

InterBase 6.0 DSQL syntax

YACC grammar notation

List of possible statements

top	: statement.
	statement ';'
	;
statement	: alter
	blob
	commit
	create
	declare
	delete
	drop
	grant
	insert
	invoke_procedure
	revoke
	rollback
	select
	set
	update
	;

GRANT statement

grant	: GRANT privileges ON prot_table_name TO user_grantee_list grant_option
	GRANT proc_privileges ON PROCEDURE simple_proc_name TO user_grantee_list grant_option
	GRANT privileges ON prot_table_name TO grantee_list
	GRANT proc_privileges ON PROCEDURE simple_proc_name TO grantee_list
	GRANT role_name_list TO role_grantee_list role_admin_option
	;
prot_table_name	: simple_table_name
	TABLE simple_table_name
	;
privileges	: ALL
	ALL PRIVILEGES
	privilege_list
	;
privilege_list	: privilege
	privilege_list ',' privilege
	;

proc_privileges	: EXECUTE
	;
privilege	: SELECT
	INSERT
	DELETE
	UPDATE column_parens_opt
	REFERENCES column_parens_opt
	;
grant_option	: WITH GRANT OPTION
	/* empty */
	;
role_admin_option	: WITH ADMIN OPTION
	/* empty */
	;
simple_proc_name	: symbol_procedure_name
	;

REVOKE statement

revoke	: REVOKE rev_grant_option privileges ON prot_table_name FROM user_grantee_list
	REVOKE rev_grant_option proc_privileges ON PROCEDURE simple_proc_name FROM user_grantee_list
	REVOKE privileges ON prot_table_name FROM user_grantee_list
	REVOKE proc_privileges ON PROCEDURE simple_proc_name FROM user_grantee_list
	REVOKE privileges ON prot_table_name FROM grantee_list
	REVOKE proc_privileges ON PROCEDURE simple_proc_name FROM grantee_list
	REVOKE role_name_list FROM role_grantee_list
	;
rev_grant_option	: GRANT OPTION FOR
grantee_list	: grantee
	grantee_list ',' grantee
	grantee_list ',' user_grantee
	user_grantee_list ',' grantee
	;
grantee	: PROCEDURE symbol_procedure_name
	TRIGGER symbol_trigger_name
	VIEW symbol_view_name
	;
user_grantee_list	: user_grantee
	user_grantee_list ',' user_grantee
	;
user_grantee	: symbol_user_name
	USER symbol_user_name
	GROUP symbol_user_name
	;

role_name_list	: role_name
	role_name_list ',' role_name
	;
role_name	: symbol_role_name
	;
role_grantee_list]]	: role_grantee
	role_grantee_list ',' role_grantee
	;
role_grantee	: symbol_user_name
	USER symbol_user_name
	;

DECLARE operations

declare	: DECLARE declare_clause
	;
declare_clause	: FILTER filter_decl_clause
	EXTERNAL FUNCTION udf_decl_clause
	;
udf_decl_clause	: symbol_UDF_name arg_desc_list1 RETURNS return_value1 ENTRY_POINT sql_string
	MODULE_NAME sql_string
	;
udf_data_type	: simple_type
	BLOB
	CSTRING '(' pos_short_integer ')' charset_clause
	;
arg_desc_list1	: /* empty */
	arg_desc_list
	'(' arg_desc_list ')'
	;
arg_desc_list	: arg_desc
	arg_desc_list ',' arg_desc
	;
arg_desc	: init_data_type udf_data_type
	;
return_value1	: return_value
	'(' return_value ')'
	;
return_value	: init_data_type udf_data_type
	init_data_type udf_data_type FREE_IT
	init_data_type udf_data_type BY VALUE
	PARAMETER pos_short_integer
	;
filter_decl_clause	: symbol_filter_name INPUT_TYPE blob_subtype OUTPUT_TYPE blob_subtype
	ENTRY_POINT sql_string MODULE_NAME sql_string
	;

CREATE metadata operations

create	: CREATE create_clause ;
create_clause	: EXCEPTION symbol_exception_name sql_string unique_opt order_direction INDEX symbol_index_name ON simple_table_name index_definition
	PROCEDURE procedure_clause
	TABLE table_clause
	TRIGGER def_trigger_clause
	VIEW view_clause
	GENERATOR generator_clause
	DATABASE db_clause
	DOMAIN domain_clause
	SHADOW shadow_clause
	ROLE role_clause
	;

CREATE INDEX

unique_opt	: UNIQUE
	/* empty */
	;
index_definition	: column_list
	column_parens
	;

CREATE SHADOW

shadow_clause	: pos_short_integer manual_auto conditiona sql_string first_file_length sec_shadow_files ;
manual_auto	: MANUAL
	AUTO
	/* empty */
	;
conditional	: /* empty */
	CONDITIONAL
	;
first_file_length	: /* empty */
	LENGTH equals long_integer page_noise
	;
sec_shadow_files	: /* empty */
	db_file_list
	;

db_file_list	: db_file
	db_file_list db_file
	;

CREATE DOMAIN

domain_clause	: column_def_name as_opt data_type begin_trigger domain_default_opt end_trigger domain_constraint_clause collate_clause ;
as_opt	: AS /* empty */ ;
domain_default_opt	: DEFAULT begin_trigger default_value /* empty */ ;
domain_constraint_clause	: /* empty */ domain_constraint_list ;
domain_constraint_list	: domain_constraint_def domain_constraint_list domain_constraint_def ;
domain_constraint_def	: domain_constraint ;
domain_constraint	: null_constraint domain_check_constraint ;
null_constraint	: NOT NULL ;
domain_check_constraint	: begin_trigger CHECK '(' search_condition ')' end_trigger ;

CREATE GENERATOR

generator_clause	: symbol_generator_name ;
------------------	------------------------------

CREATE ROLE

role_clause	: symbol_role_name
-------------	--------------------

CREATE DATABASE

db_clause	: db_name db_initial_desc1 db_rem_desc1 ;
equals	: /* empty */ = ;

db_name	: sql_string
	;
db_initial_desc1	: /* empty */
	db_initial_desc
	;
db_initial_desc	: db_initial_option
	db_initial_desc db_initial_option
	;
db_initial_option	: PAGE_SIZE equals pos_short_integer
	LENGTH equals long_integer page_noise
	USER sql_string
	PASSWORD sql_string
	SET NAMES sql_string
	;
db_rem_desc1	: /* empty */
	db_rem_desc
	;
db_rem_desc	: db_rem_option
	db_rem_desc db_rem_option
	;
db_rem_option	: db_file
	db_log
	db_log_option
	DEFAULT CHARACTER SET symbol_character_set_name
	;
db_log_option	: GROUP_COMMIT_WAIT equals long_integer
	CHECK_POINT_LEN equals long_integer
	NUM_LOG_BUFS equals pos_short_integer
	LOG_BUF_SIZE equals unsigned_short_integer
	;
db_log	: db_default_log_spec
	db_rem_log_spec
	;
db_rem_log_spec	: LOGFILE '(' logfiles ')' OVERFLOW logfile_desc
	LOGFILE BASENAME logfile_desc
	;
db_default_log_spec	: LOGFILE
	;
db_file	: file1 sql_string file_desc1
	;
logfiles	: logfile_desc
	logfiles ',' logfile_desc
	;
logfile_desc	: logfile_name logfile_attrs
	;

logfile_name	: sql_string
	;
logfile_attrs	: /* empty */
	logfile_attrs logfile_attr
logfile_attr	: SIZE equals long_integer
	;
file1 : FILE	;
file_desc1	: /* empty */
	file_desc
file_desc	: file_clause
	file_desc file_clause
file_clause	: STARTING file_clause_noise long_integer
	LENGTH equals long_integer page_noise
file_clause_noise	: /* empty */
	AT
	AT PAGE
	;
page_noise	: /* empty */
	PAGE
	PAGES
	;

CREATE TABLE

table_clause	: simple_table_name external_file '(' table_elements ')'
	;
external_file	: EXTERNAL FILE sql_string
	EXTERNAL sql_string
	/* empty */
table_elements	: table_element
	table_elements ',' table_element
table_element	: column_def
	table_constraint_definition
	;

Column definition

column_def	: column_def_name data_type_or_domain default_opt end_trigger column_constraint_clause collate_clause
	column_def_name non_array_type def_computed
	column_def_name def_computed
	;
def_computed	: computed_by '(' begin_trigger value end_trigger ')'
	;
computed_by	: COMPUTED BY
	COMPUTED
	;
data_type_or_domain	: data_type begin_trigger
	simple_column_name begin_string
	;
collate_clause	: COLLATE symbol_collation_name
	/* empty */
	;
column_def_name	: column_name
	;
simple_column_def_name	: simple_column_name
	;
data_type_descriptor	: init_data_type data_type
init_data_type	: /* empty */
default_opt	: DEFAULT default_value
	/* empty */
	;
default_value	: constant
	USER
	null_value
	datetime_value_expression
	;
column_constraint_clause	: /* empty */
	column_constraint_list
	;
column_constraint_list	: column_constraint_def
	column_constraint_list column_constraint_def
	;
column_constraint_def	: constraint_name_opt column_constraint
column_constraint	: NOT NULL
	REFERENCES simple_table_name column_parens_opt referential_trigger_action
	check_constraint
	UNIQUE
	PRIMARY KEY
	;

Table constraints

table_constraint_definition	: constraint_name_opt table_constraint ;
constraint_name_opt	: CONSTRAINT symbol_constraint_name ;
table_constraint	: unique_constraint primary_constraint referential_constraint check_constraint ;
unique_constraint	: UNIQUE column_parens ;
primary_constraint	: PRIMARY KEY column_parens ;
referential_constraint	: FOREIGN KEY column_parens REFERENCES simple_table_name column_parens_opt referential_trigger_action ;
check_constraint	: begin_trigger CHECK '(' search_condition ')' end_trigger ;
referential_trigger_action	: /* empty */ update_rule delete_rule delete_rule update_rule update_rule delete_rule ;
update_rule	: ON UPDATE referential_action ;
delete_rule	: ON DELETE referential_action ;
referential_action	: CASCADE SET DEFAULT SET NULL NO ACTION ;

PROCEDURE

procedure_clause	: symbol_procedure_name input_parameters output_parameters AS begin_string var_declaration_list full_proc_block end_trigger ;
alter_procedure_clause	: symbol_procedure_name input_parameters output_parameters AS begin_string var_declaration_list full_proc_block end_trigger ;
input_parameters	: '(' proc_parameters ')' /* empty */ ;

output_parameters	: RETURNS input_parameters
	/* empty */
	;
proc_parameters	: proc_parameter
	proc_parameters ',' proc_parameter
	;
proc_parameter	: simple_column_def_name non_array_type
	;
	;
var_declaration_list : var_declarations	/* empty */
	;
	;
var_declarations	: var_declaration
	var_declarations var_declaration
	;
var_declaration	: DECLARE VARIABLE column_def_name non_array_type ';
	;
	;
proc_block	: proc_statement
	full_proc_block
	;
full_proc_block	: BEGIN proc_statements END
	BEGIN proc_statements excp_statements END
	;
proc_statements	: proc_block
	proc_statements proc_block
	;
proc_statement	: assignment ';
	delete ';
	EXCEPTION symbol_exception_name ';
	exec_procedure
	for_select
	if_then_else
	insert ';
	POST_EVENT value ';
	singleton_select
	update ';
	while
	SUSPEND ';
	EXIT ';
	;
	;
exec_procedure	: EXECUTE PROCEDURE symbol_procedure_name proc_inputs proc_outputs ';
	;
	for_select : FOR select INTO variable_list cursor_def DO proc_block
	;

if_then_else	: IF '(' search_condition ')' THEN proc_block ELSE proc_block
	IF '(' search_condition ')' THEN proc_block
	;
singleton_select	: select INTO variable_list ';'
	;
variable	: ':' symbol_variable_name
	;
proc_inputs	: var_const_list
	'(' var_const_list ')'
	/* empty */
	;
proc_outputs	: RETURNING_VALUES variable_list
	RETURNING_VALUES '(' variable_list ')'
	/* empty */
	;
var_const_list	: variable
	constant
	column_name
	null_value
	var_const_list ',' variable
	var_const_list ',' constant
	var_const_list ',' column_name
	var_const_list ',' null_value
	;
variable_list	: variable
	column_name
	variable_list ',' column_name
	variable_list ',' variable
	;
while	: WHILE '(' search_condition ')' DO proc_block
	;
cursor_def	: AS CURSOR symbol_cursor_name
	/* empty */
	;
excp_statements	: excp_statement
	excp_statements excp_statement
	;
excp_statement	: WHEN errors DO proc_block
	;
errors	: err
	error ',' err
	;

err		: SQLCODE signed_short_integer
		GDSCODE symbol_gdscodename
		EXCEPTION symbol_exception_name
		ANY
		;

Direct EXECUTE PROCEDURE

invoke_procedure	: EXECUTE PROCEDURE symbol_procedure_name prc_inputs
	;
prc_inputs	: prm_const_list
	'(' prm_const_list ')'
	/* empty */
	;
prm_const_list	: parameter
	constant
	null_value
	prm_const_list ',' parameter
	prm_const_list ',' constant
	prm_const_list ',' null_value
	;

CREATE VIEW

view_clause	: symbol_view_name column_parens_opt AS begin_string select_view check_opt end_string
	;
select_view	: select_view_expr
	;
select_view_expr	: SELECT distinct_clause select_list from_view_clause where_clause group_clause having_clause plan_clause
	;
from_view_clause	: FROM from_view_list
	;
from_view_list	: view_table
	from_view_list ',' view_table
	;
view_table	: joined_view_table
	table_name
	;
joined_view_table	: view_table join_type JOIN view_table ON search_condition
	'(' joined_view_table ')'
	;

These rules will capture the input string for storage in metadata

begin_string	: /* empty */
	;
end_string	: /* empty */
	;
begin_trigger	: /* empty */
	;
end_trigger	: /* empty */
	;
check_opt	: WITH CHECK OPTION
	/* empty */
	;

CREATE TRIGGER

def_trigger_clause	: symbol_trigger_name FOR simple_table_name trigger_active trigger_type trigger_position begin_trigger trigger_action end_trigger ;
trigger_active	: ACTIVE
	INACTIVE
	/* empty */
	;
trigger_type	: BEFORE INSERT
	AFTER INSERT
	BEFORE UPDATE
	AFTER UPDATE
	BEFORE DELETE
	AFTER DELETE
	;
trigger_position	: POSITION nonneg_short_integer
	/* empty */
	;
trigger_action	: AS begin_trigger var_declaration_list full_proc_block ;

ALTER statement

alter	: ALTER alter_clause ;
alter_clause	: EXCEPTION symbol_exception_name sql_string
	TABLE simple_table_name alter_ops
	TRIGGER alter_trigger_clause
	PROCEDURE alter_procedure_clause
	DATABASE init_alter_db alter_db
	DOMAIN simple_column_name alter_domain_ops
	INDEX alter_index_clause ;

alter_domain_ops	: alter_domain_op
	alter_domain_ops alter_domain_op
	;
alter_domain_op	: SET begin_string default_opt end_trigger
	ADD CONSTRAINT domain_check_constraint
	ADD domain_check_constraint
	DROP DEFAULT
	DROP CONSTRAINT
	TO simple_column_name
	TYPE init_data_type non_array_type
	;
alter_ops	: alter_op
	alter_ops ',' alter_op
	;
alter_op	: DROP simple_column_name drop_behaviour
	DROP CONSTRAINT symbol_constraint_name
	ADD column_def
	ADD table_constraint_definition
	col_opt simple_column_name POSITION nonneg_short_integer
	col_opt simple_column_name TO simple_column_name
	col_opt alter_col_name TYPE alter_data_type_or_domain end_trigger
	;
col_opt	: ALTER
	ALTER COLUMN
	;
alter_data_type_or_domain	: non_array_type begin_trigger
	simple_column_name begin_string
alter_col_name	: simple_column_name
drop_behaviour	: RESTRICT
	CASCADE
	/* empty */
	;
alter_index_clause	: symbol_index_name ACTIVE
	symbol_index_name INACTIVE
	;

ALTER DATABASE

init_alter_db	: /* empty */
	;
alter_db	: db_alter_clause
	alter_db db_alter_clause
	;

db_alter_clause	: ADD db_file_list
	DROP LOGFILE
	SET db_log_option_list
	ADD db_log
	;
db_log_option_list	: db_log_option
	db_log_option_list ',' db_log_option
	;

ALTER TRIGGER

alter_trigger_clause	: symbol_trigger_name trigger_active new_trigger_type trigger_position begin_trigger new_trigger_action end_trigger ;
new_trigger_type	: trigger_type
	/* empty */
	;
new_trigger_action	: trigger_action
	/* empty */
	;

DROP metadata operations

drop	: DROP drop_clause ;
drop_clause	: EXCEPTION symbol_exception_name
	INDEX symbol_index_name
	PROCEDURE symbol_procedure_name
	TABLE symbol_table_name
	TRIGGER symbol_trigger_name
	VIEW symbol_view_name
	FILTER symbol_filter_name
	DOMAIN symbol_domain_name
	EXTERNAL FUNCTION symbol_UDF_name
	SHADOW pos_short_integer
	ROLE symbol_role_name
	;

These are the allowable datatypes

data_type	: non_array_type
	array_type
	;
non_array_type	: simple_type
	blob_type
	;

array_type	: non_charset_simple_type '[' array_spec ']'
	character_type '[' array_spec ']' charset_clause
	;
array_spec	: array_range
	array_spec ',' array_range
	;
array_range	: signed_long_integer
	signed_long_integer ':' signed_long_integer
	;
simple_type	: non_charset_simple_type
	character_type charset_clause
	;
non_charset_simple_type	: national_character_type
	numeric_type
	float_type
	integer_keyword
	SMALLINT
	DATE
	SQL DATE
	TIME
	TIMESTAMP
	;
integer_keyword	: INTEGER
	INT
	;
blob_type	: BLOB blob_subtype blob_segsize charset_clause
	BLOB '(' unsigned_short_integer ')'
	BLOB '(' unsigned_short_integer ',' signed_short_integer ')'
	BLOB '(' ',' signed_short_integer ')'
	;
blob_segsize	: SEGMENT SIZE unsigned_short_integer
	/* empty */
	;
blob_subtype	: SUB_TYPE signed_short_integer
	SUB_TYPE symbol_blob_subtype_name
	/* empty */
	;
charset_clause	: CHARACTER SET symbol_character_set_name
	/* empty */]
	;

Character type

national_character_type	: national_character_keyword '(' pos_short_integer ')'
	national_character_keyword
	national_character_keyword VARYING '(' pos_short_integer ')'
	;
character_type	: character_keyword '(' pos_short_integer ')'
	character_keyword
	varying_keyword '(' pos_short_integer ')'
	;
varying_keyword	: VARCHAR
CHARACTER VARYING	
CHAR VARYING	;
character_keyword	: CHARACTER
	CHAR
	;
national_character_keyword	: NCHAR
	NATIONAL CHARACTER
	NATIONAL CHAR
	;

Numeric type

numeric_type	: NUMERIC prec_scale
	decimal_keyword prec_scale
	;
ordinal	: pos_short_integer
	;
prec_scale	: /* empty */
	'(' signed_long_integer ')'
	'(' signed_long_integer ',' signed_long_integer ')'
	;
decimal_keyword	: DECIMAL
	DEC
	;

Floating point type

float_type	: FLOAT precision_opt
	LONG FLOAT precision_opt
	REAL
	DOUBLE PRECISION
	;
precision_opt	: '(' nonneg_short_integer ')'
	/* empty */
	;

SET statements

set	: set_transaction
	set_generator
	set_statistics
	;
set_generator	: SET GENERATOR symbol_generator_name TO signed_long_integer
	SET GENERATOR symbol_generator_name TO NUMBER64BIT
	SET GENERATOR symbol_generator_name TO '-' NUMBER64BIT
	;

Transaction statements

commit	: COMMIT optional_work optional_retain
	;
rollback	: ROLLBACK optional_work
	;
optional_work	: WORK
	/* empty */
	;
optional_retain	: RETAIN opt_snapshot
	/* empty */
	;
opt_snapshot	: SNAPSHOT
	/* empty */
	;
set_transaction	: SET TRANSACTION tran_opt_list_m
	;
tran_opt_list_m	: tran_opt_list
	/* empty */
	;
tran_opt_list	: tran_opt
	tran_opt_list tran_opt
	;
tran_opt	: access_mode
	lock_wait
	isolation_mode
	tbl_reserve_options
	;
access_mode	: READ ONLY
	READ WRITE
	;
lock_wait	: WAIT
	NO WAIT
	;

isolation_mode	: ISOLATION LEVEL iso_mode
	iso_mode
	;
iso_mode	: snap_shot
	READ UNCOMMITTED version_mode
	READ COMMITTED version_mode
	;
snap_shot	: SNAPSHOT
	SNAPSHOT TABLE
	SNAPSHOT TABLE STABILITY
	;
version_mode	: VERSION
	NO VERSION
	/* empty */
	;
tbl_reserve_options : RESERVING restr_list;	
lock_type	: SHARED
	PROTECTED
	/* empty */
	;
lock_mode	: READ
	WRITE
	;
restr_list	: restr_option
	restr_list ',' restr_option
	;
restr_option	: table_list table_lock
	;
table_lock	: FOR lock_type lock_mode
	/* empty */
	;
table_list	: simple_table_name
	table_list ',' simple_table_name
	;
set_statistics	: SET STATISTICS INDEX symbol_index_name

SELECT statement

select	: union_expr order_clause for_update_clause
	;
union_expr	: select_expr
	union_expr UNION select_expr
	union_expr UNION ALL select_expr
	;

order_clause	: ORDER BY order_list
	/* empty */
	;
order_list	: order_item
	order_list ',' order_item
	;
order_item	: column_name collate_clause order_direction
	ordinal collate_clause order_direction
	;
order_direction	: ASC
	DESC
	/* empty */
	;
for_update_clause	: FOR UPDATE for_update_list
	/* empty */
	;
for_update_list	: OF column_list
	/* empty */
	;

SELECT expression

select_expr	: SELECT distinct_clause select_list from_clause where_clause group_clause having_clause plan_clause
	;
distinct_clause	: DISTINCT
	all_noise
	;
select_list	: select_items
	'*'
	;
select_items	: select_item
	select_items ',' select_item
	;
select_item	: rhs
	rhs symbol_item_alias_name
	rhs AS symbol_item_alias_name
	;

FROM clause

from_clause	: FROM from_list
	;
from_list	: table_reference
	from_list ',' table_reference
	;

table_reference	: joined_table
	table_proc
	;
joined_table	: table_reference join_type JOIN table_reference ON search_condition
	'(' joined_table ')'
	;
table_proc	: symbol_procedure_name proc_table_inputs symbol_table_alias_name
	symbol_procedure_name proc_table_inputs
	;
proc_table_inputs	: '(' null_or_value_list ')'
	/* empty */
	;
null_or_value_list	: null_or_value
	null_or_value_list ',' null_or_value
	;
null_or_value	: null_value
	value
	;
table_name	: simple_table_name
	symbol_table_name symbol_table_alias_name
	;
simple_table_name	: symbol_table_name
	;
join_type	: INNER
	LEFT
	LEFT OUTER
	RIGHT
	RIGHT OUTER
	FULL
	FULL OUTER
	/* empty */
	;

Other clauses in the **SELECT** expression

group_clause	: GROUP BY grp_column_list
	/* empty */
	;
grp_column_list	: grp_column_elem
	grp_column_list ',' grp_column_elem
	;
grp_column_elem	: column_name
	column_name COLLATE symbol_collation_name
	;

having_clause	: HAVING search_condition
	/* empty */
	;
where_clause	: WHERE search_condition
	/* empty */
	;

PLAN clause to specify an access plan for a query

plan_clause	: PLAN plan_expression
	/* empty */
	;
plan_expression	: plan_type '(' plan_item_list ')'
	;
plan_type	: JOIN
	SORT MERGE
	MERGE
	SORT
	/* empty */
plan_item_list	;
	: plan_item
	plan_item ',' plan_item_list
plan_item	;
	: table_or_alias_list access_type
	plan_expression
table_or_alias_list	;
	: symbol_table_name
	symbol_table_name table_or_alias_list
access_type	;
	: NATURAL
	INDEX '(' index_list ')'
	ORDER symbol_index_name
index_list	;
	: symbol_index_name
	symbol_index_name ',' index_list
insert	;
	: INSERT INTO simple_table_name column_parens_opt VALUES '(' insert_value_list ')'
	INSERT INTO simple_table_name column_parens_opt select_expr
insert_value_list	;
	: rhs
	insert_value_list ',' rhs

INSERT statement

insert	: INSERT INTO simple_table_name column_parens_opt VALUES '(' insert_value_list ')'
	INSERT INTO simple_table_name column_parens_opt select_expr
	;
insert_value_list	: rhs
	insert_value_list ',' rhs
	;

DELETE statement

delete	: delete_searched
	delete_positioned
	;
delete_searched	: DELETE FROM table_name where_clause
	;
delete_positioned	: DELETE FROM table_name cursor_clause
	;
cursor_clause	: WHERE CURRENT OF symbol_cursor_name
	;

UPDATE statement

update	: update_searched
	update_positioned
	;
update_searched	: UPDATE table_name SET assignments where_clause
	;
update_positioned	: UPDATE table_name SET assignments cursor_clause
	;
assignments	: assignment
	assignments ',' assignment
	;
assignment	: column_name '=' rhs
	;
rhs	: value
	null_value
	;

BLOB get and put

blob	: READ BLOB simple_column_name FROM simple_table_name filter_clause segment_clause
	INSERT BLOB simple_column_name INTO simple_table_name filter_clause segment_clause
	;
filter_clause	: FILTER FROM blob_subtype_value TO blob_subtype_value
	FILTER TO blob_subtype_value
	/* empty */
	;
blob_subtype_value	: blob_subtype
	parameter
	;
blob_subtype	: signed_short_integer
	;

segment_clause	: MAX_SEGMENT segment_length
	/* empty */
	;
segment_length	: unsigned_short_integer
	parameter
	;

Column specifications

column_parens_opt	: column_parens
	/* empty */
	;
column_parens	: '(' column_list ')'
	;
column_list	: column_name
	column_list ',' column_name
	;
column_name	: simple_column_name
	symbol_table_alias_name '.' symbol_column_name
	symbol_table_alias_name '.' '*'
	;
simple_column_name	: symbol_column_name
	;

Boolean expressions

search_condition	: predicate
	search_condition OR search_condition
	search_condition AND search_condition
	NOT search_condition
	;
predicate	: comparison_predicate
	between_predicate
	like_predicate
	in_predicate
	null_predicate
	quantified_predicate
	exists_predicate
	containing_predicate
	starting_predicate
	unique_predicate
	'(' search_condition ')'
	;

Comparisons

comparison_predicate	: value '=' value
	value '<' value
	value '>' value
	value '>=' value
	value '<=' value
	value NOT_GTR value
	value NOT_LSS value
	value NEQ value
	;

Quantified comparisons

quantified_predicate	: value '=' ALL '(' column_select ')'
	value '<' ALL '(' column_select ')'
	value '>' ALL '(' column_select ')'
	value '>=' ALL '(' column_select ')'
	value '<=' ALL '(' column_select ')'
	value NOT_GTR ALL '(' column_select ')'
	value NOT_LSS ALL '(' column_select ')'
	value NEQ ALL '(' column_select ')'
	value '=' some '(' column_select ')'
	value '<' some '(' column_select ')'
	value '>' some '(' column_select ')'
	value '>=' some '(' column_select ')'
	value '<=' some '(' column_select ')'
	value NOT_GTR some '(' column_select ')'
	value NOT_LSS some '(' column_select ')'
	value NEQ some '(' column_select ')'
	;
some	: SOME
	ANY
	;

Other predicates

between_predicate	: value BETWEEN value AND value
	value NOT BETWEEN value AND value
	;
like_predicate	: value LIKE value
	value NOT LIKE value
	value LIKE value ESCAPE value
	value NOT LIKE value ESCAPE value
	;
in_predicate	: value IN scalar_set
	value NOT IN scalar_set
	;

containing_predicate	: value CONTAINING value
	value NOT CONTAINING value
	;
starting_predicate]]	: value STARTING value
	value NOT STARTING value
	value STARTING WITH value
	value NOT STARTING WITH value
	;
exists_predicate	: EXISTS '(' select_expr ')'
	;
unique_predicate	: SINGULAR '(' select_expr ')'
	;
null_predicate	: value IS NULL
	value IS NOT NULL
	;

Set values

scalar_set	: '(' constant_list ')'
	'(' column_select ')'
	;
column_select	: SELECT distinct_clause value from_clause where_clause group_clause having_clause plan_clause
	;
column_singleton	: SELECT distinct_clause value from_clause where_clause group_clause having_clause plan_clause
	;

Value types

value	: column_name	
	array_element	
	function	
	u_constant	
	parameter	
	variable	
	udf	
	'-' value	
	'+' value	
	value '+' value	
	value ' ' ' value	
	value COLLATE symbol_collation_name	
	value '-' value	
	value '*' value	
	value '/' value	
	'(' value ')'	
	'(' column_singleton ')'	
	USER	
	DB_KEY	
	symbol_table_alias_name '.' DB_KEY	
	VALUE	
	datetime_value_expression	
	;	
datetime_value_expression	: CURRENT_DATE	
	CURRENT_TIME	
	CURRENT_TIMESTAMP	
array_element	;	
	: column_name '[' value_list ']'	
value_list	: value	
	value_list ',' value	
	;	
constant	: u_constant	
	'-' u_numeric_constant	
	;	
u_numeric_constant	: NUMERIC	
	NUMBER	
	FLOAT	
	NUMBER64BIT	
	SCALEDINT	
	;	

u_constant	: u_numeric_constant	
	sql_string	
	DATE STRING	
	TIME STRING	
	TIMESTAMP STRING	
	;	
constant_list	: constant	
	parameter	
	current_user	
	constant_list ',' constant	
	constant_list ',' parameter	
	constant_list ',' current_user	
	;	
parameter	: '?'	
	;	
current_user : USER	;	
sql_string	: STRING	
	INTRODUCER STRING	
	;	
signed_short_integer	: nonneg_short_integer	
	'-' neg_short_integer	
	;	
nonneg_short_integer	: NUMBER	
	;	
neg_short_integer	: NUMBER	
	;	
pos_short_integer	: nonneg_short_integer	
	;	
unsigned_short_integer	: NUMBER	
	;	
signed_long_integer	: long_integer	
	'-' long_integer	
	;	
long_integer	: NUMBER	
	;	

function	: COUNT '(' '*' ')'	
	COUNT '(' all_noise value ')'	
	COUNT '(' DISTINCT value ')'	
	SUM '(' all_noise value ')'	
	SUM '(' DISTINCT value ')'	
	AVG '(' all_noise value ')'	
	AVG '(' DISTINCT value ')'	
	MINIMUM '(' all_noise value ')'	
	MINIMUM '(' DISTINCT value ')'	
	MAXIMUM '(' all_noise value ')'	
	MAXIMUM '(' DISTINCT value ')'	
	CAST '(' rhs AS data_type_descriptor ')'	
	UPPER '(' value ')'	
	GEN_ID '(' symbol_generator_name ',' value ')'	
	EXTRACT '(' timestamp_part FROM value ')'	
	;	
udf	: symbol_UDF_name '(' value_list ')'	
	symbol_UDF_name '(' ' ')	
	;	
timestamp_part	: YEAR	
	MONTH	
	DAY	
	HOUR	
	MINUTE	
	SECOND	
	WEEKDAY	
	YEARDAY	
all_noise	: ALL	
	/* empty */	
	;	
null_value	: NULL	
	;	

Performs special mapping of keywords into symbols

symbol_UDF_name	: SYMBOL
	;
symbol_blob_subtype_name	:SYMBOL
	;
symbol_character_set_name	:SYMBOL
	;
symbol_collation_name	:SYMBOL
	;
symbol_column_name	:SYMBOL
	;

symbol_constraint_name	:SYMBOL
	;
symbol_cursor_name	:SYMBOL
	;
symbol_domain_name	:SYMBOL
	;
symbol_exception_name	:SYMBOL
	;
symbol_filter_name	:SYMBOL
	;
symbol_gdscode_name	:SYMBOL
	;
symbol_generator_name	:SYMBOL
	;
symbol_index_name	:SYMBOL
	;
symbol_item_alias_name	:SYMBOL
	;
symbol_procedure_name	:SYMBOL
	;
symbol_role_name	:SYMBOL
	;
symbol_table_alias_name	:SYMBOL
	;
symbol_table_name	:SYMBOL
	;
symbol_trigger_name	:SYMBOL
	;
symbol_user_name	:SYMBOL
	;
symbol_variable_name	:SYMBOL
	;
symbol_view_name	:SYMBOL
	;

[back to top of page](#)

Any comments? Send an email to register@ibexpert.biz

COPYRIGHT © 2002-2022 HK-Software, IBExpert Ltd. All rights reserved.

All IBExpert brand and product names are trademarks or registered trademarks of IBExpert Ltd in Malta and other countries. InterBase, Delphi, CodeGear and C++Builder are trademarks or

registered trademarks of Embarcadero Technologies Inc. in the United States and other countries. Firebird is a registered trademark of the FirebirdSQL Foundation. Turbo Pascal is a registered trademark of Borland International, Inc. Sun, Java, JavaScript and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the United States and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through "The Open Group". Oracle is a registered trademark of Oracle Corporation in the United States and other countries. All Microsoft brand and product names are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. AS/400, DB2, IBM, Informix and iSeries are trademarks or registered trademarks of IBM Corporation in the United States and other countries. Linux is a registered trademark of Linux Torvalds. All other product names mentioned herein and throughout the entire web site are trademarks of their respective owners.

From:

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

Permanent link:

<https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-09-sql-language-references:interbase6.0-dsql-syntax>

Last update: **2023/08/04 16:09**

