Firebird 2 Cheat Sheet

Author: Lorenzo Alberton, https://www.alberton.info

Firebird SQL Cheat Sheet - details

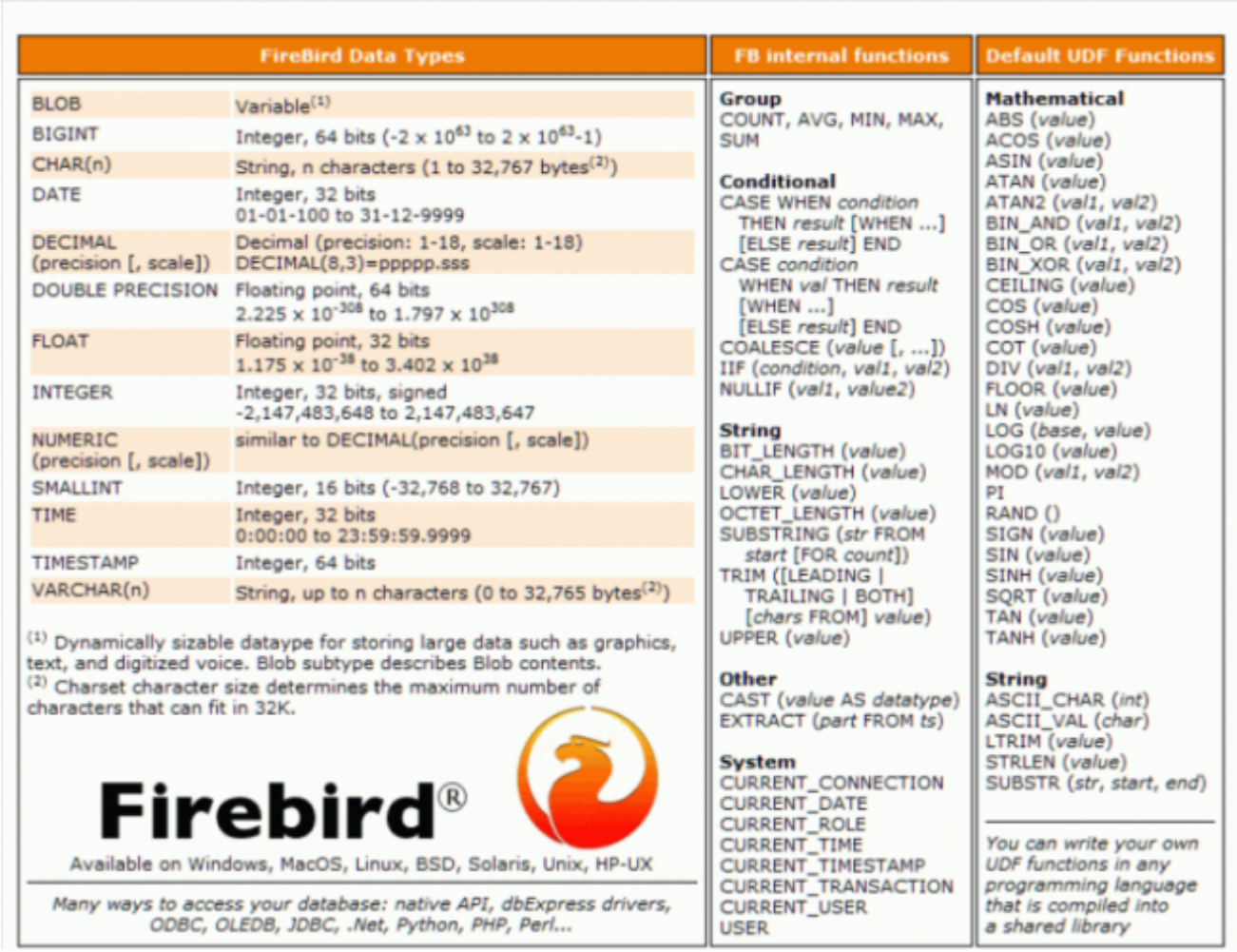
The cheat sheet is organized in 5 sections. The first section contains a list of the available datatypes, their description and the range of values that each of them supports.

The second section contains a list of the internal functions. The ones listed here are the Firebird 2 built-in functions; they're grouped by field of interest (aggregate, conditional, string functions).

The third section contains a list of the *Default UDF functions*. Firebird bundles a UDF library with some useful functions not included in the core. These functions are listed here, grouped by field of interest (mathematical and string functions).

The fourth section contains some useful queries, like the most useful queries to manage TRANSACTIONs, SAVEPOINTs, SEQUENCEs, a sample query with a LIMIT / OFFSET clause, and some queries against the System Tables to retrieve a list of the tables, fields, indices and constraints.

The last section holds a list of the PHP ibase_* functions. PHP has a Firebird/Interbase module and this is used by PHP developers to connect to, and query, a Firebird database. This section lists the functions available in PHP for connecting to and managing a Firebird database.



FireBird Data Types

BLOB	Variable[1]
BIGINT	Integer, 64 bits (-2 x 10^{63} to 2 x 10^{63}-1)
CHAR(n)	String, n characters (1 to 32,767 bytes[2])
DATE	Integer, 32 bits 01-01-100 to 31-12-9999
DECIMAL (precision [, scale])	Decimal (precision: 1-18, scale: 1-18) DECIMAL(8,3)=ppppp.sss
DOUBLE PRECISION	Floating point, 64 bits 2.225 x 10^{-308} to 1.797 x 10^{308}
FLOAT	Floating point, 32 bits 1.175 x 10^{-38} to 3.402 x 10^{38}
INTEGER	Integer, 32 bits, signed -2,147,483,648 to 2,147,483,647
NUMERIC (precision [, scale])	similar to DECIMAL(precision [, scale])
SMALLINT	Integer, 16 bits (-32,768 to 32,767)
TIME	Integer, 32 bits 0:00:00 to 23:59:59.9999
TIMESTAMP	Integer, 64 bits
VARCHAR(n)	String, up to n characters (0 to 32,765 bytes[2])

[1] Dynamically sizable datatype for storing large data such as graphics, text, and digitized voice. Blob subtype describes Blob contents.
[2] Charset character size determines the maximum number of characters that can fit in 32K.

Firebird®

Available on Windows, MacOS, Linux, BSD, Solaris, Unix, HP-UX

Many ways to access your database: native API, dbExpress drivers, ODBC, OLEDB, JDBC, .Net, Python, PHP, Perl...

FB internal functions

Group
COUNT, AVG, MIN, MAX, SUM

Conditional
CASE WHEN *condition* THEN *result* [WHEN ...] [ELSE *result*] END
CASE *condition* WHEN *val* THEN *result* [WHEN ...] [ELSE *result*] END
COALESCE (*value* [, ...])
IIF (*condition*, *val1*, *val2*)
NULLIF (*val1*, *value2*)

String
BIT_LENGTH (*value*)
CHAR_LENGTH (*value*)
LOWER (*value*)
OCTET_LENGTH (*value*)
SUBSTRING (*str* FROM *start* [FOR *count*])
TRIM ([LEADING | TRAILING | BOTH] [*chars* FROM] *value*)
UPPER (*value*)

Other
CAST (*value* AS *datatype*)
EXTRACT (*part* FROM *ts*)

System
CURRENT_CONNECTION
CURRENT_DATE
CURRENT_ROLE
CURRENT_TIME
CURRENT_TIMESTAMP
CURRENT_TRANSACTION
CURRENT_USER
USER

Default UDF Functions

Mathematical
ABS (*value*)
ACOS (*value*)
ASIN (*value*)
ATAN (*value*)
ATAN2 (*val1*, *val2*)
BIN_AND (*val1*, *val2*)
BIN_OR (*val1*, *val2*)
BIN_XOR (*val1*, *val2*)
CEILING (*value*)
COS (*value*)
COSH (*value*)
COT (*value*)
DIV (*val1*, *val2*)
FLOOR (*value*)
LN (*value*)
LOG (*base*, *value*)
LOG10 (*value*)
MOD (*val1*, *val2*)
PI
RAND ()
SIGN (*value*)
SIN (*value*)
SINH (*value*)
SQRT (*value*)
TAN (*value*)
TANH (*value*)

String
ASCII_CHAR (*int*)
ASCII_VAL (*char*)
LTRIM (*value*)
STRLEN (*value*)
SUBSTR (*str*, *start*, *end*)

You can write your own UDF functions in any programming language that is compiled into a shared library

Useful queries

-- Limit query
SELECT FIRST *limit* SKIP *offset* * FROM *table_name* ...
-- List tables
SELECT RDB$RELATION_NAME FROM RDB$RELATIONS WHERE RDB$SYSTEM_FLAG=0 AND RDB$VIEW_BLR IS NULL
-- List table fields
SELECT RDB$FIELD_NAME FROM RDB$RELATION_FIELDS WHERE RDB$RELATION_NAME='*table_name*'
-- List table constraints
SELECT RDB$INDEX_NAME FROM RDB$INDICES WHERE RDB$RELATION_NAME='*table_name*' AND (RDB$UNIQUE_FLAG IS NOT NULL OR RDB$FOREIGN_KEY IS NOT NULL)
-- List table indices
SELECT RDB$INDEX_NAME FROM RDB$INDICES WHERE RDB$RELATION_NAME='*table_name*' AND RDB$UNIQUE_FLAG IS NULL AND RDB$FOREIGN_KEY IS NULL
-- Handle sequences (NB: they're called GENERATORs in FB 1.x)
- CREATE|DROP SEQUENCE *name*;
- SET SEQUENCE *name* TO *value*;
- SELECT GEN_ID(*name*, *increment*) FROM RDB$DATABASE;
- SELECT NEXT VALUE FOR *name* FROM RDB$DATABASE;

-- Transactions
- SET TRANSACTION [READ WRITE | READ ONLY] [WAIT | NO WAIT] [ISOLATION LEVEL {SNAPSHOT [TABLE STABILITY] | READ COMMITTED [[NO] RECORD_VERSION]}]
- SAVEPOINT *name*;
- ROLLBACK [WORK] [TO [SAVEPOINT] *name*];
- COMMIT;

FireBird functions in PHP5

ibase_add_user	ibase_field_info
ibase_affected_rows	ibase_free_event_handler
ibase_backup	ibase_free_query
ibase_blob_add	ibase_free_result
ibase_blob_cancel	ibase_gen_id
ibase_blob_close	ibase_maintain_db
ibase_blob_create	ibase_modify_user
ibase_blob_echo	ibase_name_result
ibase_blob_get	ibase_num_fields
ibase_blob_import	ibase_num_params
ibase_blob_info	ibase_param_info
ibase_blob_open	ibase_pconnect
ibase_close	ibase_prepare
ibase_commit_ret	ibase_query
ibase_commit	ibase_restore
ibase_connect	ibase_rollback_ret
ibase_db_info	ibase_rollback
ibase_delete_user	ibase_server_info
ibase_drop_db	ibase_service_attach
ibase_errcode	ibase_service_detach
ibase_errmsg	ibase_set_event_handler
ibase_execute	ibase_timefmt
ibase_fetch_assoc	ibase_trans
ibase_fetch_object	ibase_wait_event
ibase_fetch_row	

Available for free from **www.alberton.info**

You can download the cheat sheet here or view at Lorenzo's website:

https://www.alberton.info/firebird_cheat_sheet.html

From:
https://www.ibexpert.com/docu/ - **IBExpert**

Permanent link:
https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-08-firebird-documentation:firebird-2-cheat-sheet

Last update: **2023/07/13 17:44**