

Data types and subtypes

BIGINT data type

Added in: 1.5

Description

BIGINT is the SQL99-compliant 64-bit signed integer type. It is available in Dialect 3 only.

BIGINT numbers range from -2^{63} .. $2^{63}-1$, or -9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807.

Example

```
create table WholeLottaRecords (  
  id bigint not null primary key,  
  description varchar(32)  
)
```

See also:

[BIGINT](#)

BLOB data type

Text BLOB support in functions and operators

Changed in: 2.1

Description

Text **BLOBs** of any length and [character set](#) (including multi-byte sets) are now supported by practically every internal text function and [operator](#). In a few cases there are limitations or bugs.

Level of support

- Full support for:
 - = (assignment);
 - =, <>, <div></div>, <div>#c3c3c3</div><div></div>, >, >= and synonyms (comparison);
 - || (concatenation);
 - [BETWEEN](#), [IS \[NOT\] DISTINCT FROM](#), [IN](#), [ANY | SOME](#) and [ALL](#).
- Partial support for [STARTING \[WITH\]](#), [LIKE](#) and [CONTAINING](#): an error is raised if the second argument is 32 KB or longer.
- [SELECT DISTINCT](#), [ORDER BY](#) and [GROUP BY](#) work on the **BLOB ID**, not the contents. This makes

them as good as useless, except that `SELECT DISTINCT` weeds out multiple `NULLS`, if present. `GROUP BY` behaves oddly in that it groups together equal rows if they are adjacent, but not if they are apart.

- Any issues with BLOBs in [internal functions](#) and [aggregate functions](#) are discussed in their respective sections.

Various enhancements

Changed in: 2.0

Description

In Firebird 2.0, several enhancements have been implemented for text BLOBs:

- DML `COLLATE` clauses are now supported.
- Equality comparisons can be performed on the full BLOB contents.
- Character set conversions are possible when assigning a `BLOB` to a `BLOB` or a string to a `BLOB`.

When defining binary BLOBs, the mnemonic `binary` can now be used instead of the integer `0`.

Examples

```
select NameBlob from MyTable
  where NameBlob collate pt_br = 'João'

create table MyPictures (
  id int not null primary key,
  title varchar(40),
  description varchar(200),
  picture blob sub_type binary
)
```

See also:

- [BLOB](#)
- [Blob filter](#)

[back to top of page](#)

New character sets

Added in: 1.0, 1.5, 2.0, 2.1

The following table lists the [character sets](#) added in Firebird.

Table 4.1. Character sets new in Firebird

Name	Max bytes/ch.	Languages	Added in
------	---------------	-----------	----------

CP943C	2	Japanese	2.1
DOS737	1	Greek	1.5
DOS775	1	Baltic	1.5
DOS858	1	= DOS850 plus € sign	1.5
DOS862	1	Hebrew	1.5
DOS864	1	Arabic	1.5
DOS866	1	Russian	1.5
DOS869	1	Modern Greek	1.5
GBK	2	Chinese	2.1
ISO8859_2	1	Latin-2, Central European	1.0
ISO8859_3	1	Latin-3, Southern European	1.5
ISO8859_4	1	Latin-4, Northern European	1.5
ISO8859_5	1	Cyrillic	1.5
ISO8859_6	1	Arabic	1.5
ISO8859_7	1	Greek	1.5
ISO8859_8	1	Hebrew	1.5
ISO8859_9	1	Latin-5, Turkish	1.5
ISO8859_13	1	Latin-7, Baltic Rim	1.5
KOI8R	1	Russian	2.0
KOI8U	1	Ukrainian	2.0
TIS620	1	Thai	2.1
UTF8 (*)	4	All	2.0
WIN1255	1	Hebrew	1.5
WIN1256	1	Arabic	1.5
WIN1257	1	Baltic	1.5
WIN1258	1	Vietnamese	2.0

(*) In Firebird 1.5, UTF8 is an alias for UNICODE_FSS. This character set has some inherent problems. In Firebird 2, UTF8 is a character set in its own right, without the drawbacks of UNICODE_FSS.

See also:

- [Character sets](#)
- [Default character set](#)
- [SET NAMES](#)
- [Firebird 2.0 Language Reference Update: Character set NONE](#)
- [Firebird 2.1 Release Notes: International language support \(INTL\)](#)
- [Firebird 2.1 Release Notes: Appendix B: International character sets](#)
- [Overview of the main character sets in Firebird](#)
- [Character sets and Unicode in Firebird](#)
- [Convert your Firebird applications to Unicode](#)

Character set NONE handling changed

Changed in: 1.5.1

Description

Firebird 1.5.1 has improved the way [character set](#) `NONE` data are moved to and from [fields](#) or [variables](#) with another character set, resulting in fewer transliteration errors. For more details, see the [Note](#) at the end of the book.

[back to top of page](#)

New collations

Added in: 1.0, 1.5, 1.5.1, 2.0

The following table lists the collations added in Firebird. The **Details** column is based on what has been reported in the [Release Notes](#) and other documents. This information is almost certainly incomplete; some collations with an empty Details field may still be case insensitive (*ci*), accent insensitive (*ai*) or dictionary-sorted (*dic*).

Please note that the default – binary – collations for new character sets are not listed here, as doing so would add no meaningful information.

Table 4.2. Collations new in Firebird

Character set	Collation	Language	Details	Added in
CP943C:	CP943C_UNICODE	Japanese		2.1
GBK:	GBK_UNICODE	Chinese		2.1
ISO8859_1:	ES_ES_CI_AI	Spanish	<i>ci, ai</i>	2.0
	FR_FR_CI_AI	French	<i>ci, ai</i>	2.1
	PT_BR Brazilian	Portuguese	<i>ci, ai</i>	2.0
ISO8859_2:	CS_CZ	Czech		1.0
	ISO_HUN	Hungarian		1.5
	ISO_PLK	Polish		2.0
ISO8859_13:	LT_LT	Lithuanian		1.5.1
UTF8:	UCS_BASIC	All		2.0
	UNICODE	All	<i>dic</i>	2.0
	UNICODE_CI	All	<i>ci</i>	2.1
WIN1250:	BS_BA	Bosnian		2.0
	PXW_HUN	Hungarian	<i>ci</i>	1.0
	WIN_CZ	Czech	<i>ci</i>	2.0
	WIN_CZ_CI_AI	Czech	<i>ci, ai</i>	2.0
WIN1251:	WIN1251_UA	Ukrainian and Russian		1.5
WIN1252:	WIN_PTBR	Brazilian Portuguese	<i>ci, ai</i>	2.0
WIN1257:	WIN1257_EE	Estonian	<i>dic</i>	2.0
	WIN1257_LT	Lithuanian	<i>dic</i>	2.0
	WIN1257_LV	Latvian	<i>dic</i>	2.0
KOI8R:	KOI8R_RU	Russian	<i>dic</i>	2.0
KOI8U:	KOI8U_UA	Ukrainian	<i>dic</i>	2.0

TIS620:	TIS620_UNICODE	Thai		2.1
---------	----------------	------	--	-----

A note on the UTF8 collations: The UCS_BASIC collation sorts in Unicode code-point order: A, B, a, b, á... This is exactly the same as UTF8 with no collation specified. UCS_BASIC was added to comply with the SQL standard.

The UNICODE collation sorts using UCA (Unicode Collation Algorithm): a, A, á, b, B...

UNICODE_CI is truly case-insensitive. In a search for e.g. 'Apple', it will also find 'apple', 'APPLE' and 'aPPLe'.

[back to top of page](#)

Unicode collations for all character sets

Added in: 2.1

Firebird now comes with UNICODE collations for all the standard character sets. However, except for the ones listed in the new collations table in the previous section, these collations are not automatically available in your databases. Instead, they must be added with the CREATE COLLATION statement, like this:

```
create collation IS08859_1_UNICODE for IS08859_1
```

The new Unicode collations all have the name of their character set with _UNICODE added. (The built-in Unicode collations for UTF8 are the exception to the rule.) They are defined, along with the other collations, in the manifest file fbintl.conf in Firebird's intl subdirectory.

Collations may also be registered under a user-chosen name, e.g.:

```
create collation LAT_UNI for IS08859_1 from external ('IS08859_1_UNICODE')
```

See [CREATE COLLATION](#) for the full syntax.

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

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-09-sql-language-references:firebird2.1-language-reference:data-types-and-subtypes>

Last update: 2023/08/01 13:42

