

Field

A field can be defined as the intersection in a [table](#) where a [row](#) meets a [column](#), containing a clearly differentiated atomic piece of information. Each data field should be [unique](#) and represent an indivisible quantity of information.

DEPT_NO	DEPARTMENT	HEAD_DEPT	MNGR_NO	BUDGET	LOCATION	PHONE_NO
000	Corporate Headquarters	<null>	105	1 000 000,00	Monterey	(408) 555-1234
100	Sales and Marketing	000	85	2 000 000,00	San Francisco	(415) 555-1234
110	Pacific Rim Headquarters	100	34	600 000,00	Kauai	(808) 555-1234
115	Field Office: Japan	110	118	500 000,00	Tokyo	3 5350 0901
116	Field Office: Singapore	110	<null>	300 000,00	Singapore	3 55 1234
120	European Headquarters	100	36	700 000,00	London	71 235-4400
121	Field Office: Switzerland	120	141	500 000,00	Zurich	1 211 7767
123	Field Office: France	120	134	400 000,00	Cannes	58 68 11 12
125	Field Office: Italy	120	121	400 000,00	Milan	2 430 39 39
130	Field Office: East Coast	100	11	500 000,00	Boston	(617) 555-1234
140	Field Office: Canada	100	72	500 000,00	Toronto	(416) 677-1000
180	Marketing	100	<null>	1 500 000,00	San Francisco	(415) 555-1234
600	Engineering	000	2	1 100 000,00	Monterey	(408) 555-1234
620	Software Products Div.	600	<null>	1 200 000,00	Monterey	(408) 555-1234
621	Software Development	620	<null>	400 000,00	Monterey	(408) 555-1234
622	Quality Assurance	620	9	300 000,00	Monterey	(408) 555-1234
623	Customer Support	620	15	650 000,00	Monterey	(408) 555-1234
670	Consumer Electronics Div.	600	107	1 150 000,00	Burlington, VT	(802) 555-1234
671	Research and Development	670	20	460 000,00	Burlington, VT	(802) 555-1234
672	Customer Services	670	94	850 000,00	Burlington, VT	(802) 555-1234

Each database field has a name, which enables the data to be accessed. A database field can be based on a [domain definition](#) or defined individually in the IBExpert [Create Table](#) or [Table Editors](#), in which case Firebird/InterBase® automatically creates a [system domain](#) for the field definition.

[back to top of page](#)

Adding new field (insert field) using the Field Editor

Fields can be inserted into a table at the time of table creation, using the IBExpert [DB Explorer](#) or menu item *New Table*. It is however often necessary to add new fields after the table has been created. This can be easily done in IBExpert by opening the [Table Editor](#) (double-click on the relevant table in the IBExpert DB Explorer) or using the DB Explorer right-click menu *Edit Table ...* (or key combination [Ctrl + O]), and then inserting a field using the



Add *Field* icon (or [Ins] key) or the Table Editor right-click menu Insert Field, to open the Adding New Field Editor.

Adding New Field

Table: ☒ Not NULL

Field: ☐ Primary Key

Domain: Edit Domain

Collate: New Domain

Domain Info

VARCHAR(5) CHARACTER SET NONE
CHECK (VALUE > '99999')
COLLATE NONE

OK Cancel

The *Adding New Field* Editor displays the table name, into which the field is to be inserted. The new field name can be specified by the user, along with the parameters [Not NULL](#) and [Primary Key](#). Further options are to be found on the [Default](#) and [Check](#) pages, and the usual [IBExpert Desc](#) (= [Description](#)) and [DDL](#) (= [Data Definition Language](#)) information pages are also included.

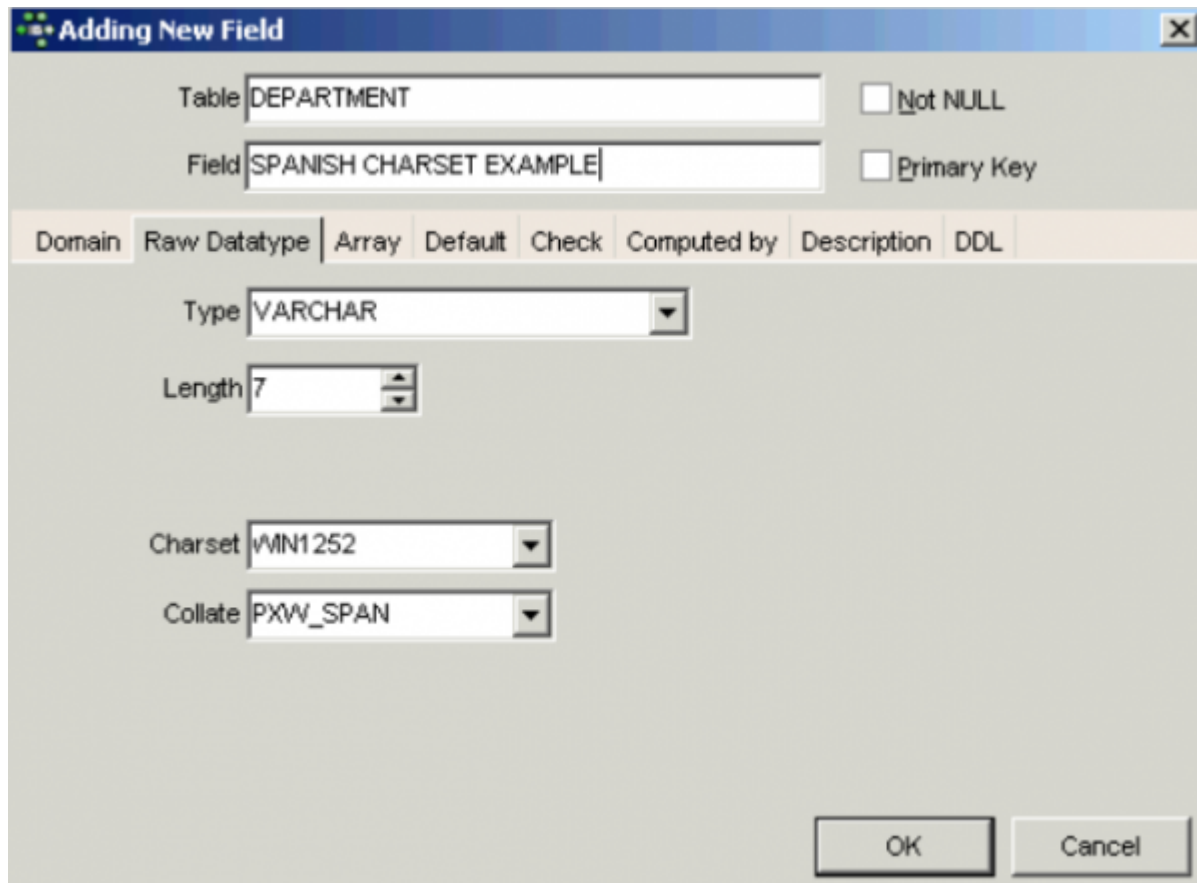
The new field may be based upon an existing domain (which may be edited using the [Edit](#) button) or a new domain can be created directly from the [New Field](#) Editor. All existing domains (in the connected database) can be viewed in the [Domain](#) drop-down list. The domain information can be viewed in the Editor's lower panel.

It is also possible to define certain numeric formats as standard using the [IBExpert Options menu](#) item, [Environment Options#Grid|Environment Options / Grid / Display Formats](#), if wished. These format standards can be overwritten in individual fields here in the [Field Editor](#).

The screenshot shows the 'Adding New Field' dialog box. The 'Table' field contains 'DEPARTMENT'. The 'Field' field is empty. The 'Type' dropdown is open, showing a list of data types: SMALLINT, INTEGER, BIGINT, FLOAT, DOUBLE PRECISION, NUMERIC, DECIMAL, DATE, TIME, TIMESTAMP, CHAR, VARCHAR, and BLOB. The 'Domain' tab is selected. There are checkboxes for 'Not NULL' and 'Primary Key'. At the bottom are 'OK' and 'Cancel' buttons.

Of course a new field doesn't have to be based on a domain. The [data type](#) can be specified using the pull-down list under the *Raw Data* type tab. However, Firebird/InterBase® automatically generates a system domain for all specified fields, so when a new field is inserted, or existing field altered, Firebird/InterBase® inserts or alters the respective system domain.

Additional context-sensitive input fields appear, relevant to the data type selected (e.g. when [VARCHAR](#) is selected, options for specifying *Length*, *Charset*, and *Collate* are offered; in the case of [NUMERIC](#), *Precision* and *Scale* can be specified).



Adding New Field

Table: ☐ Not NULL

Field: ☐ Primary Key

Domain | Raw Datatype | Array | Default | Check | Computed by | Description | DDL

Type:

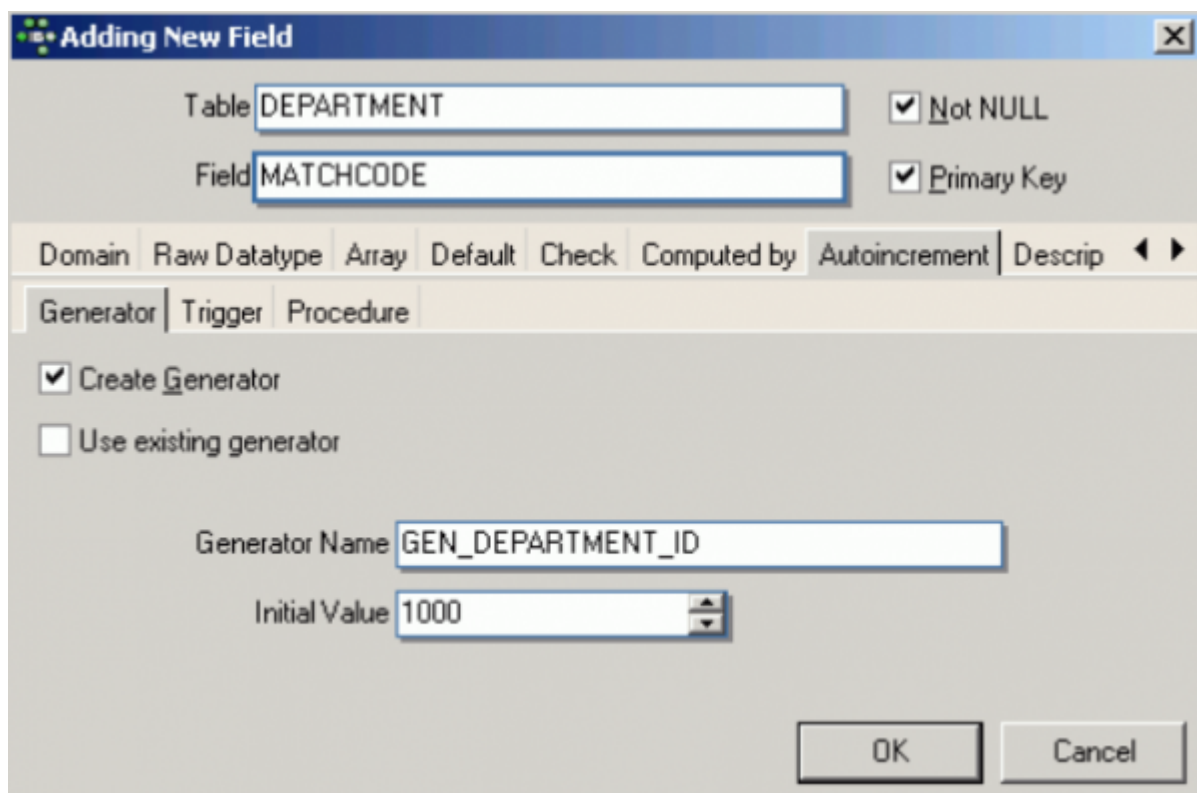
Length:

Charset:

Collate:

OK Cancel

Furthermore [arrays](#) can be defined, as well as default values, [check constraints](#), “computed by” calculations and [autoincrements](#).



Adding New Field

Table: ☒ Not NULL

Field: ☒ Primary Key

Domain | Raw Datatype | Array | Default | Check | Computed by | Autoincrement | Descrip

Generator | Trigger | Procedure

☒ Create Generator

☐ Use existing generator

Generator Name:

Initial Value:

OK Cancel

The *Autoincrement* page allows new [generators](#) to be created, or an existing generator to be selected. New [triggers](#) and [procedures](#) can also be created directly here in this Editor for this field, if desired.

The screenshot shows the 'Adding New Field' dialog box with the 'Trigger' tab selected. The 'Table' field is set to 'DEPARTMENT' and the 'Field' is 'MATCHCODE'. The 'Not NULL' and 'Primary Key' checkboxes are checked. The 'Create Trigger' checkbox is also checked. The trigger definition text area contains the following SQL code:

```
CREATE TRIGGER DEPARTMENT_BI FOR DEPARTMENT
ACTIVE BEFORE INSERT POSITION 0
AS
BEGIN
  IF (NEW.MATCHCODE IS NULL) THEN
    NEW.MATCHCODE = GEN_ID(GEN_DEPARTMENT_ID,1);
  END
```

At the bottom are 'OK' and 'Cancel' buttons.

The last two pages display the object *Description* (which can be inserted, edited and deleted here by the user as wished), and the [DDL](#) page,

The screenshot shows the 'Adding New Field' dialog box with the 'DDL' tab selected. The 'Table' field is 'DEPARTMENT' and the 'Field' is 'MATCHCODE'. The 'Not NULL' and 'Primary Key' checkboxes are checked. The DDL text area contains the following SQL code:

```
ALTER TABLE DEPARTMENT
ADD MATCHCODE SMALLINT
NOT NULL PRIMARY KEY
```

At the bottom are 'OK' and 'Cancel' buttons.

which displays the SQL code for the field as specified by the user.

For those preferring to specify their fields in the [SQL Editor](#), please refer to the [Table](#) chapter for the [CREATE TABLE](#) or [ALTER TABLE](#) syntax.

[back to top of page](#)

Edit field/alter field

Similar to [Alter Domain](#), certain field attributes may be altered. For example, the [data type](#), the field name and the field position may be altered; CHECK instructions and default values may be added, altered or deleted. There are some limitations to modifying columns if they are used as part of a constraint, view, etc. Such dependencies can be viewed in the [Domain Editor](#) on the [Used By](#) page.

Fields can be altered in the [Table Editor](#) by double-clicking on the selected field, right-clicking and selecting *Edit Field* from the menu, or pressing the [Enter] key to open the Field Editor:

The screenshot shows a dialog box titled "Edit field BUDGET". At the top, there are two text boxes: "Table" containing "DEPARTMENT" and "Field" containing "BUDGET". To the right of the "Table" box is a checkbox labeled "Not NULL". Below these are four tabs: "Domain", "Default", "Autoincrement", and "Description". The "Domain" tab is selected, showing a dropdown menu with "BUDGET" and two buttons: "Edit Domain" and "New Domain". Below the tabs is a section titled "Domain Info" which contains the following text: "DECIMAL(12,2)", "DEFAULT 50000", and "CHECK (VALUE > 10000 AND VALUE <= 2000000)". At the bottom of the dialog are "OK" and "Cancel" buttons.

You will notice that you need to switch to the [Domain Editor](#) to perform any actual changes, as even if the field is not based on a user-defined domain, Firebird/InterBase® automatically creates a system domain for all field definitions. Simply click *Edit Domain* to spring to the Domain Editor:

Domain : EMPLOYEE_FB2 (C:\Programme\Firebird\Firebird_2_1\e...

Name:

Type: ☐ Not Null

Length: Scale:

Description Default Check Array DDL Used By

```
CREATE DOMAIN BUDGET AS
DECIMAL (12,2)
DEFAULT 50000
CHECK (VALUE > 10000 AND VALUE <= 2000000)
```

OK Cancel

The desired alterations can then be easily made to the user-defined or system domain and executed and checked before finally committing:

Changing domain BUDGET...

Statement List

Operation	Result	Copy
Altering Domain...	Successful	☒

Statement

```
update RDB$FIELDS set
RDB$FIELD_PRECISION = 13
where RDB$FIELD_NAME = 'BUDGET'
```

Copy Script Commit Rollback

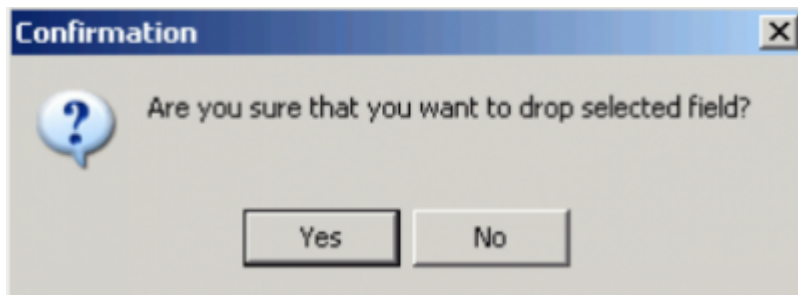
Please refer to [Alter Domain](#) and Alter Table for further information.

[back to top of page](#)

Drop field/delete field

Fields can be dropped directly in the [Table Editor](#) on the [Fields](#) page, by using the “-” icon in the [Table Editor toolbar](#), selecting from the right-click menu or using the key combination [Shift + Del].

IBExpert asks for confirmation:



before finally dropping the field. Once dropped, it cannot be retrieved.

When dropping fields, it is important to note that the field may not be part of the table's [primary key](#), have a [foreign key](#) relationship with another table, contain a unique [constraint](#), be part of a table constraint or part of another column's [CHECK constraint](#).

The [Constraints](#) page in the [Table Editor](#) lists all such fields, so that the developer can quickly ascertain whether constraint alterations/deletions are necessary, before dropping the field in question (or whether, in fact, the field should be dropped at all!).

Using SQL the syntax is:

```
ALTER TABLE <table_name>
DROP <field_name>;
```

From:
<https://www.ibexpert.com/docu/> - IBExpert

Permanent link:
<https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:field>

Last update: **2023/08/16 17:44**

