2010

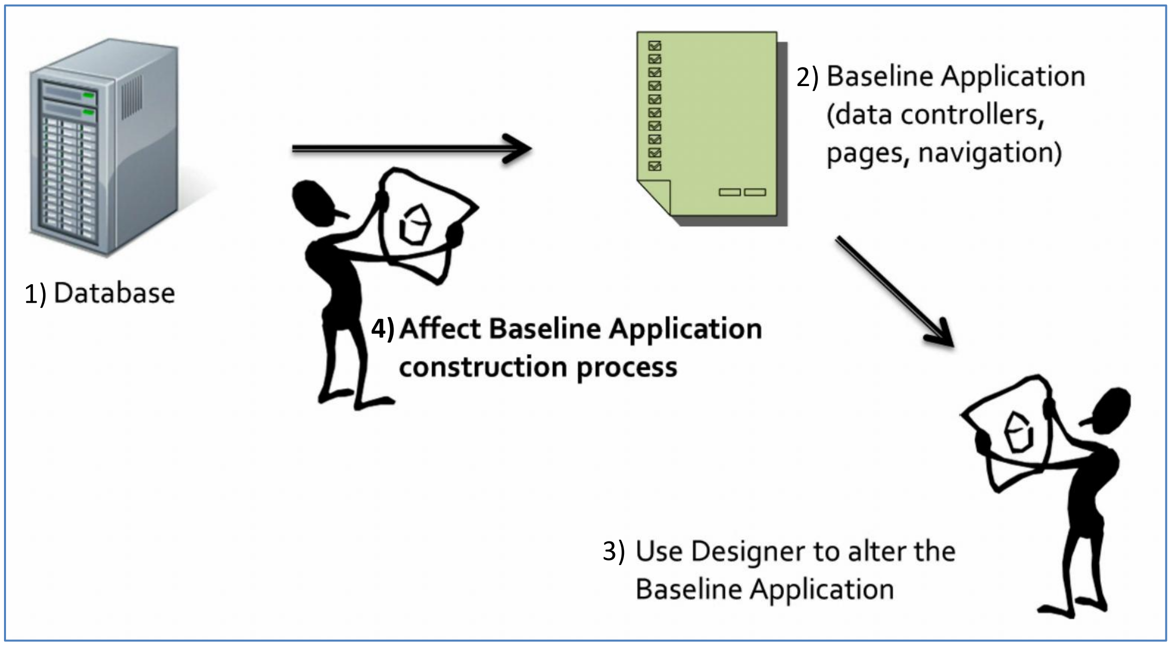
USER GUIDE: CHAPTER 1
PROJECT WIZARD

Business Logic Layer – Part 2

Application Construction Process

First, you start off with a database. Specify the connection string to your database, and Code On Time Generator will generate a baseline application, including data controllers, pages, and navigation. From there, you can use the Designer to customize the application to fit your needs.

You can also affect the baseline application construction process, so that you can spend less time designing.


1) Database
2) Baseline Application (data controllers, pages, navigation)
3) Use Designer to alter the Baseline Application
4) Affect Baseline Application construction process

Code On Time Generator creates all of the necessary infrastructure and pages for a baseline application, based on the supplied database. Multiple variables, such as virtual foreign keys and hidden fields, allow a high degree of control over the baseline application construction process.

Affecting the Construction Process

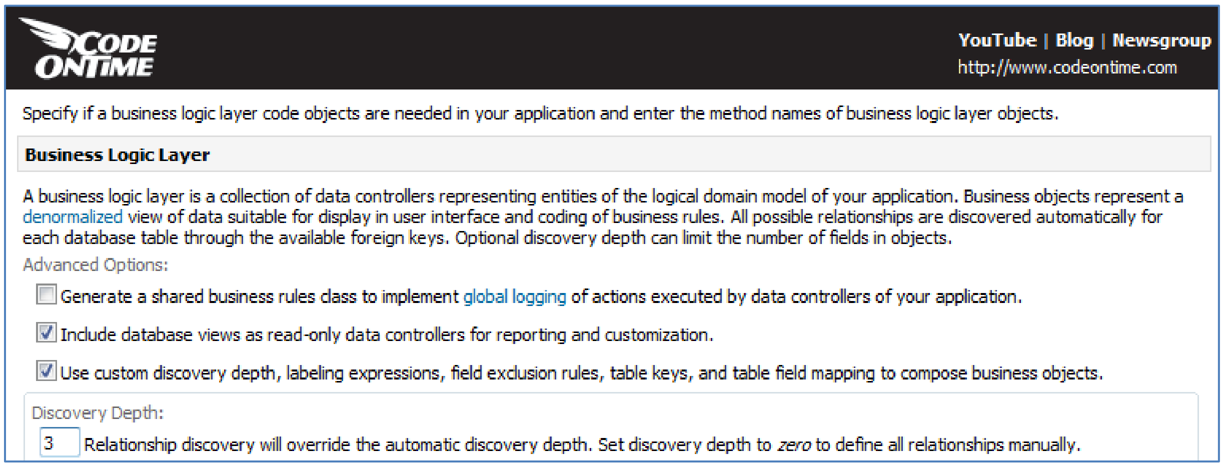
There are many ways that you can alter the baseline application, including the following:

- Declaring virtual foreign keys
- Altering automatically created pages
- Create new pages
- Customize the automatically created navigation menu
- Customize the generated data controllers
- Integrate existing applications into the new application

There are many settings you can change in the Project Wizard.

Discovery Depth

In Code On Time applications, master tables will contribute fields to the detail data controllers. The option *Discovery Depth* will allow you to define how many relationships will be looked into for field contribution. The default value is three levels up.



Specify if a business logic layer code objects are needed in your application and enter the method names of business logic layer objects.

Business Logic Layer

A business logic layer is a collection of data controllers representing entities of the logical domain model of your application. Business objects represent a denormalized view of data suitable for display in user interface and coding of business rules. All possible relationships are discovered automatically for each database table through the available foreign keys. Optional discovery depth can limit the number of fields in objects.

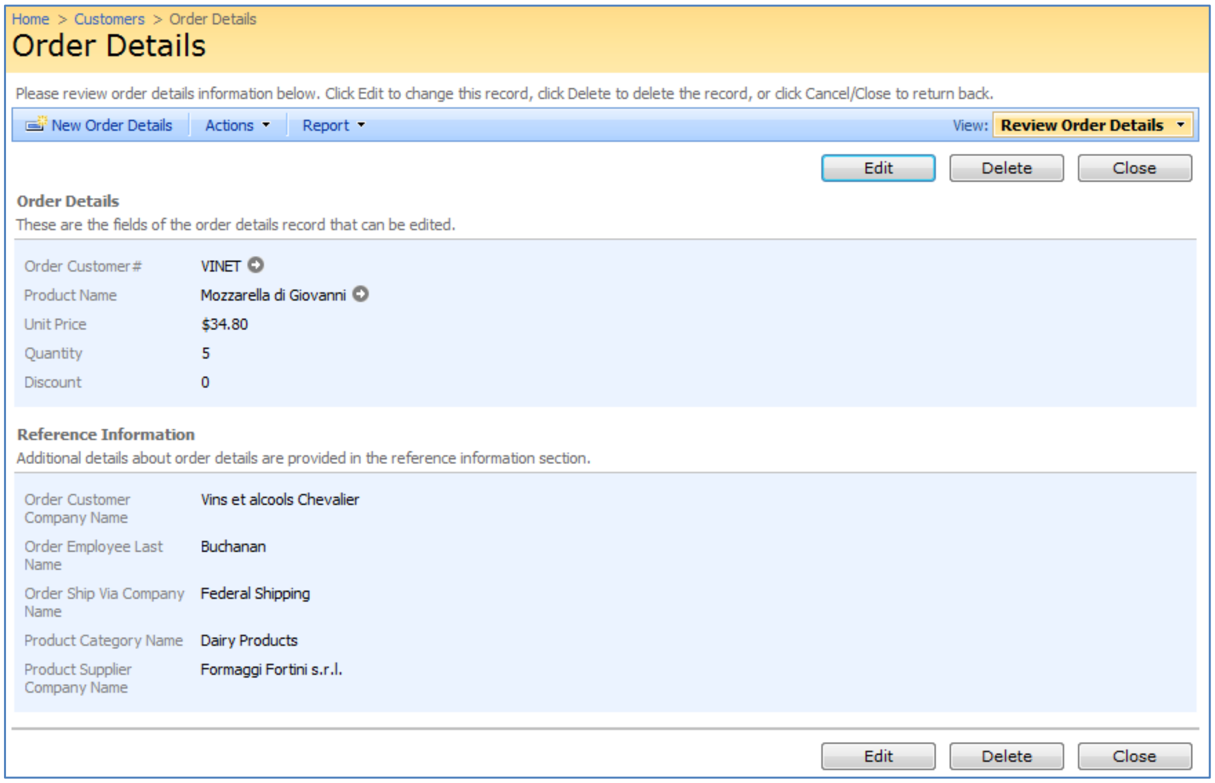
Advanced Options:

- [ ] Generate a shared business rules class to implement global logging of actions executed by data controllers of your application.
- [x] Include database views as read-only data controllers for reporting and customization.
- [x] Use custom discovery depth, labeling expressions, field exclusion rules, table keys, and table field mapping to compose business objects.

Discovery Depth:

3 Relationship discovery will override the automatic discovery depth. Set discovery depth to *zero* to define all relationships manually.

In the Northwind sample database, the *Suppliers* table is two relationships away from *Order Details*. With the default discovery depth of three, several fields from *Suppliers* are contributed to the *Order Details* page, including *Ship Via Company Name*, *Order Customer Company Name*, and *Product Category Name*.



Order Details

Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Order Details

These are the fields of the order details record that can be edited.

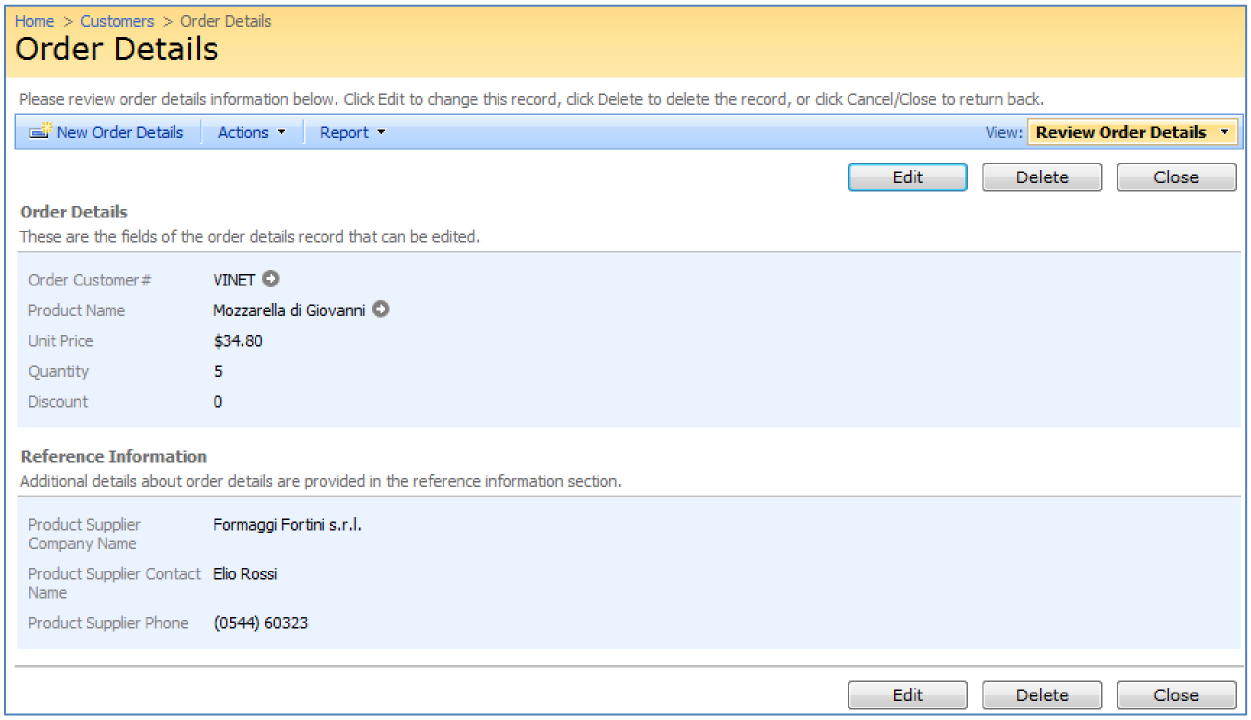
Order Customer#	VINET
Product Name	Mozzarella di Giovanni
Unit Price	$34.80
Quantity	5
Discount	0

Reference Information

Additional details about order details are provided in the reference information section.

Order Customer Company Name	Vins et alcools Chevalier
Order Employee Last Name	Buchanan
Order Ship Via Company Name	Federal Shipping
Product Category Name	Dairy Products
Product Supplier Company Name	Formaggi Fortini s.r.l.

If we change the discovery depth to one, then the relationship to *Suppliers* table will not be followed during the generation, and the fields will not be present in the *Order Details* page, as you can see in the picture below.



Order Details

Please review order details information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Order Details

These are the fields of the order details record that can be edited.

Order Customer#	VINET
Product Name	Mozzarella di Giovanni
Unit Price	$34.80
Quantity	5
Discount	0

Reference Information

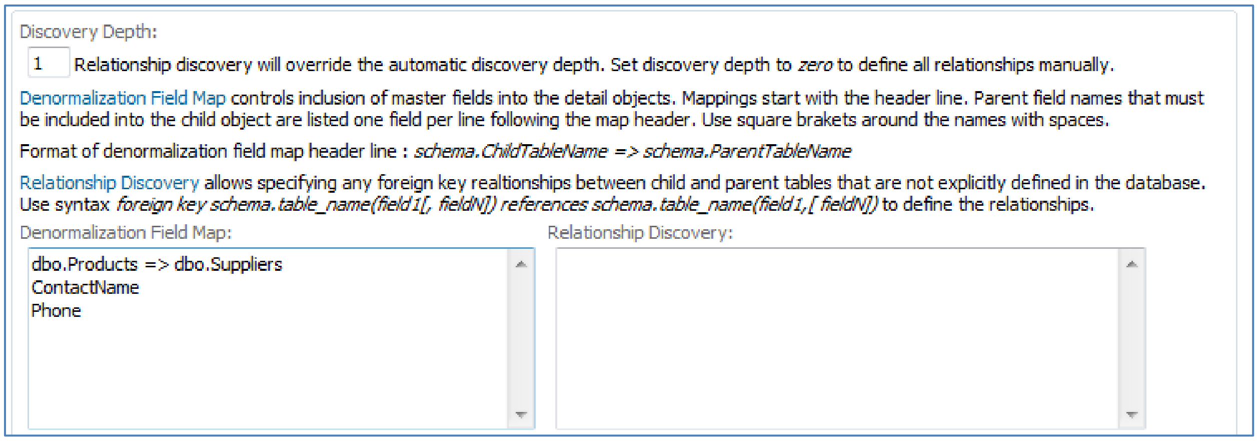
Additional details about order details are provided in the reference information section.

Product Supplier Company Name	Formaggi Fortini s.r.l.
Product Supplier Contact Name	Elio Rossi
Product Supplier Phone	(0544) 60323

De-Normalization Field Map

By default, only one field from each master table will contribute to the child data controller. You have the option of explicitly defining which fields you would like to be contributed into the child data controller.

Let's specify a few fields to copy from the *Suppliers* table into the *Products* page. These fields will be *ContactName* and *Phone*. The correct text is displayed in the image below.



Discovery Depth:

1 Relationship discovery will override the automatic discovery depth. Set discovery depth to *zero* to define all relationships manually.

Denormalization Field Map controls inclusion of master fields into the detail objects. Mappings start with the header line. Parent field names that must be included into the child object are listed one field per line following the map header. Use square brakets around the names with spaces.

Format of denormalization field map header line : *schema.ChildTableName => schema.ParentTableName*

Relationship Discovery allows specifying any foreign key realtionships between child and parent tables that are not explicitly defined in the database. Use syntax *foreign key schema.table_name(field1[, fieldN]) references schema.table_name(field1,[fieldN])* to define the relationships.

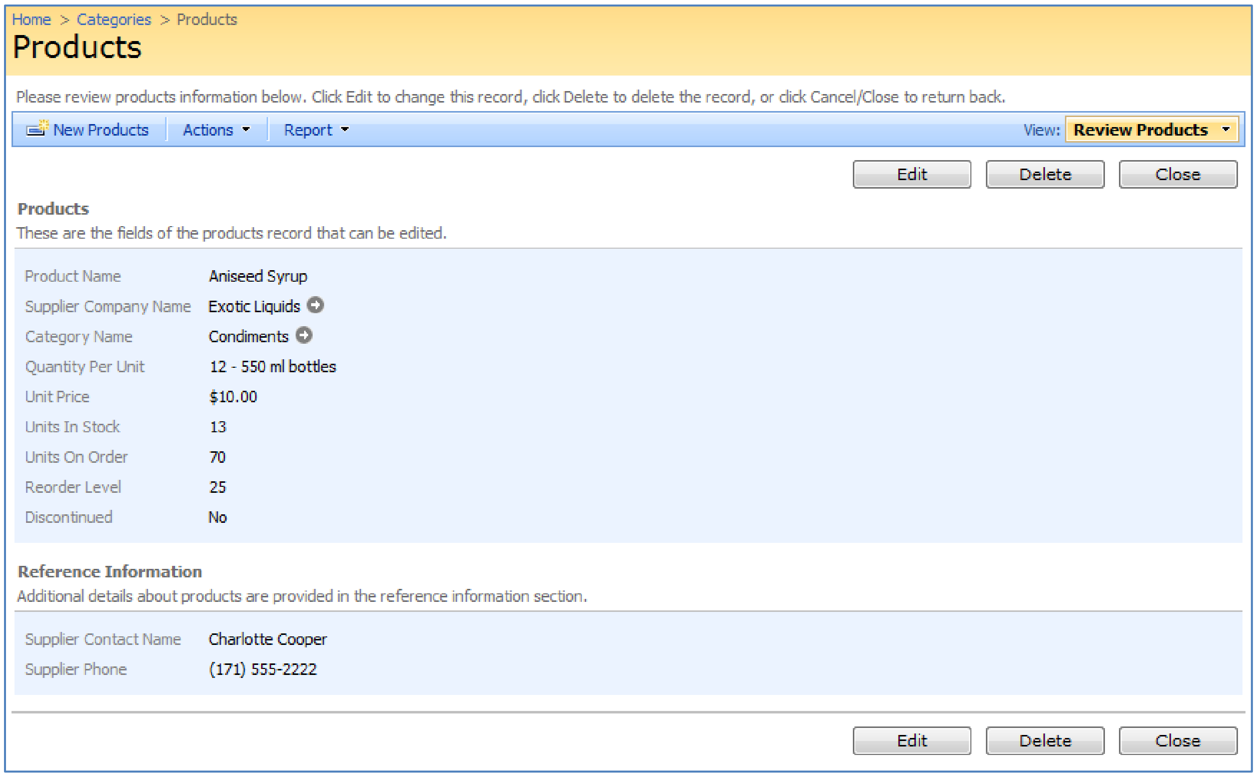
Denormalization Field Map:

```
dbo.Products => dbo.Suppliers
ContactName
Phone
```

Relationship Discovery:

Now, generate the application.

When it finishes, navigate to the *Products* page. You can see that the fields are listed under Reference Information for each record.



Products

Please review products information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Products
These are the fields of the products record that can be edited.

Product Name	Aniseed Syrup
Supplier Company Name	Exotic Liquids
Category Name	Condiments
Quantity Per Unit	12 - 550 ml bottles
Unit Price	$10.00
Units In Stock	13
Units On Order	70
Reorder Level	25
Discontinued	No

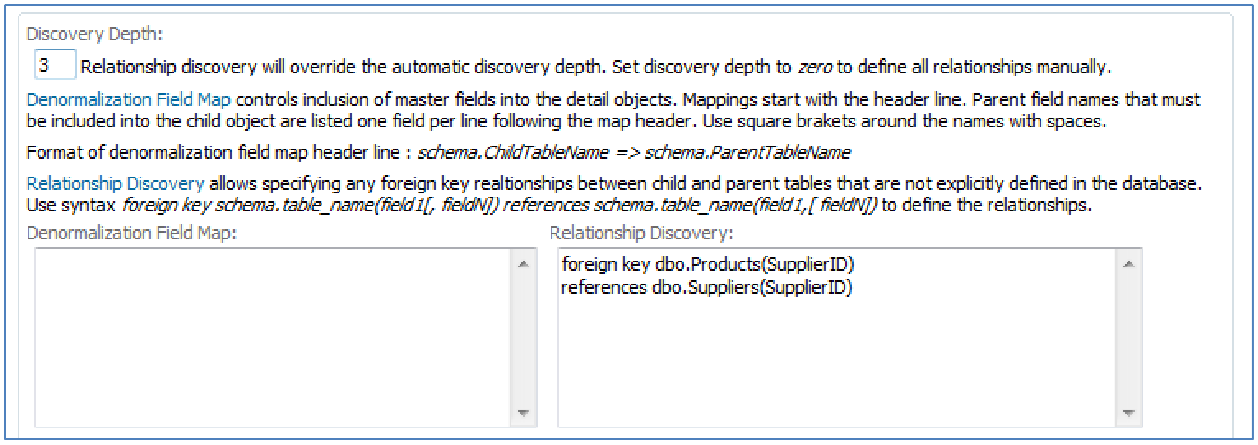
Reference Information
Additional details about products are provided in the reference information section.

Supplier Contact Name	Charlotte Cooper
Supplier Phone	(171) 555-2222

Relationship Discovery

These may dramatically change the generation of your application.

Let's put in a virtual foreign key into *Products* from the *Suppliers* table. This will insert a reference to the *SupplierID* field in the Products page. The proper text is shown in the picture below.



Discovery Depth:
3 Relationship discovery will override the automatic discovery depth. Set discovery depth to *zero* to define all relationships manually.

Denormalization Field Map controls inclusion of master fields into the detail objects. Mappings start with the header line. Parent field names that must be included into the child object are listed one field per line following the map header. Use square brakets around the names with spaces.

Format of denormalization field map header line : *schema.ChildTableName => schema.ParentTableName*

Relationship Discovery allows specifying any foreign key realtionships between child and parent tables that are not explicitly defined in the database. Use syntax *foreign key schema.table_name(field1[, fieldN]) references schema.table_name(field1,[fieldN])* to define the relationships.

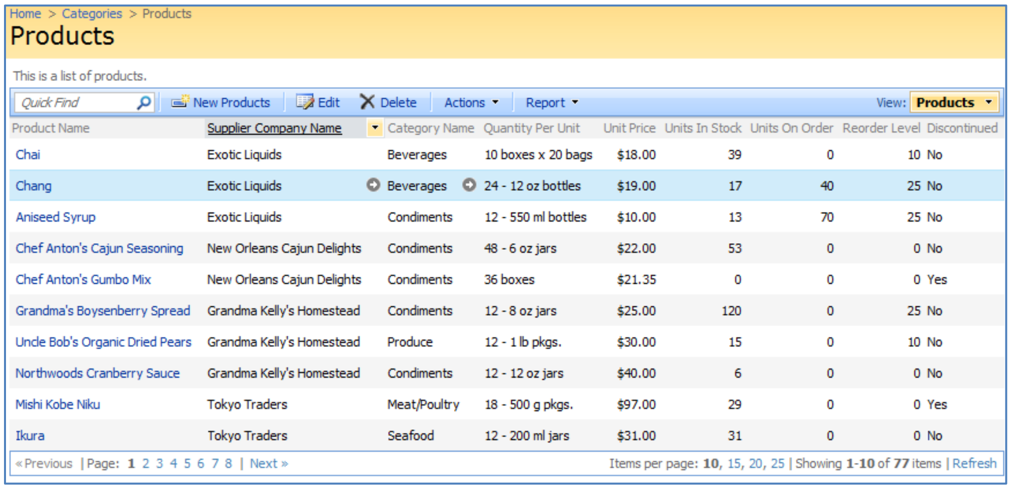
Denormalization Field Map:

Relationship Discovery:

```
foreign key dbo.Products(SupplierID)
references dbo.Suppliers(SupplierID)
```

Now, generate the application.

If you navigate to the *Products* table, you can see a foreign key reference to *SupplierID* in each of the products.



Home > Categories > Products

Products

This is a list of products.

Quick Find | New Products | Edit | Delete | Actions | Report | View: Products

Product Name	Supplier Company Name	Category Name	Quantity Per Unit	Unit Price	Units In Stock	Units On Order	Reorder Level	Discontinued
Chai	Exotic Liquids	Beverages	10 boxes x 20 bags	$18.00	39	0	10	No
Chang	Exotic Liquids	Beverages	24 - 12 oz bottles	$19.00	17	40	25	No
Aniseed Syrup	Exotic Liquids	Condiments	12 - 550 ml bottles	$10.00	13	70	25	No
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	Condiments	48 - 6 oz jars	$22.00	53	0	0	No
Chef Anton's Gumbo Mix	New Orleans Cajun Delights	Condiments	36 boxes	$21.35	0	0	0	Yes
Grandma's Boysenberry Spread	Grandma Kelly's Homestead	Condiments	12 - 8 oz jars	$25.00	120	0	25	No
Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead	Produce	12 - 1 lb pkgs.	$30.00	15	0	10	No
Northwoods Cranberry Sauce	Grandma Kelly's Homestead	Condiments	12 - 12 oz jars	$40.00	6	0	0	No
Mishi Kobe Niku	Tokyo Traders	Meat/Poultry	18 - 500 g pkgs.	$97.00	29	0	0	Yes
Ikura	Tokyo Traders	Seafood	12 - 200 ml jars	$31.00	31	0	0	No

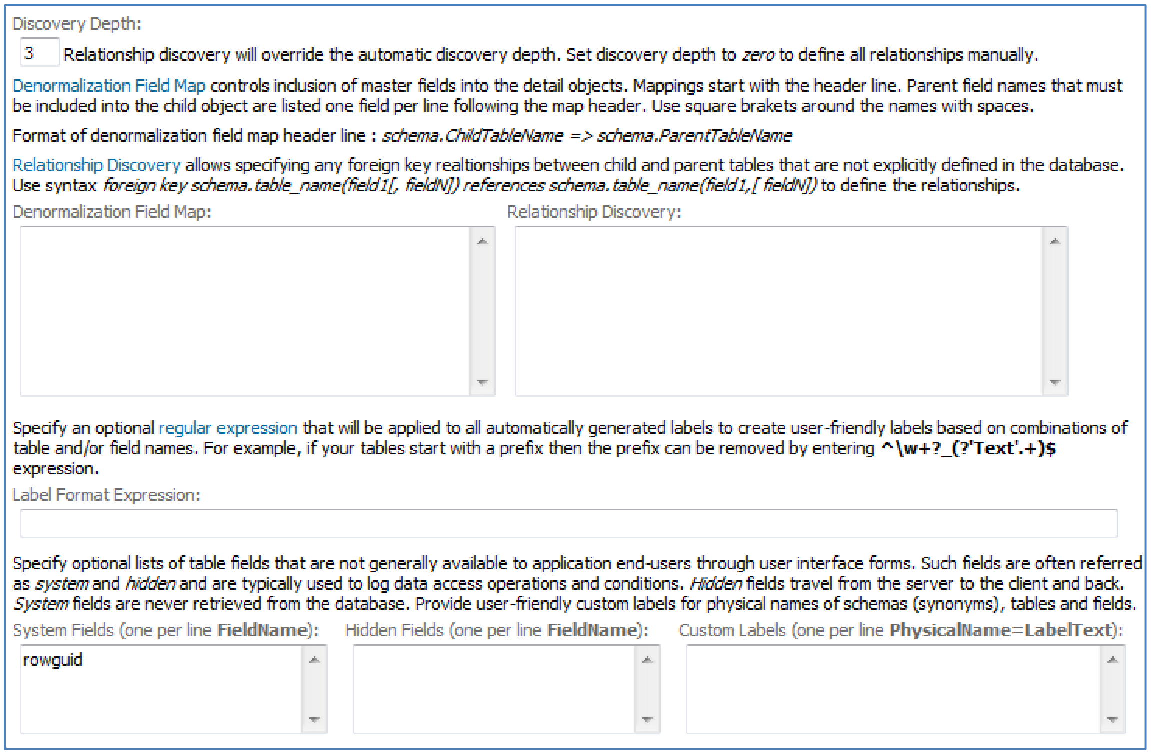
« Previous | Page: 1 2 3 4 5 6 7 8 | Next » Items per page: 10, 15, 20, 25 | Showing 1-10 of 77 items | Refresh

If the key already exists in the application, then specifying a virtual foreign key will have no effect.

System Fields

System fields are present in the database to help external processes and applications manage and manipulate the data. These fields should not be visible to end users and business logic of the generated application.

When listed under *System Fields*, it will be excluded from the application design.



Discovery Depth:

3 Relationship discovery will override the automatic discovery depth. Set discovery depth to *zero* to define all relationships manually.

Denormalization Field Map controls inclusion of master fields into the detail objects. Mappings start with the header line. Parent field names that must be included into the child object are listed one field per line following the map header. Use square brakets around the names with spaces.

Format of denormalization field map header line : *schema.ChildTableName => schema.ParentTableName*

Relationship Discovery allows specifying any foreign key realtionships between child and parent tables that are not explicitly defined in the database. Use syntax *foreign key schema.table_name(field1[, fieldN]) references schema.table_name(field1,[fieldN])* to define the relationships.

Denormalization Field Map:

Relationship Discovery:

Specify an optional regular expression that will be applied to all automatically generated labels to create user-friendly labels based on combinations of table and/or field names. For example, if your tables start with a prefix then the prefix can be removed by entering **^\w+?_(?'Text'.+)$** expression.

Label Format Expression:

Specify optional lists of table fields that are not generally available to application end-users through user interface forms. Such fields are often referred as *system* and *hidden* and are typically used to log data access operations and conditions. *Hidden* fields travel from the server to the client and back. *System* fields are never retrieved from the database. Provide user-friendly custom labels for physical names of schemas (synonyms), tables and fields.

System Fields (one per line **FieldName**): rowguid

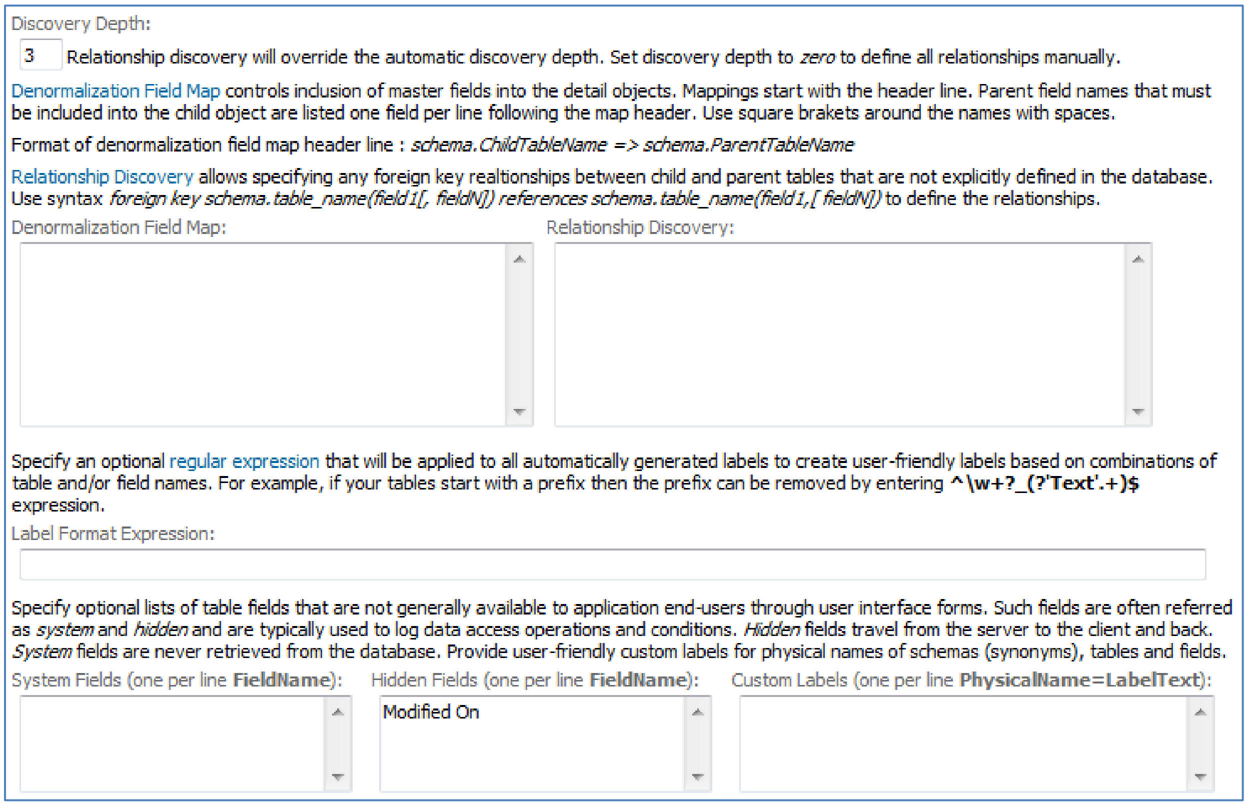
Hidden Fields (one per line **FieldName**):

Custom Labels (one per line **PhysicalName=LabelText**):

Hidden Fields

Hidden fields are used to describe data in multiple table rows, such as *Modified By* and *Modified On*. These fields can be hidden from the user interface.

Merely type in the name of the field in the *Hidden Fields* list, and it will not be visible to the end user of the application.



Discovery Depth:

3 Relationship discovery will override the automatic discovery depth. Set discovery depth to *zero* to define all relationships manually.

Denormalization Field Map controls inclusion of master fields into the detail objects. Mappings start with the header line. Parent field names that must be included into the child object are listed one field per line following the map header. Use square brakets around the names with spaces.

Format of denormalization field map header line : *schema.ChildTableName => schema.ParentTableName*

Relationship Discovery allows specifying any foreign key realtionships between child and parent tables that are not explicitly defined in the database. Use syntax *foreign key schema.table_name(field1[, fieldN]) references schema.table_name(field1,[fieldN])* to define the relationships.

Denormalization Field Map:

Relationship Discovery:

Specify an optional regular expression that will be applied to all automatically generated labels to create user-friendly labels based on combinations of table and/or field names. For example, if your tables start with a prefix then the prefix can be removed by entering **^\w+?_(?'Text'.+)$** expression.

Label Format Expression:

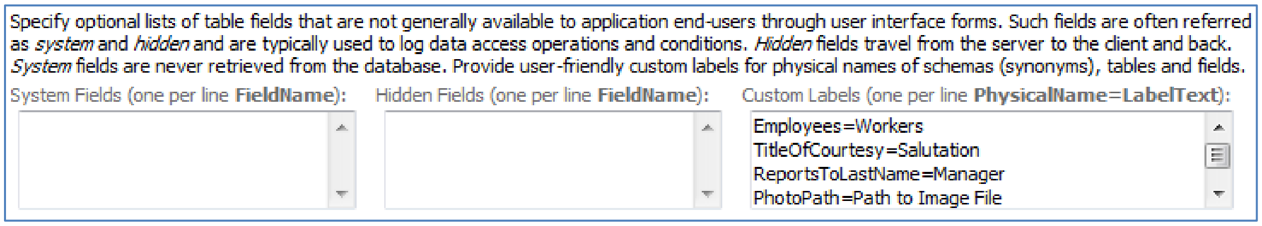
Specify optional lists of table fields that are not generally available to application end-users through user interface forms. Such fields are often referred as *system* and *hidden* and are typically used to log data access operations and conditions. *Hidden* fields travel from the server to the client and back. *System* fields are never retrieved from the database. Provide user-friendly custom labels for physical names of schemas (synonyms), tables and fields.

System Fields (one per line **FieldName**):	Hidden Fields (one per line **FieldName**):	Custom Labels (one per line **PhysicalName=LabelText**):
	Modified On	

Custom Labels

This allows creation of user-friendly labels for tables and fields. These defined custom labels do not affect the physical source code files and object identifiers, which remain the same as defined in the database.

In this example, let's change the fields *TitleOfCourtesy* to *Salutation*, *Employees* to *Workers*, *Reports to Last Name* to *Manager*, and *PhotoPath* to *Path to Image File*.

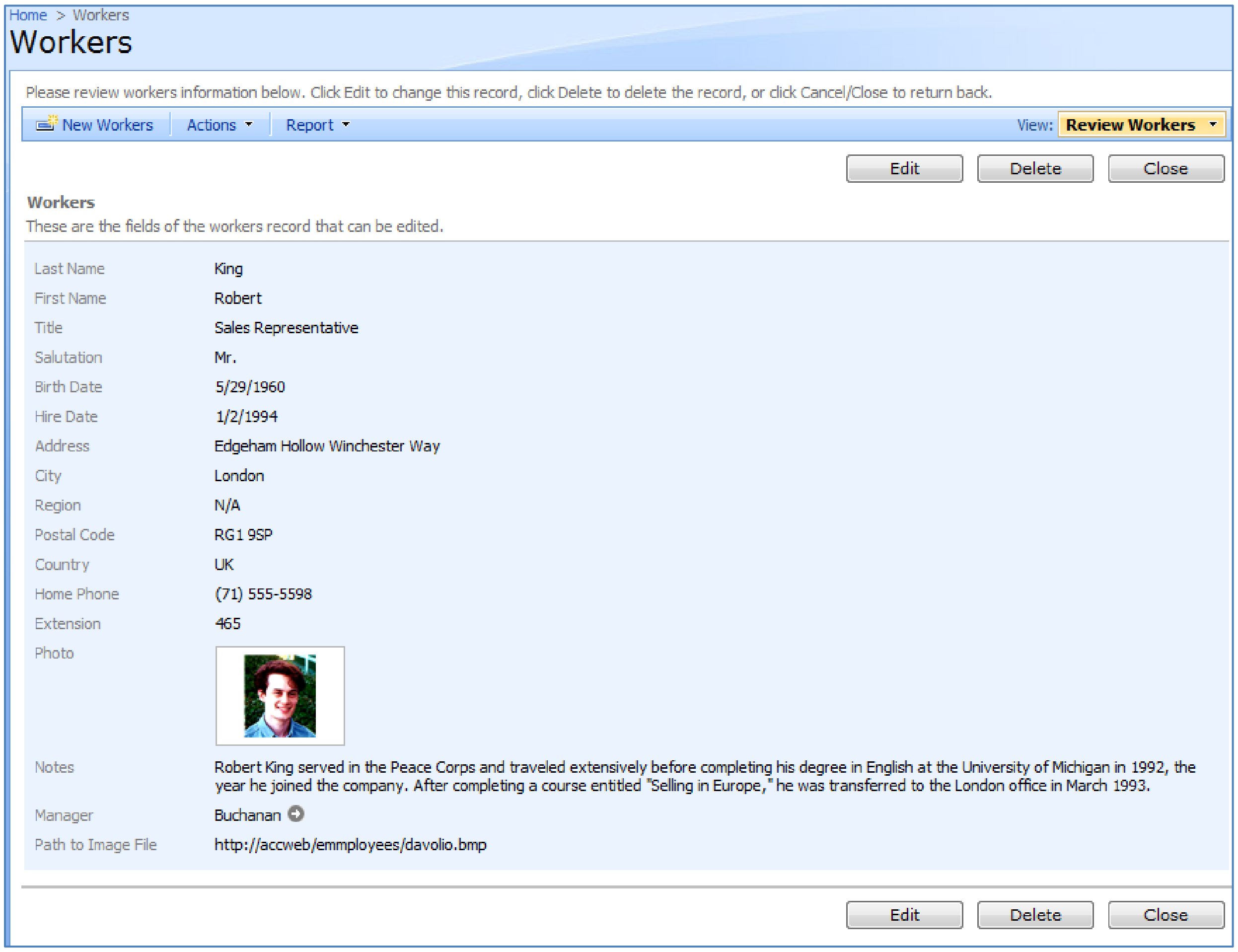


Specify optional lists of table fields that are not generally available to application end-users through user interface forms. Such fields are often referred as *system* and *hidden* and are typically used to log data access operations and conditions. *Hidden* fields travel from the server to the client and back. *System* fields are never retrieved from the database. Provide user-friendly custom labels for physical names of schemas (synonyms), tables and fields.

System Fields (one per line **FieldName**):	Hidden Fields (one per line **FieldName**):	Custom Labels (one per line **PhysicalName=LabelText**):

		Employees=Workers
		TitleOfCourtesy=Salutation
		ReportsToLastName=Manager
		PhotoPath=Path to Image File

Regenerate the application. When you generate the application again, you can see that *Employees* page is now called *Workers* page. The field *Title Of Courtesy* has been renamed to *Salutation*. *Reports To Last Name* is now *Manager*, and *Photo Path* is now *Path to Image File*.



Workers

Please review workers information below. Click Edit to change this record, click Delete to delete the record, or click Cancel/Close to return back.

Workers

These are the fields of the workers record that can be edited.

Last Name	King
First Name	Robert
Title	Sales Representative
Salutation	Mr.
Birth Date	5/29/1960
Hire Date	1/2/1994
Address	Edgeham Hollow Winchester Way
City	London
Region	N/A
Postal Code	RG1 9SP
Country	UK
Home Phone	(71) 555-5598
Extension	465
Photo	
Notes	Robert King served in the Peace Corps and traveled extensively before completing his degree in English at the University of Michigan in 1992, the year he joined the company. After completing a course entitled "Selling in Europe," he was transferred to the London office in March 1993.
Manager	Buchanan
Path to Image File	http://accweb/emmployees/davolio.bmp