文章
姚 鑫 · 六月 7, 2022 阅读大约需 9 分钟

第二章 数据类型（一）

第二章 数据类型（一）

指定 SQL 实体（如列）可以包含的数据类型。

描述

此处描述了以下主题：
- 支持的 DDL 数据类型及其类属性映射表
- 数据类型优先级用于从具有不同数据类型的数据值中选择最具包容性的数据类型
- 日期、时间、PosixTime 和时间戳数据类型
  - 使用SqlCategory和用户定义的标准
  - 对 1840 年 12 月 31 日之前的日期的可配置支持

- 支持字符串数据类型、列表数据类型和流数据类型
- 支持 ROWVERSION 数据类型
- IRIS® 数据平台 ODBC / JDBC 公开的数据类型
- 使用查询元数据方法和数据类型整数代码确定列的数据类型
- 创建用户定义的数据类型
- 处理未定义的数据类型
- 数据类型转换函数

数据类型指定列可以保存的值的种类。在使用 CREATE TABLE 或 ALTER TABLE 定义字段时指定数据类型。定义 SQL 字段时，可以指定下表（左列）中列出的 DDL 数据类型。当指定其中一种 DDL
数据类型时，它会映射到右侧列中列出的IRIS 数据类型类。在IRIS 中定义字段时，可以指定 DDL
数据类型或数据类型类。 DDL 数据类型名称不区分大小写。数据类型类名称区分大小写。 %Library
数据类型类可以通过全名（例如，%Library.String）或短名（%String）来指定。

它们映射到的 DDL 数据类型和数据类型类通常提供不同的参数和参数默认值。数据类型类通常提供比 DDL
数据类型更多的参数来定义允许的数据值。

要查看当前系统数据类型映射，请转到管理门户，选择系统管理、配置、SQL 和对象设置、系统 DDL 映射。

还可以定义其他用户数据类型。要创建或查看用户数据类型映射，请转到管理门户，选择系统管理、配置、SQL 和对象设置、用户 DDL 映射。

DDL 数据类型表

DDL Data Type	Corresponding IRIS Data Type Class
BIGINT	%Library.BigInt (MAXVAL=9223372036854775807, MINVAL=-9223372036854775807)如果 BIGINT 列可以同时包含 NULL 和极小的负数，需要重新定义索引空标记以支持标准索引排序规则。

DDL Data Type	Corresponding IRIS Data Type Class
BIGINT(%1)	%Library.BigInt %1 被忽略。相当于 BIGINT。提供 MySQL 兼容性。
BINARY	%Library.Binary(MAXLEN=1)
BINARY(%1)	%Library.Binary(MAXLEN=%1)
BINARY VARYING	%Library.Binary(MAXLEN=1)
BINARY VARYING(%1)	%Library.Binary(MAXLEN=%1)
BIT	%Library.Boolean .
CHAR	%Library.String(MAXLEN=1)
CHAR(%1)	%Library.String(MAXLEN=%1)
CHAR VARYING	%Library.String(MAXLEN=1)
CHAR VARYING(%1)	%Library.String(MAXLEN=%1)
CHARACTER	%Library.String(MAXLEN=1)
CHARACTER VARYING	%Library.String(MAXLEN=1)
CHARACTER VARYING(%1)	%Library.String(MAXLEN=%1)
CHARACTER(%1)	%Library.String(MAXLEN=%1)
DATE	%Library.Date
DATETIME	%Library.DateTime
DATETIME2	%Library.DateTime
DEC	%Library.Numeric MAXVAL=999999999999999, MINVAL=-999999999999999, SCALE=0.
DEC(%1)	%Library.Numeric 一个 64 位有符号整数。如果 %1 小于 19，则 MAXVAL 和 MINVAL 是 %1 位数。例如，DEC(8) MAXVAL=99999999，MINVAL=-99999999，SCALE=0 。 %1 的最大有意义值是 19； %1 值大于 19 不会产生错误，但默认为 19。如果 %1 为 19 或更大：MAXVAL=9223372036854775807，MINVAL=-9223372036854775808，SCALE=0。
DEC(%1,%2)	%Library.Numeric (MAXVAL=<¦'$$maxval^%apiSQL(%1,%2)'¦>, MINVAL=<¦'$$minval^%apiSQL(%1,%2)'¦>, SCALE=%2)
DECIMAL	%Library.Numeric MAXVAL=999999999999999, MINVAL=-999999999999999, SCALE=0.
DECIMAL(%1)	%Library.Numeric 一个 64 位有符号整数。如果 %1 小于 19，则 MAXVAL 和 MINVAL 是 %1 位数。例如，DECIMAL(8) MAXVAL=99999999，MINVAL=-99999999，SCALE=0。 %1 的最大有意义值是 19； %1 值大于 19 不会产生错误，但默认为 19。如果 %1 为 19 或更大：MAXVAL=9223372036854775807，MINVAL=-9223372036854775808，SCALE=0。
DECIMAL(%1,%2)	%Library.Numeric (MAXVAL=<¦'$$maxval^%apiSQL(%1,%2)'¦>, MINVAL=<¦'$$minval^%apiSQL(%1,%2)'¦>, SCALE=%2)
DOUBLE	%Library.Double 这是 IEEE 浮点标准。具有此数据类型的 SQL 列返回的默认精度为 20。
DOUBLE PRECISION	%Library.Double 这是 IEEE 浮点标准。具有此数据类型的 SQL 列返回的默认精度为 20。
FLOAT	已弃用 — %Library.Double 这是 IEEE 浮点标准。具有此数据类型的 SQL 列返回的默认精度为 20。 FLOAT(%1) 已弃用 — %Library.Double 这是 IEEE 浮点标准。具有此数据类型的 SQL 列返回的默认精度为 20。
IMAGE	%Stream.GlobalBinary
INT	%Library.Integer (MAXVAL=2147483647, MINVAL=-2147483648)
INT(%1)	%Library.Integer (MAXVAL=2147483647, MINVAL=-2147483648)。 %1 被忽略。相当于 INT。提供 MySQL 兼容性。
INTEGER	%Library.Integer (MAXVAL=2147483647,

DDL Data Type	Corresponding IRIS Data Type Class
	MINVAL=-2147483648)
LONG	%Stream.GlobalCharacter
LONG BINARY	%Stream.GlobalBinary
LONG RAW	%Stream.GlobalBinary
LONGTEXT	%Stream.GlobalCharacter 等效于 LONG。提供 MySQL 兼容性。
LONG VARCHAR	%Stream.GlobalCharacter
LONG VARCHAR(%1)	%Stream.GlobalCharacter The %1 is ignored.
LONGVARBINARY	%Stream.GlobalBinary
LONGVARBINARY(%1)	%Stream.GlobalBinary The %1 is ignored.
LONGVARCHAR	%Stream.GlobalCharacter
LONGVARCHAR(%1)	%Stream.GlobalCharacter The %1 is ignored.
MEDIUMINT	%Library.Integer(MAXVAL=8388607,MINVAL=-8388608) 提供 MySQL 兼容性。
MEDIUMINT(%1)	%Library.Integer(MAXVAL=8388607,MINVAL=-8388608) %1 被忽略。提供 MySQL 兼容性。
MEDIUMTEXT	%Stream.GlobalCharacter
MONEY	%Library.Currency(MAXVAL=922337203685477.5807, MINVAL=-922337203685477.5808, SCALE=4)
NATIONAL CHAR	%Library.String(MAXLEN=1)
NATIONAL CHAR(%1)	%Library.String(MAXLEN=%1)
NATIONAL CHAR VARYING	%Library.String(MAXLEN=1)
NATIONAL CHAR VARYING(%1)	%Library.String(MAXLEN=%1)
NATIONAL CHARACTER	%Library.String(MAXLEN=1)
NATIONAL CHARACTER(%1)	%Library.String(MAXLEN=%1)
NATIONAL CHARACTER VARYING	%Library.String(MAXLEN=1)
NATIONAL CHARACTER VARYING(%1)	%Library.String(MAXLEN=%1)
NATIONAL VARCHAR	%Library.String(MAXLEN=1)
NATIONAL VARCHAR(%1)	%Library.String(MAXLEN=%1)
NCHAR	%Library.String(MAXLEN=1)
NCHAR(%1)	%Library.String(MAXLEN=%1)
NTEXT	%Stream.GlobalCharacter
NUMBER	%Library.Numeric 一个 64 位有符号整数。 (MAXVAL=9223372036854775807, MINVAL=-9223372036854775808, SCALE=0)
NUMBER(%1)	%Library.Numeric 一个 64 位有符号整数。如果 %1 小于 19，则 MAXVAL 和 MINVAL 是 %1 位数。例如，NUMBER(8) MAXVAL=99999999，MINVAL=-99999999，SCALE=0。 %1 的最大有意义值是 19；%1 值大于 19 不会产生错误，但默认为 19。如果 %1 为 19 或更大：MAXVAL=9223372036854775807，MINVAL=-9223372036854775808，SCALE=0。
NUMBER(%1,%2)	%Library.Numeric (MAXVAL=<¦'$$maxval^%apiSQL(%1,%2)'¦>, MINVAL=<¦'$$minval^%apiSQL(%1,%2)'¦>, SCALE=%2)
NUMERIC	%Library.Numeric MAXVAL=999999999999999, MINVAL=-999999999999999, SCALE=0.
NUMERIC(%1)	%Library.Numeric 一个 64 位有符号整数。如果 %1 小于 19，则 MAXVAL 和 MINVAL 是 %1 位数。例如，NUMERIC(8) MAXVAL=99999999，MINVAL=-99999999，SCALE=0。 %1 的最大有意义值是 19；%1 值大于 19 不会产生错误，但默认为 19。如果 %1 为 19 或更大：MAXVAL=9223372036854775807，MINVAL=-9223372036854775808，SCALE=0。
NUMERIC(%1,%2)	%Library.Numeric (MAXVAL=<¦'$$maxval^%apiSQL(%1,%2)'¦>, MINVAL=<¦'$$minval^%apiSQL(%1,%2)'¦>, SCALE=%2)
NVARCHAR	%Library.String(MAXLEN=1)

DDL Data Type	Corresponding IRIS Data Type Class
NVARCHAR(%1)	%Library.String(MAXLEN=%1)
NVARCHAR(%1,%2)	%Library.String(MAXLEN=%1)
NVARCHAR(MAX)	%Stream.GlobalCharacter 等效于 LONGVARCHAR。提供 TSQL 兼容性。
POSIXTIME	%Library.PosixTime MAXVAL=1406323805406846975, MINVAL=-6979664624441081856, SCALE=0.
RAW(%1)	%Library.Binary(MAXLEN=%1) REAL 已弃用 — %Library.Double 这是 IEEE 浮点标准。具有此数据类型的 SQL 列返回的默认精度为 20。
ROWVERSION	%Library.RowVersion(MAXVAL=9223372036854775807, MINVAL=1) 系统分配的顺序整数。
SERIAL	%Library.Counter System-generated: (MAXVAL=9223372036854775807, MINVAL=1). User-supplied: (MAXVAL=9223372036854775807, MINVAL=-9223372036854775807)
SMALLDATETIME	%Library.DateTime MAXVAL= ’ 2079-06-06 23:59:59 ’ ; MINVAL= ’ 1900-01-01 00:00:00 ’)
SMALLINT	%Library.SmallInt (MAXVAL=32767, MINVAL=-32768)
SMALLINT(%1)	%Library.SmallInt %1 被忽略。相当于 SMALLINT。提供 MySQL 兼容性。
SMALLMONEY	%Library.Currency SCALE=4
SYSNAME	%Library.String(MAXLEN=128)
TEXT	%Stream.GlobalCharacter
TIME	%Library.Time
TIME(%1)	%Library.Time(精度=%1)。 PRECISION 是小数秒位数，一个介于 0 到 9 之间的整数值。
TIMESTAMP	%Library.PosixTime
TIMESTAMP2	%Library.TimeStamp
TINYINT	%Library.TinyInt (MAXVAL=127, MINVAL=-128)
TINYINT(%1)	%Library.TinyInt %1 被忽略。相当于 TINYINT。提供 MySQL 兼容性。
UNIQUEIDENTIFIER)	%Library.UniqueIdentifier
VARBINARY)	%Library.Binary(MAXLEN=1)
VARBINARY(%1))	%Library.Binary(MAXLEN=%1)
VARCHAR)	%Library.String(MAXLEN=1)
VARCHAR(%1))	%Library.String(MAXLEN=%1)
VARCHAR(%1,%2))	%Library.String(MAXLEN=%1)
VARCHAR2(%1))	%Library.String(MAXLEN=%1)
VARCHAR(MAX))	%Stream.GlobalCharacter 等效于 LONGVARCHAR。仅提供 TSQL 兼容性。

重要提示：上面显示的每个 DDL 或 IRIS 数据类型表达式实际上都是一个连续的字符串。这些字符串可能包含空格字符，但通常不包含任何类型的空格。为了便于阅读，此表中出现了一些空白。

指定 MAXLEN

- 无 MAXLEN：没有 MAXLEN
  值的字段可以取任意长度的值，最多为最大字符串长度。要定义最大长度的字符串字段，请指定 VARCHAR('')，这将创建数据类型为 %Library.String(MAXLEN="") 的属性。 VARCHAR() 创建数据类型为 %Library.String(MAXLEN=1) 的属性。要定义没有 MAXLEN 值的二进制字段，请指定 VARBINARY('')，这将创建数据类型为 %Library.Binary(MAXLEN="") 的属性。 VARBINARY() 创建数据类型为 %Library.Binary(MAXLEN=1) 的属性。
- 大 MAXLEN：**具有大 MAXLEN 值的字段仅分配实际数据值所需的空间。指定 %Library.String 数据类型时，指定的 MAXLEN 值不必与数据的实际大小密切对应。如果字段值为 “ ABC ”，仅使用磁盘、全局缓冲区和私有进程内存中的那么多空间。即使使用 MAXLEN=1000 声明该字段，私有进程内存也不会为该字段分配那么多空间。只为字段值的实际大小分配内存，而不管声明的长度如何。**

过大的 MAXLEN 值可能会影响 ODBC 应用程序。 ODBC 应用程序尝试根据来自服务器的元数据来决定所需字段的大小，因此应用程序可能会分配比实际需要更多的缓冲区空间。出于这个原因， 提供系统范围的默认 ODBC VARCHAR 最大长度 4096；此系统范围的默认设置可使用管理门户进行配置：从系统管理中选择配置，然后选择 SQL 和对象设置，然后选择 SQL。查看或设置 VARCHAR 选项的默认长度。要确定当前设置，请调用 $SYSTEM.SQL.CurrentSettings()。 ODBC 驱动程序从 TCP 缓冲区获取数据并将其转换为应用程序缓冲区，因此 MAXLEN 大小不会影响我们的 ODBC 客户端。

过大的 MAXLEN 值不应影响 JDBC 应用程序。 Java 和 .Net
没有应用程序分配缓冲区。客户端仅分配将数据保存为本机类型所需的内容。

精确度和范围

NUMERIC(6,2) 等数值数据类型具有两个整数值 (p,s) 精度和小数位数。这些映射到 ObjectScript %Library 类数据类型。在 SQL 数据类型中指定时，以下内容适用于 Windows 系统（最大值在其他系统上可能不同）：
- Precision 精度：0 到
19+s（含）之间的整数。该值确定最大和最小允许值。这通常是数字中的总位数；但是，其确切值由 %Library 类数据类型映射决定。最大整数值为 9223372036854775807。大于 19+s 的精度默认为 19+s。
- Scale：一个整数，指定允许的最大十进制（小数）位数。可以是正整数、0 或负整数。如果 s 大于或等于 p，则只允许小数，实际 p 值被忽略。允许的最大比例为 18，对应于 0.999999999999999999。大于 18 的比例默认为 18。

以下示例显示了精度和比例的不同组合的最大值：

```
ClassMethod PrecisionScale()
{
    for i = 0 : 1 : 6 {
        w "Max for (",i,",2)=",$$maxval^%apiSQL(i,2),!
    }
}
```

```
DHC-APP>d ##class(PHA.TEST.SQLFunction).PrecisionScale()
Max for (0,2)=.99
Max for (1,2)=.99
Max for (2,2)=.99
Max for (3,2)=9.99
Max for (4,2)=99.99
Max for (5,2)=999.99
Max for (6,2)=9999.99
```

#SQL #Caché

源 URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E7%AB%A0-%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B%EF%BC%88%E4%B8%80%EF%BC%89