムに差分を反映し、ER ダイアグラムとデータベースを完全比較すると、相違点をデータベースに反映できます。



---

#### Model Manager とは？

Model Manager は、ERwin 使ったチーム開発を強力に支援するモデル管理システムです。開発資産はサーバー上のリポジトリ データベースで一括管理されます。Model Manager には、チーム開発で発生する問題を解決するためのさまざまなツールが用意されており、モデリングにおける品質と生産性の向上が可能になります。



## 完全比較とは

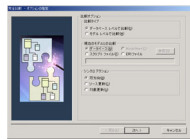
完全比較の対象には、データベース、SQL スクリプト、ER1 ファイル、および Model Manager に保存されているモデルの 4 種類があります。このチュートリアルでは、データベースとの完全比較を学習します。

Lesson 17 で、チュートリアルで作成した ER ダイアグラムをデータベースにスキーマ生成しました。現状は、ER ダイアグラムとデータベースのテーブル構造は一致しているはずですが、ER ダイアグラムに修正を加えて相違点を作り、完全比較の機能を使用して相違点を解決してみます。

### 完全比較処理のフロー

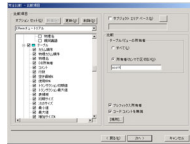
完全比較機能の処理フローを示します。

#### 1 オプションの指定



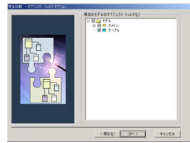
完全比較の対象を指定します。

#### 2 比較項目



対象データでサポートされている項目で、シンクロ時に比較したい項目を指定します。データベースと完全比較を行うときは、テーブルビューの所有者で、現在開いているモデルに対してフィルタをかけることができます。

#### 3 オブジェクト フィルタ オプション



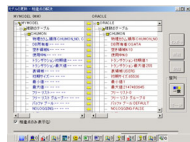
現在開いているモデルで比較対象とするオブジェクトを指定します。

#### 2 その他のモデル フィルタ オプション



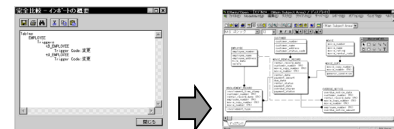
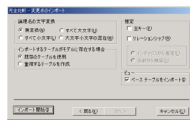
他方のモデルまたはデータベースで比較対象とするモデルオブジェクトを指定します。

#### 4 相違点の解決



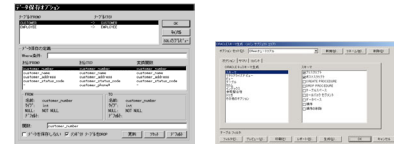
検出された相違点を ER ダイアグラムにインポートするか、データベースを更新するか、または相違点を削除するかを指定します。

#### 5 変更点のインポート



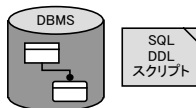
相違点をダイアグラムにインポートします。

#### 6 相違点のエクスポート



データ保存オプションやスキーマ生成オプションを指定します。

#### 7 DDL の生成



指定したオプションに従い、相違点を解決する DDL をデータベース、またはファイルに生成します。

## 完全比較の比較対象の絞り込み

通常、データベースには複数のユーザーが作成した複数のデータベース オブジェクトが存在します。ER ダイアグラムと不特定多数のデータベース オブジェクトとを比較した場合、比較対象の範囲が広すぎるために意味のない膨大な相違点がレポートされる可能性があります。この非効率的な比較を避けるために、ERwin の完全比較機能では比較対象を絞り込むことができます。たとえば、[完全比較 - 比較項目] ダイアログ ボックスでは、データベース オブジェクト タイプを指定できます。また、[完全比較 - その他のモデル フィルタ オプション] ダイアログ ボックスでは、データベース オブジェクトの所有者を指定できます。




## オプション セット

[完全比較] ダイアログ ボックスのオプション セットは階層構造になっています。階層の最上位にある [モデル] チェック ボックスをオフにすると、下階層のすべてのチェック ボックスが自動的にオフになります。[テーブル] チェック ボックスをオフにすると、その下の全階層のチェック ボックスが自動的にオフになります。すべてのチェック ボックスがオフの状態でも、下階層のチェック ボックスをオンにすると、そのチェック ボックスが依存する上位のオプションも自動的にオンになります。たとえば [カラム] チェック ボックスの [物理名] チェック ボックスをオンにすると、このチェック ボックスが依存する [モデル] チェック ボックス、[テーブル] チェック ボックス、および [カラム] チェック ボックスも自動的にオンになります。オプションの選択方法によって検出される相違点も変わります。あらかじめ最適なオプションを指定しておけば、確認したい相違点だけを検出させることができます。

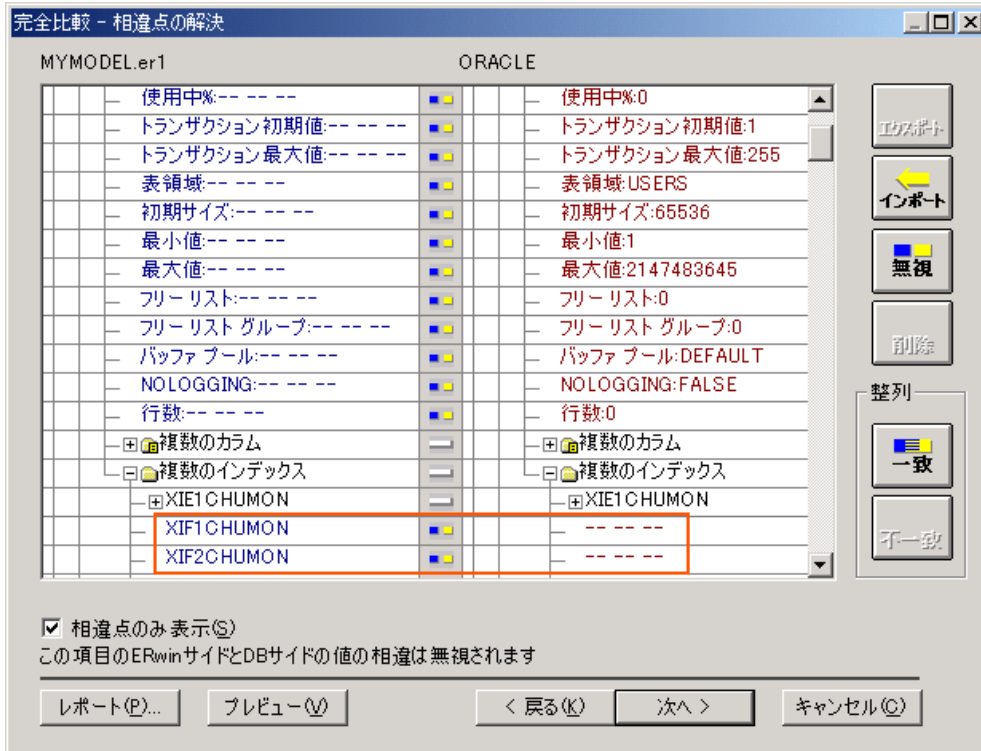
## 現状の ER ダイアグラムとデータベースとの相違点の確認

現状の ER ダイアグラムと Lesson 17 でデータベースに生成したテーブル構造は一致しているはずですが、完全比較機能を使用して、実際に一致しているかを確認してみましょう。既にデータベース テーブルを変更してしまった場合は、Lesson 17 に戻ってスキーマ生成をやり直してください。

### 相違点を確認する

1. [ツール] メニューの [完全比較] をクリックするか、[データベース] ツールバーの  をクリックして [完全比較 - オプションの指定] ダイアログ ボックスを開きます。
2. [比較タイプ] の [データベース レベルで比較] をクリックし、[現在のモデルとの比較] の [データベース] をクリックします。次に、[シンクロ アクション] の [双方向] をクリックして [次へ] をクリックします。
3. [完全比較 - 比較項目] ダイアログ ボックスで、[オプション セット] ボックスに「ERwin チュートリアル」と入力し、[新規] をクリックします。これにより、完全比較のオプション設定が「ERwin チュートリアル」という名前で保存されます。
4. [オプション セット] ボックスで、階層の最上位にある [モデル] チェック ボックスをオフにし、ドメイン情報以外のすべてのチェック ボックスをオフにします。次に、[テーブル] チェック ボックスをオンにし、[テーブル] チェック ボックス以下のすべてのチェック ボックスをオンにします。
5. 比較項目の設定を行ったら、[更新] をクリックします。
6. [テーブル/ビューの所有者] の [すべて] をクリックします。これにより、現在開いているモデルのすべてのテーブル/ビューに対して完全比較を行います。[次へ] をクリックします。
7. [完全比較 - オブジェクト フィルタ オプション] ダイアログ ボックスで、[次へ] をクリックします。
8. [完全比較 - その他のモデル フィルタ オプション] ダイアログ ボックスで、[所有者] の [所有者 (カンマで区切る)] をクリックし、下のボックスにスキーマ生成時にデータベースに接続したユーザーの名前を入力します。これにより、完全比較の対象がデータベースのオブジェクトのうち、接続したユーザーのものだけとなります。[所有者を含める] チェック ボックスをオフにします。現在の ER ダイアグラムにはテーブルの所有者情報がないため、このチェック ボックスをオンにしても効果はありません。[次へ] をクリックします。
9. [Oracle 接続] ダイアログ ボックスが表示されたら、適切な情報を入力し、[接続] をクリックします。

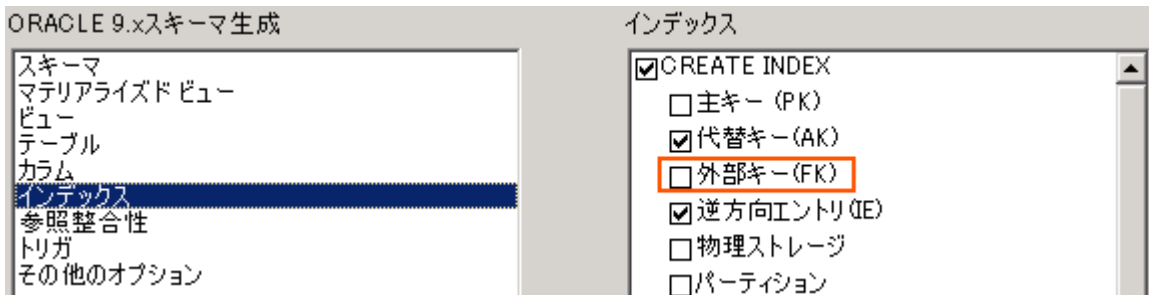
10. ER ダイアグラムとデータベースとの相違点が検出され、[完全比較 - 相違点の解決] ダイアログ ボックスが表示されます。相違点を確認したら、[キャンセル] をクリックしてダイアログ ボックスを閉じます。



11. 現在の ER ダイアグラムとデータベースには相違点がないはずなのに、何故インデックスに関する相違点が発見されたのでしょうか？

これを解く鍵は、ER ダイアグラムのスキーマをデータベースに生成したときのスキーマ生成オプションにあります。

[ORACLE スキーマ生成] ダイアログ ボックスで、[オプション セット] ボックスの一覧の [ERwin チュートリアル] をクリックすると、以下のようなスキーマ生成オプションを指定したことが確認できます。ER ダイアグラム中には、外部キー インデックスが作成されていますが、スキーマ生成時には [外部キー(FK)] チェック ボックスをオンにしなかったために相違点が発生しているのです。



12. 検出された外部キー インデックスの相違点が予期したものであることが確認できたら、[キャンセル] をクリックして [ORACLE スキーマ生成] ダイアログ ボックスを閉じます。

## ER ダイアグラムを変更して相違点を作り出す

現在の ER ダイアグラムとデータベースには相違点がないことがわかりました。ここで仕様変更が発生したと想定して、以下の表に従って ER ダイアグラムを変更してください。これにより、ER ダイアグラムとデータベースとの間に相違点を作り出して相違点を解決する手順を学習します。


### ER ダイアグラムへの変更点とデータベースへの影響






#	変更点	データベースへの影響
1	「住所」ドメインのデータ型を varchar2(40) から varchar2(50) へと変更する。	「住所」ドメインを設定した [KOKYAKU] テーブルの「kokyaku_jusho」カラムと [JUGYOIN] テーブルの「jugyoin_jusho」カラムのデータ型 varchar2 の長さが 40 から 50 へと変更される。この実現には、ALTER TABLE 文が必要である。
2	[注文] エンティティ ([CHUMON] テーブル) に「注文備考」属性 ([chumon_comment] カラム) を追加する。 論理属性名 : 注文備考 物理カラム名 : chumon_comment データ型 : varchar2(200) NULL オプション : NULL	[CHUMON] テーブルに「chumon_comment」カラムを追加する。この実現には、ALTER TABLE 文が必要である。
3	[JUGYOUIN] テーブルの「jugyoin_telno」カラムを削除する。	[JUGYOIN] テーブルから、「jugyoin_telno」カラムを削除する。この実現には、ALTER TABLE 文が必要。(なお、Oracle 8i より前のバージョンでは、Oracle の仕様で既存カラムの削除は実現できないため、テーブルを一度削除してから再作成する必要がある。)
4	[CHUMON_MEISAI] テーブルの「chumon_suuryo」カラムのデータ型を CHAR(3) から CHAR(2) へと変更する。	Oracle の仕様で桁数を短くする操作は ALTER TABLE 文では実現できない。この場合、テーブルを一度削除してから再作成する必要がある。さらに、[CHUMON_MEISAI] テーブルは [SEIHIN] テーブルと [CHUMON] テーブルを参照しているため、両テーブルからの外部キー制約の削除が前提となる。また、テーブルを削除すると既存のデータ (現在はデータは入っていません…) も同時に削除されてしまう問題が発生する。

これまでの、これらの仕様変更を実現するために、データベースへの変更を行う DDL を作成し、テーブル定義書や模造紙などに書いていた ER ダイアグラムを更新する必要がありました。ERwin を使用すれば、ERwin の ER ダイアグラムで設計内容を変更した後は、完全比較機能で相違点を検出し、検出された相違点をデータベースにエクスポートするだけです。また、テーブル定義書は更新された ER ダイアグラムを元に再度レポートを行うだけで完了です。

実際、上の表の#4 のようなテーブルの削除を伴う仕様変更が発生した場合は、既存データのバックアップを取り、テーブル間の依存関係を考慮して問題を取り除いた後でテーブルを削除し、最後に新しいテーブルに既存データを移行するスクリプトを手作業で作成する必要がありました。

では、ERwin の ER ダイアグラムに加えた変更点をデータベースにエクスポートしてみましょう。

1. [ツール] メニューの [完全比較] をクリックするか、[データベース] ツールバーの  をクリックして [完全比較 - オプションの指定] ダイアログ ボックスを開きます。
2. [比較タイプ] の [データベース レベルで比較] をクリックし、[現在のモデルとの比較] の [データベース] をクリックします。次に、[シンクロ アクション] の [双方向] をクリックします。[次へ] をクリックします。
3. [オプション セット] ボックスの一覧の [ERwin チュートリアル] をクリックします。これにより、前回と同じ比較設定で比較が行われます。[ドメイン] チェック ボックスと [テーブル] チェック ボックスだけが、オンになっていることを確認します。
4. [テーブル/ビューの所有者] の [すべて] をクリックします。[次へ] をクリックします。
5. [完全比較 - オブジェクト フィルタ オプション] ダイアログ ボックスで、[次へ] をクリックします。
6. [完全比較 - その他のモデル フィルタ オプション] ダイアログ ボックスでは、[所有者] の [所有者 (カンマで区切る)] をクリックし、下のボックスにスキーマ生成時にデータベースに接続したユーザーの名前を入力します。[所有者を含める] チェック ボックスをオフにします。[次へ] をクリックします。
7. [Oracle 接続] ダイアログ ボックスが表示されたら、適切な情報を入力し、[接続] をクリックします。
8. ER ダイアグラムとデータベースとの相違点が検出され、[完全比較 - 相違点の解決] ダイアログ ボックスが表示されます。
9. まず、ダイアログ ボックス全体を見てください。ダイアログ ボックス左半分に ERwin の ER ダイアグラム情報が表示され、右半分に完全比較の対象となるデータベース情報が表示されます。ダイアログ ボックスの最上段を見ると左側に比較の元となる現在の ER ダイアグラムのファイル名が、右側に比較対象となるデータベース (このチュートリアルでは Oracle) が表示されます。ダイアログ ボックス左下の [相違点のみ表示] チェック ボックスがデフォルトではオンになっています。これにより、両者の相違点だけが表示されます。このチェック ボックスをオフにしてください。完全比較のオプション セットで選択したオプションに従い一致している項目も表示されます。一致している項目は淡色で表示され、相違点は色付きで表示されていることが確認できます。
10. 次に、検出された相違点を見てください。ER ダイアグラム上で行った変更によってデータベースとの間に相違が生じ、ERwin が相違点を検出したことがわかります。
11. では、[KOKYAKU] テーブルの [複数のカラム] を展開してください。この相違点は 113 ページの #1 でドメインの桁数を変更したことにより発生した、既存カラムの桁数の変更 ( [KOKYAKU] テーブルの「kokyaku\_jusho」カラム) が相違点として表示されています。この相違点をデータベースにエクスポートしますので、相違点の行をマウスでクリックしてください。これにより、ダイアログ ボックス右側の [エクスポート] が有効になりますので、[エクスポート] をクリックします。ダイアログ ボックスを左右に分割していたバーに青い右矢印が表示され、相違点をエクスポートすることが確認できます。同様に [JUGYOIN] テーブルの [複数のカラム] を展開し、「jugyoin\_jusho」カラムの相違点もエクスポートを設定します。

<input type="checkbox"/> 複数のカラム		<input type="checkbox"/> 複数のカラム
<input checked="" type="checkbox"/> kokyaku_no		<input checked="" type="checkbox"/> KOKYAKU_NO
<input checked="" type="checkbox"/> kokyaku_name		<input checked="" type="checkbox"/> KOKYAKU_NAME
<input type="checkbox"/> kokyaku_jusho		<input type="checkbox"/> KOKYAKU_JUSHO
データタイプ: VARCHAR2 (50)		データタイプ: VARCHAR2 (40)

12. [CHUMON] テーブルの [複数のカラム] を展開してください。#2 で追加した新規カラム（[CHUMON] テーブルの「chumon\_comment」カラム）が相違点として表示されています。この相違点も [エクスポート] をクリックしてデータベースにエクスポートします。

元のデータベース	相違点	ターゲットデータベース
[-] 複数のカラム	[>]	[-] 複数のカラム
[+] chumon_no	[<]	[+] CHUMON_NO
[+] kokyaku_no (FK)	[<]	[+] KOKYAKU_NO (FK)
[+] tantou_eigyō_no (FK)	[<]	[+] TANTOU_EIGYO_NO (FK)
[+] chumon_status	[<]	[+] CHUMON_STATUS
[+] chumon_comment	[>]	[ ] -----

13. [CHUMON\_MEISAI] テーブルの [複数のカラム] を展開してください。#4 で行った「chumon\_suuryo」カラムの桁数の変更が相違点として表示されています。この相違点も [エクスポート] をクリックしてデータベースにエクスポートします。すべての相違点の解決方法を指定したら、ダイアログ ボックス右下の [次へ] をクリックします。

元のデータベース	相違点	ターゲットデータベース
[-] 複数のカラム	[>]	[-] 複数のカラム
[+] chumon_no (FK)	[<]	[+] CHUMON_NO (FK)
[+] seihin_no (FK)	[<]	[+] SEIHIN_NO (FK)
[+] chumon_suuryo	[>]	[+] CHUMON_SUURYO
データタイプ:CHAR(2)	[>]	データタイプ:CHAR(3)

14. [JUGYOIN] テーブルの [複数のカラム] を展開してください。#3 で行った「jugyoin\_telno」カラムの削除が相違点として表示されています。このカラム削除に関する相違点は ER ダイアグラムへの誤った操作と想定して、データベース上のカラムを ER ダイアグラムにインポートします。ダイアログ ボックス右側の [インポート] をクリックすると、ダイアログ ボックスを左右に分割していたバーに黄色い左矢印が表示され、相違点をインポートを確認できます。

元のデータベース	相違点	ターゲットデータベース
[-] 複数のカラム	[<]	[-] 複数のカラム
[+] jugyoin_no	[<]	[+] JUGYOIN_NO
[+] jugyoin_namae	[<]	[+] JUGYOIN_NAMAE
[+] jugyoin_jusho	[<]	[+] JUGYOIN_JUSHO
[ ] -----	[<]	[+] JUGYOIN_TELNO

15. [次へ] をクリックして [完全比較 - 変更点のインポート] ダイアログ ボックスを開きます。[インポート開始] をクリックしてインポートを実行します。

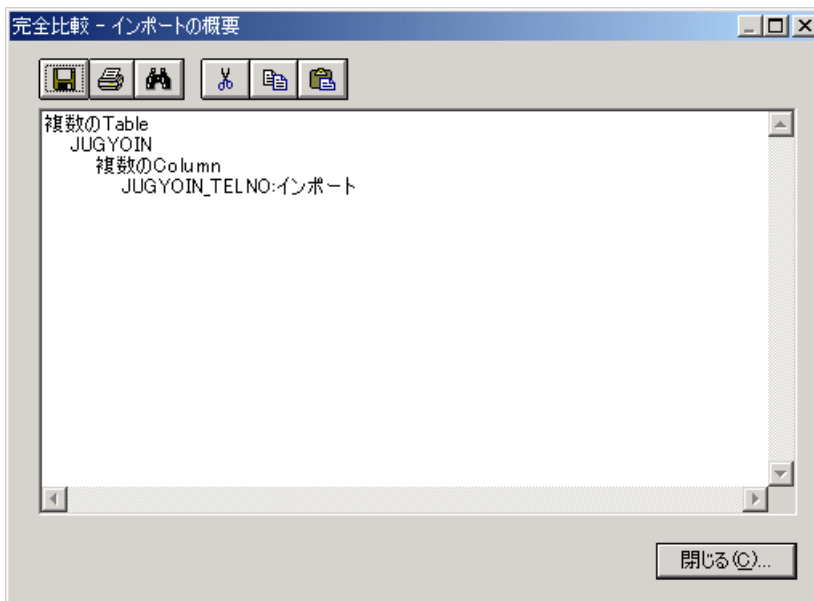
完全比較 - 変更点のインポート

<p>論理名の文字変換</p> <p><input checked="" type="radio"/> 無変換(N)    <input type="radio"/> すべて大文字(U)</p> <p><input type="radio"/> すべて小文字(L)    <input type="radio"/> 大文字小文字の混在(M)</p>	<p>推定</p> <p><input type="checkbox"/> 主キー(E)</p> <p><input type="checkbox"/> リレーションシップ(R)</p> <p><input type="radio"/> インデックスから推定(F)</p> <p><input type="radio"/> 名前から推定(Q)</p>
<p>インポートするテーブルがモデルに存在する場合</p> <p><input checked="" type="radio"/> 既存のテーブルを使用</p> <p><input type="radio"/> 重複するテーブルを作成</p>	<p>ビュー</p> <p><input checked="" type="checkbox"/> ベース テーブルをインポート(F)</p>
<p>[インポート開始(F)]    &lt; 戻る(K)    次へ &gt;    キャンセル(Q)</p>	



[推定] の [リレーションシップ] チェック ボックス外部キー制約のような宣言型データ整合性を使用せずに、トリガのような手続き型データ整合性を使用して作成されたデータベースからリバース エンジニアリングを行った場合、リバース エンジニアリングしてきた ER ダイアグラムにはリレーションシップがありません。このような場合、[推定] の [リレーションシップ] チェック ボックスを使用します。[インデックスから推定] をクリックすると、ERwin はカラムに設定されたインデックス情報を元にリレーションシップを推定します。[名前から推定] をクリックすると、主キー カラムの名前を元にリレーションシップを推定します。これにより、データベースの隠れた参照整合を ER ダイアグラムにリレーションシップとして表示できます。

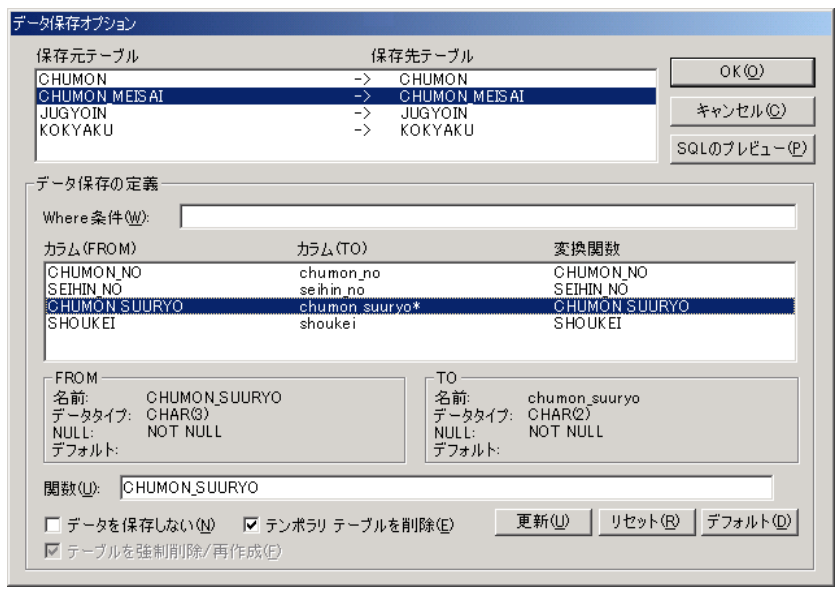
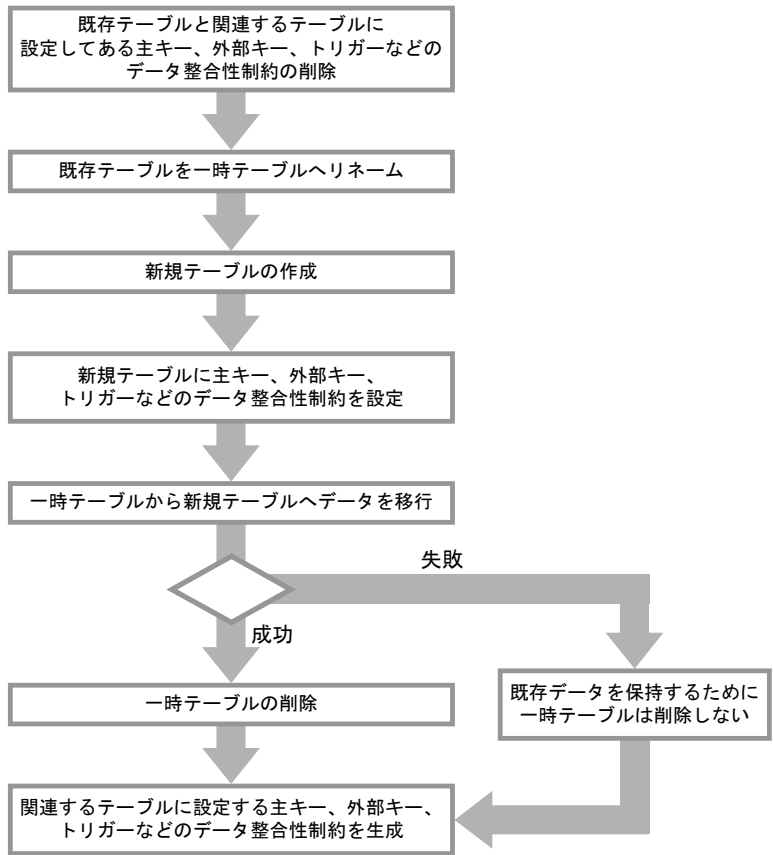
16. インポートが完了して [次へ] が有効になったら、[結果の表示] をクリックしてください。インポート結果が以下のように階層構造で表示され、[JUGYOIN] テーブルに「JUGYOIN\_TELNO」カラムがインポートされたことが確認できます。[閉じる] をクリックし [完全比較 - 変更点のインポート] ダイアログ ボックスに戻ります。後ろの ER ダイアグラムが見えるように、いったんダイアログ ボックスを横にずらしてください。[JUGYOUIN] テーブルに「JUGYOIN\_TELNO」カラムがインポートされたことが確認できたでしょうか？ [次へ] をクリックします。







ERwin のデータ移行フロー



20. 最後に、[完全比較 - 変更点のエクスポート] ダイアログ ボックスの右上にある [エクスポート開始] をクリックしてデータベースの変更スクリプトを生成します。すべてのスクリプトが問題なく完了したら、ダイアログ ボックス下段の [完了] をクリックして [完全比較 - 変更点のエクスポート] ダイアログ ボックスを閉じます。

## リバース エンジニアリング

ここでは、物理データベース、または SQL スクリプト (DDL) からデータモデルを逆生成する機能について学習します。

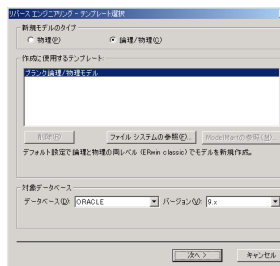
この機能は、データモデルから物理データベース スキーマを生成するフォワード エンジニアリングに対して「リバース エンジニアリング」と呼びます。なお、ここでは今まで作成してきた MYMODEL.er1 は使用しません。

### リバース エンジニアリングとは

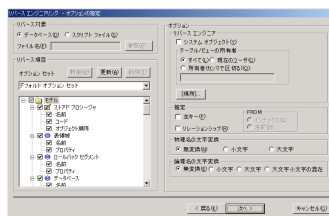
ERwin では、物理データベースまたは SQL スクリプト (DDL) からリバース エンジニアリングを行うことができます。データモデルを逆生成するためのソースが異なっても、新規の ER ダイアグラムにデータモデルを逆生成する場合は、共通して「リバース エンジニアリング」と呼びます。リバース エンジニアリングと似た機能にインポートがあります。インポートとは ERwin で開いている ER ダイアグラムとデータベース、ER1 ファイル、SQL スクリプト (DDL)、および Model Manager に格納されている ER ダイアグラムとをシンクロナイズして、シンクロナイズ先からオブジェクトを読み込む機能です。

### リバース エンジニアリング機能の処理フロー

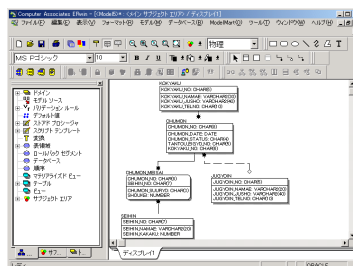
#### 1 新規モデルのタイプと対象データベースの指定



#### 2 オプションの指定



#### 3 新規 ER ダイアグラムにデータモデルの逆生成



リバース エンジニアリングして逆生成するモデルのタイプを選択します。また、リバース エンジニアリング対象のデータベースを指定します。対象が SQL スクリプトの場合は、その SQL スクリプトがどのデータベースのシンタックスで定義されているかを指定します。

リバース エンジニアリング対象のオブジェクト タイプを指定します。対象がデータベースの場合は、オブジェクトの所有者 (オーナー) でフィルタをかけることができます。

指定したオプションに従い、リバース エンジニアリングしたオブジェクト情報をグラフィカルなデータモデルで表示します。

## データベースからのリバース エンジニアリング

テーブル、カラム、インデックス、リレーションシップ、トリガ、ストアド プロシージャ、物理ストレージなどのデータベースの情報を取り込み、これに基づいて新規の ER ダイアグラムにデータモデルを逆生成します。この機能を使用すれば、正確なドキュメントが残っていない運用中のデータベースからリバース エンジニアリングを行い、データベースの構造をグラフィカルなデータモデルで表現し、現行データベース システムの正確なドキュメントを作成することもできます。また現行システムを更新する必要がある場合は、リバース エンジニアリングした ER ダイアグラムに変更内容を定義し、完全比較機能で変更点をデータベースにエクスポートするだけで現行システムを再設計できます。

データベースからリバース エンジニアリングを行うときは、[リバース エンジニアリング] ダイアログ ボックスの [オプション セット] ボックスで、データベース オブジェクト単位 (テーブルやストアド プロシージャなど) でリバース エンジニアリング対象を指定します。各データベース オブジェクトの中の特定のデータベース オブジェクトを指定したいときは (テーブルというデータベース オブジェクトから [EMP] テーブルだけをリバースする) 、インポート機能をお使いください。

### データベースからリバース エンジニアリングを行う

データベースからリバース エンジニアリングを行って、ER ダイアグラムに読み込むことができる情報は、接続ユーザーがデータベースに対して持っている権限に依存します。データベースのすべてのシステム カタログに対して SELECT 権限を持っていないユーザーを使用した場合、データベースからエラーが返ってくる場合があります。

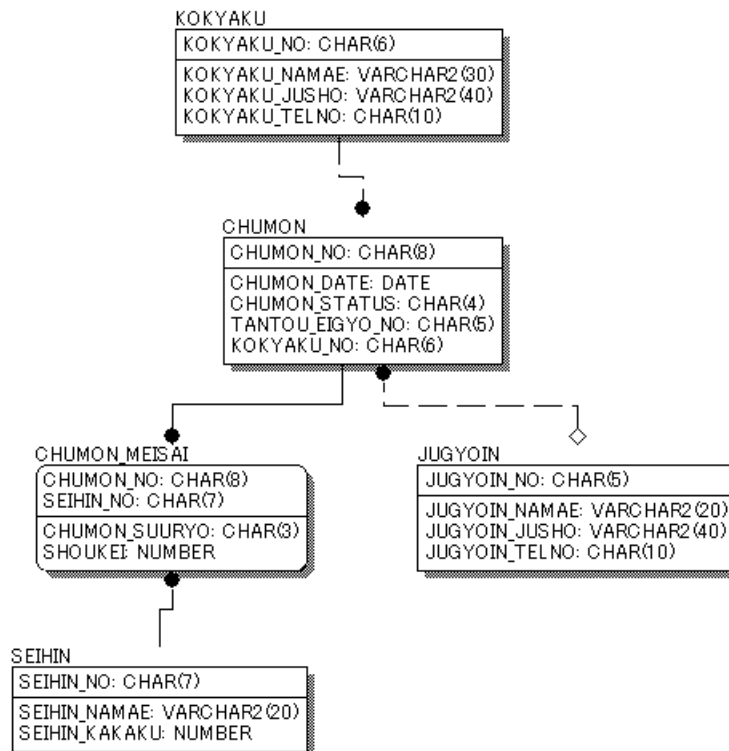
1. [ツール] メニューの [リバース エンジニアリング] をクリックし、[リバース エンジニアリング - テンプレート選択] ダイアログ ボックスを開きます。[新規モデルのタイプ] でリバース エンジニアリングして作成するモデルの種類を指定します。ここでは、[論理/物理] をクリックします。接続可能なデータベースとバージョンを [対象データベース] で指定して、[次へ] をクリックします。
2. [リバース エンジニアリング - オプションの指定] ダイアログ ボックスが開いたら、ダイアログ ボックス左上の [リバース対象] の [データベース] をクリックします。これにより、リバース エンジニアリング対象がデータベースとなります。
3. ダイアログ ボックス右上の [テーブル/ビューの所有者] の [現在のユーザー] をクリックします。これにより、[Oracle 接続] ダイアログ ボックスの [ユーザー名] ボックスに入力した所有者で、フィルタを掛けてリバース エンジニアリングを行います。
4. ダイアログ ボックス右側中段の [推定] の [主キー] チェック ボックスと [リレーションシップ] チェック ボックスをオフにします。データベース側に実存する主キー制約だけが主キーとして、また外部キー制約だけがリレーションシップとしてリバース エンジニアリングされます。
5. ダイアログ ボックス右側下段の [物理名の文字変換] と [論理名の文字変換] は、両方とも [無変換] をクリックします。これにより、データベースに登録されている文字でデータベース オブジェクトをリバース エンジニアリングされます。
6. ダイアログ ボックス左側の [オプション セット] ボックスで、すべてのオプションがチェックされていることを確認したら、ダイアログ ボックス右下の [次へ] をクリックします。[Oracle 接続] ダイアログ ボックスが表示されますので、適切な情報を入力して [接続] をクリックしてください。

7. 接続ユーザーが権限を持っていないデータベースのカatalog テーブルに検索をかけた場合、以下のようなエラーが表示されます（このエラーは ORACLE に SCOTT で接続した場合のサンプルです。エラー メッセージは接続ユーザーの持つ権限や接続データベースによって異なります）。



これらのエラーは、メッセージに表示されたカatalog テーブルに対する適切な権限をユーザーに与えることにより解決します。

8. リバース エンジニアリングが完了すると、以下のような ER ダイアグラムが表示されます。リバース エンジニアリングされたデータモデルは接続するデータベースやユーザーにより異なります。



9. リバース エンジニアリングされたテーブルを右クリックし、ショートカット メニューの [テーブル プロパティ] をポイントして [物理プロパティ] をクリックしてください。以下のようなダイアログ ボックスが表示され、ERwin がデータベースからテーブルのストレージ情報をリバース エンジニアリングしてきたことがわかります。

The screenshot shows the 'ORACLE テーブル' dialog box with the following details:

- テーブル名: CHUMON
- 名前: CHUMON
- 所有者: OGATA
- タブ: コメント | 容量計算 | **物理プロパティ** | インデックスによる構成 | パーティション | UDP | 履歴
- 表領域: USERS
- 空領域: 10% | 使用中: 0% | トランザクション初期値: 1 | トランザクション最大値: 255
- ストレージ:
  - 初期サイズ: 65536 | 最小値: 1 | フリーリスト値: 0
  - 次のサイズ: | 最大値: 214748 | フリーグループ値: 0
  - 増加サイズ: |
  - バッファプール値:  維持  リサイクル  デフォルト
- パラレルタイプ:  値: |  REDOLOG |  LOGGING  NOLOGGING
- 物理のみ:  | 生成:  | DBシンク: | OK | キャンセル

10. リバース エンジニアリングしてきた ER ダイアグラムは、この後のチュートリアルでは使用しません。余裕がある方は以下の練習を行ってみてください。

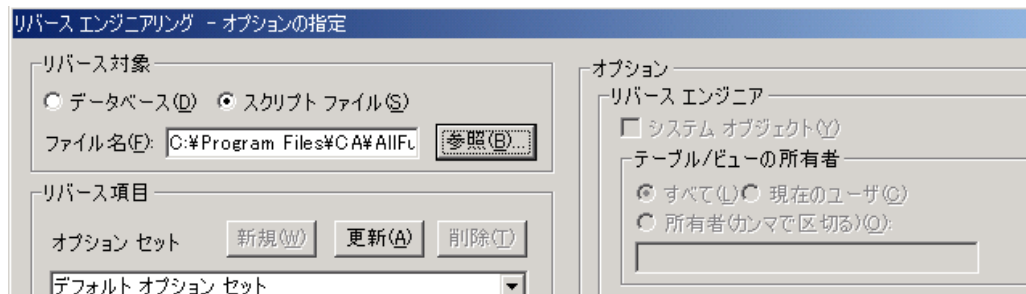
- Lesson 19 「レポート作成」を参考にして、テーブル定義書を作成する。
- Lesson 20 「完全比較」を参考にして、データモデルへの変更をデータベースに反映させる。

#### ER ダイアグラムへの変更例：

- テーブルやカラムの名前を変更する。
- カラムのデータ型を変更する。
- 新規にテーブルやカラムなどを追加する。

## SQL スクリプト (DDL) からのリバース エンジニアリング

リバース エンジニアリング対象がデータベースから SQL スクリプトに変わること以外、機能的な違いはほとんどありません。オペレーションもほぼ同じで、[リバース エンジニアリング - オプションの指定] ダイアログ ボックスで [リバース対象] の [スクリプト ファイル] をクリックしてファイルの保存場所を完全パスで指定します。



データベースからのリバース エンジニアリングと異なる点は、テーブルの所有者によるフィルタ機能が無効になっている点です。他のオペレーションは、データベースからのリバース エンジニアリングと同様ですので、「データベースからリバース エンジニアリングを行う」を参考にして実際にリバース エンジニアリングを行ってみてください。

既存の SQL スクリプトで ERwin にリバース エンジニアリングしたいものがあれば積極的に使用してください。適当な SQL スクリプトをお持ちでない方は、チュートリアルで作成した ER ダイアグラムから SQL スクリプトを生成して利用してみてください。

### データベース スキーマを SQL スクリプトに保存する

1. [ツール] メニューの [フォワード エンジニアリング/スキーマ生成] をクリックして [ORACLE スキーマ生成] ダイアログ ボックスを開きます。
2. SQL スクリプトで定義したいオプションのチェック ボックスをオンにして、[レポート] をクリックします。[生成 ORACLE/SQL スキーマ レポート] ダイアログ ボックスが開いたら、SQL スクリプトの保存場所とファイル名を指定して [保存] をクリックします。

## インポート機能

リバース エンジニアリング機能のメリットは、特定のデータベース ユーザーの特定のオブジェクトタイプを一括して ER ダイアグラムに読み込める点ですが、データベース オブジェクトを個別に指定して柔軟にリバース エンジニアリングすることはできません。これを補完する機能がインポートとなります。インポート機能の対象には、データベース、ER1 ファイル、SQL スクリプト (DDL)、および Model Manager に格納されている ER ダイアグラムがあり、各リバース エンジニアリング対象からオブジェクトを個別に指定して読み込むことができます。

### インポート機能

インポート方法	オペレーションと特徴
完全比較	[ツール] - [完全比較] Lesson 20 で学習したとおり、検出した相違点を双方向にインポート、エクスポート、および削除する最も柔軟な機能
モデル ソースとシンク □	[ツール] - [モデル ソースとシンクロ] [モデル ソースの追加] や [新規モデルの導出] により作成したソース モデルとターゲット モデルを比較し、検出した相違点を双方向にインポート、エクスポート、および削除する機能



☆ **MEMO** ☆