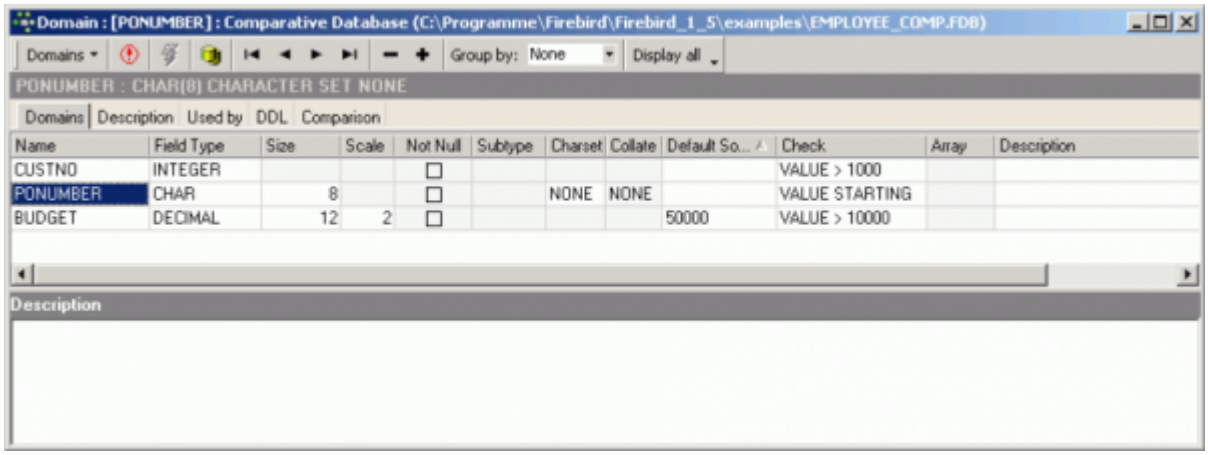
Domain

A domain is a user-defined custom data type global to the database. It is used for defining the format and range of columns, upon which actual column definitions in tables may be based.

This is useful if fields/columns in one or several database tables have the same properties, as it is much simpler to describe such a column type and its behavior as a domain. The columns can then simply be defined by specifying the domain name in the column definition. The column properties (e.g. field length, type, NOT NULL, constraints, arrays etc.) only need to be defined once in the domain. Domains help you create a uniform structure for your regular fields (e.g. ID, address and currency fields) and add more understanding to your database structure. You can define a number of characteristics including: data type, an optional default value, optional disallowing of NULL values, an optional CHECK constraint and an an optional collation clause.

Certain attributes specified in the domain can be overwritten in the table field definition, i.e. a column can be based upon a domain; however small changes may still possibly be made for this column. The domain default, collation clause and NOT NULL settings can be overridden by the field definition, and a field based on a domain can add additional CHECK constraints to the domain's CHECK constraint.

A domain is a database object and is part of the database's metadata, and can be created, modified and dropped as all other Firebird/InterBase® objects in the IBExpert DB Explorer.



When developing a normalized database, the question arises in how far domains are necessary (multiple fields, multiple data etc.). However, it does make life easier, should column alterations be necessary; e.g. zip code alteration from 4 to 5 digits (as was the case in Germany after the reunion), change of currency (e.g. from DM or Lire to Euro). In such cases, only the domain needs to be altered, and not each relevant column in each table individually throughout the database.

It should also be noted, that if user-defined domains are not explicitly defined and used for table column definitions, Firebird/InterBase® generates a new domain for every single table column created! All domains are stored in the system table RDB$FIELDS.

Domain integrity

Domain integrity ensures that a column is kept within its allowable limits. This is achieved by keys

and constraints.

back to top of page

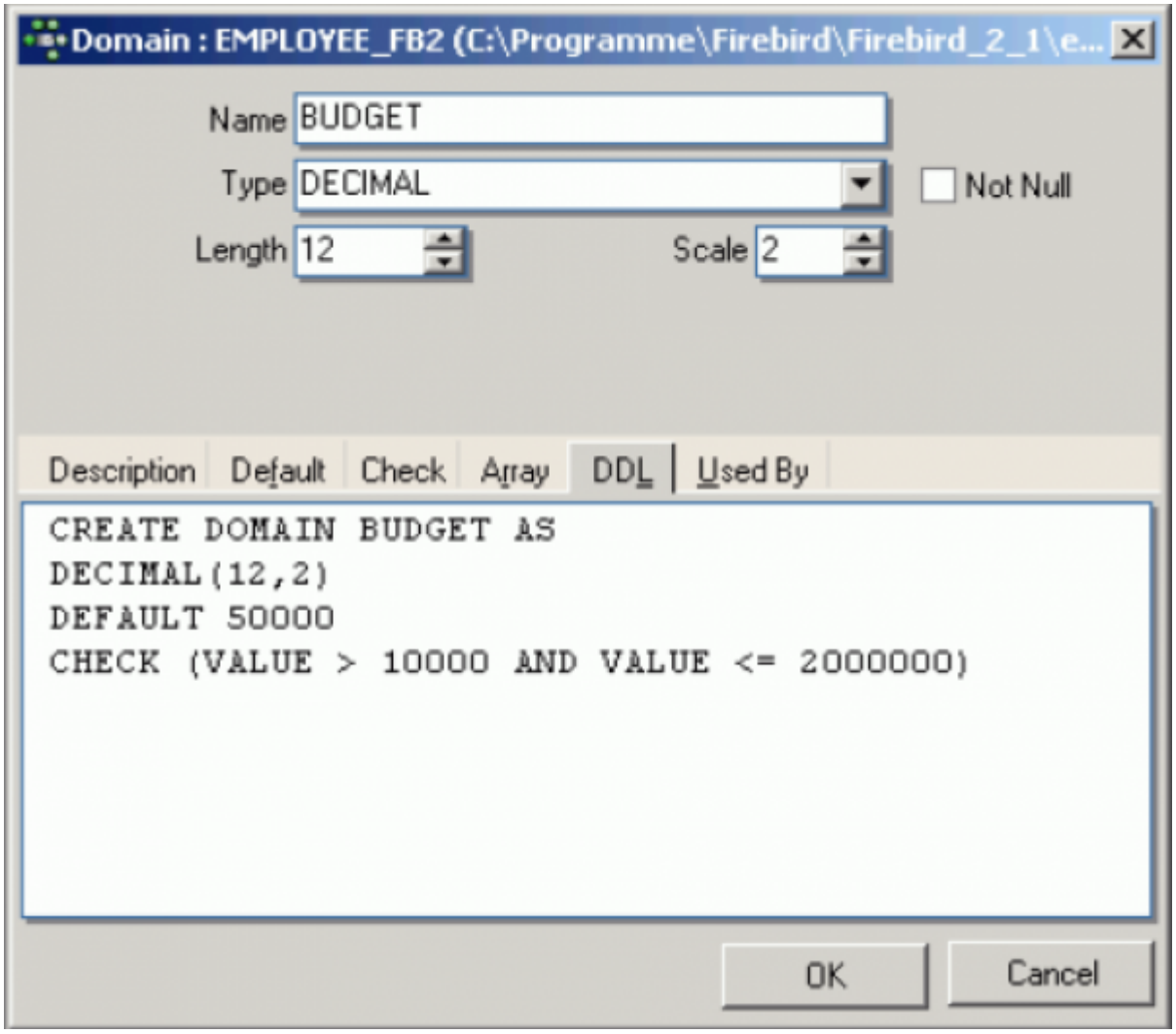
New domain / Domain Editor

A new domain can be created for a connected database, either by using the menu item *Database / New Domain*, or using the DB Explorer right-click menu (or key combination [Ctrl + N], when the domain node of the relevant connected database is highlighted), or the New Domain icon on the New Database Object toolbar.

A *New Domain* dialog appears, with its own toolbar, and a pull-down menu (domain button). The Domain Editor toolbar offers the following options:

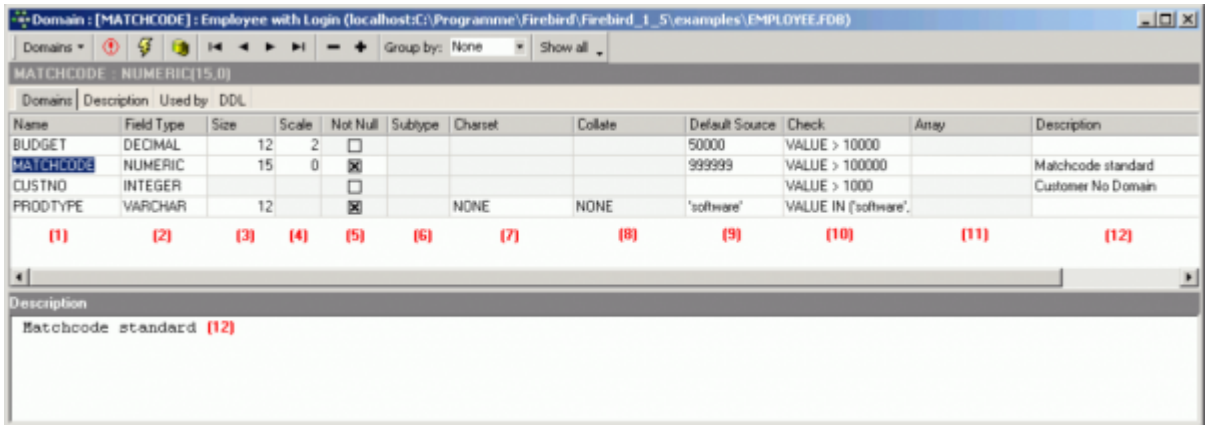
- Enable direct modifying of system tables
- Compile
- Duplicate the selected domain
- Navigational buttons
- Group by either Type or Charset
- Display all domains

For those users preferring to use the old IBExpert Modal Editor, check the Use old-style Modal Editor option in the IBExpert Options menu: Object Editor Options / Domains Editor.



A domain can also be created or selected and edited, when a new field is created or an existing field edited in a table, using IBExpert's Table Editor. (Please refer to Insert Field for further information).

The following illustrates the creation of a new domain using the Domain Editor: initially a domain name is specified (1) in the first column on the first page *Domains*:



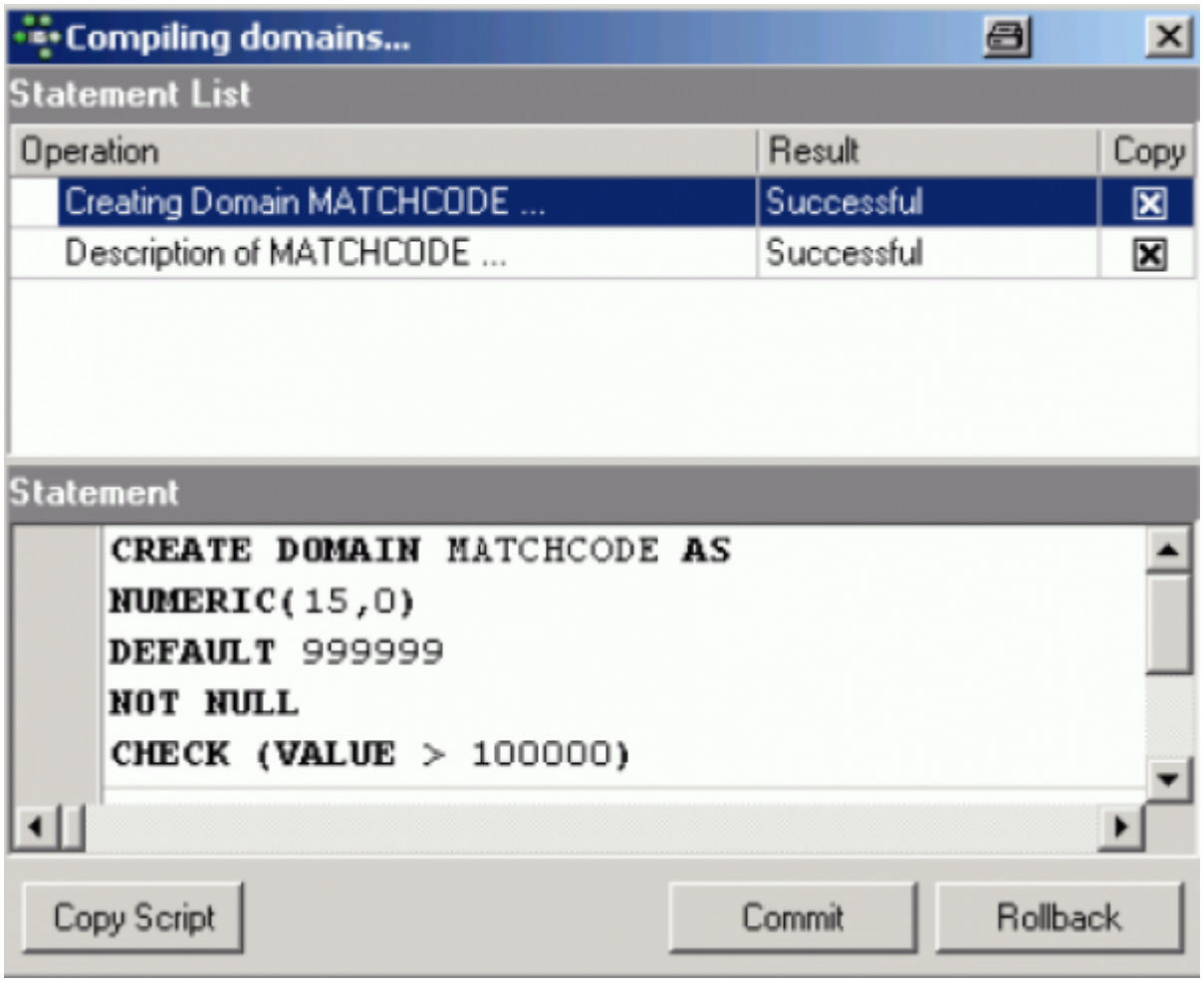
(Illustration displays the default *Domain Editor*.)

- (2) **Field Type:** Here the data type can be specified.
- (3) **Size:** Specifies the field size.
- (4) **Scale:** Here the number of decimal places can be specified for all numerical fields.
- (5) **Not Null:** This check box can be marked by double-clicking or using the space bar. NOT

NULL forces data to be entered in this field (i.e. the field may not be left empty).

- (6) **Subtype:** A subtype should be specified for blob fields.
- (7) **Charset:** A character set may be specified for individual domains. This overrides the database default character set. Although this is seldom used, it may be necessary should, for example, Asian, Russian or Arabic addresses need to be input and collated in a database with a European default character set. If no character set is specified, the domain uses the default database character set. If you do not specify a default character set, the character set defaults to NONE. Using character set NONE data is stored and retrieved just as it is originally entered. You can load any character set into a column defined with NONE, but you cannot load that same data into another column that has been defined with a different character set. In these cases, no transliteration is performed between the source and destination character sets, so errors can occur during assignment.
- (8) **Collate:** Determines collation for a character set specified for a domain.
- (9) **Default Source:** Here a default data entry (text or numeric, depending upon the specified data type) can be specified, e.g. the text NOT KNOWN can be specified as a default source, if an address field cannot be input by the user, because the information is unavailable.
- (10) **Check:** Each data set is examined for validity according to an expression defined in brackets. Certain conditions can be specified (see Check constraint) causing an automatic database examination during data input, to ensure data consistency in the tables and among each other.
- (11) **Array:** Although arrays contradict all the rules of database normalization, there are certain situations (for example storing measurement data), when they are necessary.
- (12) **Description:** Useful for database documentation. The Description page should be used to describe the domain; the *Description* field for describing the field.

Several domains can be created simultaneously in the *New Domain Editor*. After creating the new domain(s), including all necessary parameters, don´t forget to compile (using [Ctrl + F9] or the respective icon):

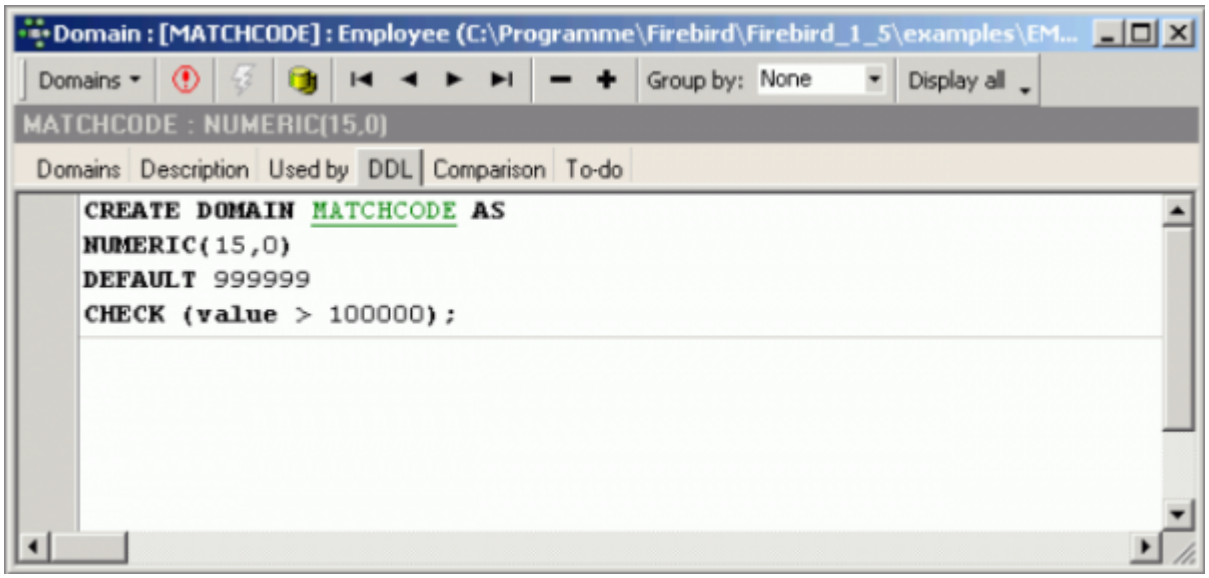


and finally committing, or should amendments be necessary, rolling back.

Tip: by clicking on the column headers (i.e. *PK, FK, Field Name* etc.), the fields can be sorted into ascending or descending order based upon that column. Double-clicking on the right edge of the column header adjusts the column width to the ideal width.

In addition to the Domains page, there are also Description, Used By, DDL, Comparison and To-Do pages:

- **Description:** this displays the description for the highlighted domain (i.e. the domain, where the cursor is currently standing).
- **Used By:** this displays those database objects which use or depend upon this domain.
- **DDL:** the DDL page displays the SQL statement created by IBExpert to create all specifications made by the user on the Domains page.
- **Comparison:** this allows you to compare a domain in the main database with a domain in a comparative database (for further information please refer to Comparison).
- **To-Do:** this feature can be used to organize your database development. You can add ToDo items for each object in the database.



Domains can also be created and edited directly from the *New Field* Editor (please refer to Insert Field).

Domains can, of course, also be created using DDL directly in the SQL Editor, using the following syntax:

CREATE DOMAIN domain [AS] <datatype>
[DEFAULT {literal | NULL | USER}]
NOT NULL] CHECK (<dom_search_condition>)]
COLLATE collation];

<datatype> =
{SMALLINT|INTEGER|FLOAT|DOUBLE PRECISION} [<array_dim>]

| {DATE|TIME|TIMESTAMP} [<array_dim>]

| {DECIMAL | NUMERIC} [(precision [, scale])] [<array_dim>]
| {CHAR | CHARACTER | CHARACTER VARYING | VARCHAR} [(int)]
[<array_dim>] CHARACTER SET charname]

| {NCHAR | NATIONAL CHARACTER | NATIONAL CHAR}
[VARYING] [(int)] [<array_dim>]

| BLOB SUB_TYPE {int | subtype_name}] SEGMENT SIZE int]
[CHARACTER SET charname]

| BLOB [(seglen [, subtype])]

```
<array_dim> = [[x:]y [, [x:]y …]]
```

<dom_search_condition> = {
VALUE <operator> value
VALUE [NOT] BETWEEN value AND value
| VALUE [NOT] LIKE value [ESCAPE value]
| VALUE [NOT] IN (value [, value …])

| VALUE IS [NOT] NULL
| VALUE [NOT] CONTAINING value
| VALUE [NOT] STARTING [WITH] value
| (<dom_search_condition>)
| NOT <dom_search_condition>
| <dom_search_condition> OR <dom_search_condition>
| <dom_search_condition> AND <dom_search_condition>
}

<operator> = {= | < | > | ⇐ | >= | !< | !> | <> | !=}

Argument	**Description**
domain	Unique name for the domain.
datatype	SQL data type.
DEFAULT	Specifies a default column value that is entered when no other entry is made; possible values are:
	• literal — Inserts a specified string, numeric value, or date value.
	• NULL — Enters a NULL value.
	• USER — Enters the user name of the current user.
	The column must be of compatible character type to use the default.
NOT NULL	Specifies that the values entered in a column cannot be NULL.
CHECK (dom_search_condition)	Creates a single CHECK constraint for the domain.
VALUE	Placeholder for the name of a column eventually based on the domain.
COLLATE collation	Specifies a collation sequence for the domain.

Example:

```
CREATE DOMAIN MATCHCODE
  AS INTEGER
     DEFAULT  999999
     NOT NULL
     CHECK (VALUE > 100000);
```

back to top of page

Edit domain/alter domain

A domain can be altered in the Domain Editor, opened by double-clicking on the domain name in the DB Explorer. Alternatively use the DB Explorer´s right mouse-click menu item *Edit Domain* or key combination [Ctrl + O].

CHECK instructions and default values may be added, altered or deleted. In fact, any aspect of an existing domain may be altered, in certain cases this is achieved by dropping the feature, such as a CHECK constraint, and recreating it. In fact, the only attribute that cannot be altered is the domain's NOT NULL setting. Here it is necessary to drop the whole domain and recreate it. And if problems are encountered altering from a certain data type to another, you may also need to first drop the old domain and recreate it with the new features (see Drop Domain.)

Please note that if you want to change the CHECK constraint for a domain that already has a constraint defined, the existing constraint must first be dropped and then the new one added. ADD CHECK does not replace the current constraint with the new one. It is also important to realize that altering a CHECK constraint does not cause existing database rows to be revalidated; CHECK constraints are only validated when an INSERT or UPDATE is performed. One way of overcoming this limitation is to perform an UPDATE query using a dummy operation. If existing rows violate the new CHECK constraint, the query fails. These rows can then be extracted by performing a SELECT.

Any changes made apply immediately to all columns using the domain definition, unless, of course, the column´s (field) definition overrides these.

The SQL syntax for this command is:

```
ALTER DOMAIN name {
SET DEFAULT {literal | NULL | USER}
| DROP DEFAULT //
| ADD [CONSTRAINT] CHECK (<dom_search_condition>)
| DROP CONSTRAINT | new_col_name
| TYPE datatype};
```

```
<dom_search_condition> = {
VALUE <operator> <val>
| VALUE [NOT] BETWEEN <val> AND <val>
| VALUE [NOT] LIKE <val> [ESCAPE <val>]
| VALUE [NOT] IN (<val> [, <val> …])
| VALUE IS [NOT] NULL
| VALUE [NOT] CONTAINING <val>
| VALUE [NOT] STARTING [WITH] <val>
| (<dom_search_condition>)
| NOT <dom_search_condition>
| <dom_search_condition> OR <dom_search_condition>
| <dom_search_condition> AND <dom_search_condition>
}
```

```
<operator> = {= | < | > | <= | >= | !< | !> | <> | !=}
```

Argument	**Description**
name	Name of an existing domain.
SET DEFAULT	Specifies a default column value that is entered when no other entry is made. Values:
	• literal — Inserts a specified string, numeric value, or date value.
	• NULL — Enters a NULL value.
	• USER — Enters the user name of the current user; column must be of compatible text type to use the default.
	• Defaults set at column level override defaults set at the domain level.
DROP DEFAULT	Drops an existing default.
ADD [CONSTRAINT] CHECK, dom_search_condition	Adds a CHECK constraint to the domain definition; a domain definition can include only one CHECK constraint.

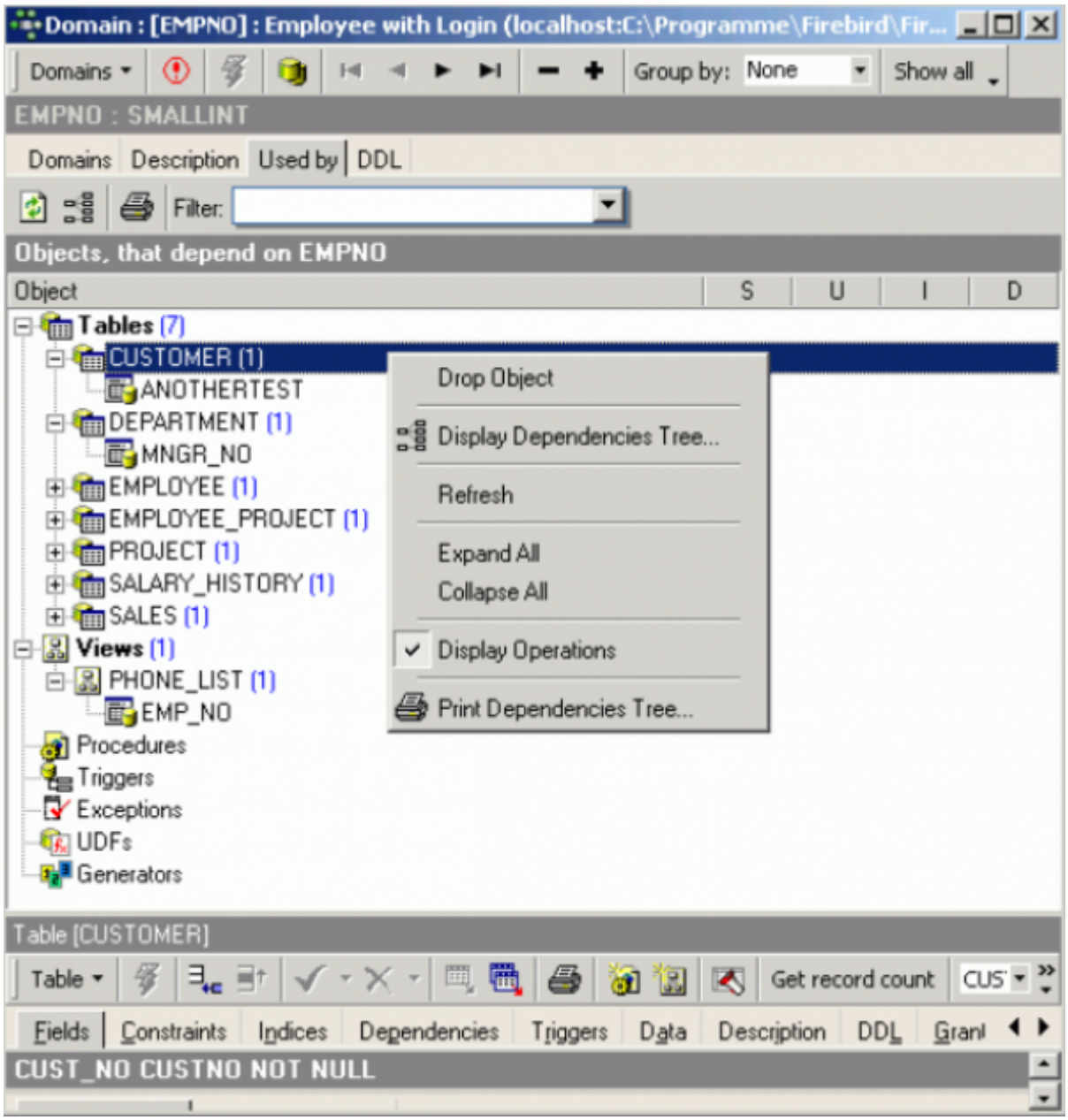
Argument	Description
DROP CONSTRAINT	Drops CHECK constraint from the domain definition.
new_col_name	Changes the domain name.
TYPE data_type	Changes the domain data type.

A domain may be altered by its creator, the SYSDBA user, and any users with operating system root privileges.

back to top of page

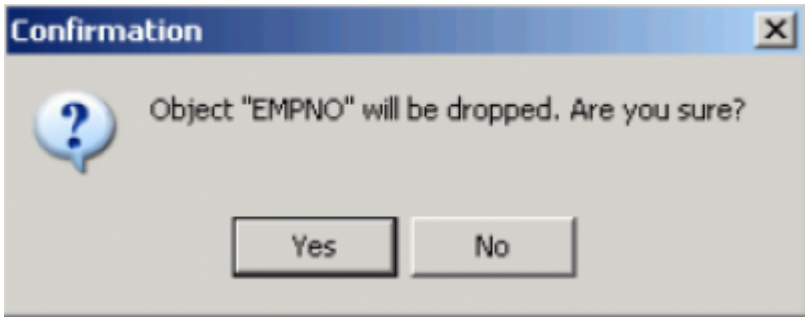
Drop domain/delete domain

A domain may only be dropped if it is not currently being used by any of the database tables. The Domain Editor's *Used By* page shows which database objects use this domain. The dependent objects may also be directly dropped here, if wished, using the right-click menu on the selected object, and choosing the menu item *Drop Object* or [Ctrl + Del].



To drop a domain use the DB Explorer right-click and select the menu item Drop Domain or [Ctrl + Del].

Alternatively, a domain can be dropped directly from the Domain Editor using the Domains pull-down menu or the "-" icon in the Domain Editor toolbar. IBExpert asks for confirmation:



before finally dropping the domain. Once dropped it cannot be retrieved; the domain has to be recreated if a mistake has been made!

Using SQL the syntax is:

```
DROP DOMAIN <domain_name>;
```

A domain can only be dropped by its creator, the SYSDBA and any users with operating system root privileges.

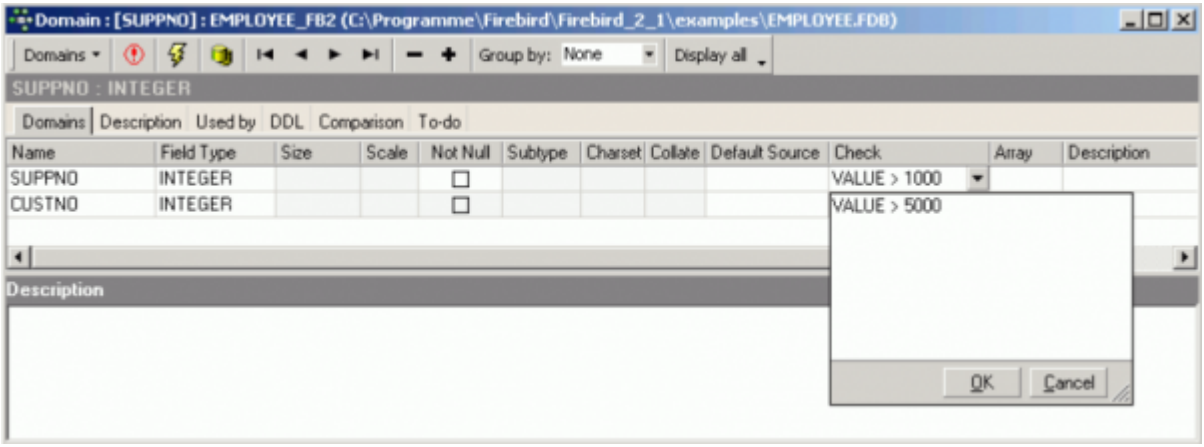
back to top of page

Duplicate domain

It is possible to create a new domain, based on an existing domain, using the Domain Editor's menu item Duplicate Domain, or the

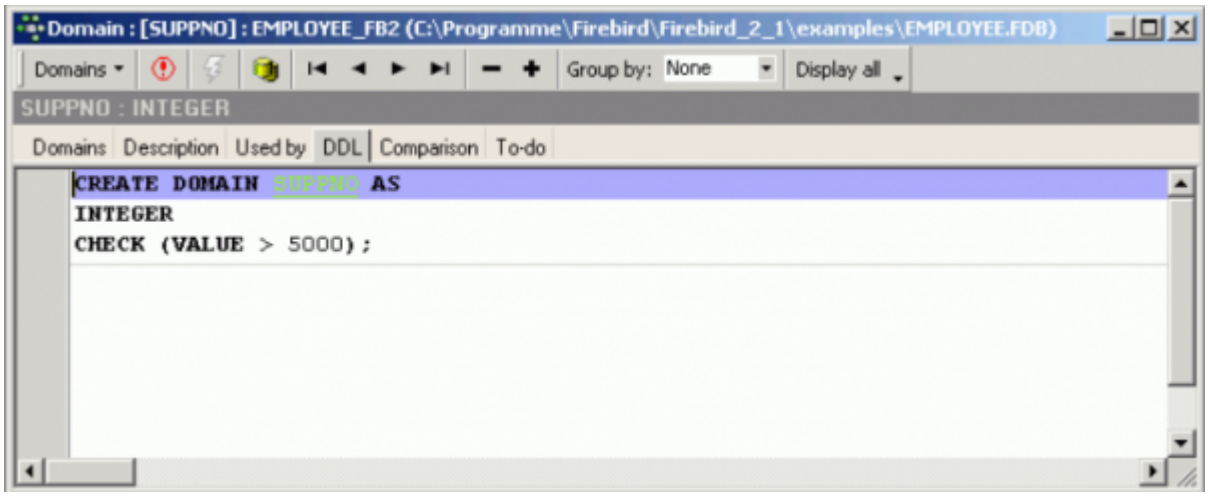
icon in the Domain Editor toolbar.

An exact copy of the selected domain is made, and can then be adapted as wished. For example a new domain, SUPPNO could be based on the CUSTNO domain in the EMPLOYEE database, by duplicating it and then, for example, renaming it and altering the CHECK VALUE to > 5000.



This saves time creating several similar domains; all you need to do is copy a domain, perform any minor alterations necessary, compile and finally commit.

The Domain Editor's DDL page displays the actual statement used to create the new domain:



Duplicating domains from one database to another

If you have already created a wide range of domains in one database, and would like to duplicate them in another new database, simply take the following steps in IBExpert:

1. Copy the domain DDL (Data Definition Language) into the SQL Editor and execute it.
2. Drag ´n´ drop the domain from the source database into the Domain Editor of the target database.