

获取SQLServer元数据的几种方法计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022__E8_8E_B7_E5_8F_96SQLS_c98_644530.htm

元数据简介 元数据 (metadata)

最常见的定义为"有关数据的结构数据"，或者再简单一点就是"关于数据的信息"，日常生活中的图例、图书馆目录卡和名片等都可以看作是元数据。在关系型数据库管理系统 (DBMS) 中，元数据描述了数据的结构和意义。比如在管理、维护 SQL Server 或者是开发数据库应用程序的时候，我们经常要获取一些涉及到数据库架构的信息：某个数据库中的表和视图的个数以及名称，某个表或者视图中列的个数以及每一列的名称、数据类型、长度、精度、描述等，某个表上定义的约束，某个表上定义的索引以及主键/外键的信息。下面我们将介绍几种获取元数据的方法。

获取元数据 使用系统存储过程与系统函数访问元数据 获取元数据最常用的方法是使用 SQL Server 提供的系统存储过程与系统函数。系统存储过程与系统函数在系统表和元数据之间提供了一个抽象层，使得我们不用直接查询系统表就能获得当前数据库对象的元数据。常用的与元数据有关的系统存储过程有以下一些：

- 系统存储过程 `sp_columns` 返回指定表或视图的列的详细信息。
- `sp_databases` 返回当前服务器上的所有数据库的基本信息。
- `sp_fkeys` 若参数为带有主键的表，则返回包含指向该表的外键的所有表，若参数为带有外键的表名，则返回所有同过主键/外键关系与该外键相关联的所有表。
- `sp_pkeys` 返回指定表的主键信息。
- `sp_server_info` 返回当前服务器的各种特性及其对应取值。
- `sp_proc_columns` 返回指定存储过程的输入、输出

参数的信息。 `sp_statistics` 返回指定的表或索引视图上的所有索引以及统计的信息。 `sp_stored_procedures` 返回当前数据库的存储过程列表，包含系统存储过程。 `sp_tables` 返回当前数据库的所有表和视图，包含系统表。常用的与元数据有关的系统函数有以下一些：系统函数 `DESCRIPTION` 返回有关列或过程参数的信息，如是否允许空值，是否为计算列等。 `COL_LENGTH` 返回指定数据库的指定属性值，如是否处于只读模式等。 `DATABASEPROPERTYEX` 返回指定数据库的指定选项或属性的当前设置，如数据库的状态、恢复模型等。 `OBJECT_ID` 返回指定数据库对象名的标识号。 `OBJECT_NAME` 返回指定数据库对象标识号的对象名。 `OBJECTPROPERTY` 返回指定数据库对象标识号的有关信息，如是否为表，是否为约束等。 `fn_listextendedproperty` 返回数据库对象的扩展属性值，如对象描述、格式规则、输入掩码等。由于我们无法直接利用到存储过程与函数的返回结果，因此只有在我们关心的只是查询的结果，而不需要进一步利用这些结果的时候，我们会使用系统存储过程与系统函数来查询元数据。例如，如果要获得当前服务器上所有数据库的基本信息，我们可以在查询分析器里面运行：`EXEC sp_databases` `GO` 在返回结果中我们可以看到数据库的名称、大小及备注等信息。但是如果我们要引用这部分信息，或者存储这部分信息以供后面使用，那么我们必须借助中间表来完成这个操作：`CREATE TABLE #sp_result (DATABASE_NAME sysname, DATABASE_SIZE int, REMARKS varchar(254) NULL)` `GO` `INSERT INTO #sp_result EXEC (sp_databases)` `GO` 使用信息架构视图访问元数据 信息架构视图基于 SQL-92 标准中针对

架构视图的定义，这些视图独立于系统表，提供了关于 SQL Server 元数据的内部视图。信息架构视图的最大优点是，即使我们对系统表进行了重要的修改，应用程序也可以正常地使用这些视图进行访问。因此对于应用程序来说，只要是符合 SQL-92 标准的数据库系统，使用信息架构视图总是可以正常工作的。常用的信息架构视图有以下一些：信息架构视图描述 INFORMATION_SCHEMA.CHECK_CONSTRAINTS 返回有关列或过程参数的信息，如是否允许空值，是否为计算列等。INFORMATION_SCHEMA.COLUMNS 返回当前数据库中当前用户可以访问的所有列及其基本信息。

INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE 返回当前数据库中定义了约束的所有列及其约束名。

INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE 返回当前数据库中定义了约束的所有表及其约束名。

INFORMATION_SCHEMA.KEY_COLUMN_USAGE 返回当前数据库中作为主键/外键约束的所有列。

INFORMATION_SCHEMA.SCHEMATA 返回当前用户具有权限的所有数据库及其基本信息。

INFORMATION_SCHEMA.TABLES 返回当前用户具有权限的当前数据库中的所有表或者视图及其基本信息。

INFORMATION_SCHEMA.VIEWS 返回当前数据库中的当前用户可以访问的视图及其所有者、定义等信息。由于这些信息架构都是以视图的方式存在的，因此我们可以很方便地获得并利用需要的信息。例如，我们要得到某个表有多少列，可以使用以下语句：

```
SELECT COUNT(*) FROM  
INFORMATION_SCHEMA.COLUMNS WHERE
```

TABLE_NAME=mytable 使用系统表访问元数据 虽然使用系统存储过程、系统函数与信息架构视图已经可以为我们提供了相当丰富的元数据信息，但是对于某些特殊的元数据信息，我们仍然需要直接对系统表进行查询。因为SQL Server 将所有数据库对象的信息均存放在系统表中，作为 SQL Server 的管理、开发人员，了解各个系统表的作用将有助于我们了解 SQL Server 的内在工作原理。SQL Server 的系统表非常多，其中最常用的与元数据查询有关的表有如下一些：系统表描述 syscolumns 存储每个表和视图中的每一列的信息以及存储过程中的每个参数的信息。 syscomments 存储包含每个视图、规则、默认值、触发器、CHECK 约束、DEFAULT 约束和存储过程的原始 SQL 文本语句。 sysconstraints 存储当前数据库中每一个约束的基本信息。 sysdatabases 存储当前服务器上每一个数据库的基本信息。 sysindexes 存储当前数据库中的每个索引的信息。 sysobjects 存储数据库内的每个对象(约束、默认值、日志、规则、存储过程等)的基本信息。 sysreferences 存储所有包括 FOREIGN KEY 约束的列。 systypes 存储系统提供的每种数据类型和用户定义数据类型的详细信息。将系统存储过程、系统函数、信息架构视图与系统表结合使用，可以方便地让我们获得所有需要的元数据信息。示例：1、获得当前数据库所有用户表的名称。SELECT OBJECT_NAME (id) FROM sysobjects WHERE xtype = U AND OBJECTPROPERTY (id, IsMSShipped) = 0 其中主要用到了系统表 sysobjects 以及其属性 xtype，还有就是用到了 OBJECTPROPERTY 系统函数来判断是不是安装 SQL Server 的过程中创建的对象。2、获得指定表上所有的索引名称 SELECT name FROM sysindexes

WHERE id = OBJECT_ID (mytable) AND indid 100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com