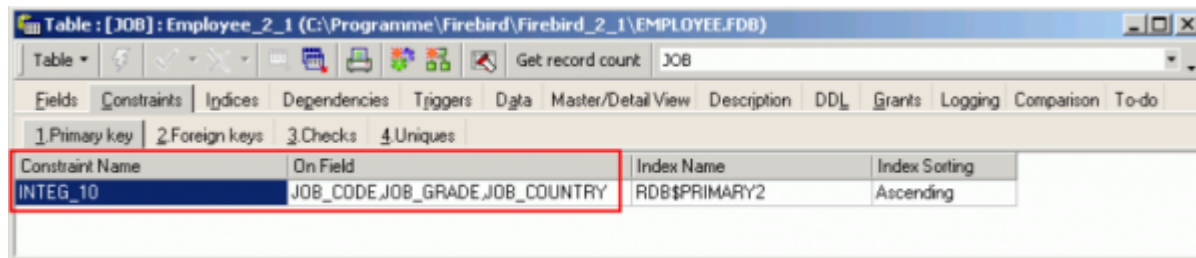


Composite/Compound key

A composite key consists of two or more [columns](#) , designated together as a table's [primary key](#). Multiple-column primary keys can be defined only as table-level constraints:



The screenshot shows the 'Constraints' tab for a table named 'JOB'. A red box highlights the 'Primary key' section, showing a constraint named 'INTEG_10' on fields 'JOB_CODE, JOB_GRADE, JOB_COUNTRY'. The 'Index Name' is 'RDB\$PRIMARY2' and the 'Index Sorting' is 'Ascending'.

Constraint Name	On Field	Index Name	Index Sorting
INTEG_10	JOB_CODE, JOB_GRADE, JOB_COUNTRY	RDB\$PRIMARY2	Ascending

Single-column primary keys can be defined at either the column or the table level (but not both). For example, the following code states that the table's primary key consists of three columns, `JOB_CODE`, `JOB_GRADE`, and `JOB_COUNTRY`. Neither of these columns is required to be unique by itself, but their combined value must be unique (and `NOT NULL`).

```
CREATE TABLE  
COLUMN_defs ...  
PRIMARY KEY (JOB_CODE, JOB_GRADE, JOB_COUNTRY);
```

Unfortunately such keys have two huge disadvantages: firstly they slow the database performance considerably, as Firebird/InterBase® needs to check all contents of all columns designated in such a composite key; secondly the sequence of the fields concerned must be identical in all referenced tables.

Basically composite keys should be avoided! It is much preferable to use an internal ID key (so-called artificial key) as the primary key for each table.

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:compositve-compound-key>

Last update: 2023/08/14 10:19

