

	Description	RPG IV			CLP		C		DDS		SQL		Value	Format		
		Type ¹	Length	Keyword	Type	Length	Type	Length	Type	Keyword	Type	Length				
CHARACTER	Character fixed length	A <i>char</i>	max 16.773.104		*CHAR	max 32767	_CHAR CHAR[n] _CHAR[n] _CHAR * _UCHAR _SCHAR			A		CHAR CHARACTER	max 32766 ²	default is blank ³		
	Character varying length ⁴	A <i>varchar</i>	max 16.773.100	varying	*CHAR ⁵	max 32767				A	VARLEN	VARCHAR CHAR VARYING CHARACTER VARYING	max 32740 ⁶	default length is zero		
	UCS-2 fixed length	C <i>ucs2</i>	max 8.386.552		--					G		GRAPHIC ⁷	max 16383 ⁸			
	UCS-2 varying length	C <i>varucs2</i>	max 8.386.550	varying	--					G	VARLEN	VARGRAPHIC ⁹ GRAPHIC VARYING	max 16370 ¹⁰	X'0000' to X'FFFF' default is blanks (X'0020')	bytes = length * 2	
	Graphic ¹¹ fixed length	G <i>graph</i>	max 8.386.552		--					J, E, O, G		GRAPHIC	max 16383 ¹²			
Graphic varying length	G <i>vargraph</i>	max 8.386.550	varying	--					J, E, O, G	VARLEN	VARGRAPHIC GRAPHIC VARYING	max 16370 ¹³	X'0000' to X'FFFF' default is X'4040'			
Indicator / Boolean	N <i>ind</i>	1			*LGL	1	_BOOL			A	1	BOOLEAN		'0' (*off), '1' (*on) default is '0' for DB2 (case insensitive): TRUE ('t', 'true', 'y', 'yes', 'on', '1', number <= 0) FALSE ('f', 'false', 'n', 'no', '0', '1', number = 0)		
BINARY ¹⁴	Binary fixed length	SQLTYPE(BINARY: length)	max 32766		--					--		BINARY	max 32766			
	Binary varying length	SQLTYPE(VARBINARY: length)	max 32740		--					--		VARBINARY BINARY VARYING	max 32740			
LOB	Character Large Object	SQLTYPE(CLOB: length)	max 16.773.100		--					--		CLOB ¹⁵ CHAR LARGE OBJECT CHARACTER LARGE OBJECT	max 2.147.483.647			
	Double Byte Large Object	SQLTYPE(DBCLOB: length)	max 8.386.550		--					--		DBCLOB	max 1.073.741.823			
	Binary Large Object	SQLTYPE(BLOB: length)	max 16.773.100		--					--		BLOB	max 2.147.483.647			
	LOB file reference	SQLTYPE(CLOB_FILE) SQLTYPE(DBCLOB_FILE) SQLTYPE(BLOB_FILE)			--					--						
	LOB locator	SQLTYPE(CLOB_LOCATOR) SQLTYPE(DBCLOB_LOCATOR) SQLTYPE(BLOB_LOCATOR)			--					--						
LOB XML	XML	SQLTYPE(XML_CLOB: length) SQLTYPE(XML_DBCLOB: length) SQLTYPE(XML_BLOB: length)	max 16.773.100 max 8.386.550 max 16.773.100		--					--		XML ¹⁶	max 2.147.483.647		default CCSID 1208 (UTF-8)	
	XML file reference	SQLTYPE(XML_CLOB_FILE) SQLTYPE(XML_DBCLOB_FILE) SQLTYPE(XML_BLOB_FILE)			--					--						
	XML locator	SQLTYPE(XML_LOCATOR)			--					--						
NUMERIC ¹⁷	exact	Binary ¹⁸	B <i>bindec</i>	1 to 4 5 to 9		--				B		SMALLINT INTEGER		0 to 9.999 0 to 999.999.999	2 byte = integer 5, 0 4 byte = integer 10, 0	
		Integer ¹⁹	I <i>int</i>	3, 0 5, 0 ²⁰ 10, 0 20, 0		*INT	2 4	_INT2 (short int) _INT4 (int / long int)		--		SMALLINT INTEGER BIGINT ²¹		±2 ⁷ , -128 to +127 ±2 ¹⁵ , -32.768 to +32.767 ±2 ³² , -2.147.483.648 a 2.147.483.647 ±2 ⁶⁴ , -9.223.372.036.854.775.808 to 9.223.372.036.854.775.807	3 digits, 1 byte 5 digits, 2 bytes 10 digits, 4 bytes 20 digits, 8 bytes	
		Unsigned	U <i>uns</i>	3, 0 5, 0 10, 0 20, 0		*UINT	2 4	_UINT2 (short unsigned int) _UINT4 (unsigned int / long unsigned int)		--				2 ⁸ , 0 to 255 2 ¹⁶ , 0 to 65.535 2 ³² , 0 to 4.294.967.295	3 digits, 1 byte 5 digits, 2 bytes 10 digits, 4 bytes	
		Packed decimal	P <i>packed</i> ²²	max 63, 63		*DEC	max 15, 9	decimal(n, p)		P		DEC DECIMAL	max 63, 63		2 ⁶⁴ , 0 to 184.467.440.737.095.851.615 (-10 ⁶³) + 1 to (10 ⁶³) - 1 X'F' for positive numbers X'D' for negative numbers	digits = 2n - 1 ²³ max length: 32 bytes
		Zoned decimal	S <i>zoned</i> ²⁴	max 63, 63		--				S		NUM NUMERIC	max 63, 63		X'F' for positive numbers X'D' for negative numbers	
	approximated	Float ²⁵	F <i>float</i>	4 8		--		_FLOAT4 _FLOAT8			F		REAL FLOAT(n) DOUBLE FLOAT ²⁶ FLOAT(n) DECFLOAT(n) ²⁷	n = 1 to 24 n = 25 to 53 n = 16 n = 34	-3.4-10 ³⁸ to -1.18-10 ³⁸ 0.0-10 ³⁸ to +3.4-10 ³⁸ -1.8-10 ³⁰⁸ to -2.23-10 ³⁰⁸ , 0.0-10 ⁰ , +2.23-10 ³⁰⁸ to +1.8-10 ³⁰⁸	8 digits, 4 bytes 16 digits, 8 bytes
		Complex8	D5	real 4F imaginary 4F		-		struct _COMPLEX8								
		Complex16	D5	real 8F imaginary 8F		-		struct _COMPLEX16								
DATE/TIME ²⁸	Date	D <i>date</i>	6 8 10	datfmt(*JUL) ²⁹ datfmt(*YMD) datfmt(*DMY) datfmt(*MDY) datfmt(*ISO) ³⁰ datfmt(*EUR) datfmt(*USA) datfmt(*JIS)	--				L	DATFMT DATSEP	DATE ³¹		40/001 to 39/365, def. 40/001 40/01/01 to 39/12/31 def. 40/01/01 01/01/40 to 31/12/39 def. 01/01/40 01/01/40 to 12/31/39 def. 01/01/40 01/01/0001 to 31/12/9999 def. 01/01/0001 01.01.0001 to 31.12.9999 def. 01.01.0001 01/01/0001 to 12/31/9999 def. 01/01/0001 0001-01-01 to 9999-12-31 def. 0001-01-01	2 digits yy/ddd 2 digits yy/mm/dd 2 digits dd/mm/yy 2 digits mm/dd/yy 4 digits yyyy-mm-dd 4 digits dd.mm.yyyy 4 digits mm/dd/yyyy 4 digits yyyy-mm-dd		
	Time	T <i>time</i>	8	timfmt(*HMS) ³² timfmt(*ISO) timfmt(*USA) timfmt(*EUR) timfmt(*JIS)	--				T	TIMFMT TIMSEP	TIME ³³		00:00:00 to 24:00:00, def. 00:00:00 00:00:00 to 24:00:00, def. 00:00:00 00:00 AM to 12:00 AM, def. 00:00 AM 00:00:00 to 24:00:00, def. 00:00:00 00:00:00 to 24:00:00, def. 00:00:00	hh:mm:ss hh:mm:ss hh:mm AM or hh:mm PM hh:mm:ss hh:mm:ss		
	Timestamp	Z <i>timestamp</i>	26-32 ³⁴	*ISO	--				Z		TIMESTAMP ³⁵		0001-01-01-00.00.00.000000 to 9999-12-31-24.00.00.000000 def. 0001-01-01-00.00.00.000000	YYYY-MM-DD- hh.mm.ss.mmmmm ³⁶		
POINTER	Pointer	* <i>pointer</i>	16 byte		*PTR		* _POINTER		--					default *NULL		
	Proc. pointer	* <i>pointer(*proc)</i>	16 byte	procptr	-		pointer to function		--					default *NULL		
	Data structure		max 16.773.104		--		struct		--		CHAR ³⁷			max element in array DS: 16.773.104		
	Java Object	O <i>object</i>		CLASS(*JAVA: 'string') EXTPROC(*JAVA: 'string') *CONSTRUCTOR	--									default *NULL		
	Datalink	-			--				--		DATALINK	max 32718 254 comm.				
	Row identifier ³⁸	SQLTYPE(ROWID)			--				--		ROWID	40		No CCSID conversion		
	ResultSet locator	SQLTYPE(RESULT_SET_LOCATOR)			--				--							
	Feedback token (fc)	DS	sev 5U msgno 5U flags 1A facid 3A isi 10U		--		_FEEDBACK		--							
	User-defined types (distinct type, array type)	--			--				--		CREATE TYPE					

¹ In *brown italic font* the free-form syntax
² max 32765 if null capable
³ Blank is x'40' in EBCDIC and x'20' in ASCII o UTF-8
⁴ data portion address = %addr(fieldname:*data). Fixed form operations MOVE, MOVE, CAT, SUBST and XLATE **do not** change the length of variable-length result fields. You can set the length of a variable-length field yourself using the %LEN built-in function on the left-hand-side of an EVAL operation.
⁵ %BIN(var 1 2) = length of variable, %SST(var 3 n) = value
⁶ max 32739 if null capable
⁷ with CCSID 13488 or 1200 (SQLTYPE(468))
⁸ max 16382 if null capable
⁹ with CCSID 13488 or 1200 (SQLTYPE(468))
¹⁰ max 16369 if null capable
¹¹ A graphic string is a sequence of double-byte characters
¹² max 16382 if null capable
¹³ max 16369 if null capable
¹⁴ CCSID 65535
¹⁵ CLOB variables can be defined in all host languages except REXX, RPG/400 and Cobol/400
¹⁶ Must be a well-formed document
¹⁷ default value is zero
¹⁸ **We recommend using corresponding integer data type**
¹⁹ Suggested for index of arrays, or fields in file feedback areas, or parameters of ILE C procedures
²⁰ Data type integer(5:0) must be used for indicator variables in embedded SQL programs
²¹ Data type for identity columns. Big integer host variables can only be used in C, C++, ILE Cobol, ILE RPG
²² Packed format is the default internal format for numeric stand-alone fields
²³ If specified the keyword PACKEVEN the number of digits is 2(n-1)
²⁴ Zoned format is the default internal format for numeric data structure subfields
²⁵ Float variables are intended to represent "scientific" values. Float should not be used when you need to represent numbers exactly to a specific number of decimal places, such as monetary amounts. Float format should not be used when more than 16 digits of precision are needed. Default is 0.0-10³⁸. Values min and max are approximated.
²⁶ Default size is 53. Floating-point variables can be used in all host languages except RPG/400 and Cobol/400.
²⁷ Decimal floating-point support both exact and approximations representations of real numbers. It can also represent the following three special values: infinity, Quiet NaN, Signaling NaN. Decimal floating-point variables can only be used in C.
²⁸ A 0 after format Indicates NO separator (ex. *ISO00)
²⁹ Valid separator for 2 digits format: / - . , &
³⁰ The default format for date, time e timestamp fields is *ISO
³¹ The internal representation is a string of 4 bytes that contains an integer (scaliger number)
³² Valid separator: . , &
³³ The internal representation is a string of 3 bytes that contains two packed decimal digits.
³⁴ From 7.2 the length can be until 32
³⁵ The internal representation is a string of 3 of between 7 and 13 bytes
³⁶ The fractional part is facultative. The length could be from 1 to 12. The default is 6.
³⁷ SQLTYPE(452)
³⁸ It is a value that uniquely identifies a row in a table