

CREATE INDEX

Available in: [DSQL](#), [ESQL](#)

Description

Creates an [index](#) on a [table](#) for faster searching, sorting and/or grouping.

Syntax

```
CREATE [UNIQUE] [ASC[ENDING] | [DESC[ENDING]] INDEX indexname
      ON tablename
      { (colname [, colname ...]) | COMPUTED BY (expression) }
```

<col> ::= a column not of type ARRAY, BLOB or COMPUTED BY

UNIQUE indices now allow NULLs

Changed in: 1.5

Description

In compliance with the SQL-99 standard, [NULLs](#) – even multiple – are now allowed in [columns](#) that have a [UNIQUE](#) index defined on them. For a full discussion, see [CREATE TABLE :: UNIQUE constraints now allow NULLs](#). As far as [NULLs](#) are concerned, the rules for unique indices are exactly the same as those for unique [keys](#).

Indexing on expressions

Added in: 2.0

Description

Instead of one or more columns, you can now also specify a single COMPUTED BY [expression](#) in an index definition. Expression indices will be used in appropriate queries, provided that the expression in the [WHERE](#), [ORDER BY](#) or [GROUP BY](#) clause exactly matches the expression in the index definition. Multi-segment expression indices are not supported, but the expression itself may involve multiple columns.

Examples

```
create index ix_upname on persons computed by (upper(name));
commit;
```

```
-- the following queries will use ix_upname:
```

```
select * from persons order by upper(name);
select * from persons where upper(name) starting with 'VAN';
delete from persons where upper(name) = 'BROWN';
delete from persons where upper(name) = 'BROWN' and age > 65;

create descending index ix_events_yt
  on MyEvents
  computed by (extract(year from StartDate) || Town);
commit;

-- the following query will use ix_events_yt:
select * from MyEvents
  order by extract(year from StartDate) || Town desc;
```

[back to top of page](#)

Maximum index key length increased

Changed in: 2.0

Description

The maximum length of index keys, which used to be fixed at 252 bytes, is now equal to 1/4 of the [page size](#), i.e. varying from 256 to 4096. The maximum indexable string length in bytes is 9 less than the key length. The table below shows the indexable string lengths in characters for the various page sizes and [character sets](#).

Table 5.1. Maximum indexable (VAR)CHAR length

Page size	Maximum indexable string length per charset type			
	1 byte/char	2 bytes/char	bytes/char	4 bytes/char
1024	247	123	82	61
2048	503	251	167	125
4096	1015	507	338	253
8192	2039	1019	679	509
16384	4087	2043	1362	1021

[back to top of page](#)

Maximum number of indices per table increased

Changed in: 1.0.3, 1.5, 2.0

Description

The maximum number of 65 [indices](#) per [table](#) has been removed in Firebird 1.0.3, reintroduced at the higher level of 257 in Firebird 1.5, and removed once again in Firebird 2.0.

Although there is no longer a “hard” ceiling, the number of indices attainable in practice is still limited by the database [page size](#) and the number of [columns](#) per index, as shown in the table below.

Table 5.2. Max. indices per table, Firebird 2.0

Page size	Number of indices depending on column count		
	1 col	2 cols	3 cols
1024	50	35	27
2048	101	72	56
4096	203	145	113
8192	408	291	227
16384	818	584	454

Please be aware that under normal circumstances, even 50 indices is way too many and will drastically reduce mutation speeds. The maximum was raised to accommodate data-warehousing applications and the like, that perform lots of bulk operations with the indices temporarily inactivated.

For a full table also including Firebird versions 1.0–1.5, see the [Notes](#) at the end of the book.

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

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-09-sql-language-references:firebird2.5-language-reference-update:ddl-statements:index:create-index>

Last update: 2023/07/20 08:36

